

## **FUNDAMENTOS DEL HARDWARE**

**UT 1 – *Conceptos básicos. Representación de la información.***

## **Índice de contenido**

1. Introducción.....	4
2. Historia de los ordenadores.....	5
2.1 Primera generación: 1940 – 1960.....	5
2.2 Segunda generación: 1960 – 1965.....	6
2.3 Tercera generación: 1965 – 1975.....	7
2.4 Cuarta generación: 1975 – 1990.....	7
2.5 Quinta generación: 1990 – hoy.....	8
3. Sistemas y códigos de numeración.....	9
3.1 Sistema binario.....	9
3.1.1 Conversión binario – decimal.....	10
3.1.2 Conversión decimal – binario.....	10
3.2 Sistema Octal.....	10
3.2.1 Conversión octal – decimal.....	10
3.2.2 Conversión decimal – octal.....	10
3.2.3 Conversión binario – octal.....	10
3.2.4 Conversión octal – binario.....	11
3.3 Sistema Hexadecimal.....	11
3.3.1 Conversión hexadecimal – decimal.....	11
3.3.2 Conversión decimal – hexadecimal.....	11
3.3.3 Conversión binario – hexadecimal.....	11
3.3.4 Conversión hexadecimal – binario.....	11
3.4 Resumen de conversiones numéricas.....	12
3.5 Operaciones aritméticas con sistema binario.....	13
3.5.1 Suma.....	13
3.5.2 Resta.....	14
3.5.3 Multiplicación.....	15
4. Códigos alfanuméricos.....	16
4.1 El código ASCII.....	16
4.2 El código UNICODE.....	17
5. Componentes básicos de un computador.....	18

## **1. Introducción.**

En esta unidad trataremos una serie de conceptos básicos relacionados con los elementos hardware de un ordenador. Para ello, comenzaremos estudiando la historia y evolución de los ordenadores y posteriormente su estructura básica, los elementos que lo forman y como se representa y almacena la información dentro de ellos.

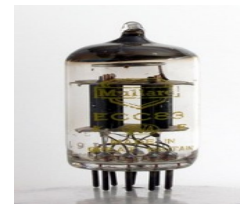
## **2. Historia de los ordenadores.**

Los computadores actuales no han surgido de la noche a la mañana. De hecho, podemos decir que se ha recorrido un largo camino hasta diseñar las máquinas actuales. En concreto, se diferencian cinco generaciones de ordenadores a lo largo del tiempo, cada una de ellas marcada por una serie de avances tecnológicos que permitían crear computadores más rápidos y potentes que sus antecesores.

Dichas generaciones son las siguientes:

### **2.1 Primera generación: 1940 – 1960.**

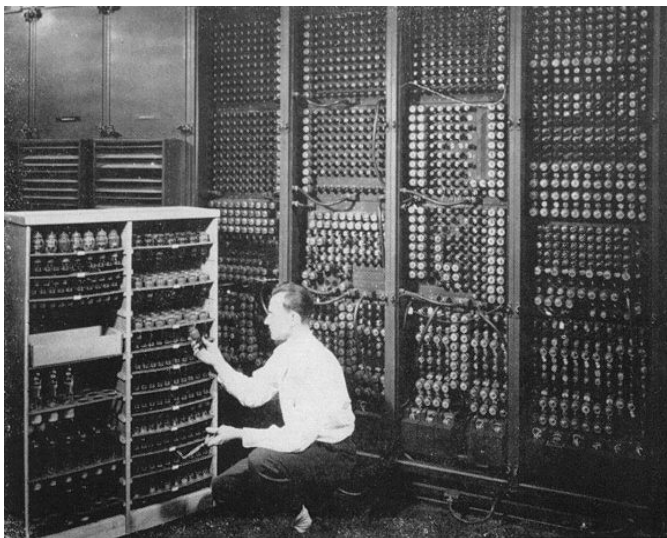
Fueron los primeros ordenadores que se construyeron, su principal característica era que usaban válvulas de vacío.



