

## 1. Informática e información

La informática (**INFORM**ación + auto**MÁTICA**) estudia el tratamiento automático de la información mediante sistemas informáticos (S.I.). Un S.I. recibe información, la procesa siguiendo unas instrucciones y genera unos resultados en cuestión de milisegundos. Para poder realizar estas operaciones, toda la información debe representarse de una determinada manera para que el sistema pueda interpretarlo.



## 2. Representación binaria de la información

Los S.I. representan internamente la información utilizando dos estados de electricidad, corriente y no corriente, que se identifican mediante los valores binarios 0 y 1. De forma simplificada, un nivel eléctrico de 5 V puede asociarse al valor 1 (verdadero/sí) y un nivel de 0 V al valor 0 (falso/no). Cada 0 o 1 recibe el nombre de dígito binario o **bit** (**B**inary **d**ig**I**T). El bit es, por tanto, la unidad mínima de información que puede manejar un S.I., pero por sí sola no tiene significado.



Los bits se agrupan para representar información más compleja con significado propio. Un conjunto de 8 bits forma un **byte** (B) o palabra. Por ejemplo, 11001001 es un byte formado por ocho bits. A partir del byte se construyen unidades mayores que permiten expresar letras, señales de control, el tamaño de los datos, la dirección de almacenamiento...

## 3. Múltiplos de la información

Cuando la cantidad de información aumenta, resulta poco práctico expresarla únicamente en bytes. Por ello se utilizan múltiplos. En este tema se emplean las siguientes equivalencias:

8 bits (b) = 1 Byte (B)

1024 KB = 1 MegaByte (MB)

1024 GB = 1 TeraByte (TB)

1024 B = 1 KiloByte (KB)

1024 MB = 1 GigaByte (GB)

1024 TB = 1 PetaByte (PB).

PISTA: Para resolver problemas de capacidad o velocidad es necesario convertir todas las cantidades a una misma unidad. Si, por ejemplo, conocemos una capacidad en GB y el tamaño de cada archivo en MB, primero debemos convertir los GB a MB. Después podremos dividir la capacidad disponible entre el tamaño medio de cada archivo.

## 4. Problemas de capacidad y velocidad de transmisión

**4.1.** Si tengo un MP3 de 4 GB y sé que de promedio una canción pesa 5 MB...

1.A. ¿Cuántas canciones caben en mi MP3?

$$4 \text{ GB} \cdot \frac{1 \text{ canción}}{5 \text{ MB}} \cdot \frac{1024 \text{ MB}}{1 \text{ GB}} = \frac{4096}{5} \text{ canciones} = 819,2 \text{ canciones}$$

Solución: "Solo" cabrán 819 canciones, no se podrá guardar la fracción de 0,2 restante.

1.B. Estimando que la duración media de una canción es de 3 min., ¿cuántas horas de música caben?

$$819 \text{ canciones} \cdot \frac{3 \text{ min}}{1 \text{ canción}} = 2.457 \text{ min} = 40\text{h y } 57'$$

Solución: Se podría escuchar 40 horas y 57 minutos de música ininterrumpidamente sin repetir ninguna canción.

**4.2.** En un PC de la clase de informática se han descargado todas las fotos del viaje a Italia, en total son 2,100 fotos. Si tengo un pendrive de 8 GB y sé que de promedio una foto pesa unos 4 MB... ¿Cuántas veces tendré que ir a la clase a llenar mi pendrive para llevármelas todas?

$$2100 \text{ fotos} \cdot \frac{4 \text{ MB}}{1 \text{ foto}} \cdot \frac{1 \text{ vez}}{8 \text{ GB}} \cdot \frac{1 \text{ GB}}{1024 \text{ MB}} = \frac{8400}{8192} \text{ veces} = 1,02 \text{ veces}$$

*Solución: Hay que ir más de 1 vez, concretamente 2 veces.*

**4.3.** Si mi contrato de conexión a internet especifica que tengo una conexión de 20 Megas ... (20 Mb/sg) ¿En cuánto tiempo me bajaré una película de aproximadamente 1 GB?

