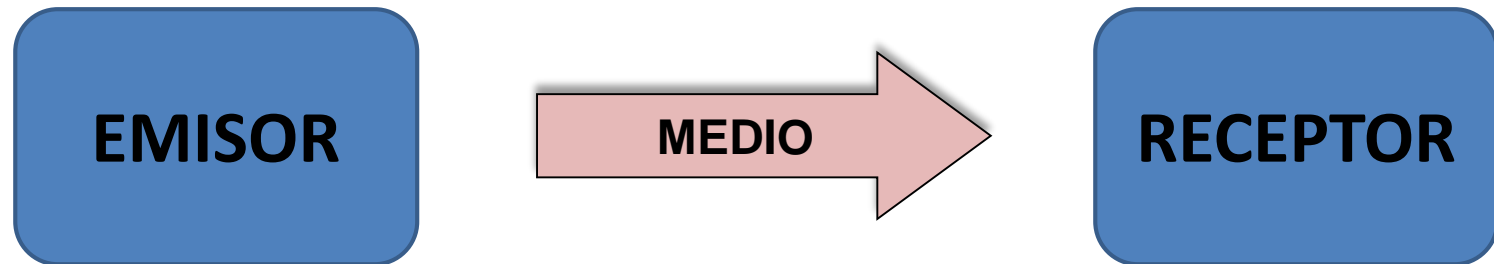


# representación de la información



# Sistema de comunicación

2

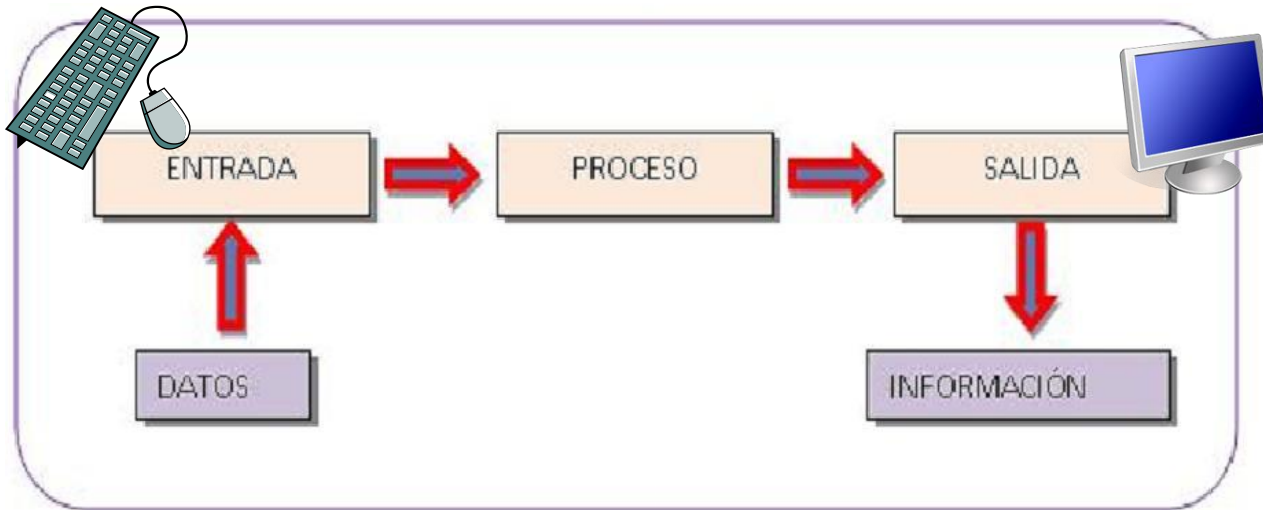


Elementos de un sistema de comunicación

Emisor y receptor pueden intercambiar sus papeles o incluso realizar ambos papeles de forma simultánea.

# Sistema de información

3



Relación establecida entre los datos y la información a través de un proceso de datos.

P.E. transmisión de datos entre el ser humano y el ordenador

# Transmisión de la información entre humano y PC

4

- ❑ Mediante **caracteres alfanuméricos** (teclado)
- ❑ Mediante **sonidos** (micrófono altavoces)
- ❑ Mediante **vídeos** (cámara de vídeo)
- ❑ Mediante **imágenes** (cámara de fotos, escáner)
- ❑ En general cualquier tipo de dato enviado/recibido por un **periférico** del ordenador.
- ❑ En cada canal los datos se representan de formas diferentes. Es necesario traducir o **codificar** los datos cuando los códigos utilizados por el emisor, el canal y el receptor difieren.

# ¿Quién es el emisor, el receptor y el medio?

5

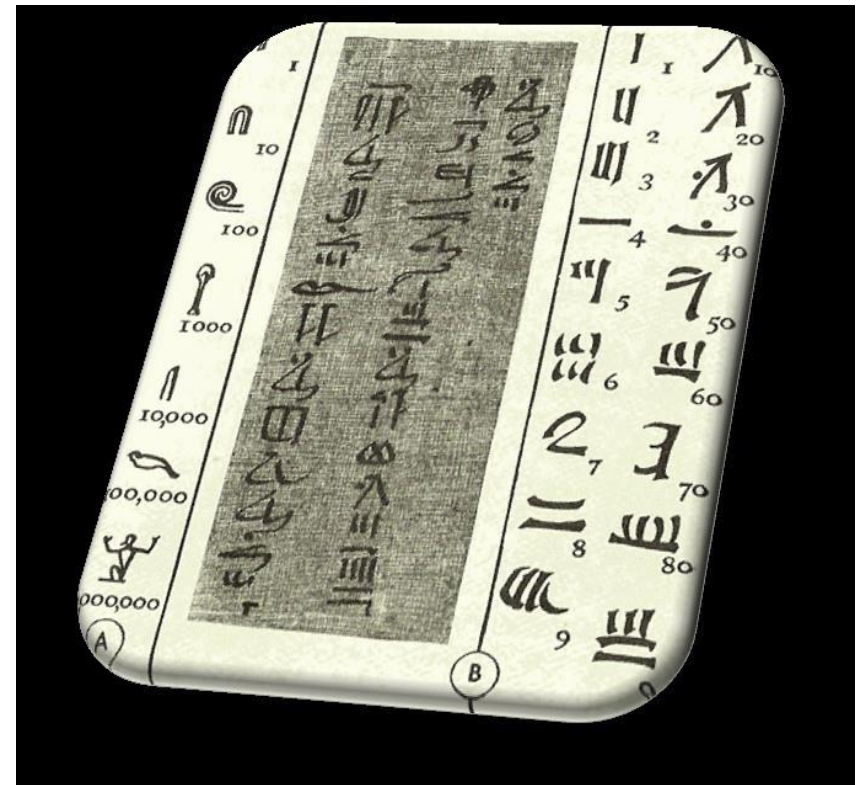
- ☐ **Una noticia en la radio**
- ☐ **Dos amigos hablando en la calle**
- ☐ **Un ordenador descargando un archivo de internet.**

# Código

6

Un CÓDIGO está compuesto de:

- Un **conjunto de reglas y convenios de transformación** del alfabeto fuente.
- Un nuevo alfabeto que sustituirá al alfabeto original.



