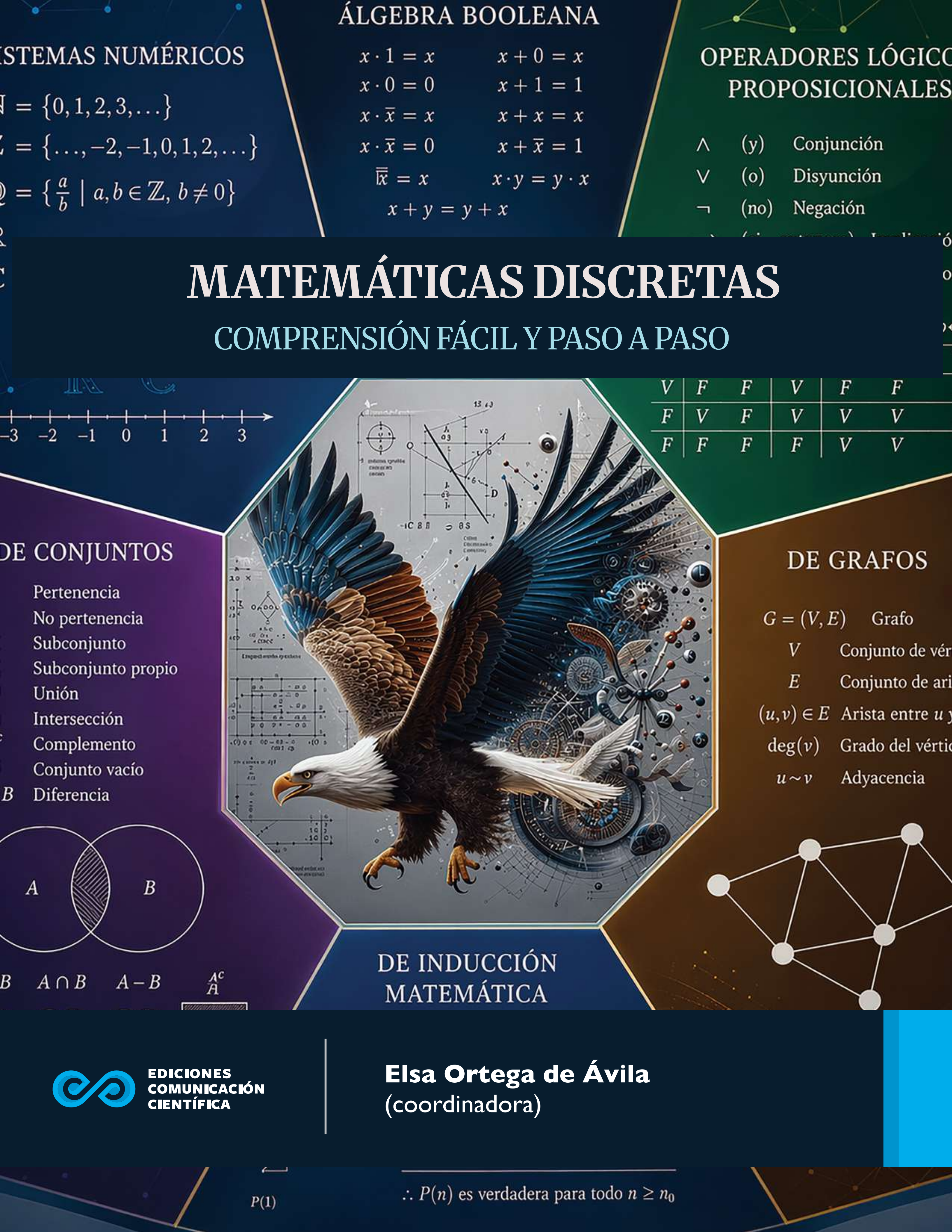
MATEMÁTICAS DISCRETAS

COMPRENSIÓN FÁCIL Y PASO A PASO



EDICIONES COMUNICACIÓN CIENTÍFICA

Elsa Ortega de Ávila
(coordinadora)

Matemáticas discretas: comprensión fácil y paso a paso

EDICIONES
COMUNICACIÓN
CIENTÍFICA

Ediciones Comunicación Científica se especializa en la publicación de conocimiento científico de calidad en español e inglés en soporte de libro impreso y digital en las áreas de humanidades, ciencias sociales y ciencias exactas. Guía su criterio de publicación cumpliendo con las prácticas internacionales: dictaminación de pares ciegos externos, autentificación antiplagio, comités y ética editorial, acceso abierto, métricas, campaña de promoción, distribución impresa y digital, transparencia editorial e indexación internacional.

Cada libro de la Colección Ciencia e Investigación es evaluado para su publicación mediante el sistema de dictaminación de pares externos y autentificación antiplagio. Invitamos a ver el proceso de dictaminación transparentado, así como la consulta del libro en Acceso Abierto.

www.comunicacion-cientifica.com

DOI.ORG/10.52501/cc.440

COMUNICACIÓN CIENTÍFICA PUBLICACIONES ARBITRADAS
HUMANIDADES, SOCIALES Y CIENCIAS

CC+I
COLECCIÓN CIENCIA e INVESTIGACIÓN

Matemáticas discretas: comprensión fácil y paso a paso

Elsa Ortega de Ávila
(coordinadora)

EDICIONES
COMUNICACIÓN
CIENTÍFICA

Matemáticas discretas : compresión fácil y paso a paso / coordinadora Elsa Ortega de Ávila.— Ciudad de México : Comunicación Científica, 2026. (Colección Ciencia e Investigación).

200 páginas : ilustraciones ; 23 x 16.5 centímetros

DOI: 1052501/cc.440

ISBN: 978-968-9738-82-4

1. Matemáticas discretas. 2. Sistemas numéricos. 3. Teoría de conjuntos. I. Ortega de Ávila, Elsa, coordinadora.

LC QA297.4 M38 DEWEY: 511.1 M38

Primera edición, 2026
Diseño de portada: Francisco Zeledón • Interiores: Guillermo Huerta

Ediciones Comunicación Científica, S. A. de C. V., 2026,
Av. Insurgentes Sur 1602, piso 4, suite 400,
Crédito Constructor, Benito Juárez, 03940, Ciudad de México,
Tel.: (52) 55-5696-6541 • Móvil: (52) 55-4516-2170
info@comunicacion-cientifica.com • www.comunicacion-cientifica.com
comunicacioncientificapublicaciones @ComunidadCient2

ISBN CC: 978-968-9738-82-4
DOI: 10.52501/cc.440

Esta obra fue dictaminada mediante el sistema de pares ciegos externos.

El proceso transparentado puede consultarse, así como el libro en acceso abierto,

en https://doi.org/10.52501/cc.440

Índice

Introducción

DOI: https://doi.org/10.52501/cc.440.00.01

El presente libro no es una investigación científica, sin embargo, sí surge de una problemática.

Problemática

En la actualidad, los estudiantes de la Licenciatura en Ingeniería en Sistemas Computacionales enfrentan serias dificultades en el proceso de aprendizaje de las Matemáticas Discretas, asignatura fundamental en la formación profesional debido a su aplicación directa en áreas como la algoritmia, estructuras de datos, lenguajes formales, teoría de la computación y diseño de *software*.

Una de las principales barreras detectadas radica en los materiales de estudio disponibles: la mayoría de los libros de matemáticas discretas son extensos, densos en contenido y fueron redactados con un enfoque excesivamente teórico. Si bien cumplen con el rigor académico, suelen resultar poco atractivos y desmotivadores para el alumnado actual; en su mayoría, los estudiantes muestran resistencia hacia textos voluminosos y de difícil comprensión.

Ante este panorama, se vuelve necesario diseñar un recurso didáctico alternativo que responda a las características de los estudiantes contemporáneos, privilegiando la claridad, la concisión y la practicidad. La elaboración de un libro sencillo, estructurado de manera accesible y con ejemplos aplicados al ámbito de la ingeniería busca favorecer la comprensión y motivación del alumnado, contribuyendo así con un aprendizaje más efectivo y significativo de las matemáticas discretas.

Justificación

La elaboración de un libro de matemáticas discretas que sea accesible y práctico se justifica en la necesidad de adaptar los procesos de enseñanza-aprendizaje a las características de las nuevas generaciones de estudiantes universitarios. En un contexto donde predominan el exceso de información y la inmediatez, es indispensable contar con materiales que simplifiquen la comprensión de los conceptos sin perder el rigor académico que exige la disciplina.

Este proyecto pretende ser un recurso didáctico innovador, diseñado con un lenguaje claro, ejemplos aplicados y una estructura amigable que permita al estudiante avanzar de manera gradual en el dominio de los temas. El libro busca equilibrar teoría y práctica, proporcionando ejercicios resueltos y actividades que favorezcan la aplicación de los conocimientos en situaciones propias de la ingeniería en sistemas computacionales. Además, se alinea con el propósito de fomentar en los alumnos el desarrollo del pensamiento lógico, analítico y crítico, competencias esenciales para su formación profesional y para enfrentar con éxito los retos del campo tecnológico. Con ello, no solo se espera facilitar la comprensión de las matemáticas discretas, sino también aumentar la motivación y el interés hacia su estudio, fortaleciendo así el desempeño académico y la preparación integral de los futuros ingenieros.

Finalmente, cabe señalar que no pretendemos abarcar en este libro la totalidad de los temas de matemáticas discretas que aparecen en la bibliografía internacional, ya que dichos contenidos suelen ser extensos y, en ocasiones, exceden los requerimientos curriculares de nuestra institución. En este sentido, el material se centra exclusivamente en los temas establecidos en el programa oficial del Tecnológico Nacional de México (TecNM), con el objetivo de ofrecer un recurso de estudio específico, pertinente y alineado con las competencias que demanda el plan de estudios de la carrera de Ingeniería en Sistemas Computacionales. De esta manera, se garantiza que los estudiantes cuenten con un apoyo concreto, focalizado y útil para su formación profesional.

Capítulo 1. Introducción general a las matemáticas discretas

Jesús Javier Arana Castillo[1]
Iván Alejandro González Torres[2]
José Francisco de la Rosa Ibarra[3]

DOI: https://doi.org/10.52501/cc.440.01

Resumen

Las matemáticas discretas constituyen un campo fundamental que se orienta al estudio de estructuras finitas, como los conjuntos, funciones, relaciones, grafos y la lógica matemática. Su importancia radica en que proporcionan bases teóricas esenciales para disciplinas como la informática, las ciencias exactas y la educación. En el ámbito informático, estas matemáticas son indispensables para el diseño y análisis de algoritmos, estructuras de datos, criptografía y lenguajes formales, ya que permiten comprender tanto la eficiencia de los programas como la seguridad de los sistemas. Asimismo, su aplicación en redes y procesos computacionales posiciona a las matemáticas discretas como un eje central del desarrollo tecnológico contemporáneo. En las ciencias exactas, las matemáticas discretas se utilizan para modelar y simular fenómenos. La teoría de grafos, por ejemplo, permite analizar redes moleculares en química y estudiar interacciones biológicas complejas, mientras que la probabilidad discreta contribuye a la estadística aplicada y a la optimización de recursos en investigaciones científicas. En el ámbito educativo, su enseñanza fortalece el pensamiento lógico y crítico, fomenta la abstracción y desarrolla competencias matemáticas esenciales, además de preparar a los estudiantes para el razonamiento computacional y su vinculación con áreas emergentes, como la inteligencia artificial y la robótica.

Palabras clave: *matemáticas discretas, aplicación educativa, estructuras finitas.*

---

[1] Maestro en Ciencias en Ingeniería Administrativa. Jefe del departamento de Ciencias Básicas en Tecnológico Nacional de México. Correo electrónico: Jesusj.ac@itz.edu.mx

[2] Maestro en Administración. Jefe del Departamento de Actividades Extraescolares en el Tecnológico Nacional de México.

[3] Maestro en Matemática Educativa. Docente en Tecnológico Nacional de México.

Importancia y aplicaciones en informática, ciencias exactas y educación

Las **matemáticas discretas** constituyen un área esencial en el estudio de estructuras finitas, tales como los conjuntos, funciones, relaciones, grafos y lógica matemática. Su relevancia radica en que proporcionan el fundamento teórico para la informática, las ciencias exactas y la educación.

En **informática**, las matemáticas discretas resultan indispensables para el diseño y análisis de algoritmos, estructuras de datos, criptografía, lenguajes formales y redes. Como señalan Rosen y Krithivasan (2018), la lógica proposicional, la teoría de grafos y la combinatoria permiten comprender la eficiencia de los programas y la seguridad de los sistemas computacionales.

En el ámbito de las **ciencias exactas**, las matemáticas discretas se aplican en problemas de modelación y simulación. Por ejemplo, la teoría de grafos se utiliza en química para analizar redes moleculares y en biología para estudiar interacciones entre genes y ecosistemas (Martínez, 2019). Asimismo, la probabilidad discreta contribuye a la estadística aplicada y la optimización de recursos en investigaciones científicas (Anderson, 2012).

Finalmente, en **educación**, las matemáticas discretas tienen un papel formativo al fortalecer el pensamiento lógico y crítico. De acuerdo con Aranda (2020), su enseñanza favorece el desarrollo de competencias matemáticas, fomenta la abstracción y prepara a los estudiantes para el razonamiento computacional. Además, su carácter interdisciplinario vincula la enseñanza con áreas emergentes como la inteligencia artificial y la robótica.

En síntesis, las matemáticas discretas son un pilar en la **ciencia y la educación**, pues articulan teoría y práctica, contribuyendo tanto a la innovación tecnológica como a la formación de profesionales capaces de resolver problemas complejos.

Breve historia de las matemáticas discretas

El desarrollo de las **matemáticas discretas** se remonta a la **Antigüedad**, cuando las civilizaciones ya empleaban conceptos de aritmética y combinatoria en problemas prácticos, como el conteo, la organización de calendarios y el diseño de juegos de azar (Katz, 2018). En la **Edad Media**, matemáticos árabes y europeos realizaron aportes al estudio de las secuencias numéricas y teoría de números, que más tarde se consolidarían como pilares de esta rama (Martínez, 2019).

En el **siglo XVII**, el trabajo de **Blaise Pascal** en el triángulo aritmético y el de **Pierre de Fermat** en probabilidad sentaron las bases de la combinatoria y el análisis discreto. Posteriormente, **Leonhard Euler** realizó aportaciones fundamentales en el siglo XVIII con la teoría de grafos —al resolver el famoso problema de los puentes de Königsberg— y con avances en la teoría de números (Biggs, 2002).

Durante el **siglo XX**, con el desarrollo de la computación, las matemáticas discretas adquirieron un papel central. Autores como **George Boole** (lógica matemática), **Kurt Gödel** (teoría de la incompletitud) y **Alan Turing** (fundamentos de la computación) consolidaron su aplicación en ciencias de la información y la informática moderna (Rosen y Krithivasan, 2018).

Hoy en día, las matemáticas discretas son consideradas un área interdisciplinaria que no solo sustenta la teoría de la computación, sino también la inteligencia artificial, la criptografía, la biología matemática y la educación en ciencias exactas.

Referencias

Anderson, D. R. (2012). *An introduction to discrete mathematics and its applications*. Cengage Learning.

Aranda, R. (2020). La enseñanza de las matemáticas discretas en la educación superior: un enfoque por competencias. *Revista Iberoamericana de Educación Superior, 11*(32), 45-62. https://doi.org/10.22201/iisue.20072872e.2020.32.588

Biggs, N. (2002). *Discrete mathematics*. Oxford University Press.

Katz, V. J. (2018). *A history of mathematics: An introduction* (4a ed.). Pearson.

Martínez, C. (2019). Aplicaciones de la teoría de grafos en ciencias naturales. *Revista Colombiana de Matemáticas Aplicadas, 10*(2), 77-89.

Rosen, K. H., y Krithivasan, K. (2018). *Discrete mathematics and its applications* (8a ed.). McGraw-Hill.

Capítulo 2. Sistemas numéricos

Elsa Ortega de Ávila[1]

Néstor Alejandro Carrillo López[2]

DOI: https://doi.org/10.52501/cc.440.02

Resumen

Los sistemas numéricos surgieron en las antiguas civilizaciones como respuesta a la necesidad de contar, medir y registrar información en actividades como el comercio, la agricultura y la construcción. A lo largo del tiempo, diversas culturas desarrollaron sistemas adaptados a sus contextos. Entre ellos destacan el sistema decimal, utilizado por los egipcios y perfeccionado en la India con la notación posicional; el sistema babilónico de base 60, aún vigente en la medición del tiempo y los ángulos; el sistema egipcio jeroglífico; y el sistema romano, limitado para cálculos complejos. Asimismo, el sistema maya introdujo el concepto de cero en una base vigesimal, mientras que el sistema binario, formalizado por Leibniz, se convirtió en la base de la informática moderna. El sistema indoarábigo, con su notación posicional y uso del cero, se consolidó como el más eficiente y universal.

En la actualidad, los sistemas numéricos se definen como estructuras que permiten representar cantidades y realizar operaciones, diferenciándose por su base y número de símbolos; los principales son el decimal (base 10), el binario (base 2), el octal (base 8) y el hexadecimal (base 16), estos últimos ampliamente utilizados en informática. Además, es posible convertir números entre distintos sistemas mediante métodos específicos, generalmente usando el sistema binario como intermediario. Estas conversiones implican procesos como la agrupación de dígitos y el uso de potencias o divisiones sucesivas. En conjunto, los sistemas numéricos constituyen un elemento esencial para las matemáticas, la ciencia y la tecnología.

[1] Doctora en Ciencias de la Educación. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0000-0003-0405-2978 ; correo electrónico: elsa.ortega@itz.edu.mx

[2] Maestro en Informática Administrativa. Encargado del laboratorio de Manufactura en Tecnológico Nacional de México.

Palabras clave: *sistemas numéricos, binario, octal, hexadecimal, decimal.*

Sistemas históricos y contemporáneos

Los sistemas numéricos tienen su origen en las antiguas civilizaciones, puesto que necesitaban contar, medir y registrar información para actividades prácticas como el comercio, la agricultura y la construcción. Con el crecimiento de las sociedades y la complejidad de sus actividades, surgió la necesidad de desarrollar métodos organizados y eficientes para representar números (Ifrah, 2000).

1. Sistema decimal (base 10). Civilización mesopotámica y egipcia

Uno de los primeros sistemas numéricos adoptados ampliamente fue el decimal, basado en diez unidades, posiblemente porque los humanos tenemos diez dedos, lo que facilita contar con ellos. Los antiguos egipcios y babilonios usaron este sistema para representar cantidades en sus registros (Menninger, 1992). Más adelante, el sistema decimal fue perfeccionado en India y transmitido al mundo occidental por matemáticos árabes, quienes desarrollaron la notación posicional y promovieron el uso del sistema decimal como base para los cálculos (Joseph, 2011).

2. Sistema babilónico (base 60)

Hace más de 4000 años, los babilonios de la antigua Mesopotamia desarrollaron un sistema numérico sexagesimal (base 60) que se sigue usando hoy en día para medir el tiempo (minutos y segundos) y los ángulos (Andonegui, 2007; Ifrah, 2000). Este sistema combinaba características de las bases 10 y 60, lo cual demuestra una sofisticación notable en el desarrollo de sistemas numéricos (Robson, 2008).

3. Sistema egipcio (base 10 con jeroglíficos)

Los egipcios empleaban un sistema decimal que utilizaba jeroglíficos específicos para representar valores como 1, 10, 100, y así sucesivamente, sumando estos símbolos para formar otros números (Clagett, 1989). Este sistema carecía de un cero y se usaba principalmente en registros administrativos y de contabilidad, en un formato adecuado para sus necesidades prácticas (Ifrah, 2000).

4. Sistema romano (números romanos)

Desarrollado en la antigua Roma, el sistema romano utiliza letras como “I”, “V”, “X”, “L”, “C”, “D” y “M” para representar valores y sigue un sistema aditivo y sustractivo. Este sistema carecía de un cero y era menos adecuado para cálculos matemáticos complejos, sin embargo, se mantuvo en uso durante la Edad Media en Europa (Menninger, 1992).

5. Sistema maya y mesoamericano (base 20)

Los mayas desarrollaron un sistema vigesimal (base 20) que incorpora el concepto de cero, uno de los primeros ejemplos en la historia de su uso, lo cual les permitió avances significativos en astronomía y en el desarrollo de un calendario preciso (Closs, 1986; Ifrah, 2000). Otros pueblos mesoamericanos utilizaron sistemas similares basados en la base 20, pero adaptados a sus necesidades culturales y prácticas.

6. Sistema binario (base 2). China y Europa moderna

Aunque el sistema binario moderno fue formalmente desarrollado en Europa por el matemático Gottfried Wilhelm Leibniz en el siglo XVII, sus orígenes se encuentran en el antiguo texto chino *I Ching*, que emplea secuencias binarias para representar conceptos (Gardner, 1983). El sistema binario es la base de la informática moderna, ya que permite representar datos y realizar operaciones en términos de 0 y 1 (Ifrah, 2000).

7. Sistema indoarábigo (números arábigos - base 10)

Desarrollado en la India antigua y difundido por matemáticos árabes, el sistema indoarábigo incluyó el concepto de cero y una notación posicional que facilitó cálculos complejos. Este sistema se consolidó como la base de la aritmética moderna y se usa en todo el mundo por su simplicidad y eficiencia (Joseph, 2011).

Los sistemas numéricos tienen diversos orígenes según las necesidades y contextos de cada civilización. Sin embargo, sistemas como el decimal y el indoarábigo se difundieron ampliamente, demostrando ser fundamentales en matemáticas, ciencia e ingeniería hasta hoy. En la página de la Universidad autónoma de Ciudad Juárez (Robledo et al., s.f.), comparten que:

Los sistemas de numeración permiten representar cantidades mediante símbolos organizados de acuerdo con reglas específicas de valor posicional. En estos sistemas, el valor de cada cifra depende tanto del símbolo utilizado como de la posición que ocupa dentro del número, la cual está determinada por una base numérica. De esta manera, el sistema decimal utiliza diez símbolos fundamentales (0–9), mientras que otros sistemas, como el binario o hexadecimal, emplean diferentes bases para representar información y realizar operaciones matemáticas (Rosen, 2012, p. 111).

De los sistemas mencionados, destacan y se siguen utilizando los sistemas de valor posicional, donde todos se refieren a una base específica de numeración (Andonegui, 2007).

Definiciones de los sistemas contemporáneos

Los sistemas numéricos son estructuras fundamentales en matemáticas que permiten representar cantidades y realizar operaciones aritméticas. De acuerdo con la Universidad Nacional a Distancia, los sistemas se diferencian por el número de símbolos que permiten, por ejemplo, el sistema binario consta de dos dígitos: el cero y el uno; el octal consta de ocho dígitos; el decimal, de diez dígitos; y el hexadecimal, de dieciséis dígitos (UNAD, s.f.). A continuación, se describen los principales sistemas numéricos.

Sistema decimal (base 10)

El sistema decimal es el más utilizado en la vida cotidiana. Está compuesto por diez dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9) y es un sistema posicional, lo que significa que el valor de cada dígito depende de su posición en el número (Katz, 2009).

Sistema binario (base 2)

El sistema binario es esencial en informática y electrónica. Solo utiliza dos dígitos (0 y 1), y también es un sistema posicional; cada posición representa una potencia de 2 (Knuth, 1997).

Sistema octal (base 8)

El sistema octal emplea ocho dígitos (0 a 7). Aunque no es tan común en el uso diario, ha sido importante en ciertas áreas de la informática, especialmente en la programación de sistemas antiguos (Tanenbaum, 2016).

Sistema hexadecimal (base 16)

El sistema hexadecimal utiliza 16 símbolos: los dígitos del 0 al 9 y las letras A, B, C, D, E y F, que representan los valores 10, 11, 12, 13, 14 y 15, respectivamente. Es muy utilizado en programación y desarrollo de *software* (Hennessy y Patterson, 2017).

Los sistemas octales y hexadecimales

Ejercicio 1. Hacer los números del 1 al 120 en el sistema octal.

1	2	3	4	5	6	7	10
11	12	13	14	15	16	17	20

Ejercicio 2. Completar los números del 1 al 120 de los siguientes hexadecimales.

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20
21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30

Conversiones entre los sistemas numéricos

Los números se pueden convertir de un sistema a otro, como las monedas de cambio; en seguida se muestra una tabla de los métodos para convertirlos. Por *default* se recomienda el sistema binario como sistema de cambio.

Binario	Octal	Decimal	Hexadecimal
Si tengo el número en binario.	Se agarran los números de 3 en 3 del número en binario y se convierte a octal en individual.	Se toma el número que le corresponde por su posición de la tabla de potencias binarias y se suma.	Se toman de a 4 números y se convierte a símbolo hexadecimal.
Se despliega de cada número octal a 3 números de binario.	**Si se tiene el número en octal, primero se convierte a binario.**	Se toma el número binario y se pone en la tabla para calcular su valor.	Se agarra el número en binario y se agarran de 4 en 4.
Se puede hacer por tabla o por división. Por división se divide entre 2 y se va guardando el residuo, ya que el residuo es el resultado, como se muestra en el método 7 de conversión.	Se agarra del binario de 3 en tres, como se muestra en el método binario a octal.	**Si se tiene el número en decimal, primero se convierte a binario.**	Se agarra del binario de 4 en cuatro como se muestra en el método binario a hexadecimal.
Se despliegan de cada número hexadecimal cuatro dígitos binarios, de acuerdo al símbolo que se tenga como se muestra en la tabla del método 3.	Del número binario se toman de 3 en 3, y se saca el número en octal como se muestra en la tabla del método 1.	Se convierte el binario a decimal, recomendable por método de tabla.	**Si se tiene el número en hexadecimal, primero se convierte a binario. Método inverso de binario a hexadecimal.**

Métodos de conversión

1. **De binario a octal.** Se toman 3 números de los binarios y se sustituyen por el número que les corresponde según la tabla siguiente:

En binario	En octal
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Por lo tanto, el número 010100 corresponderá al número 24 en octal.

2. **De binario a decimal**. Se toma un número de la tabla de potencia binaria y se pone en la tercera fila, de derecha a izquierda, como se muestra a continuación (en morado); luego se deben multiplicar los 1 por el número en rojo que le corresponde y, al final, hay que sumar todos los valores resultantes.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1
			1	0	1	0	1	0	1	0
			128	+	32	+	8	+	2	

El ejemplo de la tabla anterior corresponde al número 170 decimal.

3. **De binario a hexadecimal**. Se toman los dígitos binarios de 4 en 4 y se sustituye el número de acuerdo con la siguiente tabla:

En binario	En octal
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Por ejemplo, el número 00110101 en binario, al tomarlo de 4 en 4 nos da 0011, que equivale a 3, y 0101, que equivale a 5, por lo que el número resultante en hexadecimal es 35.

4. **De octal a binario**. Se toma en cuenta la tabla para cambiarlo. Por ejemplo, si tengo 734 se despliega cada número: 7 equivale 111, 3 equivale 011 y 4 equivale a 100. Por lo tanto queda así: 111 011 100.

En octal	En binario
1	001
2	010
3	011
4	100
5	101
6	110
7	111

5. **De octal a decimal**. Primero se pasa a binario y luego se aplica el método de binario a decimal, mediante la tabla que se muestra en el método 7.

6. **De octal a hexadecimal**. Primero se pasa a binario y luego de binario a hexadecimal, como se explica en el punto 3.

7. **De decimal a binario**. Se puede hacer de 2 formas: por tabla o por división.

a. Por tabla. Se coloca el número en la tercera línea, se va poniendo un 1 en los números que sumados den el número exacto. En la tabla contigua se pone un ejemplo del número 423:

Pasos:

1. Primero se elige un número menor o igual al número 423, en este caso sería el 256, entonces pongo un 1 debajo del 256; como no es igual al 423 sigo al paso 2.
2. Después le sumo el número inmediato a la derecha, si no me paso de 423 pongo el 1 y se suma el 256 al 128 = 384, como es menor a 423, vuelvo a repetir el paso 2.
3. Se repite el paso 2 hasta que la suma de los números que contengan 1 nos den 423.
4. En el ejemplo se sumaron 256 + 128 + 32 + 4 + 2 + 1 = 423

211	210	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2048	1024	512	256	128	64	32	16	8	4	2	1
			1	1	0	1	0	0	1	1	1
			256	128		32			4	2	1

b. **Por división** se divide el número entre la base, en este caso se divide entre 2, y se anota el residuo, si es 1 o es 0.

Número	Entre	Toca a:	Y sobra
423	entre 2	211	1
211	entre 2	105	1
105	entre 2	52	1
52	entre 2	26	0
26	entre 2	13	0
13	entre 2	6	1
6	entre 2	3	0
3	entre 2	1	1
1	entre 2	0	1

Luego se toma la columna de lo que sobra, de abajo hacia arriba, y nos queda: 110100111.

Ejercicio 3. Convertir el número 1000 en cada columna a los otros sistemas numéricos.

Binario	Octal	Decimal	Hexadecimal
1000			
	1000		
		1000	
			1000

Ejercicio 4. Completar los espacios en blanco convirtiendo entre sistemas numéricos.

Binario	Octal	Decimal	Hexadecimal
1111000			
	734		
		823	
			FF

Operaciones de los sistemas numéricos

Operaciones binarias

Primero vamos a ver las operaciones con binarios, para lo cual utilizaremos lo siguiente.

Suma en binario

1 + 0 =1

0 + 1 =1

0 + 0 = 0

1 + 1 =10

La suma de 1+1 sabemos que es 2, debe escribirse en binario con dos cifras (10), por tanto, 1+1 es 0 y se arrastra una unidad, que se suma a la posición continua hacia la izquierda. En la siguiente tabla se muestran siete ejercicios, uno por cada columna.

1	2	3	4	5	6	7
10101	111	110 0 1 1	1 1 1 0 0 0	1 1 1 1 1	1 0 0 1 1 1	1 1 1 1 1
11100	110	111 0 0 0	1 0 1 0 1 0	1 0 1 0 1	1 0 1 0 1 0	1 1 0 1 1
110001	110	1101 0 1 1	1 1 1 1 1 1	1 0 0 1 0	1 1 0 0 1 1	1 0 1 0 1
	10011		1 0 1 0 0 0 0 1	100 0 1 1 0	100 0 0 1 0 0	100 1 1 1 1

Ejemplos. Los números rojos son los resultados de las sumas.

Restas con binarios

Reglas

1 - 1	1 - 0	1 0 - 1	0 - 0
0	1	1	0

El número de arriba debe ser mayor que el de abajo.

Ejemplos. El resultado es la tercer línea.

1	2	3	4	5
1 1 0 0	1 1 0 1 1	1 1 1 1 1	1 1 0 0 1 1	1 1 1 0 0 0
1 0 1	1 0 1 0 1	1 1 0 1	0 1 1 0 1	0 0 1 0 1 1
1 1 1	**0 0 1 1 0**	**1 0 0 1 0**	**1 0 0 1 1 0**	**1 0 1 1 0 1**

Multiplicaciones con binarios

Para las multiplicaciones es muy fácil, ya que cero por cualquier número te da cero y uno por cualquier número te da el mismo número: 1 x 0 = 0 y 1 x 1 = 1.

Realicemos a continuación algunos ejercicios.

1	2	3	4	5
11101 x110	111x 110	111 0 1 1x1110	1 1 1 0 0 0x1010	1 1 1 1 1 x 1 0 1 0 1
0 0 0 0 0	0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1
1 1 1 0 1	1 1 1	1 1 1 0 1 1	1 1 1 0 0 0	0 0 0 0 0
1 1 1 0 1	1 1 1	1 1 1 0 1 1	0 0 0 0 0 0	1 1 1 1 1
1 0 1 0 1 1 1 0	1 0 1 0 1 0	1 1 1 0 1 1	1 1 1 0 0 0	0 0 0 0 0
		1 1 0 0 1 1 1 0 1 0	1 0 0 0 1 1 0 0 0 0	1 1 1 1 1
				1 0 1 0 0 0 1 0 1 1

División de binarios

Para dividir primero debemos verificar si el dividendo es mayor que el divisor, si es así continuamos.

```
       0 1 1 1
11 | 1 0 1 1 0
     1 1
      1 0 1
         11
       ------
         10 0
           11
       ------
            1
```

Supongamos que queremos dividir el número binario **10110** (22 en decimal) entre **11** (3 en decimal). El proceso sería el siguiente:

1. Dividimos el primer grupo de bits que sea igual o mayor al divisor:
   - **101** (equivale a 5 en decimal, mayor que 3).
2. **101** dividido entre **11** es igual a **1** con un residuo de **10**.
3. Bajamos el siguiente bit (el **1**), formando **101** (equivale a 5 en decimal).

4. Repetimos el proceso:
   - **101** entre **11** es **1** con un residuo de **10**.
5. Bajamos el siguiente bit (**0**), formando **100** (equivale a 4 en decimal).
6. **100** dividido entre **11** es igual a **1** con un residuo de **1**.
7. Como no hay más bits para bajar, hemos terminado.

Ejemplos:

```
         01 1 0 0
111 | 101 1 0 01
        1 11
      ----------
        1 0 0 0
          11 1
      ----------
          0 0 0 1 0 1
      ----------
```

```
         1001100
101 | 101111111
      101
      ---------
      000111
         101
      ---------
         0101
          101
      ---------
          00 011
      ---------
```

```
        101101
1 0 | 1011011
      10
      -------
      0011
        10
      -------
        010
         10
      -------
         0011
           10
      -------
           01
      -------
```

Operaciones octales

Suma octal

Para realizar sumas octales necesitamos la siguiente tabla:

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Entonces lo que se hace es que si quiero sumar 2 + 3, busco la intersección: fila 2 y columna 3 y ese es el resultado.

Ejemplo. Se suman 3 + 5 y si vemos en la tabla de suma octal es 10, entonces se pone el 0 y se acarrea 1; luego 3 + 1 de acarreo son 4, más 4 nos vuelve a dar 10, por lo que se vuelve a poner el 0 y se acarrea 1; luego 7 + 1 de acarreo es 10, esto más 2 es igual a 12, se pone el 2 y se acarrea el 1; luego 2 + 1 de acarreo da 3, más 3 nos da 6.

$$\begin{array}{r} 2\,7\,3\,3 \\ 3\,2\,4\,5 \\ \hline \mathbf{6\,2\,0\,0} \end{array}$$

En la tabla siguiente se muestran cinco ejercicios de sumas

Ej. 1	Ej.2	Ej.3	Ej.4	Ej. 5
734	533	222	127	437
527	247	777	276	717
1463	**1002**	**1221**	**425**	**1356**

Restas con octales

Primero debemos verificar que el número de arriba sea mayor que el de abajo, si no lo es intercambiamos posiciones, como se observa más adelante en el ejemplo, y luego procedemos a restar los números. En el ejemplo tenemos 4 menos 7, puesto que el 4 es más pequeño debe pedirle prestado al 5 (disminuyéndolo a 4) y se convierte en 14; después buscamos el número 14 en la tabla de suma octal, precisamente en la fila 7, y lo encontramos en la columna 5, ese número es el que se pone abajo; finalmente a 4 le quitamos 2 y nos quedan 2.

$$\begin{array}{r} 27 \\ 54 \\ \hline \end{array} \qquad \begin{array}{r} 5\,4 \\ 2\,7 \\ \hline \mathbf{2\;\,5} \end{array}$$

Realicemos algunos ejemplos:

Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5	Ejemplo 6	Ejemplo 7
734	533	703	276	716	100	221
527	247	576	127	437	76	167
205	264	105	147	257	002	032

Multiplicaciones octales

La multiplicación se hace de la misma manera en que la realizaríamos con los decimales, solo que al multiplicar usamos la siguiente tabla y cuando se trata de sumar retomamos la tabla anterior de la suma octal para no caer en errores.

Tabla de multiplicación octal

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Comencemos la explicación con un ejemplo:

	6	5	4	3	x	1	3	5
		4	1	3	5	7		
+	2	4	0	5	1			
	6	5	4	3				
1	1	5	6	3	6	7		

Explicación de la multiplicación

Primera línea de multiplicación

Comenzamos con los primeros números: 5 × 3 nos da 17, entonces ponemos el 7 y se acarrea 1; luego 5 × 4 es 24, pero se había acarreado 1 y se le suma, entonces se tiene 25, se pone el 5 y se acarrea el número 2; ahora sigue multiplicar 5 × 5, que es 31, más 2 de acarreo son 33, se pone el 3 y se acarrea 3; toca multiplicar 5 × 6, que es igual a 36, más 3 de acarreo se suman con la tabla octal y nos da 41.

Segunda línea de multiplicación

En la segunda línea se multiplican 3 × 3, lo que nos da 11, se pone 1 y se sube 1 de acarreo; luego sigue multiplicar 3 × 4, que si buscamos en la fila 3 y columna 4 nos dará 14, más 1 que llevábamos de acarreo son 15, se pone el 5 y se sube el 1 de acarreo; después multiplicamos 3 × 5, que son 17, más 1 de acarreo tenemos 20, se pone el 0 y se sube 2 de acarreo, sigue multiplicar 3 × 6 y nos da 22, más 2 de acarreo nos da 24, y ya se pone el 24. La segunda línea queda así: 24051.

Tercera línea

Se multiplica por el 1, y como todo número multiplicado por 1 es el mismo número, se pone el número 6543 tal como está. Ahora se procede a la suma de los números, como se vio en el tema de las sumas.

Divisiones octales

Como en las multiplicaciones, el proceso de las divisiones es igual que en decimal, lo que cambia son las tablas de multiplicar y se resta en octal.

Ejemplos

$$
\begin{array}{r|l}
\multicolumn{1}{r}{} & \quad 211 \\ \cline{2-2}
351 & 765\ 45 \\
\multicolumn{1}{r}{} & 722 \\ \cline{2-2}
\multicolumn{1}{r}{} & 043\ 4 \\
\multicolumn{1}{r}{} & \ \ 35\ 1 \\ \cline{2-2}
\multicolumn{1}{r}{} & \ \ 00635 \\
\multicolumn{1}{r}{} & \quad 351 \\ \cline{2-2}
\multicolumn{1}{r}{} & \quad 264 \\ \cline{2-2}
\end{array}
$$

Operaciones con números hexadecimales

Al igual que los octales, para realizar todas sus operaciones se utilizan las siguientes tablas y se suman renglón por columna.

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Suma hexadecimal

Como en el sistema hexadecimal, también se hace por medio de la tabla correspondiente de la suma para no confundir con la suma decimal.

Ejemplos

Ejemplo 1. En este ejemplo se suman E + E, si vemos en la tabla de suma hexadecimal, fila E con columna E es 1C, entonces se pone C y se acarrea 1; luego B + F es fila B con columna F, que da 1A y le sumamos 1 de acarreo, por lo que es 1B, se pone la B y se acarrea 1; luego E + A resulta 18, más 1 de acarreo da 19, se pone el 9 y se acarrea el 1; finalmente, B + C nos da 17 más 1 de acarreo es 18.

$$\begin{array}{r} B\ E\ B\ E \\ C\ A\ F\ E \\ \hline \mathbf{1\,8\ 9\ B\,C} \end{array}$$

Restas con hexadecimales

Primero debemos verificar que el número de arriba sea mayor que el de abajo, si no lo es intercambiamos posiciones. Como se puede observar en la tabla siguiente, en la primera columna la palabra BEBE es menor que CAFÉ, porque el primer dígito que está en el primer número, una B, es menor que C. Entonces intercambiamos posiciones, luego procedemos a restar los números: en el ejemplo tenemos E − E, y nos da cero; en seguida está F − B, por lo que buscamos una F en la tabla de la suma, en el renglón B, y la encontramos en la columna 4, lo cual significa que B + 4 es F, por lo que ponemos el 4; después toca restar A − E, y observamos que A es menor que E, por lo tanto le pide prestado un número a C y se convierte en 1A, buscamos en la columna E el valor 1A y lo encontramos en la columna C, de modo que ponemos la C; por último, sigue restar B – B, recordemos que la C prestó uno y se quedó solo con B, por lo tanto B − B es igual a 0.

B E B E	C A F E
C A F E	B E B E
	0 C 4 0

Multiplicaciones hexadecimales

Para realizar las multiplicaciones se necesita la siguiente tabla.

Multiplicación

*	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

La forma de realizar estas multiplicaciones es la siguiente: se hace la multiplicación como si fuera en decimal, pero en lugar de poner las tablas de los números decimales se toman las tablas de los hexadecimales. Cuando se hacen las sumas se utiliza también la tabla de suma, como se ve en la columna donde se suma el 4 con el 6 que nos da A.

