## 2. Sistemas de numeración

Se define **sistema de numeración** como el conjunto de símbolos utilizados para la representación de cantidades, así como las reglas que rigen dicha representación.

Un sistema de numeración se distingue por su **base**, que es el número de símbolos que utiliza, y se caracteriza por ser el coeficiente que determina cuál es el valor de cada símbolo dependiendo de su posición.

El sistema de numeración que utilizamos normalmente es el **sistema decimal**, de base 10. El sistema decimal utiliza diez dígitos o símbolos:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Dependiendo de la posición que ocupe un dígito dentro de una cifra, representará las unidades, decenas, centenas, millares, etc. Por esto, se dice que los sistemas de numeración son **posicionales.**

Por ejemplo, en este sistema el valor del número 6 839 se puede expresar como sumas de potencias de la **base 10:**

$$(6 \cdot 10^3) + (8 \cdot 10^2) + (3 \cdot 10^1) + (9 \cdot 10^0) = 6\,839$$

Que, de hecho, es como expresamos oralmente esta cifra:

«seis mil/ochocientos/treinta/y nueve»

Podemos definir también un sistema de numeración como un conjunto de dígitos y reglas que permiten representar datos numéricos. La **principal regla** es que un *mismo dígito tiene distinto valor según la posición que ocupe.*

**Ejemplos**

Si tenemos el número 555, el dígito 5 tiene distinto valor dependiendo de la posición que ocupe. Cada posición tiene un peso asociado, siendo en este caso 5 las unidades, 50 las decenas y 500 las centenas. El dígito más a la derecha tendrá peso 0, el siguiente 1, el siguiente 2, y así sucesivamente.

Podremos representar este número como las sumas de las potencias de la base 10 elevada al peso:

$(5 \cdot 10^2) + (5 \cdot 10^1) + (5 \cdot 10^0)$

**Errores frecuentes**

Recuerda que cualquier número elevado a 0 es igual a 1.

### 2.1. Teorema fundamental de la numeración

Este teorema relaciona una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal; es decir, el valor decimal de una cantidad expresada en otro sistema de numeración que utiliza otra base. Viene dado por la fórmula:

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

Donde:

- $i$ = posición respecto a la coma. Para los dígitos de la derecha la $i$ es negativa, empezando en –1; para los de la izquierda es positiva, empezando en 0.
- $d$ = número de dígitos a la derecha de la coma.
- $n$ = número de dígitos a la izquierda de la coma –1.
- dígito = cada uno de los que componen el número.
- base = base del sistema de numeración.

El número en decimal será el sumatorio de multiplicar cada dígito por la base elevada a su posición. $i$ indica la posición del dígito respecto a la coma; si el número tiene decimales, $i$ se iniciará con valor negativo.

El teorema, aplicado a la inversa, servirá para obtener la representación de una cantidad decimal en cualquier otra base por medio de divisiones sucesivas por dicha base. Esto se verá más adelante.

### Caso práctico 1

**En este caso práctico, vamos a ver cómo se expone el sumatorio del 6578:**

1. Calculamos los valores de la fórmula:
   - $d = 0$, no hay coma
   - $i = -d = 0$
   - $n = 3$
2. Calculamos los pesos asociados a los dígitos según la posición. El peso 0 lo tiene el dígito de la derecha, y el peso $n$ el de la izquierda (véase la Tabla 1.1).

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

Tabla 1.1. *Pesos asociados a la cantidad 6578.*

3. Sumamos según la fórmula:
   $(6 \cdot 10^3) + (5 \cdot 10^2) + (7 \cdot 10^1) + (8 \cdot 10^0) \rightarrow$
   $\rightarrow 6000 + 500 + 70 + 8 = 6578$

### Caso práctico 2

**En este caso práctico, vamos a ver cómo se expresa una cantidad con decimales, por ejemplo 34,275:**

1. Calculamos los valores de la fórmula:
   - $d = 3$, dígitos a la derecha de la coma.
   - $i = -d = -3$, el valor $i$ empezará en menos 3; los pesos se muestran en la Tabla 1.2.
   - $n = 2 - 1 = 1$, dígitos a la izquierda de la coma.
2. Calculamos los pesos asociados a los dígitos según la posición de la coma. El peso más pequeño, –3 (valor inicial de la *i*), lo tiene el dígito situado más a la derecha, el peso más alto *n*, lo tiene el dígito situado más a la izquierda (véase Tabla 1.2).

| Pesos | 1 | 0 | | -1 | -2 | -3 |
|---|---|---|---|---|---|---|
| | $10^1$ | $10^2$ | | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ |
| Dígitos | 3 | 4 | , | 2 | 7 | 5 |

Tabla 1.2. *Pesos asociados a una cantidad con decimales.*

Derecha de la coma:
$(2 \cdot 10^{-1}) + (7 \cdot 10^{-2}) + (5 \cdot 10^{-3}) \rightarrow 0{,}2 + 0{,}07 +$
$+ 0{,}005 \rightarrow 0{,}275$

Izquierda de la coma:
$(3 \cdot 10^1) + (4 \cdot 10^0) \rightarrow 30 + 4 \rightarrow 34$

El sumatorio será:
$(3 \cdot 10^1) + (4 \cdot 10^0) + (2 \cdot 10^{-1}) + (7 \cdot 10^{-2}) +$
$+ (5 \cdot 10^{-3}) \rightarrow 34{,}275$

### Caso práctico 3

**La cantidad 112,02 está expresada en el sistema de numeración de base 3, que emplea los dígitos 0, 1 y 2 para representar las cantidades. Vamos a ver cuál es la representación de este número en el sistema decimal.**

1. Calculamos los valores de la fórmula:
   - $d = 2$, dígitos a la derecha de la coma.
   - $i = -d = -2$.
   - $n = 3 - 1 = 2$, dígitos a la izquierda de la coma.
2. Calculamos los pesos asociados a los dígitos según la posición de la coma (véase la Tabla 1.3). En este ejemplo la base es 3, con lo que multiplicamos por 3, no por 10.

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

Tabla 1.3. *Pesos asociados a una cantidad en base 3.*

Derecha de la coma:
$(0 \cdot 3^{-1}) + (2 \cdot 3^{-2}) \rightarrow 0 + 0{,}2222 \rightarrow 0{,}2222$

Izquierda de la coma:
$(1 \cdot 3^2) + (1 \cdot 3^1) + (2 \cdot 3^0) \rightarrow 9 + 3 + 2 \rightarrow 14$

El sumatorio será:
$(1 \cdot 3^2) + (1 \cdot 3^1) + (2 \cdot 3^0) + (0 \cdot 3^{-1}) + (2 \cdot 3^{-2}) \rightarrow$
$\rightarrow 14{,}2222$

**Actividades**

2. **Expresar las cantidades 76 890 y 234,765 según el teorema fundamental de la numeración.**
3. **Expresa en decimal estas cantidades dadas en diversos sistemas de numeración y bases distintas:**
   *a)* 201,12 en base 4 (sistema que utiliza los dígitos 0, 1, 2, 3).
   *b)* 340,31 en base 5 (sistema que utiliza los dígitos 0, 1, 2, 3, 4).
   *c)* 215,241 en base 6 (sistema que utiliza los dígitos 0, 1, 2, 3, 4, 5).

