TEMA 2.
CODIFICACIÓN DE LA INFORMACIÓN

1. **INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.**
   - **Sistema binario.**
   - **Sistema octal.**
   - **Sistema hexadecimal.**
2. **REPRESENTACIÓN DE TEXTOS.**
3. **REPRESENTACIÓN DE DATOS NUMÉRICOS.**
   - **Números naturales.**
   - **Números enteros.**
   - **Números reales.**
4. **REPRESENTACIÓN DE SONIDOS.**
5. **REPRESENTACIÓN DE IMÁGENES.**

1. INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.

- **COMPUTADOR**: Sistema digital.
  - Trabaja con dos niveles de información.
  - La unidad mínima que puede manejar es el **BIT**: '0' o '1'
- Un SISTEMA DE NUMERACIÓN de **base n** utiliza un conjunto de **n símbolos** para representar los números.
- Un **número** se expresará como un conjunto de cifras. Cada una contribuye con un valor que depende de:
  - El **valor** que representa la cifra.
  - La **posición** que ocupa.
- El **sistema decimal** consta de los símbolos $\mathbf{S_{10} = \{0,1,2,3,4,5,6,7,8,9\}}$

$$234 = 2\cdot10^2 + 3\cdot10^1 + 4\cdot10^0 = 2\cdot100 + 3\cdot10 + 4$$

$$234.21 = 2\cdot10^2 + 3\cdot10^1 + 4\cdot10^0 + 2\cdot10^{-1} + 1\cdot10^{-2} = 2\cdot100 + 3\cdot10 + 4 + 2\cdot0.1 + 1\cdot0.01$$

1. INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.

- **SISTEMA BINARIO (base 2):**
  - El conjunto de símbolos es $S_2 = \{0, 1\}$
  - Un número se representará como una secuencia de ceros y unos.
  - Transformación de binario a decimal:

    $01101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 = 13_{10}$
  - Transformación de decimal a binario: Método de las divisiones sucesivas.

    13 | 2
    (1) 6 | 2
    (0) 3 | 2
    (1) (1)

    $13_{10} = 1101_2$
  - Con n bits, se pueden representar 2^n números.
    - Desde el 0 hasta el 2^n-1

1. INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.

- **SISTEMA OCTAL (base 8):**
  - El conjunto de símbolos es $S_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$
  - Es un sistema auxiliar. Se usa porque es muy sencillo transformar de binario a octal y viceversa.

Oct.	Bin.
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

  - Transformación de binario a octal. Se forman grupos de tres bits desde el menos hasta el más significativo y se convierte a octal cada grupo individual:

    $1010101111_2 = 1257_8$
  - Transformación de octal a binario. Se convierte a binario cada cifra octal.

    $327_8 = 11010111_2$

1. INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.

- **SISTEMA OCTAL:**
  - Transformación de octal a decimal:

    $327_8 = 3 \cdot 8^2 + 2 \cdot 8^1 + 7 \cdot 8^0 = 215_{10}$

  - Transformación de decimal a octal: Divisiones sucesivas por 8.

    327 | 8
    (7) 40 | 8
    (0) (5)

    $327_{10} = 507_8$

1. INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.

- **SISTEMA HEXADECIMAL (base 16):**
  - Conjunto de símbolos: $S_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
  - Es un sistema auxiliar. Se usa porque es muy sencillo transformar de hexadecimal a binario y viceversa.

Oct.	Bin.
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Oct.	Bin.
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

- Transformación de binario a hexadecimal. Se forman grupos de cuatro bits desde el menos hasta el más significativo y se convierte a hexadecimal cada grupo individual:

  $10101010001111_2 = 2A8F_{16}$

- Transformación de hexadecimal a binario. Se convierte a binario cada cifra hexadecimal.

