Prólogo

Conozco a José María Martín desde hace más de 20 años. Recuerdo cuando compartíamos transporte juntos, con 18 años, en diferentes universidades, pero con ambiciones e ilusiones comunes.

Nuestros caminos profesionales han estado separados hasta que hace unos 7 años volvimos a encontrarnos para compartir un proyecto tan interesante como el de Ciudades Digitales para el municipio de Aranjuez, del cual yo era Concejal Delegado de Desarrollo Tecnológico.

Fue un período de unos 18 meses extremadamente intenso, en el que pude comprobar que la calidad humana de José María se había incrementado, lo cual ya era difícil, pero su capacidad de trabajo alcanzó cotas que la mayoría de los mortales sólo llegan a imaginar. Con dos fantásticos hijos, una casa que le da bastante trabajo (en la que compartimos largas horas de trabajo interrumpidas por alguna que otra barbacoa), una mujer maravillosa y sus otros compromisos profesionales, no mermaban la calidad y cantidad de trabajo, propuestas y proyectos que me ponía encima de la mesa. Reconozco que, siendo su jefe, me hacía trabajar a mí más que yo a él.

Durante ese período, todavía tuvo tiempo para escribir varios libros de gran prestigio, que son utilizados como textos oficiales por prestigiosas universidades y otros como elementos de consulta para muchísimos estudiantes. Es todo un honor, pues, poder contribuir humildemente a esta su nueva obra.

A buen seguro que este trabajo servirá a muchos jóvenes estudiantes para consultar dudas de una forma sencilla a la vez que técnicamente complejas, y a otros muchos profesionales, o simplemente aficionados, del mundo de la informática para conocer la arquitectura de los equipos y de los sistemas informáticos.

Lo que en muchas ocasiones nos provoca dolor de cabeza, porque el ordenador se “nos ha colgado”, o va muy lento, o simplemente no cubre nuestras expectativas, tiene, en muchas ocasiones, una solución a nuestro alcance gracias a los diferentes trabajos de José María a lo largo de estos años.

Desde los primeros equipos desarrollados en humildes garajes a los superordenadores modernos han pasado apenas 30 años, pero parecen siglos si comparamos lo que compramos hoy con lo que hace apenas 20 años estaba en los mercados. El disco duro de mi primer ordenador tenía 2000 veces menos capacidad que un simple DVD que nuestros hijos manejan con total soltura y naturalidad hoy.

Este increíble avance de la tecnología hace que los trabajos y actualizaciones de profesionales como José María sean tan valiosos, porque hacen que nos mantengamos al día aquellos que trabajamos como usuarios con las aplicaciones informáticas, cuyo soporte físico es precisamente el alma de este libro.

Como ya he dicho antes, es un honor poder escribir el prólogo de este libro, cuando antes que yo, en otras publicaciones, lo han hecho personas de la calidad humana y capacidad de análisis como Santiago Lorente, que nos abandonó tristemente hace pocos años.

Querido José María, mucha suerte y sirva este prólogo para hacerte llegar mi más profunda amistad y admiración. A los lectores de España y en general de todos los países a los que nos hermanamos a través del castellano, les deseo que encuentren en la lectura de este trabajo la misma satisfacción, al menos, que siento yo con cada libro de este fantástico autor. Si hubieras nacido 15 años antes en los Estados Unidos, Bill Gates, los creadores de Apple, Google, Yahoo y tantos iconos de la informática, hoy mundialmente conocidos, entre otras cosas por sus colosales fortunas, estarían en tu círculo de amistades. Me alegro que no haya sido así, porque soy yo el que se encuentra cerca de ti, incluso desde mi residencia en el Reino Unido.

Carlos de Fuentes Artasánchez
Ingeniero Técnico Industrial, Universidad Pontificia de Comillas. ICAI
Ex Concejal de Desarrollo Tecnológico del Ayuntamiento de Aranjuez
European Supply Manager, Unilever Europe

Introducción

La relevancia que ha llegado a tener el ordenador personal o PC en el mundo de hoy es indiscutible. El PC nos acompaña en casi todas nuestras actividades cotidianas, tanto laborales como domésticas e, incluso, en nuestros momentos de diversión y esparcimiento. Lo que también resulta indiscutible es la forma como el PC ha acortado las distancias globales, permitiendo la comunicación casi de forma instantánea entre personas que se encuentran en lugares muy lejanos, geográficamente hablando, permitiendo el trabajo en colaboración. De hecho, esa es la forma en la que fue gestado parte de este libro. Los autores, separados por un océano de distancia (en el sentido más literal del término: José Leonardo Simancas desde Colombia y José María Martín en España), con dos ordenadores y el software de comunicación oportuno, han trabajado como si de forma física se encontrasen en el mismo despacho para, al final, conseguir que usted tenga este buen libro en sus manos. No muy diferente ha sido el trabajo con respecto a la edición, maquetación… Este es un ejemplo que deja patente la necesidad de conocer una máquina que realmente ha cambiado el mundo y abre un abanico de posibilidades impensable hasta no hace mucho.

Las características de los componentes de un ordenador cambian continuamente. Muchos usuarios encuentran la información más actualizada en revistas de informática y anuncios pero, obviamente, cada marca favorece sus propios productos. Este libro ofrece información de diferentes fabricantes, atendiendo más a las características de sus productos que al despliegue de marketing realizado. De manera objetiva y nunca parcial, proporciona cifras y parámetros que intentan ayudar al lector en la evaluación acerca de la conveniencia real de un dispositivo, y cuál será su rendimiento previsible.

