CAPÍTULO 1

INTRODUCCIÓN.

TECNOLOGÍA Y ORGANIZACIÓN DE COMPUTADORES

1º Grado en Ingeniería Informática.

Tema 1. Introducción

RESUMEN:

- En este tema se cubren conceptos básicos para facilitar el resto seguimiento del resto de la asignatura.
- Se define la estructura funcional de un computador.
- Se indican distintos niveles de abstracción desde los que se puede describir un computador.
- Se muestran las distintas representaciones de datos numéricos que procesa el computador .
- Se introducen los sistemas analógicos y digitales, con sus diferencias en el modo de procesamiento.

OBJETIVOS:

- Conocer la organización básica y componentes de un computador
- Identificar los factores que determinan las prestaciona básicas de un computador
- Comprender la conveniencia de describir un computador en diferentes niveles de abstracción para facilitar su comprensión, su diseño y su utilización.

Tema 1. Introducción

CONTENIDOS:

1.1. Conceptos básicos
1.2. Estructura funcional de un computador.
1.3. Representación de datos numéricos.
1.4. Niveles conceptuales de descripción de un computador.
1.5. Sistemas analógicos y digitales.

BIBLIOGRAFÍA:

[PRI06]:1 ; [PRI05]:1

Tema 1. Introducción

CONTENIDOS:

1.1. Conceptos básicos

1.2. Estructura funcional de un computador.

1.3. Representación de datos numéricos.

1.4. Niveles conceptuales de descripción de un computador.

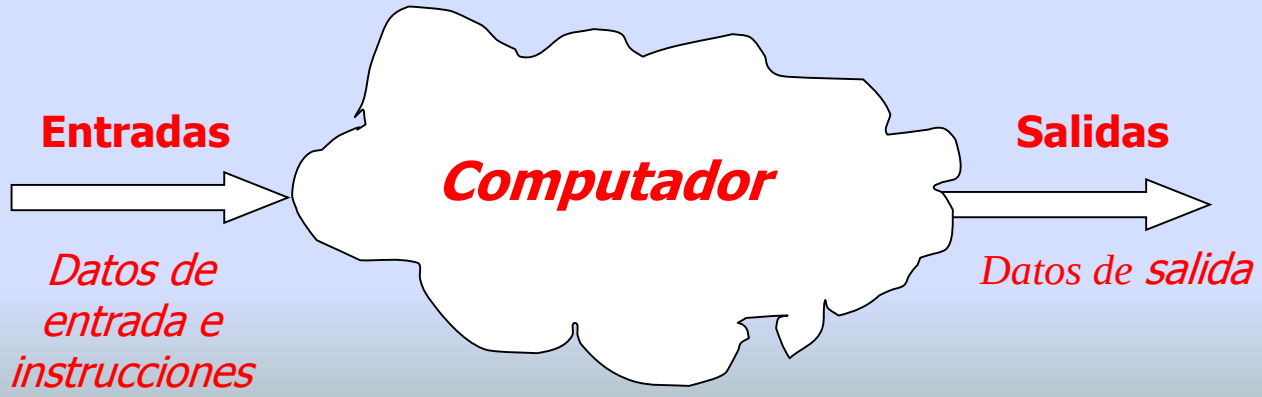
1.5. Sistemas analógicos y digitales.

1.1 Conceptos básicos

- **Informática:**
  - Es el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras electrónicas.

1.1 Conceptos básicos

- **Computador, Computadora u ordenador:**
  - Es una máquina capaz de aceptar unos datos de entrada, efectuar con ellos operaciones lógicas y aritméticas, y proporcionar la información resultante a través de un medio de salida; todo ello sin intervención de un operador humano y bajo el control de un programa de instrucciones previamente almacenado en el propio computador.



1.1 Conceptos básicos

- **Calculadora:**
  - Es una máquina capaz de efectuar operaciones aritméticas bajo el control directo del usuario.

- **Datos:**
  - Son conjuntos de símbolos utilizados para expresar o representar un valor numérico, un hecho, un objeto o una idea; en la forma adecuada para ser objeto de tratamiento.

1.1 Conceptos básicos

- **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.

1.1 Conceptos básicos

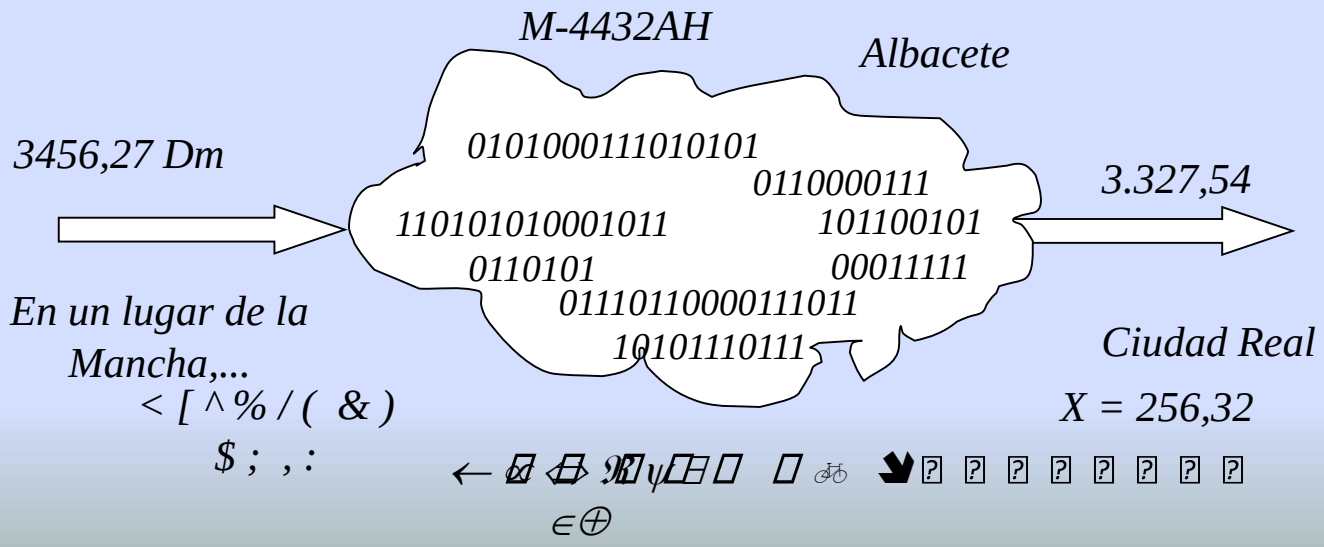
- **Ejemplo de código (binario):**
  - Código Morse (1791-1872)

A ·–	J ·–––	R ·–·
B –···	K –·–	S ···
C –·–·	L ·–··	T –
D –··	M––	U ··–
E ·	N –·	V ···–
F ··–·	Ñ––·––	W ·––
G ––·	O–––	X –··–
H ····	P ·––·	Y –·––
I ··	Q––·–	Z ––··

1 ·———–	6 –····
2 ··–––	7 ––···
3 ···––	8 –––··
4 ····–	9 ––––·
5 ·····	0 –––––

1.1 Conceptos básicos

- 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 (código binario) representados por 0 y 1. En las E/S se efectúa la transformación.