$$1 \text{ GB} \cdot \frac{1 \text{ sg}}{20 \text{ Mb}} \cdot \frac{1024 \text{ M}}{1 \text{ G}} \cdot \frac{8 \text{ b}}{1 \text{ B}} = \frac{8192}{20} \text{ sg} = 409,6 \text{ sg} \approx 6'49''$$

*Solución: Me bajaré una película de 1 GB en aproximadamente 6 minutos y 49 segundos con una conexión de 20 Mbps.*

## 5. Tipos de información en un S.I.

Toda la información que maneja un sistema informático (S.I.) puede agruparse, de forma simplificada, en dos grandes grupos dividido en otros dos subgrupos:

### ▪ INSTRUCCIONES:

- Señales de control
- Direcciones

### ▪ DATOS:

- Datos numéricos
- Datos alfanuméricos

Las **instrucciones** indican al S.I. qué debe hacer y cómo debe actuar. Dentro de ellas distinguiremos las **señales de control** y las **direcciones**.

Los **datos** constituyen la información sobre la que trabaja el S.I. Dentro de ellos distinguiremos los datos **numéricos** y los datos **alfanuméricos**.

### 5.1. Instrucciones tipo Señales de control:

Las **señales de control** permiten **coordinar y controlar el funcionamiento de los diferentes componentes hardware** de un S.I. Mediante estas señales se indica qué acción debe realizar un determinado componente y en qué momento debe hacerlo.

Por ejemplo, cuando un videojuego necesita mostrar un objeto en la pantalla, el S.I. genera las señales necesarias para que el hardware gráfico **establezca el color de determinados píxeles**. Si uno de esos píxeles debe mostrarse de color rojo, se envía la información correspondiente para que ese píxel adopte los valores de color adecuados.

Algo similar ocurre con el sonido. Cuando una aplicación necesita reproducir un sonido por los altavoces, el S.I. procesa los datos de audio y genera las señales necesarias para que el **hardware de sonido** produzca la señal que finalmente llegará a los altavoces y se transformará en sonido.

Por tanto, las señales de control permiten que las instrucciones que ejecuta el S.I. **se traduzcan en acciones concretas realizadas por el hardware**, como mostrar información en la pantalla, reproducir un sonido, escribir información en una unidad de almacenamiento o comunicarse con un periférico.

## 5.2. Instrucciones tipo Direcciones:

Las **direcciones** permiten al S.I. saber **dónde se encuentra almacenada una determinada información** y, por tanto, dónde debe buscarla para poder utilizarla. También permiten indicar **dónde debe guardarse** el resultado de una operación.

La información de un S.I. puede encontrarse almacenada en diferentes tipos de memoria o dispositivos de almacenamiento.

Supongamos que queremos realizar una suma, abrimos la calculadora y escribimos la operación **25 + 17**. Los valores 25 y 17 van a estar almacenados en determinadas **direcciones de la memoria RAM**, por ejemplo, el 25 va a la posición 10011100 y el 17 a la posición 10011101 que son las primeras posiciones libres que había en la RAM. El S.I. utiliza esas direcciones para almacenar y localizar ambos datos, realiza la suma y vuelve a guardar el resultado, **42**, en otra dirección de memoria libre 10011110.

| RAM       |        |
|-----------|--------|
| dirección | dato   |
| ...       | ...    |
| 10011011  | xxxxxx |
| 10011100  | 25     |
| 10011101  | 17     |
| 10011110  | 42     |
| 10011111  |        |
| ...       | ...    |

*Recuerda que los datos se guardarían en binario no en decimal*

## 5.3. Datos Numéricos:

Los **datos numéricos** representan números y cantidades sobre los que pueden realizarse operaciones básicas de matemáticas (sumas, restas, divisiones, multiplicaciones, ...) y lógicas (<, >, =, ≠, ≤, ≥, ...).

