SISTEMAS DE NUMERACIÓN

Un sistema de numeración es un conjunto de símbolos y reglas que permiten representar datos numéricos. La norma principal en un sistema de numeración posicional es que ***un mismo símbolo tiene distinto valor según la posición que ocupe***.

Sistema de numeración decimal:

El sistema de numeración que utilizamos habitualmente es el **decimal,** que se compone de diez símbolos o dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9) a los que otorga un valor **dependiendo de la posición** que ocupen en la cifra: unidades, decenas, centenas, millares, etc.

El valor de cada dígito está asociado al de una potencia de base 10, número que coincide con la cantidad de símbolos o dígitos del sistema decimal, y un exponente igual a la posición que ocupa el dígito menos uno, contando desde la derecha.

En este sistema el número 528, por ejemplo, significa:

5 centenas + 2 decenas + 8 unidades, es decir:

500 + 20 + 8 o, lo que es lo mismo,

$$5\cdot10^2+2\cdot10^1+8\cdot10^0=528$$

En el caso de números con decimales, la situación es análoga aunque, en este caso, algunos exponentes de las potencias serán negativos, concretamente el de los dígitos colocados a la derecha del separador decimal. Por ejemplo, el número 8245,97 se calcularía como:

8 millares + 2 centenas + 4 decenas + 5 unidades + 9 décimos + 7 céntimos

8000 + 200 + 40 + 5 + 0,9 + 0,07 = 8245,97

$$8\cdot10^3+2\cdot10^2+4\cdot10^1+5\cdot10^0+9\cdot10^{-1}+7\cdot10^{-2}=8245{,}97$$

Sistema de numeración binario.

El sistema de numeración binario utiliza sólo dos dígitos, el cero (0) y el uno (1), que tienen distinto valor dependiendo de la posición que ocupen. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición del dígito menos uno. Se puede observar que, tal y como ocurría con el sistema decimal, la base de la potencia coincide con la cantidad de dígitos utilizados (2) para representar los números.

De acuerdo con estas reglas, el número binario 1011 tiene un valor que se calcula así:

$$1\cdot2^3+0\cdot2^2+1\cdot2^1+1\cdot2^0=8+0+2+1=11$$

y lo escribimos así: $1011_2=11_{10}$

Conversión entre números decimales y binarios

Convertir un número decimal al sistema binario es muy sencillo: basta con realizar divisiones sucesivas por 2 y colocar los restos obtenidos, en cada una de ellas. Para formar el número binario tomaremos los restos **en orden inverso** al que han sido obtenidos. Por ejemplo:

77 : 2 = 38 Resto: 1
38 : 2 = 19 Resto: 0
19 : 2 = 9 Resto: 1
9 : 2 = 4 Resto: 1
4 : 2 = 2 Resto: 0
2 : 2 = 1 Resto: 0
1 : 2 = 0 Resto: 1

$77_{10} = 1\ 0\ 0\ 1\ 1\ 0\ 1_2$

La cantidad de dígitos necesarios, para representar un número en el sistema binario, dependerá del valor de dicho número en el sistema decimal. En el caso anterior, para representar el número 77 han hecho falta siete dígitos. Para representar números superiores harán falta más dígitos. Por ejemplo, para representar números mayores de 255 se necesitarán más de ocho dígitos, porque $2^8=256$ y, por tanto, 255 es el número más grande que puede representarse con ocho dígitos.

Es importante distinguir entre los números que pueden representarse con n dígitos binarios, que es 2^n, y el mayor de esos números, que es una unidad menos, es decir, $2^n - 1$.

El proceso para convertir un número del sistema binario al decimal es aún más sencillo; basta con desarrollar el número, teniendo en cuenta que el valor de cada dígito está asociado a una potencia de 2, cuyo exponente es 0 en el bit situado más a la derecha, y se incrementa en una unidad según vamos avanzando posiciones hacia la izquierda, tal y como se muestra en el siguiente ejemplo:

$$1010011 = 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 83$$

$$1010011_2 = 83_{10}$$

SISTEMAS DE NUMERACIÓN OCTAL Y HEXADECIMAL

El inconveniente de la codificación binaria es que la representación de algunos números resulta muy larga. Por este motivo se utilizan otros sistemas de numeración que resulten más cómodos de escribir: el sistema octal y el sistema hexadecimal. Afortunadamente, resulta muy fácil convertir un número binario a octal o a hexadecimal.

Sistema de numeración octal

En el sistema octal, los números se representan mediante **ocho** dígitos diferentes: 0, 1, 2, 3, 4, 5, 6 y 7. Cada dígito tiene, naturalmente, un valor distinto dependiendo del lugar que ocupen. El valor de cada una de las posiciones viene determinado por las potencias de base 8. La conversión de un número decimal a octal, y viceversa, se realiza del mismo modo que la de los números binarios, aunque, lógicamente, se emplea como base el número 8 en vez del 2.

La conversión de un número decimal a octal se hace del mismo modo: mediante divisiones sucesivas por 8 y colocando los restos obtenidos en orden inverso. Por ejemplo:

122 : 8 = 15 Resto: 2
15 : 8 = 1 Resto: 7
1 : 8 = 0 Resto: 1

$122_{10} = 172_8$

La conversión de un número octal a decimal es igualmente sencilla. Por ejemplo:

$$237_8 = 2\cdot 8^2 + 3\cdot 8^1 + 7\cdot 8^0 = 128 + 24 + 7 = 159_{10}$$

$$237_8 = 159_{10}$$

SISTEMA DE NUMERACIÓN HEXADECIMAL

En este sistema, los números se representan con dieciséis símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. Se utilizan los caracteres A, B, C, D, E y F representando las cantidades decimales 10, 11, 12, 13, 14 y 15 respectivamente, porque no hay dígitos mayores que 9 en el sistema decimal. El valor de cada uno de estos símbolos depende, como es lógico, de su posición, que se calcula mediante potencias de base 16.

Ensayemos la conversión decimal a hexadecimal del número 1735:

1735 : 16 = 108	Resto: 7	
108 : 16 = 6	Resto: C (12_{10})	$1735_{10} = 6C7_{16}$
6 : 16 = 0	Resto: 6	

Ensayemos también la conversión inversa, de hexadecimal a decimal del número 1A3F:

$$1A3F_{16} = 1\cdot 16^3 + A\cdot 16^2 + 3\cdot 16^1 + F\cdot 16^0 = 6719_{10}$$

$$1A3F_{16} = 6719_{10}$$

Conversión de números binarios a octales y hexadecimales

Cada dígito de un número octal equivale a tres dígitos en el sistema binario. Por tanto, el modo de convertir un número entre estos sistemas de numeración equivale a "expandir" cada dígito octal a tres dígitos binarios, o en "contraer" grupos de tres caracteres binarios a su correspondiente dígito octal. Por ejemplo:

$$101001011_2 = 513_8$$

$$750_8 = 111101000_2$$

Análogamente, la conversión entre números hexadecimales y binarios se realiza "expandiendo" o "contrayendo" cada dígito hexadecimal a cuatro dígitos binarios. Por ejemplo:

$$101001110011_2 = A73_{16}$$

$$1F6_{16} = 000111110110_2$$

En caso de que los dígitos binarios no formen grupos completos (de tres o cuatro dígitos, según corresponda), se deben añadir ceros a la izquierda hasta completar el último grupo. Por ejemplo:

$$101110_2 = 00101110_2 = 2E_{16}$$

ARITMÉTICA BINARIA

La Unidad Aritmético Lógica, en la CPU del procesador, es capaz de realizar operaciones aritméticas, con datos numéricos expresados en el sistema binario. Naturalmente, esas operaciones incluyen la adición, la sustracción, el producto y la división. Las operaciones se hacen del mismo modo que en el sistema decimal, pero debido a la sencillez del sistema de numeración, pueden hacerse algunas simplificaciones que facilitan mucho la realización de las operaciones.

