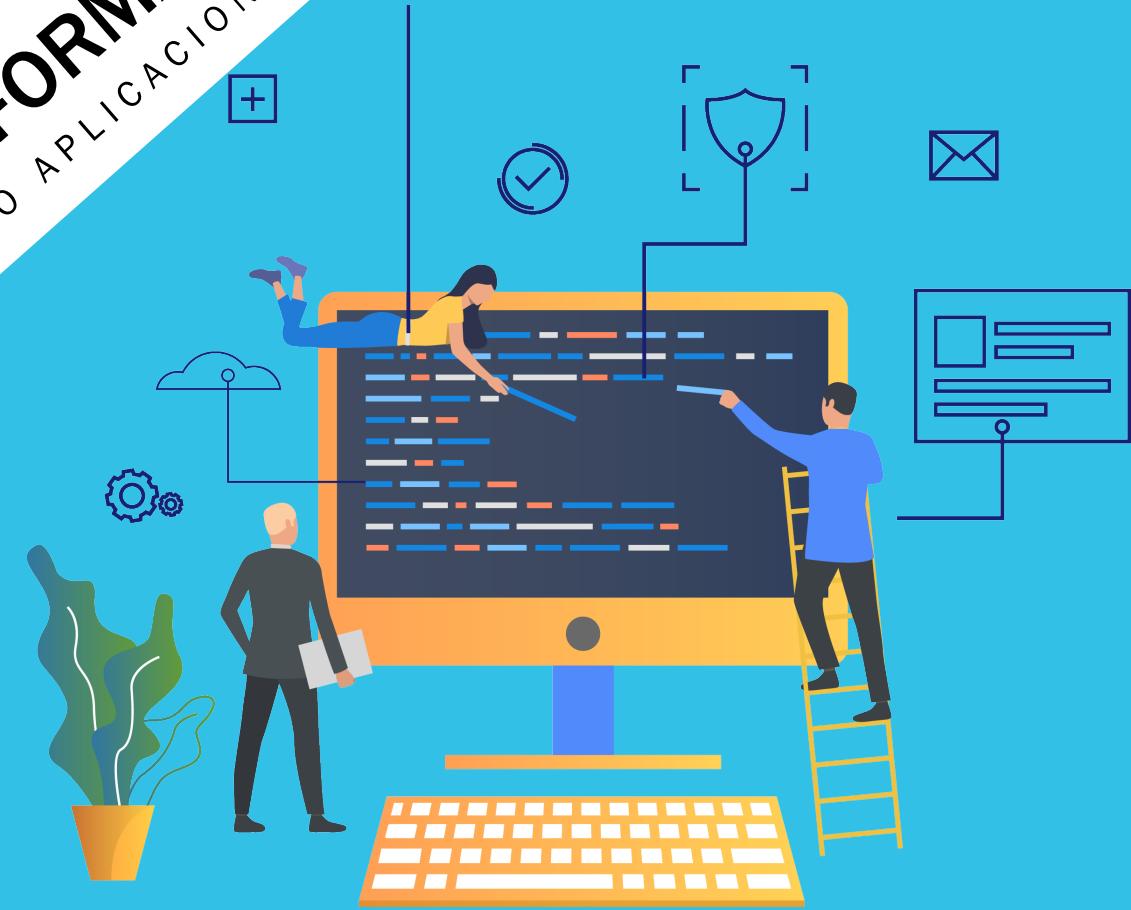


# TEMA 1. INTRODUCCIÓN A LOS SISTEMAS INFORMÁTICOS

CFGS DESARROLLO APLICACIONES MULTIPLATAFORMA



# INTRODUCCIÓN A LOS S.I. CONTENIDOS DEL TEMA

- SISTEMAS INFORMÁTICOS
- COMPONENTE SOFTWARE
- COMPONENTE HARDWARE
- MEDIDAS DE INFORMACIÓN

PARTE I. ¿QUÉ ES UN SISTEMA  
INFORMÁTICO?

What?

# 1. INTRODUCCIÓN

- Es el **ordenador** la herramienta que actualmente nos permite el tratamiento automático de la información, facilitándonos en gran medida su organización, proceso, transmisión y almacenamiento.
- El término **informática** ha ido evolucionando a lo largo del tiempo, pero en la actualidad se considera la ciencia que estudia el tratamiento automático de la información.
- **Informática ....procede de la fusión de dos palabras: información + automática.**

## 2. SISTEMA INFORMÁTICO, SOFTWARE Y HARDWARE

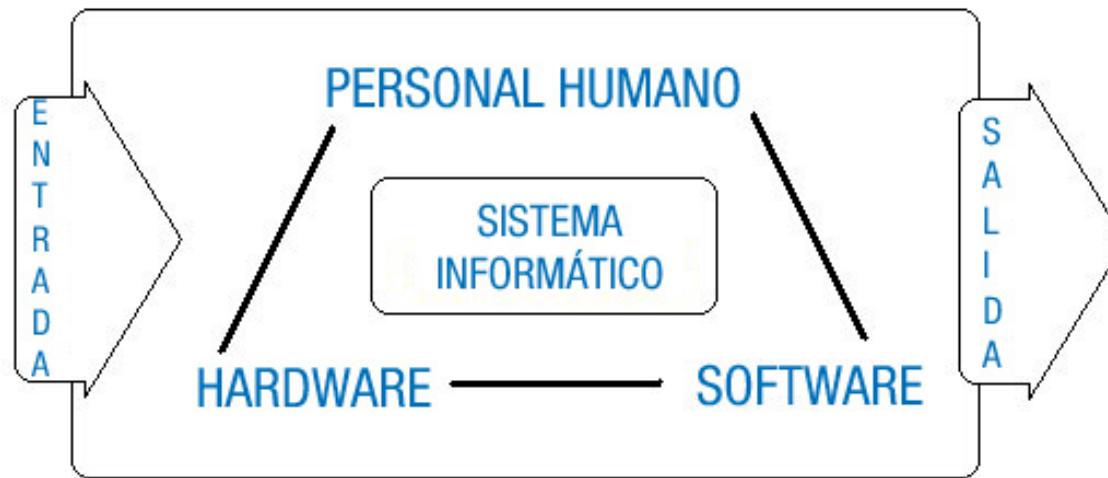
- Físicamente, se puede definir como una máquina compuesta de elementos físicos (**hardware**), en su mayoría de origen eléctrico-electrónico.
- Un **ordenador** está formado por un conjunto de componentes electrónicos que por sí mismos no son capaces de realizar demasiadas funciones
- Estos componentes electrónicos necesitan de otros componentes no físicos que los pongan en un funcionamiento: **programas** (**software**).
- Para que los componentes electrónicos de un ordenador sean capaces de funcionar y realizar un proceso determinado, es necesario ejecutar un conjunto de **órdenes o instrucciones**.
- Estas instrucciones, ordenadas y agrupadas de forma adecuada, constituyen un **programa**

## 2. SISTEMA INFORMÁTICO, SOFTWARE Y HARDWARE

- Los **programas** nos servirán para nuestro fin: procesar datos (**información**).
- Cuando un programa está compuesto por varios programas forma lo que se denomina una **aplicación informática**.
- **Instrucciones, programas y aplicaciones informáticas**, en general, quedan definidos bajo el término **software**.

## 2. SISTEMA INFORMÁTICO, SOFTWARE Y HARDWARE

- Se define **sistema informático** como un conjunto de elementos físicos (**hardware**) y de elementos lógicos (**software**) interconectados entre sí, destinados a gestionar el tratamiento automático y racional de la información, entendiendo por esto, su organización, su transmisión, su procesamiento y/o su almacenamiento.



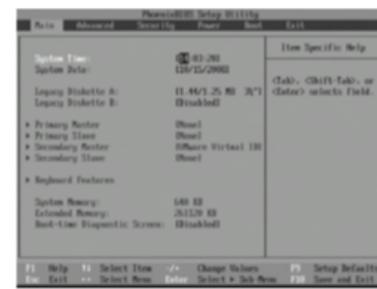
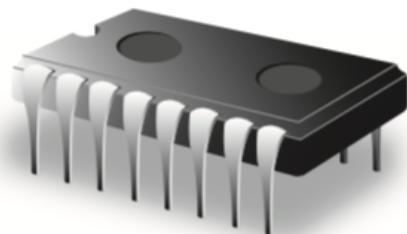
- Se incluye como parte fundamental del sistema informático al conjunto de personas que lo utiliza, ya sean usuarios, administradores, programadores, etc

## 2. SISTEMA INFORMÁTICO, SOFTWARE Y HARDWARE

- Pero un programa no funciona por sí solo.
- Es decir, tenemos los componentes electrónicos y los programas, pero el componente que falta, que también es un componente software, es el sistema operativo.
- El **sistema operativo** es el componente software de un sistema informático capaz de hacer que los programas (software) procesen información (datos) sobre los componentes electrónicos de un ordenador o sistema informático (hardware)

## 2. SISTEMA INFORMÁTICO, SOFTWARE Y HARDWARE

- Entre software y hardware existe otro concepto importante dentro de un sistema informático: **firmware**. Es la parte intangible (software) de componentes del hardware.
- Es el caso del software con el que están programadas las memorias ROM, que son hardware.
- El firmware no es fácilmente modificable. Una vez que se introduce o se graba en un componente hardware, queda prácticamente invariable a lo largo de la vida del ordenador.
- El **firmware es, por tanto, software introducido en componentes electrónicos o hardware**.



# RESUMEN

En un Sistema Informático se debe distinguir entre hardware y software:

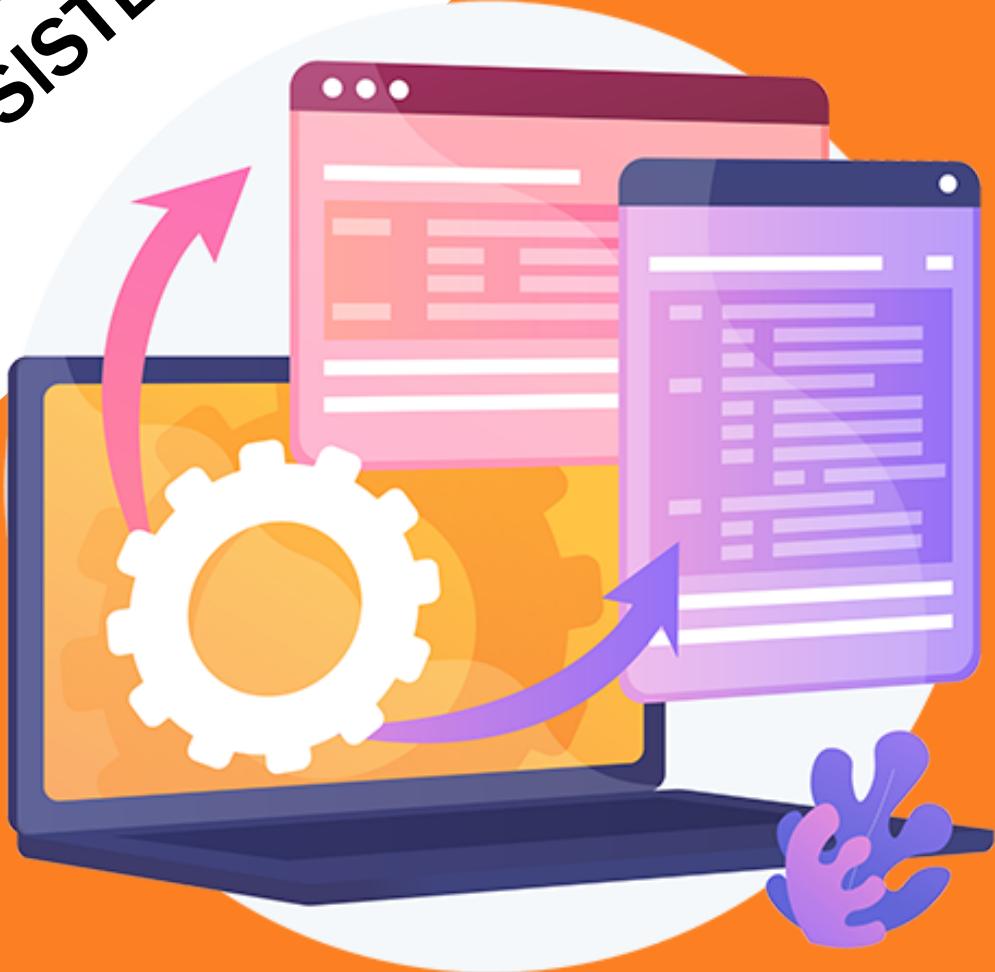
- ✓ **Hardware** es todo lo que forma parte del ordenador, que puede ser tocado físicamente. Es decir; teclado, ratón, monitor, placa base, procesador, memoria, disco duro, cables, etc. Es la "maquinaria" necesaria utilizada para el tratamiento automático de la información.
- ✓ **Software** es el elemento lógico, es todo aquello que es "intangible". Es el conjunto de programas y datos que permiten manejar el hardware, controlando y coordinando su funcionamiento para que realice las tareas deseadas.

# RESUMEN

El **Software** lo integran tanto los programas como los datos:

- Los programas están formados por un conjunto de órdenes o instrucciones que se utilizan para procesar los datos que se le introducen como información. Son necesarios para la gestión y el control de los equipos y de los trabajos de los usuarios.
- Los datos son en sí la información que los programas deben procesar, utilizando para ello los diferentes elementos hardware que componen el sistema informático. Son, en definitiva, **el objeto o razón de ser del sistema informático**.

## PARTE II. COMPONENTE SOFTWARE DE UN SISTEMA INFORMÁTICO.



### **3. COMPONENTES SOFTWARE. SISTEMA OPERATIVO Y APLICACIONES.**

**El software se compone de dos partes fundamentales:**

- **El software básico(S.O)**
- **El software de aplicaciones.**
- **Software básico**
  - Se define como software básico **aquella parte del software sin la cual el ordenador no puede funcionar**. También recibe el nombre de **sistema operativo**:
    - Sirve de comunicación entre el usuario y el hardware de la máquina.
    - Controla los recursos hardware de la máquina según las necesidades, los programas de aplicación, el lugar donde se almacenan los datos, el momento en que hay que imprimir,... etc.

### **3. COMPONENTES SOFTWARE. SISTEMA OPERATIVO Y APLICACIONES.**

#### **Software de aplicaciones**

- El software de aplicaciones **es la parte del software que sirve para procesar la información de forma personalizada**. Lo integran los programas y los datos.