**Claves y consejos**

La **cantidad de dígitos** de un número en binario dependerá del valor de dicho número en el sistema decimal. Hemos visto que para representar el número 25 necesitamos cinco **dígitos binarios**. Para representar cualquier número decimal nos guiaremos de la siguiente tabla:

| 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* |

Tabla 1.5. *Cantidad de dígitos para representar un número decimal.*

## 2.2. El sistema binario

El sistema de numeración binario utiliza solo **dos dígitos** (0 y 1) para representar cantidades, por lo que su **base es 2.** Cada dígito de un número representado por este sistema se denomina **bit *(binary digit).***

Los bits tienen distinto valor dependiendo de la posición que ocupan; por eso este sistema también es posicional. Estos valores vienen determinados por una potencia de base 2 a la que vamos a llamar *peso*. Así, por ejemplo, el número binario 1 011,01 expresado en decimal quedaría así:

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

En la Tabla 1.4 se muestran los pesos en potencia de 2 asociados según la posición del dígito. Para convertir a decimal, basta con colocar los dígitos en las columnas correspondientes y sumar los pesos donde hay un 1, hasta obtener la cantidad.

| Pesos asociados | | | | | | | | Número 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 |

Tabla 1.4. *Conversión binario-decimal sumando los pesos donde hay un 1.*

### A. Conversión de un número decimal a binario

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

Por tanto, si queremos representar el número decimal 25 en binario, realizaremos divisiones sucesivas por 2 hasta obtener un cociente menor de 2. El número resultante será **el último cociente** y tras él **los restos** obtenidos en cada una de las divisiones, empezando por el último. En la Figura 1.5 se muestra el resultado de las divisiones y el último cociente y el orden en el que deben colocarse.

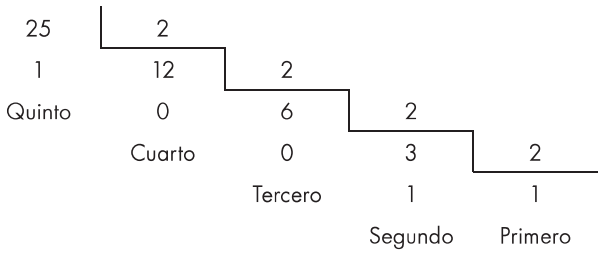


Fig. 1.5. *Conversión del número 25 a binario.*

De esta manera, **el número decimal 25** será el **11 001 en el sistema binario.**

## B. Conversión de una fracción decimal a binario

La forma más sencilla para convertir una fracción decimal a binario consiste en multiplicar sucesivamente la parte fraccionaria por 2 hasta que dé 0 como resultado. La parte entera de cada multiplicación formará los bits del número binario (véase el Caso práctico 5).

A veces, puede ocurrir que la parte fraccionaria no desaparece; es decir, no sale 0. En estos casos se realizan varias multiplicaciones hasta tener los suficientes dígitos que permitan no sobrepasar un determinado error. Por ejemplo, si se desea un error inferior a $2^{-10}$ (0,0000000002), calcularemos hasta 10 dígitos (véase el Caso práctico 6).

### Caso práctico 4

**Expresar un número decimal en sistema binario.**

Vamos a pasar a binario el número decimal 54. Para ello:

1. Calculamos el número de dígitos **N** necesarios para representar 54. El número 54 es mayor que $2^5 = 32$, pero es menor que $2^6 = 64$; entonces, con **seis dígitos binarios** podremos representar el **número decimal 54.**
2. Podremos realizar una tabla tal como la 1.4, con los seis dígitos, y luego sumar los pesos donde hay un 1, como muestra la Tabla 1.6.

| Pesos asociados | | | | | | | |
|---|---|---|---|---|---|---|---|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| | | 1 | 1 | 0 | 1 | 1 | 0 |
| | | 32 + | 16 + | 0 + | 4 + | 2 + | 0 → 54 |

Tabla 1.6. *Conversión a binario de 54 mediante divisiones sucesivas.*

3. O bien realizamos divisiones sucesivas del número 54 por 2 hasta llegar a un cociente menor de 2 (véase la Figura 1.6).

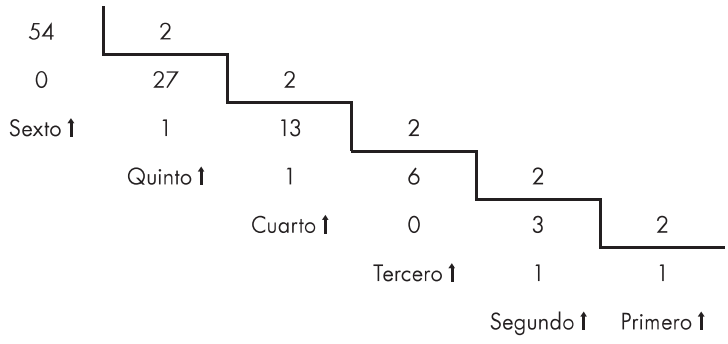


Fig. 1.6. *Conversión a binario de 54 mediante divisiones sucesivas.*

Por tanto, el número decimal 54 se representa en código binario como **110110.** Lo escribimos así:

$$\mathbf{54_{(10} \rightarrow 110110_{(2}}$$

**Errores frecuentes**

En el método de las divisiones sucesivas, no olvides comenzar por el último cociente.

**Caso práctico 5**

**Conversión de decimal a binario de una fracción.**

Vamos a convertir a base 2 $12{,}125_{(10}$:

1. Parte entera: sumamos los pesos; para ello nos fijamos en la Tabla 1.6:
   $12_{(10} \rightarrow 1100_{(2}$
2. Parte fraccionaria:
   $0{,}\mathbf{125} \cdot 2 = 0{,}250 \rightarrow \mathbf{0}$ (el primer dígito es 0; la nueva parte fraccionaria es 0,250).
   $0{,}\mathbf{250} \cdot 2 = 0{,}500 \rightarrow \mathbf{0}$ (el segundo dígito es 0; la nueva parte fraccionaria es 0,500).
   $0{,}\mathbf{500} \cdot 2 = 1{,}000 \rightarrow \mathbf{1}$ (el tercer dígito es 1; como la parte fraccionaria es 0, finaliza la conversión).
3. Resultado: $\mathbf{12{,}125_{(10} \rightarrow 1100{,}001_{(2}}$

**Caso práctico 6**

**Conversión de decimal a binario de una fracción, con un error mínimo.**

Vamos a convertir $0{,}6_{(10}$ a base 2, con un error inferior a $2^{-7}$.

Pasos:

1. Parte fraccionaria:
   $0{,}6 \cdot 2 = 1{,}2 \rightarrow \mathbf{1}$
   $0{,}2 \cdot 2 = 0{,}4 \rightarrow \mathbf{0}$
   $0{,}4 \cdot 2 = 0{,}8 \rightarrow \mathbf{0}$
   $0{,}8 \cdot 2 = 1{,}6 \rightarrow \mathbf{1}$
   $0{,}6 \cdot 2 = 1{,}2 \rightarrow \mathbf{1}$
   $0{,}2 \cdot 2 = 0{,}4 \rightarrow \mathbf{0}$
   $0{,}4 \cdot 2 = 0{,}8 \rightarrow \mathbf{0}$
2. Resultado: $\mathbf{0{,}6_{(10} \rightarrow 0{,}1001100_{(2}}$

## C. Conversión de una fracción binaria a decimal

Para esta conversión se utiliza el teorema fundamental de la numeración. El resultado es la suma de los productos de los resultados de multiplicar cada dígito por la base elevado a la posición que ocupa pero en negativo (véase el Caso práctico 7).

