



1. INTRODUCCIÓN



1.1 Definiciones

Informática, ordenador y sus componentes

1.2 Hardware. Estructura de un ordenador

1.2.1 Unidades de Entrada/Salida

1.2.2 Memoria

1.2.3 Unidad Central de Proceso (UCP/CPU)

1.2.4 Representación de los datos (Bit)

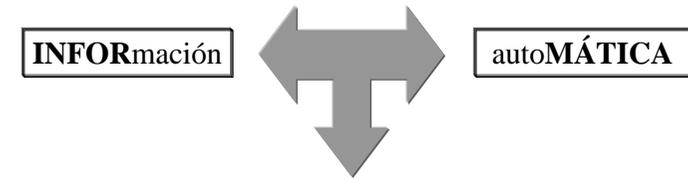
1.3 Software

1.3.1 Sistema Operativo

1.3.2 Aplicaciones Generales

1

1.1 Informática: Definición



INFORMÁTICA

Informática = ciencia encargada del tratamiento automático de la información

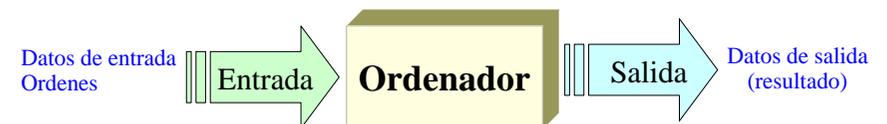
2

1.1 Otras definiciones

- ⇒ **Conjunto de conocimiento científicos y técnicos que hacen posible el tratamiento automático de la información por medio de ordenadores (RAE)**
- ⇒ **Ciencia responsable del tratamiento automático y racional de la información considerada como soporte del conocimiento de la sociedad y las comunicaciones en los campos social, económico y técnico.**
- ⇒ **Área de conocimiento que reúne todos los aspectos necesarios para el diseño y uso de los ordenadores.**

3

1.1 Ordenador



“El ordenador es una máquina electrónica digital que realiza operaciones aritmético-lógicas con los datos de entrada hasta que obtiene el resultado”

4

1.1 Ordenador

El ordenador maneja **dos tipos de información**

Ordenes: especifican lo que debe hacer el ordenador

Datos: información que utilizan (datos de entrada) o generan (datos de salida o resultados) las ordenes

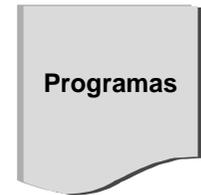
5

1.1 Componentes del ordenador: Hardware y Software

Los componentes físicos (circuitos integrados, cables, teclado, ...) de la máquina constituyen lo que se denomina el **soporte físico** o **hardware**.

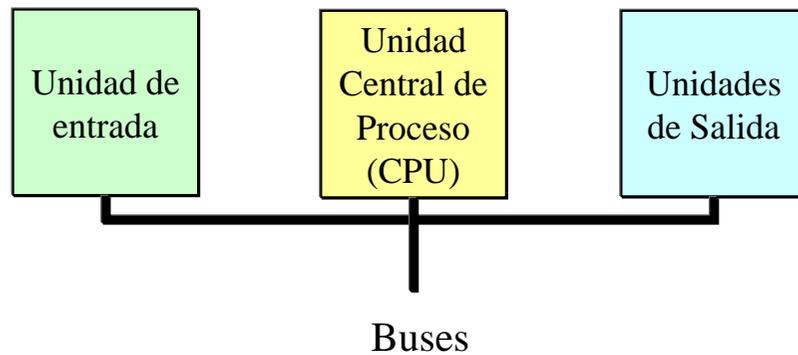


Los componentes intangibles (datos y programas) constituyen el **componente lógico** o **software**. En muchos ámbitos cuando hablamos de software hablamos tan solo de programas o aplicaciones.



