

**MÓDULO:**

**Sistemas Informáticos**

# **Unidad 1**

**INTRODUCCIÓN A LOS SISTEMAS  
MICROINFORMÁTICOS**

## INDICE DE CONTENIDOS

<b>1. HISTORIA DE LA INFORMÁTICA. GENERACIONES.....</b>	<b>4</b>
<b>2. TERMINOLOGÍA Y CONCEPTOS GENERALES.....</b>	<b>5</b>
2.1 Definición de Informática .....	5
2.2 Elementos y conceptos básicos.....	6
2.3 Concepto de información.....	7
2.4 Sistema de información.....	7
<b>3. LOS COMPONENTES FÍSICOS DE UN ORDENADOR. ....</b>	<b>9</b>
3.1 EL MICROPROCESADOR (UCP).....	10
3.1.1 UNIDAD DE CONTROL (UC).....	11
3.1.2 UNIDAD DE PROCESO (UP) .....	11
3.1.3 MEMORIA CACHE .....	11
3.2 LA MEMORIA .....	11
3.2.1 MEMORIA INTERNA.....	11
3.2.2 MEMORIA EXTERNA $\cong$ Memoria secundaria $\cong$ Memoria Auxiliar .....	12
3.3 BUSES.....	12
3.3.1 TIPOS DE BUSES .....	12
3.4 PERIFÉRICOS .....	13
3.4.1 CLASIFICACIÓN DE LOS PERIFÉRICOS.....	13
<b>4. LA INFORMACIÓN Y SU REPRESENTACIÓN INTERNA.....</b>	<b>14</b>
4.1 Sistemas de numeración .....	14
4.2 Codificación de la información .....	14
4.3 Bit, byte y sus múltiplos.....	15
4.4 Sistemas Posicionales. ....	16
4.5 Teorema fundamental de la numeración.....	16
4.6 El sistema decimal. ....	16
4.7 El sistema binario.....	18
4.8 El sistema octal .....	19
4.9 El sistema hexadecimal.....	19
4.10 Conversión entre Decimal, Binario, Hexadecimal y Octal. ....	20
4.10.1 Conversión de Binario a Decimal. ....	20
4.10.2 Conversión de decimal a base b.....	20
Conversión De Decimal A Binario:.....	20
Conversión De Decimal A Octal:.....	21
4.10.3 Conversión de Hexadecimal a Binario.....	22
4.10.4 Conversión de Binario a Hexadecimal.....	22
4.10.5 Conversión de Octal a Binario. ....	23
4.10.6 Conversión de Binario a Octal. ....	23
4.11 Representación interna de números enteros. ....	23
4.11.1 Binario puro.....	24
4.11.2 Decimal desempaqueado .....	24
4.11.3 Decimal empaquetado .....	25
4.11.4 Exceso a $2^{n-1}$ .....	25
4.12 Representación interna de números reales. ....	26
4.13 Codificación alfanumérica. ....	27
4.14 Operaciones elementales con números binarios.....	28
4.14.1 Suma en binario .....	29
4.14.2 Sustracción en binario.....	29
4.14.3 Complemento a dos .....	30

4.14.4	Complemento a uno.....	30
4.14.5	Restar en binario usando el complemento a dos.....	31
4.14.6	Multiplicación binaria.....	31
4.14.7	División binaria.....	32
<b>5.</b>	<b>CIRCUITOS INTEGRADOS.....</b>	<b>33</b>
5.10	Clasificación de los Circuitos Integrados de acuerdo a su estructura. ....	33
5.11	Puertas lógicas.....	34
5.11.1	Puerta lógica AND. ....	34
5.11.2	Puerta lógica OR. ....	35
5.11.3	Puerta lógica NOT.....	35
5.11.4	Puerta lógica NAND.....	36
5.11.5	Puerta lógica NOR. ....	36
5.11.6	Ejemplos con puertas lógicas. ....	36

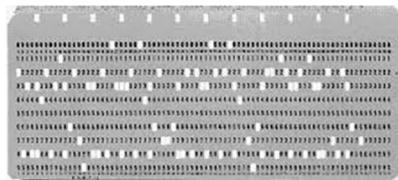
## 1. HISTORIA DE LA INFORMÁTICA. GENERACIONES

La aparición de la Informática se ha debido a la gran demanda de información que tenemos a nuestra disposición y a la dificultad para manejarla personalmente. Un ordenador es una máquina capaz de manipular datos y proporcionar resultados, siguiendo una serie de instrucciones. Debido a los rápidos avances en el mundo de la electrónica, sobre todo a partir de 1946, los ordenadores se clasifican por generaciones. Cada una de estas generaciones se caracteriza por los componentes que forman parte de un ordenador.

### • 1ª Generación

Abarca desde **1946 hasta 1957** y se caracteriza porque todos los ordenadores que pertenecen a ella estaban contruidos por medio de **válvulas electrónicas y tubos de vacío**. Estos ordenadores eran de gran tamaño, muy pesados, consumían mucha energía y se averiaban con bastante frecuencia. Los datos les

eran proporcionados por medio de **fichas o cintas perforadas** y se dedicaban, fundamentalmente, al cálculo científico. El lenguaje que se utilizaba para comunicarse con este tipo de ordenadores era lenguaje máquina. El ordenador más conocido fue el



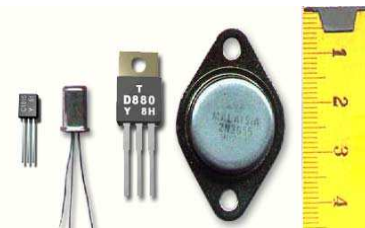
ENIAC.

### • 2ª Generación

Pertenecen a esta los ordenadores desarrollados desde **1958 a 1964**. En estos ordenadores los circuitos estaban hechos de **transistores y la memoria de núcleos de ferrita**.

Este hecho hizo que los ordenadores fueran mucho más pequeños que los contruidos hasta ese momento, tuvieran menos consumo y fueran capaces de ejecutar alrededor de 10 millones de operaciones por minuto. Los datos para estos ordenadores eran

suministrados por medio de **cintas magnéticas y se utilizaba lenguajes simbólicos**, tipo FORTRAN y COBOL. Comienzan a utilizarse para tareas administrativas y admiten algo de trabajo en cadena. El primer ordenador de esta generación fue el **TRADIC de los Laboratorios Bell**.



### • 3ª Generación

Se incluyen aquí los ordenadores que aparecieron entre **1965 y 1971**. Estos contienen **circuitos integrados o chips** y dieron lugar a la **microelectrónica**, es decir, el desarrollo de componentes electrónicos de tamaño microscópico. Con esa nueva tecnología, la velocidad de los ordenadores llegó a ser de alrededor de 100 millones de operaciones por segundo, y se consiguió un menor tamaño en los mismos. Los avances de esta tercera generación dan paso al multiproceso, es decir a la capacidad de realizar varios procesos a la vez; y a los lenguajes de programación.

### • 4ª Generación

Va desde **1972 a 1981** y se caracteriza por la aparición de los **circuitos integrados a gran escala**, es decir, más evolucionados. Los nuevos ordenadores ya no sólo son utilizados en las grandes empresas, sino que se utilizan de forma personal. Son más pequeños, más baratos e incorporan la posibilidad de utilizar diferentes programas para



diferentes aplicaciones.

- **5ª Generación**

Se incluyen en éste todos los ordenadores desarrollados a **partir de 1981**. Estos ordenadores siguen utilizando circuitos integrados, pero son de una gran velocidad. Es en esta generación en la que han aparecido los **ordenadores personales**. En esta generación de ordenadores ha dado comienzo el **desarrollo de la inteligencia artificial**, es decir, las investigaciones con el fin de diseñar ordenadores que sean capaces de desarrollar determinadas funciones del cerebro humano.



## 2. TERMINOLOGÍA Y CONCEPTOS GENERALES

### 2.1 Definición de Informática

¿Te gustan las películas del oeste, las películas de indios y vaqueros? ¿Recuerdas que los indios solían utilizar señales de humo para transmitirse mensajes, generalmente para avisar de la presencia del enemigo? Seguro que te suena también de alguna película el uso de espejos y destellos con este mismo fin. El hombre



siempre ha sentido la necesidad de comunicarse, es decir, de **transmitir información**, y estos son sólo dos ejemplos de los numerosos y distintos mecanismos que el hombre ha utilizado a lo largo de la historia para lograr tal objetivo. También podríamos hacer mención de mecanismos más recientes, como los mensajes transmitidos a través de cables utilizando el código Morse o la transmisión de voz por medio del teléfono.

**Se conoce con el nombre de Telecomunicaciones a la disciplina que se encarga de estudiar los métodos y tecnologías para la transmisión de información.**

Por otra parte, el hombre ha tenido también **la necesidad de recoger, tratar, almacenar y mostrar información**. Seguro que conoces que ya en la época de los romanos se elaboraban censos para contabilizar y registrar a la población. Pero esa época queda ya muy atrás. Con la revolución tecnológica de los últimos siglos aparecen nuevos métodos y máquinas para procesar información de manera automatizada, y es precisamente **la Informática, la disciplina o ciencia encargada del estudio y desarrollo de estas máquinas y métodos**.

Hoy en día, con la aparición y proliferación de las redes de ordenadores, ambas disciplinas están estrechamente relacionadas, al haberse convertido la máquina utilizada en el mundo de la informática, el ordenador, en una herramienta utilizable para comunicarse. De hecho, surge una nueva disciplina, llamada **Telemática**, que bebe de ambas y se encarga de **estudiar el ordenador como medio de comunicación**.



La disciplina de **la informática nace con la intención de ayudar al hombre en aquellos trabajos rutinarios y repetitivos, generalmente de cálculo y de gestión, donde es frecuente la repetición de tareas**. La idea es que una máquina puede realizar dichos trabajos mucho mejor y más rápidamente que el hombre, aunque siempre bajo la supervisión de éste. No debemos tener miedo a que las máquinas vayan a hacer prescindible al ser humano, al trabajador o trabajadora, simplemente

tendremos que hacernos a la idea de que los recursos humanos se van a tener que dedicar a realizar otro tipo de trabajos.

