

Programación I

Tema 1

Grado en Ingeniería Informática

Tema 1. Introducción a la Programación

1. Fundamentos de Informática

1.1. Definiciones Básicas

1.2. Estructura de una Computadora

1.2.1. Unidades Funcionales

1.3. Prestaciones de una Computadora

1.4. Tipos de Computadoras

1.5. Software de una Computadora

1.5.1. Software de Control

1.5.2. Utilidades

1.5.3. Aplicaciones

Tema 1. Introducción a la Programación

2. Conceptos Básicos de Programación

2.1. Instrucciones y Programas

2.2. Algoritmos

2.3. Lenguajes de Programación

2.4. Metodología

2.5. Pseudocódigo y Diagramas de Flujo

1.1. Definiciones Básicas

- **Informática:** Información + automática
- “Conjunto de conocimientos científicos y técnicas que hacen posible el *tratamiento automático de la información* por medio de computadoras.” (RAE)
- “Es una ciencia que estudia métodos, técnicas, procesos, con el fin de almacenar, procesar y transmitir información y datos en formato digital.”
“La rama de la tecnología que estudia el tratamiento automático de la información.” (Wikipedia)

Informática

1. Confluyen una serie de disciplinas:

- Ciencias de la computación, programación, ingeniería del software, arquitectura de computadores, redes de computadores, inteligencia artificial, robótica, sistemas operativos, ...

2. Se aplica a infinidad de áreas:

- Medicina, gestión empresarial y de negocios, procesos industriales, turismo, transportes, educación, investigación, ...

Información y datos

- **Información:**

- “Es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.” (Wikipedia)

- **Datos:**

- “Es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa. Los datos describen hechos empíricos, sucesos y entidades.” (Wikipedia)



Un dato por sí mismo no constituye información, es el procesamiento de los datos lo que nos proporciona información.

1. Fundamentos de Informática

Codificación

Transformación que representa los elementos de un conjunto mediante los de otro, de forma que a cada elemento del primer conjunto le corresponde un elemento distinto del segundo.

Ordenadores - codificación binaria

Toda la información
un computador
como unos y ceros.



Codificación

Solo con unos y ceros es posible representar casi cualquier información:

- Números (binario):
01011001 => 89
- Letras (codificaciones de caracteres):
01011001 => Y [ASCII]
- Imágenes, sonido, voz, ...

Decimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010

Unidades de información

BIT (Binary digiT): unidad de información más pequeña que puede procesar una computadora.

Byte: número de bits necesarios para almacenar un carácter.
Generalmente: 1 byte = 8 bits.

Múltiplos:

1 Kilo (K)	$2^{10} = 1024 \approx 10^3$
1 Mega (M)	$2^{10} K = 2^{20} = 1.048.576 \approx 10^6$
1 Giga (G)	$2^{10} M = 2^{30} = 1.073.741.824 \approx 10^9$
1 Tera (T)	$2^{10} G = 2^{40} \approx 10^{12}$
1 Peta (P)	$2^{10} T = 2^{50} \approx 10^{15}$
1 Exa (E)	$2^{10} P = 2^{60} \approx 10^{18}$

Ordenador - Computadora

- Es una máquina capaz de aceptar unos datos de entrada, efectuar con ellos operaciones lógicas y aritméticas, y proporcionar la información resultante a través de un medio de salida. Todo ello sin intervención de un operador humano y bajo el control de un programa de instrucciones previamente almacenado en el propio ordenador.



Ordenador - Computadora

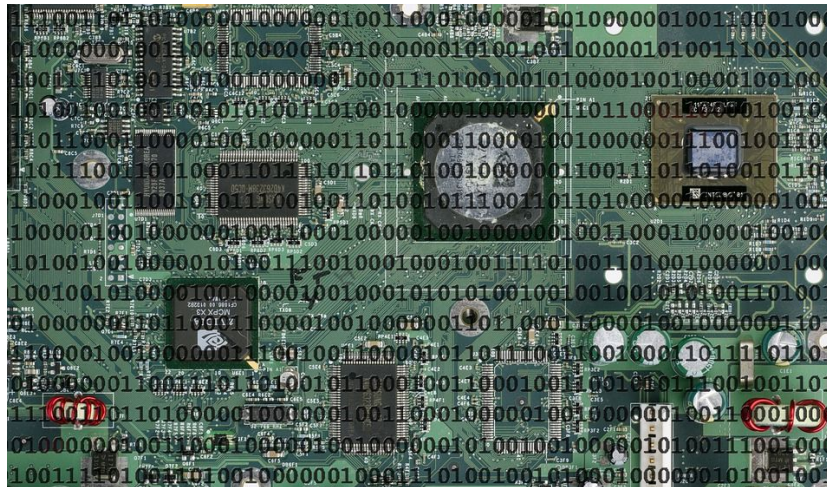
- *“Máquina electrónica capaz de almacenar información y tratarla automáticamente mediante operaciones matemáticas y lógicas controladas por programas informáticos.”*