SUMA EN BINARIO

La tabla de sumar, en binario, es mucho más sencilla que en decimal. Sólo hay que recordar cuatro combinaciones posibles. Recuerda que en el sistema decimal había que memorizar unas 100 combinaciones.

SUMA	0	1
0	0	1
1	1	0 + a

Las sumas 0+0, 0+1 y 1+0 son evidentes:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

Pero la suma de 1+1, que sabemos que es 2, debe escribirse en binario con dos cifras (10) y, por tanto 1+1 es 0 y se arrastra una unidad, que se suma a la posición siguiente a la izquierda.

Veamos algunos ejemplos:

$$\begin{array}{r} 010 \\ \underline{101} \\ 111 \end{array} \qquad \begin{array}{r} 2_{10} \\ \underline{5_{10}} \\ 7_{10} \end{array} \qquad\qquad \begin{array}{r} 001101 \\ \underline{100101} \\ 110010 \end{array} \qquad \begin{array}{r} 13_{10} \\ \underline{37_{10}} \\ 50_{10} \end{array}$$

$$\begin{array}{r} 1011011 \\ \underline{1011010} \\ 10110101 \end{array} \qquad \begin{array}{r} 91_{10} \\ \underline{90_{10}} \\ 181_{10} \end{array} \qquad\qquad \begin{array}{r} 110111011 \\ \underline{100111011} \\ 1011110110 \end{array} \qquad \begin{array}{r} 443_{10} \\ \underline{315_{10}} \\ 758_{10} \end{array}$$

SUSTRACCIÓN EN BINARIO

RESTA	0	1
0	0	1
1	1 + a	0

Restar en binario es, nuevamente, igual que la misma operación en el sistema decimal. Pero conviene repasar la operación de restar en decimal para comprender la operación binaria, que es más sencilla. Los términos que intervienen en la resta se llaman minuendo, sustraendo y diferencia.

Las sumas 0-0, 1-0 y 1-1 son evidentes:

0 – 0 = 0

1 – 0 = 1

1 – 1 = 0

La resta 0 - 1 se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente: 10 -1, es decir, $2_{10} - 1_{10} = 1$

Esa unidad prestada debe devolverse, sumándola, a la posición siguiente. Veamos algunos ejemplos:

$$\begin{array}{r} 111 \\ \underline{101} \\ 010 \end{array} \qquad \begin{array}{r} 7_{10} \\ \underline{5_{10}} \\ 2_{10} \end{array} \qquad\qquad \begin{array}{r} 10001 \\ \underline{01010} \\ 00111 \end{array} \qquad \begin{array}{r} 17_{10} \\ \underline{10_{10}} \\ 7_{10} \end{array}$$

$$\begin{array}{r} 11011001 \\ \underline{10101011} \\ 00101110 \end{array} \qquad \begin{array}{r} 217_{10} \\ \underline{171_{10}} \\ 46_{10} \end{array} \qquad\qquad \begin{array}{r} 111101001 \\ \underline{101101101} \\ 001111100 \end{array} \qquad \begin{array}{r} 489_{10} \\ \underline{365_{10}} \\ 124_{10} \end{array}$$

A pesar de lo sencillo que es el procedimiento de restar, es facil confundirse. Tenemos interiorizado el sistema decimal y hemos aprendido a restar mecánicamente, sin detenernos a pensar en el significado del arrastre. Para simplificar las restas y reducir la posibilidad de cometer errores hay varias soluciones:

- Dividir los números largos en grupos. En el siguiente ejemplo, vemos cómo se divide una resta larga en tres restas cortas:

$$\begin{array}{r} 100110011101 \\ \underline{010101110010} \\ 010000101011 \end{array} = \begin{array}{r} 1001 \\ \underline{0101} \\ 0100 \end{array} \quad \begin{array}{r} 1001 \\ \underline{0111} \\ 0010 \end{array} \quad \begin{array}{r} 1101 \\ \underline{0010} \\ 1011 \end{array}$$

- Utilizando el **Complemento a dos**

Complemento a dos

El ***complemento a dos*** de un número ***N***, con ***n*** cifras, se define como $C_2^N = 2^n - N$.

Veamos un ejemplo: tomemos el número $N = 101101_2$ que tiene 6 cifras, y calculemos el complemento a dos de ese número:

$N = 45_{10}$ $n = 6$ $2^6 = 64$ y, por tanto: $C_2^N = 64 - 45 = 19 = 010011_2$

Complemento a uno

El ***complemento a uno*** de un número ***N***, con ***n*** cifras es, por definición, una unidad menor que el complemento a dos, es decir:

$$C_1^N = C_2^N - 1 \quad \text{y, por la misma razón,} \quad C_2^N = C_1^N + 1$$

Calculemos el ***complemento a uno*** del mismo número del ejemplo anterior:

$$C_1^N = C_2^N - 1 \qquad \begin{array}{r} 010011 \\ \underline{000001} \\ 010010 \end{array} \qquad C_1^N = 010010$$

Da la sensación de que no va a ser más sencillo restar utilizando el complemento a dos, porque el procedimiento para calcular el complemento a dos es más difícil y laborioso que la propia resta. Pero es mucho más sencillo de lo que parece.

En realidad, el ***complemento a uno*** de un número binario es el número resultante de invertir UNOS y CEROS.

Si $N = 101101$

su ***complemento a uno*** es: $C_1^N = 010010$

y su ***complemento a dos*** es: $C_2^N = C_1^N + 1 = 010011$

Veamos otro ejemplo de cálculo de complementos:

Si $N = 0110110101$

El complemento a uno es: $C_1^N = 1001001010$ $C_1^N = 1001001010$

y el complemento a dos es: $C_2^N = 1001001011$

Restar en binario usando el complemento a dos

Y, por fin, vamos a ver cómo facilita la resta el complemento. La resta binaria de dos números puede obtenerse ***sumando al minuendo el complemento a dos del sustraendo***. Veamos algunos ejemplos:

a) Hagamos la siguiente resta, 91 – 46 = 45, en binario:

$$\begin{array}{r} 1011011 \\ \underline{0101110} \\ 0101101 \end{array} \qquad \begin{array}{r} 91_{10} \\ \underline{46_{10}} \\ 45_{10} \end{array}$$

Tiene alguna dificultad, cuando se acumulan los arrastres a la resta siguiente. Pero esta misma resta puede hacerse como una suma, utilizando el complemento a dos del sustraendo:

$$\begin{array}{r} 1011011 \\ \underline{1010010} \\ 10101101 \end{array}$$

En el resultado nos sobra un bit, que se desborda por la izquierda. Como el número resultante no puede ser más largo que el minuendo, el bit sobrante se desprecia.

b) Hagamos esta otra resta, 219 – 23 = 196, utilizando el complemento a dos:

$219_{10} = 11011011_2$
$23_{10} = 00010111_2$

$C_2^{23} = 11101001$

$$\begin{array}{r} 11011011 \\ \underline{11101001} \\ 111000100 \end{array}$$

Y, despreciando el bit que se desborda por la izquierda, llegamos al resultado correcto:

$11000100_2 = 196_{10}$

¡Qué fácil!

MULTIPLICACIÓN BINARIA

POR	0	1
0	0	0
1	0	1

La multiplicación en binario es más fácil que en cualquier otro sistema de numeración.