El término **informática** proviene del francés *informatique*, implementado por el ingeniero **Philippe Dreyfus** a comienzos de la década del '60. La palabra es, a su vez, un acrónimo de *information* y *automatique*.

De esta forma, la informática se refiere al **procesamiento automático de información** mediante **dispositivos electrónicos** y **sistemas computacionales**. Los sistemas informáticos deben contar con la capacidad de cumplir tres tareas básicas: **entrada** (captación de la información), **procesamiento** y **salida** (transmisión de los resultados). El conjunto de estas tres tareas se conoce como **algoritmo**.

La informática reúne a muchas de las técnicas que el hombre ha desarrollado con el objetivo de potenciar sus capacidades de pensamiento, memoria y comunicación. Su área de aplicación no tiene límites: la informática se utiliza en la **gestión de negocios**, en el **almacenamiento de información**, en el **control de procesos**, en las **comunicaciones**, en los **transportes**, en la **medicina** y en muchos otros sectores.

## 2.2 Elementos y conceptos básicos.

Antes de empezar a hablar de los elementos y detalles relacionados con el mundo de la informática, conviene definir sus componentes más importantes. Desde el punto de vista informático, el elemento físico utilizado para el tratamiento de la información es el ordenador, que puede ser definido de las siguientes maneras:



**El ordenador es una máquina compuesta de elementos físicos, en su mayoría de origen electrónico, capaz de realizar una gran variedad de trabajos a gran velocidad y con gran precisión, siempre que se le den las instrucciones adecuadas.**

**El ordenador es una máquina digital capaz de resolver cualquier problema que este especificado mediante una serie de instrucciones (programa).**

- El **conjunto de órdenes que se dan a un ordenador** para realizar un proceso determinado se denomina **programa**,
- El **conjunto de uno o varios programas**, más la documentación correspondiente para realizar un determinado trabajo, se denomina **aplicación informática**.

Podemos decir que una **aplicación informática** es un macroprograma que consta de **varios programas independientes aunque interrelacionados**; es decir, programas que funcionan de forma autónoma pero que pueden necesitar información que se ha procesado en otros programas dentro del macroprograma. Podemos imaginarnos una aplicación informática bancaria formada por varios programas como, por ejemplo, un programa para el tratamiento de nóminas, otro para gestión de préstamos hipotecarios, un programa de contabilidad, etc.

Un **sistema informático** se define como el **sistema compuesto de equipos y de personal pertinente**, que realiza funciones de entrada, proceso, almacenamiento, salida y control, con el fin de llevar a cabo una secuencia de operaciones con datos.

Anteriormente se ha definido ordenador como “una máquina compuesta de elementos físicos, en su mayoría de origen electrónico”. Dichos componentes físicos son los que conocemos de manera genérica con el nombre de hardware.



Llamamos **hardware** de un ordenador a **todo elemento físico de un sistema informático**, es decir, todos los materiales que lo componen, como los chips que lo componen, los dispositivos externos que se conectan a él, los cables, los soportes de información y, en definitiva, todos aquellos elementos que tienen entidad física.

Por contraposición, el software de un sistema informático es la parte lógica de un sistema informático; es decir, aquella que dota al equipo físico de capacidad para realizar cualquier tipo de trabajos. Por software nos estamos refiriendo a lo que no es materia física y que tradicionalmente se ha considerado programación o programas informáticos, que le indican al ordenador cómo debe realizar sus tareas.

## 2.3 Concepto de información.

Anteriormente:

- Se mencionó la necesidad del hombre de transmitir y tratar información.
- Se señaló a las Telecomunicaciones y a la Informática como las disciplinas encargadas del estudio respectivo de ambos frentes de actuación sobre la información.

Pero, ¿qué es “información”, qué es realmente aquello que estamos continuamente diciendo que transmitimos y tratamos? Se define **información** como **el conjunto de símbolos que representan algún hecho, concepto u objeto del mundo real**. Por otra parte, llamamos **datos** al **conjunto de expresiones que denotan valores, magnitudes, condiciones, estados, etc.** Normalmente, en la vida común, los términos información y dato se usan indistintamente como sinónimos, si bien en el mundo de la informática hay un pequeño matiz diferenciador entre ambos.



Veamos mediante un ejemplo en qué consiste dicha diferencia. Imagínate una señal de STOP. Cuando vemos dicha señal de tráfico sabemos que tenemos que detenernos, ¿verdad?, sabemos que nos tenemos que parar. En este ejemplo, la señal de STOP sería el dato (señal = STOP), una expresión en forma de dibujo con unos colores y forma determinados. Por otra parte, lo que nos “dice” dicha señal, que paremos, lo que nosotros interpretamos o entendemos cuando vemos la señal, eso es lo que llamamos información. Imagina ahora un semáforo en rojo. Nos encontramos ahora ante un dato totalmente distinto al anterior (semáforo = rojo); sin embargo, dicho dato transmite exactamente la misma información que la señal de STOP, pues nos está diciendo lo mismo: que nos paremos. Podemos observar, por tanto, que no es lo mismo información que dato. Si somos estrictos deberíamos entonces decir que **el ordenador trabaja exclusivamente con datos y que somos nosotros, las personas, los que al interpretar dichos datos extraemos la información que llevan asociada**.

Dentro de los múltiples y variados datos que maneja habitualmente un ordenador, hay una categoría especial: las **instrucciones**. Las instrucciones no son más que una serie de **caracteres, organizados en grupos, que representan órdenes codificadas para el ordenador y que sirven para actuar sobre datos, por ejemplo, modificándolos**. Las instrucciones informan al ordenador sobre las operaciones a realizar, el modo de ejecutarlas, los medios y datos a emplear y sobre los que operar, el tiempo de la ejecución, etc.

## 2.4 Sistema de información.

Hemos aprendido que el ordenador es la máquina encargada de realizar el tratamiento automático de la información pero, para que esto suceda, **la información**

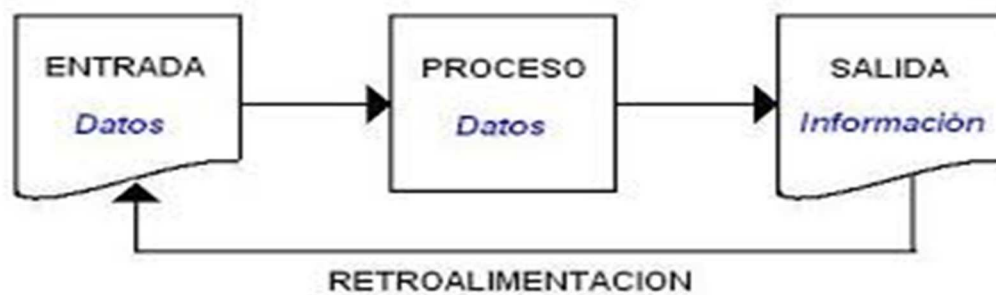


**tendrá que transmitirse o trasladarse desde el exterior hasta el interior de la máquina.** Como en cualquier otro sistema, independientemente de cómo sea éste, para que exista transmisión de información son necesarios tres elementos:

- Un elemento **emisor** que dé origen a la información, en nuestro caso, un elemento exterior.
- Un **medio** que permita la transmisión de la información y que la haga llegar desde el emisor hasta el destinatario de la información.
- Un elemento **receptor** que reciba la información del exterior a través del medio de transmisión y que, en nuestro sistema, será el propio ordenador.

Una vez que la información está en el interior del ordenador se puede empezar a actuar sobre la misma, realizando las transformaciones que sean necesarias para la consecución del objetivo que se persiga. **Al conjunto de operaciones que se realizan sobre una información se le denomina tratamiento de la información.**

Estas operaciones que se realizan sobre la información y que comienzan en las mismas “puertas” del ordenador, pueden ser divididas de manera lógica tal y como se representa a continuación:



```

Procedimiento Ordenar (L)
//Comentario: L = (L1, L2, ..., Ln) es una lista con n elementos//
k ← 0;
Repetir
  intercambio ← Falso;
  k ← k + 1;
  Para i ← 1 Hasta n - k Con Paso 1 Hacer
    Si Li > Li+1, Entonces
      intercambiar (Li, Li+1)
      intercambio ← Verdadero;
  FinSi
FinPara
Hasta Que intercambio = Falso;
FinProcedimiento
  
```



- |                           |               |                               |
|---------------------------|---------------|-------------------------------|
| • Recogida de Datos.      | • Aritmético. | • Recogida de Resultados.     |
| • Depuración de Datos.    | • Lógico.     | • Distribución de Resultados. |
| • Almacenamiento de Datos |               |                               |

En términos generales, se denomina **entrada** al conjunto de operaciones cuya misión es **tomar los datos del exterior**, del medio, **y depositarlos en el interior del ordenador**; para ello, en ocasiones es necesario realizar operaciones de **depuración o validación** de los mismos, para evitar la introducción de datos erróneos o que no cumplan los requerimientos que se espera de ellos. Por ejemplo, supongamos que se está introduciendo un dato que representa la edad de una persona, evidentemente éste dato deberá ser un número positivo, pues nadie tiene una edad negativa. Durante el proceso de entrada deberá comprobarse o validarse, por lo tanto, que el dato de entrada cumple estos requerimientos y en caso de no ser así, deberá avisarse al usuario para que depure su error.

Los datos, una vez dentro del ordenador, deben quedar depositados en la memoria del mismo para su posterior tratamiento. Al **conjunto de operaciones** que se elaboran sobre los datos de entrada para obtener los resultados o datos de salida se le llama **proceso o algoritmo**, y consiste generalmente en una combinación adecuada de operaciones de tipo aritmético (operaciones de suma, resta,



multiplicación, etc.) y comprobaciones lógicas (comparaciones de igualdad, desigualdad, etc.) de cuya ejecución secuencial se obtiene la solución a un problema.