Por ejemplo, si pensamos en la aplicación del *cuaderno de Séneca*, donde se guardan vuestras notas de cada asignatura, allí se realizan las medias y automáticamente la aplicación decide si estáis aprobados o suspensos. Imaginemos que en digitalización solo se van a hacer 3 pruebas en este trimestre, y las notas que sacáis son 6, 3 y 9. Esos 3 valores se almacenarán como datos numéricos, ya que hay que hacer operaciones matemáticas sobre ellos, habrá que hacer la media aritmética  $(6+3+9)/3$ . Y al resultado habrá que aplicarle una operación lógica: ¿Es mayor o igual que 5? (para ver si está aprobado). Así que dentro del S.I. tendrá que producirse esta operación  $[(6+3+9)/3] \geq 5$ , siendo todos estos datos de tipo numérico para poder trabajar con ellos y que el sistema automáticamente pueda devolver *aprobado* o *suspenso*.

## 5.4. Datos Alfanuméricos:

Los **datos alfanuméricos** son aquellos que representan **texto**, es decir, combinaciones de **letras, números, espacios, signos de puntuación y otros símbolos**.

Por ejemplo, podemos escribir en un mensaje: “Voy a sacar un **diez** en esta prueba” o “Voy a sacar un **10** en esta prueba”, En ambos casos, para nosotros, «diez» y «10» hacen referencia al mismo valor. Sin embargo, dentro de esas

frases están formando parte de un texto, por lo que el S.I. los trata como caracteres, es decir, como datos alfanuméricos.

## 6. Datos alfanuméricos y código ASCII

Para que un S.I. pueda almacenar y procesar letras, números y símbolos, estos caracteres deben convertirse en valores numéricos. Uno de los sistemas utilizados para ello es ASCII, siglas de *American Standard Code for Information Interchange*. ASCII establece una correspondencia entre cada carácter y un número determinado.

