

SISTEMAS INFORMÁTICOS

UD 1: INTRODUCCIÓN A LOS SISTEMAS INFORMÁTICOS.

IES HLANZ

Creado por: ISAÍAS FERNÁNDEZ LOZANO

Índice:

1. Introducción.....	4
2. Sistemas Informáticos.	4
2.1. Hardware de un sistema informático.	4
2.2. Software de un sistema informático.....	5
A) Tipos de Software.	5
2.3. Clasificación de los sistemas informáticos.....	6
3. Un poco de historia.....	7
4. La información y su representación.....	13
4.1. Teorema Fundamental de la Numeración.	15
4.2. Sistemas de Numeración.....	16
A) Sistema decimal o de base 10.	16
B) Sistema Binario o de Base 2.	16
C) Sistema Octal o de Base 8.	17
D) Sistema Hexadecimal o de Base 16.....	17
4.3. Conversión entre sistemas.	18
A) Conversión de un número decimal a binario.....	18
B) Conversión de un número binario a decimal.	20
C) Conversión de un número decimal a octal.....	20
D) Conversión de un número octal a decimal.....	21
E) Conversión de un número octal a binario.....	21
F) Conversión de un número binario a octal.....	22
G) Conversión de un número decimal a hexadecimal.....	22
H) Conversión de un número hexadecimal a decimal.....	22
I) Conversión de un número hexadecimal a binario.....	23
J) Conversión de un número binario a hexadecimal.....	23
K) Otras conversiones.	23
4.4. Operaciones Binarias.....	24
A) Suma Binaria.....	24
B) Resta Binaria.....	25
4.5. Operaciones Lógicas.....	25
4.6. Medidas de la información.....	26

4.7. Otros tipos de datos.	29
A) Texto.	29
B) Imagen.	29
C) Sonido.	30
D) Vídeo.	30

1. Introducción.

Desde siempre, el ser humano ha tenido que enfrentarse a infinidad de problemas cada vez más complicados. Para ello, se ha ido rodeando de herramientas que le permitieran resolverlos, desde el primer ábaco hasta el último de los superordenadores. A medida que estas herramientas se hacían más complejas, se fue haciendo necesario un intermediario que facilitara la comunicación entre el usuario y la máquina. Y es entonces cuando aparecen los sistemas operativos.

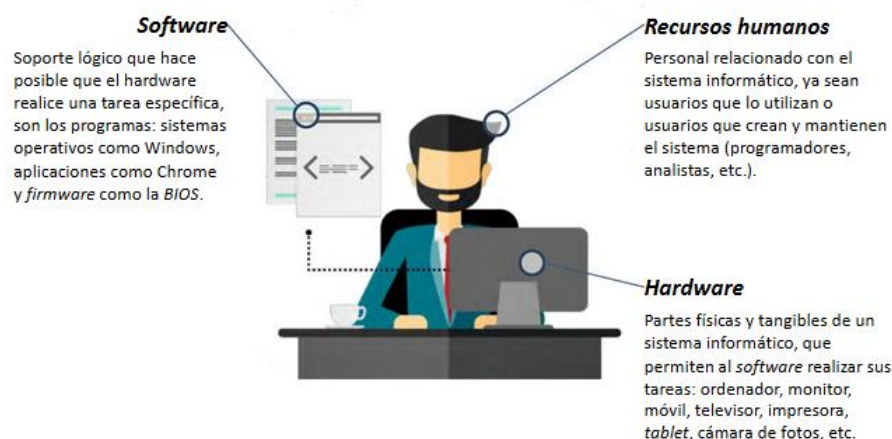
Los sistemas operativos soportan desde simples videojuegos hasta complejas aplicaciones de negocios, y son una parte esencial de la vida cotidiana. Evolucionando de forma rápida y continua, se adaptan para que el usuario pueda utilizar cualquier tipo de dispositivo.

De esta forma, es necesario que el alumno conozca qué son, cómo funcionan y con qué otros componentes se relacionan, para tener una visión global y poder trabajar con ellos de forma sencilla y eficiente.

2. Sistemas Informáticos.

Actualmente, en pleno siglo XXI, no se puede entender una sociedad civilizada sin los ordenadores, los móviles o Internet: desde comprar las entradas para un concierto hasta buscar trabajo, pasando por quedar con los amigos, escuchar música, comprar comida, etc., todas estas acciones de la vida cotidiana son inconcebibles sin los sistemas informáticos.

Un sistema informático es un conjunto de recursos formados por el *hardware*, el *software* y las personas que lo utilizan, que trabajan de forma coordinada para almacenar y procesar información con el objetivo de dar respuesta a infinidad de problemas.



2.1. Hardware de un sistema informático.

El *hardware* es una parte fundamental en un sistema informático y ha ido evolucionando de una forma muy rápida, ofreciendo prestaciones que hace unas décadas era impensable alcanzar.

Generalmente, se puede clasificar el *hardware* según la forma de conectarse:

- En el interior del ordenador:* son los componentes que están conectados a la placa base directamente: microprocesador (CPU), tarjetas (red, sonido, vídeo, etc.), memoria RAM, disco duro interno, etc.

b) *En el exterior del ordenador:* son los componentes que se conectan al ordenador desde el exterior de la carcasa, para poder comunicar el ordenador con el usuario. Se les denomina periféricos y se clasifican, a su vez, en:

- *Periféricos de entrada.* Introducen datos en el ordenador: teclado, ratón, escáner, micrófono, etc.
- *Periféricos de salida.* Muestran la información procesada por el ordenador: monitor, impresora, proyector, altavoces, etc.
- *Periféricos de entrada/salida (E/S, o en inglés, input/output).* Se pueden usar tanto para introducir datos como para mostrar información: memorias USB (*pendrives*), discos duros externos, *routers*, etc.



2.2. Software de un sistema informático.

El *software* es una parte imprescindible en un sistema informático, ya que es clave para hacer funcionar el *hardware* y ofrecer al usuario recursos para realizar las tareas de una forma cómoda y eficiente.

A) Tipos de Software.

Generalmente, el *software* se clasifica según su función dentro del sistema, de la siguiente manera:

- a) *Software base.* Es fundamental para el sistema, puesto que permite comunicar a los usuarios y la mayoría del *software* con el *hardware*, como los sistemas operativos.
- b) *Software de aplicación.* Son las aplicaciones o programas encargados de realizar tareas específicas. Se pueden diferenciar dos tipos:
 - De aplicación horizontal: comprende las aplicaciones de tipo genérico que pueden solucionar tareas de diversa índole, como los procesadores de texto, las hojas de cálculo, etc.
 - De aplicación vertical: es más específico y se crea para solventar unas necesidades determinadas del usuario o empresa, como los videojuegos.
- c) *Software de diagnóstico.* Ayuda a mantener, reparar y encontrar posibles fallos en el ordenador, como los antivirus, programas de chequeo de memoria, etc.



Existe un *software* especial denominado *firmware*, integrado en la mayoría de los componentes *hardware* de un sistema informático, que hace de traductor entre los *drivers* del sistema operativo y los circuitos electrónicos del componente.

La BIOS es uno de los *firmwares* más conocidos; se encarga de iniciar y comprobar el *hardware* del sistema para que, posteriormente, se pueda cargar un sistema operativo.

2.3. Clasificación de los sistemas informáticos.

Según los componentes *hardware* y *software* que utiliza el sistema informático para procesar la información, se pueden encontrar los siguientes tipos de sistemas informáticos: supercomputador, *mainframe*, ordenador personal y portátil, servidor, dispositivo móvil y sistema de control, cuyas características se detallan en la figura 1.5.