	1	F	3	2	x	B	0	E
				1	B	4	B	C
+				0	0	0	0	
		1	5	7	2	6		
		1	5	8	D	A	B	C

Divisiones hexadecimales

Para las divisiones lo primero que se hace es observar cuántos dígitos tiene el divisor, y se tantea cuántas veces cabe el divisor en los mismos dígitos del dividendo; se pone el número en el cociente y se multiplica por el divisor; luego se realiza la resta, y si la resta es más pequeña que el divisor estamos bien, si no se borra y se eleva el cociente un número más. Por supuesto, se usan las tablas de multiplicar y sumar hexadecimales.

	1 3 1
90	A B C D
	9 0
	1 B C
	1 B 0
	0 0 C D
	9 0
	3 D

Aplicaciones de los sistemas numéricos en la computación

Los sistemas numéricos son fundamentales en el ámbito de la computación, ya que permiten representar y manipular datos en formatos entendibles para las máquinas. A continuación, se explica cómo se aplican los sistemas numéricos en la computación moderna.

Sistema binario (base 2) y representación de datos

En la computación, el sistema binario es crucial porque los dispositivos digitales operan a través de circuitos que pueden encontrarse en un estado de encendido o apagado, representado como 1 y 0. Estos valores binarios constituyen la base de los datos en cualquier computadora, ya que toda la información se procesa y almacena en bits, los cuales representan dígitos en binario (Tanenbaum y Austin, 2013). Además, el sistema binario permite representar instrucciones y datos, desde números hasta caracteres y gráficos, en un lenguaje que los circuitos electrónicos pueden entender (Stallings, 2017).

Sistema octal y su rol en codificación

Aunque su uso es menos común hoy en día, el sistema octal (base 8) fue usado en las primeras generaciones de computadoras para simplificar el trabajo con números binarios largos. Cada dígito octal representa tres bits binarios, lo que facilitaba el trabajo en sistemas de bajo nivel como el ensamblador y en la programación de *hardware* (Mano y Kime, 2013). En el caso de algunos sistemas operativos antiguos, como UNIX, el octal también se utiliza para representar permisos de archivos, proporcionando una notación compacta y accesible para operaciones de bajo nivel (Tanenbaum y Bos, 2014).

Sistema hexadecimal (base 16) y programación de bajo nivel

El sistema hexadecimal (base 16) es ampliamente utilizado en computación, especialmente en programación de bajo nivel y depuración, debido a su capacidad para representar grandes cantidades de datos en menos espacio. Dado que cada dígito hexadecimal equivale a cuatro bits, el sistema hexadecimal permite expresar secuencias binarias de una forma más compacta, lo cual es útil en la lectura y escritura de direcciones de memoria y en la representación de colores en la programación gráfica (Hennessy y Patterson, 2017). Así, los programadores y técnicos pueden trabajar eficientemente en la manipulación de datos binarios largos.

Sistema decimal (base 10) y aplicaciones de alto nivel

Aunque la computadora internamente utiliza sistemas como el binario o hexadecimal, los resultados y datos suelen traducirse al sistema decimal para que los usuarios puedan interpretarlos fácilmente. Esto se debe a que el sistema decimal es el más natural para los humanos, y las interfaces de usuario suelen utilizarlo para mostrar datos financieros, cálculos científicos y otro tipo de información en un formato comprensible (Stallings, 2017). En este sentido, las aplicaciones de *software* suelen convertir las operaciones realizadas en binario o hexadecimal a una representación decimal para facilitar la interacción humana.

En resumen, los sistemas numéricos, especialmente el binario, octal, hexadecimal y decimal, son indispensables en la computación, ya que permiten la representación y manipulación eficiente de datos y código. Esta estructura numérica es la base del funcionamiento de las computadoras modernas, al facilitar tanto la comunicación entre máquinas como la interacción con el usuario.

Referencias

Andonegui Zabala, M. (2007). *El sistema numérico decimal*. Federación Internacional Fe y Alegría.

Clagett, M. (1989). *Ancient egyptian science: a source book*. University of Pennsylvania Press.

Closs, M. P. (1986). *Native American Mathematics*. University of Texas Press.

Coronado Padilla, J. A. (2014). *Sistemas numéricos residuales: fundamentos lógico-matemáticos* (1.ª ed.). Ediciones Unisalle.

Gardner, M. (1983). *Logic machines and diagrams*. Harvard University Press.

Hennessy, J. L., y Patterson, D. A. (2017). *Computer architecture: a quantitative approach* (6.ª ed.). Morgan Kaufmann.

Ifrah, G. (2000). *The universal history of numbers: from prehistory to the invention of the computer*. John Wiley y Sons.

Johnsonbaugh, R. (1999). *Matemáticas discretas* (4.ª ed.). Editorial Pearson.

Joseph, G. G. (2011). *The crest of the peacock: non-european roots of mathematics*. Princeton University Press.

Katz, V. J. (2009). *A history of mathematics: an introduction* (3.ª ed.). Addison-Wesley.

Knuth, D. E. (1997). *The art of computer programming: volume 2: seminumerical algorithms* (3.ª ed.). Addison-Wesley.

Mano, M. M., y Kime, C. R. (2013). *Logic and computer design fundamentals*. Pearson.

Menninger, K. (1992). *Number words and number symbols: a cultural history of numbers*. Dover Publications.

Robledo, I., Mendoza, A., y Perea, R. (s.f.). *Sistemas numéricos*. Universidad Autónoma de Ciudad Juárez.

Robson, E. (2008). *Mathematics in ancient Iraq: a social history*. Princeton University Press.

Rosen, K. H. (2012). *Matemática discreta y sus aplicaciones* (7ª ed.). McGraw-Hill.

Stallings, W. (2017). *Computer organization and architecture: designing for performance*. Pearson.

Tanenbaum, A. S. (2016). *Structured computer organization* (6.ª ed.). Pearson.

Tanenbaum, A. S., y Austin, T. (2013). *Structured computer organization*. Pearson.

Tanenbaum, A. S., y Bos, H. (2014). *Modern operating systems*. Pearson.

UNAD. (s.f.). *Sistemas numéricos*.

Capítulo 3. Introducción a la teoría de conjuntos

José Martín Barajas Guerrero[1]
Blanca Verónica Alvarado de la Torre[2]

DOI: https://doi.org/10.52501/cc.440.03

Resumen

La teoría de conjuntos constituye uno de los fundamentos esenciales de las matemáticas, ya que permite comprender y estructurar la mayoría de los objetos matemáticos. Un conjunto se define como una colección de elementos bien definidos y diferenciables, que pueden representarse mediante distintas notaciones: por extensión (listando elementos), por comprensión (mediante propiedades) o mediante intervalos en el caso de números reales. Conceptos básicos como la pertenencia (∈) y la contención (⊆) permiten establecer relaciones entre elementos y conjuntos, así como entre conjuntos mismos, facilitando su organización y análisis. Estas relaciones suelen representarse visualmente mediante diagramas de Venn.

Las operaciones fundamentales con conjuntos incluyen la unión, que agrupa los elementos de dos conjuntos; la intersección, que identifica los elementos comunes; la diferencia, que señala los elementos que pertenecen a un conjunto, pero no a otro; el complemento, que considera los elementos fuera de un conjunto dentro de un universo; y la diferencia simétrica, que reúne los elementos exclusivos de cada conjunto. Estas operaciones pueden extenderse a múltiples conjuntos y son clave para el análisis matemático.

Asimismo, la teoría de conjuntos se apoya en diversas propiedades que facilitan la manipulación y simplificación de expresiones, como las propiedades conmutativa, asociativa, distributiva, de identidad, de complemento e idempotencia. Estas reglas garantizan consistencia en los resultados y permiten un manejo más eficiente de los conjuntos. En conjunto,

[1] Maestro en Ciencias en Ingeniería Administrativa. Jefe de Laboratorio de Cómputo en Tecnológico Nacional de México. ORCID: https://orcid.org/0000-0002-2443-9590 ; correo electrónico: martin.barajas@zacatecas.tecnm.mx

[2] Maestra en Informática Admirativa. Docente en Tecnológico Nacional de México.

este marco teórico es indispensable para el desarrollo del pensamiento lógico y la formalización matemática.

Palabras clave: *conjuntos, propiedades de conjuntos, operaciones de conjuntos*.

Fundamentos de la teoría de conjuntos

Intuitivamente, un **conjunto** se refiere a una colección o agrupación de elementos que tienen alguna característica común y puede representarse de varias maneras, como en forma de lista o por propiedad (Lewin, 2011).

Formalmente, un **conjunto** se define como una colección de objetos bien definidos y diferenciables entre sí, que se conocen como los *elementos* o *miembros* del conjunto. Un ejemplo es el conjunto de los números enteros pares entre el cero y diez. Casi todos los objetos matemáticos son, ante todo, conjuntos. Por consiguiente, la teoría de los conjuntos es la base sobre la cual se construyen las matemáticas. La notación y representación de conjuntos se realiza de las siguientes maneras:

a) **Uso de llaves {}**. Los conjuntos se representan generalmente usando llaves. Por ejemplo, un conjunto de números naturales menores que 5 puede escribirse como {1,2,3,4}.

b) **Notación de conjuntos con propiedades**. Esta notación describe los elementos de un conjunto en términos de una propiedad que comparten. Se escribe como {x | propiedad de x}. Por ejemplo, el conjunto de números pares positivos menores que 10 se puede escribir como {x | x es par y $0<x<10$}.

c) **Notación de intervalos**. Se usa principalmente para representar subconjuntos de números reales. Los intervalos tienen diferentes notaciones según si incluyen o no los extremos:

- Intervalo cerrado: [a,b], incluye los extremos a y b.
- Intervalo abierto: (a,b), no incluye los extremos.
- Intervalo semiabierto o semicerrado: [a,b) o (a,b], incluye solo uno de los extremos (Lipschutz, 1991).

Pertenencia

La pertenencia se refiere a la relación entre un elemento y un conjunto. Cuando un elemento pertenece a un conjunto, decimos que es un **miembro** de ese conjunto y usamos el símbolo $\in$ (Cantor, 2006).

Ejemplo:

Si tenemos el conjunto $A = \{1,2,3,4\}$, podemos decir que:

$2 \in A$ (porque 2 es un elemento de A).

$5 \notin A$ (porque 5 no es un elemento de A).

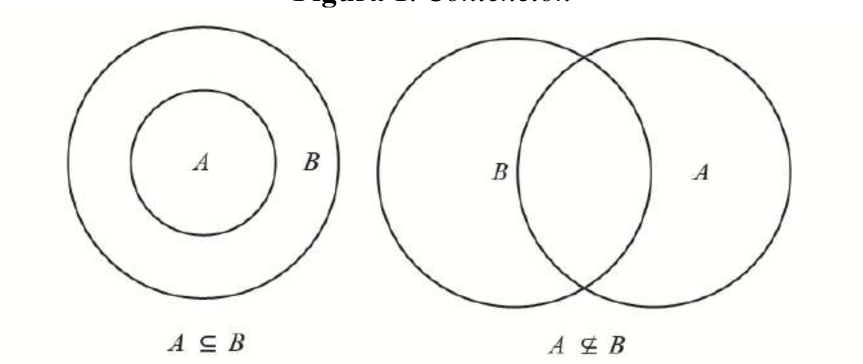
Contención

La contención se refiere a la relación entre conjuntos, específicamente cuando todos los elementos de un conjunto están dentro de otro conjunto. En este caso, decimos que uno es un subconjunto del otro y usamos el símbolo $\subseteq$.

Si un conjunto B es un subconjunto de A, significa que todos los elementos de B también están en A. Esto se expresa como $B \subseteq A$.

Ejemplo: si todos los elementos de A son también elementos de B, esto es si cuando $x \in A$, entonces $x \in B$, decimos que A es un subconjunto de B o que A está contenido en B y se escribe $A \subseteq B$. Si A no es un subconjunto de B, se escribe $A \nsubseteq B$ (véase la figura 1).

Figura 1. *Contención*



Fuente: elaboración propia.

Los diagramas, como la figura 1, se usan para mostrar las relaciones entre los conjuntos y se llaman diagramas de Venn en honor del lógico británico John Venn.

Ejemplo:

Si A = {1,2,3,4,5,6}, B = {2,4,5}, C = {1,2,3,4,5}

entonces $B \subseteq A$, $B \subseteq C$, $C \subseteq A$.

Sin embargo $A \nsubseteq B$, $A \nsubseteq C$, $C \nsubseteq B$

Operaciones básicas con conjuntos

Las operaciones básicas con conjuntos son herramientas fundamentales en teoría de conjuntos. Las operaciones principales son la unión, intersección, diferencia y complemento (Smith, 2000).

Unión (∪)

La unión de dos conjuntos A y B se define como el conjunto que tiene todos los elementos que pertenecen a A o B y se indica como $A \cup B$.

$A \cup B = \{x \mid x \in A \text{ o } x \in B\}$

Observe que $x \in A \cup B$ si $x \in A$ o $x \in B$ o x pertenece a ambos.

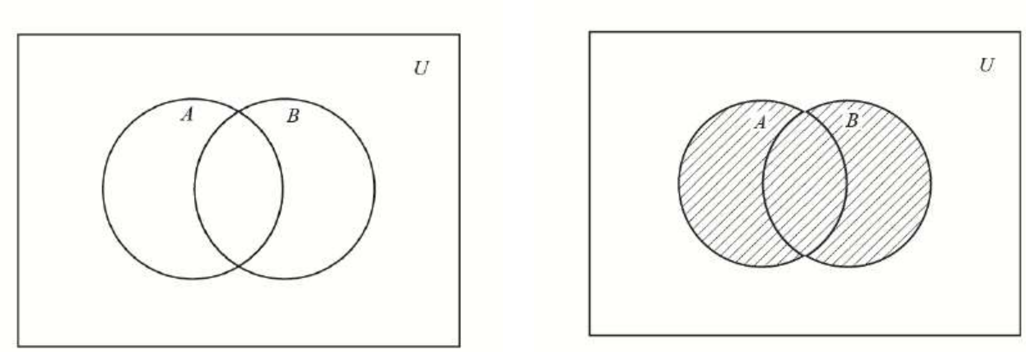
Ejemplo:

Si A = {1,2,3} y B = {3,4,5}, entonces:

$A \cup B = \{1,2,3,4,5\}$

Se puede ilustrar la unión de dos conjuntos con un diagrama de Venn de la siguiente manera: si A y B son los conjuntos dados en la figura 2(a), entonces $A \cup B$ es el conjunto de puntos en la región sombrada, como lo indica la figura figura2(b).

Figura 2a y 2b. *Unión de dos conjuntos A y B*



Fuente: elaboración propia.

Intersección (∩)

La intersección de dos conjuntos A y B se define como el conjunto que contiene todos los elementos que están en ambos conjuntos, se denota como A∩B (Smith, 2000).

Por lo tanto, $A \cap B = \{x \mid x \in A \text{ y } x \in B\}$

Ejemplo:

Si A = {a,b,c,e,f}, B = {b,e,f,r,s} y C = {a,t,u,v}

Encuentre A∩B, A∩C y B∩C

Entonces: los elementos b, e y f son los únicos que pertenecen a A y B, por lo cual

A∩B = {b ,e, f}

De igual manera,

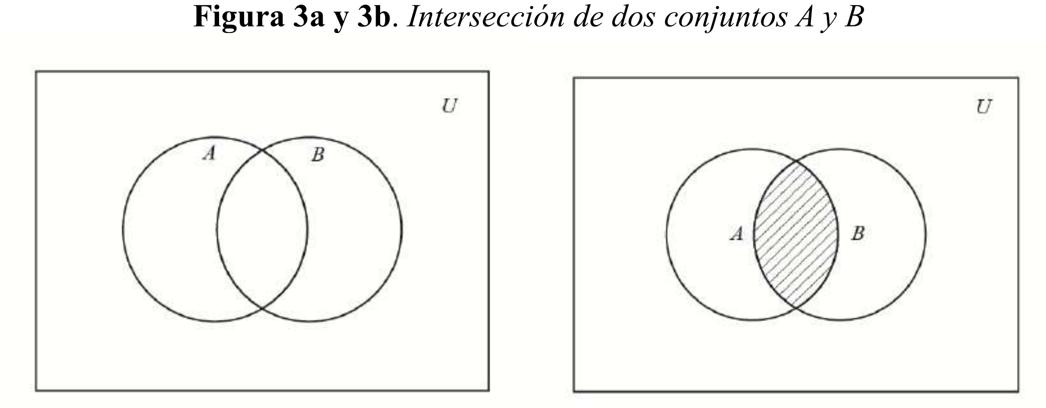
A∩C = {a}

No existen elementos que pertenezcan tanto a A como a B, por lo que

B∩C

=∅

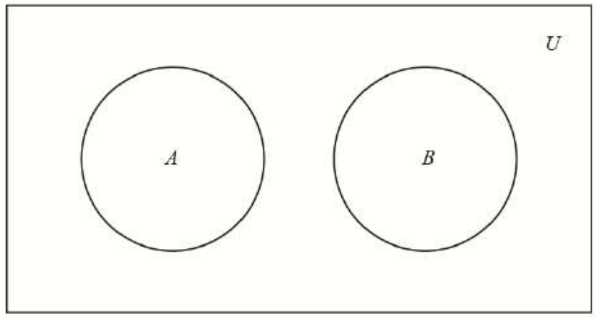
Figura 3a y 3b. *Intersección de dos conjuntos A y B*



Fuente: elaboración propia.

Es posible ilustrar la intersección de dos conjuntos por el diagrama de Venn. Si A y B son los conjuntos dados en la figura 3(a), entonces A∩B es la región sombreada de la figura 3(b). La figura 4 ilustra un diagrama de Venn con dos conjuntos ajenos (disjuntos) (Smith, 2000).

Figura 4. *Diagrama de Venn con dos conjuntos ajenos (disjuntos)*



Fuente: elaboración propia.

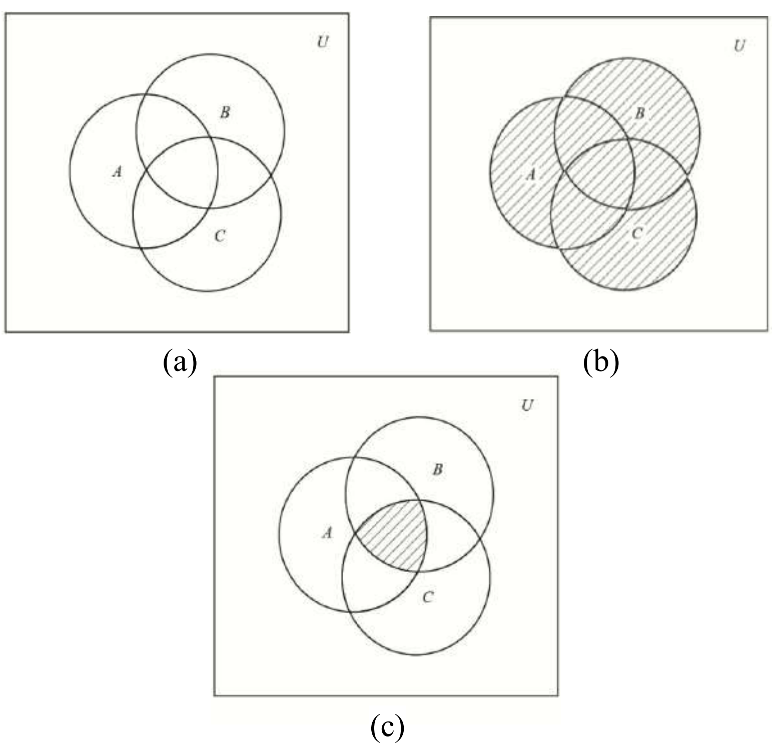
Las operaciones de unión e intersección pueden generalizarse para tres o más conjuntos. Así pues,

$A \cup B \cup C = \{x \mid x \in A \text{ o } x \in B \text{ o } x \in C\}$

$A \cap B \cap C = \{x \mid x \in A \text{ y } x \in B \text{ y } x \in C\}$

La región sombreada en la figura 5(b) es la unión de los conjuntos A, B y C mostrada en la figura 5(a) y la región sombreada en la figura 5(c) es la intersección de los conjuntos A, B y C (Smith, 2000).

Figura 5. *Unión e intersección para tres conjuntos*



Fuente: elaboración propia.

Ejemplo:

Sea A = {1,2,3,4,5,7}, B = {1,3,8,9}, C = {1,3,6,8}

Entonces A ∩ B ∩ C es el conjunto de elementos que pertenecen a A, B y C. Por tanto, A ∩ B ∩ C={1,3}.

Diferencia (− o \\)

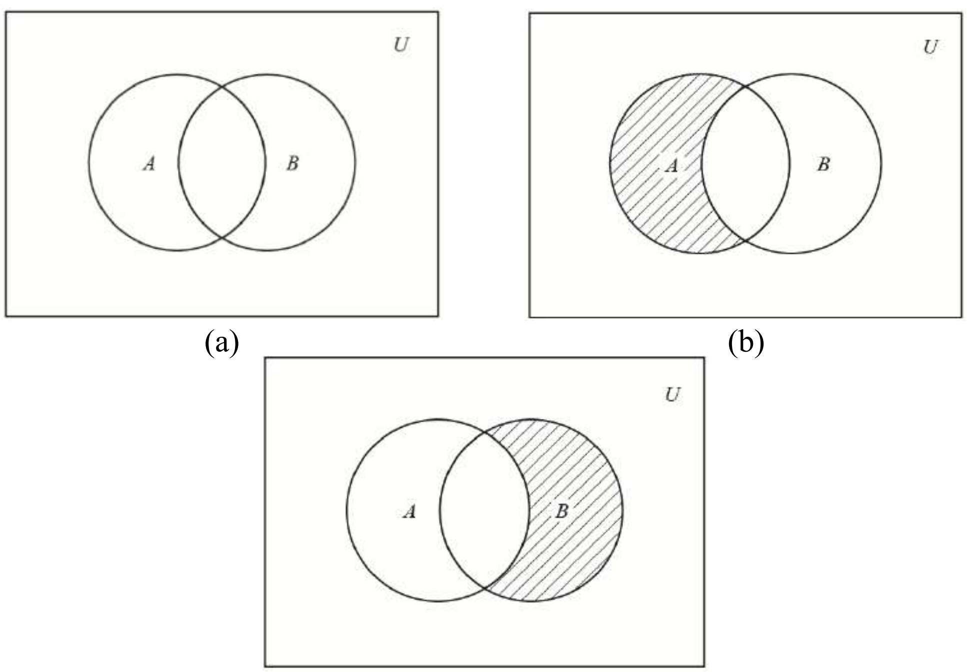
Si A y B son conjuntos, se define el complemento de B con respecto a A como el conjunto de todos los elementos que pertenecen a A pero no a B y se indica A – B (Smith, 2000).

Por lo tanto

$A - B = \{ x \mid x \in A \text{ y } x \notin B \}$

Si A y B son los conjuntos en la figura 6(a), entonces A – B y B – A son los conjuntos de puntos en las regiones sombreadas de la figura 6(b) y (c), respectivamente.

Figura 6. *Diferencia*



(a) (b)

(c)

Fuente: elaboración propia.

Ejemplo:

Si A = {1,2,3} y B = {3,4,5}, entonces:

A – B = {1,2} y B – A= {4,5}

Complemento (A′o A^c)

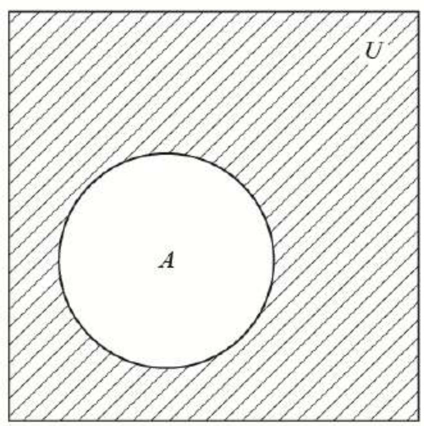
El complemento de un conjunto A es el conjunto de todos los elementos que no están en A, pero sí están en un conjunto universal U (el conjunto que contiene todos los elementos de interés en ese contexto) (Smith, 2000). Se denota como A′o A^c.

Por lo tanto

$A^c = \{ x \mid x \notin A \}$

Si A es el conjunto en la figura 7, su complemento es la región sombreada.

Figura 7. *Complemento de un conjunto A*



Fuente: elaboración propia.

Ejemplo:

Si el conjunto universal U = {1,2,3,4,5} y A = {1,2,3}, entonces:

$A^c = U - A = \{4,5\}$

Diferencia simétrica

Si A y B son dos conjuntos, su diferencia simétrica se define como el conjunto de todos los elementos que pertenezcan a A o B, pero no a ambos, y se indica por A⊕B (Bruckner, 2008). Por lo tanto, $A \oplus B = \{ x \mid (x \in A \text{ y } x \notin B) \text{ o } (x \in B \text{ y } x \notin A)\}$

Ejemplo:

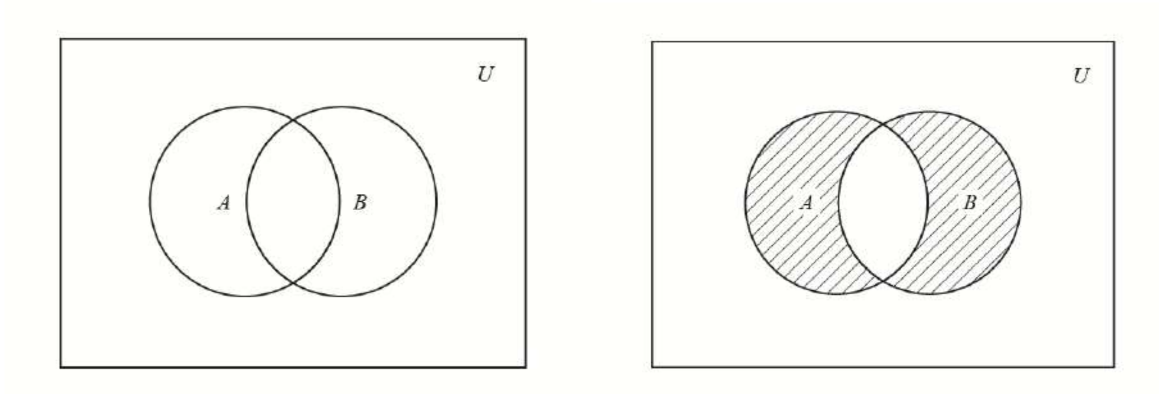
A = {a,b,c,d} y B {a,c,e,f,g}

Entonces

A⊕B = {b,d,e,f,g}

Si A y B son como se indican en la figura 8(a), su diferencia simétrica es la región sombreada en la figura 8(b). Es fácil ver que A⊕B = (A-B) ∪ (B-A).

Figura 8a y 8b. *Diferencias simétricas*



Fuente: elaboración propia.

Propiedades de los conjuntos

Las propiedades de los conjuntos se refieren a reglas que se aplican en las operaciones de unión, intersección y complemento. Estas propiedades facilitan el trabajo con conjuntos en matemáticas y ayudan a simplificar expresiones (Villanueva, 2015). A continuación, describo las propiedades más importantes con ejemplos.

Teorema 1. *Las operaciones con conjuntos antes definidas satisfacen las siguientes propiedades*.

Propiedad conmutativa

Esta propiedad indica que el orden en la unión e intersección de conjuntos no afecta el resultado.

Unión: A∪B = B∪A

Intersección: A∩B = B∩A

Ejemplo**:**

Si A = {1,2} y B = {2,3}, entonces:

$A \cup B = \{1,2,3\} = B \cup A$

$A \cap B = \{2\} = B \cap A$

Propiedad asociativa

Esta propiedad indica que al unir o intersectar varios conjuntos, el resultado no depende de cómo se agrupen.

Unión: $(A \cup B) \cup C = A \cup (B \cup C)$

Intersección:$(A \cap B) \cap C = A \cap (B \cap C)$

Ejemplo**:**

Si A = {1}, B = {1,2}, y C = {2,3}, entonces:

$(A \cup B) \cup C = \{1,2,3\} = A \cup (B \cup C)$

$(A \cap B) \cap C = \emptyset = A \cap (B \cap C)$

Propiedad distributiva

Esta propiedad indica cómo se distribuyen las operaciones de unión e intersección entre sí.

Unión sobre intersección: $A \cup (B \cap C)=(A \cup B) \cap (A \cup C)$

Intersección sobre unión: $A \cap (B \cup C)=(A \cap B) \cup (A \cap C)$

Ejemplo:

Si A={1}, B={1,2}, y C={2,3}, entonces:

$A \cup (B \cap C)=\{1\} \cup \{2\}=\{1,2\}=(A \cup B) \cap (A \cup C)$

Propiedad de identidad

Unir un conjunto con el conjunto vacío o intersectarlo con el conjunto universal no cambia el conjunto original.

Unión: $A \cup \emptyset = A$

Intersección: U = $A \cap U=A$ (donde U es el conjunto universal)

Ejemplo:

Si A = {1,2}, entonces:

$A \cup \emptyset = \{1,2\}$

$A \cap U = \{1,2\}$ (si U es el conjunto universal)

Propiedad de complemento

Estas propiedades describen la relación entre un conjunto y su complemento.

Ley de complemento: $A \cup A^c = U$

$A \cap A^c = \emptyset$

Ley de doble complemento: $(A^c)^c = A$

Ejemplo:

Si $A = \{1,2\}$A y $U = \{1,2,3\}$, entonces:

$A^c = \{3\}$

$A \cup A^c = \{1,2,3\}$

$A \cap A^c = \emptyset$

Leyes de idempotencia

Unir o intersectar un conjunto consigo mismo no cambia el conjunto.

Unión: $A \cup A = A$

Intersección: $A \cap A = A$

Ejemplo:

Si $A = \{1,2\}$, entonces:

$A \cup A = \{1,2\}$,

$A \cap A = \{1,2\}$

Leyes de Morgan

Las leyes de Morgan expresan cómo se relacionan la unión e intersección con el complemento.

Primera ley: $(A \cup B)^c = A^c \cap B^c$

Segunda ley: $(A \cap B)^c = A^c \cup B^c$

Ejemplo**:**

Si $A = \{1,2\}$, $B = \{2,3\}$ y $U = \{1,2,3,4\}$, entonces:

$A \cup B = \{1,2,3\}$

$(A \cup B)^c = \{4\}$

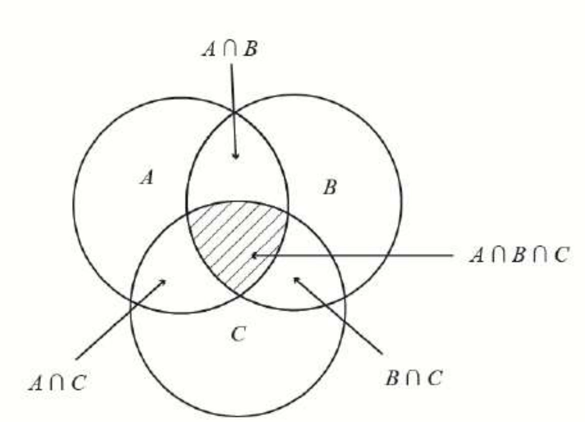
$A^c = \{3,4\}$, $B^c = \{1,4\}$

$A^c \cap B^c = \{4\}$

Teorema 2. *Si A y B son conjuntos finitos, entonces* $|A \cup B| = |A| + |B| - |A \cap B|$ (Seymour y Lipson, 2007).

Esta situación para tres conjuntos es más complicada, como se muestra en la figura 9. Ahora se expondrá el principio de adición para tres conjuntos.

Figura 9. *Intersección de tres conjuntos*



Fuente: elaboración propia.

Teorema 3. *Sean A, B y C conjuntos finitos . Entonces* $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$

Ejemplo:

Una compañía de computación necesita contratar a 25 programadores para tareas de programación de sistemas y 40 programadores para la programación de computación, de estos empleados, se espera que 10 realicen tareas de dos tipos. ¿Cuántos programadores deberá contratar?

Donde

A = Programadores de sistema

B = Programadores de aplicación

Entonces $|A| = 25$, $|B| = 40$, y $|A \cap B| = 10$. Así pues, el número que debemos emplear es $|A \cup B| = |A| + |B| + |C| - |A \cap B| = 25+40-10 = 55$.

Relaciones entre conjuntos

Una relación entre dos conjuntos es una correspondencia entre los elementos de un conjunto con los elementos de otro conjunto. Se puede representar como un conjunto de pares ordenados (a,b), donde el primer elemento pertenece al conjunto A y el segundo elemento pertenece al conjunto B (Arroyo, 2018).

Suponga que R es una relación de A a B. Entonces R es un conjunto de pares ordenados donde el primer elemento proviene de A y el segundo proviene de B. Es decir, para cada par $a \in A$ y $b \in B$ es verdadera exactamente una de las siguientes proposiciones:

i) $(a, b) \in R$; entonces se dice "a está relacionado con b", lo que se escribe como aRb.

ii) $(a, b) \notin R$; entonces se dice "a no está relacionado con b", lo que se escribe como $a \not R b$. Si R es una relación del conjunto A en sí mismo; es decir, si R es un subconjunto de $A^2 = A \times A$, entonces se dice que R es una relación sobre A.

El dominio de una relación R es el conjunto de todos los primeros elementos de los pares ordenados que pertenecen a R, y el rango es el conjunto de los segundos elementos.

Relación binaria

Describe una relación entre los elementos de dos conjuntos. Formalmente, una relación binaria entre dos conjuntos A y B es un subconjunto del producto cartesiano $A \times B$.

Por ejemplo, si tenemos los conjuntos $A = \{1,2,3\}$ y $B = \{a,b\}$, una relación entre A y B podría ser el conjunto de pares $\{(1,a),(2,b)\}$.

Sean A y B conjuntos. Una relación binaria, o simplemente una relación de A a B, es un subconjunto del producto cartesiano $A \times B$ (Seymour y Lipson, 2007). Por ejemplo, si tenemos los conjuntos $A = \{1,2,3\}$ y $B = \{a,b\}$, una relación entre A y B podría ser el conjunto de pares $\{(1,a),(2,b)\}$.

Formalmente, una relación R en A es un subconjunto del producto cartesiano $A \times A$, es decir: $R \subseteq A \times A$.

Propiedades de las relaciones binarias

Dependiendo de las propiedades que cumpla R, puede clasificarse de diversas maneras:

- Reflexiva: $\forall a \in A, (a,a) \in R$.
- Simétrica: $\forall a,b \in A, (a,b) \in R \Rightarrow (b,a) \in R$.

- Antisimétrica: $\forall a,b \in A, (a,b)\in R \wedge (b,a)\in R \Rightarrow a=b$.
- Transitiva: $\forall a,b,c \in A, (a,b)\in R \wedge (b,c)\in R \Rightarrow (a,c)\in R$

En el caso especial donde A = B se dice que R es una relación en A.

Relaciones reflexivas

Una relación R sobre un conjunto es reflexiva si aRa para toda a ∈ A; es decir, si (a, a) ∈ R para toda a ∈ A. Por tanto, R no es reflexiva si existe a ∈ A tal que (a, a) ∉R.

Ejemplo 1:

Considere las cinco relaciones siguientes sobre el conjunto A = {1, 2, 3, 4}:

R1 = {(1, 1), (1, 2), (2, 3), (1, 3), (4, 4)}

R2 = {(1, 1)(1, 2), (2, 1), (2, 2), (3, 3), (4, 4)}

R3 = {(1, 3), (2, 1)}

R4 = ∅, la relación vacía

R5 = A × A, la relación universal

Determine cuáles de las relaciones son reflexivas.

Puesto que A contiene los cuatro elementos 1, 2, 3 y 4, una relación sobre A es reflexiva si contiene los cuatro pares (1, 1),(2, 2), (3, 3) y (4, 4). Así, solo R2 y la R5 = A × A son reflexivas. Observe que R1, R3 y R4 no son reflexivas porque, por ejemplo, (2, 2) no pertenece a ninguna de ellas (Quesada, 2021).

Relaciones simétricas y antisimétricas

Una relación R sobre un conjunto A es simétrica si siempre que aRb entonces bRa; es decir, siempre que (a, b) ∈ R entonces (b, a) ∈ R. Por tanto, R no es simétrica si existen a, b ∈ A, tales que (a, b) ∈ R pero (b, a)∉R.

Ejemplo:

a) Determine cuáles de las relaciones son simétricas del ejemplo 1:

R1 no es simétrica porque (1, 2) ∈ R1 pero (2, 1)∉R1.

R3 no es simétrica porque (1, 3)∈R3 pero (3, 1)∉R3.

Las otras relaciones son simétricas.

Observación: las propiedades de ser simétrica y ser antisimétrica no son negaciones entre sí. Por ejemplo, la relación R = {(1, 3), (3, 1), (2, 3)} no es simétrica ni antisimétrica.

Por otra parte, la relación $R^c = \{(1, 1), (2, 2)\}$ es tanto simétrica como antisimétrica (Quesada, 2021).

Relaciones transitivas

Una relación R sobre un conjunto A es transitiva si siempre que aRb y bRc entonces aRc; es decir, siempre que $(a, b), (b, c) \in R$ entonces $(a, c) \in R$ (Quesada, 2021). Por tanto, R no es transitiva si existe $a, b, c \in R$ tal que $(a, b), (b, c) \in R$ pero $(a, c) \notin R$.

Ejemplo:

a) Determine cuáles de las relaciones en el ejemplo 1 son transitivas.

La relación R3 no es transitiva porque $(2, 1), (1, 3) \in R3$ pero $(2, 3) \notin R3$. Todas las otras relaciones son transitivas.

Cardinalidad de conjuntos

La cardinalidad de un conjunto se refiere a la cantidad de elementos que contiene (Weisstein, 1996). Es una forma de medir el *tamaño* del conjunto, sin importar cuáles son los elementos, sino cuántos hay.

La cardinalidad de un conjunto A se representa como $|A|$ y describe el número de elementos en el conjunto.

Ejemplos:

Si $A = \{1,2,3\}$, entonces la cardinalidad de A es $|A|=3$, porque hay tres elementos en el conjunto.

Si $B = \{a,e,i,o,u\}$, entonces $|B|=5$, ya que contiene cinco elementos.

Un conjunto vacío, $\emptyset$ o $\{\}$, tiene una cardinalidad de $|\emptyset|=0$ porque no contiene elementos.

La comparación de cardinalidades permite determinar si dos conjuntos tienen la misma cantidad de elementos (Weisstein, 1996). Algunas posibles comparaciones son las siguientes:

Igualdad de cardinalidad: dos conjuntos A y B tienen la misma cardinalidad si el número de elementos en cada uno es igual. Esto se escribe como $|A|=|B|$.

Ejemplo:

$C = \{1,2\}$ y $D = \{a,b\}$ tienen la misma cardinalidad porque $|C| = |D| = 2$.

Cardinalidad de un conjunto mayor que otro: Si un conjunto A tiene más elementos que otro conjunto B, entonces decimos que $|A| > |B|$.

Ejemplo:

Si $E = \{1,2,3,4,5\}$ y $F = \{x,y\}$, entonces $|E| > |F|$ porque $|E| = 5$ y $|F| = 2$.

Cardinalidad de conjuntos infinitos: hay conjuntos infinitos, como el conjunto de los números naturales $\mathbb{N} = \{1,2,3,\ldots\}$ o el conjunto de los números reales $\mathbb{R}$. Aunque ambos conjuntos son infinitos, no todos los infinitos son iguales en tamaño.

Infinitos contables: los conjuntos como los números naturales $\mathbb{N}$ tienen una cardinalidad infinita contable, es decir, se pueden enumerar (cada elemento se puede asociar con un número natural).

Infinitos no contables: el conjunto de los números reales $\mathbb{R}$, por otro lado, tiene una cardinalidad mayor, ya que no se puede enumerar de la misma manera; es un infinito no contable.

Ejemplos:

Si $G = \{1,3,5\}$ y $H = \{a,b,c\}$, entonces $|G| = |H| = 3$.

Si L = {perro, gato} y J = {manzana}, entonces $|L| > |J|$ porque $|L| = 2$ y $|J| = 1$.

Conjuntos especiales

Conjunto vacío

($\emptyset$): representa un conjunto sin elementos, denotado como $\emptyset$. Es el único conjunto que no contiene ningún elemento (Bruckner, 2008).