Otra clasificación del software de aplicación se hace según este sea **estándar** o a **medida**:

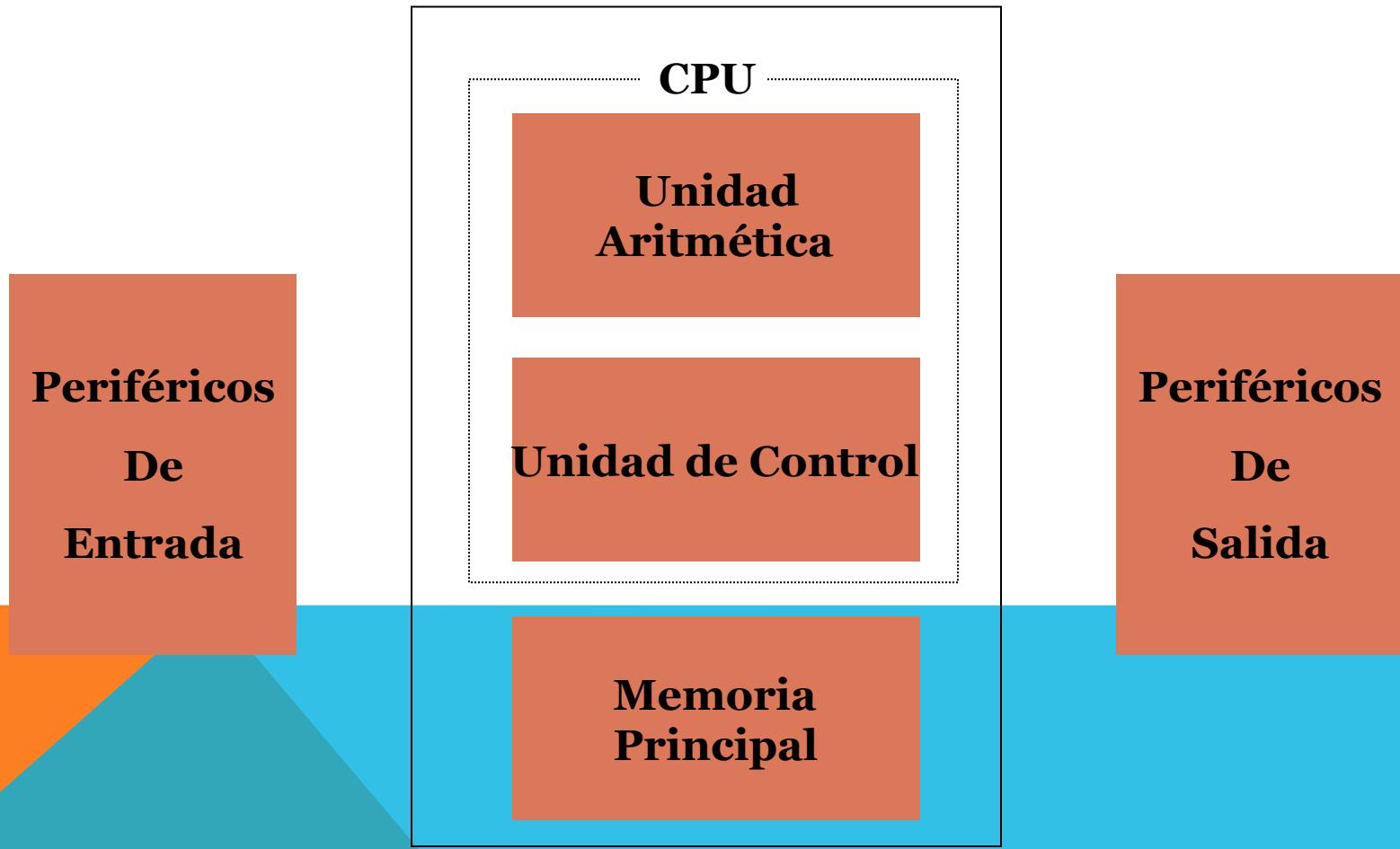
- El estándar es el que encontramos en el mercado y está a disposición del usuario con unas características predeterminadas. Este software lo utiliza el usuario adaptado a las características del propio software.
- El software a medida es el que diseñan analistas e implementan programadores atendiendo a las necesidades concretas de cada usuario.

# PARTE III: COMPONENTES FÍSICOS. HARDWARE



# VON NEUMANN

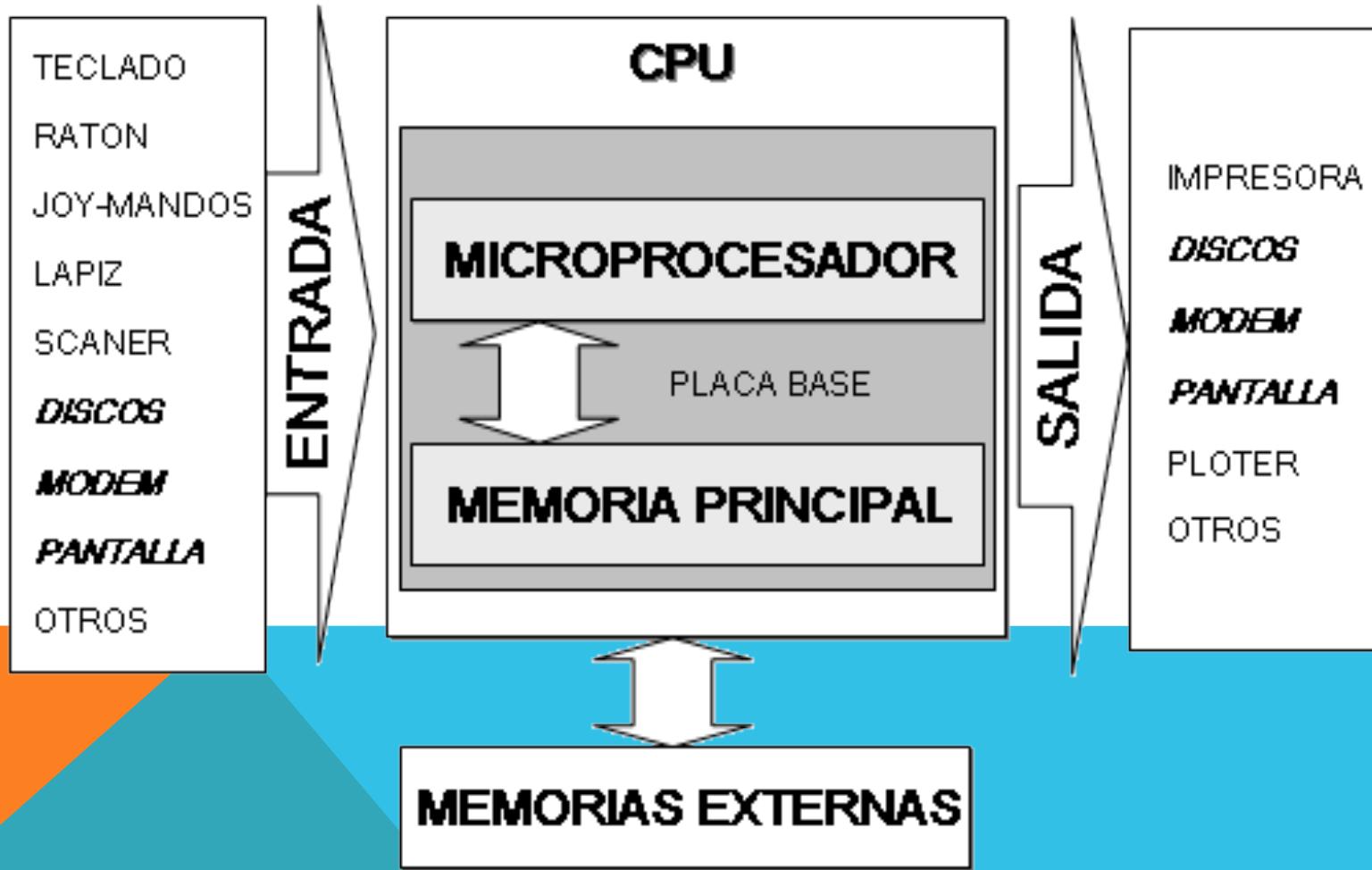
La primera arquitectura apareció en los 40 con Von Neumann.



# ARQUITECTURA BÁSICA DE UN ORDENADOR

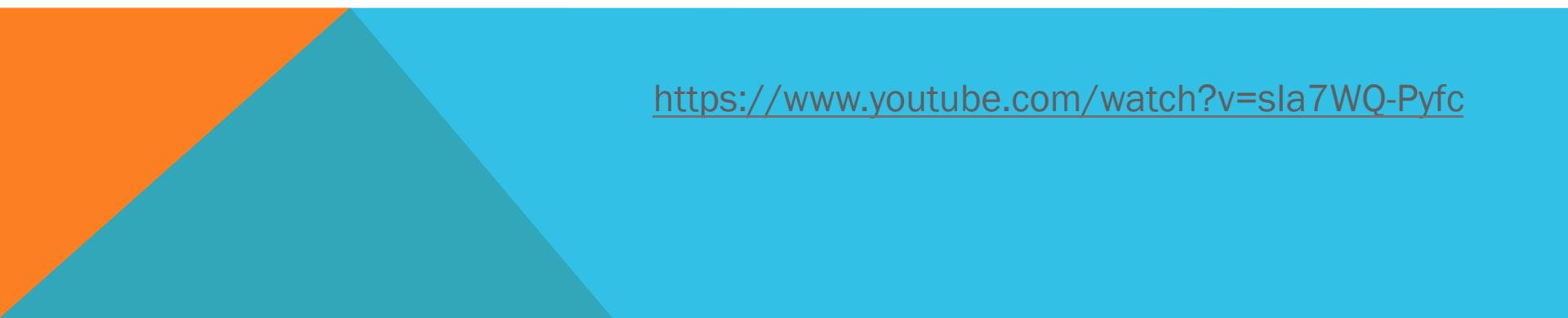
**Se podría decir que el hardware de un ordenador está constituido, básicamente, por la Unidad Central de Proceso (CPU), que se encarga de procesar los datos, la memoria que almacena la información, y los periféricos de entrada y/o salida, que permiten el intercambio de datos o información con el exterior, así como su almacenamiento.**

# ARQUITECTURA BÁSICA DE UN ORDENADOR



# ARQUITECTURA BÁSICA DE UN ORDENADOR

- Los programas deben de seguir un orden a la hora de ser ejecutados:
  - Primero, el usuario da la orden de ejecutar el programa.
  - Se crea una copia de ejecución, en memoria principal. (Todo programa antes de ejecutarse, debe almacenarse en memoria principal).
  - La cpu comienza con la ejecución de dicha instancia llamada (proceso), instrucción por instrucción.
  - Para cada instrucción, la cpu debe ir a memoria y traérsela para ser ejecutada.
  - Se dan las órdenes oportunas para que esa instrucción sea ejecutada.
  - Se realizan las operaciones necesarias.
  - Se almacenan resultados en memoria principal.
  - Se repite el mismo proceso, pero con la instrucción siguiente.



<https://www.youtube.com/watch?v=sla7WQ-Pyfc>

# ARQUITECTURA BÁSICA DE UN ORDENADOR

- La CPU o Microprocesador

- Se encarga de ejecutar todas las instrucciones cargadas de memoria principal.
- Cada instrucción es interpretada por el microprocesador.
- Se encarga de dar órdenes a todos los dispositivos del ordenador.
- Se considera como la parte más fundamental, (el cerebro del computador).
- Su velocidad de computación, viene por millones de operaciones por segundo.
- Su velocidad depende del ciclo de reloj y del tipo de arquitectura.

# ARQUITECTURA BÁSICA DE UN ORDENADOR

- ¿Qué son las instrucciones?
  - Son cada una de las tareas indivisibles con las que se componen los programas.
  - Cada instrucción tiene una tarea determinada.
  - Ser dividen en:
    - ▣ Instrucciones de lectura/escritura a memoria principal.
    - ▣ Instrucciones de lectura/escritura a dispositivos de E/S.
    - ▣ Instrucciones de operaciones como (suma, resta, multiplicación, etc).
    - ▣ Instrucciones de carga de datos entre registros.
    - ▣ ETC.

# ARQUITECTURA BÁSICA DE UN ORDENADOR

- Ejemplo de un programa informático en lenguaje natural.
  - Se pretende diseñar un programa informático que lea dos valores positivos enteros, se sumen y se les muestre al usuario.
  - ¿Qué es lo que necesitamos?
    - Instrucciones que permitan capturar los operandos.
    - Instrucción que permita operar (suma).
    - Instrucción que permita almacenar temporalmente el resultado.
    - Instrucción que permita mostrar el resultado al usuario.

# ARQUITECTURA BÁSICA DE UN ORDENADOR

- Ejemplo de un programa informático en lenguaje natural.
  - Cuando realizamos la suma con una calculadora, ¿Qué pasos realizamos?.
    1. Buscamos las teclas de los dos operandos en la calculadora.
    2. Pulsamos las teclas y a continuación
    3. Buscamos la tecla de suma y la pulsamos.
    4. Buscamos y pulsamos la tecla de igual.
    5. Visualizamos el resultado.

# ARQUITECTURA BÁSICA DE UN ORDENADOR

- Ejemplo de un programa informático en lenguaje natural.
  - Cuando realizamos la suma con un programa o algoritmo en lenguaje natural. ¿Qué pasos realizamos?.
    1. Informamos al usuario que debe teclear un primer operando.
    2. La cpu se espera a que el usuario pulse una tecla numérica.
    3. Informamos al usuario que debe teclear un segundo operando.
    4. La cpu se espera a que el usuario pulse una tecla numérica.
    5. Se buscan los operandos almacenados temporalmente en memoria y se opera con la suma. Su resultado se almacena en memoria principal.
    6. La cpu busca el resultado y lo muestra por la pantalla.

# ARQUITECTURA BÁSICA DE UN ORDENADOR

- Ejemplo de un programa informático en lenguaje natural.

## **Programa Suma**

Variables op1, op2, res : entero

### **Inicio**

Imprimir\_pantalla (“Introduce primer operando\n”)  
leer\_teclado(op1)

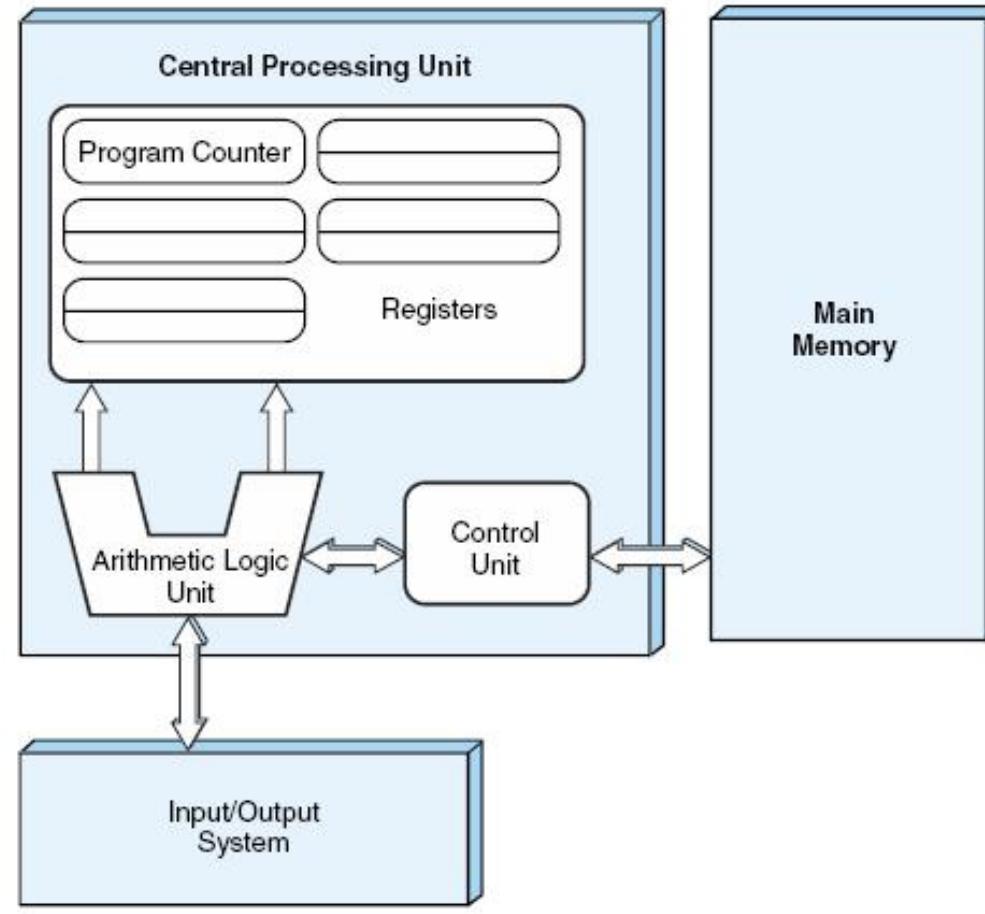
Imprimir\_pantalla (“Introduce segundo operando\n”)  
leer\_teclado(op2)

res = suma(op1, op2)

Imprimir\_pantalla (“La suma de “, op1,” y de “,op2,” es “,res);

### **final**

# ARQUITECTURA BÁSICA DE UN ORDENADOR



# 4. COMPONENTES FÍSICOS. EL HARDWARE

Los componentes físicos del ordenador se pueden clasificar en los siguientes:

1 Unidad central de proceso (UCP). Consta de:

- Unidad aritmético-lógica (UAL).
- Unidad de control (UC).