*Válvula de vacío*

Los principales ordenadores de este tipo fueron:

- ENIAC (Electronic Numerical Integrator and Computer). Este computador usaba tarjetas perforadas y pesaba más de 30 toneladas.

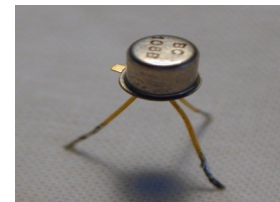


- UNIVAC (UNIVersAL Computer). Era un computador que disponía de 1000 palabras de memoria central (cada palabra era un carácter o letra) y podía leer cintas magnéticas.



## **2.2 Segunda generación: 1960 – 1965.**

- Comienzan a construirse ordenadores sustituyendo las válvulas de vacío por transistores. Un transistor es un componente electrónico que en función de si se le aplica o no una corriente, puede actuar como conductor o como aislante, permitiendo de esta forma que un circuito pueda estar en dos posibles estados (presencia de tensión o ausencia de tensión).
- En este periodo se crearon también los discos magnéticos y lenguajes de programación denominados de alto nivel.
- No obstante, en esos momentos el precio de los ordenadores era todavía muy elevado.



### **2.3 Tercera generación: 1965 – 1975.**

- Aparecen los circuitos integrados, formados por un circuito impreso que acumulaba millones de transistores.
- IBM crea su primer computador basado en circuitos impresos, llamado serie 360. Este modelo incorporaba un sistema operativo llamado OS, que incluía técnicas para el manejo de la memoria y el procesador.
- Estos computadores se vendían a precios asequibles para el usuario corriente.



IBM 360

### **2.4 Cuarta generación: 1975 – 1990.**

- Se extiende la fabricación y venta de ordenadores PC de bajo coste.
- Aparecen los primeros microprocesadores, el primero de los cuales fue el INTEL 4004. Posteriormente surgirían los procesadores INTEL 8086, 286, 386 ...
- Se emplean también memorias de semiconductores eliminando de esta forma las antiguas memorias de ferritas.



### **2.5 Quinta generación: 1990 – hoy.**

- Surge una gran competencia entre fabricantes como INTEL, AMD o Apple, lo cual acelera la aparición de nuevos microprocesadores más potentes, desde la gama Pentium o AMD K7 hasta los modernos INTEL i5, INTEL i7 o AMD AM3+ y AMD FM1
- Actualmente las líneas de mejora pasan por buscar una mayor integración y velocidad del hardware y también el concepto de procesamiento paralelo.

### **3. Sistemas y códigos de numeración.**

Un sistema de numeración es el conjunto de símbolos utilizados para la representación de cantidades, así como las reglas que rigen esta representación.

Los sistemas de numeración actuales cumplen las siguientes condiciones:

- Todo número está formado por un conjunto de símbolos llamados dígitos.
- El número de símbolos que podemos usar en un sistema de numeración, se corresponde con la “base” de este.
- El valor de un número dependerá de la posición de cada dígito y la base de numeración, pudiendo expresarse como una suma de productos, en la que el exponente asociado a la base aumenta de derecha a izquierda, empezando por 0.

En la vida cotidiana todos nosotros usamos el sistema de numeración decimal, no obstante, en el entorno de los ordenadores se emplean los sistemas binario, octal y hexadecimal.

#### **3.1 Sistema binario.**

Es un sistema de numeración polinomial de base 2, con lo que solo usa dos símbolos el 0 y el 1.

Su rango de representación con  $n$  dígitos es de  $2^n$  números.

Este sistema se usa a nivel hardware en los computadores debido a la dificultad de crear circuitos integrados con componentes capaces de representar más de dos estados físicos.

