



# **Representación de la Información**

---

**... en los Computadores**



# Información e Informática

---

- Un computador es una máquina que procesa información.
- La ejecución de un programa implica el tratamiento de los datos.
- Para que el computador ejecute un programa es necesario darles dos tipos de información:
  - las instrucciones que forman el programa y
  - los datos con los que debe operar ese programa.
- Los aspectos más importantes de la Informática relacionados con la información son:
  - cómo <representarla> y
  - cómo <materializarla> o <registrarla> físicamente.



## Cómo se da la información a un computador?

---

- Se la da en la forma usual escrita que utilizan los seres humanos;
  - con ayuda de un alfabeto o conjunto de símbolos, denominados caracteres.
- **Categorías de los caracteres:**
  - **Caracteres alfabéticos:** son los mayúsculas y minúsculas del abecedario inglés:  
A, B, C, D, E,..., X, Y, Z, a, b, c, d,..., x, y, z
  - **Caracteres numéricos:** están constituidos por las diez cifras decimales:  
Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9
    - El cero suele marcarse con una raya inclinada (ø) para evitar posibles confusiones con la O mayúscula.



## Cont...

---

- **Caracteres especiales:** son los símbolos no incluidos en los grupos anteriores, entre otros los siguientes:  
    ) ( , \* / ; : Ñ ñ = ! ? . ≈ ´ & > # < { Ç } SP
  - Con SP representamos el carácter o espacio en blanco, tal como el que separa dos palabras.
- **Carácter de control:** representan órdenes de control, como el carácter indicador de fin de línea o el carácter indicador de sincronización de una transmisión de que se emita un pitido en un terminal, etc.
  - Muchos de estos son generados e insertados por el propio computador.
- **Caracteres Gráficos:** son símbolos o módulos con los que se pueden representar figuras (o iconos) elementales.



## Cont...

---

- Toda comunicación con un computador convencional se realiza según los caracteres que admitan sus dispositivos de E / S. 101  
101
- Toda instrucción o dato se representará por un conjunto de caracteres tomados del alfabeto definido en el sistema a utilizar. 0110  
010
- El diseño de un sistema informático resulta mas fácil, su realización menos compleja y su funcionamiento muy fiable, si se utilizan solo dos valores o estados posibles. 01  
0
- Estos valores conceptualmente se representan por 0101  
0  
cero (0) y    apagada y    0 voltios y    0110  
uno (1) encendida    3.5 voltios    0110  
etc. (**BIT**)    01



# Codificación de la Información

---

- **Codificación** es una transformación que representa los elementos de un conjunto mediante los de otro, de forma tal que a cada elemento del primer conjunto le corresponda un elemento distinto del segundo.
- **Ejemplo:**
  - código de provincia en las matrículas de los coches;
  - código de enfermedades definido por la Organización Mundial de la Salud (OMS)
  - número de cedula de identidad
- Los códigos se permiten comprimir y estructurar la información
- En el interior de los computadores la información se almacena y se transfiere de un sitio a otro según un código que utiliza sólo dos valores (un *código binario*) representados por 0 y 1.



## Cont...

---

- **Codificación y Decodificación**

- Al tener que <traducir> toda la información suministrada al computador a ceros y unos, es necesario establecer una correspondencia entre el conjunto de todos los caracteres

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

- y el conjunto binario

$\beta = \{ 0, 1 \}^n$

- Estos códigos de transformación se denominan **códigos de Entrada / Salida (E/S)** o **códigos externos**.

- Las operaciones aritméticas con datos numéricos se suelen realizar en una representación más adecuada para este objetivo que la obtenida con el **código de E/S**.



# La unidad más elemental de información...

---

- ...es un valor binario, conocido como **BIT**.
- El origen de este término es inglés:

**BIT** = **B**i nary y digi**T**

- Un **bit** es una posición o variable que toma el valor 0 o 1.
- Es la capacidad mínima de almacenamiento de información en el interior de un computador
- El bit es la unidad de información mínima





# Información – caracteres – BIT ...

---

- A cada carácter le corresponde cierto número de bits.

**Byte** : número de bits necesarios para almacenar un caracter

- Byte se utiliza como sinónimo de 8 bits u octeto.
- La capacidad de almacenamiento (computador, soporte de información) se mide en bytes.
- Byte es una unidad relativamente pequeña
- Se utiliza múltiplos:
  - 1 **Kilobyte** = **1KB** =  $2^{10}$  bytes = 1024 bytes =  $2^{10}$  bytes
  - 1 **Megabyte** = **1MB** =  $2^{10}$  Kb = 1048576 bytes =  $2^{20}$  bytes
  - 1 **Gigabyte** = **1GB** =  $2^{10}$  Mb = 1073741824 bytes =  $2^{30}$  bytes
  - 1 **Terabyte** = **1TB** =  $2^{10}$  Gb = 1099511627776 bytes =  $2^{40}$  bytes
  - 1 **Pentabyte** = **1PB** =  $2^{10}$  Tb = 11258999906842624 bytes =  $2^{50}$  bytes.
  - 1 **Exabyte** = **1EB** =  $2^{10}$  Pb = 11258999906842624 bytes =  $2^{60}$  bytes.
  - 1 Zetabyte
  - 1 Yottabyte



## Cont...

---

**DATO:** Característica de una información expresada en forma adecuada para su tratamiento.

- Representación de los datos (valores):
  - Valores analógicos.
  - Valores discretos o digitales.
- Necesidad de convertir los valores analógicos a discretos.
  - **Sistema digital:** Sistema de N estados estables
  - **Dígito:** Variable capaz de asumir un estado.
- Los dígitos se agrupan para representar más estados.