# Código Morse

7

<b>A</b>	<b>.-</b>	<b>M</b>	<b>--</b>	<b>Y</b>	<b>-.--</b>	<b>6</b>	<b>-....</b>
<b>B</b>	<b>-...</b>	<b>N</b>	<b>-. .</b>	<b>Z</b>	<b>---..</b>	<b>7</b>	<b>---...</b>
<b>C</b>	<b>-.-.</b>	<b>O</b>	<b>---</b>	<b>Ä</b>	<b>.-.-</b>	<b>8</b>	<b>---..</b>
<b>D</b>	<b>-..</b>	<b>P</b>	<b>.--.</b>	<b>Ö</b>	<b>---.</b>	<b>9</b>	<b>----.</b>
<b>E</b>	<b>.</b>	<b>Q</b>	<b>--.-</b>	<b>Ü</b>	<b>..--</b>	<b>.</b>	<b>.-.-.-</b>
<b>F</b>	<b>...-</b>	<b>R</b>	<b>.-.</b>	<b>Ch</b>	<b>----</b>	<b>,</b>	<b>---..--</b>
<b>G</b>	<b>--.</b>	<b>S</b>	<b>...</b>	<b>0</b>	<b>-----</b>	<b>?</b>	<b>..--..</b>
<b>H</b>	<b>....</b>	<b>T</b>	<b>-</b>	<b>1</b>	<b>.----</b>	<b>!</b>	<b>..--.</b>
<b>I</b>	<b>..</b>	<b>U</b>	<b>..-</b>	<b>2</b>	<b>..----</b>	<b>:</b>	<b>---...</b>
<b>J</b>	<b>.----</b>	<b>V</b>	<b>...-</b>	<b>3</b>	<b>...--</b>	<b>“</b>	<b>.-...-</b>
<b>K</b>	<b>-. -</b>	<b>W</b>	<b>.--</b>	<b>4</b>	<b>....-</b>	<b>‘</b>	<b>.-----</b>
<b>L</b>	<b>.-...</b>	<b>X</b>	<b>-..-</b>	<b>5</b>	<b>.....</b>	<b>=</b>	<b>-...-</b>

# Código Internacional de Señales: Banderas

					
Alfa	Bravo	Charlie	Delta	Echo	Foxtrot
					
Golf	Hotel	India	Julieta	Kilo	Lima
					
Mike	November	Oscar	Papa	Quebec	Romeo
					
Sierra	Tango	Uniform	Victor	Whiskey	Xray
					
Yankee	Zulu	Gallardete Característico	Primer Repetidor	Segundo Repetidor	Tercer Repetidor
					
1	2	3	4	5	6
					
7	8	9	0		



# Sistemas de numeración

9

- Se define un **sistema de numeración** como el **conjunto de símbolos** utilizados para la representación de cantidades, así como las **reglas** que rigen dicha representación.
- **Base:  $n^{\circ}$  de símbolos** que utiliza. Coeficiente que determina cuál es el valor de cada símbolo dependiendo de su posición.

# Representación interna del ordenador

10

- El ordenador utiliza en sus cálculos y en las distintas comunicaciones principalmente *tres sistemas numéricos* el **binario**, el **decimal** y el **hexadecimal**.  
**¿ejemplos?**
- El **sistema decimal** (diez símbolos) es un **sistema posicional**, el valor del dígito depende de su posición.  
En realidad posee dos valores, uno absoluto marcado por el valor del dígito, y otro relativo marcado por su posición.

# Sistema posicional

11

Si tenemos un número decimal:

**6872**

Se lo puede representar de la siguiente manera:

6 millares

8 centenas

7 decenas

2 unidades

Es decir se puede representar como sumas de **potencias de base 10**:

$$6 \times 10^3 + 8 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$$

De hecho, es como lo expresamos oralmente:

**seis mil / setecientos / setenta / y dos**

# Teorema Fundamental de la Numeración

El valor de un número en *base decimal*, expresado en otro sistema de numeración que utiliza otra base viene dado por la fórmula:

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

Donde:

*i* = posición respecto a la coma. Para los dígitos a la dcha., la *i* es negativa empezando en -1. Para los de la izda. son positivos empezando por 0.

*d* = n° de dígitos a la dcha. de la coma.

*n* = n° de dígitos a la izda. de la coma – 1.

*dígito* = cada uno de los que componen el número.

*base* = base del sistema de numeración

$$... + X_4 * B^4 + X_3 * B^3 + X_2 * B^2 + X_1 * B^1 + X_0 * B^0 + X_{-1} * B^{-1} + X_{-2} * B^{-2} + X_{-3} * B^{-3} + ...$$

donde *X<sub>i</sub>* es el dígito y *B* la base.

# Ejemplo 1

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

**(6578)<sub>10</sub>**

d = 0 , n° de dígitos a la dcha. de la coma

i = -d = 0 , posición respecto a la coma.

n = 3 , n° de dígitos a la izda. de la coma – 1.

i	3	2	1	0
Pesos	$10^3$	$10^2$	$10^1$	$10^0$
Dígitos	6	5	7	8

Sumamos según la fórmula:

$$(6 \cdot 10^3) + (5 \cdot 10^2) + (7 \cdot 10^1) + (8 \cdot 10^0) = 6000 + 500 + 70 + 8 = 6578$$

## Ejemplo 2

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

**(34,275)<sub>10</sub>**

d=3 dígitos a la dcha. de la coma.

i=-d=-3 el valor *i* empieza en -3.

n=2-1=1 dígitos a la izda. de la coma.