  $1C37_{16} = 1110000110111_2$

1. INTRODUCCIÓN. SISTEMAS DE NUMERACIÓN EN INFORMÁTICA.

- **SISTEMA HEXADECIMAL:**
  - Transformación de hexadecimal a decimal:

    $5A8_{16} = 5 \cdot 16^2 + 10 \cdot 16^1 + 8 \cdot 16^0 = 1448_{10}$

  - Transformación de decimal a octal: Divisiones sucesivas por 16.

1500 | 16
(12) 93 | 16
(13) (5)

$1500_{10} = 5DC_{16}$

FUNDAMENTOS DE INFORMÁTICA
1º INGENIERÍA INDUSTRIAL

2. REPRESENTACIÓN DE TEXTOS.

- Cualquier información escrita se representa por medio de caracteres:
  - Caracteres alfabéticos
    (A, B, C, ..., Y, Z, a, b, c, ... , y, z).
  - Caracteres numéricos
    (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
  - Caracteres especiales (Símbolos ortográficos y matemáticos)
    (+ - < > % & ( ) ? ¿ , . : (espacio) / ...)
  - Caracteres geométricos e iconos
    (●❍■❐⬧◆⌂⌧⌛⌨🖰🗏🗀🗁🗎🗄🕐👓☝💧👎☺😐☹💻✉...)
  - Caracteres de control (Salto de línea, comienzo de linea...)

- Cada uno de estos caracteres debe tener asignado un código en formato binario.
  - Con 2 bits (n = 2), nº combinaciones = $2^2 = 4$. Se pueden codificar 4 caracteres distintos.
  - Con 3 bits, 8 caracteres.
  - Con n bits, 2^n caracteres.

- Para codificar m símbolos, hay que buscar el menor número de bits n que cumpla
  $m < 2^n$ o bien $n >= \log_2 m$

2. REPRESENTACIÓN DE TEXTOS.

- **EBCDIC**
  - Desarrollado por IBM.
  - 8 bits, 256 caracteres.
- **ASCII**
  - Desarrollado en ppio para transmitir datos por líneas telegráficas.
  - 7 bits, 128 caracteres.
  - Los 32 primeros son caracteres de control de transmisión de datos
- **ASCII Extendido**
  - 8 bits, 256 caracteres.
  - 128 primeros caracteres normalizados (Código ASCII).
  - 128 restantes no normalizados (Propios de cada idioma).
- **UNICODE**
  - Desarrollado para procesamiento de texto de diferentes sistemas de escritura.
  - Persigue cubrir la mayoría de lenguajes escritos actuales.
  - 16 bits, 65356 símbolos.

FUNDAMENTOS DE INFORMÁTICA
1º INGENIERÍA INDUSTRIAL

2. REPRESENTACIÓN DE TEXTOS.

TABLA DE CÓDIGOS ASCII

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

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS ENTEROS (POSITIVOS Y NEGATIVOS)

- **BCD.**
  - Se representa cada número mediante 4 bits (16 combinaciones posibles).
  - Representación poco eficiente (de las 16 combinaciones posibles sólo se usan 10)
  - Operaciones matemáticas muy complicadas.

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

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS ENTEROS (POSITIVOS Y NEGATIVOS)

- **MÉTODO MAGNITUD-SIGNO:**
  - El primer bit representa el signo del número y el resto su módulo.
    - (-) ➔ 1 (+) ➔ 0

      -10 = 1 1010

      +10 = 0 1010
  - Fácil conversión de binario a decimal.
  - Las operaciones se complican. Para realizar una suma, es necesario comprobar previamente el signo de los operandos involucrados y su valor absoluto para conocer el signo del resultado.

$$A+B\begin{cases} A>0,B>0 & (+)A+B \\ A<0,B<0 & (-)A+B \\ A>0,B<0 & |A|>|B|\Rightarrow(+)A-B,,|A|<|B|\Rightarrow(-)B-A \\ A<0,B>0 & |A|>|B|\Rightarrow(-)A-B,,|A|<|B|\Rightarrow(+)B-A \end{cases}$$