## Cont...

---

- **Código:** Ley de correspondencia entre valores de información y combinaciones de dígitos de un sistema digital utilizadas para representarlos.
- **Codificación:** Información -> Código

azul	---->	0	azul	---->	100
verde	---->	1 ó	verde	---->	101
rojo	---->	2	rojo	---->	111
- **Decodificación:** Código -> Información

azul	<----	0	azul	<----	100	
verde	<----	1	ó	verde	<----	101
rojo	<----	2	rojo	<----	111	
- **Código binario:** Cuando el sistema digital utilizado tiene sólo 2 estados (0,1).



# Sistemas de numeración usuales en informática

---

- Los computadores suelen efectuar las operaciones aritméticas utilizando una representación para los datos numéricos basada en el sistema de numeración base dos (sistema **binario**).
- También se utilizan los sistemas de numeración, preferentemente el **octal** y **hexadecimal**, para obtener códigos intermedios.
- Un número expresado en uno de estos dos códigos puede transformarse directa y fácilmente a binario y viceversa.
  - Por lo que a veces se utilizan como paso intermedio en las transformaciones de decimal a binario y viceversa.



## Representación posicional de los números

---

- Un sistema de numeración en base **b** utiliza para representar los números un alfabeto compuesto por **b** símbolos o cifras.
- Todo número se expresa por un conjunto de cifras, contribuyendo cada una de ellas con un valor que depende de:

**a) la cifra en sí, y**

**b) la posición que ocupe dentro del número.**

- En el sistema de numeración decimal (sistema en base 10):
  - $b = 10$  y el alfabeto está constituido por diez símbolos o cifras decimales:

**{0,1,2,3,4,5,6,7,8,9}**



## Cont...

---

- por ejemplo, el número 3278.52 puede obtenerse como suma de:

$$\begin{array}{r} 3000 \\ 200 \\ 70 \\ 8 \\ 0.5 \\ 0.02 \\ \hline 3278.52 \end{array}$$

- se verifica que:

$$\mathbf{3278.52} = 3*10^3 + 2*10^2 + 7*10^1 + 8*10^0 + 5*10^{-1} + 2*10^{-2}$$



## Cont...

---

### Representación de un número en una base b:

Forma abreviada:

$$N = \dots n_4 n_3 n_2 n_1 n_0 . n_{-1} n_{-2} n_{-3} \dots$$

Valor:

$$N = \dots n_4 * b^4 + n_3 * b^3 + n_2 * b^2 + n_1 * b^1 + n_0 * b^0 + n_{-1} * b^{-1} \dots$$

- **Para representar un número:**

- Resulta más cómodo que los símbolos (cifras) del alfabeto o la base de numeración sean los menos posibles, pero ,
- Cuanto menos es la base, mayor es el número de cifras que se necesitan para representar una cantidad dada.



# Sistemas de Numeración

- Binario
- Octal
- Hexadecimal

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





# Sistema de numeración binario

---

- La base es 2 ( $b=2$ ) sólo
- se necesitan dos símbolos :

**{ 0, 1 }**

# Conversión de Decimal a Binario

- Se aplica el método de las “**divisiones y multiplicaciones**” sucesivas con la base como divisor y multiplicador ( $b = 2$ ).
- Ejemplo:**  $26.1875_{10} = 11010.0011_2$
- Para la parte entera:

$$\begin{array}{r}
 26 \overline{) 2} \\
 \underline{0} \phantom{0} \\
 13 \overline{) 2} \\
 \underline{1} \phantom{0} \\
 6 \overline{) 2} \\
 \underline{0} \phantom{0} \\
 3 \overline{) 2} \\
 \underline{1} \phantom{0} \\
 1
 \end{array}$$

A red arrow points from the final remainder '1' to the fractional part conversion section below.

- Para la parte fraccionaria:

$0.1875$	$0.3750$	$0.7500$	$0.5000$
$\times 2$	$\times 2$	$\times 2$	$\times 2$
$\hline 0.3750$	$\hline 0.7500$	$\hline 1.5000$	$\hline 1.0000$

A red arrow points from the bottom of the fractional part conversions to the right.



## Conversión de Binario a Decimal

---

- Se desarrolla la representación binaria (con  $b=2$ ) y se opera el polinomio en decimal.
- **Ejemplos:**

$$110100_{(2)} = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \\ = 52_{(10)}$$

$$10100.001_{(2)} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + \\ 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 20.125_{(10)}$$

- Realmente basta con sumar los pesos ( $2_i$ ) de las posiciones ( $i$ ) en las que hay un 1.



