

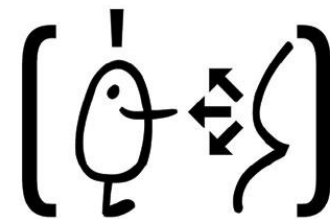
Juan Manuel Orduña Huertas

Fundamentos de computadores I



VNIVERSITAT
DE VALÈNCIA

Escola Tècnica Superior
d'Enginyeria **ETSE-UV** 



VICERECTORAT
DE PARTICIPACIÓ
I PROJECCIÓ
TERRITORIAL

Universitat i Societat

CULLERA



Profesor

- Juan Manuel Orduña Huertas.
- <https://www.uv.es/uvweb/universidad/es/ficha-persona-1285950309813.html?p2=orduna&idA=true>
- Departamento de Informática. ETSE
- Despacho 1.2.14. ETSE
- Juan.orduna@uv.es
- Tutorías: Jueves 10:00 a 11:30



Objetivo

- ¿Cómo es posible que los computadores hagan todo tipo de cálculos? ¿Son tan listos como nosotros?
- ¿Por qué funcionan sólo con ceros y unos? ¿Cómo pueden hacer todo lo que hacen si sólo tienen ceros y unos?
- ¿Cómo funciona un computador (y todos los aparatos «digitales» de hoy en día) ?



Metodología

- Presentaciones
- EJERCICIOS Y PROBLEMAS EN CLASE

BIBLIOGRAFÍA RECOMENDADA:

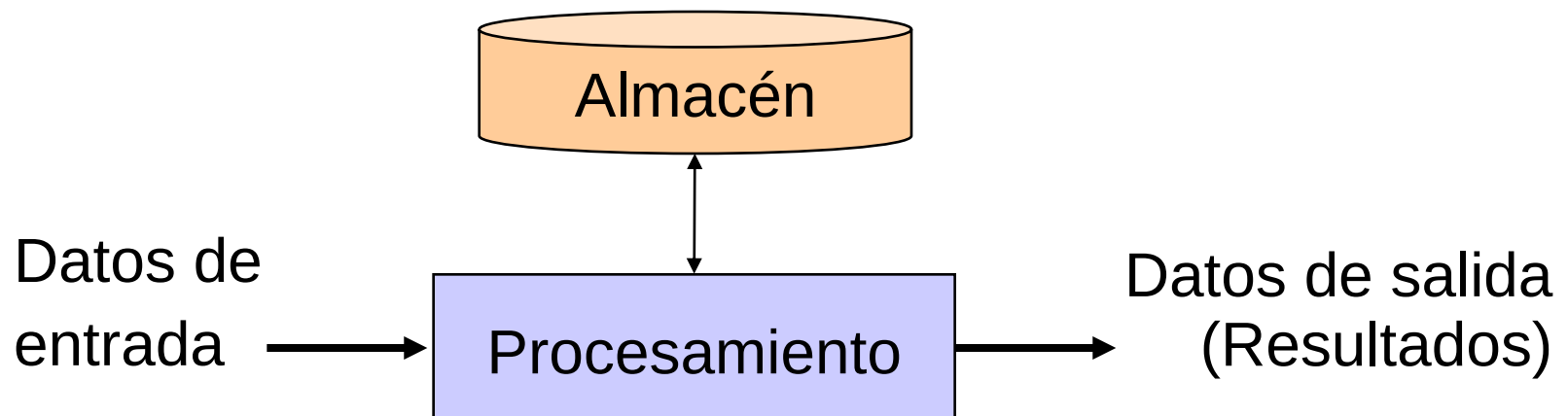
T.L. Floyd. «Fundamentos de Sistemas Digitales»
Sexta Edición. Ed. Prentice-Hall. ISBN 84-89660-21-2

«Educación es lo que queda después
de olvidar lo que se ha aprendido en la
escuela»

Albert Einstein

Tecnologías de la Información (IT)

- **Informática** = INFORmación + autoMÁTICA
 - disciplina encargada del tratamiento de la información mediante métodos automáticos.
- **Ordenador**: dispositivo electrónico digital programable capaz de almacenar, recuperar y procesar información.





Fundamentos de computadores

1. Sistemas de numeración
2. Circuitos combinacionales y secuenciales
3. Estructura de computadores



Sección 1. Sistemas de numeración

1. Sistemas numéricos posicionales
2. Conversión entre sistemas numéricos
3. Representación alfanumérica
4. Introducción a conceptos digitales

La memoria

- La memoria es el dispositivo que se encarga de **almacenar** toda la información.
- En la arquitectura de Von Neumann, **tanto los datos como las instrucciones** para procesarlos se almacenan internamente.
- El contenido de la memoria se almacenará codificado en forma de **bits (0 ó 1)**.

