

REPRESENTACIÓN DE LA INFORMACIÓN



INDICE



- 1.- Introducción a los sistemas de Numeración
- 2.- Decimal
- 3.- Binario
- 4.- Octal
- 5.- Hexadecimal
- 6.- Unidades de Medida de la Información

UTILIDADES



Ejemplos de Binario

- Redes: Calcular la subredes de una IP

Ejemplos de Octal

- LINUX: `chmod alumnos.odt 764`

Ejemplos de Hexadecimal

- Códigos Digital+: A3 FF 1D 46 88 AE 12 58
- Lenguaje Ensamblador: `Mov R4, 1D`
- Programación HTML: `<body bgcolor="#ffffff" text="#000000" link="#0000ff" vlink="#800080" alink="#ff0000">`

INFORMÁTICA

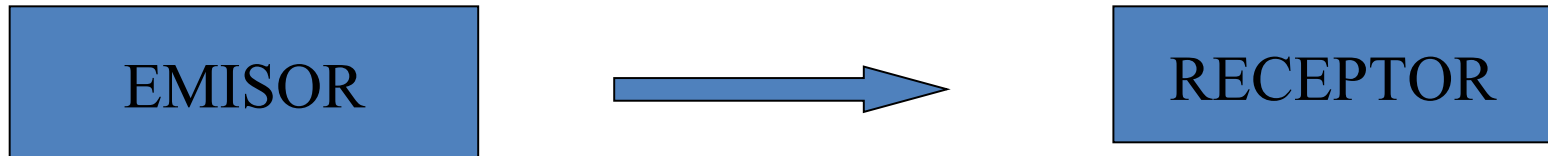


- Es la ciencia que estudia el tratamiento automático de la información.
- La informática usa las computadoras u ordenadores para el tratamiento y el proceso de la información.

Elementos de un sistema de comunicación



Está formado por los siguientes elementos básicos:



Emisor= el que genera o emite la información

Receptor= el que recibe la información

Medio o canal= vía de transmisión de la información

Elementos de un sistema de comunicación



- El emisor y el receptor pueden intercambiar sus papeles o realizar ambos papeles de forma simultánea.
- En la transmisión de información entre el ser humano y el ordenador

Emisor -> persona

Receptor -> ordenador (o viceversa)

Medio -> periféricos de entrada y salida del ordenador.

Transmisión de la información entre el ser humano y el ordenador

- Mediante caracteres alfanuméricos (letras {a, b, c,...,z} y números {0,1,2,...,9})

Ej. Los introducidos al ordenador mediante un teclado.

- Mediante sonidos: como los introducidos al ordenador a través de un micrófono, o que salen del ordenador por los altavoces



Transmisión de la información entre el ser humano y el ordenador

- Mediante vídeos: como las imágenes obtenidas a través de una cámara de vídeo.
- Mediante gráficos e imágenes: por ejemplo, una imagen introducida por un escáner o fotografías descargadas de una cámara digital.



Transmisión de la información entre el ser humano y el ordenador

En cada caso el **canal** es diferente => para proceder a la comunicación de los datos es necesario cambiar la forma en que los datos se representan

Esto es la **codificación** => es la traducción de los datos entre el emisor y el receptor.

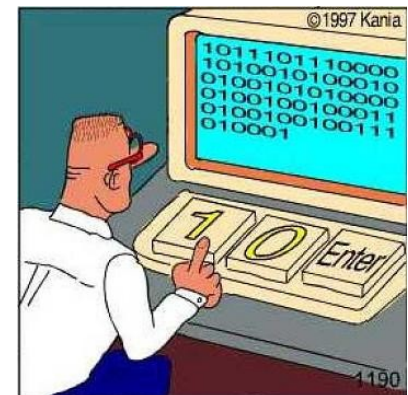


Código



- Llamaremos **código** a la ley de correspondencia entre el emisor y el receptor. Ej: ASCII
- Un código está compuesto de:
 - Un conjunto de reglas y convenios de transformación del alfabeto fuente.
 - Un nuevo alfabeto que sustituirá al original.

- La representación interna de la información en los ordenadores debe darse en forma de impulsos eléctricos.
- Esto se efectúa empleando señales biestables con dos posibles estados activado (1) – desactivado (0).
- Hay que codificar la información utilizando un código con dos únicos símbolos que representen a los dos estados.





- Utilizaremos el **1** para indicar que hay impulso
- Utilizaremos el **0** para indicar que no lo hay
- Todo el lenguaje se transcribirá a combinaciones de ceros y unos para que el ordenador lo pueda entender.
- Este código es el código binario, que está basado en el sistema de numeración binario, cuyos símbolos son el 0 y el 1.

SISTEMAS DE NUMERACIÓN

- Se define como el conjunto de símbolos utilizados para la representación de cantidades.
- Un sistema de numeración se distingue por su base
- Base= nº de símbolos que utiliza
- Ej. En el **sistema decimal** (base 10) utilizamos 10 dígitos o símbolos 0,1,2,3,4,5,6,7,8 y 9

<i>Binario</i>	0 1 © carlospes.com
<i>Octal</i>	0 1 2 3 4 5 6 7
<i>Decimal</i>	0 1 2 3 4 5 6 7 8 9
<i>Hexadecimal</i>	0 1 2 3 4 5 6 7 8 9 A B C D E F



SISTEMAS DE NUMERACIÓN

- Dependiendo de la posición que ocupe un dígito dentro de una cifra, representará las unidades, decenas, centenas, millares, etc.
- Por eso se dice que los sistemas de numeración **son posicionales**.
- Ej: sistema decimal (base 10) nº 6839
se puede expresar como suma de potencias de la base 10

$$(6 * 10 * 10 * 10) + (8 * 10 * 10) + (3 * 10) + (9 * 10^0) = 6839$$



Teorema fundamental de la numeración



El valor decimal de una cantidad expresada en otro sistema de numeración que utiliza otra base viene dado por la fórmula

$$N_i = \sum_{i=-d,n} (\text{dígito})_i * (\text{base})^i$$

i =indica la posición del dígito respecto a la coma.

El n° en decimal será el sumatorio de multiplicar cada dígito por la base elevada a su posición.

Si el n° tiene comas, i se iniciará con valor negativo.