# Operaciones aritméticas con variables binarias

- Las operaciones aritméticas básicas son la suma, resta, multiplicación y división.

## Suma aritmética con varias variables

a	b	a + b
0	0	0
0	1	1
1	0	1
1	1	0 y me llevo 1

## Restar aritmética con varias variables



a	b	a - b
0	0	0
0	1	1 y me adeudo 1
1	0	1
1	1	0



## Multiplicación aritmética con varias variables

a	b	a * b
0	0	0
0	1	0
1	0	0
1	1	1

## División aritmética con varias variables

a	b	a / b
0	0	indeterminado
0	1	0
1	0	$\infty$
1	1	1



## Ejemplos:

Efectuar las siguientes operaciones aritméticas binarias:

---

$$\begin{array}{r} 1110101 \\ + 1110110 \\ \hline 11101011 \end{array}$$

$$\begin{array}{r} 1101010 \\ - 1010111 \\ \hline 0010011 \end{array}$$

$$\begin{array}{r} 1101010 \\ \times 11 \\ \hline 1101010 \\ + 1101010 \\ \hline 10011110 \end{array}$$

$$\begin{array}{r} 1010011 \\ \times 10 \\ \hline 0000000 \\ + 1010011 \\ \hline 10100110 \end{array}$$

$$\begin{array}{r} 1101.010 \\ - 101 \downarrow \downarrow \downarrow \downarrow \\ \hline 00110 \\ - 101 \downarrow \downarrow \\ \hline 00110 \\ - 101 \\ \hline 001 \end{array} \quad \begin{array}{r} 101 \\ \hline 10.101 \end{array}$$



# Sistema de numeración octal

---

- La base es 8
- El conjunto de símbolos es:  
**{ 0, 1, 2, 3, 4, 5, 6, 7 }**

## **Conversión de octal a decimal**

- Se desarrolla el polinomio con  $b=8$  y se opera en decimal.

## **Conversión de decimal a octal**

- Aplicar el método de "divisiones y productos" con divisor y multiplicador 8.

## **Conversión "rápida" de binario a octal**

- Agrupar cifras binarias de 3 en 3 y transformar con la tabla 1.

## **Conversión "rápida" de octal a binario**

- Convertir cada cifra octal mediante la tabla



## Cont...

---

- Ejemplo:

- Haciendo uso de la tabla convertir

$$10001101100.11010_{(2)} = N_{(8)}$$

$$10|001|101|100.110|10_{)2} = 2154.64_{)8}$$

- Ejemplo:

- Haciendo uso de la tabla convertir  $537.24_{)8} = N_{)2}$

$$537.24_{)8} = 101|011|111.010|100_{)2}$$



# Sistema de numeración hexadecimal

---

- La base es 16
- El conjunto de símbolos es:

**{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}**





## Cont...

---

### **Conversión de Hexadecimal a decimal**

- Se desarrolla el polinomio con  $b=16$  y se opera en decimal.

### **Conversión de Decimal a hexadecimal**

- Aplicar el método de “divisiones y productos” con divisor y multiplicador 16.