	
VALOR 1	VALOR 0



0 0



0 1



1 0



1 1

Almacenamiento digital

- **BIT (BInary digiT)**: unidad mínima de información que sólo puede contener 0 ó 1.
- Denominaremos **byte** a un conjunto de 8 bits.
- A partir de ahí:
 - ◆ 1 **Kilobyte** (1 Kb) = 2^{10} bytes = 1.024 bytes $\approx$ 1.000 bytes
 - ◆ 1 **Megabyte** (1 Mb) = 2^{20} bytes = 1.048.576 bytes $\approx$ 1₁000.000 bytes
 - ◆ 1 **Gigabyte** (1 Gb) = 2^{30} bytes = 1.073.741.824 bytes $\approx$ 1.000.000.000 bytes
 - ◆ 1 **Terabyte** (1 Tb) = 2^{40} bytes $\approx$ 1.000.000.000.000 bytes

Ejemplo de pr. en lenguaje máquina

CARACTERÍSTICAS DE LA MÁQUINA EJEMPLO

PALABRAS DE 2 BYTES

INSTRUCCIONES DE 4 BITS

DIRECCIONES DE 4 BITS

DATOS DE 16 BITS

4 REGISTROS

JUEGO DE INSTRUCCIONES

instrucción	operación
0001	SET
0010	MOVMR
0011	MOVVM
0100	ADD
0101	SUBTRACT
0110	MULTIPLY
0111	DIVIDE
1000	EQUAL
1001	GREATER
1010	LEAST
1100	JUMP
1101	IFJUMP
1111	STOP

PROGRAMA 1

pos_bin	pos.	MEMORIA	en lenguaje ensamblador
0001	01	0010 1011 0001 0000	MOVMR 11, 1
0010	02	0010 1011 0010 0000	MOVMR 11, 2
0011	03	0001 0011 0001 0000	SET 3, 1
0100	04	0101 0001 0011 0001	SUBTRACT 1, 3, 1
0101	05	1010 0001 0011 0100	LEAST 1, 3, 4
0110	06	1100 0100 1001 0000	IFJUMP 4, 09
0111	07	0110 0010 0001 0010	MULTIPLY 2, 1, 2
1000	08	1100 0100 0000 0000	JUMP 04
1001	09	0011 0010 1100 0000	MOVVM 2, 12
1010	10	1111 0000 0000 0000	STOP
1011	11	dddddddddddddddd	<i>dato (valor de entrada)</i>
1100	12	rrrrrrrrrrrrrrrrrr	<i>resultado</i>

num_bin	num	REGISTROS
0001	1	
0010	2	
0011	3	
0100	4	

CONTADOR DE PROGRAMA

1

Ejemplo de pr. en lenguaje máquina

CARACTERÍSTICAS DE LA MÁQUINA EJEMPLO

PALABRAS DE 2 BYTES

INSTRUCCIONES DE 4 BITS

DIRECCIONES DE 4 BITS

DATOS DE 16 BITS

4 REGISTROS

JUEGO DE INSTRUCCIONES

instrucción	operación
0001	SET
0010	MOVMR
0011	MOVRM
0100	ADD
0101	SUBSTRACT
0110	MULTIPLY
0111	DIVIDE
1000	EQUAL
1001	GREATER
1010	LEAST
1100	JUMP
1101	IFJUMP
1111	STOP

PROGRAMA 2

pos_bin	pos.	MEMORIA
0001	01	MOVMR 13, 1
0010	02	MOVMR 14, 2
0011	03	EQUAL 1, 2, 3
0100	04	IFJUMP 3, 11
0101	05	GREATER 1, 2, 3
0110	06	IFJUMP 3, 09
0111	07	SUBSTRACT 2, 1, 2
1000	08	JUMP 03
1001	09	SUBSTRACT 1, 2, 1
1010	10	JUMP 03
1011	11	MOVRM 1, 15
1100	12	STOP
1101	13	dato 1
1110	14	dato 2
1111	15	resultado

num_bin	num	REGISTROS
0001	1	
0010	2	
0011	3	
0100	4	

CONTADOR DE PROGRAMA

1

valor de entrada 1

valor de entrada 2



Sistemas numéricos posicionales

La notación posicional es un modo de escritura numérica en el cual cada símbolo posee un valor diferente que depende de su posición relativa. Queda definida por la base, que es el número de símbolos necesarios para escribir cualquier número. Así pues podemos expresar un número $N_{(b)}$ de n símbolos en base b como:

$$N_{(b)} = s_{n-1} \cdot b^{n-1} + s_{n-2} \cdot b^{n-2} + \dots + s_1 \cdot b^1 + s_0 \cdot b^0$$

Esta expresión se suele escribir de forma simplificada utilizando sólo los símbolos:

$$N_{(b)} = s_{n-1} s_{n-2} \dots s_1 s_0$$



Sistemas numéricos posicionales

Sistema decimal (base 10)

- Necesitamos 10 símbolos diferentes.
- Los símbolos son: 0,1,2,3,4,5,6,7,8,9
- El peso de cada símbolo según su posición es:
$$N_{(10)} = s_{n-1} \cdot 10^{n-1} + s_{n-2} \cdot 10^{n-2} + \dots + s_1 \cdot 10^1 + s_0 \cdot 10^0$$
- Por ejemplo, el número $325_{(10)}$ representa el valor:
$$325_{(10)} = 3 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0 = 325$$



Sistemas numéricos posicionales

Sistema binario (base 2)

- Necesitamos 2 símbolos diferentes.
- Los símbolos son: 0,1
- El peso de cada símbolo según su posición es:
$$N_{(2)} = s_{n-1} \cdot 2^{n-1} + s_{n-2} \cdot 2^{n-2} + \dots + s_1 \cdot 2^1 + s_0 \cdot 2^0$$
- Por ejemplo, el número $101_{(2)}$ representa el valor:
$$101_{(2)} = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$$



Sistemas numéricos posicionales

Sistema binario (base 2)
EJERCICIOS

- Calcular el valor en base 10 de los siguientes números binarios:

110 :

0110:

1000:

1010:

1100:

10001:

10101:



Sistemas numéricos posicionales

Sistema hexadecimal (base 16)

- Necesitamos 16 símbolos diferentes.

