

UD1 :: Explotación de sistemas microinformáticos

0. Relación con el currículo

Resultados de aprendizaje y criterios de evaluación

RA1. Evalúa sistemas informáticos, e identifica sus componentes y sus características.

- CA1.1. Reconoce los componentes físicos de un sistema informático y sus mecanismos de interconexión.
- CA1.2. Se describió y se realizó el proceso de puesta en marcha de un equipo.
- CA1.3. Se ensamblaron equipos microinformáticos, con interpretación de la documentación técnica, y se verificó el resultado final.
- CA1.4. Se establecieron los parámetros de configuración (hardware y software) de un equipo microinformático.
- CA1.5. Se clasificaron, instalaron y configuraron dispositivos periféricos de diversos tipos.

Contenidos básicos

- Componentes de un sistema informático: chasis, alimentación y refrigeración; placas base, procesadores y memorias; dispositivos de almacenamiento; controladoras.
- Periféricos. Adaptadores para la conexión de dispositivos.
- Mecanismos y técnicas de interconexión.
- Secuencia de arranque de un equipamiento.
- Normas de seguridad y prevención de riesgos laborales en el taller informático.
- Manejo de documentación técnica en los idiomas más empleados por la industria.
- Webs de soporte técnico.
- Configuración y verificación de equipos.

1. Introducción

La palabra **informática** proviene de la contracción de los vocablos INFORmación y autoMÁTICA. El diccionario de la Real Academia Española la define como "el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores".

Básicamente, la informática permite el tratamiento de información, y para ello necesita representarla internamente. Derivado de los sistemas eléctricos y electrónicos, la base de representación de la información es binario, y a partir del **binario**, el octal, menos utilizado, y el **hexadecimal**.

La representación de información en una **base n** establece que los valores que se pueden utilizar son desde el **0** hasta **n-1**.

Se utiliza el sistema numérico posicional, de manera que para obtener el valor que se representa, se realiza una suma ponderada de los valores de cada posición multiplicada por la base elevada a la posición menos 1.

La base natural de los humanos es la base 10, lo que quiere decir que los dígitos que se pueden utilizar para la representación son 0,1,2,3,4,5,6,7,8,9. Para representar el valor setemil trescientos ochenta y dos sería :

$$7 \cdot 10^3 + 3 \cdot 10^2 + 8 \cdot 10^1 + 2 \cdot 10^0 = 7382$$

Representación en binario.

Para representar valores numéricos en binario, la base de representación es el 2. Ésto quiere decir que los dígitos que se utilizan son el 0 y el 1. Todos los valores se representan únicamente con ceros y unos.

Dos procesos, obtención del valor representado, que se mostrará en base 10, y por otra parte, la representación en binario de un valor numérico dado en base 10.

Obtención del valor representado en binario.

$$1011001_2 \\ 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 89_{10}$$

Y para obtener la representación en binario de un valor representado en decimal, es proceso se basa en la división entera entre dos hasta que ya no se pueda seguir dividiendo, cociente cero, y presentarlo utilizando los restos de la división, pero en sentido contrario al que se obtuvieron.

$$\begin{array}{r} 89 \mid 2 \\ 1 \quad 44 \mid 2 \\ \quad 0 \quad 22 \mid 2 \\ \qquad 0 \quad 11 \mid 2 \\ \qquad \quad 1 \quad 5 \mid 2 \\ \qquad \qquad 1 \quad 2 \mid 2 \\ \qquad \qquad \quad 0 \quad 1 \mid 2 \\ \qquad \qquad \qquad 1 \quad 0 \end{array}$$

$89_{10} = 1011001_2$

Representación de octal

La representación en octal es menos utilizada, pero el modus operandi es exactamente el mismo. Al ser base 8, los dígitos que se pueden utilizar son 0, 1, 2, 3, 4, 5, 6, 7.

$$371_8 = 3 \cdot 8^2 + 7 \cdot 8^1 + 1 \cdot 8^0 = 3 \cdot 64 + 7 \cdot 8 + 1 \cdot 1 = 192 + 56 + 1 = 249_{10}$$

Y para obtener el valor en octal.

$$\begin{array}{r} 249 \mid 8 \\ 1 \quad 31 \mid 8 \\ \quad 7 \quad 3 \mid 8 \\ \qquad 3 \quad 0 \end{array}$$

$$249_{10} = 371_8$$

Representación en hexadecimal.