El sistema binario



- Utiliza sólo dos dígitos (0 y 1) para representar cantidades por lo que su base es 2.
- Cada dígito de un número representado por este sistema se denomina bit (binary digit).
- Es un sistema posicional, los bits tienen distinto valor según la posición que ocupan

Conversión de binario a decimal

$2^0 = 1$	$2^{11} = 2048$
$2^1 = 2$	$2^{12} = 4096$
$2^2 = 4$	$2^{13} = 8192$
$2^3 = 8$	$2^{14} = 16384$
$2^4 = 16$	$2^{15} = 32768$
$2^5 = 32$	$2^{16} = 65536$
$2^6 = 64$	$2^{17} = 131072$
$2^7 = 128$	$2^{18} = 262144$
$2^8 = 256$	$2^{19} = 524288$
$2^9 = 512$	$2^{20} = 1048576$
$2^{10} = 1024$	

$$101111_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 47_{10}$$

$$10101_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21_{10}$$



Conversión de binario a decimal

Diagram illustrating the conversion of the binary number 110101_2 to the decimal number 53_{10} using positional weights:

The binary number 110101_2 is expanded as:

$$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

The corresponding powers of 2 are calculated:

$$32 + 16 + 0 + 4 + 0 + 1 = 53$$

Therefore, $110101_2 = 53_{10}$.

la séptima posición en el número

posición cero en el número

sistema de numeración en base 2

$$11011010 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

$$11011010 = 128 + 64 + 0 + 16 + 8 + 0 + 2 + 0 = 218$$

↑

Número 218 en el sistema binario

↑

El mismo número en el sistema decimal



Conversión de binario a decimal

Ejemplo:

Tenemos el número binario 1011,01

Cómo quedaría expresado en decimal?

Para ello aplicamos el teorema fundamental de la numeración

$$(1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) + (0 \cdot 2^{-1}) + (1 \cdot 2^{-2}) = 11,25$$



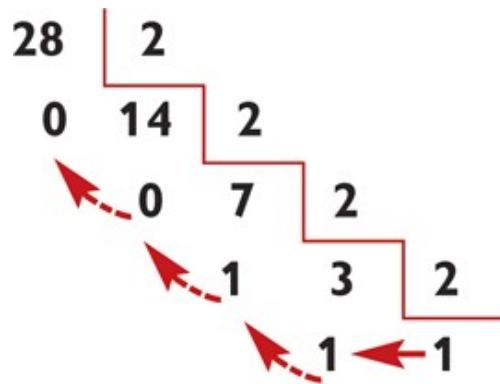
Conversión de un número decimal a binario

- Para representar un número en el sistema binario sólo podemos utilizar los dígitos 0 y 1.
- La forma de convertir a binario es dividir sucesivamente el número decimal y los cocientes que se van obteniendo por 2 hasta que el cociente sea menor de 2.
- La unión del último cociente y todos los restos obtenidos escritos en orden inverso será el número expresado en binario.

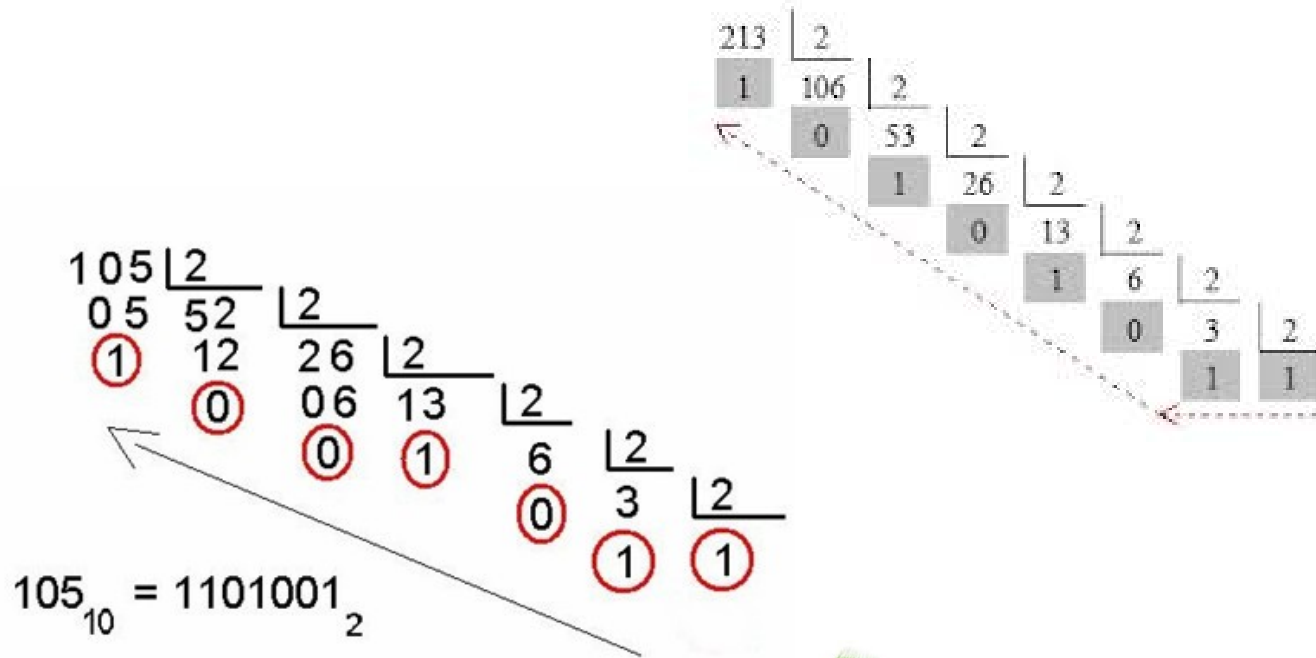


Conversión de un número decimal a binario

Ejemplo: el número decimal 28 en decimal es el número 11100 en binario.



$$28 = 11100_2$$



$$105_{10} = 1101001_2$$

Conversión de una fracción decimal a binario



- Consiste en multiplicar la parte fraccionaria por 2 hasta que de 0 como resultado.
- La parte entera formará los bits del número binario.
- A veces al multiplicar la parte fraccionaria por 2 nunca se llega a 0 -> se realizan varias multiplicaciones hasta tener los suficientes dígitos que permitan no sobrepasar un determinado error
Ej. Para un error inferior a

$$2^{-10}$$

Calcularemos hasta 10 dígitos

Decimal a binario con parte decin



Nº 12,125

Paso 1 pasar de decimal a binario la parte entera

$12:2=6$ resto 0

$6:2=3$ resto 0

$3:2=1$ resto 1

Parte entera en binario es: 1100

Decimal a binario con parte decin



- **Paso 2** pasar de decimal a binario la parte fraccionaria

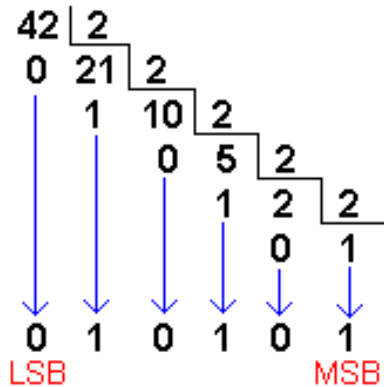
$0,125 * 2 = 0,25 \rightarrow 1^{\circ}$ dígito parte entera 0