Por último, se denomina **salida** al conjunto de operaciones que **proporcionan los resultados de un proceso** a las personas o entidades externas correspondientes. Se engloban en la salida también aquellas operaciones que dan forma a los resultados y los distribuyen adecuadamente. El algoritmo necesario para la resolución de un problema queda definido cuando una aplicación informática es analizada, de tal forma que posteriormente cada proceso se codifica en un lenguaje que sea reconocible por la máquina (directa o indirectamente) y, tras una preparación final, obtendremos una solución ejecutable por el ordenador. La automatización de un problema para que pueda ser desarrollado por un ordenador se representa en el esquema siguiente:



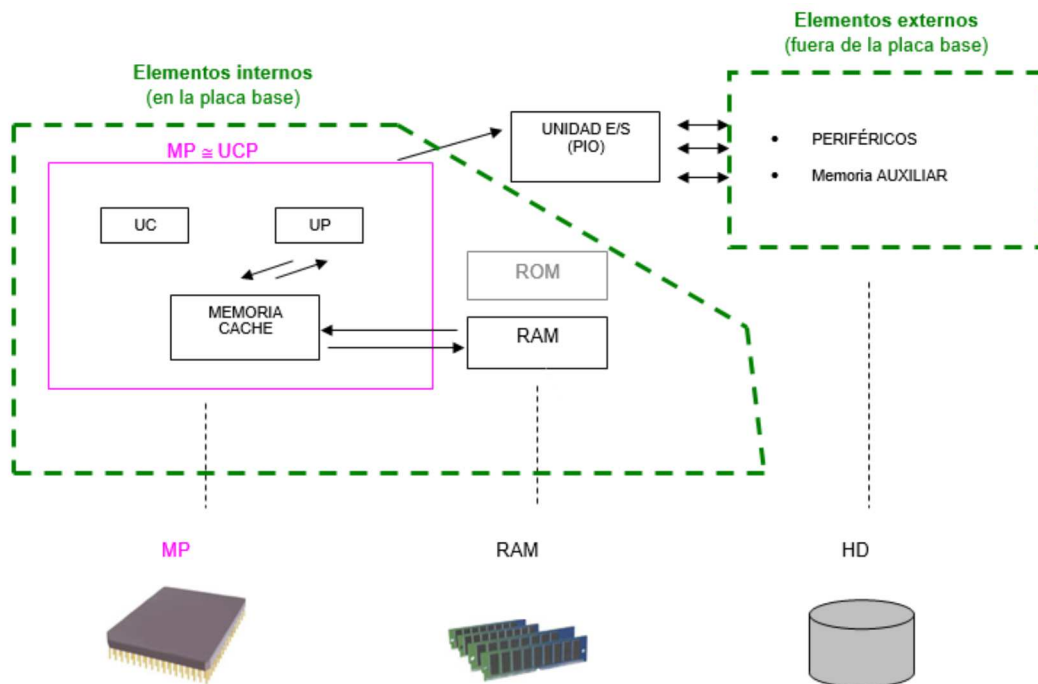
### 3. LOS COMPONENTES FÍSICOS DE UN ORDENADOR.

Los componentes físicos de un ordenador se pueden clasificar:

1. **Microprocesador (MP)  $\equiv$  procesador  $\equiv$  Unidad central de proceso (UCP  $\equiv$  CPU).**
2. **Memoria**
3. **Buses**
4. **Periféricos**

Dichos componentes se encuentran interconectados entre sí, con objeto de realizar la función principal del ordenador (ejecutar programas).

Esquema de los componentes físicos de un ordenador:



**Memoria CACHE**  $\cong$  Memoria Interna al MP

**Memoria RAM**  $\cong$  Memoria Interna  $\cong$  Memoria Principal  $\cong$  Memoria Central (MC)

**Memoria AUXILIAR**  $\cong$  Memoria Externa  $\cong$  Memoria Secundaria - Ejemplo - HD  $\cong$  Disco duro

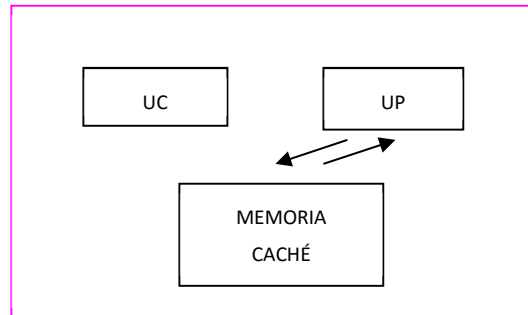
### 3.1 EL MICROPROCESADOR (UCP)

El microprocesador tiene como misión controlar las operaciones del ordenador, es decir, leer las instrucciones, interpretarlas, y ejecutarlas, incluyendo en dicha ejecución tanto las funciones necesarias para el procesamiento de datos, operaciones aritméticas y lógicas, como el envío de las órdenes necesarias a las unidades externas, como puede ser la memoria, los periféricos, etc, con el fin de realizar el tratamiento automático de la información. Es decir, realmente el microprocesador es el encargado de gobernar el funcionamiento del ordenador. El microprocesador es la parte pensante del ordenador.

Microprocesador (**MP**)  $\cong$  procesador  $\cong$  Unidad central de proceso (**UCP**)  $\cong$  Central Process Unit (**CPU**).

Está formado por:

- Unidad de control (UC)
- Unidad de proceso (UP)
- Memoria Caché

MP  $\equiv$  UCP

### 3.1.1 UNIDAD DE CONTROL (UC)

Realiza lo siguiente:

- Controla, coordina e interpreta las instrucciones.
- Gestiona y supervisa el correcto funcionamiento de la unidad de proceso.
- Controla y dirige los componentes externos a ella mediante el envío de señales de control. Entonces les envía esas señales (órdenes) por medio de impulsos para decirles cuando tienen que hacer algo, o sea les dice cuando tienen que ponerse en funcionamiento y cuando tienen que parar.

### 3.1.2 UNIDAD DE PROCESO (UP)

Trata los datos, realiza operaciones sobre los datos y obtiene un resultado. Es la que realiza el trabajo. La Unidad de proceso está formada por:

- Unidad Aritmético – Lógica (**UAL  $\equiv$  ALU** - Arithmetic Logic Unit)
- Registros

### 3.1.3 MEMORIA CACHE

**Memoria de pequeña capacidad que incorporan los procesadores** para que los cálculos de las instrucciones sean más rápidos, pues la memoria caché es mucho ms rápida que la RAM. O sea si tiene que ejecutar una instrucción, en vez de ejecutarla en la RAM, la ejecuta en la caché para que vaya más rápido.

Esta memoria **se coloca entre la CPU y la memoria RAM** y almacena y procesa temporalmente la información. Es una memoria ultrarrápida que ayuda al micro en operaciones con datos que maneja constantemente.

## 3.2 LA MEMORIA

Es el dispositivo donde se almacenan los datos y los programas con los que vamos a trabajar. La memoria con la que puede trabajar el ordenador puede ser de dos tipos:

### 3.2.1 MEMORIA INTERNA

Dos tipos de memorias internas:

- **RAM** (Random Access Memory  $\cong$  Memoria de acceso aleatorio)

Es una memoria de lectura y escritura.

- **ROM** (Read Only Memory  $\cong$  Memoria de sólo lectura)

Es una memoria de sólo lectura, cuya información no se puede modificar.

### 3.2.2 MEMORIA EXTERNA $\cong$ Memoria secundaria $\cong$ Memoria Auxiliar

Permite guardar información en grandes cantidades. La memoria externa o soportes de almacenamiento son: Los disquetes, los CD, los DVD, ...

## 3.3 BUSES

Son un conjunto de líneas eléctricas que permiten la transmisión de señales (información) entre los diferentes componentes del ordenador.

El Bus sirve de interconexión:

- De los periféricos con la memoria interna (RAM).
- De la memoria interna (RAM) con el procesador.
- De la conexión entre las diferentes partes del procesador.

Los buses transmiten la información en paralelo, esto quiere decir, que los datos van por todos los hilos del bus simultáneamente.

El bus es como una autopista en la que el tráfico es muy intenso. Por eso, el tipo de bus que incorpore nuestro ordenador contribuirá a que este sea más rápido o más lento.

### 3.3.1 TIPOS DE BUSES

Hay diferentes tipos de buses:

#### 3.3.1.1 *Bus de control*

Envía las “señales” de la UC a todos los demás elementos. Así por ejemplo, se el bus de control se utiliza, entre otras cosas, para comunicarle a la memoria si lo que se quiere es leer o escribir en ella, para que ésta sepa, respectivamente, si tiene que poner el contenido de la celda que indique el bus de direcciones y ponerlo en el bus de datos, o tiene que recoger lo que le llegue por el bus de datos y almacenarlo en la celda de memoria que indique el bus de direcciones.

#### 3.3.1.2 *Bus de datos*

Envía “datos” entre los periféricos, la RAM y el MP. Conforme han ido evolucionando los ordenadores, el tamaño del bus de datos ha ido creciendo y pasando por tamaños de 8, 16, 32 y 64 bits. Se puede pensar en este tamaño como si fuese el número de carriles que tiene una autopista, cuantos más

carriles más coches pueden circular por ella por segundo. Del mismo modo, cuanto mayor es el ancho de este tipo de buses, mayor es el rendimiento de la máquina, pues mayor caudal de datos puede transportarse en menos tiempo y, de esta manera, se minimiza el tiempo que el procesador tiene que estar esperando a que le lleguen los datos que ha pedido leer o escribir, generalmente de memoria. Por otra parte, los buses también tienen una velocidad asociada que, evidentemente, influye en el rendimiento de la máquina. Evidentemente, no es lo mismo una autovía en el límite de velocidad sea de 80 Km/hora que una en la que se pueda circular a 120.

### 3.3.1.3 Bus de dirección

Envía y recibe “direcciones” de todos los elementos para conocer dónde están los datos.

Permiten al microprocesador seleccionar una de las tantas posiciones de memoria, ya sea para lectura o escritura. Se dice que es unidireccional, pues tan sólo es el procesador el que puede poner información en este bus; el resto de elementos del sistema tan sólo puede leerlo. Cuanto mayor sea este bus, mayor será la cantidad de memoria que el microprocesador puede direccionar o encontrar y, por tanto, marca el máximo de memoria principal que un ordenador puede tener. Así, por ejemplo, con un bus de direcciones de 32 bits, se pueden direccionar  $2^{32}$  posiciones de memoria, o lo que es lo mismo, la memoria puede ser de  $2^{32}$  bytes. Esto es, aproximadamente, un tamaño de 4 GB (Gigabytes).

