Presentación ***resumen*** del libro:

"EMPEZAR DE CERO A PROGRAMAR EN lenguaje C"

Autor: Carlos Javier Pes Rivas (correo@carlospes.com)

Capítulo 2

REPRESENTACIÓN DE LOS DATOS

carlospes.com

OBJETIVOS

- Entender cómo la computadora digital es capaz de **representar** a distintos tipos de datos.

- Saber realizar **conversiones de base** entre los Sistemas Decimal, Binario, Octal y Hexadecimal.

- Conocer los tipos de **codificación binaria** más utilizados para representar a los **números enteros**, **números reales** y **caracteres**.

  – Las computadoras se utilizan para procesar información (gráficos, sonidos, textos,...). Pero, ¿cómo es capaz la computadora digital de representar a toda esta información con tan solo dos símbolos, el cero (0) y el uno (1)?

CONTENIDO

2.1 INTRODUCCIÓN (1/2)

- **Codificación de la información:**

  – Para que el ordenador pueda manejar la misma información que los humanos (textos, sonidos, imágenes, etc.) hay que realizar una **conversión** de los signos de nuestros lenguajes a ceros y unos (bits).

  – Tanto las instrucciones de los programas como los datos que estos manejan, deben codificarse en bits. A una sucesión de bits se le denomina **código binario** o ***código máquina***.

  – En la memoria principal del ordenador, cada bit se presenta por medio de un transistor. Si se utiliza **lógica positiva**, el transistor estará ***"encendido"*** para simbolizar a un **uno** (1) ó ***"apagado"*** para representar a un **cero** (0).

2.1 INTRODUCCIÓN (2/2)

- **Byte:** Los bits suelen agruparse en bloques de 8. A dicho bloque se le denomina **byte** u *octeto*. Múltiplos de byte son:
  - **Kilobyte** (KB). 1KB = 1024 bytes = 2^{10} bytes
  - **Megabyte** (MB). 1MB = 1024 Kb = 1024^{2} bytes
  - **Gigabyte** (GB). 1GB = 1024 Mb = 1024^{2} Kb = 1024^{3} bytes
  - **Terabyte** (TB). 1TB = 1024 Gb = 1024^{2} Mb = 1024^{3} Kb = 1024^{4} bytes
  - **Petabyte** (PB). 1PB = 1024 Tb = 1024^{2} Gb = ... = 1024^{5} bytes
- **Palabra:** Máximo número de bits con que la CPU puede trabajar en paralelo (a la vez).
  - Suele ser múltiplo de un byte: 8, 16, 32, 64 bits, etc.

2.2 SISTEMAS DE NUMERACIÓN (1/8)

- **Sistema de numeración:** define a un conjunto de signos y reglas para expresar a los números.

- **Sistema de Numeración Romano:**
  - **Símbolos: I** (uno), **V** (cinco), **X** (diez), **L** (cincuenta), **C** (cien), **D** (quinientos) y **M** (mil).
    - CLI = C + L + I = 100 + 50 + 1
    - MMXL = 1000 + 1000 - 10 + 50 = 2040
  - **Inconveniente:** No facilita la realización de cálculos matemáticos por escrito.
  - **No es un sistema de numeración posicional.**

2.2 SISTEMAS DE NUMERACIÓN (2/8)

- **Sistema de Numeración Arábigo o Decimal:**
  - Ingeniado por los hindúes en el siglo III a. C., aprox.
  - Se pueden representar infinitos números reales.
  - **Diez dígitos: 0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8** y **9**.
  - Signos *más* (**+**) y *menos* (**-**), y *punto* (**.**) o *coma* (**,**) para separar la parte entera de la fraccionaria.
    - **Número real = parte entera , parte fraccionaria**
    - **EJEMPLO:** -502,12 = -500 -2 -0,1 -0,02

  - **Sí es un sistema de numeración posicional.**
    - Adecuado para realizar operaciones matemáticas por escrito.

2.2 SISTEMAS DE NUMERACIÓN (3/8)

- **Sistemas de numeración posicional (1/5):**
  - Cada cifra representa a un ***valor relativo*** diferente, dependiendo de su ***valor absoluto*** y de su ***posición*** en una secuencia de números.
  - **EJEMPLO:** 444 = 400 + 40 + 4
  - Un sistema de numeración posicional se caracteriza por su **base**, que viene determinada por el número de dígitos que utiliza.
    - **Sistema Binario (base 2):** 0 y 1.
    - **Sistema Octal (base 8):** 0, 1, 2, 3, 4, 5, 6 y 7.
    - **Sistema Decimal (base 10):** 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.
    - **Sistema Hexadecimal (base 16):** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F.

2.2 SISTEMAS DE NUMERACIÓN (4/8)

- **Sistemas de numeración posicional (2/5):**
  - En cualquier sistema de numeración posicional, una secuencia de dígitos se puede representar, formalmente, de la siguiente manera:

$$\mathbf{N_b = a_{p-1}\ a_{p-2}\ ...\ a_1\ a_0\ ,\ a_{-1}\ a_{-2}\ ...\ a_{-q+1}\ a_{-q}}$$

  - Siendo:
    - (**N**) el número o secuencia de signos.
    - (**b**) la base.
    - (**p**) el número de dígitos de la parte entera.
    - (**q**) el número de dígitos de la parte fraccionaria.
    - ($\mathbf{a_i}$) las cifras del número.
    - (**i**) la posición de cada cifra con respecto a la coma (**,**).
  - Cumpliéndose que para todo dígito **a**,
    - **0** £ **a** £ **b-1**
  - Y para toda posición **i**,
    - **-q** £ **i** £ **p-1**