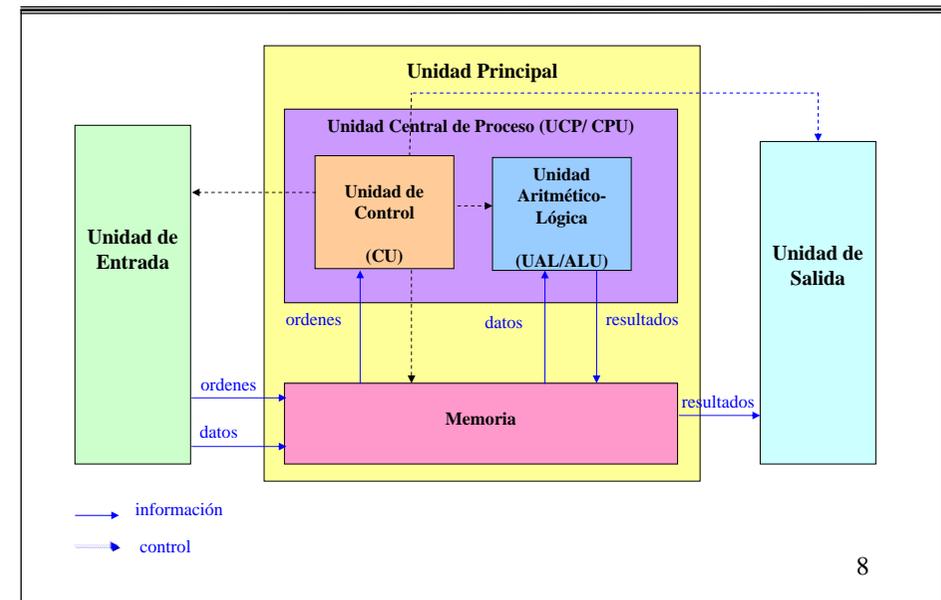
6

1.2 Hardware. Estructura de un ordenador



7

1.2 Estructura del Ordenador



8

1.2.1 UCP (CPU) Unidad de Control (UC)

- ⇒ **Dirige y controla el funcionamiento del ordenador**
- ⇒ **Coge las ordenes, las interpreta y se encarga de que se ejecuten**
- ⇒ **Características:**
 - ⇒ *Bit (8, 16, 32, 64, ...)*
 - ⇒ *Velocidad (2Ghz)*
 - ⇒ *Pentium, Athlon, Duron, PowerPC,...*

9

1.2.3.1 UCP (Hz)

HERTZIO o **hercio**. La Unidad de Control contiene un reloj interno (generador de impulsos) que sincroniza todas las operaciones elementales del ordenador.

El periodo de esta señal se denomina **tiempo de ciclo**, y su frecuencia puede darse en millones de ciclos por segundo denominados Mega Hertzios (**MHz**).

80386: 25Mhz. (Megahertzios)

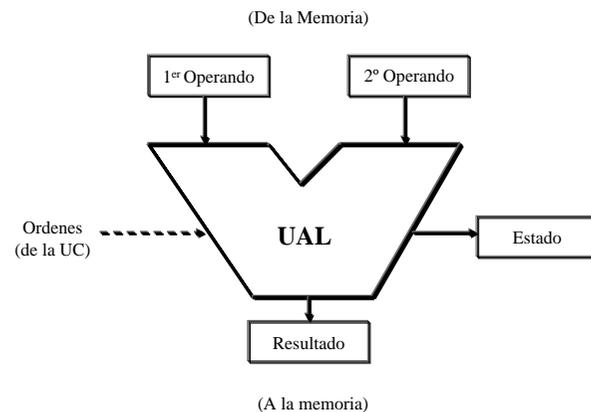
88486: 55, 66 Mhz

Pentium, AMD: 100,133 MHz, ..., 2GHz (Gigahertzios)

10

1.2.1 UCP (CPU) Unidad Aritmético-Lógica (UAL/ALU)

Realiza operaciones aritméticas (suma, resta, multiplicación, división) y lógicas (comparación,...)



11

1.2.2 Unidades de Entrada/Salida (E/S)

- ⇒ **Puente entre el ordenador y el exterior**
- ⇒ **Salida**
 - ⇒ *pantalla, impresora, altavoces, disco,*
- ⇒ **Entrada**
 - ⇒ *teclado, ratón, scanner, webcam, disco,*

12

1.2.3 Memoria

- ⇒ Almacén del ordenador
 - ⇒ **Instrucciones** que constituyen los programas
 - ⇒ **Datos**
 - ⇒ Resultados parciales
- ⇒ Clasificación
 - ⇒ Memoria **principal** o memoria interna
 - ⇒ Memoria **auxiliar** o memoria de almacenamiento masivo

13

1.2.2 Memoria principal

- ⇒ Gran Velocidad
- ⇒ Poca Capacidad
 - ⇒ **RAM**
 - ⇒ Almacena los datos e instrucciones que va a utilizar el procesador
 - ⇒ **Volátil**
 - ⇒ **ROM**
 - ⇒ Contiene los programas y rutinas de E/S que necesita el ordenador para arrancar
 - ⇒ No volátil. No se puede modificar
 - ⇒ **Caché**
 - ⇒ Contiene una copia de los datos/instrucciones más recientes

