

Tema 1. Informática Básica

Introducción

1.1 Informática Básica

1.2 Codificación de la Información

1.3 El ordenador. Fundamentos estructurales y de funcionamiento

Jesús María Aransay Azofra

Sistemas Informáticos

Universidad de La Rioja 2011/2012

Introducción

Objetivos del tema:

- Definir las nociones de Informática y sistema informático (Sección 1.1)
 - Comprender algunos de los fundamentos de la codificación digital de la información (Sección 1.2)
 - Estudiar algunas de las partes fundamentales de un sistema informático (Sección 1.3)
-

1.1 Informática básica

Informática: conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento *automático* de la *información* por medio de ordenadores

Sistema Informático: hardware + software + “factor humano”

1.2 Codificación de la información

1.2.1 Señal digital

1.2.2 Codificación de caracteres

1.2.1 Señal digital

- ❑ Basada en almacenar un número finito de estados
- ❑ Bit: binary digit. Almacena 2 estados, “0” ó “1”
- ❑ Octeto o byte: 8 bits, $2^8 = 256$ estados

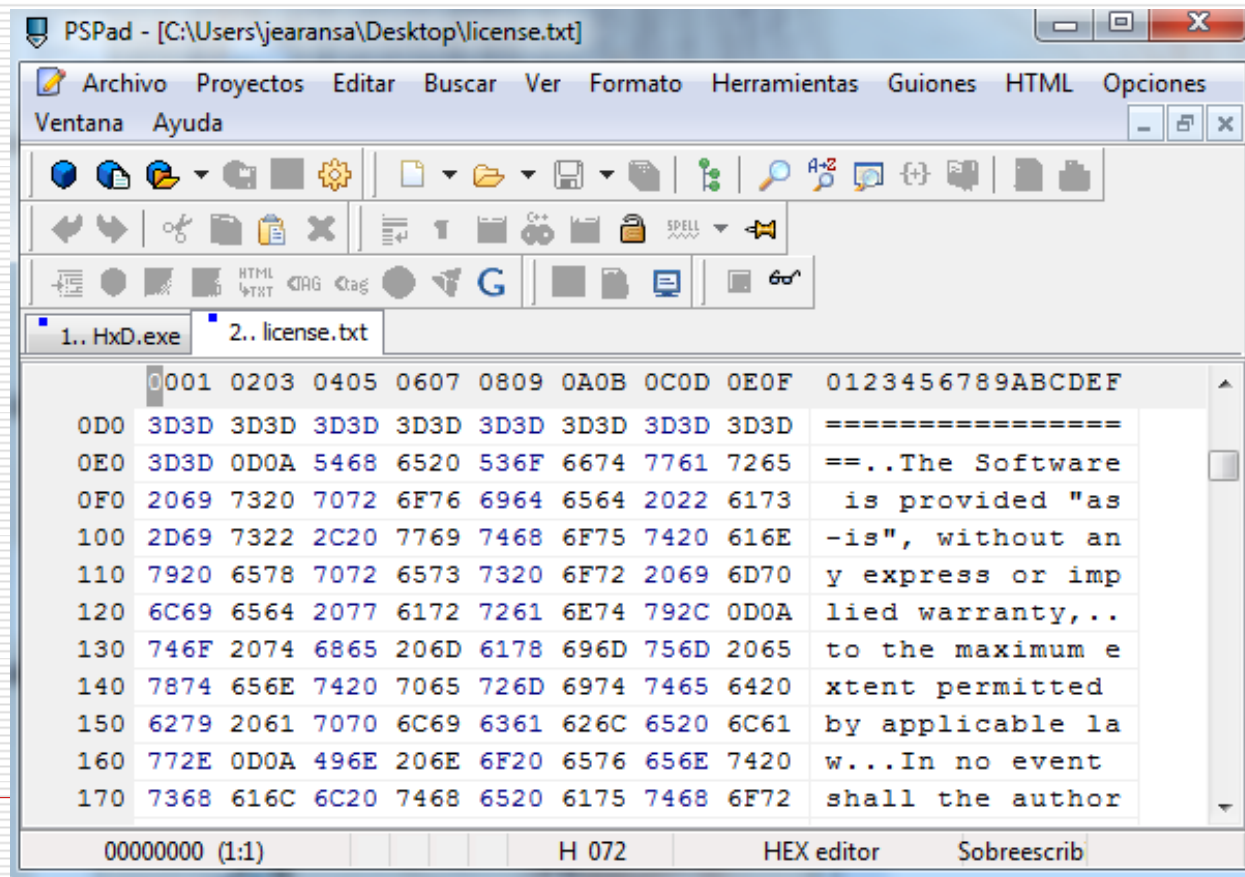
Generalmente, los sistemas informáticos siempre trabajarán como mínimo con bytes, ya que los bits no permiten almacenar cantidades de información significativas

1.2.1 Señal digital

- ❑ Lo que signifique la información contenida en un bit (o byte) depende del tipo de archivo en el que dicho bit o byte se encuentre (puede representar un color, en un archivo de imagen, o un carácter, en un fichero de “texto plano”).
 - ❑ Los ficheros de “texto plano” permiten almacenar información (de tipo textual) que no contiene mucho procesamiento o información relativa a su formato (tamaños o tipos de letra, márgenes...). Se usan ampliamente para almacenar código de programas, como ficheros de configuración, para almacenar datos...
 - ❑ En un fichero de texto plano, es crucial conocer la “tabla de codificación” con respecto a la cual el mismo ha sido guardado (y con la cual debe ser abierto)
-