## 3.4 PERIFÉRICOS

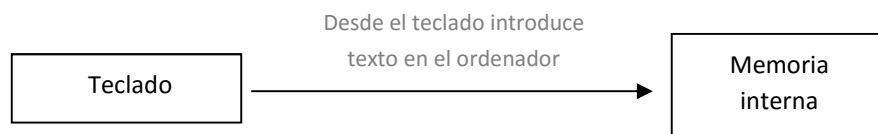
Son dispositivos que sirven para introducir y sacar datos del ordenador.

### 3.4.1 CLASIFICACIÓN DE LOS PERIFÉRICOS

#### 3.4.1.1. Periféricos de entrada

Son los que sirven para introducir datos en el ordenador.

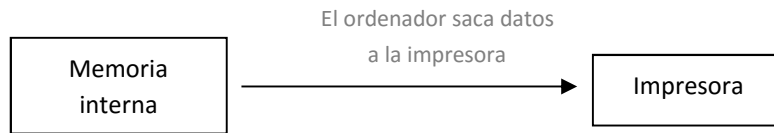
Ejemplos - Teclado, escáner, unidad lectora de CD-ROM, ratón (mouse), joystick....



#### 3.4.1.2. Periféricos de salida

Son los que sirven para sacar datos del ordenador.

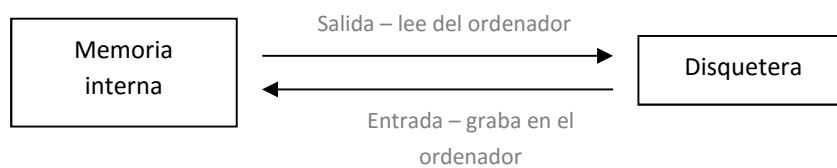
Ejemplos - Impresora, plotter, monitor (pantalla), .....



### 3.4.1.3. Periféricos de entrada/salida

Son los que sirven para ambas cosas: introducir datos en el ordenador y sacar datos del ordenador.

Ejemplos - Unidades de disquete  $\equiv$  disqueteras, los módems, pantallas táctiles, unidades ZIP, CD-RW, discos duros.....



## 4 LA INFORMACIÓN Y SU REPRESENTACIÓN INTERNA

Anteriormente se ha visto que el ordenador es la máquina que se utiliza para procesar (recoger, tratar, almacenar y mostrar) información. Pero, ¿cómo representa internamente dicha información el ordenador? En este apartado se tratará de dar respuesta a tal pregunta y se verá cómo el ordenador representa la información internamente cuando ésta es numérica (números) y cuando es alfanumérica (letras y caracteres especiales).

**Los ordenadores, debido a su construcción, solamente pueden trabajar en forma binaria.** Un ordenador está compuesto de circuitos electrónicos sobre los cuales sólo se puede evaluar si hay o no hay corriente; por lo tanto, sólo se reconocen dos estados o valores:

- “1” si hay tensión o corriente en un punto
- “0” si no hay tensión.

Sin embargo, el ordenador, para comunicarse con nosotros (para mostrarnos la información), no utiliza el sistema binario, sino que utiliza otros sistemas de numeración como son el **octal**, **hexadecimal** y **decimal**.

Antes de continuar avanzando, debemos tener claro qué es un sistema de numeración.

### 4.1 Sistemas de numeración

Los sistemas de numeración son un conjunto de símbolos y reglas que nos sirven para representar ciertas cantidades.

### 4.2 Codificación de la información

En informática es frecuente codificar la información. Codificación es la transformación que representa los elementos de un conjunto mediante los de otro, de tal forma que a cada elemento del primer conjunto le corresponde un elemento distinto del segundo. Ejemplos de códigos son:

- El código de provincia en las matrículas de los coches.
- El código de enfermedades por la OMS.
- El número de carnet de identidad.

Con los códigos se puede comprimir y estructurar la información. Pueden definirse códigos con significado.

En el interior de los computadores la información se almacena y se transfiere de un sitio a otro según un código binario representado por 0 y 1. En el interior de los computadores se efectúan automáticamente los cambios de código oportunos para que en su exterior la información sea directamente comprendida por los usuarios.

### 4.3 Bit, byte y sus múltiplos

**Bit:** La unidad más elemental de la información es un valor binario, conocido como BIT. El origen de este término es inglés, y se suele considerar que procede de la contracción de las palabras Blnary y digiT. Un bit es, por tanto, una posición o variable que toma el valor 0 o 1. Obviamente la capacidad mínima de almacenamiento de información en el interior de un computador es el bit, pudiéndose medir la capacidad de memoria de un computador en bits. La información se representa por bits. Por tanto a cada carácter le corresponde un cierto número de bits

**Byte:** Un byte es el número de bits necesarios para almacenar un carácter. Este número depende del código utilizado por el computador, siendo generalmente 8, por lo que habitualmente byte se utiliza como sinónimo de 8 bits u octeto. La capacidad de almacenamiento de un ordenador se mide en bytes y sus múltiplos.

#### Múltiplos:

- **KiloByte KB**  
1 024 Bytes =  $2^{10}$
- **MegaByte MB**  
1 024 KB  
1 048 576 Bytes =  $2^{20}$
- **GigaByte GB**  
1 024 MB  
1 048 576 KB  
1 073 741 824 Bytes =  $2^{30}$
- **TeraByte TB**  
1 024 GB  
1 048 576 MB  
1 073 741 824 KB  
1 099 511 627 776 Bytes =  $2^{40}$
- **PetaByte PB**  
1 024 TB  
1 048 576 GB  
1 073 741 824 MB  
1 099 511 627 776 KB  
1 125 899 906 842 624 Bytes =  $2^{50}$



#### 4.4 Sistemas Posicionales.

Los sistemas de numeración son sistemas posicionales. Eso significa que cada dígito tiene un valor definido por su posición:

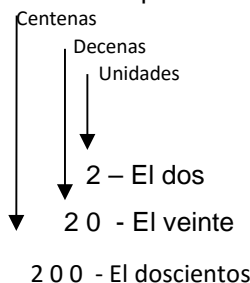
- El nº más a la derecha representa las unidades
- El siguiente más a la izquierda representa las decenas
- El siguiente las centenas
- Etc....

Ejemplo:

El número 256:



El número 2 dependiendo de la posición, será:



#### 4.5 Teorema fundamental de la numeración.

Este teorema fundamental de la numeración expone que la representación de una cantidad en un sistema, es dado por la fórmula:

$$N = \sum_{i=-d}^n X_i x B^i$$

Dónde:

B = base del sistema de numeración.

i = posición respecto al punto.

d = número de cifras a la derecha.

n = Numero de cifras a la izquierda del numero menos 1.

X = Cada una de las cifras que componen el número.

#### 4.6 El sistema decimal.

El sistema decimal es el más conocido por nosotros, pues es el que utilizamos todos los días. ¿Te has preguntado alguna vez por qué usamos el sistema decimal en vez de cualquier otro? Puede parecer un tanto absurdo, pero esto es muy probablemente debido a que tenemos 10 dedos para contar.

El sistema decimal:

- Es uno de los sistemas denominados **posicionales**, al igual que el resto de sistemas que vamos a estudiar, pues utiliza un conjunto de símbolos cuyo **significado o valor depende de su posición relativa al punto decimal**. Es decir, sabemos que no tiene el mismo valor un 1 si está situado en la posición de las unidades que si está situado en la posición de las centenas.
- La **base de este sistema de numeración es 10**, que es también la cantidad de cifras o símbolos distintitos que utiliza el sistema para la composición de los números.
- Los símbolos o cifras que utiliza el sistema decimal son los siguientes: **0 1 2 3 4 5 6 7 8 9**.

En este sistema, un número se expresa como una cadena de estas cifras, donde cada cifra aporta un valor al número, valor que depende tanto del valor intrínseco de la propia cifra, como de la posición que ocupa en la cadena, al ser un sistema de numeración posicional.

Un determinado valor o cantidad, que denominaremos número decimal, se puede expresar de la siguiente forma:

$$valor = \sum_{i=0}^{n-1} (dígito_i * base^i)$$

donde:

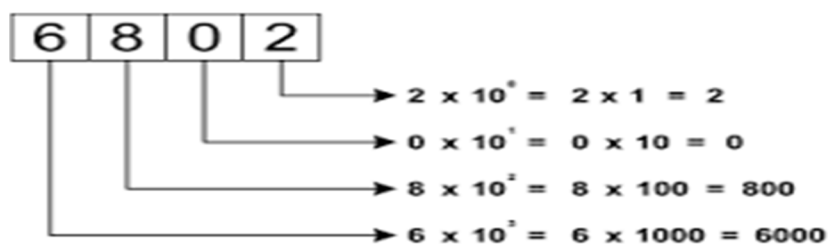
base = 10

i = Posición de la cifra en la cadena numérica, donde la posición 0 es la de más a la derecha

n = Número de dígitos

dígito = Cada una de las cifras que componen el número

*El número 6802, por ejemplo, tiene 4 cifras. En el sistema decimal, se construye de la siguiente forma, respetando las posiciones correspondientes:*



A ver si puedes contestar a estas preguntas:

- ¿Cuántos números distintos se pueden representar con 1 cifra decimal? Evidentemente, se pueden representar 10 números, los que van del 0 al 9.
- ¿Cuántos números distintos se pueden representar con 2 cifras decimales? Con dos cifras se pueden representar 100 números, los que van del 0 (00) al 99.
- ¿Cuántos números distintos se pueden representar con 3 cifras decimales? Con tres cifras decimales se pueden representar 1000 números, los que van del 0 (000) al 999.
- ¿cuántos números distintos se pueden representar con n cifras decimales? **Con n cifras decimales se pueden representar  $10^n$  números, los que van del 0 al  $10^n-1$ .**

## 4.7 El sistema binario.

Es el sistema que utilizan internamente los circuitos digitales que conforman el hardware de los ordenadores.

