

Sistemas numéricos

**Breve descripción:**

Los sistemas numéricos representan valores en diversas bases, como decimal, binario, octal y hexadecimal, cada uno con aplicaciones específicas. El álgebra de Boole, clave en sistemas digitales, utiliza operaciones lógicas (suma, multiplicación y negación) para simplificar circuitos electrónicos. Propiedades como identidad, inversa y el teorema de Morgan optimizan procesos y facilitan el análisis de sistemas electrónicos y computacionales.

**Diciembre 2024**

Tabla de contenido

[Introducción 1](#_Toc185345960)

[1. Sistemas numéricos 2](#_Toc185345961)

[2. Sistema binario 6](#_Toc185345962)

[Síntesis 12](#_Toc185345963)

[Material complementario 13](#_Toc185345964)

[Glosario 14](#_Toc185345965)

[Referencias bibliográficas 15](#_Toc185345966)

[Créditos 16](#_Toc185345967)

Introducción

Los sistemas numéricos son métodos que permiten representar valores numéricos de diversas formas, siendo el sistema decimal el más utilizado en la vida cotidiana debido a su base de 10 dígitos (0-9). Otros sistemas, como el binario, octal, hexadecimal y romano, son ampliamente empleados en diferentes contextos, desde sistemas electrónicos hasta representaciones históricas. Comprender estos sistemas facilita la conversión de números entre bases y su aplicación en diversos campos.

El álgebra de Boole es fundamental en el ámbito digital, ya que utiliza números binarios (0 y 1) para modelar operaciones lógicas. Este sistema algebraico permite trabajar con circuitos integrados, predecir el comportamiento de sistemas electrónicos y simplificar expresiones lógicas. A través de operaciones básicas como suma lógica (+), multiplicación lógica (·) y negación (¬), el álgebra de Boole optimiza el diseño de sistemas digitales y electrónicos.

Las propiedades del álgebra de Boole, como la identidad, inversa, conmutativa y el teorema de Morgan, proporcionan herramientas esenciales para simplificar y resolver expresiones lógicas complejas. Estas propiedades no solo facilitan el análisis y diseño de circuitos, sino que también son clave para optimizar procesos en la programación y el funcionamiento de sistemas electrónicos, consolidando su relevancia en la ingeniería y la informática.

# Sistemas numéricos

Existen algunas formas de representar números, es decir, un mismo número se puede representar de varias formas. Por eso se habla de sistemas numéricos. El más común es el sistema decimal, que se utiliza en la vida cotidiana y que se representa con 10 números del **0 al 9 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)**.

A continuación, se realiza un comparativo de los diferentes sistemas numéricos y en el estudio del material se explicará cómo conocer la equivalencia de un número en los diferentes sistemas.

1. Sistemas numéricos

| Binario | Octal | Decimal | Hexadecimal | Romano |
| --- | --- | --- | --- | --- |
| 0001 | 01 | 1 | 1 | I |
| 0101 | 05 | 5 | 5 | V |
| 1010 | 12 | 10 | A | X |
| 1111 | 17 | 15 | F | XV |

**Sistema decimal**

Se llama decimal porque utiliza diez dígitos (0, 1, 2...9) para representar cualquier número. El peso está dado en potencias de 10. La importancia de los dígitos está acorde a su ubicación: el dígito que esté más hacia la izquierda tiene mayor importancia.

A continuación, se presenta un ejemplo:

2158 = 2000 + 100 + 50 + 8

2158 = 21000 + 1100 + 510 + 8

2158 = 210³ + 110² + 510¹ + 8\*10⁰

Los cuatro dígitos se multiplican por una potencia de 10 y cuanto más a la izquierda se sitúe el dígito, mayor será la potencia de diez por la que se multiplica.

**Sistema octal**

Utiliza solo ocho dígitos (0, 1, 2...7) para representar cualquier número. El peso está dado en potencias de 8.

A continuación, se representa el número 4156 que está octal en decimal:

**Ejemplo:**

4156 = 48³ + 18² + 58¹ + 68⁰

= 2048 + 64 + 40 + 6

**= 2158 (decimal)**

Para convertir de decimal a octal:

• 2158/8 = 269 y sobran 6

• 269/8 = 33 y sobran 5

• 33/8 = 4 y sobra 1

• 4/8 = No está, entonces el residuo es 4

**2158 (decimal) = 4156 (octal)**

**Sistema hexadecimal**

Se llama hexadecimal porque utiliza dieciséis dígitos (0, 1, 2...9, A, B, C, D, E, F).