2.2 SISTEMAS DE NUMERACIÓN (5/8)

- **Sistemas de numeración posicional (3/5):**
  - **EJEMPLO:** En el **Sistema Decimal**, el número real **4305,86** se puede expresar como:
    - $\mathbf{4305,86_{10}}$
  - Siendo:
    - (**N = 4305,86**) el número.
    - (**b = 10**) la base.
    - (**p = 4**) el número de dígitos de la parte entera.
    - (**q = 2**) el número de dígitos de la parte fraccionaria.
    - ($a_3 = 4$, $a_2 = 3$, $a_1 = 0$, $a_0 = 5$, $a_{-1} = 8$ y $a_{-2} = 6$) las cifras del número.
  - Cumpliéndose que para todo dígito **a**,
    - **0** £ **a** £ **9**
  - Y para toda posición **i**,
    - **-2** £ **i** £ **3**

2.2 SISTEMAS DE NUMERACIÓN (6/8)

- **Sistemas de numeración posicional (4/5):**
  - **EJEMPLO:** En el **Sistema Binario**, el número real **11010,001** se puede expresar como:
    - **$11010{,}001_2$**
  - Siendo:
    - (**N = 11010,001**) el número.
    - (**b = 2**) la base.
    - (**p = 5**) el número de dígitos de la parte entera.
    - (**q = 3**) el número de dígitos de la parte fraccionaria.
    - ($a_4 = 1$, $a_3 = 1$, $a_2 = 0$, $a_1 = 1$, $a_0 = 0$, $a_{-1} = 0$, $a_{-2} = 0$ y $a_{-3} = 1$) las cifras del número.
  - Cumpliéndose que para todo dígito **a**,
    - **0** £ **a** £ **1**
  - Y para toda posición **i**,
    - **-3** £ **i** £ **4**

2.2 SISTEMAS DE NUMERACIÓN (7/8)

- **Sistemas de numeración posicional (5/5):**

  – Con **n** dígitos se pueden representar $\mathbf{b^n}$ números diferentes.

  – **EJEMPLO:** Con tres dígitos, en el Sistema Decimal se pueden representar $\mathbf{10^3}$ números enteros positivos distintos, es decir, mil números: del $\mathbf{000_{10}}$ al $\mathbf{999_{10}}$, ambos inclusive.

  – **EJEMPLO:** con tres dígitos
    - **En Binario:** $\mathbf{2^3}$ = 8 números ($\mathbf{000_2}$ ... $\mathbf{111_2}$)
    - **En Octal:** $\mathbf{8^3}$ = 512 números ($\mathbf{000_8}$ ... $\mathbf{777_8}$)
    - **En Hexadecimal:** $\mathbf{16^3}$ = 4096 números ($\mathbf{000_{16}}$ ... $\mathbf{FFF_{16}}$)

2.2 SISTEMAS DE NUMERACIÓN (8/8)

- **Teorema Fundamental de la Numeración (TFN):**

$$N_b = a_{p-1} \cdot b^{p-1} + a_{p-2} \cdot b^{p-2} + ... + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + ... a_{-q+1} \cdot b^{-q+1} + a_{-q} \cdot b^{-q}$$

  - Dígito más significativo (de **mayor peso**): a_{p-1}
  - Dígito menos significativo (de **menor peso**): a_{-q}
  - Fórmula:

$$\boxed{N_b = \sum_{i=-q}^{p-1} a_i \cdot b^i}$$

- **EJEMPLO:**

$4305{,}86_{10} = 4 \cdot 10^3 + 3 \cdot 10^2 + 0 \cdot 10^1 + 5 \cdot 10^0 + 8 \cdot 10^{-1} + 6 \cdot 10^{-2}$

$4305{,}86_{10} = 4000 + 300 + 0 + 5 + 0{,}8 + 0{,}06$

2.3 CONVERSIONES DE BASES (1/6)

- **De base (b) a base 10:** hacemos uso del TFN.

- **EJEMPLOS:**

$10{,}101_2 = 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 2 + 0 + 0{,}5 + 0 + 0{,}125 = 2{,}625_{10}$

$703{,}4_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 3 \cdot 8^0 + 4 \cdot 8^{-1} = 448 + 0 + 3 + 0{,}5 = 451{,}5_{10}$

$6C{,}1_{16} = 6 \cdot 16^1 + C \cdot 16^0 + 1 \cdot 16^{-1} = 96 + 12 + 0{,}0625 = 108{,}0625_{10}$

2.3 CONVERSIONES DE BASES (2/6)

- **De base 10 a base (b):**

  – **PASO 1: Convertir la parte entera** del número $\mathbf{N_{10}}$, dividiéndola, sucesivamente, entre **b**, hasta obtener un cociente más pequeño que **b**.

    - La parte entera del número que estamos buscando lo compondrán el **último cociente** y los **restos** que se hayan ido obteniendo, tomados en orden inverso.

  – **PASO 2: Convertir la parte fraccionaria** del número $\mathbf{N_{10}}$, multiplicándola, repetidamente, por **b**, hasta obtener un cero en la parte fraccionaria o hasta que se considere oportuno.

    - La parte fraccionaria del número buscado lo formarán las **partes enteras** de los números que se hayan ido obteniendo en cada producto, cogidas en ese mismo orden.

2.3 CONVERSIONES DE BASES (3/6)

- **De base 10 a base (b):**

- **EJEMPLO (De base 10 a base 2):** $13{,}3125_{10}$ a base 2

Paso 1: parte entera

$13 \;|\; 2$

$a_0 = 1 \leftarrow \quad \underline{1} \quad 6 \;|\; 2$

$a_1 = 0 \leftarrow \quad \underline{0} \quad 3 \;|\; 2$

$a_2 = 1 \leftarrow \quad \underline{1} \quad \underline{1} \quad \rightarrow a_3 = 1$

$$\mathbf{13_{10} = 1101_2}$$

Paso 2: parte fraccionaria

$0{,}3125 \times 2 = \underline{0}{,}625 \quad \rightarrow a_{-1} = 0$

$0{,}625 \times 2 = \underline{1}{,}25 \quad \rightarrow a_{-2} = 1$

$0{,}25 \times 2 = \underline{0}{,}5 \quad \rightarrow a_{-3} = 0$

$0{,}5 \times 2 = \underline{1} \quad \rightarrow a_{-4} = 1$

$$\mathbf{0{,}3125_{10} = 0{,}0101_2}$$

$$\mathbf{13{,}3125_{10} = 1101{,}0101_2}$$

2.3 CONVERSIONES DE BASES (4/6)

- **De base (b) a base (c), ambas distintas de 10:**
  - **PASO 1:** Convertir el número $\mathbf{N_b}$ de base **(b)** a base **10**.
  - **PASO 2:** Convertir el número $\mathbf{N_{10}}$ de base **10** a base **(c)**.