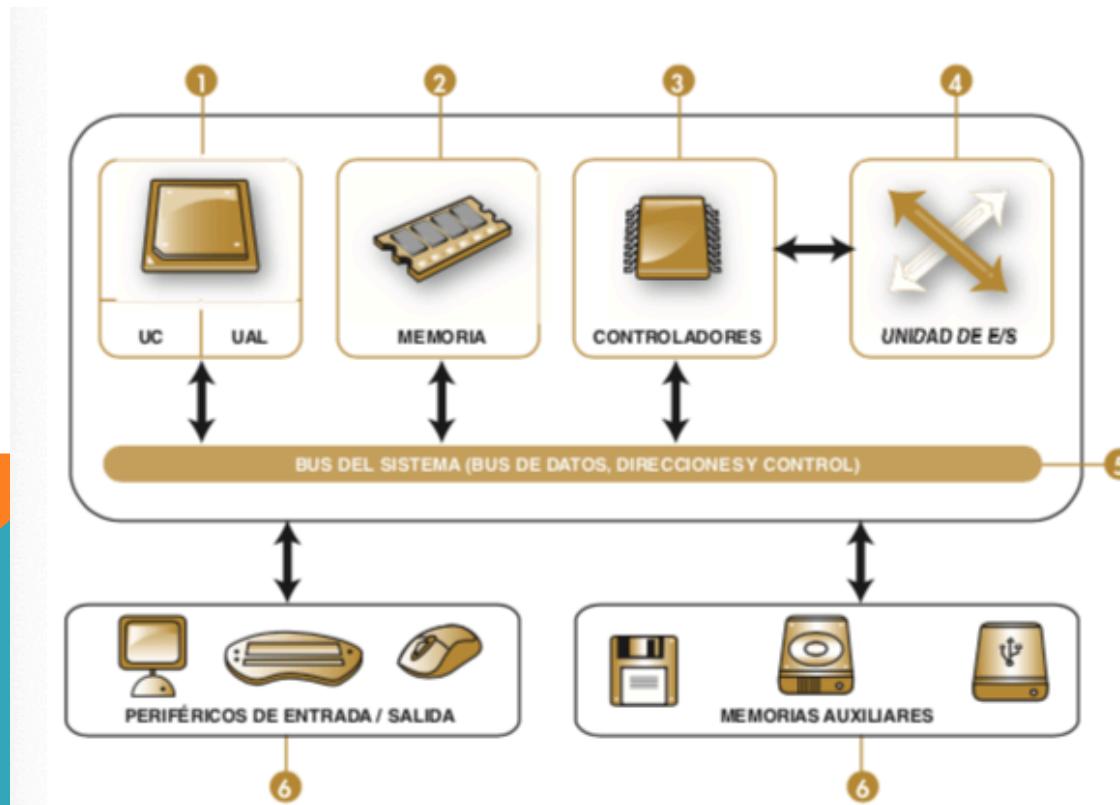
2 Memoria central (MC) o RAM.

3 Controladores.

4 Unidad de entrada/salida (E/S).

5 Buses.

6 Unidades periféricas o periféricos de entrada/salida.



## 4.1. UNIDAD CENTRAL DE PROCESO: CPU

La **unidad central de proceso** o **UCP**, también denominada procesador, es el elemento encargado del control y ejecución de las operaciones que se efectúan dentro del ordenador con el fin de realizar el tratamiento automático de la información.

Es la parte fundamental del ordenador. Se encarga de controlar todas las tareas y procesos que se realizan dentro de él. Está formado por la unidad de control (**UC**), la unidad aritmético- lógica (**UAL**)

Para que el procesador pueda trabajar necesita, además, otros componentes hardware del sistema informático: la memoria principal(**RAM**), la unidad de entrada/salida, los periféricos de entrada/salida, los controladores y los buses



## A. UNIDAD DE CONTROL (UC)

La **unidad de control o UC** es la parte pensante del ordenador; es como el director de una orquesta, ya que se encarga del gobierno y funcionamiento de los aparatos que la componen. Tareas:

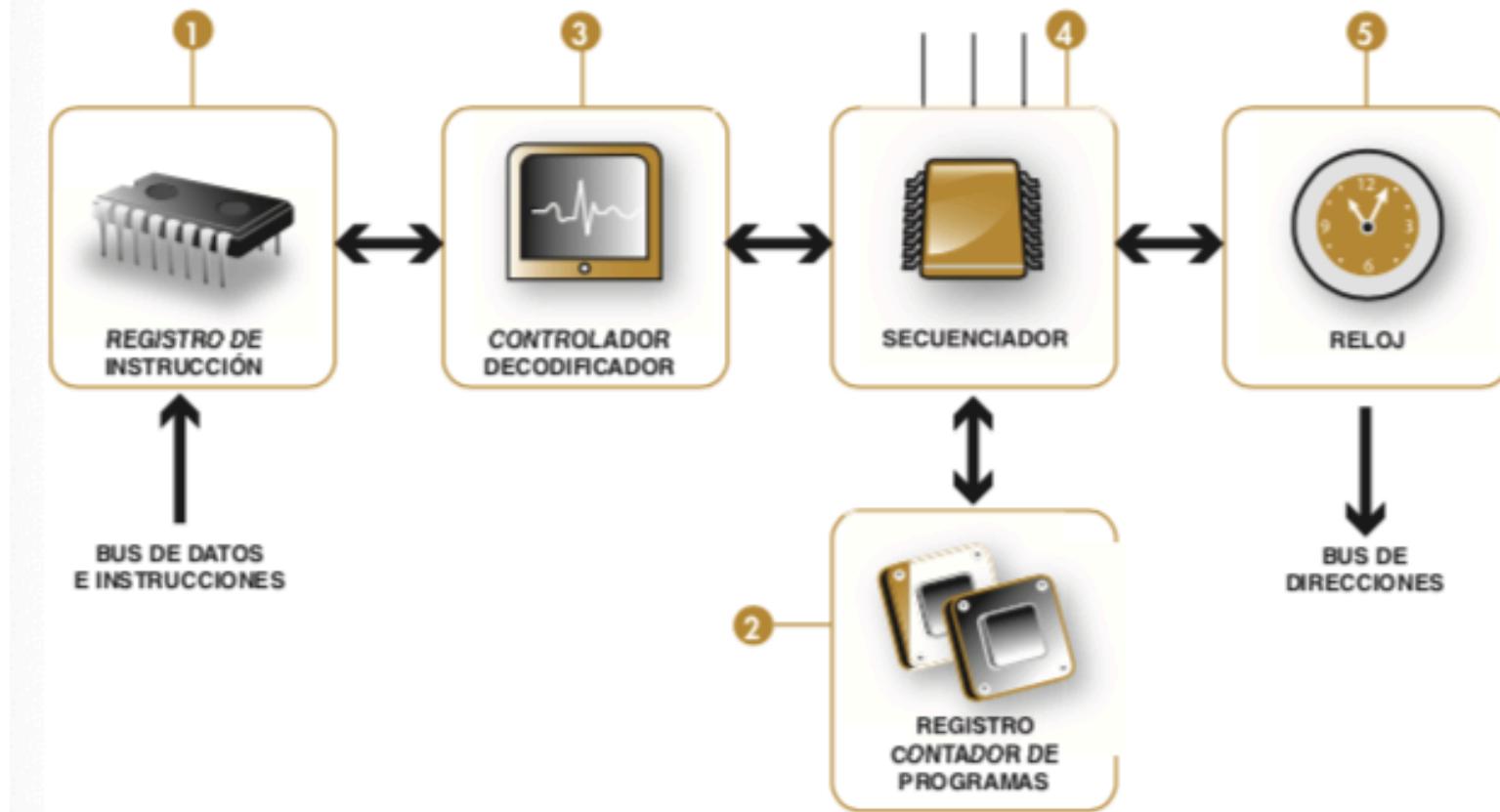
- La tarea fundamental de la UC es recibir información para interpretarla y procesarla después mediante las órdenes que envía a los otros componentes del ordenador
- Se encarga de traer a la memoria interna o central del ordenador (**RAM**) las instrucciones necesarias para la ejecución de los programas y el procesamiento de los datos. Estas instrucciones y datos se extraen, normalmente, de los soportes de almacenamiento externo.
- La **UC interpreta y ejecuta las instrucciones en el orden adecuado** para que cada una de ellas se procese en el debido instante y de forma correcta.

## A. UNIDAD DE CONTROL (UC)

Para realizar todas estas operaciones, la UC dispone de algunos aliados, pequeños espacios de almacenamiento que son su esencia. Estos espacios de almacenamiento se denominan **registros**. Además de los registros, tiene otros componentes. Todos ellos se detallan a continuación:

- 1 **Registro de instrucción.** Es el encargado de almacenar la instrucción que se está ejecutando.
- 2 **Registro contador de programas.** Contiene la dirección de memoria de la siguiente instrucción a ejecutar.
- 3 **Controlador y decodificador.** Se encarga de interpretar la instrucción para su posterior proceso. Es el encargado de extraer el código de operación de la instrucción en curso.
- 4 **Secuenciador.** Genera las microórdenes necesarias para ejecutar la instrucción.
- 5 **Reloj.** Proporciona una sucesión de impulsos eléctricos a intervalos constantes.

## A. UNIDAD DE CONTROL



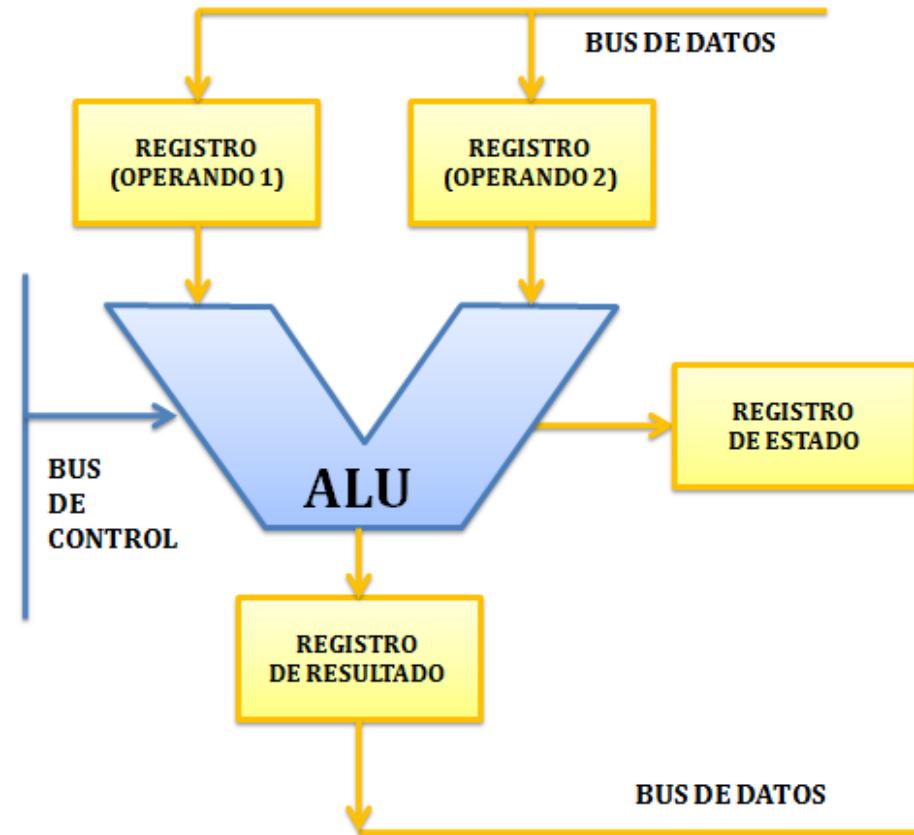
- 1 Registro de instrucción.** Es el encargado de almacenar la instrucción que se está ejecutando.
- 2 Registro contador de programas.** Contiene la dirección de memoria de la siguiente instrucción a ejecutar.
- 3 Controlador y decodificador.** Se encarga de interpretar la instrucción para su posterior proceso. Es el encargado de extraer el código de operación de la instrucción en curso.
- 4 Secuenciador.** Genera las microórdenes necesarias para ejecutar la instrucción.
- 5 Reloj.** Proporciona una sucesión de impulsos eléctricos a intervalos constantes.

# RESUMEN: UNIDAD DE CONTROL

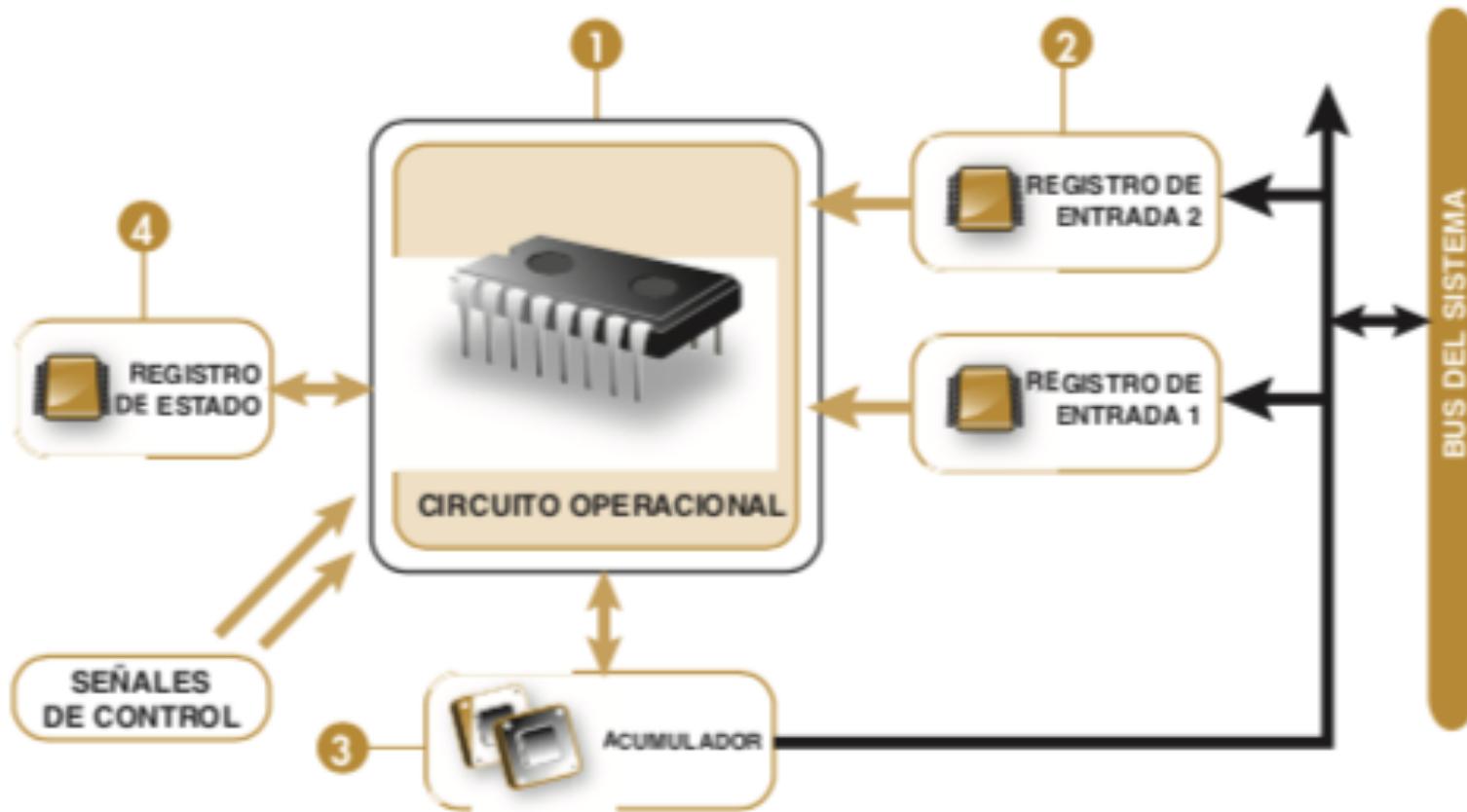
- Unidad de Control.
  - Todas las instrucciones se deben de ejecutar como máximo en 5 pasos:
    1. Búsqueda de la instrucción en memoria principal.
    2. Decodificación de la instrucción.
    3. Búsqueda de los operandos.
    4. Ejecutar operación de la instrucción.
    5. Almacenar resultados.