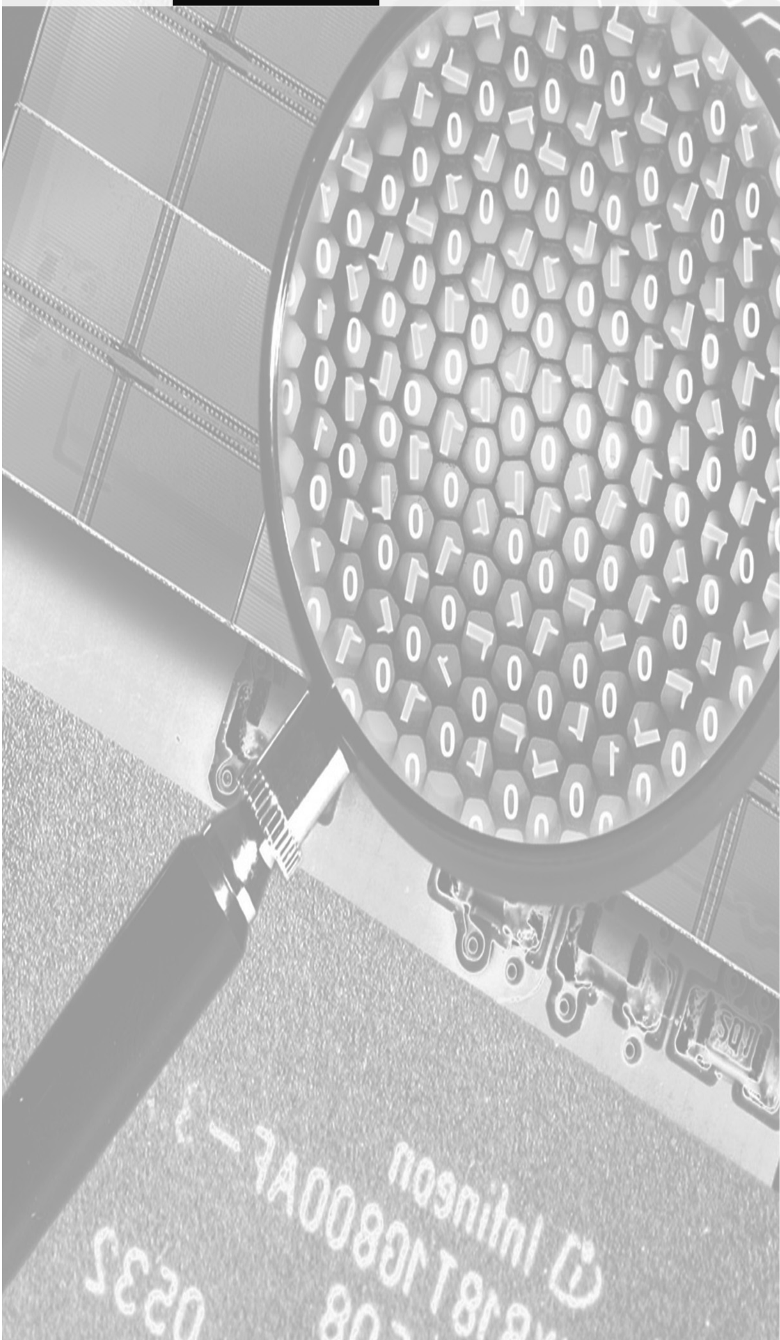
En base a la experiencia, se ha determinado que la mejor manera de introducirse de lleno en el mundo de los ordenadores, ya como usuario que actualiza y da mantenimiento a su propio equipo o como profesional en el área que brinda soporte técnico a usuarios inexpertos, es el conocimiento del amplio abanico de posibilidades tecnológicas que convergen en un ordenador. Así, el lector podrá comprobar la exhaustividad en el reconocimiento de los diversos dispositivos de procesamiento, entrada y salida de datos y de almacenamiento. La idea final ha sido la de ofrecer un panorama amplio y concreto donde el lector pudiera tomar sus mejores decisiones con respecto a la selección y combinación específica de los elementos de hardware con los que construirá su equipo ordenador, desde los últimos monitores LCD hasta los potentes microprocesadores *CORE 2 DUO*, pasando por impresoras y demás componentes, periféricos y accesorios.

El texto, con un marcado carácter didáctico, está dirigido de forma específica a estudiantes de informática y profesionales de los sistemas de información que deseen tener un conocimiento sólido y profundo para su aplicación en el terreno laboral de la Microinformática. Este planteamiento no es restrictivo para todas aquellas personas interesadas en el mundo de los ordenadores que deseen realizar el ensamblaje de un PC con componentes de última tecnología.

Como elemento diferenciador, en este libro encontrará el fundamento electrónico de cada uno de los componentes integrantes del PC. En este sentido no sólo se realiza un análisis a nivel global de las tarjetas sino, además, un estudio previo de puertas lógicas, biestables y circuitos tipo. Una parte importante del texto está dedicada a la lógica microprogramable, tratando PLD, CPLD, GAL… desarrollada por José Leonardo Simancas que, de forma meticulosa, analiza la potencialidad de estos montajes.

Los autores

1 Tecnologías digitales: codificación de la información



Objetivos del capítulo

- ✓ Conocer diferentes sistemas de numeración.
- ✓ Operar con diferentes sistemas y encontrar equivalencias entre ellos.
- ✓ Relacionar la información con su representación numérica.
- ✓ Entender los códigos ASCII y Unicode como básicos para la representación alfanumérica.

La evolución de la Informática ha sido consecuencia directa del modo de codificar la información. Algo tan etéreo como son los datos impone dificultades en su tratamiento si no es posible establecer una magnitud que ayude a cuantificar los valores. Ahora bien, ¿cuál debe ser la medida? La información no se mide en longitud, peso o volumen. Las premisas sentadas por George Boole fueron las apropiadas para que casi 100 años después, Claude Shanon, matemático e ingeniero estadounidense, descubriese la relación que se podía establecer entre información, Álgebra de Boole y Electrónica. Una vez llegados a este punto, la solución era relativamente sencilla...

La Historia informática está llena de diseños e ingenios que intentaban –y conseguían en ocasiones– manejar la información. La dificultad de codificación de la información se suplía con esfuerzo del usuario, que era quien debía introducir los parámetros de forma manual para conseguir los resultados... Artificios mecánicos, en su gran mayoría, hacían el resto.

La facilidad de interpretación de la información a través del Álgebra de Boole supone un avance importante del que ni tan siquiera el propio Boole fue consciente. El hecho de reducir el universo informacional a sólo dos dígitos, sin perder por ello precisión o detalle en los datos, simplifica colosalmente el problema de la gestión.

