SISTEMAS INFORMÁTICOS

UD 1: INTRODUCCIÓN A LOS SI. COMPONENTES FÍSICOS

CURSO 23/24

IES P. H. LANZ

Parte 1. Introducción a los SI y Representación de Información

1. Introducción a los Sistemas Informáticos (SI)

Sistemas Informáticos: un SI se puede definir como una serie de elementos físicos (Hardware) y otros lógicos (Software) capaz de realizar muchas tareas a gran velocidad y con gran precisión, automatizando el procesamiento de la información.

El **software**: es un conjunto de órdenes o instrucciones que hacen que los componentes físicos realicen tareas determinadas, estas instrucciones ordenadas y agrupadas de forma adecuada constituyen los programas y el conjunto de estos últimos se les denomina aplicaciones informáticas.

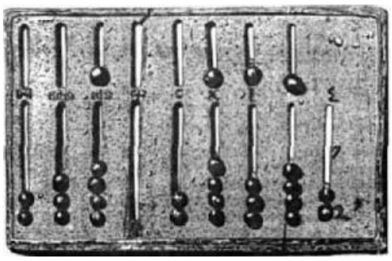
El **hardware** o parte física de un equipo informático representa los elementos tangibles que lo componen, es decir todas aquellas piezas electrónicas con las cuales montamos un computador. En la actualidad existen multitud de piezas para los computadores y no todas ellas son necesarias en todos los sistemas informáticos. Las piezas que lo componen dependerán sobre todo de la finalidad para la cual se haya instalado dicho sistema.

1.1. 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.

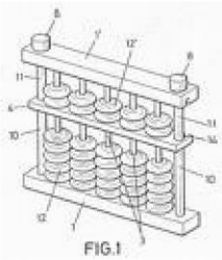
Prehistoria: Generación 0.

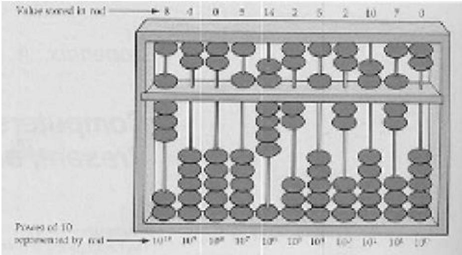
El hombre primitivo usó los diez dedos de las manos para contar (de ahí nace el sistema de numeración decimal), posteriormente se ayudó de guijarros o piedras agrupadas en montones y se sofisticó el método de contar haciendo surcos en una bandeja cubierta de arena (**tablas de arena**[1]), en la que se representaban los números por piedras y sus posiciones en los surcos.

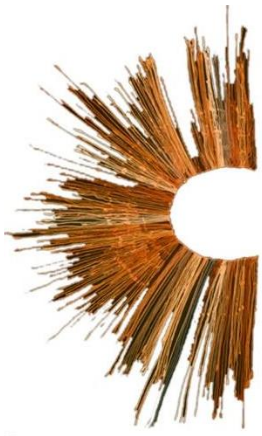


[1] **En las tablas de arena, el primer surco de la derecha puede corresponder a las unidades, el segundo a las decenas, el tercero a las centenas, y así sucesivamente. La suma consiste en añadir piedras en la hendidura derecha; cuando se completa con diez, se quitan todas y se añade una en la fila siguiente, y así sucesivamente.**

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.

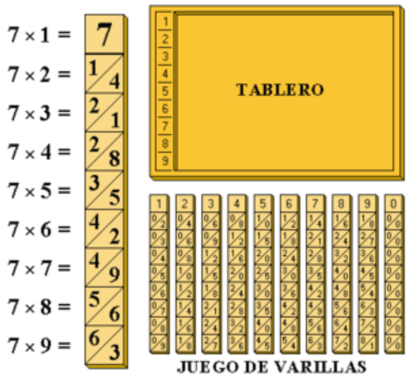


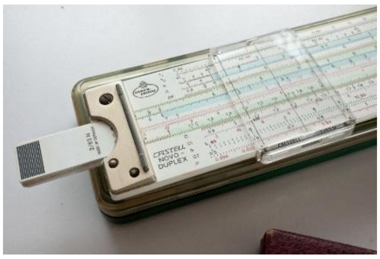




El primer instrumento construido por la humanidad para facilitar el cálculo fue el **ábaco**, que simplificaba las operaciones aritméticas básicas (suma, resta, multiplicación y división). Puede ser considerado como el origen de las máquinas de calcular. A pesar de su antigüedad, entre el 2000 y 1000 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:

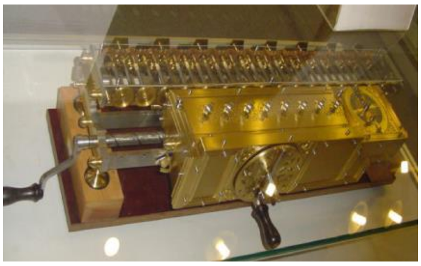
En **1642** el joven filósofo, matemático y físico francés **Blaise Pascal**, inventó una máquina para sumar y restar, que patentó en 1647. A dicha máquina se le dio el nombre de **Pascalina**. Tenía las dimensiones de una caja de zapatos y en su interior disponía unas ruedas dentadas conectadas entre 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.