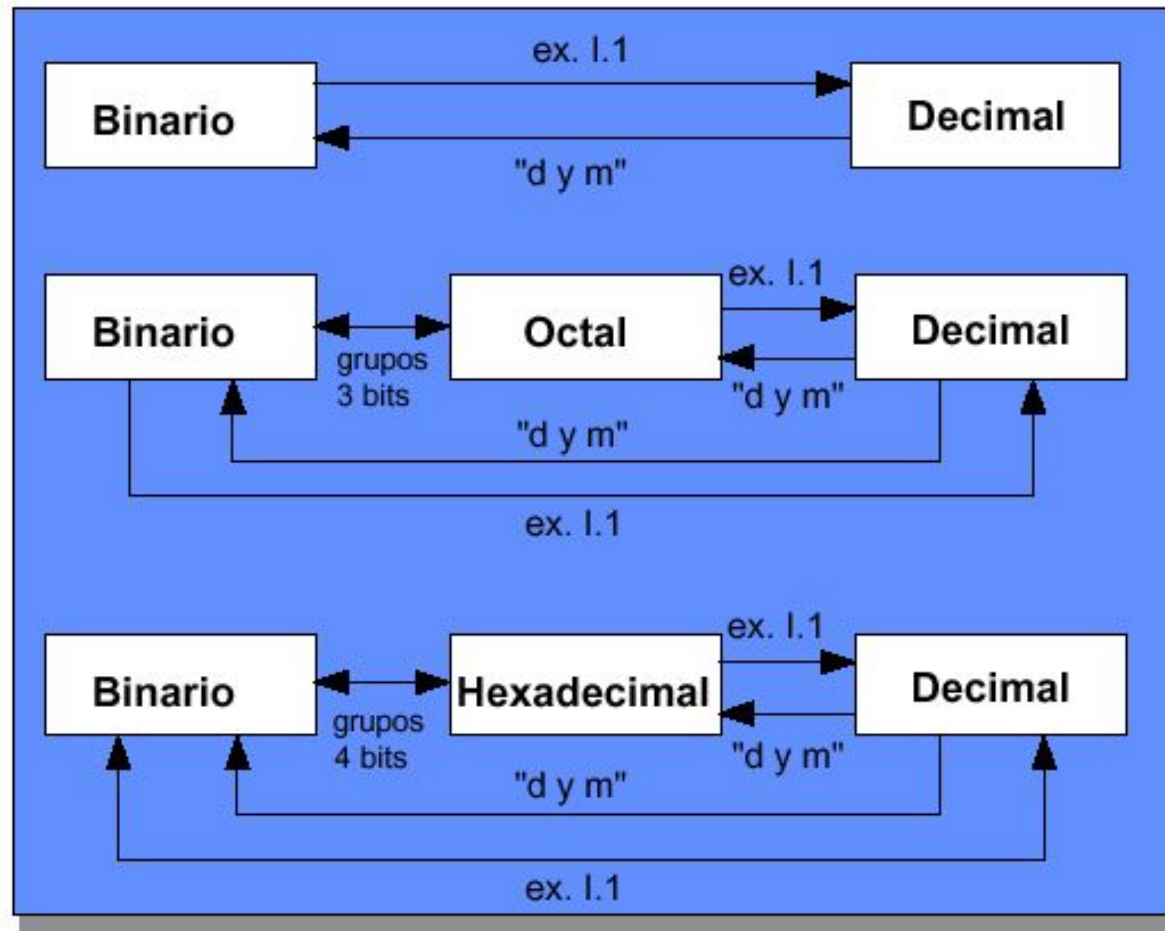
### **Conversión “rápida” de binario a hexadecimal**

- Agrupar cifras binarias de 4 en 4 y transformar con la tabla
  - Ejemplo:  $0010|0101|1101|1111 . 1011|1010_{(2)} = 25DF.BA_{(16)}$

### **Conversión “rápida” de hexadecimal a binario**

- Convertir cada cifra hexadecimal mediante la tabla
  - Ejemplo:  $1ABC.C4_{(16)} = 0001|1010|1011|1100 . 1100|0100_{(2)}$

# Resumen de cambios de base



- Hacer las operaciones en binario:

- $101011101_{(2)} + 101001010_{(2)} = N_{(8)}$
- $1100101011_{(2)} + 100101101_{(2)} = N_{(10)}$
- $101011101_{(2)} - 10001010_{(2)} = N_{(16)}$
- $110001011_{(2)} - 10101101_{(2)} = N_{(16)}$
- $10101.0101_{(2)} * 2_{(10)} = N_{(2)}$
- $1101.1010_{(2)} * 25_{(10)} = N_{(10)}$
- $1010100_{(2)} / 2_{(10)} = N_{(8)}$
- $10101.101_{(2)} / 101_{(2)} = N_{(2)}$



# Representación de datos Numéricos

---

- Para la representación de los datos numéricos se debe tener en cuenta que las operaciones de la ALU están sujetas a las siguientes restricciones:
  - Los registros son de tamaño fijo.
  - Puede existir desbordamiento.
  - Presentan problemas con los números negativos.
- Es necesario, por ello, introducir **nuevas formas de numeración** basadas, por supuesto, en la representación binaria.
- Al conjunto de estas representaciones y su funcionamiento se le denomina **aritmética binaria**.
- En aritmética binaria debemos distinguir:
  - Representación para números enteros
  - Representación de números reales.



# Representación de textos

---

## Códigos de Entrada/Salida

- Asocian a cada símbolo una determinada combinación de bits.

$a = \{0,1,2,\dots,8,9,A,B,\dots,Y,Z,a,b,\dots,y,z,*,",/,,\dots\}$

$b = \{0,1\}^n$

- Con  $n$  bits podemos codificar  $m=2^n$  símbolos distintos
- Para codificar  $m$  símbolos distintos se necesitan  $n$  bits,

$$n \geq \log_2 m = 3.32 \log (m)$$



## Ejemplo:

---

- Para codificar las cifras decimales  $\{0,1,2,3,4,5,6,7,8,9\}$  se necesitarán :  
 **$n \geq 3.3221 \log(m) = 3.322 \text{ bits}$**
- es decir, 4 bits (para que se cumpla la relación)
- Por lo menos se necesitan 4 bits, pero pueden hacerse codificaciones con más bits de los necesarios. **Tabla 2**
- Con 4 bits no se usan  $2^4 - 10 = 6$  combinaciones, y con 5 bits  $2^5 - 10 = 22$  combinaciones.