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS ENTEROS (POSITIVOS Y NEGATIVOS)

- **COMPLEMENTO A UNO:**
  - Se utilizan palabras de n bits.
  - Como en el método anterior, el bit que se encuentra más a la izquierda indica el signo 1(-), 0(+).
  - Si el número que se quiere representar es positivo, el primer bit (0) representa el signo y el resto (n-1 bits) el módulo.
  - Si se trata de un número negativo, se intercambian ceros por unos y unos por ceros.
  - Ej. Para n = 8 bits:

    +10 = 00001010

    -10 = 11110101

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

- COMPLEMENTO A UNO: Para n = 4 bits:

C1	Base 10
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-7
1001	-6
1010	-5
1011	-4
1100	-3
1101	-2
1110	-1
1111	-0

- El 0 tiene doble representación.
- El rango de representación es:

  $[-(2^{n-1} - 1), 2^{n-1} -1]$

- Las operaciones son más sencillas. Siempre se suma, independientemente de si se trata una suma o una resta.
- En complemento a 1, los números se suman igual que en binario, teniendo en cuenta que si aparece acarreo en la suma de los bits más significativos, se debe sumar este acarreo al resultado.
- Debemos tener en cuenta que el resultado debe estar dentro del rango de representación para n bits.

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS ENTEROS (POSITIVOS Y NEGATIVOS)

- **COMPLEMENTO A DOS:**
  - Se utilizan palabras de n bits.
  - Como en el método anterior, el bit que se encuentra más a la izquierda indica el signo 1(-), 0(+).
  - Si el número que se quiere representar es positivo, el primer bit (0) representa el signo y el resto (n-1 bits) el módulo.
  - Si se trata de un número negativo, se complementa el número positivo y se le suma 1 al resultado, despreciando el último acarreo en caso de existir.
  - Ej. Para n = 4 bits:

    +5 = 0101

    -5 ➔ 10000 ➔ 1011

    - 0101

    01011

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

- COMPLEMENTO A DOS: Para n = 4 bits:

C2	Base 10
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

- El 0 ya no tiene doble representación.
- El rango de representación es:

  $[-2^{n-1}, 2^{n-1} -1]$

- Las operaciones son más sencillas. Siempre se suma, independientemente de si se trata una suma o una resta.
- En complemento a 2, los números se suman igual que en binario, despreciando el último acarreo en caso de existir.
- Debemos tener en cuenta que el resultado debe estar dentro del rango de representación para n bits.

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS REALES

- **COMA FIJA:**
  - La coma ocupa una posición fija y predeterminada.
  - Se opera por separado la parte entera y la decimal.
  - No todo número real con un número finito de decimales es posible representarlo en binario. (Ej. 0.3).
  - Los números negativos se representan igual que los enteros.
  - Paso de binario a decimal:
    - $0110.11 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 6.75$
  - Paso de decimal a binario:
    - 6.75 ➔ 6 | 2 → 0, 3 | 2 → 1, 1 ; 0.75 x 2 → 1.50 ; 0.50 x 2 → 1.00 ➔ 110.11

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS REALES

- COMA FLOTANTE:
  - Se utiliza notación exponencial, representando los números como: $N = M \cdot B^E$. (M = Mantisa, B = Base, E = Exponente).

    $13.745 = 0.1374 \cdot 10^{+2}$ $\quad -0.0000312 = -0.312 \cdot 10^{-4}$

    $0110.11 = 0.11011 \cdot 2^3$ $\quad 0.00101011 = 0.101011 \cdot 2^{-2}$

  - Norma IEEE para la representación de datos de tipo real:
    - Se utilizan 3 campos para representar un dato:

S	E'	M'

    - Exponente (n bits): $E' = E + (2^{n-1} - 1)$
      - De este modo, con n bits reservados para el exponente se pueden incluir exponentes negativos o positivos sin utilizar un bit de signo explícito).
    - Mantisa (m bits): Almacena la parte fraccionaria del número normalizado sin incluir el 1 de después de la coma: $M = 0.1M'$
  - Simple precisión (float) ➔ 32 bits ➔ n = 8, m = 23 bits
  - Doble precisión (double) ➔ 64 bits ➔ n = 11, m = 52 bits

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS REALES

- COMA FLOTANTE:
  - Ejemplo. Representación en simple precisión.