Supercomputador

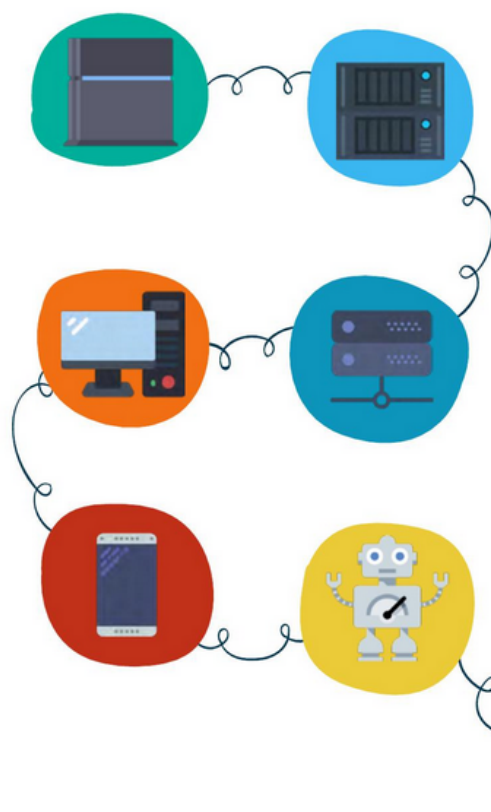
Procesa una gran cantidad de información de forma muy rápida para realizar cálculos complejos. Se emplea sobre todo para la investigación en campos como la medicina (cáncer, Alzheimer, etc.), la meteorología (tsunamis y predicción del tiempo, etc.).

Ordenador personal y portátil

(PC o *desktop* y *laptop*). Realiza tareas sencillas como navegar por Internet, consumir contenido multimedia, confeccionar documentos, etc. Además, el portátil, gracias a su batería, puede transportarse con relativa facilidad.

Dispositivo móvil

(*tablet*, *smartphone*, *smartwatch*, etc.). Pequeño ordenador con pantalla táctil y batería que, además de hacer las mismas tareas que los ordenadores (con limitaciones en cuanto a capacidad de procesamiento), permiten realizar llamadas, controlar constantes vitales, etc.



Mainframe

(computadora central). Posee una gran capacidad de procesamiento como el supercomputador, pero centrado en el manejo de operaciones de entrada y salida, proporcionando una alta fiabilidad. Se utiliza sobre todo en grandes empresas para dar soporte a una gran cantidad de usuarios.

Servidor

Proporciona servicios como compartición de datos, aplicaciones o dispositivos (como impresoras o discos duros) a través de una red a un conjunto de usuarios. Su capacidad se calcula en función de los servicios que ofrece y del número de usuarios que atiende. Se usa sobre todo para alojar sitios web y compartir recursos.

Sistemas de control

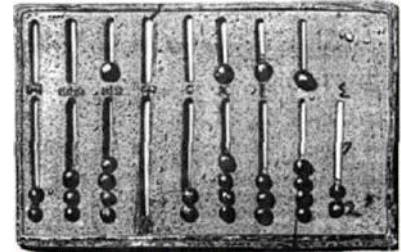
Gestionan y controlan el comportamiento de equipos o máquinas. Suelen ser de tamaño reducido y en ocasiones tienen restricciones de tiempo real. Están instalados en microondas, lavadoras, coches, cajeros, etc.

3. Un poco de historia...

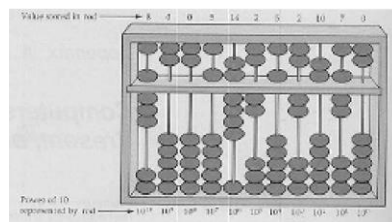
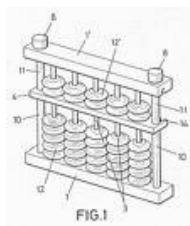
La historia del hardware se puede dividir en varias eras, dependiendo de la “máquinas” creadas y de su tecnología de construcción.

Prehistoria: Generación 0.

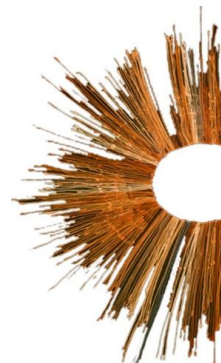
El hombre primitivo usó los diez dedos de las manos para contar (de ahí nace el sistema de numeración decimal), posteriormente se ayudó de guijarros o piedras agrupadas en montones y se sofisticó el método de contar haciendo surcos en una bandeja cubierta de arena (tablas de arena), en la que se representaban los números por piedras y sus posiciones en los surcos.



El quipu fue un sistema nemotécnico mediante cuerdas de lana o algodón y nudos, fue usado como sistema de contabilidad y una forma de escritura de los incas en el 2500 a.C.



Ábaco

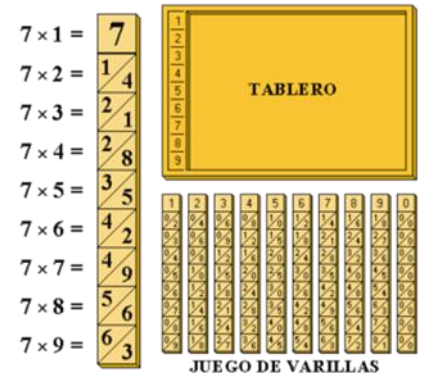


Quipu



El primer instrumento construido por la humanidad para facilitar el cálculo fue el ábaco, que simplificaba las operaciones aritméticas básicas (suma, resta, multiplicación y división). Puede ser considerado como el origen de las máquinas de calcular. A pesar de su antigüedad, entre el 2000 y 1000 a.C., se sigue usando hoy día en algunos países orientales.

El escocés John Napier (1550-1617), además de inventar los logaritmos, ideó en 1615 un sencillo instrumento mecánico denominado Varillas de Napier, con el que se podían realizar con gran facilidad multiplicaciones y divisiones.



Entre 1620 y 1630 se utilizaron las reglas de cálculo, que utilizaban los logaritmos como medio para realizar ciertos cálculos.

La Era Mecánica: Generación 1.

En la generación mecánica empezaron a aparecer diferentes tipos de máquinas capaces de realizar operaciones matemáticas sencillas, además empezaron a introducir conceptos actuales, algunas de ellas en orden cronológico son:

En 1642 el joven filósofo, matemático y físico francés **Blaise Pascal**, inventó una máquina para sumar y restar, que patentó en 1647. A dicha máquina se le dio el nombre de **Pascalina**. Tenía las dimensiones de una caja de zapatos y en su interior disponía unas ruedas dentadas conectadas entre sí que formaban una cadena de transmisión. Las ruedas representaban el sistema decimal de numeración; cada rueda constaba de diez pasos, para lo cual estaba convenientemente marcada con números del 0 al 9. El número total de ruedas ascendía a ocho, distribuidas de la siguiente manera:



- 6 ruedas para representar los números enteros, y
- 2 ruedas más en el extremo derecho para indicar dos posiciones decimales.

Con esta disposición se podía manejar números entre 0.01 y 999,999.99. Para sumar o restar, se hacía girar una manivela en el sentido apropiado, con lo que las ruedas corrían los pasos necesarios.

Aun cuando el logro de Pascal fue apreciado en toda Europa, la Pascalina fue un fracaso financiero, debido a que era Pascal la única persona que podía repararla, además, en esta época el trabajo manual en cálculos aritméticos era muchísimo más barato que la máquina.

En 1671 el filósofo y matemático alemán Gottfried Wilhelm Leibniz construyó la Calculadora Universal, una máquina que no solo sumaba y restaba, sino que también podía multiplicar y dividir. Pero el aporte más significativo de Leibniz a la ciencia de la computación es la invención del Sistema de Numeración Binario. Hoy en día, el sistema de numeración binario rige totalmente el lenguaje que hablan las computadoras actuales.

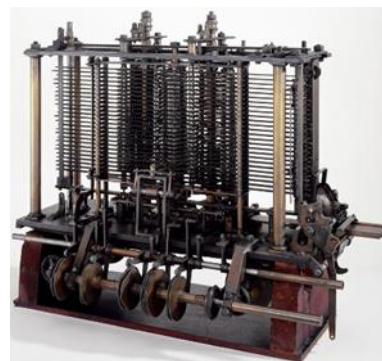


Las máquinas de Pascal y Leibniz sirvieron de base para toda una serie de mejoras que se desarrollaron a lo largo de los siglos XVIII y XIX.

Hacia 1820 se comercializó una máquina denominada Arithmometer ideada por el francés Thomas de Colmar, de gran precisión y robustez.



Leonardo da Vinci, Pascal y Leibniz construyeron las primeras máquinas, pero fue Charles **Babbage**, en **1832**, el que creó la primera capaz de encadenar varias operaciones automáticamente y que permitía resolver ecuaciones de primer grado con su **máquina diferencial**.