## Cont... Tabla 2

---

Alfabeto	Código I	Código II
0	0000	00000
1	1000	10001
2	0100	01001
3	1100	11000
4	0010	00101
5	1010	10100
6	0110	01100
7	1110	11101
8	0001	00011
9	1001	10010



# Ejemplos de Códigos de E/S

---

- **Código ASCII**

- El código ASCII se utiliza para representar caracteres.
- Formado por 8 bits (cada carácter se expresa por un número entre 0 y 255)
- Es un código estándar, independiente del lenguaje y del ordenador
- Podemos distinguir dos grupos:
  - Los 128 primeros caracteres se denominan código ASCII estándar
    - Representan los caracteres que aparecen en una maquina de escribir convencional
  - Los 128 restantes se denominan código ASCII ampliado
    - Este código asocia un numero a caracteres que no aparecen en la maquina de escribir y que son muy utilizados en el ordenador tales como caracteres gráficos u operadores matemáticos.

- **Código EBCDIC**

- Extended Binary Coded Decimal Interchange Code
  - Código Ampliado de Caracteres Decimales Codificados en Binario para Intercambio de Información
- Es un sistema de codificación de caracteres alfanuméricos.
- Cada carácter queda representado por un grupo de 8 bits.

- **Código Unicode**

- Es de 16 bits, por lo que puede representar 65536 caracteres.
- Es una extensión del ASCII para poder expresar distintos juegos de caracteres (latino, griego, árabe, kanji, cirílico, etc).





# Cont...

---

- **Código ASCII.**

## CARACTERES DE CONTROL

0	<b>NUL</b>	(nulo)	16	<b>DLE</b>	(escape de enlace de datos)
1	<b>SOH</b>	(comienzo de cabecera)	17	<b>DC1</b>	(control de dispositivo 1)
2	<b>STX</b>	(comienzo de texto)	18	<b>DC2</b>	(control de dispositivo 2)
3	<b>ETX</b>	(fin de texto)	19	<b>DC3</b>	(control de dispositivo 3)
4	<b>EOT</b>	(fin de transmisión)	20	<b>DC4</b>	(control de dispositivo 4)
5	<b>ENQ</b>	(pregunta)	21	<b>NAK</b>	(acuse de recibo negativo)
6	<b>ACK</b>	(acuse de recibo)	22	<b>SYN</b>	(sincronización)
7	<b>BEL</b>	(campana sonora)	23	<b>ETB</b>	(fin de bloque transmisión)
8	<b>BS</b>	(retroceso de un espacio)	24	<b>CAN</b>	(anulación)
9	<b>HT</b>	(tabulación horizontal)	25	<b>EM</b>	(fin de medio físico)
10	<b>LF</b>	(cambio de renglón)	26	<b>SUB</b>	(carácter de sustitución)
11	<b>VT</b>	(tabulación vertical)	27	<b>ESC</b>	(escape)
12	<b>FF</b>	(página siguiente)	28	<b>FS</b>	(separador de ficheros)
13	<b>CR</b>	(retroceso del carro)	29	<b>GS</b>	(separador de grupos)
14	<b>SO</b>	(fuera de código)	30	<b>RS</b>	(separador de registros)
15	<b>SI</b>	(en código)	31	<b>US</b>	(separador de unidades)



# Cont...

---

## CARACTERES GRAFICOS

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

# Cont...

## ASCII-1967 (US-ASCII)

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

ódigos ASCII (0-127).

Carácteres no imprimibles				Carácteres imprimibles								
Nombre	Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.
ulo	0	00	NUL	32	20	Espacio	64	40	@	96	60	`
inicio de cabecera	1	01	SOH	33	21	!	65	41	A	97	61	a
inicio de texto	2	02	STX	34	22	"	66	42	B	98	62	b
fin de texto	3	03	ETX	35	23	#	67	43	C	99	63	c
fin de transmisión	4	04	EOT	36	24	\$	68	44	D	100	64	d
inquiry	5	05	ENQ	37	25	%	69	45	E	101	65	e
acknowledge	6	06	ACK	38	26	&	70	46	F	102	66	f
campanilla (beep)	7	07	BEL	39	27	'	71	47	G	103	67	g
backspace	8	08	BS	40	28	(	72	48	H	104	68	h
tabulador horizontal	9	09	HT	41	29	)	73	49	I	105	69	i
alto de línea	10	0A	LF	42	2A	*	74	4A	J	106	6A	j
tabulador vertical	11	0B	VT	43	2B	+	75	4B	K	107	6B	k
alto de página	12	0C	FF	44	2C	,	76	4C	L	108	6C	l
retorno de carro	13	0D	CR	45	2D	-	77	4D	M	109	6D	m
shift fuera	14	0E	SO	46	2E	.	78	4E	N	110	6E	n
shift dentro	15	0F	SI	47	2F	/	79	4F	O	111	6F	o
escape línea de datos	16	10	DLE	48	30	0	80	50	P	112	70	p
control dispositivo 1	17	11	DC1	49	31	1	81	51	Q	113	71	q
control dispositivo 2	18	12	DC2	50	32	2	82	52	R	114	72	r
control dispositivo 3	19	13	DC3	51	33	3	83	53	S	115	73	s
control dispositivo 4	20	14	DC4	52	34	4	84	54	T	116	74	t
neg acknowledge	21	15	NAK	53	35	5	85	55	U	117	75	u
incronismo	22	16	SYN	54	36	6	86	56	V	118	76	v
fin bloque transmitido	23	17	ETB	55	37	7	87	57	W	119	77	w
cancelar	24	18	CAN	56	38	8	88	58	X	120	78	x
fin medio	25	19	EM	57	39	9	89	59	Y	121	79	y
substituto	26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
escape	27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
separador archivos	28	1C	FS	60	3C	<	92	5C	\	124	7C	
separador grupos	29	1D	GS	61	3D	=	93	5D	]	125	7D	}
separador registros	30	1E	RS	62	3E	>	94	5E	^	126	7E	~





