UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Telecomunicación

CODIFICACIÓN DE LA INFORMACIÓN (ANEXO)

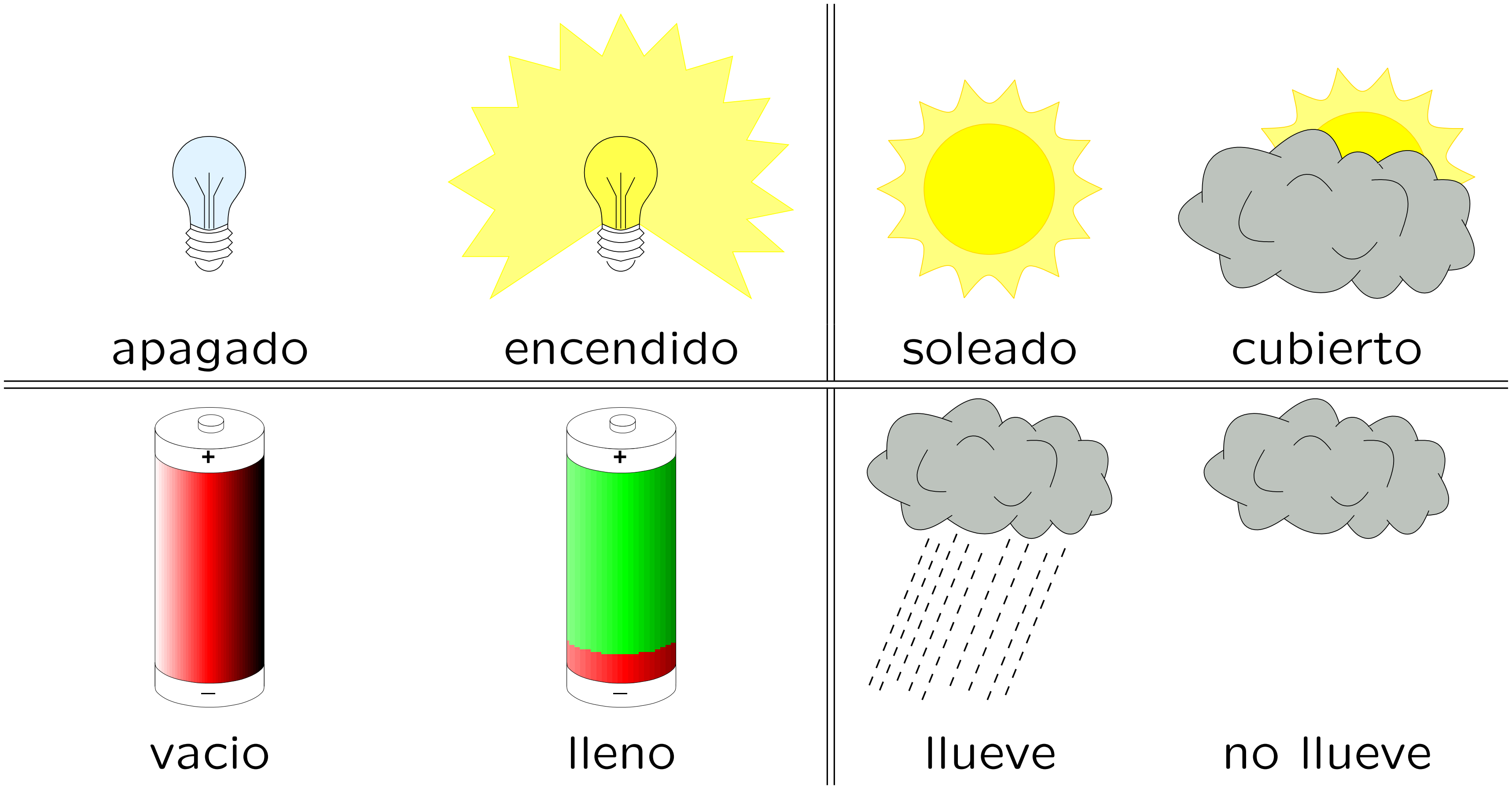
Programación I

Anexo: CODIFICACIÓN DE LA INFORMACIÓN

1. Codificación de la información
2. Representación Posicional de los Números
3. Sistemas de Numeración. Conversiones entre Bases
4. Representación de Números Naturales y Enteros
5. Códigos de Entrada/Salida

