

# Introducción a la representación de la Información

## Contenidos:

1. Tipos de Información Elementales.
  1. Variables.
  2. Señales.
2. Codificación de la información en un computador digital: por qué tenemos que emplear la codificación binaria de la información
3. Técnicas Básicas de Codificación Binaria de la Información.
  1. Codificación directa.
  2. Codificación por campos.
  3. Secuencias de Códigos: Códigos de Control
4. Bits, bytes y sus múltiplos.
  1. Múltiplos.
  2. Diferencias de terminología entre el Mundo de la Informática y el Mundo de las Comunicaciones.
  3. Submúltiplos del Segundo (Tiempo).

## 1. Tipos de Información Elementales

*En este apartado el objetivo será comprender qué tipos de variables te puedes encontrar en tu trabajo. De nuevo, recuerda que las variables representan la información que te interesará sobre lo que estés trabajando. Con esa información podrás realizar las operaciones que sean necesarias.*

- Supongamos que estamos observando un determinado fenómeno o tema de estudio (ej.: los alumnos en el aula). Para cada **variable** (propiedad o característica) que nos interese del fenómeno siempre tendremos que tener en cuenta:
  - a) **Tipo** de la variable: determinará el **conjunto de valores posibles** que puede adoptar o tomar esa variable.

### Ejemplo 1:

- Variable: número de alumnos en el aula
- Tipo: número natural.
  - ~ conjunto de valores posibles:  
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

- b) **Señal**: el valor que toma la variable puede variar dependiendo de otra variable (el tiempo, la frecuencia, etc.)

### Ejemplo 2:

- Variable: número de alumnos en el aula.
  - Esta variable tendrá un único valor en un momento del día dado (ej. 14 alumnos) pero puede ir variando a lo largo de la mañana (cambio de turno) o a lo largo de una clase (un alumno sale al servicio, un alumno llega tarde).
  - A cada segundo de la mañana se le puede asociar el número de alumnos que hay en el aula IF4:
    - segundo 1: 2 alumnos
    - segundo 2: 2 alumnos
    - segundo 3: 3 alumnos
    - segundo 4: 4 alumnos
    - ...
    - segundo 20: 15 alumnos
    - ...
- Esto representa el concepto de **señal** la variación de una variable dependiendo de otra variable que, en este caso, es la variable tiempo.

## 1.1. Variables

- Una **variable** es una **propiedad** o **atributo** de un objeto físico o abstracto que puede ser cuantificada (= medida) o calificada obteniéndose así su **valor**.

Es muy importante que se cumplan estas útiles condiciones:

- a) Una variable toma sus valores de entre un **conjunto de valores posibles**.

Este conjunto debe estar bien definido (sin ambigüedades): es el **conjunto de definición** de la variable o el **tipo** de la variable.

- b) Los valores que puede adoptar la variable deben ser:

- ~ **incompatibles**: la variable no puede adoptar dos valores al mismo tiempo.

- ~ **exhaustivos**: todo valor que pueda tomar la variable debe estar dentro de ese conjunto de posibles valores (es decir, no nos podemos olvidar ninguno).

Esta propiedad es relativa y depende del fenómeno que se esté estudiando.

Por ejemplo, si la variable de estudio es el "color de pelo", los posibles conjuntos exhaustivos serán diferentes según el conjunto de individuos que consideremos (a veces denominado el **universo del discurso**), por ejemplo:

- ~ alumnos de una clase: {castaño, rubio, negro}

- ~ alumnos de un instituto: {castaño, rubio, negro, pelirrojo, "sin pelo"}

- ~ alumnos de una región:

- {castaño, rubio, negro, pelirrojo, "sin pelo", "rojo", "verde"}

**Es decir, la variable siempre presenta en cada momento dado un valor, y sólo uno, de entre todos los posibles valores del conjunto de definición.**

- Todos los tipos de variables caen dentro de una de las siguientes categorías:

### a) Variables cualitativas

- Son aquellas que cumplen:

- 1º) Los valores de su tipo se pueden **enumerar** (listar, dar una lista).

- 2º) No se pueden ordenar sus valores unos con respecto a otros.

Esto es así porque, debido a su propia naturaleza, no tiene sentido plantearse un orden.

- 3º) No se puede **operar aritméticamente** (adición, substracción, multiplicación, división) con sus valores.

- Por lo tanto, las únicas **operaciones primitivas** (fundamentales) a realizar con estas variables son:

- a) **Comprobación de pertenencia** de un valor a un tipo.

- b) **Comprobación de igualdad/desigualdad** entre dos valores.

- Ejemplos:

- ~ variable color del pelo:

- {negro, castaño, pelirrojo, rubio, verde}

- ~ variable droga usada ilegalmente por ciudadano:

- {hachís, barbitúrico, anfetamina, LSD, morfina, heroína, cocaína, marihuana, mescalina}

- En los lenguajes de alto nivel, suelen representarse mediante un tipo de dato que recibe el nombre de **enumeración**.

### b) Variables ordinales

- Una variable pertenece a esta clase si cumple:

- 1º) Los valores de su tipo se pueden **enumerar** (listar, dar una lista).

- 2º) Se puede establecer un **orden** (primero va este, luego este otro, ...) entre los valores.