1.1 Conceptos básicos

- **UNIDADES DE INFORMACIÓN:**
  - **Bit (b)** →
    - unidad más elemental o capacidad mínima de información.
    - Es una posición o variable que toma el valor 0 ó 1.
  - **Byte (B)** →
    - En la actualidad se considera sinónimo de grupo de 8 bits.
    - (Históricamente: nº de bits necesarios para almacenar un carácter.)

1.1 Conceptos básicos

- ***Ejemplo*** → ***G24 B***

G	→	**0100 0111**
2	→	**0011 0010**
4	→	**0011 0100**
SP	→	**0010 0000**
B	→	**0100 0010**

1.1 Conceptos básicos

- ***MÚLTIPLOS (de bit, Byte, palabra,...):***
  - 1 **Kilo** (K) $= 2^{10} = 1024 \approx 10^3$
  - 1 **Mega** (M) $= 2^{10}K = 2^{20} = 1.048.576 \approx 10^6$
  - 1 **Giga** (G) $= 2^{10}M = 2^{30} = 1.073.741.824 \approx 10^9$
  - 1 **Tera** (T) $= 2^{10}G = 2^{40} \approx 10^{12}$
  - 1 **Peta** (P) $= 2^{10}T = 2^{50} \approx 10^{15}$
  - 1 **Exa**(E) $= 2^{10}P = 2^{60} \approx 10^{18}$

1.1 Conceptos básicos. Prefijos binarios

- *kibi-* (símbolo *Ki*), 2^{10} = 1.024
- *mebi-* (símbolo *Mi*), 2^{20} = 1.048.576
- *gibi-* (símbolo *Gi*), 2^{30} = 1.073.741.824
- *tebi-* (símbolo *Ti*), 2^{40} = 1.099.511.627.776
- *pebi-* (símbolo *Pi*), 2^{50} = 1.125.899.906.842.624
- *exbi-* (símbolo *Ei*), 2^{60} = 1.152.921.504.606.846.976

- Los prefijos SI no se usan para indicar múltiplos binarios.
- La parte *bi* del prefijo viene de la palabra binario, por ejemplo, kibibyte significa un kilobinario byte, que son 1.024 bytes
- Ejemplos: kibibit (Kibit o Kib) o kibibyte (KiB), Mibibyte (MiB).

  http://es.wikipedia.org/wiki/Prefijos_binarios

1.1. Conceptos básicos. Programas e instrucciones

- Las instrucciones se forman con elementos o símbolos tomados de un determinado repertorio, y se construyen siguiendo unas reglas precisas.

- Todo lo relativo a los símbolos y reglas para construir o redactar con ellos un programa se denomina lenguaje de programación.

1.1. Conceptos básicos. Programas e instrucciones

- Una instrucción *es un conjunto de símbolos que representan una orden de operación o tratamiento para la computadora.* Las operaciones suelen realizarse con datos.

- Un programa *es un conjunto ordenado de instrucciones que se dan a la computadora indicándole las operaciones o tareas que se desea realice.*

1.1. Conceptos básicos. Tipos de instrucciones.

- **Instrucciones de transferencias de datos.**
  - Transferir datos de una unidad a otra. Por ejemplo, de E/S.
- **Instrucciones de tratamiento.**
  - Instrucciones aritmético-lógicas.
- **Instrucciones de bifurcación y saltos.**
  - Permiten alterar el orden secuencial de ejecución.
  - Saltos y llamadas/retornos de subrutinas (procedimientos)
- **Otras instrucciones.**
  - Detener el funcionamiento de la computadora, a la espera de una acción del operador,...

1.1. Conceptos básicos. Lenguaje máquina

- El lenguaje máquina es el único que entienden los circuitos del computador (CPU). Las instrucciones se forman por bits agrupados en campos:
  - **Campo de código de operación** indica la operación correspondiente a la instrucción.
  - **Campos de dirección** especifican los lugares (o posición) dónde se encuentra o donde ubicar los datos con los que se opera.

1.1. Conceptos básicos. Lenguaje de alto nivel

- **El lenguaje máquina tiene serios inconvenientes:**
  - depende del modelo de procesador;
  - el repertorio de instrucciones es muy reducido, conteniendo sólo operaciones muy elementales;
  - es muy laborioso programar con él por tener que utilizar sólo números; etc.

- **Para evitar estos problemas:**
  - Se han ideado **lenguajes de alto nivel**, que no dependen de la computadora, para facilitar la tarea de programación.
  - Ejemplos: BASIC, FORTRAN, COBOL, Pascal, Logo, C, Ada, Prolog, Lisp, etc.

Tema 1. Introducción

CONTENIDOS:

1.1. Conceptos básicos.

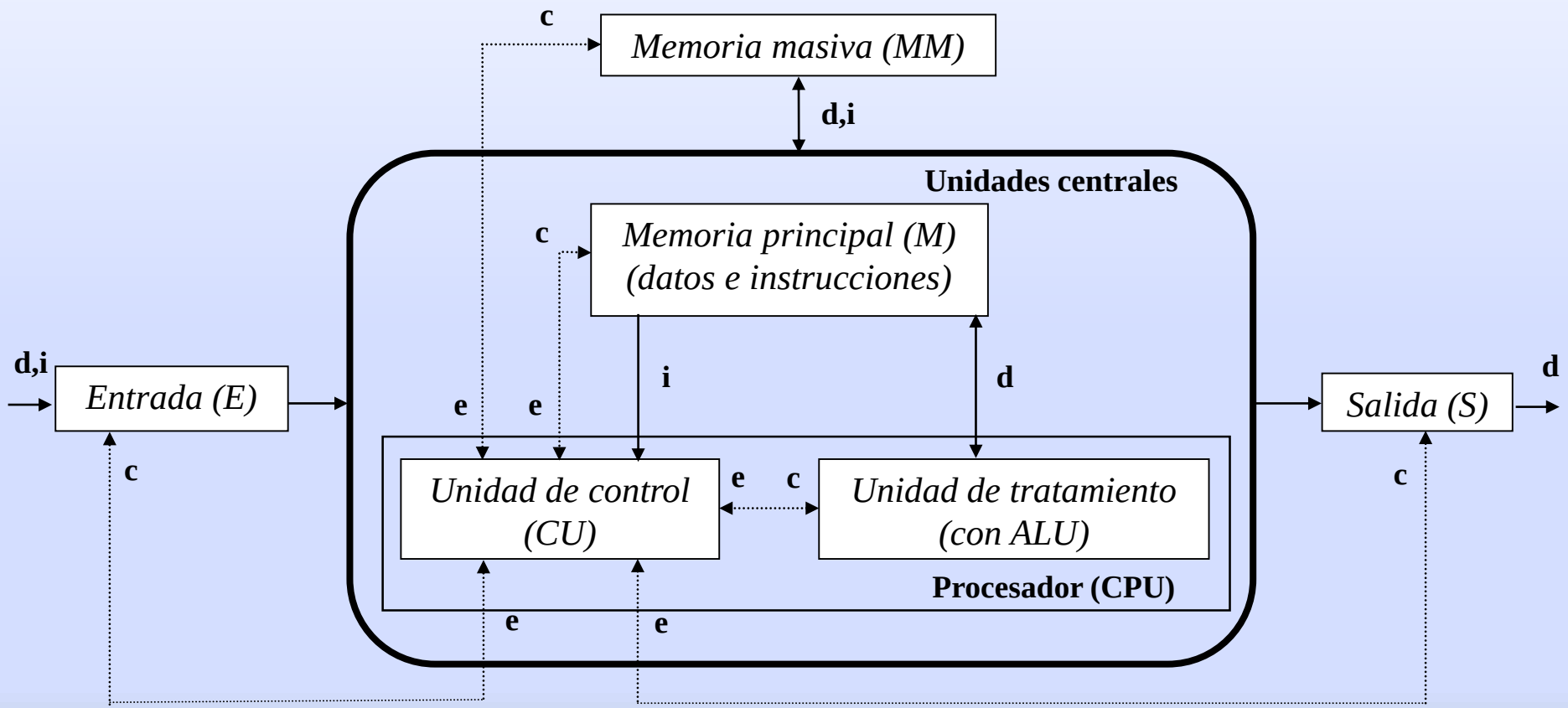
1.2. Estructura funcional de un computador.