La representación en hexadecimal, base 16, es muy usada, por ejemplo, para representar el contenido de algunos archivos. Presenta la particularidad de que se necesitan 16 símbolos o dígitos para representar la información, y para representar los símbolos a partir del 9, se utilizan las letras mayúsculas en orden, con lo cual los símbolos o dígitos serían : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

$$B0CA_{16} = 11 \cdot 16^3 + 0 \cdot 16^2 + 12 \cdot 16^1 + 10 \cdot 16^0 = 45056 + 0 + 192 + 10 = 45258_{10}$$

Y para pasar de decimal a hexadecimal.

$$\begin{array}{r}
 45258 \quad | \quad \underline{16} \\
 10 \quad 2828 \quad | \quad \underline{16} \\
 12 \quad 176 \quad | \quad \underline{16} \\
 0 \quad 11 \quad | \quad \underline{16} \\
 11 \quad 0
 \end{array}$$

$45258_{10} = B0CA_{16}$

Todo ésto se utiliza para información numérica, pero en un sistema informático es necesario manejar información textual, por lo menos.

Unidades de información

La información que se maneja en un sistema informático se representa internamente con valores en binario, con lo que la unidad de información más elemental tiene dos posibles valores, 0 y 1. Se denomina **bit**.

Los bits se agrupan en lo que se denomina palabras, que por convención se ha determinado que sean de 8 bits, y a esta palabra se la denomina **byte**. De facto es la unidad de información más pequeña que se maneja.

El byte, evidentemente, tiene múltiplos, que se presentan en la siguiente tabla.

| Unidad | Nombre | Equivalencia | Sistema | Notación |
|--------|------------|--------------|---------|---|
| 1KB | 1 Kilobyte | 1000 bytes | Decimal | SI [Sistema Internacional] |
| 1KiB | 1 Kibibyte | 1024 bytes | Binario | IEC [International Electrotechnical Commission] |

Por encima, el resto de múltiplos son los siguientes.

| SI | | IEC | |
|-------------------|--------------|--------------------|--------------|
| Unidad | Equivalencia | Unidad | Equivalencia |
| 1 Megabyte [1MB] | 1000KB | 1 Mebibyte [1MiB] | 1024KiB |
| 1 Gigabyte [1B] | 1000GB | 1 Gibibyte [1iB] | 1024GiB |
| 1 Terabyte [1B] | 1000TB | 1 Tebibyte [1iB] | 1024TiB |
| 1 Petabyte [1B] | 1000PB | 1 Pebibyte [1iB] | 1024PiB |
| 1 Exabyte [1B] | 1000EB | 1 Exbibyte [1iB] | 1024EiB |
| 1 Zetabyte [1B] | 1000ZB | 1 Zebibyte [1iB] | 1024ZiB |
| 1 Yottabyte [1B] | 1000YB | 1 Yobibyte [1iB] | 1024YiB |
| 1 Brontobyte [1B] | 1000BB | 1 Bronbibyte [1iB] | 1024BiB |

El último de la lista no está reconocido oficialmente por la IEC.

Representación de información textual

La solución que se adoptó en su momento fue establecer una tabla en la que a cada carácter que se representaba visualmente le correspondiese una cadena en binario, que tendría su representación en decimal y hexadecimal.

A esa tabla se la denominó **ASCII** [*American Standard Code for Information Interchange*] que utilizaba 7 bits y posteriormente se extendió a 8. De esta forma, si un sistema compartía información con otro, internamente representaban la información de la misma manera.