**Caso práctico 7**

**Conversión de una fracción binaria a decimal.**

Vamos a convertir $110{,}0011_{(2}$ a base 10:

1. Parte entera: sumamos los pesos; para ello nos fijamos en la Tabla 1.6:
   $\mathbf{110_{(2} \rightarrow 6_{(10}}$
2. Parte fraccionaria:
   $(0 \cdot 2^{-1}) + (0 \cdot 2^{-2}) + (1 \cdot 2^{-3}) + (1 \cdot 2^{-4}) \rightarrow$
   $\rightarrow (0 \cdot 0{,}5) + (0 \cdot 0{,}25) + (1 \cdot 0{,}125) + (1 \cdot 0{,}0625) \rightarrow 0{,}1875$
3. Resultado: $\mathbf{110{,}0011_{(2} \rightarrow 6{,}1875_{(10}}$

### Actividades

**4. Expresa estas cantidades en código binario:**

*a)* $75_{(10}$ *c)* $345_{(10}$

*b)* $129_{(10}$ *d)* $1590_{(10}$

**5. Expresa estas cantidades en código binario, con un error inferior a $2^{-6}$:**

*a)* $123{,}75_{(10}$ *c)* $4{,}234_{(10}$

*b)* $7{,}33_{(10}$ *d)* $15{,}91_{(10}$

**6. Expresa estas cantidades en código decimal:**

*a)* $111{,}011_{(2}$ *c)* $110110{,}11001_{(2}$

*b)* $11100{,}101_{(2}$

**7. Completa la información que falta en la Tabla 1.7.**

| Pesos | | | | | | | | | | | | | Número |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | , | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 | , | 0,5 | 0,25 | 0,125 | 0,0625 | 0,03125 | |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | , | 1 | 1 | 0 | 1 | | |
| | | 1 | 1 | 0 | 1 | 1 | , | 1 | 0 | 1 | | | |
| | 1 | 1 | 0 | 0 | 1 | 1 | , | 0 | 0 | 1 | 1 | 1 | |

Tabla 1.7. *Convertir a decimal sumando pesos.*

## D. Suma y resta en binario

Al igual que con el sistema decimal, en el sistema binario podemos realizar las operaciones aritméticas: suma, resta, multiplicación y división. La suma binaria es parecida a la suma en decimal, con la diferencia de que se manejan solo dos dígitos, el 0 y el 1. Si el resultado de la suma excede de 1, se agrega un acarreo a la suma parcial siguiente. Para realizar sumas nos fijaremos en la tabla de sumar (véase la Tabla 1.8) y para realizar restas nos fijaremos en la tabla de restar (véase la Tabla 1.9).

| Suma binaria |
|---|
| Suma binaria |
| 0 + 0 = 0 |
| 0 + 1 = 1 |
| 1 + 0 = 1 |
| 1 + 1 = 0, acarreo 1 |

Tabla 1.8. *Tabla para la suma binaria.*

| Resta binaria |
|---|
| 0 – 0 = 0 |
| 0 – 1 = 1, acarreo 1, que se suma al siguiente sustraendo |
| 1 – 0 = 1 |
| 1 – 1 = 0 |

Tabla 1.9. *Tabla para la resta binaria.*

### Caso práctico 8

Sumas y restas en binario, con acarreo, sin acarreo y con decimales.

*a)* Suma sin acarreos:

```
        1 0 0 0 0 → 16
+       1 0 1 0 0 1 → 41
        1 1 1 0 0 1 → 57
```

*b)* Suma con acarreos:

```
            1 1 1      Acarreos
            ↓ ↓ ↓
    1 0 1 0 1 1 1 → 87
+   1 0 0 0 0 1 → 33
    1 1 1 1 0 0 0 → 120
```

*(Continúa)*

## Caso práctico 8

*(Continuación)*

Cuando nos encontramos con tres unos, la suma da 1 y de acarreo 1.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | 1 | 1 | 1 | | Acarreos |
| | ↓ | | | ↓ | ↓ | ↓ | | |
| | | 1 | 1 | 0 | 1 | 1 | 1 → | 55 |
| + | | 1 | 0 | 0 | 0 | 1 | 1 → | 35 |
| | 1 | 0 | 1 | 1 | 0 | 1 | 0 → | 90 |

*c)* Sumas con decimales:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | | 1 | | Acarreos |
| | ↓ | ↓ | ↓ | ↓ | | ↓ | | |
| | | 1 | 1 | 0 | , | 1 | 1 → | 6,75 |
| + | | 1 | 0 | 1 | , | 0 | 1 → | 5,25 |
| | 1 | 1 | 0 | 0 | , | 0 | 0 → | 12,00 |

*d)* Resta sin acarreos:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 0 | 1 | 0 | 1 → | 117 |
| + | | 1 | 0 | 0 | 0 | 0 | 1 → | 33 |
| | 1 | 0 | 1 | 0 | 1 | 0 | 0 → | 84 |

*e)* Resta con acarreos:

- Cuando nos encontramos con el primer 0 – 1, el resultado es 1 y nos llevamos 1, que sumaremos al siguiente sustraendo.
- Si al sumar nos volvemos a llevar 1 (caso de sumar 1 de acarreo + 1 en sustraendo), ese 1 pasa al siguiente sustraendo, y así sucesivamente hasta que dé 0.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 0 | 1 | 0 | 1 → | 117 |
| – | ↓1 | ↓1 | ↓1 | | ↓1 | | | Acarreos |
| | | 1 | 1 | 1 | 0 | 1 | 0 → | 58 |
| | 0 | 1 | 1 | 1 | 0 | 1 | 1 → | 59 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 0 | 0 | 1 | 0 | 1 → | 101 |
| – | | ↓1 | ↓1 | | ↓1 | | | Acarreos |
| | | | 1 | 1 | 0 | 1 | 1 → | 27 |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 → | 74 |

*f)* Resta con decimales:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 1 | , | 0 | 1 → | 17,25 |
| – | ↓1 | ↓1 | ↓1 | ↓1 | ↓1 | | | | Acarreos |
| | | 1 | 0 | 1 | 1 | , | 1 | 1 → | 11,75 |
| | 0 | 0 | 1 | 0 | 1 | | 1 | 0 → | 5,5 |

## Actividades

**8. Realiza las siguientes operaciones binarias**

*a)* 111000 + 100010

*b)* 101010 + 101101

*c)* 111010 – 111001

*d)* 110100 – 101

*e)* 1011,111 – 0,01

*f)* 11001,1101 – 1110,01

## E. Multiplicación binaria

Se realiza como en la multiplicación decimal, con la diferencia de que luego se hacen las sumas en binario. Para los productos, utilizaremos la Tabla 1.10.

| Multiplicación binaria |
|---|
| 0 · 0 = 0 |
| 0 · 1 = 0 |
| 1 · 0 = 0 |
| 1 · 1 = 1 |

**Tabla 1.10.** *Tabla para la multiplicación binaria.*

Si en la suma de una multiplicación nos juntamos con cuatro 1 en una columna, primero sumamos 1 + 1 = 0, y me llevo 1 para la siguiente suma de la siguiente columna; continuamos sumando 1 + 1 = 0, y me vuelvo a llevar 1 para sumar a la siguiente columna, con lo que el resultado será 0 y me llevo dos 1, que se sumarán con los elementos de la columna siguiente.

Véase el Caso práctico 9.

## F. División binaria

Se efectúa como en la división decimal, pero las multiplicaciones y las restas internas se hacen en binario.

Véase el Caso práctico 9.

### Caso práctico 9

**Multiplicaciones y divisiones en binario.**

*a)* Multiplicar 25 (11001) por 5 (101).

```
          1  1  0  0  1   → 25
       ×        1  0  1   →  5
    ---------------------
          1  1  0  0  1
+    0  0  0  0  0
   1  1  0  0  1
-------------------------
1  1  1  1  1  0  1   →125
```

*b)* Multiplicar 23 (10111) por 14 (1110). Nos encontramos con columnas en las que hay que sumar cuatro 1:

```
             1  0  1  1  1   → 23
          ×     1  1  1  0   → 14
       ------------------
             0  0  0  0  0
+       1  0  1  1  1
      1  0  1  1  1
   1  0  1  1  1
-----------------------------
1  0  1  0  0  0  0  1  0   →322
```

*(Continúa)*

### Actividades

**9. Realiza sumas binarias de las siguientes cantidades dadas en decimal:**

*a)* 25 + 21

*b)* 15,125 + 16,75

*c)* 47 + 15

**10. Realiza las siguientes operaciones binarias:**

*a)* 1100010100 – 110101

*b)* 1101010,1101 – 1010,001

*c)* 110110 · 1010

*d)* 10001001 / 1010

*e)* 10001000100 / 101010

### Caso práctico 9

*(Continuación)*

*c)* Dividir 10 (1010) entre 2 (10):

```
1 0 1 0 | 1 0
0 0     | 1 0 1     Cociente → 5
  1 0
  0 0               Resto → 0
```

*d)* Dividir 59 (111011) entre 5 (101):

```
1 1 1 0 1 1 | 1 0 1
0 1 0 0 1   | 1 0 1 1   Cociente → 11
  1 0 0 1
    1 0 0             Resto → 4
```