1.3. Representación de datos numéricos.

1.4. Niveles conceptuales de descripción de un computador.

1.5. Sistemas analógicos y digitales.

1.2. Estructura funcional de un computador



d: datos ; i: instrucciones
e: señales de estado c: señales de control

1.2. Estructura funcional de un computador

- **UNIDAD DE ENTRADA (E).**
  - Dispositivo por donde se introducen en la computadora los datos e instrucciones. Transforman las informaciones de entrada en señales binarias de naturaleza eléctrica. Ejemplos: un teclado, un digitalizador, una lectora de tarjetas de crédito, etc..
- **UNIDAD DE SALIDA (S).**
  - Dispositivo por donde se obtienen los resultados de los programas ejecutados en la computadora. Transforman las señales eléctricas binarias en caracteres escritos o gráficos visualizados. Ejemplos: un monitor de vídeo, una impresora o un registrador gráfico

1.2. Estructura funcional de un computador

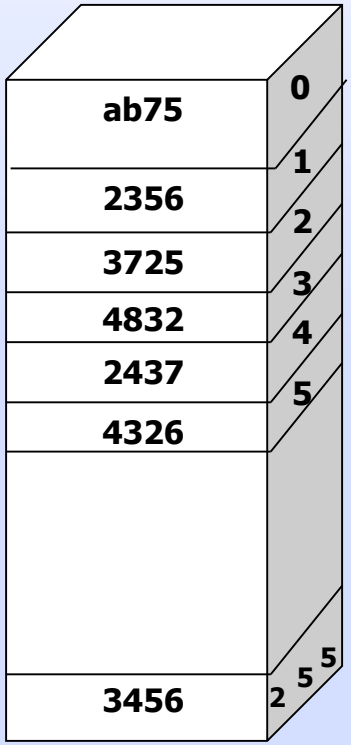
- **MEMORIA (M)**
  - Es la unidad donde se almacenan tanto los datos como las instrucciones. Existen dos tipos básicos de memoria, diferenciados principalmente por su velocidad.
    - Memoria principal, o central, o interna.
    - Memoria masiva auxiliar, secundaria o externa.

1.2. Estructura funcional de un computador

- **MEMORIA (M)**
  - Memoria principal, o central, o interna.
    - Actúa con gran velocidad → ligada directamente a las unidades más rápidas (UC y ALU).
    - Para que un programa se ejecute debe estar almacenado (**cargado**) en la memoria principal.
    - Son circuitos integrados (IC).
    - Estructurada en **posiciones** (**palabras** de memoria) de un determinado número de bits.
    - Para leer o escribir una información es necesario dar la dirección de la posición.

1.2. Estructura funcional de un computador

- Se accede (lee o escribe) a las posiciones (palabras) de memoria, por medio de direcciones.



Dirección	Contenido
0	ab75
1	2356
2	3725
3	4832
4	2437
5	4326
.	.
.	.
.	.
255	3456

Es muy importante distinguir entre contenidos y direcciones

Tema 1. Introducción

CONTENIDOS:

1.1. Conceptos básicos.
1.2. Estructura funcional de un computador.
1.3. Representación de datos numéricos.
1.4. Niveles conceptuales de descripción de un computador
1.5. Sistemas analógicos y digitales.

1.3 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.

1.3 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.

1.3 Representación de datos numéricos

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

Datos de tipo real

1.3 Representación de datos numéricos. 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

1.3 Representación de datos numéricos. 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	

1.3 Representación de datos numéricos. 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.**

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

87	16
07	5

  - Signo y magnitud: como N>0 → 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 + 1000000 = 11010111

1.3 Representación de datos numéricos. 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.**

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

  87 | 16
  07 | 5

  - Signo y magnitud: como N<0 → 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 + 1000000 = 00101001

1.3 Representación de datos numéricos. Datos de tipo entero representados en binario

- Ejemplo: representaciones de datos de n=4 bits de tipo entero:

Nº Decimal	Sin signo	Signo y magnitud	Complemento a 1	Complemento a 2	Sesgada (Sesgo=8)
(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

1.3 Representación de datos numéricos. Datos de tipo entero representados en binario. Extensión del signo.

- La extensión del signo es una consecuencia directa de la utilización de un número mayor de bits que el estrictamente necesario para representar un dato numérico.

- Supóngase que se tienen datos enteros con n'=6 bits y se quiere utilizar una representación para datos enteros con n=8 bits y se utiliza la representación con signo en complemento a 2. Se desea estudiar cómo se haría la extensión del signo para los datos enteros con signo

$$+24)_{10} \text{ y } -24)_{10}$$

$+24)_{10} = 011000)_2$

$-24)_{10} = C_2\ (+24) = C_2\ (011000) = C_1\ (011000)+1 = 100111+1 = 101000)_2$

El bit en rojo sería el bit de signo.

1.3 Representación de datos numéricos. Datos de tipo entero representados en binario. Extensión del signo.

1. Para el caso de datos enteros positivos (en la representación complemento a 2), para la extensión del signo basta con completar con ceros a la izquierda del dato representado hasta llegar a completar la palabra. Por ejemplo:

   $+24)_{10} = 011000)_2$; con n = 8 bits $+24)_{10} = 00011000)_2$

2. Para el caso de datos enteros negativos (en la representación complemento a 2), para la extensión del signo basta con completar con unos a la izquierda del dato representado en complemento a 2 hasta llegar a completar la palabra. Por ejemplo:

   $-24)_{10} = 101000$; con n = 8 bits $-24)_{10} = 11101000$

- El bit de signo se extiende hasta completar todos los bits de la palabra.

1.3 Representación de datos numéricos. Datos de tipo entero representados en binario. Extensión del signo.

En el caso de datos enteros negativos (en la representación de complemento a 2), la extensión del signo sería equivalente a hacer el complemento a 2 de todo el dato positivo extendido considerando todos los bits de la representación. Por ejemplo:

$+24)_{10} = 011000)_2$; con n = 8 bits $+24)_{10} = 00011000)_2$

$C_2\ (+24) = C_2\ (00011000) = \quad = C_1\ (00011000) + 1 =$

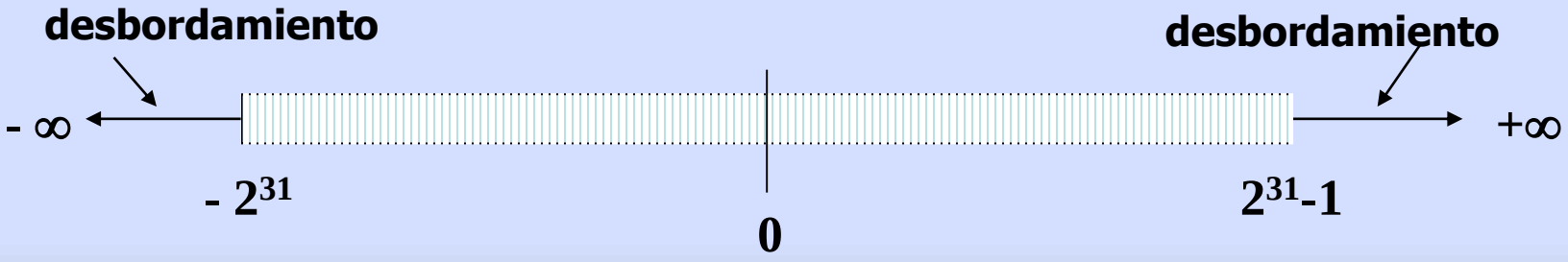
$11100111 + 1 = 11101000$

1.3 Representación de datos numéricos. 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(máximo) = $2^{31}-1$ = 2.147.483.647

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



1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

- Este tipo de representación numérica se utiliza para representar números reales (expresados estos en cualquier base) con un número fijo de cifras (p) para representar la parte entera y otro número fijo de cifras (q) para representar la parte fraccionaria, separadas ambas por una coma o punto decimal.

- En algunos casos se considera que la representación de números enteros es una representación en coma fija en la que no hay cifras en la parte fraccionaria (q=0).

- Ejemplos:
  - En base 10: 356,2378 (p=3 , q=4)
  - En base 2 : 1100,01011 (p=4 , q=5)

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

- Al considerar un número fijo de cifras (p) para representar la parte entera y otro número fijo de cifras (q) para representar la parte fraccionaria se pueden cometer inexactitudes en la representación del dato numérico.

- Esto da lugar a errores en la representación del dato numérico bien sea por truncamiento o por redondeo.

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

Recordemos que (Seminario 1), para la conversión de un número de decimal a binario se procede como sigue:

1. La **parte entera** del número binario se obtiene dividiendo entre 2 la parte entera del número decimal de partida, y de los cocientes que sucesivamente se vayan obteniendo (sin obtener decimales en el cociente).

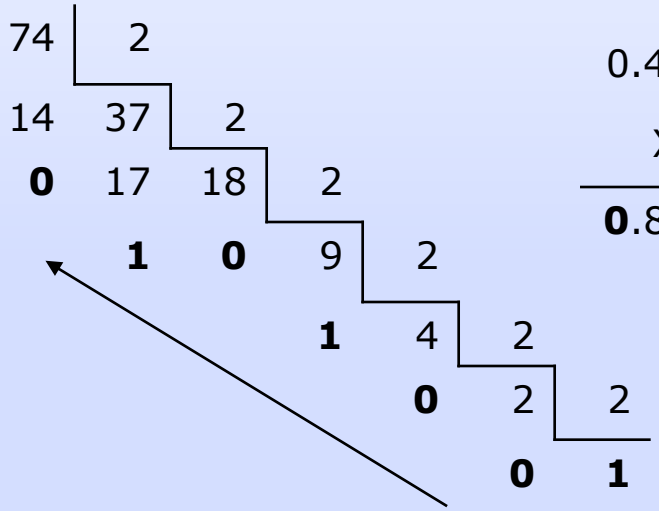
2. La **parte fraccionaria** del número binario se obtiene multiplicando por 2 sucesivamente la parte fraccionaria del número decimal de partida y las partes fraccionarias que se van obteniendo en los productos sucesivos.

3. El número binario se forma con **los restos de las divisiones** y el **último cociente** (tomando el último cociente como el bit más significativo y el primer residuo como el bit menos significativo), “,”, las **partes enteras de los productos** obtenidos (siendo el bit más significativo el del primer producto, y el menos significativo el del último producto).

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

- **Ejemplo:**

  Transformar de decimal a binario: $74,423)_{10}$



0.423	0.846	0.692	0.384	0.768
X 2	X 2	X 2	X 2	X 2
0.846	**1**.692	**1**.384	**0**.768	**1**.536 ...

Parte fraccionaria: 01101...

Parte entera:1001010

$$74,423)_{10} = 1001010, 01101...)_2$$

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

- Sea un número decimal con una parte entera (E) y una parte fraccionaria (F).
- Para representar en binario la parte entera (E) del número decimal se requieren, al menos, p bits tales que:

$$2^{p-1} \leq E < 2^{p}$$

- La representación en binario de la parte fraccionaria (F) de un número decimal con cifras fraccionarias puede dar lugar a una parte fraccionaria en binario con un número de cifras mucho mayor que las de la parte fraccionaria del número decimal o incluso infinitas cifras. Si el número binario se almacena con un número prefijado (q) de bits se producirá en la representación binaria un error de truncamiento.

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

- **Ejemplo:**

  Transformar de decimal a binario: $74{,}423)_{10}$

  a) Parte entera (E):

  - Número de bits (p) mínimo para representar la parte entera:

  $$2^{p-1} \leq 74 < 2^{p} \; ; 2^{6} = 64 \leq 74 < 2^{7} = 128 \Rightarrow p = 7 \; bits$$

  - Parte entera: 1001010

  b) Parte fraccionaria (F):

  - Número de bits (q) para representar la parte fraccionaria: a priori desconocido.
  - Suponiendo q = 5 bits para representar la parte fraccionaria:
  - Parte fraccionaria: 01101

  c) Representación binaria en coma fija del número decimal:

  $$\mathbf{74{,}423)_{10} = 1001010,\ 01101)_{2}}$$

  d) Representación decimal del dato binario 1001010,01101 = 74,40625

  e) Se produce un error de truncamiento.

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto fijo.

- **PROBLEMAS:**

1. Puede ser que no haya suficientes bits (p) para realizar la representación binaria de la parte entera (E) de números decimales. Con p bits se puede llegar a representar un número entero (sin signo) como máximo de $2^p - 1$.
2. La representación de la parte fraccionaria del número decimal (F) puede requerir un número infinito de bits. Si se elige una representación de la parte fraccionaria con q bits se produce un error entre el número decimal y el número representado en binario.
3. Si se realizan multiplicaciones de dos datos binarios representados con B = p+q bits, para representar el resultado se requieren 2*B bits, que excede el número fijo de bits para la representación.

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

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

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

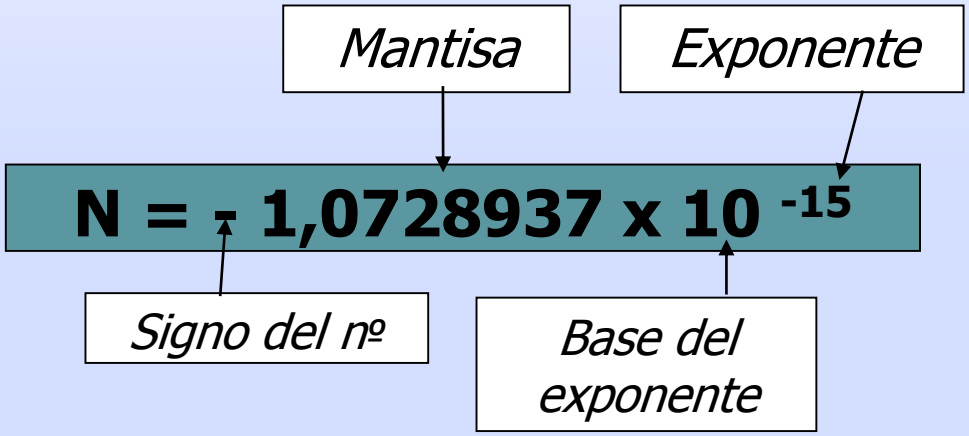
- En este caso es como si la coma decimal fuera variando o "flotando" de una parte a otra del número.

