

CAPÍTULO 3

REPRESENTACIÓN DE LA INFORMACIÓN EN LOS COMPUTADORES

*TECNOLOGÍA Y ORGANIZACIÓN DE
COMPUTADORES*

1º Grado en Ingeniería Informática.

Tema 3. Representación de la información en los computadores

RESUMEN:

- En este tema se analizan los aspectos relacionados con la representación de la información en el interior de los computadores.
- Se consideran la representación de textos, de sonidos, de imágenes y de valores numéricos.
- Para poder comprender esta representación, es necesario conocer previamente los sistemas de numeración desde el punto de vista de su aplicación en informática.

OBJETIVOS:

- Conocer las distintas formas básicas de representación de la información en un computador.

Tema 3. Representación de la información en los computadores.

CONTENIDOS:

- 3.0.Introducción
- 3.1.Representación de textos
- 3.2.Representación de sonidos
- 3.3.Representación de imágenes
- 3.4.Representación de video
- 3.5.Representación de datos numéricos

BIBLIOGRAFÍA:

- [PRI05]: Capítulo 2 (excepto Redondeos y algoritmos de compresión)
- [PRI06]: Capítulo 4 (excepto Redondeos y algoritmos de compresión)

Introducción

- Un computador es una máquina que procesa, memoriza y transmite **información**.
- La información se representa en el interior de la máquina de acuerdo con un **código binario**.
- La información se utiliza principalmente bajo las formas de:
 - Textos
 - Sonidos
 - Imágenes
 - Valores numéricos

Introducción

- Los sistemas que combinan textos, imágenes, sonidos y vídeo se denominan sistemas **multimedia**.
- En este capítulo vamos a estudiar cómo se representa la información de cada una de estas formas.
- Las aplicaciones multimedia se suelen almacenar en ficheros o enviarse por un canal de comunicación (streaming), con lo que
 - el volumen requerido en disco puede ser muy elevado
 - el tiempo de transmisión por red excesivo



Introducción

- Solución: **compresión de datos**
 - El archivo, antes de ser almacenado o transmitido se comprime mediante un algoritmo de compresión, y
 - cuando se recupera para procesarlo o visualizarlo se aplica la técnica inversa para descomprimirlo.
- Tipos:
 - Compresión sin pérdidas
 - Compresión con pérdidas (codificación perceptual)

Introducción

¿Qué quiere decir una compresión de 3:1? (f_c : 1)

Esta expresión indica que la capacidad antes (C_a) es 3 veces la de después de comprimirlo (C_d).

factor de compresión $\rightarrow f_c = \frac{C_a}{C_d}$

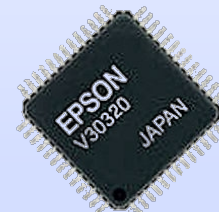
- Otra forma de representar la compresión es mediante el **porcentaje de compresión**:

porcentaje de compresión $\rightarrow p_c = \left[1 - \frac{C_d}{C_a} \right] \cdot 100 \% = \left[1 - \frac{1}{f_c} \right] \cdot 100 \%$

NOTA: Sólo para estos dos conceptos, seguir lo indicado en esta transparencia, independientemente de lo indicado en [PRI05] o [PRI06].

Introducción

- Un **CODEC** (COmprime ó COdifica/DEsComprime ó DECodifica): es software, hardware o mezcla de ambos que codifica en binario las muestras de la señal según el formato del fichero de salida y aplica un algoritmo de compresión (si es el caso), y a la inversa.



- Parámetros que caracterizan un CODEC:
 - **Número de canales:** una o más señales de audio simultáneamente: "mono" (un canal), "estéreo" (dos canales) o multicanal.
 - **Frecuencia de muestreo**
 - **Número de bits por muestra:** Determina la precisión con la que se reproduce la señal original y el rango dinámico de la misma.
 - **Algoritmo de compresión** (si hay compresión)

Tema 3. Representación de la información en los computadores

CONTENIDOS:

- 3.1. Representación de textos
- 3.2. Representación de sonidos
- 3.3. Representación de imágenes
- 3.4. Representación de video
- 3.5. Representación de datos numéricos

3.1.Representación de textos

- La información se suele introducir en el computador utilizando el lenguaje escrito:
 - **Caracteres alfabéticos:** letras mayúsculas y minúsculas del abecedario inglés {A, B, C, D, E,..., X, Y, Z, a, b, c, d,..., x, y, z}
 - **Caracteres numéricos:** {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
 - **Caracteres especiales:** símbolos no incluidos en los grupos anteriores. Por ejemplo: {) (, * / ; : + Ñ ñ = ! ? . " & > # <] Ç [SP }

3.1.Representación de textos

- **Caracteres geométricos y gráficos:** símbolos o módulos con los que se pueden representar figuras (o iconos). Por ejemplo: ♣ ♦ ♥ ♠ α β ∫ ∩ ∪ ∩ ∪ Σ
- **Caracteres de control:** representan órdenes de control, como:
 - Salto de línea,
 - sincronización de una transmisión,
 - pitido en un terminal, etc.

Muchos de los caracteres de control son generados e insertados por el propio ordenador.

3.1.Representación de textos

- En el ordenador toda la información se almacena en forma de 0s y 1s, así que hay que **traducir** ó **codificar** ó establecer una correspondencia entre los dos conjuntos:

$$\alpha \equiv \{A,B,C,D,\dots,Z,a,b,\dots,z,0,1,2,3,\dots,9,/,+,(,),\dots\} \rightarrow \beta \equiv \{0,1\}^n$$

de forma tal que a cada elemento de α le corresponda un elemento distinto de β .

3.1.Representación de textos

- Si tenemos que codificar o traducir un conjunto de **m** símbolos (α) ¿cuántos bits, **n**, necesitaremos?
- El número mínimo de bits, **n**, dependerá del número de elementos, **m**, a codificar.
 - Con $n=2$ bits se pueden hacer $2^2=4$ combinaciones → se pueden codificar hasta $m=4$ símbolos.
 - Con $n=3$ bits podemos hacer $2^3=8$ combinaciones → se pueden codificar hasta $m=8$ símbolos.
 - Con n bits podemos hacer 2^n combinaciones → se pueden codificar hasta $m=2^n$ símbolos.

- Es decir:

$$\mathbf{m \leq 2^n \text{ ó } n \geq \log_2(m) = 3.32 \log(m)}$$

3.1.Representación de textos

- Los códigos que llevan a cabo esta codificación se denominan **códigos de E/S** o códigos externos o códigos-texto, y pueden definirse de forma arbitraria.
- No obstante, existen códigos de E/S **normalizados** que son utilizados por diferentes constructores de computadores. Por ejemplo: BCD de intercambio normalizado, Fieldata, EBCDIC, ASCII, etc.



3.1.Representación de textos

- Ejemplos de códigos normalizados:
 - CÓDIGO SBCD (6 bits → 64 caracteres)
 - CÓDIGO EBCDIC (8 bits → 256 caracteres)
 - CÓDIGO ASCII
 - UNICODE



3.1.Representación de textos

- **Código ASCII** (American Standard Code for Information Interchange).
 - La mayor parte de las transmisiones de datos entre dispositivos se realizan en esta codificación.
 - Hay distintas versiones.
 - La versión ASCII ANSI-X3.4 (1968) utiliza 7 bits y ha sido de los más usuales.
 - Existen numerosas versiones ampliadas que utilizan 8 bits y respetan los códigos del ASCII básico.