- 3º) No se puede operar aritméticamente (adición, substracción, multiplicación, división) con sus valores.

- Así pues, las únicas operaciones fundamentales (a partir de las cuales se pueden construir todas las demás) a realizar con estas variables son:

- a) **Comprobación de pertenencia** de un valor a un tipo.

- b) **Comprobación de igualdad/desigualdad** ( = , ≠ ) entre dos valores.

- c) **Comprobación de relación de orden** entre dos valores (≤).

- Ejemplos:

- ~ variable interés de un ciudadano en *Ropax*, detergente de última generación:

- {Mucho, Algo, Poco, Nada}

- ~ satisfacción del ciudadano con su gobierno:

- {Total, Mucha, Moderada, Media, Baja, Ninguna}

### c) Variables cuantitativas (magnitudes)

- Por último, una variable que sea de tipo cuantitativo debe cumplir:

- 1º) Los valores de su tipo se pueden denotar de alguna forma (bien por extensión o por comprensión).

- ~ ej. de denotar por extensión:  $\{1, 2, 3, 4, 5, 6\}$
- ~ ej. de denotar por comprensión:  $\{n \mid 1 \leq n \leq 2324234\}$
- 2º) Se pueden ordenar sus valores unos con respecto a otros.
- 3º) Se puede operar aritméticamente (adición, sustracción, multiplicación, división) con sus valores.
- Las únicas operaciones fundamentales a realizar con estas variables son:
  - a) Comprobación de pertenencia de un valor a un tipo.
  - b) Comprobación de igualdad/desigualdad ( $=$ ,  $\neq$ ) entre dos valores.
  - c) Comprobación de relación de orden entre dos valores ( $\leq$ ).
  - d) **Operaciones aritméticas** fundamentales.
- Dentro de las variables cuantitativas te encontrarás, además, con 2 tipos distintos:
  - c.1) **Variables cuantitativas discretas (variables digitales)**
    - Estas variables toman únicamente valores numéricos discretos (es decir, no hay números con infinitos decimales: 3,143214... ).  
Expresado de una forma más formal (es decir, precisa): los valores de estas variables pertenecen a un conjunto de valores representable por un subconjunto de los **números enteros**.
    - Ejemplos:
      - ~ variable número de pelos en la cabeza de una persona.
      - ~ variable cantidad de dinero de Pablo Rodríguez García en un momento dado.
      - ~ variable número que sale al tirar un dado.
    - La denominación “digital” procede del término dígito o cifra. Esto se debe a que las magnitudes digitales pueden representarse **de forma exacta** mediante un número finito de cifras o guarismos.
  - c.2) **Variables cuantitativas continuas (variables analógicas, variables continuas)**
    - Informalmente, una variable pertenece a este tipo si puede presentar valores numéricos con infinitas cifras decimales.  
(Ej.  $\pi = 3.14159265\dots$ ).
    - Más formalmente, una variable es continua si puede presentar valores representables por un subconjunto de los números reales.
    - Ejemplos:
      - ~ variable altura de una persona: podemos medir a una persona y obtener 1,81 metros. Pero si conseguimos un instrumento de medida todavía más preciso, podríamos obtener 1,812343m. Con otro instrumento más preciso podríamos obtener 1,812343879383477 metros. Y así sucesivamente.<sup>1</sup>
      - ~ variable tiempo de respuesta a un estímulo auditivo: 1,31 segundos, 1,31332 segundos, 1,31332324323 segundos, 1,31332324323423423424234234234234 segundos.
      - ~ variable intensidad de luz en una sala de estar.
    - Las variables analógicas, al contrario de las digitales, pueden no poder representarse de forma exacta con un número finito de cifras: sólo pueden representarse **de manera aproximada**.  
Lo que sí podemos es aumentar arbitrariamente la **precisión de la representación** a costa de utilizar más y más cifras.

## 1.2. Señales

- Pero he aquí que las cosas del mundo están cambiando y la información que poseemos de ellas debe cambiar para adaptarse a la nueva situación. Sabes bien que el valor que toma una variable bajo estudio en un momento dado debe ser único. Pero puede cambiar a otro valor que pasará a ser único.  
Este concepto de variable que cambia de valor y que además lo hace en función de otra variable (llamada **variable independiente**) es a lo que hace referencia el importantísimo concepto de **señal**.

<sup>1</sup> Por cierto, una cosa es la altura de una persona y otra muy distinta es el número que empleamos para representarla 1,81 y la unidad de medida: el metro.