- **EJEMPLO (De base 8 a base 2):** $16{,}51_8$ a base 2

Paso 1:

$$16{,}51_8 = 1 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} + 1 \cdot 8^{-2} = 8 + 6 + 0{,}625 + 0{,}015625 = 14{,}640625_{10}$$

Paso 2:

$$
\begin{array}{r|l}
14 & 2 \\
\hline
\underline{0} \quad 7 & 2 \\
\hline
\underline{1} \quad 3 & 2 \\
\hline
\underline{1} & \underline{1}
\end{array}
$$

$a_0 = 0 \leftarrow$

$a_1 = 1 \leftarrow$

$a_2 = 1 \leftarrow$

$\rightarrow a_3 = 1$

$0{,}640625 \times 2 =$	$\underline{1}{,}28125$	$\rightarrow a_{-1} = 1$
$0{,}28125 \times 2 =$	$\underline{0}{,}5625$	$\rightarrow a_{-2} = 0$
$0{,}5625 \times 2 =$	$\underline{1}{,}125$	$\rightarrow a_{-3} = 1$
$0{,}125 \times 2 =$	$\underline{0}{,}25$	$\rightarrow a_{-4} = 0$
$0{,}25 \times 2 =$	$\underline{0}{,}5$	$\rightarrow a_{-5} = 0$
$0{,}5 \times 2 =$	$\underline{1}$	$\rightarrow a_{-6} = 1$

$$\mathbf{16{,}51_8 = 14{,}640625_{10} = 1110{,}101001_2}$$

2.3 CONVERSIONES DE BASES (5/6)

- **Conversión directa (Octal/Binario):**

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

- **EJEMPLOS:**

De base 8 a base 2

$\underline{1}\ \underline{6}\ ,\ \underline{5}\ \underline{1}_{8}$

$00\underline{1}\ \underline{110}\ ,\ \underline{101}\ \underline{001}_{2}$

De base 2 a base 8

$0\underline{10}\ \underline{000}\ \underline{110}\ \underline{000}\ ,\ \underline{010}\ \underline{1}00_{2}$

$\underline{2}\ \underline{0}\ \underline{6}\ \underline{0}\ ,\ \underline{2}\ \underline{4}_{8}$

2.3 CONVERSIONES DE BASES (6/6)

- **Conversión directa (Hexadecimal/Binario):**

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

- **EJEMPLO:** De base 16 a base 2

$$1\ B\ ,\ 8_{16}$$

$$0001\ \ 1011\ ,\ 1000_{2}$$

EJERCICIOS RECOMENDADOS

- **Resueltos:** 1, 2, 3, 4, 5, 6, 7, 8 y 9.
- **Propuestos:** 1, 2, 3, 4, 5, 6 y 7.

2.4 ARITMÉTICA BINARIA (1/4)

- **Suma binaria:**

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 0 \text{ y acarreo } 1$$

- **EJEMPLO:** Sumar 10010_2 y 110_2

		1	1			← Acarreos
	1	0	0	1	0	← 1º Sumando
+			1	1	0	← 2º Sumando
	1	1	0	0	0	← Resultado

2.4 ARITMÉTICA BINARIA (2/4)

- **Resta binaria:**

0 - 0 = 0
0 - 1 = 1 y acarreo 1
1 - 0 = 1
1 - 1 = 0

- **EJEMPLO:** Restar 101001_2 y 1011_2

	1	0	1	0	0	1	← Minuendo
			1	0	1	1	← Sustraendo
-	1	1	1	1			← Acarreos
	0	1	1	1	1	0	← Resultado

2.4 ARITMÉTICA BINARIA (3/4)

- **Multiplicación binaria:**

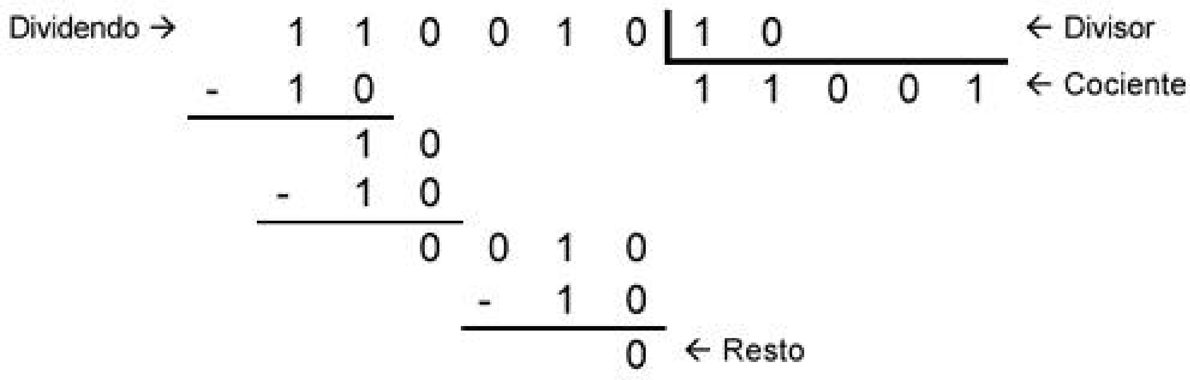
0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

- **EJEMPLO:** Multiplicar 10101_2 por 101_2

			1	0	1	0	1	← Multiplicando
		x			1	0	1	← Multiplicador
			1	0	1	0	1	
		0	0	0	0	0		
+	1	0	1	0	1			
	1	1	0	1	0	0	1	← Resultado