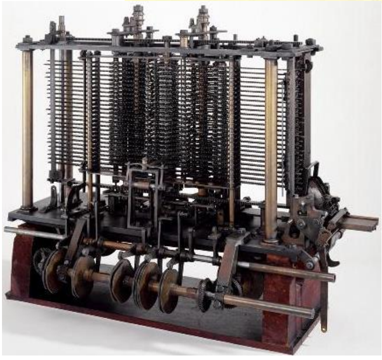
Aun cuando el logro de Pascal fue apreciado en toda Europa, la Pascalina fue un fracaso financiero, debido a que era Pascal la única persona que podía repararla, además, en esta época el trabajo manual en cálculos aritméticos era muchísimo más barato que la máquina.

En **1671** el filósofo y matemático alemán Gottfied 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.

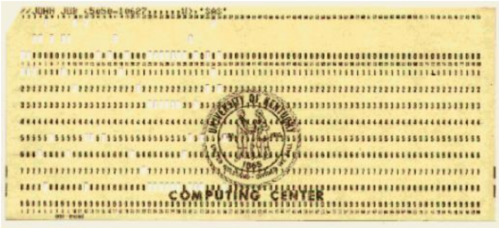


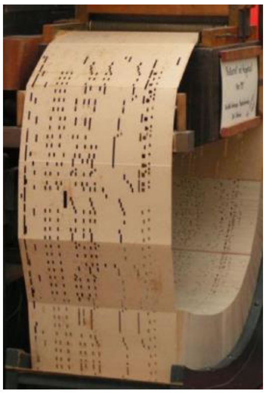
Leonardo da Vinci, Pascal y Leibniz construyeron las primeras máquinas, pero fue Charles **Babbage**, en **1832**, el que creó la primera capaz de encadenar varias operaciones automáticamente y que permitía resolver ecuaciones de primer grado con su **máquina diferencial**.
En **1833**, la **máquina 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 creo **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**[2]. 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.





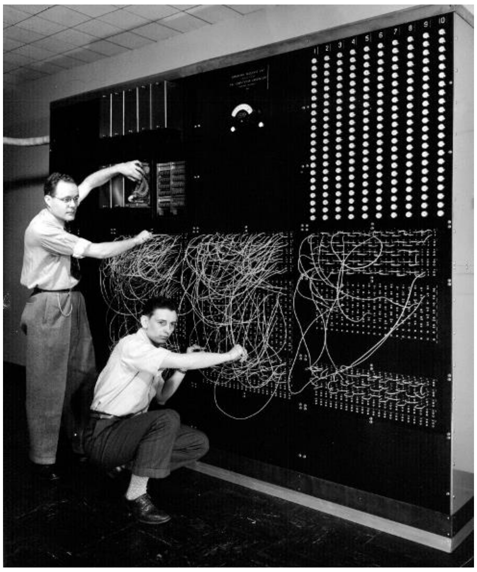
El Dr. Hollerith, en **1896** fundó la **Tabulating Machine Company** y vendió sus productos en todo el mundo. La demanda de sus máquinas llegó hasta Rusia, en donde el primer censo ruso efectuado en 1897 se efectuó usando la Máquina Tabuladora del Dr. Hollerith. En **1911** la Tabulating Machine Company 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**.

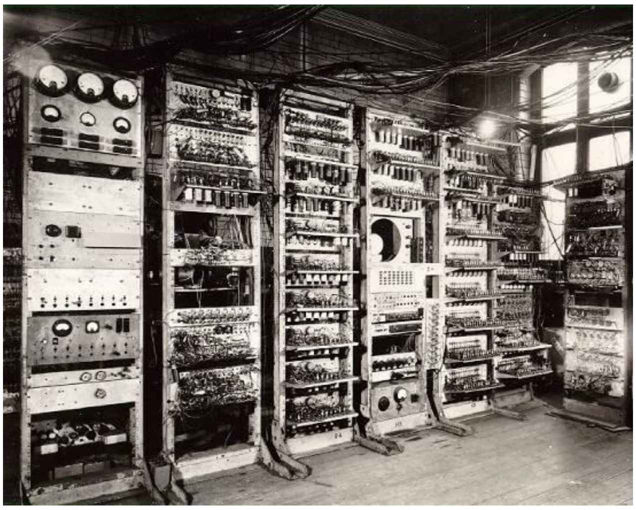
En **1924**, Watson cambió el nombre de su compañía CTR por el de **International Business Machines Corporation (IBM)**, a día de hoy es la vigésima empresa más grande del mundo y es la empresa en EEUU con más patentes dentro del mundo de la tecnología y tiene nueve laboratorios de investigación. La "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.