- Los símbolos son:

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

- El peso de cada símbolo según su posición es:

$$N_{(16)} = s_{n-1} \cdot 16^{n-1} + s_{n-2} \cdot 16^{n-2} + \dots + s_1 \cdot 16^1 + s_0 \cdot 16^0$$

- Por ejemplo, el número $1A0_{(16)}$ representa el valor:

$$1A0_{(16)} = 1 \cdot 16^2 + A \cdot 16^1 + 0 \cdot 16^0 = 256 + 16 = 272$$



Sistemas numéricos posicionales

Sistema hexadecimal (base 16)
EJERCICIOS

- Hallar el valor en el sistema decimal de los números hexadecimales:

D8 :

23:

78:

F:

FF:

A46:

Conversión entre sistemas numéricos



Conversión de base b ($b \neq 10$) a base
10

$$N_{(b)} = s_{n-1}s_{n-2}\dots s_1s_0$$



$$N_{(10)} = [s_{n-1} \cdot b^{n-1} + s_{n-2} \cdot b^{n-2} + \dots + s_1 \cdot b^1 + s_0 \cdot b^0]_{(10)}$$

Conversión de base 2 a base 10

$$110101_2 = 53_{10}$$

Conversión entre sistemas numéricos



Conversión de base 16 a base 10

$1A0_{(16)}$

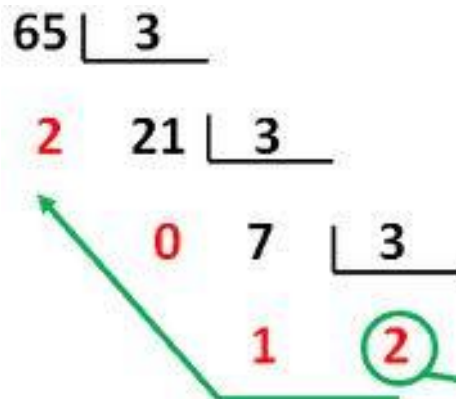


$$1 \cdot 16^2 + 10 \cdot 16^1 + 0 \cdot 16^0 = 416_{(10)}$$

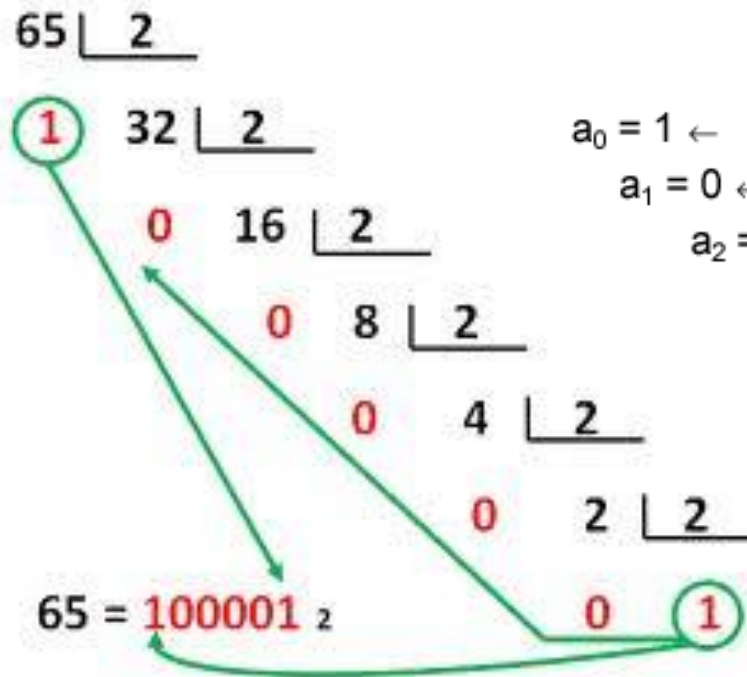
Conversión entre sistemas numéricos



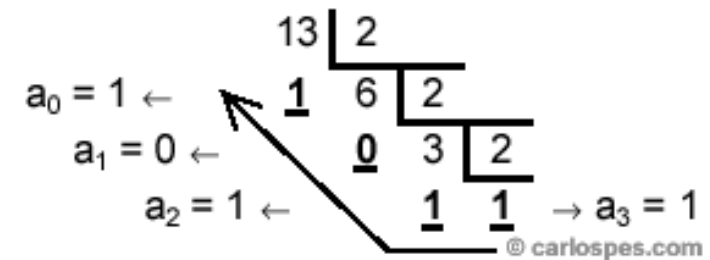
Conversión de base 10 a base b
($b \neq 10$)



$$65 = 2102_3$$



$$65 = 100001_2$$



Conversión entre sistemas numéricos



Conversión de base 10 a base b
($b \neq 10$)

$$\begin{array}{r|l} 460 & 16 \\ \hline 12 & 28 \quad 16 \\ \hline 12 & \textcircled{1} \end{array}$$

$$\begin{array}{r|l} 4711 & 16 \\ \hline 7 & 294 \quad 16 \\ \hline 6 & 18 \quad 16 \\ \hline 2 & \textcircled{1} \end{array}$$

$$\begin{array}{r|l} 13 & 2 \\ \hline 1 & 6 \quad 2 \\ \hline 0 & 3 \quad 2 \\ \hline 1 & 1 \quad \rightarrow a_3 = 1 \end{array}$$