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

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

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

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



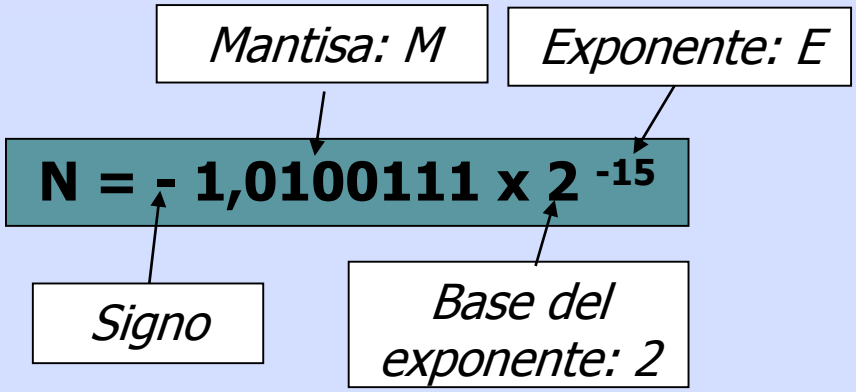
- Denominación:
  - notación exponencial,
  - notación científica
  - notación en punto o coma flotante.

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

±: signo del número, M: mantisa, B: base, E: exponente

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- 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: $\mathbf{N = \pm M \cdot 2^E}$
- Por ejemplo:

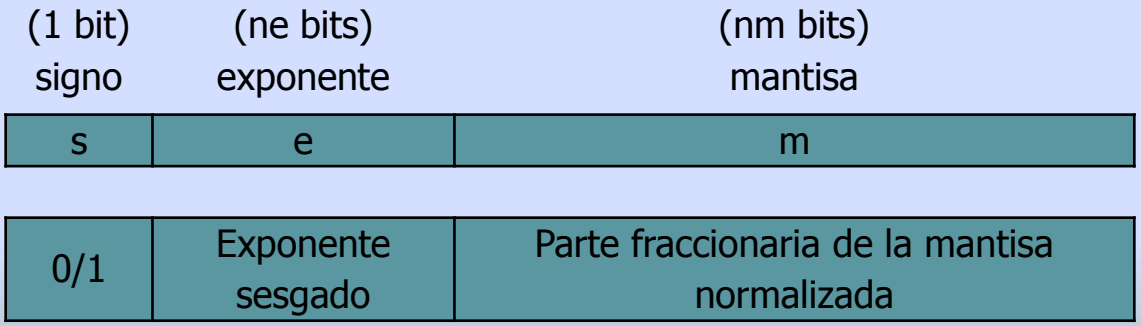


1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

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



1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- 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: $S = 2^7-1 = 127$.

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

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

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

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- **Signo:** negativo → 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	→ 4	→ 0100
0,4473046 x 16	= 7,1568736	→ 7	→ 0111
0,1568736 x 16	= 2,5099776	→ 2	→ 0010
0,5099776 x 16	= 8,1596416	→ 8	→ 1000
0,1596416 x 16	= 2,5542656	→ 2	→ 0010
0,5542656 x 16	= 8,8682496	→ 8	→ 1000
0,8682496 x 16	= 13,8919936	→ D	→ 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

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- **Ejemplo**: Obtener el valor decimal del nº cuya representación interna es: 1 0011 1110 0011 110, suponiendo n=16 y ne = 8

  n = 1 + ne + nm ⇨ 16 = 1 + 8 + nm ⇨ nm = 7

s	ne=8	nm=7
1	00111110	0011110

  Signo: s = 1 ⇨ N es negativo

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

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

  Mantisa:

  La mantisa está normalizada:

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

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

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

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

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

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

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- Patrones asociados a situaciones especiales:

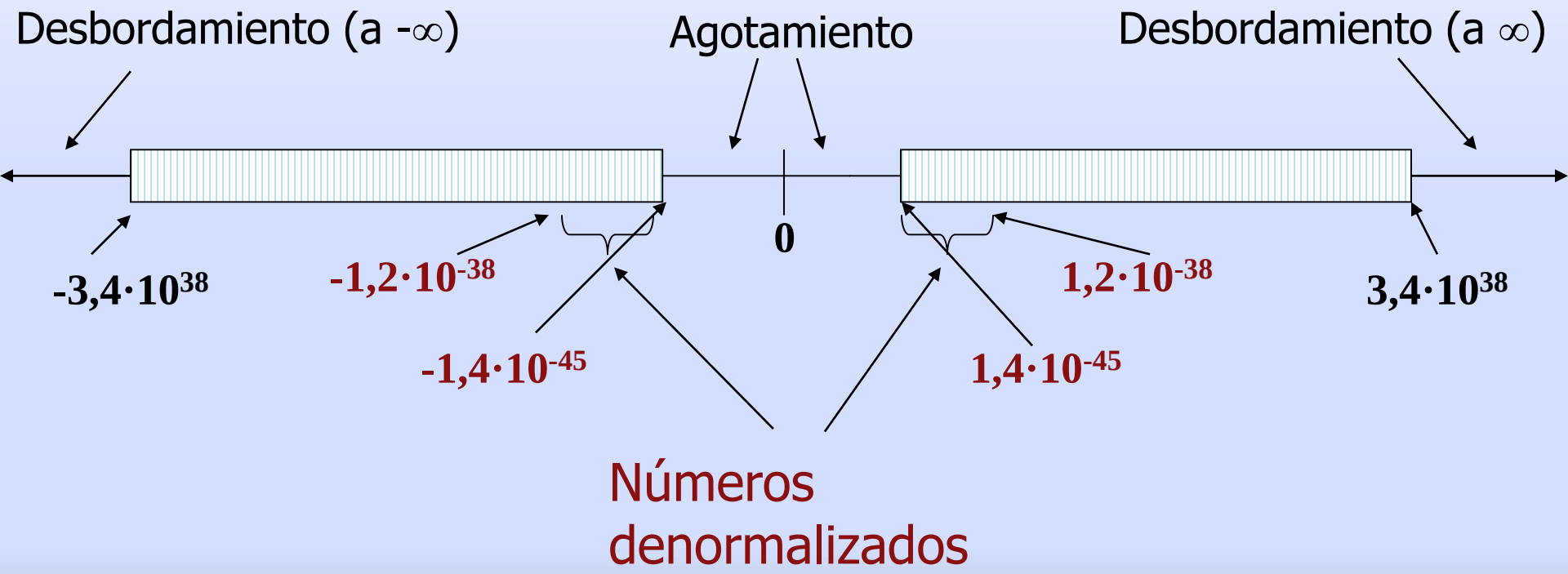
		Signo	Exponente	Mantisa
Nº denormalizado	→	0/1	0000 0000	m ≠ 0
Cero	→	0	0000 0000	000 0000 0000 0000 0000 0000
+ ∞	→	0	1111 1111	000 0000 0000 0000 0000 0000
- ∞	→	1	1111 1111	000 0000 0000 0000 0000 0000
Indeterminado (NaN)		0	1111 1111	m ≠ 0

Número denormalizado: tiene la parte entera igual a 0

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

1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- IEEE 754 simple precisión:



1.3 Representación de datos numéricos. Datos de tipo real. Representación en coma o punto flotante.

- 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.