Para comprobar si la división es correcta, multiplicamos el divisor (5) por el cociente (11) y sumamos el resto (4); esto en binario:

```
          1 0 1 1     Cociente → 11
    ×     1 0 1       Divisor → 5
    -------------
          1 0 1 1
  + 0 0 0 0
  1 0 1 1
  -------------
  1 1 0 1 1 1         → 55
  +       1 0 0       Resto → 4
  -------------
  1 1 1 0 1 1         Dividendo → 59
```

*e)* Dividir 282 (100011010) entre 10 (1010):

```
1 0 0 0 1 1 0 1 0 | 1 0 1 0
0 0 1 1 1 1       | 1 1 1 0 0   Cociente → 28
  0 1 0 1 0
    0 0 0 0 1 0               Resto → 2
```

## 2.3. El sistema octal

Los primeros sistemas informáticos utilizaban solo el sistema binario para interpretar y transformar los datos, con lo que las labores de programación eran bastante tediosas; se recurrió entonces al uso de sistemas intermedios que permitían una fácil traducción hacia y desde el sistema binario. Estos sistemas son el *octal* y el *hexadecimal*.

El **sistema octal** tiene como base de numeración 8, es decir, utiliza ocho símbolos para representar las cantidades. Estos símbolos son 0, 1, 2, 3, 4, 5, 6, 7.

Para convertir de decimal a octal, y viceversa, procederemos como en el sistema binario:

- **Conversión de un número decimal a octal.** Lo más sencillo son las divisiones sucesivas. En la Figura 1.7 se convierte a octal el número 925.

| 925 | 8 | | |
|---|---|---|---|
| 920 | 115 | 8 | |
| 45 | 35 | 14 | 8 |
| 5 | 3 | 6 | 1 |
| Cuarto ↑ | Tercero ↑ | Segundo ↑ | Primero ↑ |

**Fig. 1.7.** *Conversión a octal de 925 mediante divisiones sucesivas.*

- Para **convertir un número octal a decimal,** emplearemos el teorema fundamental de la numeración. Nos podremos guiar por los pesos asociados a cada dígito dependiendo de su posición. En la Tabla 1.11 se muestra la cantidad 1635 en octal. Para pasar a decimal, multiplicamos el dígito por la base elevada a su posición:

| Pesos asociados en el sistema octal | | | |
|---|---|---|---|
| $8^3$ | $8^2$ | $8^1$ | $8^0$ |
| 512 | 64 | 8 | 1 |
| 1 | 6 | 3 | 5 |

Tabla 1.11. *Pesos asociados en el sistema octal.*

$$(1 \cdot \mathbf{8^3}) + (6 \cdot \mathbf{8^2}) + (3 \cdot \mathbf{8^1}) + (5 \cdot \mathbf{8^0}) \longrightarrow (1 \cdot \mathbf{512}) + (6 \cdot \mathbf{64}) + (3 \cdot \mathbf{8}) + (5 \cdot \mathbf{1}) \longrightarrow \mathbf{925}$$

$$\mathbf{1635_{(8} \longrightarrow 925_{(10}}$$

- **Conversión de una fracción decimal a octal.** Se procede como en el sistema binario, con el método de multiplicaciones sucesivas, lo único que cambia es la base (véase el Caso práctico 10).
- **Conversión de una fracción octal a decimal.** Se procede a realizar esta conversión aplicando el teorema fundamental de la numeración: cada dígito tiene un peso según la posición que ocupe. El primer dígito de la parte fraccionaria se multiplica por la base elevada a –1; el segundo por la base elevada a –2, y así sucesivamente (véase el Caso práctico 11).

### Caso práctico 10

**Conversión a octal de una fracción decimal.**

Vamos a convertir a octal el número $12{,}0625_{(10}$.

1. La parte entera se calcula por divisiones (véase la Figura 1.7):

$$\mathbf{12_{(10} \longrightarrow 14_{(8}}$$

2. Para la parte fraccionaria, realizamos multiplicaciones sucesivas por 8, quedándonos con la parte entera y multiplicando por la fraccionaria, hasta que dé 0. Si las fracciones no llegan a 0, se realizan varias multiplicaciones hasta tener los suficientes dígitos que permitan no sobrepasar un determinado error:

$$0{,}0625 \cdot 8 = \mathbf{0},5 \longrightarrow 0{,}5 \cdot 8 = \mathbf{4},0$$

3. Resultado: $\mathbf{12{,}0625_{(10} \longrightarrow 14{,}04_{(8}}$

### Caso práctico 11

**Conversión a decimal de una fracción octal.**

Vamos a convertir a decimal el número $11{,}3016_{(8}$.

1. En primer lugar, hacemos los cálculos:

$$(1 \cdot 8^1) + (1 \cdot 8^0) + (\mathbf{3} \cdot 8^{-1}) + (\mathbf{0} \cdot 8^{-2}) + (\mathbf{1} \cdot 8^{-3}) + (\mathbf{6} \cdot 8^{-4}) \longrightarrow$$

$$\longrightarrow 8 + 1 + 3/8 + 0 + 1/512 + 6/4096 \longrightarrow$$

$$\longrightarrow 8 + 1 + 0{,}375 + 0 + 0{,}001953125 + 0{,}00146484375 \longrightarrow 9{,}37841796875$$

2. Resultado: $\mathbf{11{,}0316_{(8} \longrightarrow 9{,}37841796875_{(10}}$

### Actividades

**11. Convierte a octal los siguientes números decimales:**

*a)* 28,25 *c)* 5,125

*b)* 15,75 *d)* 6,33

**12. Convierte a decimal los siguientes números octales:**

*a)* 13,5763 *c)* 3,7701

*b)* 25,6625 *d)* 7,6543

## 2.4. El sistema hexadecimal

El **sistema hexadecimal** tiene como base de numeración 16, es decir, utiliza dieciséis símbolos para representar las cantidades. Estos símbolos son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

A los símbolos A, B, C, D, E y F se les asignan los valores que se muestran en la Tabla 1.12.

| Símbolo | Valor asignado |
|---|---|
| A | 10 |
| B | 11 |
| C | 12 |
| D | 13 |
| E | 14 |
| F | 15 |

Tabla 1.12. *Sistema hexadecimal: valores asignados a los símbolos A, B, C, D, E y F.*

Para convertir de hexadecimal a decimal, y viceversa, procederemos como en los casos anteriores.

- **Conversión de un número decimal a hexadecimal.** Se realizan divisiones sucesivas, y para los restos entre 10 y 15 utilizamos las letras correspondientes, como se muestra en la Tabla 1.8. En la Figura 1.8 se convierte el número 41 565 a hexadecimal.

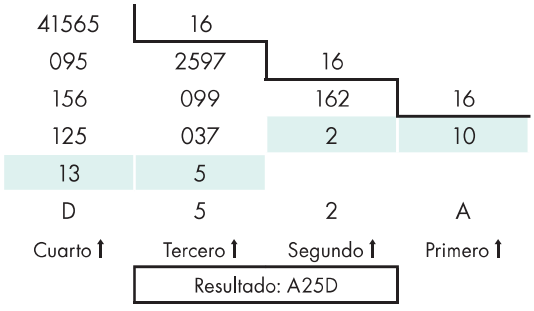


Fig. 1.8. *Conversión a hexadecimal de 41 565.*

- Para **convertir un número hexadecimal a decimal,** utilizaremos el teorema fundamental de la numeración. En la Tabla 1.13 se muestran los pesos asociados por cada posición. Para pasar la cantidad $A1D_{(16}$ a decimal, multiplicamos el dígito por la base elevada a su posición:

$$(A \cdot 256) + (1 \cdot 16) + (D \cdot 1) = (10 \cdot 256) + 16 + 13 = 2\,560 + 16 + 13 = 2\,589$$

$$\mathbf{A1D_{(16} = 2589_{(10}}$$

- **Conversión de una fracción decimal a hexadecimal.** Se procede como en los casos anteriores (véase el Caso práctico 12).

| Pesos asociados en el sistema hexadecimal | | | |
|---|---|---|---|
| $16^3$ | $16^2$ | $16^1$ | $16^0$ |
| 4096 | 256 | 16 | 1 |
| | A | 1 | D |

Tabla 1.13. *Pesos para la conversión hexadecimal-decimal.*

**¿Sabías que...?**

El sistema hexadecimal actual fue introducido en el ámbito de la computación por primera vez en 1963 por IBM.