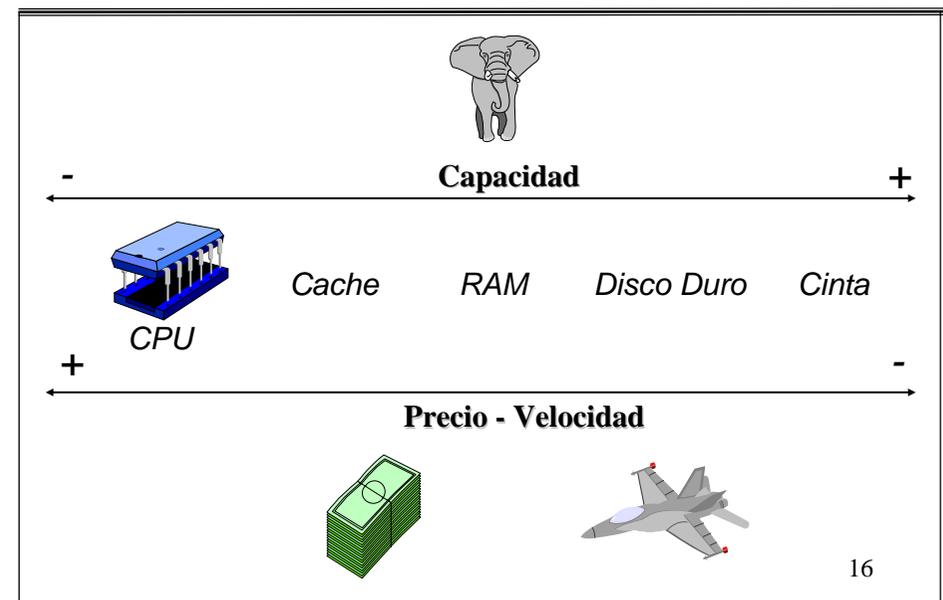
14

1.2.3 Memoria auxiliar

- ⇒ Velocidad de acceso baja
- ⇒ Gran Capacidad
- ⇒ Almacena programas y datos (Ej. Sistema operativo, aplicaciones)
- ⇒ Almacena la información cuando se apaga el ordenador
- ⇒ **Memorias magnéticas**
 - ⇒ Disco duro (1-100 GB)
 - ⇒ Diskettes (1,44 MB)
 - ⇒ ZIP, JAZ, cintas, ...
- ⇒ **Memorias ópticas**
 - ⇒ CD-ROM (650 MB)
 - ⇒ CD-R, CD-RW (650 MB)
 - ⇒ DVD, ...

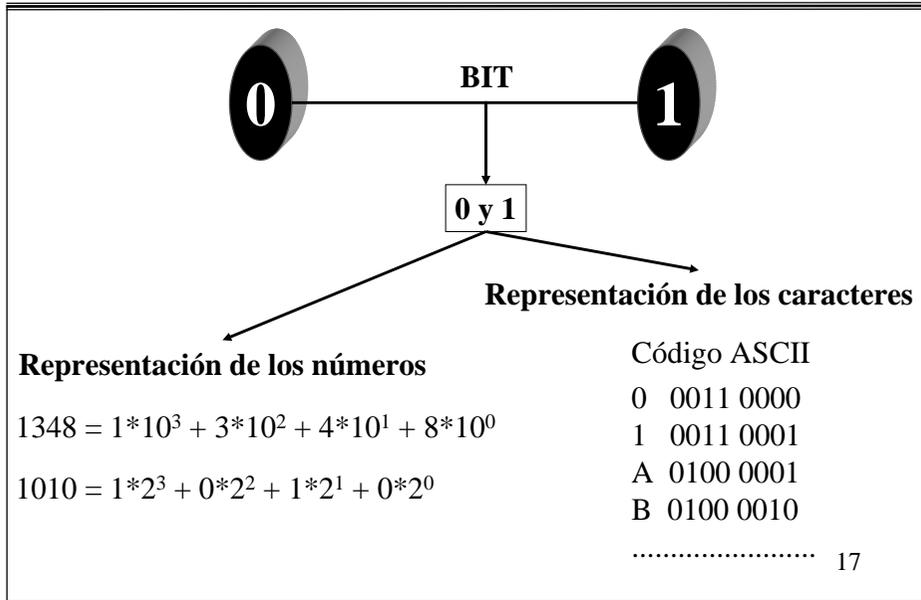
15

1.2.3 Memoria



16

1.2.3.1 Representación de la Información



1.2.3.1 Unidad de Información (Memoria)

- Los **ordenadores** utilizan un sistema de numeración **binario**. Esto es debido a que los componentes básicos del ordenador pueden indicar **dos estados** o condiciones posibles.