El sistema binario:

- La base de este sistema de numeración es 2. La base es 2, lo que quiere decir que sólo disponemos de 2 símbolos {0,1} para construir todos los números binarios.
- Al igual que el sistema decimal, el sistema binario es un sistema de numeración posicional, que recordemos que quiere decir que el valor de cada cifra viene dado tanto por su valor intrínseco como por su posición dentro de la cadena de cifras que forman el número binario.
- Los símbolos o cifras que se utilizan para la representación de los números son exclusivamente los siguientes: 0 1.
- Cada cifra o dígito de un número representado en este sistema se denomina bit, que es la menor unidad de información posible en un ordenador.

Así, por ejemplo:

1 bit = se refiere a un número de 1 cifra binaria 2 bits = se refiere a un número de 2 cifras binarias . . . n bits = se refiere a un número de n cifras binarias

En la figura inferior podemos ver el teorema fundamental de la numeración aplicado al sistema binario.

$$N = d_n \dots d_1 d_0 d_{-1} \dots d_{-k} =$$

$$d_n \cdot 2^n + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0 + d_{-1} \cdot 2^{-1} + \dots + d_{-k} \cdot 2^{-k} =$$

$$N = \sum_{i=-k}^n d_i \cdot 2^i$$

En el apartado anterior hemos formulado una serie de preguntas cuyas respuestas seguro que has encontrado sencillas al tratarse del sistema decimal. ¿Crees que serías capaz de responder a las mismas preguntas pero formuladas sobre el sistema de numeración binario? Comprobémoslo:

- ¿Cuántos números distintos se pueden representar con 1 cifra binaria; es decir, con 1 bit? Con un bit se pueden representar  $2^1$  números; es decir, 2 números, los que van del 0 al 1.
- ¿Cuántos números distintos se pueden representar con 2 cifras binarias? Con dos bits se pueden representar  $2^2$  números; es decir, 4 números:

00 = valor 0    01 = valor 1

10 = valor 2    11 = valor 3

Es decir, los que van del 00 al 11, o lo que es lo mismo, los que van del valor 0 al valor 3.

- ¿Cuántos números distintos se pueden representar con 3 cifras binarias? Con tres bits se pueden representar  $2^3$  números; es decir, 8 números:

000 = valor 0    001 = valor 1    010 = valor 2    011 = valor 3

100 = valor 4    101 = valor 5    110 = valor 6    111 = valor 7

Es decir, los que van del 000 al 111, o lo que es lo mismo, los que van del valor 0 al valor 7.

- Generalizando, ¿cuántos números distintos se pueden representar con  $n$  cifras binarias? **Con  $n$  cifras binarias se pueden representar  $2^n$  números, los que van del 0 al  $2^n - 1$ .**

#### Ejemplos:

El número binario 111 está formado por un solo símbolo repetido tres veces. No obstante, cada uno de esos símbolos tiene un valor diferente, que depende de la posición que ocupa en el número. Así, el primer 1 (empezando por la izquierda) representa un valor de 4 en decimal ( $2^2$ ), el segundo de 2 ( $2^1$ ) y el tercero de 1 ( $2^0$ ), dando como resultado el valor del número:  $111 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 4 + 2 + 1 = 7$ .

### 4.8 El sistema octal.

El sistema octal:

- La base de este sistema de numeración es 8. La base es 8, lo que quiere decir que disponemos de 8 símbolos  $\{0,1,2,3,4,5,6,7\}$  para construir todos los números.
- Al igual que el sistema decimal y binario, el sistema octal es un sistema de numeración posicional.
- Los símbolos o cifras que se utilizan para la representación de los números son exclusivamente los siguientes: 0 1 2 3 4 5 6 7.

En la figura inferior podemos ver el teorema fundamental de la numeración aplicado al sistema octal.

$$\begin{aligned}
 N &= d_n \dots d_1 d_0, d_{-1} \dots d_{-k} &= \\
 d_n \cdot 8^n + \dots + d_1 \cdot 8^1 + d_0 \cdot 8^0 + d_{-1} \cdot 8^{-1} + \dots + d_{-k} \cdot 8^{-k} &= \\
 N &= \sum_{i=-k}^n d_i \cdot 8^i
 \end{aligned}$$

### 4.9 El sistema hexadecimal.

El sistema hexadecimal actual fue introducido en el ámbito de la computación por primera vez por IBM en 1963.

El sistema hexadecimal:

- La base de este sistema de numeración es 16. La base es 16, lo que quiere decir que disponemos de 16 símbolos  $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$  (dado que el sistema usual de numeración es de base decimal y, por ello, sólo se dispone de diez dígitos, se adoptó la convención de usar las seis primeras letras del alfabeto latino para suplir los dígitos que nos faltan  $A = 10$ ,  $B = 11$ ,  $C = 12$ ,  $D = 13$ ,  $E = 14$  y  $F = 15$ ) para construir todos los números.
- Al igual que el sistema decimal, binario y octal, el sistema hexadecimal es un sistema de numeración posicional.
- Los símbolos o cifras que se utilizan para la representación de los números son exclusivamente los siguientes: 0 1 2 3 4 5 6 7 8 9 A B C D E F.

En la figura inferior podemos ver el teorema fundamental de la numeración aplicado al sistema hexadecimal.

## 4.10 Conversión entre Decimal, Binario, Hexadecimal y Octal.

### 4.10.1 Conversión de Binario a Decimal.

Para poder transformar números binarios en su correspondiente decimal basta multiplicar el dígito binario (que sólo puede ser 0 o 1) por 2 elevado a la potencia correspondiente según la cantidad de dígitos de la cifra. Luego se suman los valores obtenidos y se consigue el número final.

Ejemplos:

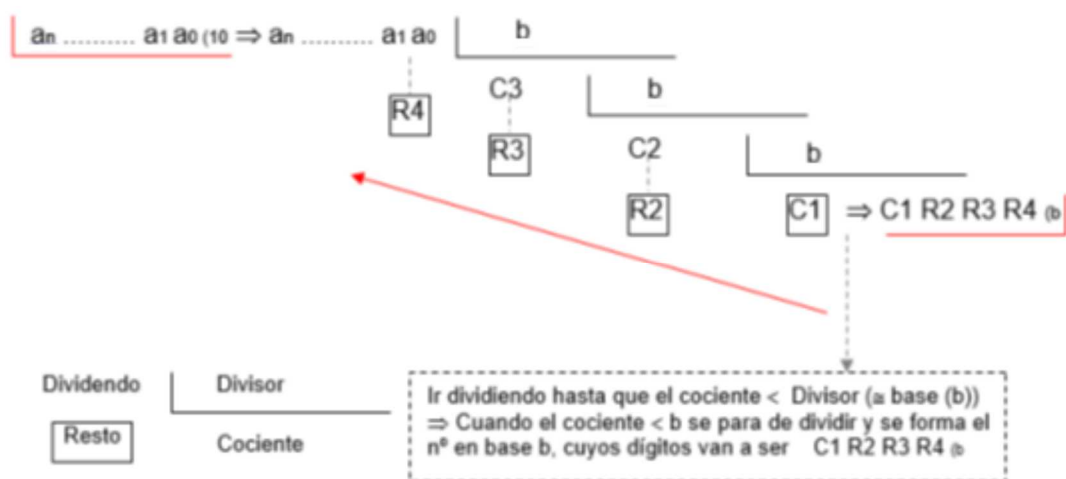
$$100011,011 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 32 + 0 + 0 + 0 + 2 + 1 + 0 + 0,25 = 35,25$$

$$101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = 5$$

### 4.10.2 Conversión de decimal a base b.

Para la parte entera:

El primer paso para transformar un número decimal en otro en base b es realizar sucesivas divisiones enteras del número por la base b.

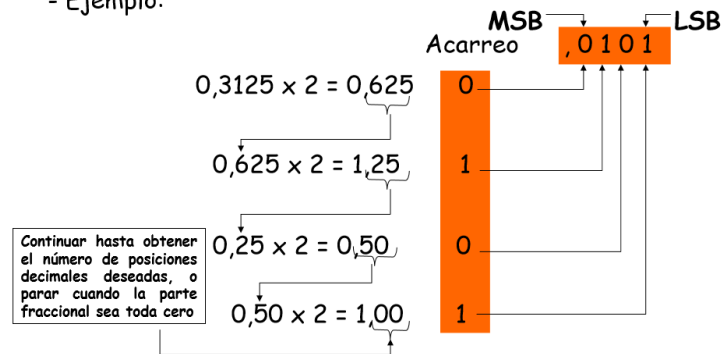


En el caso de que tenga parte fraccionaria:

Se multiplica la parte fraccionaria por la base y se van cogiendo la parte entera de resultado. Continuamos cogiendo la parte fraccionaria del resultado y volviendo a realizar la operación anterior, se va repitiendo esto hasta que la parte fraccionaria sea 0.

Ejemplo para pasar a base 2:

- Ejemplo:



### Conversión De Decimal A Binario:

Para convertir un número decimal a otro sistema, el número decimal es sucesivamente dividido por la base del sistema, en este caso la base del sistema binario es 2 el número será sucesivamente dividido entre 2 y el resultado del cociente será nuevamente dividido entre 2 y así sucesivamente hasta que el cociente sea 0. El resto de cada división es un número binario que conforma el número resultante de la conversión. El primer resultado producido (el primer resto obtenido) corresponde al bit más próximo al punto decimal (o lo que se conoce como bit de menor peso). Los sucesivos bits se colocan a la izquierda del anterior. Nótese que esto es como escribir en sentido contrario al empleado normalmente.

**Ejemplo:** Pasamos el número decimal 42 a binario.

Dividimos el número 42 entre 2.

Dividimos el cociente obtenido por 2 y repetimos el mismo proceso hasta que el cociente sea 1.

El numero binario lo formamos tomando el primer dígito el ultimo cociente, seguidos por los restos obtenidos en cada división, seleccionándolos de derecha a izquierda, como se muestra en el siguiente esquema.



El resultado será 101010, si hacemos la transformación inversa de nuevo a decimal tenemos:

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 32 + 0 + 8 + 0 + 2 + 0 = 42$$

### Conversión De Decimal A Octal:

En este caso basta usar el mismo método de conversión que con los números binarios. Pero en vez de hacer divisiones sucesivas entre 2 hay que efectuarlas

entre 8. Nótese que el divisor corresponde a la base del sistema al cual se va a convertir.