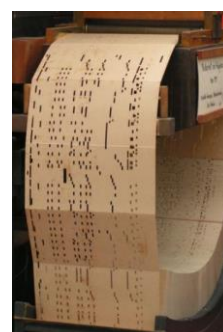
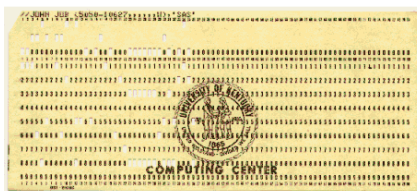
En **1833**, la **máquina analítica** de Babbage, establece los principios de funcionamiento de los ordenadores actuales con conceptos como dispositivo de Entrada-Salida (E/S), memoria, unidad de control y aritmético-lógica. Posiblemente la más importante de esta era, aunque nunca llegase a ser construida. El software lo creó **Ada Lovelace o Ada Byron** la primera programadora de la historia.

La Era Electromecánica: Generación 2.

En **1885** la compañía Brunsviga comercializó una máquina denominada **Dupla** que fue utilizada (con sucesivas versiones mejoradas) hasta finales de la década de 1950 en oficinas y empresas como ayuda en la realización de cálculos rutinarios (contabilidad, etc.). Era una versión mejorada de las calculadoras mecánicas.



Un problema grave que se le presentó al Gobierno de los EEUU al final de la década de los años 1880 era que con los medios de que disponía preveían que tardarían en realizar el censo de 1880 unos 12 años, con lo que se solaparía con el de 1890 (en EEUU el censo se hace cada 10 años). Por este motivo, el Gobierno convocó un concurso para encontrar la mejor forma de realizar censos posteriores. Así, en **1887** el Dr. Hermann Hollerith, un especialista en estadística que trabajaba en la oficina de censos, concibió la idea de usar tarjetas perforadas para mecanizar el procesamiento de datos de los censos, desarrollando así una máquina denominada **tabuladora**. Esta máquina tuvo un gran éxito e hizo posible que el censo de 1890 se efectuase en 3 años, en lugar de los doce inicialmente previstos.



El Dr. Hollerith, en 1896 fundó la Tabulating Machine Company y vendió sus productos en todo el mundo. La demanda de sus máquinas llegó hasta Rusia, en donde el primer censo ruso efectuado en 1897 se efectuó usando la Máquina Tabuladora del Dr. Hollerith.

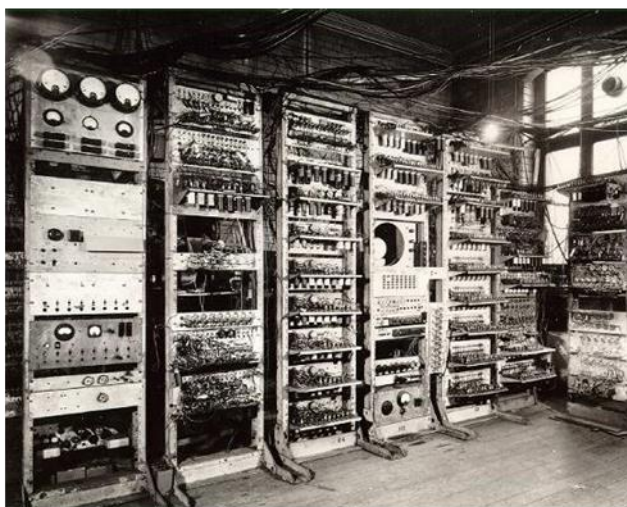
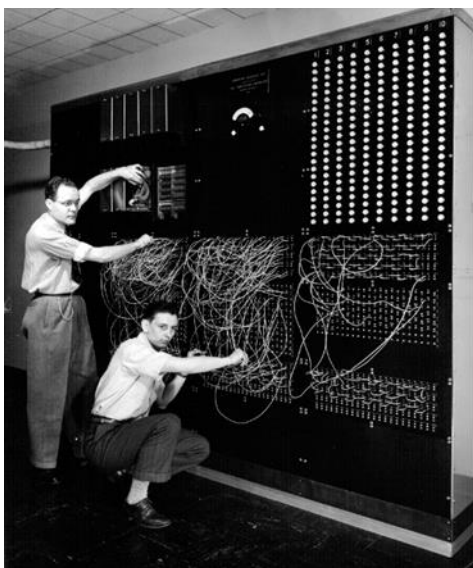
En 1911 la Tabulating Machine Company se unió con varias otras compañías para formar la Computing Tabulating Recording Company (CTR).

En 1914 es nombrado gerente general de la compañía Thomas J. Watson. Los resultados dados por la máquina tabuladora tenían que escribirse a mano antes de 1919, año en que la Computing Tabulating Recording Company presentó al mercado la Impresora / Listadora.

En 1924, Watson cambió el nombre de su compañía CTR por el de International Business Machines Corporation (IBM), a día de hoy es la vigésima empresa más grande del mundo y es la empresa en EEUU con más patentes dentro del mundo de la tecnología y tiene nueve laboratorios de investigación. La “Gran Azul” como se la conoce ha dado grandes inventos al mundo de la computación: el disquete, el disco duro, la banda magnética, el modelo relacional, el Universal Product Code, la DRAM y el sistema de inteligencia artificial Watson, entre otros.

Entre 1938 y 1941, aparece la primera computadora electromecánica y unas versiones mejoradas de esta misma. El Z1, Z2 y Z3 de Konrad Zuse, menospreciado por haberlo creado durante la Alemania Nazi y la Segunda Guerra Mundial.

En 1944, el profesor Howard H. Aiken, de la Universidad de Harvard y con el apoyo de IBM, comenzó la construcción de la primera computadora electromecánica (MARK I). Usaba tarjetas perforadas y componentes electromecánicos. Medía 15 metros de largo, pesaba 10 toneladas y tenía 800 kilómetros de cables. Fue la máquina que hizo de puente entre la era mecánica y la siguiente era... la era electrónica.

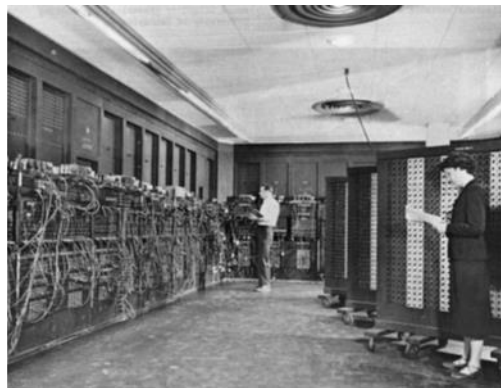


La Era Electrónica: Generación 3.

La era electrónica podemos dividirla en 5 etapas claramente diferenciadas por diferentes inventos que revolucionaron todo el sector.

1ª Generación: Válvulas de Vacío

1944: Entra en funcionamiento el prototipo, Colossus Mark I, la primera máquina electrónica programable de la historia y una sucesión de estas. Las máquinas Colossus fueron los primeros dispositivos calculadores electrónicos usados por los británicos para leer las comunicaciones cifradas alemanas durante la Segunda Guerra Mundial. Colossus fue uno de los primeros computadores digitales. Desarrollada por Thomas (Tommy) Harold Flowers con ayuda de otros como Alan Turing.



1943-1946: Se crea el ENIAC de Eckert y Mauchly, considerado el primer computador electrónico, introducen el primer elemento electrónico, hasta que se desclasificó la información del anterior

1944: El doctor John Von Neumann desarrolla la idea de programa interno o almacenado y escribe el fundamento teórico de la construcción de un computador electrónico.

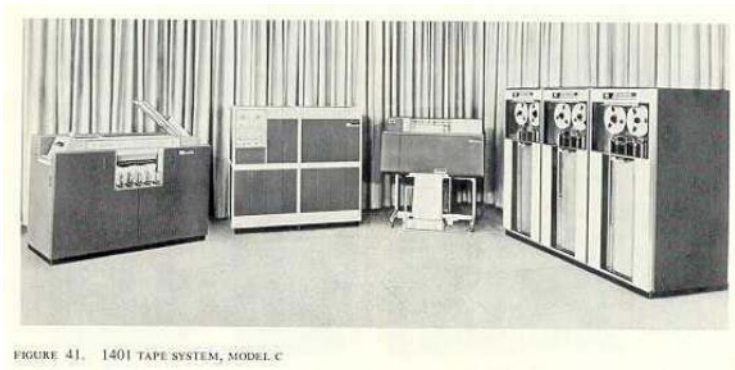
1945-1951: Se crea el EDVAC, trabaja con programas almacenados por lo demás era muy parecido al ENIAC.