1. Fundamentos de Informática

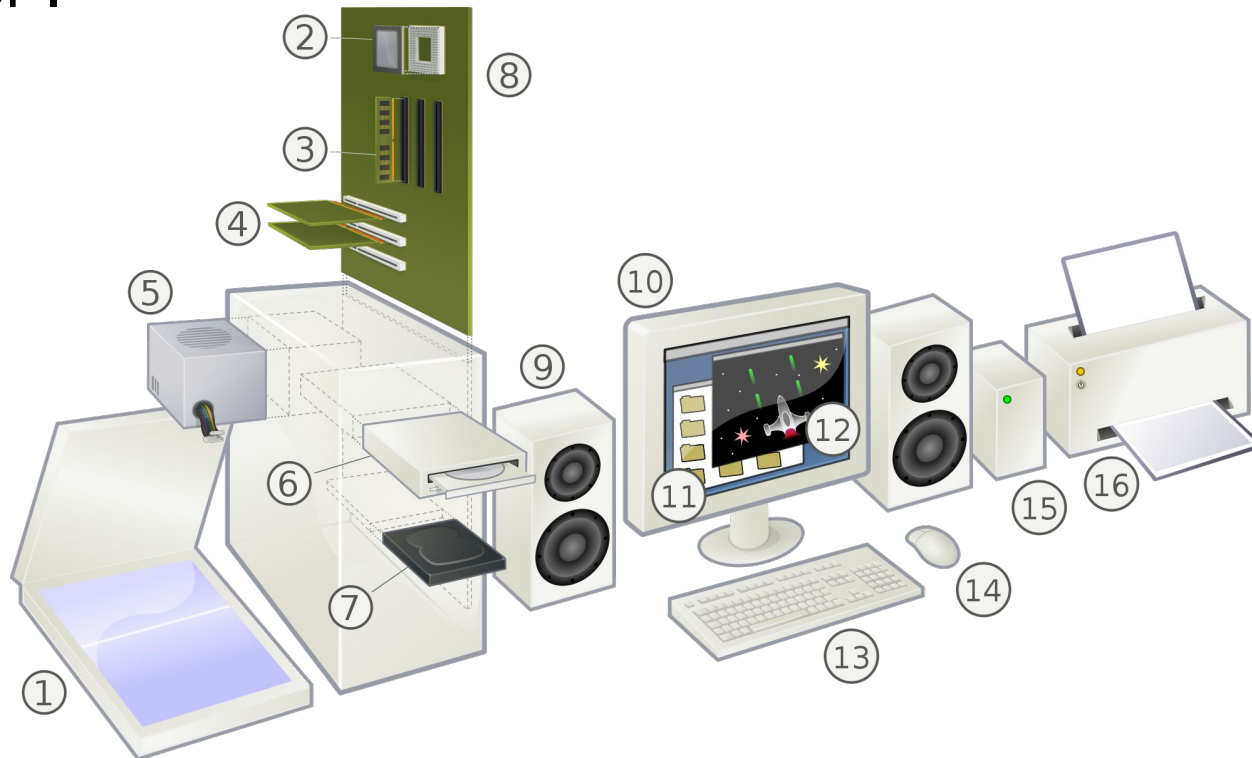
Sistema informático

- “Es un sistema que permite almacenar y procesar información” (Wikipedia)
- “Conjunto de elementos físicos y lógicos necesarios para la realización y explotación de aplicaciones informáticas”
- Hardware + Software



Hardware

Soporte Físico o **Hardware** de un ordenador es el conjunto de circuitos electrónicos, cables, dispositivos electromecánicos, y otros elementos físicos que forman el ordenador .