Conjunto universal

(U): es el conjunto que contiene todos los elementos posibles en un contexto específico. Dependiendo del problema o dominio, U representa el *todo* en el universo de discurso definido (Christensen, 2001).

Propiedades del conjunto vacío y el conjunto universal

1.- Unión con el conjunto vacío:

$A \cup \emptyset = A$: la unión de cualquier conjunto A con el conjunto vacío es A.

Ejemplo:

Si $A = \{1,2,3\}$, entonces $A \cup \emptyset = \{1,2,3\}$.

2.- Intersección con el conjunto vacío:

$A \cap \emptyset = \emptyset$: la intersección de cualquier conjunto A con el conjunto vacío es $\emptyset$.

Ejemplo:

Si A = {1,2,3}, entonces $A \cap \emptyset = \emptyset$.

3.- Unión con el conjunto universal:

$A \cup U = U$: la unión de cualquier conjunto A con el conjunto universal es U.

Ejemplo:

Si U = {1,2,3,4,5} y A = {1,2,3}, entonces $A \cup U$ = {1,2,3,4,5} = U.

4.- Intersección con el conjunto universal:

$A \cap U = A$: la intersección de cualquier conjunto A con el conjunto universal A.

Ejemplo:

Si A = {1,2,3} y U = {1,2,3,4,5}, entonces $A \cap U$ = {1,2,3} = A.

5.-Complemento del conjunto vacío:

$\emptyset^c = U$: el complemento del conjunto vacío es el conjunto universal, ya que todos los elementos posibles están fuera del conjunto vacío.

Ejemplo:

Si el conjunto universal es U = {1,2,3,4,5}, entonces el complemento de $\emptyset$ es U = {1,2,3,4,5}.

6.- Complemento del conjunto universal:

$U^c = \emptyset$: el complemento del conjunto universal es el conjunto vacío, ya que no existen elementos fuera del conjunto universal.

Ejemplo:

Si U = {1,2,3,4,5}, entonces el complemento de U es $\emptyset$ = {}.

La relación con otros conjuntos es con las leyes de Morgan, las cuales nos muestran cómo se relacionan el conjunto vacío y el conjunto universal al trabajar con complementos (Arroyo, 2018).

Ejemplos de leyes de Morgan

Primera ley de Morgan: $(A\cup B)^c = A^c \cap B^c$

Ejemplo:

Si U = {1,2,3,4,5,6}, A = {1,2}, y B = {2,3,4}, entonces:

$A\cup B$ = {1,2,3,4}

$(A\cup B)^c$ = {5,6}

A^c = {3,4,5,6}, B^c = {1,5,6}

$A^c \cap B^c$ = {5,6}, que coincide con $(A\cup B)^c$.

Segunda ley de Morgan: $(A\cap B)^c = A^c \cup B^c$

Ejemplo:

Con los mismos conjuntos A y B del ejemplo anterior:

$A\cap B$ = {2}

$(A\cap B)^c$ = {1,3,4,5,6}

A^c = {3,4,5,6}, B^c = {1,5,6}

$A^c \cup B^c$ = {1,3,4,5,6}, que coincide con $(A\cap B)^c$.

Las propiedades y relaciones ayudan a simplificar operaciones y establecer relaciones de inclusión entre conjuntos.

Conjuntos iguales y desigualdad de conjuntos

Conjuntos iguales

Dos conjuntos A y B son iguales si contienen exactamente los mismos elementos. Esto se representa como A = B. Para que dos conjuntos sean iguales, cada elemento de A debe estar en B, y cada elemento de B debe estar en A. No importa el orden de los elementos ni cómo estén escritos, siempre que el contenido sea idéntico (Coto, s.f.).

Ejemplo:

Si A = {1,2,3} y B ={3,2,1}, entonces A = B, ya que ambos contienen los mismos elementos.

Desigualdad de conjuntos

Dos conjuntos son desiguales cuando no contienen exactamente los mismos elementos, es decir, al menos uno de los conjuntos tiene un elemento que no está en el otro. Esto se representa como $A \neq B$ (Coto, s.f.).

Ejemplo:

Si $A = \{1,2,3\}$ y $B = \{1,2,4\}$, entonces $A \neq B$, ya que B contiene el elemento 4, que no está en A, y A contiene el elemento 3, que no está en B.

Igualdad por extensión e igualdad por intensión

Existen dos formas de definir y comparar conjuntos para determinar si son iguales: igualdad por extensión e igualdad por intensión.

Igualdad por extensión

En la igualdad por extensión, se enumeran explícitamente todos los elementos del conjunto. Si dos conjuntos tienen listados exactamente los mismos elementos, son iguales por extensión (Restrepo, 2003).

Ejemplo:

Si $A = \{1,2,3\}$ y $B = \{1,2,3\}$, entonces $A = B$ por extensión, porque ambos conjuntos contienen exactamente los mismos elementos listados.

Igualdad por intensión

En la igualdad por intensión, se describe a los conjuntos mediante una propiedad o condición que define a todos sus elementos, sin listarlos individualmente. Dos conjuntos son iguales por intensión si cumplen la misma condición.

Ejemplo:

Si $A = \{x \in \mathbb{N} \mid \}$ y $B = \{1,2,3\}$, entonces $A = B$ por intensión, ya que ambos conjuntos se definen por la misma condición (los números naturales menores que 4).

Conjuntos especiales

Los conjuntos especiales son conjuntos que tienen propiedades o características específicas que los hacen únicos o interesantes en el contexto de las matemáticas (Arroyo, 2018). A continuación veremos algunos ejemplos de conjuntos especiales comunes.

Conjuntos finitos

Un conjunto es finito si tiene un número específico y limitado de elementos.

Ejemplo:

$\{1,2,3,4\}$ es un conjunto finito porque contiene solo cuatro elementos.

Otros ejemplos de conjuntos finitos incluyen el conjunto de los días de la semana, los meses del año, o los números naturales del 1 al 10.

Conjuntos infinitos

Un conjunto es infinito si tiene un número ilimitado de elementos, es decir, no tiene un final.

Ejemplo:

El conjunto de los números naturales $\mathbb{N}=\{1,2,3,4,\ldots\}$ es infinito porque no hay un último número en la secuencia.

Distinción entre conjunto infinito numerable y no numerable

Conjuntos infinitos numerables (o contables):

Son aquellos conjuntos infinitos cuyos elementos se pueden poner en correspondencia uno a uno con los números naturales. Esto significa que, aunque el conjunto es infinito, sus elementos pueden enumerarse (Stillwell, 2010).

Ejemplo:

El conjunto de los números naturales $\mathbb{N} = \{1,2,3,\ldots\}$.

El conjunto de los números enteros $\mathbb{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$.

El conjunto de los números racionales $\mathbb{Q}$ (todos los números que pueden expresarse como una fracción de dos enteros).

Conjuntos infinitos no numerables (o no contables)

Son aquellos conjuntos infinitos cuyos elementos no pueden ponerse en correspondencia uno a uno con los números naturales. Esto implica que tienen *más* elementos que un conjunto numerable, en un sentido técnico (Stillwell, 2010).

Ejemplo:

El conjunto de los números reales $\mathbb{R}$, que incluye todos los números racionales e irracionales en la recta numérica.

Un subconjunto notable de los reales es el conjunto de los números irracionales (números que no pueden expresarse como una fracción, como $\sqrt{2}$, π, *e*).

Conjuntos numerables y no numerables

Conjuntos numerables

Un conjunto es numerable (o contable) si sus elementos se pueden enumerar o poner en correspondencia uno a uno con los números naturales. Es decir, si podemos asignar un número natural único a cada elemento del conjunto, el conjunto es numerable (Stillwell, 2010).

Características:

- Puede ser finito o infinito.
- En el caso de los conjuntos infinitos, los numerables tienen el mismo tamaño que el conjunto de los números naturales, en términos de *cantidad* de elementos.

Ejemplos:

Números naturales ($\mathbb{N}$): el conjunto de los números naturales es numerable. $\mathbb{N} = \{1,2,3,4,\ldots\}$.

Números enteros ($\mathbb{Z}$): incluye números positivos, negativos y el cero. $\mathbb{Z} = \{\ldots,-3,-2,-1,0,1,2,3,\ldots\}$.

Es numerable porque se puede organizar en una secuencia que corresponde con los números naturales (por ejemplo, ordenando: $0,1,-1,2,-2,3,-3,\ldots$).

Números racionales ($\mathbb{Q}$): incluye todos los números que se pueden expresar como fracciones de la forma $\frac{a}{b}$, donde a y b son enteros y b ≠ 0.

Aunque pueda parecer que hay más números racionales que naturales, es posible enumerarlos de manera que correspondan a los números naturales. Por lo tanto, $\mathbb{Q}$ es numerable.

Conjuntos no numerables

Un conjunto es no numerable (o no contable) si no es posible contar sus elementos ni ponerlos en correspondencia uno a uno con los números naturales. Esto significa que el conjunto tiene *más* elementos que cualquier conjunto numerable.

Características:

- Siempre es infinito.
- Los elementos de un conjunto no numerable no pueden organizarse de manera que cada elemento corresponda a un número natural.

Ejemplos:

Números reales ($\mathbb{R}$): incluye todos los números en la recta numérica, tanto racionales como irracionales. Existen muchos más números reales que naturales o racionales, y no se pueden enumerar todos en una lista que se corresponda con los números naturales.

Ejemplos de números reales incluyen: $\sqrt{2}$, π, *e*, así como números racionales como $\frac{1}{2}$.

Números irracionales: estos son los números que no pueden expresarse como fracciones, como $\sqrt{2}$, π, *e*, y son un subconjunto de los números reales. Los números irracionales son no numerables, ya que no se pueden poner en correspondencia con los números naturales.

Tabla 1. *Conjuntos y su numerabilidad*

Conjunto	**Símbolo**	**Numerable/No Numerable**	**Ejemplos**
Números naturales	$\mathbb{N}$	Numerable	$\{1,2,3,\ldots\}$
Números enteros	$\mathbb{Z}$	Numerable	$\{\ldots,-1,0,1,\ldots\}$
Números racionales	$\mathbb{Q}$	Numerable	$\{\frac{1}{2}, -\frac{3}{4}, 5\}$
Números reales	$\mathbb{R}$	No numerable	$\{\sqrt{2}, \pi, e\}$
Números irracionales	-	No numerable	$\{\sqrt{3}, \pi\}$

Fuente: elaboración propia con datos de Stilwell (2010).

Conjuntos potencia

Los conjuntos potencia (o conjuntos de partes) son un concepto fundamental en la teoría de conjuntos. Dado un conjunto *A*, el conjunto potencia de *A*, denotado como P(A), es el conjunto de todos los subconjuntos posibles de A, incluyendo el conjunto vacío y el propio conjunto A.

Definición formal

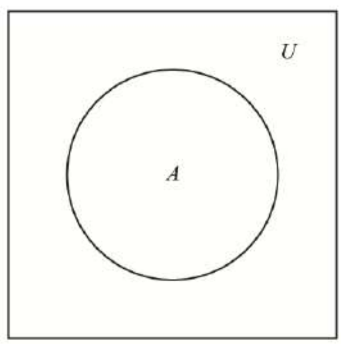
Si A es un conjunto, entonces su conjunto potencia P(A) es el conjunto de todos los subconjuntos de A. Matemáticamente: $P(A) = \{X : X \subseteq A\}$.

Ejemplo:

Sea $A = \{1,2,3\}$ entonces P(A) se compone de los siguientes subconjuntos de:

$A = \emptyset, \{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\},\{1,2,3\} = A$

Figura 10. *Conjunto A*



Fuente: elaboración propia.

Propiedades

1.- Cantidad de elementos: si un conjunto A tiene *n* elementos, su conjunto potencia P(A) tendrá 2^n elementos.

2.- Incluye el conjunto vacío: siempre incluye el conjunto vacío ($\emptyset$) y el propio conjunto A como parte de sus subconjuntos.

Ejemplo

Supongamos que $A = \{1,2\}$. Los subconjuntos posibles de A son:

$\emptyset,\{1\}, \{2\},\{1,2\}$

Por lo tanto, el conjunto potencia P(A) es: $P(A) = \{\emptyset,\{1\},\{2\},\{1,2\}\}$

Ejemplo:

Sea A = [{1, 2, 3}, {4, 5}, {6, 7, 8}].

a) Enumere los elementos de A. El conjunto tiene tres elementos, los conjuntos {1, 2, 3}, {4, 5} y {6, 7, 8}.

b) Encuentre n(A). n(A) = 3.

Aplicaciones de la teoría de conjuntos

En álgebra, los conjuntos desempeñan un rol fundamental como base estructural para definir y entender los conceptos algebraicos. Aquí tienes algunos aspectos clave del rol de los conjuntos en álgebra.

Definición de estructuras algebraicas

Los conjuntos son el marco en el que se construyen las estructuras algebraicas, como grupos, anillos, cuerpos y espacios vectoriales. Estas estructuras son conjuntos equipados con operaciones (como suma o multiplicación) que cumplen ciertas propiedades (Lang, 1976).

Por ejemplo:

Un grupo es un conjunto G con una operación binaria que satisface propiedades como asociatividad, existencia de un elemento neutro y existencia de inversos.

Un cuerpo es un conjunto con dos operaciones (suma y multiplicación) que cumplen reglas adicionales, como la existencia de inversos para la multiplicación (excepto el cero).

Relaciones y funciones

En álgebra, los conjuntos permiten definir relaciones y funciones entre sus elementos, lo que es crucial para establecer conceptos como homomorfismos, isomorfismos y automorfismos. Estas nociones son esenciales para estudiar cómo se comportan y relacionan las estructuras algebraicas entre sí (Artin, 2011).

Propiedades de operaciones

Los conjuntos permiten estudiar las propiedades de las operaciones definidas en ellos, como cerradura, conmutatividad, asociatividad y distributividad. Estas propiedades son fundamentales para caracterizar las operaciones algebraicas y las estructuras asociadas (El Proyecto NROC, s.f.).

Base para álgebra booleana y lógica

El álgebra booleana, que es una rama del álgebra usada en lógica matemática y computación, está profundamente ligada a la teoría de conjuntos. Operaciones como unión, intersección y complemento tienen paralelismos con operaciones lógicas como el "OR", "AND" y "NOT" (González, s.f.).

La teoría de conjuntos tiene una relación estrecha con la lógica matemática, particularmente en el tratamiento de proposiciones y operaciones lógicas.

Cada proposición lógica puede representarse con un conjunto de condiciones que se cumplen o no.

Por ejemplo, la proposición: "Los números mayores que 3".

Conjunto asociado: $A = \{x \in \mathbb{N} : x>3\}$.

Construcción de polinomios y espacios vectoriales

En álgebra, los conjuntos se utilizan para definir elementos como polinomios o vectores (Camino, 2003).

Por ejemplo:

Los polinomios se consideran elementos de un conjunto con operaciones de suma y multiplicación.

Los espacios vectoriales están formados por un conjunto de vectores, que satisfacen axiomas sobre operaciones de suma y multiplicación escalar.

Conjuntos finitos e infinitos

El estudio de conjuntos finitos e infinitos permite explorar propiedades como cardinalidad y bases algebraicas. Esto es esencial, por ejemplo, en la teoría de matrices y determinantes o al analizar si un espacio vectorial tiene dimensión finita o infinita (Fontes, 2020).

Relación entre conjuntos y lógica matemática

Conjuntos como base de la lógica

Los conjuntos permiten representar proposiciones y relaciones lógicas.

Por ejemplo:

Conjuntos universales y subconjuntos: un conjunto universal U puede representar todas las posibilidades, mientras que un subconjunto representa los elementos que cumplen ciertas condiciones lógicas (Arroyo, 2018).

Operaciones entre conjuntos: operaciones como la unión, la intersección y el complemento tienen su equivalente en lógica:

Unión ($A \cup B$) → Disyunción (OR): representa el conjunto de elementos que están en A, B o ambos.

Intersección ($A \cap B$) → Conjunción (AND): representa el conjunto de elementos que están simultáneamente en A y B.

Complemento (A^c) → Negación (NOT): representa los elementos que no están en A.

Relaciones lógicas mediante conjuntos

Los conjuntos ayudan a visualizar conceptos fundamentales de la lógica:

Implicación lógica: si A es subconjunto de B, esto puede interpretarse como $A \Rightarrow B$ (si x pertenece a A, entonces x pertenece a B).

Equivalencia lógica: si $A = B$, entonces ambas proposiciones o conjuntos tienen los mismos elementos, lo cual equivale a una bicondicional ($A \Leftrightarrow B$).

Diagramas de Venn

Los diagramas de Venn son una herramienta gráfica basada en la teoría de conjuntos para representar proposiciones lógicas y sus intersecciones (Arroyo, 2018). Estos diagramas son útiles para:

- Identificar relaciones entre varias proposiciones.
- Representar visualmente la validez de argumentos lógicos.

Conjuntos como interpretación semántica de la lógica

En la lógica matemática, los conjuntos se utilizan para interpretar proposiciones en términos de verdad:

Un conjunto de elementos puede representar todas las interpretaciones para las cuales una proposición es verdadera.

Los modelos lógicos, fundamentales en lógica de primer orden, se construyen usando conjuntos que satisfacen ciertas propiedades (Arroyo, 2018).

Lógica en la teoría de conjuntos

La teoría de conjuntos está formalizada con herramientas de la lógica matemática.

Por ejemplo:

Axiomas de la teoría de conjuntos: Sistemas como los axiomas de Zermelo-Fraenkel (ZF) con el axioma de elección (ZFC) usan lógica para definir las propiedades de los conjuntos (Arroyo, 2018).

Cuantificadores**:** en lógica, los cuantificadores universales ($\forall$) y existenciales ($\exists$) se usan para describir propiedades de conjuntos:

Ejemplo:

-$\forall x \in A$, $P(x)$ significa que $P(x)$ es verdadera para todos los elementos del conjunto A.

-$\exists x \in A, P(x)$ significa que existe al menos un elemento en A para el cual $P(x)$ es verdadera.

Aplicaciones prácticas

La combinación de conjuntos y lógica es esencial para:

Sistemas de computación: conjuntos y lógica forman la base del álgebra booleana y los circuitos lógicos digitales.

Demostraciones matemáticas: las proposiciones y argumentos se representan mediante conjuntos y sus relaciones lógicas.

Teoría de modelos: los conjuntos se usan para interpretar estructuras lógicas complejas, como números, grafos o espacios.

Paradojas y limitaciones de la teoría de conjuntos: introducción a las paradojas

La paradoja de Russell

La paradoja de Russell fue formulada por el matemático y filósofo Bertrand Russell en 1901 y es una de las paradojas más conocidas en la teoría de conjuntos (Arroyo, 2018). Se plantea de la siguiente manera:

Supongamos que existe un conjunto R que contiene todos los conjuntos que no se contienen a sí mismos.

Nos preguntamos: ¿R se contiene a sí mismo?

Si R∈R (es decir, R se contiene a sí mismo), entonces, por definición, R no debería contenerse a sí mismo.

Si R∉R (es decir, R no se contiene a sí mismo), entonces, por definición, debería contenerse a sí mismo.

Esto lleva a una contradicción lógica, lo que muestra que la teoría de conjuntos ingenua (que permite definir cualquier conjunto) tiene inconsistencias.

Impacto de la paradoja de Russell

La paradoja de Russell expone que la teoría de conjuntos ingenua no es un sistema consistente para formalizar las matemáticas. Esto motivó a los matemáticos a desarrollar teorías de conjuntos más rigurosas.

Teoría axiomática de conjuntos (ZF/ZFC)

Se introdujo la teoría de conjuntos de Zermelo-Fraenkel (ZF), que incluye un conjunto de axiomas para evitar paradojas.

El axioma de separación, por ejemplo, limita la posibilidad de definir conjuntos arbitrarios como RRR en la paradoja de Russell.

Teoría de tipos

Propuesta por Russell como una solución, esta teoría organiza los conjuntos en *tipos* jerárquicos para evitar que un conjunto pueda referirse a sí mismo.

La paradoja de Russell y otras similares llevaron al desarrollo de sistemas más rigurosos para las matemáticas, como la teoría axiomática de conjuntos y la lógica formal. Esto muestra cómo es que las paradojas, aunque inicialmente son problemáticas, han impulsado avances significativos en la comprensión y formalización de las matemáticas (Lipschutz, 1991).

Referencias

Arroyo, J. R. (2018). *Lógica y teoría de conjuntos*. Universidad de Costa Rica.

Artin, M. (2011). *Álgebra* (2.ª ed.). Pearson.

Bruckner, A. (2008). *Análisis real elemental*. Prentice-Hall.

Camino, R. L. (2003). *Espacios vectoriales*. Universidad de Granada.

Cantor, G. (2006). *Fundamentos para una teoría general de conjuntos*. Editorial Crítica.

Castañeda Campos, C. (s.f.). *Funciones reales de variable real*. Universidad Nacional de Huancavelica.

Coto, M. N. (s.f.). *Números reales y sucesiones: introducción a la teoría matemática*. Instituto de Matemáticas; Universidad Nacional Autónoma de México.

Christensen, H. (2001). *Estadística paso a paso*. Trillas.

El Proyecto NROC. (s.f.). *Matemáticas de desarrollo*.

Facultad Regional San Francisco. (s.f.). *Funciones*. Universidad Tecnológica Nacional.

Fontes, F. G. (2020). *Conjuntos finitos y conjuntos infinitos*.

González Carlomán, A. (s.f.). *Retículo completo de Boole, lógica matemática, teoría de conjuntos*. Asturias.

Lang, S. (1976). *Álgebra lineal*. Fondo Educativo Interamericano.

Lewin, R. (2011). *La teoría de conjuntos y los fundamentos de la matemática*. J. C. Sáez, ed.

Lipschutz, S. (1991). *Teoría de conjuntos y temas afines*. McGraw-Hill.

Quesada, E. V. (2021). *Matemática discreta con apoyo de software*. Alpha Editorial.

Restrepo, G. (2003). *Fundamentos de las matemáticas*. Universidad del Valle.

Seymour, L., y Lipson, M. (2007). *Matématicas discretas*. Mc Graw Hill.

Smith, S. A. (2000). *Álgebra de conjuntos*. Pearson Educación.

Stillwell, J. C. (2010). *Caminos al infinito: las matemáticas de la verdad y la prueba*. CRC Press.

Villanueva, E. (2015). *Teoría de conjuntos: introducción axiomática*. Universidad Nacional Autónoma de México.

Weisstein, E. W. (1996). *The book of numbers*. Springer-Verlag.

Capítulo 4. Relaciones y funciones

María Salome de la Rosa Olvera[1]
Ma. Lourdes Trejo Calzada[2]
Rosario Vega Rodríguez[3]

DOI: https://doi.org/10.52501/cc.440.04

Resumen

Las relaciones en matemáticas se establecen entre elementos de uno o dos conjuntos, siendo una relación binaria un subconjunto del producto cartesiano. Estas relaciones poseen propiedades fundamentales, como ser reflexivas, irreflexivas, simétricas, antisimétricas, asimétricas y transitivas, las cuales permiten analizarlas y clasificarlas. A partir de estas propiedades surgen tipos importantes, como las relaciones de equivalencia (reflexivas, simétricas y transitivas) y las relaciones de orden (parcial y total), esenciales para estructurar y comparar elementos en distintos contextos matemáticos. Por otra parte, las funciones constituyen un tipo especial de relación en la que a cada elemento del dominio le corresponde un único valor en el codominio. Se clasifican en inyectivas, suprayectivas y biyectivas, según la forma en que se relacionan ambos conjuntos. Además, existen diversos tipos de funciones, como las reales, algebraicas (polinomiales y racionales), trascendentes (trigonométricas, exponenciales y logarítmicas) y especiales (implícitas, inversas, escalonadas e hiperbólicas), cada una con aplicaciones específicas. Las funciones permiten modelar fenómenos en áreas como la física, economía, ingeniería y vida cotidiana, favoreciendo el pensamiento lógico, la abstracción y la toma de decisiones fundamentadas.

[1] Maestra en Administración. Jefa del Laboratorio de Química en Tecnológico Nacional de México. Correo electrónico: salome.ro@itz.edu.mx
[2] Maestra en Ciencias en Enseñanza de las Ciencias. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0009-0007-4761-069X
[3] Maestro en Ciencias en Enseñanza de ciencias básicas con especialidad en física. Docente en Tecnológico Nacional de México.

Palabras clave: *relaciones matemáticas, propiedades de las relaciones, funciones.*

Propiedades de relaciones

Las **relaciones** en matemáticas (particularmente en álgebra y teoría de conjuntos) se establecen entre los elementos de un conjunto o entre dos conjuntos. Una **relación binaria** R sobre un conjunto A es un subconjunto del producto cartesiano A× A \times AA × A.

Estas relaciones pueden tener diferentes **propiedades** que permiten clasificarlas y comprender su estructura.

Propiedades principales de las relaciones binarias

Reflexiva

Una relación R sobre A es reflexiva si **todo elemento se relaciona consigo mismo**.

$$\forall a \in A, (a,a) \in R$$

Ejemplo: la relación “ser igual a” en los números reales.

Irreflexiva

Una relación es irreflexiva si **ningún elemento se relaciona consigo mismo**.

$$\forall a \in A, (a, a) \notin R$$

Ejemplo: la relación “ser menor que” en los números reales.

Simétrica

Una relación es simétrica si siempre que un elemento está relacionado con otro, el segundo también lo está con el primero.

$$\forall\ a, b \in A, (a, b) \in R \implies (b, a) \in R$$

Ejemplo: la relación “ser hermano de”.

Antisimétrica

Una relación es antisimétrica si, cuando dos elementos se relacionan en ambos sentidos, entonces deben ser iguales.

$$\forall a, b \in A, (a, b) \in R \text{ y } (b, a) \in R \implies a = b$$

Ejemplo: la relación “ser menor o igual que” (≤).

Asimétrica

Una relación es asimétrica si, cuando un elemento se relaciona con otro, el inverso nunca ocurre.

$$\forall a,b \in A,\ (a, b)\in R \Longrightarrow (b, a)\notin R$$

Ejemplo: la relación "ser padre de".

Transitiva

Una relación es transitiva si siempre que un elemento está relacionado con un segundo y el segundo lo está con un tercero, entonces el primero está relacionado con el tercero.

$$\forall a,b,c\in A,\ (a,b)\in R \text{ y } (b,c)\in R \Longrightarrow (a,c)\in R$$

Ejemplo: la relación "ser mayor que".

Clasificación de relaciones según propiedades

- **Relación de equivalencia**: cumple **reflexiva, simétrica y transitiva**.
  Ejemplo: "tener la misma edad".
- **Relación de orden parcial**: cumple **reflexiva, antisimétrica y transitiva**.
  Ejemplo: "ser menor o igual que".
- **Relación de orden total**: es un orden parcial en el que todos los elementos son comparables.
  Ejemplo: el orden en los números reales con $\leq$.

Características de las funciones

El concepto de función es uno de los pilares más importantes en la matemática, ya que permite establecer relaciones entre variables y modelar fenómenos de la vida real. De acuerdo con Zill y Wright (2011), una función puede definirse como una regla que asocia a cada elemento de un conjunto de entrada un único elemento de salida. Esta definición sencilla es la base para comprender cómo se aplican las funciones en diversos campos, desde la física y la economía hasta la informática y la biología.

Comprender las funciones no se limita a memorizar fórmulas, sino a analizar cómo se comportan y de qué manera reflejan procesos reales. Así, el estudio de funciones se convierte en una herramienta de pensamiento crítico, modelación y resolución de problemas

(Martínez, 2017). A continuación, se desarrolla un análisis de los principales tipos de funciones, sus características y sus aplicaciones prácticas.

Las funciones se denotan generalmente como f:A→B, donde A es el dominio y B es el codominio, y para cada a∈A existe un único b∈B tal que f(a) = b.

Una función es un tipo especial de relación entre dos conjuntos, donde a cada elemento del conjunto de partida (o dominio) le corresponde exactamente un elemento del conjunto de llegada (o codominio); (Arroyo et al, 2018).

En términos más formales, una función f de un conjunto A a un conjunto B se define como una relación tal que para cada a∈A existe un único b∈B tal que $f(a) = b$. Las funciones se denotan generalmente como f:A→B, donde A es el dominio y B es el codominio.

Dominio: el conjunto de todos los elementos de partida A.

Codominio: el conjunto B donde llegan los valores de la función.

Rango o imagen: el conjunto de los elementos de B que son imágenes de los elementos de A.

Ejemplo:

Si A = {1,2,3,4} y B = {a,b,c,d}, y sea f = {(1,a),(2,a),(3,d),(4,c)}

Entonces F es una función, ya que ningún elemento de A aparece como primer elemento de dos pares ordenados diferentes por lo que se tiene: $f(1) = a$,$f(2) = a$, $f(3) = d$, $f(4) = c$.

Esta función es una correspondencia en la que a cada elemento del dominio A le corresponde uno y solo un elemento en el codominio B (Castañeda, s.f.). El codominio de f, Cod(f) = {a,d,c}. En algunas ocasiones se escribirá el codominio de la función f como f(A). Obsérvese que el dominio a∈B aparece como segundo elemento de dos diferentes pares ordenados en f. Esto no causa conflicto con la definición de una función. Por lo tanto, la función puede tomar el mismo valor en dos elementos diferentes de A.

Tipos de funciones

Funciones por la relación del dominio por el codominio

Las funciones pueden clasificarse según cómo se relacionan los elementos del dominio con los del codominio (Universidad Tecnológica Nacional, Facultad Regional San Francisco, s.f). Los tres tipos principales son la inyectiva, la suprayectiva y la biyectiva.

Función inyectiva (inyección)

Una función f:A→B es inyectiva si asigna diferentes elementos del dominio a diferentes elementos del codominio. Por tanto, diferentes valores de A dan diferentes valores en B. Es decir, si $f(a1)=f(a2)$ implica que a1=a2.

Ejemplo:

La función f(x) = 2x en el conjunto de los números reales R es inyectiva, ya que diferentes valores de x dan diferentes valores de f(x).

Función suprayectiva (sobreyectiva)

Una función f:A→B es suprayectiva si cada elemento del codominio tiene al menos un elemento del dominio que se le asigna. Por lo que todos los valores de B están cubiertos por valores de A. Es decir, para cada b∈B existe al menos un a∈A tal que f(a) = b.

Ejemplo:

La función f(x) = x^2 en el conjunto de los números reales $\mathbb{R}$ no es suprayectiva sobre $\mathbb{R}$, ya que no todos los números reales son cuadrados perfectos (por ejemplo, no hay un x real tal que $x^2 = -1$). Sin embargo, si restringimos el codominio a los números reales no negativos, entonces f(x) = x^2 sí es suprayectiva en ese conjunto.

Función biyectiva (biyección)

Una función f:A→B es biyectiva si es a la vez inyectiva y suprayectiva. En otras palabras, cada elemento de A se asocia con un único y diferente elemento de B, y todos los elementos de B están cubiertos. Por lo que cumple las propiedades de inyectiva y suprayectiva. Las funciones biyectivas también se conocen como correspondencias uno a uno.

Ejemplo:

La función f(x) = x+1 en el conjunto de los números enteros $\mathbb{Z}$ es biyectiva, ya que cada entero se asigna a otro único entero, y todos los enteros están representados en el codominio (Villanueva, 2015).

Esta clasificación se debe a la forma en que relacionan los elementos del dominio con el codominio. Una función inyectiva es aquella en la que no existen dos elementos distintos del dominio que tengan la misma imagen. En contraste, una función suprayectiva asegura que cada elemento del codominio tenga al menos un elemento en el dominio que lo relacione. Finalmente, una función biyectiva combina ambas propiedades: es inyectiva y suprayectiva a la vez, por lo que establece una correspondencia biunívoca entre los dos conjuntos (Zill y Wright, 2011).

Estas distinciones tienen aplicaciones importantes en áreas como la informática, en donde se requiere asegurar que cada dato tenga una correspondencia única dentro de un sistema de codificación. Según López y Hernández (2020), "las funciones biyectivas son esenciales en el diseño de algoritmos criptográficos, pues garantizan que cada mensaje codificado tenga una única interpretación posible" (p. 60).

Asimismo, la comprensión de estas funciones permite entender mejores conceptos como la existencia de funciones inversas, herramienta clave en el álgebra y el cálculo.

Funciones reales de variable real

Una función real de variable real es aquella en la que tanto el dominio como el rango pertenecen al conjunto de los números reales. Esto significa que a cada valor real de la variable independiente x le corresponde un valor real de la variable dependiente y (Zill y Wright, 2011). La representación gráfica de estas funciones en el plano cartesiano facilita la comprensión de su comportamiento.

Por ejemplo, la función lineal $f(x) = mx + b$ describe relaciones proporcionales, como la velocidad constante en un movimiento rectilíneo. De igual forma, las funciones cuadráticas permiten representar trayectorias parabólicas, muy comunes en fenómenos físicos como el lanzamiento de proyectiles (Mora y Castillo, 2018).

Desde la educación, este tipo de funciones es fundamental, pues permite a los estudiantes relacionar de forma concreta los conceptos matemáticos con situaciones reales. En palabras de Perdomo (2020), "la enseñanza de las funciones reales favorece el desarrollo del pensamiento lógico y la abstracción matemática" (p. 92).

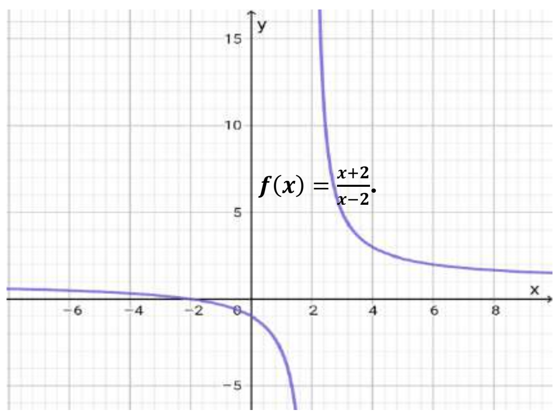
Funciones algebraicas: polinomiales y racionales

Las funciones algebraicas se dividen en **polinomiales** y **racionales**. Las primeras se definen por expresiones en las que la variable aparece elevada a potencias enteras no negativas. Así, funciones lineales, cuadráticas, cúbicas y de mayor grado son ejemplos de polinomiales. Su importancia radica en que modelan fenómenos con gran precisión, desde la trayectoria de un objeto hasta el análisis de poblaciones (Mendoza y Rodríguez, 2019).

Las funciones racionales, en cambio, se expresan como el cociente de dos polinomios. Este tipo de funciones son frecuentes en la física y la química, ya que describen relaciones de proporcionalidad inversa, como la ley de gravitación universal o la intensidad de una reacción química. De acuerdo con Cárdenas (2019), "las funciones racionales permiten estudiar procesos en los que una variable depende inversamente de otra, siendo esenciales en la economía y la administración" (p. 80).

En la educación, estas funciones representan un reto didáctico, pero también una oportunidad para fortalecer la capacidad de abstracción y análisis en los estudiantes.

Gráfica 1. *Ejemplo de una función racional*



Fuente: elaboración propia.

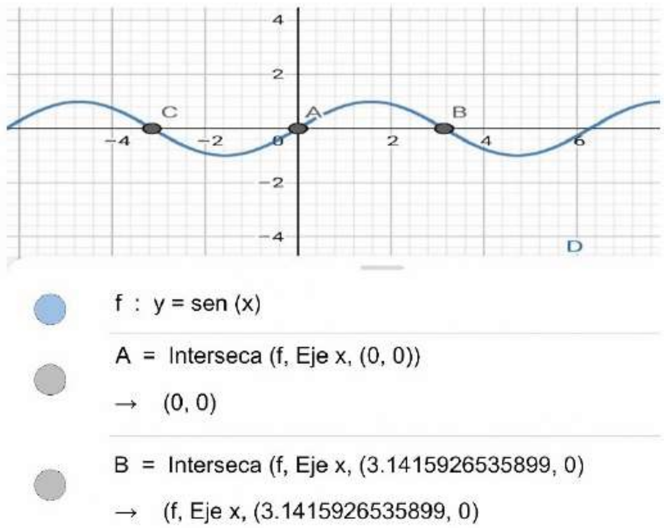
Funciones trascendentes: trigonométricas, exponenciales y logarítmicas

Más allá de las algebraicas, las **funciones trascendentes** incluyen aquellas que no pueden expresarse como polinomios. Dentro de este grupo se encuentran las trigonométricas, exponenciales y logarítmicas.

Funciones trigonométricas

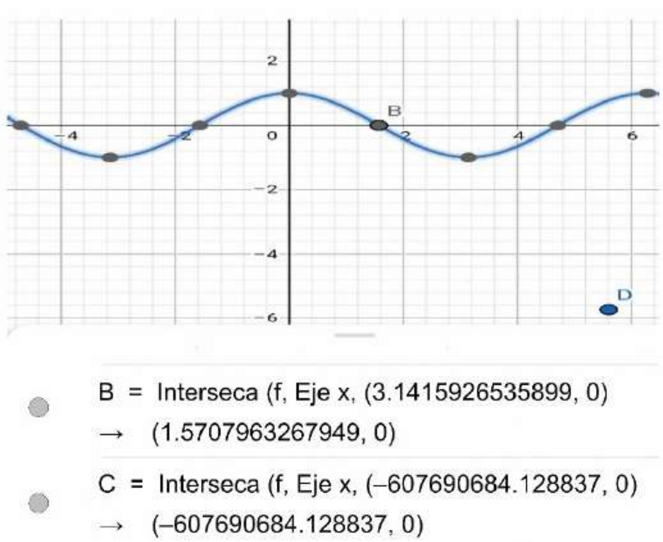
Son esenciales en la geometría, la física y la ingeniería. Permiten describir fenómenos periódicos como las ondas sonoras o la propagación de la luz (Salas y Pineda, 2021). La función seno, por ejemplo, modela el movimiento armónico simple.

Gráfica 2. *Función seno*



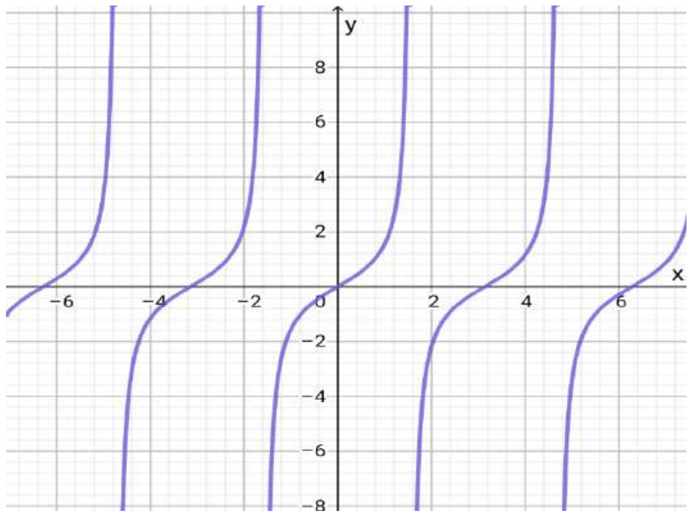
Fuente: elaboración propia.

Gráfica 3. *Función coseno*



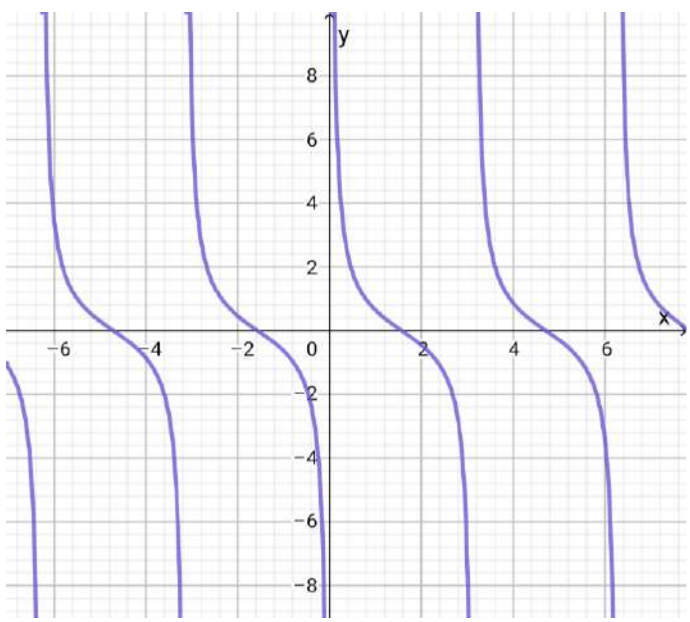
Fuente: elaboración propia.