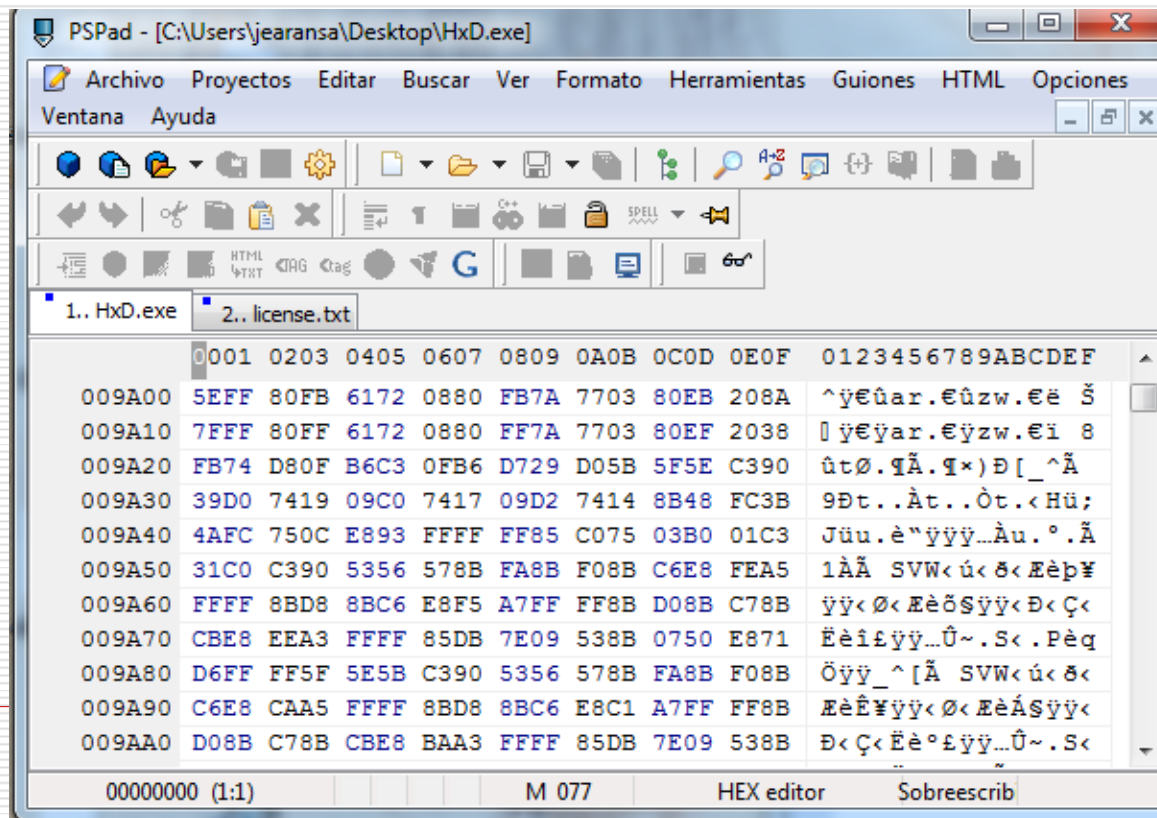
1.2.2 Codificación de caracteres

□ Archivos de texto plano (sin formato)



1.2.2 Codificación de caracteres

- ❑ Archivos binarios (también representados en formato binario)



Archivos de texto

Cada carácter del fichero es almacenado en memoria por una secuencia de “0”s y “1”s (por lo general, de uno o varios octetos).

¿Cómo es el proceso de almacenamiento de un carácter en un fichero de texto plano en memoria?

1. Usuario escribe y guarda: “A”
2. Ordenador comprueba: ¿cuál es la tabla de caracteres activa? ¿Y su codificación de caracteres respectiva?
3. Ordenador calcula: ¿cuál es la codificación del carácter “A” en esa regla de codificación?
4. Ordenador guarda en memoria la codificación (los bytes) del carácter “A” con respecto a la tabla de caracteres activa

Y viceversa...

¿Cómo es el proceso de lectura de un carácter desde memoria?

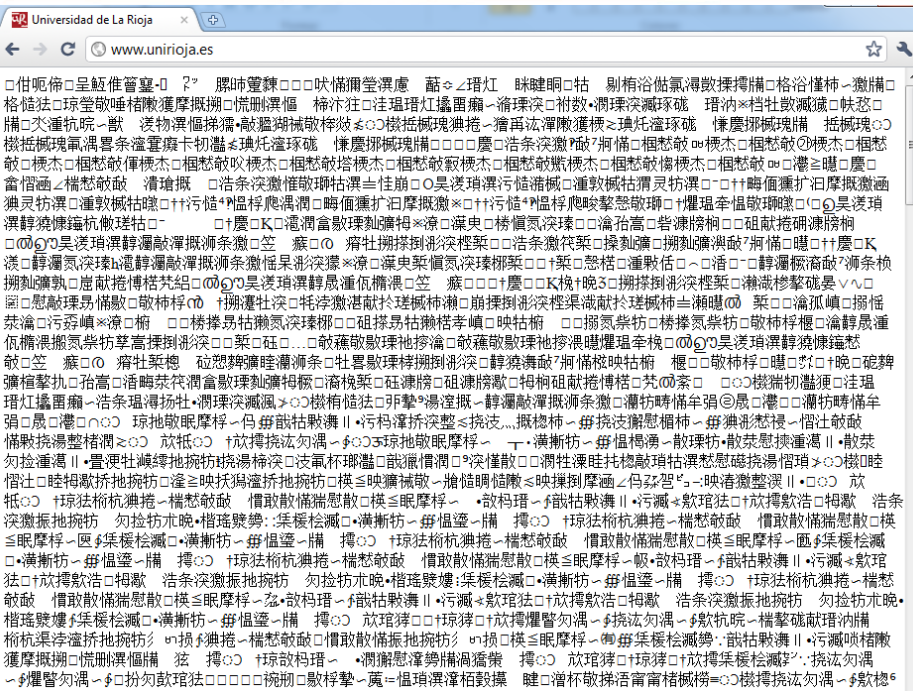
1. Abrimos un fichero de texto plano con un programa (bloc de notas, PsPad, Notepad++...)
 2. El ordenador comprueba la tabla de caracteres que usa este programa por defecto, o la infiere del archivo
 3. El ordenador lee la primera secuencia de bytes (no de bits)
 4. El ordenador comprueba a qué carácter corresponde esa secuencia en la codificación de esa tabla de caracteres, y muestra ese carácter
-

¿Cuántas cosas pueden ir mal en el anterior proceso?

- A nivel teórico unas cuantas, en la práctica generalmente una: que la tabla de caracteres que usamos al guardar un archivo NO sea la misma que la que usamos al abrirlo (porque hemos cambiado de programa, de ordenador...)
 - Que la regla (o algoritmo) de codificación haya cambiado (UTF-16 Big Endian vs. UTF-16 Little Endian)
-



www.unirioja.es abierta con la misma codificación con que se guardó (ISO-8859-1)



www.unirioja.es abierta con otra codificación distinta de la que se guardó (UTF-16LE)

Algunas definiciones (no estándar):

Repertorio de caracteres: caracteres que una codificación pretende almacenar

Ejemplo: {a, e, i, o, u, A, E, I, O, U, á, é, í, ó, ú}

Tabla de caracteres: una asignación entre los caracteres de un repertorio y números enteros positivos

Ejemplo: {a=1, e=2, i=3, o=4, u=5, A=6, E=7, I=8, O=9, U=10, á=11, é=12, í=13, ó=14, ú=15}

Codificación de caracteres: algoritmo que a cada número de la tabla de caracteres le asigna su representación en binario (y por tanto la que se almacena en memoria)

Ejemplo: {1 = 00000001, 2 = 00000010...}

Notas

Siguiendo con el ejemplo anterior, ["a" = 1 = 00000001], lo podemos leer como 'El carácter "a" cuya posición en la tabla de caracteres es 1 se codifica (y almacena) en binario como 00000001'

La posición en la tabla de codificación y la codificación en binario **no tienen por qué coincidir** (por ejemplo, la "O" en la tabla ASCII ocupa la posición 79, y su codificación en binario es "11001111" = 207). De ahí la importancia de conocer no sólo la tabla de caracteres, sino también la codificación o regla correspondiente

Notas

Cómo convertir a binario o hexadecimal un número entero:

Base 2. Símbolos usados: {0,1}

$$135 = 67 * 2 + 1$$

$$67 = 33 * 2 + 1$$

$$33 = 16 * 2 + 1$$

$$16 = 8 * 2 + 0$$

$$8 = 4 * 2 + 0$$

$$4 = 2 * 2 + 0$$

$$2 = 1 * 2 + 0$$

$$1 = 0 * 2 + 1$$

135 = 10000111 (cogemos los restos de dividir por 2 en orden inverso)

Base 16. Símbolos usados: {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

$$143 = 8 * 16 + (15 = F)$$

$$8 = 0 * 16 + 8$$

143 = 8F (cogemos los restos de dividir por 16 en orden inverso)

Algunos sistemas de codificación

ASCII (American Standard Code for Information Interexchange)

- ☐ Origen occidental
 - ☐ Tiene su origen en los años 60
 - ☐ Repertorio de caracteres con sólo 128 elementos
-

Tabla ASCII

b₇ → b₆ → b₅ →					0	0	0	0	1	1	1	1	
					0	0	0	1	0	1	0	1	
Column → Row ↓					0	1	2	3	4	5	6	7	
Bits	b ₄ ↓	b ₃ ↓	b ₂ ↓	b ₁ ↓	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(	8	H	X	h	x
	1	0	0	1	9	HT	EM	)	9	I	Y	i	y
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FC	,	<	L	\	l	/
	1	1	0	1	13	CR	GS	-	=	M	]	m	}
	1	1	1	0	14	SO	RS	.	>	N	^	n	~
	1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Codificación ASCII

1. Tomamos la posición en la tabla ASCII del carácter (g = 103)
2. Conversión a binario (en 7 dígitos)

1. $103 = 51 * 2 + 1$

2. $51 = 25 * 2 + 1$

3. $25 = 12 * 2 + 1$

4. $12 = 6 * 2 + 0$

5. $6 = 3 * 2 + 0$

6. $3 = 1 * 2 + 1$

7. $1 = 0 * 2 + 1$

Representación en binario: 01100111

3. Añadimos el bit de paridad (1 si hay un número impar de unos, 0 en otro caso) en el octavo dígito (g = 101100111)

ISO – 8859 - ...

- ❑ Repertorio de 256 caracteres (8 bits)
 - ❑ Los caracteres 1 a 128 coinciden con los caracteres imprimibles de ASCII
 - ❑ Contiene variantes para zonas geográficas:
 - ISO-8859-1: ISOLatin1 (Europa Occidental)
 - ISO-8859-2: ISOLatin2 (Europa Oriental)
 - ...
 - ISO-8859-15: ISOLatin9 (Europa, incluye €)
-

Tabla de codificación ISO-8859-1

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	<u>NUL</u> 0000	<u>STX</u> 0001	<u>SOT</u> 0002	<u>ETX</u> 0003	<u>EOT</u> 0004	<u>ENQ</u> 0005	<u>ACK</u> 0006	<u>BEL</u> 0007	<u>BS</u> 0008	<u>HT</u> 0009	<u>LF</u> 000A	<u>VT</u> 000B	<u>FF</u> 000C	<u>CR</u> 000D	<u>SO</u> 000E	<u>SI</u> 000F
10	<u>DLE</u> 0010	<u>DC1</u> 0011	<u>DC2</u> 0012	<u>DC3</u> 0013	<u>DC4</u> 0014	<u>NAK</u> 0015	<u>SYN</u> 0016	<u>ETB</u> 0017	<u>CAN</u> 0018	<u>EM</u> 0019	<u>SUB</u> 001A	<u>ESC</u> 001B	<u>FS</u> 001C	<u>GS</u> 001D	<u>RS</u> 001E	<u>US</u> 001F
20	<u>SP</u> 0020	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	:	;	<	=	>	?
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	<u>DEL</u> 007F
80																
90																
A0	<u>NBSP</u> 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	­ 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

DOS - ...

- ❑ Repertorio de 256 caracteres (8 bits)
 - ❑ Los caracteres 33 a 128 coinciden con los caracteres imprimibles de ASCII
 - ❑ Generalmente usada en la consola de MSDOS (chcp)
 - ❑ Contiene variantes para zonas geográficas:
 - DOS-437: DOSUS (Estados Unidos)
 - DOS-850: DOSLatin1 (Europa Occidental)
-

Windows - ...

- ❑ Repertorio de 256 caracteres (8 bits)
 - ❑ Los caracteres 33 a 128 coinciden con los caracteres imprimibles de ASCII, suelen ser compatibles con las ISO correspondientes
 - ❑ Generalmente usada en aplicaciones Windows (80s y 90s)
 - ❑ Contiene variantes para zonas geográficas:
 - Windows-1252: WinLatin1 (Europa Occidental)
 - Windows-1250, Windows-1251 (Europa Oriental)
-

Tabla de caracteres Win- 1252

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	:	;	<	=	>	?
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‘ 201A	’ 0192	“ 201E	… 2026	† 2020	‡ 2021	~ 02C6	% 2030	Š 0160	< 2039	Œ 0152		Ž 017D	
90		‚ 2018	‘ 2019	“ 201C	” 201D	• 2022	— 2013	— 2014	~ 02DC	™ 2122	Š 0161	> 203A	œ 0153		Ž 017E	Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	­ 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ø 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Unicode

- ❑ Repertorio y tablas de más de 100.000 caracteres (diversas codificaciones!!)
 - ❑ Compatible con ASCII (del 1 al 128)
 - ❑ Estándar de facto: usado internamente en Windows (desde NT), Java, .NET, Unix, Mac OS X, KDE...
 - ❑ No dispone de variantes, la idea es incluir cualquier carácter de cualquier idioma
 - ❑ Dispone de varias codificaciones (UTF-8, UTF-16, UTF-32)
-

