SISTEMAS INFORMÁTICOS UD-1

CONCEPTOS BÁSICOS DE LOS SISTEMAS INFORMÁTICOS (I)

SOFTWARE Y HARDWARE

ÍNDICE

[EL SISTEMA INFORMÁTICO: SOFTWARE Y HARDWARE 3](#_Toc116105535)

[COMPONENTES SOFTWARE: SISTEMA OPERATIVO Y APLICACIONES 3](#_Toc116105536)

[COMPONENTES FÍSICOS: EL HARDWARE 4](#_Toc116105537)

[A. ESTRUCTURA HARDWARE GENERAL 4](#_Toc116105538)

[B. UNIDAD CENTRAL DE PROCESADO (UCP) 4](#_Toc116105539)

[C. MEMORIA 5](#_Toc116105540)

[1. JERARQUÍA DE LA MEMORIA 5](#_Toc116105541)

[2. MEMORIA RAM (RANDOM ACCESS MEMORY) 6](#_Toc116105542)

[3. MEMORIA ROM (READ ONLY MEMORY) 9](#_Toc116105543)

[4. INSTRUCCIONES DE UN PROGRAMA 9](#_Toc116105544)

[D. UNIDADES DE ENTRADA/SALIDA (E/S) 10](#_Toc116105545)

[E. LOS PERIFÉRICOS 11](#_Toc116105546)

[COMPONENTES LÓGICOS: EL SOFTWARE 11](#_Toc116105547)

[A. LOS DATOS DE INFORMACIÓN 11](#_Toc116105548)

# EL SISTEMA INFORMÁTICO: SOFTWARE Y HARDWARE

El **ordenador** o **sistema informático** es una máquina compuesta por:

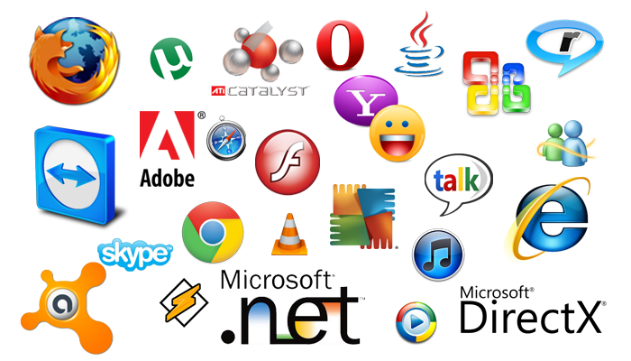
* **Elementos físicos (*hardware*)** los cuales son origen eléctrico-electrónico y son capaces de realizar una gran variedad de trabajos a gran velocidad y con gran precisión.
* **Elementos lógicos (*software*)** los cuales permiten que los elementos físicos realicen sus funciones. Son los **programas**.

El **sistema operativo** es el software base de un sistema informático capaz de hacer que el software procesen información sobre el hardware.

Otro concepto importante es el ***firmware*** que es la parte software propia de los componentes del hardware, por tanto, es un programa básico que controla los circuitos electrónicos de cualquier dispositivo. Se va a encargar de controlar qué lo que tiene que hacer el hardware de un dispositivo y se va a asegurar de que el funcionamiento básico es correcto. Ejemplos de firmware serían la **ROM-BIOS (*Basic Input/Output System*)** o el software de un disco duro que permite que el mecanismo de lectura se pueda mover.

# COMPONENTES SOFTWARE: SISTEMA OPERATIVO Y APLICACIONES

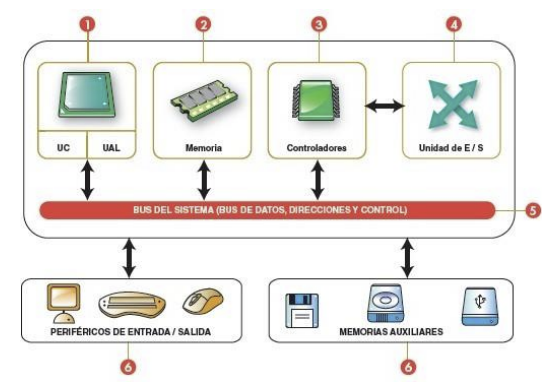
El software se compone de 2 partes:

* **Software básico**. Aquella parte del software sin la cual el sistema no puede funcionar. Es el **sistema operativo**.
* **Software de aplicaciones**. Parte del software que sirven para procesar la información de forma personalizada. Son los **programas** y los **datos**.