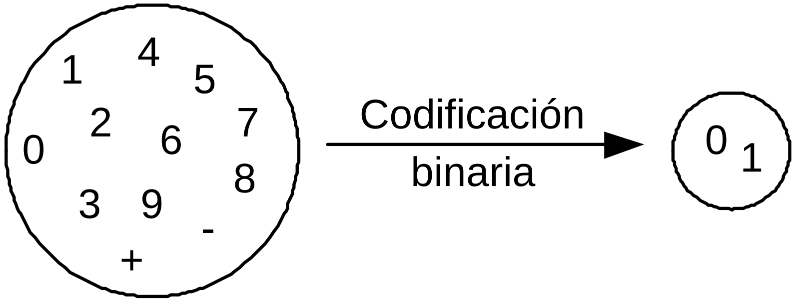
2.4 ARITMÉTICA BINARIA (4/4)

- **División binaria:**
- **EJEMPLO:** Dividir 110010_2 entre 10_2



2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (1/13)

- **Codificación binaria:**



- **Sistemas de representación en Coma Fija (CFI):**
  - Binario Puro
  - Signo Magnitud
  - Complemento a 1
  - Complemento a 2
  - Exceso a 2^{n-1}

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (2/13)

- **Binario Puro (BP):**
  - Sólo pueden ser positivos.
  - **Rango de representación:**

$$\boxed{0_{10} \leq x \leq (2^n - 1)_{10}}$$

- **Fórmula:** De BP a base 10

$$\boxed{N_{BP} = \left(\sum_{i=0}^{n-1} a_i \cdot 2^i \right)_{10}}$$

- **EJEMPLO:** Para n = 8

$$0_{10} \leq x \leq (2^8 - 1)_{10}$$

$$0_{10} \leq x \leq (256 - 1)_{10}$$

$$\mathbf{0_{10} \leq x \leq 255_{10}}$$

- **EJEMPLO:** 23_{10} en BP

	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
23_{10} =	0	0	0	1	0	1	1	1	en BP

- **EJEMPLO:** -68_{10} y 379 no son representables

- **EJEMPLO:** $10110001_{BP} = (1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^0)_{10} = 177_{10}$

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (3/13)

- **Desbordamiento:** Sucede cuando el resultado de una operación está fuera del rango de representación.

- **EJEMPLO:** Para n = 8, al sumar 11001000_{BP} y 11001011_{BP}

	1	1			1					← Acarreos
		1	1	0	0	1	0	0	0	← 200_{10}
+		1	1	0	0	1	0	1	1	← 203_{10}
	1	1	0	0	1	0	0	1	1	← 403_{10}

$200_{10} + 203_{10} = \mathbf{403_{10}}$ (está fuera del rango de representación)

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (4/13)

- Para saber de antemano el número **n** de cifras necesarias para representar en **Binario Puro** a un determinado número **N** del Sistema Decimal, se puede calcular el logaritmo en base 2 del número decimal:

$$n \geq \log_2 N_{10}$$

- **EJEMPLO:** ¿Cuántas cifras se necesitan para representar al número 27_{10} en Binario Puro?

$$\log_2 27_{10} = 4{,}754887502 \cong 5$$

Se necesitan 5 bits ($27_{10} = 11011_{BP}$)

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (5/13)

- **Signo Magnitud (SM):**
  - Números positivos y negativos.

a_{n-1}	a_{n-2}	…	a_1	a_0
bit de signo	bits de la magnitud			

	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
23_{10} =	0	0	0	1	0	1	1	1	en SM
	bit de signo	bits de la magnitud							

	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
-23_{10} =	1	0	0	1	0	1	1	1	en SM
	bit de signo	bits de la magnitud							

- **Rango de representación:**

$$\boxed{(-2^{n-1} + 1)_{10} \leq x \leq (2^{n-1} - 1)_{10}}$$

- **EJEMPLO:** Para n = 8

$$(-2^{8-1} + 1)_{10} \leq x \leq (2^{8-1} - 1)_{10}$$

$$(-2^{7} + 1)_{10} \leq x \leq (2^{7} - 1)_{10}$$

$$(-128 + 1)_{10} \leq x \leq (128 - 1)_{10}$$

$$\mathbf{-127_{10} \leq x \leq 127_{10}}$$

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (6/13)

- **Fórmula:** De SM a base 10

$$\boxed{N_{SM} = \left((1 - 2\cdot a_{n-1}) \cdot \sum_{i=0}^{n-2} a_i \cdot 2^i \right)_{10}}$$

- **EJEMPLOS:**

$$11100001_{SM} = \left((1 - 2\cdot 1) \cdot (1\cdot 2^6 + 1\cdot 2^5 + 1\cdot 2^0) \right)_{10} =$$
$$= \left((1 - 2) \cdot (64 + 32 + 1) \right)_{10} = \left((-1) \cdot (97) \right)_{10} = -97_{10}$$

$$00011010_{SM} = \left((1 - 2\cdot 0) \cdot (1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^1) \right)_{10} =$$
$$= \left((1 - 0) \cdot (16 + 8 + 2) \right)_{10} = \left((1) \cdot (26) \right)_{10} = 26_{10}$$

- **Inconvenientes:**
  - Desbordamiento.
  - El número $0_{10} = 0000000_{SM} = 10000000_{SM}$ (dos representaciones)

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (7/13)

- **Complemento a 1 (C1):**
  - Números positivos igual que en SM o BP:

	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
9503_{10} =	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	1	en C1

  - Números negativos en **Complemento a la Base Menos 1**.
    - $C_{b-1}(N) = b^n - 1 - N$
    - $C_1(N) = 2^n - 1 - N_2 = N_{C1}$

$$C_1(N) = C_1(10010100011111) =$$
$$= 2^n - 1 - N =$$
$$= 2^{16} - 1 - 10010100011111 =$$
$$= 10000000000000000 - 1 - 10010100011111 =$$
$$= 1101101011100000_{C1}$$