Como los factores de la multiplicación sólo pueden ser CEROS o UNOS, el producto sólo puede ser CERO o UNO. En otras palabras, la tabla de multiplicar es muy fácil de aprender

En un ordenador, sin embargo, la operación de multiplicar se realiza mediante sumas repetidas. Eso crea algunos problemas en la programación porque cada suma de dos UNOS origina un arrastre, que se resuelven contando el número de UNOS y de arrastres en cada columna. Si el número de UNOS es par, la suma es un CERO y si es impar, un UNO. Luego, para determinar los arrastres a la posición superior, se cuentan las parejas de UNOS.

DIVISIÓN BINARIA

Igual que en el producto, la división es muy fácil de realizar, porque no son posibles en el cociente otras cifras que UNOS y CEROS.

Consideremos el siguiente ejemplo, 42 : 6 = 7, en binario:

```
(Dividendo)    1 0 1 0 1 0  | 110  (Divisor)
               - 1 1 0      | 111  (Cociente)
                 1 0 0 1
                 - 1 1 0
                   0 1 1 0
                     1 1 0
                     0 0 0
```

Se intenta dividir el dividendo por el divisor, empezando por tomar en ambos el mismo número de cifras (100 entre 110, en el ejemplo). Si no puede dividirse, se intenta la división tomando un dígito más (1001 entre 100).

Si la división es posible, entonces, el divisor sólo podrá estar contenido **una vez** en el dividendo, es decir, la primera cifra del cociente es un UNO. En ese caso, el resultado de multiplicar el divisor por 1 es el propio divisor. Restamos las cifras del dividendo del divisor y bajamos la cifra siguiente.

El procedimiento de división continúa del mismo modo que en el sistema decimal.

EJERCICIOS

1. Expresa, en código binario, los números decimales siguientes:
   c) 47
   d) 191
   e) 25
   f) 67
   g) 99
   h) 135
   i) 276.

2. Expresa, en el sistema decimal, los siguientes números binarios:
   a) 110111
   b) 111000
   c) 010101
   d) 101010
   e) 1111110

3. Dados dos números binarios: 01001000 y 01000100 ¿Cuál de ellos es el mayor? ¿Podrías compararlos sin necesidad de convertirlos al sistema decimal?

4. ¿Cuántos números diferentes se pueden escribir, utilizando el sistema binario de numeración, con sólo 3 dígitos? ¿Y con 16 dígitos?

5. Convierte los siguientes números octales en decimales:
   a) 45_8
   b) 125_8
   c) 625_8

6. Convierte los siguientes números decimales en octales:
   a) 63
   b) 513
   c) 119

7. Convierte los siguientes números binarios en octales:
   a) 1101101
   b) 101110
   c) 11011011
   d) 101101011

8. Convierte los siguientes números octales en binarios:
   a) 25_8
   b) 372_8
   c) 2753_8

9. Realiza las siguientes sumas de números binarios:
   a) 111011 + 110
   b) 111110111 + 111001
   c) 10111 + 11011 + 10111

10. Realiza las siguientes sumas de números octales:
    a) 365 + 23
    b) 2732 + 1265
    c) 65 + 1773

11. Suma los siguientes números hexadecimales:
    a) 17A + 3C
    b) 20F5 + 31B
    c) 2E70C + 1AA7F

12. Realiza las siguientes restas de números binarios:
    a) 111011 - 110
    b) 111110111 - 111001
    c) 1010111 - 11011 – 10011

13. Resta los siguientes números octales:
    a) 365 - 23
    b) 2732 - 1265
    c) 1773 – 65

14. Realiza las siguientes restas de números hexadecimales:
    a) 17A - 3C
    b) 20F5 - 31B
    c) 2E70C - 1AA7F

SISTEMAS DE NUMERACIÓN

La base de un sistema de numeración es la cantidad de dígitos diferentes que posee el sistema.

Sistema	Base
Decimal	10
Binario	2
Octal	8
Hexadecimal	16

Conversión entre los diferentes Sistemas de Numeración

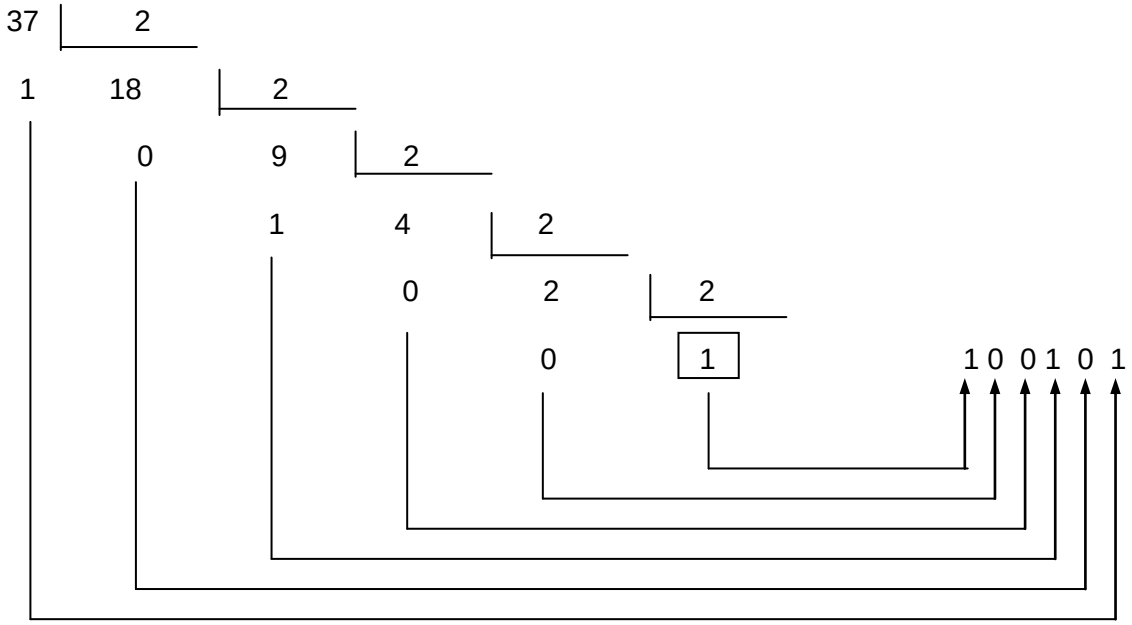
➤Decimal a:

✓Binario

Se divide el número decimal por 2 y se anota el resto, el resultado de la división se divide nuevamente por 2 y se vuelve a anotar el resto. Se continua este proceso hasta que el último resultado de la división sea menor que 2.

Se toma el último cociente y cada uno de los restos en el orden inverso a como fueron obtenidos, es decir que el primer resto será el bit menos significativo.

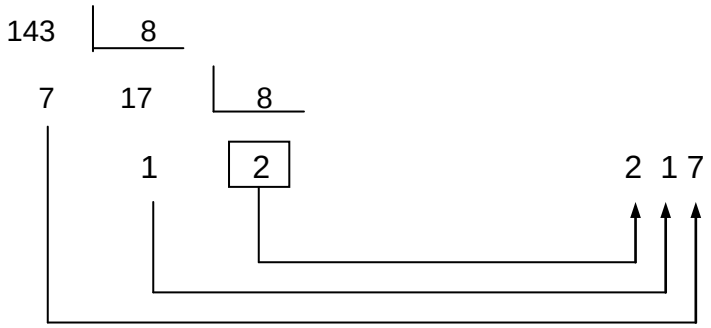
$$37_{10} = 100101_2$$



✓Octal

Se realiza el proceso de divisiones sucesivas, el divisor es 8.