3.1.Representación de textos

- ASCII (Ampliaciones)

Denominación	Estándar	Área geográfica
Latín-1	ISO 8859-1	Oeste y Europa del este
Latín-2	ISO 8859-2	Europa central y del este
Latín-3	ISO 8859-3	Europa sur, maltés y esperanto
Latín-4	ISO 8859-4	Europa norte
Alfabeto latín/cirílico	ISO 8859-5	Lenguajes eslavos
Alfabeto latín/árabe	ISO 8859-6	Lenguajes árabigos
Alfabeto latín/griego	ISO 8859-7	Griego moderno
Alfabeto latín/hebraico	ISO 8859-8	Hebreo y Yiddish
Latín-5	ISO 8859-9	Turco
Latín-6	ISO 8859-10	Nórdico (Sámi, Inuit e islandés)
Alfabeto Latín/Thai	ISO 8859-11	Lenguaje Thai
Latín-7	ISO 8859-13	Báltico <i>Rim</i>
Latín-8	ISO 8859-14	Céltico
Latín-9 (<i>alias Latín-0</i>)	ISO 8859-15	Latín 1 con ligeras modificaciones (símbolo €)

3.1.Representación de textos

ASCII (ISO 8859-1, Latín 1)

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00	0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10	16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20	32	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
80	128																
90	144																
A0	160		ı	ş	£	¤	¥	¦	§		©	ª	«	¬	®	¯	
B0	176	°	±	²	³	´	µ	¶	·	,	:	»	¼	½	¾	¿	
C0	192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï

$$G = 40 + 7 = 47)_{16} = 0100\ 0111)_2$$

3.1.Representación de textos

- **Inconvenientes de los códigos anteriores** (sobre todo con Internet):
 - Los símbolos codificados son **insuficientes** para representar los caracteres especiales que requieren numerosas aplicaciones.
 - Los símbolos y códigos añadidos en las versiones ampliadas a 8 bits **no están normalizados**.
 - Están basados en los caracteres latinos, existiendo otras culturas que utilizan otros símbolos muy distintos.
 - Los lenguajes escritos de diversas culturas orientales, como la china, japonesa y coreana se basan en la utilización de **ideogramas** o símbolos que representan palabras, frases o ideas completas, siendo, por tanto, inoperantes los códigos que sólo codifican letras individuales.

3.1.Representación de textos

- **Unicode (ISO/IEC 10646)** es propuesto por un consorcio de empresas y entidades que trata de hacer posible escribir aplicaciones que sean capaces de procesar texto de muy diversas culturas. Trata de conseguir:
 - **Universalidad**, trata de cubrir la mayoría de lenguajes escritos existentes en la actualidad,
 - **Unicidad**, a cada carácter se le asigna exactamente un único código y
 - **Uniformidad**, ya que todos los símbolos se representan con un número fijo de bits (16).



3.1.Representación de textos

Características:

- Cada carácter se codifica con 16 bits $\Rightarrow 2^{16} = 65.356$ símbolos
- No se codifican los caracteres de control
- Incluye caracteres combinados: ñ, ä, ç, etc.
- No determina la forma o imagen concreta de cada carácter
- Para evitar duplicidades, caracteres parecidos en distintos idiomas tienen igual posición en el código

3.1.Representación de textos

Asignación de códigos **UNICODE**

ZONA	CÓDIGO	SÍMBOLO	Nº CARACT
A	0000	ASCII Latín-1	1681
	3FFF	Caracteres latinos, griegos, cirílicos, armenios, hebreos, árabes, sirios, etc. Símbolos generales, caracteres fonéticos chinos, japoneses, coreanos	
I	4000 9FFF	Ideogramas	24576
O	A000 DFFF	Pendiente de asignación	16384
R	E000 FFFF	Caracteres locales y propios de los usuarios. Compatibilidad con otros códigos	8192

3.1.Representación de textos

UNICODE (zona A):

Rango Unicode	Se corresponde con
0000 a 007F	Latín Básico (00 a 7F), definidos en la norma ASCII ANSI-X3.4.
0080 a 00FF	Suplemento Latín-1 (ISO 8859-1)
0100 a 017F	Ampliación A de Latín
0180 a 024F	Ampliación B del Latín
0250 a 02AF	Ampliación del Alfabeto Fonético Internacional (IPA)
02BF a 02FF	Espaciado de letras modificadoras
0300 a 036F	Combinación de marcas diacríticas (tilde, acento grave, etc.)
0370 a 03FF	Griego
0400 a 04FF	Cirílico
0530 a 058F	Armenio
0590 a 05FF	Hebreo
0600 a 06FF	Árabe
0700 a 074F	Sirio
etc.	etc.

3.1.Representación de textos

ZONA A: LATÍN
ZONA B: Cancheros y locales



3.1.Representación de textos

- **DETECCIÓN DE ERRORES:**

- A veces, al codificar, se introducen **redundancias** (bits extras) de acuerdo con algún algoritmo predeterminado par que los códigos pueden ser verificados automáticamente.
- Por ejemplo, en ASCII, se suele incluir un octavo bit de paridad.
- Uno de estos algoritmos añade al código inicial de cada carácter un nuevo bit: el **bit de paridad**. Existen dos criterios para introducir este bit:
 - **Bit de paridad, criterio par:** se añade un bit (0 ó 1) de forma tal que el número total de unos de código que resulte sea par.
 - **Bit de paridad, criterio impar:** se añade un bit (0 ó 1) de forma tal que el número total de unos del código que resulte sea impar.

3.1.Representación de textos

- Ejemplos:

<i>Mensaje inicial</i>
100 0001
101 1011
101 0000
110 1000

<i>Mensaje con bit de paridad (criterio par)</i>
0100 0001
1101 1011
0101 0000
1110 1000

bit de paridad ↗

<i>Mensaje inicial</i>
000 0000
100 0001
101 1011
101 0000
110 1000

<i>Mensaje o con bit de paridad (criterio impar)</i>
1000 0000
1100 0001
0101 1011
1101 0000
0110 1000

bit de paridad ↗

Tema 3. Representación de la información en los computadores.

CONTENIDOS:

3.1.Representación de textos

3.2.Representación de sonidos

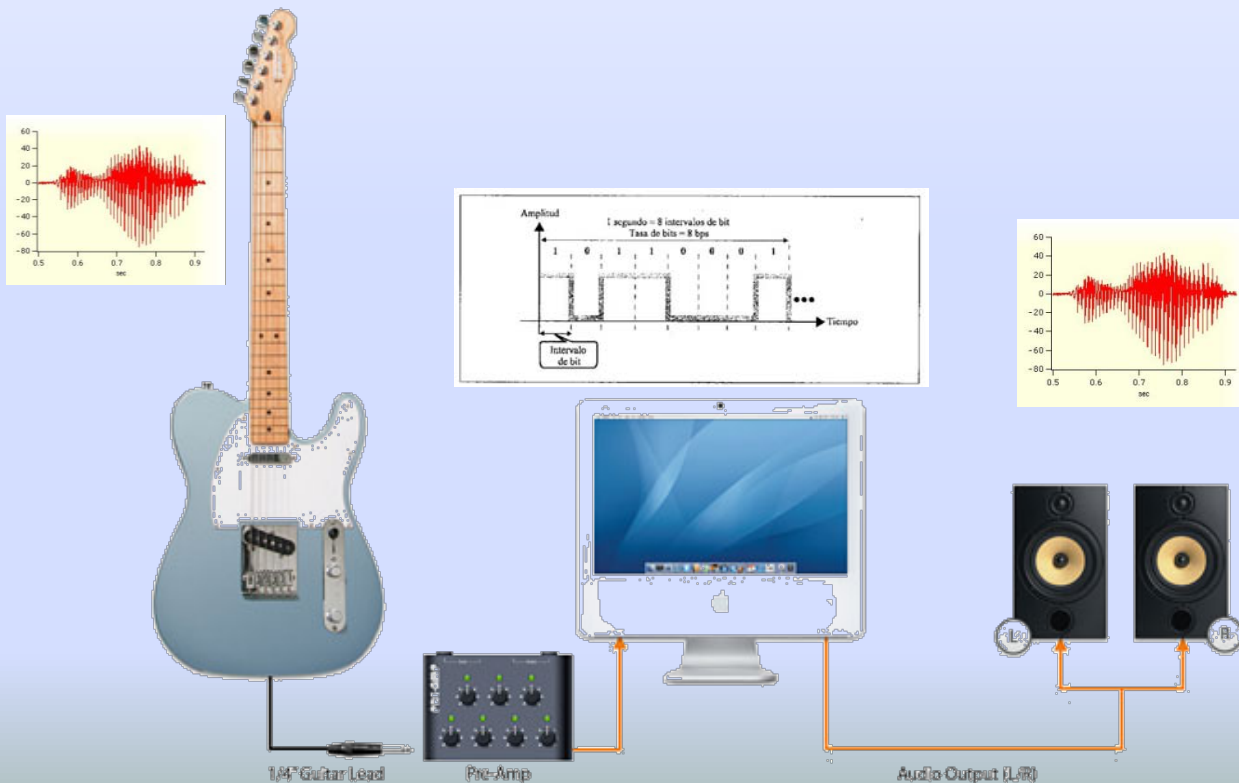
3.3.Representación de imágenes

3.4.Representación de video

3.5.Representación de datos numéricos

3.2.Representación de sonidos

- ¿Cómo se graba, almacena y reproduce una señal de audio en un computador?