UTF-8

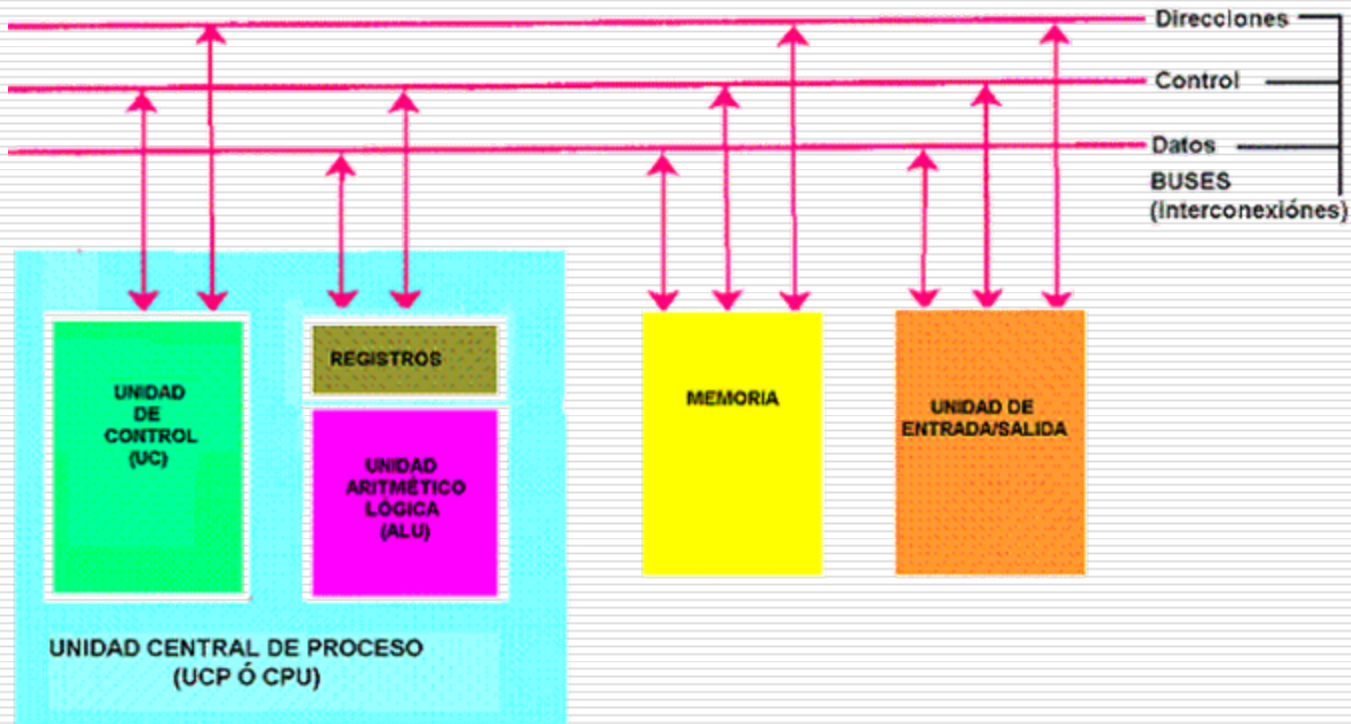
- ❑ Codificación de Unicode de longitud variable (uno o más bytes)
 - ❑ Los caracteres ASCII los codifica en un byte
 - ❑ Para cualquier otro carácter usa 2 o más bytes (mirar los apuntes)
 - ❑ A menudo añade al principio de los archivos el BOM, marca que avisa de que el archivo está en formato UTF-8
-

UTF-16

- ❑ Codificación de Unicode de longitud variable (2 o más bytes)
 - ❑ Incompatible con archivos ASCII, UTF-8, ISO-8859-... (recordar el ejemplo de www.unirioja.es)
 - ❑ Los caracteres en el BMP (los 4096 primeros) los codifica en 2 bytes (los demás en 4)
-

1.3 El ordenador. Fundamentos estructurales y de funcionamiento

Estructura de programa almacenado



UCP (unidad central de proceso)

- ❑ Registros: son un número reducido de áreas de almacenamiento en las que se tiene la información con la que en cada momento trabaja la UCP. Se utilizan para guardar direcciones o datos de memoria (MAR, MDR), de las instrucciones llevadas a cabo (IR) o de las que se han de ejecutar (PC).
 - ❑ UAL (Unidad aritmético-lógica): se encarga de realizar las operaciones aritméticas o lógicas con los datos que se encuentren en los registros correspondientes.
 - ❑ UC (Unidad de control): dirige todo el proceso, leyendo del registro correspondiente la instrucción a ejecutar y generando las señales de control para que se realicen las operaciones pertinentes.
-

UCP: simulación de uso

□ <http://arity.ca/2010-11/CSE101/resources/RSCPU/>

Ejemplo de UCP con bus único

IR: Instruction register

PC: Program counter

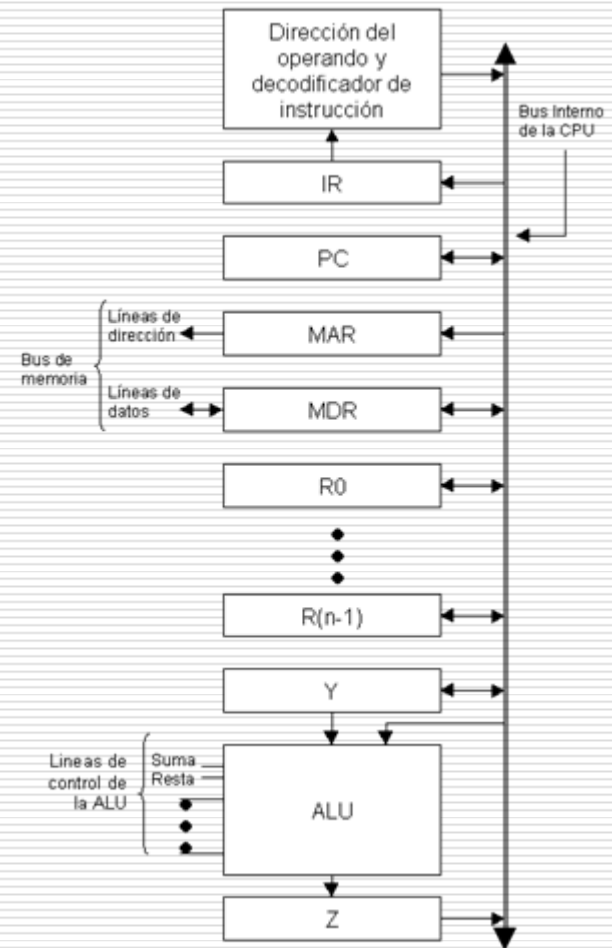
MAR: memory address register

MDR: memory data register

ALU: Unidad aritmético-lógica

R0...R(n-1): registros de aplicación especial

Y, Z: registros de uso interno y almacenamiento temporal



UCP: lenguaje máquina

- ❑ La UCP tiene un lenguaje propio de instrucciones que permiten llevar a cabo operaciones sobre memoria, datos...
 - ❑ Estas instrucciones “consumen” ciclos del procesador. La cantidad de ciclos (e decir, de instrucciones) que un procesador puede completar por segundo se mide en hercios ($1,6 \text{ GHz} = 1,6 * 10^9$ ciclos por segundo)
 - ❑ Las instrucciones de un procesador son específicas del mismo (x86, tarjetas gráficas...); por eso, mayor velocidad no siempre es sinónimo de mayor rendimiento (si el lenguaje máquina estás más optimizado...)
-

UCP:lenguaje máquina

- ❑ http://en.wikipedia.org/wiki/X86_instruction_listings
 - ❑ http://oopweb.com/Assembly/Documents/ArtOfAssembly/Volume/Chapter_6/CH06-1.html
 - ❑ http://en.wikipedia.org/wiki/Intel_Core_i7
-

UCP: lenguaje máquina

- Uno de los lenguajes máquina, o de instrucciones, más extendidos es lenguaje ensamblador (es lo más legible a la vez que cercano al lenguaje máquina propio de la UCP). El mismo es específico del procesador sobre el que lo ejecutemos.
 - http://es.wikipedia.org/wiki/Lenguaje_ensamblador#Ejemplos_de_lenguaje_ensamblador
-