$0,250 * 2 = 0,5 \rightarrow 2^{\circ}$ dígito parte entera 0

$0,5 * 2 = 1 \rightarrow 3^{\circ}$ dígito es 1

Como la parte decimal es 0 hemos llegado al final
Nº en binario 1100,001

Decimal a binario con parte decin



$$\begin{aligned}
 0.375 \times 2 &= 0.75 \\
 0.75 \times 2 &= 1.50 \\
 0.50 \times 2 &= 1.00
 \end{aligned}$$

Blue arrows show the progression of the fractional part conversion, leading to the binary digits .011.

Resultado: 101010.011



Conversion of the decimal number 13.80 to binary:

13	2
6	1
3	0
1	1
0	1

Blue arrows indicate the flow of the conversion process. The final binary digits are 101010.011.

$$\begin{aligned}
 13.80 \\
 0.80 \times 2 &= 1.6 \quad \mathbf{1} \\
 0.60 \times 2 &= 1.2 \quad \mathbf{1}
 \end{aligned}$$

Conversión decimal a binario con un mínimo error

Ej. Convertir $0,6_{(10)}$ a binario con un error inferior a 2^{-7} .

Parte entera: 0 en decimal es 0 en binario

Parte fraccionaria:

$0,6 * 2 = 1,2 \rightarrow 1^{\circ}$ dígito 1

$0,2 * 2 = 0,4 \rightarrow 2^{\circ}$ dígito 0

$0,4 * 2 = 0,8 \rightarrow 3^{\circ}$ dígito 0

$0,8 * 2 = 1,6 \rightarrow 4^{\circ}$ dígito 1

$0,6 * 2 = 1,2 \rightarrow 5^{\circ}$ dígito 1

$0,2 * 2 = 0,4 \rightarrow 6^{\circ}$ dígito 0

$0,4 * 2 = 0,8 \rightarrow 7^{\circ}$ dígito 0

Resultado en binario 0,1001100

Nota: si no se expresa el error mínimo se toman los dígitos hasta que el resultado en las multiplicaciones se repite, lo que es periódico

Ejemplo 0,1001



SUMA EN BINARIO



$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=0 \text{ con acarreo de } 1$$

Ejemplo suma sin acarreos



		1	0	0	0	0	->16
<u>±</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>->41</u>
	1	1	1	0	0	1	->57

Suma con acarreos



				1	1	1		acarreos
	1	0	1	0	1	1	1	->87
+	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>->33</u>
	1	1	1	1	0	0	0	->120

Cuando nos encontramos con 3 unos suma=1 y
acarreo=1

				1	1	1		acarreos
		1	1	0	1	1	1	->55
+	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>->35</u>
	1	0	1	1	0	1	0	->90

Suma con acarreos



$$\begin{array}{r} + 68 \\ 27 \\ \hline \end{array}$$

$$\begin{array}{r} + 01000100 \\ 00011011 \\ \hline \end{array}$$

SUMAR LOS PRIMEROS DOS
NÚMEROS.
PRIMERA SUMA

$$\begin{array}{r} + 95 \\ 14 \\ \hline \end{array}$$

$$\begin{array}{r} + 01011111 \\ 00001110 \\ \hline \end{array}$$

SUMAR TERCER NÚMERO.
SEGUNDA SUMA

$$\begin{array}{r} + 109 \\ 18 \\ \hline \end{array}$$

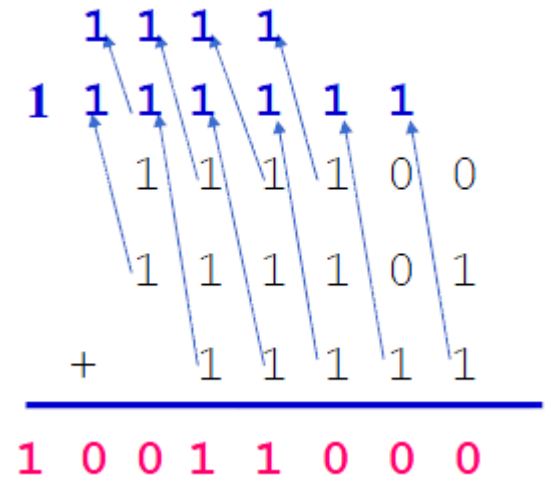
$$\begin{array}{r} 01101101 \\ + 00010010 \\ \hline \end{array}$$

SUMAR EL CUARTO NÚMERO .
SUMA FINAL

127

01111111

RESULTADO



La suma de un numero impar de unos da de
resultado un 1

La suma de un numero par de unos da de
resultado un 0

Suma binaria con decimales



	1	1	1	1		1	acarreos
		1	1	0	,	1	1 ->6,75
+		<u>1</u>	<u>0</u>	<u>1</u>	,	<u>0</u>	<u>1 ->5,25</u>
	1	1	0	0	,	0	0 ->12,00

RESTA EN BINARIO



$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$0 - 1 = 1 \text{ con acarreo de } 1$$

$$1 - 1 = 0$$

Resta sin acarreos



$$\begin{array}{r} 1110101 \rightarrow 117 \\ - \quad \underline{100001} \rightarrow 33 \\ \hline 1010100 \rightarrow 84 \end{array}$$

- Hay que restar de arriba hacia abajo
- Los acarreos se ponen en el hueco de en medio

Resta con acarreos



	1	1	1	0	1	0	1	->117
	1	1	1		1			acarreos
=		<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	->58
	0	1	1	1	0	1	1	->59
	1	1	0	0	1	0	1	->101
		1	1		1			acarreos
=			<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	->27
	1	0	0	1	0	1	0	->74

Resta con decimales



	1	0	0	0	1	,	0	1->17,25
	1	1	1	1	1			acarreos
-		<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	,	<u>1</u>	<u>1->11,75</u>
	0	0	1	0	1		1	0->5,5

Resta con decimales



	1	0	0	0	1	,	0	1->17,25
	1	1	1	1	1			acarreos
-		<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	,	<u>1</u>	<u>1->11,75</u>
	0	0	1	0	1		1	0->5,5

Multiplicación binaria



$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$

Se realiza como la multiplicación decimal, con la diferencia de que luego se hacen las sumas en binario.