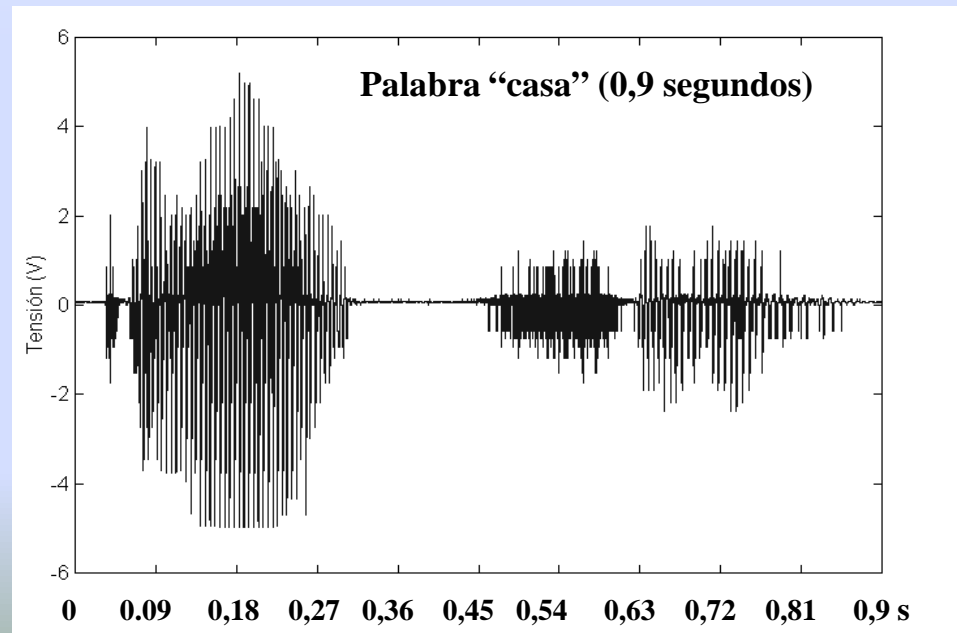


3.2.Representación de sonidos

- **Grabación de una señal de audio:**

1. Una señal de audio **se capta** por medio de un micrófono que produce una **señal analógica** (señal que puede tomar cualquier valor dentro de un determinado intervalo continuo).

La señal analógica se amplifica para encajarla dentro de dos valores límites,
(Ej. Entre $-5V$ y $+5V$)



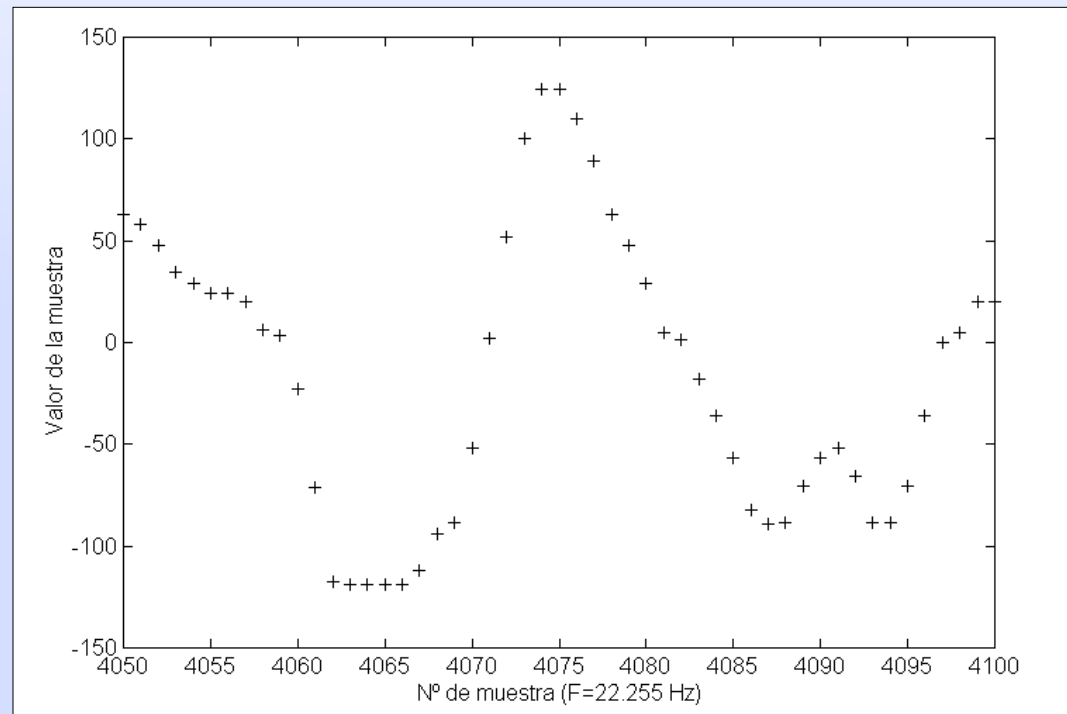
3.2.Representación de sonidos

3. Por medio de un conversor A/D **se muestrea y digitaliza**

- Frecuencia de muestreo:
 F_s (22,05 KHz); periodo
de muestreo:

$$T_s = \frac{1}{F_s} = 0.045 \text{ ms}$$

- En la figura: muestras de
la 4050 a la 4100 (0,184
a 0,186 segundos)



3.2.Representación de sonidos

Almacenamiento de la señal de audio:

- La **señal de sonido** queda representada por una serie de valores de 8 bits que corresponden a cada muestra de la señal analógica y que se almacenan en un archivo.

Posición	contenido	posición	contenido
1	2	4057	20
2	2	4058	6
3	2	4059	3
4	2	4060	-23
.	.	4061	-71
.	.	4062	-118
4050	63	.	.
4051	58	.	.
4052	48	19996	1
4053	35	19997	1
4054	29	19998	1
4055	24	19999	1
4056	24	20000	1

(c)

- Los valores obtenidos en la conversión (muestras) se almacenan en posiciones consecutivas.
- Antes de las muestras se incluye una cabecera, con información sobre el tipo de fichero y sus parámetros.

3.2.Representación de sonidos

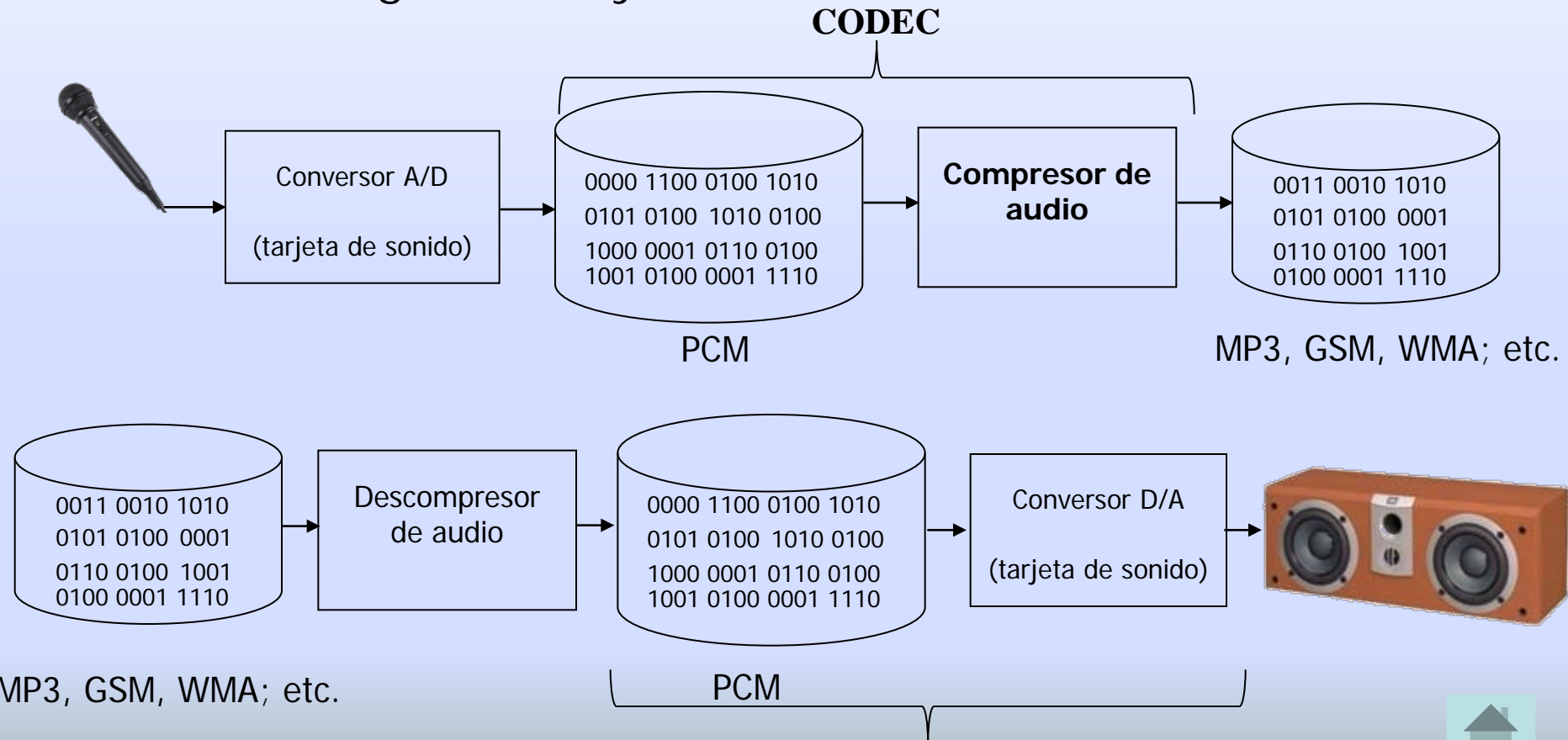
- La **calidad y capacidad** necesaria para almacenar una señal de audio dependen de los siguientes parámetros:
 - **Frecuencia de muestreo** (suficiente para no perder la forma de la señal original)
 - **Número de bits por muestra** (precisión)
- Tasa de datos:** $T = f_s \times N \times C$