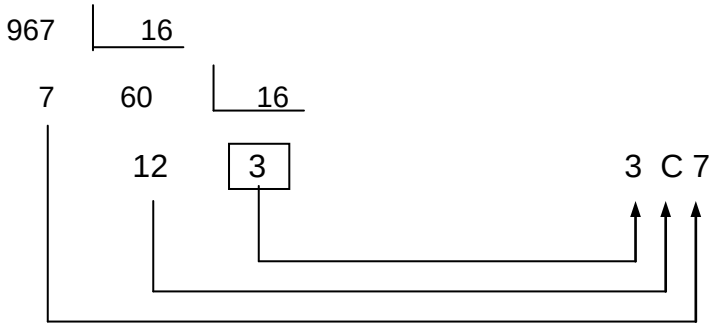
$$143 = 217_8$$



✓Hexadecimal

Se realiza el proceso de divisiones sucesivas, el divisor es 16.

$$967 = 3C7_H$$



➤Binario a:

✓Decimal

La conversión se obtiene resolviendo el polinomio de numeración de base 2.

$$10111_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 0 + 4 + 2 + 1 = 23_{10} = 23$$

✓Octal

La conversión puede realizarse directamente agrupando de a tres bits, de derecha a izquierda, y reemplazando los números por su equivalente octal, obtenidos de la tabla de conversión.

$$101111010_2 = 1\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6 + 1\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 =$$

$$= \left(1\cdot 2^2 + 0\cdot 2^1 + 1\right)\cdot 2^6 + \left(1\cdot 2^2 + 1\cdot 2^1 + 1\right)\cdot 2^3 + \left(0\cdot 2^2 + 1\cdot 2^1 + 0\right)\cdot 2^0 =$$

$$= 5\cdot 8^3 + 7\cdot 8^1 + 2\cdot 8^0 = 572_8$$

✓Hexadecimal

La conversión puede realizarse directamente, agrupando de a cuatro bits de derecha a izquierda y reemplazando a los mismos por su equivalente hexadecimal, obtenidos de la tabla de conversión.

$$11101001_2 = 1\cdot 2^7 + 1\cdot 2^6 + 1\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 =$$

$$= \left(1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0\right)\cdot 2^4 + \left(1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\right)\cdot 2^0 =$$

$$= 14\cdot 16^1 + 9\cdot 16^0 =$$

$$= E9_H$$

➢Octal a:

✓Decimal

Se resuelve el polinomio de numeración de base 8.

$$347_8 = 3\cdot 8^2 + 4\cdot 8^1 + 7\cdot 8^0 = 192 + 32 + 7 = 231_{10} = 231$$

✓Binario

Se toma cada dígito octal y se reemplaza por su equivalente binario, obtenidos de la tabla de conversión.

$$753_8 = 111101011_2$$

✓Hexadecimal

Conviene pasar de octal a binario y luego a hexadecimal.

$$654_8 = 110101100_2 = 1AC_H$$

➤Hexadecimal a:

✓Decimal

Se resuelve el polinomio de numeración de base 16.

$$2B9_H = 2 \cdot 16^2 + B \cdot 16^1 + 9 \cdot 16^0 =$$

$$= 2 \cdot 16^2 + 11 \cdot 16^1 + 9 \cdot 16^0 = 697_{10} = 697$$

✓Binario

Se toma cada dígito hexadecimal y se reemplaza por su equivalente binario, obtenidos de la tabla de conversión.

$$5A7_H = 10110100111_2$$

✓Octal

Conviene pasar a binario y luego a octal.

$$6C_H = 1101100_2 = 154_8$$

TABLA DE CONVERSIÓN

DECIMAL	BINARIO	OCTAL	HEXADECIMAL
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Completa la siguiente tabla según corresponda:

DECIMAL	BINARIO	OCTAL	HEXADECIMAL
57	111001	71	39
114	1110010	162	72
95	1011111	137	5F
989	1111011101	1735	3DD
752	1011110000	1360	2F0
2840	101100011000	5430	B18
1567	11000011111	3037	61F
1505	10111100001	2741	5E1

UNIDAD 2: TEORIA DE CONJUNTOS

Definiciones:

1.- **Conjunto**: es una lista, clase o colección de objetos bien definidos, objetos que, pueden ser cualesquiera: números, personas, letras, etc. Estos objetos se llaman elementos o miembros del conjunto.

Ejemplos: { 1, 3, 7, 10}
$\{x/x^2 -3x -2= 0\}$
{ Inglaterra, Francia, Dinamarca}

2.-**Subconjunto**: A es subconjunto de B si todo elemento de A lo es también de B.

Notación: $A \subset B \Leftrightarrow \forall x \in A \Rightarrow x \in B$

Ejemplo:
El conjunto C = {1,3,5} es un subconjunto del D = {5,4,3,2,1} ya que todo elemento de C pertenece al conjunto D.

3.- **Conjunto Universal**: es aquel conjunto que no puede ser considerado un subconjunto de otro conjunto, excepto de si mismo. Todo conjunto se debe considerar un subconjunto del Conjunto Universal.

Notación: U

Ejemplo:
A = {1,3,5} B = {2,4,6,8}

U = {0,1,2,3,4,5,6,7,8,9}

4.- **Conjunto Potencia:** se denomina conjunto potencia de A, P(A), a la familia de todos los subconjuntos del conjunto A. Sí el conjunto A tiene n elementos, el conjunto potencia de A tendrá 2^n elementos.

Notación:

Ejemplo:
A = {3,4,5}
P(A)= 2^3 = 8, lo que significa que pueden formarse 8 subconjunto de A.

P(A)= { {3}, {4}, {5}, {3,4}, {3,5}, {4,5}, {3,4,5}, ϕ }.

5.- **Conjunto Vacío**: es aquel que no posee elementos y es subconjunto de cualquier otro conjunto.

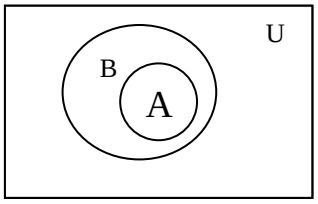
Notación: $\phi = \{ x / x \neq x \}$

Ejemplo:
$B= \{x/x^2 = 4$, x es impar}. B es entonces un conjunto vacío.

6.-**Diagrama de Venn**: Los diagramas de venn permiten visualizar gráficamente las nociones conjuntistas y se representan mediante círculos inscritos en un rectángulo. Los círculos corresponden a los conjuntos dados y el rectángulo al conjunto universal.

Ejemplo:

$A \subset B$



7.-**Conjuntos Finitos o Infinitos**: Los conjuntos serán finitos o infinitos, si sus elementos son o no factibles de contar.

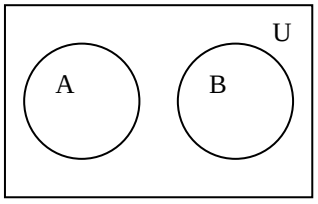
Ejemplo:

M= {a,e,i,o,u}, M es finito.

N={1,3,5,7...}, N es infinito.

8.- **Conjuntos disjuntos**: Dos conjuntos son disjuntos si no tienen elementos comunes.

Gráficamente:



Ejemplo:

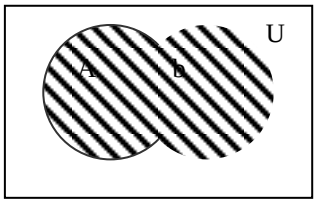
A= {1,3,8}, B={2,4,9}; A y B son conjuntos disjuntos.