Software

Soporte lógico o **Software** de un ordenador es el conjunto de programas ejecutables por el ordenador o sistema informático:

- Sistemas operativos
- Aplicaciones
- Firmware
- Apps, ...

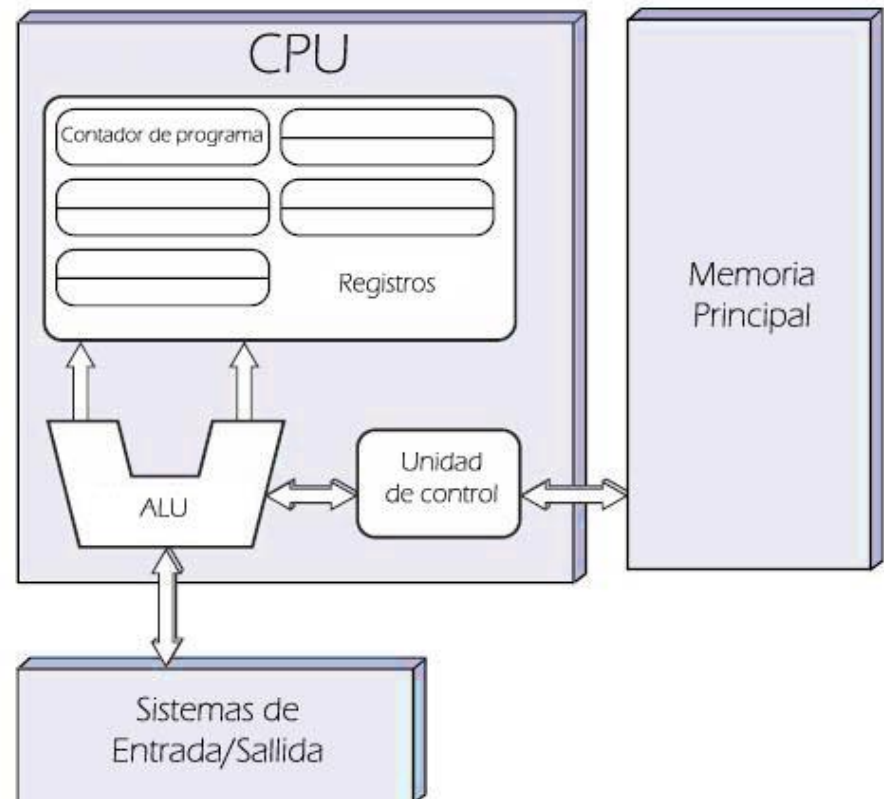


1.2. Estructura de una Computadora

La base de la estructura de un computador actual fue definida en 1945 por Von Neumann.