Multiplicar 25(11001) por 5 (101)

			1	1	0	0	1	
	<u>x</u>				<u>1</u>	<u>0</u>	<u>1</u>	
			1	1	0	0	1	
+		0	0	0	0	0		
	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<hr/>		
	1	1	1	1	1	0	1	->125



Multiplicar 23(10111) por 14(1110)
 resultado 322



$$\begin{array}{r}
 10111 \\
 \times 1110 \\
 \hline
 00000 \\
 10111 \\
 + 10111 \\
 \hline
 101110000
 \end{array}$$

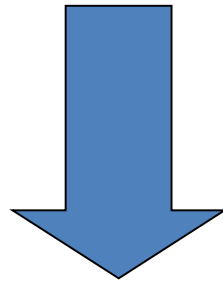
Multiplicación Binaria con decimales



División binaria



Se efectúa como la división decimal pero



Las multiplicaciones y las restas internas se hacen en binario.

$$\begin{array}{r}
 \text{Dividendo} \rightarrow \quad 1 \ 1 \ 0 \ 0 \ 1 \ 0 \mid 1 \ 0 \quad \leftarrow \text{Divisor} \\
 \underline{- \quad 1 \ 0} \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \leftarrow \text{Cociente} \\
 \quad \quad \quad 1 \ 0 \\
 \quad \quad \underline{- \quad 1 \ 0} \\
 \quad \quad \quad \quad 0 \ 0 \ 1 \ 0 \\
 \quad \quad \quad \quad \underline{- \quad 1 \ 0} \\
 \quad \quad \quad \quad \quad \quad 0 \quad \leftarrow \text{Resto}
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \mid 1 \ 0 \ 1 \\
 -1 \ 0 \ 1 \\
 \hline
 1 \ 1 \ 1 \\
 1 \ 0 \ 1 \\
 \hline
 0 \ 1 \ 0 \ 1 \\
 1 \ 0 \ 1 \\
 \hline
 0 \ 0 \ 0 \ 0 \ 1
 \end{array}$$

RESIDUO
0 1

Dividir en Binario



$ \begin{array}{r} 100100 \overline{) 11} \\ \underline{100} \\ 0011 \\ \underline{0011} \\ 0000 \end{array} $	$ \begin{array}{r} 110000 \overline{) 110} \\ \underline{110} \\ 000000 \end{array} $
$ \begin{array}{r} 1100 \overline{) 100} \\ \underline{100} \\ 0100 \\ \underline{0100} \\ 0000 \end{array} $	$ \begin{array}{r} 10100 \overline{) 100} \\ \underline{100} \\ 0010 \\ \underline{0010} \\ 0100 \\ \underline{0100} \\ 0000 \end{array} $

Dividir en Binario



$ \begin{array}{r} 100100 \overline{) 11} \\ \underline{100} \\ 0011 \\ \underline{0011} \\ 0000 \end{array} $	$ \begin{array}{r} 110000 \overline{) 110} \\ \underline{110} \\ 000000 \end{array} $
$ \begin{array}{r} 1100 \overline{) 100} \\ \underline{100} \\ 0100 \\ \underline{0100} \\ 0000 \end{array} $	$ \begin{array}{r} 10100 \overline{) 100} \\ \underline{100} \\ 0010 \\ \underline{0010} \\ 0000 \end{array} $

Dividir 282 (100011010) entre 10(1010)



1	0	0	0	1	1	0	1	0		<u>1</u>	0	1	0	
0	0	1	1	1	1					1	1	1	0	0
		0	1	0	1	0								
			0	0	0	0	1	0						

cociente 28 (11100)

Resto 2 (0010)

El sistema octal



- Los primeros sistemas informáticos utilizaban solo el sistema binario para interpretar y transformar los datos.
- Se recurrió a sistemas intermedios para traducir hacia y desde el sistema binario.
- Esos sistemas son el sistema octal y hexadecimal

El sistema octal



- Tiene como base de numeración 8
- símbolos {0, 1, 2, 3, 4, 5, 6 y 7}
- Es un sistema posicional
- Para convertir de decimal a octal se realizan divisiones sucesivas entre 8

- Ej.: 925 en decimal

$925:8=115$ resto 5

$115:8=14$ resto 3

$14:8=1$ resto 6

Nº en octal 1635

Sistema Decimal	Sistema Octal	Sistema Binario
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111

Convertir octal a decimal



- Empleamos el teorema fundamental de la numeración
- Ejemplo nº 1635 en octal

$$1635_{(8)} = (1 \cdot 8^3) + (6 \cdot 8^2) + (3 \cdot 8^1) + (5 \cdot 8^0) = 925_{(10)}$$

- Conversión de una fracción decimal a octal

Igual que con el sistema binario con el método de las multiplicaciones sucesivas ej. $12,0625_{(10)}$

parte entera $12_{(10)} = 14_{(8)}$

parte fraccionaria $0,0625 \cdot 8 = 0,5$

$$0,5 \cdot 8 = 4,0$$

Resultado en octal $14,04_{(8)}$

Conversión de una fracción octal a decimal

Se aplica el teorema fundamental de la numeración

Ejemplo: $11,3016_{(8)}$

$$(1 \cdot 8^1) + (1 \cdot 8^0) + (3 \cdot 8^{-1}) + (0 \cdot 8^{-2}) + (1 \cdot 8^{-3}) + (6 \cdot 8^{-4}) = 9,37841796875_{(10)}$$



Sistema hexadecimal



- Tiene como base de numeración 16
- Usa 16 símbolos para representar cantidades

Estos símbolos son {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F}

- Es un sistema posicional.