El peso está dado en potencias de 16.

**Ejemplo:**

2158/16 = 134 y sobran 14 = E

134/16 = 8 y sobran 6

8/16 = No está, entonces el residuo es 8

2158 en decimal **es igual a 86E en hexadecimal**

86C = 8\*162 + 6\*161 + E\*160

= 8\*(256) + 6\*(16) + 14\*(1)

= 2048 + 96 + 14 = 2158 decimal

**Sistema numérico romano**

Este sistema ya tiene establecida una tabla de conversión o equivalencias.

1. Sistema de numeración romano

| Decimal | Romano |
| --- | --- |
| 1 | I |
| 10 | X |
| 50 | L |
| 100 | C |
| 500 | D |
| 1000 | M |

**Ejemplo:**

2158 = 2000 + 100 + 50 + 8

= MM C L VII

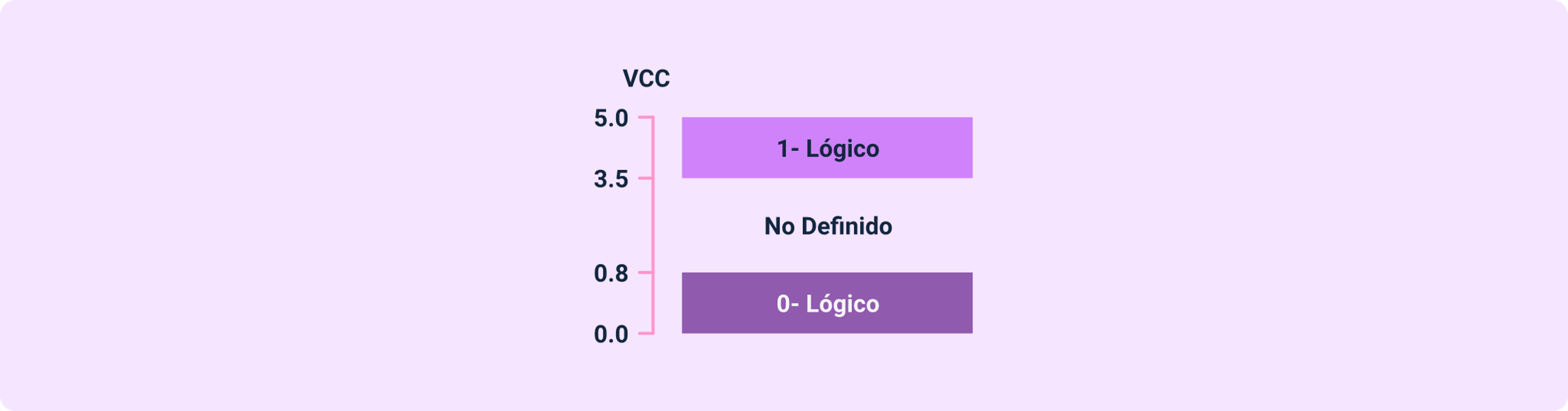
= MMCLVIII (romano)

2158 en decimal es igual a MMCLVII en Romano

# Sistema binario

El sistema binario utiliza solo dos dígitos (0 y 1) para representar cualquier número. Es el sistema que se utiliza como lenguaje para los sistemas electrónicos, gracias a los dos estados o niveles lógicos que se encuentran en los circuitos integrados: abierto o cerrado (1 o 0 lógico).