[2] En la tabuladora las tarjetas van posicionándose una a una en una estación de lectura, cuyo soporte es una placa conductora. Unas varillas metálicas telescópicas entran en contacto con la superficie superior de la tarjeta; en los sitios que hay perforación las varillas tocan la placa metálica, cerrándose así un contacto eléctrico. Este circuito provoca el desplazamiento de un dial, que va contabilizando el número de tarjetas que tienen perforación en la posición correspondiente y hace que la tarjeta vaya al cajetín de clasificación correspondiente.

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.



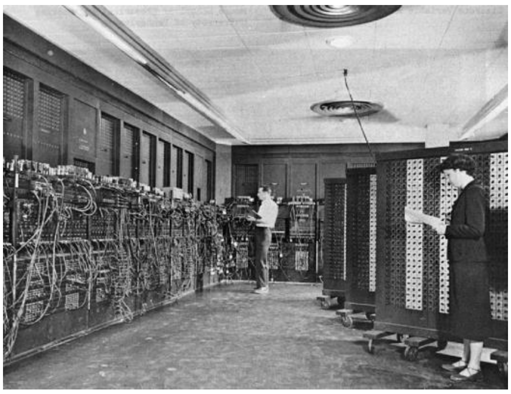


La Era Electrónica: Generación 3.

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 desclasifico la información del anterior

1944: El . 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 Feranti 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.

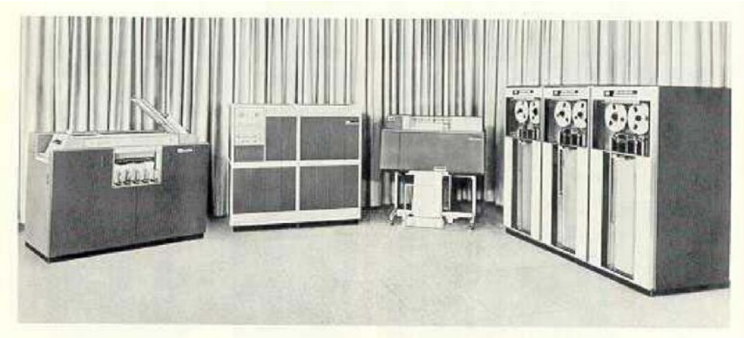
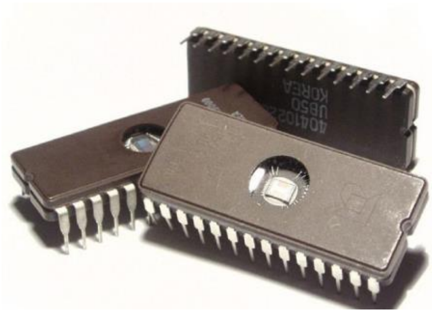


FIGURE 41. 1401 TAPE SYSTEM, MODEL C

3ª Generación: Chips

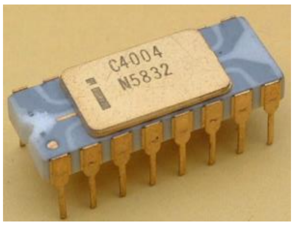
1964-1974: Aparece el IBM 370. Ordenadores basados en circuitos integrados, supuso la minimización de los ordenadores.

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.