A estos valores 0 y 1 se les denomina bit (binary digit) y son la menor cantidad de información que podemos representar en un computador. A su vez existen diversas unidades formadas por conjuntos de bits:

1 Byte = 8 bits  
1 KByte (kilobyte) = 1024 Bytes  
1 MByte (megabyte) = 1024 KBytes  
1 GByte (gigabyte) = 1024 MBytes  
1 TByte (terabyte) = 1024 GBytes



### **3.1.1 Conversión binario – decimal.**

Por medio del teorema fundamental de numeración.

### **3.1.2 Conversión decimal – binario.**

Por medio del método de las divisiones sucesivas.

Se divide el número decimal entre 2, sucesivamente hasta que se obtenga un cociente inferior a 2. El número binario resultante será este cociente seguido de los restos obtenidos en las divisiones.

Si hubiese parte fraccionaria, se multiplica esta por 2 cogiendo la parte entera de cada multiplicación y repitiendo después la operación con la siguiente parte fraccionaria.

## **3.2 Sistema Octal.**

Utiliza 8 símbolos que van del 0 al 7.

Se usa a nivel software para representar números de una forma más cómoda que con el sistema binario. Esto es

Su rango de representación con n dígitos es de  $8 \text{ Exp } n$ .

### **3.2.1 Conversión octal – decimal.**

Por medio del teorema fundamental de numeración.

### **3.2.2 Conversión decimal – octal.**

Por medio del método de las divisiones sucesivas.

Se divide el número decimal entre 8, sucesivamente hasta que se obtenga un cociente inferior a 8. El número binario resultante será este cociente seguido de los restos obtenidos en las divisiones.

### **3.2.3 Conversión binario – octal.**

Cogemos los bits de derecha a izquierda en grupos de 3 y los convertimos a decimal. El resultado de cada conversión será un dígito octal válido.

### **3.2.4 Conversión octal – binario.**

Seleccionamos cada dígito octal y lo convertimos a un conjunto de 3 bits binarios como si se tratara de un número decimal corriente.

## **3.3 Sistema Hexadecimal.**

Utiliza 16 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Se usa a nivel software para representar números de una forma más cómoda que con el sistema binario. Principalmente para indicar direcciones de memoria.

Su rango de representación con n dígitos es de  $16^{\text{Exp } n}$ .

### **3.3.1 Conversión hexadecimal – decimal.**

Por medio del teorema fundamental de numeración.

### **3.3.2 Conversión decimal – hexadecimal.**

Por medio del método de las divisiones sucesivas.

Se divide el número decimal entre 16, sucesivamente hasta que se obtenga un cociente inferior a 16. El número binario resultante será este cociente seguido de los restos obtenidos en las divisiones.

Si hubiese parte fraccionaria, se multiplica esta por 16 cogiendo la parte entera de cada multiplicación y repitiendo después la operación con la siguiente parte fraccionaria.

### **3.3.3 Conversión binario – hexadecimal.**

Cogemos los bits de derecha a izquierda en grupos de 4 y los convertimos a decimal. El resultado de cada conversión será un dígito octal válido.

### 3.3.4 Conversión hexadecimal – binario.

Seleccionamos cada dígito octal y lo convertimos a un conjunto de 4 bits binarios como si se tratara de un número decimal corriente.

### 3.4 Resumen de conversiones numéricas.

Seguidamente se añade a modo de resumen, una tabla con las reglas de conversiones numéricas tratadas anteriormente.