<b>i</b>	<b>1</b>	<b>0</b>		<b>-1</b>	<b>-2</b>	<b>-3</b>
<b>Pesos</b>	<b>10<sup>1</sup></b>	<b>10<sup>0</sup></b>		<b>10<sup>-1</sup></b>	<b>10<sup>-2</sup></b>	<b>10<sup>-3</sup></b>
<b>Dígitos</b>	<b>3</b>	<b>4</b>	<b>,</b>	<b>2</b>	<b>7</b>	<b>5</b>

Sumamos según la fórmula:

$$(3 \cdot 10^1) + (4 \cdot 10^0) + (2 \cdot 10^{-1}) + (7 \cdot 10^{-2}) + (5 \cdot 10^{-3}) = 30 + 4 + 0,2 + 0,07 + 0,005 = \mathbf{34,275}$$

# Actividad

La cantidad

**$(112,02)_3$**

está expresada en el sistema de numeración de **base 3**, que emplea los dígitos 0, 1, 2 para representar las cantidades. Averigua cuál es la representación de este número en el sistema decimal.

i	2	1	0		-1	-2
Pesos	$3^2$	$3^1$	$3^0$		$3^{-1}$	$3^{-2}$
Dígitos	1	1	2	,	0	2

$$(1 \cdot 3^2) + (1 \cdot 3^1) + (2 \cdot 3^0) + (0 \cdot 3^{-1}) + (2 \cdot 3^{-2}) = 14,2222$$

# actividades

16

- Expresar las cantidades **76890** y **234,765** según el teorema fundamental de la numeración.
- Expresar en decimal estas cantidades dadas en diversos sistemas de numeración y bases distintos:
  - a) 201,12 en base 4 (dígitos 0, 1, 2, 3)
  - b) 340,31 en base 5 (dígitos 0, 1, 2, 3, 4)
  - c) 215,241 en base 6 (dígitos 0, 1, 2, 3, 4, 5)



# Sistema binario



# Sistema binario

18

- El ordenador utiliza el **sistema binario** o **base 2**, es decir, sólo emplea dos símbolos, los dígitos 0 y 1.
- Por tanto se representa mediante potencias de base 2 (que llamamos **peso** como hemos visto).

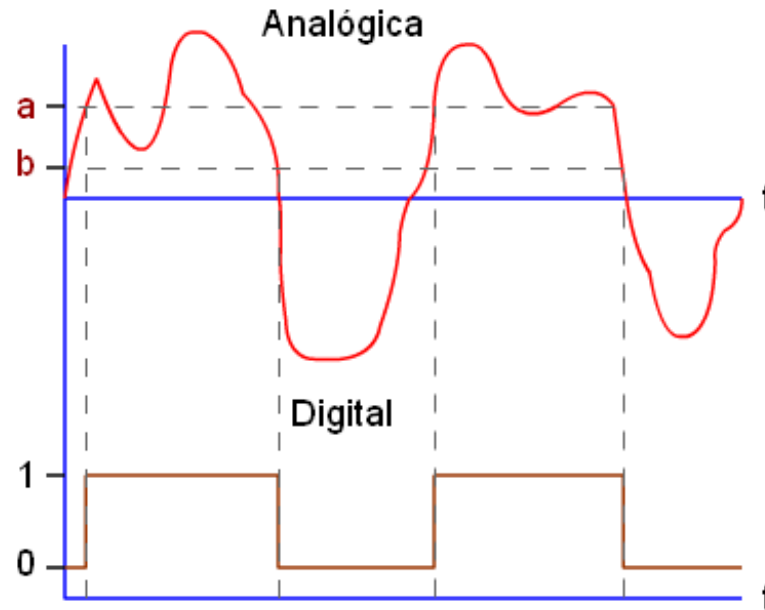
**Nº  $(1011,11)_2$  expresado en decimal:**

$$(1011,11)_2 = (1 \cdot 2^3) + (0 \cdot 2^2) + 1 \cdot 2^1 + (1 \cdot 2^0) + (1 \cdot 2^{-1}) + (1 \cdot 2^{-2}) = (11,75)_{10}$$

- **Esto es, conversión de Binario a Decimal.**

# ¿por qué usar sistema binario?

19



Empleo de señales **biestables** con dos posibles estados:  
**Tensión No, Tensión Sí.**

Mínimo incremento para cambiar de estado, evitando Ruido. Típicamente **5 Voltios** de incremento, pero varía según el sistema y su finalidad.

# Conversión binario a decimal

- Para convertir a decimal, basta con sumar los pesos donde corresponda un **1** del binario.

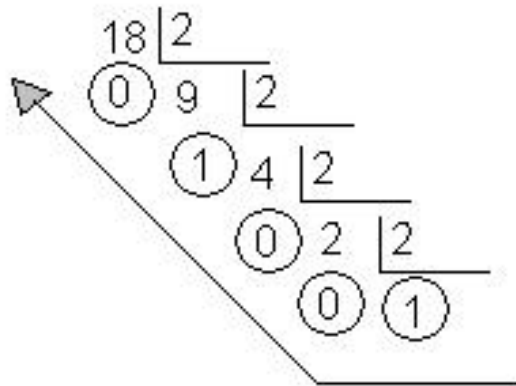
Pesos asociados								Nº decimal
$2^3$	$2^2$	$2^1$	$2^0$	,	$2^{-1}$	$2^{-2}$	$2^{-3}$	
8	4	2	1	,	0,5	0,25	0,125	
	1	1	0					6
1	0	1	1	,	0	1		11,25
1	1	0	1	,	1	0	1	13,625
		1	1	,	1			3,5
1	0	0	1	,	1	1		9,75

NO ENTRA

# Conversión de decimal a binario

21

- Dividir sucesivamente entre 2, y después, tomar **el último cociente y todos los restos en orden inverso** a como han aparecido

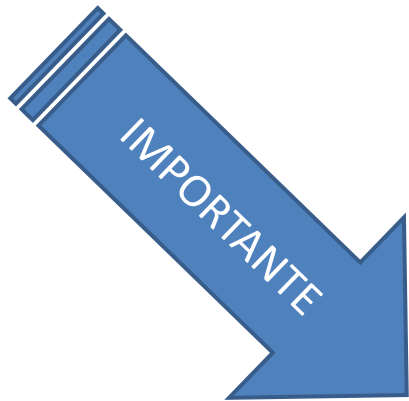


Por lo tanto, el número  $(18)_{10}$  equivale al número  $(10010)_2$

# Consejo

22

- La **cantidad de dígitos** de un número en binario dependerá del valor de dicho número en el sistema decimal.
- Para representar cualquier cifra decimal en binario podemos guiarnos por las potencias de 2:



Número decimal	Dígitos en binario
Menor que 2 ( $2^1$ )	1
Menor que 4 ( $2^2$ )	2
Menor que 8 ( $2^3$ )	3
Menor que 16 ( $2^4$ )	4
Menor que 32 ( $2^5$ )	5
Menor que 64 ( $2^6$ )	6
...	...
Menor que $2^n$	n

# Actividad

1. Indica, sin pasar a decimal, cuál es el mayor de los dos:  
a) 01001000  
b) 01000010
2. Expresa, en código binario, los números  $(55)_{10}$  y  $(205)_{10}$ .
3. ¿Cuántos caracteres diferentes se pueden representar utilizando el sistema numérico binario con 3 dígitos? ¿Y con 8?  
¿Qué números serían los más grandes que se pueden representar con ambos casos?
4. Pasar de binario a decimal las siguientes cantidades:  
 $(10010111)_2$  y  $(01110110)_2$

# Solución

24

1. Indica, sin pasar a decimal, cuál es el mayor de los dos:  
a)  $(01001000)_2$  **MAYOR**  
b)  $(01000010)_2$
2. Expresa, en código binario, los números  $(55)_{10}$  y  $(205)_{10}$ .  
 **$(110111)_2$  y  $(11001101)_2$**
3. ¿Cuántos caracteres diferentes se pueden representar utilizando el sistema numérico binario con 3 dígitos? ¿Y con 8?  
¿Qué números serían los más grandes que se pueden representar en ambos casos?  
**8 y 256 caracteres. 7 y 255**
4. Pasar de binario a decimal las siguientes cantidades:  
 $(10010111)_2$  y  $(01110110)_2$   
 **$(151)_{10}$  y  $(118)_{10}$**



# Conversión de decimal a binario de una fracción

NO ENTRA

25

Para convertir una fracción decimal a binario, esta fracción debe ser **multiplicada x 2**, y tomamos la parte entera del resultado, repetimos el proceso con la parte fraccionaria del resultado anterior, dándonos una nueva parte entera, y así sucesivamente hasta que la parte fraccionaria se haga cero o tengamos suficientes decimales **por debajo de un determinado error**.