	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
-9503_{10} =	1	1	0	1	1	0	1	0	1	1	1	0	0	0	0	0	en C1

  - **Rango de representación**:

$$\boxed{(-2^{n-1} + 1)_{10} \le x \le (2^{n-1} - 1)_{10}}$$

EJEMPLO: Para n = 16

$$(-2^{16-1} + 1)_{10} \le x \le (2^{16-1} - 1)_{10}$$
$$(-2^{15} + 1)_{10} \le x \le (2^{15} - 1)_{10}$$
$$(-32768 + 1)_{10} \le x \le (32768 - 1)_{10}$$
$$\mathbf{-32767_{10} \le x \le 32767_{10}}$$

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (8/13)

- **Fórmulas:** De C1 a base 10 (para números positivos)

$$\boxed{N_{C1}(\text{positivo}) = \left((1 - 2\cdot a_{n-1}) \cdot \sum_{i=0}^{n-2} a_i \cdot 2^i \right)_{10}} \quad \text{o} \quad \boxed{N_{C1}(\text{positivo}) = \left(\sum_{i=0}^{n-1} a_i \cdot 2^i \right)_{10}}$$

- **Fórmula:** De C1 a base 10 (para números negativos)

$$\boxed{N_{C1}(\text{negativo}) = -\left(N_{C1}(\text{positivo}) \right)_{10}}$$

- **EJEMPLO:**

$1111101011111101_{C1}(\text{negativo}) \rightarrow 0000010100000010_{C1}(\text{positivo})$

$0000010100000010_{C1} = (1\cdot 2^{10} + 1\cdot 2^{8} + 1\cdot 2^{1})_{10} = (1024 + 256 + 2)_{10} = 1282_{10}$

- **Inconvenientes:** $1111101011111101_{C1} = -1282_{10}$
  – Desbordamiento.
  – El número $0_{10} = 00000000000_{SM} = 1111111111111111_{SM}$

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (9/13)

- **Complemento a 2 (C2):**
  - Números positivos igual que en C1, SM o BP:

	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
9503_{10} =	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	1	en C2

  - Números negativos en **Complemento a la Base**.
    - $C_{b-1}(N) = b^n - N$
    - $C_1(N) = 2^n - N_2 = N_{C2}$

$C_2(N) = C_2(10010100011111) =$

$= 2^n - N =$

$= 2^{16} - 10010100011111 =$

$= 10000000000000000 - 10010100011111 =$

$= 1101101011100001_{C2}$

	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
-9503_{10} =	1	1	0	1	1	0	1	0	1	1	1	0	0	0	0	1	en C2

  - **Rango de representación**:

$$\boxed{(-2^{n-1})_{10} \leq x \leq (2^{n-1} - 1)_{10}}$$

EJEMPLO: Para n = 16

$$(-2^{16-1})_{10} \leq x \leq (2^{16-1} - 1)_{10}$$

$$(-2^{15})_{10} \leq x \leq (2^{15} - 1)_{10}$$

$$(-32768)_{10} \leq x \leq (32768 - 1)_{10}$$

$$\mathbf{-32768_{10} \leq x \leq 32767_{10}}$$

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (10/13)

- **Fórmulas:** De C2 a base 10 (para números positivos)

$$\boxed{N_{C1}(\text{positivo}) = \left((1 - 2\cdot a_{n-1}) \cdot \sum_{i=0}^{n-2} a_i \cdot 2^i \right)_{10}} \quad \text{o} \quad \boxed{N_{C1}(\text{positivo}) = \left(\sum_{i=0}^{n-1} a_i \cdot 2^i \right)_{10}}$$

- **Fórmula:** De C2 a base 10 (para números negativos)

$$\boxed{N_{C2} = \left(-a_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} a_i \cdot 2^i \right)_{10}}$$

- **EJEMPLO:**

$$1001000000000110_{C2} = \left((-1 \cdot 2^{15}) + (1 \cdot 2^{12} + 1 \cdot 2^{2} + 1 \cdot 2^{1}) \right)_{10} =$$

$$= \left((-32768) + (4102) \right)_{10} = -28666_{10}$$

- **Inconveniente:**
  – Desbordamiento.

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (11/13)

- **Exceso a 2^{n-1}:**
  - Números positivos y negativos:
    - $N_{EX} = N + 2^{n-1}$
  - **Rango de representación**:

$$\boxed{(-2^{n-1})_{10} \leq x \leq (2^{n-1} - 1)_{10}}$$

- **EJEMPLO:** Para n = 16

$$(-2^{16-1})_{10} \leq x \leq (2^{16-1} - 1)_{10}$$

$$(-2^{15})_{10} \leq x \leq (2^{15} - 1)_{10}$$

$$(-32768)_{10} \leq x \leq (32768 - 1)_{10}$$

$$\mathbf{-32768_{10} \leq x \leq 32767_{10}}$$

- **EJEMPLOS:**

$9503_{10} = (\ 9503_{10} + (2^{16-1})_{10}\ )_{EX} =$

$= (\ 9503_{10} + (2^{15})_{10}\ )_{EX} =$

$= (\ 9503_{10} + 32768_{10}\ )_{EX} =$

$= (\ 42271_{10}\ )_{EX} =$

$= (\ 1010010100011111_2\ )_{EX} =$

$= 1010010100011111_{EX.\ a\ 32768}$

$-9503_{10} = (\ -9503_{10} + (2^{16-1})_{10}\ )_{EX} =$

$= (\ -9503_{10} + (2^{15})_{10}\ )_{EX} =$

$= (\ -9503_{10} + 32768_{10}\ )_{EX} =$

$= (\ 23265_{10}\ )_{EX} =$

$= (\ 101101011100001_2\ )_{EX} =$

$= 0101101011100001_{EX.\ a\ 32768}$

2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (12/13)

- **Fórmula:** De Exceso a 2^{n-1} a base 10

$$\boxed{N_{EX} = \left(\left(\sum_{i=0}^{n-1} a_i \cdot 2^i \right) - 2^{n-1} \right)_{10}}$$

- **EJEMPLO:**

$$0000000001001001_{EX.\ a\ 32768} = \left((1\cdot2^6 + 1\cdot2^3 + 1\cdot2^0) - 2^{15} \right)_{10} =$$

$$= \left((64 + 8 + 1) - 32768 \right)_{10} = -32695_{10}$$

- **Inconveniente:**
  – Desbordamiento.