DE	A
Cualquier base	Decimal
<p>Utilizamos el sistema de las potencias.  Ejemplos:  El número binario 110 en decimal seria: <math>0*2^0 + 1*2^1 + 1*2^2</math>  El número octal 763 en decimal seria: <math>3*8^0 + 6*8^1 + 7*8^2</math></p>	
Decimal	Cualquier base
<p>Utilizamos el sistema de las divisiones sucesivas. El divisor será igual a la base a la que queremos convertir. Por ejemplo, si queremos convertir a binario la base será 2 y a octal será 8.</p>	
Binario	Octal
<p>Cogemos los bits de derecha a izquierda en grupos de 3 y los convertimos a decimal. El resultado de cada conversión será un dígito octal válido.  Ejemplo: 111 101 001 -&gt; 7 5 1</p>	
Octal	Binario
<p>Seleccionamos cada dígito octal y lo convertimos a un conjunto de 3 bits binarios como si se tratara de un número decimal corriente.  Ejemplo: 431 -&gt; 100 011 001</p>	
Binario	Hexadecimal
<p>Cogemos los bits de derecha a izquierda en grupos de 4 y los convertimos a decimal. El resultado de cada conversión será un dígito hexadecimal válido.  Ejemplo: 1111 0001 1010 -&gt; F 1 A</p>	
Hexadecimal	Binario
<p>Seleccionamos cada dígito octal y lo convertimos a un conjunto de 4 bits binarios como si se tratara de un número decimal corriente.</p>	

*Fundamentos del Hardware*  
*UT1 – Conceptos básicos. Representación de la información.*

Ejemplo: 3CE -> 0011 1100 1110

### 3.5 Operaciones aritméticas con sistema binario.

Una vez conocemos el funcionamiento de los principales sistemas de numeración, es conveniente tratar la forma en la que se realizan algunas de las operaciones aritméticas fundamentales.

En concreto estudiaremos la suma, resta y multiplicación binaria.

#### 3.5.1 Suma.

Para sumar en números binarios, tendremos en cuenta la siguiente tabla:

	Resultado	Acarreo
<b>0 + 0</b>	0	0
<b>0 + 1</b>	1	0
<b>1 + 0</b>	1	0
<b>1 + 1</b>	0	1

**Ejemplos:**

				1	1	1	Acarreos	
				↓	↓	↓		
	1	0	1	0	1	1	1	→ 87
+		1	0	0	0	0	1	→ 33
<hr/>								
	1	1	1	1	0	0	0	→ 120

				1	1	1	Acarreos	
				↓	↓	↓		
		1	1	0	1	1	1	→ 55
+		1	0	0	0	1	1	→ 35
<hr/>								
	1	0	1	1	0	1	0	→ 90

### 3.5.2 Resta.

Para sumar en números binarios, tendremos en cuenta la siguiente tabla:

	Resultado	Acarreo
<b>0 - 0</b>	0	0
<b>0 - 1</b>	1	1
<b>1 - 0</b>	1	0
<b>1 - 1</b>	0	0

**Ejemplos:**

	1	1	1	0	1	0	1	→ 117
-	↓1	↓1	↓1		↓1			Acarreos
		1	1	1	0	1	0	→ 58
	0	1	1	1	0	1	1	→ 59
	1	1	0	0	1	0	1	→ 101
-		↓1	↓1		↓1			Acarreos
			1	1	0	1	1	→ 27
	1	0	0	1	0	1	0	→ 74

### 3.5.3 Multiplicación.

Tendremos en cuenta la siguiente tabla.

	Resultado
<b>0 * 0</b>	0
<b>0 * 1</b>	0
<b>1 * 0</b>	0
<b>1 * 1</b>	1

**Ejemplos:**

$\begin{array}{r} 1100010 \\ \times 1010 \\ \hline 0000000 \\ 1100010 \\ 0000000 \\ 1100010 \\ \hline 1111010100 \end{array}$	$\begin{array}{r} 10110100 \\ \times 1011001 \\ \hline 10110100 \\ 00000000 \\ 00000000 \\ 10110100 \\ 10110100 \\ 00000000 \\ 10110100 \\ \hline 11111010010100 \end{array}$
---	---

## **4. Códigos alfanuméricos.**

Son aquellos que sirven para representar caracteres alfanuméricos: números, letras, caracteres especiales y caracteres de control.

Todos los códigos alfanuméricos han de cumplir dos condiciones:

- Deben ser capaces de representar todos los números, letras del alfabeto y signos de puntuación.
- Deben de asignar códigos numéricos ordenados para representar cada carácter.

Entre los principales códigos de representación alfanumérica tenemos los códigos ASCII y UNICODE.