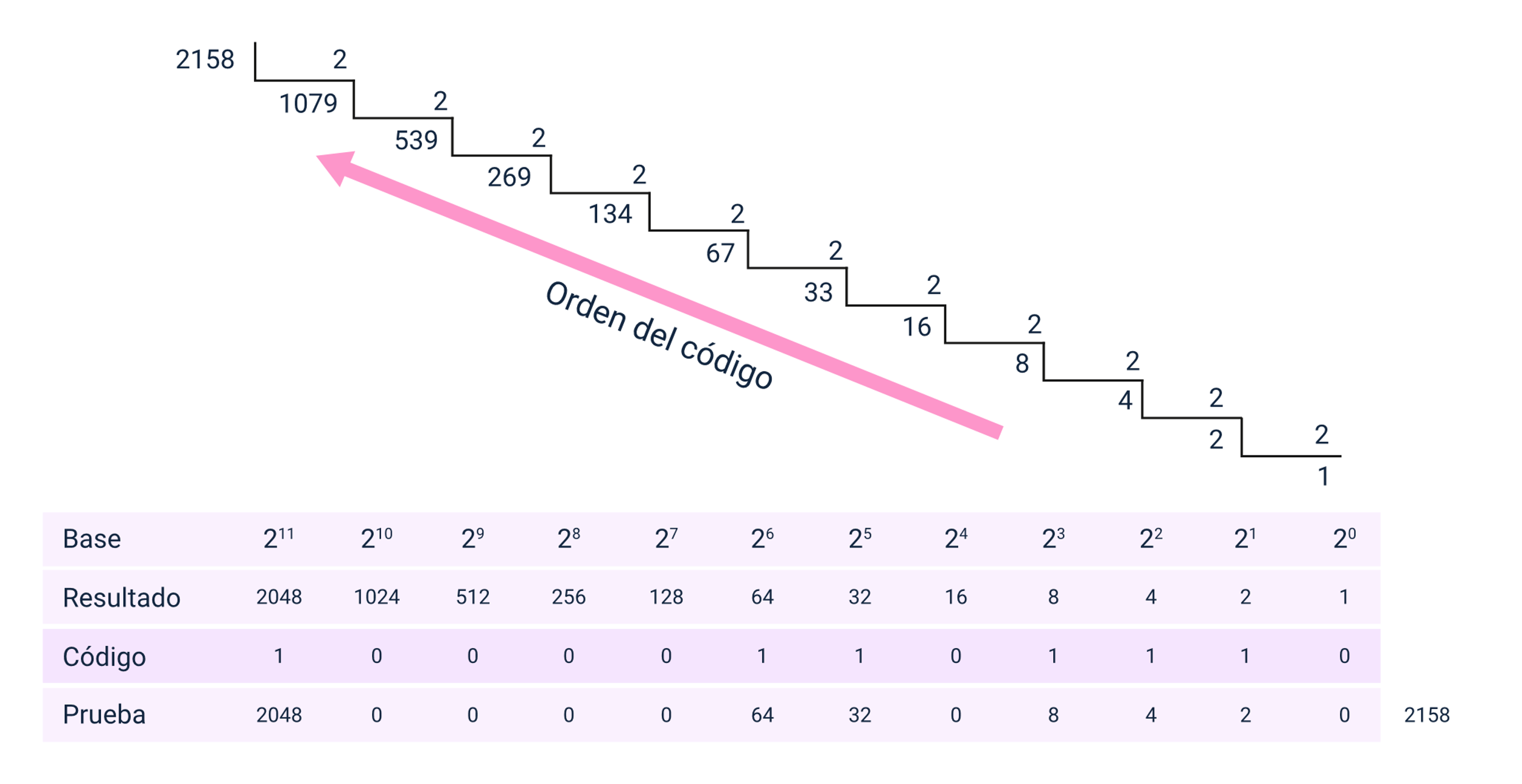
1. Niveles de voltaje en lógica digital



En la electrónica tradicional se usan 5 voltios para el dígito '1' y 0 voltios para el dígito '0'.

**Ejemplo**. Representar 2158 (decimal) en binario

1. Conversión de un número decimal a binario



El procedimiento para convertir un número decimal a binario utilizando divisiones sucesivas por 2 es el siguiente:

1. Se divide el número decimal entre 2, obteniendo el cociente y el resto. El resto será uno de los dígitos del número binario.
2. Si el resto es 0, se registra un 0 como dígito binario; si el resto es 1, se registra un 1.
3. El cociente de la división se toma como el nuevo número decimal.
4. Se repite el proceso dividiendo nuevamente por 2, registrando los restos como los dígitos binarios.
5. El procedimiento finaliza cuando el cociente alcanza el valor de 0. Los restos obtenidos se leen de abajo hacia arriba para formar el número binario.
6. **Divida el número decimal por 2.**
7. **Si hay resto, escriba 1; si no, 0.**
8. **Continúe hasta que no quede número decimal.**

**Álgebra de Boole**

El álgebra de Boole se utiliza para programar circuitos integrados a través de números binarios y realizar operaciones binarias. Esto permite predecir el comportamiento en la entrada y salida del circuito.