DECIMAL	BINARIO	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversión de un número decimal a hexadecimal

Se realizan divisiones sucesivas

Ejemplo $41565_{(10)}$

$41565:16=2597$ resto $13=D$

$2597:16=162$ resto 5

$162:16=10(A)$ resto 2

Resultado en hexadecimal A25D



Convertir un nº hexadecimal a decimal

- Utilizaremos el teorema fundamental de la numeración
- Ejemplo $A1D_{(16)}$
 $(10 \cdot 16^2) + (1 \cdot 16^1) + (13 \cdot 16^0)$

Resultado en decimal 2589



Conversión de una fracción decimal a hexadecimal



- Vamos a convertir a hexadecimal el nº $28,1975_{(10)}$
- La parte entera $28_{(10)} \rightarrow 1C_{(16)}$
- Para la parte decimal, realizaremos multiplicaciones sucesivas por 16

$$0,1975 * 16 = 3,16$$

$$0,36 * 16 = 5,76$$

$$0,16 * 16 = 2,56$$

$$0,76 * 16 = 12,16$$

$$0,56 * 16 = 8,96$$

$$0,16 * 16 = 2,56$$

$$0,96 * 16 = 15,36$$

se repite de nuevo

$$12,1975_{(10)} \rightarrow 1C,328F5C2$$

Conversión de una fracción decimal a hexadecimal



- Vamos a convertir a hexadecimal el nº $28,1975_{(10)}$
- La parte entera $28_{(10)} \rightarrow 1C_{(16)}$
- Para la parte decimal, realizaremos multiplicaciones sucesivas por 16

$$0,1975 * 16 = 3,16$$

$$0,36 * 16 = 5,76$$

$$0,16 * 16 = 2,56$$

$$0,76 * 16 = 12,16$$

$$0,56 * 16 = 8,96$$

$$0,16 * 16 = 2,56$$

$$0,96 * 16 = 15,36$$

se repite de nuevo

$$12,1975_{(10)} \rightarrow 1C,328F5C2$$

Conversión de una fracción hexadecimal a decimal



- Se procede aplicando el teorema fundamental de la numeración

Vamos a convertir a decimal el número $1AF,3A_{(16)}$

$$(1 \cdot 16^2) + (A \cdot 16^1) + (F \cdot 16^0) + (3 \cdot 16^{-1}) + (A \cdot 16^{-2}) = 431,2265625_{(10)}$$






Conversión entre sistemas



Decimal	Binario	Octal	Hexadecimal
0	000	0	0
1	001	1	1
2	010	2	2
3	011	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
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Conversión entre sistemas



	Binario	Octal	Decimal	Hexadecimal
Binario		Agrupar de a 3 bits	Formula Polinomial (potencias de 2)	Agrupar de a 4 bits
Octal	Escribir cada dígito en binario (3 bits)		Formula Polinomial (potencias de 8)	Pasar por binario
Decimal	Entera / 2 Fraccionaria * 2	Entera / 8 Fraccionaria * 8		Entera / 16 Fraccionaria * 16
Hexadecimal	Escribir cada dígito en binario (4 bits)	Pasar por binario	Formula polinomial (potencia de 16)	

Conversión de hexadecimal a binario

- Se sustituye cada dígito hexadecimal (0, 1, 2, 3, 4, ..., D, E, F) por su representación binaria utilizando 4 dígitos
- Ejemplo pasar a binario 73B,F1₍₁₆₎

7	3	B,	F	1
0111	0011	1011,	1111	0001



Conversión de binario a hexadecimal



- Se agrupan los dígitos binarios de 4 en 4 a partir del punto decimal a la izquierda y hacia la derecha
- Se sustituye cada grupo de 4 por su valor en hexadecimal
- Ejemplo: Pasar a hexadecimal $101011011_{(2)}$

0001	0101	1011
1	5	B

Resultado $15B_{(16)}$

Conversión de octal a binario

- Se sustituye cada dígito octal por su representación binaria utilizando tres dígitos binarios

- Pasar a binario $527_{(8)}$

5	2	7
101	010	111

$527_{(8)} \rightarrow 101010111_{(2)}$

- Pasar a binario $712,46_{(8)}$

7	1	2,	4	6
111	001	010	, 100	110

$712,46_{(8)} \rightarrow 111001010,100110_{(2)}$



Conversión de binario a octal

- Se agrupan los dígitos de 3 en 3 a partir del punto decimal a la izquierda y hacia la derecha
- Se sustituye cada grupo de tres por su equivalencia en octal

- Pasar a octal $10101100_{(2)}$

010 101 100

2 5 4

$10101100_{(2)} \rightarrow 254_{(8)}$

- Pasar a octal $1110110,1100111_{(2)}$

001 110 110 , 110 011 100

1 6 6 , 6 3 4

$11101,1100111_{(2)} \rightarrow 166,634_{(8)}$



Conversión de hexadecimal a octal

- Pasar 1AB0C,FB2₍₁₆₎ a octal

1º paso a binario 1AB0C,FB2₍₁₆₎

1 A B 0 C, F B 2

0001 1010 1011 0000 1100 ,1111 1011 0010

2º paso la cifra binaria a octal

011 010 101 100 001 100,111 110 110 010

3 2 5 4 1 4, 7 6 6 2

Resultado 325414,7662₍₈₎



Conversión de octal a hexadecimal



- Se pasa de octal a binario y de binario a hexadecimal.
- Pasar $3710,142_{(8)}$ a hexadecimal

3 7 1 0, 1 4 2

011 111 001 000, 001 100 010

lo dividimos en grupos de cuatro

0111 1100 1000,0011 0001

7 C 8, 3 1

resultado $7C8,31_{(16)}$

Conversión de un número decimal a hexadecimal

Se realizan divisiones sucesivas

Ejemplo $41565_{(10)}$

$41565:16=2597$ resto $13=D$

$2597:16=162$ resto 5

$162:16=10(A)$ resto 2

Resultado en hexadecimal A25D



Actividades 1: Sistemas de Numeración

Binario				111101
Decimal		123		
Octal	16			
Hexadecimal			5D	

Binario		11111111		
Decimal			228	
Octal				151
Hexadecimal	CA			



Actividades 2: Sistemas de Numeración

Binario			11101010	
Decimal		501		
Octal				257
Hexadecimal	1DB			

Binario	110101001			
Decimal		987		
Octal			701	
Hexadecimal				FEA



Representación de números con Signo



Números Enteros: son los números positivos y negativos mas el 0.

Signo y Magnitud: El 0 para el signo positivo y el 1 para el signo negativo

Problema: El cero tiene dos posibles representaciones, esto hay que solucionarlo con otras representaciones.

Ej: La cadena de bits **01001** se considera como 9 (binario sin signo) o + 9 (binario con signo) porque el bit de la extrema izquierda es cero.

Ej: La cadena de bits **11001** representa el equivalente binario a 25 cuando se considera un número sin signo, o -9 cuando se le considera un número con signo; ello se debe a que el uno de la posición extrema izquierda indica que será negativo y los cuatro bits representan 9 en binario.

