



## ¿Qué es una PC?

PC son las siglas en inglés de Personal Computer, que traducido significa Computadora Personal. Hay otras que se denominan Computadoras de escritorio, que son la gama de equipos utilizados en el hogar o en las oficinas y que no son portátiles, aunque esta categoría también podría considerarse una computadora personal.

### ¿Como Funciona Mi PC?

A medida que el usuario va tomando confianza con su computadora surgen numerosas inquietudes sobre el significado de las siglas y términos utilizados en la jerga informática. Así en muchas ocasiones no sabe para qué sirven o qué representa. A continuación, intentaremos aclarar algunos de estos interrogantes.

### ¿Qué es software y qué es hardware?

Se denomina **software** a todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos, que permite al usuario realizar una tarea, y software de sistema como un sistema operativo, que permite al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones.

Probablemente la definición más formal de software es la atribuida al **Instituto de Ingenieros Eléctricos y Electrónicos**, la suma total de los programas de cómputo, procedimientos, reglas documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo. Bajo esta definición, el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente, binario o ejecutable, además de su documentación: es decir, todo lo intangible.



El término “software” fue usado por primera vez en este sentido por John W. Tukey en 1957. En las ciencias de la computación y la ingeniería de software, el software es toda la información procesada por los sistemas informáticos: programas y datos. El concepto de leer diferentes secuencias de instrucciones de la memoria de un dispositivo para controlar cálculos fue inventado por Charles Babbage como parte de su máquina diferencial. La teoría que forma la base de la mayor parte del software moderno fue propuesta por vez primera por Alan Turing en su ensayo de 1936, *Los números computables, con una aplicación al problema de decisión*.

Se denomina **hardware** o **soporte físico** al conjunto de elementos materiales que componen un ordenador. Hardware también son los componentes físicos de una computadora tales como el disco duro, CD-ROM, disquetera (floppy), etc.





En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, periféricos de todo tipo y otros elementos físicos.

El hardware se refiere a todos los componentes físicos (que se pueden tocar) de la computadora: discos, unidades de disco, monitor, teclado, ratón (mouse), impresora, placas, chips y demás periféricos. En cambio, el software es intangible, existe como ideas, conceptos, símbolos, pero no tiene sustancia. Una buena metáfora sería un libro: las páginas y la tinta son el hardware, mientras que las palabras, oraciones, párrafos y el significado del texto son el software. Una computadora sin software sería tan inútil como un libro con páginas en blanco.

## El lenguaje de la PC

### **Sistema Binario: Historia**

El antiguo matemático Indio Pingala presentó la primera descripción que se conoce de un sistema de numeración binario en el siglo tercero antes de Cristo, lo cual coincidió con su descubrimiento del concepto del número cero.

El sistema binario moderno fue documentado en su totalidad por Leibniz en el siglo XVII en su artículo "*Explication de l'Arithmétique Binaire*". Leibniz usó el 0 y el 1, al igual que el sistema de numeración binario actual.

En 1854, el matemático británico George Boole, publicó un artículo que marcó un antes y un después, detallando un sistema de lógica que terminaría denominándose Álgebra de Boole. Dicho sistema jugaría un papel fundamental en el desarrollo del sistema binario actual, particularmente en el desarrollo de circuitos electrónicos.

En 1937, Claude Shannon realizó su tesis doctoral en el MIT, en la cual implementaba el Álgebra de Boole y aritmética binaria utilizando relés y conmutadores por primera vez en la historia. Titulada *Un Análisis Simbólico de Circuitos Conmutadores y Relés*, la tesis de Shannon básicamente fundó el diseño práctico de circuitos digitales.

En noviembre de 1937, George Stibitz, trabajando por aquel entonces en los Laboratorios Bell, construyó un ordenador basado en relés - al cual apodó "Modelo K" (porque lo construyó en una cocina, en inglés "kitchen")- que utilizaba la suma binaria para realizar los cálculos. Los Laboratorios Bell autorizaron un completo programa de investigación a finales de 1938, con Stibitz al mando. El 8 de enero de 1940 terminaron el diseño de una Calculadora de Números Complejos, la cual era capaz de realizar cálculos con números complejos. En una demostración en la conferencia de la Sociedad Americana de Matemáticas, el 11 de septiembre de 1940, Stibitz logró enviar comandos de manera remota a la Calculadora de Números Complejos a través de la línea telefónica mediante un teletipo. Fue la primera máquina computadora utilizada de manera remota a través de la línea de teléfono. Algunos participantes de la conferencia que presenciaron la demostración fueron John Von Neumann, John Mauchly y Norbert Wiener, el cual escribió acerca de dicho suceso en sus diferentes tipos de memorias en la cual alcanzó diferentes logros.