N: bits o Bytes/muestra, f_s : frecuencia de muestreo, c: número de canales

Aplicación	F_s (KHz)	N B/muestra	Nº de canales	T_b (Kbps)	Capacidad 1 minuto
Telefonía	8	1	1	62,5 Kbps	468,7 KB
Radio AM	11	1	1	85,9 Kbps	644,5 KB
Radio FM	22,05	2	2	689,1 Kbps	5,05 MB
CD	44,1	2	2	1,35 Mbps	10,1 MB
TDT	48	2	2	1,46 Mbps	10,99 MB

3.2.Representación de sonidos

- Proceso de grabación y almacenamiento:



3.2.Representación de sonidos

FORMATOS DE AUDIO		
Sin compresión	Con compresión	
	Sin pérdidas	Con pérdidas
Audio-CD, WAV, AU, AIFF, ...	ALAC, DST, FLAC, LA, LPAC, LTAC, MPL, MPEG-4, ...	MP1, MP2, MP3, DTS, OGG, WMA, AAC, AC3, ADPCM, ...

3.2.Representación de sonidos

- Ejemplo: fichero de audio en diversos formatos
 - Estéreo, $F_s = 44100$ Hz, 32 bits/muestra

 PetShopBoys-BlueSky.aiff	41.176 KB	Sonido en formato AIFF
 PetShopBoys-BlueSky.au	41.176 KB	Sonido en formato AU
 PetShopBoys-BlueSky.mp3	3.731 KB	Sonido en formato MP3
 PetShopBoys-BlueSky.raw	41.175 KB	Archivo RAW
 PetShopBoys-BlueSky.wav	41.176 KB	Archivo de sonido
 PetShopBoys-BlueSky.wma	3.761 KB	Archivo de audio de Windows Media

- El formato mp3 es un formato de compresión de audio estándar ISO .
 - Cada canal se codifica con un n° de bits proporcional a su amplitud
 - Se consigue hasta un 92% de compresión

Tema 3. Representación de la información en los computadores.

CONTENIDOS:

- 3.1.Representación de textos
- 3.2.Representación de sonidos
- 3.3.Representación de imágenes**
- 3.4.Representación de video
- 3.5.Representación de datos numéricos

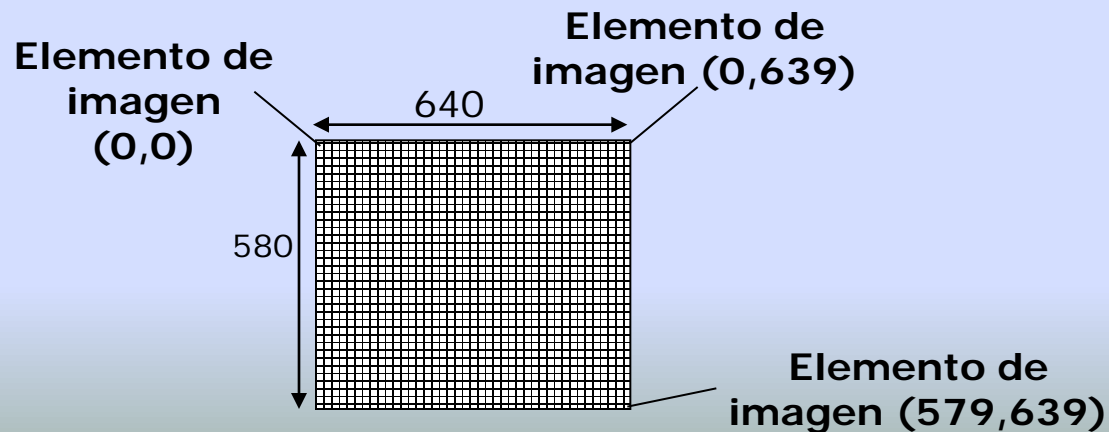
3.3.Representación de imágenes

- Las **imágenes** se adquieren por medio de periféricos tales como escáneres, cámaras de video o cámaras fotográficas.
- Una imagen se representa por **patrones de bits**, generados por el periférico correspondiente.
- **Formas básicas de representación:**
 - Mapa de bits
 - Mapa de vectores

3.3.Representación de imágenes

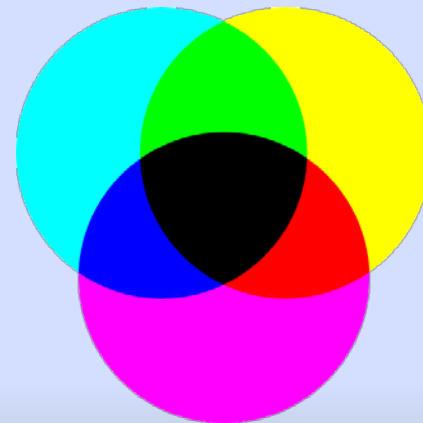
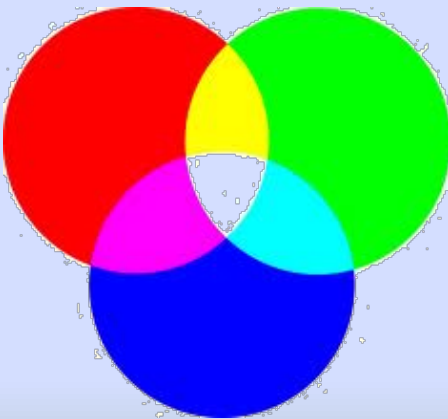
MAPAS DE BITS

- La imagen se divide en una fina retícula de celdas o **elementos de imagen** o **píxeles**.
- La **resolución** es el número de e.i. horizontales x nº e.i. verticales.
- A cada elemento de imagen (e.i.) se le asocia un valor, **atributo**, que se corresponde con su nivel de gris (b/n) o color, medio en la celda.



3.3.Representación de imágenes

- El **color en las pantallas** se codifica con las intensidades de tres colores básicos: **R** (rojo), **G** (verde) y **B** (azul) (RGB)
- El **color en papel** se codifica con las intensidades de 4 colores básicos sustractivos: **C** (Cian), **M** (Magenta), **Y** (Amarillo) y **K** (Negro)



3.3.Representación de imágenes

- La imagen se **memoriza**, almacenando de forma ordenada y sucesiva los atributos de los distintos elementos de imagen, precedidos de una cabecera con información sobre la imagen.

posición	Contenido (código de color)
0	00101110
1	01011011
2	00101110
3	01001111
...	...
...	...
371.196	00101110
371.197	00011011
371.198	01001110
371.199	11001110

(b)

3.3.Representación de imágenes

- La **calidad** de la imagen depende de
 - La resolución y
 - Codificación del atributo (número de bits)
- **La capacidad** depende de esos parámetros:
 - Por ejemplo, una imagen de 16 niveles de grises (b/n) y con resolución de 640x350: 110 KBytes
 - Por ejemplo, una imagen con resolución XGA (1024X768) con 256 niveles (32 bits) para cada color básico: 2,25 MBytes (≈ 20 veces más)

3.3.Representación de imágenes

- Resoluciones usuales para codificar imágenes:

		Resolución (horizontal x vertical)
Convencionales	Fax (A4)	(100,200,400)x(200,300,400) ei/''
	Foto (8'' x 11'')	128,400,1200 ei/''
Televisión	Videoconferencia	176 x 144
	TV	720 x 480 NTSC 720 x 576 PAL
	HDTV (alta definición) 16:9	1920 x 1080 1280 x 720

TV: Resolución en ei/imagen

3.3.Representación de imágenes

Formatos de Mapas de bits:

FORMATOS DE IMÁGENES		
Sin compresión	Con compresión	
	Sin pérdidas	Con pérdidas
BMP, TIFF	GIF, PNG	JPEG

3.3.Representación de imágenes

- PRUEBAS DE OCUPACIÓN REALIZADOS CON PHOTOSHOP 7.0

Imagen de 1024 x 768 píxeles

- BMP (profundidad de color, 32 bits): 3MB
- JPEG (calidad óptima): 523 KB
- TIFF: 3MB
- GIF (paleta de 256 colores): 550 KB
- GIF (paleta de 128 colores): 461 KB
- PNG-24: 1,15 MB
- PNG-8 (256 colores): 530 KB

3.3.Representación de imágenes

- Foto de un paisaje (802 X 416 pix):
TIFF: 996 KB, BMP: 2.305 KB, PNG: 1.783 KB, GIF: 405 KB, JPG: 293 KB



3.3.Representación de imágenes

- Foto de un paisaje:
 - JPG: 12,84 KB (calidad baja)

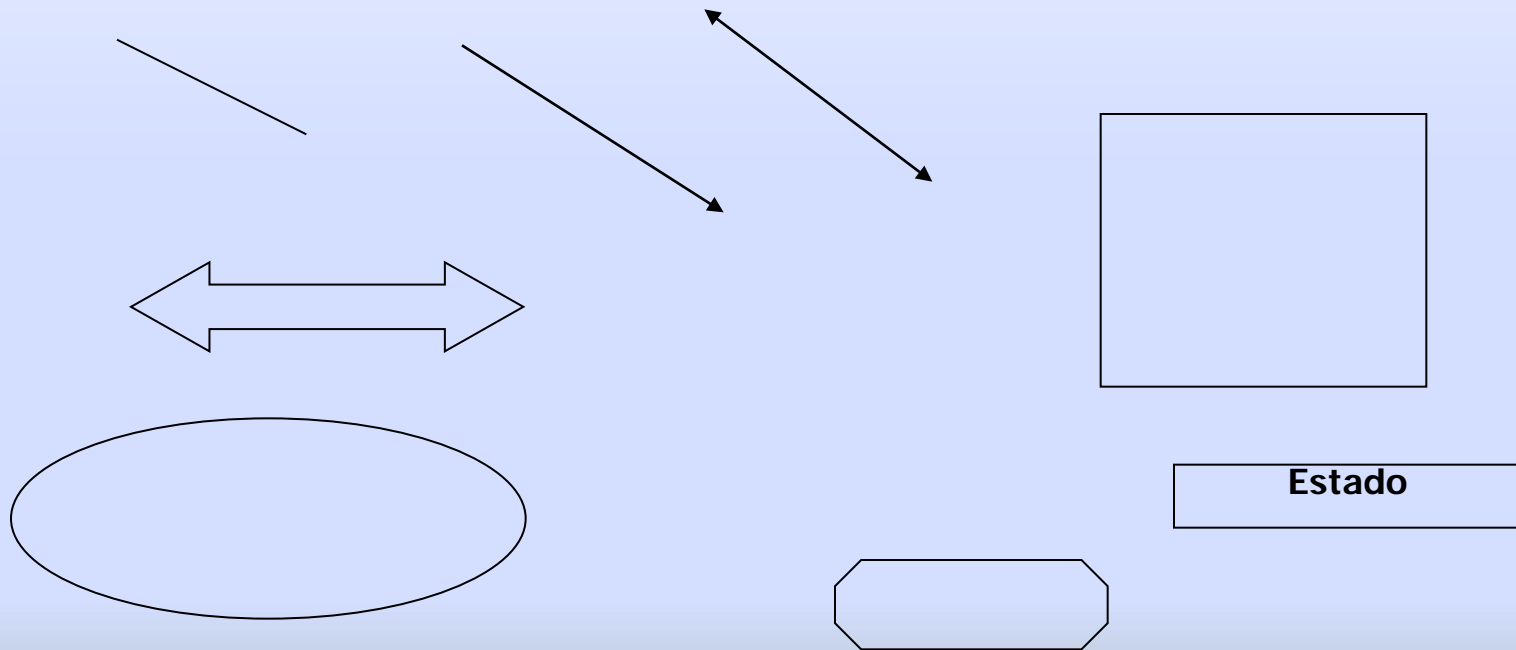


3.3.Representación de imágenes

- **Gráficos vectoriales o gráficos orientados a objetos**
 - Se descompone la imagen en una colección de **objetos** tales como líneas, polígonos y textos con sus respectivos atributos o detalles (grosor, color, etc.) modelables por medio de vectores y ecuaciones matemáticas que determinan tanto su forma como su posición dentro de la imagen.
 - Se almacena el código del objeto y sus parámetros (no los puntos)
 - Para visualiza una imagen, un programa evalúa las ecuaciones y escala los vectores generando la imagen concreta a ver.
 - Ocupa menos que un mapa de bits, y
 - Es más rápido hacer cambios de escala y representarlos en pantalla (*rastrering*).
 - Adecuada para gráficos de tipo geométrico (no imágenes reales): CAD/CAM, esquemas, logotipos, etc.

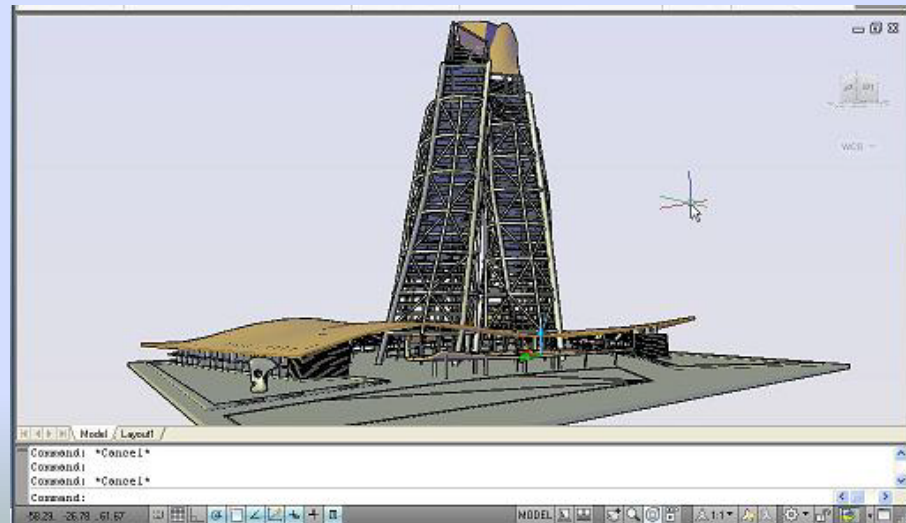
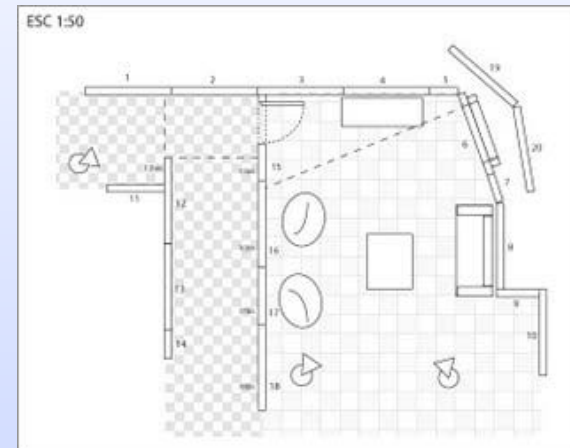
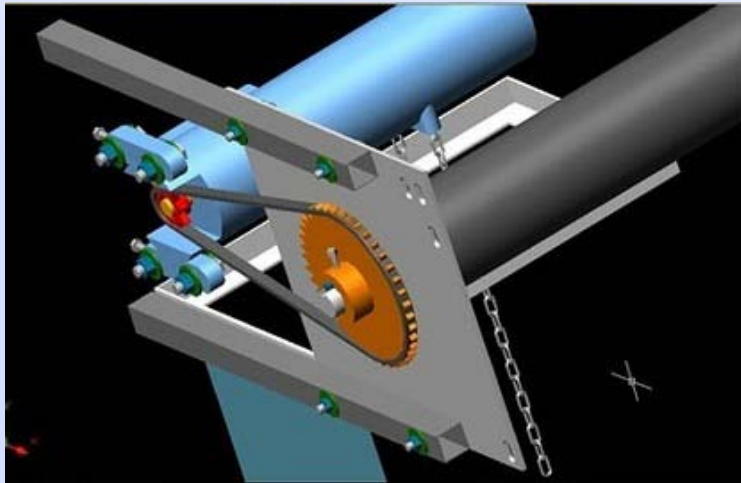
3.3.Representación de imágenes