Tema 1. Introducción

CONTENIDOS:

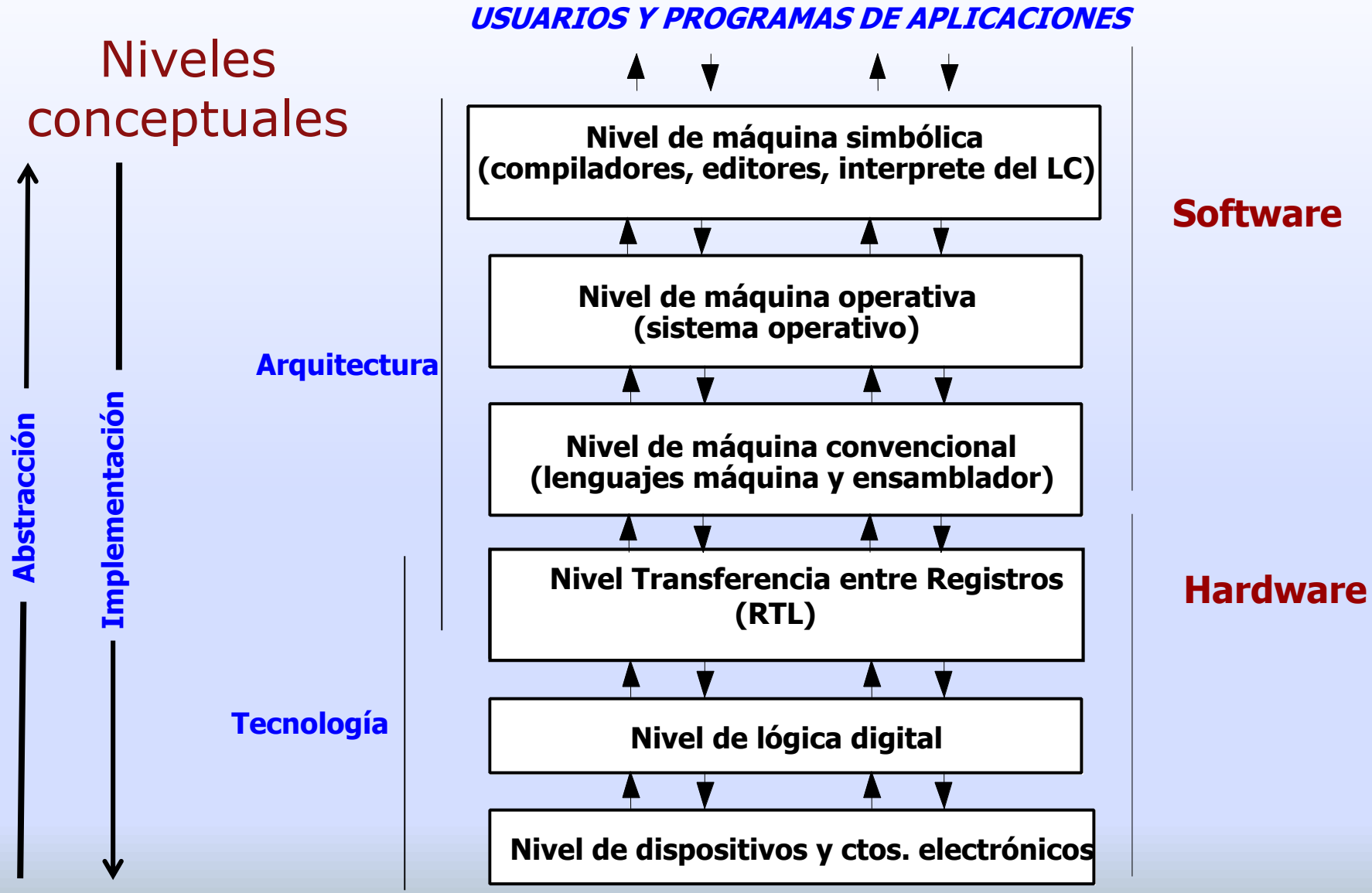
1.1. Conceptos básicos.
1.2. Estructura funcional de un computador.
1.3. Representación de datos numéricos.
1.4. Niveles conceptuales de descripción de un computador
1.5. Sistemas analógicos y digitales.

1.4. Niveles conceptuales de descripción de un computador

- El soporte físico, o hardware de un computador es la máquina en sí: el conjunto de circuitos electrónicos, cables, armarios, dispositivos electromecánicos, y otros elementos físicos que forman el computador.

- El soporte lógico, software o logical de un computador es el conjunto de programas (del sistema operativo, de utilidades, y de los usuarios) ejecutables por el computador.

Niveles conceptuales



1.4 NIVELES DE COMPLEJIDAD EN LA DESCRIPCIÓN DE UN SISTEMA DIGITAL

NIVEL	COMPORTAMIENTO	COMPONENTES ESTRUCTURALES
Nivel de Máquina convencional	Instrucciones máquina	Procesadores, controladores, memorias, ASIC
Nivel de Transferencia entre Registos	Algoritmos Diagramas de flujo Cartas ASM	registros, contadores, memorias, ALUs, MUXs, DEMUXs, etc.
Nivel de Lógica Digital	Ecuaciones booleanas Tablas de estado	Puertas lógicas y biestables
Nivel de dispositivos y Circ. Electrónicos	Ecuaciones diferenciales Diagramas corriente-tensión	N. de Dispositivos: Difusiones N+, P+, etc. N. De Circ.:Transistores, resistencias, condensadores

1.4 Niveles de complejidad en la descripción de un sistema digital.

- El análisis o diseño de un sistema digital se realiza en varios pasos de distinta complejidad o niveles de abstracción.

  - **Nivel de sistema**: identifica los grandes componentes estructurales del sistema digital.
  - **Nivel de procesador**: identifica los componentes de mayor nivel, su comportamiento y sus interconexiones. (CPU, memoria, periféricos).
  - **Nivel de registro**: estudia el comportamiento de las unidades funcionales que constituyen el sistema.
  - **Nivel lógico**: relaciona los detalles del sistema desde un punto de vista técnico (Diseño lógico → puertas lógicas).
  - **Nivel electrónico**: se construyen circuitos electrónicos que constituyen los bloques diseñados en el nivel lógico.
  - **Nivel físico**: detalles a nivel físico microscópico para implementar o fabricar el sistema.