Ej1: Representa en binario los siguientes números con signo: -38, +76, -85, +124, -

Ej2: Representa en decimal con y sin signo los siguientes números en binario: 11010, 011001, 1110011, 0110010, 110101010

Representación de la Informac



Complemento a 1:

- El bit mas a la izquierda es para el signo.
- Los números positivos se quedan igual
- Los negativos se cambian los ceros por unos y los unos por ceros.
- Existe una desventaja a la hora de utilizar el complemento a uno para representar números negativos que hace más adecuado el complemento a 2, y es que existen dos posibles representaciones para el número cero.

número binario:
0101 0110

Ca1: 1010 1001
+1
Ca2: 1010 1010

Complemento a 2:

- El bit mas a la izquierda es para el signo.
- Los positivos se quedan igual
- A los negativos se calcula el Ca1 y se le suma 1, si en el último dígito hay acarreo se desprecia.
- El cero tiene una única representación

Equivalencia Signo, Ca1 y Ca2



Decimal Con signo	Signo y magnitud	Complemento a 2's	Complemento a 1's
+15	0.1111	0.1111	0.1111
+14	0.1110	0.1110	0.1110
+13	0.1101	0.1101	0.1101
+12	0.1100	0.1100	0.1100
+11	0.1011	0.1011	0.1011
+10	0.1010	0.1010	0.1010
+9	0.1001	0.1001	0.1001
+8	0.1000	0.1000	0.1000
+7	0.0111	0.0111	0.0111
+6	0.0110	0.0110	0.0110
+5	0.0101	0.0101	0.0101
+4	0.0100	0.0100	0.0100
+3	0.0011	0.0011	0.0011
+2	0.0010	0.0010	0.0010
+1	0.0001	0.0001	0.0001
0	0.0000	0.0000	0.0000
	(1.0000)		(1.1111)
-1	1.0001	1.1111	1.1110
-2	1.0010	1.1110	1.1101
-3	1.0011	1.1101	1.1100
-4	1.0100	1.1100	1.1011
-5	1.0101	1.1011	1.1010
-6	1.0110	1.1010	1.1001
-7	1.0111	1.1001	1.1000
-8	1.1000	1.1000	1.0111
-9	1.1001	1.0111	1.0110
-10	1.1010	1.0110	1.0101
-11	1.1011	1.0101	1.0100
-12	1.1100	1.0100	1.0011
-13	1.1101	1.0011	1.0010
-14	1.1110	1.0010	1.0001
-15	1.1111	1.0001	1.0000
-16	-	1.0000	-

Actividades Signo, Ca1 y Ca2



Rellena los huecos de esta tabla

Cantidad	Con Signo	En C a 1	En C a 2
-121			
57			
-35			
112			
	110100010		
	011111		
		011100100	
		100010110	
			100000110
			011101110

Ejercicios: ¿Cuántos bits?



Números **sin** signo = $0 \dots 2^n - 1$ Números **con** signo = $-(2^{n-1} - 1) \leq x \leq 2^{n-1} - 1$

donde **n** es el n° de bits

Cuántos bits se requieren para representar los siguientes números decimales?

a) $17 = 2^5 - 1 = 31$ Se necesitan 5 bits

b) $35 = 2^6 - 1 = 63$ Se necesitan 6 bits

c) $-49 = 2^{7-1} - 1 = 63$ Se necesitan 7 bits

d) $+68 = 2^{8-1} - 1 = 127$ Con 8 bits se puede contar hasta 127

e) 31 =

f) 64 =

g) 114 =

h) -122 =

i) +205 =

j) +512 =

Ejercicios: ¿Cuántos bits?



- 1.- Dados n bits, ¿cuántos símbolos distintos puedo representar con ellos?
- 2.- Si dispongo de 6 bits para representar caracteres, ¿cuántos caracteres diferentes puedo representar?
a) 32 b) 64 c) 16
d) Todos los que quiera siempre que no se repitan las combinaciones
- 3.- Si se desea definir un código que permita representar los 100 caracteres del alfabeto chino ¿cuántos bits serían necesarios? Justifica la respuesta.
- 4.- Si se desea definir un código que permita representar los 27 caracteres del alfabeto castellano, tanto en mayúscula como en minúscula, así como los números del 0 al 9, ¿cuántos bits serían necesarios? Justifica la respuesta
- 5.- Si se desea definir un código que permita representar los 10 números decimales, junto con los caracteres especiales coma, punto, punto y coma, asterisco, comillas, más, menos y comilla simple, ¿cuántos bits serían necesarios? Justifica la respuesta

Representación interna de la información

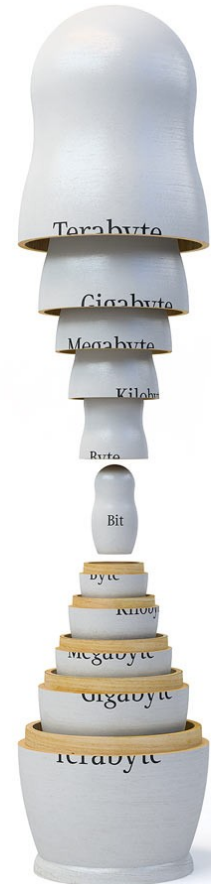


- El bit es la unidad mínima de almacenamiento
- Un bit puede almacenar dos valores 0 y 1
- Un carácter ocupa 8 bits
- 1 byte = 8 bits => un carácter ocupa 1 byte
- Cuando decimos que un archivo ocupa 5000 bytes, queremos decir que contiene el equivalente a 5000 caracteres. Hola = 4 bytes = 32 bits
- Palabra = tamaño de la información manejada en paralelo en el sistema
- Son comunes las palabras de 8, 16, 32 y 64 bits . A mayor tamaño de palabra mayor es la precisión y la potencia de cálculo del ordenador