## B. UNIDAD ARITMÉTICO-LÓGICA (UAL)

- La unidad aritmético-lógica o UAL es la parte de la UCP encargada de realizar operaciones aritméticas y lógicas sobre la información:
- Las operaciones aritméticas pueden ser suma, resta, multiplicación, división, potenciación, etc.
- Las lógicas son normalmente de comparación, para las que se emplean los operadores del álgebra de Boole



## B. UNIDAD ARITMÉTICO-LÓGICA (UAL)



- 1 Operacional o circuito operacional.** Realiza las operaciones con los datos de los registros de entrada.
- 2 Registros de entrada.** Contienen los operandos de la operación.
- 3 Acumulador.** Almacena los resultados de las operaciones.
- 4 Registro de estado.** Registra las condiciones de la operación anterior.

# BUS DEL SISTEMA

- Buses internos de la CPU.
  - Son las auténticas autopistas por donde circulan los bits.
  - Distinguimos tres tipos de bus:
    - Bus de datos. Sólo circulan los bits de datos. Suelen ser de 16, 32 bits, 64, etc.
    - Bus de direcciones. Sólo circulan los bits de las direcciones en memoria a las que se quiere acceder.
    - Bus de control. Sólo circulan los bits de aquellas órdenes que la unidad de control ordena al resto de dispositivos y estos contesten, por ejemplo, una orden de lectura o escritura a memoria principal. El tamaño del bus de datos, depende del número de accesos que la cpu haga a memoria a por un dato. Por ejemplo, un bus de 16 bits, para acceder a un dato de 32 bits, se realizan 2 accesos.

## 4.2 LA MEMORIA. FUNCIONES, TIPOS Y CARÁCTERÍSTICAS



## 4.2 LA MEMORIA. FUNCIONES, TIPOS Y CARÁCTERÍSTICAS

El ordenador almacena dentro de su memoria interna todos los programas y datos con los que se va a trabajar y que van a ser procesados. Los dos tipos de memoria esenciales con los que puede trabajar el ordenador son:

- ✓ **Memorias de almacenamiento externo: soportes de almacenamiento**
- ✓ **Memoria interna: RAM y ROM**

## 4.2 LA MEMORIA. FUNCIONES, TIPOS Y CARÁCTERÍSTICAS

### NIVELES DE LA JERARQUÍA DE MEMORIA



# MEMORIAS DE ALMACENAMIENTO EXTERNO

- Son memorias externas: discos duros, pen drives, etc., y aunque estén físicamente dentro de la carcasa del ordenador, como es el caso de los discos duros, la denominación de externas es para diferenciarlas precisamente de la propia RAM.
- Estas memorias **son más lentas que la propia memoria principal**, ya que constan de componentes electrónicos y mecánicos.
- Son **no volátiles**, de tal forma que la información permanece en ellas incluso después de quitar el suministro de energía eléctrica al ordenador.

# MEMORIAS DE ALMACENAMIENTO INTERNO

Dentro del ordenador existen varios tipos de memorias que no son consideradas externas. Son las siguientes:

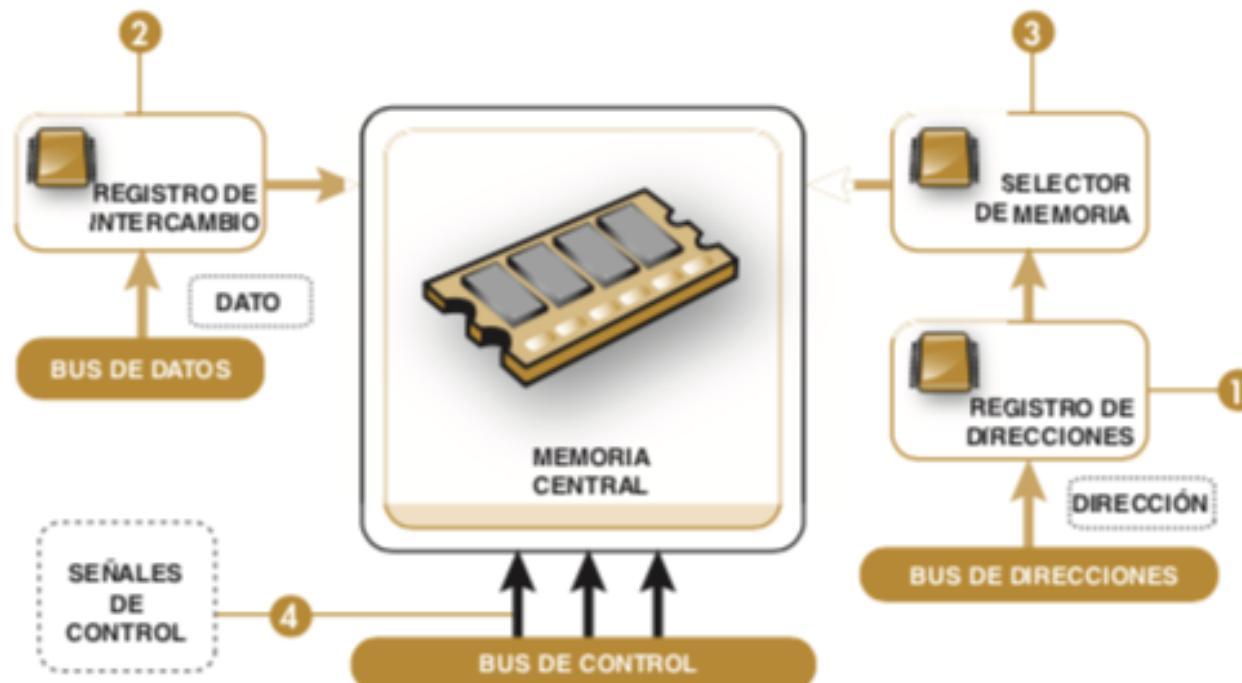
- **RAM (Random Access Memory).** En ella es posible almacenar y modificar información y es lo que se conoce como memoria principal, memoria central o memoria de acceso directo.
- **ROM (Read Only Memory).** Es una memoria de solo lectura, cuya información no puede ser modificada y que sirve básicamente para poder inicializar el sistema informático.

# MEMORIAS DE ALMACENAMIENTO INTERNO: RAM

- La memoria RAM es un componente necesario para que se pueda procesar la información.
- Casi todo, por no decir todo, lo que se tiene que procesar dentro del ordenador debe pasar tarde o temprano por la memoria central.
- Los elementos que componen la memoria central o principal:

- 1 **Registro de direcciones.** Contiene la dirección de la celda o posición de memoria a la que se va a acceder.
- 2 **Registro de intercambio.** Recibe los datos en operaciones de lectura y almacena los datos en las operaciones de escritura.
- 3 **Selector de memoria.** Se activa cada vez que hay que leer o escribir conectando la celda o posición de memoria con el registro de intercambio.
- 4 **Señales de control.** Indica si una operación es de lectura o escritura.

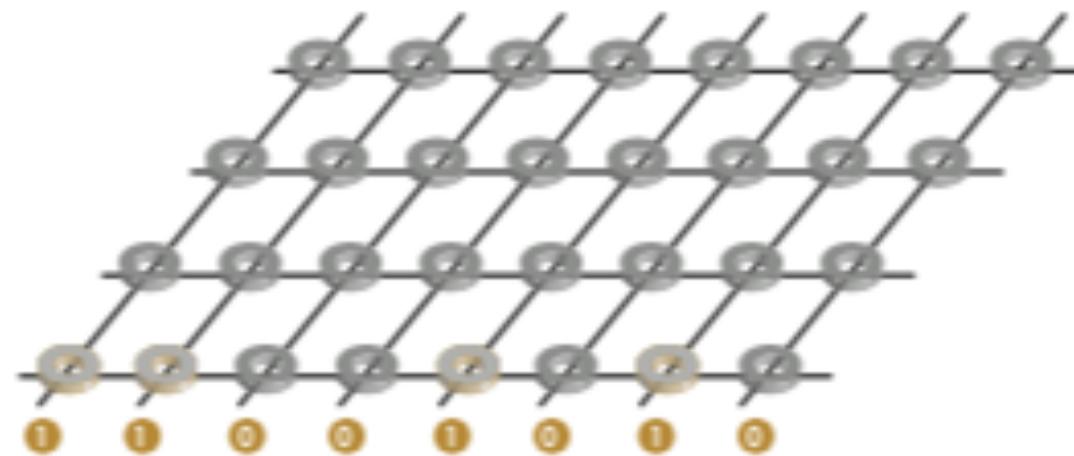
# RAM



- 1 Registro de direcciones.** Contiene la dirección de la celda o posición de memoria a la que se va a acceder.
- 2 Registro de intercambio.** Recibe los datos en operaciones de lectura y almacena los datos en las operaciones de escritura.
- 3 Selector de memoria.** Se activa cada vez que hay que leer o escribir conectando la celda o posición de memoria con el registro de intercambio.
- 4 Señales de control.** Indica si una operación es de lectura o escritura.

# RAM

- La memoria central está formada por componentes electrónicos (**biestables**) capaces de almacenar información en forma de ceros y unos (sistema binario), que actúan como pequeños condensadores, de tal forma que la presencia de energía dentro de ellas puede traducirse como un uno (1) lógico y la ausencia de energía como un cero (0) lógico.
- Cada información de este tipo recibe el nombre de **bit**.



# MEMORIA CACHÉ

Además de la memoria central, lo normal es que los ordenadores incorporen otro tipo de memoria para agilizar los cálculos que realizan los programas.

- Suelen ser memorias intermedias colocadas entre la RAM y el procesador, que almacenan temporalmente la información a procesar.
- Este tipo de memorias reciben el nombre de **memoria caché** y almacenan la información que se utiliza con más frecuencia.



# ¿CÓMO FUNCIONAN LAS MEMORIAS?

- La información en memoria se suele almacenar en bloques. Estos bloques suelen ser de ocho celdillas; es decir, equivalen a 8 bits y se denominan byte (combinación de ceros y unos).
- Cada conjunto de ellos representa un carácter, es decir, cualquier letra o número como combinación de 8 bits.
- Estos condensadores, como tales, cierto tiempo, se van descargando. transcurrido
- Evidentemente, para no perder la información de la memoria, el propio sistema informático tendrá que proceder a recargarlos antes de que se descarguen definitivamente. Este proceso es conocido como **refresco de memoria**.

ONE BYTE  
11010111  
=  
8 BITS

# MEMORIA RAM: TIPOS

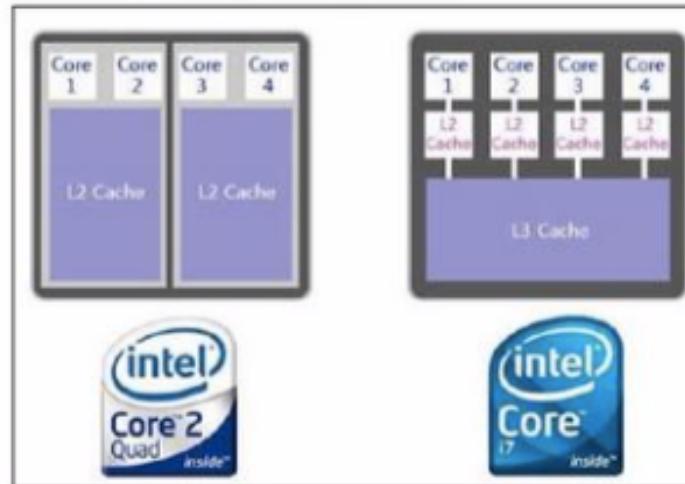
- **DRAM (Dynamic RAM)**. Es un tipo de memoria RAM electrónica **construida mediante condensadores**. Cuando un condensador está cargado se dice que almacena un BIT a uno. Si está descargado, el valor del BIT es cero. Para mantener las celdillas cargadas, este tipo de memoria necesita refrescarse cada cierto tiempo: el refresco de una memoria RAM consiste en recargar nuevamente con energía los condensadores que tienen almacenado un uno para evitar que la información se pierda (de ahí lo de Dynamic).
- La memoria DRAM es más lenta que la memoria SRAM, pero mucho más barata de fabricar.

# MEMORIA RAM: TIPOS

- **SRAM (Static RAM)**. Es un tipo de memoria RAM alternativa a la DRAM que **no necesita refrescarse**.
- **SRAM y DRAM son memorias volátiles**, lo que significa que cuando se corta el suministro de corriente, los datos almacenados se pierden.
- **Debido al alto coste de fabricación de la SRAM y a su alta velocidad, suele utilizarse como memoria caché.**

**Caché de 1º nivel (L1):** Esta caché está integrada en el núcleo del procesador, trabajando a la misma velocidad. La cantidad de memoria caché L1 varía de un procesador a otro, estando normalmente entre los 64KB y los 256KB. Se divide en dos partes, una para instrucciones y otra para datos.

**Caché de 2º nivel (L2):** Integrada también en el procesador, aunque no directamente en el núcleo de este, tiene las mismas ventajas que la caché L1, aunque es algo más lenta que esta. La caché L2 suele ser mayor que la caché L1, pudiendo llegar a superar los 2MB.

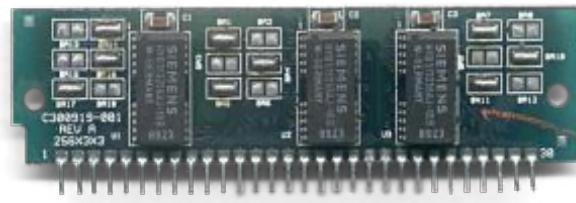


## MEMORIA RAM: TIPOS

- **SDRAM (Synchronous Dynamic RAM)**. Es una memoria que incorpora la capacidad de la DRAM y la velocidad de la SRAM; es decir, necesita refresco de sus celdas, pero en un intervalo superior de tiempo. **Esta memoria es la que incorporan en la actualidad la mayoría de los ordenadores personales.**
- **DDRAM (Double Data Rate)** o memoria de doble recarga o memoria de doble tasa de transferencia. Compuesta por memorias SDRAM, tiene la característica de que se refresca dos veces por impulso de reloj. Es una memoria de funcionamiento muy complejo, pero tiene la ventaja de ser prácticamente el doble de rápida que cualquiera de las anteriores.

# MEMORIA RAM: CARACTERÍSTICAS

- En la actualidad, una de las características fundamentales de las memorias RAM es la velocidad con que la información se puede almacenar en ellas.
- Esta **velocidad es mayor cuanto menos se tarde en acceder a la posición de memoria requerida en cada instante**. La velocidad se mide en nanosegundos (60, 70, 80, 100,...). Cuanto menor sea el tiempo de acceso, más rápido será el acceso que se pueda realizar a cualquier posición de memoria para poder grabar o leer su información.