### Caso práctico 12

**Conversión a hexadecimal de una fracción decimal.**

Vamos a convertir a hexadecimal el número $28{,}1975_{(10}$.

1. La parte entera: $28_{(10} \rightarrow 1C_{(16}$
2. Para la parte decimal, realizamos multiplicaciones sucesivas por 16:
   $0{,}1975 \cdot 16 = \mathbf{3},16$
   $0{,}16 \cdot 16 = \mathbf{2},56$
   $0{,}56 \cdot 16 = \mathbf{8},96$
   $0{,}96 \cdot 16 = \mathbf{15},36$
   $0{,}36 \cdot 16 = \mathbf{5},76$
   $0{,}76 \cdot 16 = \mathbf{12},16$
   $0{,}16 \cdot 16 = \mathbf{2,56}$, se repite de nuevo.
3. Resultado: $\mathbf{12{,}1975_{(10} \rightarrow 1C{,}328F5C28F5C2\ldots_{(8}}$

### Actividades

**13. Expresa en código decimal estas cantidades en octal:**

*a)* $123{,}6_{(8}$ *c)* $265{,}021_{(8}$

*b)* $27{,}34_{(8}$

**14. Expresa estas cantidades en decimal a código octal:**

*a)* $91{,}23_{(10}$ *c)* $459{,}901_{(10}$

*b)* $28{,}32_{(10}$

**15. Expresa en decimal:**

*a)* $F03{,}E_{(16}$ *c)* $2C5{,}02A_{(16}$

*b)* $2F{,}3C_{(16}$

**16. Expresa en hexadecimal:**

*a)* $123{,}8_{(10}$ *c)* $978{,}105_{(10}$

*b)* $98{,}32_{(10}$

- **Conversión de una fracción hexadecimal a decimal.** Se procede como en los casos anteriores, aplicando el teorema fundamental de la numeración.

### Caso práctico 13

**Conversión a decimal de una fracción hexadecimal.**

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

1. Realizamos los cálculos:

$$(1 \cdot 16^2) + (A \cdot 16^1) + (F \cdot 16^0) + (3 \cdot 16^{-1}) + (A \cdot 16^{-2}) =$$
$$= 256 + 160 + 15 + 0{,}1875 + 0{,}0390625 = 431{,}2265625$$

2. Resultado: $\mathbf{1AF{,}3A_{(16} \rightarrow 431{,}2265625\ _{(10}}$

## 2.5. Conversiones entre sistemas

De la misma manera que convertimos del sistema decimal al binario, octal y hexadecimal, y viceversa. También podemos convertir del binario al octal y hexadecimal y del hexadecimal al octal, etc. (véase la Tabla 1.14).

| DEC | BIN | OCT | HEX |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |
| 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 |
| 29 | 11101 | 35 | 1D |
| 30 | 11110 | 36 | 1E |

**Tabla 1.14.** *Equivalencias entre sistemas decimal, binario, octal y hexadecimal.*

### A. Conversión hexadecimal-binario

Se sustituye cada dígito hexadecimal (0, 1, 2, …, D, E, F) por su representación binaria utilizando cuatro dígitos; así, el 0 se representa por 0000, el 1 por 0001, el 2 por 0010, etc. Se utilizan cuatro dígitos porque el valor más alto de este código, el 15, que se representa con la F, necesita cuatro dígitos: 1111 (véase el Caso práctico 14).

**Caso práctico 14**

**Conversión de hexadecimal a binario.**

Pasar a binario $73B,F1_{(16}$:

$$\underset{\downarrow}{7} - \underset{\downarrow}{3} - \underset{\downarrow}{B} , \underset{\downarrow}{F} - \underset{\downarrow}{1}$$
$$0111 \quad 0011 \quad 1011 , 1111 \quad 0001$$

$$\mathbf{73B,F1_{(16} \rightarrow 11100111011,11110001_{(2}}$$

### B. Conversión binario-hexadecimal

Se agrupan los dígitos binarios de cuatro en cuatro a partir del punto decimal hacia la izquierda y hacia la derecha, y se sustituye cada grupo de cuatro por su valor correspondiente en hexadecimal (véase el Caso práctico 15).

**Caso práctico 15**

Pasar a hexadecimal $101011011_{(2}$:

$$\underset{\downarrow}{0001} - \underset{\downarrow}{0101} - \underset{\downarrow}{1011}$$
$$1 \quad 5 \quad B$$

$$\mathbf{101011011_{(2} \rightarrow 15B_{(16}}$$

### C. Conversión octal-binario

Procedemos como en la conversión hexadecimal-binario; se sustituye cada dígito octal por su representación binaria utilizando tres dígitos binarios. Se utilizan tres porque el valor más alto, el 7, necesita tres dígitos binarios: 111 (véase el Caso práctico 16).

**Caso práctico 16**

Pasar a binario $527_{(8}$:

$$\underset{\downarrow}{5} - \underset{\downarrow}{2} - \underset{\downarrow}{7}$$
$$101 \quad 010 \quad 111$$

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

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

$$\underset{\downarrow}{7} - \underset{\downarrow}{1} - \underset{\downarrow}{2} , \underset{\downarrow}{4} - \underset{\downarrow}{6}$$
$$111 \quad 001 \quad 010 , 100 \quad 110$$

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

## D. Conversión binario-octal

Se agrupan los dígitos de tres en tres a partir del punto decimal hacia la izquierda y hacia la derecha, sustituyendo cada grupo de tres por su equivalente en octal (véase el Caso práctico 17).

### Caso práctico 17

Pasar a octal $10101100_{(2}$:

| 010 | 101 | 100 |
|---|---|---|
| ↓ | ↓ | ↓ |
| 2 | 5 | 4 |

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

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

| 001 | 110 | 110 | , | 110 | 011 | 100 |
|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | | ↓ | ↓ | ↓ |
| 1 | 6 | 6 | | 6 | 3 | 4 |

**$1110110,1100111_{(2} \rightarrow 166,634_{(8}$**

## E. Conversión hexadecimal-octal

En esta conversión se realiza un paso intermedio; primero se pasa de hexadecimal a binario y luego de binario a octal (véase el Caso práctico 18).

### Caso práctico 18

Pasar $1AB0C,1B2_{(16}$ a octal.

- Convertir a binario $1AB0C,1B2_{(16}$:

| 1 | A | B | 0 | C | , | 1 | B | 2 |
|---|---|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ | | ↓ | ↓ | ↓ |
| 0001 | 1010 | 1011 | 0000 | 1100 | , | 0001 | 1011 | 0010 |

- Convertir a octal $11010101100001100, 000110110010_{(2}$

| 011 | 010 | 101 | 100 | 001 | 100 | , | 000 | 110 | 110 | 010 |
|---|---|---|---|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | | ↓ | ↓ | ↓ | ↓ |
| 3 | 2 | 5 | 4 | 1 | 4 | , | 0 | 6 | 6 | 2 |

**$1AB0C,1B2_{(16} \rightarrow 325414,0662_{(8}$**

## F. Conversión octal-hexadecimal

Se realiza como la anterior, pero en este caso, primero se pasa de octal a binario y luego de binario a hexadecimal (véase el Caso práctico 19).

### Caso práctico 19

Pasar $3710,142_{(8}$ a hexadecimal.