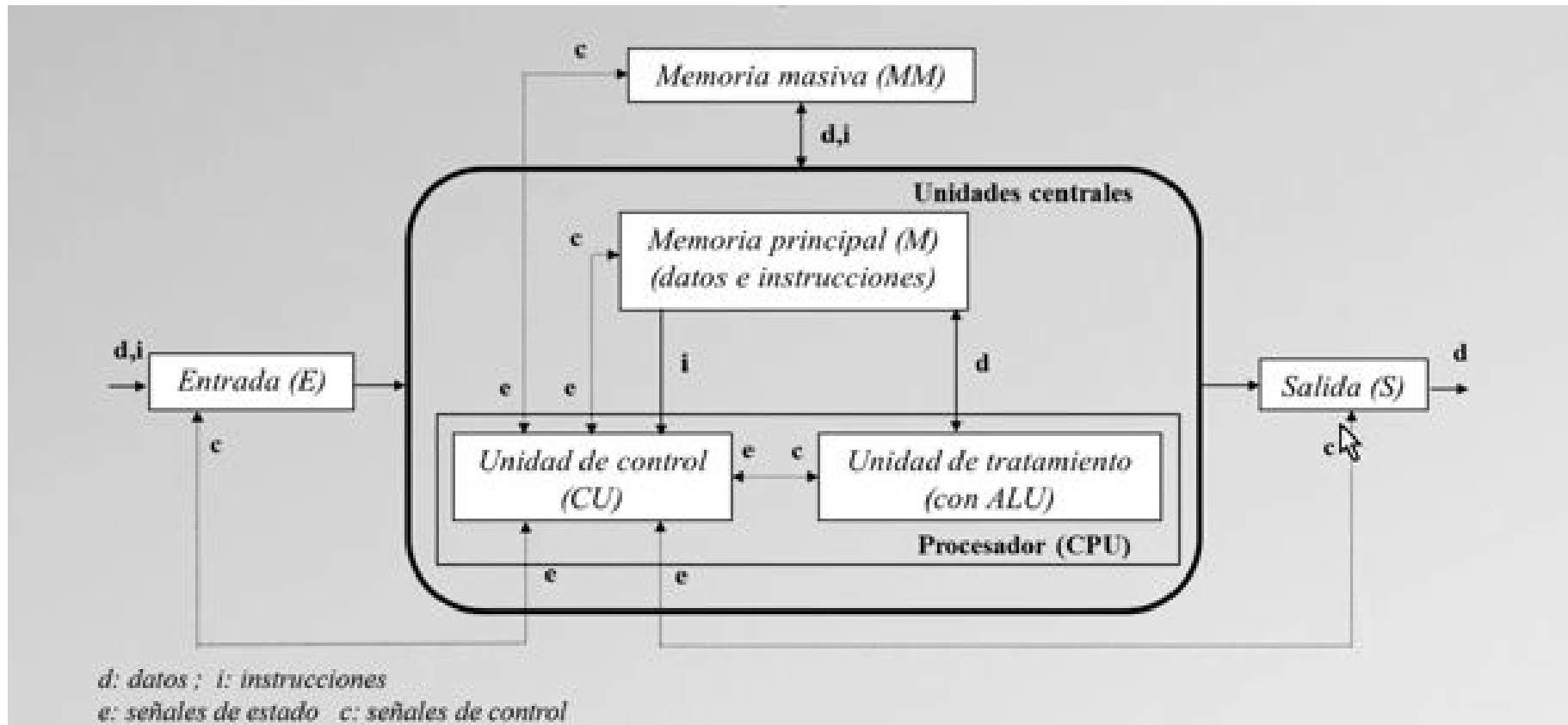
Componentes:

- CPU
- Memoria
- Entrada/Salida



1. Fundamentos de Informática

1.2.1. Unidades Funcionales



Prieto, A., Lloris, A., & Torres, J. C. (2006). Introducción a la Informática. McGraw-Hill.

1. Fundamentos de Informática

Unidades de entrada y de salida

Unidad de Entrada: dispositivo por donde se introducen en la computadora los datos e instrucciones: un teclado, un scanner, un lápiz óptico, ...

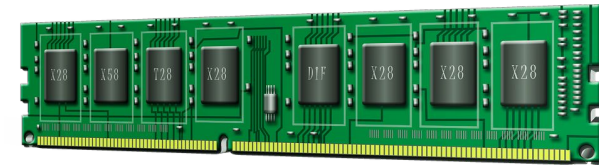


Unidad de Salida: dispositivo por donde se obtienen los resultados de los programas ejecutados en el ordenador: monitor, impresora, plotter, ...



Memoria

Memoria: es la unidad donde se almacena tanto los datos como las instrucciones (programas).



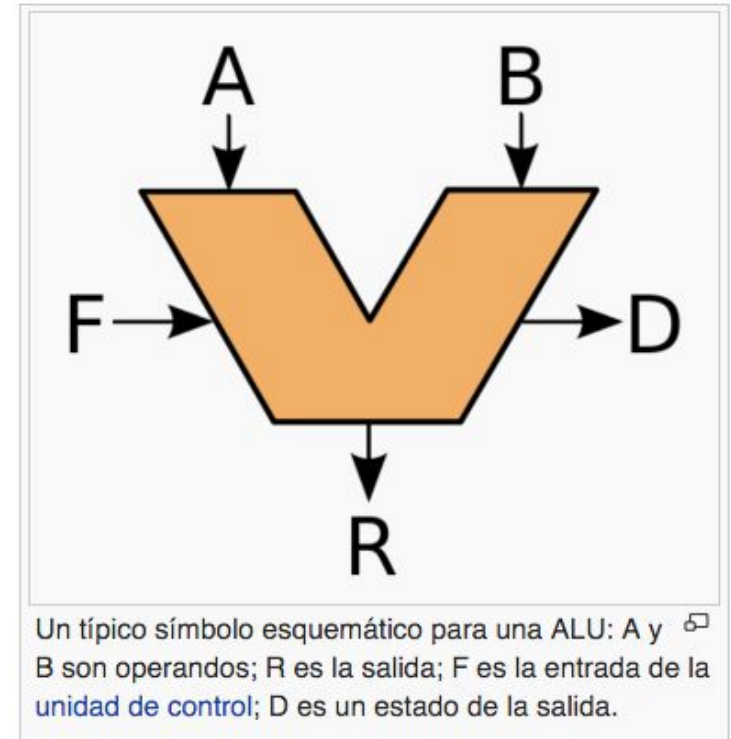
- **Memoria principal**, central o interna. Circuitos internos. Gran velocidad. Es volátil.
- **Memoria auxiliar**, secundaria o externa: más lenta. Es permanente. Discos, dispositivos USB,...