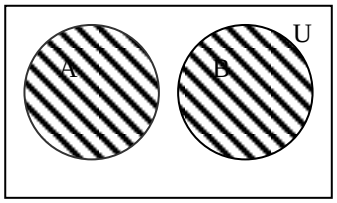
OPERACIONES CON CONJUNTOS

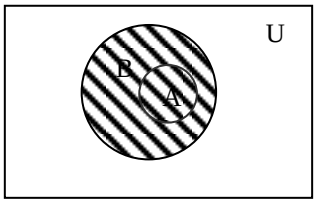
1.-**Unión de conjuntos**: La unión de dos conjuntos A y B es un conjunto cuyos elementos pertenecen a A o a B.

Notación: $A \cup B = \{x / x \in A \vee x \in B\}$

Gráficamente:







Ejemplo

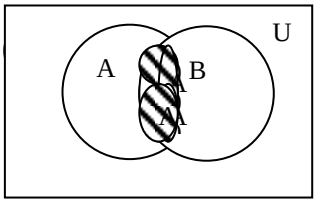
A={3,4,5,8,9} B={5,7,8,9,10}

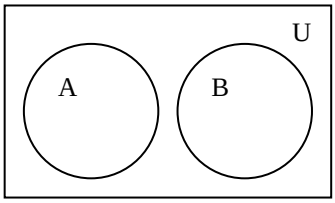
$A \cup B$={3,4,5,7,8,9,10}

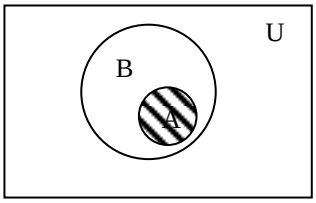
2.- **Intersección de conjuntos**: La intersección de dos conjuntos A y B, es un conjuntos cuyos elementos son comunes a A y B.

Notación: $A \cap B = \{x / x \in A \wedge x \in B\}$

Gráficamente:







Ejemplo:

A={7,8,9,10,11,12} B={5,6,9,11,13,14}

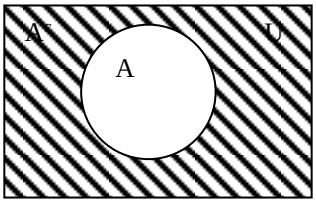
$A \cap B$={9, 11}

3.-**Complemento**: El complemento de un conjunto A, son todos los elementos que no están en el conjunto A
y que están en el universo.

Notación: $A^c = \{x / x \in U \wedge x \notin A\}$

$A^c = U - A$

Gráficamente:



Ejemplo:

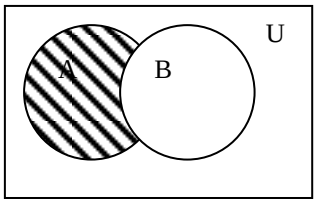
U= {1,2,3,...10} y A={ 3,4,6,7}

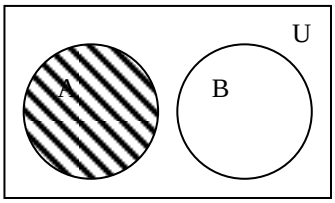
A^c= {1,2,5,8,9,10}

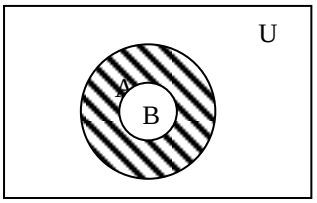
4.- **Diferencia de conjuntos**: La diferencia de dos conjuntos A y B, es un conjunto cuyos elementos son aquellos que están en el conjunto A, pero no en el conjunto B.

Notación: $A - B = \{x / x \in A \wedge x \notin B\}$

Gráficamente:







Ejemplo:

$C = \{u, v, x, y, z\}$ $D = \{s, t, z, v, p, q\}$

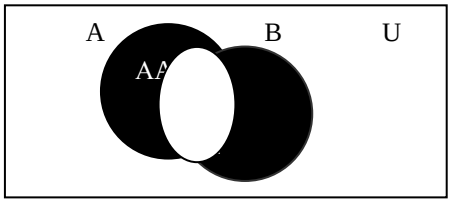
$C - D = \{x, y, u\}$

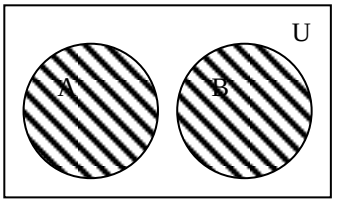
5.- **Diferencia Simétrica**: La diferencia simétrica de dos conjuntos A y B es un conjunto cuyos elementos son aquellos que están en A, pero no en B, unidos con aquellos que están en B, pero no en A.

Notación: $A \Delta B = \{x / x \in A \wedge x \notin B\} \cup \{x / x \notin A \wedge x \in B\}$

$A \Delta B = (A - B) \cup (B - A)$

Gráficamente:





Ejemplo:

$A = \{1,3,4,5,6,7,20,30\}$ $B=\{2,6,20,40,50\}$

$A \Delta B = \{1,3,4,5,7,30\} \cup \{2,40,50\}$

$A \Delta B = \{1,2,3,4,5,7,30,40,50\}$

6.-**Producto cartesiano**: El producto cartesiano entre dos conjuntos A y B es el conjunto de todos los pares ordenados que tienen como primera componente un elemento de A y como segundo componente un elemento de B.

Notación: $A \times B = \{(a, b) / a \in A \wedge b \in B\}$

Ejemplo:

A= {1,2} B={3,4,5}

A x B = {(1,3), (1,4), (1,5), (2,3), (2,4), (2,5)}

Observaciones:

1.- $n(A) = n \wedge n(B) = s \Rightarrow n(A \times B) = n \bullet s$

2.-Si $A = \phi \quad B = \phi \Leftrightarrow A \times B = \phi$

3.- $A \times B \neq B \times A$ siempre que se cumpla que $A \neq B$

7.- **Cardinalidad**:

$$n(A \cup B) = n(A) + n(B) - n(A \cap B)$$

$$n(A \cup (B \cup C)) = n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n(A \cap (B \cap C))$$

LEYES DE ALGEBRA DE CONJUNTO

1.- **Asociatividad**:

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

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

2.- **Conmutatividad**:

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

3.- **Distributividad**:

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

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

4.- **Absorción**:

$A \cup (A \cap B) = A$

$A \cap (A \cup B) = A$

5.- **Idempotencia**:

$A \cup A = A$

$B \cap B = B$

6.- **Identidad**:

$A \cup \phi = A$ $A \cap U = A$

$A \cup U = U$ $A \cap \phi = \phi$

7.-**Complemento**:

$A \cup A^c = U$ $A \cap A^c = \phi$

$(A^c)^c = A$ $U' = \phi$, $\phi' = U$

8.- **Ley de Morgan**:

$(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$

$A - B = A \cap B^c$

EJERCICIOS RESUELTOS

Demuestre:

1.-$(A - B) \cap B = \phi$

$(A \cap B^c) \cap B = \phi$

$A \cap (B^c \cap B) = \phi$

$A \cap \phi = \phi$

$\phi = \phi$

2.- $(A - B) \cap (A - C) = A - (B \cup C)$

$(A \cap B^c) \cap (A \cap C^c) = A - (B \cup C)$

$A \cap (B^c \cap A) \cap C^c = A - (B \cup C)$

$(A \cap A) \cap (B^c \cap C^c) = A - (B \cup C)$

$A \cap (B \cup C)^c = A - (B \cup C)$

$A - (B \cup C) = A - (B \cup C)$

3.- $n[A \cup (B \cup C)] = n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n[A \cap (B \cap C)]$

$n[A \cup (B \cup C)] = n(A) + n(B \cup C) - n[A \cap (B \cup C)]$

$= n(A) + n(B) + n(C) - n(B \cap C) - n[(A \cap B) \cup (A \cap C)]$

$= n(A) + n(B) + n(C) - n(B \cap C) - n(A \cap B) - n(A \cap C) + [n(A \cap B) \cap (A \cap C)]$