**Lo mismo podríamos hacer para pasar de decimal a hexadecimal, divisiones sucesivas entre 16.**

#### 4.10.3 Conversión de Hexadecimal a Binario.

El sistema hexadecimal se utiliza principalmente como método abreviado para representar números binarios, cada dígito hexadecimal se convierte en su equivalente binario de 4 bits.

La conversión entre Hexadecimal y Binario es casi inmediata, ya que cada dígito Hexadecimal tiene una correspondencia directa con un conjunto de 4 dígitos en binario, por ser ambos bases múltiplos de 2.

Para realizar la conversión se toma cada uno de los dígitos Hexadecimal y se pasan a grupos de 4 dígitos binarios con la siguiente tabla.

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

9F2(H)= 100111110010 (2)

9= 1001 F= 1111 2= 0010

10100111 = A7<sub>(16)</sub>      2E = 00101110<sub>(2)</sub>

1010<sub>(2)</sub> = 10<sub>(10)</sub> = A<sub>(16)</sub>      E<sub>(16)</sub> = 14<sub>(10)</sub> = 1110<sub>(2)</sub>

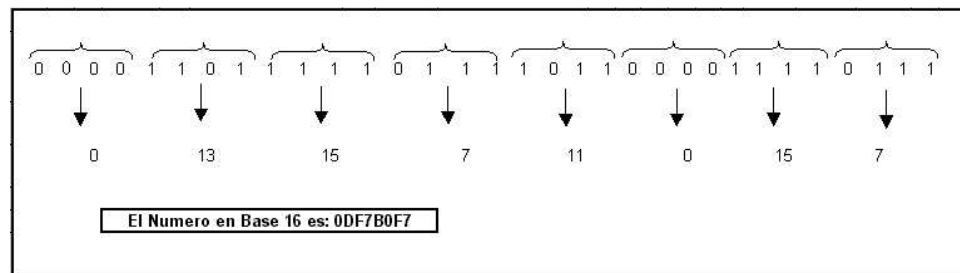
#### 4.10.4 Conversión de Binario a Hexadecimal.

Para realizar la conversión de binario a hexadecimal:

- Agrupamos la cantidad binaria en grupos de 4 en 4 iniciando por el lado derecho. Si al terminar de agrupar no completas 4 dígitos, entonces agrega ceros a la izquierda.
- Posteriormente cada grupo de 4 se pasa a decimal y se pone el dígito que le corresponde teniendo en cuenta que a partir del 10 van las letras de la A a la F:

1111 1101 0111 0011  
 ← ← ← ←  
 F D 7 3





#### 4.10.5 Conversión de Octal a Binario.

La conversión entre Octal y Binario es casi inmediata, ya que cada dígito Octal tiene una correspondencia directa con un conjunto de 3 dígitos en binario, por ser ambas bases múltiplos de 2.

Para realizar la conversión se toma cada uno de los dígitos Octal y se pasan a grupos de 3 dígitos binarios con la siguiente tabla.

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

#### 4.10.6 Conversión de Binario a Octal.

Para realizar la conversión de binario a Octal:

- Agrupamos la cantidad binaria en grupos de 3 en 3 iniciando por el lado derecho. Si al terminar de agrupar no completas 3 dígitos, entonces agrega ceros a la izquierda.
- Posteriormente cada grupo de 3 se pasa a decimal y se pone el dígito que le corresponde.

### 4.11 Representación interna de números enteros.

¿Recuerdas cuáles son los números enteros? Básicamente, los números enteros son los que no tienen decimales, cubriendo desde el menos infinito hasta el infinito; es decir,  $-\infty, \dots, -2, -1, 0, 1, 2, \dots, +\infty$ .

¿Puede representar el ordenador todos estos números? Y, de ser así, ¿cómo los representa internamente? En este apartado trataremos de dar respuesta a estas preguntas.

Coge una calculadora y empieza a realizar operaciones encadenadas de multiplicación de manera indiscriminada. ¿Qué sucede? Verás que llega un momento en el que aparece en la pantalla un símbolo extraño, generalmente como una E al revés. Este símbolo quiere decir que el número resultante de la operación no "cabe" en la pantalla, se trata de un número no representable que desborda la capacidad de representación de números de la calculadora. Es decir, es un número mayor de lo que la calculadora puede manejar.

Algo idéntico ocurre con los ordenadores, que son máquinas al fin y al cabo y, por tanto, limitadas, pues tienen una capacidad de almacenamiento finita. El ordenador, al igual que la calculadora, tiene una capacidad de cálculo limitada, y habrá unos límites en cuanto a los números que es capaz de representar y manejar.

Estas limitaciones vendrán dadas por la arquitectura del mismo ordenador, generalmente por el tamaño de los espacios de memoria que utiliza para realizar las

operaciones, pero también por la manera que tenga de representar los números internamente, algo que también influirá en la cantidad de números que se pueden representar.

Por eso es tan importante conocer, al menos un poco, la manera que tiene el ordenador de representar los números internamente, para ser conscientes de qué es representable y de qué no, para conocer los límites de la máquina. Fíjate si es importante que muchos de los errores más graves que se han producido en la historia de las aplicaciones informáticas, y que han provocado grandes incidentes y el fracaso de proyectos enteros, han tenido su origen en la no consideración de aspectos de este tipo.

Supón que queremos almacenar en el ordenador información acerca de la edad de una persona en años. Cuando la persona no llega al año, su edad será de cero, y además, no puede haber una persona que tenga una edad negativa. Por lo tanto, de todos los números enteros, para almacenar información de este tipo, sólo necesitamos los números positivos y el cero.

Dentro de los números enteros, existe un subconjunto, el de los números naturales que son los enteros positivos más el cero; es decir, los números: 0, 1, 2, 3, 4,...  $\infty$ . En el mundo de la informática los números naturales se conocen con el nombre de enteros sin signo. Los ordenadores suelen usar una capacidad de 32 bits para almacenar los números enteros, aunque es algo que depende de la arquitectura del propio procesador. Este espacio, el número de bits que se van a poder utilizar para representar los números va a determinar, evidentemente, la cantidad de números representables.

Para **representar los números enteros** en el ordenador utiliza lo que se conoce con el nombre de **representación en Coma fija o punto fijo**.

Hay 3 formas de representar los números en coma fija:

### 4.11.1 Binario puro

Se utiliza una combinación de 32 bits, en la que:

El bit de la izquierda sirve para representar el signo:

- El 0 para el signo +
- El 1 para el signo -

Los restantes 31 bits sirven para representar el valor del número

El  $n^{\circ}$  -10 tendría la siguiente forma:

Signo																													2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
0 +	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
1 -	32	31																											4	3	2	1

### 4.11.2 Decimal desempacotado

Representa cada número decimal, de tal forma que cada una de sus cifras ocupa un byte u octeto (8 bits), en los que:

- Los 4 bits de la izquierda son cuatro unos denominados bits de zona.
- Los 4 bits de la derecha representan las cifras

Exceptuando los 8 bits de más a la derecha:

- Los 4 bits de la izquierda representan el signo:
  - El 1100 para el signo +
  - El 1101 para el signo -
- Los 4 bits de la derecha representan la cifra.

Representar el n° 2371 decimal en decimal desempaquetado:

1 1 1 1 0 0 1 0   1 1 1 1 0 0 1 1   1 1 1 1 0 1 1 1   1 1 0 0 0 0 0 1

2                      3                      7                      Signo+                      1

### 4.11.3 Decimal empaquetado

Representa cada cifra con un conjunto de 4 bits. El conjunto de 4 bits de más a la derecha se usa para representar el signo con la misma combinación que en el caso anterior.

- El 1100 para el signo +
- El 1101 para el signo -

Representar el nº 2371 decimal en decimal empaquetado:

0 0 1 0   0 0 1 1   0 1 1 1   0 0 0 1   1 1 0 0  
2            3            7            1            Signo +

#### 4.11.4 Exceso a $2^{n-1}$

En Exceso a  $2^{n-1}$ , si se dispone de **n** bits para representar a un número entero (**N**) positivo o negativo, dicho número se representa como **N +  $2^{n-1}$** , por tanto:

$$NEX = N + 2^{n-1}$$

- Ejemplo:

Si  $n=8$  bits  $\rightarrow$  Exceso =  $2^{8-1} = 2^7 = 128 \rightarrow$  Cuando nos den un n° a representar, le sumaremos 128

$$10 \rightarrow 10 + 128 = 138 \rightarrow 10001010$$

$$-10 \rightarrow -10 + 128 = 118 \rightarrow 01110110$$

$$109 \rightarrow 109 + 128 = 237 \rightarrow 11101101$$

$$-109 \rightarrow -109+128 = 19 \rightarrow 00010011$$

Por otro lado, para calcular el valor en base 10 de un número entero (**N**) escrito en Exceso a  $2^{n-1}$ , se tiene que utilizar la fórmula:

$$N_{EX} = ( (\sum_{i=0}^{n-1} a_i \cdot 2^i) - 2^{n-1} )_{10}$$

Figura - Fórmula para calcular, en base 10, el valor de un número entero escrito en Exceso a  $2^{n-1}$ .

## 4.12 Representación interna de números reales.

Ya hemos visto un poco cómo representa el ordenador los números enteros. Pero los números sin decimales no son suficientes. En algunas ocasiones necesitamos números con decimales, por ejemplo, para representar los precios de las cosas en euros. Aparte de los números enteros existe lo que se conoce con el nombre de números reales; es decir, números que tienen parte entera y parte decimal, y estos también son representables y manejables por el ordenador. Para representar los números reales el ordenador utiliza lo que se conoce con el nombre de **representación en punto flotante**. Además, este tipo de representación no sólo sirve para representar los números reales, sino también para representar aquellos números enteros no representables con los mecanismos descritos en el apartado anterior para la representación de números enteros. Veamos cómo es esta forma de representación de números.

Para codificar un número real en punto flotante primero hay que pasar éste a notación científica. Así, por ejemplo, el número 30'0008 vendría expresado como  $0'300008 * 10^2$ .