**Ej. : Convertir el número 0,90625 a fracción binaria**

$$0,90625 \times 2 = 1,8125$$

$$0,8125 \times 2 = 1,625$$

$$0,625 \times 2 = 1,25$$

$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,$$

$$(0,90625)_{10} = (0,11101)_2$$

# Conversión de decimal a binario de una fracción, con error mínimo

NO ENTRA

26

Convertimos  $(0,6)_{10}$  a base 2, con un error inferior a  $2^{-7}$

$$0,6 \times 2 = 1,2 \rightarrow 1$$

$$0,2 \times 2 = 0,4 \rightarrow 0$$

$$0,4 \times 2 = 0,8 \rightarrow 0$$

$$0,8 \times 2 = 1,6 \rightarrow 1$$

$$0,6 \times 2 = 1,2 \rightarrow 1$$

$$0,2 \times 2 = 0,4 \rightarrow 0$$

$$0,4 \times 2 = 0,8 \rightarrow 0$$



posición de los dígitos a la dcha.  
de la coma en binario

Resultado:

$(0,6)_{10} = (0,1001100)_2$  con error  $2^{-7}$  en la conversión

# Conversión de una fracción decimal a binario

NO ENTRA

27

- Si se desea convertir un número que tiene parte entera y decimal a binario, se deberá operar cada parte **por separado**, y luego obtener la suma de los resultados.

Por ejemplo:

$$(12,125)_{10} = (1100,001)_2 \quad \text{¡COMPRUEBALO!}$$

$$\text{Parte entera por pesos: } 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 12$$

$$(1100)_2$$

Parte fraccionaria:

$$0,125 \times 2 = 0,250 \rightarrow 0$$

$$0,250 \times 2 = 0,500 \rightarrow 0$$

$$0,500 \times 2 = 1,000 \rightarrow 1$$

$$(0,001)_2$$

# Conversión de una fracción binaria a decimal

NO ENTRA

28

- Para convertir una fracción binaria a decimal, basta con sumar los pesos donde corresponda un 1 del binario a la derecha de la coma.

Pesos asociados								Nº decimal
$2^3$	$2^2$	$2^1$	$2^0$	,	$2^{-1}$	$2^{-2}$	$2^{-3}$	
8	4	2	1	,	0,5	0,25	0,125	
				,	0	1		0,25
				,	1	0	1	0,625
				,	1			0,5
				,	1	1		0,75

# ¿Cuántos bits usar?

- **Bit (binary digit):** Unidad elemental de informática (puede ser 0 o 1) su símbolo **b minúscula**.
- Si sumamos todos los símbolos que conocemos: Números + letras + caracteres raros + signos de puntuación... aprox. 150
- Con **7 bits = 128 estados**. ¡Nos quedamos cortos!
- Con **8 bits = 256 estados**. OK!
- **Byte: agrupación de 8 bits**. Llamado también **OCTETO**, su símbolo **B mayúscula**.

# Byte y bit

- ❑ ¡¡¡NO es lo mismo KB que Kb!!!
- ❑ Recuerda no confundir los bits (b) con los bytes (B).
- ❑ 1 byte = 8 bits = número binario de 8 dígitos.

# Bit y Byte

31

- ❑ **Múltiplos del Bit**
- ❑ Puedo representar  **$2^8 = 256$**  combinaciones posibles.
- ❑ Los ordenadores actuales trabajan siempre con agrupaciones de 1, 2, 4 y 8 bytes, es decir, con bloques de 8, 16, 32 y 64 bits, pero siempre múltiplos de 8 bits.

Unidad	Símbolo	Equivalencias		
1 byte	B	8 bits		
1 Kilobyte	KB	$2^{10}$ bytes	1024 bytes	
1 Megabyte	MB	$2^{20}$ bytes	$1024^2$ bytes	1024 KB
1 Gigabyte	GB	$2^{30}$ bytes	$1024^3$ bytes	1024 MB
1 Terabyte	TB	$2^{40}$ bytes	$1024^4$ bytes	1024 GB
1 Petabyte	PB	$2^{50}$ bytes	$1024^5$ bytes	1024 TB

# ¿1 GB = $10^9$ bytes?

32

## □ Tarea:

- **En parejas** averiguar porqué la capacidad del disco duro en GB es menor al número de bytes que aparece en Windows y dejad vuestra respuesta en un documento de Word, con nombre ***TUS\_APELLIDOS\_GB***, y déjalo en la carpeta.
- Si ya sabes el por qué de antemano escríbelo individualmente y déjalo en la misma carpeta.



NACHO\_8GB

Tipo: Disco extraíble

Sistema de archivos: NTFS

■ Espacio usado:	2,330,738,688 bytes	2.16 GB
■ Espacio disponible:	5,409,124,352 bytes	5.03 GB

Capacidad: 7,739,863,040 bytes 7.20 GB



Unidad F:



# ¿1 GB = $10^9$ bytes?

33

Un Gigabyte (1 GB) significa  $10^9 = 1.000.000.000$  bytes usando potencias de 10.

Sin embargo, un sistema operativo muestra la capacidad de almacenamiento usando potencias de 2 para la definición de  $1 \text{ GB} = 2^{30} = 1.073.741.824$  bytes, y por lo tanto muestra menos capacidad de almacenamiento.

Entonces, viendo las propiedades de un disco duro de 160 GB, el sistema mostrará algo como 160.039.239.680 bytes que vienen siendo aproximadamente 149 GB ( $160.039.239.680 \text{ bytes} / 1024 / 1024 / 1024$ ), para uno de supuestamente 320 GB el sistema mostrará 320.070.287.360 bytes, que son 298 GB efectivos y para uno disco de 500 GB (499.983.122.432 bytes) en realidad 465 GB estaría disponibles.

En resumen, la diferencia está en que los vendedores de discos duros usan el Sistema Internacional (con base 10) y no potencias con base 2, que sería lo correcto.