# MEMORIAS DE ALMACENAMIENTO INTERNO: ROM

- La memoria ROM o **memoria de solo lectura** contiene programas especiales que sirven para cargar e iniciar el arranque del ordenador.
- En ella se encuentra almacenada toda la información referente a los componentes hardware de los que consta nuestro equipo.
- Posteriormente, será labor del sistema operativo realizar el resto de operaciones para poder empezar a utilizar el ordenador.
- El software que integra la ROM forma el **BIOS** del ordenador (Basic Input Output System) o sistema básico de entrada/salida.



# MEMORIA ROM: BIOS

- Inicialmente, los BIOS se programaban sobre memorias de tipo ROM, lo que implicaba que cualquier modificación en el sistema no podía realizarse a menos que lo hiciese el fabricante. Había que sustituir el componente electrónico para modificar la configuración del BIOS.
- Por eso, posteriormente, el BIOS se montó en memorias de tipo PROM (Programmable Read Only Memory), que son programables una sola vez y después de haber sido montadas en la placa.
- El BIOS es un código que localiza y carga el sistema operativo en la RAM; es un software elemental instalado en una pequeña ROM de la placa base que permite que esta comience a funcionar. Proporciona las órdenes básicas para poner en funcionamiento el hardware indispensable para empezar a trabajar.

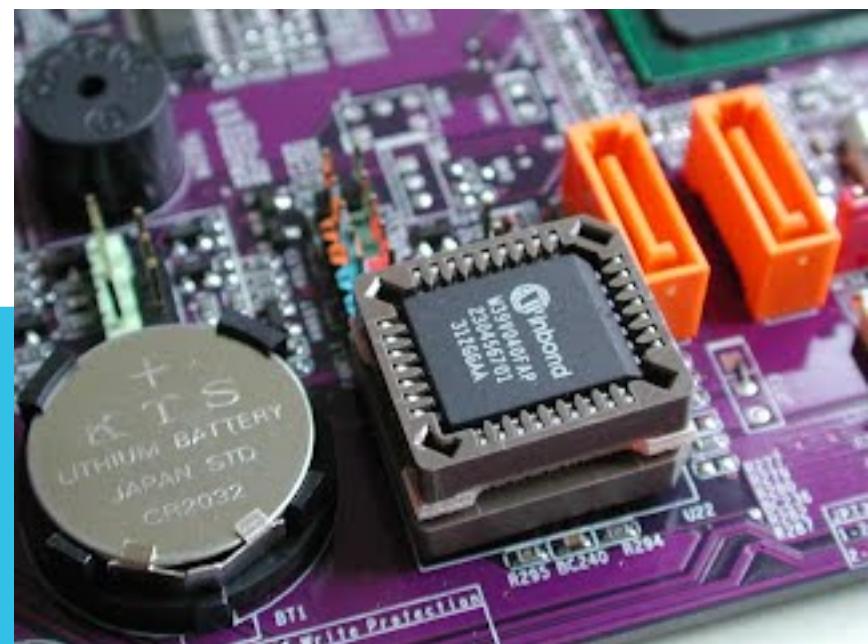
# MEMORIA ROM: BIOS

En la actualidad, se utilizan las memorias de tipo **EPROM** (Erasable Programmable Read Only Memory), que permiten cambiar la configuración asignada. Este proceso es complejo, pero no implica realizar operaciones físicas sobre los componentes que están montados.

La **CMOS** (Complementary Metal Oxide Semiconductor) es un tipo de memoria interna del ordenador que se caracteriza por consumir muy poca energía eléctrica, lo que la hace idónea para almacenar datos del BIOS.

# MEMORIA ROM: BIOS

- El BIOS es un tipo de memoria que en los ordenadores se utiliza para guardar los datos básicos de hardware y de configuración.
- Para que toda la información que mantiene no se borre, es necesario que la CMOS siempre tenga corriente eléctrica. Cuando el ordenador está apagado, obtiene energía de una pequeña **pila** o batería ubicada en la placa base.
- La configuración del BIOS se puede modificar si instalamos un nuevo disco duro, si queremos cambiar la fecha, la hora del sistema, etc. Para acceder al BIOS y poder modificar sus valores, hay que pulsar las teclas F2 o Supr durante el proceso de inicio del equipo, dependiendo del BIOS de cada equipo.



# OTROS TIPOS DE MEMORIA INTERNA

- VRAM o memoria de vídeo se utiliza para almacenar las imágenes que se quieren visualizar, en vez de hacerlo directamente sobre la RAM. Actualmente, este tipo de memoria es fundamental debido a la evolución de la tecnología multimedia.
- En la actualidad, la mayoría de los ordenadores incorporan en la propia tarjeta o adaptador gráfico la llamada SGDRAM (Super Graphics Dynamic Random Access Memory). Se trata de una memoria de elevada capacidad, a veces verdaderamente elevada, que se caracteriza por su alta velocidad y bajo consumo.

# OTROS TIPOS DE MEMORIA INTERNA

La CDRAM, por ejemplo, es un tipo de memoria que **actúa entre el procesador y el periférico correspondiente**. En algunos casos, estas memorias actúan como memorias caché (memoria intermedia de alta velocidad). Suelen ir asociadas a determinados dispositivos, como unidades de CD-ROM y dispositivos de entrada/salida, para liberar a la RAM de operaciones innecesarias.

# ESTRUCTURA DE LA MEMORIA RAM

- Los fabricantes de ordenadores han dividido desde un principio la estructura en esas tres partes fundamentales, que son gestionadas de forma diferente por cada sistema operativo.
- Así, por ejemplo, MS-DOS gestiona la memoria dividiéndola realmente en estas tres capas.
- Windows9X, XP, Vista 2003 Server, 2008 Server, UNIX o Linux gestionan la memoria de forma diferente, olvidándose casi por completo de esta estructura.
- Ya veremos qué gestión se hace de la memoria en cada uno de los sistemas operativos.
- En cualquier caso, esta cuestión es fundamental, pues una cosa es el modo en que los ordenadores reconocen la memoria después de ser fabricados (tres capas) y otra muy distinta el modo en que el sistema operativo gestiona las diferentes capas de la RAM.

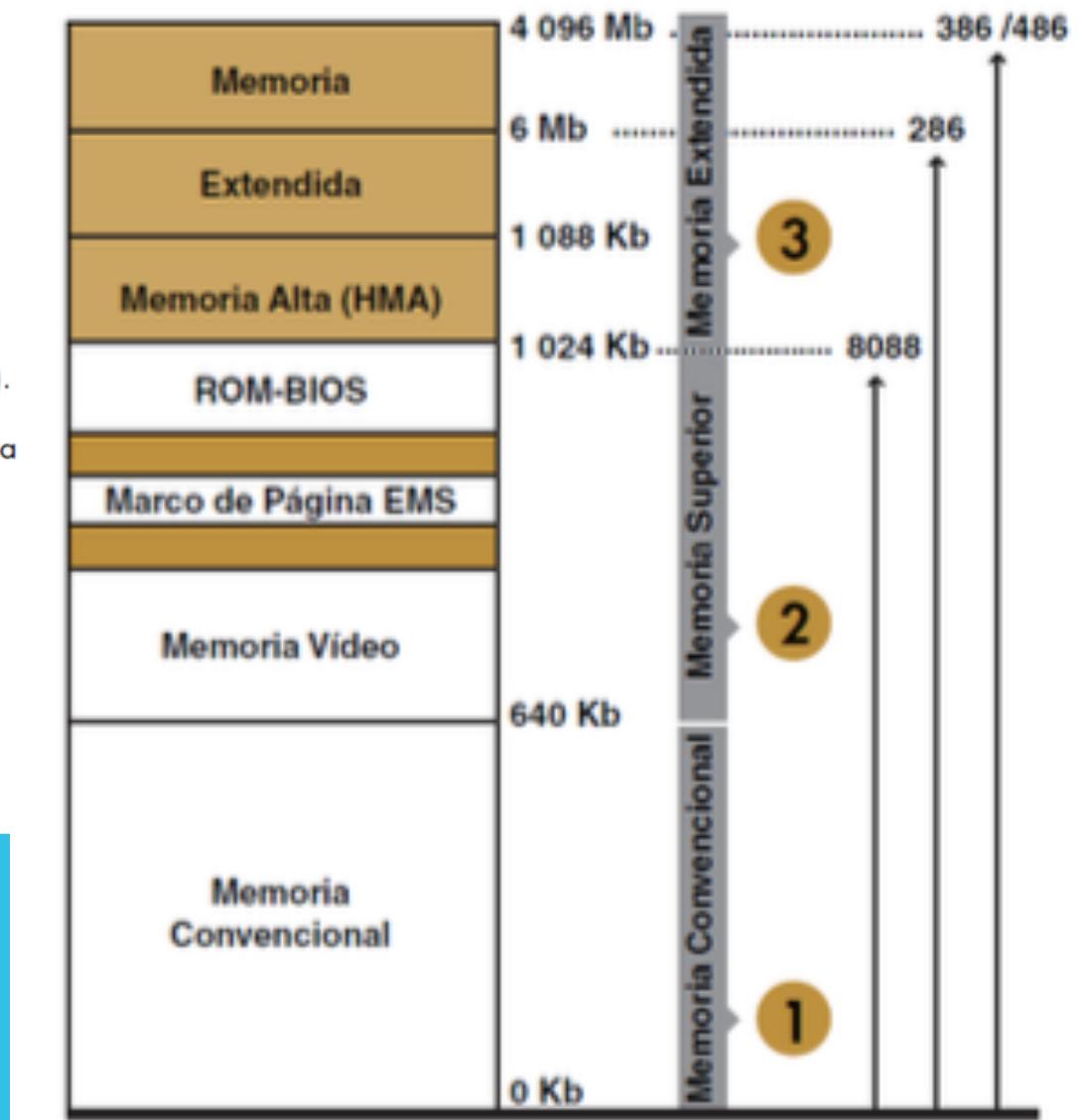
# ESTRUCTURA DE LA MEMORIA RAM

**En cuanto a la estructura de la memoria, desde un principio, e independientemente del sistema operativo, la memoria se ha estructurado en varios niveles**

- 1 Memoria convencional.** De 0 a 640 Kb.
- 2 Memoria superior.** De 641 Kb hasta 1024 Kb (1 Mb).
- 3 Memoria extendida.** De 1 025 Kb hasta el límite de la placa base del equipo.

# ESTRUCTURA DE LA MEMORIA RAM

- 1 Memoria convencional. De 0 a 640 Kb.
- 2 Memoria superior. De 641 Kb hasta 1024 Kb (1 Mb).
- 3 Memoria extendida. De 1 025 Kb hasta el límite de la placa base del equipo.



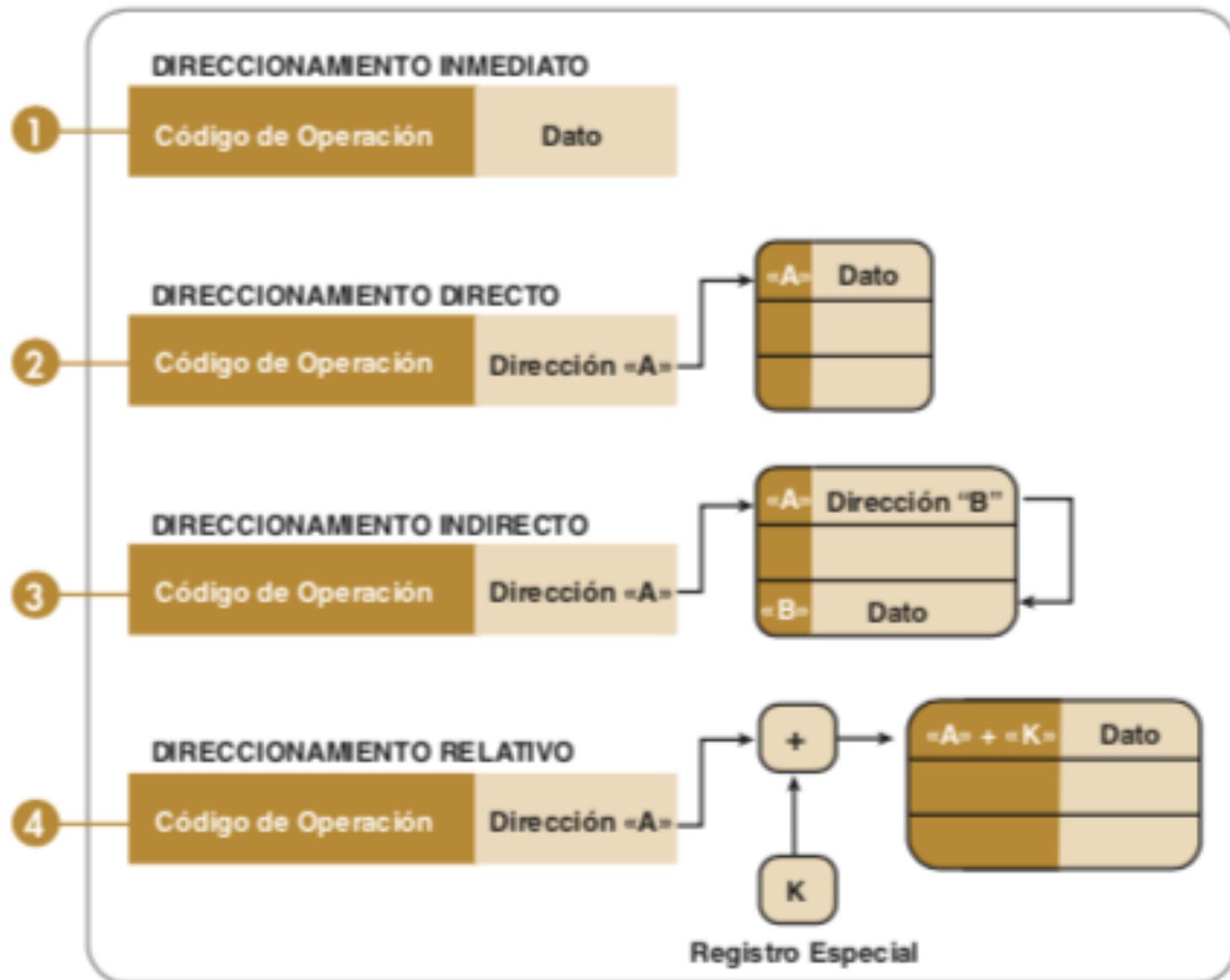
# ESTRUCTURA DE LA MEMORIA RAM

- Para acceder a las celdillas de memoria hay que atender al concepto de **dirección de memoria**.
- Esta dirección es la situación del componente electrónico dentro del conjunto de componentes de la memoria. De esta forma, cuando se accede a una dirección de memoria, lo que se hace es acceder a un conjunto de biestables (condensadores).
- Cada uno de estos biestables físicos referencia un bit lógico (0,1). El bit se define como la mínima unidad de información.
- El conjunto de 8 bits a los que se accede se denomina byte, carácter o palabra. A partir de aquí, la información se mide como conjunto de bytes, es decir, bloques de 8 bits.

# MEMORIA RAM: DIRECCIONAMIENTO

- El **direccionamiento** es una operación que se realiza cuando el procesador ejecuta o interpreta una instrucción.
- Toda **instrucción** está compuesta por un **código de operación** y un **operando**.
- El **código de operación** es la **instrucción en sí** (suma, resta, multiplicación, etc.) y el **operando** es el **dato o información** que se va a procesar.
- Según el método utilizado, la rapidez de ejecución de un programa será mayor o menor. Los llamados **modos de direccionamiento** son las diferentes formas de acceder a los operandos en cada instrucción.