2. Representación de la información

1. Conceptos Fundamentales

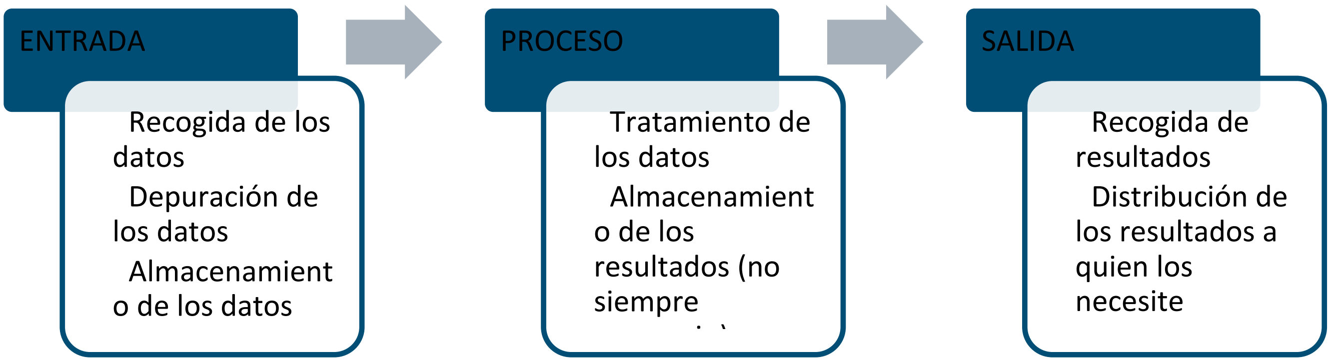
El término **ordenador** procede del francés y no del inglés *computer*, del que deriva computadora. "*Ordinateur*" significa quien ordena, pero también tiene una connotación religiosa, ya que se usa como referencia a Dios, quien pone orden en el mundo. De modo que podría decirse que en Francia y en España las computadoras son casi deidades.

Ambos términos se refieren al mismo aparato pero en castellano se suele usar preferentemente el primero. Por lo cual, tanto un término como otro se pueden usar indistintamente para referirse a dicha máquina.

Un **ordenador** es una máquina compuesta de elementos físicos tanto mecánicos como electrónicos. Los primeros permiten que se ponga en marcha una unidad de disco, por ejemplo. Los segundos nos permiten realizar trabajos con gran velocidad y precisión siempre, claro está, que le demos el programa adecuado. Cuando para realizar una determinada tarea son precisos varios programas hablamos de **aplicación informática**.

Para poder ejecutar los programas y/o aplicaciones precisamos de **un sistema informático** que es el conjunto de elementos que nos permiten introducir la información, tratarla y obtener los resultados deseados. Este conjunto de operaciones con la información se denomina **tratamiento**.

Las tres partes del tratamiento son:



En todo tratamiento de la información intervienen tres elementos:

- El *hardware* o parte física y tangible del sistema informático.
- El *software* o parte lógica (sistemas operativos, programas y/o aplicaciones).
- El operador o usuario.

A este tratamiento de forma automatizada es a lo que llamamos informática, definiéndola de una forma más adecuada, la **informática** es la ciencia tecnológica que estudia el tratamiento automático y racional de la información, con el fin de obtener de ella la máxima utilidad y funcionalidad.

Para entender el término *información* hay que hacerlo en conjunto de los componentes que forman un sistema de comunicación completo, este sistema de comunicación está dividido en varios elementos básicos:

- **Emisor**, fuente o transmisor: El que genera la información.
- **Receptor**: El que recibe la información generada.
- **Canal**: medio por el que se transmite la información generada.
- **Mensaje**: es información compuesta por un conjunto de señales y reglas conocidas por el emisor y el receptor.
- **Código**: sistema de signos empleados (código morse, idioma inglés, etc.)
- **Contexto**: circunstancias en las que se emite el mensaje

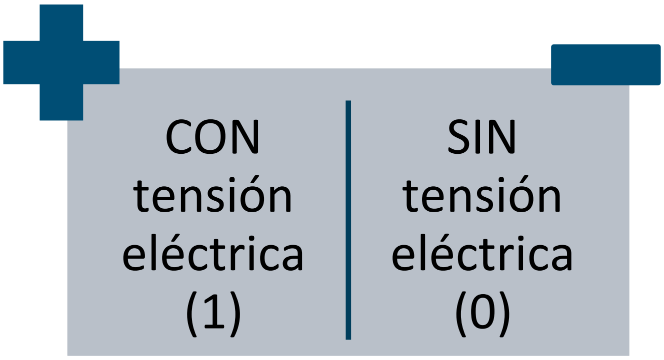
Con estos elementos podemos definir la **información** como la representación de hechos, objetos, valores, ideas, etc., que permiten la comunicación entre el emisor y el receptor, y la adquisición del conocimiento de las cosas.

La transmisión en términos generales, como hemos podido observar, se efectúa de muy diversas formas pero un computador, *¿cómo se comunica realmente?*

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, aunque ordenadores diferentes emplean códigos distintos, e incluso se usan códigos diferentes en las distintas partes del mismo ordenador. La codificación puede ser más o menos arbitraria dependiendo de si los datos se van a utilizar para la representación, para la transmisión, para la realización de operaciones aritméticas, etc.