### **4.1 El código ASCII**

Es uno de los más extendidos y sus siglas significan Standar Code for Information Interchange.

Actualmente se usa el ASCII extendido, que codifica cada carácter con un byte. Seguidamente se incluye una tabla ASCII:



*Fundamentos del Hardware*  
*UTI – Conceptos básicos. Representación de la información.*

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

Estos son los códigos básicos, el resto pueden verse en la tabla que sigue

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ù	161	A1	í	193	C1	ł	225	E1	β
130	82	é	162	A2	ó	194	C2	Ť	226	E2	Γ
131	83	â	163	A3	ú	195	C3	ţ	227	E3	Π
132	84	ä	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	†	229	E5	σ
134	86	å	166	A6	ª	198	C6	ƒ	230	E6	μ
135	87	ç	167	A7	º	199	C7	‡	231	E7	τ
136	88	ê	168	A8	¿	200	C8	℔	232	E8	Φ
137	89	ë	169	A9	ƒ	201	C9	℥	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	℥	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	℥	235	EB	δ
140	8C	î	172	AC	¼	204	CC	‡	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	=	237	ED	∞
142	8E	Ä	174	AE	«	206	CE	‡	238	EE	ε
143	8F	Å	175	AF	»	207	CF	±	239	EF	Π
144	90	É	176	B0	☐	208	D0	℔	240	FO	≡
145	91	æ	177	B1	☐	209	D1	Ť	241	F1	±
146	92	Æ	178	B2	☐	210	D2	π	242	F2	≥
147	93	ó	179	B3		211	D3	℔	243	F3	≤
148	94	ö	180	B4	†	212	D4	℔	244	F4	[
149	95	ò	181	B5	‡	213	D5	ƒ	245	F5	]
150	96	û	182	B6	‡	214	D6	℥	246	F6	÷
151	97	ù	183	B7	π	215	D7	‡	247	F7	≈
152	98	ÿ	184	B8	ŕ	216	D8	‡	248	F8	°
153	99	Ö	185	B9	‡	217	D9	Ť	249	F9	•
154	9A	Û	186	BA		218	DA	ŕ	250	FA	·
155	9B	÷	187	BB	ŕ	219	DB	■	251	FB	√
156	9C	£	188	BC	ŕ	220	DC	■	252	FC	¤
157	9D	¥	189	BD	ŕ	221	DD	■	253	FD	¢
158	9E	℔	190	BE	ŕ	222	DE	■	254	FE	■
159	9F	ƒ	191	BF	ŕ	223	DF	■	255	FF	□

## 4.2 El código UNICODE

Sus siglas significan Universal Code y es el estándar más extendido en Internet.

Se caracteriza porque codifica cada carácter con 2 bytes.

Se creó por tanto para solucionar los límites de representación de los códigos anteriores, permitiendo entre otras cosas la creación de aplicaciones multilingües.

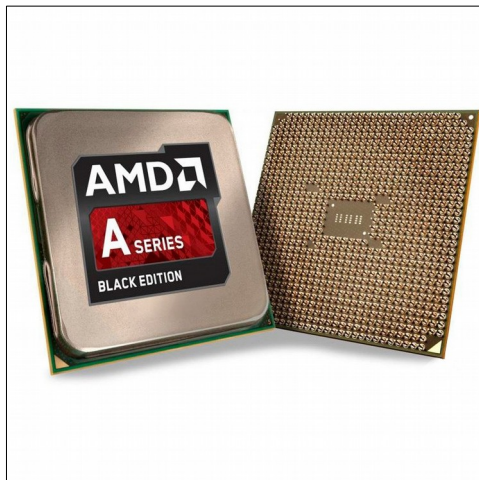
Por último hay que indicar que existen versiones de UNICODE como UTF8 que usa grupos de 8 bits para representar un subconjunto de símbolos de UNICODE.