- Al igual que el sistema decimal utiliza diez dígitos del 0 al 9, el sistema binario utiliza solamente **dos dígitos** para representar toda su información básica; **cero** (0) y **uno** (1). Estos dígitos binarios se denominan **bits**.

- Se requiere una **codificación** que haga corresponder los caracteres utilizados por el **usuario** ('A', 'B', 'C', ..., 'a', 'b', 'c', ..., '1', '2', '3', ..., '/', '*', '(', ...) y los utilizados por la **máquina** (combinaciones de **0s** y **1s**), es decir, se necesita representar los elementos de un conjunto (usuario) con los elementos de otro conjunto (máquina).

1.2.3.1 Unidad de Información (Memoria)

- Hay diferentes estándares de codificación de caracteres:

- **ASCII**: *American Standard Code for Information Interchange*. La tabla básica de caracteres ASCII esta compuesta por 128 caracteres incluyendo símbolos y caracteres de control. En ASCII cada carácter está representado por **7 bits** (unos ó ceros). Existe una versión extendida de 256 caracteres, representada por **8 bits**.

- **EBCDIC**: *Extended Binary Coded Decimal Interchange Code*. Codificación de caracteres de **8 bits** utilizada en ordenadores IBM.

- **Unicode**: es una norma más reciente de **16 bits** cuyo objetivo es asignar a cada carácter de cada lenguaje una codificación única, a diferencia de la mayor parte de los juegos ISO como el ISO-8859-1, que sólo definen los necesarios para un idioma o zona geográfica.

1.2.3.1 Unidad de Información (Memoria)

- Las **operaciones aritméticas** con datos numéricos se suelen realizar mediante una **representación basada en el sistema de numeración en binario**.

- Al ser **una representación numérica posicional** es muy apta para la realización de operaciones aritméticas.

Ejemplo con 2 bits con 3 bits con 4 bits:

00	000	100	0000	0100	1000	1100			
01	001	101	0001	0101	1001	1101			
10	2 ²	010	110	2 ³	0010	0110	1010	1110	2 ⁴
11		011	111		0011	0111	1011	1111	

1.2.3.1 Sistemas de Numeración y Binario

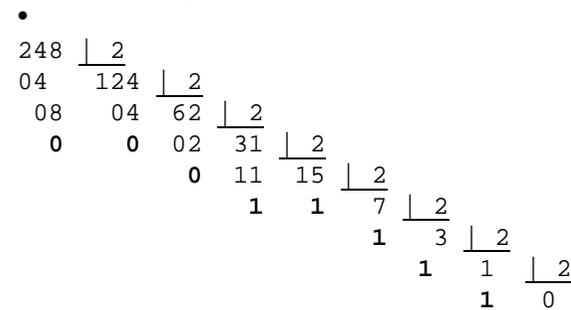
- Los **humanos** trabajamos normalmente en sistema **decimal** y el **ordenador** en **binario**.
- El sistema de numeración decimal (**base 10**), utiliza diez símbolos (**0, 1, 2, 3, ..., 9**) o dígitos.
- El sistema de numeración binaria (**base 2**) utiliza solamente dos símbolos (**0 y 1**).
- La posición de uno de estos símbolos en un número indica la potencia que se asigna a este símbolo "*Sistema posicional*".
- Sistema de numeración decimal (base 10):

$$837 = 8 * 10^2 + 3 * 10^1 + 7 * 10^0$$
- Sistema de numeración binario (base 2):

$$1101000101_2 = 1 * 2^9 + 1 * 2^8 + 0 * 2^7 + 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 837_{10}$$

1.2.3.1 Conversión Decimal a Binario

- Regla: se divide el número decimal por la base (en nuestro caso 2) hasta que el cociente se anule y se obtienen los dígitos en el resto, del de menor peso al de mayor.