### Ejemplo 3:

- Variable: dinero que tengo.
- Esta variable tendrá un único valor en un momento dado (ej. 1000 €) pero puede ir variando a lo largo del día (ej.: compro algo, me dan dinero, etc).
- Podemos listar cuánto dinero tengo en cada minuto del día, empezando a contar desde las 08:00:
  - minuto 1: 1000 €
  - minuto 2: 1000 €
  - minuto 3: 1000 €
  - minuto 4: 1000 €
  - minuto 5: 100 € (compré algo en el quiosco)
  - minuto 6: 100 €
  - minuto 7: 100 €
  - minuto 8: 100 €
  - ...
  - minuto 60: 1000000 € (me tocó el cupón de la ONCE)
  - ..
- También podemos aumentar la resolución de la medición y medir el avance de la variable tiempo (variable independiente) en segundos ("deshacer los grumos"):
  - segundo 1: 1000 €
  - segundo 2: 1000 €
  - ....
  - segundo 240: 1000 €
  - segundo 241: 900 € (compré un bolígrafo)
  - segundo 242: 900 €
  - segundo 243: 900 €
  - segundo 244: 800 € (compré un paquete chicles)
  - ....
  - ....
  - segundo 3600: 100 €
  - segundo 3601: 100 €
  - segundo 3602: 100 €
  - segundo 3603: 1000000 €
  - ..
- Decimos que la variable tiempo es una **variable independiente** porque, para nuestro estudio, no depende de nada. La variable dinero que tengo sí depende del momento de la mañana, representado por un número de segundos transcurridos desde las 08:00.

- En el sentido más abstracto (es decir, más general) la palabra señal significa el proceso de representar en un **formato** dado la información que cambia. Esta definición incluye tanto las formas más primitivas de transmisión de la información, tales como las señales de humo utilizadas por los indios norteamericanos o el sonido del tam-tam de algunas tribus de África, como las formas actuales de comunicación más sofisticadas.
- Para ti, como futuro informático (o como actual informático) el término señal puede significar 2 cosas:
  - a) La evolución de una variable de interés de un fenómeno bajo estudio que cambia en función de otra.
  - b) Una serie de medidas u observaciones del fenómeno bajo estudio que permiten asociar a cada valor de interés de la variable independiente un valor de la variable dependiente: la representación, en un lenguaje humano, del fenómeno.  
 Estas medidas u observaciones las representaremos en un formato dado (una lista de pares de valores, un gráfico, etc.)
- Además, para que podamos entendernos los informáticos entre nosotros y con los matemáticos, etc. debemos añadir algunas definiciones adicionales que serán extremadamente útiles después. Serán nuestra "jerga" particular:
  - ~ nos solemos referir a las señales, para abreviar, con símbolos como:  $v$ ,  $t$ ,  $x$ ,  $y$ ,  $s$ , ... (ejs.:  $t$  de tiempo,  $s$  de señal,  $v$  de voltaje eléctrico,  $i$  de intensidad eléctrica).
  - ~ Así pues, por ejemplo,  $x$  puede significar, según lo que estemos estudiando:
    - ~ los valores que toma el voltaje de una onda de radio en un intervalo particular de tiempo (la variable dependiente voltaje depende de la variable tiempo).
    - ~ la historia a lo largo del tiempo de un proceso económico.
    - ~ el cambio en la temperatura ambiente a medida que escalamos una montaña (la variable dependiente temperatura ambiente depende de la variable independiente altura).

~ el color de pelo de una persona, en función del champú usado.

En estos ejemplos, se habla de la evolución de ciertas variables (voltaje, magnitud económica, temperatura) en función de otras variables que se consideran independientes (tiempo, altura, tipo champú):

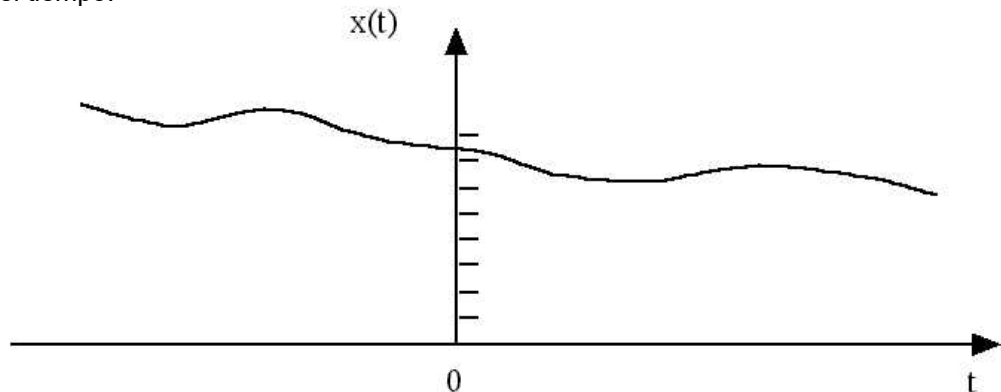
- ~ por lo tanto, el concepto de señal implica siempre la dependencia entre dos variables o magnitudes.
- ~ las señales que aparecen con más frecuencia son las que tienen al tiempo como variable independiente y se las denomina **señales dependientes del tiempo**.
- ~ dado que una señal es, simplemente, una función de una variable, el valor en el instante o momento  $t$  de una señal que depende del tiempo se representará como  $x(t)$ .

Hay que tener cuidado con no confundir el símbolo  $x$ , que indica la historia de la variable  $x$  en el tiempo, con el símbolo  $x(t)$  que especifica el valor de la señal en el instante  $t$ .