1951: UNIVAC I. Considerada la primera computadora comercial en ser vendida, aunque se le adelantó la británica Ferranti Mark I por unos meses, y nunca se tuvo en cuenta la Z4 que se adelantó casi un año. Los doctores Mauchly y Eckert fundaron la compañía Universal Computer (Univac), y su primer producto fue esta máquina. El primer cliente fue la Oficina del Censo de Estados Unidos.

1952-1955: MANIAC-I, MANIAC-II, UNIVAC-II, evoluciones de los modelos anteriores, aparecen los discos duros de núcleos de ferrita.

2ª Generación: Transistores

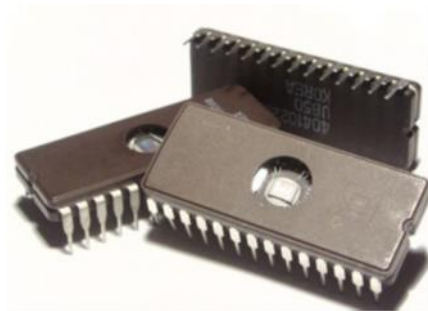
1955-1964: Aparece el IBM 1401. Se sustituye la válvula de vacío por el transistor. Los tamaños se reducen por 100, y ganaban potencia, fiabilidad y rapidez en la misma proporción. Se introducen los lenguajes de alto nivel Cobol, Fortran y Algol.



3ª Generación: Chips

1964-1974: Aparece el IBM 370. Ordenadores basados en circuitos integrados, supuso la minimización de los ordenadores.

1969: Aparecen las primeras memorias de semiconductores que vienen a sustituir a las memorias de núcleo de ferrita. Texas Instruments comienza a fabricarlos en serie.



4ª Generación: Microchips

1975-1980: Intel 4004. Presenta toda la CPU en un circuito integrado, el microprocesador. Aparecen los PCs. Gran perfeccionamiento de las unidades de almacenamiento (disquete).



5ª Generación: Circuitos integrados

1983 - 1998: Se sigue creando los mismos componentes, pero con una miniaturización de las tecnologías de construcción (VLSI, ULSI) (a día de hoy 14nm tanto en AMD como en Intel).

A principios de los años ochenta, aparecen los primeros ordenadores personales (PC) basados en el sistema operativo MS-DOS. Permiten trabajar con una gran variedad de programas, y su uso se extiende rápidamente. La información se graba en disquetes magnéticos.

Se desarrollan nuevos modelos de PC cada vez más potentes y rápidos (8088, 80286, 80386, 80486, Pentium). Avanza, progresivamente, la tecnología que permite conectar varios ordenadores entre sí y, de esta forma, intercambiar información y compartir recursos. Se generaliza el uso de un tipo particular de ordenadores, los portátiles.



Se amplían las posibilidades de los ordenadores y aparece el concepto de multimedia, que engloba todas las nuevas capacidades gráficas, de sonido, vídeo, etc. Esto trae consigo la utilización de otros sistemas de almacenamiento masivo como, por ejemplo, los CD-ROM y DVD. En 1984 se presenta el primer Macintosh (<https://www.youtube.com/watch?v=c4mDbwoG5y4>). En 1985 se crea Windows Microsoft y en 1998 se funda Google.

6ª Generación: IoT

1999-Actualidad: Arquitecturas paralelas con múltiples procesadores trabajando a la vez. Métodos de almacenamiento mejorados, capacidad medida en GB y TB. Nuevos lenguajes de programación. Grandes avances en disciplinas relacionadas con la robótica, redes neuronales, juegos. Aparición de redes sociales, ordenadores portátiles, dispositivos móviles, inteligencia artificial, etc.

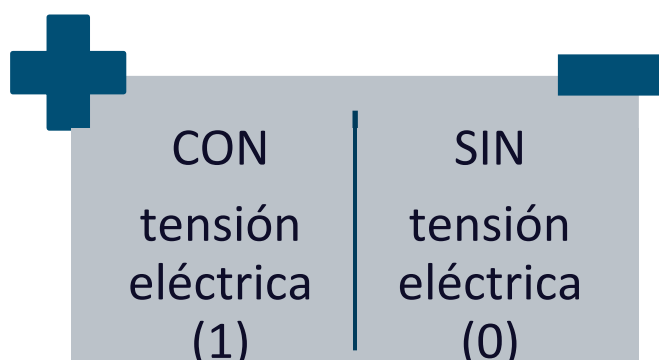
Se potencian notablemente las comunicaciones entre ordenadores; cada vez son menos los ordenadores que trabajan de forma aislada, sino que suelen conectarse con otros ordenadores para aprovechar sus recursos, bien en una red de área local (Intranet), bien a través de Internet. Aparece Wikipedia en 2001, aparecen las impresoras 3D.

4. La información y su representación.

La **transmisión de la información** entre el ser humano y la computadora puede hacerse de muchas formas, mediante caracteres alfanuméricos, sonidos, vídeos e imágenes. Esta transmisión es totalmente lógica para los seres humanos pero los computadores solo trabajan con impulsos eléctricos, por lo tanto, es necesario una codificación para toda esta infinidad de información y que sea común para que pueda interpretarse en todos los computadores.

Los seres humanos tenemos los idiomas, con ellos codificamos imágenes, valores o hechos en palabras para transmitir la información. Para que dicha información sea funcional tanto el emisor como el receptor deben de hablar el mismo idioma para poder entenderse correctamente, lo mismo pasa en los ordenadores.

Por lo tanto, internamente la representación de la información se efectúa empleando señales eléctricas con dos estados posibles (biestables) que se corresponden con los estados apagado/encendido o activado/desactivado:



A este lenguaje o código se le denomina **Código Binario**, puesto que se basa en dos símbolos, 0 y 1.

A esta correspondencia se le denomina “codificación de la información”, y al proceso inverso, “decodificación”. Todos los datos empleados por los ordenadores están codificados.

Los computadores suelen efectuar las operaciones aritméticas utilizando una representación para los datos numéricos basada en el sistema de numeración base dos (binario).

También se utilizan los sistemas de numeración octal y hexadecimal, para obtener códigos intermedios. Un número expresado en uno de estos dos códigos puede transformarse directa y fácilmente a binario y viceversa, y es por ello que se utilizan con gran frecuencia como paso intermedio en las transformaciones de decimal a binario y viceversa.

Toda la información que maneja o se almacena en un computador está representada mediante bits utilizando el sistema binario.

Un **bit** (BInary digiT) es la unidad mínima de información. Equivale a 0 ó 1.

Un **nibble** o Cuado es el conjunto de cuatro dígitos binarios (bits) o medio octeto. Su interés se debe a que cada cifra en hexadecimal (0, 1, 2,..., 9, A, B, C, D, E, F) se puede representar con un cuarteto, puesto que $2^4=16$. También el cuarteto es la base del sistema de codificación BCD.

Un **byte** es la unidad mínima inteligible y está formado por 8 bits.

Una **palabra** es la unidad mínima que maneja el procesador (8, 16, 32 ó 64 bits).

Como el equipo solo *entiende* de ceros y unos, toda la información debe convertirse en una sucesión de estos elementos, lo que se denomina sistema binario $\{0,1\}$. Sin embargo, ese idioma es muy complejo para el usuario, por lo que existen otros sistemas de numeración más cercanos:

- a) *Octal*: utiliza ocho símbolos $\{0,1,2,3,4,5,6,7\}$.
- b) *Hexadecimal*: utiliza dieciséis símbolos $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$.
- c) *Decimal*: utiliza diez símbolos $\{0,1,2,3,4,5,6,7,8,9\}$.

Equivalencia entre sistemas de numeración

Decimal	Binario	Octal	Hexadecimal
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10

RECUERDA

- ✓ Un sistema de numeración es un conjunto de símbolos y reglas que permite construir todos los números válidos.