- Dibujos hechos con **primitivas geométricas** (objetos): líneas y polilíneas, Polígonos, Círculos y elipses, Curvas de Bézier, Texto



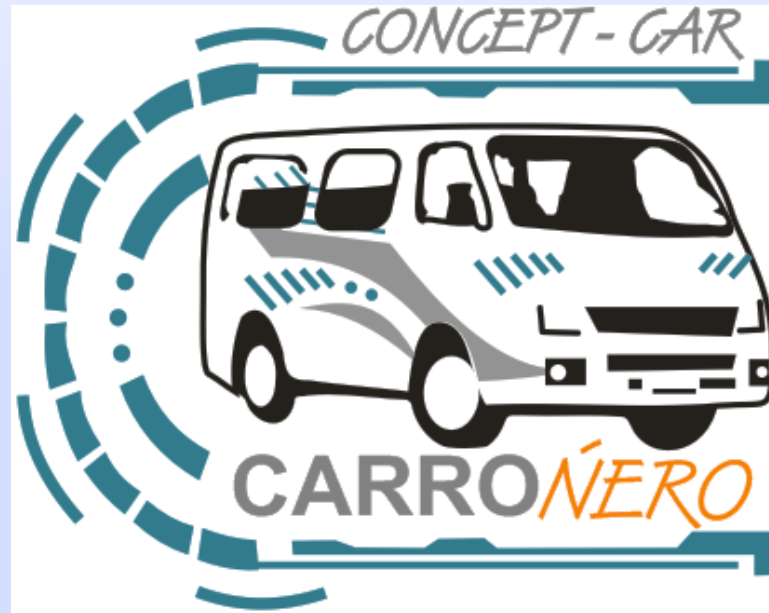
3.3.Representación de imágenes

- Ejemplos de imágenes con primitivas geométricas:



3.3.Representación de imágenes

- Ejemplos vectorización:



3.3.Representación de imágenes

Formato	Origen	Descripción
IGES (Initial Graphics Exchange Specification)	ASME/ANSI	Estándar para intercambio de modelos y datos CAD (usable en AutoCAD, etc.)
DXF (Document eXchange Format)		Formato original del AutoCAD
PICT (PICTure)	Apple Comp.	Imágenes vectoriales que pueden incluir objetos que son imágenes en mapa de bits
EPS (Encapsulated Postscript)	Adobe Sys.	Ampliación para imágenes del lenguaje de impresión Postscript, con la que se pueden insertar imágenes en distintos formatos como TIFF, WMF, PICT o EPSI
TrueType	Apple comp..	Alternativa de Apple y Microsoft para el EPS

Tema 3. Representación de la información en los computadores.

CONTENIDOS:

- 3.1.Representación de textos
- 3.2.Representación de sonidos
- 3.3.Representación de imágenes
- 3.4.Representación de video
- 3.4.Representación de datos numéricos

3.4 Representación de vídeo

- **Vídeo:** sucesión de imágenes a una determinada frecuencia (fotogramas por segundo, fps), con sonido.

	fps
Imagen en movimiento	15
Cine	24
TV	25 (PAL) 30 (NTFS)
HDTV	60

- Todos los formatos de vídeo llevan compresión.

3.4 Representación de vídeo

- Formatos de vídeo:

DV (Digital Video)	Estándar internacional 1996
MPEG (Motion JPEG)	estándar de codificación de audio y vídeo normalizado
MPEG-1, MPEG-2, MPEG-4	estándares ISO evolucionados del MPEG
WMV (Windows media video)	CODEC de Microsoft
DivX	Compresión de audio MP3, codifica y comprime de forma que ocupa un DVD de 80 min.
XviD	CODEC libre basado en MPEG-4
SWF (ShockWaveFlash)	Formato completo para multimedia muy extendido en la web desarrollado por Adobe.

3.4 Representación de vídeo

- Los vídeos no sólo contienen imágenes en movimiento sino audio, subtítulos, etc.
- **Metafichero**: un fichero compuesto por varios ficheros de distintos tipos (audio, vídeo, subtítulos, etiquetas, etc.) así como metadatos con información.
- **Contenedor**: contiene la descripción de un metafichero.

Contenedores de vídeo	
MP4	Contenedor del formato MPEG-4
3GP	Adaptación de MP4 para servicios multimedia UMTS
AVI	Contenedor de Microsoft (1992)
ASF	Microsoft. Diseñado para streaming
FLV	Contenedor de Adobe Flash
Matroska	Estándar de código abierto
Ogg	Contendor libre de Xiph.org
QuickTime	Desarrollado por Apple

Tema 3. Representación de la información en los computadores.

CONTENIDOS:

- 3.1.Representación de textos
- 3.2.Representación de sonidos
- 3.3.Representación de imágenes
- 3.4.Representación de video
- 3.5.Representación de datos numéricos

3.5 Representación de datos numéricos

- Los datos se introducen en el ordenador en **lenguaje escrito** y por tanto se codifican como cualquier texto según el código de E/S. Es decir, los números son tratados y codificados como caracteres de texto.
- Esta codificación es inapropiada para operar, ya que no se basa en un sistema de numeración matemático.
- Si un número se va a utilizar en un programa como un dato numérico, el ordenador efectúa una transformación entre códigos binarios, obteniéndose una representación en el sistema de numeración en base 2, y, por tanto, apta para realizar operaciones aritméticas.

3.5 Representación de datos numéricos

- **Ejemplo:** $255 + 42 = 297$

- Notación en ASCII:

$255 = (0011\ 0010\ 0011\ 0101\ 0010\ 0101)_{\text{ASCII}}$

$42 = (0011\ 0100\ 0011\ 0010)_{\text{ASCII}}$

$$\begin{array}{r} \\ + \\ \hline 0011\ 0010\ 0110\ 1001\ 0101\ 0111 \end{array} \rightarrow 2iW)_{\text{ASCII}} \rightarrow \text{MAL!}$$

- Notación aritmética (binario natural):

$$\begin{array}{rcl} + & 255 & = & 1111\ 1111 \\ & 42 & = & 0010\ 1010 \\ \hline & 297 & = & 1\ 0010\ 1001 \end{array}$$

- Las cantidades ocupan menos!
- Algoritmos muy eficientes para hacer operaciones aritméticas!

3.5 Representación de datos numéricos

- Al introducir un número en el ordenador se codifica y almacena como un texto cualquiera.
- Cuando un programa va a utilizar un dato, según las operaciones que se vayan a realizar con él, el programador le asocia un tipo u otro.
- Los lenguajes de programación contienen reglas para poder determinar si un dato concreto se va a utilizar como texto, como número, como número real, etc.
- Cuando se traduce el programa a lenguaje máquina o cuando se ejecuta, los datos se transforman al tipo especificado por el programador de forma que se realicen las operaciones con ellos de forma adecuada.