- Si ahora centramos nuestro estudio en las señales dependientes del tiempo, podremos distinguir 2 clases de señales muy importantes en la informática:

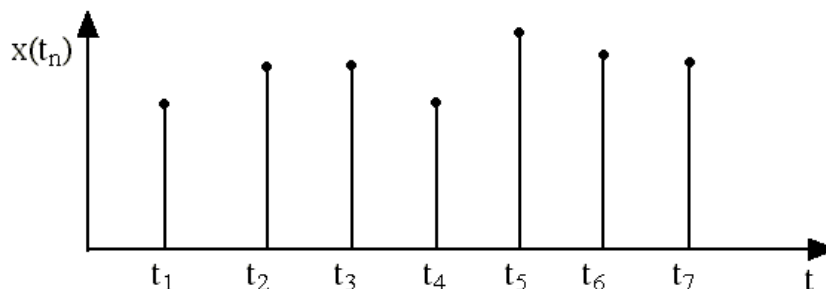
a) **Señales analógicas** (*señales continuas en el tiempo*)

- El tiempo se considera como una **variable analógica**: el valor de la señal puede cambiar en cualquier momento del tiempo.  
Su nombre pretende reflejar la dependencia continua de la señal con respecto al tiempo.
- El ejemplo siguiente representa en un formato muy acostumbrado la variación de la temperatura de una habitación a medida que pasa el tiempo:



b) **Señales discretas** (*señales discretas/digitales en el tiempo*)

- En este caso, la variable dependiente sólo puede cambiar en instantes específicos del tiempo, o bien, la señal sólo está definida en esos instantes.
- Puedes ver ahora un ejemplo que muestra la evolución de la variable número de personas en una sala de espera, a lo largo del período laboral:



## 2. Codificación de la información en un computador digital: por qué tenemos que emplear la codificación binaria de la información

*En este apartado, el objetivo es llegar a comprender por qué para representar cualquier tipo de información (datos + instrucciones de transformación de datos) los ordenadores sólo emplean dos símbolos: 0 y 1. Los humanos empleamos montones de símbolos: a, b, c, ..., 0, 1, 2, ..., integral, diferencial, ... .*

- Los **computadores digitales** actuales se basan en una tecnología electrónica que permite representar los datos y las acciones a realizar sobre ellos mediante combinaciones de **circuitos de tipo flip-flop** o **biestables**:
  - ~ estos elementos básicos sólo admiten dos estados, representados por el nivel de tensión (o corriente eléctrica, a veces) a su salida.
  - ~ este tipo de información, que se representa mediante la combinación de elementos que sólo admiten dos estados, se denomina **información binaria**.
  - ~ cada uno de los **elementos de representación de la información binaria** recibe el nombre de **bit** (binary digit) y se codifica mediante el empleo de uno de los dos únicos símbolos **0** o **1**: **cualquier dato que se deba procesar en un computador digital deberá estar representado por una secuencia de ceros y unos**.
- Así pues, los computadores digitales sólo manejan **magnitudes digitales**, que:
  - ~ se representan por un número finito de dígitos.
  - ~ son discretas en el tiempo: no cambian continuamente sino que se mantienen constantes durante intervalos de tiempo (cambian “a saltos”).

**El hecho de que la información que procesan estos computadores presente estas características ha obligado a desarrollar una serie de técnicas específicas para la representación de las instrucciones y los datos. En el apartado siguiente estudiaremos algunas de ellas.**
- En todo computador digital existen, al menos 2 **flujos de información**:
  - a) El **flujo de datos**; los datos que han de ser manipulados para producir los resultados deseados. Los datos pueden ser de naturaleza diversa; por ejemplo:
    - ~ **datos numéricos**: de tipo natural, entero, real.
    - ~ **datos alfabéticos**: las letras del alfabeto, caracteres de control, etc.
  - b) El **flujo de control**, que expresa las manipulaciones a realizar con dichos datos. Se compone de las órdenes (instrucciones) que debe ejecutar el computador, y que reciben el nombre de **códigos de instrucción**.

**«La naturaleza de la información (numérica, alfabética o código de instrucción) influye significativamente sobre el sistema de codificación elegido para su representación.»**

### 3. Principales Sistemas de Codificación Binaria de la Información

*En este apartado, el objetivo será que aprendas las técnicas fundamentales para generar códigos con sólo 2 símbolos (**códigos binarios**) con el objetivo de representar distintos tipos de datos.*

- La codificación binaria consiste en establecer unas reglas que definan una aplicación<sup>2</sup> entre cada **elemento de información (dato)** y la secuencia de bits que constituye su **código**:
  - ~ existen varios métodos genéricos para establecer esta aplicación, que dan lugar a tipos diferentes de códigos: estos criterios se denominan **sistemas de codificación**.
  - ~ dicho de otra forma, debe existir un código binario y uno sólo para cada significado (información) que quieras codificar (y el método de codificación establece una aplicación entre una "tabla semántica" y una "tabla de códigos")
  - ~ existen 3 sistemas básicos de codificación: los estudiarás ahora por orden creciente de complejidad.