# CÓDIGO EBCDIC

---

## *Caracteres de control EBCDIC:*

NUL	Nulo	IGS	Separador para intercambio de grupos
SOH	Comienzo de cabeza	IRS	Separador para intercambio de registros
SOT	Comienzo de texto	IUS	Separador para intercambio de unidad
EOT	Final de texto	DS	Selección de dígito
PF	Perforadora desconectada	SOS	Comienzo de significado
HT	Tabulación horizontal	FS	Separador de campo
LC	Minúscula	BYP	Desviar
DEL	Eliminar, borrar	LF	Alimentación de línea
RLF	Alimentación de línea invertida	ETB	Final de bloque de transmisión
SMM	Comienzo mensaje manual	ESC	Escape
VT	Tabulación vertical	SM	Fijar modo
FF	Alimentación de hoja	ENQ	Solicitud, petición
CR	Retorno de carro	ACK	Acuse de recibo
SO	Fuera de código	BEL	Pitido
SI	Dentro de código	SYN	Sincronización
DLE	Escape del enlace de datos	PN	Perforadora conectada
TM	Marca de cinta	RS	Detener lectora
RES	Restaurar	UC	Mayúsculas
NL	Pasar a línea siguiente	EOT	Fin de transmisión
BS	Retroceso de un espacio	NACK	Acuse de recibo negativo
IL	<i>sin función</i>	SUB	Sustituir
CAN	Cancelar	DCi	Control dispositivo <i>i</i>
EM	Final de soporte	CUi	Control usuario <i>i</i>
CC	Control del cursor		
SP	Espacio en blanco		
IFS	Separador para intercambio de archivos		

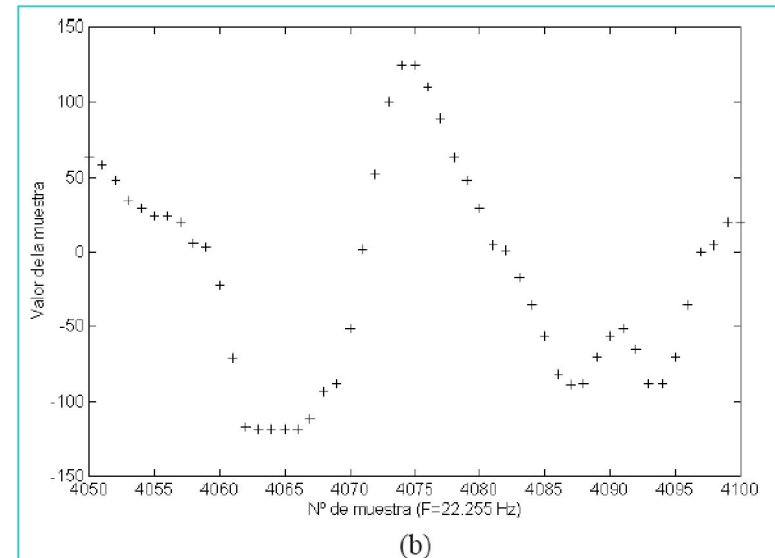
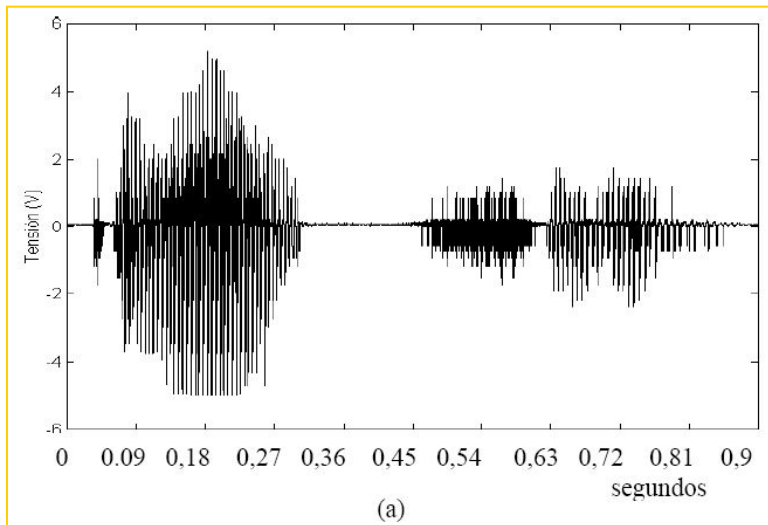