←←←←← El resultado se lee en este sentido ←←←←←

$248_{10} \rightarrow 11111000_2$
 $1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 0 * 2^0 = 248$

1.2.3.1 Sistema de numeración Hexadecimal

- A menudo queremos introducir los datos tal y como la entiende el ordenador, es decir, como series de unos y ceros. Como esto puede resultar tedioso y de fácil confusión es más práctico utilizar el sistema **hexadecimal**, es decir, en **base 16**.
- El sistema decimal utiliza diez dígitos del 0 al 9. El sistema hexadecimal requiere seis dígitos más, por lo que a partir del 9 utilizaremos letras de la A (10_{decimal}) a la F (15_{decimal}).
- Existe una correspondencia directa entre cuatro dígitos binarios y uno hexadecimal:

$$h = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$
- Para representar cualquier información almacenada en un **octeto** (byte), se podrá realizar con **dos dígitos hexadecimales**

1.2.3.1 Sistema de numeración Hexadecimal

Binario	Octal	Decimal	Hexadecimal
0000	00	00	0
0001	01	01	1
0010	02	02	2
0011	03	03	3
0100	04	04	4
0101	05	05	5
0110	06	06	6
0111	07	07	7
1000	10	08	8
1001	11	09	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10

1.2.3.1 Sistema de numeración Hexadecimal

El proceso de conversión de hexadecimal a decimal será similar a los tratados con anterioridad:

$$\begin{array}{r|l} 248 & | \ 16 \\ 088 & 15 \ | \ 16 \\ 8 & 15 \ 0 \\ & =\mathbf{F} \end{array}$$

$$248_{10} = F8_{16}$$

$$F8_{16} = F \cdot 16^1 + 8 \cdot 16^0 = 15 \cdot 16 + 8 \cdot 1 = 128 + 8 = 248_{10}$$

1.2.3.1 Sistema de Binario (Operaciones aritméticas)

<u>SUMA</u>	<u>RESTA</u>
0+0=0	0-0=0
0+1=1	0-1=1 y debo 1
1+0=1	1-0=1
1+1=0 y llevo 1	1-1=0

<u>MULTIPLICACION</u>	<u>DIVISION</u>
0.0=0	0:0=-
0.1=0	0:1=0
1.0=0	1:0=oo
1.1=1	1:1=1

1.2.3.1 Sistema de Binario (Operaciones Lógicas)

Otro tipo de operaciones son las booleanas u operaciones lógicas : AND, OR y NOT. Estas operaciones se rigen según las siguientes tablas:

<u>SUMA BOOLEANA (OR)</u>	<u>PRODUCTO BOOLEANO (AND)</u>
0 OR 0=0	0 AND 0=0
0 OR 1=1	0 AND 1=0
1 OR 0=1	1 AND 0=0
1 OR 1=1	1 AND 1=1

INVERSION BOOLEANA (NOT)

NOT 0 = 1
NOT 1 = 0

1.2.3.1 Representación Interna de la Información

- Los ordenadores procesan dos tipos de información: las **instrucciones** que forman parte del programa y los **datos** que manejarán dichas instrucciones.
- En la **Unidad Central de Proceso** la información se transmite y **procesa** en unidades denominadas **palabras**. La longitud de la palabra depende de la **estructura interna de cada modelo de ordenador**, pudiendo ser las mas normales de **8, 16, 32, 64 bits**.

1.2.3.1 Representación Interna de la Información

• Para leer o escribir un dato o ejecutar una instrucción del programa almacenado en la **memoria principal** se da la **dirección de la palabra** donde se quiere leer o escribir, por tanto para obtener un **buen aprovechamiento** de la **memoria**, la **longitud de la palabra**, debe ser un **múltiplo del número de bits utilizados para representar un carácter**.

• Así en los ordenadores de 8, 16, 32, 64 bits se utilizan códigos de E/S de 8 bits (EBCDIC o ASCII) y tanto **las longitudes de las instrucciones como la longitud de los datos serán múltiplos de 8**.

29

1.2.3.1 Tipos de Datos (Representación Interna)

La **representación interna** de datos **depende del tipo de dato y del lenguaje de programación**. Los tipos de datos más significativos pueden ser:

- **Texto o carácter**
- **Lógico**
- **Entero**
- **Real de precisión simple o doble**

30

1.2.3.1 Representación Interna de los Datos Tipo Texto o Carácter

• Los **datos tipo carácter o texto** se suelen denominar **alfabéticos** si están compuestos solamente por letras y **alfanuméricos** si están compuestos por letras, números y/o caracteres especiales.

• Estos tipos de datos **se almacenan utilizando la codificación de E/S correspondiente**. En el caso del **ASCII o EBCDIC**, un carácter por byte, sin realizarse internamente ninguna transformación.

• Ejemplo: MICRO en código ASCII será **4d-49-43-52-4f** en hexadecimal ó **77-73-67-82-79** en decimal.