| Decimal | Binario  | Carácter | Decimal | Binario  | Carácter | Decimal | Binario  | Carácter | Decimal | Binario  | Carácter |
|---------|----------|----------|---------|----------|----------|---------|----------|----------|---------|----------|----------|
| 0       | 00000000 | NUL      | 32      | 00100000 | espacio  | 64      | 01000000 | @        | 96      | 01100000 | `        |
| 1       | 00000001 | SOH      | 33      | 00100001 | !        | 65      | 01000001 | A        | 97      | 01100001 | a        |
| 2       | 00000010 | STX      | 34      | 00100010 | "        | 66      | 01000010 | B        | 98      | 01100010 | b        |
| 3       | 00000011 | ETX      | 35      | 00100011 | #        | 67      | 01000011 | C        | 99      | 01100011 | c        |
| 4       | 00000100 | EOT      | 36      | 00100100 | \$       | 68      | 01000100 | D        | 100     | 01100100 | d        |
| 5       | 00000101 | ENQ      | 37      | 00100101 | %        | 69      | 01000101 | E        | 101     | 01100101 | e        |
| 6       | 00000110 | ACK      | 38      | 00100110 | &        | 70      | 01000110 | F        | 102     | 01100110 | f        |
| 7       | 00000111 | BEL      | 39      | 00100111 | '        | 71      | 01000111 | G        | 103     | 01100111 | g        |
| 8       | 00001000 | BS       | 40      | 00101000 | (        | 72      | 01001000 | H        | 104     | 01101000 | h        |
| 9       | 00001001 | TAB      | 41      | 00101001 | )        | 73      | 01001001 | I        | 105     | 01101001 | i        |
| 10      | 00001010 | LF       | 42      | 00101010 | *        | 74      | 01001010 | J        | 106     | 01101010 | j        |
| 11      | 00001011 | VT       | 43      | 00101011 | +        | 75      | 01001011 | K        | 107     | 01101011 | k        |
| 12      | 00001100 | FF       | 44      | 00101100 | ,        | 76      | 01001100 | L        | 108     | 01101100 | l        |
| 13      | 00001101 | CR       | 45      | 00101101 | -        | 77      | 01001101 | M        | 109     | 01101101 | m        |
| 14      | 00001110 | SO       | 46      | 00101110 | .        | 78      | 01001110 | N        | 110     | 01101110 | n        |
| 15      | 00001111 | SI       | 47      | 00101111 | /        | 79      | 01001111 | O        | 111     | 01101111 | o        |
| 16      | 00010000 | DLE      | 48      | 00110000 | 0        | 80      | 01010000 | P        | 112     | 01110000 | p        |
| 17      | 00010001 | DC1      | 49      | 00110001 | 1        | 81      | 01010001 | Q        | 113     | 01110001 | q        |
| 18      | 00010010 | DC2      | 50      | 00110010 | 2        | 82      | 01010010 | R        | 114     | 01110010 | r        |
| 19      | 00010011 | DC3      | 51      | 00110011 | 3        | 83      | 01010011 | S        | 115     | 01110011 | s        |
| 20      | 00010100 | DC4      | 52      | 00110100 | 4        | 84      | 01010100 | T        | 116     | 01110100 | t        |
| 21      | 00010101 | NAK      | 53      | 00110101 | 5        | 85      | 01010101 | U        | 117     | 01110101 | u        |
| 22      | 00010110 | SYN      | 54      | 00110110 | 6        | 86      | 01010110 | V        | 118     | 01110110 | v        |
| 23      | 00010111 | ETB      | 55      | 00110111 | 7        | 87      | 01010111 | W        | 119     | 01110111 | w        |
| 24      | 00011000 | CAN      | 56      | 00111000 | 8        | 88      | 01011000 | X        | 120     | 01111000 | x        |
| 25      | 00011001 | EM       | 57      | 00111001 | 9        | 89      | 01011001 | Y        | 121     | 01111001 | y        |
| 26      | 00011010 | SUB      | 58      | 00111010 | :        | 90      | 01011010 | Z        | 122     | 01111010 | z        |
| 27      | 00011011 | ESC      | 59      | 00111011 | ;        | 91      | 01011011 | [        | 123     | 01111011 | {        |
| 28      | 00011100 | FS       | 60      | 00111100 | <        | 92      | 01011100 | \        | 124     | 01111100 |          |
| 29      | 00011101 | GS       | 61      | 00111101 | =        | 93      | 01011101 | ]        | 125     | 01111101 | }        |
| 30      | 00011110 | RS       | 62      | 00111110 | >        | 94      | 01011110 | ^        | 126     | 01111110 | ~        |
| 31      | 00011111 | US       | 63      | 00111111 | ?        | 95      | 01011111 | _        | 127     | 01111111 | DEL      |

Gracias a esta codificación, caracteres como letras mayúsculas y minúsculas, cifras, signos de puntuación y otros símbolos pueden representarse internamente mediante valores binarios. La tabla ASCII muestra precisamente la relación entre el carácter y sus diferentes representaciones numéricas.

Todos los sistemas del mundo utilizan esta codificación para comunicarse entre ellos.

La tabla ASCII original utiliza 7 bits, por lo que puede representar únicamente 128 caracteres (0–127), suficientes para las letras del alfabeto inglés, números, signos y caracteres de control. Sin embargo, resultaba insuficiente para representar caracteres utilizados en otros idiomas, como la ñ, á, é, í, ó, ú, ü, además de otros símbolos.



Por este motivo surgieron diferentes versiones de ASCII extendido, que utilizan 8 bits (1 byte) y permiten representar hasta 256 valores (0–255). Los primeros 128 mantienen normalmente los caracteres del ASCII original y los valores del 128 al 255 se utilizan para caracteres adicionales. Para utilizarlo, a cada carácter se le asigna un código numérico que el S.I. almacena en forma binaria; por ejemplo, si un carácter tiene asignado el código decimal 65, el S.I. lo representa como 01000001. Es importante señalar que **no existe una única tabla ASCII extendida universal**: los caracteres del 128 al 255 **dependen de la página de códigos utilizada por cada país**.