Incluye tres operaciones fundamentales:

1. Operaciones lógicas básicas

| Operación + | Operación ⋅ o ∗ | Operación de negación |
| --- | --- | --- |
| (Es el mismo símbolo de la suma que usualmente usamos, pero no es la misma operación) | (Es el mismo símbolo del producto o multiplicación que usualmente usamos, pero no es la misma operación) | (Permite invertir o negar el estado lógico, bit o dígito así) |
| Entradas | Salida | Entradas | Salida | Entradas | Salida |
| A + B = C  0 + 0 = 0  0 + 1 = 1  1 + 0 = 1  1 + 1 = 1 | A · B = C  0 · 0 = 0  0 · 1 = 0  1 · 0 = 0  1 · 1 = 1 | A = A  0 = 1  1 = 0 |

**Dato importante**

La negación se expresa con una línea arriba del nombre asignado a la entrada o a la salida o con una comilla, así: A = A'

**Función lógica**

Es la expresión algebraica formada por variables binarias que se relacionan mediante las operaciones básicas del álgebra de Boole.

**Ejemplo:**

F = a + b (la función sería “a + b” que equivale a la operación booleana “+” o de suma).

Donde a y b son las variables binarias.

**Tabla de verdad**

Es una representación gráfica de todos los valores que puede tomar la función lógica para cada una de las posibles combinaciones de las variables de entrada.

El número de combinaciones posibles es 2^n, siendo n el número de variables.

**Ejemplo**: la tabla para la función lógica F = (A + B) · C sería:

n = 3 (porque las variables son A, B y C).

23 = 8 (número de posibles combinaciones y filas que debe tener la tabla).

El álgebra de Boole se basa en un conjunto de propiedades y reglas fundamentales que permiten simplificar y analizar expresiones lógicas. Estas reglas son esenciales para optimizar circuitos y resolver problemas lógicos de manera estructurada. A continuación, se presentan las principales propiedades y ejemplos asociados:

1. Las 8 posibles combinaciones

| Variables o entrada | Función lógica o salida |
| --- | --- |
| A B C | F = (A + B) · C |
| 0 0 0 | 0 |
| 0 0 1 | 0 |
| 0 1 0 | 0 |
| 0 1 1 | 1 |
| 1 0 0 | 0 |
| 1 0 1 | 1 |
| 1 1 0 | 0 |
| 1 1 1 | 1 |

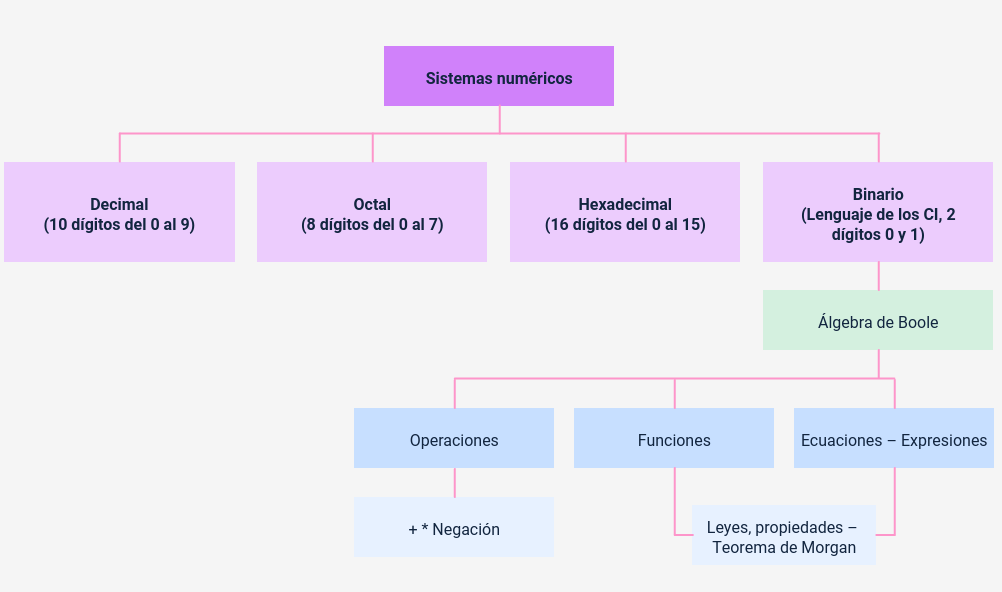
El álgebra de Boole se basa en un conjunto de propiedades y reglas fundamentales que permiten simplificar y analizar expresiones lógicas. Estas reglas son esenciales para optimizar circuitos y resolver problemas lógicos de manera estructurada. A continuación, se presentan las principales propiedades y ejemplos asociados.