Por este motivo se utilizan códigos normalizados como el **ASCII** (*American Standard Code for Information Interchange*) (el ASCII básico utiliza 7 bits) para la entrada y salida de datos, mientras que para operaciones aritméticas se utilizan codificaciones en binario natural o *BCD natural*, y, en ocasiones se utilizan los sistemas *octal* y *hexadecimal* por la facilidad y simplicidad que ofrecen en su transformación al decimal y al binario.

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.

2. La información y su representación

Como ya hemos comentado anteriormente los ordenadores trabajan con impulsos eléctricos, sólo son capaces de interpretar si un elemento está cargado o no de electricidad. Por lo tanto 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).

Los datos e instrucciones se le dan al ordenador en la forma usual escrita que usamos los seres humanos, con ayuda de un alfabeto. Por tanto, la información suministrada al ordenador hay que traducirla a ceros y unos por lo que se necesita una codificación.

Codificación: representación de los elementos de un conjunto α, donde α = {A,B,C,...,Z,a,b,c...z,0,1,...(,)...} mediante los de otro conjunto, β, donde β = {0,1} n de forma que a cada elemento de α le corresponde un elemento distinto de β.

Estos códigos se llaman **códigos de E/S** y existen varios que están normalizados y que son utilizados por distintos sistemas informáticos. Las operaciones aritméticas con datos numéricos se suelen realizar en una representación fundamentada en el sistema de numeración en base 2, sistema que puede considerarse como una codificación en binario más, pero que al ser posicional es muy apta para hacer operaciones aritméticas.

2.1. Teorema Fundamental de la Numeración

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:

$$N_i = \sum_{i=-d}^{n} (digito)_i \cdot (base)^i$$

Donde:

i = posición respecto a la coma.

d= número de dígitos a la derecha de la coma.

n= número de dígitos a la izquierda de la coma -1.

dígito= cada uno de los que componen el número.

base= base del sistema de numeración.

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:$(1 \cdot 3^2)+(1 \cdot 3^1)+(2 \cdot 3^0)+(0 \cdot 3^{-1})+(2 \cdot 3^{-2}) \rightarrow 14{,}2222_{(10}$

2.2. Sistemas de Numeración

2.2.1. 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 decima 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:

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

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

2.2.2. 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 →4 números; con 3 bits →8 números; con 4 bits →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:

Números que podemos representar = $2^{\text{nº de bits que utilicemos}}$

2.2.3. 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:

$$N=(1\cdot8^3)+(0\cdot8^2)+(0\cdot8^1)+(1\cdot8^0)$$

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

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

2.2.4. 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:

$N=(2\cdot16^2)+(C\cdot16^1)+(A\cdot16^0)$

$N=(2\cdot16^2)+(12\cdot16^1)+(10\cdot16^0)$

$N=512+192+10=714$

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

1.

2.3. CONVERSIÓN ENTRE SISTEMAS Y OPERACIONES BINARIAS

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.

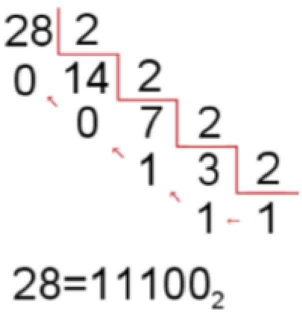
Decimal → otro sistema: Se realiza mediante divisiones sucesivas usando como divisor la base a la que queremos convertir el número. Se toma para la siguiente división sólo la parte entera del cociente y se van guardando los restos obtenidos. Cuando el último cociente es menor que la base, se toma éste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

2.3.1. Conversión de un número decimal a binario

Para transformar un número de decimal a binario hacemos:

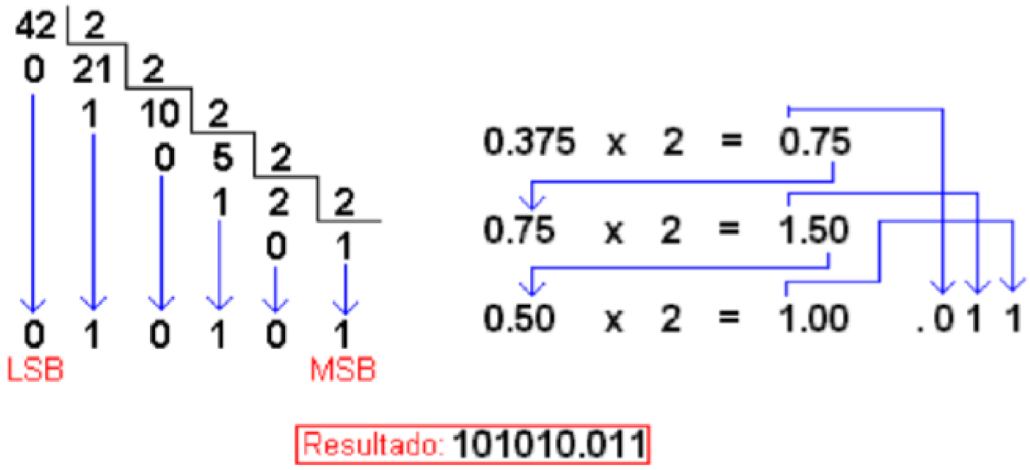
- **a.** La parte entera del nuevo número (binario) se obtiene dividiendo por 2 (sin obtener decimales en el cociente) la parte entera del número decimal de partida, y los cocientes que sucesivamente se vayan obteniendo.
- **b.** Los restos de estas divisiones y el último cociente (que serán siempre 0s o 1s) son las cifras binarias. El último cociente será el bit más significativo y el primer resto el bit menos significativo.
- **c.** 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: $28_{(10} \rightarrow ¿?_{(2}$