3.5 Representación de datos numéricos

3.5.1. Datos de tipo entero representados en binario

- Enteros sin signo: valor absoluto
- Enteros con signo
 - Signo y magnitud
 - Complemento a uno
 - Complemento a dos
 - Sesgada

3.5.2. Datos de tipo entero representados en BCD

3.5.3. Datos de tipo real

3.5.1 Datos de tipo entero representados en binario

- **Enteros sin signo:**
 - Los n bits representan el valor absoluto del número.
 - Por ejemplo, si $n=8$:

Nº decimal	Enteros sin signo
	Valor absoluto
24	00011000

3.5.1 Datos de tipo entero representados en binario

- Enteros con signo:

		(1 bit)	(n-1 bits)
Signo y magnitud:		0/1	Valor absoluto de N
Complemento a 1:	$N > 0$	0	Valor absoluto de N
	$N < 0$	1	Complemento a 1 de N
Complemento a 2:	$N > 0$	0	Valor absoluto de N
	$N < 0$	1	Complemento a 2 de N
Sesgado:	$S = 2^{n-1}$	$N + S$	

3.5.1 Datos de tipo entero representados en binario

- **Ejemplo:** obtener la representación en las cuatro formas vistas del número entero $N = 87$ con $n = 8$ bits.

$$87)_{10} = 57)_{16} = 01010111)_2$$

$$\begin{array}{r|l} 87 & 16 \\ \hline 07 & 5 \end{array}$$

- Signo y magnitud: como $N > 0 \rightarrow S = 0$
 $N = 01010111$
- Complemento a 1: como $N > 0$, $N = |N|$
 $N = 01010111$
- Complemento a 2: como $N > 0$, $N = |N|$
 $N = 01010111$
- Sesgada: $N + S$ donde $S = 2^{n-1} = 2^7 = 10000000$
 $N = 01010111 + 10000000 = 11010111$

3.5.1 Datos de tipo entero representados en binario

- **Ejemplo:** obtener la representación en las cuatro formas vistas del número entero $N = -87$ con $n = 8$ bits.

$$87)_{10} = 57)_{16} = 01010111)_2$$

$$\begin{array}{r|l} 87 & 16 \\ \hline 07 & 5 \end{array}$$

- Signo y magnitud: como $N < 0 \rightarrow S = 1$
 $N = 11010111$
- Complemento a 1: como $N < 0$, $C_1(|N|)$
 $N = 10101000$
- Complemento a 2: como $N < 0$, $C_2(|N|)$
 $N = 10101001$
- Sesgada: $N + S$ donde $S = 2^{n-1} = 2^7 = 10000000$
 $N = -01010111 + 10000000 = 00101001$

3.5.1 Datos de tipo entero representados en binario

- Representaciones de datos de n=4 bits de tipo entero:

Nº Decimal	Sin signo	Signo y magnitud	Complemento a 1	Complemento a 2	Sesgada
(8-15)	...	no	no	no	no
+7	0111	0111	0111	0111	1111
+6	0110	0110	0110	0110	1110
+5	0101	0101	0101	0101	1101
+4	0100	0100	0100	0100	1100
+3	0011	0011	0011	0011	1011
+2	0010	0010	0010	0010	1010
+1	0001	0001	0001	0001	1001
+0	0000	0000	0000	0000	1000
-0		1000	1111	--	--
-1		1001	1110	1111	0111
-2		1010	1101	1110	0110
-3		1011	1100	1101	0101
-4		1100	1011	1100	0100
-5		1101	1010	1011	0011
-6		1110	1001	1010	0010
-7		1111	1000	1001	0001
-8				1000	0000

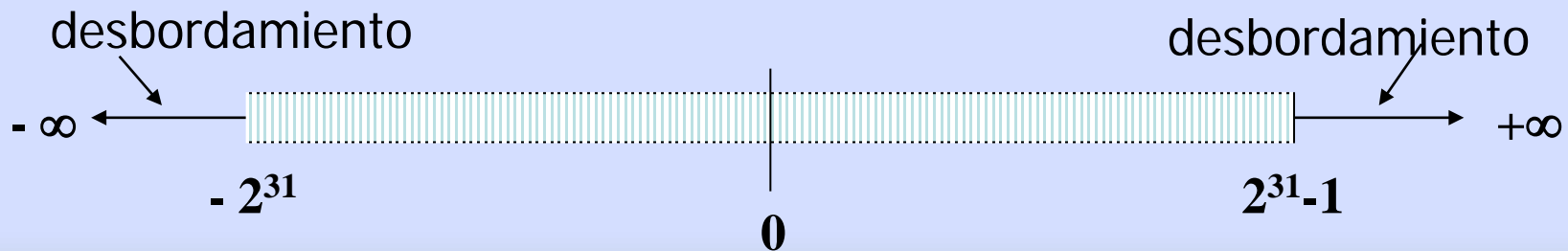
Sesgo=8

3.5.1 Datos de tipo entero representados en binario

- Si como resultado de una operación se obtiene un número fuera de los límites máximo y mínimo se dice que se ha producido un **desbordamiento**.
- Por ejemplo: si $n=32$ bits, en complemento a 2:

$$N(\text{máximo}) = 2^{31}-1 = 2.147.483.647$$

$$N(\text{mínimo}) = -(2^{31}) = - 2.147.483.648$$



3.5.2 Datos de tipo entero representados en BCD

Representación de dígitos decimales codificados en binario (BCD):

- Se codifica aisladamente cada dígito decimal con cuatro dígitos binarios.
- Por ejemplo:

$$0111\ 0010\ 1001)_{\text{BCD}} = 729)_{10}$$

$$3795)_{10} = 0011\ 0111\ 1001\ 0101)_{\text{BCD}}$$

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

3.5.3 Datos de tipo real

- Un **número real** se puede representar de distintas formas, por ejemplo:

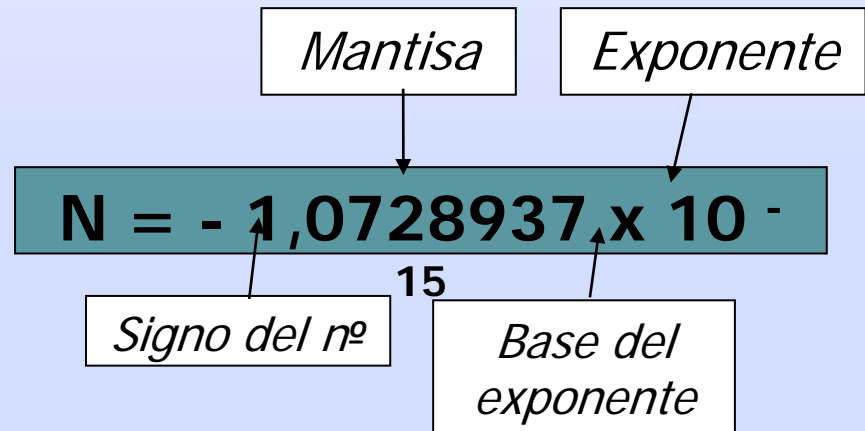
$$N = 3257,3285 = 3257,3285 \cdot 10^0 = 3,2573285 \cdot 10^3 = \\ = 32573285 \cdot 10^{-4} = 3257328900 \cdot 10^{-6} = \dots$$

Se dice que el numero está **normalizado** cuando la cifra mas significativa esta en la posición de las unidades:

$$N = 3,2573285 \cdot 10^3$$

3.5.3 Datos de tipo real

- Es decir, podemos transformar la representación de un número real, N , conservando su valor, cambiando el exponente, E , y reajustando adecuadamente la mantisa, M .



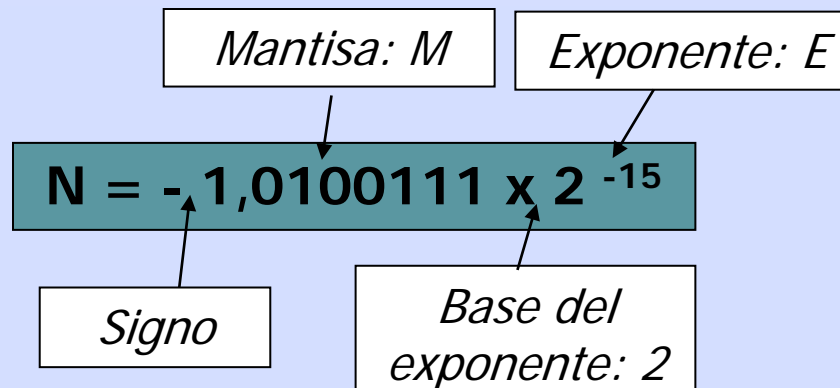
- Denominación:
 - notación exponencial,
 - notación científica
 - notación en punto o coma flotante.

$$N = \pm M \cdot B^E$$

\pm : signo del número, M : mantisa, B : base, E : exponente

3.5.3 Datos de tipo real

- Vamos a ver la **Normalización IEEE 754**, que tiene una aceptación prácticamente universal.
- Se transforma N a binario natural, con base del exponente $B=2$: $N = \pm M \cdot 2^E$
- Por ejemplo:



3.5.3 Datos de tipo real

- **Campo del signo, s (1 bit):** El bit de signo es cero para los números positivos y uno para los números negativos.
- **Campo del exponente, e (ne bits):** El exponente se almacena en forma de “entero sesgado”:

$$e = S + E = 2^{ne-1} - 1 + E$$

- **Campo de la mantisa, n (nm bits):** parte fraccionaria de la mantisa (M) normalizada.