Unidades de medida de la informac



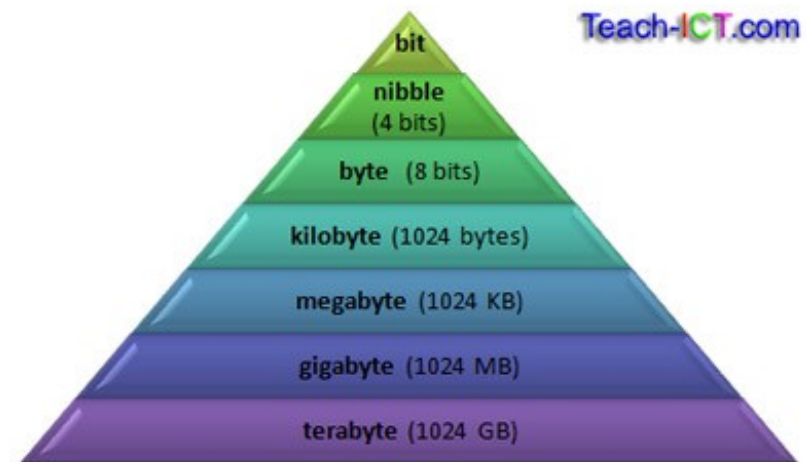
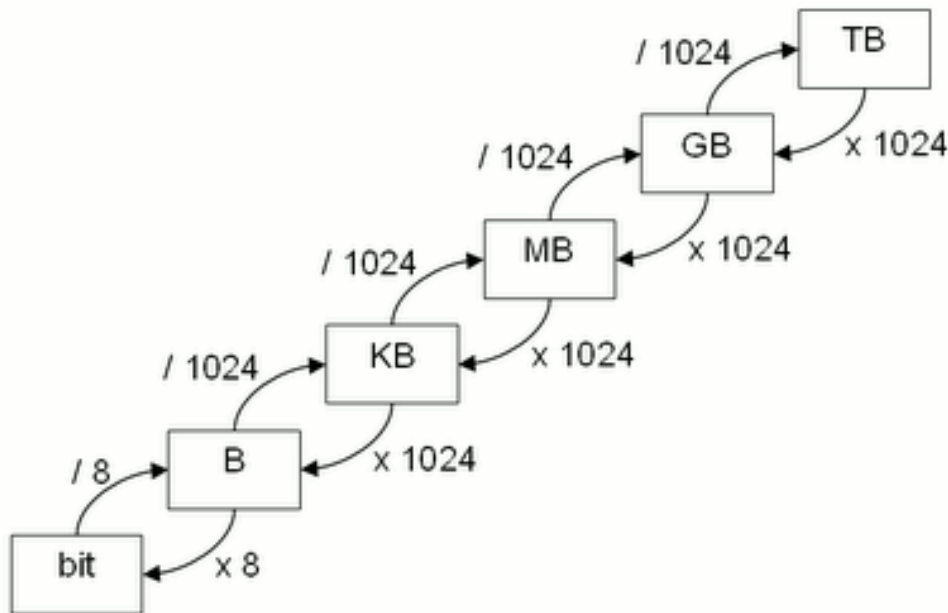
		Factor binario
Bytes	B	$2^0 = 1$
KiloBytes	Kb	$2^{10} = 1024$
MegaBytes	Mb	$2^{20} = 1\,048\,576$
GigaBytes	Gb	$2^{30} = 1\,073\,741\,824$
TeraBytes	Tb	$2^{40} = 1\,099\,511\,627\,776$
PetaBytes	Pb	$2^{50} = 1\,125\,899\,906\,842\,624$
ExaBytes	Eb	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
ZettaBytes	Zb	$2^{70} = 118\,059\,162\,071\,741\,130\,342\,4$
YottaBytes	Yb	$2^{80} = 1208\,925\,8196\,146\,291\,747\,061\,76$



Recuerda la **B** representa **bytes** y la **b** quiere decir **bit**

Ej: **30 MB** = 30 **megabytes** y **30 Mbs** = 30 **megabits** por segundo

Unidades de medida de la informac



Teach-ICT.com

Unidades de medida de la informac



- 1.- ¿Se quieren grabar canciones en MP3 en un CD de 700 MB para escuchar en el auto. Cuántos temas se pueden grabar si cada uno pesa en promedio 2,5 MB?
- 2.- ¿Cuántos KB son 0.14 TB?
- 3.- ¿Cuántas canciones se pueden guardar en un reproductor de MP3 de 1 GB si se desean almacenar archivos de música de 3 MB?
- 4.- Cuántas fotos puede almacenar una cámara digital con memoria interna de 2 GB, si cada foto tiene un tamaño de 2 MB?
- 5.- ¿Cuántas copias de dos archivos, uno de 548 KB y otro de 6 MB, pueden realizarse en un CD con capacidad de 700 MB?
- 6.- ¿Cuál es la cantidad de MB libres en un disco con una capacidad de almacenamiento de 3276,80 MB, si su espacio utilizado es 1 GB?

Unidades de medida de la informac



7.- ¿Cuál de los siguientes valores indican mayor peso de información almacenada?

1576648 bytes

1,2 MB

1675 KB

12000000 bits

8.- En un pendrive de 512 MB

a) ¿Cuánto espacio en MB me sobra si almaceno 15 archivos de 1210 bytes y 12 de 23420 KB cada uno?

b) ¿Cuántos archivos de tamaño 20200000 Bits cabrían en dicho espacio libre?

9.- Hemos comprado un lápiz de memoria de 4 GB de capacidad.

Tenemos tres archivos **sicho.avi** ocupa **950 MB**, **back_up.rar** ocupa **2880 MB** y **Nero.exe** **48 MB**. El lápiz se encuentra recién formateado.

a) ¿Caben o no los archivos en el pendrive?

b) ¿sobraría espacio en el caso que cupiesen los archivos anteriores para grabar más archivos?

c) si sobrara espacio ¿cuántas fotos de 1 Mb de capacidad podría grabar hasta agotar la capacidad del lápiz de memoria?

Unidades de medida de la informac



10.- Su cuenta de correo electrónico le permite enviar a sus contactos archivos de hasta 1 MB. Indique en cada caso si podrá enviar los siguientes archivos

- a. Una fotografía de sus vacaciones de 1.317 KB:
- b. Un archivo de música en formato mp3 987.424 bytes:

11.- Actualmente, en España, existe un canon digital que grava prácticamente todo aquello que es susceptible de ser utilizado para reproducir contenido con derechos de autor. Javier Cañete quiere hacer una copia de seguridad de los **6 GB** de fotografías personales que hay en su ordenador y ha pensado en utilizar **CD de 700 MB**. Si cada CD tiene un **canon de 0.17 €**, ¿cuánto tendrá que pagar nuestro amigo Javier debido al canon?

12.- Un capítulo de **Perdidos** ocupa **500 MB**. Si pretendo grabar los **25 capítulos** de la primera temporada en DVD de **4.7 GB**, ¿cuántos DVD me hacen falta?

Tamaño Memoria

**Con n bits hay 2^n Celdas, Casillas o direcciones de Memoria,
Que van desde la 0 a la $2^n - 1$**

Capacidad de Casilla o Celda = 8 bits o 16 o 32 o 64

Tamaño Mem. = N° Casillas x Capacidad Casilla



Ejercicios: ¿Cuanta Memoria?

1.- En la mayoría de la microcomputadoras las direcciones de las localidades de la memoria se especifican en hexadecimal. Estas direcciones son números secuenciales que identifican cada celda de la memoria.

a) Una microcomputadora en lo particular puede almacenar un numero de 8 bits en cada localidad de la memoria que va de 00 a FF ¿Cuántas localidades o celdas de memoria hay? ¿Cual es el tamaño en KB de esta memoria?