| Decimal | Binario  | Carácter |
|---------|----------|----------|
| 128     | 10000000 | Ç        |
| 129     | 10000001 | ü        |
| 130     | 10000010 | é        |
| 131     | 10000011 | â        |
| 132     | 10000100 | ä        |
| 133     | 10000101 | à        |
| 134     | 10000110 | å        |
| 135     | 10000111 | ç        |
| 136     | 10001000 | ê        |
| 137     | 10001001 | ë        |
| 138     | 10001010 | è        |
| 139     | 10001011 | ï        |
| 140     | 10001100 | î        |
| 141     | 10001101 | ì        |
| 142     | 10001110 | À        |
| 143     | 10001111 | Á        |
| 144     | 10010000 | Ê        |
| 145     | 10010001 | æ        |
| 146     | 10010010 | Æ        |
| 147     | 10010011 | ô        |
| 148     | 10010100 | ö        |
| 149     | 10010101 | ò        |
| 150     | 10010110 | ù        |
| 151     | 10010111 | ù        |
| 152     | 10011000 | ÿ        |
| 153     | 10011001 | Ö        |
| 154     | 10011010 | Ü        |
| 155     | 10011011 | €        |
| 156     | 10011100 | £        |
| 157     | 10011101 | ¥        |
| 158     | 10011110 | Pts      |
| 159     | 10011111 | f        |

| Decimal | Binario  | Carácter |
|---------|----------|----------|
| 160     | 10100000 | á        |
| 161     | 10100001 | í        |
| 162     | 10100010 | ó        |
| 163     | 10100011 | ú        |
| 164     | 10100100 | ñ        |
| 165     | 10100101 | Ñ        |
| 166     | 10100110 | ª        |
| 167     | 10100111 | º        |
| 168     | 10101000 | ¿        |
| 169     | 10101001 | ¡        |
| 170     | 10101010 | ¬        |
| 171     | 10101011 | ½        |
| 172     | 10101100 | ¼        |
| 173     | 10101101 | ı        |
| 174     | 10101110 | «        |
| 175     | 10101111 | »        |
| 176     | 10110000 |          |
| 177     | 10110001 |          |
| 178     | 10110010 |          |
| 179     | 10110011 |          |
| 180     | 10110100 |          |
| 181     | 10110101 |          |
| 182     | 10110110 |          |
| 183     | 10110111 |          |
| 184     | 10111000 |          |
| 185     | 10111001 |          |
| 186     | 10111010 |          |
| 187     | 10111011 |          |
| 188     | 10111100 |          |
| 189     | 10111101 |          |
| 190     | 10111110 |          |
| 191     | 10111111 |          |

| Decimal | Binario  | Carácter |
|---------|----------|----------|
| 192     | 11000000 | ˆ        |
| 193     | 11000001 | ˆ        |
| 194     | 11000010 | ˆ        |
| 195     | 11000011 | ˆ        |
| 196     | 11000100 | ˆ        |
| 197     | 11000101 | ˆ        |
| 198     | 11000110 | ˆ        |
| 199     | 11000111 | ˆ        |
| 200     | 11001000 | ˆ        |
| 201     | 11001001 | ˆ        |
| 202     | 11001010 | ˆ        |
| 203     | 11001011 | ˆ        |
| 204     | 11001100 | ˆ        |
| 205     | 11001101 | ˆ        |
| 206     | 11001110 | ˆ        |
| 207     | 11001111 | ˆ        |
| 208     | 11010000 | ˆ        |
| 209     | 11010001 | ˆ        |
| 210     | 11010010 | ˆ        |
| 211     | 11010011 | ˆ        |
| 212     | 11010100 | ˆ        |
| 213     | 11010101 | ˆ        |
| 214     | 11010110 | ˆ        |
| 215     | 11010111 | ˆ        |
| 216     | 11011000 | ˆ        |
| 217     | 11011001 | ˆ        |
| 218     | 11011010 | ˆ        |
| 219     | 11011011 | ˆ        |
| 220     | 11011100 | ˆ        |
| 221     | 11011101 | ˆ        |
| 222     | 11011110 | ˆ        |
| 223     | 11011111 | ˆ        |