$a_0 = 1 \leftarrow$
 $a_1 = 0 \leftarrow$
 $a_2 = 1 \leftarrow$

© carlospes.com

Conversión entre sistemas numéricos



Conversión entre base 2 y base 16

Binario

Hexadecimal

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

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

Decimal, Hexadecimal, Octal, Binario

Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin
0	0	000	000000000	16	10	020	00010000	32	20	040	00100000	48	30	060	00110000
1	1	001	000000001	17	11	021	000100001	33	21	041	001000001	49	31	061	001100001
2	2	002	000000010	18	12	022	000100010	34	22	042	001000010	50	32	062	001100010
3	3	003	000000011	19	13	023	000100011	35	23	043	001000011	51	33	063	001100011
4	4	004	000000100	20	14	024	000101000	36	24	044	001001000	52	34	064	001101000
5	5	005	000000101	21	15	025	000101001	37	25	045	001001001	53	35	065	001101001
6	6	006	000000110	22	16	026	000101010	38	26	046	001001010	54	36	066	001101010
7	7	007	000000111	23	17	027	000101011	39	27	047	001001011	55	37	067	001101011
8	8	010	00001000	24	18	030	00011000	40	28	050	00101000	56	38	070	00111000
9	9	011	00001001	25	19	031	00011001	41	29	051	00101001	57	39	071	00111001
10	A	012	00001010	26	1A	032	00011010	42	2A	052	00101010	58	3A	072	00111010
11	B	013	00001011	27	1B	033	00011011	43	2B	053	00101011	59	3B	073	00111011
12	C	014	00001100	28	1C	034	00011100	44	2C	054	00101100	60	3C	074	00111100
13	D	015	00001101	29	1D	035	00011101	45	2D	055	00101101	61	3D	075	00111101
14	E	016	00001110	30	1E	036	00011110	46	2E	056	00101110	62	3E	076	00111110
15	F	017	00001111	31	1F	037	00011111	47	2F	057	00101111	63	3F	077	00111111
Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin
64	40	100	01000000	80	50	120	01010000	96	60	140	01100000	112	70	160	01110000
65	41	101	01000001	81	51	121	01010001	97	61	141	01100001	113	71	161	01110001
66	42	102	01000010	82	52	122	01010010	98	62	142	01100010	114	72	162	01110010
67	43	103	01000011	83	53	123	01010011	99	63	143	01100011	115	73	163	01110011
68	44	104	01000100	84	54	124	01010100	100	64	144	01100100	116	74	164	01110100
69	45	105	01000101	85	55	125	01010101	101	65	145	01100101	117	75	165	01110101
70	46	106	01000110	86	56	126	01010110	102	66	146	01100110	118	76	166	01110110
71	47	107	01000111	87	57	127	01010111	103	67	147	01100111	119	77	167	01110111
72	48	110	01001000	88	58	130	01011000	104	68	150	01101000	120	78	170	01111000
73	49	111	01001001	89	59	131	01011001	105	69	151	01101001	121	79	171	01111001
74	4A	112	01001010	90	5A	132	01011010	106	6A	152	01101010	122	7A	172	01111010
75	4B	113	01001011	91	5B	133	01011011	107	6B	153	01101011	123	7B	173	01111011
76	4C	114	01001100	92	5C	134	01011100	108	6C	154	01101100	124	7C	174	01111100
77	4D	115	01001101	93	5D	135	01011101	109	6D	155	01101101	125	7D	175	01111101
78	4E	116	01001110	94	5E	136	01011110	110	6E	156	01101110	126	7E	176	01111110
79	4F	117	01001111	95	5F	137	01011111	111	6F	157	01101111	127	7F	177	01111111

Conversión entre sistemas numéricos



EJERCICIOS

Decimal	Binario	Hexad.
32		
	01110111	
128		
		2C
79		
	10001110	
		210
		AF
153		

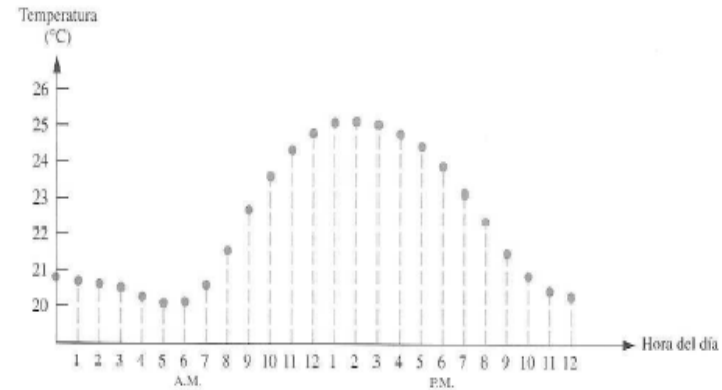
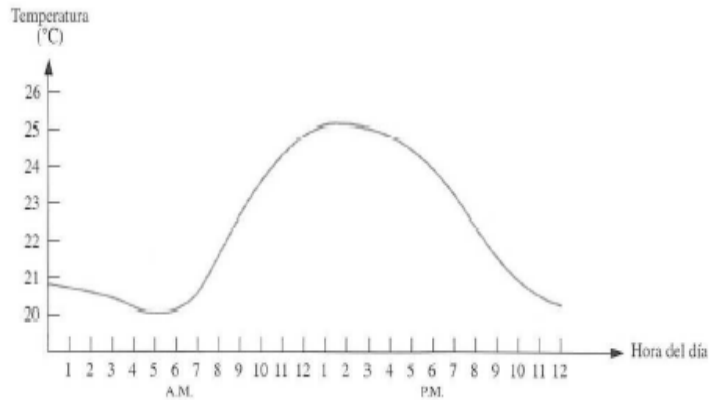