$= n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n[A \cap (B \cap C)]$

4.- $(A \cup A) \cap (A \cup B^c) = A$

$A \cap (A \cup B^c) = A$

$A = A$

$A = A$

5.- $(B \cap C) \cup A = (B \cup A) \cap (C \cup A)$

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

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

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

Simplificar:

1.- $A \cup [(B \cap (A \cup B)) \cap (A \cup (A \cap B))]$

$A \cup [(B \cap A) \cup (B \cap B)] \cap (A \cup A) \cap (A \cup B)$

$A \cup [(B \cap A) \cup (B \cap B)] \cap A \cap (A \cup B)$

$A \cup [B \cap A] \cup B = B$

$(A \cup B) \cap (A \cup A)$

$(A \cup B) \cap (A)$

A

Una encuesta aplicada a un grupo de jóvenes, acerca de las preferencias por alguna radio F.M. de la región, señaló que:

277 preferían Carolina
233 preferían Manquehue
405 preferían Tiempo
165 preferían Manquehue y Tiempo
120 preferían Manquehue y Carolina
190 preferían Carolina y Tiempo
105 preferían las tres estaciones de radio mencionadas

Determine:
a) ¿Cuántos jóvenes fueron encuestados?
b) ¿Cuántos jóvenes prefieren sólo Carolina?
c) ¿Cuántos jóvenes prefieren sólo Carolina y Tiempo?

Solo C= 277-120+105-190+105-105
Solo C= 72 jóvenes

Solo M= 233-120+105-105-165+105
Solo M= 53 jóvenes

Solo C y M= 120-105= 15 Jóvenes

Solo C y T= 190-105= 85 jóvenes

Solo M y T= 165-105= 60 jóvenes

Sólo T= 405-190+105-165+105-105= 545 jóvenes

Total jóvenes encuestados= 72+53+15+85+60+155+105= 545 jóvenes

a) Fueron encuestados 545 jóvenes
b) Sólo Carolina prefieren 72 jóvenes
c) Solo Carolina y Tiempo prefieren 85 jóvenes

EJERCICIOS PROPUESTOS

Demostrar:

1.- $A \cup (A^c \cap B) = A \cup B$

2.- $A \cap (A^c \cup B) = A \cap B$

3.- $(A - B) \cap B = \phi$

4.- $A-(B \cap C)=(A-B) \cup (A-C)$

5.- $A - (B \cap A) = A - B$

6.- $A^c - B^c = B - A$

Simplificar:

1.- Si $A \subset B \Rightarrow A \cup (B-A)=$

2.- $A \cap (A \cap B^c) \cup (A^c \cap B)=$

3.- $[A \cap (A \cup B)]^c \cup B$

4.- $[(A-B) \cap (B-C)] \cup (C-A)$

Ejercicios

1.- Una encuesta realizada a 2000 hombres reveló lo siguiente respecto a sus gustos por distintos tipos de mujeres:

800 preferían las rubias;
950 preferían las morenas;
750 preferían las colorinas;
150 preferían las rubias y morenas;
300 preferían las morenas y colorinas
250 preferían las rubias y colorinas
200 Sólo morenas y colorinas

Determine el número de hombres que :

a) Preferían los tres tipos de mujeres encuestados.
b) No preferían estos tipos de mujeres.

2.- En una reunión se determina que 40 personas son aficionadas al juego, 39 son aficionadas al vino y 48 a las fiestas, además hay 10 personas que son aficionadas al vino, juego y fiestas, existen 9 personas aficionadas al juego y vino solamente, hay 11 personas que son aficionadas al juego solamente y por último nueve a las fiestas y el vino.

Determinar:

a) El número de personas que es aficionada al vino solamente.
b) El número de personas que es aficionada a las fiestas solamente

3.- En una encuesta realizada a 320 alumnos de Ingeniería Comercial de la Universidad de Valparaíso, se descubrió que estos prefieren tres lugares para sus “carretes” de fin de semana:

95 prefieren ir al “Kamikaze”;
90 prefieren ir al “Playa”;
120 prefieren ir al “Bar de los Cuatro Vientos”;
30 prefieren ir al “Kamikaze” y al “Playa”
10 prefieren ir al “Kamikaze” y al “Bar de los Cuatro Vientos”
40 prefieren ir al “Playa” solamente
60 prefieren ir al “Kamikaze” solamente

Determine el número de estudiantes que prefieren:

a) Sólo ir al “Bar de los Cuatro Vientos”
b) Ir a los tres lugares
c) No salir y quedarse estudiando el fin de semana

1. Conceptos Básicos

Lógica
Es la ciencia que se encarga de estudiar las formas de las formas de las estructuras o esquemas del razonamiento formal. Está establece los principios fundamentales y proporciona los métodos necesarios que permite determinar que cierto razonamiento sea válido o no lo sea.

Lógica Proposicional
Es un sistema formal cuyos elementos más simples representan proposiciones y cuyas constantes lógicas llamadas conectivas lógicas, representan operaciones sobre proposiciones, capaces de formar otras proposiciones, capaces de formar otras proposiciones de mayor complejidad.

También se puede decir, que es la parte de la lógica que se encarga específicamente de las proposiciones.

Proposiciones
Son oraciones o enunciados susceptible de ser clasificada, inequívocamente, de verdadera o falsa, es decir, las proposiciones son aquellas expresiones que afirman algo, para la cual se dispone de algún criterio que nos permita establecer, sin lugar a duda, si tal afirmación es verdadera o falsa, por lo que su valor de verdad o valor veritativo (“V” o “F”).

Para la lógica proposicional, una proposición puede ser el resultado o composición de otras proposiciones, la primera se llamaría proposición resultante (también proposición compuesta) y la otra proposición componente (o proposición simple).

Proposición Simple
Se denomina enunciado o proposición simple o atómica a aquel enunciado o proposición que no tiene conectores lógicos.

Proposición Compuesta
Es un enunciado o proposición que está formada por dos o más enunciado o proposiciones simples; por consiguiente, están separadas por diferentes conectores lógicos.

Variables de proposiciones
Una proposición podrá ser representada por medio de una letra minúscula (p, q, r, s y t), o también una letra minúscula con subíndice de ser necesario. Las letras o símbolos utilizados para este fin se conocen como variables proposicionales. Ejemplo:

$$\neg(p \text{ A } q) \rightarrow \neg r$$

2. Constantes lógicas, operadores, conectores, conectivos.

En la lógica proposicional, las conectivas lógicas son tratados como funciones de verdad, es decir, como funciones que toman conjuntos de valores de verdad y devuelven valores de verdad.

El significado de las conectivas lógicas no es nada más que su comportamiento como funciones de verdad. Cada conectiva lógica se distingue de las otras por los valores de verdad que devuelve

frente a las distintas combinaciones de valores de verdad que puede recibir. Esto quiere decir que el significado de cada conectiva lógica puede ilustrarse mediante una tabla que despliegue los valores de verdad que la función devuelve frente a todas las combinaciones posibles de valores de verdad que puede recibir.

Conectivas

Una *conectiva lógica* o simplemente *conectiva*, (también llamado *operador lógico* o *conectores lógicos*) es un símbolo o palabra que se utilia para conectar dos fórmulas bien formadas o sentencias, de modo que el valor de verdad de la formula compuesta depende del valor de verdad de las fórmulas componentes.