#### 1. Codificación Directa

- La **codificación directa** es el sistema de codificación más simple de todos: consiste en establecer una **correspondencia biyectiva (aplicación)** entre un conjunto de informaciones y un conjunto de códigos binarios, por ejemplo, mediante una **tabla**:
  - ~ el número mínimo de bits,  $N_{\min}$ , a emplear en la codificación viene dado por la expresión:  

$$N_{\min} = \lceil \log_2 (\text{número de símbolos a codificar}) \rceil$$
 Puede ser que se desperdicien algunos códigos binarios posibles con  $N_{\min}$  bits disponibles si el número de símbolos a codificar no es potencia exacta de 2.
  - ~ por supuesto, pueden utilizarse códigos binarios con un número de bits superior al estrictamente necesario.
- Ejemplo:

Símbolo	Código	
A	000	
B	001	ABBA = 000 001 001 000
C	010	= 000001001000
D	011	
E	100	

- Esta estrategia de codificación se emplea día a día en: el código ASCII, los códigos numéricos de enteros, etc.

#### 2. Codificación por Campos

- En la **codificación por campos** los códigos binarios que codifican los elementos de información se componen de **campos de bits** de longitud fija, donde cada campo codifica un atributo del elemento de información particular a codificar dando lugar a un código parcial:
  - ~ el código binario final se forma escogiendo los códigos parciales adecuados para cada campo y juntando los códigos parciales de todos los campos en un orden dado.
  - ~ la codificación de cada campo, en la que se asigna a cada atributo un código binario dado, se realiza, por ejemplo, empleando codificación directa.

<sup>2</sup> Una aplicación es una asociación entre los elementos de dos conjuntos, por ejemplo  $A = \{\text{alumnos de 1º ASL}\}$  y  $B = \{\text{nombres de los alumnos de 1º ASL}\}$ . Esta asociación, asigna a cada elemento del conjunto A un elemento, y sólo uno, del conjunto B. Expresado de una forma más precisa: una aplicación (función) es un subconjunto, S, del producto cartesiano de 2 conjuntos A y B ( $S \subseteq A \times B$ ). Este subconjunto S verifica una importante propiedad: cada elemento de A aparece en un par de S y en sólo uno.



Letra	Campo 1	Tamaño	Campo 2	Símbolo	Código
A	000	Mayúscula	0	A	0000
B	001	Minúscula	1	a	0001
C	010			B	0010
D	011			b	0011
E	100			C	0100
				c	0101
				D	0110
				d	0111
				E	1000
				e	1001

campo 1    campo 2

▲            ▲

b = 0 0 1 1

▼

código

- En el ejemplo anterior se observa:
  - ~ cada campo se codifica por una tabla independiente
  - ~ la suma de las longitudes de estas tablas es inferior a la de una tabla directa única: mientras que en la codificación directa hubiera sido necesario formar una única tabla de longitud 5x2, la suma de las dos tablas parciales formadas en la codificación por campos es 5 + 2 = 7.
 Esta longitud total es inferior en tres elementos a la correspondiente a la codificación directa, y dicha diferencia va aumentando enormemente al ir creciendo el número de campos y/o el de bits por campo: en general, si tenemos los campos  $c_1, c_2, \dots, c_n$ , con  $b_1, b_2, \dots, b_n$  bits cada uno, tendremos:
  - codificación directa:  $2^{(b_1+b_2+\dots+b_n)}$  códigos directos
  - codificación por campos:  $2^{b_1} + 2^{b_2} + \dots + 2^{b_n}$  códigos parciales
 Por ejemplo: para códigos de tres campos de 5, 2 y 3 bits cada uno se tendría:
  - codificación directa:  $2^{(5+2+3)} = 1024$  códigos directos
  - codificación por campos:  $2^5 + 2^2 + 2^3 = 44$  códigos parciales.
- La codificación por campos resulta habitualmente más sencilla que la codificación directa: no obstante, esta última presenta en contrapartida una total libertad en la asignación de códigos, lo que permite en muchos casos facilitar las operaciones de proceso de la información codificada.

### 3. Secuencias de Códigos: Códigos de Control

- Con frecuencia, los elementos de información no se procesan o almacenan aisladamente sino en conjunto (todos juntos seguidos, formando lo que muchas veces se denomina **flujo** o **stream**): en estos casos los códigos de los sucesivos datos suelen ser procesados o registrados **secuencialmente**:
  - ~ el **tratamiento secuencial** de los códigos abre una nueva posibilidad consistente en hacer que determinados códigos especiales (**códigos de control** o **códigos de escape**) modifiquen la interpretación de los que aparezcan a continuación.
  - ~ ejemplo:

TABLA NORMAL		TABLA ESPECIAL	
Letra	Campo 1	Símbolo	Código
A	000	A'	000
B	001	E'	100
C	010		
D	011		
E	100		
usar tabla especial	101	BEBE' = 001 100 001 101 100	

Aparecen las letras A, B, C, D, E y además los símbolos especiales (letras) A', B':

- ~ si la aparición de los símbolos A', B' es poco frecuente, se les puede agrupar en una combinación única, usando un código adicional a continuación para indicar de qué símbolo especial se trata;
- ~ las letras acentuadas se codifican por tanto mediante una secuencia de dos códigos:
  - ~ el **código de control**, que indica que no es una letra normal: afecta a la codificación del elemento siguiente en secuencia, para el que no se usa la tabla de códigos normales sino la tabla especial.
  - ~ el código correspondiente a la letra especial.