| Decimal | Binario  | Carácter |
|---------|----------|----------|
| 224     | 11100000 | α        |
| 225     | 11100001 | β        |
| 226     | 11100010 | Γ        |
| 227     | 11100011 | π        |
| 228     | 11100100 | Σ        |
| 229     | 11100101 | σ        |
| 230     | 11100110 | μ        |
| 231     | 11100111 | τ        |
| 232     | 11101000 | Φ        |
| 233     | 11101001 | Θ        |
| 234     | 11101010 | Ω        |
| 235     | 11101011 | δ        |
| 236     | 11101100 | ∞        |
| 237     | 11101101 | φ        |
| 238     | 11101110 | ε        |
| 239     | 11101111 | η        |
| 240     | 11110000 | ≡        |
| 241     | 11110001 | ±        |
| 242     | 11110010 | ≥        |
| 243     | 11110011 | ≤        |
| 244     | 11110100 | ∫        |
| 245     | 11110101 | ∫        |
| 246     | 11110110 | ÷        |
| 247     | 11110111 | ≈        |
| 248     | 11111000 | °        |
| 249     | 11111001 | ·        |
| 250     | 11111010 | ·        |
| 251     | 11111011 | √        |
| 252     | 11111100 | n        |
| 253     | 11111101 | ²        |
| 254     | 11111110 | ■        |
| 255     | 11111111 | espacio  |

## 7. Datos numéricos y sistemas de numeración

Los datos numéricos pueden representarse utilizando diferentes sistemas de numeración. En este tema se trabajan tres sistemas fundamentales: binario, decimal y hexadecimal.

En binario es como entienden las máquinas la información (corriente/no corriente), en decimal es como entendemos los seres humanos las cantidades y el sistema hexadecimal está a caballo entre lo que entienden las máquinas y lo que entiende el ser humano.

El sistema binario utiliza únicamente los dígitos 0 y 1; el decimal utiliza los dígitos del 0 al 9; y el hexadecimal utiliza los dígitos del 0 al 9 y las letras A, B, C, D, E y F para representar los valores del 10 al 15.

Un mismo valor puede escribirse de formas diferentes dependiendo de la base utilizada. Por ejemplo,  $00011011_2$ ,  $27_{10}$  y  $1B_{16}$  representan el mismo valor. Cambia su escritura, pero no la cantidad que representan y debemos saber pasar de un a otro en esta asignatura.

## 8. Conversión de binario a decimal

Para convertir un número binario a decimal se utiliza el método de las potencias de 2. En primer lugar se numeran las posiciones de los bits de derecha a izquierda, comenzando por 0. Cada posición corresponde a una potencia de 2:  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$ , etc.

Después se toman únicamente las posiciones en las que aparece un 1 y se calcula la potencia de 2 correspondiente. Finalmente se suman todos esos valores.

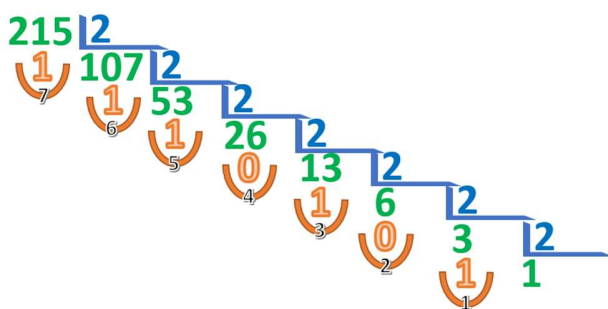
Por ejemplo, en  $10011011_2$  aparecen “unos” en las posiciones 7, 4, 3, 1 y 0, por lo que se suman  $2^7 + 2^4 + 2^3 + 2^1 + 2^0 = 128 + 16 + 8 + 2 + 1 = 155$ . Por tanto,  $10011011_2 = 155_{10}$ .