| Dec | Hex | Oct | Char | Dec | Hex | Oct | Char | Dec | Hex | Oct | Char | Dec | Hex | Oct | Char |
|-----|-----|-----|------|-----|-----|-----|-------|-----|-----|-----|------|-----|-----|-----|------|
| 0 | 0 | 000 | NUL | 32 | 20 | 040 | Space | 64 | 40 | 100 | @ | 96 | 60 | 140 | ` |
| 1 | 1 | 001 | SOH | 33 | 21 | 041 | ! | 65 | 41 | 101 | A | 97 | 61 | 141 | a |
| 2 | 2 | 002 | STX | 34 | 22 | 042 | \" | 66 | 42 | 102 | B | 98 | 62 | 142 | b |
| 3 | 3 | 003 | ETX | 35 | 23 | 043 | # | 67 | 43 | 103 | C | 99 | 63 | 143 | c |
| 4 | 4 | 004 | EOT | 36 | 24 | 044 | \$ | 68 | 44 | 104 | D | 100 | 64 | 144 | d |
| 5 | 5 | 005 | ENQ | 37 | 25 | 045 | % | 69 | 45 | 105 | E | 101 | 65 | 145 | e |
| 6 | 6 | 006 | ACK | 38 | 26 | 046 | & | 70 | 46 | 106 | F | 102 | 66 | 146 | f |
| 7 | 7 | 007 | BEL | 39 | 27 | 047 | ' | 71 | 47 | 107 | G | 103 | 67 | 147 | g |
| 8 | 8 | 010 | BS | 40 | 28 | 050 | (| 72 | 48 | 110 | H | 104 | 68 | 150 | h |
| 9 | 9 | 011 | TAB | 41 | 29 | 051 |) | 73 | 49 | 111 | I | 105 | 69 | 151 | i |
| 10 | A | 012 | LF | 42 | 2A | 052 | * | 74 | 4A | 112 | J | 106 | 6A | 152 | j |
| 11 | B | 013 | VT | 43 | 2B | 053 | + | 75 | 4B | 113 | K | 107 | 6B | 153 | k |
| 12 | C | 014 | FF | 44 | 2C | 054 | , | 76 | 4C | 114 | L | 108 | 6C | 154 | l |
| 13 | D | 015 | CR | 45 | 2D | 055 | - | 77 | 4D | 115 | M | 109 | 6D | 155 | m |
| 14 | E | 016 | SO | 46 | 2E | 056 | . | 78 | 4E | 116 | N | 110 | 6E | 156 | n |
| 15 | F | 017 | SI | 47 | 2F | 057 | / | 79 | 4F | 117 | O | 111 | 6F | 157 | o |
| 16 | 10 | 020 | DLE | 48 | 30 | 060 | 0 | 80 | 50 | 120 | P | 112 | 70 | 160 | p |
| 17 | 11 | 021 | DC1 | 49 | 31 | 061 | 1 | 81 | 51 | 121 | Q | 113 | 71 | 161 | q |
| 18 | 12 | 022 | DC2 | 50 | 32 | 062 | 2 | 82 | 52 | 122 | R | 114 | 72 | 162 | r |
| 19 | 13 | 023 | DC3 | 51 | 33 | 063 | 3 | 83 | 53 | 123 | S | 115 | 73 | 163 | s |
| 20 | 14 | 024 | DC4 | 52 | 34 | 064 | 4 | 84 | 54 | 124 | T | 116 | 74 | 164 | t |
| 21 | 15 | 025 | NAK | 53 | 35 | 065 | 5 | 85 | 55 | 125 | U | 117 | 75 | 165 | u |
| 22 | 16 | 026 | SYN | 54 | 36 | 066 | 6 | 86 | 56 | 126 | V | 118 | 76 | 166 | v |
| 23 | 17 | 027 | ETB | 55 | 37 | 067 | 7 | 87 | 57 | 127 | W | 119 | 77 | 167 | w |
| 24 | 18 | 030 | CAN | 56 | 38 | 070 | 8 | 88 | 58 | 130 | X | 120 | 78 | 170 | x |
| 25 | 19 | 031 | EM | 57 | 39 | 071 | 9 | 89 | 59 | 131 | Y | 121 | 79 | 171 | y |
| 26 | 1A | 032 | SUB | 58 | 3A | 072 | : | 90 | 5A | 132 | Z | 122 | 7A | 172 | z |

| | | | | | | | | | | | | | | | |
|----|----|-----|-----|----|----|-----|---|----|----|-----|---|-----|----|-----|-----|
| 27 | 1B | 033 | ESC | 59 | 3B | 073 | ; | 91 | 5B | 133 | [| 123 | 7B | 173 | { |
| 28 | 1C | 034 | FS | 60 | 3C | 074 | < | 92 | 5C | 134 | \ | 124 | 7C | 174 | |
| 29 | 1D | 035 | GS | 61 | 3D | 075 | = | 93 | 5D | 135 |] | 125 | 7D | 175 | } |
| 30 | 1E | 036 | RS | 62 | 3E | 076 | > | 94 | 5E | 136 | ^ | 126 | 7E | 176 | ~ |
| 31 | 1F | 037 | US | 63 | 3F | 077 | ? | 95 | 5F | 137 | _ | 127 | 7F | 177 | DEL |

Esta tabla, aún en su versión extendida, no daba solución para representar los caracteres necesarios, por ejemplo, para otros alfabetos, con lo que se propusieron otros conjuntos de caracteres o sistemas de codificación. El más utilizado es **UTF-8**, o **UNICODE**. Existen también UTF-16 y UTF-32.

Se podría suponer que UTF-32, UTF-16 y UTF-8 utilizan respectivamente 32, 16 y 8 bits, pero no, UTF-32 en efecto utiliza 32 bits para representar los caracteres.

Sin embargo, UTF-8 tiene longitud variable, de la siguiente manera.

Si el primer bit es un 0, equivale a ASCII o **ISO-LATIN-15**.

Si el primer bit es un 1, los siguientes 1's representan el número de bytes que se utilizan para representar la información.

Lo que se consigue con este sistema es que los caracteres más utilizados ocupen menos espacio.

Ejemplo.

```
01000001_11000011_10000010_11100100_10111000_10101101_11110000_10011111_10001111_10000000
```

01000001 → Comienza con 0 → ASCII → "A"

11000011_10000010 → Comienza con 110 → 2 bytes → "Â"

11100100_10111000_10101101 → Comienza con 1110 → 3 bytes → "中"

11110000_10011111_10001111_10000000 → Comienza con 11110 → 4 bytes → "🏀"