0100 1101 0100 1001 0100 0011 0101 0010 0100 1111

31

1.2.3.1 Representación Interna de los Datos Lógico

• Representan un valor del álgebra de Boole binaria:

- 0: falso
- 1: verdadero

• Dependiendo del lenguaje de programación también se considera verdadero a cualquier valor distinto de 0

• La representación interna de este tipo de dato es muy variada siendo quizá la más común la de completar todo el espacio de la palabra a ceros o a unos dependiendo del caso, o el de identificar el dato solamente con el bit extremo derecho 0 ó 1.

32

1.2.3.1 Representación Interna de los Datos Entero

- La representación en **binario puro** consiste en el almacenamiento de los números, descritos mediante el sistema de numeración decimal, en el sistema de numeración binario.

Ejemplo:

En una palabra de (16 bits) almacenar el número entero 15.

$$\begin{array}{r}
 15 \quad | \quad 2 \\
 1 \quad 7 \quad | \quad 2 \\
 \quad 1 \quad 3 \quad | \quad 2 \\
 \quad \quad 1 \quad 1 \quad | \quad 2 \\
 \quad \quad \quad 1 \quad 0
 \end{array}$$

| 0000 | 0000 | 0000 | 1111 |

33

1.2.3.1 Representación Interna de los Datos Entero (Binario con Signo)

- El signo se puede representar en el bit extremo izquierdo de la palabra mediante:

	0 ... +
	1 ... -
+15	0000 0000 0000 1111
-15	1000 0000 0000 1111

- De esta forma, el mayor número almacenable en 2 bytes sería representado por 15 unos con lo que se podría almacenar un número:

$$2^{15} = 32768 \text{ números (0 a 32767)}$$

donde el bit 16 es el signo (-32767, ..., -1, -0, 0, 1, ..., 32767)

34

1.2.3.1 Representación Interna de los Datos Entero (Complemento a 2)

- Para **representar** un **número negativo** se puede utilizar el **complemento a dos**.

- El **complemento a dos** de un número **A** compuesto por **n** dígitos (es **importante** el número de dígitos) se define como:

- $C_2(A) = 2^n - A$

- El equivalente decimal sería el complemento a 10:

- $C_{10}(A) = 10^n - A$

- Así, para 5 dígitos decimales, el complemento a 10 del 345 sería:

- $100000 - 345 = 99655$

35

1.2.3.1 Representación Interna de los Datos Entero (con signo)

- Para calcular el complemento a 2 de un número basta con calcular el complemento a 1 (obtener lo que le falta a cada dígito para llegar a 1) y sumarle 1.

- Ejemplo: complemento a 2 con 6 bits del 9_{10} :

- 9_{10} en binario con 6 bits: 001001_2

- Complemento a 1 de 001001_2 : 110110

- Complemento a 2 (sumando 1): $110110 + 1 = 110111$

- Ejemplo: complemento a 10 del 8342 con 5 cifras decimales:

1. Complemento a 9 con 5 cifras: $99999 - 8342 = 91657$

2. Sumando 1: $91657 + 1 = 91658$

3. Comprobamos: $10^5 - 8342 = 91658$

36

1.2.3.3 Representación de la información

⇒ La ventaja de la utilización del complemento a dos de los números negativos radica en que, al igual que **restar** es *sumar al minuendo el opuesto del sustraendo*, podemos decir que **restar es sumar al minuendo el complemento a dos del sustraendo**, descartando las posibles llevadas.

⇒ Ejemplo: 65 (0100 0001) – 23 (0001 0111) con 8 bits

⇒ 0100 0001 – 0001 0111 =

⇒ 0100 0001 + 1110 1000 + 1 =

0100 0001

+ 1110 1001

10010 1010 = 32 + 8 + 2 = 42 (descartando la llevada)

1.2.3.1 Representación de Enteros

Enteros con complemento a 2				
Binario	Decimal	Compl. a 1	Compl. a 2	Equiv. decimal
0000	0	1111	0000	0
0001	1	1110	1111	-1
0010	2	1101	1110	-2
0011	3	1100	1101	-3
0100	4	1011	1100	-4
0101	5	1010	1011	-5
0110	6	1001	1010	-6
0111	7	1000	1001	-7
1000	-8	0111	1000	-8
1001	-7	0110	0111	7
1010	-6	0101	0110	6
1011	-5	0100	0101	5
1100	-4	0011	0100	4
1101	-3	0010	0011	3
1110	-2	0001	0010	2
1111	-1	0000	0001	1

1.2.3.1 Unidad de Información (Memoria)

Bit. Unidad básica de información (un hueco de 0 o 1)

• **1 Byte = 8 BITS**

• **1 Kilobyte = 2¹⁰ bytes = 1024 bytes**

• **1 Megabyte = 2²⁰ bytes = 1024 Kbytes**

• **1 Gigabyte = 2³⁰ bytes = 1024 Mbytes**

• **1 Terabyte = 2⁴⁰ bytes = 1024 Gbytes**

• Exabyte = 2⁵⁰ bytes = 1024 Tbytes

• Petabyte, Zettabyte, Yottabyte, ...