## Archivo Binario

Un **Archivo binario** es un archivo informático que contiene información de cualquier tipo, codificada en forma binaria para el propósito de almacenamiento y procesamiento en ordenadores. Por ejemplo, los archivos informáticos que almacenan texto formateado o fotografías.

Muchos formatos binarios contienen partes que pueden ser interpretados como texto. Un archivo binario que *sólo* contiene información de tipo textual sin información sobre el formato del mismo se dice que es un archivo de texto plano. Habitualmente se contraponen los términos 'archivo binario' y 'archivo de texto' de forma que los primeros no contienen solamente texto.

Habitualmente se piensa en los archivos binarios como una secuencia de bytes lo que implica que los **dígitos binarios (bits)** se agrupan de ocho en ocho. Los archivos binarios contienen bytes que suelen ser interpretados como alguna otra cosa que no sean caracteres de texto. Un ejemplo típico son los programas de ordenador compilados; de hecho, las aplicaciones o programas compilados son conocidos como **binarios**, especialmente entre los programadores. Pero un archivo binario puede almacenar imágenes, sonido, versión comprimida de otros archivos, etc. En pocas palabras, cualquier tipo de información.

Algunos archivos binarios tienen una cabecera. Esta cabecera es un bloque de metadatos que un programa informático usará para interpretar correctamente la información contenida. Por ejemplo, un archivo GIF puede consistir en múltiples imágenes y la cabecera se usa para identificar y describir cada bloque de datos de cada imagen. Si el archivo binario no tiene cabecera se dice que es un **archivo binario plano**.

## Bit, lo más pequeño del lenguaje

**Bit** es el acrónimo de **Binary digit**. (dígito binario). Un bit es un dígito del sistema de numeración binario. La Real Academia Española (RAE) ha aceptado la palabra bit con el plural bits.

Mientras que en nuestro sistema de numeración decimal se usan diez dígitos, en el binario se usan sólo dos dígitos, el 0 y el 1. Un bit o dígito binario puede representar uno de esos dos valores, **0 ó 1**.

Podemos imaginarnos un bit como una bombilla que puede estar en uno de los siguientes dos estados:



El bit es la unidad mínima de información empleada en informática, en cualquier dispositivo digital, o en la teoría de la información. Con él, podemos representar dos valores cualquiera, como verdadero o falso, abierto o cerrado, blanco o negro, norte o sur, masculino o femenino, amarillo o azul, etc. Basta con asignar uno de esos valores al estado de "apagado" (0), y el otro al estado de "encendido" (1).



## Origen del termino Bit

Claude E. Shannon primero usó la palabra *bit* en un trabajo académico de 1948. Él atribuyó su origen a John W. Tukey, que había escrito una nota en los laboratorios Bell el 9 de enero de 1947 en la cual contrajo las palabras "binary digit" (dígito binario) a simplemente "bit", formando una palabra combinada. Curiosamente, Vannevar Bush había escrito en 1936 sobre los "bits de información" que podían ser almacenados en las tarjetas perforadas usadas en las computadoras mecánicas de ese tiempo.

## Combinación de Bit

Con un bit podemos representar solamente dos valores. Para representar o codificar más información en un dispositivo digital, necesitamos una mayor cantidad de bits. Si usamos dos bits, tendremos cuatro combinaciones posibles:

- **0 0** - los dos están "apagados"
- **0 1** - el primero (de derecha a izquierda) está "encendido" y el segundo "apagado"
- **1 0** - el primero (de derecha a izquierda) está "apagado" y el segundo "encendido"
- **1 1** - los dos están "encendidos"

Hay 4 combinaciones posibles con dos bits

Bit 1	Bit 0
 0	 0
 0	 1
 1	 0
 1	 1

Con estas cuatro combinaciones podemos representar hasta cuatro valores diferentes, como, por ejemplo, los colores rojo, verde, azul y negro.

A través de secuencias de bits, se puede codificar cualquier valor discreto como números, palabras, e imágenes. Cuatro bits forman un nibble, y pueden representar hasta  $2^4 = 16$  valores diferentes; ocho bits forman un octeto, y se pueden representar hasta  $2^8 = 256$  valores diferentes. En general, con n número de bits pueden representarse hasta  $2^n$  valores diferentes.

Un byte y un octeto no son la misma cosa. Mientras que un octeto siempre tiene 8 bits, un byte contiene *un número fijo de bits*, que no necesariamente son 8. En los computadores antiguos, el byte podría estar conformado por 6, 7, 8 ó 9 bits. Hoy en día, en la inmensa mayoría de las computadoras, y en la mayoría de los campos, un byte tiene 8 bits, siendo equivalente al octeto, pero hay excepciones.