1. Propiedades o reglas del álgebra de Boole

| Propiedad o Regla | Ejemplo |
| --- | --- |
| Simplificación de negación | A = A |
| Simplificación de igualdad | A + A = A  A · A = A |
| Inversa | A + A = 1  A · A = 0 |
| Identidad | A + 0 = A  A · 1 = A  A + 1 = 1  A · 0 = 0 |
| Conmutativa | A + B = B + A  A · B = B · A |
| Asociativa | A + (B + C) = (A + B) + C  A · (B · C) = (A · B) · C |
| Distributiva | A · (B + C) = (A · B) + (A · C)  (A + B) · C = (A · C) + (B · C) |
| Más reglas de simplificación o leyes de absorción | A + A · B = A  A · (A + B) = A |
| Teorema de Morgan | (A · B) = A´ + B´  (A · B) = A´ · B´ |

Síntesis

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.



Material complementario

| Tema | Referencia | Tipo de material | Enlace del recurso |
| --- | --- | --- | --- |
| Sistemas numéricos | Electrónica FP. (2018). Sistemas numéricos: Binario, Decimal y Hexadecimal. | Video | <https://www.youtube.com/watch?v=g9-MRBBcvdg> |
| Sistema binario | Matemáticas profe Alex. (2023). Números Binarios | Introducción | Video | <https://www.youtube.com/watch?v=F1xB6oxmMWQ&t=52s> |
| Sistema binario | EDteam. (2023). ¿Qué es el sistema binario y como funciona? | Video | <https://www.youtube.com/watch?v=-V4AbHyR_x4> |

Glosario

**Álgebra de Boole**: rama del álgebra que trabaja con operaciones lógicas sobre valores binarios (0 y 1).

**Circuitos digitales**: sistemas electrónicos que procesan información mediante niveles lógicos (0 y 1).

**Multiplicación lógica**: operación booleana que devuelve 1 solo si todas las entradas son 1.

**Negación**: operación booleana que invierte el estado lógico, cambiando 0 a 1 y viceversa.

**Sistema binario**: sistema numérico de base 2 que utiliza los dígitos 0 y 1, empleado en sistemas electrónicos.

**Sistema decimal**: sistema numérico de base 10 que utiliza los dígitos del 0 al 9.

**Sistemas numéricos**: métodos de representación de números en diferentes bases como decimal, binario, octal y hexadecimal.

**Suma lógica (+)**: operación booleana que devuelve 1 si al menos una entrada es 1.

**Tabla de verdad**: representación de todas las posibles combinaciones de entrada y salida de una función lógica.

**Teorema de Morgan**: regla del álgebra de Boole que simplifica la negación de expresiones lógicas compuestas.

Referencias bibliográficas

Carlospes, (s.f), Sistema de numeración romano. (s.f.). En Sitio web de Carlos Pes. <https://www.carlospes.com/curso_representacion_datos/02_01_sistema_de_numeracion_romano.php>

Cidead, (s.f), material interactivo sobre Lógica Binaria.

Junta de Extremadura consejería de educación y empleo, (s.f), Laboratorio virtual de Lógica Binaria.

Lloris, A. (2006). Capítulo 1: Introducción a los Sistemas digitales. En Sistemas Digitales 2 Ed. McGraw Hill.

Créditos

| Nombre | Cargo | Centro de Formación y Regional |
| --- | --- | --- |
| Milady Tatiana Villamil Castellanos | Responsable del ecosistema | Dirección General |
| Olga Constanza Bermúdez Jaimes | Responsable de línea de producción | Centro de Servicios de Salud - Regional Antioquia |
| Magda Melissa Rodríguez Celis | Experto temático | Centro de Desarrollo Agroempresarial - Regional Cundinamarca |
| Paola Alexandra Moya Peralta | Evaluadora instruccional | Centro de Servicios de Salud - Regional Antioquia |
| Juan Daniel Polanco Muñoz | Diseñador de contenidos digitales | Centro de Servicios de Salud - Regional Antioquia |
| Edgar Mauricio Cortés García | Desarrollador full stack | Centro de Servicios de Salud - Regional Antioquia |
| Jaime Hernán Tejada Llano | Validador de recursos educativos digitales | Centro de Servicios de Salud - Regional Antioquia |
| Margarita Marcela Medrano Gómez | Evaluador para contenidos inclusivos y accesibles | Centro de Servicios de Salud - Regional Antioquia |
| Daniel Ricardo Mutis Gómez | Evaluador para contenidos inclusivos y accesibles | Centro de Servicios de Salud - Regional Antioquia |