4.1. Teorema Fundamental de la Numeración.

Con el fin de poder convertir cualquier número al sistema binario se utiliza el Teorema Fundamental de la Numeración, que se basa en los sistemas de numeración posicionales, donde se contempla el valor relativo de la cifra dentro del número.

Por ejemplo, en el número, 121 en base 10, el primer uno indica las centenas, mientras que el último son las unidades. Siguiendo este esquema, el anterior número podría expresarse de la siguiente forma: $121_{10} = 1 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 = 100 + 20 + 1$.

El TFN o Teorema Fundamental de la Numeración nos permite expresar en el sistema decimal cualquier cantidad expresada en otro sistema de numeración. Viene dado por la siguiente fórmula:

Si el número es $d_{(n-1)} \dots d_2 d_1 d_0, d_{-1} d_{-2} \dots d_{-k}$ entonces el número en base b es:

$$N = \sum_{i=-k}^n (\text{dígito})_i \cdot (\text{base})^i$$

Donde:

b → es la base del sistema de numeración al que la queremos pasar.

n → es el número de dígitos de la parte entera.

k → el número de dígitos de la parte decimal.

Así, para obtener el valor en decimal del número $112,02_{(3)}$, desarrollamos la fórmula como sigue: $112,02_{(3)} \rightarrow x_{(10)}$

Aplicando el TFN, la sumatoria será: $(1 \cdot 3^2) + (1 \cdot 3^1) + (2 \cdot 3^0) + (0 \cdot 3^{-1}) + (2 \cdot 3^{-2}) \rightarrow 14,2222_{(10)}$

4.2. Sistemas de Numeración.

A) Sistema decimal o de base 10.

Es el sistema numérico usado normalmente por el ser humano. Su base es 10 y utiliza los símbolos o dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Normalmente, no se suele aplicar el teorema de la numeración al sistema decimal ya que el resultado es el propio número.

Los múltiplos y submúltiplos se representan como potencias de 10, por ejemplo: $1000=10^3=1K$ o $0.1=10^{-1}$.

Por ejemplo, el número decimal 3.278,52 puede obtenerse como la suma de:

$$\begin{array}{r} 3000 \\ 200 \\ 70 \\ + 8 \\ 0,5 \\ 0,02 \\ \hline 3.278,52 \end{array}$$

Es decir, se verifica: $3.278,52 = 3 \cdot 10^3 + 2 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0 + 5 \cdot 10^{-1} + 2 \cdot 10^{-2}$

B) Sistema Binario o de Base 2.

El **sistema binario** es el que utiliza el hardware del ordenador ya que sólo usa dos símbolos para la representación de cualquier número, {0, 1}. Por lo tanto, su base es 2.

Aplicando el teorema fundamental de la numeración, el número $1001_{(2)}$ será: $9_{(10)}$

Con 3 bits se pueden representar 2^3 enteros binarios que se corresponden con los números decimales del 0 al 7.

BINARIO	DECIMAL
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

La cantidad de números que se pueden representar en binario depende, como ya se ha podido ver, del número de cifras binarias o bits que utilicemos, de forma que si utilizamos un bit podremos representar solamente dos números; con dos bits \rightarrow 4 números; con 3 bits \rightarrow 8 números; con 4 bits \rightarrow 16 números diferentes (entre el 0 y el 15 en decimal).

Así, para saber la cantidad de números distintos que se pueden representar tenemos la siguiente fórmula:

$$\text{Números que podemos representar} = 2^{n^{\circ} \text{ de bits que utilicemos}}$$

C) Sistema Octal o de Base 8.

El sistema octal utiliza los símbolos {0, 1, 2, 3, 4, 5, 6, 7}.

El número 1001_8 será, en decimal:

$$N = (1 \cdot 8^3) + (0 \cdot 8^2) + (0 \cdot 8^1) + (1 \cdot 8^0)$$

$$N = 512 + 0 + 0 + 1 = 513$$

En la siguiente tabla se muestran los números enteros binarios que se pueden formar con 3 bits, que se corresponden con los números octales del 0 al 7:

OCTAL	BINARIO
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

D) Sistema Hexadecimal o de Base 16.

El sistema hexadecimal utiliza 16 símbolos {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(=10), B(=11), C(=12), D(=13), E(=14) y F(=15)}.

Debido a que la conversión entre binario y hexadecimal, y viceversa, es directa (como veremos en el apartado siguiente), se utiliza mucho en el software informático, como el octal.

El número $2CA_{16}$ es:

$$N=(2 \cdot 16^2)+(C \cdot 16^1)+(A \cdot 16^0)$$

$$N=(2 \cdot 16^2)+(12 \cdot 16^1)+(10 \cdot 16^0)$$

$$N=512+192+10=714$$

La siguiente tabla muestra la correspondencia entre los números binarios y hexadecimales.

HEXADECIMAL	BINARIO
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

4.3. Conversión entre sistemas.

Vamos a explicar solamente la conversión de números enteros (excepto en binario), ya que para el tratamiento de los decimales suelen usarse métodos más complejos que los que vamos a describir aquí.

- **Otro sistema → decimal:** Se aplica el teorema fundamental de la numeración.
- **Decimal → otro sistema:** Se realiza mediante divisiones sucesivas usando como divisor la base a la que queremos convertir el número. Se toma para la siguiente división sólo la parte entera del cociente y se van guardando los restos obtenidos. Cuando el último cociente es menor que la base, se toma éste como primer dígito y a continuación de éste (a la derecha) se van situando los restos obtenidos en orden inverso.

A) Conversión de un número decimal a binario.

Para transformar un número de decimal a binario hacemos:

- La parte entera del nuevo número (binario) se obtiene dividiendo por 2 (sin obtener decimales en el cociente) la parte entera del número decimal de partida, y los cocientes que sucesivamente se vayan obteniendo.
- Los restos de estas divisiones y el último cociente (que serán siempre 0s o 1s) son las cifras binarias. El último cociente será el bit más significativo y el primer resto el bit menos significativo.
- 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. El número binario se forma con las partes enteras (que serán 0s o 1s) de los productos obtenidos.

Ejemplo 1: $28_{(10)} \rightarrow ?_{(2)}$

$$\begin{array}{r}
 28 \div 2 = 14 \text{ resto } 0 \\
 14 \div 2 = 7 \text{ resto } 0 \\
 7 \div 2 = 3 \text{ resto } 1 \\
 3 \div 2 = 1 \text{ resto } 1 \\
 1 \div 2 = 0 \text{ resto } 1
 \end{array}$$

$28 = 11100_2$

Ejemplo 2: $42,375_{(10)} \rightarrow ?_{(2)}$

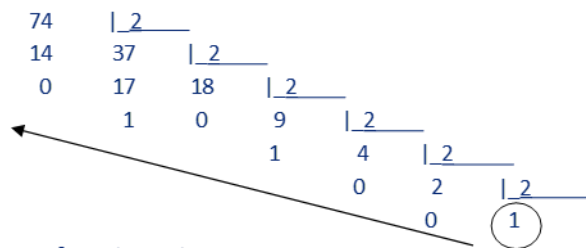
Esta conversión es de un número entero, para convertir una fracción decimal a binario se multiplicará sucesivamente la parte fraccionaria (la que está a la derecha de la coma) por 2 hasta que dé como resultado.

$ \begin{array}{r} 42 \div 2 = 21 \text{ resto } 0 \\ 21 \div 2 = 10 \text{ resto } 1 \\ 10 \div 2 = 5 \text{ resto } 0 \\ 5 \div 2 = 2 \text{ resto } 1 \\ 2 \div 2 = 1 \text{ resto } 0 \\ 1 \div 2 = 0 \text{ resto } 1 \end{array} $ <p>Arriba: LSB Abajo: MSB</p>	$ \begin{array}{l} 0.375 \times 2 = 0.75 \\ 0.75 \times 2 = 1.50 \\ 0.50 \times 2 = 1.00 \end{array} $ <p>Resultados fraccionarios: .011</p>
--	--

Resultado: 101010.011

Ejemplo 3: $74,423_{(10)} \rightarrow ?_{(2)}$

a) Parte entera:



b) Parte fraccionaria:

0,423	0,846	0,692	0,384	0,768
<u>x 2</u>	<u>x 2</u>	<u>x 2</u>	<u>x 2</u>	<u>x 2</u>
0,846	1,692	1,384	0,768	1,536

Con lo cual: $74,423_{(10)} = 1001010,01101_{(2)}$

B) Conversión de un número binario a decimal.