Existen otras opciones de codificación de la información, como puede ser el caso de la Lógica Difusa de la compañía Omrom que, además de establecer dos niveles lógicos opuestos entre sí, contempla una infinidad de valores entre ambos, pero lo cierto es que, a día de hoy, seguimos trabajando según las directrices que hace más de siglo y medio (1847) marcó un inglés en *El análisis matemático del pensamiento*. Ésta es la importancia de la codificación y el porqué de este capítulo.

1.1 CUANTIFICACIÓN DE LA INFORMACIÓN

Antes de poder abordar el modo de establecer una mesura en la información, es necesario definir qué se entiende por información. Lo cierto es que existen numerosas definiciones según el autor que se desee consultar. Los hay que definen el concepto como una unidad básica resultado de la inversa del logaritmo neperiano de la probabilidad de que ocurra un proceso, hasta los que lo interpretan como un ente abstracto con capacidad de representación. Quizás, entre todas ellas, la más clara sea la que entiende la información como el cambio producido desde un espacio del universo con respecto a otro mediante un mensaje.

Una buena definición es la propuesta argumentada con un ingenioso ejemplo por D. José Luis Lorente Guash, profesor de Física de la UNED en uno de sus libros. Así, supóngase una moneda que se lanza al aire. Esta simple acción denota, además de una incertidumbre sobre el resultado (cara o cruz), una falta de información que es, precisamente, el lado por el que caerá la moneda. Admitida, por tanto, la idea de que la moneda aportará información en función de su posición de caída, se puede establecer un símil válido con un bit ya que la citada posición podrá variar entre dos posibilidades: cara o cruz. De este modo, se dispone de un elemento que, susceptible de adoptar uno de dos estados opuestos entre sí, aporta una información concreta. Ésa es la idea de un bit.

1.2 SISTEMAS DE NUMERACIÓN

Habitualmente trabajamos en sistema decimal, es decir, contamos desde 0 hasta 9 y, a partir de ahí, añadimos un nuevo dígito y comenzamos de nuevo la numeración. Cada vez que lleguemos a 9, incrementaremos el número añadido hasta que éste también llegue a 9, momento en el que se repetirá el proceso descrito. Este sistema de numeración, heredado de los árabes, permite un fácil manejo a la vez que una importante comodidad aritmética. Matemáticamente, su definición corresponde a la suma de N paquetes donde cada uno está formado por el producto del dígito original por la base de numeración elevada a un peso determinado. Dicho peso se obtiene de la posición del dígito original, comenzando a contar de derecha a izquierda menos uno, es decir:

$$Número = N_n \cdot B^{Pn-1} + N_{n-1} \cdot B^{Pn-2} + \ldots$$

donde

N_x corresponde al dígito de posición x en el número N.
B corresponde a la base de numeración que, en este caso, es 10.
Pn-1 corresponde al peso del dígito de posición n-1.

Por ejemplo, el número 128 formaría tres paquetes (1, 2 y 8), donde cada uno multiplicaría el dígito en cuestión (nuevamente 1, 2 y 8) por la base, que en nuestro caso es 10, elevada al peso del dígito menos uno. Así, según el orden, se obtendrían pesos de 1 para el 8, 2 para el 2 y 3 para el 1 (contando de derecha a izquierda). Si a éstos se les resta la unidad, se obtendría los pesos (de derecha a izquierda) de 0, 1 y 2. El resultado es, por tanto:

$$128 = 1 \cdot 10^2 + 2 \cdot 10^1 + 8 \cdot 10^0 = 1 \cdot 100 + 2 \cdot 10 + 8 \cdot 1 = 100 + 20 + 8 = 128$$

Según su base de numeración, existen distintos sistemas. Otro ejemplo habitual es el horario. Se sugiere al lector que determine la forma de numeración en la formación de minutos, segundos, días, meses... No obstante, el sistema decimal no es el único usado. Dependiendo de la finalidad, se pueden usar otras bases de numeración. Este capítulo describirá las más comunes.

1.3 ANALÓGICO VERSUS DIGITAL

Existe la falsa creencia de que un sistema digital es más exacto que uno analógico. Nada más lejos de la realidad: un sistema digital dispone de un número finito de valores para representar cantidades y, por tanto, es preciso realizar una aproximación. Un sistema analógico dispone de infinitas posibilidades de representación. Ahora bien, en un sistema analógico es la persona quien debe interpretar qué valor se está representando, mientras que en un sistema digital se limita a leerlo. Por ejemplo, el velocímetro digital de un coche indicará la velocidad del mismo con la exactitud con la que ha sido definido el marcador (número de dígitos). De este modo, si el velocímetro dispone de tres dígitos para números enteros y uno para decimales, si circula a 80,876 Km/h será preciso aproximar a 80,9 Km/h. El mismo caso en un velocímetro analógico (de “aguja”) representaría exactamente el valor de la velocidad pero sería el usuario quien, probablemente, no fuese capaz de interpretarlo con tanta exactitud.

1.4 COMPLEMENTO Y BASE DE NUMERACIÓN

Por *complemento* de un número se entenderá aquel que, sumado al número original, obtenga la base de numeración menos la unidad. Supóngase la necesidad de encontrar en el sistema decimal el número complementario al 3. El planteamiento y desarrollo responde a:

Complemento (N) = Base Numeración - N –1;
Complemento (3) = 10 - 3 - 1 = 6

Base de un sistema responderá al número de símbolos diferentes definidos dentro del lenguaje. Así, se definirían:

Base binaria	*(2)*	*Binario =*	*{0, 1}*
Base octal	*(8)*	*Octal =*	*{0, 1, 2, 3, 4, 5, 6, 7}*
Base decimal	*(10)*	*Decimal =*	*{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}*
Base hexadecimal	*(16)*	*Hexadecimal =*	*{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}*

1.5 PONDERACIÓN EN LOS CÓDIGOS

Dentro de los diferentes códigos que existen se puede establecer una división para catalogarlos: los códigos ponderados y los no ponderados. Los códigos ponderados son aquellos que, como su nombre indica, tienen un valor concreto. Dicho valor lo recibe el dígito según la posición que ocupa dentro del número (se dice que tienen “peso”).