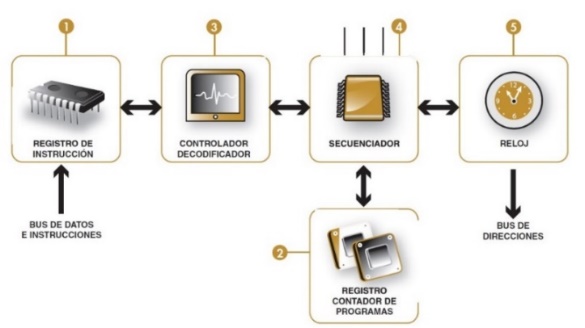
# COMPONENTES FÍSICOS: EL HARDWARE

## ESTRUCTURA HARDWARE GENERAL

La estructura hardware general de un ordenador es:

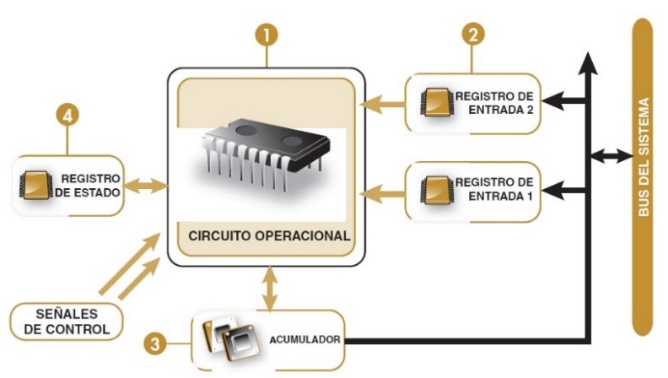
1. **Unidad central de procesado (UCP).** Consta de:
   1. **Unidad de control (UC)**.
   2. **Unidad aritmético-lógica (UAL)**.
2. **Memoria central (MC)** o RAM.
3. **Controladores**.
4. **Unidad de entrada/salida (E/S)**.
5. **Buses**.
6. **Unidades periféricas** o periféricos de entrada/salida.

## UNIDAD CENTRAL DE PROCESADO (UCP)

Es el procesador del ordenador y se encarga de la ejecución de todas las instrucciones que en él ocurren. Se divide en:

* **Unidad de control (UC)**.Gestiona y coordina las instrucciones. Está formada por:

1. **Registro de instrucción**.Almacena 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 qué debe hacer la instrucción.
4. **Secuenciador**. Genera micro-órdenes necesarias para ejecutar la instrucción.
5. **Reloj**. Proporciona una sucesión de intervalos eléctricos a intervalos constantes que permiten que el hardware de un ordenador se pueda comunicar entre sí.

* **Unidad aritmético-lógica (UAL)**. Se encarga de realizar operaciones aritméticas y lógicas sobre la información.

Las lógicas son normas de comparación, para las que se emplean los **operadores del álgebra de Boole**.

|  |  |
| --- | --- |
| OPERADOR | SIGNIFICADO |
| > | Mayor que |
| < | Menor que |
| > = / < = | Mayor o igual / Menor o igual |
| NOT | “No” lógico |
| AND | “Y” lógico |
| OR | “O” lógico |

Contiene los siguientes componentes:

1. **Operacional o circuito operacional**. Realiza las operaciones.
2. **Registros de entrada**. Almacenan los operandos de la operación. Existe un registro por cada operando.
3. **Acumulador**. Almacena los resultados de las operaciones. Se comunica con los registros de entrada para guardar esos resultados en caso de tener que hacer más operaciones.
4. **Registro de estado**. Registra las condiciones de la operación anterior (si es posible o no).