Para pasar un número que esté escrito en base 2 a base decimal utilizaremos el teorema fundamental de la numeración. Dicho teorema nos dice que cada dígito del número se multiplica por la base elevada a la posición que ocupa dicho dígito en el número.

$$11010_{(2)} = 26_{(10)}$$

$$1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 8 + 0 + 2 + 0 = 26$$

$$11010,011_{(2)} = 26,375_{(10)}$$

$$1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 16 + 8 + 2 + 0,25 + 0,125 = 26,375_{(10)}$$

****TRUCO****

Indicamos los exponentes sobre los dígitos y sólo hay que sumar los que tienen 1

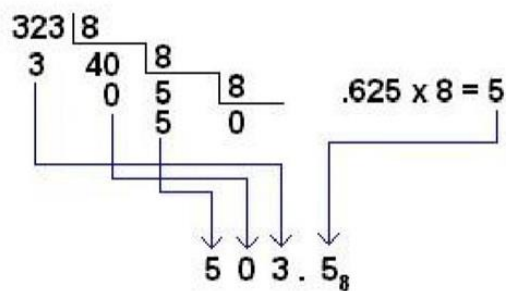
7	6	5	4	3	2	1	0	exponentes
1	1	0	0	1	0	1	1	

$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 = 128 + 64 + 8 + 2 + 1 = 203$$

C) Conversión de un número decimal a octal.

La forma para convertir un número o un número fraccionario de base decimal a octal, es exactamente igual que en binario, pero cambiando la base que en octal es 8.

Convertir 323.625 a octal



D) Conversión de un número octal a decimal.

Se aplica el TFN, por el que se va multiplicando cada dígito octal por la base (8) elevado a la posición que ocupa el dígito en el número y se suman los resultados.

Ejemplo:

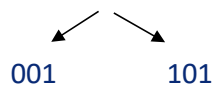
Transformamos el número 47₈ a decimal: $47_8 = 4 \cdot 8^1 + 7 \cdot 8^0 = 39_{10}$

E) Conversión de un número octal a binario.

Se pasa convirtiendo individualmente a binario (tres bits) cada cifra octal manteniendo el orden del número original. Se utiliza la tabla de conversión.

Ejemplo:

Transformamos 15₈ a binario: 1 5



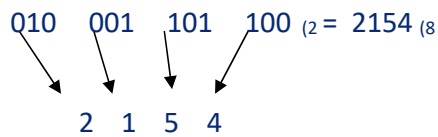
Con lo cual: $15_8 = 001\ 101_2 = 1101_2$

Otro ejemplo: $537_8 = 101\ 011\ 111_2$

OCTAL	BINARIO
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

F) Conversión de un número binario a octal.

Se forman grupos de tres cifras binarias de derecha a izquierda y se convierte a octal cada grupo individual

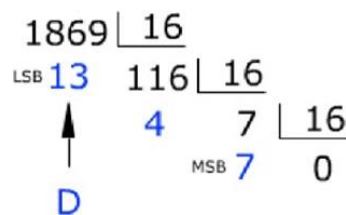


OCTAL	BINARIO
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

G) Conversión de un número decimal a hexadecimal.

Lo primero es aclarar que el sistema hexadecimal o base 16 tiene una particularidad y es que utiliza símbolos pertenecientes a letras para expresar sus cantidades más altas, por lo tanto, para las primeras diez es igual que en el sistema decimal (0-9), y para el resto de los elementos usa letras (A-F). Estas letras tienen un valor equivalente en el sistema decimal (10-15).

El procedimiento para convertir un número de decimal a hexadecimal es exactamente igual al procedimiento en octal y en binario.



Hay que tener cuidado a la hora de representar el número en hexadecimal puesto que hay que cambiar las cantidades más grandes por las letras determinadas, en este ejemplo sería 74D.

H) Conversión de un número hexadecimal a decimal.

Se hace igual que en el caso octal o binario, es decir, se va multiplicando cada dígito hexadecimal por la base (16) elevado a la posición que ocupa el dígito en el número y se suman los resultados.

Ejemplo:

Transformamos el número 3F1_(H) a decimal:

$$3F1_{(H)} = 3 \cdot 16^2 + 15 \cdot 16^1 + 1 \cdot 16^0 = 1009_{(10)}$$

I) Conversión de un número hexadecimal a binario.

Se pasa convirtiendo individualmente a binario (cuatro bits) cada cifra hexadecimal manteniendo el orden del número original.

$$\begin{array}{c} 7 \text{ F } \text{B} \text{ C } 2 \\ \swarrow \quad \downarrow \quad \swarrow \quad \searrow \quad \searrow \\ 7\text{FBC}2_{(16)} = 0111 \ 1111 \ 1011 \ 1100 \ 0010_{(2)} \end{array}$$
$$7\text{FB},\text{C}2_{(16)} = 0111 \ 1111 \ 1011 \ , \ 1100 \ 0010_{(2)}$$

J) Conversión de un número binario a hexadecimal.

Se forman grupos de cuatro cifras binarias de derecha a izquierda y se convierte a hexadecimal cada grupo individual.

HEX	BINARIO	HEX	BINARIO
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

$$\begin{array}{c} 0010 \ 0101 \ 1101 \ , \ 1111_{(2)} = 25\text{D}.\text{F}_{(16)} \\ \swarrow \quad \downarrow \quad \downarrow \quad \swarrow \\ 2 \ 5 \ \text{D} \ . \ \text{F} \end{array}$$

K) Otras conversiones.

Otra forma menos ortodoxa de cambiar diferentes cantidades de una base a otras es mediante el uso de la tabla de equivalencias.

Tabla de equivalencias entre sistemas decimal, binario, octal y hexadecimal.

<u>DECIMAL</u>	<u>BINARIO</u>	<u>HEXADECIMAL</u>	<u>OCTAL</u>
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Los cambios de base de Octal y Hexadecimal a binario se pueden realizar directamente utilizando la tabla de equivalencias anteriormente mostrada.

$100100_2 \rightarrow 44_8$ En Octal se cogen los bits de 3 en 3.

$00100100_2 \rightarrow 24_{16}$ En Hexadecimal se cogen los bits de 4 en 4 si faltan se completa con ceros a la izquierda.

4.4. Operaciones Binarias.

A) Suma Binaria.

Al igual que con el sistema decimal, en el sistema binario podemos realizar las operaciones aritméticas. La suma binaria es parecida a la suma en decimal, con la diferencia de que se manejan solo dos dígitos, el 0 y el 1. Si el resultado de la suma excede de 1, se agrega un acarreo a la suma parcial siguiente. Para realizar sumas nos fijaremos en la tabla de sumar.

SUMA BINARIA			
$0 + 0 = 0$	$0 + 1 = 1$	$1 + 0 = 1$	$1 + 1 = 0, \text{acarreo } 1$

Ejemplo de la suma entre el número $111 + 01110$ en binario

acarreo \longrightarrow

$$\begin{array}{r}
 \begin{array}{ccccccc}
 & 1 & 1 & 1 & & & \\
 & \swarrow & \swarrow & \swarrow & & & \\
 1 & 0 & 1 & 1 & 1 & 0 & \\
 + & 0 & 1 & 1 & 1 & 0 & \\
 \hline
 1 & 0 & 0 & 1 & 0 & 0 &
 \end{array}
 \end{array}$$

B) Resta Binaria.

Al igual que para la suma para la resta utilizaremos una tabla, pero distinta, la diferencia básicamente es que el acarreo se suma al siguiente sustraendo.

RESTA BINARIA

0 - 0 = 0	0 - 1 = 1, acarreo 1*	1 - 0 = 1	1 - 1 = 0
-----------	-----------------------	-----------	-----------

*El acarreo en la resta se suma al sustraendo.

Ejemplo 1:

Resta los números 11011 y 1101 en binario.

$$\begin{array}{r} 11011 \\ - 1101 \\ \hline 1110 \end{array}$$

minuendo
sustraendo

La resta de dos números binarios puede obtenerse sumando al minuendo el complemento a dos del sustraendo. Se utiliza porque la unidad aritmecológica no resta números binarios, suma binarios negativos, por eso esta conversión al negativo.