## **5. Componentes básicos de un computador.**

En este punto, se realizará una breve introducción a los principales elementos hardware de un computador, cada uno de los cuales será explicado en mayor profundidad durante las siguientes unidades de trabajo.

En concreto, los principales elementos hardware que forman un ordenador son los siguientes:

- **El microprocesador.** Encargado de coordinar el funcionamiento del resto de elementos del sistema, para ejecutar las instrucciones de los programas que se procesan en un computador.  
Actualmente hay dos empresas que acapara el mercado de procesadores INTEL y AMD.



- **La placa base.** Es el dispositivo que permite la comunicación entre los componentes del PC, como por ejemplo, el microprocesador, la memoria, la tarjeta gráfica, etc. De hecho, todos estos elementos van soldados o conectados a la placa mediante zócalos de diferentes tipos.

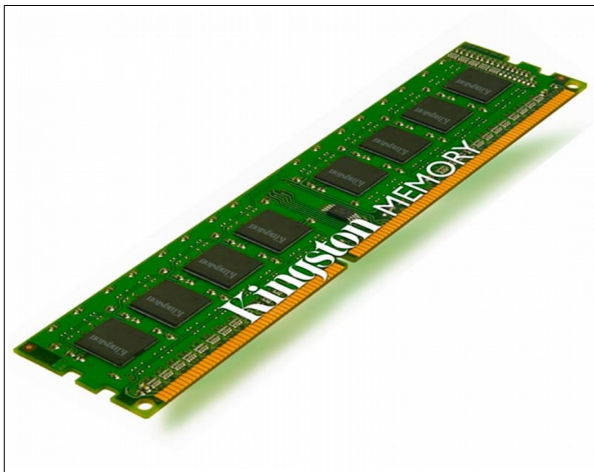


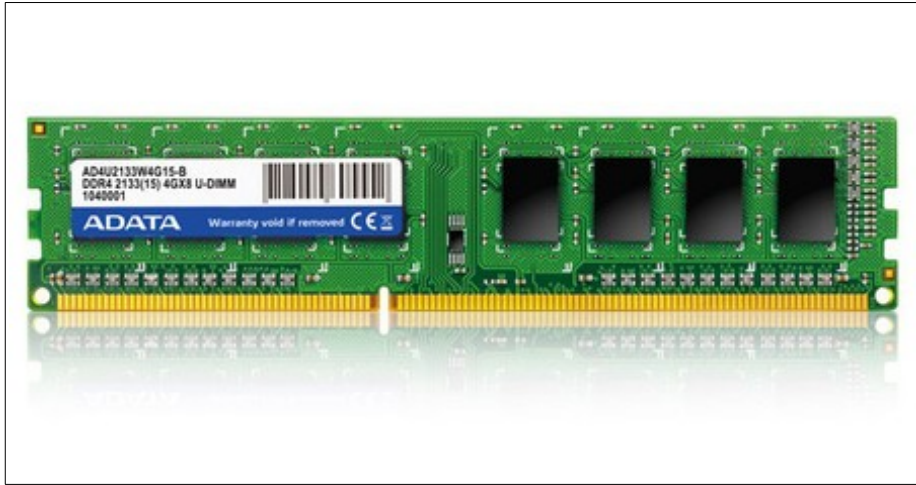
- **La memoria.** Este componente, permite almacenar los programas que se ejecutan en el computador, los datos que se procesan y los resultados de estos.

Memoria SO-DIM para portátiles



Memorias DDR3 y DDR4 para PC estandard.





- **Los dispositivos de almacenamiento.** Permiten almacenar grandes cantidades de información de forma permanente, manteniéndolos incluso al apagar el ordenador.



- **Los dispositivos periféricos de entrada/salida.** Permiten introducir o extraer información del computador. Pueden dividirse en tres grupos:
  - Periféricos de salida. Ejemplos de ellos son el monitor, la impresora, etc.
  - Periféricos de entrada. Como el teclado o el ratón.
  - Periféricos de entrada y salida. Por ejemplo una tarjeta de audio.