Memoria interna

- ❑ Almacén de datos y programas
 - ❑ Es volátil (desaparece al apagar el equipo)
 - ❑ Se compone de varias cachés y de la memoria interna
 - ❑ Todas ellas son de tipo RAM (aunque generalmente llamamos RAM a la interna)
-

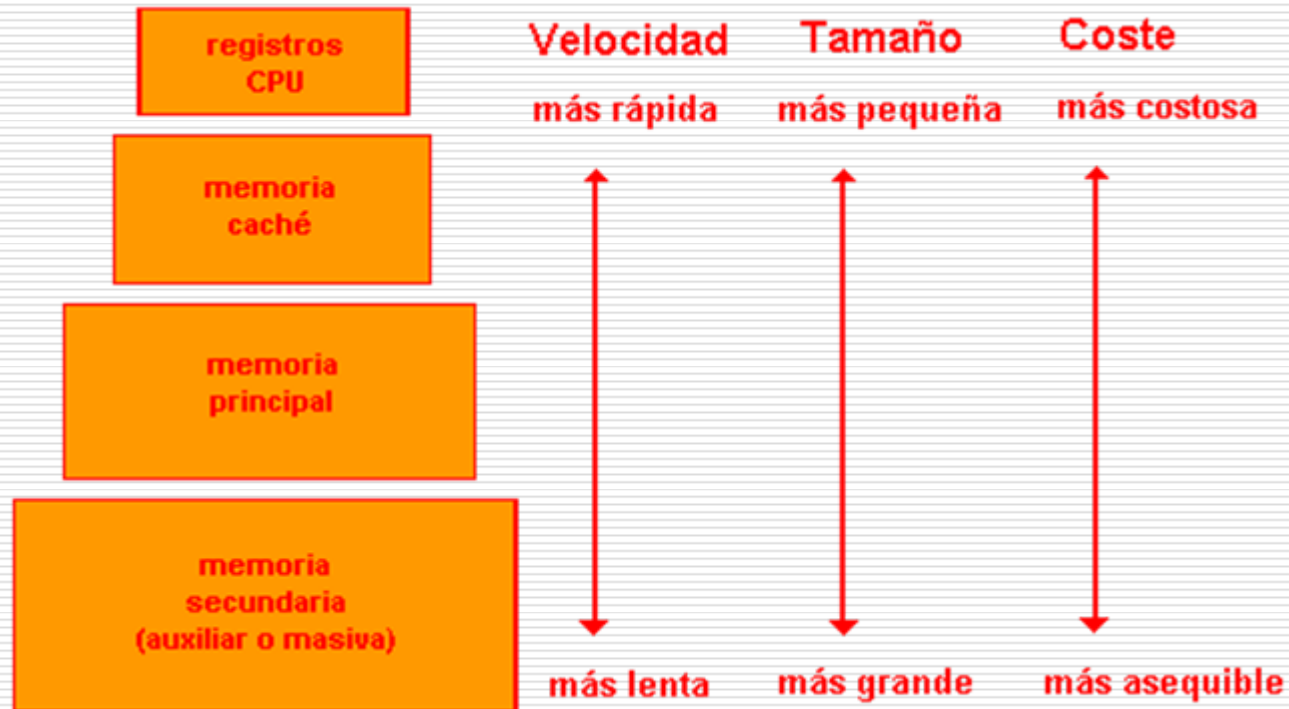
Memoria interna

Se estructura en varios niveles. Cuanto más próxima a la UCP, resulta más cara, más pequeña y más rápida

Niveles:

- 1.Registros de memoria (RDM, RAM, IR, PC)
 - 2.Memoria caché (nivel 1, 2, 3...)
 - 3.Memoria principal (o RAM)
 - 4.Memoria secundaria o externa (disco duro...)
-

Memoria interna



Memoria interna

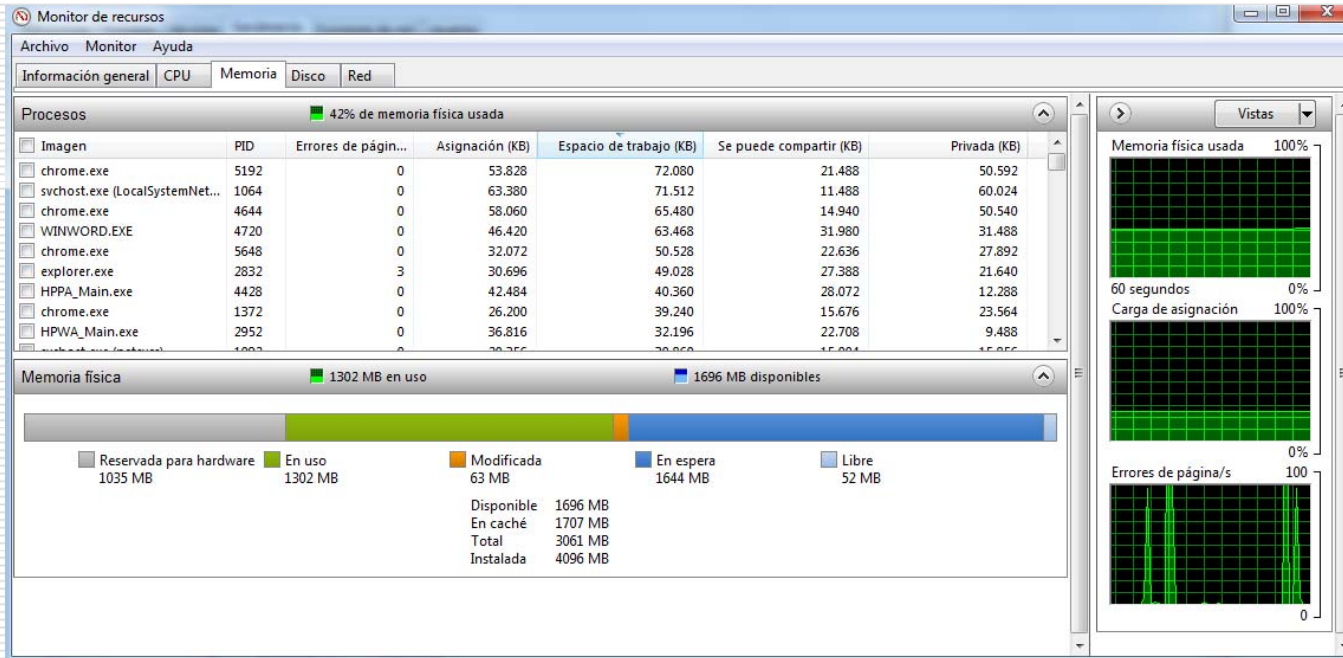
```
Memtest86+ v1.55 | Pass 0%
Athlon 64 (0.13) 2701 Mhz | Test 68% #####
L1 Cache: 128K 22136MB/s | Test #2 [Moving inversions, ones & zeros]
L2 Cache: 512K 5274MB/s | Testing: 112K - 1024M 1024M
Memory : 1024M 2455MB/s | Pattern: ffffffff
Chipset : nVidia nForce4 (ECC : Disabled)
Settings: RAM : 192 MHz (DDR385) / CAS : 2.5-3-3-8 / Dual Channel (128 bits)

WallTime  Cached  RsvdMem  MemMap  Cache  ECC  Test  Pass  Errors  ECC Errs
-----
0:00:05  1024M    276M  e820-Std   on  off  Std    0      1      0

-----
Tst  Pass  Failing Address          Good      Bad      Err-Bits  Count Chan
---  ---  -
2    0  0001a3effa8 - 419.9MB  ffffffff ffffffe9 00000016    1

(ESC)Reboot (c)configuration (SP)scroll_lock (CR)scroll_unlock
```

Memoria interna



Otra forma de observar la memoria interna del equipo y su uso.