Ejemplo 2: $42{,}375_{(10}$ ¿?$_{(2}$

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.



Ejemplo 3: $74{,}423_{(10}$ ¿?$_{(2}$

a) Parte entera:

74	|2					
14	37	|2				
0	17	18	|2			
	1	0	9	|2		
			1	4	|2	
				0	2	|2
					0	1

b) Parte fraccionaria:

0,423	0,846	0,692	0,384	0,768
x 2	x 2	x 2	x 2	x 2
0,846	1,692	1,384	0,768	1,536

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

2.3.2. 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

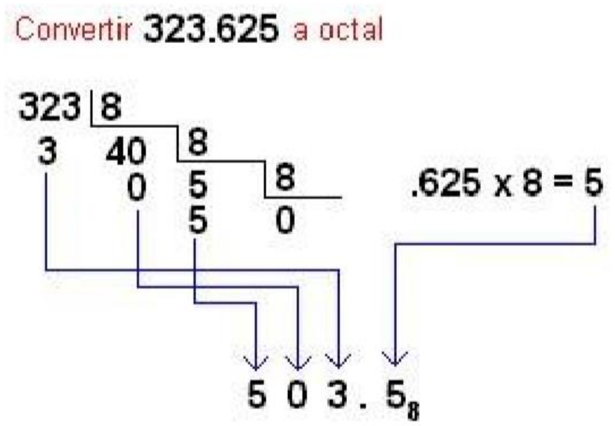
Indicamos los exponentes sobre los dígitos y sólo hay que sumar los que tienen 1

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

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

2.3.3. 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.



2.3.4. 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 $47_{(8}$ a decimal:

$47_{(8} = 4 \cdot 8^1 + 7 \cdot 8^0 = 39_{(10}$

2.3.5. 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.

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

Ejemplo:

Transformamos $15_{(8}$ a binario: 1 5

001 101

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

2.3.6. 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

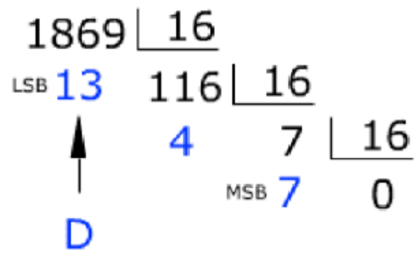
$$010 \quad 001 \quad 101 \quad 100_{(2} = 2154_{(8}$$

$$2 \quad 1 \quad 5 \quad 4$$

2.3.7. Conversión de un número decimal a hexadecimal.

Lo primero es aclarar que el sistema hexadecimal o base 16 tiene una particularidad es que utiliza símbolos pertenecientes a letras para expresar sus cantidades más altas, por la 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.

2.3.8. 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:

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

2.3.9. Conversión de un número hexadecimal a binario.

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

7 F B C 2

$7FBC2_{(16} = 0111\ 1111\ 1011\ 1100\ 0010_{\ (2}$

7FB,C2$_{(16}$ = 0111 1111 1011 , 1100 0010 $_{(2}$

2.3.10. Conversión de un número de 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

0010 0101 1101 , 1111 $_{(2}$ = 25D.F $_{(16}$

2 5 D . F

2.3.11. 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.

DECIMAL	BINARIO	HEXADECIMAL	OCTAL
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.