Representación alfanumérica

Código ASCII (American Standard Code for Information Interchange)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Magnitudes analógicas y digitales



▢ Magnitud Analógica: toma valores continuos:

- Por ejemplo la temperatura no varía de entre 20°C y 25°C de forma instantánea, sino que toma todos los infinitos valores intermedios

▢ Magnitud Digital: toma únicamente un conjunto de valores discretos: 22

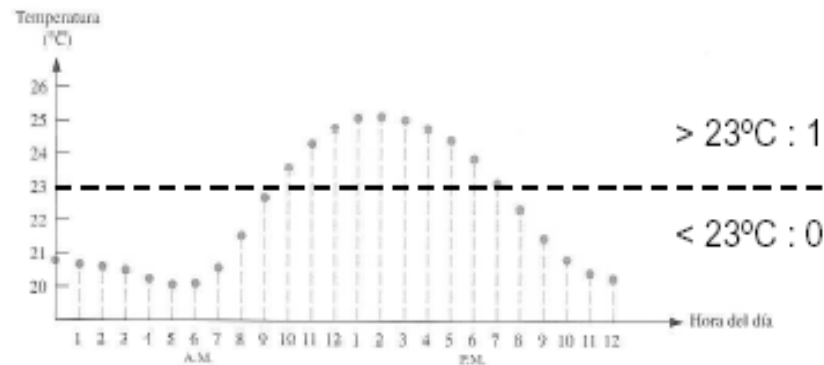
- Por ejemplo, medir la temperatura obteniendo muestras cada hora.

Formato digitalizable

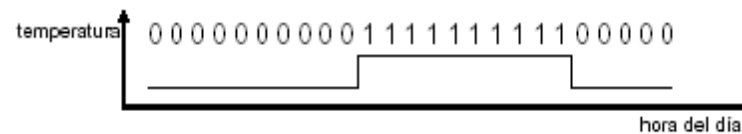
- Hemos obtenido muestras que representan el valor de la temperatura durante un conjunto de intervalos discretos.
- Hemos convertido una magnitud de naturaleza analógica en un formato que se puede digitalizar.
- Es importante tener en cuenta que la figura anterior no es la representación digital de una magnitud analógica, es sólo un formato digitalizable.

Conversión analógico - digital

- Para digitalizar la magnitud es necesario asignar a cada valor muestreado un código digital.
- Supongamos que creamos dos categorías de temperatura