Conectiva	Expresión en el lenguaje natural	Símbolo		Ejemplo
Negación	No	¬	~	**No** está lloviendo
Conjunción	Y	∧	&	Está lloviendo **y** está nublado
Disyunción	O	∨		Está lloviendo **o** está nublado
Condicional material	Si … Entonces	→	⊃	**Si** está soleado, **entonces** es de día
Bicondicional	Si y solo si	↔	≡	Está nublado **si y solo si** hay nubles visible
Disyunción opuesta	Ni … ni	↓		**Ni** está soleado **ni** está nublado
Disyunción exclusiva	O bien … o bien	Δ	Θ, ≢, ↮, W	**O bien** está soleado **o bien** está nublado

***Negación* (¬)**

Es una operación sobre la proposición donde su valor de verdad es un valor semántico. La negación de una proposición es ***verdadera*** cuando dicha proposición es falsa, y viceversa. Su formalización es de la forma ***¬p***.

***Conjunción* (∧)**

Es un conector lógico binario entre dos proposiciones, cuyo valor de la verdad resulta ***cierto*** sólo si ambas proposiciones son verdaderas, y es ***falso*** de cualquier otra forma. Su formalización es de la forma ***p ∧ q***.

***Disyunción* (∨)**

Es un conector lógico binario entre dos proposiciones, cuyo valor de la verdad resulta ***falso*** sólo si ambas proposiciones son falsas, y ***cierto*** de cualquier otra forma. Su formalización es de la forma **p ∨ *q***.

***Condición material* (→)**

También conocido como condicional funcional de verdad o como implicación material, y es una conectiva lógica que conecta dos proposiciones. El condicional es una función de verdad binaria, que se vuelve ***falso*** cuando **B** es falso siendo **A** verdadera, y se vuelve se ***verdadero*** en cualquier otro caso. Su formalización es de la forma ***p → q***.

***Bicondicional* (↔)**

También llamado equivalencia o doble implicación. Este es un operador lógico binario que conecta dos proposiciones y el valor de verdad es ***verdadero*** cuando ambas proposiciones tienen el mismo valor de verdad, es decir, ambas son verdaderas o falsas simultáneamente, de lo contrario es ***falso***. Su formalización es de la forma ***p ↔q***.

***Disyunción opuesta* (↓)**
También conocidas como la negación conjunta o flecha de peirse, es una conectiva lógica cuyo valor de verdad resulta ***verdadero*** sólo si ambas proposiciones son falsas, y ***falso*** de cualquier otra forma. Su formalización es de la forma ***p*** ↓***q****.*

***Disyunción exclusiva* (Δ)**
También llamada exclusivo o desigualdad material, es un operador lógico entre dos proposiciones; yel valor de verdad es ***verdadera***

3. Formalización de proposiciones

La operación consistente en sustituir las expresiones del lenguaje natural por símbolos lógicos se llama *formalización*. A la proposición debidamente formalizada se conocerá como *fórmula*.

Formalización
Por formalización se entienden dos aspectos relacionados:

- *Como proceso*: se refiere al proceso de traducción o simbolización de las proposiciones del lenguaje natural del lenguaje cotidiano al lenguaje lógico.
- *Como estructura*: la formalización nos permite explicitar la estructura ordenada o forma lógica de las proposiciones del lenguaje natural que se simbolizan o se traducen al lenguaje lógico.

Sintaxis
Todos los lenguajes se componen de unos símbolos y de unas reglas sintácticas que nos permite indicar que combinaciones de símbolos son correctos y cuales no los son.

Reglas para la formalización de fórmulas bien formuladas (FBF)
Una forma enunciativa es una expresan, en la que interviene variables de enunciados y conectivas que puede formarse utilizando las siguientes reglas:

- Toda proposición es FBF.
- Si *A* es un FBF, entonces ¬*A* también es FBF.
- Si *A* y *B* son FBF, entonces (A∧B), (A∨B) y (A→B) también es FBF.

Normas para la escritura de formas enunciativas

a) Una conectiva afecta a las letras proposicionales inmediatas o a los conjuntos inmediatos a ella que están entre paréntesis.
b) Reglas de precedencia:

Nivel 1:	( ), [ ], { }
Nivel 2:	¬
Nivel 3:	A, ∨
Nivel 3:	→, ↔

Pasos para la formalización

- Definir las proposiciones simples (p, q, r, …)
- Formalizar según su jerarquía

  - Menor jerarquía $\begin{cases} Coma \\ Punto\ y\ coma \end{cases}$

  - Mayor jerarquía $\begin{cases} Punto \\ Dos\ signos\ de\ puntuación \end{cases}$

Formalización de la conjunción

Proposición en lenguaje natural: Los perros son listos y los gatos egoístas
p: Los perros son listos
q: Los gatos son egoístas
Formalización: **p ^ q**

Proposición en lenguaje natural: Estudiaré, pero también veré la tele
p: Estudiaré
q: Veré la tele
Formalización: **p ^ q**

Proposición en lenguaje natural: Además de comer tarta, beberé sidra
p: Comeré tarta
q: Beberé sidra
Formalización: **p ^ q**

Formalización de la disyunción

Proposición en lenguaje natural: Voy al cine o al teatro
p: Voy al cine
q: Voy al teatro
Formalización: **p v q**

Proposición en lenguaje natural: Puedo ir al cine o no
p: Iré al cine
q: No iré al cine
Formalización: **p v q**

Formalización del condicional

Proposición en lenguaje natural: Si Misha es un gato, entonces escupirá bola de pelos
p: Misha es un gato
q: Misha escupirá bola de pelos
Formalización: **p → q**

Proposición en lenguaje natural: Si vas a la playa, te broncearás
p: Vas a la playa
q: Te broncearás
Formalización: **p → q**

Proposición en lenguaje natural: Pégame y tendrás tu merecido
p: Pégame
q: Tendrás tu merecido
Formalización: **p → q**

Proposición en lenguaje natural: Asistir a clase es condición necesaria para aprobar
p: Se asiste a clase
q: Se aprueba
Formalización: **q → p**

Formalización de la negación

Proposición en lenguaje natural: No voy a solucionarte el
problemap: Voy a solucionarte el problema
Formalización: **¬p**

Proposición en lenguaje natural: No es cierto que haya estado en ese cine
p: He estado en ese cine
Formalización: **¬p**

También se puede realizar *formalizaciones combinadas*, como se muestra a continuación:
Proposición en lenguaje natural: No voy a ir a París, pero si voy, me acordaré de ti y de tu madre.
p: Voy a París
q: Me acordaré de ti
r: Me acordaré de tu madre
Formalización: **¬p** A **[p → (q** A **r)]**

Proposición en lenguaje natural: Si vas al cine, entonces, o compras palominas o me envidiarás si tienes hambre.
p: Vas al cine
q: Compras palomitas
r: Me envidias
s: Tienes hambre
Formalización: **p → [q** ∨ **(s→r)]**

Proposición en lenguaje natural: Me quieras o no, tendrás que soportarme
p: Me quieres
q: Tienes que soportarme
Formalización: **(p** ∨ **¬p) → q**

Proposición en lenguaje natural: Si en Marte no hay agua, entonces no hay vida; en consecuencia, no hay marcianos ni platillos voladores,
p: Marte hay agua
q: Marte hay vida
r: Hay marcianos
s: Hay platillos voladores
Formalización: **(¬p → ¬q) → (¬r ∧ ¬s)** ó **(¬p → ¬q) →(r ↓ s)**

4. Tabla de la Verdad

Una tabla de verdad o tabla de valores de verdad, es una tabla que muestra el valor de verdad de una proposición compuesta, para cada combinación de verdad que se pueda asignar a sus componentes. También se puede decir, que es una estrategia de la lógica simple que permite establecer la validez de una o varias proposiciones compuestas en cuanto a cualquier situación, osea, determinan las condiciones necesarias para que sea verdadero la proposición propuesta. Cada proposición posee 2 valores de verdad, los cuales son ***verdadero*** o ***falso***, en el cual se podrá definir en la tabla de verdad como "*V*" y "*F*". Estudiaremos la tabla de verdad de cada uno de los conectivos ante mencionado.