  $-133.5 = -10000101.1 \cdot 2^0 = -0.100001011 \cdot 2^8$

  M' = 00001011

  $E' = 8 + (2^8-1 - 1) = 135 = 10000111$

1	10000111	00001011000000000000000

- Exponentes en simple precisión (float).

E	E'	E' (binario)	
0	0+127	0111 1111	
127	127+127=254	1111 1110	→ Exp. Máximo
-126	127-126 = 1	0000 0001	→ Exp. Mínimo

3. REPRESENTACIÓN DE DATOS NUMÉRICOS

REPRESENTACIÓN DE NÚMEROS REALES

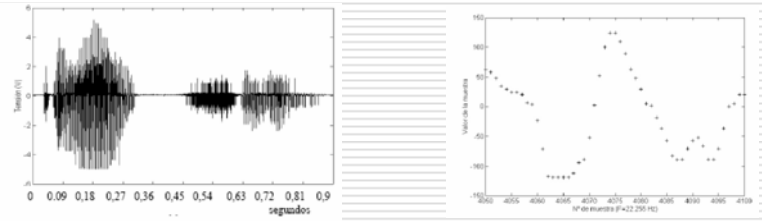
- COMA FLOTANTE:
  - Casos especiales:

a)	0	0000 0000	000 0000 0000 0000 0000 0000	Cero
b)	0	0000 0000	!= 0	Número denormalizado
c)	0	1111 1111	000 0000 0000 0000 0000 0000	+ Infinito
d)	1	1111 1111	000 0000 0000 0000 0000 0000	- Infinito
e)	0	1111 1111	!= 0	Indeterminación

Cuando E' = 0, el número se presenta denormalizado, es decir, el 1 más significativo de la mantisa no se encuentra implícito. En este caso, $E' = E + (2^{n-1} - 2)$

4. REPRESENTACIÓN DE SONIDOS

- Por medio de un micrófono, se capta una señal analógica.
- La señal es amplificada para encajarla dentro de dos valores límite, p. ej., entre -5 y +5 voltios.
- Se toman muestras con una frecuencia determinada.
- Las muestras se digitalizan (se transforman a binario) con un conversor analógico/digital. Con esto, la señal queda almacenada como una secuencia de valores, por ejemplo, de 8 bits.
- Cuanto mayor es la frecuencia de muestreo y el número de bits por muestra, mayor es la calidad del sonido y el volumen del archivo.
- Para reducir el tamaño del fichero generado, se utilizan los CODEC.



4. REPRESENTACIÓN DE SONIDOS

	Nº de bits/muestra	Frecuencia de muestro (F_s, KHz)	Periodo de muestreo (T_s, µsegundos)
PCM teléfono	8	8	125
Calidad telefónica	8	11,025	90,7
Radio	8	22,05	45,4
CD	16(1)	44,1	22,7

(1) Numero de bits/muestra por canal, con sonido estereofónico hay que multiplicar por 2

- Tipos de CODEC:
  - PCM (Pulse Code Modulation). Se graba un tren de pulsos correspondientes a cada muestra.
  - DPCM. (Differential PCM). En vez de almacenar los valores absolutos, se muestra la diferencia de cada muestra con la anterior.
  - ADPCM (Adaptive DPCM), u-law. Dada una secuencia de muestras, un algoritmo predice el valor de la muestra siguiente y se almacena el error entre el valor predicho y el real.
  - MPEG Audio Capa-III (Formatos MP2, MP3). Varían el número de bits y la frecuencia de muestreo en función del rango de frecuencias de la señal de audio (guardan mas muestras para las frecuencias a las que el oído es mas sensible).