- Este sistema de codificación consigue reducir la longitud de las tablas de codificación a costa de aumentar el código



asignado a ciertos símbolos:

- ~ si estos símbolos especiales aparecieran entre los elementos de información de los datos con una frecuencia similar a los símbolos normales, este sistema de codificación representaría los elementos de información con códigos excesivamente largos, ocupando en memoria más espacio que los códigos obtenidos mediante el empleo de los otros sistemas de codificación;
  - ~ por lo tanto, la condición de que los símbolos que se representan mediante los códigos de control deban de aparecer con relativamente poca frecuencia constituye un criterio fundamental para decidir emplear la codificación mediante este sistema.
- En el caso de la codificación por campos, los sucesivos campos de un código pueden ser considerados como una secuencia de códigos parciales, pudiendo por tanto un campo afectar a la codificación de los siguientes, variando su longitud o añadiendo o suprimiendo otros.

Ejemplo:

- ~ el primer campo, además de identificar la letra, actúa como código de control indicando si existe o no el campo 3.
- ~ los códigos no tienen ahora la misma longitud, según se pone de manifiesto en la tabla equivalente de codificación directa.

existe				
Letra	Campo 3	Campo 1	Símbolo	Código
A	sí	000	A	00000
B	no	001	A'	00001
C	no	010	a	00010
D	no	011	a'	00011
E	sí	100	B	0010
			b	0011
			C	0100
			c	0101
			D	0110
			d	0111
			E	10000
			E'	10001
			e	10010
			e'	10011

		Acento	Campo 3
		no	0
		sí	1

Tamaño	Campo 2
Mayúscula	0
Minúscula	1

## 4. Bits, bytes y sus múltiplos

Tenemos ahora claro que **bit** (*Binary digiT*) significa dígito binario, normalmente "0" y "1".

- es el abecedario de los ordenadores.
- cualquier tipo de dato sea este un libro, una foto, una película, un sonido los ordenadores lo guardan como una secuencia de 1s y 0s: es su lengua.
- sólo entienden frases como: "100101010101" , "11111010101001011010101110101000" y "10101010000000000000000000000001".
- y son muy buenos y rápidos trabajando con este tipo de lenguaje: son los mejores: para un ordenador una secuencia o cadena de 1000 unos y ceros (o de 1 millón de 1s y 0s) es tan sencillo como para ti la palabra "hola".
- así pues un bit es una de las dos letras de su abecedario.

Pero los ordenadores no trabajan de bit en bit: trabajan con grupos de bits simultáneamente.

- a un grupo de 8 bits se le denomina "**byte**" u "**octeto**".
- además, a un grupo de dos bytes se le pueden llamar "**palabra**" (**word**) (a veces, se denomina palabra a un grupo de 32 bits; en el caso del término palabra no hay tanto consenso como en el caso del byte).

Normalmente, el termino "bit" se abrevia con una "**b**" minúscula y el término byte con una "**B**" mayúscula.

- por ejemplo: 34**k**b significa 34 kilobits, 128**M**B significa 128 megabytes.

**Es un error muy serio, conceptualmente hablando, confundir el término bit con el término byte: un BYTE es 8 veces más que un BIT.**

### 1. Múltiplos.

Y hablando de múltiplos. La tabla siguiente resume los principales múltiplos de bit y byte empleados en el Mundo de la Informática. Fíjate que el Mundo de la Informática se trabaja con potencias de 2:

Múltiplos de bit		Múltiplos de byte	
kilobit ( <b>kb</b> )	$2^{10}$ bits = 1024 bits	kilobyte ( <b>KB</b> )	$2^{10}$ bytes = 1024 bytes
megabit ( <b>Mb</b> )	$2^{20}$ bits = 1048576 bits	megabyte ( <b>MB</b> )	$2^{20}$ bytes = 1048576 bytes
gigabit ( <b>Gb</b> )	$2^{30}$ bits = 1073741824 bits	gigabyte ( <b>GB</b> )	$2^{30}$ bytes = 1073741824 bytes
terabit ( <b>Tb</b> )	$2^{40}$ bits = 1099511627776 bits	terabyte ( <b>TB</b> )	$2^{40}$ bytes = 1099511627776 bytes

### 2. Diferencias de terminología entre el Mundo de la Informática y el Mundo de las Comunicaciones.

Nos queda realizar una importante aclaración: existen dos significados distintos para los múltiplos estudiados en el apartado anterior.

- el significado real de un término dependerá del contexto en el que aparezca.
- esto se debe a que en el Mundo de las Comunicaciones se trabaja con potencias de 10 en vez de potencias de 2.

La tabla siguiente resume estas diferencias:

Múltiplos de BIT			
Mundo de la Informática		Mundo de las Comunicaciones	
kilobit ( <b>kb</b> )	$2^{10}$ bits = 1024 bits	kilobit ( <b>kb</b> )	$10^3$ bits = 1000 bits
megabit ( <b>Mb</b> )	$2^{20}$ bits = 1048576 bits	megabit ( <b>Mb</b> )	$10^6$ bits = 1000000 bits
gigabit ( <b>Gb</b> )	$2^{30}$ bits = 1073741824 bits	gigabit ( <b>Gb</b> )	$10^9$ bits = 1000000000 bits
terabit ( <b>Tb</b> )	$2^{40}$ bits = 1099511627776 bits	terabit ( <b>Tb</b> )	$10^{12}$ bits = 1000000000000 bits