Fuente: web de Toshiba.

# Gibibyte

- El **gibibyte** (apócope de *giga binary byte*, simbolizado **GiB**) es una unidad de información utilizada como un múltiplo del byte. 1 GiB equivale a  $2^{30}$  bytes = 1024 Mebibyte (MiB) = 1 073 741 824 bytes.

- No se debe confundir con el gigabyte (GB), pues a pesar de que están relacionados tienen valores diferentes:

$$1 \text{ GB} = 10^9 \text{ bytes} = 1\,000\,000\,000 \text{ bytes} \approx \mathbf{0,93 \text{ GiB}}$$

$$\mathbf{1 \text{ GiB}} = 2^{30} \text{ bytes} = 1\,073\,741\,824 \text{ bytes} \approx \mathbf{1,07 \text{ GB}}$$

- El uso de estos términos intenta disipar una confusión muy común en torno a los medios de almacenamiento. Los dos números están relativamente cercanos, pero el confundir uno con otro ha llevado ocasionalmente a problemas aún discutidos por la comunidad informática. Por ejemplo, las unidades de almacenamiento como discos duros, pendrives y DVD suelen expresar su capacidad en la unidad más pequeña (GB), mientras que Windows los expresa en **GiB** (a pesar de usar erróneamente el símbolo "GB") pareciendo que las unidades tuviesen menos de lo estipulado.
- Forma parte de la norma ISO/IEC 80000-13 (Desde febrero del año 1999).

# Gibibyte



NACHO\_8GB

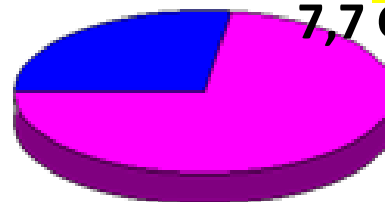
Tipo: Disco extraíble

Sistema de archivos: NTFS

Espacio usado: 2,330,738,688 bytes 2.16 GB

Espacio disponible: 5,409,124,352 bytes 5.03 GB

Capacidad: 7,739,863,040 bytes 7.20 GB



Unidad F:

GIBIBYTE

7,7 GB

7,2 GiB

# Unidades de Información.

## Gibibyte vs. Gigabyte.

36

Unidades de información (del byte)			
Sistema Internacional (decimal)		ISO/IEC 80000-13 (binario)	
Múltiplo (símbolo)	SI	Múltiplo (símbolo)	ISO/IEC
kilobyte (kB)	$10^3$	kibibyte (KiB)	$2^{10}$
megabyte (MB)	$10^6$	mebibyte (MiB)	$2^{20}$
gigabyte (GB)	$10^9$	<b>gibibyte (GiB)</b>	$2^{30}$
terabyte (TB)	$10^{12}$	tebibyte (TiB)	$2^{40}$
petabyte (PB)	$10^{15}$	pebibyte (PiB)	$2^{50}$
exabyte (EB)	$10^{18}$	exbibyte (EiB)	$2^{60}$
zettabyte (ZB)	$10^{21}$	zebibyte (ZiB)	$2^{70}$
yottabyte (YB)	$10^{24}$	yobibyte (YiB)	$2^{80}$

# Piensa...

37

- La dirección IP de un equipo está formada por **4 bytes** de la siguiente manera:

**BYTE 1º . BYTE 2º . BYTE 3º . BYTE 4º**

- Así, una IP puede ser expresada en *sistema decimal* por 4 números decimales, por ejemplo:

**192 . 168 . 0 . 1**

- Por tanto ¿Cuántos bits en *sistema binario* forman una IP?
- ¿Se te ocurre cuántas IPs puede haber disponibles en todo el mundo?

# Ejercicios

38

- Haciendo uso de los múltiplos del byte y la tabla de Unidades de Información (diapositivas 22 y 27) convierte las siguientes unidades:
  - 7GB = MB
  - $2.5 \cdot 10^6$ MB = GB
  - 200 MB = KB
  - 16038 bits = KB
  - 10 B = bits

# Sistemas octal y hexadecimal



# Sistemas octal y hexadecimal

40

- El inconveniente de la codificación binaria es que la representación de algunos números, especialmente en programación, resulta muy larga (largas cadenas de 0 y 1). Por ello se recurre a sistemas intermedios:
  - ▣ Sistema OCTAL
  - ▣ Sistema HEXADECIMAL
- La formas de conversión a decimal y viceversa son parecidas a las ya vistas.
- OCTAL utiliza (0,1,2,3,4,5,6,7)
- HEXADECIMAL utiliza (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

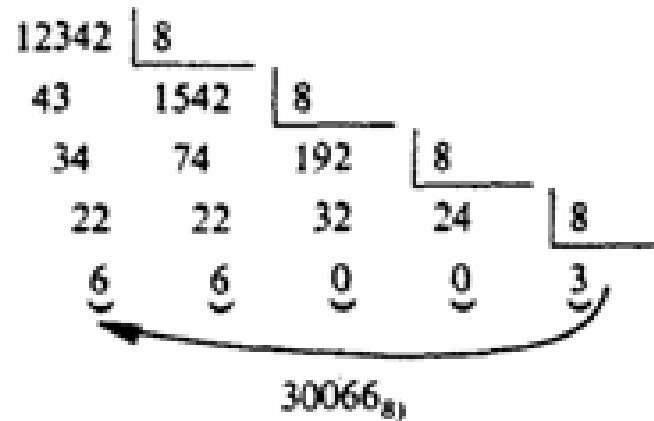


# Conversión decimal a octal

41

## □ *Parte entera decimal a octal:*

Lo más sencillo es realizar divisiones sucesivas entre 8



## □ *Fracciones decimales a octal:*

Igual que con el binario, mediante multiplicaciones sucesivas x 8.

$$¿ (12,0625)_{10} = ( \quad )_8 ?$$

# Conversión octal a decimal.

42

## □ *Parte entera octal a decimal:*

Se hará de forma similar a la explicada para convertir de binario a decimal. Pero se tendrá en cuenta que la **base** para los pesos ahora es **8**.