En este sistema, un número se representa con tres campos:

- **1 bit de signo**, que será 0 si el número es positivo y 1 si es negativo.
- **Exponente**. Recoge la codificación del número que hace de exponente en la notación científica; en el ejemplo anterior habría que codificar el 2, pues era  $10^2$ . Esto no es exactamente así, puesto que este campo supone que la base de exponenciación es 2 en vez de 10, por lo que primero habría que resolver la siguiente ecuación:  $10^2 = 2^x$  y calcular el valor de x, que es lo que habría que codificar en el campo Exponente.
- **Mantisa**. Recoge la codificación del número después del 0' de la notación científica.

**Representación de los reales.**

$$N^o = \text{mantisa} * \text{base}^{\text{Exponente}}$$

Por ejemplo...

$$345 = 0.345 * 10^3$$

Para que los números representados en punto flotante fueran posibles intercambiar entre distintas arquitecturas se establece el **estándar IEEE 754** que define el formato y las operaciones con estos. El estándar IEEE 754 usa la base 2 ( $b = 2$ ) y su mantisa normalizada es de la forma 1,F. La representación utiliza el signo, el exponente (codificado usando representación de números con signo en exceso  $2^{n-1}$ ) y la parte fraccional de la mantisa (lo que está después de la “coma binaria”). En definitiva el número se representa por la terna de códigos binarios S (codificación del signo), E (codificación del exponente en desplazamiento), F (codificación de la parte fraccional de la mantisa en binario).

El estándar define tres formatos (en función de la cantidad de bits utilizados):

	S (bits)	E (bits)	F (bits)	Total (bytes)
Simple Precisión	1	8	23	4
Doble Precisión	1	11	52	8
Precisión Extendida	1	15	64	10

Los números se almacenan de la siguiente forma:



### 4.13 Codificación alfanumérica.

Ya hemos visto cómo el ordenador trabaja con la información de tipo numérica, ya sea con números enteros o reales. Pero, aparte de con números, el ordenador también trabaja con letras o, dicho de manera más genérica, con caracteres. En el mundo de la informática a este tipo de dato recibe el nombre de datos alfanuméricos. Entre dichos datos alfanuméricos están los siguientes:

- Caracteres alfabéticos:
  - Letras mayúsculas. De la A a la Z (sin la Ñ)
  - Letras minúsculas. De la a a la z (sin la ñ)
- Cifras decimales:
  - Son las cifras 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, pero tratados como caracteres.
- Caracteres especiales:
  - Caracteres como: el punto, la coma, el punto y coma, el asterisco, etc.
  - Órdenes de control: NUL, CR, ACK, etc. Estos caracteres no tienen representación con un símbolo, sino que sirven para mandarle al ordenador alguna orden o instrucción especial.

Sin embargo, como sabemos, el ordenador sólo es capaz de trabajar en binario; es decir, de representar y almacenar ceros y unos. ¿Cómo puede entonces el ordenador representar y almacenar un carácter? ¿No hay aquí una cierta contradicción? Para la representación de información alfanumérica se utiliza lo que se conoce con el nombre de código. Un código no es más que una tabla de equivalencia en la que a cada carácter o símbolo que se quiere representar se le asigna un número binario, de tal manera que lo que guarda el ordenador internamente es dicho número binario. Sin embargo, a la hora de interpretarlo, se busca en la tabla y se toma el carácter alfanumérico asociado.

**Los primeros códigos utilizados fueron de 6 bits.** Esto quiere decir que a cada carácter se le asigna un número de 6 bits, lo que implica que este tipo de códigos permitían la representación de  $2^6$  caracteres distintos (64 caracteres) que corresponde a 26 letras mayúsculas, 10 cifras numéricas (0 1 2 3 4 5 6 7 8 9) y 28 caracteres denominados especiales. Un ejemplo de código de 6 bits es el código **FIELDATA**, que se muestra a continuación:

bits	5 4 3							
2 1 0	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0	@	C	K	S	)	*	0	8
0 0 1	[	D	L	T	-	(	1	9
0 1 0	]	E	M	U	+	%	2	'
0 1 1	#	F	N	V	<	:	3	;
1 0 0	^	G	O	W	=	?	4	/
1 0 1	SP	H	P	X	>	!	5	.
1 1 0	A	I	Q	Y	&	,	6	"
1 1 1	B	J	R	Z	\$	\	7	_

En dicho código, para almacenar el carácter T, el ordenador internamente deberá guardar el número binario de 6 bits 011001.

Con el nacimiento de lenguajes de programación de alto nivel comenzaron a utilizarse **códigos de 7 bits**, que permiten la utilización de los mismos caracteres que existían en el FIELDATA, añadiendo las letras minúsculas y caracteres cuyo significado son órdenes de control entre periféricos. Un ejemplo de códigos de 7 bits es el código **ASCII de 7 bits**, que se muestra a continuación:

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

Hoy en día los códigos más utilizados son el **ASCII extendido, de 8 bits**, y **UNICODE, de 16 bits** y de amplio uso en Internet y el mundo de las redes en general.



#### 4.14 Operaciones elementales con números binarios.

La Unidad Aritmético Lógica, en la CPU del procesador, es capaz de realizar operaciones aritméticas, con datos numéricos expresados en el sistema binario. Naturalmente, esas operaciones incluyen la adición, la sustracción, el producto y la división. Las operaciones se hacen del mismo modo que en el sistema decimal, pero





La resta  $0 - 1$  se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente. Esa unidad prestada debe devolverse, sumándola, a la posición siguiente.

-	0	1
0	0	1
1	1 + 1	0

Veamos algunos ejemplos:

$$\begin{array}{r}
 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0 \\
 - 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1 \\
 \hline
 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1
 \end{array}
 \quad \text{acarreo}$$

$$111 - 101 = 010$$

$$10001 - 01010 = 00111$$

$$11011001 - 10101011 = 00101110$$

$$111101001 - 101101101 = 001111100$$

#### 4.14.3 Complemento a dos

El **complemento a dos** de un número  $N$ , compuesto por  $n$  bits, se define como:

$$C2N = 2^n - N$$

Veamos un ejemplo: tomemos el número  $N = 101101_2$ , que tiene 6 bits, y calculemos su complemento a dos:

$$N = 45_{10} \quad n = 6 \quad 2^6 = 64 \quad \text{y, por tanto: } C2N = 64 - 45 = 19 = 010011_2$$

#### 4.14.4 Complemento a uno

El **complemento a uno** de un número  $N$ , compuesto por  $n$  bits es, por definición, una unidad menor que el complemento a dos, es decir:

$$C1N = C2N - 1$$

y, por la misma razón:

$$C2N = C1N + 1$$

Calculemos el complemento a uno del mismo número del ejemplo anterior:



Siendo  $N = 101101$ , y su complemento a dos  $C2N = 010011$

$$C1N = C2N - 1 = 010011 - 000001 = 010010$$

$$C1N = 010010$$

En realidad, **el complemento a uno de un número binario es el número resultante de invertir los UNOS y CEROS de dicho número.**

Los pasos más rápidos para pasar a complemento a 1 y complemento a 2 son:

- Primero se pasa a complemento a 1 cambiando los 0 por 1 y 1 por 0.
- Y a continuación si queremos el complemento a 2 se coge el complemento a 1 creado de la forma anterior y se le suma 1 en binario.

$$\begin{array}{r}
 9 = 1001 \\
 \quad 0110 \rightarrow \text{Complemento a 1} \\
 + \quad 1 \rightarrow \text{Se suma 1 al LSB} \\
 \hline
 \quad 0111 \rightarrow \text{Complemento a 2}
 \end{array}$$

#### 4.14.5 Restar en binario usando el complemento a dos

Y, por fin, vamos a ver cómo facilita la resta el complemento. La resta binaria de dos números puede obtenerse sumando al minuendo el complemento a dos del sustraendo.

**EJEMPLO1:** Hagamos la siguiente resta:

$$91 - 46 = 45, \text{ en binario:}$$

$$1011011 - 0101110 = 0101101$$

Tiene alguna dificultad, cuando se acumulan los arrastres a la resta siguiente. Pero esta misma resta puede hacerse como una suma, utilizando el complemento a dos del sustraendo:

$$1011011 + 1010010 = 0101101$$

En el resultado de la suma nos sobra un bit, que se desborda por la izquierda. Pero, como el número resultante no puede ser más largo que el minuendo, el bit sobrante se desprecia.

**EJEMPLO2:** Hagamos esta otra resta,  $219 - 23 = 196$ , utilizando el complemento a dos:

$$219_{(10)} = 11011011_2$$

$$23_{(10)} = 00010111_2$$

$$C2\ 23 = 11101001$$

$$\text{El resultado de la resta será: } 11011011 + 11101001 = 111000100$$

Y, despreciando el bit que se desborda por la izquierda, llegamos al resultado correcto:

$$11000100_2 = 196_{(10)}$$

#### 4.14.6 Multiplicación binaria

La multiplicación en binario es más fácil que en cualquier otro sistema de numeración. Como los factores de la multiplicación sólo pueden ser CEROS o UNOS, el producto sólo puede ser CERO o UNO. En otras palabras, las tablas de multiplicar del cero y del uno son muy fáciles de aprender:

	x	0	1
--	---	---	---

0	0	0
1	0	1

En un ordenador, sin embargo, la operación de multiplicar se realiza mediante sumas repetidas.

Veamos, por ejemplo, una multiplicación:

$$\begin{array}{r}
 110100010101 \\
 \times \quad \quad 1101 \\
 \hline
 110100010101 \\
 000000000000 \\
 110100010101 \\
 110100010101 \\
 \hline
 1010101000010001
 \end{array}$$

#### 4.14.7 División binaria

Igual que en el producto, la división es muy fácil de realizar, porque no son posibles en el cociente otras cifras que UNOS y CEROS.

Consideremos el siguiente ejemplo,  $42:6 = 7$ , en binario:

$$\begin{array}{r}
 101010 \quad | \quad 110 \\
 -110 \quad \quad 111 \\
 \hline
 1001 \\
 -110 \\
 \hline
 0110 \\
 110 \\
 \hline
 000
 \end{array}$$

Se intenta dividir el dividendo por el divisor, empezando por tomar en ambos el mismo número de cifras (101 entre 110, en el ejemplo). Si no puede dividirse, se intenta la división tomando un dígito más (1010 entre 110).