# Esquema de asignación de códigos en Unicode

---

<i>Zona</i>	<i>Códigos</i>		<i>Símbolos codificados</i>	<i>Nº de caracteres</i>
A	0000	0000 00FF	Latín-1	256
			otros alfabetos	7.936
	2000		Símbolos generales y caracteres fonéticos chinos, japoneses y coreanos	8.192
I	4000		Ideogramas	24.576
O	A000		Pendiente de asignación	16.384
R	E000 FFFF		Caracteres locales y propios de los usuarios. Compatibilidad con otros códigos	8.192

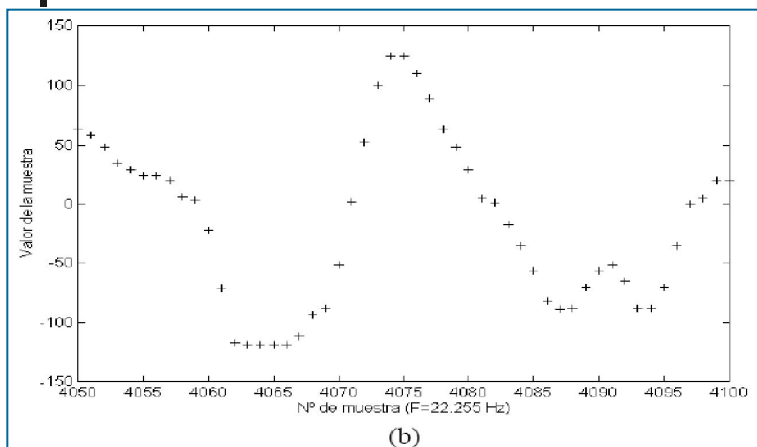
# Representación de Sonidos



- Grabación de una señal de sonido:
  - **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 amplificada** para encajarla dentro de dos valores límites, p.e. entre -5 voltios y +5 voltios.

- a) Señal analógica captada por un micrófono al pronunciar la palabra "casa";
- b) Tramo de muestras comprendido entre 0,184 a 0,186 segundos;

# Cont...



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
- c) Valores de las muestras obtenidos por un conversor A/D y que representan a la señal de voz.
- Principales parámetros de grabación:
  - Frecuencia de muestreo (suficiente para no perder la forma de la señal original)
  - Número de bits por muestra (precisión)
- La **capacidad** necesaria para almacenar una señal de audio depende de los dos parámetros anteriores:
  - 1 minuto de audio estéreo con calidad CD, necesita 10 MB (sin compresión de datos)





# Representación de Imágenes

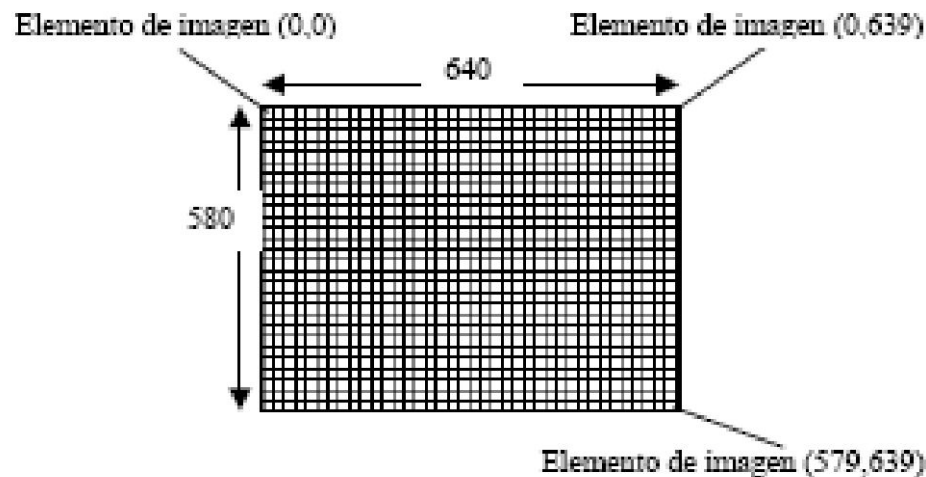
- Las imágenes se adquieren por medio de periféricos tales como escáneres, cámaras de video o cámaras fotográficas.
- Una imagen se representa por patrones de bits, generados por el periférico correspondiente.
- Formas básicas de representación:
  - Mapa de bits
  - Mapa de vectores

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)	Consortio 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 Graphics 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

# Imágenes de Mapas de Bits

Estructura de una imagen con resolución de 640x580 elementos.

- La imagen se considera dividida en una fina retícula de celdas o **elementos de imagen** (pixels).
- 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.
- La **resolución** es
  - (nº e.i. horizontales x nº e.i. verticales).
- Se memoriza, almacenando ordenada y sucesivamente los atributos de los distintos elementos de imagen.