Ejemplo 2:

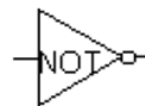
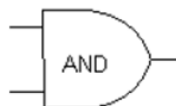
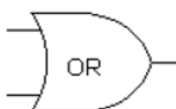
Restar los siguientes números binarios: 010011 y 110000

110000 \rightarrow C1 \rightarrow 001111 \rightarrow C2 \rightarrow 010000

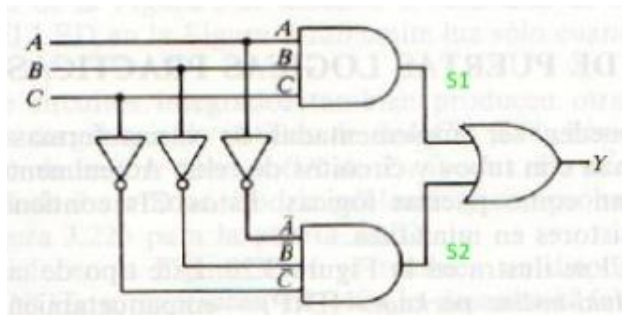
$$\begin{array}{r} 010011 \\ + 010000 \\ \hline 100011 \end{array}$$

4.5. Operaciones Lógicas.

OR			AND			XOR			NOT	
A	B	A OR B	A	B	A AND B	A	B	A XOR B	A	NOT A
0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	1	1	1	0
1	0	1	1	0	0	1	0	1		
1	1	1	1	1	1	1	1	0		



Ejemplo: Tabla de verdad del siguiente circuito



ENTRADAS			SALIDA		
A	B	C	S1	S2	Y
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	0	1

EJERCICIO: Comprueba los resultados implementando el circuito en: <https://logic.ly/demo/> en su versión online.

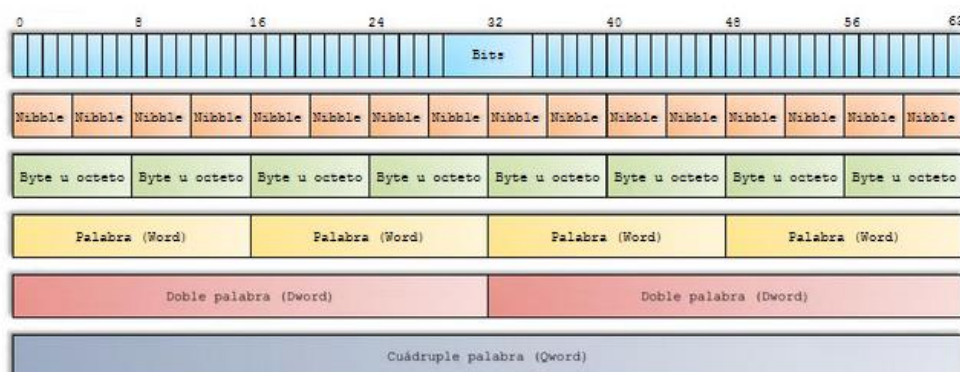
4.6. Medidas de la información.

Continuamente se están midiendo elementos: ¿cuánto lloverá mañana?, ¿cuánto vale este ordenador?, o ¿cuánto corre este coche? Cada respuesta implica una unidad de medida diferente, en estos ejemplos, l/m², € o km/h.

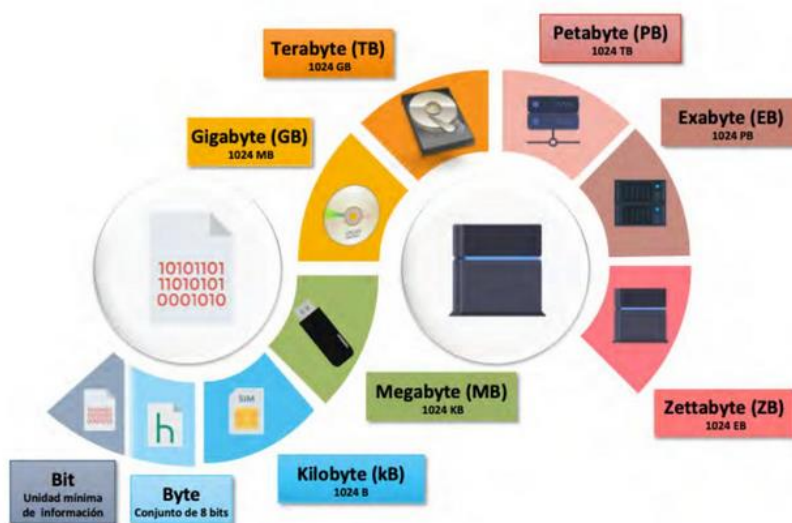
En informática, las medidas de información se utilizan para calcular la cantidad de información que se puede procesar, almacenar, enviar o recibir en los dispositivos electrónicos.

La unidad mínima de información que se utiliza en informática es el bit (dígito binario o *binary digit*), que equivale a cada uno de los dos estados comprensibles por una máquina: apagado (0) y encendido (1).

Además, los grupos de bits, dependiendo de su longitud, reciben un nombre específico:



Así como cuando se quiere medir una distancia se utiliza el metro con sus múltiplos como el kilómetro o submúltiplos como el milímetro, cuando se trabaja con información se utilizan el byte y sus múltiplos:



Nombre Decimal	Sistema Decimal	Nombre Binario	Sistema Binario
bit	0 - 1	bit	0 - 1
Byte	$10^0 = 1 \text{ B}$	Byte	$2^0 = 1 \text{ B (8bits)}$
1 Kilobyte (KB)	$10^3 = 1000 \text{ B}$	Kibibyte	$2^{10} = 1024 \text{ B}$
1 Megabyte (MB)	$10^6 = 1000 \text{ KB}$	Mebibyte	$2^{20} = 1024 \text{ KiB}$
1 Gigabyte (GB)	$10^9 = 1000 \text{ MB}$	Gibibyte	$2^{30} = 1024 \text{ MiB}$
1 Terabyte (TB)	$10^{12} = 1000 \text{ GB}$	Tebibyte	$2^{40} = 1024 \text{ GiB}$
1 Petabyte (PB)	$10^{15} = 1000 \text{ TB}$	Pebibyte	$2^{50} = 1024 \text{ TiB}$
1 Exabyte (EB)	$10^{18} = 1000 \text{ PB}$	Exbibyte	$2^{60} = 1024 \text{ PiB}$
1 Zettabyte (ZB)	$10^{21} = 1000 \text{ EB}$	Zebibyte	$2^{70} = 1024 \text{ EiB}$
1 Yottabyte (YB)	$10^{24} = 1000 \text{ ZB}$	Yobibyte	$2^{80} = 1024 \text{ ZiB}$

Para operar con las diferentes unidades de medida es necesario dividir (múltiplos) o multiplicar (submúltiplos) por $1024 = 2^{10}$.

Generalmente se utiliza una 'b' para indicar bit y una 'B' para indicar byte.



PARA SABER MÁS

¿Por qué si la velocidad de bajada de Internet es de 1 giga por segundo se tarda más de dos segundos en descargar un vídeo de 2 GB? Esto es debido a que la velocidad de transmisión, generalmente, se mide en bits y no en bytes, por lo que, si la velocidad de bajada es de 1 Gb/s, se tardará aproximadamente 16 segundos en descargar el vídeo.

CAMBIO DE UNIDAD DE MEDIDA BINARIA (en informática encontraremos el nombre decimal aunque haga referencia a la unidad binaria, aunque no es lo recomendado)

a. CAMBIO A UNA UNIDAD MAYOR

b → B → KB → MB → GB → TB → ...
 /8 /1024 /1024 /1024 /1024 /1024

Ejemplo

¿Cuántos MB son 5678990 bits?

$5678990 \text{ bits} / 8 = 709873,75 \text{ B} / 1024 = 693,236 \text{ KB} / 1024 = 0,68 \text{ MB}$

b. CAMBIO A UNA UNIDAD MENOR

b ← B ← KB ← MB ← GB ← TB ← ...
 x8 x1024 x1024 x1024 x1024 x1024

Ejemplo

¿Cuántos bits son 2 MB?