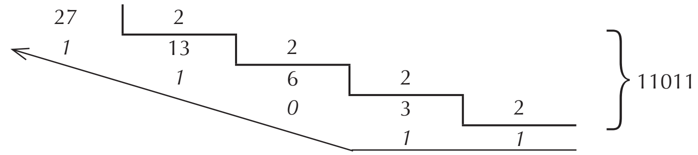
De esta forma, dentro de los códigos ponderados están el binario natural, el BCD o el AIKEN y dentro de los no ponderados aparecen el exceso a 3 y el código Gray (o reflejado).

1.6 SISTEMA BINARIO

Está basado en el uso exclusivo de dos números: 0 y 1. Esto facilita su posterior identificación electrónica, ya que el modo de funcionamiento de los semiconductores se realizará en corte o saturación, siendo sencillo el asociar un estado de tensión para cada número.

El sistema de formación es el mismo que el decimal pero sustituyendo la base 10 por base 2. Para convertir cualquier número decimal a binario, se realizan sucesivas divisiones por dos hasta llegar a un resto de 0 ó 1. Finalmente se "recogen" los módulos y el último cociente formando así el número binario. Así, la conversión del número 27 a binario correspondería a

Tabla 1.1 Conversión de decimal a binario



Luego 27 en binario es el número 11011. Para realizar la conversión de un número binario a decimal, el proceso consiste en multiplicar cada dígito del número binario obtenido por 2 elevado a la posición que ocupa el dígito comenzando por 0. Dicha posición se cuenta comenzando por la derecha. Al final, se suman los resultados obtenidos y se obtiene el número decimal. Realicemos la conversión del número 11011, obtenido como resultado en el último ejercicio:

$$11011=1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0=16 + 8 + 2 + 1=27$$

1.7 SISTEMA OCTAL

Otro sistema de numeración usado es el denominado octal. Recibe su nombre del número de elementos que lo conforman. La numeración evoluciona de forma idéntica a la decimal desde 0 hasta 7. Una vez superado este valor, se añade un dígito más comenzando la sucesión. La equivalencia con binario y decimal sería la siguiente:

Tabla 1.2 Equivalencias base 10, 2, 8

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

Para pasar un número en octal a binario, basta con tomar independientemente cada uno de sus dígitos y transformarlos por su equivalente binario en tres bits. Así, el número 436 en base octal correspondería al 100011110 en binario, tal y como se demuestra:

$$\frac{\begin{array}{ccc}4 & 3 & 6\end{array}}{\begin{array}{ccc}100 & 011 & 110\end{array}} = 100011110$$

Para realizar la conversión de binario a octal, el proceso es el inverso: se agruparán elementos de tres en tres y se sustituirán por su equivalente en octal. Así, el ejercicio anterior tendría la conversión de:

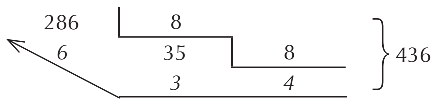
$$\frac{\begin{array}{ccc}100 & 011 & 110\end{array}}{\begin{array}{ccc}4 & 3 & 6\end{array}} = 436$$

Si se desea pasar del octal al decimal, la conversión se realiza igual que si se tratase de un número binario, a diferencia de que la base de numeración será 8. Haciendo la inversa del ejercicio anterior obtendríamos:

$$436_{(8} = 4 \cdot 8^2 + 3 \cdot 8^1 + 6 \cdot 8^0 = 4 \cdot 64 + 3 \cdot 8 + 6 \cdot 1 = 256 + 24 + 6 = 286_{(10}$$

La última de las conversiones pasaría por transformar el sistema decimal en octal, para lo que será necesario realizar divisiones sucesivas por 8 (base de la numeración) y reservar los restos y el último cociente tomados de derecha a izquierda.

Tabla 1.3 Conversión de decimal a octal



1.8 SISTEMA HEXADECIMAL

Este sistema aparece como una forma condensada del sistema binario. La base de trabajo es 16 y se usa habitualmente en calculadoras o máquinas pequeñas. Es más “compacto” que el binario, ya que aúna grupos de cuatro dígitos, opción que es aprovechada por pequeñas máquinas de 4, 8, 16, 32... bits que lo usan como sistema de numeración. Básicamente responde a la siguiente tabla:

Tabla 1.4 Equivalencias base 10, 2, 16

Decimal	Binario	Hexadecimal	Decimal	Binario	Hexadecimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

Así, si se desea pasar un número de binario a hexadecimal, bastará con aunarlo en grupos de cuatro dígitos y transcribir directamente su equivalente en hexadecimal. Supóngase el número 1101010011110110:

$$\frac{1101 \quad 0100 \quad 1111 \quad 0110}{D \quad\quad 4 \quad\quad F \quad\quad 6} = D4F6$$

Si la operación fuese al contrario (pasar de hexadecimal a binario), el proceso sería el inverso: se deben observar cada uno de los dígitos y transcribirlos por su equivalente en binario de cuatro bits. Supóngase el caso justo inverso al ya hallado: el número D4F6:

$$\frac{D \quad\quad 4 \quad\quad F \quad\quad 6}{1101 \quad 0100 \quad 1111 \quad 0110} = 1101010011110110$$

comprobándose así la correcta traslación de un valor entre distintos sistemas de numeración. El ejemplo puede hacerse extensivo al sistema decimal.

1.9 CÓDIGO ASCII

Se trata del código por excelencia de representación de caracteres en el PC. A estas alturas es harto sabido por parte del lector que el PC trabaja con un sistema difícilmente comprensible (o, cuando menos, incómodo) para el usuario: el código binario. Realmente y según las necesidades se usa binario o hexadecimal. Por otro lado, el usuario utiliza el denominado “lenguaje natural” que implica las mismas dificultades de comprensión para la máquina que el binario para las personas.