La memoria interna aloja los procesos (programas en ejecución) y los datos que se encuentran en funcionamiento en nuestro equipo en un momento concreto. La memoria interna se divide en páginas que son ocupadas por distintos procesos

Memoria interna

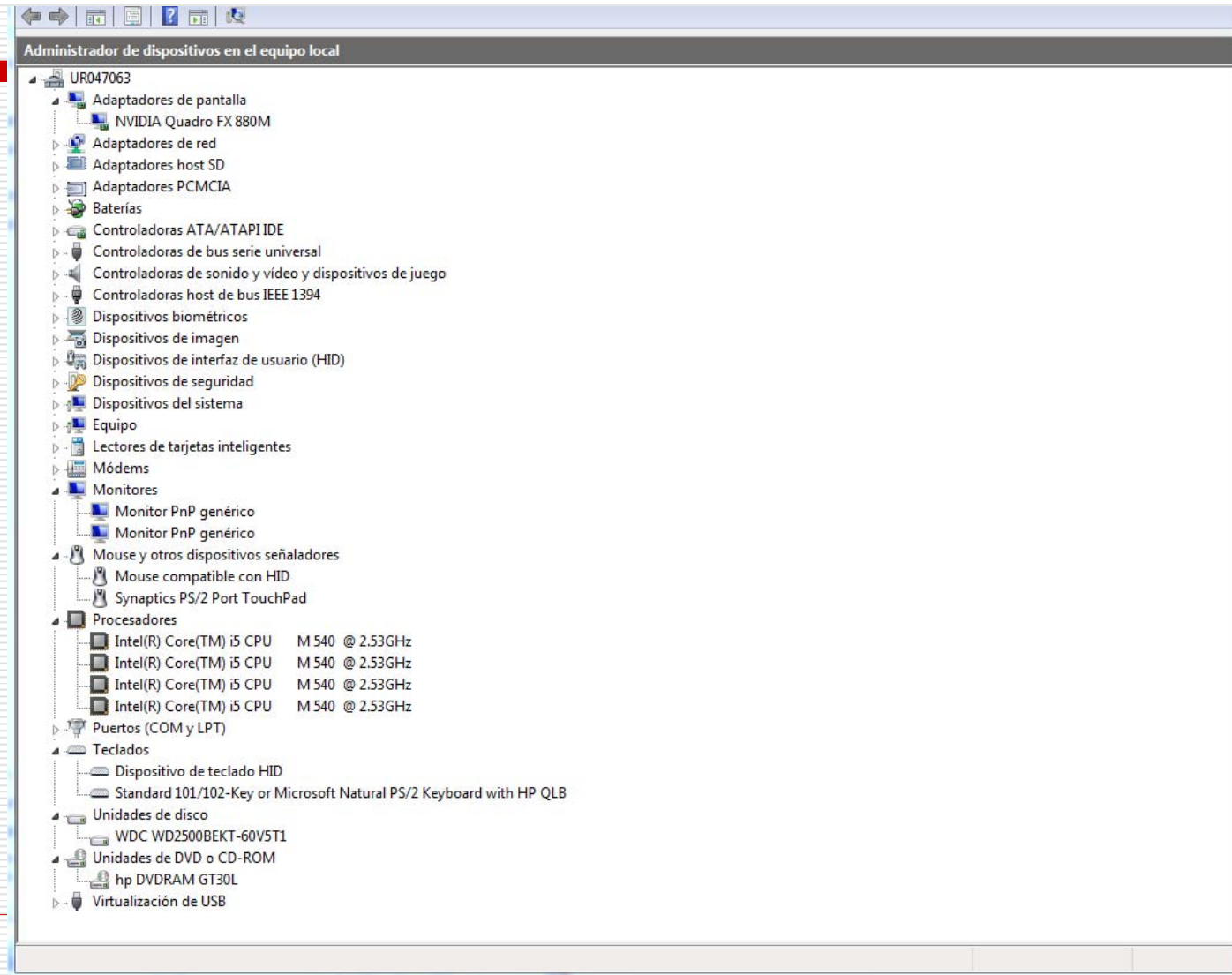
Propiedades de la memoria interna:

- ❑ Suele ser de tipo RAM (o SDRAM, Synchronous Dynamic RAM)
 - ❑ Su capacidad se mide en GB (tamaños habituales hoy en día son 2, 4 ó 8 GB)
 - ❑ Su velocidad (número de ciclos de lectura o escritura por segundo) se mide en Hz (velocidades habituales, 1,6GHz por segundo, $1,6 \cdot 10^9$ ciclos por segundo)
 - ❑ Disponen también de tiempos de latencia, que representan el tiempo (en ciclos) que consumen ciertas operaciones básicas (tiempo de acceso a una columna, a una fila y columna, de una fila a la siguiente y para activar un tablero completo). Generalmente se expresan como 6-6-6-18. A menores latencias, mayor rendimiento
-

Unidades de E/S (Entrada/Salida)

- Comunican periféricos (teclado, ratón, pantalla, impresoras) con la UCP
 - Posibilidades de comunicación:
 - E/S programada (UCP “visita” el módulo de E/S periódicamente)
 - E/S dirigida por interrupciones (bidireccional)
 - Acceso directo a memoria (independiente de la UCP)
-

Dispositivos (algunos de ellos de E/S)



Unidades de E/S

- ❑ Todos ellos requieren de un controlador (o driver) que traduce las órdenes enviadas y recibidas por la UCP al lenguaje que “entiende” el periférico o dispositivo de E/S
 - ❑ La instalación de un periférico debería ir acompañada de la instalación del controlador correspondiente (o de uno genérico) que permita la comunicación bidireccional
-

Elementos de interconexión (BUS)

Usados para transferir información entre dos puntos de la computadora

Se distinguen:

- ❑ Bus serie (datos transferidos bit a bit).

Usado en puertos USB o conexiones Firewire y cada vez más en conexiones entre los elementos de la placa base

- ❑ Bus paralelo (conjuntos de bits).