(1 bit) signo	(ne bits) exponente	(nm bits) mantisa
s	e	m
0/1	Exponente sesgado	Parte fraccionaria de la mantisa normalizada

3.5.3 Datos de tipo real

- El estándar IEEE 754 considera cuatro tamaños o precisiones posibles de datos: **simple precisión** (n=32), **simple ampliada**, **doble** (n=64), y **doble ampliada**; aunque sólo especifica completamente las precisiones sencilla y doble.

Tipos de precisión contemplados en el estándar IEEE 754

Precisión	Simple	Simple ampliada	Doble	Doble ampliada
nm + 1	24	32	53	64
Exponente máximo	127	≥ 1023	1023	≥ 16383
Exponente mínimo	-126	≤ -1022	-1022	-16 382
S (sesgo del exponente)	127	(n.e.)	1 023	(n.e.)

(n.e.: no especificado por el estándar)

- Simple precisión:
n=32, ne=8 y nm=23, sesgo: $S = 2^7 - 1 = 127$.

3.5.3 Datos de tipo real

Ejemplo: Obtener la representación interna del número decimal $-632 \cdot 10^{-16}$ según la norma IEEE 754 simple precisión ($n=32$, $n_e=8$ y $n_m=23$).

- Normalización del número: $N = -632 \cdot 10^{-16} = -6,32 \cdot 10^{-14}$
- Pasar a la forma: $N = \pm M \cdot 2^E$

$$10^{-14} = 2^x \rightarrow x = -14 \frac{\log(10)}{\log(2)} = -46,5069933284$$

el exponente tiene que ser entero:

$$N = -6,32 \cdot 2^{-46,506993} = -6,32 \cdot 2^{-0,506993} \cdot 2^{-46} = -4,4473046 \cdot 2^{-46}$$

El número a almacenar es:

$$\mathbf{N = -4,4473046 \cdot 2^{-46}}$$

3.5.3 Datos de tipo real

- **Signo:** negativo $\rightarrow S=1$
- **Mantisa:** hay que obtener 23 bits de mantisa; paso la mantisa a binario.

$$N = -4,4473046 \cdot 2^{-46})_{10}$$

	HEX	BIN
4	= 4	$\rightarrow 4 \rightarrow 0100$
$0,4473046 \times 16 = 7,1568736$	$\rightarrow 7$	$\rightarrow 0111$
$0,1568736 \times 16 = 2,5099776$	$\rightarrow 2$	$\rightarrow 0010$
$0,5099776 \times 16 = 8,1596416$	$\rightarrow 8$	$\rightarrow 1000$
$0,1596416 \times 16 = 2,5542656$	$\rightarrow 2$	$\rightarrow 0010$
$0,5542656 \times 16 = 8,8682496$	$\rightarrow 8$	$\rightarrow 1000$
$0,8682496 \times 16 = 13,8919936$	$\rightarrow D$	$\rightarrow 1101$

$$N = -0100,0111\ 0010\ 1000\ 0010\ 1000\ 1101 \cdot 2^{-46} =$$

$$= -1,00\ 0111\ 0010\ 1000\ 0010\ 1000\ 1101 \cdot 2^{-44}$$

por tanto: $m = 00011100101000001010001$ (sin redondeo)

- **Exponente:** $e = S + E = 127 - 44 = 83)_{10} = 01010011)_{2}$

1	01010011	00011100101000001010001
---	----------	-------------------------

3.5.3 Datos de tipo real

- Ejemplo:** Obtener el valor decimal del n° cuya representación interna es: 1 0011 1110 0011 110, suponiendo $n=16$ y $n_e = 8$

$$n = 1 + n_e + n_m \Rightarrow 16 = 1 + 8 + n_m \Rightarrow n_m = 7$$

Signo: $s = 1 \Rightarrow N$ es negativo

s	$n_e=8$	$n_m=7$
1	00111110	0011110

Exponente: $e = (00111110)_2 = 62_{10}$

Como $e = E + S$ y $S = 2^7 - 1 = 127 \Rightarrow E = e - S = 62 - 127 = -65$

Mantisa:

La mantisa está normalizada:

$$M = (1,0011110)_2 = 2^0 + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} = 1,234375_{10}$$

$$N = -1,234375 \cdot 2^{-65}$$

$$2^{-65} = 10^x \rightarrow -65 \log 2 = x \log 10 \rightarrow x = -19,5669$$

$$N = -1,234375 \cdot 2^{-65} = -1,234375 \cdot 10^{-19,5669} =$$

$$-1,234375 \cdot 10^{-19} \cdot 10^{0,5669} = -0,3345780 \cdot 10^{-19} = -3,345780142 \cdot 10^{-20}$$

$$\mathbf{N = - 3,345780142 \cdot 10^{-20}}$$

3.5.3 Datos de tipo real

- Patrones asociados a situaciones especiales:

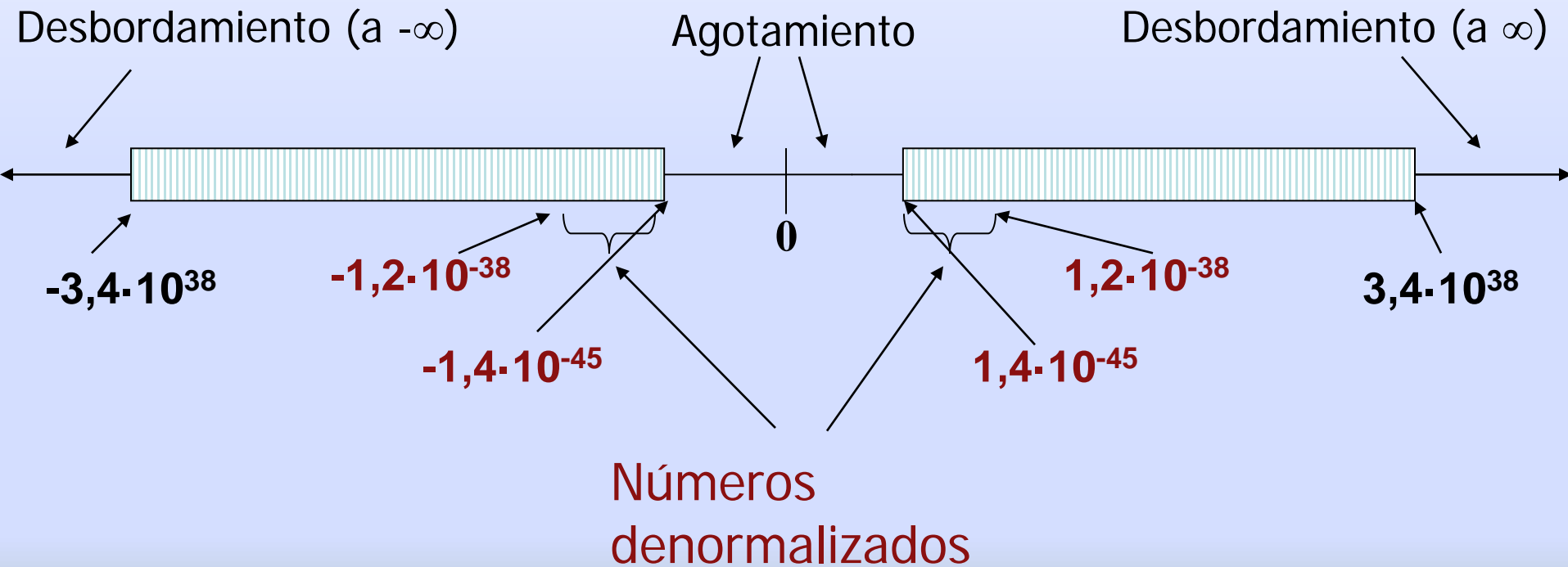
		Signo	Exponente	Mantisa
Nº denormalizado	→	0/1	0000 0000	m ≠ 0
Cero	→	0	0000 0000	000 0000 0000 0000 0000 0000
+ ∞	→	0	1111 1111	000 0000 0000 0000 0000 0000
- ∞	→	1	1111 1111	000 0000 0000 0000 0000 0000
Indeterminado (NaN)		0	1111 1111	m ≠ 0

Número denormalizado: tiene la parte entera igual a 0

$M = [0, m]$, con $M < 1$

3.5.3 Datos de tipo real

- IEEE 754 simple precisión:



3.5.3 Datos de tipo real

- Problemas por tener un número (n) limitado de bits:
 - Precisión limitada
 - La ALU debe realizar redondeos.
 - Resultados intermedios, pueden dar lugar a números excesivamente pequeños (que se aproximan a 0).
 - Resultados numéricos excesivamente altos, es decir por desbordamiento.
 - Comparación de dos números muy próximos, o iguales.

FIN

PREGUNTAS?