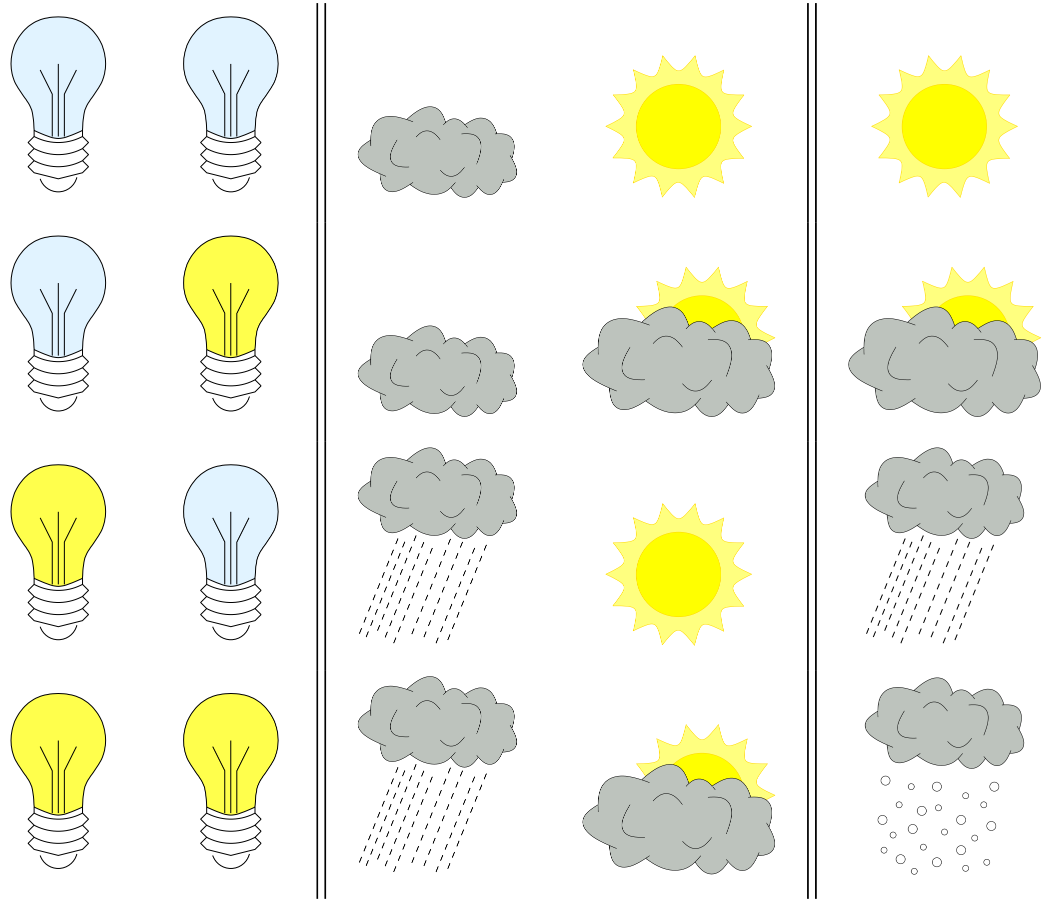
■ CODIFICACIÓN DE LA INFORMACIÓN

BIT es la unidad elemental de información (2 estados diferentes)