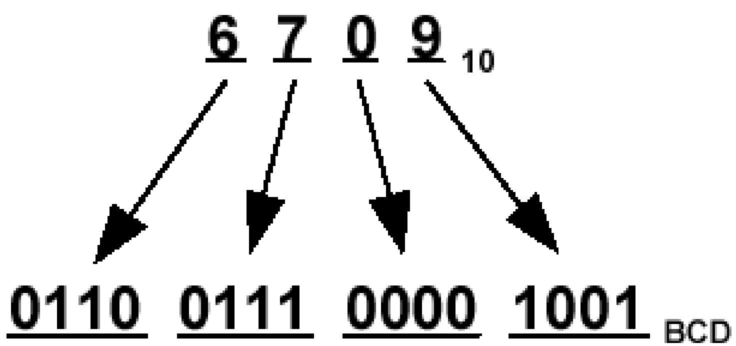
2.5 REPRESENTACIÓN DE NÚMEROS ENTEROS (13/13)

- **BCD (*Binary Coded Decimal*):**

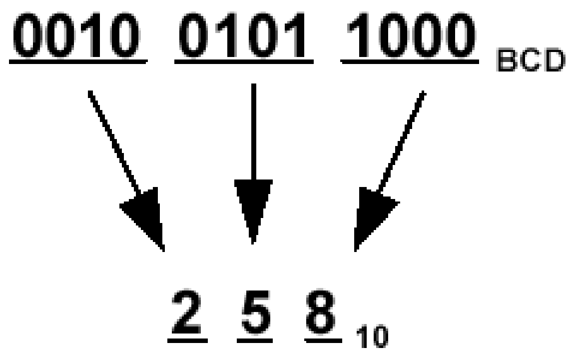
Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

- **EJEMPLOS:**

De base 10 a BCD



De BCD a base 10

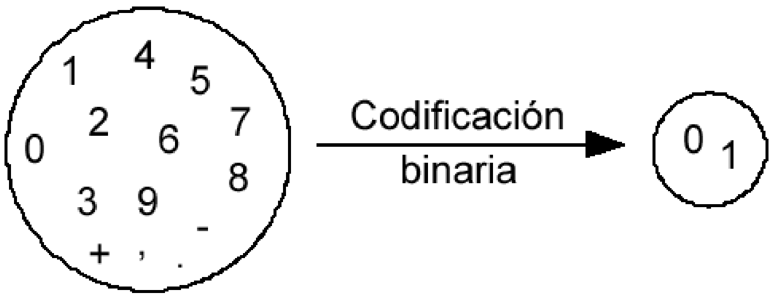


EJERCICIOS RECOMENDADOS

- **Resueltos:** 10, 11, 12, 13 y 14.
- **Propuestos:** 8, 9, 10, 11 y 12.

2.6 REPRESENTACIÓN DE NÚMEROS REALES (1/8)

- **Codificación binaria:**



- **Sistema de representación en Coma Fija Flotante (CFL):**
  - Notación científica o exponencial: $\mathbf{N = m \times b^{exp}}$
  - En las computadoras digitales (***estándar IEEE 754***): $\mathbf{N = m \times 2^{exp}}$

- **EJEMPLOS:**

$41{,}67_{10} = 4{,}167 \times 10^{1} = 0{,}4167 \times 10^{2} = 0{,}04167 \times 10^{3} = 0{,}004167 \times 10^{4} = \ldots$

$41{,}67_{10} = 416{,}7 \times 10^{-1} = 4167 \times 10^{-2} = 41670 \times 10^{-3} = 416700 \times 10^{-4} = \ldots$

2.6 REPRESENTACIÓN DE NÚMEROS REALES (2/8)

- **Normalización:** situar **la coma decimal** de su mantisa en un lugar determinado (a la izquierda o a la derecha del dígito más significativo de la mantisa).
- **EJEMPLOS:**

Sistema Decimal	$41{,}67_{10}$	$-0{,}625_{10}$
Coma decimal a la <u>derecha</u> del dígito más significativo de la mantisa	$\underline{\mathbf{4}},167 \times 10^{1}$	$-\underline{\mathbf{6}},25 \times 10^{-1}$
Coma decimal a la <u>izquierda</u> del dígito más significativo de la mantisa	$0,\underline{\mathbf{4}}167 \times 10^{2}$	$-0,\underline{\mathbf{6}}25 \times 10^{0}$

Sistema Binario	$11{,}01_{2}$	$-0{,}011_{2}$
Coma decimal a la <u>derecha</u> del bit más significativo de la mantisa	$\underline{\mathbf{1}},101 \times 2^{1}$	$-\underline{\mathbf{1}},1 \times 2^{-2}$
Coma decimal a la <u>izquierda</u> del bit más significativo de la mantisa	$0,\underline{\mathbf{1}}101 \times 2^{2}$	$-0,\underline{\mathbf{1}}1 \times 2^{-1}$

2.6 REPRESENTACIÓN DE NÚMEROS REALES (3/8)

- **Estándar IEEE 754 (precisión simple):** 32 bits
  - **<u>Signo</u>** (1 bit). **<u>Mantisa</u>** en Signo Magnitud (23 bits). **<u>Exponente</u>** en Exceso a 2^{n-1}-1 (8 bits).