Así, surge el código ASCII. Este código, formado por ocho bits, lo que origina un total de 256 caracteres numerados del 0 al 255 (2^8=256), recoge todas las posibilidades (o casi todas) de representación de un idioma. Así, se recogerán aquí todas las letras, caracteres identificativos y específicos de un idioma (como puede ser la letra eñe) e, incluso, pulsaciones de teclado como son Enter, borrado, símbolo nulo... Así mismo, los 32 primeros caracteres (del 0 al 31) tendrán la consideración de caracteres especiales o lo que es lo mismo, con alguna función asociada. A partir del código ASCII número 32 comienza la relación de símbolos normales (caracteres semigráficos, números, letras...). En este mismo capítulo se adjunta una relación del código ASCII para mayor abundancia de datos. A este respecto resulta interesante conocer que existió una primera versión del código ASCII formada por 7 bits que limitaba el conjunto a 128 caracteres (2^7=128).

Así, la máquina realizará sus cálculos en binario y, cuando deba presentarlos al usuario, hará una conversión a través del código ASCII. En caso contrario el proceso se invierte: el usuario introducirá datos en un formato ASCII y se hará una traslación a binario. Cada país tiene su propia página de códigos que recoge el código ASCII específico de un idioma. En el caso de España, la página de códigos es la 034 aunque también resulta válida la multilingüe, recogida con el número 850.

1.10 CÓDIGO UNICODE

El conjunto de caracteres Unicode pretende ser la respuesta a las limitaciones que el código ASCII impone al variar la página de códigos según los países. Dado que los países cada vez comparten más (política, moneda, empresas...), empieza a precisarse un código que recoja estas necesidades. Por poner un ejemplo significativo, el símbolo del euro (€) tiene ya sentido en varios países.

Así surge el Unicode. Se trata de un conjunto de caracteres mucho más amplio que el ASCII convencional donde el rango posible de caracteres se establece entre 0 y 65.536, como consecuencia de su formación por 16 bits (2^{16}=65.536). Como el lector recordará, el ASCII está formado por ocho bits, lo que le limita a 256 caracteres (2^8 = 256). Hay una correspondencia directa entre los 127 primeros caracteres del código ASCII y el Unicode.

Es habitual referenciar a los caracteres del código Unicode en hexadecimal (del 0000 al FFFF) dada su compresión, usando 4 dígitos y ya hay muchos lenguajes de programación que empiezan a priorizarlo sobre el ASCII, como puede ser el caso de Java. El Unicode incluye caracteres, entre otros, del latín, griego, arábico, cirílico, hebreo, katakana y hangul. En definitiva, representa los caracteres de la mayoría de los idiomas que tienen representación escrita en el ámbito mundial.

1.11 EQUIVALENCIA ENTRE CÓDIGOS

Sirva a modo de resumen las siguientes conversiones entre códigos con los procesos implicados. Se han usado los códigos más comunes, invitando al lector a que reproduzca otros ejemplos con los códigos que crea convenientes:

Decimal → Binario

$427_{(10} = 110101011_{(2}$

427	2								
1	213	2							
	1	106	2						
		0	53	2					
			1	26	2				
				0	13	2			
					1	6	2		
						0	3	2	
							1	1	

EJERCICIO RESUELTO 1.1

CONVIERTA DE SISTEMA DECIMAL A BINARIO EL NÚMERO 153.

153	2							
1	76	2						
	0	38	2					
		0	19	2				
			1	9	2			
				1	4	2		
					0	2	2	
						0	**1**	

10011001

Binario → Decimal

$110101011_{(2} = 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 = 256 + 128 + 32 + 8 + 2 + 1 = 427_{(10}$

EJERCICIO RESUELTO 1.2

CONVIERTA DE SISTEMA BINARIO A DECIMAL EL NÚMERO 10011001.

10011001 = 1·27+ 0·26+ 0·25+ 1·24+ 1·23+ 0·22+ 0·21+ 1·20 = 128 + 16 + 8 + 1 = 153

Binario → Octal

$$110101011_{(2} = \frac{110 \mid 101 \mid 011}{6 \mid 5 \mid 3} = 653_{(8}$$

Octal → Binario

$$653_{(8} = \frac{6 \mid 5 \mid 3}{110 \mid 101 \mid 011} = 110101011_{(2}$$

EJERCICIO RESUELTO 1.3

CONVIERTA DE SISTEMA BINARIO A OCTAL EL NÚMERO 10011001.

$$\frac{10}{2} \quad \frac{011}{3} \quad \frac{001}{1} = 231$$

EJERCICIO RESUELTO 1.4

CONVIERTA DE SISTEMA OCTAL A BINARIO EL NÚMERO 231.

$$\frac{2}{10} \quad \frac{3}{011} \quad \frac{1}{001} = 10011001$$

Octal → Decimal

$$653_{(8} = 6 \cdot 8^2 + 5 \cdot 8^1 + 3 \cdot 8^0 = 6 \cdot 64 + 5 \cdot 8 + 3 \cdot 1 = 384+40+3 = 427_{(10}$$

Decimal → Octal

$$\begin{array}{r|l} 427_{(10} & 8 \\ \hline \underline{3} & 53 \;\big|\; 8 \\ & \;\underline{5} \quad\; \underline{6} \end{array}$$

Decimal → Hexadecimal

$$\begin{array}{r|l} 427_{(10} & 16 \\ \hline \underline{11} & 26 \;\big|\; 16 \\ & \underline{10} \quad\; \underline{1} \end{array}$$

Binario → Hexadecimal

$110101011_{(2} =$

1	1010	1011
1	A	B

= 1ABh

Hexadecimal → Binario

1ABh=

1	A	B
1	1010	1011

$= 110101011_{(2}$

Hexadecimal → Decimal

$1ABh = 1 \cdot 16^2 + A \cdot 16^1 + B \cdot 16^0 = 256 + 160 + 11 = 427_{(10}$

Se anexa a continuación una tabla resumen de equivalencias con los códigos más usados (Decimal – Binario – Octal – Hexadecimal - ASCII):