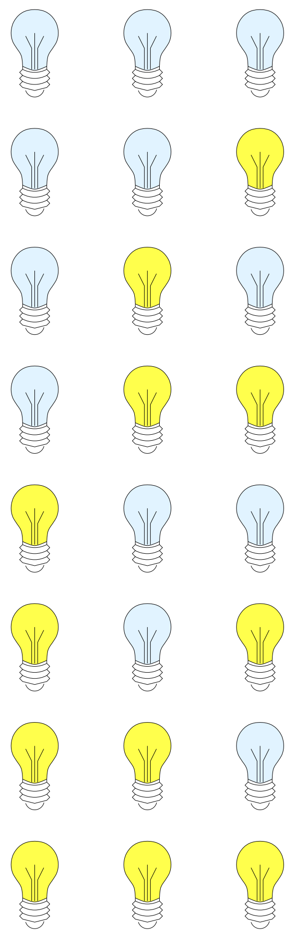
■ CODIFICACIÓN DE LA INFORMACIÓN

- 2 BITs representan 4 (2^2) estados diferentes



■ CODIFICACIÓN DE LA INFORMACIÓN

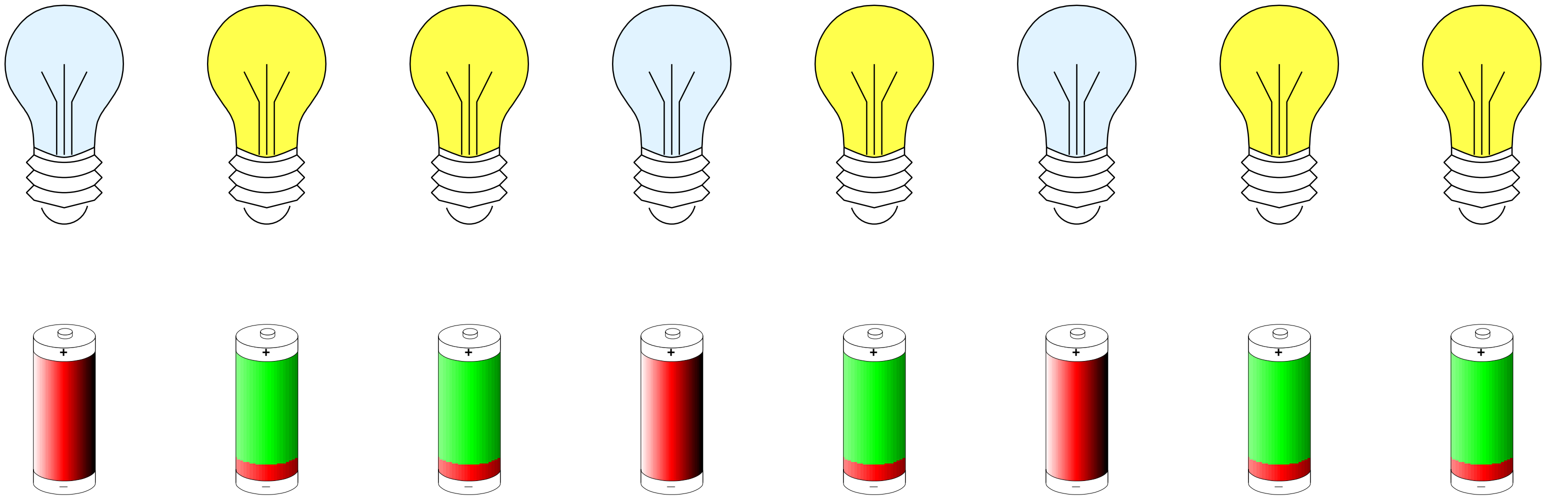
- 3 BITs representan 8 (2^3) estados diferentes



- CODIFICACIÓN DE LA INFORMACIÓN

BYTE 8 BITs. representa 256 (2^8) estados diferentes

Unidad mínima de trabajo.



0 1 1 0 1 0 1 1

Anexo: CODIFICACIÓN DE LA INFORMACIÓN

1. Codificación de la información
2. Representación Posicional de los Números
3. Sistemas de Numeración. Conversiones entre Bases
4. Representación de Números Naturales y Enteros
5. Códigos de Entrada/Salida

- REPRESENTACIÓN DE LOS NÚMEROS
  - Numeración Romana: `XIV`
  - Numeración Arábiga: `14`

■ REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS

- La representación Arábiga es posicional.
  - Un sistema de numeración en base `B` utiliza para representar los números un alfabeto compuesto por `B` símbolos o cifras.
    - ◇ Sistema Binario (base 2):

      `0 1`

    - ◇ Sistema Decimal (base 10):

      `0 1 2 3 4 5 6 7 8 9`

    - ◇ Sistema Hexadecimal (base 16):

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

- REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS
  - Cada cifra contribuye al valor total representado por el número con un valor que depende de:
    - El valor asociado a la cifra en sí
    - La posición de la cifra dentro del número

$$valor = \sum_i cifra_i \times base^i$$

- REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS

  - $3278{,}52_{(10}$

    $$3 \times 10^3 + 2 \times 10^2 + 7 \times 10^1 + 8 \times 10^0 + 5 \times 10^{-1} + 2 \times 10^{-2}$$

  - $1A6F.B3_{(16}$

    $$1 \times 16^3 + 10 \times 16^2 + 6 \times 16^1 + 15 \times 16^0 + 11 \times 16^{-1} + 3 \times 16^{-2}$$

    $$6767{,}6992_{(10}$$

Anexo: CODIFICACIÓN DE LA INFORMACIÓN

1. Codificación de la información
2. Representación Posicional de los Números
3. Sistemas de Numeración. Conversiones entre Bases
4. Representación de Números Naturales y Enteros
5. Códigos de Entrada/Salida

- SISTEMAS DE NUMERACIÓN

Binario	Hexadecimal	Decimal	Octal
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	10
1001	9	9	11
1010	A	10	12
1011	B	11	13
1100	C	12	14
1101	D	13	15
1110	E	14	16
1111	F	15	17
...	...	...	...