2.3.12. 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, acarreo 1

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

```
acarreo  ──→      1  1  1
                  1  0  1  1  0
               +  0  1  1  1  0
               ────────────────
               1  0  0  1  0  0
```

2.3.13. 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.

```
   11011     minuendo
-   1101     sustraendo
 ───────
    1110
```

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

Ejemplo 2:

Restar los siguientes números binarios: 010011 y 110000

110000 → C1 → 001111 → C2 → 010000

```
  010011
+ 010000
 ------------
```

100011

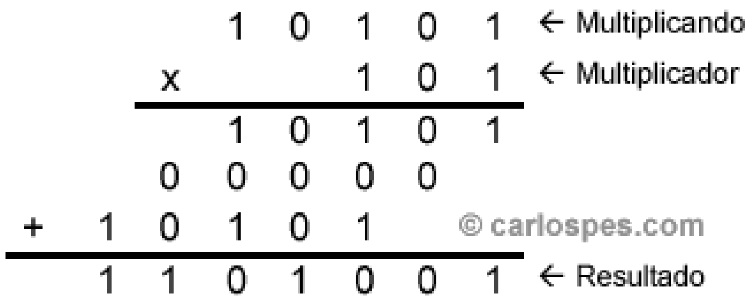
2.3.14. MULTIPLICACIÓN BINARIA.

Se realiza como en la multiplicación decimal, con la diferencia de que las sumas se hacen en binario.

MULTIPLICACIÓN BINARIA

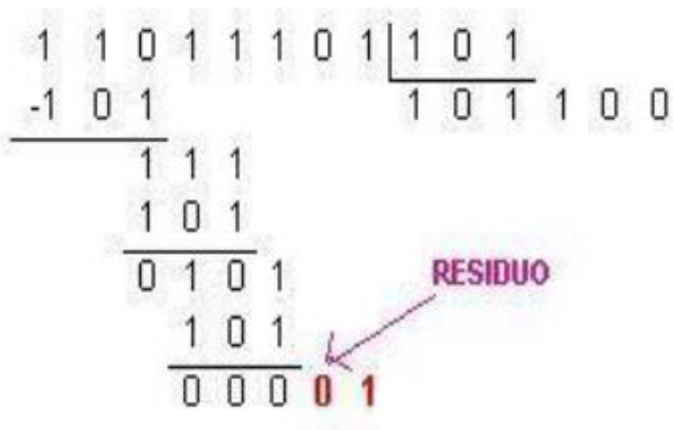
$0 \cdot 0 = 0$	$0 \cdot 1 = 0$	$1 \cdot 0 = 0$	$1 \cdot 1 = 1$

Ejemplo:



2.3.15. DIVISIÓN BINARIA.

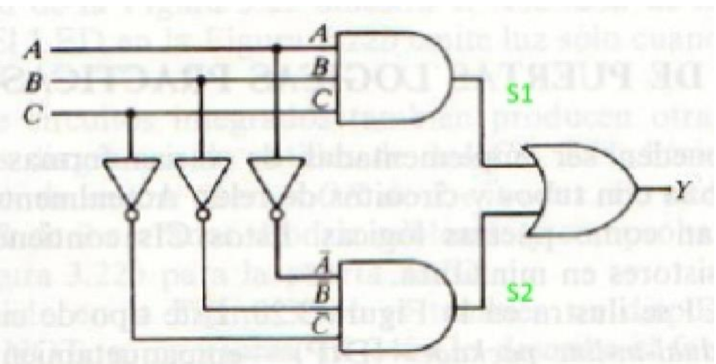
Se efectúa como en la división decimal, pero las multiplicaciones y las restas internas se hacen en binario.



2.3.16. 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.

2.4. Representación de Datos Alfanuméricos

Ya hemos visto como se almacenan las cantidades numéricas dentro del ordenador; ahora nos toca ver cómo se almacena el resto de caracteres que forman el alfabeto. Los códigos de E/S permitirán traducir la información o los datos que nosotros podemos entender a una representación que la máquina pueda interpretar y procesar.

Para ello han aparecido varios estándares que lo han regularizado:

- FIELDATA Y XS-3: Los más antiguo y en desuso utilizaba 6 bits.
- ASCII (*American Standard Code for Information Interchange* o Código Estadounidense Estándar para el Intercambio de Información): Se utilizan 8 bits pero sólo se usan 7 bits para caracteres alfanuméricos dejando el último para futuras expansiones y caracteres gráficos.

  2^7 → 128 caracteres diferentes (usa 7 bits), no incluye caracteres como la ñ y caracteres acentuados.

  2^8 → 256 caracteres diferentes (usa 8 bits)

0		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	P	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

CONVERSIÓN DE TEXTO A BINARIO

PASO 1: Buscamos el número decimal que corresponde a cada letra (cuidado con las minúsculas y mayúsculas).

PASO 2: Convertimos ese número decimal a binario.

CONVERSIÓN DE BINARIO A TEXTO

PASO 1: Convertimos ese número binario a decimal.