31	30 … 23	22 … 0
s	exp	m
signo	exponente	mantisa

- **EJEMPLO:** **$-5{,}75_{10}$** $= -101{,}11_2$

  Normalizar: $-101{,}11_2 = -1{,}0111_2 \times 2^2$

  Exponente en Exceso 2^{n-1}-1: $2_{10} + (2^{8-1}-1)_{10} = 129_{10} = 10000001_{\text{EX. A 127}}$

  Mantisa con <u>bit implícito</u>: 01110000000000000000000

31	30 … 23	22 … 0
1	10000001	01110000000000000000000
signo	exponente	mantisa

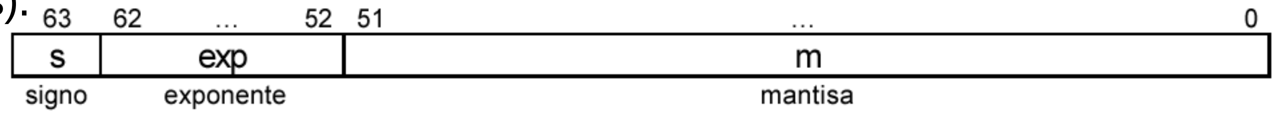
<u>1100</u> <u>0000</u> <u>1011</u> <u>1000</u> <u>0000</u> <u>0000</u> <u>0000</u> <u>0000</u>

<u>C</u> <u>0</u> <u>B</u> <u>8</u> <u>0</u> <u>0</u> <u>0</u> <u>0</u>

$-5{,}75_{10} = -101{,}11_2 = -1{,}0111_2 \times 2^2 = \text{C0B80000}_{\text{CFL(PRECISIÓN SIMPLE)}}$

2.6 REPRESENTACIÓN DE NÚMEROS REALES (4/8)

- **Estándar IEEE 754 (precisión doble):** 64 bits
  - **<u>Signo</u>** (1 bit). **<u>Mantisa</u>** en Signo Magnitud (52 bits). **<u>Exponente</u>** en Exceso a 2^{n-1}-1 (11 bits).

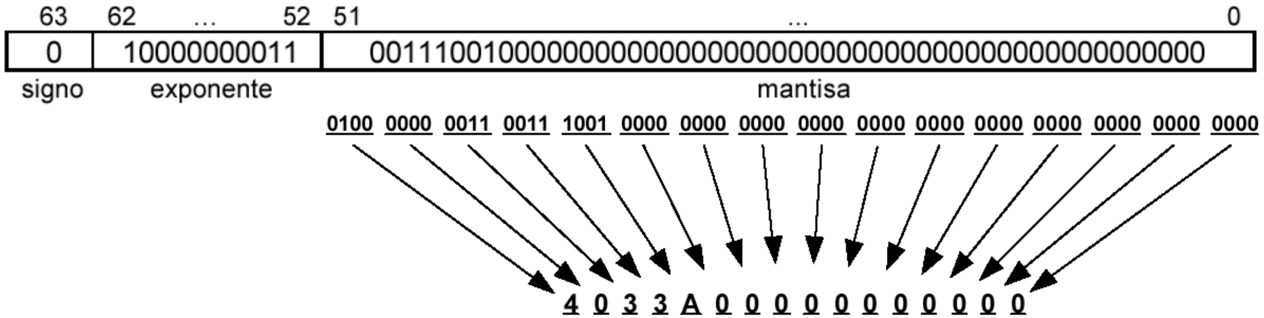


- **EJEMPLO: $19{,}5625_{10}$** = $10011{,}1001_2$

Normalizar: $10011{,}1001_2 = 1{,}00111001_2 \times 2^4$

Exponente en Exceso 2^{n-1}-1: $4_{10} + (2^{11-1}-1)_{10} = 1027_{10} = 10000000011_{EX.\ A\ 1023}$

Mantisa con <u>bit implícito</u>: 0011100100000000000000000000000000000000000000000000



$19{,}5625_{10} = 10011{,}1001_2 = 1{,}00111001_2 \times 2^4 = 4033A00000000000_{CFL(PRECISIÓN\ DOBLE)}$

2.6 REPRESENTACIÓN DE NÚMEROS REALES (5/8)

- **Casos especiales en el estándar IEEE 754 (simple y doble):**

Signo (s)	***Exponente (exp)***	***Mantisa (m)***	***Significado***
Positivo (0)	Todo unos (11…1)	Todo ceros (00...0)	Más infinito (+∞)
Negativo (1)	Todo unos (11…1)	Todo ceros (00...0)	Menos infinito (-∞)
0 ó 1	Todo unos (11…1)	Distinta de todo ceros	No es un número (*Not a Number*, **NaN**)
0 ó 1	Todo ceros (00...0)	Todo ceros (00...0)	Representa al cero (0)
0 ó 1	Todo ceros (00...0)	Distinta de todo ceros	Número muy pequeño cercano al cero (0)

- Si todos los bits del exponente son ceros (00...0), **no** se está utilizando **bit implícito**.
- Si, además, la mantisa es distinta de todo ceros, el número es **muy pequeño**. (Exponente -126 ó -1022).

2.6 REPRESENTACIÓN DE NÚMEROS REALES (6/8)

- **Del estándar IEEE 754 con precisión simple a base 10:**

Signo (s)	*Exponente (exp)*	*Mantisa (m)*	*Valor en base 10*
0 ó 1	0 < **exp** < 255	Indiferente	$(-1)^{\mathbf{s}} \cdot 1,\mathbf{m} \cdot 2^{\mathbf{exp}-127}$
0	**exp** = 255	**m** = 0	$+\infty$
1	**exp** = 255	**m** = 0	$-\infty$
0 ó 1	**exp** = 255	**m** ≠ 0	NaN
0 ó 1	**exp** = 0	**m** = 0	0
0 ó 1	**exp** = 0	**m** ≠ 0	$(-1)^{\mathbf{s}} \cdot 0,\mathbf{m} \cdot 2^{-126}$

- **Rango de representación** en el estándar IEEE 754 con precisión simple:

$$\boxed{(\,(2^{-23} - 2) \cdot 2^{127})_{10} \leq x \leq (\,(2 - 2^{-23}) \cdot 2^{127})_{10}}$$