1.3 Software

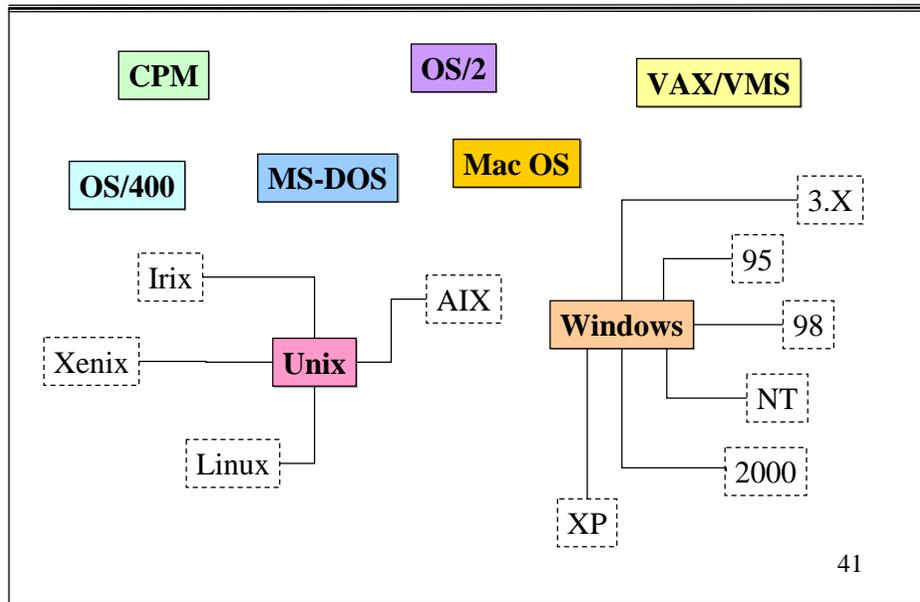
⇒ **Categorías de Software**

⇒ **Sistemas Operativos:** MS-DOS, Unix, Windows, Linux, VMS, MacOS, ...

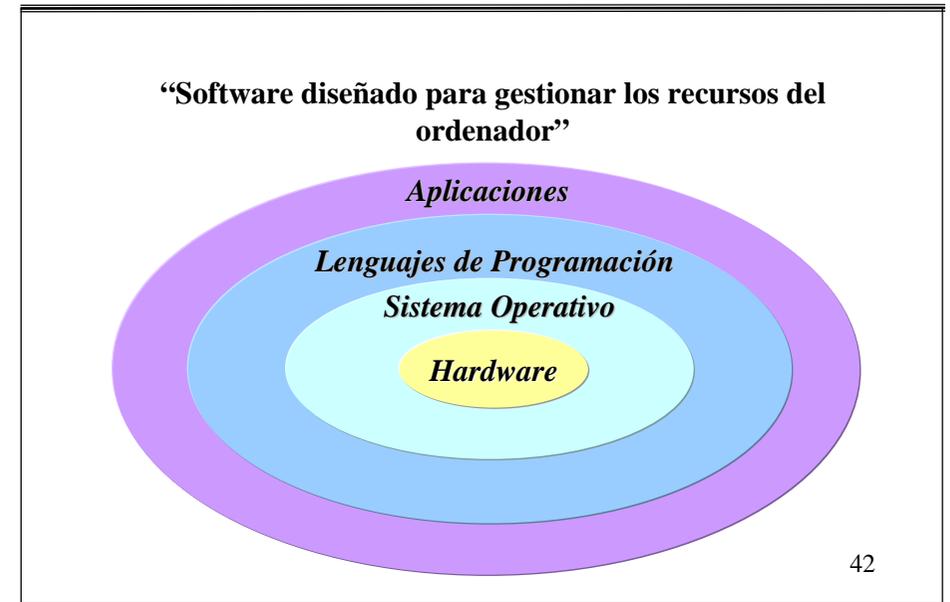
⇒ **Aplicaciones:** Procesadores de texto, presentaciones, hojas de cálculo, navegadores, sistemas de gestión de bases de datos, reproductores de música, juegos, ...