Gráfica 4. *Función tangente*



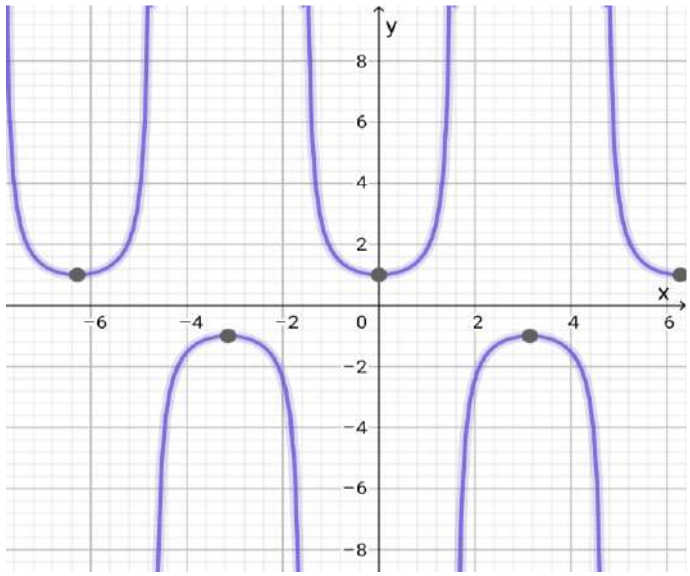
Fuente: elaboración propia.

Figura 5. *Cotangente*



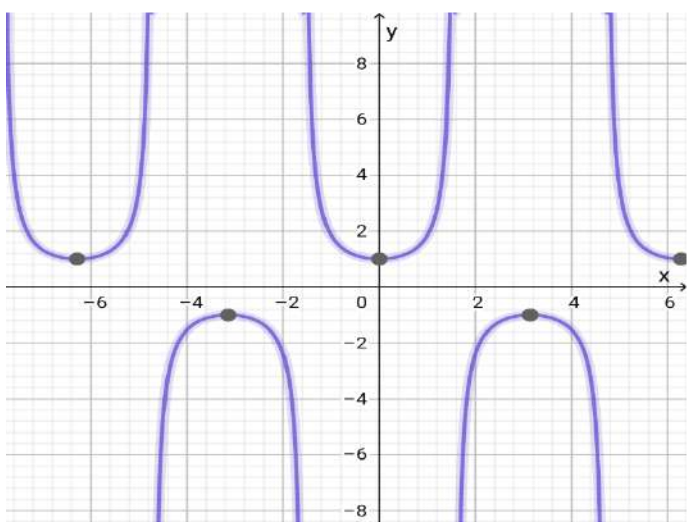
Fuente: elaboración propia.

Gráfica 6. *Secante*



Fuente: elaboración propia.

Gráfica 7. *Cosecante*

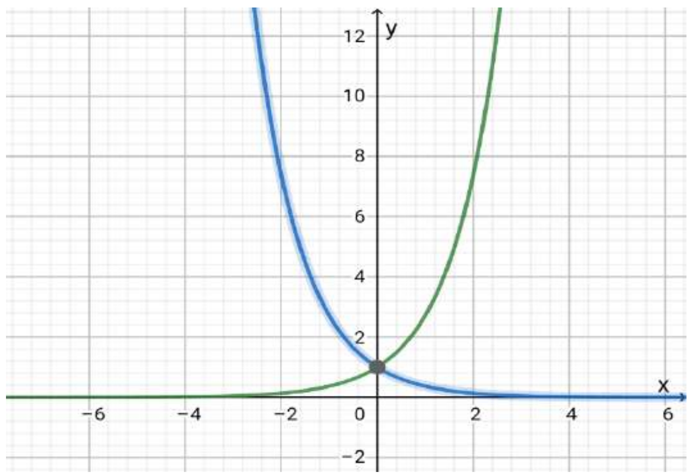


Fuente: elaboración propia.

Funciones exponenciales

Describen fenómenos de crecimiento o decrecimiento acelerado, como la expansión de una población, el interés compuesto en finanzas o la desintegración radiactiva (Martínez, 2017).

Gráfica 8. *Función exponencial*

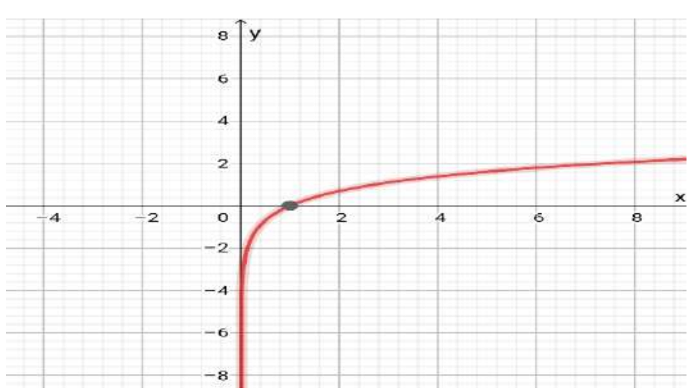


Nota: $f(x) = e^{x}$ *está de color verde y*
$h(x) = e -^{x}$ *se encuentra de color azul.*
Fuente: elaboración propia.

Funciones logarítmicas

Son la inversa de las exponenciales y se aplican en la medición de fenómenos que crecen de manera rápida, pero que necesitan ser expresados en escalas más manejables, como la magnitud de los terremotos en la escala de Richter (Mora y Castillo, 2018).

Gráfica 9. *Función logarítmica*



Fuente: elaboración propia.

En conjunto, las funciones trascendentes reflejan la riqueza de las matemáticas para interpretar fenómenos complejos y diversos.

Funciones especiales: escalonadas, implícitas, inversas e hiperbólicas

Además de las funciones clásicas, existen otras con propiedades particulares. Las **funciones escalonadas**, como la de Heaviside, representan sistemas de encendido y apagado, se utilizan ampliamente en la ingeniería eléctrica y en los sistemas de control. Por otro lado, las **funciones implícitas** permiten representar relaciones en las que no es posible despejar una variable de manera directa, como en el caso de las circunferencias o elipses.

Las **funciones implícitas** son aquellas en las que la relación entre las variables **no aparece despejada de manera explícita**, es decir, la variable dependiente *y* no está aislada en un lado de la ecuación como ocurre con las funciones explícitas.

En lugar de eso, la relación se presenta mediante una **ecuación que involucra a las dos variables** al mismo tiempo.

Ejemplo de función explícita:

Una función explícita tiene la forma: $y = f(x) \Rightarrow y = 2x + 3$

Aquí la variable dependiente *y* está aislada, y para cada valor de *x* se obtiene directamente un valor de *y*.

Ejemplo de función implícita:

Una relación como: $x2 + y2 = 25$

representa una circunferencia de radio 5.

- Aquí no aparece despejado *y*.
- Para obtener *y* en función de *x*, tendríamos que manipular la ecuación:

$y = \pm 25 - x2$

Esto muestra que la relación original define a *y* de forma implícita.

Usos de las funciones implícitas:

- En **geometría**, describen curvas como circunferencias, elipses o hipérbolas.
- En **cálculo**, se estudian para derivar expresiones cuando no es posible o no conviene despejar *y*. De ahí surge la **derivación implícita**.

- En **modelación matemática**, muchas veces los fenómenos no se expresan con una sola variable aislada, sino en relaciones más complejas que requieren esta forma.

Las **funciones inversas** son fundamentales, pues permiten "deshacer" el efecto de una función dada. Por ejemplo, la función logarítmica es la inversa de la exponencial. Finalmente, las **funciones hiperbólicas** tienen aplicaciones en la física relativista y en la teoría de hipérbolas, modelando comportamientos que no pueden explicarse con funciones trigonométricas (Zill y Wright, 2011).

De acuerdo con Ramírez (2021), "la diversidad de funciones matemáticas refleja la diversidad de los fenómenos que buscan modelarse, desde los más simples hasta los más complejos" (p. 50).

Conclusión

El análisis de las funciones matemáticas permite comprender que no son únicamente herramientas abstractas, sino que tienen un fuerte vínculo con la realidad. Desde la representación de trayectorias y el análisis de ondas hasta la predicción de fenómenos económicos, las funciones constituyen un lenguaje universal de las ciencias.

La importancia de este tema, como señalan Zill y Wright (2011), radica en que permite al estudiante y al investigador crear modelos matemáticos que simplifican la realidad sin perder su esencia. Por ello, su estudio es imprescindible en la educación matemática y en la formación científica de cualquier disciplina.

Aplicación de las funciones

Las funciones constituyen un concepto central en las matemáticas y tienen un amplio rango de aplicaciones en distintas disciplinas. En el ámbito de las ciencias naturales, permiten modelar fenómenos físicos como el movimiento, el crecimiento poblacional o la propagación de ondas, donde las relaciones entre variables dependen de expresiones funcionales (Mora y Castillo, 2018).

En la economía y administración, las funciones son fundamentales para representar costos, ingresos, utilidades y tendencias de mercado, lo que facilita la toma de decisiones estratégicas y la predicción de escenarios (Cárdenas, 2019).

En la ingeniería y la tecnología, se aplican en el diseño de estructuras, en la programación de sistemas automatizados y en el análisis de datos experimentales, ya que permiten establecer correspondencias entre variables de entrada y salida (López y Hernández, 2020).

En el campo de la educación, su enseñanza desarrolla competencias en resolución de problemas, abstracción y modelación matemática, habilidades necesarias para comprender fenómenos complejos en la vida cotidiana y en la ciencia (Ramírez, 2021).

Finalmente, en la vida diaria, las funciones se manifiestan en situaciones como el cálculo de intereses bancarios, el consumo de energía, la conversión de divisas o el análisis de gráficos estadísticos que orientan decisiones personales y sociales (Mora y Castillo, 2018).

Referencias

Arroyo Hernández, J., Ramírez Jiménez, J., y Sequeira Chavarría, F. (2018). *Lógica y teoría de conjuntos*. Editorial Universidad Nacional.

Cárdenas, J. (2019). Aplicaciones de las funciones matemáticas en economía y gestión empresarial. *Revista de Ciencias Económicas, 37*(2), 77-92.

Castañeda Campos, C. (s.f.). *Funciones reales de variable real*. Universidad Nacional de Huancavelica.

Facultad Regional San Francisco. (s.f.). *Funciones*. Universidad Tecnológica Nacional.

López, G., y Hernández, F. (2020). Funciones matemáticas y su impacto en la ingeniería aplicada. *Revista Mexicana de Ingeniería y Tecnología, 12*(1), 55-70.

Martínez, J. (2017). *Fundamentos de lógica y matemáticas*. Universidad Nacional Autónoma de México.

Mendoza, F., y Rodríguez, L. (2019). Aplicaciones de la lógica matemática en la computación. *Revista Colombiana de Matemáticas Aplicadas, 11*(2), 45-58.

Mora, A., y Castillo, P. (2018). Modelación de fenómenos naturales mediante funciones. *Revista Colombiana de Educación Matemática, 6*(2), 101-118.

Perdomo, C. (2020). El papel de la lógica matemática en la formación del pensamiento crítico. *Revista Latinoamericana de Educación, 14*(27), 89-104.

Ramírez, S. (2021). La enseñanza de las funciones como herramienta para el desarrollo del pensamiento matemático. *Revista Iberoamericana de Educación Matemática, 15*(29), 45-63.

Salas, M., y Pineda, R. (2021). Lógica, modelación y sociedad: aplicaciones contemporáneas. *Revista Iberoamericana de Filosofía y Ciencia, 8*(1), 33-52.

Villanueva, E. (2015). *Teoría de conjuntos: introducción axiomática*. Universidad Nacional Autónoma de México.

Universidad Tecnológica Nacional. (s.f.). *Funciones* [Material didáctico]. Universidad Tecnológica Nacional. https://sanfrancisco.utn.edu.ar/

Zill, D. G., y Wright, W. S. (2011). *Cálculo de una variable: trascendentes tempranas* (4.ª ed.). McGraw-Hill.

Capítulo 5. Lógica

Edgar Esaúl Saucedo Becerra[1]

DOI: https://doi.org/10.52501/cc.440.05

Resumen

La lógica matemática es una disciplina fundamental que estudia las estructuras del razonamiento válido mediante el uso de símbolos y reglas formales. Su propósito no es definir qué pensar, sino cómo hacerlo correctamente, favoreciendo la coherencia y evitando contradicciones. Este campo tiene aplicaciones en áreas diversas, como la informática, la filosofía y la inteligencia artificial, ya que fortalece habilidades de análisis, abstracción y resolución de problemas. Históricamente, surge con Aristóteles y evoluciona con aportaciones clave de Boole y Frege, quienes consolidaron la lógica simbólica moderna. En este contexto, la lógica proposicional analiza proposiciones, entendidas como enunciados con valor de verdad (verdadero o falso). Las proposiciones pueden ser simples o compuestas, y se relacionan mediante conectores lógicos, como conjunción, disyunción, negación, condicional y bicondicional. Las tablas de verdad permiten evaluar todas las combinaciones posibles de estos valores, lo cual facilita la verificación de la validez de los argumentos. Además, existe una relación entre lógica y teoría de conjuntos, donde operaciones como unión, intersección y complemento corresponden a operadores lógicos. En conjunto, estos elementos constituyen una base esencial para el pensamiento formal y la argumentación rigurosa.

Palabras clave: *lógica matemática, proposiciones, conectores lógicos.*

Proposiciones, conectores lógicos y tablas de verdad

La lógica matemática constituye un pilar esencial del pensamiento formal, y su estudio permite desarrollar habilidades para analizar, construir y evaluar argumentos con rigor. A través del uso de símbolos y estructuras formales, la lógica ofrece herramientas que trascienden las

---

[1] Maestro en Tecnología Educativa. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0000-0002-3114-2197 ; correo electrónico: edsaucedo@gmail.com

matemáticas e impacta en disciplinas como la informática, la lingüística, la filosofía, el derecho y la inteligencia artificial.

Esta disciplina tiene como objeto principal el estudio del razonamiento válido. Su propósito no es determinar qué se debe pensar, sino cómo pensar correctamente, es decir, de manera coherente, estructurada y sin contradicciones. En palabras de Mendelson (2010), “la lógica matemática es la aplicación del método matemático al estudio de los principios del razonamiento válido” (p. 1).

En este capítulo se presentan los fundamentos de la lógica proposicional y de predicados: los conceptos de proposición, verdad, conectores lógicos, tablas de verdad, inferencias válidas y reglas de deducción. A través del análisis de estos elementos, el estudiante adquiere las bases necesarias para representar argumentaciones en lenguaje simbólico y evaluar su validez mediante métodos formales.

El estudio de la lógica matemática no solo contribuye a la formación intelectual del universitario, sino que fortalece su capacidad de argumentación, toma de decisiones y resolución de problemas, habilidades cada vez más valoradas en contextos académicos, científicos y profesionales.

Lógica

La lógica matemática, también conocida como lógica simbólica o formal, se ocupa del estudio de los sistemas formales que permiten representar el razonamiento mediante símbolos, reglas y estructuras precisas, facilitando la demostración de teoremas y la validación de argumentos. En palabras de Mendelson (2010) , “es la aplicación del método matemático al estudio de los principios del razonamiento válido” (p. 1).

Así, la lógica matemática es una disciplina que estudia las estructuras del pensamiento formal, lo que permite analizar la validez de los razonamientos mediante símbolos y reglas precisas. Es fundamental en áreas como las matemáticas, la computación y la filosofía, ya que ayuda a desarrollar habilidades de análisis, abstracción y resolución de problemas (Jiménez, 2014).

¿Dónde y cuándo surgió?

La lógica surgió como disciplina formal en la antigua Grecia con Aristóteles (384-322 a. C.), quien desarrolló el silogismo como una forma de razonamiento deductivo. Sin embargo, la lógica matemática moderna nace en el siglo XIX, con el trabajo de George Boole en Inglaterra y Gottlob Frege en Alemania. George Boole publicó *The laws of thought* en 1854, donde desarrolló el álgebra booleana, base para la lógica digital. Por su parte, Gottlob Frege publicó *Begriffsschrift* en 1879, el cual se considera el primer sistema formal completo de lógica matemática.

En América Latina, su enseñanza y aplicación han cobrado fuerza en universidades y centros de investigación, especialmente en el contexto de la educación matemática y la informática (Farré et al., 2013).

Principales aportes históricos

- Aristóteles (siglo IV a.C.): fundador de la lógica clásica.
- George Boole (1854): lógica algebraica.
- Gottlob Frege (1879): primer sistema lógico formal.
- Bertrand Russell y Alfred North Whitehead (1910-1913): *Principia mathematica*, obra clave para fundamentar las matemáticas en la lógica.
- Kurt Gödel (1931): teoremas de incompletitud, límites de los sistemas formales.

Características

La lógica se divide en 5:

1. Proposicional
2. Modelos
3. Demostración
4. Conjuntos
5. Recursión

En este capítulo se toma el estudio de la lógica proposicional. Su objetivo es cuestionar los conceptos y las reglas de deducción que se utilizan en las matemáticas.

La lógica estudia las reglas de deducción formales y el pensamiento lógico se aplica en diferentes áreas. En las ciencias computacionales, la lógica proposicional sirve para verificar si se escriben correctos los programas, ya que los programas solo hacen lo que se les ponen a realizar.

Una proposición lógica es un enunciado declarativo al que puede asignarse un valor de verdad, es decir, puede determinarse si es verdadero o falso. Las proposiciones constituyen la base del razonamiento lógico y del análisis formal utilizado en la lógica matemática y en las demostraciones deductivas (Arroyo et al., 2018; Copi et al., 2016, p. 24).. Un ejemplo de ello es el conjunto de valores de verdad que satisfacen una proposición.

Características de una proposición lógica

- Debe ser un enunciado claro y preciso.
- No debe ser ambiguo.
- Debe tener un valor de verdad único: verdadero (V) o falso (F).

Ejemplos de proposiciones lógicas simples:

Proposición: p: “El número 5 es mayor que 3”.

Esta proposición es verdadera porque 5 es mayor que 3.

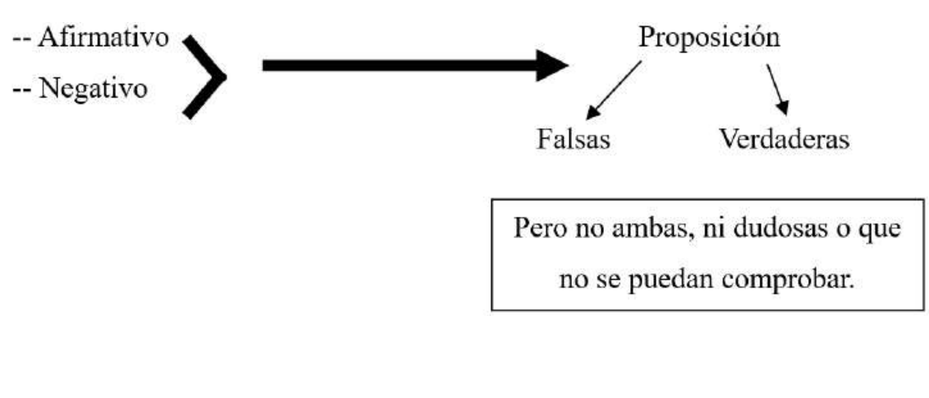
Proposición: q: “El número 2 es par”.

Esta proposición también es verdadera.

Lógica proporcional

Una proposición es un enunciado declarativo. Un enunciado es una frase de dos o más palabras o una expresión que puede ser numérica. Los tipos de enunciados son los siguientes:

- Imperativo: es una orden.
- Interrogativo: es una pregunta.
- Exclamativo: son frases con emoción.
- Declarativo:



Representación con conjuntos

Las proposiciones lógicas pueden representarse como conjuntos:

Sea A = $\{x \in \mathbb{N} : x > 3\}$, el conjunto de números mayores que 3.

Sea B = $\{x \in \mathbb{N} : x \text{ es par}\}$, el conjunto de números pares.

Conjunción: $A \cap B$: representa los números mayores que 3 y que también son pares.

Disyunción: $A \cup B$: representa los números que son mayores que 3 o que son pares.

Complemento de A: $\neg A$: representa los números que no son mayores que 3.

Clasificación de proposiciones lógicas

Proposiciones simples: son aquellas que no contienen conectores lógicos.

Ejemplo: "Hoy es lunes."

Proposiciones compuestas: son aquellas que combinan dos o más proposiciones simples mediante conectores lógicos como "y" ($\wedge$), "o" ($\vee$), "no" ($\neg$), "si... entonces" ($\rightarrow$).

Ejemplo: "Hoy hace frío y está lloviendo".

En los enunciados de la tabla 1 se evalúa si es una proposición, si es verdadera o falsa y si solo es enunciado y de qué tipo es:

Tabla 1. *Tipos de proposiciones*

	Enunciado	Tipo
1	El Tecnológico de Zacatecas está vacío de alumnos los domingos.	Declarativo, afirmativo, proposición verdadera.
2	Los marcianos nos vigilan.	Declarativo, afirmativo, no proposición porque no podemos comprobarlo
3	Ve a comprarme dos boletos para el cine.	Imperativo, no proposición.
4	3 + 4 = 5	Declarativo, afirmativo, proposición, falsa.
5	Thalía esposa de Tomy motola nació en 1869.	Declarativo, afirmativo, proposición, falsa.
6	Todos los números tienen un número de valor mayor.	Declarativo, afirmativo, proposición, verdadero
7	Los números primos se pueden dividir solo entre sí mismos y la unidad.	Declarativo, afirmativo, proposición, verdadero
8	¿Tienes novio(a)?	Interrogativo, no proposición
9	6 > 9	Declarativo, afirmativo, proposición, falsa.
10	¡Uff, qué calor!, ¡hay que sudor!	Exclamativo, no proposición

Fuente: elaboración propia.

La lógica estudia el razonamiento, analiza si el razonamiento es correcto, analiza las relaciones entre los enunciados mas no el contenido o la veracidad del contenido, sin embargo, si los dos primeros enunciados fueran verdaderos la lógica garantiza el tercero (Johnsonbaught, 1999).

Operaciones lógicas: las operaciones sobre conjuntos (unión, intersección, complemento) tienen paralelos en la lógica:

- Unión: equivalente a una disyunción lógica ("o").
- Intersección: representa una conjunción lógica ("y").
- Complemento: relacionado con la negación lógica ("no").

Estas conexiones permiten usar la teoría de conjuntos como una herramienta para formalizar sistemas lógicos.

Conexión con operaciones lógicas

1. Conjunción lógica ("y"):

   Proposición combinada: $p \wedge q$.

   Traducción: "El número 5 es mayor que 3 y el número 2 es par."

   Esto es verdadero, ya que ambas proposiciones, p y q, son verdaderas.

2. Disyunción lógica ("o"):

   Proposición combinada: $p \vee r$.

   Traducción: "El número 5 es mayor que 3 o el número 7 es divisible por 2."

   Esto es verdadero, porque al menos una de las proposiciones (ppp) es verdadera.

3. Negación ("no"):

   Negación de r: $\neg r$.

   Traducción: "El número 7 no es divisible por 2."

   Esto es verdadero, porque rrr es falsa.

4. Implicación ("si...entonces"):

   Proposición implicativa: $p \rightarrow q$.

   Traducción: "Si el número 5 es mayor que 3, entonces el número 2 es par."

   Esto es verdadero, porque si la primera proposición (p) es verdadera, la segunda (q) también lo es.

Conectores lógicos

Para hablar utilizamos conectores como “y” u “o”, a continuación se describen los operadores más usados con las proposiciones.

Conjunción o AND

Es equivalente a “y”, es la unión de dos o más proposiciones, por lo que se convierte en una proposición compuesta. El símbolo utilizado para este operador es: ^. Se utiliza V para verdadero y F para falso.

Ejemplo: si el contenido de las proposiciones es como se indica en la tabla

Proposición	Contenido	Valor de verdad
p	la casa es blanca	V
q	los alumnos son inteligentes	V

Por su parte, las proposiciones compuestas serían del siguiente modo:

p ^ q = la casa es blanca y los alumnos son inteligentes. V

Nota: las dos proposiciones tienen que ser verdaderas para que el resultado sea verdadero.

Ejemplo:

Proposición	Contenido	Valor de verdad
r	hoy es lunes	F
s	los lunes siempre llueve	F

r^s = hoy es lunes y los lunes siempre llueve. F

La tabla de verdad que contiene las combinaciones ente dos proposiciones es la siguiente para el operador conjunción.

Tabla 2. *Tabla del operador conjunción*

P	Q	P^Q
V	V	V
V	F	F
F	V	F
F	F	F

El segundo operador es el operador disyunción.

Disyunción o OR

Es la composición de dos o más proposiciones con el operador "o" y se denota como v.

Ejemplo: si se tiene las proposiciones anteriores se pueden unir con "v".

Proposición	**Contenido**	**Valor de verdad**
p	la casa es blanca	V
q	los alumnos son inteligentes	V

p v q = la casa es blanca y los alumnos son inteligentes.   V

Nota: con una proposición que sea verdadera el resultado es verdadero.

La tabla de disyunción que contiene las combinaciones entre dos proposiciones es la siguiente:

Tabla 3. *Disyunción*

P	Q	P v Q
V	V	V
V	F	V
F	V	V
F	F	F

No o not

El tercer operador es el operador negación, lo que realiza es contradecir la proposición. Para este operador solo es necesaria una proposición. Para este operador hay diferentes símbolos $\bar{P}$ ¬P ~P

Si la proposición es:

P = Hoy es lunes

~P = Hoy no es lunes

Tabla 4. *Not o negación*

P	$\bar{P}$ ¬P ~P
V	F
F	V

Condicional

El cuarto operador es la condicional, es una proposición compuesta de dos proposiciones. Se lee si p entonces q.

Tabla 5. *Condicional*

P	Q	P→Q	Q→P
V	V	V	V
V	F	F	V
F	V	V	F
F	F	V	V

Cuando la primera proposición es verdadera y la segunda falsa el resultado es falso. Todas las demás opciones son verdaderas.

Bicondicional

El quinto operador es el bicondicional. Proposición compuesta de al menos 2 proposiciones. P si solo si q.

Tabla 6. *Bicondicional*

P	Q	P←→Q
V	V	V
V	F	F
F	V	F
F	F	V

Ejercicios:

Con las proposiciones de la siguiente tabla se presentan algunos ejercicios:

Proposición	Contenido	Valor de verdad
R	3>5	F
S	5>2	V
Q	Hay un tornado en la escuela	F
P	Esta nevando	F

Ejercicios de sustitución con frase

S→R = Si 5>2 entonces 3>5. F

V →F = F

R→Q = Si 3>5 entonces hay un tornado en la escuela. V

Q→P = Si hay un tornado en la escuela entonces está nevando. V

S↔ R= 5>2 si solo si 3>5. F

V↔F = F

Q↔ P = Hay un tornado en la escuela si solo si está nevando. V

Sustitución sin el valor frase de la proposición

Suponiendo que p y r son falsas y que q, s y t son verdaderas determine los valores de verdad de cada una de las siguientes proposiciones:

Proposición	Valor de verdad
P	F
R	F
Q	V
S	V
T	V

P = F R = F Q = V S=V T = V

Ejemplo 1:

$p \rightarrow q = F \rightarrow V = V$

Pasos: primero sustituir con los valores de verdad, luego hacer operaciones desde los paréntesis más internos, hasta obtener el valor de verdad final.

Ejemplo 2:

$\neg p \rightarrow \neg q = \neg F \rightarrow \neg V = V \rightarrow F = F$

Ejemplo 3:

$\neg(p \rightarrow q) = \neg(F \rightarrow V) = \neg(V) = F$

Evaluando expresiones

Para evaluar expresiones se tienen que sacar los valores de verdad de cada una de las proposiciones individuales, luego ha que sustituirlas por el valor de verdad y los operadores, y después se realiza la operación.

En los siguientes ejercicios represente la afirmación de manera simbólica dado que p: 4<2 F q: 7<10 V r: 6<6 F y encontrar el valor de verdad.

Datos	**Simbología**
P: 4<2 F Q: 7<10 V R: 6<6 F S: 23+3 = 7 F T: Hoy es lunes F	
1.- Si 4<2 entonces 7< 10	P → q = F→V = V
2.- Si (4<2 y 6<6), entonces 7<10	(P^R) → Q = (F^F) → V = V
3.- Si no es cierto que (6< 6 y 7 no es menor que 10), entonces 6<6.	¬ (R^¬Q) → R = ¬ (F^¬V) → F = ¬ (F^F) → F = V → F = F
4.- 7<10 si y solo si (4<2 y 6 no es menor que 6)	Q ←→ (P ^ ¬R) = V ←→ (F ^¬F) V ←→ F = F
5. Si (7<10 y 6<6 y 23+3=7 y 4<2) entonces 7<10	(Q^R^S^P) → Q (V^F^F^F) → V F → V = V
6. Hoy es lunes si solo si (no es cierto que (si 4<2 entonces 6<6) y (23+3=7 o 7<10)	T ↔ ¬(p → r) ^ (s v q) = F ↔ ¬(F → F) ^ (F v V) = F ↔ ¬(V) ^(V) = F ↔ F^V = F ↔ F = V
7. No es cierto que(6<6 y 4<2) si solo si (si 7 <10 entonces 6<6)	¬(r^p) ↔ (q → r) = ¬(F^F) ↔ (V → F)= ¬(F) ↔ (F) = V ↔ F = F

Tablas de verdad

Una **tabla de verdad** es una herramienta fundamental en lógica matemática que permite representar todas las posibles combinaciones de verdad (verdadero o falso) de una o más proposiciones, y mostrar el resultado lógico de una expresión o enunciado compuesto.

2^n núm. de proposiciones = núm. de renglones = $2^{3 \text{ prop}}$= 8 renglones: la mitad verdaderos y la mitad falsos. En este ejemplo 4 verdaderos y 4 falsos.

Ejemplos:

Para sacar la tabla de ((P^Q) v R)→P se observa que solo hay 3 proposiciones que son P, Q y R. Se tiene que respetar el orden alfabético.

1

P	**Q**	**R**	**P ^ Q**	$\overline{\mathbf{(P \wedge Q)}}$	$\mathbf{(P \wedge Q)\overline{\vee R}}$	$\mathbf{((P \wedge Q)\overline{\vee R)} \rightarrow P}$
V	V	V	V	F	V	V
V	V	F	V	F	F	V
V	F	V	F	V	V	V
V	F	F	F	V	V	V
F	V	V	F	V	V	F
F	V	F	F	V	V	F
F	F	V	F	V	V	F
F	F	F	F	V	V	F

SOLUCION

2

p	**q**	**p→q**
V	V	V
V	F	F
F	V	V
F	F	V

2^n número de proposiciones = número de renglones = $2^2 = 4$

$\neg p \rightarrow \neg q$

3

p	Q	¬p	¬q	¬p → ¬q
V	V	F	F	V
V	F	F	V	V
F	V	V	F	F
F	F	V	V	V

4

p	q	(p→q)	¬(p→q)
V	V	V	F
V	F	F	V
F	V	V	F
F	F	V	F

Equivalencias lógicas y leyes de inferencia

Equivalencias lógicas

Las **equivalencias lógicas** son relaciones entre proposiciones que, aunque se escriban de manera distinta, **significan lo mismo** (tienen las mismas tablas de verdad).

Ejemplo:

$p \rightarrow q \equiv \neg p \vee q \equiv \neg p \vee q$

Esto significa que decir "si p entonces q" es lo mismo que decir "no p o q".

Ejemplo en lenguaje cotidiano

- Proposición: “Si estudio, entonces apruebo”.
- Equivalente: “No estudio o apruebo”.

  Ambas expresiones siempre tienen el mismo valor de verdad.

1. Demuestra con una tabla de verdad que:

   $\neg(p \wedge q) \equiv \neg p \vee \neg q$

   (Ley de Morgan)

2. Reescribe en forma equivalente:
   - “Si hace frío, entonces llevo chamarra”.
   - Usando p = “Hace frío”, q = “Llevo chamarra”.

¿Qué son las leyes de inferencia?

Las **leyes de inferencia** son reglas que nos permiten **deducir nuevas proposiciones verdaderas** a partir de proposiciones conocidas como verdaderas. Es decir, nos dicen cómo **razonar lógicamente**.

Nombre de Equivalencia	
Doble negación	$\neg(\neg P) \equiv P$
Conmutativa	$P \wedge Q \equiv Q \wedge P$; $P \vee Q \equiv Q \vee P$
Asociativa	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$
Distributiva	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
De Morgan	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$; $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
Condicional	$P \to Q \equiv \neg P \vee Q$
Bicondicional	$P \leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$

Leyes de inferencia

1. Aplica Modus Ponens:
   - Si estudio, entonces apruebo.
   - Estudié.
   - ¿Qué concluyes?

2. Aplica *modus tollens*:
   - Si corro, entonces sudo.
   - No sudé.
   - ¿Qué concluyes?

3. Identifica la regla usada:
   - Si hoy es lunes, entonces hay clase.
   - Hoy es lunes.
   - Conclusión: hay clase.

Lógica de predicados y cuantificadores

Lógica de predicados

La lógica de predicados, también conocida como lógica de primer orden, es una extensión de la lógica proposicional que permite representar afirmaciones más complejas mediante el uso de **predicados**, **variables** y **cuantificadores**. Esta lógica es fundamental en matemáticas, filosofía, informática y lingüística, ya que permite formalizar razonamientos sobre objetos y sus propiedades (Miquel, 2019).

Elementos de la lógica de predicados

Predicados

Un predicado es una función que asigna un valor de verdad a una o más variables. Por ejemplo, el predicado P(x) puede representar "x es par". Los predicados pueden tener una o más variables, y se combinan con conectivos lógicos como $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.

Variables y términos

Las **variables** representan elementos del dominio de discurso. Los **términos** pueden ser variables, constantes o funciones aplicadas a otros términos. Por ejemplo, en la aritmética de Peano, el término s(s(0)) representa el número 2 (Miquel, 2019).

Cuantificadores

Los **cuantificadores** permiten expresar afirmaciones generales o particulares sobre los elementos del dominio.

Cuantificador universal (∀)

Expresa que una propiedad se cumple para todos los elementos del dominio.

- Ejemplo:

$\forall x \in N$, $x+0 = x$ ("Para todo número natural x, x más 0 es igual a x").

Cuantificador existencial (∃)

Indica que existe al menos un elemento que cumple cierta propiedad.

- Ejemplo:

$\exists x \in Z$, x2=4 ("Existe un número entero cuyo cuadrado es 4").

Existencia única (∃!)

Afirma que existe **exactamente un** elemento que cumple la propiedad.

- Ejemplo:

$\exists! x \in R$, $x = 0$ ("Existe un único número real igual a cero").

Negación de cuantificadores

Negar una proposición con cuantificadores implica cambiar el tipo de cuantificador:

- $\neg(\forall x\ P(x)) \equiv \exists x\ \neg P(x)$
- $\neg(\exists x\ P(x)) \equiv \forall x\ \neg P(x)$

Ejemplo:

Negar: "Todos los estudiantes aprobaron"

Se convierte en: "Existe al menos un estudiante que no aprobó".

Fórmulas abiertas y cerradas

Una **fórmula abierta** contiene variables libres, mientras que una **fórmula cerrada** tiene todas sus variables ligadas por cuantificadores. Las fórmulas cerradas pueden evaluarse como verdaderas o falsas en un modelo (Miquel, 2019).

Ejemplos prácticos

1. **Universal**:

$\forall x \in N, x+1>x$ ("Todo número natural es menor que su sucesor").

2. **Existencial**:

$\exists x \in Z, x2 = 9$ ("Existe un número entero cuyo cuadrado es 9").

3. **Negación**:

$\neg(\forall x \in R, x2 \geq 0) \equiv \exists x \in R, x2<0$

Conclusión

La lógica de predicados permite una representación más rica y precisa del razonamiento matemático que la lógica proposicional. Gracias a los cuantificadores y predicados, es posible formalizar teorías completas como la aritmética de Peano o la teoría de conjuntos de Zermelo-Fraenkel, fundamentales en el desarrollo de las matemáticas modernas (Miquel, 2019).

Aplicación de la lógica

La lógica matemática constituye una herramienta fundamental en distintos campos del conocimiento. En el ámbito de la **informática**, se utiliza para el diseño de algoritmos, lenguajes de programación y circuitos digitales, ya que permite establecer estructuras formales para la resolución de problemas computacionales (Mendoza y Rodríguez, 2019).

En las **ciencias exactas**, la lógica facilita la construcción de demostraciones rigurosas y el análisis de estructuras abstractas, asegurando la validez de los razonamientos matemáticos (Martínez, 2017).

En la **educación**, su enseñanza contribuye al desarrollo del pensamiento crítico, la argumentación y la capacidad de análisis de los estudiantes, al favorecer la identificación de falacias y la construcción de razonamientos sólidos (Perdomo, 2020). Asimismo, fomenta competencias metacognitivas que fortalecen la resolución de problemas en distintos contextos.

Finalmente, en el campo de la **filosofía y las ciencias sociales**, la lógica matemática se aplica en la modelación de procesos de decisión, la teoría de juegos y la inteligencia artificial, lo que refleja su carácter transversal y su impacto en la sociedad contemporánea (Salas y Pineda, 2021).

Referencias

Arroyo Hernández, J., Ramírez Jiménez, J., y Sequeira Chavarría, F. (2018). *Lógica y teoría de conjuntos*. Editorial Universidad Nacional.

Boole, G. (1854). *An investigation of the laws of thought*. Macmillan.

Copi, I. M., Cohen, C., y McMahon, K. (2016). *Introducción a la lógica* (14.ª ed.). Pearson.

Farré, J., Jiménez, J., y Rodríguez, M. (2013). *La lógica matemática como base del pensamiento analítico para las matemáticas y la computación*. Universidad Abierta y a Distancia de México.

Frege, G. (1879). *Begriffsschrift. Verlag von Louis Nebert.*

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik, 38*, 173-198.

Jiménez, J. (2014). *Aplicaciones de la lógica en la computación y las matemáticas*. Universidad Nacional Autónoma de México.

Johnsonbaugh, R. (1999). *Matemáticas discretas* (4.ª ed.). Editorial Pearson.

Martínez, J. (2017). *Fundamentos de lógica y matemáticas*. Universidad Nacional Autónoma de México.

Mendelson, E. (2010). *Introducción a la lógica matemática* (5.ª ed.). Reverté.

Mendoza, F., y Rodríguez, L. (2019). Aplicaciones de la lógica matemática en la computación. *Revista Colombiana de Matemáticas Aplicadas, 11*(2), 45-58.

Miquel, A. (2019). Teorías y modelos: una introducción a la lógica de primer orden. *Publicaciones Matemáticas del Uruguay, 17*, 195-224. https://pmu.uy/pmu17/pmu17-0195.pdf

Perdomo, C. (2020). El papel de la lógica matemática en la formación del pensamiento crítico. *Revista Latinoamericana de Educación, 14*(27), 89-104.

Russell, B., y Whitehead, A. N. (1913). *Principia Mathematica* (Vol. I–III). Cambridge University Press.

Salas, M., y Pineda, R. (2021). Lógica, modelación y sociedad: aplicaciones contemporáneas. *Revista Iberoamericana de Filosofía y Ciencia, 8*(1), 33-52.