- CONVERSIONES ENTRE BASES
  - **De cualquier base a decimal**:

$$valor = \sum_i cifra_i \times base^i$$

    - $110100_{(2} = 52_{(10}$
    - $10100{,}001_{(2} = 20{,}125_{(10}$
    - $25DF.BA_{(16} = 9695{,}7265_{(10}$

■ CONVERSIONES ENTRE BASES

- **De decimal a cualquier base**.
  - Parte Entera: dividimos el número entre la base hasta que el cociente sea cero. Las cifras del número son los restos de dichas divisiones en orden inverso
  - Multiplicamos la parte fraccionaria por la base tantas veces como cifras queramos obtener. Las cifras serán las partes enteras resultados de las anteriores multiplicaciones
  - $26{,}1875_{(10} = 11010{,}0011_{(2}$

$$\begin{array}{rcrcr} 26 & \div & 2 & = & 13\sim\mathbf{0} \\ 13 & \div & 2 & = & 6\sim\mathbf{1} \\ 6 & \div & 2 & = & 3\sim\mathbf{0} \\ 3 & \div & 2 & = & 1\sim\mathbf{1} \\ 1 & \div & 2 & = & 0\sim\mathbf{1} \end{array}$$

$$\begin{array}{rcrcl} 0{,}1875 & \times & 2 & = & \mathbf{0}{,}3750 \\ 0{,}3750 & \times & 2 & = & \mathbf{0}{,}7500 \\ 0{,}7500 & \times & 2 & = & \mathbf{1}{,}5000 \\ 0{,}5000 & \times & 2 & = & \mathbf{1}{,}0000 \end{array}$$

- CONVERSIONES ENTRE BASES
  - **De cualquier base a cualquier base**.
    - Convertimos de la base original a decimal y después de decimal a la base destino.

$$\text{BASE\_ORIGINAL} \Longrightarrow \text{DECIMAL} \Longrightarrow \text{BASE\_DESTINO}$$

■ CONVERSIÓN RÁPIDA

- Es posible realizar **conversión rápida** entre bases cuando una es potencia de otra ($A = B^n$).
  - ∘ Binario $\Longleftrightarrow$ Hexadecimal ($16^1 = 2^4$)
  - ∘ Binario $\Longleftrightarrow$ Octal ($8^1 = 2^3$)
- Se realiza una **conversión directa** entre grupos de cifras indicado por el exponente.
  - ∘ 4 cifras binarias se corresponden con 1 cifra hexadecimal y viceversa
  - ∘ 3 cifras binarias se corresponden con 1 cifra octal y viceversa

▪ CONVERSIÓN RÁPIDA

- $0100101110111111.1011101_{(2} \Longleftrightarrow 25DF.BA_{(16}$

0100101110111111.1011101							(2
2	5	D	F	.	B	A	(16

- $1ABC.C4_{(16} \Longleftrightarrow 0001101010111100.11000100_{(2}$

1	A	B	C	.	C	4	(16
0001101010111100.11000100							(2

- $10001101100.11010_{(2} \Longleftrightarrow 2154.64_{(8}$

10001101100.11010							(2
2	1	5	4	.	6	4	(8

- $537.24_{(8} \Longleftrightarrow 101011111.010100_{(2}$

5	3	7	.	2	4	(8
101011111.010100						(2

Anexo: CODIFICACIÓN DE LA INFORMACIÓN

1. Codificación de la información
2. Representación Posicional de los Números
3. Sistemas de Numeración. Conversiones entre Bases
4. Representación de Números Naturales y Enteros
5. Códigos de Entrada/Salida

■ REPRESENTACIÓN DE NÚMEROS NATURALES

- Se representan en BINARIO PURO
- **N** bits permiten representar un rango de valores desde 0 hasta $2^{\mathsf{N}} - 1$

$$(\overbrace{00\cdots000}^{\mathsf{N}} \Longleftrightarrow \overbrace{11\cdots111}^{\mathsf{N}})$$