Tabla 1.5 Equivalencias entre códigos

De	Binario	Oc	He	A	De	Binario	Oc	He	A	De	Binario	Oc	He	A	De	Binario	Oc	He	A
0	00000000	000	00		64	01000000	100	40	@	128	10000000	200	80	€	192	11000000	300	C0	À
1	00000001	001	01		65	01000001	101	41	A	129	10000001	201	81		193	11000001	301	C1	Á
2	00000010	002	02		66	01000010	102	42	B	130	10000010	202	82	‚	194	11000010	302	C2	Â
3	00000011	003	03		67	01000011	103	43	C	131	10000011	203	83	ƒ	195	11000011	303	C3	Ã
4	00000100	004	04		68	01000100	104	44	D	132	10000100	204	84	„	196	11000100	304	C4	Ä
5	00000101	005	05		69	01000101	105	45	E	133	10000101	205	85	…	197	11000101	305	C5	Å
6	00000110	006	06		70	01000110	106	46	F	134	10000110	206	86	†	198	11000110	306	C6	Æ
7	00000111	007	07		71	01000111	107	47	G	135	10000111	207	87	‡	199	11000111	307	C7	Ç
8	00001000	010	08		72	01001000	110	48	H	136	10001000	210	88	ˆ	200	11001000	310	C8	È
9	00001001	011	09		73	01001001	111	49	I	137	10001001	211	89	‰	201	11001001	311	C9	É
10	00001010	012	0A		74	01001010	112	4A	J	138	10001010	212	8A	Š	202	11001010	312	CA	Ê
11	00001011	013	0B		75	01001011	113	4B	K	139	10001011	213	8B	‹	203	11001011	313	CB	Ë
12	00001100	014	0C		76	01001100	114	4C	L	140	10001100	214	8C	Œ	204	11001100	314	CC	Ì
13	00001101	015	0D		77	01001101	115	4D	M	141	10001101	215	8D		205	11001101	315	CD	Í
14	00001110	016	0E		78	01001110	116	4E	N	142	10001110	216	8E	Ž	206	11001110	316	CE	Î
15	00001111	017	0F		79	01001111	117	4F	O	143	10001111	217	8F		207	11001111	317	CF	Ï
16	00010000	020	10		80	01010000	120	50	P	144	10010000	220	90		208	11010000	320	D0	Ð
17	00010001	021	11		81	01010001	121	51	Q	145	10010001	221	91	‘	209	11010001	321	D1	Ñ
18	00010010	022	12		82	01010010	122	52	R	146	10010010	222	92	’	210	11010010	322	D2	Ò
19	00010011	023	13		83	01010011	123	53	S	147	10010011	223	93	“	211	11010011	323	D3	Ó
20	00010100	024	14		84	01010100	124	54	T	148	10010100	224	94	”	212	11010100	324	D4	Ô
21	00010101	025	15		85	01010101	125	55	U	149	10010101	225	95	•	213	11010101	325	D5	Õ
22	00010110	026	16		86	01010110	126	56	V	150	10010110	226	96	–	214	11010110	326	D6	Ö
23	00010111	027	17		87	01010111	127	57	W	151	10010111	227	97	—	215	11010111	327	D7	×
24	00011000	030	18		88	01011000	130	58	X	152	10011000	230	98	˜	216	11011000	330	D8	Ø
25	00011001	031	19		89	01011001	131	59	Y	153	10011001	231	99	™	217	11011001	331	D9	Ù
26	00011010	032	1A		90	01011010	132	5A	Z	154	10011010	232	9A	š	218	11011010	332	DA	Ú
27	00011011	033	1B		91	01011011	133	5B	[	155	10011011	233	9B	›	219	11011011	333	DB	Û
28	00011100	034	1C		92	01011100	134	5C	\	156	10011100	234	9C	œ	220	11011100	334	DC	Ü
29	00011101	035	1D		93	01011101	135	5D	]	157	10011101	235	9D		221	11011101	335	DD	Ý
30	00011110	036	1E		94	01011110	136	5E	^	158	10011110	236	9E	ž	222	11011110	336	DE	Þ
31	00011111	037	1F		95	01011111	137	5F	_	159	10011111	237	9F	Ÿ	223	11011111	337	DF	ß
32	00100000	040	20		96	01100000	140	60	`	160	10100000	240	A0		224	11100000	340	E0	à
33	00100001	041	21	!	97	01100001	141	61	a	161	10100001	241	A1	¡	225	11100001	341	E1	á
34	00100010	042	22	"	98	01100010	142	62	b	162	10100010	242	A2	¢	226	11100010	342	E2	â
35	00100011	043	23	#	99	01100011	143	63	c	163	10100011	243	A3	£	227	11100011	343	E3	ã
36	00100100	044	24	$	100	01100100	144	64	d	164	10100100	244	A4	¤	228	11100100	344	E4	ä