Para los bytes tenemos una tabla análoga:

Múltiplos de BYTE			
Mundo de la Informática		Mundo de las Comunicaciones	
kilobyte ( <b>kB</b> )	$2^{10}$ bytes = 1024 bytes	kilobyte ( <b>kB</b> )	$10^3$ bytes = 1000 bytes
megabyte ( <b>MB</b> )	$2^{20}$ bytes = 1048576 bytes	megabyte ( <b>MB</b> )	$10^6$ bytes = 1000000 bytes
gigabyte ( <b>GB</b> )	$2^{30}$ bytes = 1073741824 bytes	gigabyte ( <b>GB</b> )	$10^9$ bytes = 1000000000 bytes
terabyte ( <b>TB</b> )	$2^{40}$ bytes = 1099511627776 bytes	terabyte ( <b>TB</b> )	$10^{12}$ bytes = 1000000000000 bytes

**EJEMPLO 1:** Si hablamos de una memoria RAM de 128MB, estamos hablando de:

$$128 \times 2^{20} \text{ bytes} = 128 \times 1048576 \text{ bytes} = 134217728 \text{ bytes}$$

La memoria RAM "pertenece" al Mundo de la Informática: por eso mega significa, en este contexto,  $2^{20}$ .

**EJEMPLO 2:** Si hablamos de una línea de fibra óptica de 1Gbps por segundo, estamos hablando de:

$$1 \times 10^9 \text{ bits} = 1 \times 1\,000\,000\,000 \text{ bits} = 1\,000\,000\,000 \text{ bits por segundo.}$$

Los bits por segundo es un término que "pertenece" al Mundo de las Comunicaciones. En este contexto mega significa  $10^6$ .

### 3. Submúltiplos del Segundo (Tiempo)

Como repaso, se pone a continuación una tabla con los principales submúltiplos del segundo, que revisten una gran importancia y utilidad en el Mundo de las Comunicaciones:

Submúltiplos del Segundo	
décima de segundo	$10^{-1}$ segundos = 0,1 segundos
centésima de segundo ( <b>cs</b> )	$10^{-2}$ segundos = 0,01 segundos
milisegundo ( <b>ms</b> )	$10^{-3}$ segundos = 0,001 segundos
microsegundo ( <b>μs</b> )	$10^{-6}$ segundos = 0,000001 segundos
nanosegundo ( <b>ns</b> )	$10^{-9}$ segundos = 0,000000001 segundos
picosegundo ( <b>ps</b> )	$10^{-12}$ segundos = 0,000000000001 segundos

## Ampliación: Prefijos de múltiplos binarios

### ¿Cuándo un kilobyte es un kibibyte? ¿Y cuando un MB es un MiB?

Dicho con pocas palabras: con frecuencia. Pero, ¿qué son los mebibytes, gibibytes, tebibytes, pebibytes y exbibytes? La respuesta se encuentra en la norma IEC 60027-2, desarrollada por el TC 25 (Cantidades y unidades y sus símbolos), publicado en noviembre del 2000 y, poco a poco, se va adoptando por el mundo de las TIC. En la tabla que aparece al final, se resumen los detalles esenciales de las nuevas unidades, de dónde proceden, sus símbolos y la relación aproximada con las unidades métricas del *Système international d'unités* (SI).

¿En qué se diferencian estas nuevas unidades, recién estandarizadas, de aquellas que se han hecho tan familiares en las últimas dos o tres décadas de explosión del PC? ¿Realmente importa esta diferencia? Después de todo, la mayor parte de la gente probablemente piensa que saben todo lo que necesitan acerca de kilobytes o megabytes cuando ven que se están quedando sin memoria, recursos o capacidad de disco duro en su PC. O cuando se descargan ficheros Internet a través de un módem, donde el tamaño del fichero puede influir en el coste de la conexión.

El hecho es que, aunque puede que no le haya importado mucho al usuario medio de PC en los últimos años, un kilobyte no son necesariamente 1000 bytes o  $10^3$  bytes como el prefijo del SI "kilo" parece que indica. El SI es un sistema decimal (en base 10), pero los computadores básicamente reconocen si una señal eléctrica está activa o no, lo cual se representa por un 1 o un 0. En jerga matemática, son sistemas binarios (base 2). En lo que respecta a los científicos e ingenieros de las industrias de las telecomunicaciones y las TIC, estas fuentes de confusión y posibles incompatibilidades sí que importan, y aún más cuando los números que machacan los computadores se hacen cada vez más grandes.

La segunda edición del IEC 60027-2 (Símbolos gráficos a usar en la tecnología eléctrica - Parte 2: Telecomunicaciones y electrónica) se desarrolló para ajustarse a las necesidades expresadas por la industria en los sectores de transmisión de datos y procesamiento de datos. Elimina la confusión existente hasta el momento al definir con claridad los prefijos y sus símbolos para los múltiplos binarios, en contraposición a los múltiplos decimales.