Capítulo 6. Álgebra booleana

Pedro Tomas Ortiz y Ojeda[1]
Pedro Alfonso Guadalupe Ortiz Sánchez[2]
Patricia Guadalupe Sánchez Iturbe[3]

DOI: https://doi.org/10.52501/cc.440.06

Resumen

En el campo de los sistemas de control se distinguen dos categorías fundamentales: los servomecanismos y los sistemas de conmutación, cada uno con funciones específicas en el manejo de sistemas dinámicos. Los servomecanismos son dispositivos que integran un motor, sensores y un sistema de control con retroalimentación, permitiendo posicionar con precisión un actuador. Su funcionamiento se modela mediante ecuaciones diferenciales, lo que posibilita analizar y predecir su comportamiento ante distintas condiciones. Por otro lado, los sistemas de conmutación se enfocan en establecer y mantener conexiones entre dispositivos, basándose en principios de lógica formal para gestionar estados de encendido y apagado, esenciales para la comunicación y coordinación entre componentes. Estos sistemas operan con señales binarias, es decir, valores discretos representados como 0 y 1, lo que da origen a los circuitos lógicos. Este enfoque se fundamenta en el álgebra booleana, desarrollada por George Boole, quien estableció que los procesos lógicos pueden representarse matemáticamente mediante variables binarias. Posteriormente, Claude Shannon aplicó estos principios al diseño de circuitos digitales, consolidando su relevancia en la informática y la electrónica. El álgebra booleana se basa en funciones lógicas elementales como AND (Y), OR (O) y NOT (NO), que permiten modelar condiciones y decisiones. Asimismo, incluye postulados, teoremas y propiedades que facilitan la simplificación de expresiones lógicas. Herramientas como

[1] Doctor en Matemática Educativa. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0000-0002-3796-8504 ; correo electrónico: ptoyomx@yahoo.com
[2] Doctor en Administración. Docente en el Tecnológico Nacional de México. ORCID: https://orcid.org/0000-0002-2466-1837
[3] Doctora en Ciencias y Biotecnología de Plantas. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0000-0001-9245-3725

las tablas de verdad permiten verificar resultados al analizar todas las combinaciones posibles de variables. En conjunto, estos elementos constituyen la base del diseño y análisis de sistemas digitales y de control.

Palabras clave: *álgebra booleana, propiedades del álgebra booleana, funciones lógicas.*

Fundamentos y propiedades

En el ámbito de los sistemas de control, se identifican esencialmente dos categorías fundamentales que cumplen funciones específicas en el control de sistemas dinámicos. La primera categoría se refiere a la teoría de los servomecanismos. Un servomecanismo es un dispositivo que integra un motor, un sistema de control y un sensor de retroalimentación, cuya función es posicionar con precisión un eje o actuador. Este tipo de sistema se basa en una relación matemática descrita a través de ecuaciones diferenciales, las cuales modelan la dinámica del sistema y permiten predecir su comportamiento y respuesta ante diferentes condiciones de entrada.

La segunda categoría se enfoca en los sistemas de conmutación, cuyo propósito es establecer y mantener interconexiones efectivas entre dispositivos. A diferencia de los servomecanismos, los sistemas de conmutación dependen de las leyes de la lógica y del razonamiento lógico, que permiten gestionar los estados de conexión y desconexión entre distintos elementos del sistema, facilitando así la comunicación y coordinación entre ellos. La implementación de estos sistemas requiere un análisis estructurado de los principios de la lógica formal para asegurar la integridad y continuidad en el flujo de información y control entre los dispositivos interconectados.

En conjunto, estas dos categorías abordan aspectos complementarios en el control de sistemas, contribuyendo tanto a la precisión en la acción de control como a la eficiencia en la comunicación y coordinación entre dispositivos autónomos o interdependientes. En los sistemas de conmutación las informaciones o señales de entrada y salida varían de modo discontinuo o, al menos, así se pueden considerarse. Cada una toma dos valores distintos, entonces los dispositivos y señales se les conoce como binarios. Debido a su principio de funcionamiento los circuitos que forman parte de este sistema, reciben el nombre de circuitos lógicos pues su funcionamiento es muy similar a el raciocinio de la lógica filosófica (Barceló, 2012; Jasso y Rivlin, 2023).

Esta lógica se conoce como álgebra proposicional o álgebra booleana, en honor a George Boole (1815-1864), filósofo y matemático inglés que dio un fuerte impulso a las leyes del razonamiento lógico. Sus obras fundamentales, tituladas *Un análisis matemático de la lógica* e *Investigación de las leyes del pensamiento*, establecen que mediante leyes matemáticas los procesos lógicos del pensamiento, relacionados con la verdad o la falsedad, son expresados en proposiciones o enunciados, y pueden considerarse como variables binarias (Boole, 2023; Enciclopedia Herder, 2023).

Un siglo después, Claude E. Shannon desarrolló las bases para la teoría del diseño de circuitos digitales mediante la relación lógica numérica de los circuitos electrónicos con el álgebra booleana (Enciclopedia Herder, 2023).

Con el uso de funciones elementales y una serie de fórmulas fue posible desarrollar un álgebra en la que se aplican solamente dos valores posibles: verdad o falsedad, a las que se les llama variables binarias, y que en los procesos conmutativos tienen una importancia capital, pues un contacto puede estar abierto o cerrado, una bobina puede estar alimentada o sin tensión, o un dispositivo electrónico pueda conducir energía o no, en fin, existen una gran cantidad de elementos en dos estados posibles.

Se puede considerar que el estado de cada uno de elementos mencionados es una variable binaria en la que se puede asignar un valor de 1 a un contacto cerrado y 0 a un contacto abierto. Así la expresión del contacto si está cerrado será considerado como $A = 1$, y si está abierto entonces $A = 0$.

Para desarrollar un lenguaje básico en el álgebra booleana es necesario considerar tres funciones clásicas, llamadas funciones lógicas elementales.

La función "Y"

Esta función implica la satisfacción de dos o más condiciones para lograr un resultado final, como símbolo se pueden usar un punto, paréntesis o el signo ×, que indica producto o simplemente AB. Si consideramos las variables A y B con sus respectivos valores tenemos:

A	B	A·B
0	0	0
0	1	1
1	0	1
1	1	1

La función "O"

Se considera que indica dos o más alternativas para lograr un mismo resultado, se representa por el signo +, que corresponde a la suma. Tomando los valores las variables A y B los valores de 0 y 1 tenemos:

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

La función "NO"

Es la llamada función complemento, se considera como la negación de los dos estados posibles. Así, para los valores de 0 y 1 tenemos:

A	B	$\bar{A}$	$\bar{B}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	0

En el álgebra de Boole, la negación de una variable o función lógica recibe el nombre de **complemento** y se representa mediante una barra horizontal sobre la variable. Por ejemplo, el complemento de Ase expresa como $\bar{A}$. Esta operación lógica indica el valor opuesto de la variable original: si $A = 1$, entonces $\bar{A} = 0$; y si $A = 0$, entonces $\bar{A} = 1$.

El complemento también puede aplicarse a expresiones booleanas completas mediante las leyes de De Morgan. En términos generales, el complemento de una suma lógica $(+)$ se transforma en un producto lógico $(\cdot)$, mientras que el complemento de un producto lógico se transforma en una suma lógica. Asimismo, el complemento de 1 es 0 y el complemento de 0 es 1. Otra propiedad importante es la **ley de involución**, la cual establece que el complemento del complemento de una variable devuelve la variable original:

$$\bar{\bar{x}} = x$$

Por ejemplo, el complemento de la expresión:

$$1 \cdot A + \bar{B}C + 0$$

se obtiene aplicando las leyes de complemento y las leyes de De Morgan, dando como resultado:

$$(0 + \bar{A})(B + \bar{C}) \cdot 1$$

El álgebra de Boole constituye una herramienta fundamental en la lógica matemática y en el diseño de sistemas digitales, ya que permite representar y simplificar funciones lógicas utilizadas en circuitos electrónicos y sistemas computacionales (Riofrio y Rodríguez, 2023).

Generalizando, se considera que la presencia de algo recibe el valor de 1 y su ausencia recibe 0, así se pueden establecer los siguientes postulados del álgebra booleana (Barceló, 2012; Lifeder, 2023):

a) $1 \cdot 1 = 1$	a') $0 + 0 = 0$
b) $1 \cdot 0 = 0 \cdot 1 = 0$	b') $0 + 1 = 1 + 0 = 1$
c) $0 \cdot 0 = 0$	c') $1 + 1 = 1$
d) $\bar{1} = 0$	d') $\bar{0} = 1$
e) $\bar{\bar{1}} = 1$	e') $\bar{\bar{0}} = 0$

Con el uso de estos postulados se pueden desarrollar teoremas que simplifican las expresiones.

Se llama el dual de una expresión booleana al proceso de obtención del complemento, en el que no se complementan las variables y es utilizado como una herramienta matemática para ampliar ciertos teoremas y para efectuar simplificaciones. De manera que si se tiene una función que sea igual a uno, el hecho de obtener el dual de la función no implica que dicha función sea cero, como en el caso del complemento, si no que puede ser igual a uno a cero. Así, por ejemplo, el dual de $1 \cdot A + \bar{B}C + 0$ sería $(0+A)\ (\bar{B}+C) \cdot 1$

Teoremas y sus duales

1.a) $0\cdot X = 0$	1.b) $1 + X = 1$
2.a) $1\cdot X = X$	2.b) $0 + X = X$
3.a) $X\cdot X = X$	3.b) $X + X = X$
4.a) $X\cdot \bar{X} = 0$	4.b) $X + \bar{X} = 1$
5.a) $XY = YX$	5.b) $X + Y = Y + X$
6.a) $X\cdot Y\cdot Z = X\cdot(Y\cdot Z) = (X\cdot Y)\, Z$	6.b) $X + Y + Z = X + (Y + Z) = (X + Y) + Z$
7.a) $\overline{\bar{\bar{x}}\cdot \bar{\bar{y}}\cdot z} = \bar{X} + \bar{Y} + \cdots + \bar{Z}$	7.b) $\overline{\bar{\bar{x}} + \bar{\bar{y}} + \cdot}\ \overline{+z} = \bar{X}\,\bar{Y}\cdot\bar{Z}$
8 $\overline{\bar{F}(X, Y, \cdots, Z,, +,)} = F(\bar{X},\bar{Y},\cdots,\bar{z}, ,+,\cdot)$	8. $F(X,Y) = X + Y$
9.a) $XY + XZ = X\,(Y + Z)$	9.b) $(X + Y)\,(X + Z) = X + YZ$
10.a) $XY+X\bar{Y} = X$	10.b) $(X + Y)(X +\bar{Y}) = X$
11.a) $X+XY = X$	11.b) $X(X + Y) = X$
12.a) $X+\bar{X}Y = X+Y$	12.b) $X\,(\bar{X} + Y) = XY$
13.a) $XY + \bar{X}Z + YZ = XY + \bar{X}Z$	3.b) $(X + Y)(\bar{X} + Z\,)(Y + Z) = (X + Y)\,(\bar{X} + Z)$
14.a) $XY + \bar{X}Z = (X + Z)\,(\bar{X} + Y)$	14b) $(X + Y)(\bar{X} + Z) = XZ + \bar{X}Y$

Para demostrar un teorema o comprobar una igualdad se deben probar todas las posibilidades de los estados de las variables, esto es lo que se llama una *tabla de verdad*, es un esquema en el cual se enlistan sistemáticamente los valores de las variables independientes y los subsecuentes valores de las variables dependientes, en donde las posibles combinaciones de un sistema binario son 2^n donde n es el número de variables independientes.

Ejemplos.

1. Realice la tabla de verdad del teorema 2a.

Solución

Puesto que se tiene una variable tenemos $2^1 = 2$ el número de renglones o filas será de dos.

X	$1\cdot X = X$
1	$1\cdot 1=1$
0	$1\cdot 0=0$

2. Use la tabla de verdad para comprobar $A + AB + A\overline{C} = A$

   Solución

   Puesto que se tiene una variable tenemos $2^3 = 8$ el número de renglones o filas será de ocho.

A	B	C	AB	$\overline{C}$	$A\overline{C}$	$A+AB+A\overline{C}$
0	0	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	0	1	0	0
1	0	0	0	1	1	1
1	1	0	1	1	1	1
0	1	1	0	0	0	0
1	0	1	0	0	0	1
1	1	1	1	0	0	1

3. Aplique los teoremas 7a (teorema de de Morgan) a la expresión: $\overline{A \cdot B \cdot C \cdot Z} = \overline{A} + \overline{B} + \overline{C} + \overline{Z}$

   Sea: A = B = C = Z = 1, entonces: $\overline{1 \cdot 1 \cdot 1 \cdot 1} = \overline{1} + \overline{1} + \overline{1} + \overline{1}$

   $\overline{1} = 0 + 0 + 0 + 0$

   $0 = 0$

4. Simplifique: (A + B) (A + C)

   Solución:

   AA + AC + BA + BC

   Del teorema 3a: AA = A

   A + AC + BA + BC A(1 + C + B) + BC

   Del teorema 1b:

   (1 + C + B) = 1

   A (1) +BC = A + BC

Formas canónicas

Una expresión como: $(A + \overline{B} + C + D)\,(A + B + \overline{C} + D)$ está formada por dos términos y cada término contiene todas las variables, entonces se puede decir que una forma canónica es aquella que está compuesta por una suma o multiplicación de dos o más términos, los cuales contienen varias literales, en donde los términos están formados por todas las variables, complementadas o sin complementar. También la expresión: $A\overline{B}CD\overline{E} + A\overline{B}\,\overline{C}\,\overline{D}E + \overline{A}B\overline{C}DE$ es canónica.

Minitérminos

Cuando se tiene una expresión canónica cualquiera y se desea expresar como una suma de productos, entonces a cada uno de los términos se le llama *minitérminos*, si una función no esta en forma canónica, al pasarla como una expresión en forma de minitérminos se aplica el teorema 10a invertido. Así, si X puede indicar una o más variables y Y solo una variable entonces:

$$X = X(Y + \overline{Y}) = XY + X\overline{Y}$$

Donde: $X = A\overline{B}C$ y $Y = D$

$$A\overline{B}C = A\overline{B}C(D + \overline{D}) = A\overline{B}CD + A\overline{B}C\overline{D}$$

Maxitérminos

Se considera *maxitérminos* a una expresión canónica que está formada por producto de sumas, es decir:

$$X = (X + Y)\,(Y + \overline{Y})$$

Donde: $X = A + \overline{B} + \overline{C}$ y $Y = D$

$$A + \overline{B} + \overline{C} = (A + \overline{B} + \overline{C} + D)\,(A + \overline{B} + \overline{C} + \overline{D})$$

Simplificación de funciones booleanas

Existen básicamente los siguientes métodos básicos de simplificación:

1.- Método de simplificación algebraico, que se realiza aplicando las leyes y los teoremas del álgebra de Boole.

2.- Métodos tabulares y gráficos.

A) Mapa de Karnaugh. Es aplicable para funciones de dos a cinco variables.

B) Tablas de Quine-Mc Cluskey. Es aplicable para funciones de cinco o más variables.

Mapa de Karnaugh

Este método, conocido también como el método de mapas, matrices lógicas, es un método gráfico para simplificar y representar a las expresiones booleanas partiendo del reconocimiento usual de los implicantes o connotantes primos.

Se considera que el método requiere que las expresiones booleanas se encuentren en forma canónica, se desarrollan cuadros o casillas a cada uno de los minitérminos, es decir a cada una de las posibles combinaciones. Para el caso de una expresión de dos variables es necesario $2^2 = 4$ casillas, de manera que cada minitérmino tendrá solo un cuadro, que se ocupará un determinado lugar en el arreglo de los cuatro cuadros o casillas (González, 2016).

Ejemplo:

1. Construya el mapa que representa la siguiente función: F = A + $\bar{A}$B + $\bar{A}\bar{B}\bar{C}$

		AB			
		00	01	11	10
C	0			1	1
	1			1	1

= A

		AB			
		00	01	11	10
C	0	1			
	1				

= $\bar{A}$B

		AB			
		00	01	11	10
C	0	1			
	1				

= $\bar{A}\bar{B}\bar{C}$

		AB			
		00	01	11	10
C	0	1	1	1	1
	1		1	1	1

= A + $\bar{A}$ B + $\bar{A}\bar{B}\bar{C}$

2. Construya el mapa que representa la siguiente función: F = $\bar{x}\bar{y}$ + $\bar{z}$ + x$\bar{y}$z

		XY			
		00	01	11	10
Z	0	1	1	1	1
	1	1		1	1

3. Obtenga la forma canónica en función de minitérminos de la siguiente expresión: F = $\bar{A}\bar{B} + \bar{C}$

AB

C	00	01	11	10
0	1			
1	1			

$= \bar{A}\bar{B}$

C	00	01	11	10
0	1	1	1	1
1				

$= \bar{C}$

AB

C	00	01	11	10
0	1	1	1	1
1	1			

$= \bar{A}\bar{B} + \bar{C}$

De la lectura de la tabla por fila: F = 000 + 010 + 110 + 100 + 001 entonces:

F = $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}$C

4. Obtenga la forma canónica por dos métodos de: F = A + $\bar{A}\bar{B}$C + $B\bar{C}$

Primer método

F = A(B + $\bar{B}$)(C + $\bar{C}$) + $\bar{A}\bar{B}$C + ($A + \bar{A}$) $B\bar{C}$

F = (AB + A$\bar{B}$)(C + $\bar{C}$) + $\bar{A}\bar{B}$C + $AB\bar{C} + \bar{A}B\bar{C}$

F = ABC + AB$\bar{C}$ + A$\bar{B}$C + $A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + AB\bar{C} + \bar{A}B\bar{C}$

Eliminando: $AB\bar{C}$

La forma canónica es F = ABC + AB$\bar{C}$ + A$\bar{B}$C + $A\bar{B}\bar{C} + \bar{A}\,\bar{B}\bar{C} + \bar{A}B\bar{C}$

Segundo método. Usando los mapas:

	AB	00	01	11	10
C	0		1	1	1
	1	1		1	1

De la lectura de la tabla anterior:

F = $ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$

Simplificación de funciones

Como se dijo líneas arriba, los mapas de Karnaugh se utilizan para simplificar ecuaciones booleanas.

Ejemplos

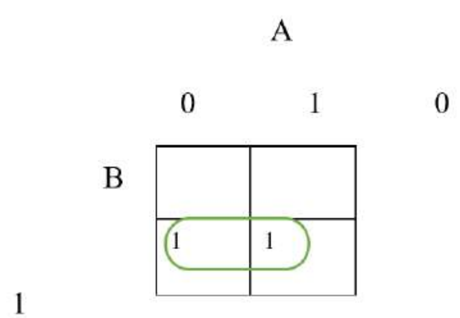
1. Simplifique la siguiente expresión: F = $\bar{A}B + AB$

Primer método:

F = $B(\bar{A} + A)$ pero: $\bar{A} + A = 1$

Así: F = B

Segundo método:



Los términos de los unos adyacentes, ya sea en forma horizontal o vertical, de manera que sea constante o común a los dos valores contiguos, será el resultado de la reducción: F = B, término de la variable que permanece constante.

2. A partir del mapa, halla la función que le corresponda.

a.-

		A	
		0	1
B	0		
	1	1	1

F = B

b.-

		A	
		0	1
B	0		1
	1		1

F = A

c.-

		A	
		0	1
B	0	1	
	1	1	

F = $\bar{A}$

d.-

		A	
		0	1
B	0	1	1
	1		

F = $\bar{B}$

3. Reduzca: F = $\bar{A}\,\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C$

C \ AB	00	01	11	10
0	1	1		1
1	1			

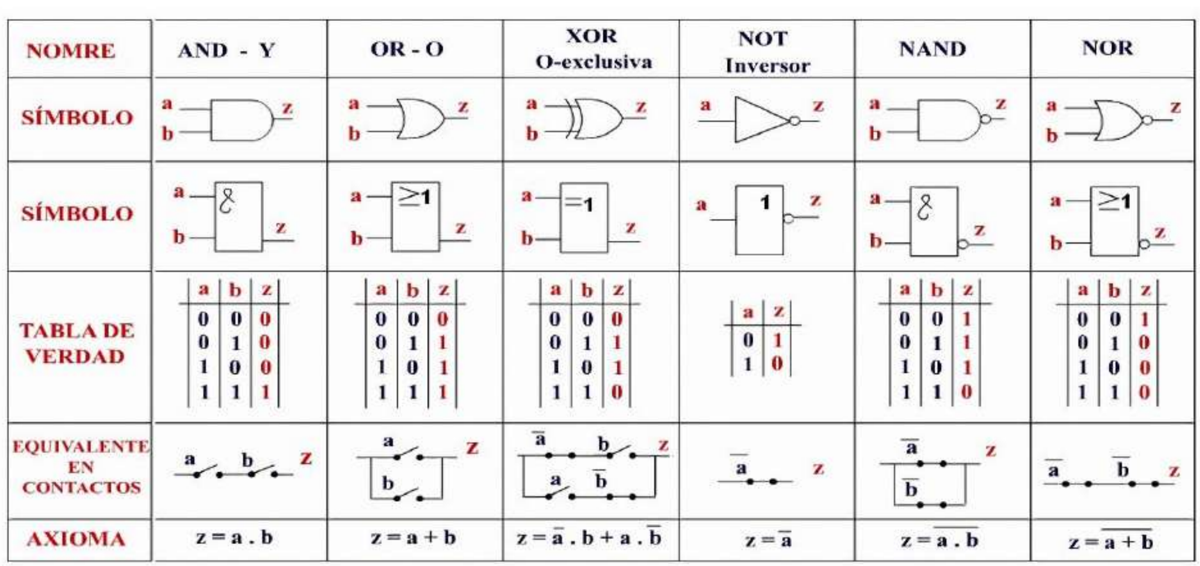
F=$A\ B+AC+BC$

Nota: en este ejemplo se observa que los cuadros externos de un mapa, que son adyacentes, pueden visualizarse formando un cilindro imaginario con el mapa, con respecto a un eje vertical, o, en su caso, teniendo un eje horizontal.

Aplicaciones en circuitos lógicos

Estas compuertas, también conocidas como operadores lógicos, son pequeños circuitos digitales integrados cuyo funcionamiento está de acuerdo con el algebra de Boole en sus operaciones y postulados, las cuales se muestran en la figura 1 (Blogspot, s.f.; Morris, 2013; Morris et al., 2015).

Figura 1. *Funciones lógicas básicas*



NOMRE	AND - Y	OR - O	XOR O-exclusiva	NOT Inversor	NAND	NOR
SÍMBOLO	a b z	a b z	a b z	a z	a b z	a b z
SÍMBOLO	a & b z	a ≥1 b z	a =1 b z	a 1 z	a & b z	a ≥1 b z
TABLA DE VERDAD	a b z 0 0 0 0 1 0 1 0 0 1 1 1	a b z 0 0 0 0 1 1 1 0 1 1 1 1	a b z 0 0 0 0 1 1 1 0 1 1 1 0	a z 0 1 1 0	a b z 0 0 1 0 1 1 1 0 1 1 1 0	a b z 0 0 1 0 1 0 1 0 0 1 1 0
EQUIVALENTE EN CONTACTOS	a b z	a b z	$\bar{a}$ b a $\bar{b}$ z	$\bar{a}$ z	$\bar{a}$ $\bar{b}$ z	$\bar{a}$ $\bar{b}$ z
AXIOMA	$z = a \cdot b$	$z = a + b$	$z = \bar{a} \cdot b + a \cdot \bar{b}$	$z = \bar{a}$	$z = \overline{a \cdot b}$	$z = \overline{a + b}$

Referencias

Barceló, A. (2012). *Introducción a la lógica intencional lógica temporal proposicional*. Instituto de Investigaciones Filosóficas.

Blogspot. (s.f.). *Diagrama de puertas lógicas* [Imagen]. Blogspot https://4.bp.blogspot.com/_HEr2cc0A3Y/VSyACzDZiYI/AAAAAAAABKc/kiR9eIiLt5Q/s1600/puertas-logicas.png

Boole, G. (2023). *El análisis matemático de la lógica* (2.ª ed.). Lincoln Press.

Enciclopedia Herder. (2023). *George Boole*. https://encyclopaedia.herdereditorial.com.

González, R. (2016). *Circuitos lógicos*. Moodle CCH Azcapotzalco.

Jasso Méndez, J., y Rivlin, L. (2023). Argumentación humana eficaz y lógica para el siglo XXI. Andamios. *Revista de Investigación Social, 20*(53), 203-221.

Lifeder. (2023, 24 de agosto). *Álgebra booleana: qué es, historia, teoremas, postulados, ejemplos*. Lifeder. https://www.lifeder.com

Morris, M. (2013). *Diseño digital*. Prentice Hall/Pearson.

Morris, M., Kime, C., y Martin, T. (2015). *Logic and computer design fundamentals*. Prentice Hall College.

Riofrio Sarmiento, E. S., y Rodríguez Cabrera, N. L. (2023). Recurso tecnológico para la enseñanza del álgebra Booleana en compuertas lógicas: una propuesta didáctica. *Revista Uniandes Episteme, 10*(2), 249–260. https://doi.org/10.61154/rue.v10i2.2906

Capítulo 7. Teoría de grafos

Jorge Oliver Bautista Acosta[1]

Carlos Antonio Martínez Cardona[2]

DOI: https://doi.org/10.52501/cc.440.07

Resumen

Un grafo es una estructura matemática que permite representar relaciones entre distintos elementos mediante vértices (nodos) y aristas (conexiones). Los vértices simbolizan entidades como personas, ciudades o dispositivos, mientras que las aristas representan los vínculos entre ellos, como rutas, comunicación o relaciones sociales. Las aristas pueden ser dirigidas o no dirigidas, tener peso o no, incluso pueden formar lazos o múltiples conexiones. Asimismo, los vértices poseen propiedades como el grado, que indica cuántas conexiones tienen, y pueden clasificarse como adyacentes, aislados o terminales. Dentro de los grafos también se estudian los caminos, que son secuencias de vértices conectados; los ciclos, que inician y terminan en el mismo punto; y la accesibilidad entre nodos, lo cual permite analizar la conectividad de la red. Los grafos pueden clasificarse según sus características en diversos tipos, como grafos simples, dirigidos, ponderados, completos, bipartitos o conexos, entre otros, lo que permite modelar diferentes situaciones reales. Para su estudio y aplicación, existen varias formas de representación, como conjuntos, matrices de adyacencia, matrices de incidencia y listas de adyacencia, cada una con ventajas en términos de almacenamiento y eficiencia en el análisis. Estas estructuras son fundamentales en múltiples aplicaciones prácticas, como redes de transporte, sistemas eléctricos, bases de datos y redes digitales, donde facilitan la organización, análisis y optimización de conexiones complejas. En síntesis, los grafos constituyen una herramienta esencial para comprender y modelar sistemas interconectados en diversos ámbitos científicos y tecnológicos.

[1] Maestro en Administración con especialidad en Comercio Internacional. Docente en Tecnológico Nacional de México. Correo electrónico: oliver.ba@itz.edu.mx

[2] Maestro en Administración. Docente en el Tecnológico Nacional de México.

Palabras clave: *grafos, elementos de los grafos, tipos de grafos.*

Conceptos básicos y representaciones

Definición

Un grafo es una representación gráfica de conexiones o relaciones entre objetos, entidades, cosas, etcétera.

Elementos, características y componentes de los grafos

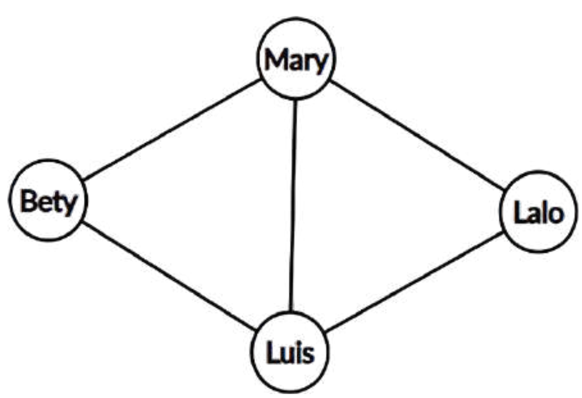
Elementos de un grafo

Básicamente un grafo puede estar formado por dos elementos o componentes:

1. Vértices o nodos: son los puntos que representan los elementos de una relación. Por ejemplo: personas, ciudades, teléfonos, etc.
2. Aristas o enlaces: son las líneas que conectan a los vértices y representan la relación que hay entre dos o más vértices. Por ejemplo: carreteras entre dos ciudades, amistad entre dos personas, etc.

En la figura 1 se muestra un grafo donde los vértices so: Bety, Mary, Luis y Lalo, mientras que las aristas pueden ser relaciones de amistad. Por ejemplo, se observa que hay relación de amistad entre Bety y Mary, ya que hay una arista (línea) entre ellos dos, pero no hay relación de amistad entre Bety y Lalo, ya que no hay ninguna arista (línea) entre ambos.

Figura 1. *Grafo entre personas*



Fuente: elaboración propia.

Consideraciones sobre las aristas

Si la arista carece de dirección se denota indistintamente {a, b} o {b, a}, donde a y b son los vértices que une.

Si {a, b} representa una arista, a los vértices a y b se les llama sus extremos.

Aristas adyacentes: se dice que dos aristas son adyacentes si convergen en el mismo vértice.

Por ejemplo, en la figura 2 las aristas (Zac, SLP) y (Ags, SLP) son aristas adyacentes.

Aristas paralelas: se dice que dos aristas son paralelas cuando están asociadas al mismo par de vértices.

Por ejemplo, en la figura 2, las aristas (Zac, SLP) y (SLP, Zac) son aristas paralelas.

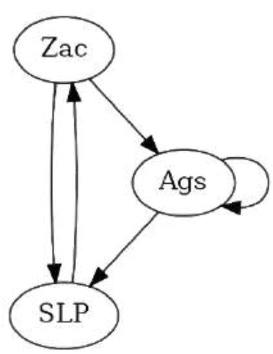
Aristas cíclicas: también llamada lazo. Es una arista que sale y entra en un mismo vértice.

Por ejemplo, en la figura 2 el vértice Ags tiene una arista que sale y entra en sí mismo, por lo tanto, esa arista es un lazo.

Cruce: son dos aristas que se cruzan en un punto.

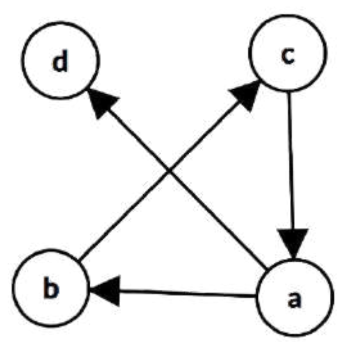
Por ejemplo, en la figura 3 las aristas (a, d) y (b, c) son aristas que se cruzan.

Figura 2. *Tipos de aristas*



Fuente: elaboración propia.

Figura 3. *Cruce de vértices*



Fuente: elaboración propia.

Consideraciones sobre los vértices

Se llama grado de un vértice al número de aristas de las que es extremo. Además, se dice que un vértice es *par* o *impar* según lo es su grado.

Vértices adyacentes: si tenemos un par de vértices de un grafo (a, b) y si tenemos una arista que los une, entonces a y b son vértices adyacentes y se dice que “a” es el vértice inicial y “b” el vértice adyacente.

Por ejemplo, en la figura 4 Ags y SLP son vértices adyacentes y se dice que Ags es el vértice inicial y SLP el vértice adyacente.

Vértice aislado: es un vértice de grado cero, es decir, un vértice que no es punto final de ninguna arista.

Por ejemplo, en la figura 4 Zac es un vértice de grado cero.

Vértice terminal: es un vértice de grado 1, es decir, que solo esta conectado a una arista.

Por ejemplo, en la figura 4, Ags es un vértice terminal.

Figura 4. *Tipos de vértices*

Fuente: elaboración propia.

Caminos

Sean x, y $\in$ V, se dice que hay un camino en G de x a y si existe una sucesión finita no vacía de aristas {x,v1}, {v1,v2},..., {vn,y}. En este caso x e y se llaman los extremos del camino. El número de aristas del camino se llama la longitud del camino.

Por ejemplo, en la figura 5, en el camino (a, b, c, d) la longitud es 3.

Si los vértices no se repiten en el camino se dice que es un camino simple o propio.

Por ejemplo, en la figura 5, en el camino (a, b, c) los vértices no se repiten.

Si hay un camino no simple entre dos vértices, también habrá un camino simple entre ellos.

Por ejemplo, en la figura 5, un camino simple del vértice "a" al vértice "b" sería (a, b) pero se puede también tener el camino no simple (a, b, c, d, b).

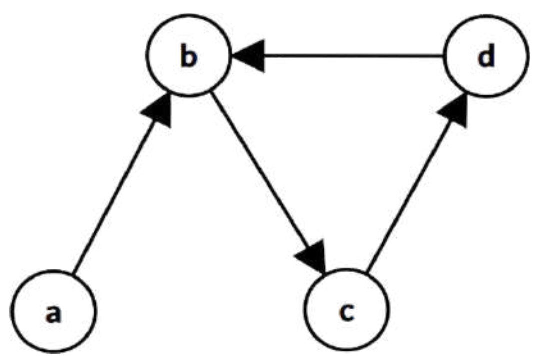
Se llama ciclo, circuito o camino cerrado a un camino simple, que comienza y termina en el mismo vértice.

Por ejemplo, en la figura 5 el camino (c, d, b, c) es un ciclo.

Un vértice "a" se dice accesible desde el vértice "b" si existe un camino entre ellos. Todo vértice es accesible respecto a sí mismo.

Por ejemplo, en la figura 5 los vértices b, c y d son accesibles desde el vértice "a", pero el vértice "a" no es accesible desde algún vértice. Otra forma de decirlo es que hay caminos de "a" a los vértices b, c, d, pero no existe ningún camino de los vértices b, c, d, al vértice "a".

Figura 5. *Caminos*



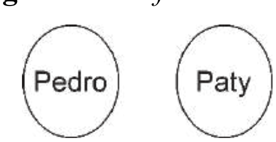
Fuente: elaboración propia.

Características de los grafos

Puede haber uno o más vértices; por ejemplo en la figura 5 se tienen 4 vértices.

Puede que el grafo no tenga aristas (con lo cual no hay relación entre vértices). Por ejemplo, en la figura 2 se aprecia que no hay arista (relación) entre Pedro y Paty.

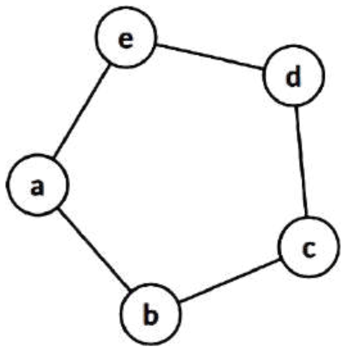
Figura 6. *Grafo sin aristas*



Fuente: elaboración propia.

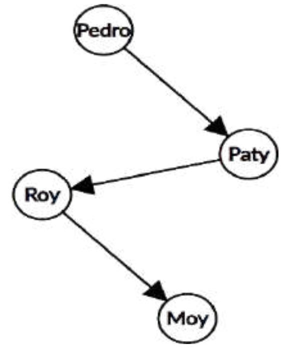
Las aristas pueden ser bidireccionales (como en la figura 7) o direccionales (como en la figura 8), donde solo van en un sentido.

Figura 7. *Grafo bidireccional*



Fuente: elaboración propia.

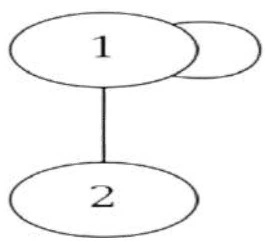
Figura 8. *Grafo dirigido*



Fuente: elaboración propia.

Puede haber aristas que salen y entran al mismo vértice, a las cuales se les llama lazo; por ejemplo, en la figura 9 el vértice 1 tiene un lazo.

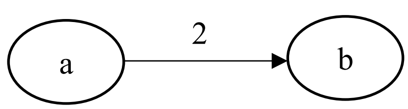
Figura 9. *Grafo con lazo*



Fuente: elaboración propia.

Las aristas pueden tener peso (valor), a lo cual se le llama grafo ponderado; por ejemplo la figura 10 tiene una arista con peso 2.

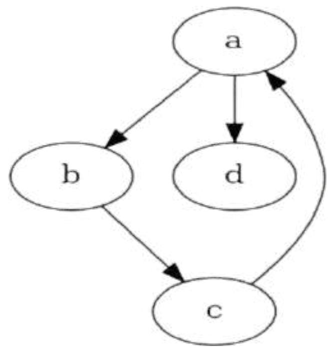
Figura 10. *Arista con peso*



Fuente: elaboración propia.

La forma de las aristas no es relevante (pueden ser líneas rectas o curvas), en la figura 11 se tienen ambos tipos de líneas.

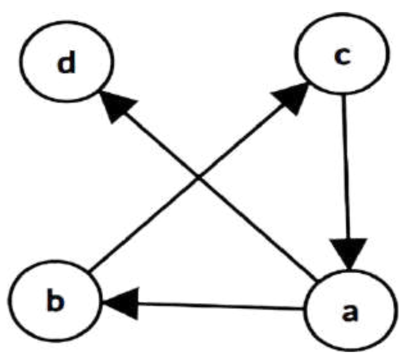
Figura 11. *Aristas rectas y curvas*



Fuente: elaboración propia.

La posición de los vértices no es relevante; por ejemplo, las imágenes 11 y 12 muestran el mismo grafo, pero acomodando los vértices y aristas en diferente posición.

Figura 12. *Grafo con cuatro vértices*



Fuente: elaboración propia.

Tipos de grafos

Más que tipos de grafos, se dan clasificaciones de grafos, de acuerdo con ciertas condiciones que cumplen; algunas de estas características no son mutuamente excluyentes, por el contrario, un grafo puede presentar varias de ellas.

Grafo simple

Es un grafo donde existe una única arista entre cualquier par de vértices, no hay lazos, las aristas no están dirigidas. Por ejemplo, véase los grafos de la figura 1 y 7.

Grafo no dirigido

Es un grafo donde las aristas entre cada par de vértices no tienen dirección (la relación va en ambos sentidos). Por ejemplo, los grafos de las figuras 1, 7 y 9. Son usados cuando la relación entre vértices es bidireccional como una carretera entre dos ciudades.

Grafo dirigido

Es un grafo donde las aristas entre cada par de vértices tienen dirección (como una flecha que va del vértice A hacia el vértice B). Por ejemplo, véase los grafos de las figuras 2, 3, 4, 5, 8, 10, 11 y 12.

Grafo no ponderado

Es un grafo donde no existe ningún valor representado en las aristas. Por ejemplo, véase los grafos de las figuras 1, 2, 3, 4, 5, 6, 7, 8, 9, 11 y 12.

Grafo ponderado

Es un grafo donde las aristas tienen un *peso* o valor, que puede representar distancia, tiempo, costo, etc. Por ejemplo, véase el grafo de la figura 10.

Multígrafo

Es un grafo que puede tener múltiples aristas entre el mismo par de vértices y puede tener bucles. Por ejemplo, véase el grafo de la figura 2. Aplicación: un sistema de transporte con varias rutas entre dos estaciones.

Pseudografo

Es un grafo que permite tanto múltiples aristas como bucles.

Subgrafo

Es un grafo cuyos vértices y aristas son subconjuntos de otro grafo.

Grafo conexo

Un grafo no dirigido es conexo si existe un camino entre cualquier par de vértices. Por ejemplo, véase los grafos de las figuras 1, 2, 7 y 9.

Grafo completo

Cada vértice está conectado con todos los demás vértices. Tiene exactamente [n(n-1)]/2 aristas, donde n es el número de vértices del grafo. Por ejemplo, véase el grafo de la figura 13.

Grafo bipartito

Sus vértices pueden dividirse en dos conjuntos disjuntos, de forma que todas las aristas conectan un vértice de un conjunto con uno del otro. Si, además, todos los vértices de un conjunto están conectados con todos los del otro, se llama bipartito completo. Ejemplo: relaciones entre estudiantes y los cursos que toman.

Grafo plano

Un grafo plano es aquel que puede dibujarse en el plano sin que ninguna arista se interseque.

Grafo regular

Es un grafo donde todos los vértices tienen el mismo grado.

Grafo nulo

Es un grafo donde los vértices no están conectados, esto es, no hay aristas.

Grafo isomorfo

Dos grafos son isomorfos cuando existe una correspondencia biunívoca (uno a uno) entre sus vértices, de tal forma que dos de estos quedan unidos por una arista en común.

Representación de los grafos

Conjuntos

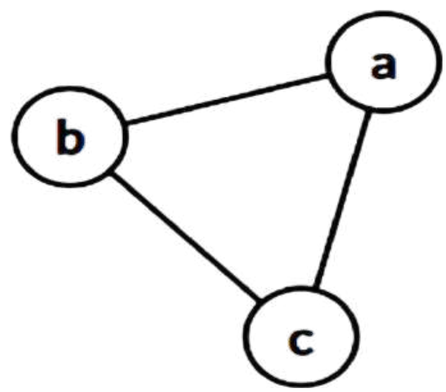
Un grafo (G) se define como un conjunto de vértices (V) no vacío y un conjunto de aristas (A), donde G = {V, A}.