5. REPRESENTACIÓN DE IMÁGENES

- Las imágenes se captan mediante periféricos especializados tales como escáneres, cámaras de video o cámaras fotográficas.
- Existen dos formas básicas de codificar las imágenes:
  - Mapa de bits.
  - Mapa de vectores.
- **MAPA DE BITS**
  - La imagen se divide en una retícula de puntos (pixeles) y a cada uno se le asigna el nivel de gris o el color medio correspondiente.
  - En el caso de imágenes de color, se suelen descomponer en 3 colores básicos: R (rojo), G (verde), B (azul), y la intensidad de cada color se codifica por separado.

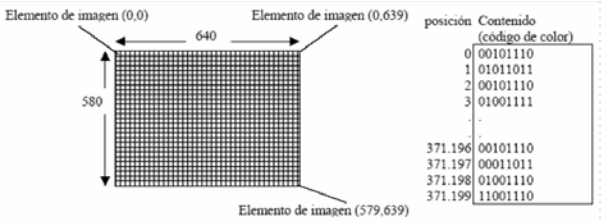


Figura 24.- Estructura de imagen con resolución 640x580 píxeles

5. REPRESENTACIÓN DE IMÁGENES

		Resolución (horizontal x vertical)	*Movimiento*
Convencionales	Fax (A4)	(100, 200,400) x (200, 300, 400) ei/"	Estática
	Foto (8"x11")	128, 400, 1200 ei/pulgada	Estática
Televisión	Videoconferencia	176 x 144 ei/imagen	10 a 36 imágenes/s
	TV	720 x 480 ei/imagen	30 imágenes/s
	HDTV (TV alta definición)	1920 x 1080 ei/imagen	30 imágenes/s
Pantalla computador	VGA SVGA XGA	640 x 480 ei 800 x 600 ei 1024 x 768 ei	

- **MAPA DE VECTORES**
  - Se descompone la imagen en una colección de objetos (líneas, polígonos, textos, …), cada cual, con sus atributos o detalles (color, grosor…) modelables mediante vectores y ecuaciones matemáticas que determinan su forma y posición en la imagen.
  - Cuando se visualiza una imagen a través de un periférico, un programa se encarga de evaluar las ecuaciones y generar la imagen correcta.
    - Adecuado para imágenes de tipo geométrico, no para imágenes reales. (Aplicaciones CAD).
    - Ocupan menos espacio que los mapas de bits.
    - Fácil redimensionamiento.
    - Menor calidad y fidelidad de la imagen.

5. REPRESENTACIÓN DE IMÁGENES

Tipo	Formato	Origen	Descripción
Mapa de bits	BMP (BitMap)	Microsoft	Usado en aplicaciones Windows
	PICT (PICTure)	Apple Comp.	Usado en Macintosh
	TIFF (Tagged Image File Formats)	Microsoft y Aldus	Usado en PC y Macintosh, muy poco compatible con otros formatos.
	JPEG (Joint Photographic Experts Group)	Grupo JPEG	Muy buena calidad para imágenes naturales. Incluye compresión, Muy usado en la web
	GIF (Graphic Interchange Format)	CompuServe	Incluye compresión. Muy usado en la web.
	PNG (Portable Network Graphics)	Consorcio www	Evolución de GIF. Muy buena calidad de colores. Incluye muy buena compresión
Mapa de vectores	DXF (Document eXchange Format)		Formato normalizado para imágenes CAD (AutoCAD , CorelDRAW, etc.)
	IGES (Initial Ghaphics Exchange Specification)	ASME/ANSI	Formato normalizado para modelos CAD (usable en AutoCAD , CorelDRAW, etc.)
	EPS (Encapsulated Poscript)	Adobe Sys.	Ampliación para imágenes del lenguaje Poscript de impresión.
	TrueType	Apple comp....	Alternativa de Apple y Microsoft para el EPS