Tema 1. Introducción

CONTENIDOS:

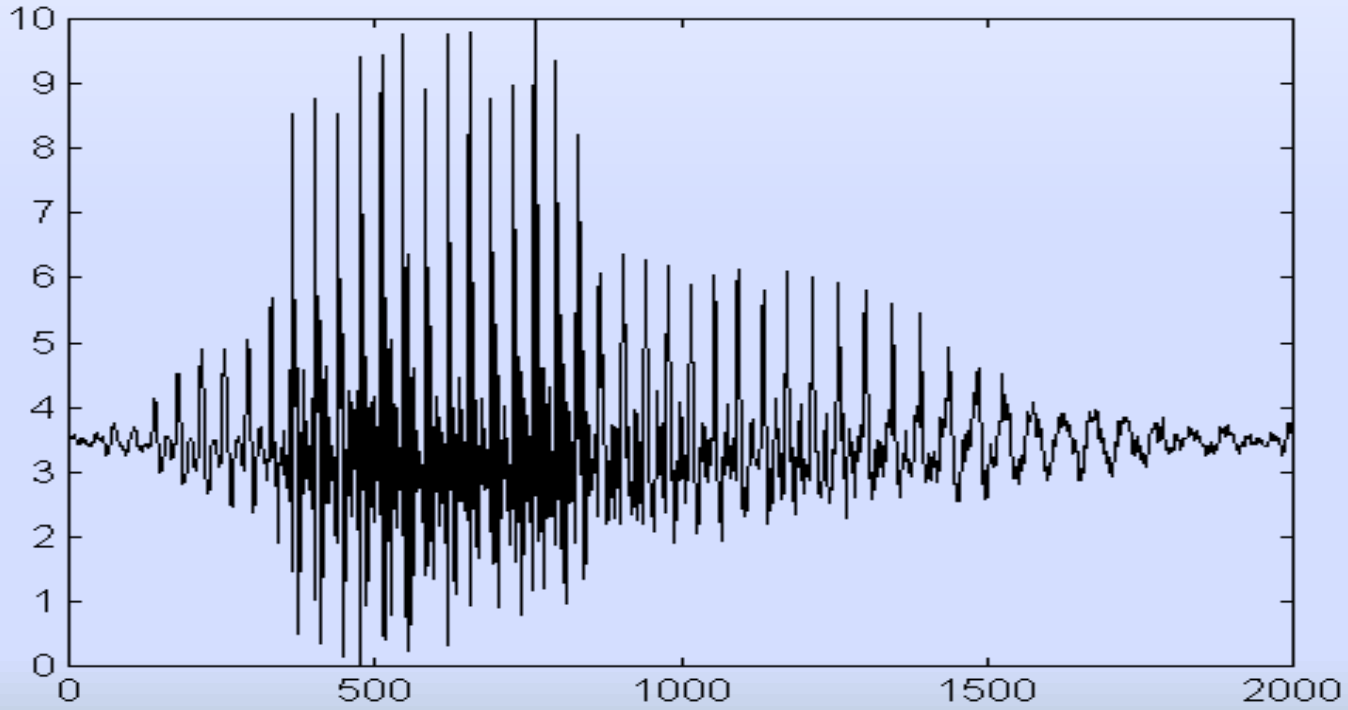
1.1. Conceptos básicos.
1.2. Estructura funcional de un computador.
1.3. Representación de datos numéricos.
1.4. Niveles conceptuales de descripción de un computador
1.5. Sistemas analógicos y digitales.

1.5. Sistemas analógicos y digitales

- Existen dos formas básicas de representar la información: analógica y digital.
- Las magnitudes físicas que pueden tomar infinitos valores y varían de forma continua se denominan variables analógicas o continuas.
- Señal analógica: señal física que se utiliza para representar una variable analógica.
- La mayor parte de las variables físicas de la naturaleza (temperatura, tensión, intensidad luminosa, posición, sonido, etc.) varían continuamente con el tiempo.
- Siempre se puede encontrar un valor entre dos valores cualesquiera.
- Un sistema analógico es aquel que procesa señales analógicas.

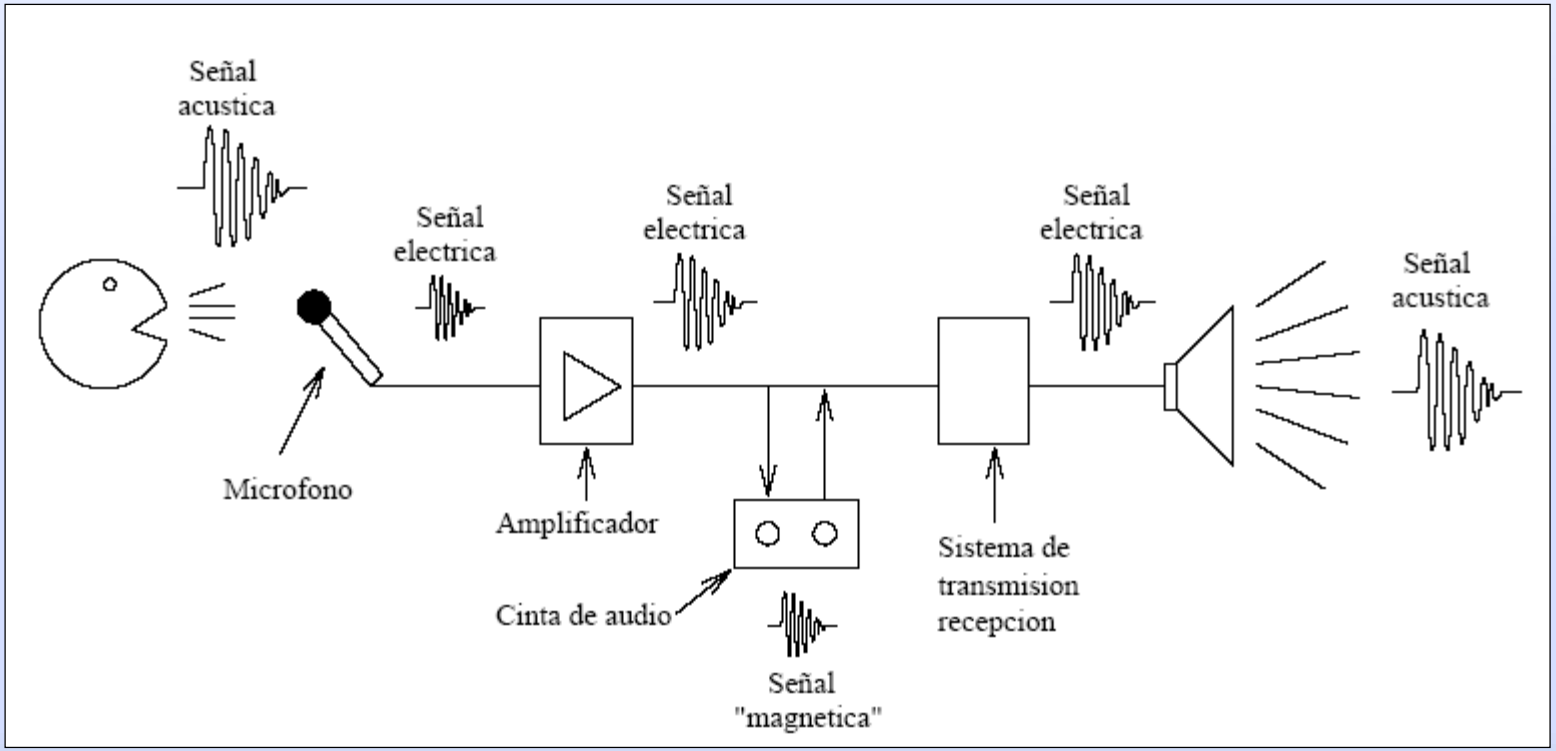
1.5 Sistemas analógicos y digitales

- Ejemplo: señal de audio que representa la palabra “mano” capturada a través de un micrófono