En este capítulo, el conjunto de aristas A se escribirán como pares ordenados, donde v1 y v2 ∈ V y (v1, v2) ∈ A.

Ejemplos:

a) Se tiene el grafo no dirigido.

Figura 13. *Grafo no dirigido*



Fuente: elaboración propia.

La representación en forma de conjuntos del grafo de la figura 13 sería la siguiente:

El conjunto de vértices (V) que forman al grafo.

V = {a, b, c}

El conjunto de pares ordenados (A) que forman las aristas representadas con enlace doble.

A = {(a,b)(b,a)(a,c)(c,a)(b,c)(c,b)}

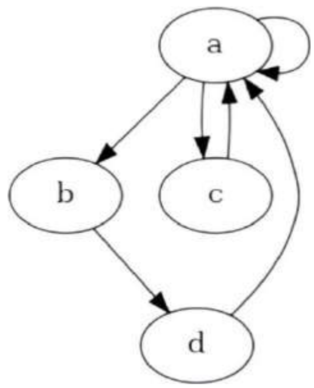
El conjunto de pares ordenados (A) que forman las aristas representadas con enlace simple.

A = {(a,b)(a,c)(b,c)}

La diferencia de la representación con enlace doble y enlace sencillo es que en un grafo no dirigido las aristas son bidireccionales, entonces en enlace doble se ponen dos pares ordenados para cada arista. Mientras tanto, en el enlace sencillo solo se pone un par ordenado y se sobreentiende que va hacia cualquier sentido.

b) Se tiene el grafo dirigido.

Figura 14. *Grafo dirigido*



Fuente: elaboración propia.

La representación en forma de conjuntos del grafo de la figura 13 sería la siguiente:

El conjunto de vértices (V) que forman al grafo.

V = {a, b, c, d}

El conjunto de pares ordenados (A) que forman las aristas del grafo.

A = {(a,a)(a,b)(a,c)(b,d)(c,a)(d,a)}

Matrices

Matriz de adyacencia

Sea G = (V,E), con V ≠ ϕ, se llama matriz de adyacencia de G a la matriz nxn, A = (a_{ij}), donde

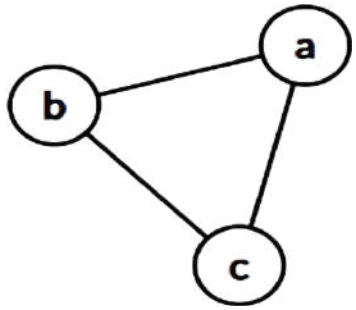
$a_{ij} = 1$ si $(v_i,v_j) \in E$ y $a_{ij} = 0$ si $(v_i,v_j) \notin E$

Ejemplos:

a) Para el grafo de la figura 13, la matriz de adyacencia es la siguiente:

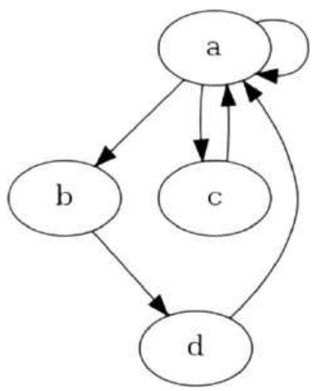
	a	b	c	d
a	1	1	1	0
b	0	0	0	1
c	1	0	0	0
d	1	0	0	0

Para un grafo no dirigido, su matriz de adyacencia siempre es simétrica.



	a	B	c
a	0	1	1
b	1	0	1
c	1	1	0

Para el grafo de la figura 14, su matriz de adyacencia es la siguiente:



Matriz de incidencia

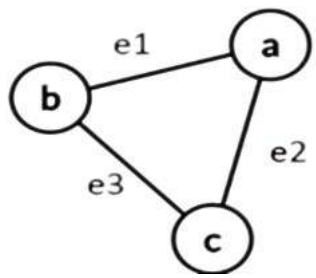
Sea G un grafo de n vértices, la matriz de incidencia A del grafo G, es una matriz $A = a_{ij} \in M_n(R)$, que cumple:

- Las columnas de la matriz representan las aristas del grafo.
- Las filas representan a los distintos vértices.
- Por cada vértice (v_i) unido por una arista (a_{ij}), se tiene que:

$a_{ij} = 1$, si v_i es adyacente a e_j y $a_{ij} = 0$, si v_i no es adyacente a e_j.

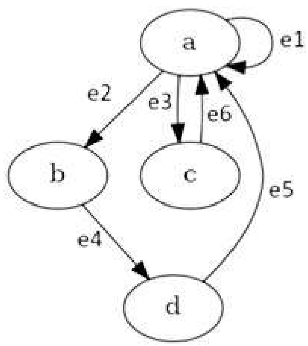
Además, si a_{ij} es entrada de vi entonces aij = -1

a) Para el grafo de la figura 13, su matriz de incidencia es la siguiente:



	e1	e2	e3
a	1	1	0
b	1	0	1
c	0	1	1

b) Para el grafo de la figura 14, su matriz de incidencia es la siguiente:



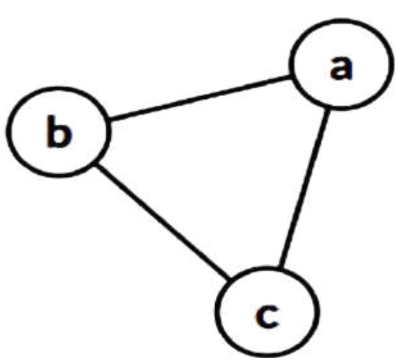
	e1	e2	e3	e4	e5	e6
a	2	1	1	0	-1	-1
b	0	-1	0	1	0	0
c	0	0	-1	0	0	1
d	0	0	0	1	1	0

Listas

Lista de adyacencia

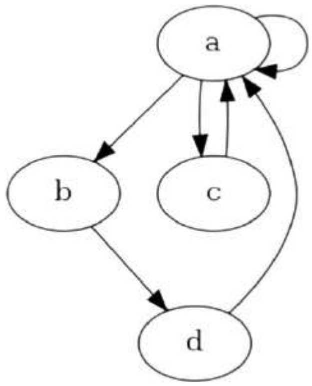
Cada vértice tiene una lista de vértices, los cuales son adyacentes a él. Esto causa redundancia en un grafo no dirigido (ya que A existe en la lista de adyacencia de B y viceversa), pero las búsquedas son más rápidas.

a) Para el grafo de la figura 13, su lista de adyacencia es la siguiente:



Nodo	Adyacentes
a	b, c
b	a, c
c	a, b

b) Para el grafo de la figura 14, su lista de adyacencia es la siguiente:



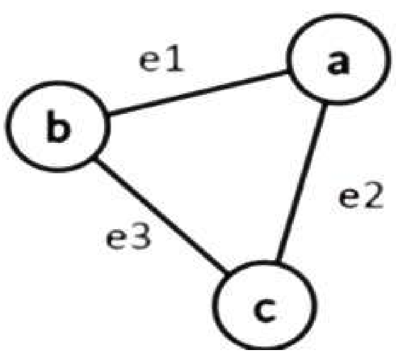
Nodo	Adyacentes
a	a, b, c
b	d
c	a
d	a

Lista de incidencia

Las aristas se representan con un vector de pares (ordenados, si el grafo es dirigido), donde cada par representa una de las aristas. Las listas de incidencia son útiles para:

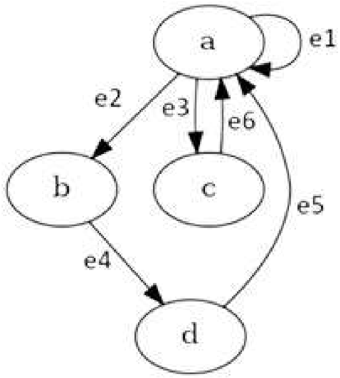
- Representar grafos de manera compacta.
- Realizar algoritmos en grafos, como búsqueda de caminos.
- Implementar estructuras de datos para grafos en *software*.

a) Para el grafo de la figura 13, su lista de incidencia es la siguiente:



Arista	Pares de incidencia
e1	(a,b)
e2	(a,c)
e3	(b,c)

b) Para el grafo de la figura 14, su lista de incidencia es la siguiente:



Arista	Pares de incidencia
e1	(a,a)
e2	(a,b)
e3	(a,c)
e4	(b,d)
e5	(d,a)
e6	(c,a)

Tabla 1. *Comparación entre algunas de las diferentes representaciones de grafos*

Representación	**Espacio**	**Búsqueda de adyacentes**	**Ver si hay conexión**
Lista de adyacencia	$O(V + E)$	Rápido	Lento (hay que buscar en la lista)
Matriz de adyacencia	$O(V^2)$	Lento (recorrer la fila)	Rápido (acceso directo)
Matriz de incidencia	$O(V \times E)$	Poca usada	Poca usada

Aplicaciones de los grafos

- Red de carreteras, donde los nodos son las ciudades y las aristas son las conexiones entre ellas.
- Red eléctrica.
- Red de drenaje de una ciudad.
- Red de páginas de Wikipedia.
- Circuitos eléctricos.
- Cronogramas de proyectos.
- Rutas más cortas.
- Análisis de relaciones sociales .
- Modelado de páginas web y enlaces en internet .
- Representación de redes con direccionalidad, como interacciones en redes sociales.
- Organización de estructuras de datos como árboles y listas enlazadas.
- Resolución de problemas en algoritmos.

Referencias

Balderas, J. (2014, 26 de noviembre). *Matemáticas discretas: relaciones y grafos*. Blogspot. https://matematicasdiscretas8.blogspot.com/2014/11/elementos-y-caracteristicas-de-los.html

Callejas, A., Cosme, A., y Serrano, A. (s.f.). *Elementos y Características de los Grafos*. Instituto Tecnológico de Querétaro. https://trabajoenequipoitq.wixsite.com/matematicas-discreta/61--elementos-y-caracteristicas

CS Academy. (s.f.). *Editor de gráficos*. CS Academy. https://csacademy.com/app/graph_editor/

Khairulin, M. (s.f.). *Draw graph*. GitHub. https://mxwell.github.io/draw-graph/?q=

Mostaccio, C., y Pérez, G. (s.f.). *Grafo Ponderado*. Grafos http://163.10.22.82/OAS/estructuras_de_grafos/grafo_ponderado.html

Perfil, V. (2017, 28 de noviembre). Elementos, características y componentes de los grafos. Matemáticas Discretas. https://tcsjmdiscretas.blogspot.com/2017/11/elementos-caracteristicas-y-componentes.html

Teoría de grafos. (2025, 17 de marzo.). En *Wikipedia*. https://es.wikipedia.org/w/index.php?title=Teor%C3%ADa_de_grafos&oldid=166169478

Universitat Politècnica de València. (2011, 21 de septiembre). *Representación de grafos. Matriz de adyacencia | 3/25 | UPV* [Video). YouTube. https://www.youtube.com/watch?v=cNAkUZaiDo4

Capítulo 8. Árboles

Mariana Ortíz García[1]

José Gabriel Navarro Favela[2]

DOI: https://doi.org/10.52501/cc.440.08

Resumen

Los árboles son estructuras fundamentales dentro de la teoría de grafos, caracterizadas por ser grafos no dirigidos, conexos y sin ciclos, donde existe un único camino entre cada par de vértices. Estas estructuras permiten representar relaciones jerárquicas de manera eficiente, siendo ampliamente utilizadas en informática para organizar datos, optimizar búsquedas y estructurar sistemas como bases de datos y archivos. Un árbol puede tener una raíz a partir de la cual se derivan nodos en relaciones padre-hijo, definiéndose conceptos clave como nivel, altura y grado, los cuales influyen en la eficiencia de los algoritmos que operan sobre ellos. Además, los árboles tienen aplicaciones en diversos campos como la biología, la lingüística y contextos cotidianos como los torneos deportivos o árboles genealógicos. En términos estructurales, un árbol con n vértices contiene n−1 aristas, lo que garantiza una conexión mínima sin redundancias. También destacan los árboles de expansión, que son subgrafos que conectan todos los vértices de una gráfica, y los árboles de expansión mínima, cuyo objetivo es minimizar el peso total de las conexiones. Para construir estos árboles se utilizan algoritmos como la búsqueda en anchura (BFS), la búsqueda en profundidad (DFS), y métodos como Prim y Kruskal, que siguen estrategias para optimizar recorridos y conexiones. Estas técnicas son esenciales en el análisis de redes, rutas y estructuras complejas. En conjunto, los árboles representan una herramienta clave para modelar, organizar y procesar información en múltiples áreas, desde la informática hasta la inteligencia artificial y la toma de decisiones.

[1] Maestra en Sistemas Computacionales. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0009-0003-5415 ; Scopus: 57982741400 ; correo electrónico: mariana.ortiz@itz.edu.mx

[2] Maestro en Sistemas Computacionales. Encargado del Laboratorio de Desarrollo de Software en Tecnológico Nacional de México.

Palabras clave: *árboles, algoritmos de búsqueda, análisis de rutas, árboles de expansión.*

Conceptos básicos y representaciones de árboles

Los árboles constituyen una de las estructuras más fundamentales en la teoría de grafos y tienen múltiples aplicaciones en diversas áreas del conocimiento, desde las ciencias de la computación hasta la biología evolutiva. En términos formales, un árbol es un tipo de grafo no dirigido, conexo y sin ciclos, lo que significa que todos sus vértices están conectados por caminos únicos y que no existen trayectorias cerradas en su estructura (Johnsonbaugh, 1999).

Desde un punto de vista práctico, los árboles ofrecen una forma eficaz de organizar datos. En ciencias de la computación, por ejemplo, son esenciales para la búsqueda eficiente de información, el almacenamiento ordenado de datos, la composición de imágenes digitales y la estructuración de sistemas jerárquicos, como bases de datos o sistemas de archivos. Conceptualmente, un árbol con raíz es aquel en el que uno de los vértices es designado como punto inicial o raíz, y a partir de él se derivan subestructuras llamadas subárboles, con relaciones de padre e hijo entre nodos (Johnsonbaugh, 1999).

El nivel de un vértice se define como la distancia desde la raíz hasta dicho vértice, mientras que la altura del árbol es el mayor nivel presente en la estructura. Estas nociones son clave en el análisis del rendimiento de algoritmos basados en árboles, ya que inciden directamente en la eficiencia de operaciones y en la búsqueda o la inserción.

Más allá de la informática, los árboles tienen aplicaciones en campos como la filogenia, donde se utilizan para representar relaciones de parentesco evolutivo entre especies, o en la lingüística histórica, para reconstruir el origen común de las lenguas. Incluso en contextos cotidianos, como los torneos deportivos de eliminación directa, las estructuras arbóreas modelan el avance de los competidores, donde el equipo ganador aparece en la raíz del árbol.

En suma, los árboles no solo constituyen una herramienta formal de gran elegancia matemática, sino que también poseen una enorme potencia práctica en la modelización de relaciones jerárquicas, optimización de estructuras y análisis de procesos evolutivos.

Un **gráfico de árbol** es una estructura especial dentro de la teoría de grafos que se caracteriza por ser **acíclica y conexa**, es decir, no contiene ciclos y permite que todos sus vértices estén enlazados entre sí mediante caminos únicos. En este sentido, el árbol es la forma más eficiente de conectar todos los puntos de un grafo sin redundancias. Como señala

Johnsonbaugh (1999), los árboles son fundamentales en matemáticas discretas porque establecen la conexión más simple y directa entre nodos dentro de una red.

En un grafo compuesto por **n vértices**, un árbol contará con **exactamente $n-1$ aristas**, lo cual refleja su estructura mínima de conexión. Además, la cantidad total de árboles distintos que se pueden formar con esos vértices es de $\mathbf{n^{n-2n}}$, lo que evidencia la enorme variedad estructural posible dentro de esta categoría de grafos. Esta propiedad fue formalizada en la fórmula de Cayley, y es ampliamente utilizada en áreas como la computación y la combinatoria (Rosen, 2012).

Desde el punto de vista formal, un **árbol T** se define como un grafo simple que cumple con la condición de que para cada par de vértices v y w, existe un **único camino simple** que los conecta. Además, si se selecciona un vértice específico como punto de inicio, se obtiene un **árbol con raíz**, donde ese nodo central sirve como referencia para establecer relaciones jerárquicas dentro de la estructura (Johnsonbaugh, 1999).

Una de las aplicaciones más prácticas de los árboles es en la **organización de datos dentro de bases de datos**, ya que permiten establecer relaciones jerárquicas entre los elementos, facilitando su clasificación y recuperación. En este contexto, los árboles sirven como modelos para representar estructuras que requieren eficiencia y orden, como los índices en bases de datos relacionales (Johnsonbaugh, 1999).

Más allá de la informática, los árboles también tienen gran utilidad en áreas como la **biología evolutiva**, donde se emplean para representar procesos de **filogenia**. En este caso, los árboles muestran cómo es que diferentes entidades están vinculadas por líneas de descendencia, lo que permite trazar relaciones de parentesco entre especies u otros sistemas como las lenguas humanas. Como afirman Semple y Steel (2003), los árboles filogenéticos constituyen herramientas clave para el estudio de la evolución.

En el campo de la **informática**, el concepto de árbol también se utiliza como una **estructura de datos abstracta** que refleja una jerarquía natural. Cada árbol contiene una **raíz**, que es el nodo principal, y a partir de esta se ramifican **subárboles**, cuyos nodos están conectados a través de relaciones padre-hijo. Esta representación se ha vuelto esencial en algoritmos de búsqueda, estructuras organizativas y almacenamiento de datos eficientes (Knuth, 1998).

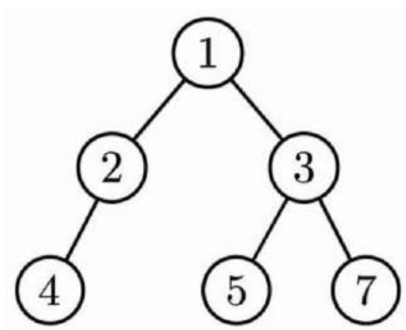
Al representar con una gráfica cualquier torneo deportivo de eliminación se obtiene un árbol, y el equipo ganador será la raíz del árbol.

Dentro de los usos más comunes de los árboles se encuentra el de almacenar un dato de tal modo que su búsqueda sea eficiente, que tenga listas ordenadas de datos y que haya una composición de imágenes digitales.

El nivel de un vértice v es la longitud del camino simple de la raíz a v. La altura de un árbol es el número máximo de nivel que aparece en el árbol. Los hijos de la raíz son los vértices subsecuentes unidos a la raíz. Un vértice padre es un vértice que está por encima de un vértice hijo y más cerca de la raíz.

Partes de un árbol

Figura 1. *Partes de un árbol*



Fuente: elaboración propia.

Si 1 es la raíz,

entonces 2 y 3 son hijos de 1,

3 es padre de 5 y de 7; y 2 es padre de 4.

4, 5 y 7 solamente son hojas.

El grado de un vértice mide lo retirado que esté de la raíz. Por ejemplo:

Vértice	1	2	3	4	5	7
Grado	0	1	1	2	2	2

El grado del árbol es el grado máximo de un vértice, en este caso es 2.

Los organigramas son ejemplos de árboles. A este tipo de árboles se les denomina árboles jerárquicos, se emplean para representar cómo se relacionan lógicamente los distintos registros dentro de una base de datos, donde se muestra el control o administración de una determinada empresa, institución u organización.

Los árboles familiares también son un tipo especial de árbol jerárquico. Se recomienda realizar un árbol genealógico de la familia donde se incluya por lo menos 25 integrantes.

Árboles de expansión

El desafío consiste en encontrar una subgráfica T dentro de una gráfica G, tal que T sea un árbol que incluya todos los vértices de G; a esta estructura se le conoce como árbol de expansión. Una gráfica tiene árbol de expansión siempre y cuando sea conexa, es decir que todos sus vértices estén interconectados (Johnsonbaugh, 1999).

¿Qué es un árbol de expansión mínima? Es un **subconjunto de las aristas de un grafo** que:

1. **Conecta todos los vértices** del grafo (sin ciclos).
2. **Tiene el menor peso total posible**.

Un algoritmo de búsqueda encuentra la única ruta más corta entre dos puntos, utilizando una heurística que ayuda a agilizar el proceso de búsqueda.

Existen varios métodos para encontrar árboles en expansión.

A) **Búsqueda a lo ancho**. Consiste en explorar todos los vértices que se encuentran en un mismo nivel antes de avanzar al siguiente. La búsqueda en anchura es una técnica que permite recorrer de forma sistemática todos los nodos de un grafo. Es especialmente útil en la resolución de problemas de optimización, donde se requiere identificar la mejor alternativa entre varias. El proceso comienza en un vértice inicial v, considerado la raíz, el cual se activa primero. Luego se marcan como visitados todos los vecinos de ese vértice que aún no hayan sido explorados. A continuación, se repite el proceso con los vecinos de los hijos de v, asegurándose de no visitar ningún nodo más de una vez. Como resultado, se genera una estructura sin ciclos, es decir, un árbol.

B) **Búsqueda en profundidad (DFS, por sus siglas en inglés)** es un algoritmo diseñado para recorrer todos los vértices de un grafo finito, siguiendo una estrategia que prioriza explorar lo más profundo posible un camino antes de retroceder. El proceso comienza seleccionando un vértice inicial o raíz, desde el cual se visitan primero sus descendientes directos, luego los hijos de estos y así sucesivamente, hasta llegar a un nodo que no tenga vecinos no visitados.

A diferencia de la búsqueda en anchura, DFS se caracteriza por investigar completamente una rama antes de considerar otras, lo que implica que atraviesa los niveles inferiores del grafo antes de regresar a los superiores. Para gestionar esta secuencia de pasos, generalmente se utiliza una **pila**, ya sea de forma explícita o mediante **recursión** que emplea la pila del sistema, lo que permite regresar a vértices anteriores cuando no se encuentran más caminos disponibles desde el nodo actual.

Cuando ya no es posible avanzar desde un vértice activo, el algoritmo **retrocede** a través de los vértices previamente visitados hasta localizar uno con una conexión pendiente por explorar. Desde allí se reinicia el proceso de avance, repitiendo este patrón de descenso y retroceso hasta que se hayan examinado todos los nodos alcanzables. Este enfoque hace de DFS una herramienta fundamental en diversos algoritmos aplicados sobre grafos, como los de ordenamiento topológico o los que evalúan propiedades como la **conectividad** o la **planitud** de una estructura gráfica.

C) **Búsqueda por menor costo (árbol de expansión mínima).** Es una subgráfica que se construye a partir de una gráfica conexa, que tiene un único camino simple de un vértice a otro y cuya suma de pesos sea mínima.

En el ámbito de la informática, estos algoritmos se emplean ampliamente para **recorrer grafos** y **buscar rutas** entre puntos conocidos como nodos. Su objetivo principal es **encontrar trayectos viables de manera eficiente**, lo que los convierte en herramientas clave para el análisis de redes y estructuras conectadas. Se destacan por su **precisión y buen desempeño**, lo que ha llevado a su uso extendido en múltiples aplicaciones computacionales.

No obstante, en contextos reales como los **sistemas de navegación o enrutamiento de viajes**, suelen ser superados por otros algoritmos más avanzados que permiten **preprocesar el grafo**, optimizando así la velocidad y eficiencia en la resolución del problema (Kurose y Ross, 2020).

Métodos para sacar árboles de expansión mínima

Algoritmo de PRIM

El algoritmo en cuestión construye un **árbol de expansión mínima** incorporando aristas de forma progresiva. En cada paso, selecciona la arista con **menor peso** que no genere un ciclo dentro del subgrafo en desarrollo. Este procedimiento pertenece a la categoría de **algoritmos codiciosos**, ya que en cada iteración toma la mejor decisión disponible en ese momento, sin considerar el impacto de decisiones anteriores (Cormen et al., 2009).

El principio que guía este método puede resumirse como una estrategia de **optimización local**, en la que se elige la mejor opción inmediata —una sola arista con el menor peso posible— con la intención de construir una solución eficiente. Sin embargo, este tipo de optimización **no siempre garantiza** que la solución final sea óptima en términos globales. Si durante el proceso no se encuentra una conexión válida hacia los vértices restantes, el algoritmo retrocede al vértice previo para verificar si existen otras conexiones posibles que aún no se han evaluado (Cormen et al., 2009).

Algoritmo de Kruskal

El algoritmo de Kruskal es un método *greedy* (codicioso) que se utiliza para encontrar un árbol de expansión mínima (MST, por sus siglas en inglés) en un grafo no dirigido, conectado y ponderado.

Consideremos un grafo dirigido y ponderado con N nodos, ninguno de ellos aislado. Sea x el nodo de inicio y un vector D de tamaño N que almacenará, al finalizar el algoritmo, las distancias mínimas desde x a cada uno de los demás nodos. El procedimiento se desarrolla de la siguiente manera:

1. Inicialmente, todas las entradas del vector D se asignan un valor infinitamente grande para representar distancias desconocidas, excepto la distancia desde x a sí mismo, que se establece en cero, dado que no existe distancia alguna para llegar a sí mismo.
2. Se define a = x, donde a es el nodo actual que se está evaluando.
3. Se examinan todos los nodos adyacentes a a que no hayan sido marcados como visitados; a estos los denominamos $v_{i.}$

4. Si la distancia almacenada en D para llegar a v_i desde x es mayor que la suma de la distancia desde x a a más la distancia desde a a v_i, entonces el valor en D para v_i se actualiza con esta suma menor.

5. El nodo a se marca como completo o visitado.

6. Se selecciona como nuevo nodo actual aquel con el valor más pequeño en D entre los nodos no visitados (esto se puede gestionar eficazmente utilizando una cola de prioridad). Se repite el proceso desde el paso 3 hasta que todos los nodos estén marcados.

Este método es una implementación típica del algoritmo de Dijkstra, ampliamente utilizado para encontrar rutas óptimas en grafos ponderados (Cormen et al., 2009).

Pasos del algoritmo de Kruskal

1. Ordena todas las aristas del grafo de menor a mayor peso.
2. Inicializa el MST como vacío (sin aristas).
3. Recorre la lista de aristas en orden creciente de peso:
4. Si agregar la arista NO forma un ciclo, agrégala al MST.
5. Si forma un ciclo, descártala.
6. Repite hasta que el MST tenga exactamente (n – 1) aristas, donde n es el número de vértices.

En la tabla 1 se observa la comparación de los diferentes algoritmos en cuanto a rutas cortas y árboles en expansión.

Tabla 1. *Comparación de algoritmos*

Algoritmo	¿Sirve para rutas más cortas?	¿Sirve para árbol de expansión?	Acepta pesos negativos
Bellman-Ford	☑ Sí	✖ No	☑ Sí
Dijkstra	☑ Sí	✖ No	✖ No (sin modificar)
Kruskal/Prim	✖ No	☑ Sí	⚠ Solo pesos positivos

El siguiente es un código de árbol en expansión a lo **ancho** en Java.

```
import java.util.*;
public class BFSExpansionTree {
    // Número de vértices
    private int V;
    // Lista de adyacencia
    private LinkedList<Integer>[] adj;
    // Constructor
    public BFSExpansionTree(int vértices) {
        this.V = vertices;
        adj = new LinkedList[vértices];
        for (int i = 0; i < vértices; ++i)
            adj[i] = new LinkedList<>();
    }

    // Método para agregar aristas (no dirigido)
    void addEdge(int v, int w) {
        adj[v].add(w);
        adj[w].add(v); // Porque es grafo no dirigido
    }

    // Método para construir e imprimir el árbol de expansión por BFS
    void bfsExpansionTree(int start) {
        boolean[] visited = new boolean[V];
        int[] parent = new int[V]; // Para registrar el "padre" de cada nodo en el árbol
        Arrays.fill(parent, -1);
        Queue<Integer> queue = new LinkedList<>();
        visited[start] = true;
        queue.add(start);
        System.out.println("Aristas del árbol de expansión (BFS):");
        while (!queue.isEmpty()) {
```

```
            int current = queue.poll();
            for (int neighbor : adj[current]) {
                if (!visited[neighbor]) {
                    visited[neighbor] = true;
                    parent[neighbor] = current;
                    queue.add(neighbor);
                    System.out.println(current + " - " + neighbor);
                }
            }
        }
    }

    // Ejemplo de uso
    public static void main(String[] args) {
        BFSExpansionTree g = new BFSExpansionTree(6);
        g.addEdge(0, 1);
        g.addEdge(0, 2);
        g.addEdge(1, 3);
        g.addEdge(1, 4);
        g.addEdge(2, 5);
        g.bfsExpansionTree(0); // Comienza desde el nodo 0
    }
}
```

Ahora el código de expansión mínima con **Kruskal**

```
import java.útil.*;
// Clase para representar una arista
class Edge implements Comparable<Edge> {
    int src, dest, weight;
    public Edge(int s, int d, int w) {
        src = s;
```

```
        dest = d;
        weight = w; }

    // Para ordenar por peso
    public int compareTo(Edge other) {
        return this.weight - other.weight;
    }
}

// Clase para representar un subconjunto (para Union-Find)
class Subset {
    int parent, rank;
}

public class KruskalMST {
    int vertices;
    List<Edge> edges = new ArrayList<>();

    public KruskalMST(int v) {
        this.vértices = v;
    }

    // Añadir arista
    void addEdge(int src, int dest, int weight) {
        edges.add(new Edge(src, dest, weight));
    }

    // Encontrar conjunto con compresión de caminos
    int find(Subset[] subsets, int i) {
        if (subsets[i].parent != i)
            subsets[i].parent = find(subsets, subsets[i].parent);
```

```
        return subsets[i].parent;
    }

    // Unir dos conjuntos por rango
    void union(Subset[] subsets, int x, int y) {
        int xroot = find(subsets, x);
        int yroot = find(subsets, y);
        if (subsets[xroot].rank < subsets[yroot].rank)
            subsets[xroot].parent = yroot;
        else if (subsets[xroot].rank > subsets[yroot].rank)
            subsets[yroot].parent = xroot;
        else {
            subsets[yroot].parent = xroot;
            subsets[xroot].rank++;
        }
    }

    // Método principal para construir el MST
    void kruskal() {
        List<Edge> result = new ArrayList<>();
        Collections.sort(edges); // Ordenar aristas por peso
        Subset[] subsets = new Subset[vertices];
        for (int i = 0; i < vertices; i++) {
            subsets[i] = new Subset();
            subsets[i].parent = i;
            subsets[i].rank = 0;
        }
        for (Edge edge : edges) {
            int x = find(subsets, edge.src);
            int y = find(subsets, edge.dest);
            if (x != y) {
```

```
            result.add(edge);
            unión(subsets, x, y);
        }
    }
    System.out.println("Aristas del Árbol de Expansión Mínima:");
    int totalWeight = 0;
    for (Edge e : result) {
        System.out.println(e.src + " - " + e.dest + " : " + e.weight);
        totalWeight += e.weight;
    }
    System.out.println("Peso total del MST: " + totalWeight);
  }
  // Ejemplo de uso
  public static void main(String[] args) {
    KruskalMST g = new KruskalMST(4);
    g.addEdge(0, 1, 10);
    g.addEdge(0, 2, 6);
    g.addEdge(0, 3, 5);
    g.addEdge(1, 3, 15);
    g.addEdge(2, 3, 4);
    g.kruskal();
  }
}
```

Aplicaciones de los árboles

Los árboles, como estructuras de datos jerárquicas, poseen una amplia aplicación en contextos reales debido a su capacidad para representar relaciones de dependencia, clasificación y organización escalonada de información. Su utilidad no es meramente teórica; por el contrario, constituyen uno de los fundamentos operativos de múltiples sistemas tecnológicos contemporáneos (Cormen et al., 2022; Goodrich et al., 2014).

En el ámbito de los sistemas de archivos, por ejemplo, la organización de carpetas y subcarpetas en sistemas operativos como Windows, macOS o Linux responde a una estructura arbórea. Cada directorio funciona como un nodo que puede contener otros nodos (subdirectorios o archivos), formando una jerarquía que facilita la localización eficiente de información. Esta estructura permite búsquedas organizadas y rápidas, especialmente cuando se implementan variantes como árboles balanceados para optimizar tiempos de acceso.

En el campo de las bases de datos, los árboles son esenciales para indexar información. Estructuras como el *B-tree* y sus variantes permiten que sistemas gestores de bases de datos encuentren registros en grandes volúmenes de información en tiempos logarítmicos. Sin estos mecanismos, consultas en bancos, hospitales o plataformas de comercio electrónico serían considerablemente más lentas. La eficiencia en la recuperación de datos se traduce directamente en mejor desempeño organizacional y experiencia del usuario (Cormen et al., 2022).

Otra aplicación significativa se encuentra en los motores de búsqueda en internet. Algoritmos como PageRank, desarrollado por Larry Page, utilizan estructuras relacionadas con grafos y árboles para jerarquizar páginas web según su relevancia. Aunque conceptualmente se apoyan en grafos, la organización jerárquica de resultados y la estructuración del contenido digital mantienen principios análogos a los árboles, particularmente en la clasificación y recuperación de información.

En el ámbito de la inteligencia artificial, los árboles de decisión constituyen herramientas fundamentales para el aprendizaje automático y la toma de decisiones automatizada. Modelos como el C4.5 permiten clasificar datos y predecir comportamientos a partir de reglas jerarquizadas. Estos modelos se emplean en diagnóstico médico, evaluación crediticia, detección de fraudes y sistemas de recomendación. La estructura arbórea facilita la interpretación de resultados, lo que aporta transparencia en entornos donde la explicabilidad es crucial.

Asimismo, en el desarrollo de compiladores y lenguajes de programación, los árboles sintácticos abstractos representan la estructura gramatical del código fuente. Cada instrucción se descompone en nodos que reflejan operaciones y operandos, permitiendo que el sistema interprete y ejecute correctamente los programas (Goodrich et al., 2014).

Desde una perspectiva más amplia, los árboles modelan fenómenos sociales y naturales: genealogías familiares, estructuras organizacionales, taxonomías biológicas y esquemas curriculares pueden representarse mediante estructuras arbóreas. Su fortaleza radica en representar relaciones jerárquicas sin ambigüedad y con eficiencia computacional.

En síntesis, los árboles no son únicamente una abstracción matemática, sino una herramienta estructural indispensable para organizar, buscar, clasificar y procesar información en múltiples ámbitos de la vida cotidiana. Su aplicación transversal confirma que la teoría algorítmica se materializa en soluciones concretas que sostienen la infraestructura digital contemporánea.

Referencias

Cormen, T. H., Leiserson, C. E., Rivest, R. L., y Stein, C. (2009). *Introduction to algorithms* (3.ª ed.). MIT Press.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., y Stein, C. (2022). *Introduction to algorithms* (4.ª ed.). MIT Press.

Goodrich, M. T., Tamassia, R., y Goldwasser, M. H. (2014). *Data structures and algorithms in Python*. Wiley.

Johnsonbaugh, R. (1999). *Matemáticas discretas* (4.ª ed.). Editorial Pearson.

Knuth, D. E. (1998). *The art of computer programming, volume 1: fundamental algorithms* (3.ª ed.). Addison-Wesley.

Kurose, J. F., y Ross, K. W. (2020). C*omputer networking: a top-down approach* (8th ed.). Pearson.

Rosen, K. H. (2012). *Discrete mathematics and its applications* (7.ª ed.). McGraw-Hill.

Semple, C., y Steel, M. (2003). *Phylogenetics*. Oxford University Press.

Stallings, W. (2017). *Computer organization and architecture: designing for performance*. Pearson.

Tanenbaum, A. S. (2016). *Structured computer organization* (6.ª ed.). Pearson.

Tanenbaum, A. S., y Austin, T. (2013). *Structured computer organization*. Pearson.

Tanenbaum, A. S., y Bos, H. (2014). *Modern operating systems*. Pearson.

UNAD. (s.f.). *Sistemas numéricos*. https://www.unadmexico.mx/LITE_36/_Un_151_OperacionesAritmeticasBasicas/escenas/2_Inicio_1.html#:~:text=B%C3%A1sicamente%20los%

Capítulo 9. Principios de conteo y combinatoria

Alfredo García Castañon[1]
José Antonio Reyes Rodríguez[2]
María del Rosario Gámez Aguilar[3]

DOI: https://doi.org/10.52501/cc.440.09

Resumen

Los principios fundamentales de conteo constituyen reglas básicas de la combinatoria que permiten determinar el número de formas en que pueden ocurrir ciertos eventos sin necesidad de enumerarlos. Entre ellos destacan el principio aditivo, que se aplica cuando las opciones son mutuamente excluyentes y se suman las posibilidades, y el principio multiplicativo, que se utiliza cuando un proceso ocurre en etapas sucesivas, multiplicando las opciones de cada paso. Asimismo, el principio de inclusión-exclusión amplía el conteo cuando existen intersecciones entre conjuntos, evitando duplicidades. Estos principios son esenciales para el estudio de técnicas más avanzadas como las permutaciones y combinaciones. Las permutaciones corresponden a los arreglos u ordenamientos de elementos donde el orden sí importa, mientras que las combinaciones se refieren a selecciones en las que el orden no es relevante. Ambas se calculan mediante fórmulas basadas en el factorial y tienen aplicaciones en diversas áreas como la probabilidad, la estadística y la informática. Además, existen variantes como las permutaciones con repetición, que ajustan el conteo cuando hay elementos iguales. El principio de inclusión-exclusión también se aplica a múltiples conjuntos, sumando los elementos individuales, restando intersecciones dobles y agregando las triples, lo que permite un conteo preciso. En conjunto, estos conceptos tienen amplias aplicaciones en campos como la ingeniería, la biología, la economía y la informática, especialmente en problemas de

[1] Maestro en Informática Administrativa. Docente en Tecnológico Nacional de México. ORCID: https://orcid.org/0009-0001-3656-8293 ; correo electrónico: alfredo.garcia@itz.edu.mx
[2] Licenciado en Informática. Docente en Tecnológico Nacional de México.
[3] Ingeniero en Materiales. Docente en Tecnológico Nacional de México.

optimización, análisis de datos y toma de decisiones. Su estudio fortalece el razonamiento lógico y la capacidad para resolver problemas complejos.

Palabras clave: *principios de conteo, permutaciones, principio de inclusión-exclusión.*

Principios fundamentales de conteo

Los **principios fundamentales de conteo** son reglas básicas de la **combinatoria** que permiten calcular el número de formas en que pueden ocurrir ciertos eventos, sin necesidad de enumerar cada posibilidad una por una. Se utilizan en situaciones de conteo y probabilidad, y constituyen la base para técnicas más avanzadas como permutaciones y combinaciones.

Principio aditivo (o de la suma)

Si un proceso puede realizarse de **dos maneras mutuamente excluyentes** (no pueden ocurrir al mismo tiempo), entonces el número total de maneras de realizarlo es la **suma** de ambas posibilidades.

Ejemplo:

En una heladería puedes elegir entre 3 tipos de paletas o 4 tipos de nieves. Como no puedes elegir ambas a la vez, hay:

$3 + 4 = 7$ posibilidades.

Principio multiplicativo (o de la multiplicación)

Si un proceso se compone de **dos pasos sucesivos** y el primero puede hacerse de m formas y el segundo de n formas, entonces el número total de formas de realizar ambos pasos es la **multiplicación**: $m \times n$.

Ejemplo:

Para vestirte, tienes 4 camisas y 3 pantalones. Cada camisa se puede combinar con cada pantalón, entonces hay:

$4 \times 3 = 12$ formas de vestirte.

Principio de inclusión-exclusión (extensión del aditivo)

Cuando las opciones **no son mutuamente excluyentes** (pueden coincidir), se debe restar la intersección.

Ejemplo:

Si en una fiesta hay 20 personas que saben inglés, 15 que saben francés y 5 que saben ambos idiomas, el total de personas que saben al menos un idioma es: 20 + 15 – 5 = 30.

Importancia

Estos principios permiten **contar de manera eficiente** y constituyen la base para:

- **Permutaciones** (ordenamientos).
- **Combinaciones** (selecciones sin orden).
- **Variaciones** y problemas de probabilidad.

Permutaciones y combinaciones

La **combinatoria** es una rama de la matemática que estudia el conteo de objetos y sus posibles disposiciones sin necesidad de enumerar cada caso. Dentro de este campo, las **permutaciones** y **combinaciones** constituyen herramientas esenciales que permiten calcular el número de formas en que se pueden ordenar o seleccionar elementos de un conjunto. De acuerdo con Camacho y Gómez (2018), el dominio de estas técnicas facilita la resolución de problemas en probabilidad, estadística, informática y teoría de decisiones. Una **permutación** es un arreglo u **ordenamiento** de todos o parte de los elementos de un conjunto, donde el **orden sí importa** (Martínez, 2017).