**1.** Convertir a binario $3710,142_{(8}$:

| 3 | 7 | 1 | 0 | , | 1 | 4 | 2 |
|---|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | | ↓ | ↓ | ↓ |
| 011 | 111 | 001 | 000 | , | 001 | 100 | 010 |

**2.** Convertir a hexadecimal $11111001000,001100010_{(2}$:

| 111 | 1100 | 1000 | , | 0011 | 0001 |
|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | | ↓ | ↓ |
| 7 | C | 8 | , | 3 | 1 |

**$3710,142_{(8} \rightarrow 7C8,31_{(16}$**

### Actividades

**17. Convierte a hexadecimal:**

*a)* $703,16_{(8}$

*b)* $1227,32_{(8}$

*c)* $2C5,02A_{(8}$

**18. Convierte a octal:**

*a)* $C127,B_{(16}$

*b)* $9A,53F2_{(16}$

*c)* $74,10D_{(16}$

## 3. Representación interna de la información

**Importante**

Ya hemos visto que en informática se utiliza el sistema binario, solo se manejan las cifras cero y uno (0 y 1), los ordenadores trabajan internamente con dos niveles de voltaje: «apagado» (0) y «encendido» (1), por lo que su sistema de numeración natural es el sistema binario.

El **bit** es la unidad mínima de información; con él podemos representar dos valores cualesquiera, como verdadero o falso, abierto o cerrado, blanco o negro, norte o sur, rojo o azul... Basta con asignar uno de esos valores al estado de «apagado» (0) y el otro al estado de «encendido» (1).

Cuando se almacena la información no se trabaja a nivel de bit, sino que se trabaja a nivel de carácter (letra, número o signo de puntuación), que ocupa lo que se denomina un **byte,** que a su vez está compuesto de 8 **bits**. El ordenador trabaja con agrupaciones de bits fáciles de manipular y suelen ser múltiplos de 2, la base del sistema binario. Los tamaños más comunes son:

- **Octeto, carácter** o **byte:** es la agrupación de 8 bits, el tamaño típico de información; con él se puede codificar el alfabeto completo (ASCII estándar).
- **Palabra:** tamaño de información manejada en paralelo por los componentes del sistema, como la memoria, los registros o los buses. Son comunes las palabras de 8, 32, 64, 128 y 256 bits: 1 byte, 4, 8, 16, 32 bytes. A mayor tamaño de palabra, mayor es la precisión y la potencia de cálculo del ordenador.

Así, cuando decimos que un archivo de texto ocupa 5000 bytes, queremos decir que contiene el equivalente a 5000 letras o caracteres (entre dos y tres páginas de texto sin formato).

Lo normal es utilizar los múltiplos del byte: el kilobyte (KB), el megabyte (MB), el gigabyte (GB), etc.

En informática se utilizan las potencias de 2 ($2^3$, $2^{10}$, $2^{20}$...) para representar las medidas de la información; sin embargo se ha extendido el uso de las potencias de 10 (uso decimal), debido a que se ha impuesto el uso del *Sistema Internacional de Medidas* (SI), o sistema métrico. Así pues, el primer término de medida que se utilizó fue el **kilobyte** (KB), y se eligió este porque $2^{10}$ es aproximadamente 1 000, que se asocia con el kilo (1 000 gramos); en realidad debería ser 1 024 bytes, ya que $2^{10}$ son 1 024.

La Tabla 1.15 muestra las unidades de medida de información más utilizadas, tanto en su uso decimal como en su uso binario:

| Nombre (símbolo) | Sistema Internacional de Unidades (SI) Estándar (uso decimal) | Prefijo binario (uso binario) | Nombre (símbolo) |
|---|---|---|---|
| Kilobyte (KB) | $1000^1 = 10^3$ bytes | $1024^1 = 2^{10}$ bytes | Kibibyte (kib) |
| Megabyte (MB) | $1000^2 = 10^6$ bytes | $1024^2 = 2^{20}$ bytes | Mebibyte (Mib) |
| Gigabyte (GB) | $1000^3 = 10^9$ bytes | $1024^3 = 2^{30}$ bytes | Gibibyte (Gib) |
| Terabyte (TB) | $1000^4 = 10^{12}$ bytes | $1024^4 = 2^{40}$ bytes | Tebibyte(Tib) |
| Petabyte (PB) | $1000^5 = 10^{15}$ bytes | $1024^5 = 2^{50}$ bytes | Pebibyte (Pib) |
| Exabyte (EB) | $1000^6 = 10^{18}$ bytes | $1024^6 = 2^{60}$ bytes | Exbibyte (Eib) |
| Zettabyte (ZB) | $1000^7 = 10^{21}$ bytes | $1024^7 = 2^{70}$ bytes | Zebibyte (Zib) |
| Yottabyte (YB) | $1000^8 = 10^{24}$ bytes | $1024^8 = 2^{80}$ bytes | Yobibyte (Yib) |

Tabla 1.15. *Unidades de medida de información en decimal y en binario.*

El **megabyte** (MB). Equivale a $10^6$ (1 000 000 bytes) o $2^{20}$ (1 048 576 bytes), según el contexto. Es el conjunto de 1 024 kilobytes, $2^{20}$ bytes → $2^{10} \cdot 2^{10} = 1\,024 \cdot 1\,024$ → → 1 048 576; también podemos decir un millón de bytes $10^6$.

Un **gigabyte** (GB) equivale a $2^{30}$ bytes o $10^9$ bytes, según el uso. Es la unidad que más se usa actualmente para especificar la capacidad de la memoria RAM, de las memorias de tarjetas gráficas, de los CD-ROM o el tamaño de los programas y de los archivos grandes. La capacidad de almacenamiento se mide habitualmente en gigabytes, es decir, en miles de megabytes. Un GB es el conjunto de 1 024 megabytes, $2^{30}$ bytes, o lo que es lo mismo, $2^{10} \cdot 2^{10} \cdot 2^{10} = 1\,024 \cdot 1\,024 \cdot 1\,024 \rightarrow 1\,073\,741\,824$; mil millones de bytes $10^9$.

**Actividades**

**19. Expresa las medidas *zettabyte* y *yottabyte*, desglosadas en las medidas inferiores, tanto en su uso binario como en el decimal.**

## 3.1. Representación de datos alfabéticos y alfanuméricos

Ya hemos visto cómo se almacenan las cantidades numéricas dentro del ordenador; ahora nos toca ver cómo se almacena el resto de caracteres que forman el alfabeto.

Los códigos de E/S permitirán traducir la información o los datos que nosotros podemos entender a una representación que la máquina puede interpretar y procesar. Los datos llegan y salen del ordenador a través de los periféricos de entrada y de salida, respectivamente. Cada fabricante de componentes de E/S podría asignar una combinación diferente al mismo símbolo de origen (por ejemplo, las letras del alfabeto); sin embargo, esto no sería nada positivo en un mercado abierto como el informático. Por eso se tiende a la estandarización de códigos, que ha llevado a la universalización de unos pocos códigos de E/S, como el BCD, EBCDIC, ASCII y Unicode. La mayoría de estos códigos representan cada carácter por medio de un byte (8 bits). Sin duda, el más importante de todos estos es el ASCII.

### A. ASCII

El **Código Estadounidense Estándar para el Intercambio de Información**, o ASCII *(American Standard Code for Information Interchange)*, es la recomendación X3.4-1977 del Instituto Estadounidense de Normas Nacionales (ANSI). Utiliza grupos de 7 bits por carácter, permitiendo $2^7 \rightarrow 128$ caracteres diferentes, lo que es suficiente para el alfabeto en letras mayúsculas y minúsculas y los símbolos de una máquina de escribir corriente, además de algunas combinaciones reservadas para su uso interno. El código *ASCII extendido* usa 8 bits por carácter, lo que añade otros 128 caracteres posibles. Este juego de códigos más amplio permite que se agreguen los símbolos de lenguajes extranjeros y varios símbolos gráficos (véanse las tablas 1.16 y 1.17 en la página siguiente).

### B. Unicode

El **Unicode Standard** es una norma de codificación universal de caracteres que se emplea en los ordenadores bajo Windows NT y en los navegadores Internet Explorer y Netscape a partir de su versión 4. Su uso se está extendiendo. Utiliza 16 bits, lo que permite codificar todos los caracteres de cualquier lenguaje, hasta 65 536.