## MEMORIA

1. JERARQUÍA DE LA MEMORIA

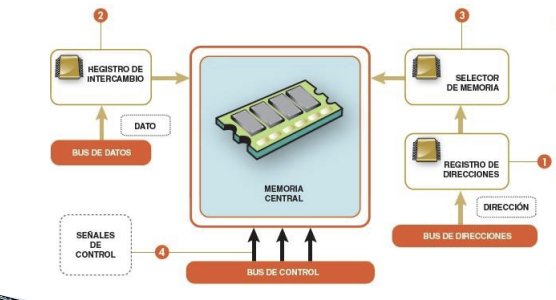
La memoria se puede estratificar en una pirámide clasificándose desde el vértice hasta la base en:

1. **Memoria interna**. Se encuentra en el interior del equipo y es con la que trabaja la CPU directamente para ejecutar los programas. Dentro de este tipo de memoria tenemos: los registros, la memoria caché y la memoria RAM.
2. **Memoria externa**. Aquí se encontrarían los discos duros tanto HDD como SSD.
3. **Memoria extraíble**. Tendríamos los dispositivos ópticos (CDs, Blu-Ray), las memorias USB y los dispositivos magnéticos (disquetes).
4. **Almacenamiento en la nube**. Serían los servidores de internet.

Cuanto más próximos a la base de esa pirámide:

* + Más grande es la memoria (tamaño de almacenamiento).
  + Más lento es el acceso a la información.

1. MEMORIA RAM (RANDOM ACCESS MEMORY)

****Es una **memoria de acceso aleatorio**, lo que significa que el tiempo que se tarda en acceder a la posición A o a la posición B es el mismo. Además, es una **memoria de lectura y escritura**. Es un componente necesario para que se pueda procesar la información ya que todo lo que se procesa en un equipo informático debe pasar por la RAM. Consta de:

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**. Indican si una operación es de lectura o de escritura.

Existen varios tipos de memoria RAM:

* + **DRAM *(Dynamic RAM)***. Necesitan refresco (es decir, necesitan electricidad continua para sobreescribir en todo momento los datos) y son volátiles (una vez que se quedan sin esa electricidad, por ejemplo, al apagar el equipo, se pierden todos los datos). Están construida normalmente mediante condensadores. Es la que utiliza la memoria RAM.
  + **SRAM *(Static RAM)***. Necesitan menos refresco, construida normalmente mediante biestables. Es más rápida y costosa que la DRAM. Es la que utiliza la caché (se ubica dentro de la CPU).
  + **SDRAM *(Synchronous DRAM)***. Incorpora la capacidad de la DRAM y la velocidad de la SRAM, es decir, necesita el refresco de las celdas, pero en un intervalo superior.
  + **DDR - SDRAM *(Double Data Rate - SDRAM***. Compuesta por memorias SDRAM, tiene la característica de que se ejecutan dos transferencias por ciclo de reloj.

Las memorias tienen diferentes buses:

* + **Bus de direcciones**. Formado por varios hilos (en función del número de posiciones de la memoria) que transmiten la información que representa **la dirección de la RAM donde vamos a tener que buscar**. Hay que tener en cuenta que el ordenador no se comunica en base 10 sino en sistema binario por lo que sólo hay 2 números disponibles (el 0 y el 1). Además, cada hilo de dirección sólo puede mandar un número por pulso, lo que se conoce como **bit**, así que si tenemos una memoria con 2 posiciones sólo necesitaremos 1 hilo (bit) para nombrar a todas las posiciones (0 y 1) pero si tenemos 4 posiciones necesitaremos 2 hilos (posiciones 00, 01, 10 y 11) y si tenemos 8 posiciones necesitaremos 3 hilos (posiciones 000, 001, 010, 011, 100, 101, 110 y 111). El número de hilos necesarios para llenar todas las posiciones de una memoria va seguir la siguiente fórmula:

Siendo x el número de hilos necesarios en cada caso. Para agilizar esto se pueden memorizar las siguientes igualdades:

21 = 2

22 = 4

23 = 8

24 = 16

25 = 32

26 = 64

27 = 128

28 = 256

29 = 512

210 = 1024

Por ejemplo, si tenemos una memoria con 128 posiciones necesitaremos de 7 hilos de dirección porque 27 es 128.