Para que un programa se ejecuta debe estar almacenado en la memoria principal.

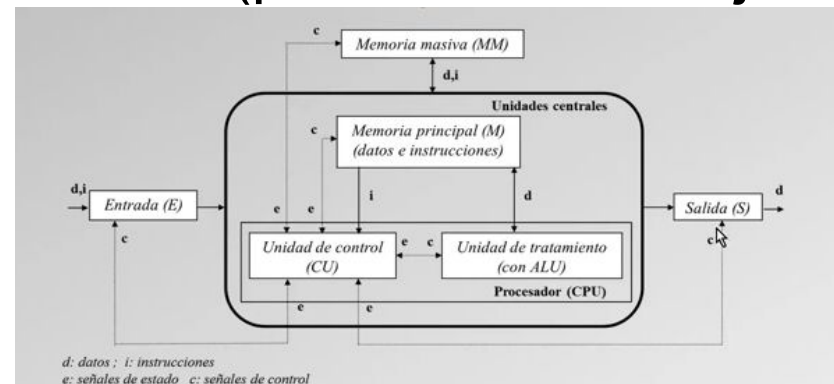
Unidad Aritmético-Lógica (ALU)

- Circuitos electrónicos con los que se realizan las operaciones de tipo aritmético y lógico.
- Registros que almacena temporalmente los datos con los que opera.



Unidad de control

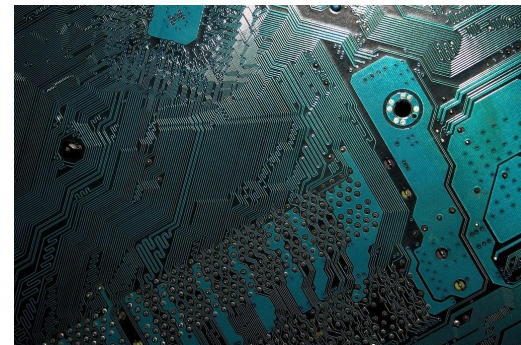
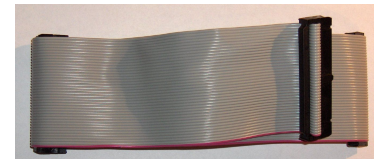
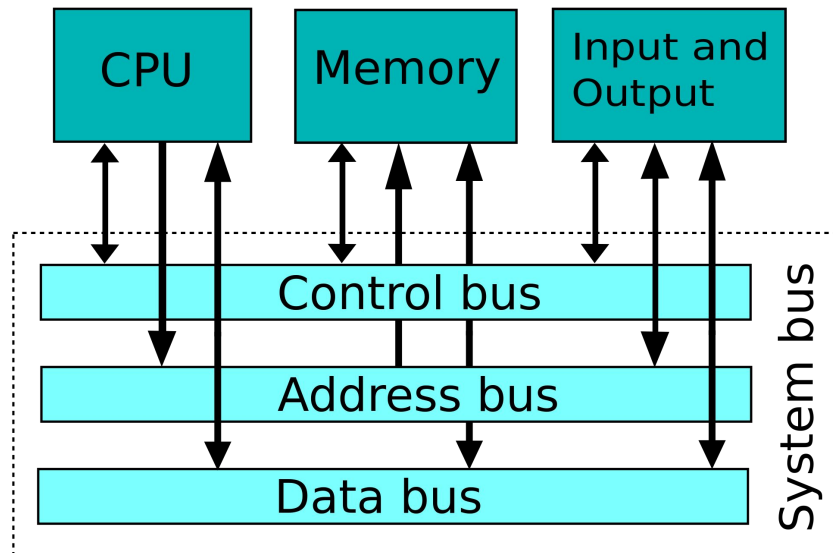
- Detecta señales de estado procedentes de las distintas unidades.
- Capta de memoria las instrucciones máquina del programa de manera secuencial.
- Genera señales de control dirigidas a todas las unidades monitorizando las operaciones que implican la ejecución de la instrucción.
- Contiene un reloj que sincroniza todas las operaciones elementales de la computadora (periodo de reloj → tiempo de ciclo).