## **Octeto o Byte**

Voz inglesa, se pronuncia **báit**, que si bien la Real Academia Española ha aceptado como equivalente a **octeto**, es decir a ocho bits, para fines correctos, un byte debe ser considerado como una secuencia de bits contiguos, cuyo tamaño depende del código de información o código de caracteres en que sea definido. Se usa comúnmente como unidad básica de almacenamiento de información en combinación con los prefijos de cantidad.

Los prefijos kilo, mega, giga, etc. se consideran múltiplos de 1024 en lugar de múltiplos de 1000. Esto es así porque 1024 es la potencia de 2 ( $2^{10}$ ) más cercana a 1000. Se utiliza una potencia de dos porque la computadora trabaja en un sistema binario.

Sin embargo, para el SI, los prefijos mantienen su significado usual de potencias de mil.

Así:

Nombre	Abrev.	Factor	Tamaño en SI
Kilo	<i>K</i>	1024	1000
Mega	<i>M</i>	1.048.576	1.000.000
Giga	<i>G</i>	1.073.741.824	1.000.000.000
Tera	<i>T</i>	1.099.511.627.776	1.000.000.000.000
Peta	<i>P</i>	1.125.899.906.842.624	1.000.000.000.000.000
Exa	<i>E</i>	1.152.921.504.606.846.976	1.000.000.000.000.000.000

## **Codificación del sistema Binario:**

American Standard Code for Information Interchange

El código **ASCII** (*acrónimo inglés de American Standard Code for Information Interchange — Código Estadounidense Estándar para el Intercambio de Información*), pronunciado generalmente [áski], es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales. Fue creado en 1963 por el Comité Estadounidense de Estándares (ASA, conocido desde 1969 como el Instituto Estadounidense de Estándares Nacionales, o ANSI) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía. Más tarde, en 1967, se incluyeron las minúsculas, y se redefinieron algunos códigos de control para formar el código conocido como **US-ASCII**.

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión. A menudo se llama incorrectamente ASCII a otros códigos de caracteres de 8 bits, como el estándar ISO- 8859-1 que es una extensión que utiliza 8 bits para proporcionar caracteres adicionales usados en idiomas distintos al inglés, como el español.

ASCII fue publicado como estándar por primera vez en 1967 y fue actualizado por última vez en 1986. En la actualidad define códigos para 33 caracteres no imprimibles, de los cuales la mayoría son caracteres de control obsoletos que tienen efecto sobre cómo se procesa el texto, más otros 95 caracteres imprimibles que les siguen en la numeración (empezando por el carácter espacio).

Casi todos los sistemas informáticos actuales utilizan el código ASCII o una extensión compatible para representar textos y para el control de dispositivos que manejan texto.



## **Los caracteres de control ASCII**

El código ASCII reserva los primeros 32 códigos (numerados del 0 al 31 en decimal) para caracteres de control: códigos no pensados originalmente para representar información imprimible, sino para controlar dispositivos (como impresoras) que usaban ASCII. Por ejemplo, el carácter 10 representa la función "nueva línea" (line feed), que hace que una impresora avance el papel, y el carácter 27 representa la tecla "escape" que a menudo se encuentra en la esquina superior izquierda de los teclados comunes.

El código 127 (los siete bits a uno), otro carácter especial, equivale a "suprimir" ("delete"). Aunque esta función se asemeja a otros caracteres de control, los diseñadores de ASCII idearon este código para poder "borrar" una sección de papel perforado (un medio de almacenamiento popular hasta la década de 1980) mediante la perforación de todos los agujeros posibles de una posición de carácter concreta, reemplazando cualquier información previa. Dado que el código 0 era ignorado, fue posible dejar huecos (regiones de agujeros) y más tarde hacer correcciones.

Muchos de los caracteres de control ASCII servían para marcar paquetes de datos, o para controlar protocolos de transmisión de datos (por ejemplo ENQuiry, con el significado: ¿hay alguna estación por ahí?, ACKnowledge: recibido o "acuse de recibo", Negative AcKnowledge: No recibido, Start Of Header: inicio de cabecera, Start of TeXt: inicio de texto, End of TeXt: final de texto, etc.). ESCape y SUBstitute permitían a un protocolo de comunicaciones, por ejemplo, marcar datos binarios para que contuviesen códigos con el mismo código que el carácter de protocolo, y que el receptor pudiese interpretarlos como datos en lugar de como caracteres propios del protocolo.