⇒ **Entornos de Programación:** Common Lisp, Turbo Pascal, Borland C++, JDK, Visual Studio (VB, C++, C#, Java, ...),

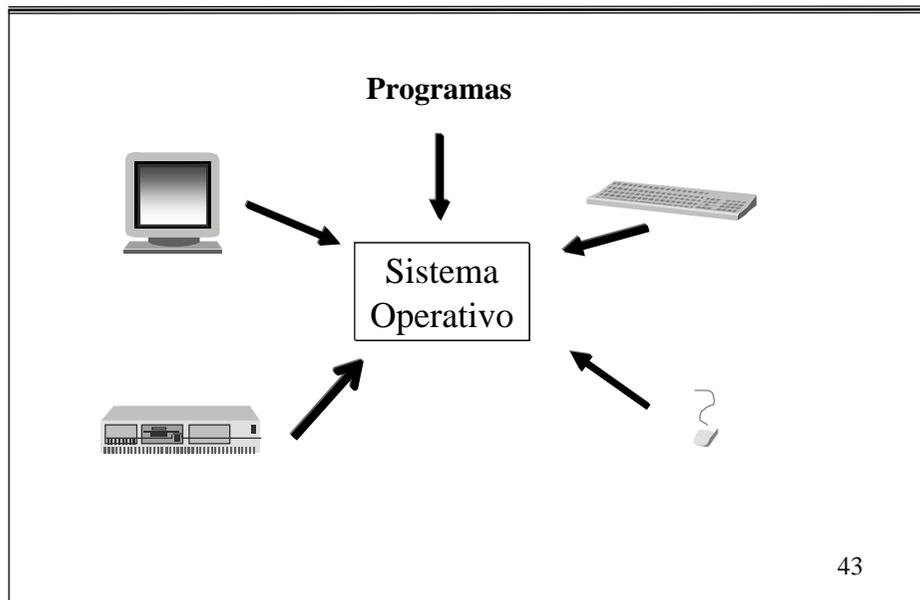
Sistemas Operativos más extendidos



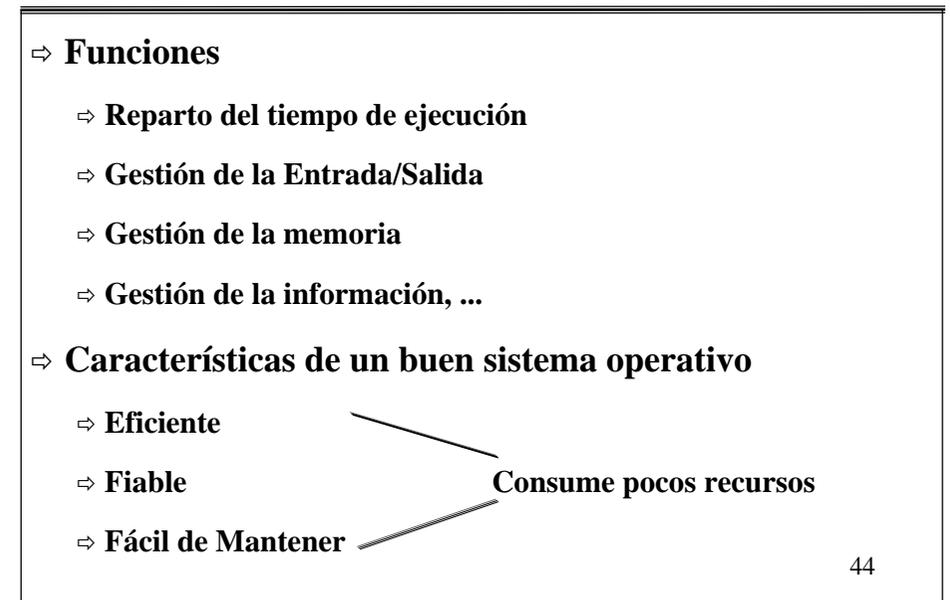
1.3.1 Sistema Operativo



Sistema Operativo



Sistema Operativo



Clasificación de los sistemas operativos

Monousuario (1) / Multiusuario (2)

1. Todos los recursos y datos están en manos del usuario que trabaja en el ordenador
2. Los datos y recursos son compartidos por los distintos usuarios que utilizan el ordenador

Monotarea (1) / Multitarea (2)

1. El procesador está dedicado en exclusiva a una única tarea
2. Realiza más de una tarea a la vez. (Reparte el tiempo de CPU entre distintas tareas)