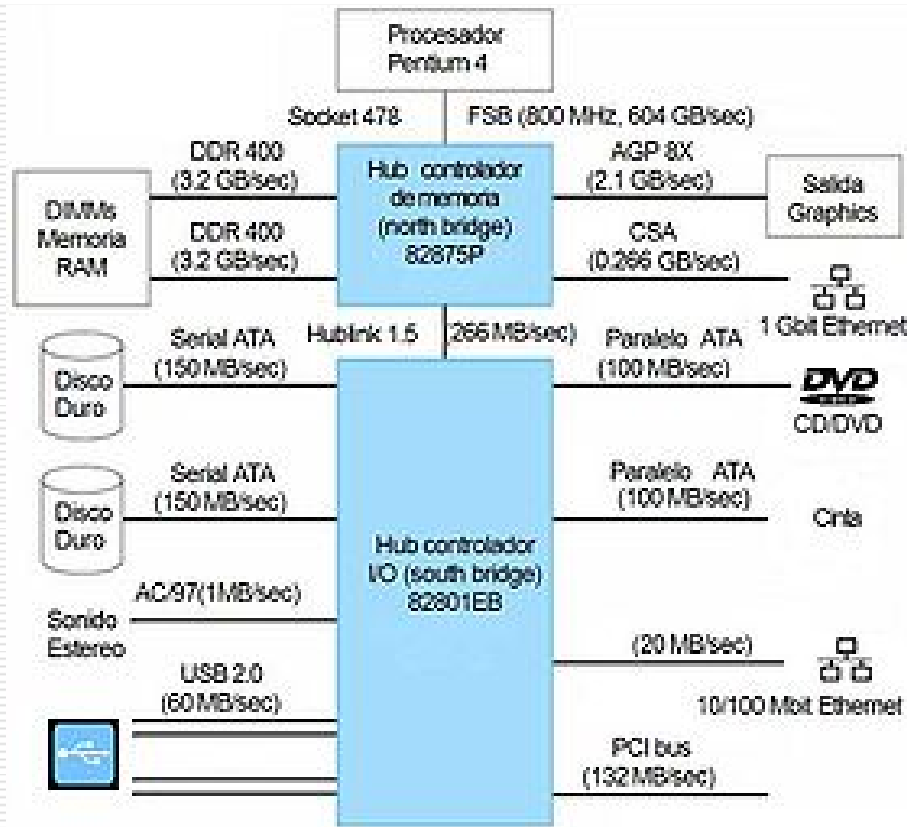
Dispone de varias líneas que realizan la comunicación de forma simultánea

Elementos de interconexión (puertos)

Son interfaces para el envío y recepción de datos entre dispositivos.

Se dividen también, por su tipo de comunicación, en puertos serie y puertos paralelos

Elementos de interconexión (BUS)



Ejemplo de dispositivo con distintos puertos y buses de comunicación con sus dispositivos externos y propios de la placa

El disco duro (dispositivo de E/S)

Es importante diferenciar entre su **estructura física** (de naturaleza mecánica) y su **estructura lógica** (dividido en unidades de asignación indexadas...)

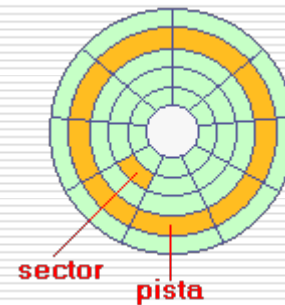
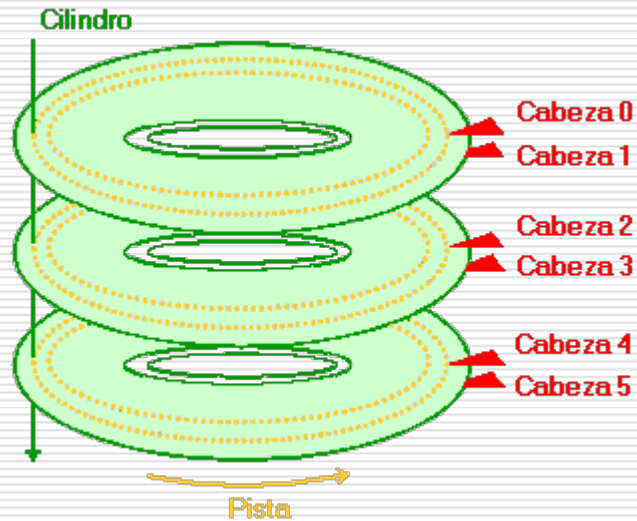
El **formateo** nos permite “olvidar” la estructura física y trabajar sobre la lógica

El disco duro (estructura física)

Estructura física (no de los SSD, solid state drive):

- ☐ Un conjunto de platos de superficie magnética sujetos a un eje que gira a miles de rpm
 - ☐ Un conjunto de cabezas de lectura/escritura sujetas sobre un brazo móvil. Cada cabeza lee sobre una superficie distinta.
 - ☐ Un bus de datos que conecta las cabezas con la electrónica de la unidad
-

El disco duro (estructura física)



El disco duro (estructura física)

- ❑ Cada sector contiene 512 bytes (se está empezando a trabajar en sectores de 4KB)
- ❑ Cada disco contiene el mismo número de cilindros
- ❑ Tenemos tantas superficies como cabezas

Tamaño del disco duro: $512 \text{ bytes} * (\text{n}^\circ \text{ de sectores por cilindro}) * (\text{cilindros por disco}) * (\text{n}^\circ \text{ de cabezas})$

Disco duro (estructura física)

No cualquier número de cilindros, cabezas y sectores está permitida (recordar práctica 2, trucos como Logical Block Addressing, LBA)

Disco duro (estructura lógica)

Por medio del formateo le damos estructura lógica al disco. Dos funciones:

1. Divide el disco duro en “unidades de asignación” (o clusteres), de tamaño 512 bytes, 1KB, 2KB, 4KB...64KB.

El cluster es un conjunto contiguo de sectores que componen la unidad más pequeña de almacenamiento (o espacio mínimo que ocupará cualquier archivo en disco)

Disco duro (estructura lógica)

Por medio del formateo le damos estructura lógica al disco. Dos funciones:

2. Crea una tabla de asignación de archivos donde se almacenan las coordenadas de cada “cluster”, el nombre del fichero allí contenido y alguna otra propiedad (depende del formato: FAT32, NTFS...)