1.5 Sistemas analógicos y digitales

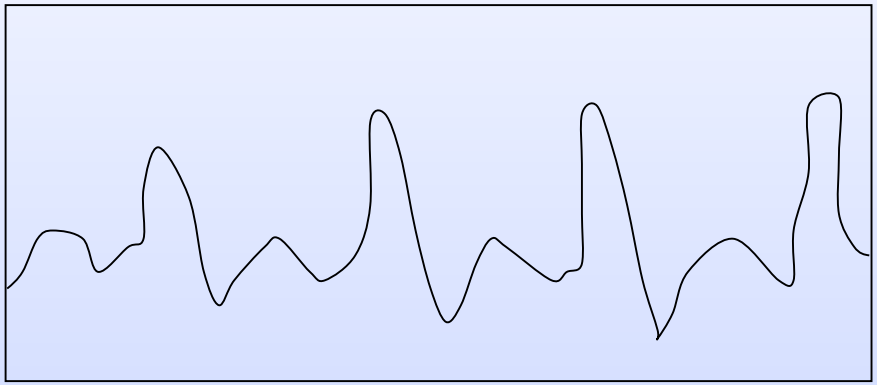
- Sistema de audio analógico:



1.5 Sistemas analógicos y digitales

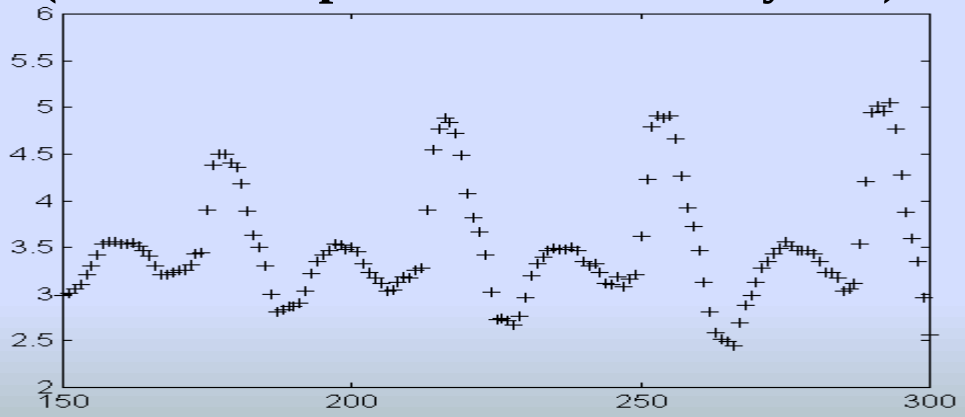
- Señal digital: señal física que se utiliza para representar una magnitud que toma valores discretos.
- Muchas señales físicas que se emplean para transmitir información son de naturaleza analógica (tensión eléctrica).
- Sin embargo, las señales analógicas se pueden cuantizar o muestrear o digitalizar, tomando un valor (una muestra) de dicha señal cada cierto tiempo (periodo de muestreo).
- Una señal digital puede caracterizarse por varios niveles, n. En este curso nos referiremos al caso particular de señales digitales binarias (n=2). En general, un nivel será 0 ó L y el otro 1 ó H.
- Un sistema digital es cualquier dispositivo destinado a generar, transmitir, procesar o almacenar señales digitales.

1.5 Sistemas analógicos y digitales



Señal muestreada a F_s= 8 KHz (T_s=0,125 ms)

(Tramo comprendido entre 150 y 300)



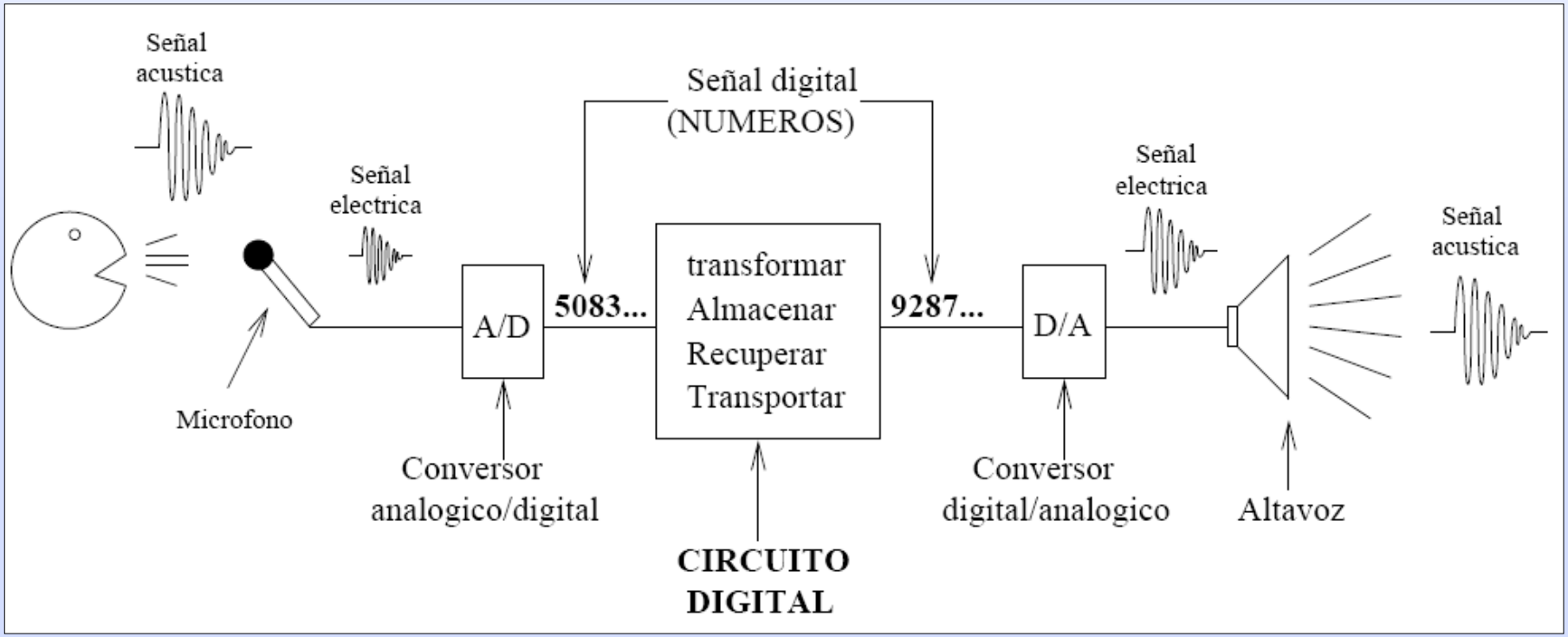
1.5 Conceptos básicos. Sistemas analógicos y digitales

Más muestras

Menos muestras

1.5 Conceptos básicos. Sistemas analógicos y digitales

- Sistema de audio digital:



1.5 Sistemas analógicos y digitales

Sistemas analógicos:

- Más sensibles a ruidos, a cambios en las condiciones (T, V, …) ↓
- Lecturas imprecisas de los valores ↓
- Contiene toda la información ↑

Sistemas digitales binarios:

- Resolución dependiente de la frecuencia de muestreo y número de bits por dato
- Compromiso velocidad/nº líneas, según transmisión paralelo o serie ↓
- Más fiabilidad y precisión en almacenamiento, procesamiento y transmisión de señales ↑
- Diseño más fácil basado en decisiones lógicas y conmutadores (Sí/No, 1/0, ON/OFF) ↑
- Metodologías de diseño y herramientas CAD altamente desarrolladas y bien conocidas ↑

CAPÍTULO 1

INTRODUCCIÓN.

TECNOLOGÍA Y ORGANIZACIÓN DE COMPUTADORES

1º Grado en Ingeniería Informática.