De	Binario	Oc	He	A	De	Binario	Oc	He	A	De	Binario	Oc	He	A	De	Binario	Oc	He	A
37	00100101	045	25	%	101	01100101	145	65	e	165	10100101	245	A5	¥	229	11100101	345	E5	å
38	00100110	046	26	&	102	01100110	146	66	f	166	10100110	246	A6	¦	230	11100110	346	E6	æ
39	00100111	047	27	′	103	01100111	147	67	g	167	10100111	247	A7	§	231	11100111	347	E7	ç
40	00101000	050	28	(	104	01101000	150	68	h	168	10101000	250	A8	¨	232	11101000	350	E8	è
41	00101001	051	29	)	105	01101001	151	69	i	169	10101001	251	A9	©	233	11101001	351	E9	é
42	00101010	052	2A	*	106	01101010	152	6A	j	170	10101010	252	AA	ª	234	11101010	352	EA	ê
43	00101011	053	2B	+	107	01101011	153	6B	k	171	10101011	253	AB	″	235	11101011	353	EB	ë
44	00101100	054	2C	,	108	01101100	154	6C	l	172	10101100	254	AC	¬	236	11101100	354	EC	ì
45	00101101	055	2D	-	109	01101101	155	6D	m	173	10101101	255	AD	–	237	11101101	355	ED	í
46	00101110	056	2E	.	110	01101110	156	6E	n	174	10101110	256	AE	®	238	11101110	356	EE	î
47	00101111	057	2F	/	111	01101111	157	6F	o	175	10101111	257	AF	¯	239	11101111	357	EF	ï
48	00110000	060	30	0	112	01110000	160	70	p	176	10110000	260	B0	°	240	11110000	360	F0	ð
49	00110001	061	31	1	113	01110001	161	71	q	177	10110001	261	B1	±	241	11110001	361	F1	ñ
50	00110010	062	32	2	114	01110010	162	72	r	178	10110010	262	B2	²	242	11110010	362	F2	ò
51	00110011	063	33	3	115	01110011	163	73	s	179	10110011	263	B3	³	243	11110011	363	F3	ó
52	00110100	064	34	4	116	01110100	164	74	t	180	10110100	264	B4	´	244	11110100	364	F4	ô
53	00110101	065	35	5	117	01110101	165	75	u	181	10110101	265	B5	µ	245	11110101	365	F5	õ
54	00110110	066	36	6	118	01110110	166	76	v	182	10110110	266	B6	¶	246	11110110	366	F6	ö
55	00110111	067	37	7	119	01110111	167	77	w	183	10110111	267	B7	·	247	11110111	367	F7	÷
56	00111000	070	38	8	120	01111000	170	78	x	184	10111000	270	B8	¸	248	11111000	370	F8	ø
57	00111001	071	39	9	121	01111001	171	79	y	185	10111001	271	B9	¹	249	11111001	371	F9	ù
58	00111010	072	3A	:	122	01111010	172	7A	z	186	10111010	272	BA	º	250	11111010	372	FA	ú
59	00111011	073	3B	;	123	01111011	173	7B	{	187	10111011	273	BB	″	251	11111011	373	FB	û
60	00111100	074	3C	<	124	01111100	174	7C	|	188	10111100	274	BC	¼	252	11111100	374	FC	ü
61	00111101	075	3D	=	125	01111101	175	7D	}	189	10111101	275	BD	½	253	11111101	375	FD	ý
62	00111110	076	3E	>	126	01111110	176	7E	~	190	10111110	276	BE	¾	254	11111110	376	FE	þ
63	00111111	077	3F	?	127	01111111	177	7F	▒	191	10111111	277	BF	¿	255	11111111	377	FF	ÿ

Actividades

- Complete la siguiente tabla y obtenga el texto ASCII:

Binario	01001000	01001111	01001100	01000001
Decimal				
ASCII				

Actividades

- Realice el ejercicio inverso a la actividad anterior:

ASCII	H	O	L	A
Decimal				
Binario				

1.12 SISTEMA BCD

El BCD o Binary Code Decimal es similar al código binario puro. Se forma de manera constante con cuatro dígitos representando los valores decimales de 0 al 9. El resto de los dígitos se realizarán como combinación de los anteriores. Es ponderado en 8-4-2-1. Así, la formación de los diez primeros será:

Tabla 1.6 Formación del código BCD

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

A partir del nueve, el resto serán combinaciones como, por ejemplo:

87	
8	7
1000	0111
10000111	

28	
2	8
0010	1000
00101000	

49	
4	9
0100	1001
01001001	

75	
7	5
0111	0101
01110101	

NOTA 1.1

Obsérvese que no hay diferencia en los diez primeros números entre el código BCD y el código binario natural, exceptuando la formación de los números con cuatro dígitos.

1.12.1 SUMA EN BCD

La suma en BCD natural se realiza bit a bit de modo que:

$0 + 0 = 0$; $0 + 1 = 1$; $1 + 0 = 0$; $1 + 1 = 0$ (más 1 de CARRY)[1];

La única novedad que deberá tenerse en cuenta es que, cuando se realice una suma y el resultado obtenido sea mayor de 9 (1001), habrá que sumar 6 (0110) a ese valor. Suponga la suma de los números 8 (1000) y 7 (0111). La suma dará 15 (1111) que obviamente es mayor que 9:

1 Carry o acarreo, representado en este texto como CARRY es un concepto ya explicado en las operaciones con el sistema binario.

	1000
	0111 +
	1111 (> 1001)
	0110 +
(CARRY) 1	0101

El resultado obtenido sería 1 0101 que, por homogeneidad se representará complementando con tres ceros a la izquierda como:

0001	0101
1	5

El ejemplo se amplía de tal modo que, para sumar 4.575 y 6.232, se realizaría:

4.575	→			0100	0101	0111	0101
6.232	→	+		0110	0010	0011	0010
				*1010	0111	*1010	**0111**
				+ 0110		+ 0110	
			0001	**0000**	+ 1	**0000**	
					1000		
10.807	→		1	0	8	0	7
*>1001							

1.12.2 RESTA EN BCD

Al igual que en el sistema decimal, se entiende la resta en BCD como la suma con un número negativo, de tal manera que 5 - 3 = 5 + (-3). Es preciso determinar cómo se representan los números negativos en BCD, o, lo que es lo mismo, los números complementarios. Así, supóngase la resta de 8.334 menos 3.890.

8.334	→			1000	0011	0011	0100
3.890	→ (CA9)[2] 6.109→	+		0110	0001	0000	1001
				*1110	**0100**	0011	*1101
				+ 0110			+ 0110
			1	**0100**		1	0011
						+ 0100	+ 1
							0100
4.444	→			4	4	4	4
*>1001							

Al igual que en la operación de adición, si el número supera el 9, se le suma 6 y los carrys se suman de un grupo a otro, con la excepción del grupo de bits más significativo, que añade su carry (si es que existe) al de menor peso.

2 Complemento a 9.

1.12.3 CÓDIGO AIKEN

Es un código ponderado como BCD en 2-4-2-1 y también usa 10 números de base formados por cuatro dígitos a partir de los cuales forma el resto.

Tabla 1.7 Formación del código AIKEN

Decimal	Aiken	Decimal	Aiken
0	0000	5	1011
1	0001	6	1100
2	0010	7	1101
3	0011	8	1110
4	0100	9	1111

A partir del nueve, el resto serán combinaciones como, por ejemplo:

87	
8	7
1110	1101
11101101	

28	
2	8
0010	1110
00101110	

49	
4	9
0100	1111
01001111	

75	
7	5
1101	1011
11011011	

1.13 SISTEMA EXCESO A 3

Se trata de un código no ponderado –que no tiene "peso"– y, además, autocomplementario ya que la suma de aquellos números "contrarios" (donde se han cambiado los 0 por 1, y viceversa) siempre da como valor el número 1111 ó 9 en decimal. La formación es la siguiente:

NOTA 1.2

Obsérvese que no hay diferencia en los cinco primeros números entre el código BCD y el AIKEN.