Disco duro (funcionamiento)

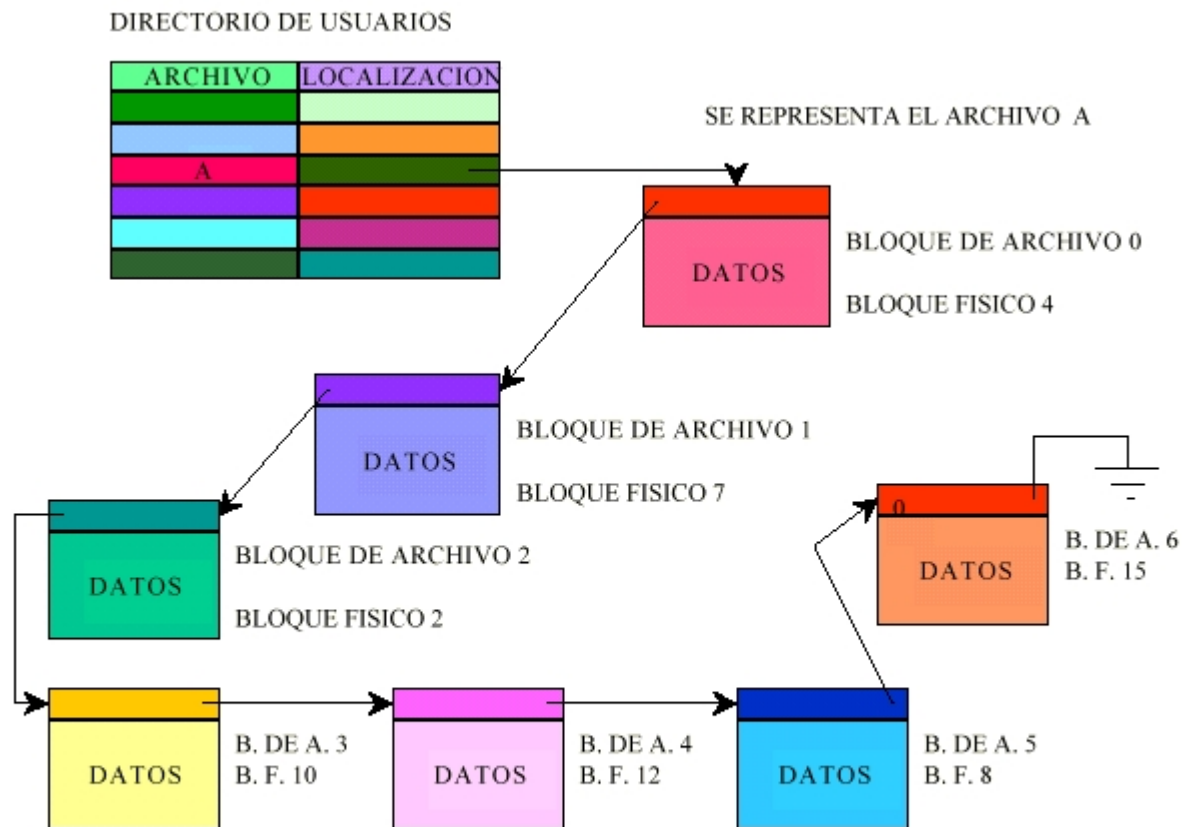


Figura ~4.4: Encadenamiento de bloques o lista ligada de bloques.

Sistema de arranque (MBR)

La BIOS (ROM BIOS) realiza comprobaciones básicas y carga el MBR (primer sector, 512 bytes) de la partición activa del disco de arranque.

Primer sector físico del disco: 512 Bytes

446 Bytes: Código máquina (gestor de arranque)

64 Bytes: [Tabla de particiones](#)

2 Bytes: Firma de unidad arrancable ("55h AAh" en hexadecimal)

Limitaciones del MBR

- ❑ Sólo puede contener 4 particiones primarias
- ❑ Sólo una de las particiones puede ser activa

0000	33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00	3A2B4. 2A284. c.
0010	06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB B9 04 00	. 1..úóxPh..Eú1..
0020	BD BE 07 80 7E 00 00 7C 0B 0F 85 10 01 83 C5 10	%4..~..fA.
0030	E2 F1 CD 18 88 56 00 55 C6 46 11 05 C6 46 10 00	âRf..U.UEF..EF..
0040	B4 41 BB AA 55 CD 13 5D 72 0F 81 FB 55 AA 75 09	'A»=Uí. r..úU²u0
0050	F7 C1 01 00 74 03 FE 46 10 66 60 80 7E 10 00 74	+â..t..pF.f'.~..t
0060	26 66 68 00 00 00 66 FF 76 08 68 00 00 68 00	sfh....fýv.h..h.
0070	7C 68 01 00 68 10 00 B4 42 8A 56 00 8B F4 CD 13	h..h.. 'B\$V.<ôf.
0080	9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00	YfA.ââ...> SV.
0090	8A 76 01 8A 4E 02 8A 6E 03 CD 13 66 61 73 1E FE	Šv.ŠW.Šn.f.fas.p
00A0	4E 11 0F 85 0C 00 80 7E 00 80 0F 84 8A 00 B2 80	N.....~.....Š.².
00B0	EB 82 55 32 E4 8A 56 00 CD 13 5D EB 9C 81 3E FE	ë,U2âSV.f. éœ.>p
00C0	7D 55 AA 75 6E FF 76 00 E8 8A 00 0F 85 15 00 B0	)U²unýv.âŠ....."
00D0	D1 E6 64 E8 7F 00 B0 DF E6 60 E8 78 00 B0 FF E6	Ñedè.. "Bæ'èx."ýæ
00E0	64 E8 71 00 B8 00 BB CD 1A 66 23 C0 75 3B 66 81	dèq.,..>f.f#âu;f.
00F0	FB 54 43 50 41 75 32 81 F9 02 01 72 2C 66 68 07	ûTCPâu2.ù..r,fh.
0100	BE 00 00 66 68 00 02 00 00 66 68 08 00 00 00 66	>..fh....fh....f
0110	53 66 53 66 55 66 68 00 00 00 00 66 68 00 7C 00	3f3fUfh....fh. .
0120	00 66 61 68 00 00 07 CD 1A 5A 32 F6 EA 00 7C 00	.fah....f.22ôé. .
0130	00 CD 18 A0 B7 07 EB 08 A0 B6 07 EB 03 A0 B5 07	.f. . .ë. ¶.â. p.
0140	32 E4 05 00 07 8B F0 AC 3C 00 74 FC EB 07 00 B4	2â...<ô<c.tû>..'
0150	0E CD 10 EB F2 2B C9 E4 64 EB 00 24 02 E0 F8 24	.f.âô+ââdè.â.â&f
0160	02 C3 49 6E 76 61 6C 69 64 20 70 61 72 74 69 74	.MInvalid partit
0170	69 6F 6E 20 74 61 62 6C 65 00 45 72 72 6F 72 20	ion table.Error
0180	6C 6F 61 64 69 6E 67 20 6F 70 65 72 61 74 69 6E	loading operatin
0190	67 20 73 79 73 74 65 6D 00 4D 69 73 73 69 6E 67	g system.Missing
01A0	20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74 65	operating syste
01B0	6D 00 00 00 00 62 7A 99 CC 45 23 77 00 00 00 20	m....bs.lE#w...
01C0	21 00 17 D3 E3 FF 00 08 00 00 00 C0 D4 01 80 00	!...ôZý.....Aô...
01D0	C1 FF 17 EF FF FF 00 CF D4 01 00 CF D4 01 00 00	âý.ýýý.Yô..Yô...
01E0	C1 FF 07 EF FF FF 00 9E A9 03 B0 9D 38 01 00 00	âý.ýýý.âø..".8...
01F0	C1 FF 0F EF FF FF B0 3B E2 04 C0 51 46 02 55 AA	âý.ýýý";â.âQF.Uâ