Fórmula general de permutación

La cantidad de permutaciones de *n* elementos tomados de r en r se expresa como:

$$\mathrm{P(n,r)} = \frac{(n-r)!}{\mathrm{n!}}$$

donde *n*! (factorial de *n*) representa el producto de todos los números enteros positivos desde 1 hasta *n*.

Ejemplo:

En una repisa hay 5 libros distintos y queremos saber de cuántas maneras se pueden ordenar 3 de ellos:

$$P(5.3) = \frac{5!}{(5-3)!} = \frac{120}{2} = 60$$

Respuesta: 60 formas distintas de ordenar 3 libros de un total de 5.

Permutaciones con repetición

Cuando algunos elementos se repiten, el número de permutaciones se reduce, ya que no todos los ordenamientos son únicos.

Ejemplo:

En la palabra "MAMÁ" hay 4 letras en total, con 2 "M" y 2 "A".

$$P = \frac{4!}{(2! \times 2!)} = 6.$$

Respuesta: 6 formas distintas de ordenar las letras.

Combinaciones

Definición: una combinación es una selección de elementos de un conjunto, donde el orden no importa (Mendoza y Rodríguez, 2019).

Fórmula general de combinación

$$C(n,r) = \frac{n!}{r!(n-r)!)}$$

Ejemplo:

De un grupo de 8 estudiantes, queremos elegir 3 para formar un comité. Como el orden de elección no importa:

$$C\,(8,3) = \frac{8!}{3!(8-3)!} = 56.$$

Respuesta: hay 56 maneras de elegir a los 3 integrantes.

Tabla 1. *Diferencias clave entre permutaciones y combinaciones*

Aspecto	Permutación	combinación
Orden	Importa	No importa
Fórmula	$P(n,r) = \frac{(n-r)!}{n!}$	$C(n,r) = \frac{n!}{r!(n-r)!)}$
Ejemplo	Ordenar 3 libros de 5	Elegir 3 estudiantes de 8
Resultado del ejemplo	60 formas	56 formas

Principio de inclusión-expulsión

El **principio de inclusión–exclusión** es una técnica de conteo usada para determinar el número de elementos que pertenecen a la **unión** de varios conjuntos, evitando la **sobrecontabilización**.

En su forma básica, para dos conjuntos A y B:

$|A \cup B|=|A|+|B|-|A \cap B|$ se suman los elementos de cada conjunto.

- Se resta la intersección porque fue contada dos veces.

Para **tres conjuntos** A, B, C:

$|A \cup B \cup C|=|A|+|B|+|C|-|A \cap B|-|A \cap C|-|B \cap C|+|A \cap B \cap C|$

La idea es: **sumar las partes individuales, restar las intersecciones dobles y agregar las triples**.

En general, para *n* conjuntos, el principio sigue el mismo patrón de **sumar y restar alternadamente** las intersecciones.

Ejemplo 1: dos conjuntos

En un grupo de estudiantes:

- 20 saben programar en Python (A).
- 15 saben programar en **C++** (B).
- 8 saben ambos lenguajes ($A \cap B$).

¿Cuántos estudiantes saben al menos un lenguaje?

$|A \cup B|=|A|+|B|-|A \cap B|=20+15-8=27$

27 estudiantes saben al menos un lenguaje.

Ejemplo 2: tres conjuntos

En una clase de ingeniería:

- 25 estudiantes aprobaron **Matemáticas** (A).
- 18 aprobaron **Física** (B).
- 20 aprobaron **Química** (C).
- 10 aprobaron Matemáticas y Física.
- 7 aprobaron Matemáticas y Química.
- 5 aprobaron Física y Química.
- 3 aprobaron las tres materias.

¿Cuántos aprobaron **al menos una**?

$|A \cup B \cup C| = 25 + 18 + 20 - (10 + 7 + 5) + 3$

$|A \cup B \cup C| = 63 - 22 + 3 = 44$

44 estudiantes aprobaron al menos una materia.

Importancia en ingeniería

El principio de inclusión-exclusión es clave en:

- **Teoría de conjuntos y probabilidad**: calcular probabilidades de eventos compuestos.
- **Análisis de datos**: evitar duplicados al contar elementos en bases de datos.
- **Optimización y computación**: algoritmos de conteo y combinatoria.

Aplicaciones

- Estadística y probabilidad: cálculo de probabilidades.
- Informática: generación de contraseñas, algoritmos de cifrado.
- Biología: combinaciones genéticas.
- Economía y administración: toma de decisiones.
- Educación: desarrollo del razonamiento lógico-matemático.

Conclusión

El estudio de las permutaciones y combinaciones constituye una base fundamental para comprender el conteo en matemáticas. Al distinguir entre situaciones donde el orden es importante y aquellas donde no lo es, se construye un marco analítico aplicable en numerosos campos. Su aprendizaje contribuye al desarrollo de competencias lógicas y de resolución de problemas, esenciales en la formación académica y profesional.

Referencias

Camacho, A., y Gómez, R. (2018). Aplicaciones de la combinatoria en problemas de probabilidad. *Revista Colombiana de Educación Matemática, 9*(1), 55-68.

Martínez, J. (2017). *Fundamentos de lógica y matemáticas*. Universidad Nacional Autónoma de México.

Mendoza, F., y Rodríguez, L. (2019). Aplicaciones de la lógica matemática en la computación. *Revista Colombiana de Matemáticas Aplicadas, 11*(2), 45-58.

Ramírez, S. (2021). La enseñanza de las funciones como herramienta para el desarrollo del pensamiento matemático. *Revista Iberoamericana de Educación Matemática, 15*(29), 45-63.

Capítulo 10. Inducción y recursión

Antonio Pérez Cortés[1]
Cecilia Guadalupe Hernández Yáñez[2]

DOI: https://doi.org/10.52501/cc.440.10

Resumen

Las demostraciones por inducción matemática constituyen una herramienta fundamental para verificar proposiciones que involucran números naturales. Este principio permite determinar si una afirmación es válida para todos los valores de n a partir de dos pasos esenciales: comprobar que la proposición es verdadera para un valor inicial (caso base) y demostrar que, si se cumple para un valor k, entonces también se cumple para $k + 1$ (paso inductivo). De esta manera, se garantiza la validez de la proposición para todos los números naturales a partir de n inicial. Este método ha sido ampliamente utilizado desde el siglo XVII, aunque fue formalizado posteriormente, y resulta clave para demostrar fórmulas como la suma de números naturales o la suma de cuadrados, así como para analizar desigualdades. Por otro lado, los algoritmos recursivos representan un enfoque complementario en matemáticas y programación, basado en la idea de resolver problemas mediante su reducción a casos más simples del mismo tipo. La recursividad se expresa a través de relaciones de recurrencia, que definen cada término de una sucesión en función de términos anteriores, junto con condiciones iniciales que permiten iniciar el proceso. Este método es ampliamente utilizado para modelar fenómenos en diversas áreas, como la economía, donde se aplica en el cálculo de intereses compuestos. Además, existe una estrecha relación entre inducción y recurrencia, ya que muchos resultados pueden demostrarse mediante ambos enfoques. En conjunto, estos conceptos fortalecen el razonamiento lógico y la resolución sistemática de problemas matemáticos complejos.

[1] Maestro en Matemáticas. Docente en Tecnológico Nacional de México. Correo electrónico: antonio.pc@itz.edu.mx

[2] Licenciada en Matemáticas. Docente en Tecnológico Nacional de México.

Palabras clave: *inducción matemática, algoritmos recursivos.*

Demostraciones por inducción

Cuando trabajamos proposiciones que involucran a los números naturales y que, además, se satisfacen para una cantidad particular de elementos, la primer pregunta que nos podemos formular es la siguiente: ¿podemos generalizar este resultado para cualquier valor natural $n = 1,2,3,\ldots$? De no ser así, ¿para qué valores de n se satisface la proposición?

El principio de inducción matemática es una herramienta usada ampliamente en matemáticas, que usualmente se enuncia a través de alguna afirmación $P(n)$ para $n \in \mathbb{N}$. Este principio es la principal herramienta para responder las preguntas iniciales que nos formulamos. Podría darse el caso de que $P(n)$ sea satisfecha para algunos valores de n y completamente falsa para otros (Velásquez, 2010). Para ilustrar esto, consideremos las afirmaciones siguientes:

1. $n^2 = n$. Podemos notar que esta afirmación se cumple para el valor 1, es decir, $P(1)$ es verdadera. Por otro lado, $P(n)$ es falsa para cualquier $n > 1$.
2. $n^2 > 0$. Claramente, el cuadrado de cualquier número natural es positivo, es decir, $P(n)$ es verdadera para toda n.

Definición (principio de inducción matemática): *Sea n_0 un número natural fijo y sea $P(n)$ una propiedad definida para cada número natural $n \geq n_0$. Además, supongamos que se satisfacen las condiciones siguientes:*

i) *$P(n_0)$ es verdadera.*

ii) *Si para todo natural $k \geq n_0$, $P(k)$ es verdadera, entonces $P(k+1)$ también lo es.*

Entonces decimos que para todo entero $n \geq n_0$, $P(n)$ es verdadera. A ii) *le llamamos hipótesis de inducción o hipótesis inductiva (idem).*

Este principio fue usado inicialmente en trabajos de Francesco Maurolico en 1575. Años después, Pierre de Fermat y Blaise Pascal utilizaron esta técnica, pero fue hasta 1883 que Augustus de Morgan describió de manera rigurosa esta técnica y la nombró *Inducción matemática* (“Inducción matemática”, 2024).

Ejemplo 1. Para ilustrar mejor este principio, partamos de un ejemplo clásico: supongamos que buscamos la suma (S_{100}) de los primeros 100 números naturales, es decir, $S_{100} = \sum_{i=1}^{100} i$

Este problema fue resuelto por Carl Friederich Gauss y después generalizado en la fórmula siguiente:

$$S_n = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$ (Chávez y González, 2018; De Nápoli, 2020).

Para demostrar esto, notemos que $P(1)$ es verdadera, ya que para $n = 1$ tenemos que $\frac{1(1+1)}{2} = 1$. Ahora supongamos que para cada entero $k \geq 1$, $P(k)$ es verdadera. Demostremos que $P(k+1)$ es verdadera, es decir,

$$S_{k+1} = \frac{(k+1)(k+2)}{2}$$

Dado que $P(k)$ es verdadera, tenemos que

$$S_k = \frac{k(k+1)}{2}$$

Luego, consideremos

$$\begin{aligned} S_{k+1} &= \sum_{i=1}^{k+1} i \\ &= \sum_{i=1}^{k} i + (k+1) \\ &= S_k + (k+1) \leftarrow \text{Aplicamos hipóteis de inducción,} \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ S_{k+1} &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

Por lo tanto, la fórmula se satisface para cualquier $n \in \mathbb{N}$.

Ejemplo 2. Demuestre que la suma de los cuadrados de los primeros n números naturales está dada por la fórmula

$$S_n = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Primero, notemos que $P(1)$ es verdadera, ya que

$$1 = \frac{1(1+1)(2 \cdot 1 + 1)}{6} = \frac{(2)(3)}{6}$$

Ahora asumimos que para todo $k \geq 1$, $P(k)$ es verdadera, esto es

$$S_k = \sum_{i=1}^{k} = \frac{k(k+1)(2k+1)}{6}$$

Nuestro objetivo ahora es demostrar que

$$S_{k+1} = \frac{(k+1)(k+2)(2k+3)}{6}$$

Tenemos que

$$\begin{aligned}
S_{k+1} &= \sum_{i=1}^{k+1} i^2 \\
&= \sum_{i=1}^{k} i^2 + (k+1)^2 \leftarrow \text{Aplicamos hipótesis de inducción,} \\
&= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 \\
&= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} \\
&= \frac{(k+1)(k(2k+1) + 6(k+1))}{6} \\
&= \frac{(k+1)(2k^2 + k + 6k + 6)}{6} \\
&= \frac{(k+1)(k+2)(2k+3)}{6}
\end{aligned}$$

Por lo tanto, la fórmula es verdadera para cualquier $n \in \mathbb{N}$.

Ejemplo 3. Ahora encontremos para que valores de n se satisface la desigualdad

$$2^n > 2n + 1$$

Notemos que para $n = 1, 2$ no se satisface, pues $2^1 < 2(1) + 1 = 3$ y $2^2 < 2(2) + 1 = 5$. Por otro lado, para $n = 3$ tenemos que $2^3 > 2(3) + 1$, es decir, $P(3)$ es verdadera. Ahora, supongamos que para todo $k \geq 3$ se satisface que $P(k)$ es verdadera, es decir, se satisface

$$2^k > 2k + 1$$

Ahora, nuestro objetivo es demostrar que $P(k+1)$ es verdadera, es decir,

$$2^{k+1} > 2k + 3$$

Tenemos que $2^{k+1} = 2^k \cdot 2$, por lo tanto, tenemos que para $2k + 2 > 3$ se cumple

$$2^{k+1} = 2 \cdot 2^k \quad > 2(2k+1) = 4k + 2 = 2k + (2k+2) > 2k + 3$$

Ejemplo 4. Demuestre que $\sum_{i=1}^{n} r^i = \frac{r^{n+1}-1}{r-1}$ para todo $n \in \mathbb{N}$ y $r \neq 1$.

Notemos que $P(n)$ está dado por

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

Por un lado, tenemos que

$$r^0 = 1 = \sum_{i=0}^{0} r^i$$

Por otro lado, tenemos

$$\sum_{i=0}^{0} r^i = \frac{r^{0+1} - 1}{r - 1} = 1$$

Concluimos que $P(0)$ es verdadera. Seguimos con el paso inductivo, es decir, asumimos que $P(k)$ es verdadera para algún entero $k \geq 0$, es decir, se satisface que

$$\sum_{i=0}^{k} = \frac{r^{k+1} - 1}{r - 1}$$

Veamos que $P(k+1)$ es verdadera. Tenemos que

$$\sum_{i=0}^{k+1} r^i = \sum_{i=0}^{k} r^i + r^{k+1}$$

Luego, dado que $P(k)$ es verdadera, se sigue que

$$\begin{aligned}
\sum_{i=0}^{k+1} r^i &= \frac{r^{k+1} - 1}{r - 1} + r^{k+1} \\
&= \frac{r^{k+1} - 1 + (r-1)(r^{k+1})}{r - 1} \\
&= \frac{r^{k+1} - 1 + r^{k+2} - r^{k+1}}{r - 1} \\
&= \frac{r^{k+2} - 1}{r - 1}
\end{aligned}$$

Concluimos que $P(n)$ es verdadera para todo $n \geq 0$.

Algoritmos recursivos

Uno de los aspectos más importantes de la programación es la recursividad, un concepto que aparece infinidad de veces en la naturaleza. Tales problemas consisten en resolver de manera directa un problema para algún conjunto de datos, y usar estos últimos para resolver el problema para el resto de datos. Una de las maneras de abordar problemas recursivos es a través de sucesiones (Ugarte, 2016).

Se puede definir una sucesión a través de su n-ésimo término, es decir, dar una fórmula para obtener el término a_n de la sucesión. De esta manera, podemos conocer cada término de la sucesión de manera inequívoca. De manera alternativa, se puede definir una sucesión a través de la recursividad, que es el proceso de definir una función que sea llamada a sí misma para de manera repetitiva (recursiva) hasta obtener una solución. En muchas ocasiones, esta función se denomina relación de recurrencia, y define cada elemento en términos de los elementos anteriores de la sucesión (Benítez, 2019).

Definición (relación de recurrencia)

Para una sucesión de términos $a_0, a_1, a_2, \ldots$, se define la relación de recurrencia como una fórmula que relaciona para cada k, el término a_k con los términos $a_{k-1}, a_{k-2}, \ldots, a_{k-i}$ (para $k - i \geq 0$). Las **condiciones iniciales** determinan los valores de $a_0, a_1, \ldots, a_m$ para algún entero fijo $m \geq 0$ (UNAM, s.f.; Doerr y Levasseur, s.f.).

Ejemplo 5. Consideremos una sucesión $c_0, c_1, c_2, \ldots$ de manera que para $k \geq 2$,

$$\begin{aligned} c_k &= c_{k-1} + kc_{k-2} + 1 \\ & c_0 = 1 y c_1 = 2 \end{aligned}$$

Para encontrar los términos $c_2, c_3 y c_4$ hacemos

$$\begin{aligned} c_2 &= c_1 + 2c_0 + 1 \\ &= 2 + 2(1) + 1 \\ c_2 &= 5 \\ c_3 &= c_2 + 3c_1 + 1 \\ &= 5 + 3(2) + 1 \\ c_3 &= 12 \\ c_4 &= c_3 + 4c_2 + 1 \\ &= 12 + 4(5) + 1 \\ c_4 &= 33 \end{aligned}$$

Sin embargo, una relación de recurrencia no es única, sino que se puede expresar de distintas maneras, y aun así para cada $k \geq 0$ obtener el mismo resultado para el término a_k. Para ilustrar esto, consideremos la sucesión dada por las relaciones de recurrencia:

$$\begin{aligned} s_k &= 3s_{k-1} - 1 \text{para} \quad k \geq 1, y \\ s_{k+1} &= 3s_k - 1, \text{para} \quad k \geq 0 \end{aligned}$$

Cabe destacar que, a pesar de que la recurrencia y la inducción son procedimientos distintos, se pueden demostrar algunos resultados por medio de inducción o de recurrencia, o incluso por medio de ambos procedimientos. Ilustramos esto en el siguiente ejemplo.

Ejemplo 5. Demostremos que el n-ésimo número impar (a_n) mayor que cero está dado por $2n - 1$. Para esto, consideramos el primer término, es decir, $n = 1$. Tenemos que

$$a_1 = 2(1) - 1 = 1$$

Supongamos que $P(k)$ es verdadera para algún $k \geq 0$. Tenemos que

$$a_k = 2k - 1$$

El objetivo es demostrar que $P(k + 1)$ es verdadera. Notemos que $a_{k+1} = a_k + 2$, ya que $a_k + 1$ es un número par. Luego, consideremos

$$\begin{aligned} a_{k+1} &= a_k + 2 \\ &= (2k - 1) + 2 \\ &= 2k + 2 - 1 \\ a_{k+1} &= 2(k + 1) - 1 \end{aligned}$$

Para este resultado usamos inducción al asumir que la relación de recurrencia era cierta para algún entero positivo k.

Además, podemos definir la recurrencia para conjuntos, que es una forma de construir cualquier elemento de un conjunto A a partir de un subconjunto $\mathcal{B} \subseteq A$ de elementos ya construidos (llamados base).

Ejemplo 6. Consideremos al conjunto de los números naturales. La base $\mathcal{B}$ de este conjunto es $\mathcal{B} = \{1\}$, y la relación de recurrencia está dada por $a_n = n + 1$.

Ejemplo 7. Uno de los ejemplos más comunes donde la recurrencia es extremadamente útil es en economía y finanzas, en particular en el cálculo de intereses compuestos con una tasa de impuestos periódica i. Para esto, definimos

n = n-ésimo periodo
a_n = Cantidad de dinero a finales del periodo n
a_0 = Cantidad inicial de dinero.

De esta manera, para cada periodo inicial, $n = 1$, tenemos que

$$a_1 = a_0 + ia_0$$

En general, para cualquier periodo k, se tiene que

$$a_k = a_{k-1} + ia_{k-1}$$

Notemos que el conjunto base es $\mathcal{B} = \{a_0\}$ y la relación de recurrencia es

$$a_n = a_{n-1} + ia_{n-1}$$

Referencias

Benítez, R. (2019). *Fundamentos de programación con Python.* Universidad de Chile; Departamento de Ciencias de la Computación.

Chávez, E., y González, C. (2018). *Matemáticas I. Libro de texto para el nivel medio superior*. Universidad Nacional Autónoma de México.

De Nápoli, P. L. (2020). *Más sobre inducción*. Departamento de Matemática; Facultad de Ciencias Exactas y Naturales; Universidad de Buenos Aires. https://mate.dm.uba.ar/~pdenapo/apuntes-algebraI/2020/2do-cuatrimestre/clase-07-induccion.pdf

Doerr, A., y Levasseur, K. [LibreTexts Español]. (s.f.). *Relaciones de recurrencia.* LibreTexts Español. https://es.libretexts.org/

Inducción matemática [Wikipedia] (2024, 5 de noviembre). En *Wikipedia*. https://es.wikipedia.org/wiki/Inducci%C3%B3n_matem%C3%A1tica

Ugarte, G. (2016). *Matemáticas discretas aplicadas a la computación.* Universidad Nacional de Colombia.

Universidad Nacional Autónoma de México. (s.f.). *Fascículo de matemáticas discretas I.* UNAM.

Velázquez, T. J. (2010). *Fascículo de inducción matemática*. Universidad Nacional Autónoma de México.

Ejercicios

Ejercicios de sistemas numéricos

Ejercicios de conversión entre los sistemas numéricos

Ejercicio 5

Binario	Octal	Decimal	Hexadecimal
110011			
	532		
		123	
			AA

Ejercicio 6

Binario	Octal	Decimal	Hexadecimal
100010			
	23		
		64	
			A1

Respuestas

Del ejercicio 5

Binario	Octal	Decimal	Hexadecimal
110011	63	51	33
101011010	532	346	15A
1111011	173	123	7B
10101010	252	170	AA

Respuesta del ejercicio 6

Binario	Octal	Decimal	Hexadecimal
100010	42	34	22
10011	23	19	13
1000000	100	64	40
10100001	241	161	A1

Ejercicios de sumas en binario

1	2	3	4	5	6	7
10101	111	110 0 1 1	1 1 1 0 0 0	1 1 1 1 1	1 0 0 1 1 1	1 1 1 1 1
11100	110	111 0 0 0	1 0 1 0 1 0	1 0 1 0 1	1 0 1 0 1 0	1 1 0 1 1
110001	110	1101 0 1 1	1 1 1 1 1 1	1 0 0 1 0	1 1 0 0 1 1	1 0 1 0 1
	10011		1 0 1 0 0 0 0 1	100 0 1 1 0	100 0 0 1 0 0	100 1 1 1 1

Ejercicios de restas en binario

1	2	3	4	5
1 1 0 0	1 1 0 1 1	1 1 1 1 1	1 1 0 0 1 1	1 1 1 0 0 0
1 0 1	1 0 1 0 1	1 1 0 1	0 1 1 0 1	0 0 1 0 1 1
1 1 1	**0 0 1 1 0**	**1 0 0 1 0**	**1 0 0 1 1 0**	**1 0 1 1 0 1**

Multiplicaciones en binario

1	2	3	4	5
11101 x110	111x 110	111 0 1 1x1110	1 1 1 0 0 0x1010	1 1 1 1 1 x 1 0 1 0 1
0 0 0 0 0	0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1
1 1 1 0 1	1 1 1	1 1 1 0 1 1	1 1 1 0 0 0	0 0 0 0 0
1 1 1 0 1	1 1 1	1 1 1 0 1 1	0 0 0 0 0 0	1 1 1 1 1
1 0 1 0 1 1 1 0	1 0 1 0 1 0	1 1 1 0 1 1	1 1 1 0 0 0	0 0 0 0 0
		1 1 0 0 1 1 1 0 1 0	1 0 0 0 1 1 0 0 0 0	1 1 1 1 1
				1 0 1 0 0 0 1 0 1 1

Divisiones en binario

Ejercicios

```
        101 000
101 | 110 01 100
      101
      001 01
        1 01
        0 0 0 100
```

```
        1 1 1 1 1 1
100 | 111 1 1 1 1 1
      100
        11 1
        10 0
        011 1
          10 0
          0 11 1
            100
            011 1
              10 0
              0 11 1
                10 0
                  1 1
```

```
        0111010
101 | 100100100
      101
      1000
        101
        0 110
          101
          00 110
              101
              00 10
```

Sumas octales

Ej. 1	Ej.2	Ej.3	Ej.4	Ej. 5
777	423	752	237	567
547	647	476	676	413

Restas octales

Ej. 1	Ej. 2	Ej. 3	E. 4	E. 5	Ej. 6	Ej. 7
736	333	443	556	544	111	333
423	244	666	227	373	777	167

Multiplicaciones octales

1	2	3
6 5 4 3 x 1 3 5	3 2 3 4 1 x 5 6 7	7 4 3 2 5 x 1 2 3
4 1 3 5 7	2 7 1 0 4 7	2 6 5 1 7 7
+ 2 4 0 5 1	+ 2 3 6 5 0 6	+ 1 7 0 6 5 2
6 5 4 3	2 0 4 1 4 5	7 4 3 2 5
1 1 5 6 3 6 7	2 3 2 7 2 6 2 7	1 1 6 2 6 4 1 7

Divisiones en octal

4 3 6
152 | 732 5 4
650
62 5
47 6
12 7 4
11 7 4
01 0 0

6543
135 | 1156367
1056
0100 3
721
00626
564
0427
427
000

5 13 4
100 | 513 47 6
500
13 4
10 0
034 7
30 0
04 7 6
4 0 0
076

4 62 0
73 | 432 117
354
56 1
542
17 1
166
003 7

Sumas hexadecimales

Realizaremos algunos ejercicios de sumas:

Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5
F E 0 0	7 2 3 4	2 A 2 B 2 C	1 2 7 C E 0	4 3 7
C A F E	1 6 4 4	1 F 3 E 4 C	3 3 3 F E 1	7 1 7
1 C 8 F E	**8 8 7 8**	**4 9 6 9 7 8**	**4 5 B C C 1**	**B 4 E**

Restas hexadecimales

Se muestran siete ejemplos de restas:

Ej 1	Ej. 2	Ej. 3	Ej.4	Ej. 5	Ej. 6	Ej.7
CE0	ABC	B10	BECA	F0CA	FECA	CABE
527	247	999	127	437	769	167
7B9	875	177	BDA3	EC93	F761	C957

Multiplicaciones hexadecimales

Ej 1	Ej 2	Ej 3
D E D 0 x F E 0	C A F E x F 0 C A	B E C A X B E B E
0 0 0 0	7 E D E C	A 6 F 0 C
+ C 2 F 6 0	+ 9 8 3 E 8	+ 1 8 3 2 A E
D 0 E 3 0	0 0 0 0	A 6 F 0 C
D D 1 2 6 0 0	B E 4 E 2 2 7	8 3 2 A E
	B E E E 4 C 6 C	8 E 2 7 8 5 E C

Divisiones hexadecimales

```
       14AB
ABC | DDDB94
      ABC
      ------
      3 2  1 B
       2 A F 0
      ------
       0 7 2 B 9
         6 B 5 8
      ------
          7 6 1 4
           7 6 1 4
      ------
           0 0 0 0
```

```
     CEFA23
47 | 396 7 5FB5
     354
     ------
      042 7
        3 E 2
     ------
        0 4  5  5
          4  2  9
     ------
          0 2   C  F
             2   C  6
     ------
             0    0   9  B
                       8  E
     ------
                         0D5
                          D5
                           00
```

```
          CAFÉ
FEA | C9E70 3 9
      BEF8
      ------
       0AEF 0
         9F2 4
      ------
         0 F C C3
            E E B6
      ------
            0 E 0 D  9
               D E C C
                0 2 0 D
```

Ejercicios de conjuntos

Operaciones con conjuntos

Dado:

- A = {1,2,3,4}
- B = {3,4,5,6}
- U = {1,2,3,4,5,6,7,8}

Encuentra:

1. A∪B
2. A∩B
3. A−B
4. A^c

Soluciones:

1. A∪B = {1,2,3,4,5,6}
2. A∩B = {3,4}
3. A−B = {1,2}
4. A^c = {5,6,7,8}

Sea U = el conjunto de los números reales A = {x | x es una solución de $x^2 - 1 = 0$}

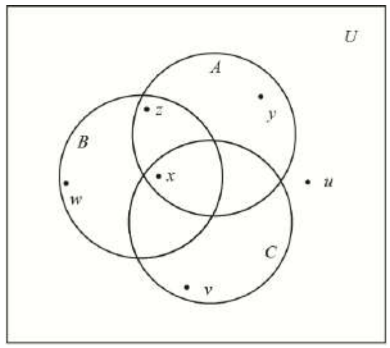
B = {-1,4}, responda si lo siguiente es verdadero o falso.

a) x ∈ A∩B∩C

b) y ∈ A∩C

c) y ∈ A∪B∪C

d) x ∈ B∩C

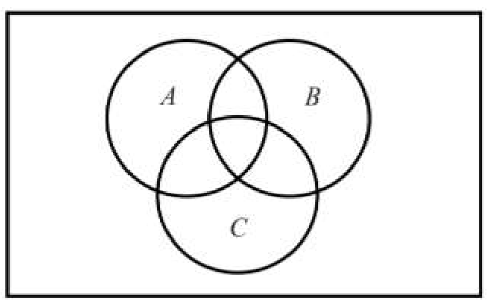


Soluciones

a) verdadero   b) verdadero   c) falso   d) falso

Ejercicios sobre diagramas de Venn

En el diagrama de Venn de la figura a) se muestran los conjuntos A, B y C.

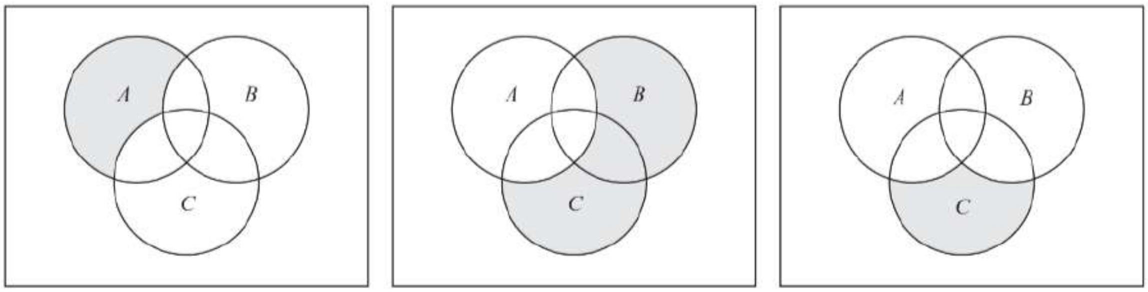


a)

Sombree los siguientes conjuntos:

a) A-(B ∪ C)   b) A^C ∩ (B ∪ C)   c) A^C ∩ (C-B).

Solución



a) b) c)

Ejercicio de conjunto finito

Determine cuáles de los siguientes conjuntos son finitos:

a) Rectas paralelas al eje x

b) Letras del alfabeto español

c) Enteros múltiplos de 5

d) Animales vivientes

Soluciones

a) Infinito; b) finito; c) infinito; d) finito.

Ejercicio de Conjunto potencia:

Dado A = [{a, b}, {c}, {d, e, f }].

a) Enumere los elementos de A.

b) Encuentre número de elementos n(A).

c) Encuentre el conjunto potencia de A.

Solución

a) Tres elementos: {a, b}, {c}, y {d, e, f }.

b) n(A) = 3.

c) P(A) tiene $2^3 = 8$ elementos como sigue:

P (A) = {A, [{a, b},{c}], [{a, b},{d, e, f }],[{c},{d, e, f }], [{a, b}], [{c}], [{d, e, f }], Ø}

Ejercicio de tipos de relaciones

Considere las cinco relaciones siguientes sobre el conjunto A = {1, 2, 3}:

R = {(1, 1), (1, 2), (1, 3), (3, 3)} ∅ = relación vacía

S = {(1, 1)(1, 2), (2, 1)(2, 2), (3, 3)} A × A = relación universal

T = {(1, 1), (1, 2), (2, 2), (2, 3)}

Determine si cada una de las relaciones indicadas sobre A es:

a) reflexiva; b) simétrica; c) transitiva; d) antisimétrica.

Solución

a) R no es reflexiva puesto que 2 ∈ A pero (2, 2) ∉R. T no es reflexiva puesto que (3, 3) ∉T y, en forma semejante, ∅ no es reflexiva. S y A × A son reflexivas.

b) R no es simétrica puesto que (1, 2) ∈ R pero (2, 1) ∉ R, y en forma semejante, T no es simétrica. S, ∅ y A × A son simétricas.

c) T no es transitiva puesto que (1, 2) y (2, 3) pertenecen a T, pero (1, 3) no pertenece a T. Las otras cuatro relaciones son transitivas.

d) S no es antisimétrica porque 1 ≠ 2 y ambos (1, 2) y (2, 1) pertenecen a S. En forma semejante, A × A no es antisimétrica. Las otras tres relaciones son antisimétricas.

Ejercicios de funciones

En el ejercicio A = {a,b,c,d} y B = {1,2,3}. Determine si las relaciones R de A en B es una función. Y si lo es, dé su codominio.

a) R={(a,1),(b,2),(c,1),(d,2)}

b) R={(a,3),(b,2),((c,1)}

Solución:

a) Observemos R:

- a→1
- b→2
- c→1
- d→2

Cada elemento de A (a,b,c,d) tiene exactamente una imagen en B. Por lo tanto, R es una función**.**

Determinar el codominio

El codominio es el conjunto B, pero la imagen de la función es el subconjunto de B con los valores que realmente se usan: Im(R) = {1,2} por lo que podemos afirmar que el codominio = {1,2}

a) Observemos R:

- $a \rightarrow 3$
- $b \rightarrow 2$
- $c \rightarrow 1$

En este caso:

Cada elemento de A relacionado en R (a,b,c) tiene una única imagen en B.

Sin embargo, el elemento $d \in A$ no está relacionado con ningún elemento de B.

Una relación R es una función si cada elemento de A está relacionado exactamente con un único elemento de B, la relación R no es una función.

Ejercicios de tipos de funciones

Sea f una función de A en B. Determine si la función f es inyectiva, suprayectiva o biyectiva.

A = B = Z;

f(a) = a-1

Solución

Verificar si f es inyectiva

Supongamos que $f(a_1) = f(a_2)$.

Esto implica: $a_1 - 1 = a_2 - 1$

Simplificando: $a_1 = a_2$

Por lo tanto, f es inyectiva, ya que valores diferentes de a producen valores diferentes de f(a).

Verificar si f es suprayectiva

Para que f sea subyectiva, todo $b \in \mathbb{Z}$ debe ser alcanzado por algún $a \in \mathbb{Z}$.

Sea $b \in \mathbb{Z}$. Queremos encontrar un $a \in \mathbb{Z}$ tal que: $f(a) = b$

Sustituyendo $f(a) = a - 1$f, tenemos: $a - 1 = b \implies a = b + 1$

Como $b + 1 \in \mathbb{Z}$, existe un $a \in \mathbb{Z}$ para cada $b \in \mathbb{Z}$.

Por lo tanto, f es suprayectiva.

La f es biyectiva ya que la f es inyectiva y suprayectiva.

Ejercicio $A = \mathbb{R}$, $B = \{x|x \text{ es real y } x \geq 0\}$

$F(a) = |a|$

Solución

Análisis de la función:

Verificar si es inyectiva:

La función no es inyectiva, ya que $f(a_1) = f(a_2)$ no implica que $a_1 = a_2$.

Por ejemplo, $f(3) = f(-3) = 3$, pero $3 \neq -3$.

Verifica si es sobreyectiva:

La función es sobreyectiva, porque para cada $b \in B$ (es decir, $b \geq 0$), siempre existe un $a \in \mathbb{R}$ tal que $f(a) = b$ (por ejemplo, $a = b$ o $a = -b$).

Por lo que la función no es inyectiva, pero es sobreyectiva y no es biyectiva.

Ejercicio de cardialidad

Encuentre el número cardinal de cada conjunto:

a) $A = \{a, b, c, . . . , y, z\}$,

b) $B = \{x \mid x \in N, x\ 2 = 5\}$,

Solución

a) $|A| = 29$, puesto que en el alfabeto español hay 29 letras.

b) $|B| = 0$ puesto que no existe ningún entero positivo cuyo cuadrado sea 5; es decir, B es vacío.

Ejercicio de proposiciones

Sean p “Hace frío” y q “Está lloviendo”. Proporcionar una oración coloquial sencilla que describa cada una de las siguientes proposiciones:

a) $\neg p$

b) $p \vee q$

c) $p \wedge q$

d) $q \vee \neg q$

Solución

En cada caso, $\wedge$, $\vee$ y $\sim$ se traducen por “y”, “o” y “es falso que” o “no”, respectivamente, y luego se simplifica la oración en lenguaje coloquial.

a) No hace frío.

b) Hace frío y está lloviendo.

c) Hace frío o está lloviendo.

d) Está lloviendo o no hace frío.

Problemas avanzados: ejercicios que incluyen aplicaciones prácticas y teóricas de los conceptos aprendidos.

Propiedades de conjuntos

Ejercicio: se hace una encuesta de los medios de transporte urbano más comunes. A cada persona se le pregunta si el CAMIÓN, el METRO o el AUTOMÓVIL es el medio más usado para ir al trabajo. Se permite más de una respuesta. El resultado de la encuesta se da a continuación:

a) 30 personas escogieron el CAMIÓN

b) 35 personas escogieron el METRO

c) 100 personas escogieron el AUTOMÓVIL

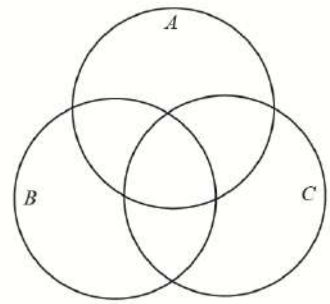
d) 15 personas escogieron el CAMIÓN y METRO

e) 15 personas escogieron el CAMIÓN y AUTOMÓVIL

f) 20 personas escogieron el METRO y AUTOMOVIL

g) 5 personas escogieron los tres medios de transporte

¿Cuántas personas respondieron a la encuesta?



Solución. Sean A, B y C los conjuntos de las personas que escogieron CAMIÓN, METRO y AUTOMOVIL, respectivamente, (véase la figura). Se sabe que $|A| = 35$, $|B| = 30$, $|C| = 100$, $|A \cap B| = 15$, $|A \cap C| = 15$, $B \cap C| = 20$, $|A \cup B \cup C| = 5$. Así pues, el número total de personas que respondió es

$$|A \cup B \cup C| = (30 + 35 + 100) - (15 + 15 + 20) + 5 = 120$$

Conjuntos finitos

En una universidad cada estudiante de humanidades debe acreditar un curso A de matemáticas y un curso B de ciencias. En una muestra de 140 estudiantes de segundo año se observó lo siguiente:

c) 60 acreditaron A

d) 45 acreditaron B

e) 20 acreditaron tanto A como B

Use un diagrama de Venn para determinar el número de estudiantes que acreditaron:

a) Por lo menos uno de A y B;

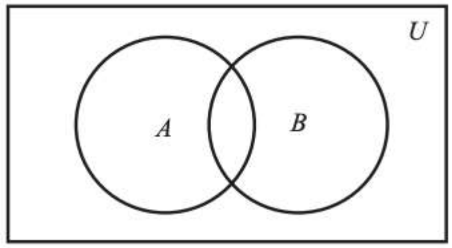
b) exactamente uno de A o B;

c) ni A ni B.

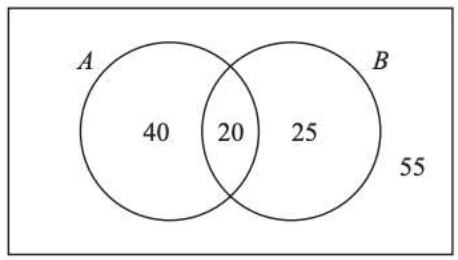
Al escribir los datos anteriores en notación de conjuntos se obtiene:

$$|A| = 60, |B| = 45, |A \cap B| = 20, |U| = 140$$

Se dibuja un diagrama de Venn de los conjuntos A y B, como en la siguiente figura.



Luego, se asignan números a las cuatro regiones:



20 acreditaron tanto A como B, de modo que $|A \cap B| = 20$.

$60 - 20 = 40$ acreditaron A pero no B, por lo que $|A\text{-}B| = 40$.

$45 - 20 = 25$ acreditaron B pero no A, por lo que $|B\text{-}A| = 25$.

$140 - 20 - 40 - 25 = 55$ no acreditaron A ni B.