La versión 3 de Unicode tiene 49194 caracteres de los utilizados en los lenguajes más importantes del mundo. El objetivo de Unicode es representar cada elemento usado en la escritura de cualquier idioma del planeta. Los idiomas actuales más importantes del mundo pueden escribirse con Unicode, incluyendo su puntuación, símbolos especiales, símbolos matemáticos y técnicos, formas geométricas, caracteres gráficos y modelos de Braille.

**¿Sabías que...?**

ASCII también se conoce como la ISO 8859-1 y es el utilizado por los sistemas operativos MS-DOS, Windows y UNIX. En las Tablas 1.16 y 1.17 pueden verse el código ASCII y el ASCII extendido, sin los caracteres de control, que son los primeros 32 caracteres, del 0 al 31.

| Código decimal | Carácter ASCII | Código decimal | Carácter ASCII | Código decimal | Carácter ASCII | Código decimal | Carácter ASCII | Código decimal | Carácter ASCII | Código decimal | Carácter ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 33 | ! | 49 | 1 | 65 | A | 81 | Q | 97 | a | 113 | q |
| 34 | " | 50 | 2 | 66 | B | 82 | R | 98 | b | 114 | r |
| 35 | # | 51 | 3 | 67 | C | 83 | S | 99 | c | 115 | s |
| 36 | $ | 52 | 4 | 68 | D | 84 | T | 100 | d | 116 | t |
| 37 | % | 53 | 5 | 69 | E | 85 | U | 101 | e | 117 | u |
| 38 | & | 54 | 6 | 70 | F | 86 | V | 102 | f | 118 | v |
| 39 | ' | 55 | 7 | 71 | G | 87 | W | 103 | g | 119 | w |
| 40 | ( | 56 | 8 | 72 | H | 88 | X | 104 | h | 120 | x |
| 41 | ) | 57 | 9 | 73 | I | 89 | Y | 105 | i | 121 | y |
| 42 | * | 58 | : | 74 | J | 90 | Z | 106 | j | 122 | z |
| 43 | + | 59 | ; | 75 | K | 91 | [ | 107 | k | 123 | { |
| 44 | , | 60 | < | 76 | L | 92 | \ | 108 | l | 124 | \| |
| 45 | - | 61 | = | 77 | M | 93 | ] | 109 | m | 125 | } |
| 46 | . | 62 | > | 78 | N | 94 | ^ | 110 | n | 126 | ~ |
| 47 | / | 63 | ? | 79 | O | 95 | _ | 111 | o | 127 | _ |
| 48 | 0 | 64 | @ | 80 | P | 96 | ` | 112 | p | | |

**Tabla 1.16.** *Código Standard ASCII (caracteres alfanuméricos).*

| Cód. | Carácter. | Cód. | Carácter. | Cód. | Carácter. | Cód. | Carácter. | Cód. | Carácter. | Cód. | Carácter. | Cód. | Carácter. | Cód. | Carácter. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | € | 144 | | 160 | | 176 | ° | 192 | À | 208 | Ð | 224 | à | 240 | ð |
| 129 | | 145 | ‘ | 161 | ¡ | 177 | ± | 193 | Á | 209 | Ñ | 225 | á | 241 | ñ |
| 130 | ‚ | 146 | ’ | 162 | ¢ | 178 | ² | 194 | Â | 210 | Ò | 226 | â | 242 | ò |
| 131 | ƒ | 147 | “ | 163 | £ | 179 | ³ | 195 | Ã | 211 | Ó | 227 | ã | 243 | ó |
| 132 | „ | 148 | ” | 164 | ¤ | 180 | ´ | 196 | Ä | 212 | Ô | 228 | ä | 244 | ô |
| 133 | … | 149 | • | 165 | ¥ | 181 | µ | 197 | Å | 213 | Õ | 229 | å | 245 | õ |
| 134 | † | 150 | – | 166 | ¦ | 182 | ¶ | 198 | Æ | 214 | Ö | 230 | æ | 246 | ö |
| 135 | ‡ | 151 | — | 167 | § | 183 | · | 199 | ç | 215 | × | 231 | ç | 247 | ÷ |
| 136 | ˆ | 152 | ˜ | 168 | ¨ | 184 | ¸ | 200 | È | 216 | Ø | 232 | è | 248 | ø |
| 137 | ‰ | 153 | ™ | 169 | © | 185 | ¹ | 201 | É | 217 | Ù | 233 | é | 249 | ù |
| 138 | Š | 154 | š | 170 | ª | 186 | V | 202 | Ê | 218 | Ú | 234 | ê | 250 | ú |
| 139 | ‹ | 155 | | 171 | « | 187 | » | 203 | Ë | 219 | Û | 235 | ë | 251 | û |
| 140 | Œ | 156 | œ | 172 | ¬ | 188 | ¼ | 204 | Ì | 220 | Ü | 236 | ì | 252 | ü |
| 141 | | 157 | | 173 | | 189 | ½ | 205 | Í | 221 | Ý | 237 | í | 253 | ý |
| 142 | Ž | 158 | ž | 174 | ® | 190 | ¾ | 206 | Î | 222 | Þ | 238 | î | 254 | þ |
| 143 | | 159 | Ÿ | 175 | ¯ | 191 | ¿ | 207 | Ï | 223 | ß | 239 | ï | 255 | ÿ |

**Tabla 1.17.** *Código Standard ASCII extendido (caracteres alfanuméricos).*

Unicode proporciona un número único para cada carácter, sin importar la plataforma, sin importar el programa, sin importar el idioma. Líderes de la industria tales como Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys y muchos otros han adoptado la norma Unicode. Unicode es un requisito para los estándares modernos tales como XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc., y es la manera oficial de aplicar la norma ISO/IEC 10646. Es compatible con numerosos sistemas operativos, con todos los exploradores actuales y con muchos otros productos. La aparición de la norma Unicode y la disponibilidad de herramientas que la respaldan se encuentran entre las más recientes e importantes tendencias en tecnología de software.

La incorporación de Unicode en sitios web y en aplicaciones de cliente-servidor o de múltiples niveles permite disminuir ostensiblemente los costos del uso de juegos de caracteres heredados. Unicode permite que un producto de software o sitio web específico se oriente a múltiples plataformas, idiomas y países, sin necesidad de rediseñarlo. Además, permite que los datos se trasladen a través de gran cantidad de sistemas distintos sin sufrir daños.

Básicamente, las computadoras solo trabajan con números. Almacenan letras y otros caracteres mediante la asignación de un número a cada uno. Antes de que se inventara Unicode, existían cientos de sistemas de codificación distintos para asignar estos números. Ninguna codificación específica podía contener caracteres suficientes; por ejemplo, la Unión Europea, por sí sola, necesita varios sistemas de codificación distintos para cubrir todos sus idiomas. Incluso para un solo idioma como el inglés no había un único sistema de codificación que se adecuara a todas las letras, signos de puntuación y símbolos técnicos de uso común.

## C. BCD y EBCDIC

BCD, que significa **decimal codificado en binario** *(Binary Coded Decimal)*, en realidad no es un código de E/S, sino una forma de codificar los símbolos numéricos del 0 al 9 que se emplean en varios códigos de E/S, entre los que figuran EBCDIC y ASCII.

BCD divide cada octeto en dos mitades o cuartetos, cada uno de los cuales almacena en binario una cifra. Con este código es muy fácil convertir del binario al sistema decimal.

El EBCDIC, o **código BCD extendido de caracteres decimales codificados en binario para el intercambio de información** (EBCDIC, *Extended BDC Interchange Code*), es un sistema de codificación que tiene como objetivo la representación de caracteres alfanuméricos. Es el utilizado por la empresa IBM para sus ordenadores de la serie IBM PC (miniordenadores y *mainframes)*.