# MEMORIA RAM: DIRECCIONAMIENTO



# MEMORIA RAM: DIRECCIONAMIENTO

- 1 **Direccionamiento inmediato.** En la instrucción está incluido directamente el operando o dato.
- 2 **Direccionamiento directo.** En la instrucción, el campo del operando contiene la dirección en memoria donde se encuentra el operando.
- 3 **Direccionamiento indirecto.** El campo del operando contiene una dirección de memoria en la que se encuentra la dirección efectiva del operando.
- 4 **Direccionamiento relativo.** La dirección del dato que interviene en la instrucción se obtiene sumando a la dirección de la propia instrucción una cantidad fija, que normalmente está contenida en un registro de tipo especial.

## 4.3. UNIDADES DE E/S Y BUSES



## 4.3. UNIDADES DE E/S Y BUSES

La **unidad de entrada/salida** sirve para comunicar el procesador y el resto de componentes internos del ordenador con los periféricos de entrada/salida y las memorias de almacenamiento



Aquí vemos los puertos que comunican las unidades de entrada y salida con el exterior.

## 4.3. UNIDADES DE E/S Y BUSES

- El **bus** es el elemento responsable de establecer una correcta interacción entre los diferentes componentes del ordenador. Es, por lo tanto, el dispositivo principal de comunicación.
- En un **sentido físico**, se define como un conjunto de líneas de hardware (metálicas o físicas) utilizadas para la transmisión de datos entre los componentes de un sistema informático. En cambio, en **sentido figurado** es una ruta compartida que conecta diferentes partes del sistema.



## **4.3. UNIDADES DE E/S Y BUSES**

- La evolución de los buses a lo largo de la historia ha sido determinante para la evolución de los sistemas operativos.
- Al fabricarse buses mucho más rápidos y con más líneas, los sistemas operativos han ido mejorando y aportando nuevas funcionalidades que antiguamente no podían ser implementadas por falta de velocidad en los buses.
- Hoy por hoy, el bus sigue determinando en gran medida la velocidad de proceso de un equipo, ya que sigue siendo uno de los componentes hardware que mayores .

# BUSES: TIPOS DE INTERCONEXIÓN

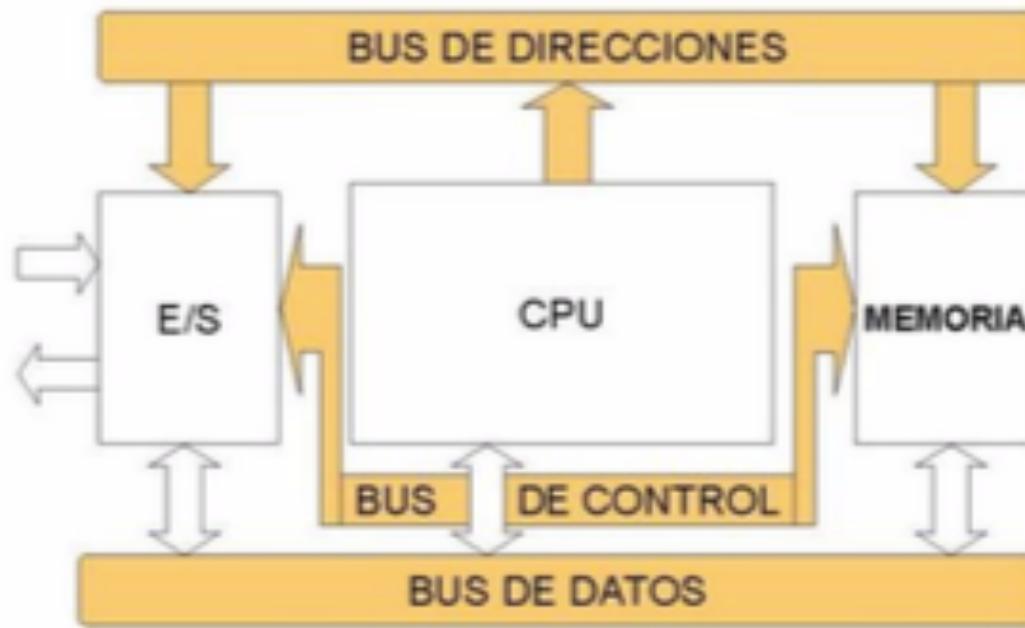
En lo referente a la estructura de interconexión mediante los buses, existen de dos tipos:

- **Bus único.** Considera a la memoria y a los periféricos como posiciones de memoria, y hace un símil de las operaciones E/S con las de escritura/lectura en memoria. Estas equivalencias consideradas por este bus hacen que no permita controladores DMA (Direct Access Memory) de acceso directo a memoria.
- **Bus dedicado.** Este, en cambio, al considerar la memoria y periféricos como dos componentes diferentes, permite controladores DMA.

# BUS DEDICADO

El bus dedicado contiene varias subcategorías más que son las siguientes:

- **Bus de datos.** Transmite información entre la CPU y los periféricos.
- **Bus de direcciones.** Identifica el dispositivo al que va destinada la información que se transmite por el bus de datos.
- **Bus de control o de sistema.** Organiza y redirige hacia el bus pertinente la información que se tiene que transmitir.



# BUSES

**La capacidad operativa del bus depende:**

- **Del propio sistema, de la velocidad de este,**
- **la «anchura» del bus (número de líneas de datos que operan en paralelo) depende de los bits que se pueden transmitir simultáneamente, según el tipo de procesador que incorpore el equipo.**

Procesador	BUS datos/BUS direcciones
8086	16 / 8
8088	16 / 8
80286	16 / 16
80386	32 / 16
80486	32 / 66
PENTIUM	50 / 66
AMD K-6 2	32 / 100
PENTIUM II	32 / 100
CELERON	32 / 66
PENTIUM III	32 / 100
AMD ATHLON	32 / 100
PENTIUM IV	32 / 64
PENTIUM CORE DUO	64 / 133

# BUSES

- La frecuencia o velocidad del bus queda determinada por los **impulsos de reloj**.
- El reloj es, por tanto, el componente que determina la velocidad, ya que a mayor frecuencia en Mhz, más rápida es la circulación de bits por las líneas del bus.
- Y no solo eso. El bus determina la arquitectura y, por tanto, su tamaño determina el del registro de instrucción. Así, el código de operación puede ser mayor, siendo posible ejecutar un mayor número de operaciones. Esto produce un aumento de potencia, no por mayor rapidez, sino por mayor complejidad de las instrucciones.

## 4.4. LOS PERIFÉRICOS

- Los periféricos son dispositivos hardware con los cuales el usuario puede interactuar con el ordenador (teclado, ratón, monitor), almacenar o leer datos y/o programas (dispositivos de almacenamiento o memorias auxiliares), imprimir resultados (impresoras),... etc
- Los periféricos se conectan con el ordenador, es decir, con la UCP y sus componentes, a través de los denominados **puertos** o conectores externos. Esta gestión la lleva a cabo otra parte esencial del ordenador: **la unidad de entrada/salida**, componente hardware usado para la gestión de periféricos.

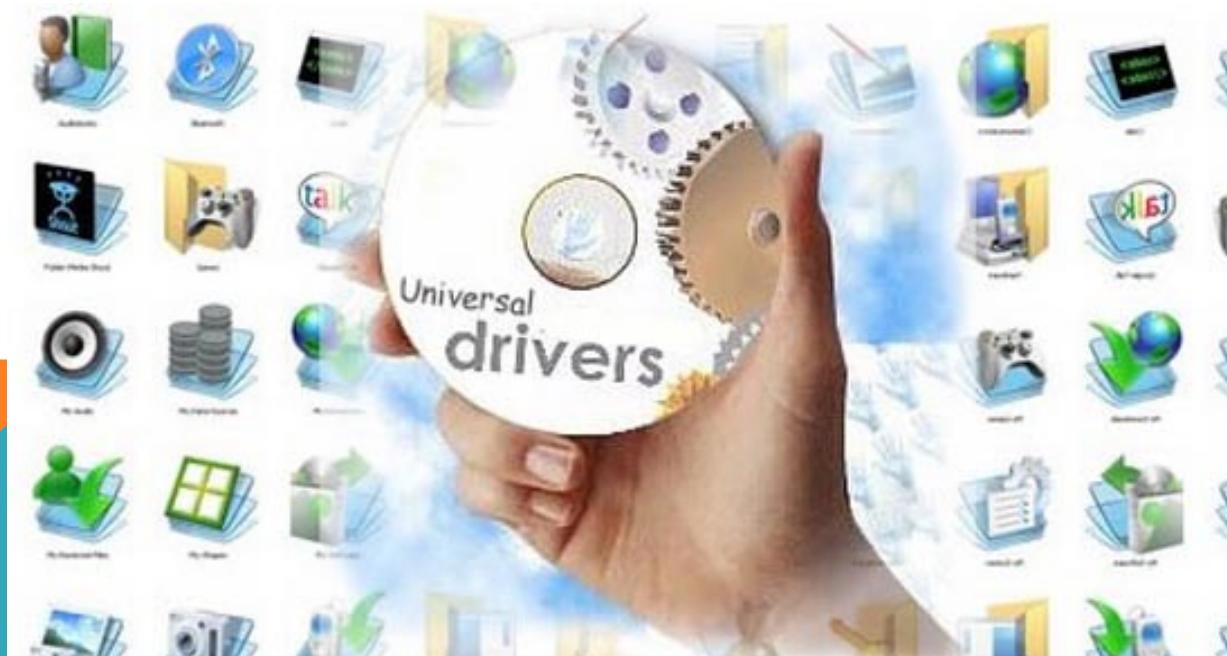
## 4.4. LOS PERIFÉRICOS

En una primera aproximación podemos hacer una clasificación de los periféricos teniendo en cuenta desde o hacia dónde envían información

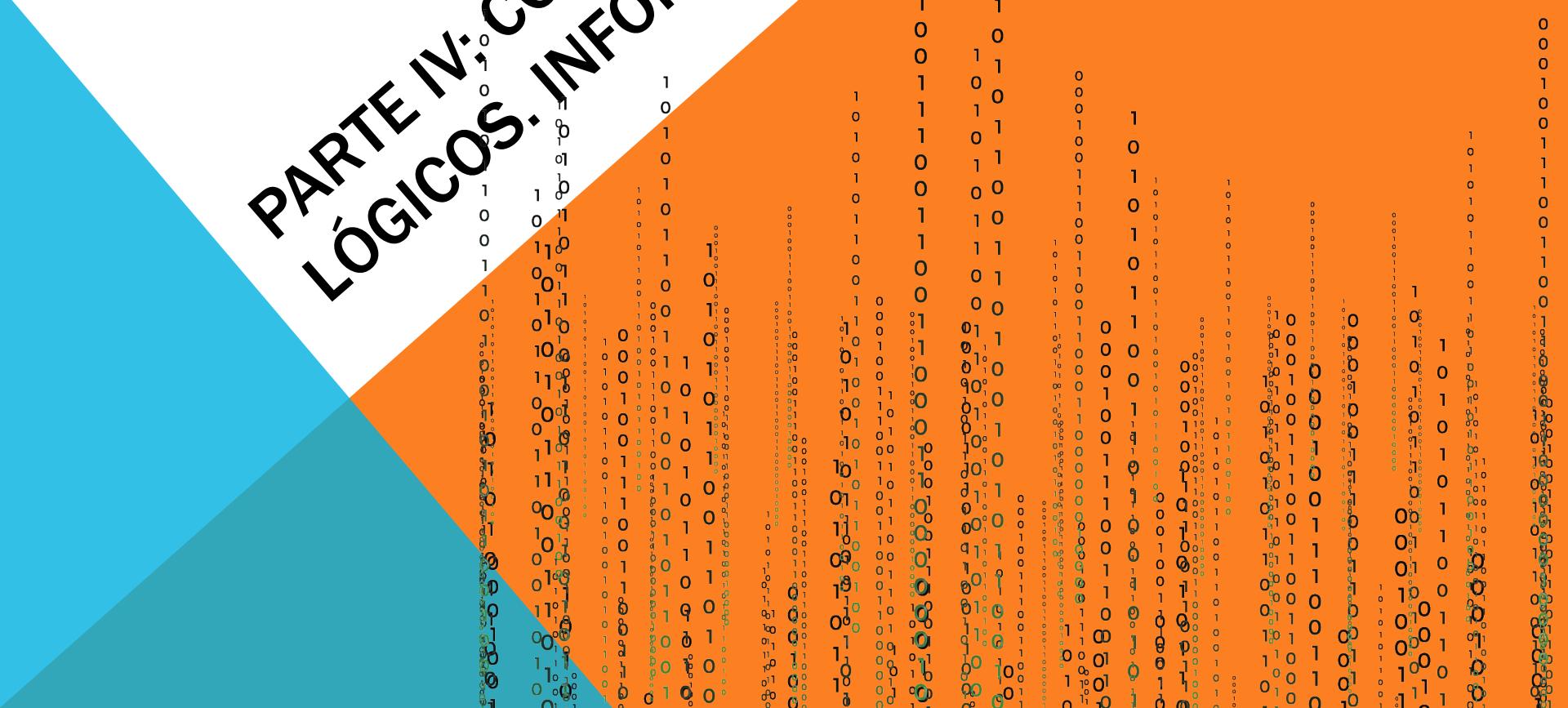


# LOS PERIFÉRICOS: DRIVERS O CONTROLADORES

- Muchos de los periféricos de entrada/salida necesitan un tipo de software especial para ser configurados.
- Esto significa que es necesario introducir dentro de la configuración del ordenador y acorde con nuestro software básico unos programas específicos que permitan al sistema operativo reconocer el periférico y utilizarlo de forma correcta. Estos programas se denominan **drivers o controladores**.



# **PARTE IV: COMPONENTES LÓGICOS. INFORMACIÓN**



## 5. COMPONENTES LÓGICOS

- Una vez vistos los componentes hardware de un sistema informático, hay que tener en cuenta que para que un sistema informático sea útil es necesario que procese información.
- La información que se procesa en un sistema informático puede ser de diferentes tipos: textos, gráficos, música, etc.

**Vamos a ver de forma general los tipos de datos que procesa un sistema informático.**

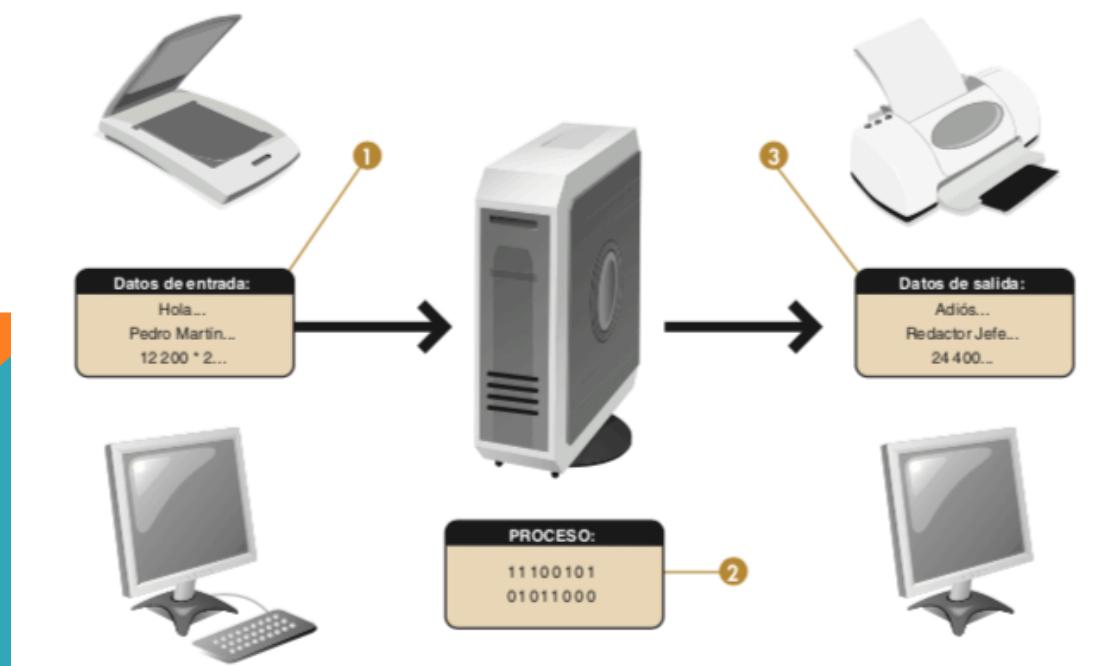
## 5.1 LOS DATOS

- Por sentido común, para que el ordenador funcione, necesita **información con la que trabajar**. Esta información es de varios tipos dependiendo de su función(textos, imágenes....etc.)
- El ordenador también maneja **información que servirá para procesar esos datos**. En este caso, nos estamos refiriendo a programas o aplicaciones informáticas, como procesadores de textos, que se utilizan para procesar datos en formato texto,...etc.
- Por último, un sistema informático necesita otro tipo de software fundamental. Este software está compuesto de programas y datos que ponen en funcionamiento las aplicaciones informáticas. En este caso nos referimos al **sistema operativo**.