Sol: 256 direcciones y 0,25 KB

b) Se especifica que otra computadora tiene 4096 localidades o celdas de memoria. ¿Qué intervalo de direcciones hexadecimales utiliza esta computadora? ¿Cual es el tamaño en KB de esta memoria si cada dirección almacena un caracter?

Sol: 256 direcciones de 000 a FFF (hex), 4 Kbytes

2.- ¿Cuál es el tamaño de una memoria (kbytes) cuyas direcciones van de la 0000 a la FFFF y en cada celda solo cabe un carácter? **Sol: 64 Kbytes**

3.- ¿Cuál es el tamaño de una memoria (kbytes) cuyas direcciones van de la 000 a la FFF y la dirección 41D tiene almacenado el dato ***hola***? **Sol: 16 Kbytes**

4.- Tenemos una memoria Cache de 32 Kbytes, en cada casilla caben 8 letras

a) ¿Cuántas **direcciones** tiene esta memoria? b) ¿y *cuantos **bits** para cada dirección?*

Sol: a) 4096 direcciones o celdas b) 12 bits, Rango de direcciones de la 000 a FFF)



Representación de datos alfabético alfanuméricos



- Hemos visto cómo se almacenan las cantidades numéricas en el ordenador => vamos a ver cómo se almacenan el resto de caracteres
- Los códigos de E/S van a traducir la información a una representación que la máquina pueda interpretar y procesar

Representación de datos alfabético alfanuméricos



- Los datos llegan y salen del ordenador a través de los periféricos de entrada y salida
- Los fabricantes de periféricos usan códigos estándar (BCD, EBCDIC, ASCII o Unicode) para la traducción de la información.

Código ASCII



- Código Estadounidense Estándar para el intercambio de información
- Utiliza grupos de 7 bits para cada carácter => permite 128 caracteres (2^7)
- El código ASCII extendido usa 8 bits por carácter => permite 256 caracteres (2^8)

Dec	Hx	Char	Dec	Hx	Html	Chr	Dec	Hx	Html	Chr	Dec	Hx	Html	Chr
0	0	NUL (null)	32	20	 	Space	64	40	@	@	96	60	`	`
1	1	SOH (start of heading)	33	21	!	!	65	41	A	A	97	61	a	a
2	2	STX (start of text)	34	22	"	"	66	42	B	B	98	62	b	b
3	3	ETX (end of text)	35	23	#	#	67	43	C	C	99	63	c	c
4	4	EOT (end of transmission)	36	24	$	\$	68	44	D	D	100	64	d	d
5	5	ENQ (enquiry)	37	25	%	%	69	45	E	E	101	65	e	e
6	6	ACK (acknowledge)	38	26	&	&	70	46	F	F	102	66	f	f
7	7	BEL (bell)	39	27	'	'	71	47	G	G	103	67	g	g
8	8	BS (backspace)	40	28	((72	48	H	H	104	68	h	h
9	9	TAB (horizontal tab)	41	29))	73	49	I	I	105	69	i	i
10	A	LF (NL line feed, new line)	42	2A	*	*	74	4A	J	J	106	6A	j	j
11	B	VT (vertical tab)	43	2B	+	+	75	4B	K	K	107	6B	k	k
12	C	FF (NP form feed, new page)	44	2C	,	,	76	4C	L	L	108	6C	l	l
13	D	CR (carriage return)	45	2D	-	-	77	4D	M	M	109	6D	m	m
14	E	SO (shift out)	46	2E	.	.	78	4E	N	N	110	6E	n	n
15	F	SI (shift in)	47	2F	/	/	79	4F	O	O	111	6F	o	o
16	10	DLE (data link escape)	48	30	0	0	80	50	P	P	112	70	p	p
17	11	DC1 (device control 1)	49	31	1	1	81	51	Q	Q	113	71	q	q
18	12	DC2 (device control 2)	50	32	2	2	82	52	R	R	114	72	r	r
19	13	DC3 (device control 3)	51	33	3	3	83	53	S	S	115	73	s	s
20	14	DC4 (device control 4)	52	34	4	4	84	54	T	T	116	74	t	t
21	15	NAK (negative acknowledge)	53	35	5	5	85	55	U	U	117	75	u	u
22	16	SYN (synchronous idle)	54	36	6	6	86	56	V	V	118	76	v	v
23	17	ETB (end of trans. block)	55	37	7	7	87	57	W	W	119	77	w	w
24	18	CAN (cancel)	56	38	8	8	88	58	X	X	120	78	x	x
25	19	EM (end of medium)	57	39	9	9	89	59	Y	Y	121	79	y	y
26	1A	SUB (substitute)	58	3A	:	:	90	5A	Z	Z	122	7A	z	z
27	1B	ESC (escape)	59	3B	;	;	91	5B	[[123	7B	{	{
28	1C	FS (file separator)	60	3C	<	<	92	5C	\	\	124	7C	|	
29	1D	GS (group separator)	61	3D	=	=	93	5D]]	125	7D	}	}
30	1E	RS (record separator)	62	3E	>	>	94	5E	^	^	126	7E	~	~
31	1F	US (unit separator)	63	3F	?	?	95	5F	_	_	127	7F		DEL



Código ASCII



1 Carácter = 1 byte = 8 bits

¿Cuántos bits se necesitan para almacenarse la frase: **El aula verde**?

¿La sección de lectores de un diario, impone como restricción para la publicación de las cartas que el texto no supere los **1500 caracteres**.
Cuál será el tamaño en KB de un archivo de texto que contenga esa información?

Unicode



- Utiliza 16 bits, lo que permite codificar 65536 caracteres.
- El objetivo de Unicode es representar cada elemento usado en la escritura de cualquier idioma del planeta
- **Unicode** está reemplazando al código **ASCII** ya que permite a los usuarios gestionar todos los lenguajes del planeta además de símbolos matemáticos que simplifican el intercambio de información científica.
- Se Utiliza en los Navegadores
- Antes de Unicode, por ejemplo, la unión Europea utilizaba cientos de sistemas de codificación distintos para cubrir todos sus idiomas.

EBCDIC



- Nombre: Código BCD extendido de caracteres decimales codificados en binario para el intercambio de información (Extended BDC Interchange Code)
- Sistema de codificación que tiene como objetivo la representación de caracteres alfanuméricos.
- Es utilizado por la empresa IBM.
- Cada carácter tienen 8 bits => podemos representar hasta 256 caracteres.