■ REPRESENTACIÓN DE NÚMEROS NATURALES

- Ejemplo para **4** bits: Rango desde **0** hasta $\mathbf{+15}$

0000	$\equiv$	0
0001	$\equiv$	1
0010	$\equiv$	2
0011	$\equiv$	3
0100	$\equiv$	4
0101	$\equiv$	5
0110	$\equiv$	6
0111	$\equiv$	7
1000	$\equiv$	8
1001	$\equiv$	9
1010	$\equiv$	10
1011	$\equiv$	11
1100	$\equiv$	12
1101	$\equiv$	13
1110	$\equiv$	14
1111	$\equiv$	15

- REPRESENTACIÓN DE NÚMEROS ENTEROS
  - Signo-Magnitud
  - Complemento a uno
  - **Complemento a dos**

- REPRESENTACIÓN **SIGNO-MAGNITUD** DE NÚMEROS ENTEROS.
  - El bit más significativo para el signo.
  - El resto la magnitud en binario puro.
  - **N** bits permiten representar un rango de valores desde $-(2^{N-1}-1)$ hasta $(2^{N-1}-1)$

$$\overbrace{00\cdots000}^{N} \Longleftrightarrow \overbrace{01\cdots111}^{N} \quad Positivos$$
$$\overbrace{10\cdots000}^{N} \Longleftrightarrow \overbrace{11\cdots111}^{N} \quad Negativos$$

- REPRESENTACIÓN **SIGNO-MAGNITUD** DE NÚMEROS ENTEROS.
  - Ventajas/Desventajas

    $\Downarrow$ El cero tiene dos representaciones.

    - $\overbrace{00\cdots000}^{\mathsf{N}} \Longleftrightarrow +0$
    - $\overbrace{10\cdots000}^{\mathsf{N}} \Longleftrightarrow -0$

    $\Downarrow$ La operación a realizar (suma o resta) depende de los operandos.

    $\Downarrow$ Diferentes circuitos para realizar sumas y restas.

■ NÚMEROS ENTEROS: **SIGNO-MAGNITUD**

- Ejemplo para **4** bits. Rango desde **-7** hasta **+7**

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

- REPRESENTACIÓN **COMPLEMENTO A UNO** DE NÚMEROS ENTEROS.

  - El signo se encuentra incluido en la representación del número.
  - Cambiar el signo a un número consiste en **cambiar ceros por unos y unos por ceros** en su representación.
  - **N** bits permiten representar un rango de valores desde $-(2^{N-1} - 1)$ hasta $(2^{N-1} - 1)$

$$\overbrace{00\cdots000}^{N} \Longleftrightarrow \overbrace{01\cdots111}^{N} \quad Positivos$$
$$\overbrace{11\cdots111}^{N} \Longleftrightarrow \overbrace{10\cdots000}^{N} \quad Negativos$$

- REPRESENTACIÓN **COMPLEMENTO A UNO** DE NÚMEROS ENTEROS.

  - Ventajas/Desventajas

    $\Downarrow$ El cero tiene dos representaciones.

    - $\overbrace{00\cdots000}^{\text{N}} \Longleftrightarrow +0$

    - $\overbrace{11\cdots111}^{\text{N}} \Longleftrightarrow -0$

    $\Uparrow$ Una sola operación para realizar sumas y restas.

    $\Downarrow$ El acarreo final se debe sumar al resultado.

▪ NÚMEROS ENTEROS: **COMPLEMENTO A UNO**

- Ejemplo para **4** bits. Rango desde -**7** hasta **+7**

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

```
  -7        1000
+ +3      + 0011
------    --------
  -4      0 1011
```

```
  +4        0100
+ +3      + 0011
------    --------
  +7      0 0111
```

```
          1 1
  +7        0111
+ -3      + 1100
------    --------
  +4      1 0011
           + ..1
           ------
            0100
```

```
          1
  -4        1011
+ -3      + 1100
------    --------
  -7      1 0111
           +...1
           ------
            1000
```

- REPRESENTACIÓN **COMPLEMENTO A DOS** DE NÚMEROS ENTEROS.
  - El signo se encuentra incluido en la representación del número.
  - Cambiar el signo a un número consiste en **cambiar ceros por unos y unos por ceros y sumar uno** en su representación.
  - **N** bits permiten representar un rango de valores desde -2^{N-1} hasta $(2^{N-1} - 1)$

$$\overbrace{00\cdots000}^{N} \iff \overbrace{01\cdots111}^{N} \quad \textit{Positivos}$$
$$\overbrace{11\cdots111}^{N} \iff \overbrace{10\cdots000}^{N} \quad \textit{Negativos}$$

- REPRESENTACIÓN **COMPLEMENTO A DOS** DE NÚMEROS ENTEROS.