PASO 2: Buscamos el número decimal que corresponde a cada letra (cuidado con las minúsculas y mayúsculas).

FIN

Ejemplo: Convierte a binario

Introduce aquí la respuesta							

- UNICODE: Es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas, además de textos clásicos de lenguas muertas. El término Unicode proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad.

  El establecimiento de Unicode ha sido un ambicioso proyecto para reemplazar los esquemas de codificación de caracteres existentes, muchos de los cuales están muy limitados en tamaño y son incompatibles con entornos plurilingües. Unicode se ha vuelto el más extenso y completo esquema de codificación de caracteres, siendo el dominante en la internacionalización y adaptación local del software informático. El estándar ha sido implementado en un número considerable de tecnologías recientes, que incluyen XML, Java y sistemas operativos modernos.

  La última revisión es la 13.0 de marzo de 2020 e incluye 143.859 caracteres.

- BCD Y EBCDIC BCD, que significa decimal codificado en binario, en realidad no es un código de E/S, sino una forma de codificar los símbolos numéricos del 0 al 9 que se emplean en varios códigos de E/S, entre los que figuran EBCDIC y ASCII.

BCD divide cada octeto en dos mitades o cuartetos, cada uno de los cuales almacena en binario una cifra. Con este código es muy fácil convertir del binario al sistema decimal.

El EBCDIC, es el código BCD Extendido de caracteres decimales codificados en binario para el intercambio de información. Utiliza 8 bits y puede representar 256 caracteres.

2.5. Unidades de Medidas de Informacion.

Repasando un poco la palabra bit deriva de Binary Digit, es la unidad mínima de almacenamiento empleada en informática pero cuando se almacena la información no se hace a nivel de bit, sino que se trabaja a nivel de un conjunto de bits. Los tamaños más comunes son:

- Octeto, carácter o byte: Es la agrupación de 8 bits, el tamaño típico de información; con él se codifica el alfabeto completo (ASCII estándar).
- KiB, MiB, GiB, etc.: son sucesivas agrupaciones de 1024 B, KiB, MiB, GiB, etc.
- Palabra: tamaño de información manejada en paralelo por los componentes del sistema. Son comunes las palabras de 8, 32, 64, 128, 256, 320, 384, 512 bits. A mayor tamaño de palabra, mayor es la precisión y la potencia de cálculo del ordenador.

Nota: "La razón por la que se utiliza el factor multiplicador 1024 en vez de 1000, como sucede en otras magnitudes físicas, es por ser una potencia entera de 2 (2^{10}=1024) y, en consecuencia, un valor mucho más conveniente para máquinas que trabajan en sistema binario."

La siguiente tabla muestra las unidades de medida de información más utilizadas, tanto en su uso decimal como en su uso binario.

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

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 bits / 8 = 709873,75 B / 1024 = 693,236 KB /1024 = 0,68 MB

b. CAMBIO A UNA UNIDAD MENOR

b	→	B	→	KB	→	MB	→	GB	→	TB	→	…
	x8		x1024		x1024		x1024		x1024		x1024	

Ejemplo

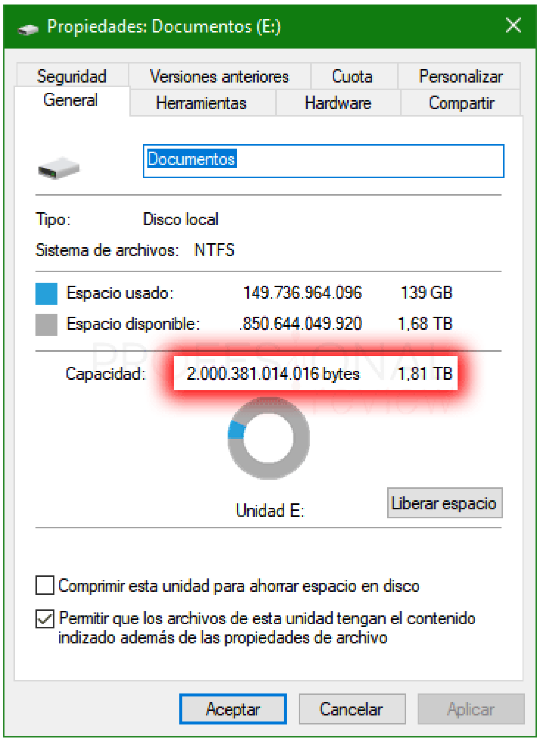
¿Cuántos bits son 2 MB?

2 MB x 1024 = 2048 KB x 1024 = 2097152 B x 8 = 16777216 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 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 menos en nuestro equipo.

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