Por el diagrama de Venn:

a) $20 + 40 + 25 = 85$ acreditaron A o B. Ahora, por el principio de inclusión-exclusión:

$|A \cup B| = |A| + |B| - |A \cap B| = 60 + 45 - 20 = 85$

b) $40 + 25 = 65$ acreditaron exactamente uno de los cursos. Es decir, $|A \oplus B| = 65$.

c) 55 no acreditaron ninguno de los cursos; es decir,

$|A^C \cap B^C| = |(A \cup B)^C| = 140 - 85 = 55$.

Ejercicios de lógica

Ejercicios de sustitución

Ejercicios:

$P = F \quad R=F \quad Q=V \quad S=V \quad T = V$

1- $(p \rightarrow q) \wedge (q \rightarrow R) = (F \rightarrow V) \wedge (V \rightarrow F) = (V) \wedge (F) = F$

2- $(p \rightarrow q) \rightarrow R= \quad (F\rightarrow V)\rightarrow F = (V)\rightarrow F \quad = F$

3- $P \rightarrow (q \rightarrow R)= F\rightarrow(V\rightarrow F) \; = F\rightarrow(F) \quad =V$

4- $(s\rightarrow(p \wedge \neg R)) \wedge \{[p\rightarrow(R \vee q)] \wedge s\} =$

$(V\rightarrow(F\wedge\neg F))\wedge((F \rightarrow(Fv\ V)) \wedge V)$

$(V\rightarrow(F))\wedge((F\rightarrow(V))\wedge V) =$

$(V\rightarrow F)\wedge((F\rightarrow V)\wedge V) =$

$F\wedge(V\wedge V)= \quad F\wedge V= \quad F$

5- $((p\wedge\neg q)\rightarrow(q\wedge R))\rightarrow(s\vee\neg q) =$

$((F\wedge\neg V)\rightarrow(V\wedge F))\rightarrow(Vv\neg V)=$

$(F\rightarrow F)\rightarrow(V)= (F\rightarrow F)\rightarrow(V)=V\rightarrow V=V$

6- $((p \wedge R) \leftrightarrow (r\ v\ s)) \rightarrow q =$

$((F \wedge F) <\!\rightarrow (F\ \ v\ V))\rightarrow V =$

$= (F <\!\rightarrow V)\rightarrow V = F \rightarrow V = V$

7- $((q\wedge s)v(R \rightarrow q)) \leftrightarrow (q \wedge R) = ((V\wedge V)v(F\rightarrow V)) \leftrightarrow (V\wedge F) = (VvV) \leftrightarrow (F) = V \leftrightarrow F = F$

8- $(q \wedge R \wedge s \wedge p)\rightarrow q = (V\wedge F\wedge V\wedge F)\rightarrow V = F\rightarrow V = V$

9- $((p \wedge q)v(p \wedge R)v(p\wedge s)) \leftrightarrow R= \; ((F \wedge V)v(F \wedge F)v(F\wedge V)) \leftrightarrow F$

$((F)v(F)v(F)) \leftrightarrow F = V$

10- $\neg(p\rightarrow(\neg(q \;\; \rightarrow s)\wedge r))= \quad \neg(F\rightarrow(\neg(V\rightarrow V)\wedge F))= \; \neg(F\rightarrow(\neg(V)\wedge F))= \; \neg(F\rightarrow(\neg(V)\wedge F))= \neg(F\rightarrow F)= \neg(V)=F$

11- $\neg(q \rightarrow r)\wedge(s \rightarrow p)= \neg(V\rightarrow F)\wedge(V\rightarrow F)= \neg(F)\wedge(F)=V\wedge F=F$

12- $\neg(r \wedge p) \leftrightarrow (q \rightarrow r)= \; \neg(F\wedge F) \leftrightarrow (V\rightarrow F)= \neg(F) \leftrightarrow (F)=V\leftrightarrow F=F$

13- $(T\leftrightarrow\neg(p\rightarrow r)\wedge(svq))= (V\leftrightarrow\neg(F\rightarrow F)\wedge(VvV)=f$

Ejercicios de tablas de verdad

1.- (p→q)^(q→r) =

P	**Q**	**R**	**P→Q**	**(Q→r)**	**(p→q)^(q→r)**
V	V	V	V	V	V
V	V	F	V	F	F
V	F	V	F	V	F
V	F	F	F	V	F
F	V	V	V	V	V
F	V	F	V	F	F
F	F	V	V	V	V
F	F	F	V	V	V

Solución

2.- (p→q)→r

P	**Q**	**R**	**P→Q**	**(p→q) →r**
V	V	V	V	V
V	V	F	V	F
V	F	V	F	V
V	F	F	F	V
F	V	V	V	V
F	V	F	V	F
F	F	V	V	V
F	F	F	V	F

Solución

3.- p→(q→r) = si fuera todo verdadero sería tautología.

P	**Q**	**R**	**q→r**	**p→(q →r)**
V	V	V	V	V
V	V	F	F	F
V	F	V	V	V
V	F	F	V	V
F	V	V	V	V
F	V	F	F	V
F	F	V	V	V
F	F	F	V	V

Solución

4.- (s→(p∧¬r))∧((p→(rVq))∧s) = cuando son todos falsos se llama contradicción.

p	q	r	S	rv q	¬ r	p∧¬ r	s→(p∧¬ r)	p→(rV q)	((p→(rVq)) ∧s)	(s→(p∧¬r))∧((p→(rVq))∧s)
V	V	V	V	V	F	F	F	V	V	F
V	V	V	F	V	F	F	V	V	F	F
V	V	F	V	V	V	V	V	V	V	V
V	V	F	F	V	V	V	V	V	F	F
V	F	V	V	V	F	F	F	V	V	F
V	F	V	F	V	F	F	V	F	F	F
V	F	F	V	F	V	V	V	F	F	F
V	F	F	F	F	V	V	V	V	F	F
F	V	V	V	V	F	F	F	V	V	F
F	V	V	F	V	F	F	V	V	F	F
F	V	F	V	V	V	F	F	V	V	F
F	V	F	F	V	V	F	V	V	F	F
F	F	V	V	V	F	F	F	V	V	F
F	F	V	F	V	F	F	V	V	F	F
F	F	F	V	F	V	F	F	V	V	F
F	F	F	F	F	V	F	V	V	F	F

5 .- ((p∧¬q)→(q∧r))→(s∨¬q) 2^4=16 renglones 8 verdaderos y 8 falsos

P	q	r	S	¬q	p∧¬q	q∧r	((p∧¬q)→(q∧r))	s∨¬q	((p∧¬q)→(q∧r))→(s∨¬q)
V	V	V	V	F	F	V	V	V	V
V	V	V	F	F	F	V	V	F	F
V	V	F	V	F	F	F	V	V	V
V	V	F	F	F	F	F	V	F	F
V	F	V	V	V	V	F	F	V	V
V	F	V	F	V	V	F	F	V	V
V	F	F	V	V	V	F	F	V	V
V	F	F	F	V	V	F	F	V	V
F	V	V	V	F	F	V	V	V	V
F	V	V	F	F	F	V	V	F	F
F	V	F	V	F	F	F	V	V	V
F	V	F	F	F	F	F	V	F	F
F	F	V	V	V	F	F	V	V	V
F	F	V	F	V	F	F	V	V	V
F	F	F	V	V	F	F	V	V	V
F	F	F	F	V	F	F	V	V	V

6 .- ((p ∧ r) <→ (r v s))→q

p	q	r	S	p∧ r	RvS	((p ∧ r) <→ (r v s))	((p ∧ r) <→ (r v s))→q
V	V	V	V	V	V	V	V
V	V	V	F	V	V	V	V
V	V	F	V	F	V	F	V
V	V	F	F	F	F	V	V
V	F	V	V	V	V	V	F
V	F	V	F	V	V	V	F
V	F	F	V	F	V	F	V
V	F	F	F	F	F	V	F
F	V	V	V	F	V	F	V
F	V	V	F	F	V	F	V
F	V	F	V	F	V	F	V
F	V	F	F	F	F	V	V
F	F	V	V	F	V	F	V
F	F	V	F	F	V	F	V
F	F	F	V	F	V	F	V
F	F	F	F	F	F	V	F

7. ((q^s)v(r →q)) ↔ (q ^r) =

Q	R	S	q∧ S	(R →Q)	(q^R)	((q∧s)v(R→q))	((q∧s)v(R→q)) ↔ (q^R)
V	V	V	V	V	V	V	V
V	V	F	F	V	V	V	V
V	F	V	V	V	F	V	F
V	F	F	F	V	F	V	F
F	V	V	F	F	F	F	V
F	V	F	F	F	F	F	V
F	F	V	F	V	F	V	F
F	F	F	F	V	F	V	F

8.-(q^r^s^p)→q

P	q	r	S	(q^R^s^p)	(q^R^s^p)→q
V	V	V	V	VV	V
V	V	V	F	F	V
V	V	F	V	F	V
V	V	F	F	F	V
V	F	V	V	F	V
V	F	V	F	F	V
V	F	F	V	F	V
V	F	F	F	F	V
F	V	V	V	F	V
F	V	V	F	F	V
F	V	F	V	F	V
F	V	F	F	F	V
F	F	V	V	F	V
F	F	V	F	F	V
F	F	F	V	F	V
F	F	F	F	F	V

Tautología

Cuando todo sale verdadero

9.- ((p^q) v (p^r) v (p^s)) ↔ r =

p	q	r	s	(p^q)	(p^r)	(p^s)	((p^q)v(p^r)v(p^s))	Resultado
V	V	V	V	V	V	V	V	V
V	V	V	F	V	V	F	V	V
V	V	F	V	V	F	V	V	F
V	V	F	F	V	F	F	V	F
V	F	V	V	F	V	V	V	V
V	F	V	F	F	V	F	V	V
V	F	F	V	F	F	V	V	F
V	F	F	F	F	F	F	F	V

F	V	V	V	F	F	F	F	F
F	V	V	F	F	F	F	F	F
F	V	F	V	F	F	F	F	V
F	V	F	F	F	F	F	F	V
F	F	V	V	F	F	F	F	F
F	F	V	F	F	F	F	F	F
F	F	F	V	F	F	F	F	V
F	F	F	F	F	F	F	F	V

Contingencia

10 .- ¬(p→(¬(q →s)^r))= 2^4 =16 renglones

P	Q	R	S	Q→S	¬ (Q→S)	¬(Q →S)^R	P→(¬(Q →S)^R)	¬(P→(¬(Q →S)^R))
V	V	V	V	V	F	F	F	V
V	V	V	F	F	V	V	V	F
V	V	F	V	V	F	F	F	V
V	V	F	F	F	V	F	F	V
V	F	V	V	V	F	F	F	V
V	F	V	F	V	F	F	F	V
V	F	F	V	V	F	F	F	V
V	F	F	F	V	F	F	F	V
F	V	V	V	V	F	F	V	F
F	V	V	F	F	V	V	V	F
F	V	F	V	V	F	F	V	F
F	V	F	F	F	V	F	V	F
F	F	V	V	V	F	F	V	F
F	F	V	F	V	F	F	V	F
F	F	F	V	V	F	F	V	F
F	F	F	F	V	F	F	V	F

11.- ¬(q →r)^(s →p) =

p		q	r	S	q→r	¬ (q→r)	(s →p)	¬(q →r)^(s →p)
V		V	V	V	V	F	V	F
V		V	V	F	V	F	V	F
V		V	F	V	F	V	V	V
V		V	F	F	F	V	V	V
V		F	V	V	V	F	V	F
V		F	V	F	V	F	V	F
V		F	F	V	V	F	V	F
V		F	F	F	V	F	V	F
F		V	V	V	V	F	F	F
F		V	V	F	V	F	V	F
F		V	F	V	V	F	F	F
F		V	F	F	F	V	V	V
F		F	V	V	V	F	F	F
F		F	V	F	V	F	V	F
F		F	F	V	V	F	F	F
F		F	F	F	V	F	V	F

12.- ¬(r ^p) ↔ (q →r)= 2^3= 8 renglones : 4 verdaderos y 4 falsos

P	Q	R	R^ P	(Q→R)	¬ (R^P)	¬(r ^p) ↔ (q →r)
V	V	V	V	V	F	F
V	V	F	F	F	V	F
V	F	V	V	V	F	F
V	F	F	F	V	V	V
F	V	V	F	V	V	V
F	V	F	F	F	V	F
F	F	V	F	V	V	V
F	F	F	F	V	V	V

13.- (t↔¬(p → r)^(s v q))= 5 proposiciones. 2^5=32 renglones: 16 verdaderos y 16 falsos

p	q	r	s	t	(p→ r)	¬ (p→r)	(s V q)	¬(p → r)^(s v q)	(t↔¬(p → Rr^(s v q))
V	V	V	V	V	V	F	V	F	F
V	V	V	V	F	V	F	V	F	V
V	V	V	F	V	V	F	V	F	F
V	V	V	F	F	V	F	V	F	V
V	V	F	V	V	F	V	V	V	V
V	V	F	V	F	F	V	V	V	F
V	V	F	F	V	F	V	V	V	V
V	V	F	F	F	F	V	V	V	F
V	F	V	V	V	V	F	V	F	F
V	F	V	V	F	V	F	V	F	V
V	F	V	F	V	V	F	F	F	F
V	F	V	F	F	V	F	F	F	V
V	F	F	V	V	F	V	V	V	V
V	F	F	V	F	F	V	V	V	F
V	F	F	F	V	F	V	F	F	F
V	F	F	F	F	F	V	F	F	V
F	V	V	V	V	V	F	V	F	F
F	V	V	V	F	V	F	V	F	V
F	V	V	F	V	V	F	V	F	F
F	V	V	F	F	V	F	V	F	V
F	V	F	V	V	V	F	V	F	F
F	V	F	V	F	V	F	V	F	V
F	V	F	F	V	V	F	V	F	F
F	V	F	F	F	V	F	V	F	V
F	F	V	V	V	V	F	V	F	F
F	F	V	V	F	V	F	V	F	V

F	F	V	F	V	V	F	F	F	F
F	F	V	F	F	V	F	F	F	V
F	F	F	V	V	V	F	V	F	F
F	F	F	V	F	V	F	V	F	V
F	F	F	F	V	V	F	F	F	F
F	F	F	F	F	V	F	F	F	V

Ejercicios de cuantificadores

Traducción lógica

"Todo número par es divisible por 2"

$\forall x \in Z, P(x) \rightarrow D(x)$

1. **Negación**

   Negar: $\exists x \in N, x<0$

Resultado: ∀x∈N, x≥0\forall x \in \mathbb{N},\ x \geq 0∀x∈N, x≥0

2. **Evaluación de verdad**

   ¿Es verdadera la proposición $\exists x \in N, x<0$?

   Respuesta: Falsa, porque no hay números naturales negativos.

3. **Traducción natural**

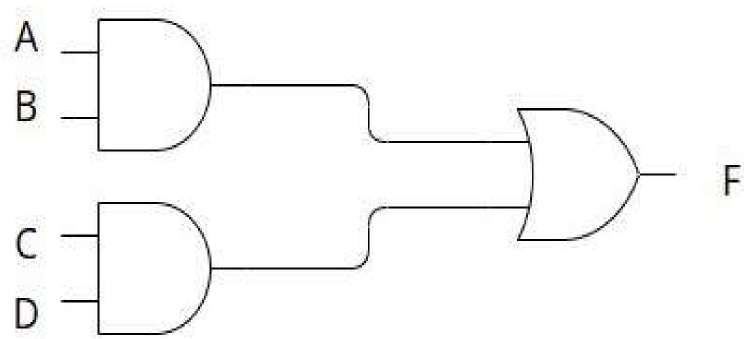
   $\forall x \in Z, x \neq 0 \rightarrow 1x \in Q$ "Todo número entero distinto de cero tiene inverso racional".

Ejercicios de Algebra booleana

1.- Usando las compuertas lógicas AND y OR construya el circuito lógico de la función:

$F = AB + CD$

Circuito lógico



2.-De acuerdo con la función lógica: $F = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$ desarrolle la reducción lógica y represente el circuito lógico que le corresponda.

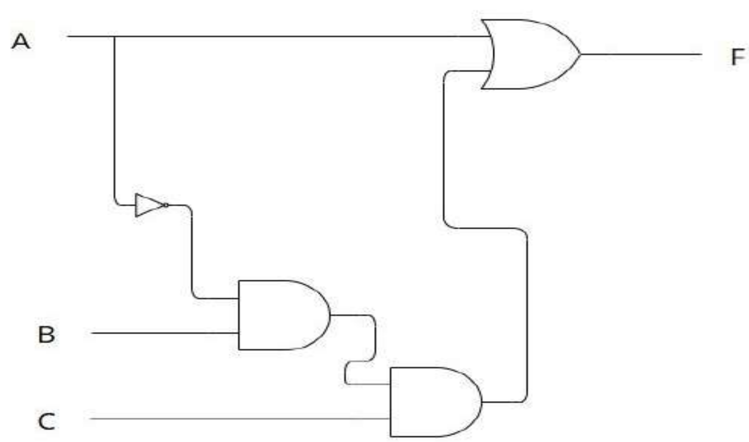
$F = \bar{A}BC + A\bar{B}(\bar{C} + C) + AB(\bar{C} + C)$

$F = \bar{A}BC + A\bar{B} + AB$

$F = \bar{A}BC + A(\bar{B} + B)$

$F = \bar{A}BC + A$

CIRCUITO LÓGICO

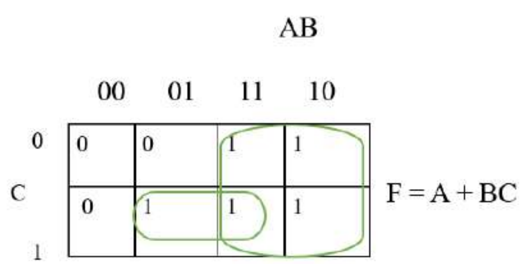


3.- Usando la tabla de verdad para: $F = \bar{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC,$ hallar el mapa de Karnaugh y el circuito lógico que corresponda.

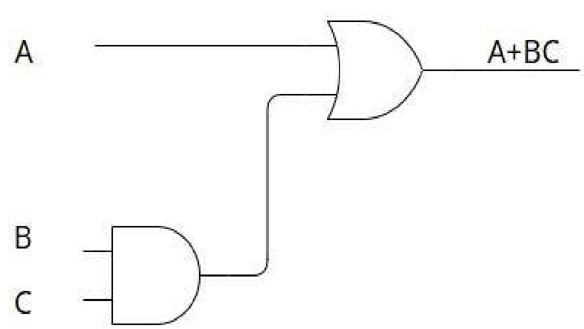
Tabla de verdad

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Mapa de Karnaugh

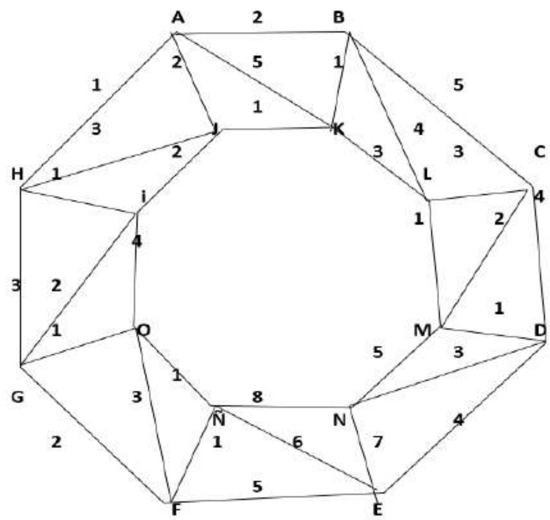


Circuito lógico



Ejercicios de árboles

1. Ejercicio: con la siguiente gráfica sacar un recorrido en profundidad si la raíz es A:

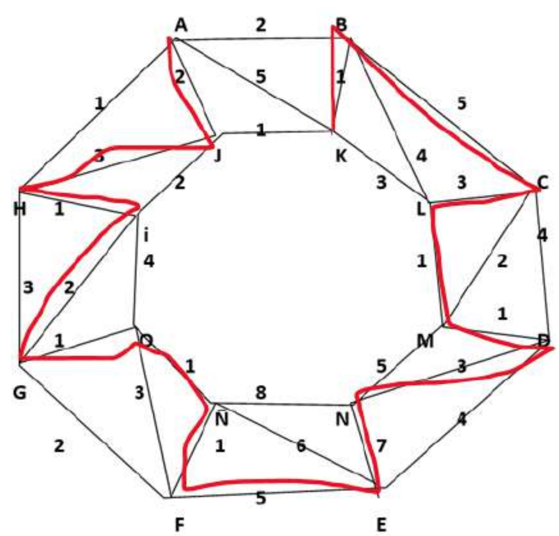


Si el punto inicial es A:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	ñ	o
	2	5	4	4	5	2	3	1	2	1	3	1	5	8	1

Costo total = 47

2. Sacar 3 recorridos diferentes iniciando en A:



Sobre los autores

Ortega de Ávila, Elsa

Doctora en Ciencias de la Educación por la Universidad Autónoma de Coahuila (México). Obtuvo la Maestría en Ingeniería con Orientación en Computación en la Universidad Autónoma de Zacatecas. Realizó la Licenciatura de Ingeniería en Sistemas Computacionales en el Tecnológico Nacional de México. Se ha desempeñado como profesora-investigadora en la misma institución como profesora de Licenciatura en Ciencias Básicas y posgrado en la Maestría de Administración, también ha colaborado en la UNID como docente de varias licenciaturas y posgrado en la Maestría de Educación y Pedagogía. En la actualidad es investigadora en nivel candidato de la SECIHTI, donde realiza investigaciones sobre educación, matemáticas y metodología de investigación. Recientemente publicó el libro *Autoaprendizajes emotivos: hábitos de estudio y metas definidas en la vida* (2024) y es primera autora del capítulo "Las TIC y el autoaprendizaje", del libro *Tecnología educativa: innovación y desafíos en el siglo XXI* de la Universidad de Guadalajara del Centro Universitario del Norte.

ORCID: https://orcid.org/0000-0003-0405-2978
Academia: https://www.academia.edu/?from_navbar=true&trigger=nav
Google Académico: https://scholar.google.es/citations?user=RjTGYEcAAAAJ&hl=es
ResearchGate: https://www.researchgate.net/profile/Elsa-De-Avila?ev=hdr_xprf

Alvarado de la Torre, Blanca Verónica

Maestra en Informática Administrativa de la Universidad Autónoma de Durango, campus Zacatecas (2015) e Ingeniera en Comunicaciones y Electrónica en la Universidad Autónoma de Zacatecas (1992). Actualmente se desempeña como profesora titular "C" de tiempo completo en el Departamento de Sistemas y Computación del Instituto Tecnológico de Zacatecas, donde ha desarrollado una trayectoria académica de más de tres décadas en formación de ingenieros en el área de tecnologías de la información. Sus principales líneas de trabajo e investigación se centran en tecnologías de la información y comunicaciones, desarrollo de *software*, electrónica analógica y digital, arquitectura de computadoras e interfaces de comunicación. Entre sus logros profesionales destaca su participación en el desarrollo de sistemas institucionales para la administración pública estatal, su liderazgo en proyectos de sistemas administrativos.

Arana Castillo, Jesús Javier

Maestro en Ciencias en Ingeniería Administrativa por el Instituto Tecnológico de Celaya (México). Obtuvo su Licenciatura en Ingeniería Industrial en Producción en el Instituto Tecnológico de Zacatecas. Se desempeña como jefe de Departamento de Ciencias Básicas y actualmente está cursando un Doctorado en Formación Profesional y Educativa. Coordina el evento Nacional de Ciencias Básicas del Instituto Tecnológico de Zacatecas, coordina el evento Olimpiada del Conocimiento de las preparatorias del Estado de Zacatecas, coordina las visitas de las secundarias a las instalaciones del Tecnológico de Zacatecas, coordina los cursos de nivelación de alumnos de nuevo ingreso, es docente en la carrera de Ingeniería Industrial del Tecnológico, participa como auditor en el Sistema de Gestión Ambiental.

Barajas Guerrero, Martín José

Maestro en Ciencias en Ingeniería Administrativa por el Instituto Tecnológico de Celaya (2006), cuenta con especialidad en Computación por la Universidad Autónoma de Zacatecas (1995) y es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Zacatecas (1992). Actualmente se desempeña como profesor titular "C" de tiempo completo y jefe de laboratorio de cómputo en el Departamento de Sistemas y Computación del Instituto Tecnológico de Zacatecas, donde ha desarrollado una trayectoria académica de más de tres décadas en formación de ingenieros en el área de tecnologías de la información. Sus principales líneas de trabajo e investigación se centran en tecnologías de la información y comunicaciones, desarrollo de *software*, computación aplicada, ciencia de datos e inteligencia artificial, así como tecnologías web y en la nube. Entre sus logros profesionales destaca su participación en el desarrollo de sistemas institucionales para la administración pública estatal, su liderazgo en proyectos de sistemas administrativos y financieros, así como su experiencia como desarrollador de aplicaciones web. Ha contribuido a la mejora de procesos académicos mediante la implementación de sistemas de control escolar y cuenta con certificación internacional como programador Java (SCJP). En cuanto a su producción académica reciente, destacan las publicaciones: *Ciencia de datos y decisiones empresariales* (2025) y *Optimización de la energía de Helmholtz mediante inteligencia artificial* (2024), en el área de computación aplicada.

ORCID: https://orcid.org/0000-0002-2443-9590

Bautista Acosta, Jorge Oliver

Maestro en Administración con especialidad en Comercio Internacional (Universidad Autónoma de Zacatecas). Tiene una especialidad en Computación (Universidad Autónoma de Zacatecas). Es Ingeniero en Comunicaciones y Electrónica (Universidad Autónoma de Zacatecas). Actualmente se desempeña como profesor de tiempo completo de nivel licenciatura dentro del Instituto Tecnológico de Zacatecas, perteneciente al Tecnológico Nacional de México. Ha realizado investigación sobre arquitecturas de cómputo, sistemas digitales, redes de telecomunicaciones y participado como coautor de los artículos *Propuesta de chat bot en la Tutoría del Tecnológico Nacional de México* (2021) e *Implicaciones de la deserción escolar a nivel superior en ingeniería en sistemas e informática* (2019).

Carrillo López, Néstor Alejandro

Maestro en Informática Administrativa, estudió su licenciatura como Ingeniero en Sistemas Computacionales en Programación, con cargo académico actual en el Tecnológico Nacional de México con sede en Zacatecas, se desempeña como docente de la Academia de Ingeniería Industrial, es el encargado del laboratorio de manufactura.
ORCID: https://orcid.org/0000-0002-2443-9590

De la Rosa Ibarra, José Francisco

Maestro en Matemática Educativa por la Universidad de Zacatecas. Obtuvo el grado de Ingeniero en Comunicaciones y Electrónica en la Universidad Autónoma de Zacatecas. Se ha desempeñado como profesor de licenciatura y posgrado en el Instituto Tecnológico de Zacatecas.

Gámez Aguilar, María del Rosario

Ingeniero en Materiales por el Instituto Tecnológico de Zacatecas (México). Se desempeña como profesora de licenciatura en el Instituto Tecnológico de Zacatecas.

García Castañon, Alfredo

Maestro en Informática Administrativa por la UAD campus Zacatecas, Ingeniero en Sistemas computacionales, egresado del Instituto Tecnológico de Zacatecas, docente del área de sistemas y computación del IT Zacatecas desde el año 2000 a la fecha, además de ser docente del posgrado en la Maestría en Sistemas Computacionales de la misma institución educativa, las líneas de investigación es sobre ingeniería de *software* y las publicaciones más recientes son los artículos: "Mobile application for healthy living" en la revista *Tecnología y Ciencia Aplicada*, y "Graph data analysis against money laundering", publicado en la misma revista. ORCID: https://orcid.org/0009-0001-3656-8293

González Torres, Iván Alejandro

Doctorante en Formación Profesional y Educativa. Maestro en Administración, por el Instituto Tecnológico de Zacatecas (México). Obtuvo su Licenciatura en Administración en el Instituto Tecnológico de Zacatecas. Se desempeña como jefe del Departamento de Actividades Extraescolares, realizando actividades de promoción y difusión cultural y deportiva, así como la organización de eventos extraescolares.

Hernández Yáñez, Cecilia Guadalupe

Licenciada en Matemáticas por la Unidad Académica de Matemáticas de la Universidad Autónoma de Zacatecas. Actualmente se desempeña como docente en el área de Ciencias Básicas en el Instituto Tecnológico de Zacatecas.

Martínez Cardona, Carlos Antonio

Maestro en Administración por la Universidad Autónoma de Zacatecas en la Facultad de Contaduría e Ingeniero en Sistemas Computacionales en el Tecnológico Nacional de Mexico Campus Instituto Tecnológico de Zacatecas. Se ha desempeñado como coordinador de las carreras de Ingeniería en Sistemas Computacionales e Informática, como jefe del depto. de Sistemas y Computación, asi como delegado sindical de la D-II-14 (D-V-163) de UNDESINTEC (Sección 61), y principalmente como profesor de licenciatura en el Instituto Tecnológico de Zacatecas. Actualmente es miembro del comité creador del COPTIZAC (Colegio de Profesionistas de Tecnologías de la Información de Zacatecas).

Navarro Favela, José Gabriel

Maestro en Sistemas Computacionales por el Instituto Tecnológico de Orizaba (México), también obtuvo la Maestría en Informática Administrativa en la Universidad Autónoma de Durango (México) y la Licenciatura en Sistemas de Computación Administrativa en el Instituto Tecnológico de Estudios Superiores de Monterrey (México). Actualmente es el encargado del Laboratorio de Desarrollo de Software del Instituto Tecnológico de Zacatecas, también se desempeña como docente del mismo instituto, además, se ha desempeñado como docente en la Facultad de Contabilidad y Administración de la Universidad Autónoma de Zacatecas. Publicó los artículos *Cómo desarrollar una línea de productos de software, un enfoque práctico* en el Congreso CIIA (Congreso Internacional de Informática Aplicada), realizado por el Instituto Tecnológica Superior de Misantla y el artículo *Uso del polimorfismo familiar para la implementación de una línea de productos de software con CaesarJ*.

Ortíz García, Mariana

Maestra en Sistemas Computacionales. Ha sido coautora de artículos de congreso, directora y codirectora de tesis de Licenciatura en Ingeniería en Sistemas Computacionales y la Maestría en Sistemas Computacionales del Instituto Tecnológico de Zacatecas en tópicos de *software* con desarrollo tecnológico con impacto en el sector agrícola, salud, así como recursos humanos en los sectores privado y público. Ha realizado investigaciones relacionadas con bases de datos de grafos como coautora que sirven como base para el modelado de redes sociales, detección de patrones y tendencias. Así mismo, es experta en el modelado de aplicaciones web y móviles fungiendo como líder en proyectos relacionados a la ingeniería de *software*.

ORCID: https://orcid.org/0000-0009-0003-5415
Scopus: 57982741400
Google Académico: https://scholar.google.com/citations?hl=es&view_op=list_wo

Ortiz Sánchez, Pedro Alfonso Guadalupe

Doctor en Administración y Doctor en Ingeniería, Maestro en Educación y en Tecnologías de la Comunicación y la Información. Profesor de Computación e Inteligencia Artificial, TecNM/campus Mérida, PTC Universidad del Valle de México. Soporte tecnológico en la

empresa, Arquitectura y Soluciones Informáticas de la ciudad de Mérida. Autor de diversas ponencias y artículos. Línea de investigación: aplicaciones científicas y educativas de las tecnologías de la comunicación y la información. Artículos recientes: *Excessive cell phone use and its implications for academic performance in engineering students* (2025) y *About the validation of a questionnaire on behavioral addiction to the internet and its relationship with the student population* (2024).
ORCID: https://orcid.org/0000-0002-2466-1837

Ortiz y Ojeda, Pedro Tomas

Doctor en Matemática Educativa. Profesor de licenciatura, maestría y doctorado en el ITTG, UNACH, UVG y UTS. Ha participado como jurado en concursos de ciencias básicas y en actividades de divulgación científica, así como en la actualización de planes y programas de estudio relacionadas con las ciencias básicas, también ha impartidos cursos a diversas instituciones nacionales. Autor de diversas ponencias y artículos, ha fungido como asesor y revisor de diversas tesis de licenciatura, maestría y doctorado. Realiza investigaciones en la línea de la epistemología y la cognición matemática. Participa en REEDIEM (Red de Investigadores Educativos en México) nodo Tuxtla Gutiérrez. Artículos recientes: *Excessive cell phone use and its implications for academic performance in engineering students* y *About the validation of a questionnaire on behavioral addiction to the internet and its relationship with the student population*.
ORCID: https://orcid.org/0000-0002-3796-8504

Pérez Cortés, Antonio

Doctorante en Ciencias Básicas de la Universidad Autónoma de Zacatecas, con especialización en Física-Matemática. Maestro en Matemáticas por la Universidad Autónoma de Zacatecas. Licenciado en Física por la Universidad Autónoma de Zacatecas. Desempeña su labor docente en el Instituto Tecnológico de Zacatecas (ITZ-TECNM), desde el semestre de agosto-diciembre 2024, y en el Centro de Actualización del Magisterio en Zacatecas (CAMZAC). Profesor-investigador con enfoque en la física-matemática, en particular, en la mecánica cuántica pseudohermítica y teoría cuántica de campos; a su vez, se dedica a estudiar geometría algebraica, con especialización en la teoría de Brill-Noether, la cual estudia haces lineales sobre curvas de género g. Ha impartido pláticas y seminarios en las unidades académicas de Física y de Matemáticas (UAM, UAF) de la Universidad Autónoma de Zacatecas; en el posgrado de matemáticas del Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) de la Universidad de Guadalajara (UDG). A su vez, impartió una conferencia titulada *Matemáticas en los huapangos*, en el Festival del Huapango Inspiración Huasteca, acompañado de la M.M. Kenia Zárate Badillo y la Dra. Graciela Astrid Reyes Ahumada. Ha impartido cursos que abarcan física general, cálculo diferencial, cálculo integral, probabilidad, estadística, álgebra, trigonometría, estadística inferencial, entre otras.

Reyes Rodríguez, José Antonio

Licenciado en Informática, estudió en el Instituto Tecnológico de Zacatecas y es docente de la Academia de Sistemas y Computación en la misma institución desde hace 22 años. Realiza actividades de atención a alumnos como: atención a residentes, asesor interno para obtención de grado de Licenciatura en Informática e Ingeniería en Sistemas Computacionales, es tutor de los grupos 4 A Ing. Informática y 4 C de ISC.

Salome de la Rosa Olvera, María

Maestra en Administración y Licenciada en Ingeniería Química. Jefa del laboratorio de Química en el Instituto Tecnológico de Zacatecas.

Sánchez Iturbe, Patricia Guadalupe

Doctora en Ciencias y Biotecnología de Plantas. Profesora de licenciatura y maestría en el ITTG, UNACH. Autora de diversas ponencias y artículos, ha fungido como asesora y revisora de diversas tesis de licenciatura, maestría y doctorado. Ha participado en actividades de divulgación científica, así como en la actualización de planes y programas de estudio, también ha impartidos cursos a diversas instituciones nacionales. Realiza investigaciones en la línea de educación y biotecnología. Participa en REEDIEM (Red de Investigadores Educativos en México) nodo Tuxtla Gutiérrez. Artículos recientes: *Excessive cell phone use and its implications for academic performance in engineering students* (2025) y *About the validation of a questionnaire on behavioral addiction to the internet and its relationship with the student population* (2024).

ORCID: https://orcid.org/0000-0001-9245-3725

Saucedo Becerra, Edgar Esaúl

Maestro en Tecnología Educativa por la Universidad Interamericana del Desarrollo (México). Obtuvo la Licenciatura de Ingeniero en Sistemas Computacionales. Se ha desempeñado como académico profesional en la Unidad Académica de Matemáticas de la UAZ como responsable de servicios de cómputo; coordinador de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Zacatecas. En la actualidad es profesor-investigador en la Universidad Autónoma de Zacatecas donde realiza investigaciones sobre la integración de la tecnología en el ambiente educativo. También es miembro de la Internet Society. Recientemente publicó el artículo *Pensamiento matemático evidenciado a través de la modelización matemática mediada con tecnología* (2025).

ORCID: https://orcid.org/0000-0002-3114-2197
ResearchGate: https://www.researchgate.net/profile/Edgar-Saucedo-2

Trejo Calzada, Ma. Lourdes

Maestra en Ciencias en Enseñanza de las Ciencias MCEC, por el Centro Interdisciplinario de Investigación y Docencia Técnica CIIIDET-Querétaro, perteneciente al Tecnológico Nacional de México (TecNM). Tiene formación de Licenciatura de Química Farmacéutica Bióloga por la Facultad de Ciencias Químicas en la Universidad Autónoma de Zacatecas. En su trayectoria profesional docente de veinticuatro años en el Instituto Tecnológico de Zacatecas ha desempeñado roles como: jefa de Laboratorio de Química, jefa del Departamento de Ciencias Básicas, secretaria de academia, auditor de sistemas de gestión de calidad, tutora grupal e individual. Asesora académica en eventos locales, regionales y nacionales. Actualmente se desempeña como docente de tiempo completo adscrita al Departamento de Ciencias básicas. Una de sus publicaciones más recientes es el artículo "Impacto del aprendizaje de química a través de las prácticas en las ingenierías del ITZ", publicado en coautoría en la revista *ExpreSciencia* (2024).

ORCID: https://orcid.org/0009-0007-4761-069X

Vega Rodríguez, Rosario

Maestra en Ciencias en Enseñanza de Ciencias Básicas con especialidad en Física por el CIIDET. Especialidad en tecnología educativa por la Universidad Autónoma de zacatecas, y Licenciatura en Ingeniería Eléctrica también por la universidad Autónoma de Zacatecas.

Matemáticas discretas: comprensión fácil y paso a paso, de
Elsa Ortega de Ávila (coordinadora), publicado por
Ediciones Comunicación Científica, S. A. de C. V.,
se terminó de imprimir en junio de 2026 en Litográfica
Ingramex, S. A. de C. V., Centeno 162-1, Granjas Esmeralda,
09810, Ciudad de México. El tiraje fue de 500 ejemplares
impresos y en versión digital para acceso abierto en los formatos
PDF, Epub y HTML. El cuidado de la edición estuvo a cargo de
Fidel Carlón Solís.

Este libro surge como una respuesta pedagógica a las dificultades recurrentes que enfrentan los estudiantes de Ingeniería en Sistemas Computacionales en el aprendizaje de las matemáticas discretas. Lejos de constituirse como un tratado exhaustivo o una obra de investigación científica el texto se orienta como un recurso didáctico accesible, claro y funcional, alineado con las necesidades y características de los estudiantes contemporáneos.

A diferencia de los enfoques tradicionales, frecuentemente densos y altamente teóricos, esta propuesta privilegia la comprensión progresiva mediante un lenguaje sencillo, ejemplos contextualizados en el ámbito de la ingeniería y una estructura organizada que facilita el aprendizaje autónomo. El contenido equilibra adecuadamente la teoría esencial con ejercicios resueltos y actividades prácticas, además, promueve que se apliquen estos conceptos en situaciones reales relacionadas con la algoritmia, estructuras de datos y desarrollo de software.

Asimismo, el libro *Matemáticas discretas: comprensión fácil y paso a paso* se encuentra estrictamente vinculado con el programa académico del Tecnológico Nacional de México, lo que garantiza su pertinencia y utilidad como herramienta de apoyo en el proceso formativo. Su enfoque no busca abarcar la totalidad del campo disciplinar, sino centrarse en los contenidos fundamentales que fortalecen el pensamiento lógico, analítico y crítico del estudiante.

En suma, esta obra representa una alternativa innovadora que favorece la motivación, mejora la comprensión y contribuye al desarrollo de competencias clave en la formación de futuros ingenieros.

Elsa Ortega de Ávila es doctora y cuenta con una trayectoria amplia en educación superior y básica con enfoque interdisciplinario. Ha ejercido la docencia, dirigido tesis y participado en investigación. Su productividad incluye 13 artículos, dos libros y dos capítulos publicados; en ellos integra investigación, innovación tecnológica y desarrollo académico con impacto social.

RENIECYT
Registro Nacional de Instituciones y Empresas Científicas y Tecnológicas
2000922

DOI.ORG/10.52501/CC.440

ISBN: 978-968-9738-82-4

EDICIONES COMUNICACIÓN CIENTÍFICA PUBLICACIONES ARBITRADAS
comunicacion-cientifica.com