- La señal digital correspondiente sería la siguiente

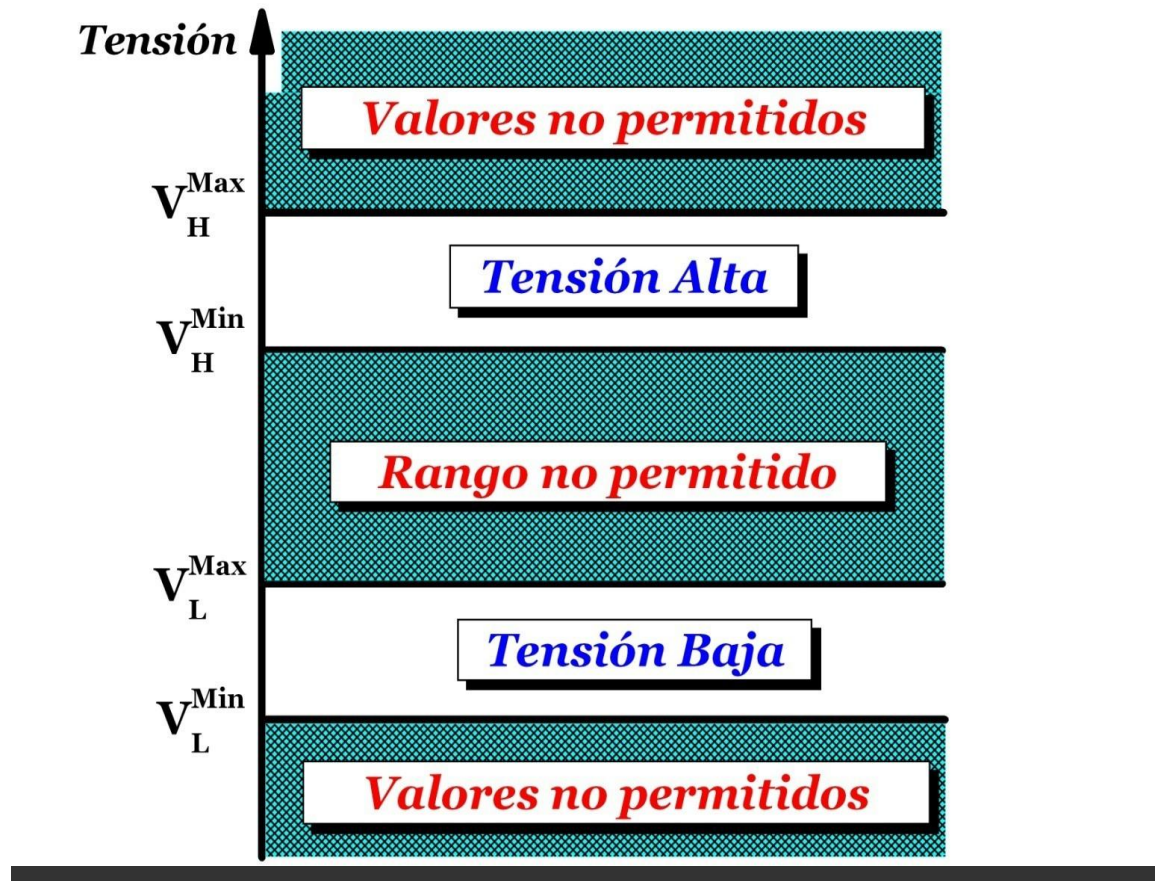


Dígitos binarios

- La electrónica digital utiliza sistemas y circuitos en los que sólo existen dos estados posibles.
- Se representan mediante dos niveles de tensión (**lógica positiva**):
 - **ALTO (HIGH) = 1**
 - **BAJO (LOW) = 0**
- Las combinaciones de los estados – códigos – representan tipos de datos.
- Este sistema de numeración se conoce como **binario (0, 1)**
- Un dígito binario es un **bit** (binary digit)

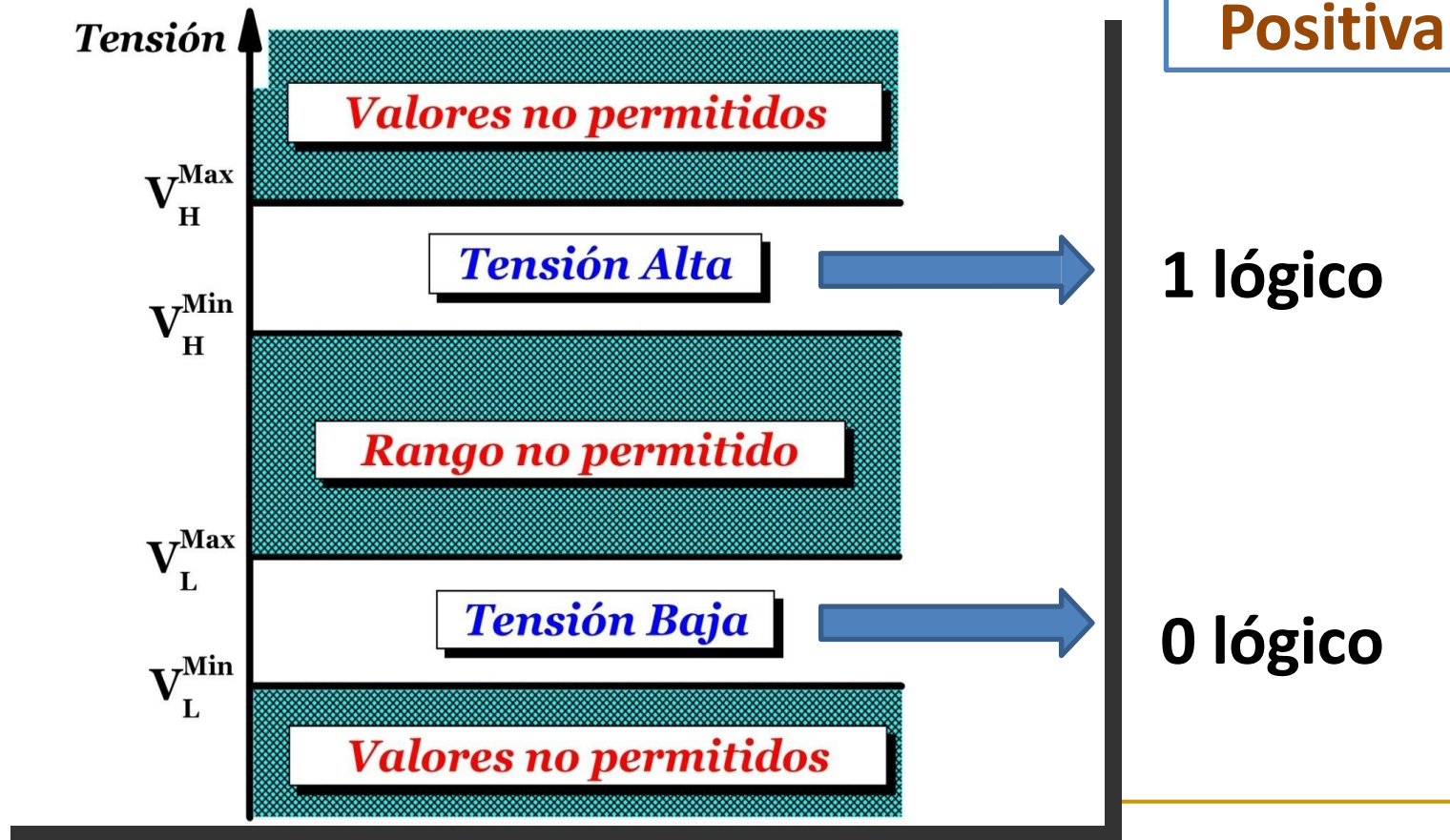
1.- Electrónica Analógica y Electrónica Digital