Los diseñadores del código ASCII idearon los caracteres de separación para su uso en sistemas de cintas magnéticas.

Dos de los caracteres de control de dispositivos, comúnmente llamados XON y XOFF generalmente ejercían funciones de caracteres de control de flujo para controlar el flujo hacia un dispositivo lento (como una impresora) desde un dispositivo rápido (como un microprocesador), de forma que los datos no saturasen la capacidad de recepción del dispositivo lento y se perdiesen.

Los primeros usuarios de ASCII adoptaron algunos de los códigos de control para representar "metainformación" como final-de-línea, principio/final de un elemento de datos, etc. Estas asignaciones a menudo entraban en conflicto, así que parte del esfuerzo de convertir datos de un formato a otro comporta hacer las conversiones correctas de metainformación. Por ejemplo, el carácter que representa el final-de-línea en ficheros de texto varía con el sistema operativo. Cuando se copian archivos de un sistema a otro, el sistema de conversión debe reconocer estos caracteres como marcas de final-de-línea y actuar en consecuencia.

Actualmente los usuarios de ASCII usan menos los caracteres de control. Los lenguajes modernos de etiquetas, los protocolos modernos de comunicación, el paso de dispositivos basados en texto a basados en gráficos, el declive de las teleimpresoras, las tarjetas perforadas y los papeles continuos han dejado obsoleta la mayoría de caracteres de control.



### Tabla de Caracteres ASCII

Como hemos visto anteriormente la tabla sirve para funciones internas de codificación, pero también el usuario puede utilizar esta tabla para poder introducir un código ASCII (Letra, Símbolo o Número) en un procesador de texto o DOS, por ejemplo la letra Ñ, suele tener problemas si se configura mal el teclado, utilizando el código ASCII, presionando la tecla ALT + el código del carácter nos da automáticamente el código en pantalla.

0	24	†	48	0	72	H	96		120	x	144	E	168	¿	192	L	216	‡	240	≡
1	25	↓	49	1	73	I	97	a	121	y	145	æ	169	ƒ	193	↳	217	‡	241	±
2	26	+	50	2	74	J	98	b	122	z	146	æ	170	ŕ	194	‡	218	ƒ	242	≥
3	27	+	51	3	75	K	99	c	123	{	147	ø	171	ŕ	195	‡	219	█	243	≤
4	28	↳	52	4	76	L	100	d	124		148	ö	172	ŕ	196	-	220	█	244	↕
5	29	"	53	5	77	M	101	e	125	}	149	ö	173	ı	197	‡	221	█	245	↕
6	30	-	54	6	78	N	102	f	126	~	150	ú	174	"	198	‡	222	█	246	+
7	31	-	55	7	79	O	103	g	127	▲	151	ù	175	"	199	‡	223	█	247	≈
8	32		56	8	80	P	104	h	128	Ç	152	ü	176	▨	200	‡	224	α	248	°
9	33	†	57	9	81	Q	105	i	129	Û	153	ó	177	▨	201	‡	225	β	249	•
10	34	"	58	:	82	R	106	j	130	é	154	Ü	178	▨	202	‡	226	Γ	250	•
11	35	#	59	:	83	S	107	k	131	ä	155	ç	179	▨	203	‡	227	Π	251	√
12	36	\$	60	<	84	T	108	l	132	ä	156	ç	180	▨	204	‡	228	Σ	252	"
13	37	%	61	=	85	U	109	m	133	ä	157	W	181	▨	205	=	229	σ	253	²
14	38	&	62	>	86	U	110	n	134	ä	158	R	182	▨	206	‡	230	μ	254	■
15	39	'	63	?	87	W	111	o	135	ç	159	f	183	▨	207	‡	231	γ	255	α
16	40	(	64	@	88	X	112	p	136	è	160	á	184	▨	208	μ	232	ø		
17	41	)	65	A	89	Y	113	q	137	è	161	í	185	▨	209	‡	233	θ		
18	42	x	66	B	90	Z	114	r	138	è	162	ó	186	▨	210	‡	234	Ω		
19	43	+	67	C	91	[	115	s	139	ÿ	163	ú	187	▨	211	‡	235	Ø		
20	44	,	68	D	92	\	116	t	140	ï	164	ñ	188	▨	212	‡	236	ω		
21	45	-	69	E	93	]	117	u	141	ï	165	Ñ	189	▨	213	‡	237	ϕ		
22	46	.	70	F	94	^	118	v	142	Ä	166	ä	190	▨	214	‡	238	€		
23	47	/	71	G	95	_	119	w	143	Ä	167	å	191	▨	215	‡	239	Π		