$$\begin{array}{ccccccc}
 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
 \downarrow & & & \downarrow & \downarrow & & \downarrow & \downarrow \\
 2^7 & & & 2^4 & 2^3 & & 2^1 & 2^0 \\
 || & & & || & || & & || & || \\
 128 & + & 16+8 & + & 2+1 & = & 155_{(10)}
 \end{array}$$

## 9. Conversión de decimal a binario

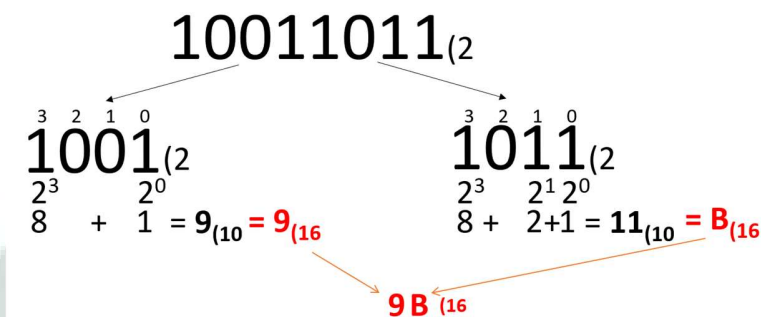
Para convertir un número decimal a binario se utiliza el método de las divisiones sucesivas entre 2. Se divide el número entre 2 mediante división entera y se anota el resto. El cociente obtenido vuelve a dividirse entre 2 y se repite el proceso hasta que el cociente sea menor que 2.

El número binario se obtiene leyendo **el último cociente y todos los restos en orden inverso**, es decir, desde el final hacia el principio. Si necesitamos representar el resultado mediante un byte, completamos con ceros por la izquierda hasta alcanzar 8 bits. El procedimiento se aplica en el ejemplo al número decimal 215.



$11010111_2$

## 10. Conversión de binario a hexadecimal



Para pasar de binario a hexadecimal se divide el número binario en grupos de 4 bits. Cada grupo de cuatro bits recibe el nombre de nibble (1 nibble = 4 bits). Si fuera necesario, se completarían los grupos con ceros por la izquierda.

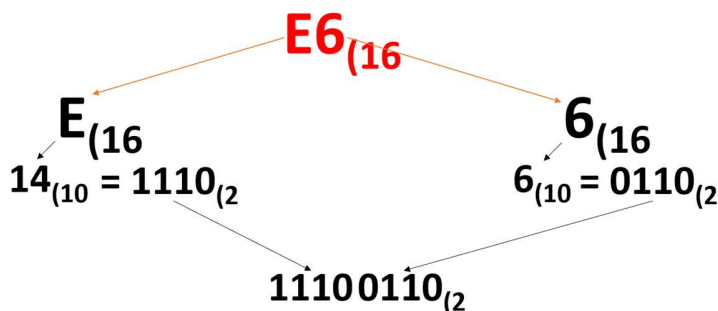
A continuación, se calcula el valor decimal de cada nibble utilizando las potencias de 2 y se sustituye ese valor por su símbolo hexadecimal.

| DECIMAL     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| HEXADECIMAL | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A  | B  | C  | D  | E  | F  |

Los valores decimales 10, 11, 12, 13, 14 y 15 se representan respectivamente mediante A, B, C, D, E y F. Finalmente se unen los símbolos obtenidos. Así,  $1011_2$  equivale a  $11_{10}$ , es decir,  $B_{16}$ ; y  $1001_2$  equivale a  $9_{10}$ , es decir,  $9_{16}$ . Por tanto,  $10111001_2 = B9_{16}$ .

## 11. Conversión de hexadecimal a binario

Para convertir de hexadecimal a binario se trabaja con cada dígito hexadecimal por separado. Primero se identifica su valor decimal y después se representa ese valor mediante un grupo de 4 bits. Cada dígito hexadecimal se convierte, por tanto, en un nibble.