## □ *Fracciones octales a decimal:*

Aplicando el Teorema Fundamental de la numeración cada dígito tiene un peso según su posición a la dcha. de la coma.

$$(65,12)_8 = (6 \times 8^1) + (5 \times 8^0) + (1 \times 8^{-1}) + (2 \times 8^{-2}) = (53,375)_{10}$$

# Conversión octal a decimal

43

- Una opción práctica puede ser usando la tabla:

Pesos asociados al sistema OCTAL								
$8^4$	$8^3$	$8^2$	$8^1$	$8^0$		$8^{-1}$	$8^{-2}$	$8^{-3}$
4096	512	64	8	1	,	0,125	0,0156	$1,95 \cdot 10^{-3}$
		6	2	7	,	0	1	

$$(627,01)_8 = (6 \times 8^2) + (2 \times 8^1) + (7 \times 1) + (0 \times 8^{-1}) + (1 \times 8^{-2}) =$$

$$(6 \times 64) + (2 \times 8) + (7 \times 1) + (0) + (1 \times 0,0156) = (407,0156)_{10}$$

$$¿(716,0316)_8 = ( \quad )_{10} ?$$

# Sistema Hexadecimal

44

- Tiene como **base** de numeración 16, esto es, 16 símbolos para representar cantidades:  
**0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.**
- También es posicional asignando a los símbolos A, B, C, D, E, F los siguientes valores:

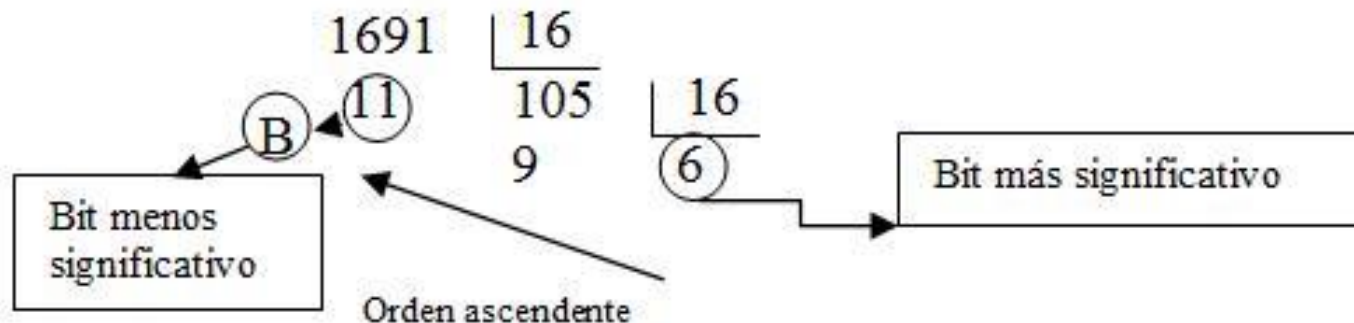
SÍMBOLO	VALOR
A	10
B	11
C	12
D	13
E	14
F	15

# Conversión decimal a hexadecimal

45

## □ *Parte entera decimal a hexadecimal:*

Lo más sencillo es realizar divisiones sucesivas entre 16:



El resultado en hexadecimal de  $1691_{10}$  es  $69B_{16}$ .

## ⊙ *Fracciones decimales a hexadecimal:*

Igual que en binario, mediante multiplicaciones sucesivas x 16.

¿  $(41565)_{10} = ( \quad )_{16}$  ?

# Conversión hexadecimal a decimal

46

## □ *Parte entera hexadecimal a decimal:*

Se hará en forma similar a la explicada para convertir de binario a decimal. Pero se tendrá en cuenta que la **base** para los pesos es **16**.

## □ *Fracciones hexadecimales a decimal:*

Aplicando el Teorema Fundamental de la numeración cada dígito tiene un peso según su posición a la dcha. de la coma.

$$\begin{aligned} (\mathbf{A1D})_{16} &= (\mathbf{A} \times 16^2) + (\mathbf{1} \times 16^1) + (\mathbf{D} \times 16^0) = \\ &= (\mathbf{10} \times 256) + (\mathbf{1} \times 16) + (\mathbf{13} \times 1) = (\mathbf{2589})_{10} \end{aligned}$$

# Conversión hexadecimal a decimal

47

- Una opción más práctica usando la tabla:

Pesos asociados al sistema HEXADECIMAL								
$16^4$	$16^3$	$16^2$	$16^1$	$16^0$		$16^{-1}$	$16^{-2}$	$16^{-3}$
65536	4096	256	16	1	,	0,0625	0,0039	$2,44 \cdot 10^{-4}$
		A	1	D	,	F	2	

$$(A1D,F2)_{16} = (A \times 16^2) + (1 \times 16^1) + (D \times 1) + (F \times 16^{-1}) + (2 \times 16^{-2}) =$$

$$(10 \times 256) + (1 \times 16) + (13 \times 1) + (15 \times 0,0625) + (2 \times 0,0039) = (2589,9453)_{10}$$

# Ejercicios

48

□ **Convierte a decimal:**

a)  $11,3016_8$

b)  $265,021_8$

c)  $A59,DC_{16}$

d)  $1F3B,01A_{16}$

□ **Expresa en octal:**

a)  $28,32_{10}$

b)  $459,901_{10}$

□ **Expresa en hexadecimal:**

a)  $1048576,32_{10}$

b)  $650,502_{10}$



# Ejercicios

49

- Abre un *terminal de comandos (cmd)* en Windows7, escribe ***ipconfig /all*** y obtén la **Dirección Física MAC**, luego pasa la MAC (que está en sistema hexadecimal) de tu equipo al sistema decimal.

La dirección MAC de tu tarjeta de red está formada por 6 números hexadecimales de 2 cifras cada uno, observa la imagen:

```
Adaptador de Ethernet Ethernet:
    Sufijo DNS específico para la conexión. . . :
    Descripción . . . . . : Realtek PCIe GBE Family Controller
    Dirección física. . . . . : D0-50-99-4D-89-C2
    DHCP habilitado . . . . . : sí
    Configuración automática habilitada . . . : sí
    Vínculo: dirección IPv6 local. . . : fe80::e982:aa1:9d9f:a654%3(Preferido)
    Dirección IPv4. . . . . : 10.0.100.39(Preferido)
    Máscara de subred . . . . . : 255.255.255.0
```

- ¿qué es la dirección física MAC de una tarjeta de red?

# Conversión octal a binario

50

- Al ser la base del octal (**8**) potencia de la base binaria (**2<sup>3</sup>**), la transformación de una base a la otra se hace en forma directa dígito a dígito. ***Cada dígito octal será reemplazado por 3 dígitos binarios*** (3 por ser la potencia que relaciona ambas bases), según la tabla que tenemos a continuación.

2	7	6,	5	3	4
010	111	110,	101	011	100

$$(276,534)_8 = (10111110,1010111)_2$$

# Conversión binario a octal

51

- Esta conversión es similar a la anterior, ***por cada tres símbolos binarios corresponde uno octal***. Para realizar correctamente esta conversión el número de dígitos a la derecha de la coma decimal debe ser múltiplo de 3, si no lo fuera deberá agregarse al final del número tantos ceros como sea necesario. Idéntico caso será a la izquierda de la coma, en dicho caso los ceros se agregan al principio del número.