### Bits y bytes

Un "bit" es un dígito binario y un "byte" es un grupo de bits, normalmente 8 (de aquí, la palabra octeto para byte). Hace años, cuando los computadores del momento tenían como mucho memorias de decenas de kilobytes, los informáticos observaron que la potencia binaria  $2^{10}$  (=1024) era muy parecida a la potencia decimal  $10^3$  (1000) y, sencillamente por comodidad, comenzaron a llamar a 1024 bytes 1 kilobyte. Después de todo, sólo existía una diferencia del 2,4% y, normalmente, los profesionales distinguían por el contexto cuándo kilobyte significaba  $10^3$  o  $2^{10}$ .

A pesar de la imprecisión y el uso inapropiado del prefijo del sistema decimal del SI, los vendedores y las tiendas encontraron el término fácil de entender; lo mismo pasó con el público. Así, por ejemplo, del omnipresente disquete de 3,5 pulgadas se dice que tiene una capacidad de 1,44MB (megabytes). Esto es falso pues la capacidad originalmente se expresó como 1440kB (kilobytes) antes de ser "traducida" a 1,44MB, es un 2% incorrecta debido al doble uso erróneo de los prefijos decimales.

A medida que pasó el tiempo, los kilobytes han pasado a ser megabytes y los megabytes gigabytes. Es bien posible que, dentro de unos pocos años, la capacidad de almacenamiento de un PC o un portátil se mida en terabytes y en los sistemas industriales muy grandes o en los sistemas científicos se mida en peta o incluso exabytes. Aparece entonces un problema y es que, incluso a la tera-escala del SI ( $10^{12}$ ), la discrepancia entre el "equivalente" binario ( $2^{40}$ ) no es del 2,4% que existía en la kilo-escala sino de cerca del 10%. En la exa-escala ( $10^{18}$  y  $2^{60}$ ) la diferencia es del 20%. Por matemáticas básicas sabemos que cuanto más grande sea el número de bytes mayor será el diferencial; de esta forma, el error - tanto para los ingenieros como para el personal de marketing y el público en general - irá creciendo más y más. Esta una de las buenas razones por la que el IEC estandarizó los prefijos para los múltiplos binarios.

Otra razón importante es que distintas partes de la industria de las TIC habían comenzado a confundirse entre sí. En el mundo de los computadores, por ejemplo, los principales fabricantes de discos duros suelen dar las cantidades del tipo kilobytes, megabytes, gigabytes, etc. de almacenamiento queriendo significar 1000B, 1000000B, 1000000000B respectivamente, en conformidad con el prefijo decimal. La memoria, por otra parte, se describe empleando un prefijo decimal pero realmente significa cantidades binarias. De esta forma, 512MB de



RAM normalmente significan 536870912B y, tal y como se muestra en la tabla, se debería decir una memoria de 512MiB (mebibytes) o 537MB.

Para empeorar las cosas todavía más, de forma frecuente se produce la misma confusión entre los sistemas operativos y las aplicaciones con respecto a sus requisitos de sistema, lo cual conduce a supuestas anomalías e incompatibilidades.

Entre los sectores de los computadores y de las telecomunicaciones del mundo de las TIC, también se han producido confusiones similares, sobre todo debido a que las velocidades de transmisión han crecido enormemente durante las últimas décadas. Generalmente, los diseñadores de redes han empleado el término megabits por segundo (Mbit/s) con el significado 1048576 bit/s mientras que los ingenieros de telecomunicaciones han utilizado el mismo término con el significado 1000000 bit/s. Otro claro ejemplo es la forma de indicar el ancho de banda máximo de un bus PCI: 133,3MB/s. Esto se basaba en que el bus corre a 33,3 MHz y que el ancho de bus es de 4 bytes (32bits): así pues,  $4 \text{ (bytes/Hz)} \times 33,3 \text{ (MHz/s)} = 133 \text{ MB/s}$ . Pero esto es falso porque la M en MHz significa 1000000 mientras que la M en MB significa 1048576.

## Múltiplos binarios

Los nuevos prefijos y símbolos para los múltiplos binarios estandarizados por IEC 60027-2 no forman parte del sistema métrico de unidades SI.

Factor	Nombre	Símbolo	Origen	Derivación del SI
$2^{10}$	kibi	Ki	kilobinario ( $2^{10}$ ) <sup>1</sup>	kilo ( $10^3$ ) <sup>1</sup>
$2^{20}$	mebi	Mi	megabinario ( $2^{10}$ ) <sup>2</sup>	mega ( $10^3$ ) <sup>2</sup>
$2^{30}$	gibi	Gi	gigabinario ( $2^{10}$ ) <sup>3</sup>	giga ( $10^3$ ) <sup>3</sup>
$2^{40}$	tebi	Ti	terabinario ( $2^{10}$ ) <sup>4</sup>	tera ( $10^3$ ) <sup>4</sup>
$2^{50}$	pebi	Pi	petabinario ( $2^{10}$ ) <sup>5</sup>	peta ( $10^3$ ) <sup>5</sup>
$2^{60}$	exbi	Ei	exabinario ( $2^{10}$ ) <sup>6</sup>	exa ( $10^3$ ) <sup>6</sup>
$2^{70}$	zebi	Zi	zettabinario ( $2^{10}$ ) <sup>7</sup>	zetta ( $10^3$ ) <sup>7</sup>
$2^{80}$	yobi	Yi	yottabinario ( $2^{10}$ ) <sup>8</sup>	yotta ( $10^3$ ) <sup>8</sup>