Valores de una señal digital



1.- Electrónica Analógica y Electrónica Digital

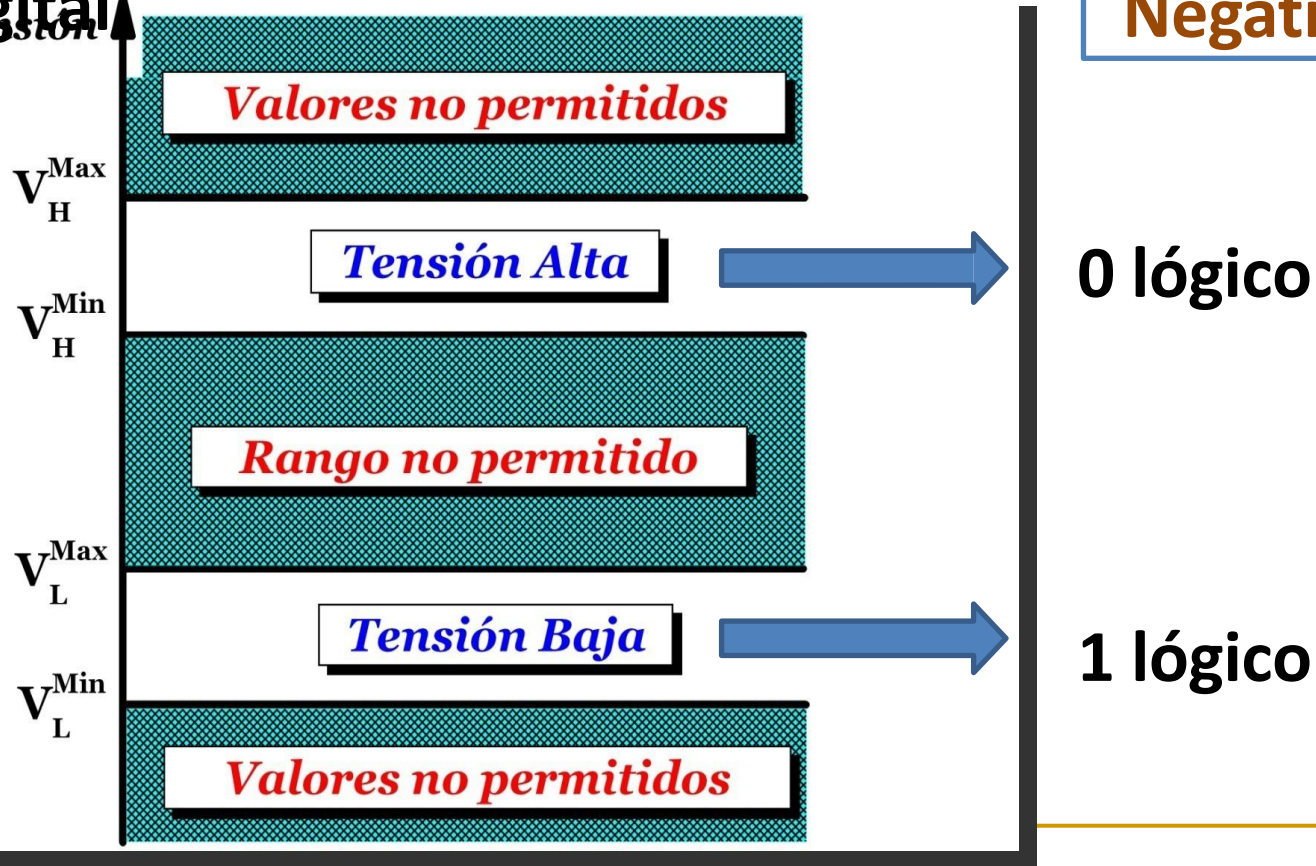
Valores de una señal digital



1.- Electrónica Analógica y Electrónica Digital

Valores de una señal

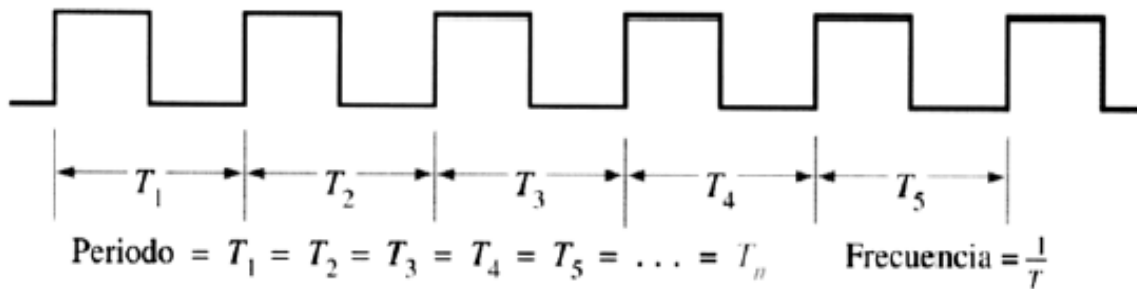
digital



Formas de onda digitales

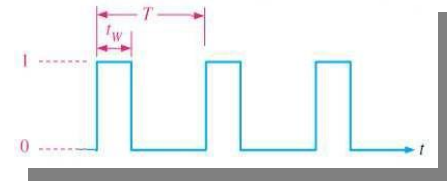
- Periódicos

:

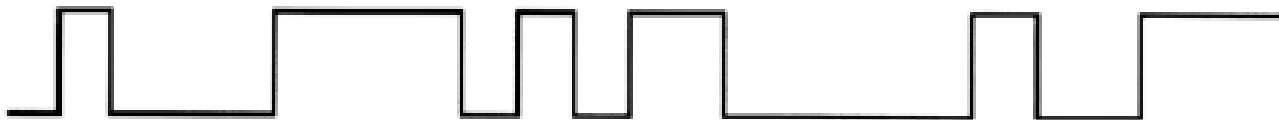


**Señales
periódicas: Ciclo
de trabajo**

$$\left(\frac{t_w}{T} \right) \times 100 \quad (\%)$$

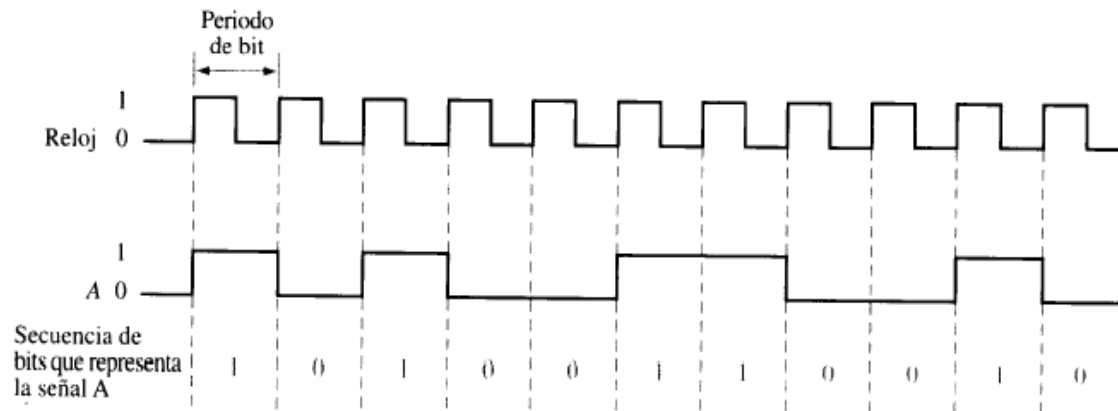


- No
periódicos



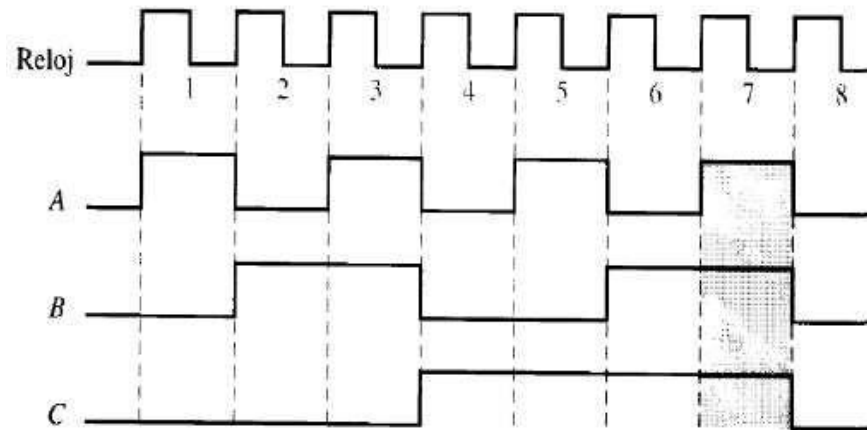
Información binaria

- Los trenes de impulsos representan secuencias de bits.
 - ▢ La señal a nivel bajo representa un 0 binario
 - ▢ La señal a nivel alto representa un 1 binario
- Cada bit dentro de una secuencia ocupa un intervalo de tiempo definido, denominado **periodo de bit**
- **Reloj:** señal que marca el periodo de los bits.
 - ▢ Se usa para sincronizar todas las señales presentes en el sistema.
 - ▢ Las duraciones de bit, vienen marcadas por la separación de dos flancos consecutivos



Información binaria

- **Cronogramas o diagramas de tiempos:** gráfica que muestra la relación temporal real entre dos o más señales (A, B, C) y cómo varía cada señal respecto a las demás.



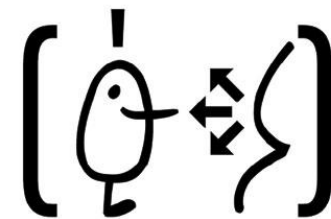
Juan Manuel Orduña Huertas

Fundamentos de computadores I



VNIVERSITAT
DE VALÈNCIA

Escola Tècnica Superior
d'Enginyeria **ETSE-UV** 



VICERECTORAT
DE PARTICIPACIÓ
I PROJECCIÓ
TERRITORIAL

Universitat i Societat

CULLERA