010	101	011,	001	100
2	5	3,	1	4

$$(10101011,0011)_2 = (253,14)_8$$

# Conversión hexadecimal a binario

52

- Por idénticas razones que el caso anterior ( $16 = 2^4$ ), la transformación de una base a la otra se hace en forma directa dígito a dígito. Cada dígito hexadecimal será reemplazado por 4 dígitos binarios (cuarteto o *nibble*), según la siguiente tabla:

Hexadecimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Hexadecimal	Binario
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Ejemplo:  $(5A8,39C)_{16} = (10110101000,0011100111)_2$

5	A	8,	3	9	C
0101	1010	1000,	0011	1001	1100

# Conversión binario a hexadecimal

53

- Esta conversión es similar a la conversión a octal, pero en lugar de tres, serán cuatro símbolos binarios los que corresponde a un hexadecimal. Para realizar correctamente esta conversión el nº de dígitos a la dcha. de la coma decimal debe ser múltiplo de 4 si no lo fuera deberá agregarse al final del número tantos ceros como sea necesario. Idéntico caso será a la izquierda de la coma, en dicho caso los ceros se agregan al principio del número.

**Convertir el binario 1010101011,00111 a hexadecimal.**

0010	1010	1011,	0011	1000
2	A	B,	3	8

$$(1010101011,00111)_2 = (2AB,38)_{16}$$

# Conversiones hexadecimal a octal y octal a hexadecimal.

54

## □ *Hexadecimal a octal:*

Se realiza mediante un paso intermedio, primero se pasa de hexadecimal a binario y luego de binario a octal (también se puede usar como sistema intermedio el decimal).

## □ *Octal a hexadecimal:*

Como en el caso anterior, pero ahora primero se hace la conversión de octal a binario y luego de binario a hexadecimal (también se puede usar como sistema intermedio el decimal).

# Conversión entre sistemas

55

DEC	BIN	OCT	HEX
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

DEC	BIN	OCT	HEX
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C

# Ejercicios

56

## ☐ Convertir a binario:

a)  $AF32,00E_{16}$

b)  $77610,2755_8$

## ☐ Convertir a Hexadecimal:

a)  $703,16_8$

b)  $10011101001,00011_2$

c)  $1227,32_8$

## ☐ Convertir a octal:

a)  $1101010111101,1011101_2$

b)  $C127,B1_{16}$

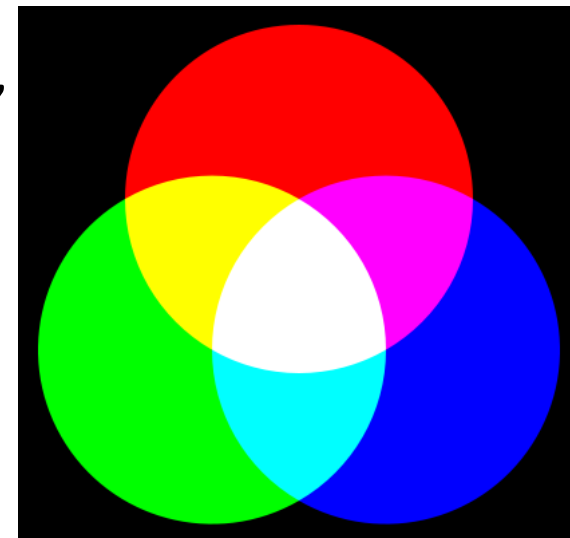
c)  $9A,53F2_{16}$



# Sistema de color RGB

57

- ❑ La codificación hexadecimal del color permite expresar fácilmente un color concreto de la escala RGB. Se utiliza, por ejemplo, en el *lenguaje HTML* y en *JavaScript*.
- ❑ Este sistema utiliza la combinación de 3 números hexadecimales de 2 dígitos cada uno para expresar las diferentes intensidades de los colores primarios RGB (*Red, Green, Blue*, rojo, verde y azul). El RGB también puede expresarse a veces como 3 números decimales cada uno correspondiendo a cada color primario.



Negro	#000000	Los tres canales están al mínimo 00, 00 y 00
Blanco	#FFFFFF	Los tres canales están al máximo FF, FF y FF

Color	Hexadecimal	Color	Hexadecimal	Color	Hexadecimal	Color	Hexadecimal
CYAN	#00FFFF	NEGRO	#000000	AZUL	#0000FF	MAGENTA	#FF00FF
GRIS	#808080	VERDE	#008000	LIMA	#00FF00	MARRÓN	#800000
AZUL MARI.	#000080	OLIVA	#808000	PÚRPURA	#800080	ROJO	#FF0000
PLATA	#C0C0C0	TURQUESA	#008080	BLANCO	#FFFFFF	AMARILLO	#FFFF00

# Ejercicios, Sistema RGB

58

- ❑ Obtener el **color magenta (combinación de rojo y azul)** en hexadecimal, y binario.
- ❑ Obtener los tres números **decimales** que corresponden al **plata**.
- ❑ Obtén en hexadecimal un tono de **gris oscuro**, razona el resultado.
- ❑ Cuantos tonos **sin color verde** hay **entre el azul y el rojo**.
- ❑ Abrir el programa **Paint** para ver cómo utiliza el sistema RGB, ¿cómo lo explicas?.

# Códigos alfanuméricos

! " # \$ % & ' ( ) \* + , - . /  
0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
@ A B C D E F G H I J K L M N O  
P Q R S T U V W X Y Z [ \ ] ^ \_  
` a b c d e f g h i j k l m n o  
p q r s t u v w x y z { | } ~

# Código ASCII

60

- **ASCII** (*American Standard Code for Information Interchange*)  
El código **ASCII** es el más utilizado, utiliza **7 bits por carácter**, permitiendo **128 diferentes caracteres**, suficiente para los signos del alfabeto en letras mayúsculas y minúsculas, y los símbolos de escribir más corrientes además de algunas combinaciones reservadas para su uso interno. El **código ASCII extendido** usa **8 bits por carácter** (256 símbolos en total). Este código más amplio permite que se agreguen los símbolos de lenguajes extranjeros y varios símbolos gráficos.