# TIPOS DE DATOS

La primera clasificación que podemos hacer de los tipos de datos:

- ① **Datos de entrada.** Son los que se suministran al ordenador desde los periféricos de entrada (teclado, ratón, módem, escáner, etc.) o desde los diferentes soportes de información (disquetes, discos duros, CD-ROM, etc.). Forman la primera fase del tratamiento automático de la información: **entrada**.
- ② **Datos intermedios.** Son aquellos que se obtienen en la segunda fase del tratamiento automático o de la información: **proceso**.
- ③ **Datos de salida.** También llamados resultados, completan el proceso del tratamiento automático de la información: **salida**.



# TIPOS DE DATOS

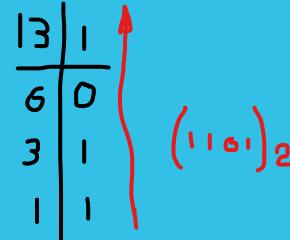
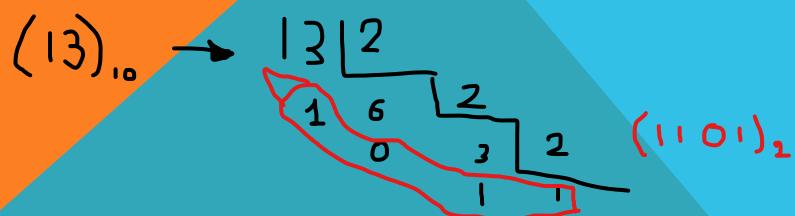
La segunda clasificación que podemos hacer de los tipos de datos, según varíen o no durante el proceso:

- **Datos fijos.** Son los que permanecerán constantes durante el proceso o programa que se les aplique. Los datos fijos reciben el nombre de constantes.
- **Datos variables.** Son aquellos que sí se modifican a lo largo del proceso según sucedan determinadas condiciones o acciones realizadas por los programas.
- **Según la forma de ser utilizados por el ordenador:**
  - **Datos numéricos.** Son los dígitos del 0 al 9.
  - **Datos alfabéticos.** Son las letras mayúsculas y minúsculas de la A hasta la Z.
  - **Datos alfanuméricos.** Son una combinación de los anteriores, más una serie de caracteres especiales (\*, /, -, %, etc.).

## 5.2 SISTEMAS DE CODIFICACIÓN

- Los **sistemas de codificación** se utilizan para procesar la información que el usuario entiende y el ordenador no. Es evidente que el usuario y el sistema informático trabajan en lenguajes diferentes.
- La memoria del ordenador, y por extensión el resto de componentes internos, no entiende de letras o números. **Solamente entiende de corriente eléctrica.**
- Por eso, cuando el usuario quiere almacenar una letra en memoria, por ejemplo, la primera letra de su documento de texto, el ordenador, gracias al sistema operativo y a los componentes de hardware, se encarga de transformar la letra y de almacenarla en un conjunto (normalmente 8 bits) de impulsos eléctricos.

$$(352)_{10} = 3 \cdot 10^2 + 5 \cdot 10^1 + 2 \cdot 10^0$$



## 5.2 SISTEMAS DE CODIFICACIÓN

Se define un sistema de numeración como el conjunto de símbolos y reglas que se utilizan para representar cantidades o datos numéricos.

- Estos sistemas se caracterizan por la base a la que hacen referencia y que determina el distinto número de símbolos que lo componen. Nosotros utilizamos el sistema de numeración en base 10, compuesto por 10 símbolos diferentes (del 0 al 9).
- Los sistemas de numeración que utilizamos son sistemas posicionales, es decir, el valor relativo que cada símbolo representa queda determinado por su valor absoluto y por la posición que ocupe dicho símbolo en un conjunto.

# SISTEMAS DE NUMERACIÓN

Todos los sistemas posicionales están basados en el **Teorema Fundamental de la Numeración (TFN)**, que sirve para relacionar una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal.

En él,  $X$  es el valor absoluto del dígito en cuestión,  $i$  es la posición que ocupa el dígito con respecto al punto decimal y  $B$  es la base. El símbolo  $\Sigma$  (sumatorio) indica que para obtener el valor del número (NÚM), será necesario sumar todos los productos.

Esta misma fórmula también se puede expresar de la siguiente forma:

$$\text{NÚM} = X_n \cdot 10^n + \dots + X_2 \cdot 10^2 + X_1 \cdot 10^1 + X_0 \cdot 10^0 + X_{-1} \cdot 10^{-1} + X_{-2} \cdot 10^{-2} \dots + X_N \cdot 10^N$$

$$283 = 2 \cdot 10^2 + 8 \cdot 10^1 + 3 \cdot 10^0 = 200 + 80 + 3$$

Ten en cuenta



El **Teorema Fundamental de la Numeración (TFN)** queda determinado por la fórmula siguiente:

$$\text{NÚM} = \Sigma X_i \cdot B^i$$

# SISTEMAS DE CODIFICACIÓN NUMÉRICOS

Son tres los sistemas de codificación que utiliza habitualmente un sistema informático:

- **Binario.** Este sistema emplea dos símbolos diferentes: el cero y el uno (0,1). Es el sistema que maneja el ordenador internamente, ya que lo utilizan sus componentes electrónicos.

Cada uno de estos símbolos recibe el nombre de **bit**, entendiendo por tal la mínima unidad de información posible.

Los símbolos del sistema decimal pueden representarse (codificarse) en binario mediante el TFN. Cada símbolo decimal puede representarse con una combinación de cuatro bits.

- **Octal.** Es un sistema en base 8 que utiliza los símbolos del 0 al 7 para representar las cantidades, las cuales quedan reproducidas posicionalmente por potencias de 8. El sistema de numeración en base 8 tiene una correspondencia directa con el binario, ya que cada símbolo en base 8 puede representarse mediante una combinación de 3 bits.
- **Hexadecimal.** Es un sistema de numeración en base 16. Utiliza 16 símbolos diferentes, del 0 al 9 y los dígitos valores (o letras) A, B, C, D, E y F. Estas letras representan, respectivamente, los dígitos 10, 11, 12, 13, 14 y 15 del sistema decimal. Este sistema también tiene una correspondencia directa con el sistema binario, ya que cada símbolo en base 16 se puede representar mediante una combinación de 4 bits.

# SISTEMAS DE CODIFICACIÓN NUMÉRICOS

- El sistema que maneja internamente un ordenador es el binario, pero, en ocasiones, por comodidad en el manejo de los datos, se suele utilizar el octal y el hexadecimal, ya que mucha de la información que nos muestra el sistema operativo, como direcciones de memoria, está expresada en hexadecimal.

Decimal	Binario	Base 8	Base 16
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

*Sistemas decimal, binario, octal y hexadecimal.*

# CAMBIOS DE BASE

De base 10 a base 2



## Caso práctico 1

Pasar a base 2 el número 90 que está en base 10

Primero dividimos el número por 2 (base destino) y el cociente que obtenemos lo dividimos de nuevo por 2. El nuevo cociente lo volvemos a dividir por 2, y así sucesivamente hasta que aparezca un cociente igual a 0.

$$90 : 2 = 45. \text{ Resto } 0.$$

$$45 : 2 = 22. \text{ Resto } 1.$$

$$22 : 2 = 11. \text{ Resto } 0.$$

$$11 : 2 = 5. \text{ Resto } 1.$$

$$5 : 2 = 2. \text{ Resto } 1.$$

$$2 : 2 = 1. \text{ Resto } 0.$$

$$1 : 2 = 0. \text{ Resto } 1.$$

Ordenamos los restos sucesivos que aparecen en las divisiones, pero en orden inverso, y obtenemos la nueva codificación en base 2:

Resultado:  $90_{(10)} = 1011010_{(2)}$

# CAMBIOS DE BASE

De base 10 a base  
8 o 16

Si queremos pasar el mismo número a base 8 y 16, la forma de proceder sería la misma, teniendo en cuenta que ahora el divisor es el 8 o el 16, respectivamente.

Paso a base 8:

$$90 : 8 = 11. \text{ Resto } 2.$$

$$11 : 8 = 1. \text{ Resto } 3.$$

$$1 : 8 = 0. \text{ Resto } 1.$$

Resultado:  $90_{(10)} = 132_{(8)}$

Paso a base 16:

$$90 : 16 = 5. \text{ Resto } 10 \text{ (A).}$$

$$5 : 16 = 0. \text{ Resto } 5.$$

Resultado:  $90_{(10)} = 5A_{(16)}$

Como se puede ver en este último caso, el primer resto ha sido 10. Pero este símbolo en hexadecimal no existe; existe la A como símbolo décimo de la base.

# CAMBIO DE BASE

De base 2 a base 10

## Caso práctico 2



Pasar el número 1001 de binario a base 10

Primero se toman los dígitos binarios, 4 en total, y se van multiplicando por potencias de 2 de izquierda a derecha. El último exponente que pondremos en base 2 y con el que multiplicaremos el último dígito será el  $n-1$ , siendo  $n$  el número de dígitos que tiene la cifra de base 2.

$$1001_{(2)} = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = 9_{(10)}$$

### Ten en cuenta



El Teorema Fundamental de la Numeración (TFN) queda determinado por la fórmula siguiente:

$$\text{NÚM} = \sum X_i \cdot B^i$$

# CAMBIO DE BASE

Cambios de base de numeración: Modo directo

## Bases equivalentes

- Los cambios de base **entre bases equivalentes**, como son las bases 2, 8 y 16, se pueden realizar de forma directa, teniendo en cuenta la equivalencia de bits con la que se puede representar cada dígito de estas bases en binario. En base 8, un dígito octal queda representado por una combinación de 3 bits ( $2^3 = 8$ , siendo 3 el número de bits). En hexadecimal, la asociación es de 4 bits ( $2^4 = 16$ ).
- Como cada número en base 8 y en base 16 tiene una correspondencia directa con el número en binario mediante un conjunto de 3 y 4 bits respectivamente, si queremos transformar un número en base 8 o en base 16 a un número en base 2 o viceversa, bastará con formar grupos de 3 o 4 bits respectivamente. **Esta transformación se llama directa.**

$$\begin{array}{r} \overbrace{\text{1}}^2 + \overbrace{\text{0}}^2 + \overbrace{\text{1}}^2 + \overbrace{\text{1}}^0 \\ \text{6} \quad \text{7} \quad \text{2} \quad \text{7} \end{array} = (6727)_8$$

$$\begin{array}{r} \overbrace{\text{1}}^8 + \overbrace{\text{1}}^4 + \overbrace{\text{0}}^1 = 13 = \text{D} \\ \text{D} \quad \text{D} \quad \text{7} \end{array} = (\text{DD7})_{16}$$

# CAMBIO DE BASE: MODO DIRECTO

## Bases equivalentes

### Caso práctico 3



#### Pasar el número 132 de base 8 a base 16

En primer lugar, pasamos el 132 que está en octal a binario de forma directa. Como cada dígito octal se puede expresar con 3 dígitos binarios, tenemos:

$$132_{18} = 001 \ 011 \ 010_2 = 001011010_2$$

Así, transformamos directamente el dígito 1 en 001, el 3 en 011 y el 2 en 010. Mirando la equivalencia de la Tabla 1.3, vemos que cada dígito en base 8 tiene su correspondencia con 3 dígitos binarios.

Ahora, para pasar a base 16, basta con hacer grupos de 4 bits empezando por la derecha. Si faltan dígitos por la izquierda, los completamos con 0, aunque en nuestro caso no son significativos, ya que como en cualquier sistema de numeración, los 0 a la izquierda no tienen valor.

$$001011010_2 = 0000 \ 0101 \ 1010_2 = 5A_{16}$$

El bloque de 4 bits de más a la derecha tiene su equivalencia con el dígito 10 en hexadecimal, pero como este símbolo no existe en este sistema de numeración, lo hacemos corresponder con su símbolo correspondiente que es la letra A. El bloque del centro se corresponde con el dígito 5 y, evidentemente, los cuatro 0 de la izquierda representan un 0, y como tal, no tiene valor precisamente por estar a la izquierda. En este caso, el resultado sería el siguiente:

Resultado:  $132_{18} = 5A_{16}$

El mismo caso sería el paso de base 16 a base 8. Para ello bastaría pasar a binario el número en hexadecimal y hacer bloques de 3 bits.

# CAMBIO DE BASE: MODO INDIRECTO

Bases no equivalentes

- También podemos realizar un cambio de base por el método indirecto, que **consiste en pasar el número de base n a base 10, y posteriormente pasarlo a base m**. En nuestro ejemplo,  $n = 8$  y  $m = 16$ . Este método se utiliza siempre que las bases de numeración no tenga correspondencia posicional.

## RESUMEN:

- Si queremos transformar un número de base 6 a base 5, es evidente que siempre necesitaremos pasar por base 10.
- Si las bases son binario, octal y hexadecimal, al ser potencias de 2 y ser equivalentes, el paso puede ser directo.

# Ejercicios



## Caso práctico 4

Pasar el número 132 en octal a base 16, pasando por base 10



## Caso práctico 5

Realizar los siguientes cambios de base:

- a) Pasar el número 0111 1011 1010 0011 que está en binario a base 16 y base 8.
- b) Pasar el número 100 101 100 que está en binario a base 8 y base 16.
- c) Pasar el número 1274 de base 8 a base 2 y a base 16.
- d) Pasar el número ABF de base 16 a base 8 y base 2.

# SISTEMAS DE CODIFICACIÓN ALFANUMÉRICOS

- Ya sabemos que los datos, además de numéricos, pueden ser alfabéticos o alfanuméricos.
- Normalmente, con los datos alfanuméricos podemos construir instrucciones y programas. Por otro lado, es lógico pensar que el ordenador no solamente procesará datos numéricos, sino también datos alfabéticos y combinaciones de los anteriores, como datos alfanuméricos.
- Los sistemas de codificación alfanumérica sirven para representar una cantidad determinada de símbolos en binario. A cada símbolo le corresponderá una combinación de un número de bits.

# SISTEMAS DE CODIFICACIÓN ALFANUMÉRICOS

**Los sistemas de codificación alfanumérica más importantes son:**

- **ASCII (American Standard Code for Information Interchange).** Este sistema utiliza una combinación de 7 u 8 bits, dependiendo del fabricante, para representar cada símbolo.
- **Es el más utilizado y el que emplea símbolos diferentes (28).** Con este código se pueden representar dígitos del 0 al 9, letras mayúsculas de la A a la Z, letras minúsculas, caracteres especiales y algunos otros denominados de control.