  - Ventajas/Desventajas

    $\Uparrow$ El cero tiene una única representación.

    - $\overbrace{00\cdots000}^{\text{N}} \Longleftrightarrow +0$

    - $\overbrace{00\cdots000}^{\text{N}} \Longleftrightarrow -0$

    $\Uparrow$ Una sola operación para realizar sumas y restas.

    $\Uparrow$ El acarreo final se desecha.

▪ NÚMEROS ENTEROS: **COMPLEMENTO A DOS**

- Ejemplo para **4** bits. Rango desde **-8** hasta **+7**

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

```
                11
   -7         1001
 + +3       + 0011
------     --------
   -4       0 1100
```

```
   +4         0100
 + +3       + 0011
------     --------
   +7       0 0111
```

```
            1 111
   +7         0111
 + -3       + 1101
------     --------
   +4         0100
```

```
            1 1
   -4         1100
 + -3       + 1101
------     --------
   -7         1001
```

Anexo: CODIFICACIÓN DE LA INFORMACIÓN

1. Codificación de la información
2. Representación Posicional de los Números
3. Sistemas de Numeración. Conversiones entre Bases
4. Representación de Números Naturales y Enteros
5. Códigos de Entrada/Salida

- CÓDIGOS DE ENTRADA/SALIDA
  - La representación de INFORMACIÓN por medio de CARACTERES es uno de los medios más usuales en el tratamiento de los datos.

	Entrada/Salida	Almacenamiento	Procesamiento
	Otros	Otros	Numérico
Inf. Abstracta	Carácter	Carácter	Numérico
	Otros	Numérico	Numérico
Inf. Numérica	Carácter	Numérico	Numérico

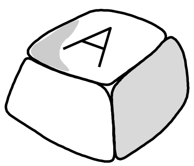
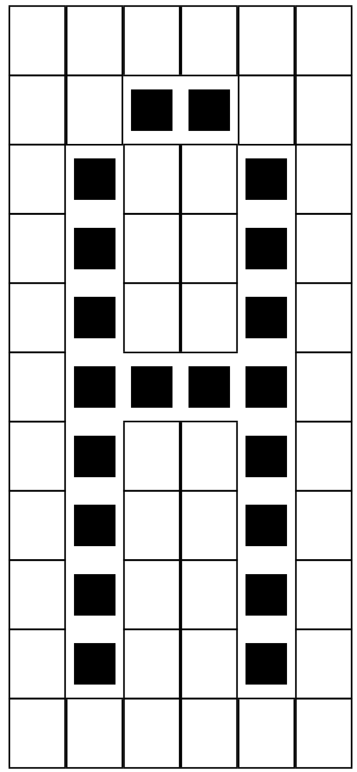
CÓDIGOS DE ENTRADA/SALIDA

- CARACTERES
  - Letras Mayúsculas: **A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**
  - Letras Minúsculas: **a b c d e f g h i j k l m n o p q r s t u v w x y z**
  - Dígitos: **0 1 2 3 4 5 6 7 8 9**
  - Símbolos de puntuación: **, ; . : ? ! ( ) [ ]** { }
  - Otros símbolos: **= + - / * > < # $ % ^ & _ ~**
  - Caracteres de control
- Se representan NUMÉRICAMENTE según una tabla de correspondencia

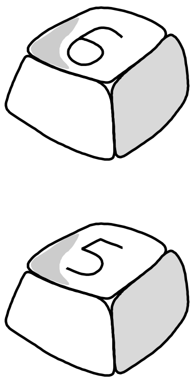
- TABLA ASCII (representación de caracteres) [0..31] Car. Control

Rep	Car	Rep	Car	Rep	Car	Rep	Car	Rep	Car	Rep	Car
32	SP	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	ˆ	110	n	126	˜
47	/	63	?	79	O	95	_	111	o	127	DEL

- Entrada/Salida de Carácter

Entrada		Rep. Interna Carácter		Salida
	$\Longrightarrow$	65	$\Longrightarrow$	

- Entrada/Salida de Números (mediante caracteres)

Entrada		Rep. Interna Numérica		Salida
	$\Longrightarrow$	65	$\Longrightarrow$