Tabla de verdad de Negación

Como ya se había mencionado anteriormente, la negación es un operador unitario debido a que solo afecta el valor de verdad de una proposición. Está es la tabla de verdad más sencilla de realizar debido a que solo se puede obtener 2 valores de verdad, los cuales serán su valor negado u opuesto de su valor de verdad. A continuación, se muestra la tabla de verdad de la negación.

p	¬p
V	F
F	V

Para la elaboración de una tabla de verdad se deberá tener en cuenta la cantidad de proposiciones existente.

Ejemplo:

Proposición en lenguaje natural: No es cierto que Rodrigo es maestro

p: Rodrigo es maestro

Formalización: **¬p**

Tabla de verdad de Conjunción

A partir de este conector se puede complicar un poco la elaboración de la tabla de verdad, debido a que este es un conector binario, el cual se refiere que une dos proposiciones. En la conjunción existe claramente un valor de verdad *verdadero*, y este se debe cuando ambas proposiciones son ciertas, de lo contario es *falso*, es decir, que para el valor de verdad sea cierto no puede existir ninguna falsa. A continuación, se muestra la tabla de verdad.

p	q	p^q
V	**V**	V
V	**F**	F
F	**V**	F
F	**F**	F

Para elaborar esta tabla de verdad se consideró que existe 2 proposiciones, en el cual para determinar el número de filas a utilizar se recurrió a una pequeña formula que es:

N° de Fila = 2^n, donde **n**= N° de proposiciones

Como *n* es igual a 2, entonces

N° de Fila = $2^2 = 4$

La distribución de los valores de verdad de cada proposición es la siguiente:

$$N° \, de \, fila = 4 \qquad p \left\langle \begin{matrix} 2\,V \\ 2\,F \end{matrix} \right. \qquad q \left\langle \begin{matrix} V \\ F \end{matrix} \right. \qquad q \left\langle \begin{matrix} V \\ F \end{matrix} \right.$$

Debe tener en cuenta al momento de elaborar una tabla de verdad que siempre el primer valor a utilizar es *verdadero* y luego es *falso*. La cantidad de fila siempre será la mitad del valor, esto se debe a que como posee 2 valores de verdad, se considerará que para cada valor de verdad será el *n° de fila* entre 2.

Ejemplo:
Proposición en lenguaje natural: Juan es futbolista y Ana es voleibolista
p: Juan es futbolista
q: Ana es voleibolista
Formalización: **p ^ q**

Tabla de verdad de Disyunción
Este conector también es binario, por lo que se deberá realizar similar al anterior, pero en este caso se debe tener en cuenta que el único valor de verdad *falso* es cuando ambas proposiciones son falsas,de lo contario es *verdadera*, es decir, que para el valor de verdad *verdadero* al menos uno de las proposiciones debe ser verdadero. A continuación, se muestra la tabla de verdad.

p	q	p∨q
V	**V**	V
V	**F**	V
F	**V**	V
F	**F**	F

Ejemplo:
Proposición en lenguaje natural: Raúl es ingeniero o es profesor
p: Raúl es ingeniero
q: Raúl es profesor
Formalización: **p ∨ q**

Tabla de verdad de Condicional
Esta conectiva lógica como ya fue definida anteriormente también posee un valor de verdad *falso* cuando la proposición **B** es falsa siendo la proposición **A** verdadera, de lo contrario el valor de verdad es *verdadero*. A continuación, se muestra la tabla de verdad.

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

Ejemplo:
Proposición en lenguaje natural: Si Carla estudia, entonces ingresará a la universidad
p: Carla estudia
q: Carla ingresará a la universidad
Formalización: **p → q**

Tabla de verdad de Disyunción opuesta
Esta conectiva lógica como define un valor de verdad *verdadero* cuando ambas proposiciones son falsas, de lo contrario el valor de verdad es *falso*. A continuación, se muestra la tabla de verdad.

p	q	p↓q
V	**V**	F
V	**F**	F
F	**V**	F
F	**F**	V

Ejemplo:
Proposición en lenguaje natural: Ni está soleado ni está lloviendo
p: Está soleado
q: Está lloviendo
Formalización: **(p ↓ q)** ó **(¬p ^ ¬q)**

Tabla de verdad de Bicondicional
Este operador lógico como define un valor de verdad *verdadero* cuando ambas proposiciones tienen el mismo valor de verdad, de lo contrario el valor de verdad es *falso*. A continuación, se muestra la tabla de verdad.

p	q	p↔q
V	**V**	V
V	**F**	F
F	**V**	F
F	**F**	V

Ejemplo:
Proposición en lenguaje natural: Ana irá a la fiesta si y solo si tiene amigas
p: Ana irá a la fiesta
q: Ana tiene amigas
Formalización: **p ↔ q**

Tabla de verdad de Disyunción exclusiva
Este operador lógico como define un valor de verdad *verdadero* cuando ambas proposiciones poseen distinto valor de verdad, de lo contrario el valor de verdad es *falso*. A continuación, se muestra la tabla de verdad.

p	q	pΔq
V	**V**	F
V	**F**	V
F	**V**	V
F	**F**	F

Ejemplo:
Proposición en lenguaje natural: O bien Manuel juega o bien estudia
p: Manuel juega
q: Manuel estudia
Formalización: **p Δ q**

Podemos resumir las tablas de verdad de la siguiente forma

p	q	¬p	p∧q	p∨q	p→q	p↓q	p↔q	pΔq
V	**V**	F	V	V	V	F	V	F
V	**F**	F	F	V	F	F	F	V
F	**V**	V	F	V	V	F	F	V
F	**V**	V	F	F	V	V	V	F

Clasificación de la tabla de verdad

Las tablas de verdad se pueden clasificar según su resultado de las siguientes formas:

- **Tautología**: Es aquella proposición compuesta que es cierta para todos los valores de verdad que se asignen a cada una de las proposiciones, independientemente de los valores de verdad de las proposiciones que la componen, es decir, que todos los valores de verdad sean *"V"*.
- **Contradicción**: Una proposición compuesta corresponde a una contracción cuando los valores de verdad que se asignen a cada una de las proposiciones son falsas sin importar el valor de las proposiciones que la forman, es decir, que todos los valores de verdad sean *"F"*.
- **Contingencia**: Una proposición compuesta cuyos valores en sus diferentes líneas de la tabla dan como resultado *"V"* y *"F"* se conoce como contingencia, inconsistencia o falacia. Tecnológicamente, las contingencias se utilizan para construir circuitos de control y automatismo.

A continuación, estudiaremos estas tablas de verdad:

Hallar la tabla de la verdad del siguiente esquema proposicional

$$[(p \vee q) \wedge \neg q] \rightarrow p$$

p	q	¬q	p∨q	(p∨q)∧¬q	[(p∨q)∧¬q]→p
V	**V**	F	V	F	V
V	**F**	V	V	V	V
F	**V**	F	V	F	V
F	**F**	V	F	F	V

Se puede concluir, que la proposición es Tautológica ya que todos sus valores de verdad son ***verdaderos***.

Hallar la tabla de la verdad del siguiente esquema proposicional

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

p	q	¬p	p∧q	(p∧q)∧¬p
V	**V**	F	V	F
V	**F**	F	F	F
F	**V**	V	F	F
F	**F**	V	F	F

Se puede concluir, que la proposición es Contradicional ya que todos sus valores de verdad son ***falsos***.