$2 \text{ MB} \times 1024 = 2048 \text{ KB} \times 1024 = 2097152 \text{ B} \times 8 = 16777216 \text{ bits}$

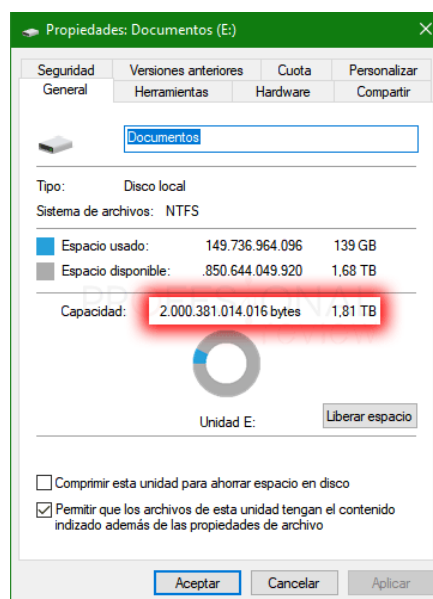
¿Por qué mi disco duro tiene menos capacidad de la que he comprado?

Las capacidades de almacenamiento en el sistema internacional son más pequeñas que las que se representan en binario. Y seguramente también nos hayamos percatado de que los discos duros, absolutamente siempre que compramos uno vienen con menos capacidad de la que en un principio prometen. Pero ¿Es esto cierto?

Lo que ocurre es que los discos duros se comercializan en términos de capacidad decimal según el sistema internacional, entonces **un Gigabyte equivale a 1.000.000.000 Bytes**. Y los sistemas operativos como Windows utilizan el sistema de numeración binario para representar estas cifras, que como hemos visto, difieren mientras mayor capacidad tengamos. Si tenemos en cuenta esto y nos dirigimos a ver las propiedades de nuestro disco duro, nos podríamos encontrar con la siguiente información:

Nosotros **hemos comprado un disco duro de 2TB**, entonces, **¿por qué solamente tenemos 1,81TB disponibles?**

Para dar la respuesta tendremos que hacer la conversión entre un sistema y otro. Si la cantidad la representamos en bytes debemos de coger el equivalente del sistema de numeración correspondiente. Entonces:



Capacidad en sistema decimal / Capacidad sistema binario
 $2.000.381.014.016 / 1.099.511.627.776 = 1,81 \text{ TB}$

Es decir, realmente nuestro disco duro tiene 2TB, pero en términos del sistema internacional, no en el sistema binario. **Windows nos lo da en términos del sistema binario** y es precisamente por este motivo por el que vemos

Para tener un disco duro de 2TB y que lo viéramos así. **Nuestro disco duro debería de ser de: $(2 \times 1.099.511.627.776) / 2.000.000.000.000 = 2,19 \text{ TB}$**

4.7. Otros tipos de datos.

Los siguientes tipos de datos se basan en el tipo de dato numérico:

A) Texto.

La representación de caracteres alfanuméricos se realiza mediante una tabla de códigos en la que se asigna a cada carácter un código numérico. Además de los símbolos gráficos, se incluyen símbolos de control como nueva línea, escape, etc. Existen diversos tipos:

- *ASCII*: utiliza 7 bits para codificar cada carácter, pensado para el inglés.
- *Latín X*: extensión del *ASCII*, emplea 8 bits y dependiendo del valor de 'X' (de 1 a 9) incluye las lenguas de Europa occidental, otras lenguas como el hebreo o el árabe o modificaciones para incluir, por ejemplo, el símbolo del euro.
- *Unicode*: código de 16 bits que engloba a todas las codificaciones e idiomas conocidos.

"La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación no"																																	
-- Albert Einstein																																	
text to ASCII numbers																																	
013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	013	010	034			
076	097	032	105	109	097	103	105	110	097	099	105	195	179	110	032	101	115	032	109	195	161	115	032	105	109	112	111	114	116	097	110	116	101
115	032	109	195	161	115	032	105	109	112	111	114	116	097	110	116	101	032	113	117	101	032	101	108	032	099	111	110	111	099	105	109	105	101
110	116	111	046	032	069	108	032	099	111	110	111	099	105	109	105	101	110	116	111	032	101	115	032	108	105	109	105	116	097	100	111	044	032
109	105	101	110	116	114	097	115	032	113	117	101	032	108	097	032	105	109	097	103	105	110	097	099	105	195	179	110	032	110	111	034	013	010
032	032	032	032	032	045	045	032	065	108	098	101	114	116	032	069	105																	

B) Imagen.

Es necesario un proceso de digitalización, compuesto por dos fases:

1. *Muestreo*: divide la imagen analógica en porciones de un tamaño determinado que poseen un valor determinado dentro de una escala (RGB, escala de grises, blanco y negro).
2. *Cuantificación*: convierte el valor de las porciones anteriores con un número determinado de bits.

Finalmente, se obtiene una imagen digital, que no es más que una matriz numérica.

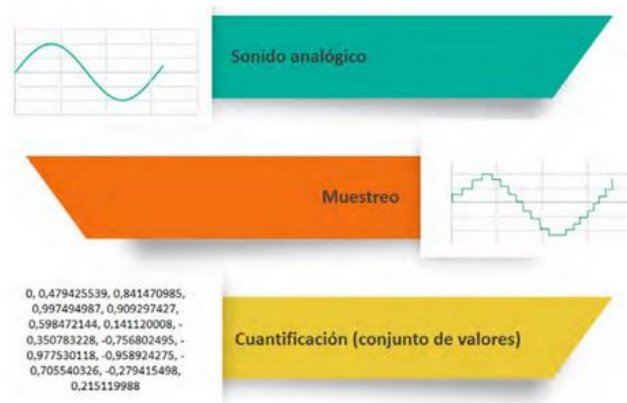


C) Sonido.

Al igual que con las imágenes, es necesario un proceso de digitalización, compuesto por dos fases:

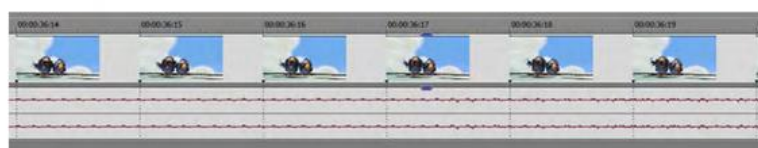
1. *Muestreo*: almacena la señal del sonido a intervalos regulares de tiempo. La frecuencia de muestreo indica el número de muestras del sonido que se toman por segundo. Se mide en hercios (Hz), por lo que $3000 \text{ Hz} = 3 \text{ KHz} = 3000 \text{ muestras por segundo}$.
2. *Cuantificación*: se encarga de convertir la señal almacenada anteriormente con un número determinado de bits.

Por lo general, el sonido puede estar en mono (1 canal) o estéreo (dos canales). Finalmente se obtiene un sonido digital, que no es más que un conjunto de valores de una onda almacenados a intervalos regulares de tiempo.



D) Vídeo.

Se puede tratar como una secuencia de imágenes y sonido en el tiempo. El ritmo de imágenes por segundo se denomina *frames* por segundo (fps).





- ✓ La cantidad de espacio que ocupa una imagen viene determinada por los bits para cada color \times resolución horizontal \times resolución vertical. Por ejemplo, una imagen de 1024×768 y 32 bits de profundidad de color ocupa 24 MB.
 - ✓ El tamaño de un sonido almacenado podría venir dado por el número de canales \times calidad de muestreo \times frecuencia (Hz) \times duración (seg.). Por ejemplo, un sonido estéreo de 32 bits de calidad con una frecuencia de 44,1 KHz durante 30 segundos ocupa 80,75 MB.
 - ✓ Para calcular el tamaño de un vídeo es necesario realizar los siguientes cálculos: fps \times duración vídeo (seg.) \times tamaño una imagen \times tamaño sonido. Por ejemplo, un vídeo de 30 segundos grabado a una resolución de 1024×768 y 32 bits de profundidad de color a 30 fps con sonido estéreo de 32 bits de calidad con frecuencia de 44,1 KHz ocupa 21,17 GB.
 - ✓ Generalmente, el sistema codifica el texto, las imágenes, el sonido y el vídeo para que ocupen menos espacio al almacenarlos utilizando algoritmos de compresión.
-