Tabla 1.8 Formación del código Exceso a 3

Decimal	Exceso a 3	Decimal	Exceso a 3	Decimal	Exceso a 3
0	0011	3	0110	6	1001
1	0100	4	0111	7	1010
2	0101	5	1000	8	1011
				9	1100

La autocomplementación se demuestra así:

$$\begin{array}{r} 0011\ (0) \\ \underline{+1100}\ (9) \\ 1111\ (9) \end{array} \quad \begin{array}{r} 0100\ (1) \\ \underline{+1011}\ (8) \\ 1111\ (9) \end{array} \quad \begin{array}{r} 0101\ (2) \\ \underline{+1010}\ (7) \\ 1111\ (9) \end{array} \quad \begin{array}{r} 0110\ (3) \\ \underline{+1001}\ (6) \\ 1111\ (9) \end{array} \quad \begin{array}{r} 0111\ (4) \\ \underline{+1000}\ (5) \\ 1111\ (9) \end{array}$$

NOTA 1.3

Obsérvese que la formación del código se realiza a partir de un binario natural comenzando la secuencia en el número 3 (0011) en lugar de hacerlo en el 0 (0000).

RESUMEN DEL CAPÍTULO

- Un elemento que, susceptible de adoptar uno de dos estados opuestos entre sí, aporta una información concreta. Ésa es la idea de un bit. A partir de esta unidad, es posible establecer un sistema de múltiplos.
- El sistema decimal responde a la suma de N paquetes donde cada uno está formado por el producto del dígito original por la base de numeración elevada a un peso determinado. Dicho peso se obtiene de la posición del dígito original, comenzando a contar de derecha a izquierda menos uno, es decir:

  $Número = N_n \cdot B^{Pn-1} + N_{n-1} \cdot B^{Pn-2} + \ldots$

  donde

  N_x *corresponde al dígito de posición x en el número N.*

  B *corresponde a la base de numeración que, en este caso, es 10.*

  $Pn-1$ *corresponde al peso del dígito de posición n-1.*
- Un sistema digital dispone de un número finito de valores para representar cantidades y, por tanto, es preciso realizar una aproximación. Un sistema analógico dispone de infinitas posibilidades de representación.
- Dentro de los diferentes códigos que existen se puede establecer una división para catalogarlos: los códigos ponderados (binario natural, el BCD o el AIKEN) y los no ponderados (exceso a 3 y código Gray).
- El código ASCII, formado por ocho bits, origina un total de 256 caracteres numerados del 0 al 255 (2^8=256) y recoge todas las posibilidades (o casi todas) de representación de un idioma.
- El código Unicode es un conjunto de caracteres mucho más amplio que el ASCII convencional donde el rango posible de caracteres se establece entre 0 y 65.536, como consecuencia de su formación por 16 bits (2^{16}=65.536).

EJERCICIOS PROPUESTOS

- **1.** Convierta de sistema decimal a sistema binario.
  a) 231
  b) 129
  c) 85
  d) 1

- **2.** Convierta de sistema binario a sistema decimal.
  a) 11100111
  b) 10000001
  c) 01010101
  d) 00000001

- **3.** Convierta de sistema binario a sistema octal.
  a) 11100111
  b) 10000001
  c) 01010101
  d) 00000001

- **4.** Convierta de sistema octal a sistema binario.
  a) 347
  b) 201
  c) 125
  d) 1

- **5.** Realice la siguiente suma en BCD: 2.325 + 4.127. Compruebe que el resultado BCD obtenido corresponde con el esperado en binario.

TEST DE CONOCIMIENTOS

1 Los sistemas digitales son más exactos que los analógicos:
a) Sí.
b) No.
c) La exactitud no es medible en estos términos.
d) Sería necesario evaluar el sistema en cuestión.

2 Para calcular un número binario a partir de otro en base 10:
a) Se realizan sucesivas divisiones por dos hasta llegar a un resto de 0 ó 1.
b) Se realizan sucesivas divisiones por 10 hasta llegar a un resto de 0 ó 1.
c) Se cambia el exponente de la operación.
d) Se realiza el complemento del número en base 2.

3 Por *complemento* de un número se entenderá aquel que:
- **a)** Restado del número original, obtenga la base de numeración.
- **b)** Sumado al número original, obtenga la base de numeración más la unidad.
- **c)** Sumado al número original, obtenga la base de numeración menos la unidad.
- **d)** Sumado al número original, obtenga la base de numeración.

4 El sistema octal:
- **a)** Trabaja con números del 0 al 8.
- **b)** Trabaja con números del 0 al 7.
- **c)** Trabaja con números del 1 al 8.
- **d)** Trabaja con números del 0 al 7.

5 El código ASCII recoge un total de:
- **a)** 64 caracteres.
- **b)** 128 caracteres.
- **c)** 256 caracteres.
- **d)** 512 caracteres.

6 El código Unicode:
- **a)** Recoge 65.536 caracteres.
- **b)** Es incompatible con el ASCII.
- **c)** Se trata del ASCII convencional.
- **d)** Se corresponde de forma directa con el ASCII en sus primeros 64 caracteres.

7 El número 43 se corresponde con el:
- **a)** 00101010 binario.
- **b)** 053 en octal.
- **c)** 2D en hexadecimal.
- **d)** 11010110 en BCD.

8 El código Aiken es:
- **a)** Un código sin peso.
- **b)** Ponderado en potencias en base 2 de 0, 1, 2…
- **c)** Ponderado como BCD en 2-4-2-1.
- **d)** Usa ocho números de base formados por tres dígitos.

9 El código Exceso a 3 es:
- **a)** Un código sin peso.
- **b)** Ponderado en potencias en base 2 de 0, 1, 2…
- **c)** Ponderado como BCD en 2-4-2-1.
- **d)** Usa 8 números de base formados por tres dígitos.

10 La resta en BCD se realiza:
- **a)** Con el CA2 del sustraendo.
- **b)** Con el CA9 del minuendo.
- **c)** Con la resta directa entre números.
- **d)** Como la suma de un número negativo.