* + **Bus de control**. Formado por 6 hilos (es decir, se pueden transmitir 6 bits). Transmite la **orden a realizar**: leer o escribir. Por ejemplo, 001101 para leer y 101111 para escribir.
  + **Bus de datos**. Se usa para los **datos** en sí, los cuales son conjuntos de 0s y 1s.

Las primeras RAMs eran de tipo **SIMM (*Single In-line Memory Module*)**, es decir, sólo tenían un conector (buses) en una cara de las caras. Las actuales son de tipo **DIMM (*Double In-line Memory Module*)**, es decir, tienen conectores en ambas caras.

Las placas base y procesadores actuales, aunque tiene 4 slots o módulos de memoria RAM sólo tienen 2 canales de RAM. Esto es debido a que se usan 2 módulos para cada canal. Debido a esto, si se conectan 2 memorias (completando los módulos que comparten canal) el procesador estaría trabajando en lo que se llama ***dual-channel***, es decir, el procesador puede escribir y leer de las 2 RAMs a la vez lo cual optimiza los procesos. Existe también el ***quad-channel*** y el ***six-channel***, aunque actualmente no están tan extendidos y sólo están presentes en la gama alta.

La **transmisión** de la información se basa en **impulsos eléctricos** de forma que si no tenemos potencial eléctrico (realmente nunca es 0V, siempre hay voltaje residual), se estaría enviando un 0 y si hay potencial, se estaría enviando un 1.

La **velocidad** de esta transmisión se basa en un **ritmo marcado por el reloj** de la UC de la CPU. Esta información del ritmo llega por otro bus extra (de 1 hilo) que transmite los ciclos de potencial eléctrico:

* + En las **SDRAM** sólo se tenía en cuenta la **bajada** de potencial.

Pero debido a que el reloj no puede ir infinitamente rápido, llegó un punto donde se alcanzó un tope de ciclos por segundo donde el circuito no era capaz de entender la diferencia entre encendido y apagado.

* + Debido a la esa limitación se idearon las **DDR - SDRAM.** En estas RAMsse tiene en cuenta tanto la **subida** como la **bajada**, lo que permite una transmisión el doble de rápida. Esto se conoce como ***pumping****.*



En relación a esto, no hay que confundir **MegaHerzios** (MHz) con **MegaTransferencias** (MTrs), cuando se habla de una RAM de 3200MHz es erróneo, serían 3200MTrs en 1600MHz (2MTrs por 1MHz).

1. MEMORIA ROM (READ ONLY MEMORY)

Es una memoria de **sólo lectura** contine programas especiales que sirven para cargar e iniciar el arranque del ordenador.

Guardan **información que no se va a alterar**. Se utiliza en el arranque del ordenador, concretamente en la BIOS.

Existen diferentes tipos de ROM:

* + **PROM *(Programmable ROM)***. Son programables una sola vez.
  + **EPROM *(Erasable PROM)***. Permite cambiar la configuración asignada mediante radiación ultravioleta.
  + **EEPROM *(Electrically EPROM)***. Permite cambiar la configuración mediante corrientes eléctricas.
  + **Flash**. Permiten ser programadas con los valores de tensión propios de los ordenadores.

1. INSTRUCCIONES DE UN PROGRAMA

Los programas se componen de instrucciones.