En este sistema de codificación, cada carácter tiene 8 bits. Al tener ocho, podremos representar hasta $2^8 \rightarrow 256$ caracteres. Será posible almacenar letras mayúsculas, minúsculas, caracteres especiales, caracteres de control para dispositivos de E/S y para comunicaciones.

**Actividades**

**20. Consultando las tablas de los códigos ASCII y EBCDIC, representa el nombre del centro en el que cursáis los estudios, cada carácter es un byte. Ponlo en hexadecimal y en binario (consulta en Internet para averiguar la equivalencia de números y letras del código EBCDIC).**

**21. Busca en Internet los códigos de colores HTML. Observa como varían los códigos RRGGBB.**

**¿Sabías que...?**

Para definir los colores en las páginas web se utilizan tres pares de números hexadecimales que representan la combinación de los tres colores primarios: rojo, verde y azul (paleta de colores RGB **R**ed-**G**reen-**Bl**ue).

La sintaxis para codificar un color en HTML es la siguiente **color=«#RRGGBB».**

Donde RR, GG y BB representan un número hexadecimal para cada color entre 00 y FF (0 a 255 en decimal) en el orden rojo, verde y azul.

Así por ejemplo el color rojo se representa por **«#FF0000»**, es decir «255» rojo, «0» verde, y «0» azul. El verde sería **«#00FF00»**, y el azul **«#0000FF»**.

El blanco se representa por **«#FFFFFF»** y el negro por **«#000000»**.

# Síntesis

**Transmisión de información** entre el ser humano y el ordenador

- Comunicación
  - Emisor = Ser humano
  - Receptor = Ordenador
  - Medio = Periféricos

La información debe ser traducida o ***codificada,*** ya que los códigos utilizados por el emisor, el canal y el receptor son diferentes.

Un sistema de numeración es un conjunto de dígitos y reglas que permiten representar datos numéricos. La **principal regla** es que un *mismo dígito tiene distinto valor según la posición que ocupe.*

El **sistema** de numeración **binario** es el que utilizan los ordenadores para almacenar la información, los circuitos digitales internos que componen los ordenadores utilizan este sistema para la interpretación de la información y codificación de la misma. Su base es 2, y cada dígito de un número representado por este sistema se denomina **bit (bi**nary digi**t).**

Otros sistemas, como el **octal** (base 8) y el **hexadecimal** (base 16), también son utilizados por los ordenadores.

**Conversión directa octal/binario**

| **Octal** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **Binario** | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

**Conversión directa hexadecimal/binario**

| **Hexadecimal** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **Binario** | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| **Hexadecimal** | 8 | 9 | A | B | C | D | E | F |
| **Binario** | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

Las unidades de medidas de la información son:

| | |
|---|---|
| Kilobyte (KB) | $1000^1 = 10^3$ bytes |
| Megabyte (MB) | $1000^2 = 10^6$ bytes |
| Gigabyte (GB) | $1000^3 = 10^9$ bytes |
| Terabyte (TB) | $1000^4 = 10^{12}$ bytes |
| Petabyte (PB) | $1000^5 = 10^{15}$ bytes |
| Exabyte (EB) | $1000^6 = 10^{18}$ bytes |
| Zettabyte (ZB) | $1000^7 = 10^{21}$ bytes |
| Yottabyte (YB) | $1000^8 = 10^{24}$ bytes |

Para almacenar los caracteres que forman el alfabeto se utilizan los **códigos de E/S** que traducen la información o los datos que nosotros podemos entender a una representación que la máquina puede interpretar y procesar. Los códigos estandarizados que se utilizan son el BCD, EBCDIC, ASCII y Unicode.

Actualmente, el Unicode (Unicode Standard) es el más extendido, se usa en los ordenadores bajo Windows y en los navegadores Internet Explorer y Netscape a partir de su versión 4. Utiliza 16 bits, lo que permite codificar todos los caracteres de cualquier lenguaje, hasta 65 536.

# Test de repaso

**1. Indica cuál de los siguientes números no está codificado en octal**

*a)* 12345,678.

*b)* 234,001.

*c)* 2347,0011.

*d)* 3221,02.

**2. Si el ancho de palabra es de 10 bits, ¿cuántos números podremos representar?**

*a)* 100.

*b)* 1000.

*c)* 1024.

*d)* 10.

**3. ¿Cuántos dígitos binarios necesito para representar el número 43?**

*a)* 5.

*b)* 6.

*c)* 4.

*d)* 7.

**4. ¿Cuántos bytes tienen tres gigabytes?**

*a)* Tres millones de bytes.

*b)* Tres mil millones de bytes.

*c)* Tres mil kilobytes.

*d)* Trescientos millones de bytes.

**5. El número 36 en octal se representa en binario a:**

*a)* 00110110.

*b)* 11001001.

*c)* 011110.

*d)* 100001.

**6. Para representar caracteres alfabéticos y alfanuméricos, utilizaremos el código:**

*a)* ANSI.

*b)* Binario.

*c)* ASCII.

*d)* IEEE754.

**7. El código EBCDIC es el utilizado por:**

*a)* Los ordenadores IBM de la serie IBM PC.

*b)* Los ordenadores bajo Windows NT.

*c)* Los equipos de la marca Compaq.

*d)* Los navegadores de Internet.

**8. De los siguientes códigos, ¿cuál es el que utiliza la mayoría de los navegadores de Internet?**

*a)* EBCDIC.

*b)* BCD.

*c)* Unicode.

*d)* ASCII.

**9. ¿Cuántos bits tienen 12 kB?**

*a)* $12 \cdot 1\,024 \rightarrow 12\,288$ bits.

*b)* $12 \cdot 1\,024 \cdot 8 \rightarrow 98\,304$ bits.

*c)* $12 \cdot 1\,000 \rightarrow 12\,000$ bits.

*d)* $12 \cdot 1\,000 \cdot 8 \rightarrow 9\,600$ bits.

**10. El número decimal 34 se representa en binario como:**

*a)* 100100.

*b)* 100010.

*c)* 100001.

*d)* 100011.

**11. El número binario 1101 equivale al número decimal:**

*a)* 23.

*b)* 14.

*c)* 15.

*d)* 13.

**Soluciones:** 1a; 2c; 3b; 4b; 5c; 6c; 7a; 8c; 9c; 10b.

# Comprueba tu aprendizaje

## I. Sistemas de numeración

**1.** Expresa la cantidad según el teorema fundamental de la numeración.

- 234,765.
- 347,21.
- 800,102.

**2.** Representa en el sistema decimal los siguientes números en distintas bases:

- $123{,}45_{(6}$.
- $4300{,}012_{(5}$.
- $1101{,}0011_{(2}$.

**3.** Convierte a binario:

- $178{,}2_{(8}$.
- $29{,}3125_{(10}$.
- $A{,}B2_{(16}$.

**4.** Convierte a hexadecimal:

- $110010{,}1101_{(2}$.
- $56{,}375_{(10}$.
- $156{,}22_{(8}$.

**5.** Convierte a octal:

- $9A{,}53F2_{(16}$.
- $29{,}3125_{(10}$.
- $1101110{,}01001_{(2}$.

## II. Operaciones en binario

**6.** Realiza las siguientes sumas en binario:

- 11111111 + 1.
- 1011,101 + 101,110.
- 11001,11 + 10,1.

**7.** Efectúa las siguientes restas en binario:

- 11111111 – 1.
- 1011,101 – 101,110.
- 11001,11 – 10,1.

**8.** Realiza las siguientes multiplicaciones en binario:

- 1011,01 · 101.
- 111 · 100.
- 11001,11 · 10,1.

**9.** Realiza las siguientes divisiones en binario:

- 101011 / 110.
- 110110110 / 1110.
- 11001,11 / 10,1.

## III. Códigos alfanuméricos utilizados por los ordenadores

**10.** Codifica en ASCII y EBCDIC las palabras:

- Instalación.
- Mantenimiento.

## IV. Medidas de almacenamiento de la información en el ordenador

**11.** Expresa en bytes las siguientes cantidades:

- 25 YB.
- 15 ZB.
- 20 PB.