2.6 REPRESENTACIÓN DE NÚMEROS REALES (7/8)

- **Del estándar IEEE 754 con precisión doble a base 10:**

Signo (s)	***Exponente (exp)***	***Mantisa (m)***	***Valor en base 10***
0 ó 1	0 < **exp** < 2047	Indiferente	$(-1)^{\mathbf{s}} \cdot 1,\mathbf{m} \cdot 2^{\mathbf{exp}-1023}$
0	**exp** = 2047	**m** = 0	$+\infty$
1	**exp** = 2047	**m** = 0	$-\infty$
0 ó 1	**exp** = 2047	**m** $\neq$ 0	NaN
0 ó 1	**exp** = 0	**m** = 0	0
0 ó 1	**exp** = 0	**m** $\neq$ 0	$(-1)^{\mathbf{s}} \cdot 0,\mathbf{m} \cdot 2^{-1022}$

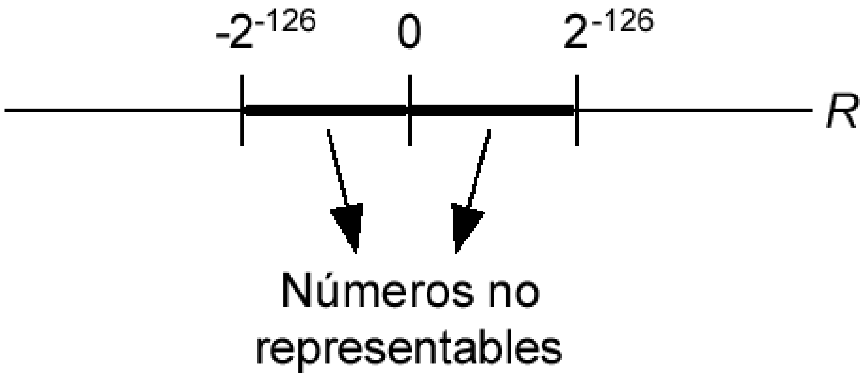
- **Rango de representación** en el estándar IEEE 754 con precisión doble:

$$\boxed{\mathbf{(\,(2^{-52} - 2) \cdot 2^{1023})_{10} \leq x \leq (\,(2 - 2^{-52}) \cdot 2^{1023})_{10}}}$$

2.6 REPRESENTACIÓN DE NÚMEROS REALES (8/8)

- **Rangos de representación discontinuos:** no pueden representar todos los números reales que existen entre dos cualesquiera de ellos.

- **EJEMPLO:** con **precisión simple**, alrededor del número cero (0), existen infinitos números reales que no son representables.



EJERCICIOS RECOMENDADOS

- **Resueltos:** 15 y 16.
- **Propuestos:** 13, 14, 15 y 16.

2.7 REPRESENTACIÓN DE CARACTERES (1/5)

- **Símbolos más usados por los humanos:**
  - **Letras minúsculas del alfabeto inglés:** { a, b, c, ..., x, y, z }
  - **Letras mayúsculas del alfabeto inglés:** { A, B, C, ..., X, Y, Z }
  - **Numéricos:** { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
  - **Especiales:** { +, -, *, /, @, #, ñ, Ñ, á, é, ... }
  - **Gráficos:** { ♣, ♦, ♥, ♠, ... }
  - **De control:** { Salto de línea, Tabulador horizontal, ... }

- **Tipos de codificación binaria:**
  - *EBCDIC*
  - *ASCII*
  - *Unicode*

2.7 REPRESENTACIÓN DE CARACTERES (2/5)

- **EBCDIC (*Extended Binary Coded Decimal Interchange Code*):**
  - Inventado por IBM.
  - Utiliza grupos de 8 bits. (2^8 = 256 caracteres).

- **ASCII (*American Standard Code for Information Interchange*):**
  - Emplea grupos de 7 bits. (2^7 = 128 caracteres).
  - También existe un **ASCII extendido** de 8 bits.

- **Unicode:**
  - Todos los símbolos de todos los idiomas.
  - Utiliza grupos de 16 bits. (2^{16} = 65536 caracteres).

2.7 REPRESENTACIÓN DE CARACTERES (3/5)

- **Tabla ASCII:**

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

2.7 REPRESENTACIÓN DE CARACTERES (4/5)

- **Caracteres (128-255) del ASCII extendido:**

	0	1	2	3	4	5	6	7	8	9
12									Ç	ü
13	é	â	ä	à	å	ç	ê	ë	è	ï
14	î	ì	Ä	Å	É	æ	Æ	ô	ö	ò
15	û	ù	ÿ	Ö	Ü	¢	£	¥	P_t	ƒ
16	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐
17	¬	½	¼	¡	«	»	░	▒	▓	│
18	┤	╡	╢	╖	╕	╣	║	╗	╝	╜
19	╛	┐	└	┴	┬	├	─	┼	╞	╟
20	╚	╔	╩	╦	╠	═	╬	╧	╨	╤
21	╥	╙	╘	╒	╓	╫	╪	┘	┌	█
22	▄	▌	▐	▀	α	ß	Γ	π	Σ	σ
23	µ	γ	Φ	θ	Ω	δ	∞	ø	∈	∩
24	≡	±	≥	≤	⌠	⌡	÷	≈	°	•
25	·	√	^{n}	^{2}	■					

2.7 REPRESENTACIÓN DE CARACTERES (5/5)

- **Caracteres "gráficos" (0-31) del ASCII:**

	0	1	2	3	4	5	6	7	8	9
0		☺	☻	♥	♦	♣	♠	•	◘	○
1	◙	♂	♀	♪	♫	☼	►	◄	↕	‼
2	¶	§	▬	↨	↑	↓	→	←	∟	↔
3	▲	▼								

GRACIAS POR SU ATENCIÓN

Para más información, puede visitar la web del autor:

http://www.carlospes.com

carlospes.com