Diagrama

Descripción generada automáticamenteEl **direccionamiento** es una operación que se realiza cuando el procesador ejecuta o interpreta una instrucción. Consta de dos partes el **código de operación** (parte roja de la imagen) que indica el tipo de operación que se va a hacer (por ejemplo, ADD – para sumar) y el **campo del operando** (parte amarilla de la imagen) puede tener diferentes tipos de datos, en función de los datos que tenga tendremos un tipo de direccionamiento u otro.

Existen 4 tipos:

1. **Direccionamiento inmediato**. En el campo estará el dato que necesitamos para la instrucción. Por ejemplo, ADD A B (código de operación y operandos).
2. **Direccionamiento directo**. En el campo estará la dirección de memoria donde está el dato que necesitamos. Por ejemplo, ADD 010 (código de operación y dirección de memoria y en esa posición de memoria 010 estaría el dato A).
3. **Direccionamiento indirecto**. En el campo estará la dirección de memoria en la que se encuentra la dirección real de dónde se encuentra el dato. Por ejemplo, ADD 010 (código de operación y dirección de memoria, en esa dirección encontraríamos una segunda dirección, por ejemplo, 111 en la que se encontraría el dato real).
4. **Direccionamiento relativo**. En el campo habrá una dirección de memoria a la que se le sumará el contenido de un registro especial que tiene un dato numérico (en binario) y el resultado de esa suma es la posición de memoria en la que se e va a encontrar el dato. Por ejemplo, ADD 010 como direccionamiento base y en el registro especial un 001 (posición 1) dando como resultado el 010 (posición 3) que es donde se encontraría el dato.

## UNIDADES DE ENTRADA/SALIDA (E/S)

Sirven para comunicar el procesador y el resto de los componentes internos con los periféricos de entrada/salida (por ejemplo, la pantalla) y las memorias de almacenamiento externo o auxiliares (por ejemplo, un HDD).

Hay un componente importante dentro del ordenador que está relacionado directamente con el procesador: el **bus**.

* + **Bus de datos.** Transmite información entre la UCP y los periféricos.
  + **Bus de direcciones.** Identifica el dispositivo al que va destinada la información que se transmite por el bus de datos. Es decir, la dirección de un periférico.
  + **Bus de control o sistema.** Organiza y redirige hacia el bus pertinente la información que se tiene que transmitir.

La estructura es la siguiente:

1. **Procesador**.
2. **Buses**
3. **Memoria** **RAM**.

**Gráfico, Gráfico de dispersión

Descripción generada automáticamente**

## LOS PERIFÉRICOS

Son dispositivos hardware con los cuales el usuario puede interactuar con el ordenador (**periféricos de entrada**: un teclado), almacenar o leer datos y/o programas (**periféricos de almacenamiento**: un HDD), imprimir resultados (**periféricos de salida**: una impresora), etc.

Se conectan con la UCP y el resto de los componentes a través de los **puertos** o **conectores externos** que están soldados a la placa base. Esta gestión la lleva a cabo la unidad entrada/salida.

Necesitan un tipo de software espacial para ser configurados: los **drivers**. Estos no se deben confundir con el firmware, son códigos que no se instalan en el dispositivo en sí sino en el SO con el que funcionan.

# COMPONENTES LÓGICOS: EL SOFTWARE

## LOS DATOS DE INFORMACIÓN

Un sistema informático procesa información continuamente. Existen tres tipos de datos:

Diagrama

Descripción generada automáticamente

1. **Datos de entrada**. Son los que se suministran desde los periféricos de entrada o desde diferentes soportes de información.
2. **Datos intermedios**. Son aquellos que se obtienen en la segunda fase de tratamiento automático o de la información. La información pasa a ser conjuntos de 1s y 0s.
3. **Datos de salida**. También llamados resultados, completan el proceso del tratamiento automático de la información.

Otra clasificación que podemos hacer de los datos, según varíen o no durante el proceso, es la siguiente:

* + **Datos fijos**. Son los que permanecen constantes durante el proceso. Reciben el nombre de **constantes**.
  + **Datos variables**. Son aquellos que sí se modifican durante el proceso.