## 61

<div><div><div><div><div>b<sub>7</sub></div><div>b<sub>6</sub></div><div>b<sub>5</sub></div><div>b<sub>4</sub></div><div>b<sub>3</sub></div><div>b<sub>2</sub></div><div>b<sub>1</sub></div></div><div>Bits</div></div><div><div>Column →</div><div>Row ↓</div></div></div></div>						0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	0	0	0	0	0	0					
						0	1	2	3	4	5	6	7					
0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p					
0	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q					
0	0	0	1	0	2	STX	DC2	"	2	B	R	b	r					
0	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s					
0	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t					
0	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u					
0	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v					
0	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w					
0	1	0	0	0	8	BS	CAN	(	8	H	X	h	x					
0	1	0	0	1	9	HT	EM	)	9	I	Y	i	y					
0	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z					
0	1	0	1	1	11	VT	ESC	+	;	K	[	k	{					
0	1	1	0	0	12	FF	FC	,	<	L	\	l						
0	1	1	0	1	13	CR	GS	-	=	M	]	m	}					
0	1	1	1	0	14	SO	RS	.	>	N	^	n	~					
0	1	1	1	1	15	SI	US	/	?	O	_	o	DEL					

# Código *ASCII* original (7 bits)

62

## □ *American Standard Code for Information Interchange*

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

**128**  
símbolos u  
ordenes

Caracteres imprimibles  
del código ASCII (7 bits):

!"#\$%&'()\*+,-./  
0123456789:;<=>?  
@ABCDEFGHIJKLMNO  
PQRSTUVWXYZ[\]^\_  
`abcdefghijklmnopqrstuvwxyz  
{|}~

# ASCII extendido (8 bits) → *Latin-1* (ISO-8859-1)

63

- Es una norma de la ISO que define la codificación del **alfabeto latino**, incluyendo los diacríticos (como letras acentuadas, ñ, ç), y letras especiales (como ß, Ø), necesarios para la escritura de las siguientes lenguas originarias de Europa occidental

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	û	161	A1	í	193	C1	ł	225	E1	β
130	82	é	162	A2	ó	194	C2	ŧ	226	E2	Γ
131	83	â	163	A3	ú	195	C3	ţ	227	E3	π
132	84	ä	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	†	229	E5	σ
134	86	ã	166	A6	ª	198	C6	‡	230	E6	μ
135	87	ç	167	A7	º	199	C7	‡	231	E7	ι
136	88	ê	168	A8	¿	200	C8	Ł	232	E8	Φ
137	89	ë	169	A9	ƒ	201	C9	ƒ	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	Ł	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	Ł	235	EB	δ
140	8C	î	172	AC	¼	204	CC	Ł	236	EC	∞
141	8D	ì	173	AD	ı	205	CD	=	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	¼	238	EE	ε
143	8F	Å	175	AF	»	207	CF	½	239	EF	Π
144	90	É	176	B0	⋮	208	D0	Ł	240	F0	≡
145	91	æ	177	B1	⋮	209	D1	Ł	241	F1	±
146	92	Æ	178	B2	⋮	210	D2	Ł	242	F2	≥
147	93	ô	179	B3		211	D3	Ł	243	F3	≤
148	94	ö	180	B4	Ł	212	D4	Ö	244	F4	
149	95	ò	181	B5	Ł	213	D5	ƒ	245	F5	
150	96	û	182	B6	Ł	214	D6	ƒ	246	F6	÷
151	97	ù	183	B7	Ł	215	D7	Ł	247	F7	≈
152	98	ý	184	B8	Ł	216	D8	Ł	248	F8	≈
153	99	Û	185	B9	Ł	217	D9	Ł	249	F9	·
154	9A	Ü	186	BA		218	DA	Ł	250	FA	·
155	9B	φ	187	BB	Ł	219	DB	■	251	FB	√
156	9C	£	188	BC	Ł	220	DC	■	252	FC	°
157	9D	¥	189	BD	Ł	221	DD	■	253	FD	²
158	9E	Ps	190	BE	Ł	222	DE	■	254	FE	■
159	9F	f	191	BF	Ł	223	DF	■	255	FF	

# Código Unicode

- El **Unicode** usa **hasta 32 bits (4 bytes)** para representar cada carácter, por lo que para los mismos caracteres emplea cuatro veces el espacio que con el código ASCII extendido, pero puede manejar muchos más caracteres. La meta de Unicode es representar cada elemento usado en la escritura de cualquier idioma del planeta. La versión 3 de **Unicode** tiene **49,194 caracteres** en lugar de los escasos pocos centenares de caracteres para ASCII. Todos los idiomas actuales más importantes del mundo, pueden escribirse con Unicode, incluidas su puntuación y símbolos especiales para matemática y geometría.



# Unicode 7.0 (ISO 10646)

65

- Poco a poco es el código que se va extendiendo pero el dominio histórico que ha llevado el código ASCII complica su popularidad
- <http://www.unicode.org/charts/>

 **Code Charts** [Home](#) | [Site Map](#) | [Search](#)

## Unicode 7.0 Character Code Charts

[SCRIPTS](#) | [SYMBOLS](#) | [NOTES](#)

Find chart by hex code:   Related links: [Name index](#) [Help & links](#)

### Scripts

European Scripts	African Scripts	South Asian Scripts	Indonesia & Oceania Scripts
<a href="#">Armenian</a>	<a href="#">Bamum</a>	<a href="#">Bengali and Assamese</a>	<a href="#">Balinese</a>
<a href="#">Armenian Ligatures</a>	<a href="#">Bamum Supplement</a>	<a href="#">Brahmi</a>	<a href="#">Batak</a>
<a href="#">Caucasian Albanian</a>	<a href="#">Bassa Vah</a>	<a href="#">Chakma</a>	

## Ejemplos de caracteres Unicode



Carácter alfabético latino  
"A" (U+0041).



Sílaba devanagari "Aum"  
(ॐ) (U+0950).



Ideograma chino "yue"  
(月)(U+6708).

# Unicode en un SGBD: MySQL

66

MySQL 5.7 Reference Manual :: 10 Globalization :: 10.1 Character Set Support :: 10.1.10 Unicode Support :: 10.1.10.5 The utf8 Character Set (3-Byte UTF-8 Unicode Encoding)

## 10.1.10.5 The utf8 Character Set (3-Byte UTF-8 Unicode Encoding)

UTF-8 (Unicode Transformation Format with 8-bit units) is an alternative way to store Unicode data. It is implemented according to RFC 3629, which describes encoding sequences that take from one to four bytes. (An older standard for UTF-8 encoding, RFC 2279, describes UTF-8 sequences that take from one to six bytes. RFC 3629 renders RFC 2279 obsolete; for this reason, sequences with five and six bytes are no longer used.)

The idea of UTF-8 is that various Unicode characters are encoded using byte sequences of different lengths:

« 10.1.10.4 The utf32 Character Set (UTF-32 Unicode Encoding)

10.1.10.6 The utf8mb3 Character Set (Alias for utf8) »

### Section Navigation [\[Toggle\]](#)

- 10.1.10 Unicode Support
- 10.1.10.1 The ucs2 Character Set (UCS-2 Unicode Encoding)
- 10.1.10.2 The utf16 Character Set

**Tip:** To save space with UTF-8, use `VARCHAR` instead of `CHAR`. Otherwise, MySQL must reserve three bytes for each character in a `CHAR CHARACTER SET utf8` column because that is the maximum possible length. For example, MySQL must reserve 30 bytes for a `CHAR(10) CHARACTER SET utf8` column.

¿Cuántos caracteres diferentes podrá  
representar este código Unicode?

# Ejercicio.

67

- **Calcula el código binario de tu nombre según el código ASCII.**

***“Solo hay 10 tipos de personas: las que saben informática y las que no”***