Por ejemplo, para convertir  $E6_{16}$ , la cifra E equivale a  $14_{10}$  y se representa como  $1110_2$ . La cifra 6 equivale a  $6_{10}$  y se representa como  $0110_2$ . Al unir ambos nibbles obtenemos  $11100110_2$ . Por tanto,  $E6_{16} = 11100110_2$ .

## 12. Conversiones entre hexadecimal, binario y decimal

Las conversiones entre los tres sistemas pueden organizarse utilizando el binario como sistema intermedio. Para pasar de hexadecimal a decimal puede realizarse primero la conversión de hexadecimal a binario y, a continuación, de binario a decimal. De este modo se aplican procedimientos ya conocidos.

De manera semejante, para pasar de decimal a hexadecimal puede convertirse primero el número decimal a binario y después agrupar los bits de cuatro en cuatro para obtener los dígitos hexadecimales. Es importante dominar las conversiones básicas, ya que permiten relacionar las tres formas de representar los datos numéricos.

**Hexadecimal**



**Binario**



**Decimal**

**Decimal**



**Binario**



**Hexadecimal**

### 13. Todo sistema informático está compuesto por hardware y software.

Un sistema informático (S.I.) necesita dos componentes fundamentales para funcionar: **hardware y software**. Ambos trabajan conjuntamente: el hardware constituye la parte física del sistema, mientras que el software contiene las instrucciones electrónicas, o comúnmente mal llamados *programas*, que indican al hardware qué operaciones debe realizar.

Esto no se limita a los ordenadores, tablets o smartphones. Muchos dispositivos y máquinas actuales, como por ejemplo un smartwatch, un smartTV, una Thermomix, un ascensor, un frigorífico o un coche, incorporan sistemas informáticos.

Podemos entenderlo con un ejemplo sencillo: un teléfono móvil tiene pantalla, procesador, memoria, cámaras y altavoces. Todo eso es hardware. Pero esos componentes, por sí solos, no saben qué tienen que hacer. Necesitan software, como el sistema operativo Android y las aplicaciones instaladas, para poder realizar tareas concretas.

**Hardware y software se necesitan mutuamente**, aunque los estudiemos por separado, **hardware y software funcionan conjuntamente**. El hardware sin software dispone de componentes físicos, pero no de las instrucciones necesarias para realizar las tareas que queremos, por lo tanto no funcionaría. Y el software necesita un hardware sobre el que ejecutarse, sin él no puede existir.

### 14. Hardware

El **hardware** es el conjunto de todos los **componentes físicos de un sistema informático**, es decir, aquellos elementos que existen físicamente y que, en muchos casos, podemos ver y tocar.

Son hardware, por ejemplo, el **teclado, ratón, pantalla, procesador, memoria RAM, placa base, tarjeta gráfica, tarjeta de red, disco duro SSD, impresora, cámara web**, etc.

No todo el hardware se encuentra a simple vista. Cuando utilizamos un PC podemos ver el monitor, el teclado o el ratón, pero en el interior de la caja existen otros componentes fundamentales. Por ejemplo, la **placa base**, el **procesador (CPU)**, la **memoria RAM**, la **memoria de almacenamiento masivo...** y muchos más.

### 15. Software

El **software** es el conjunto de **instrucciones y datos** que permiten controlar el hardware y realizar diferentes tareas. A diferencia del hardware, el software es **intangible**: no podemos tocar físicamente Windows, Android o una aplicación.

Por ejemplo, cuando hacemos clic con el ratón sobre el icono del navegador web *Google Chrome*, estamos utilizando hardware (ratón, procesador, memoria RAM, monitor, tarjeta de red, ...) para ejecutar un software. Chrome contiene las instrucciones necesarias para que el S.I. sepa qué debe hacer para mostrarnos una página web.

Son una muestra muy pequeña de ejemplos de software **Windows, Android, IOS, Word, Excel, Chrome, Spotify, Instagram, Telegram o WhatsApp**.