Si la división es posible, entonces, el divisor sólo podrá estar contenido una vez en el dividendo, es decir, la primera cifra del cociente es un UNO. En ese caso, el resultado de multiplicar el divisor por 1 es el propio divisor. Restamos las cifras del dividendo del divisor y bajamos la cifra siguiente.

El procedimiento de división continúa del mismo modo que en el sistema decimal.

## 5. CIRCUITOS INTEGRADOS

Un **circuito integrado** es una **combinación de elementos de un circuito que están miniaturizados y que forman parte de un mismo chip o soporte**. Dicho circuito aceptará una serie de valores de entrada y devuelve unas salidas que dependerán de los valores dados, además están fabricados con una función específica como pueden ser: Operaciones Aritméticas, funciones lógicas, amplificación, codificación, decodificación, controladores, etc

Los circuitos integrados están constituidos por resistencias, condensadores, diodos, transistores y demás dispositivos eléctricos que permiten el paso de la señal. La señal que se introduce en cada entrada (voltaje) generará por dicho circuito interno unos valores concretos de salida.

A su vez, estas resistencias, transistores, etc. forman las denominadas **puertas lógicas**.

Estos Circuitos Integrados por lo general se combinan para formar sistemas mucho más complejos que pueden ser desde una calculadora, un reloj digital, un videojuego, hasta una computadora, etc

El circuito integrado está elaborado con un material semiconductor, sobre el cual se fabrican los circuitos electrónicos a través de la fotolitografía. Estos circuitos, que ocupan unos pocos milímetros, se encuentran protegidos por un encapsulado con conductores metálicos que permiten establecer la conexión entre dicha pastilla de material semiconductor y el circuito impreso.

Otra de las características de los circuitos integrados es que rara vez se pueden reparar; es decir si un solo componente de un circuito integrado llegara a fallar, se tendría que cambiar la estructura completa; esto se debe al tamaño diminuto y los miles de componentes que poseen.

### 5.10 Clasificación de los Circuitos Integrados de acuerdo a su estructura.

Dependiendo del número de puertas lógicas que formen el circuito integrado podemos clasificarlo en los siguientes niveles o escalas de integración:

- **SSI**.- Significa Small Scale Integration (integración en pequeña escala) y comprende los chips que contienen menos de 12 puertas. Los primeros Circuitos Integrados eran SSI.
- **MSI**.- Significan Medium Scale Integration (integración en mediana escala), y comprende los chips que contienen de 13 a 100 puertas. ejemplos: codificadores, registros, contadores, multiplexores, de codificadores y de multiplexores.
- **LSI**.- Significa Large-Scale Integration (integración en alta escala) y comprende los chips que contienen de 100 a 1000 puertas. ejemplos: memorias, unidades aritméticas y lógicas (alu's), microprocesadores de 8 y 16 bits. La aparición de los circuitos integrados a gran escala, dio paso a la construcción del microprocesador.
- **VLSI**.- Significa Very Large Scale Integration (integración en muy alta escala) y comprende los chips que contienen de 1000 a 10.000 puertas. los cuales aparecen para consolidar la industria de los integrados y para desplazar definitivamente la tecnología de los componentes aislados y dan inicio a la era de la miniaturización de los equipos apareciendo y haciendo cada vez más común la manufactura y el uso de los equipos portátiles.

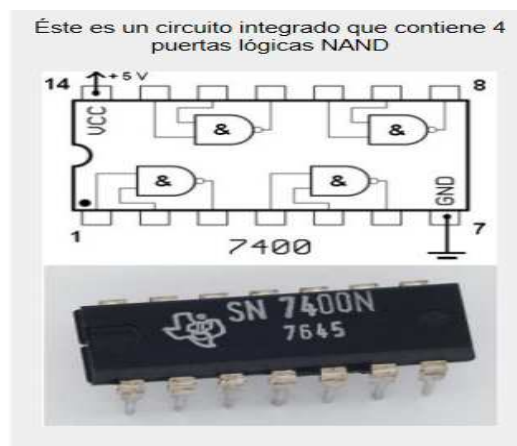
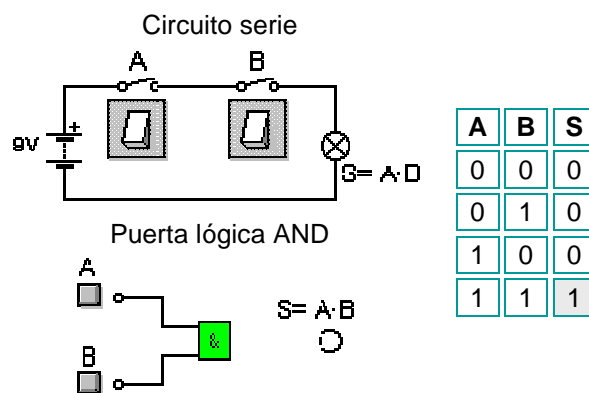
- **ULSI** Significa Ultra Large Scale Integration: Tecnología de circuitos integrados que utilizan entre 100.000 y un millón de transistores por circuito integrado, equivalentes a 10.000 a 100.000 puertas lógicas. Actualmente se utiliza para fabricar microprocesadores complejos.

## 5.11 Puertas lógicas.

Las puertas lógicas son circuitos electrónicos capaces de realizar operaciones lógicas básicas, además es la unidad básica sobre la que se diseña un circuito integrado y puede tener una o varias entradas que se convertirán en una sola salida

Una puerta lógica, o compuerta lógica, es un dispositivo electrónico con una función booleana. Suman, multiplican, niegan o afirman, incluyen o excluyen según sus propiedades lógicas.

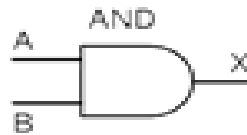
Por ejemplo, para realizar la operación producto utilizamos un circuito integrado a partir del cual se obtiene el resultado  $S = A \cdot B$



### 5.11.1 Puerta lógica AND.

La señal de salida se activa sólo cuando se activan todas las señales de entrada.

Equivala al producto lógico  $S = A \cdot B$ , representada por el siguiente símbolo y se corresponde con la siguiente tabla de la verdad (para dos entradas).

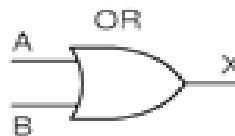


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

### 5.11.2 Puerta lógica OR.

La señal de salida se activa sólo cuando se activan alguna de las señales de entrada.

Equivale a la suma lógico  $S = A + B$  y representada por el siguiente símbolo y se corresponde con la siguiente tabla de la verdad (para dos entradas).

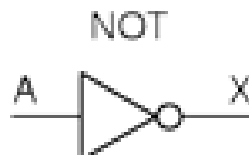


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

### 5.11.3 Puerta lógica NOT.

Esta puerta es un inversor que invierte el nivel lógico de una señal binaria. La señal de salida es la inversa de la señal de entrada.

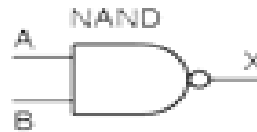
Se representa por el siguiente símbolo y se corresponde con la siguiente tabla de la verdad.



A	X
0	1
1	0

## 5.11.4 Puerta lógica NAND.

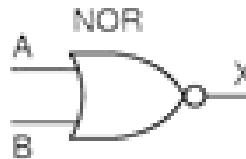
Es el complemento de la función AND (la puerta AND negada). Se representa por el siguiente símbolo y se corresponde con la siguiente tabla de la verdad.



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

## 5.11.5 Puerta lógica NOR.

Es el complemento de la función OR (la puerta OR negada). Se representa por el siguiente símbolo y se corresponde con la siguiente tabla de la verdad.

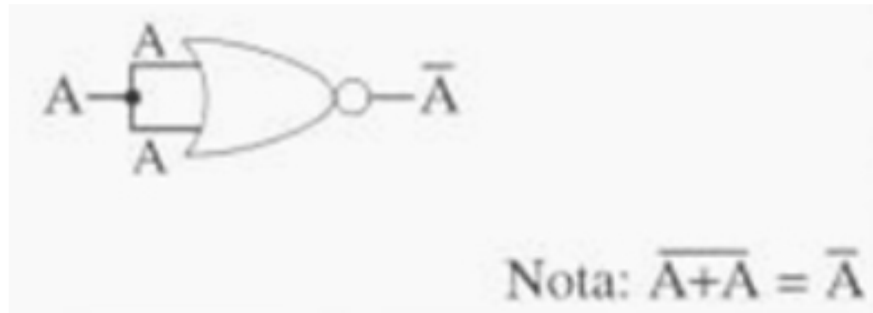


A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

## 5.11.6 Ejemplos con puertas lógicas.

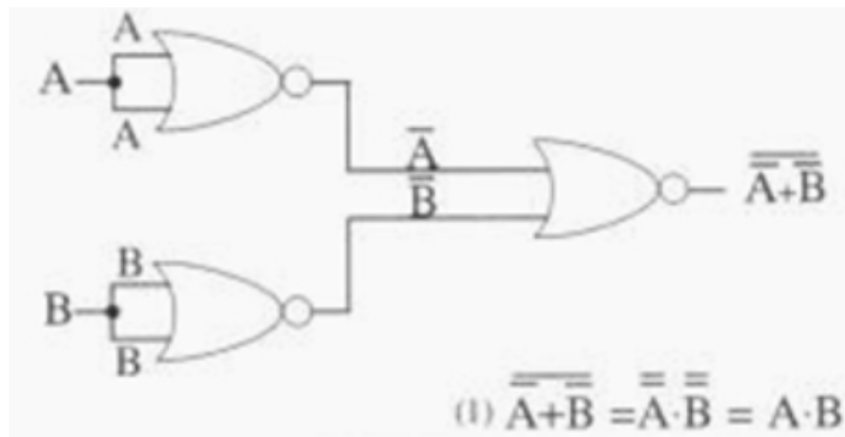
- Ejemplo de ejecución de cualquier puerta empleando solamente puertas NOR.

**NOT:**



**AND:**





**NAND:**