# CÓDIGO ASCII

Caracteres ASCII de control

00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc. vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sinc.)
23	ETB	(fin bloque trans.)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

Caracteres ASCII imprimibles

32	espacio	64	@	96	'
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	:	91	[	123	{
60	<	92	]	124	}
61	=	93	^	125	}
62	>	94	_	126	-
63	?	95	-		

ASCII extendido  
(Página de código 437)

128	ç	160	à	192	ł	224	ó
129	ù	161	í	193	ł	225	ś
130	é	162	ó	194	ł	226	ő
131	â	163	ú	195	ł	227	ô
132	ã	164	ñ	196	—	228	ö
133	à	165	ñ	197	ł	229	ò
134	à	166	*	198	ł	230	µ
135	ç	167	*	199	ł	231	þ
136	è	168	ż	200	ł	232	پ
137	ę	169	ø	201	ł	233	û
138	ę	170	ń	202	ł	234	û
139	í	171	%	203	ł	235	û
140	ł	172	%	204	ł	236	î
141	ł	173	ł	205	—	237	ÿ
142	À	174	«	206	ł	238	—
143	Á	175	»	207	ł	239	—
144	É	176	—	208	ł	240	—
145	æ	177	—	209	ł	241	—
146	æ	178	—	210	ł	242	—
147	ó	179	—	211	ł	243	—
148	ó	180	—	212	ł	244	ł
149	ó	181	ł	213	ł	245	ł
150	ó	182	ł	214	ł	246	ł
151	ó	183	ł	215	ł	247	—
152	ý	184	ø	216	ł	248	—
153	ó	185	—	217	ł	249	—
154	ü	186	—	218	ł	250	—
155	ë	187	—	219	ł	251	—
156	€	188	—	220	ł	252	—
157	ø	189	€	221	ł	253	—
158	*	190	¥	222	ł	254	—
159	f	191	—	223	ł	255	nnbsp

# SISTEMAS DE CODIFICACIÓN ALFANUMÉRICA

Los sistemas de codificación alfanumérica más importantes son:

- **EBCDIC** (Extended BCD Interchange Code). Cada símbolo se representa por una combinación de 8 bits agrupados en dos bloques de cuatro. **Es el formato extendido del BCD.**
- **UNICODE**. Es un código internacional utilizado hoy por hoy en la mayoría de los sistemas operativos. Permite que un producto software o página Web específica se oriente a múltiples plataformas, idiomas o países sin necesidad de rediseño. Concretamente, el código ASCII tiene una tabla específica para cada país, ya que los diferentes símbolos de todos los países no cabrían en una tabla. Define la codificación de caracteres, así como las propiedades y los algoritmos que se utilizan en su aplicación. Proporciona un número único para cada carácter, sin importar la plataforma (hardware), el programa (software) o el idioma.

# **SISTEMAS DE CODIFICACIÓN ALFANUMÉRICA**

**La mayoría de líderes del mercado como Apple, HP, IBM, Microsoft, Oracle, Sun, Unisys y otros, han adoptado la norma UNICODE, permitiendo crear aplicaciones y hardware estándar con XML, Java, etc. Es compatible con muchos sistemas operativos actuales, así como con la mayoría de los exploradores de Internet, permitiendo que un producto software se oriente a varias plataformas o idiomas sin necesidad de rediseño.**

# MEDIDAS DE LA INFORMACIÓN

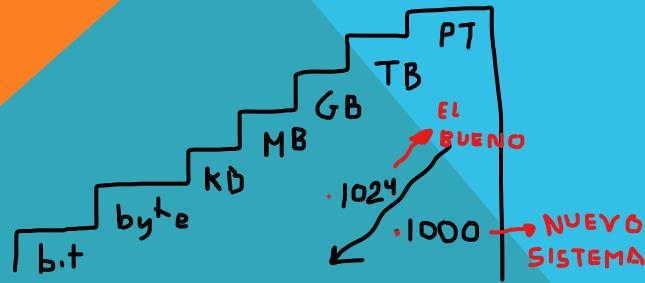
Es la mínima unidad de información. Representado por un 0 o un 1. Para almacenar mayor información tendremos que recurrir a otras unidades de medida mayores al **BIT**, es decir, a sus múltiplos

**Nibble** o cuarteto: 4 bits.

**Byte u octeto**: 8 bits. Se usa para representar un carácter alfanumérico

UNIDADES	ABREVIATURA	SE HABLA DE	REPRESENTA
1 Kilobyte	KB	Kas	1024 bytes
1 Megabyte	MB	Megas	1024 Kb (1048576 bytes)
1 Gigabyte	GB	Gigas	1024 Mb (1073741824 bytes)
1 Terabyte	TB	Teras	1024 Gb (1 billón de bytes)

Busca por Internet todos los múltiplos mayores al Terabyte



# MEDIDAS DE LA INFORMACIÓN



## Ampliación

Bit = mínima unidad de información.

4 Bits = Nibble o cuarteto.

8 Bits = 1 Byte.

1 024 Bytes = 1 Kilobyte.

1 024 Kilobytes = 1 Megabyte (Mb).

1 024 Megabytes = 1 Gigabyte (Gb).

1 024 Gigabytes = 1 Terabyte (Tb).

1 024 Terabytes = 1 Petabyte (Pb).

1 024 Petabytes = 1 Exabyte (Eb).

1 024 Exabytes = 1 Zettabyte (Zb).

1 024 Zettabytes = 1 Yottabyte (Yb).

1 024 Yottabytes = 1 Brontobyte (Bb).

1 024 Brontobytes = 1 Geopbyte (Geb).

# Solución



## Caso práctico 5

Realizar los siguientes cambios de base:

- Pasar el número 0111 1011 1010 0011 que está en binario a base 16 y base 8.
- Pasar el número 100 101 100 que está en binario a base 8 y base 16.
- Pasar el número 1274 de base 8 a base 2 y a base 16.
- Pasar el número ABF de base 16 a base 8 y base 2.

a) Primero, hacemos el cambio de base 16. Agrupamos los bits de 4 en 4 empezando por la derecha. El resultado es el siguiente: 0111 1011 1010 0011<sub>2</sub>. Localizamos los dígitos equivalentes en base 16 y el resultado que obtenemos es el siguiente: 7 **B** A 3<sub>16</sub>.

Del mismo modo, pero realizando agrupaciones de 3 en 3 bits, obtendremos el número equivalente en base 8.

$$000 \underline{111} \ 101 \ \underline{110} \ 100 \ \underline{011}_2 = 7 \ 5 \ 6 \ 4 \ 3_8$$

- b) Procediendo de forma similar al caso a), los resultados obtenidos son los siguientes:

$$\underline{100} \ 101 \ \underline{100}_2 = 4 \ 5 \ 4_8$$

$$\underline{0001} \ 0010 \ \underline{1100}_2 = 1 \ 2 \ C_{16}$$

- c) Aquí, el procedimiento es a la inversa. Tomamos de derecha a izquierda cada dígito del número de base 8 y escribimos sus equivalentes en binario. Cada dígito en base 8 corresponde a 3 dígitos binarios. Este es el resultado:

$$1 \ 2 \ 7 \ 4_8 = 001 \ \underline{010} \ \underline{111} \ \underline{100}_2$$

Obtenido el número en binario, podremos agrupar los dígitos de 4 en 4 de derecha a izquierda para obtener así el correspondiente número en base 16.

$$0010 \ 1011 \ \underline{1100}_2 = 2 \ B \ A_{16}$$

- d) De forma similar, lo primero es pasar el número de base 16 a binario, buscando su equivalencia de 4 bits por cada dígito hexadecimal.

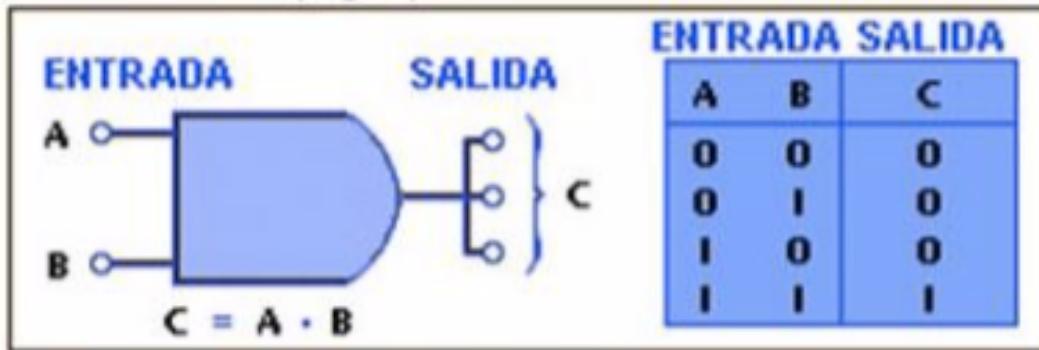
$$A \ B \ F_{16} = 1010 \ 1011 \ \underline{1111}_2$$

Luego, se agrupan los dígitos binarios de 3 en 3 de derecha a izquierda para obtener el equivalente en base 8. Así:

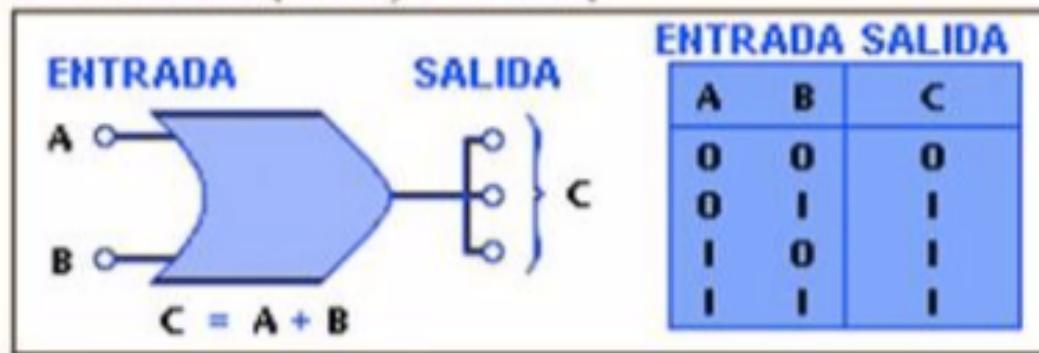
$$101 \ \underline{010} \ \underline{111} \ \underline{111}_2 = 5 \ 2 \ 7 \ 7_8$$

# PUERTAS Y OPERACIONES LÓGICAS

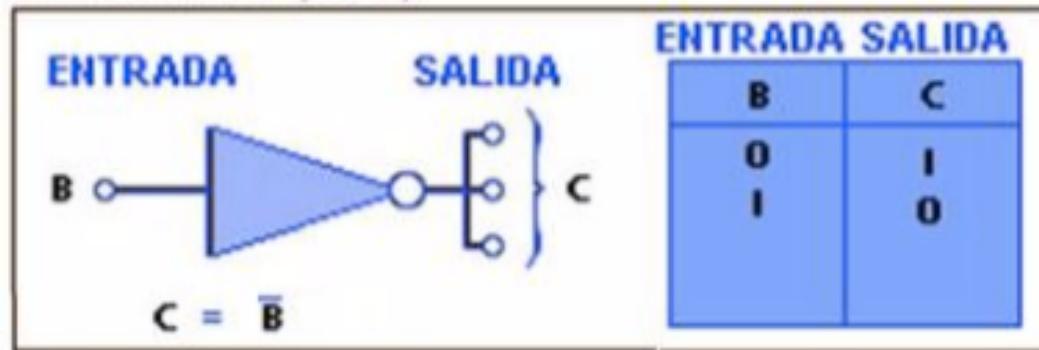
PUERTA AND (A y B)



PUERTA OR (A o B, o ambos)

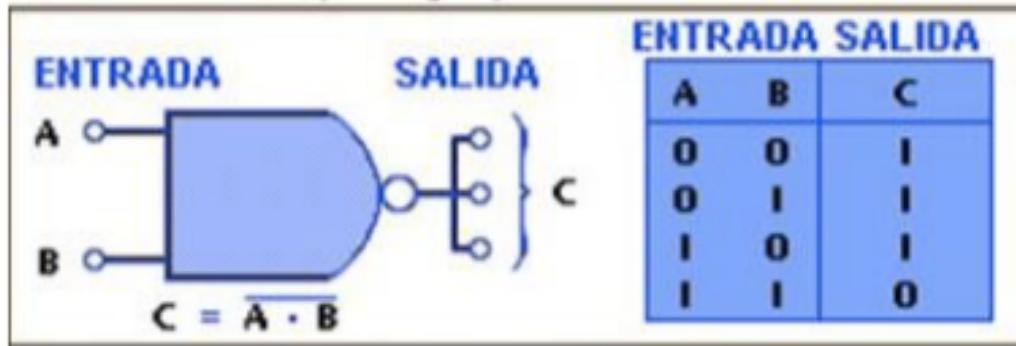


PUERTA NOT (no C)

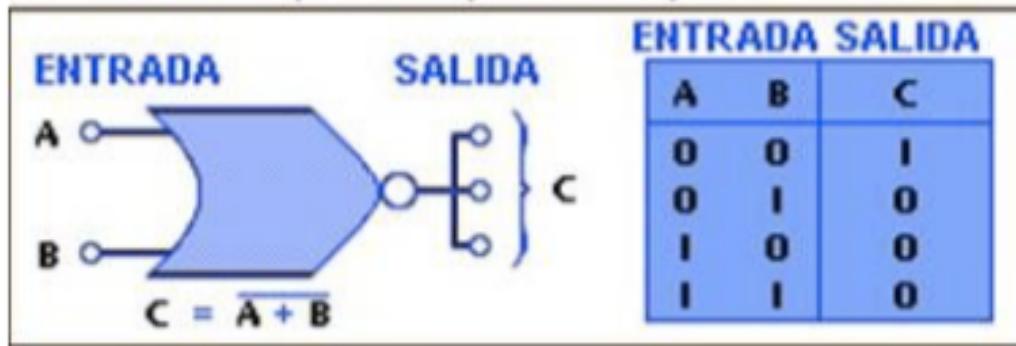


# PUERTAS Y OPERACIONES LÓGICAS

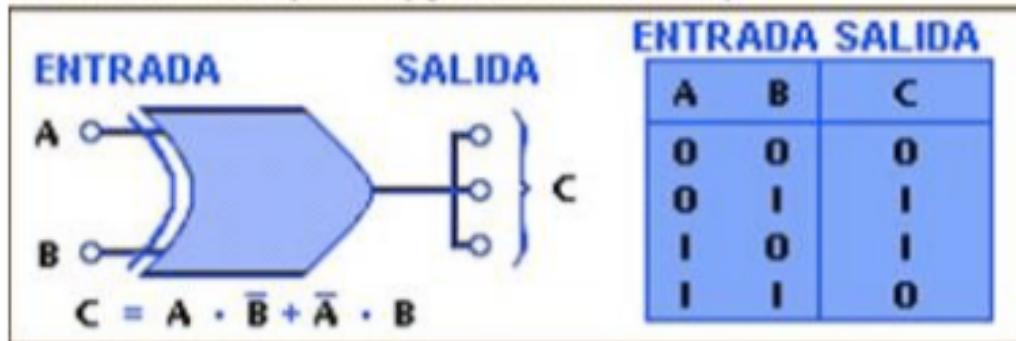
PUERTA NAND (no A y B)



PUERTA NOR (ni A ni B, ni ambos)



PUERTA XOR (A o B, pero no ambos)



"O" EXCLUSIVO