(a)

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)

# Características de algunas formas de imágenes digitalizadas

		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	

- La calidad de la imagen depende de
  - La resolución y
  - Codificación del atributo (número de bits)
- La capacidad depende de dichos parámetros:
  - Ejemplo: imagen de 16 niveles de grises (b/n) y con resolución de 640x350: 110 Kbytes
  - Ejemplo: imagen con resolución XGA con 256 niveles para cada color básico: 2,25 MBytes



# Imágenes de Mapas de Vectores

---

- 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.
- Para visualizar una imagen, un programa evalúa las ecuaciones y escala los vectores generando la imagen concreta a ver.
- Características:
  - Sólo es adecuada para gráficos de tipo geométrico (no imágenes reales)
  - Ocupan mucho menos espacio que los mapas de bits.



# Compresión De Datos

---

- Diversas aplicaciones (multimedia, etc.) requieren utilizar archivos de gran capacidad.
  - Volumen requerido para su almacenamiento en disco muy elevado
  - el tiempo de transmisión del archivo por una red resulta excesivo
- Solución: transformación denominada **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.

## Técnicas:

- Codificación por longitud de secuencias
- Codificación relativa o incremental
- Codificación dependiente de la frecuencia
- Codificación con diccionario adaptativo
- Codificación Lempel-Ziv
- Compresión GIF (imágenes)
- Compresión JPEG (imágenes)
- Compresión MPEG (imágenes)
- Compresión MP3 (sonidos)



# Detección de errores en la Información Codificada

---

- Cuantas menos codificaciones se desperdicien el código es más eficiente.
- La **eficiencia de un código** (**T**) se define como el cociente entre el número de símbolos que se representan realmente,  $m$ , dividido para el número de símbolos que en total pueden representarse.
- Con códigos binarios en que  $m = 2^n$ , se tiene:
$$T = m / m' = m / 2^n, \quad \text{con } 0 \leq T \leq 1$$
- Cuanto más eficiente sea el código, entonces  $\tau$  será mayor.



### Ejemplo 3.17

---

- Supongamos que usamos el código ASCII, para representar 95 símbolos . La eficiencia del código será:

**sin bit de paridad:**

$$T = m / m' = 95 / 2^7 = 0.742$$

**con un bit adicional de paridad:**

$$T = m / m' = 95 / 2^8 = 0.371$$



## Cont...

---

- Un código poco eficiente se dice que es **redundante**:

$$R = ( 1 - T ) \cdot 100\%$$

- (Observamos que se da en %)
- **Ejemplo 3.18**
- En los casos considerados en el ejemplo anterior, las redundancias son:
$$R = ( 1 - 0.742 ) \cdot 100\% = 28.8\%$$
$$R = ( 1 - 0.371 ) \cdot 100\% = 62.9\%$$
- En ocasiones, las redundancias se introducen deliberadamente para detectar posibles errores de transmisión o grabación de información.





## Cont...

- **Por ejemplo:**
  - necesitamos transmitir 8 símbolos {A,B,C,D,E,F,G,H}
  - Un código sin redundancia:  
 $n = 3$  bits
- Si por error varía uno de los bits obtenemos otro símbolo del alfabeto.

Alfabeto	Código I	Código II
A	000	0000
B	001	0001
C	010	0010
D	011	0011
E	100	0100
F	101	0101
G	110	0110
H	111	0111

- Esto considerando por sí mismo ( aisladamente) no puede ser detectado como erróneo.
- Pero, si usamos un código redundante, como el código II existirían algunas posibilidades de detectar errores.



## Cont...

---

- Las redundancias se introducen de acuerdo con algún algoritmo predeterminado.
- Los códigos pueden ser verificados por circuitos del computador o periféricos especializados en este objetivo.
- Uno de estos algoritmos añade al código inicial de cada carácter un nuevo bit llamado bit de paridad.



# Bit de Paridad

---

- Existen dos criterios para introducir este bit:
- **Bit de Paridad, Criterio Par:**
  - Se añade un bit ( 0 o 1 ) de forma que el número total de unos del código que resulte sea par.
- **Bit de Paridad, Criterio Impar:**
  - Se añade un bit ( 0 o 1 ) de forma que el número total de unos del código que resulte sea impar.
- El bit de paridad se introduce antes de transmitir o grabar la información ( en la memoria principal, cinta o disco magnético).



## Ejemplo:

Código inicial	Código con bit de paridad	Código con bit de paridad
	(criterio par)	(criterio impar)
100 0001	0100 0001	1100 0001
101 1011	1101 1011	0101 1011
101 0000	0101 0000	1101 0000
110 1000	1110 1000	0110 1000
	↑ bit de paridad	↑ bit de paridad

- Por **ruido o interferencia** en la transmisión puede **intercambiarse un bit** (de 0 a 1 o de 1 a 0).
- Si en el receptor se comprueba la paridad se detecta el error ya que el número de unos deja de ser par o impar (según el criterio).
- De esta manera se podría producir automáticamente la **retransmisión** del carácter erróneo.
- Si se produjese el **cambio de dos bits distintos**, no se detectaría el error de paridad.
  - Esto es poco probable que ocurra.