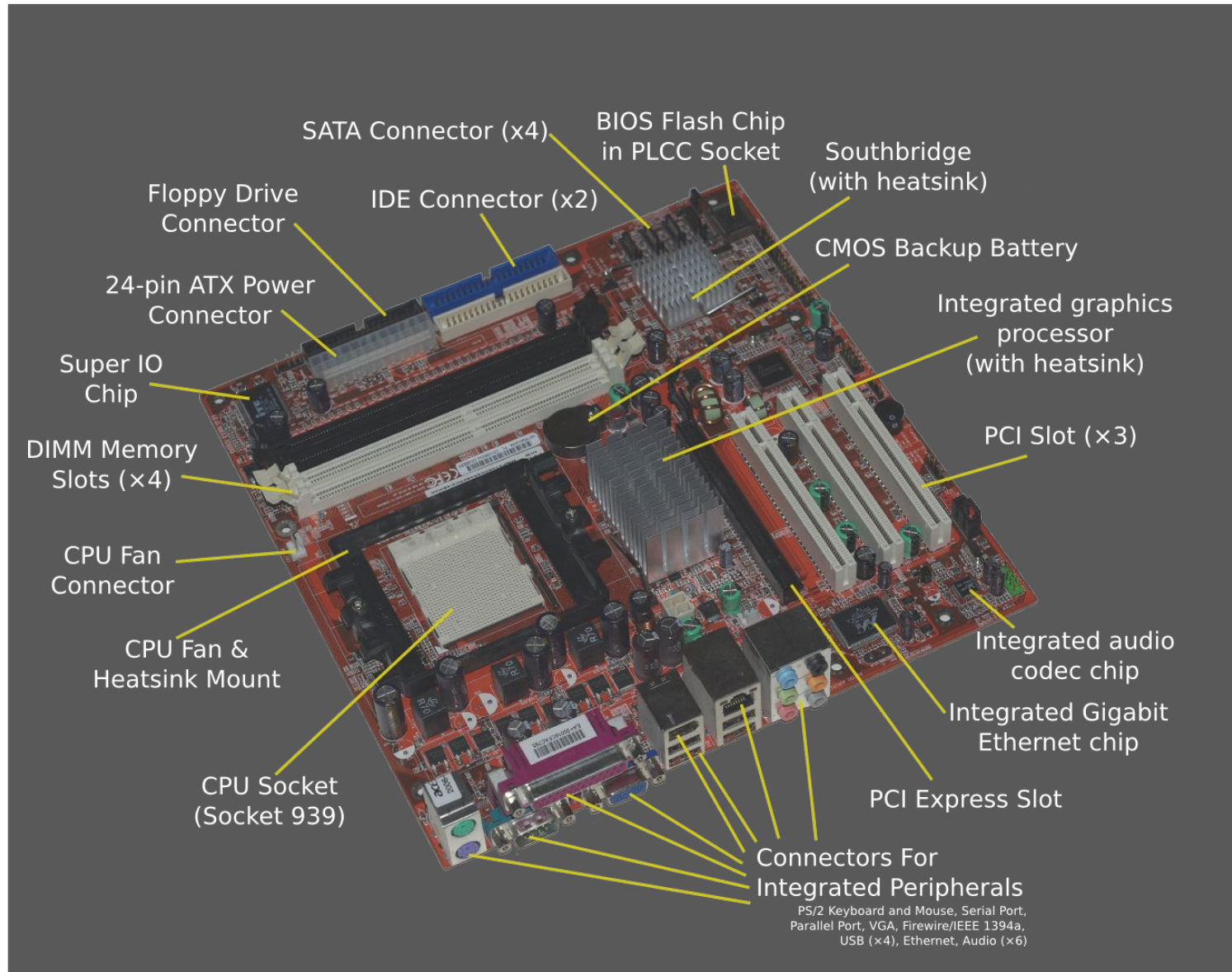


Bus

Las distintas unidades se interconectan mediante conexiones eléctricas: BUS.

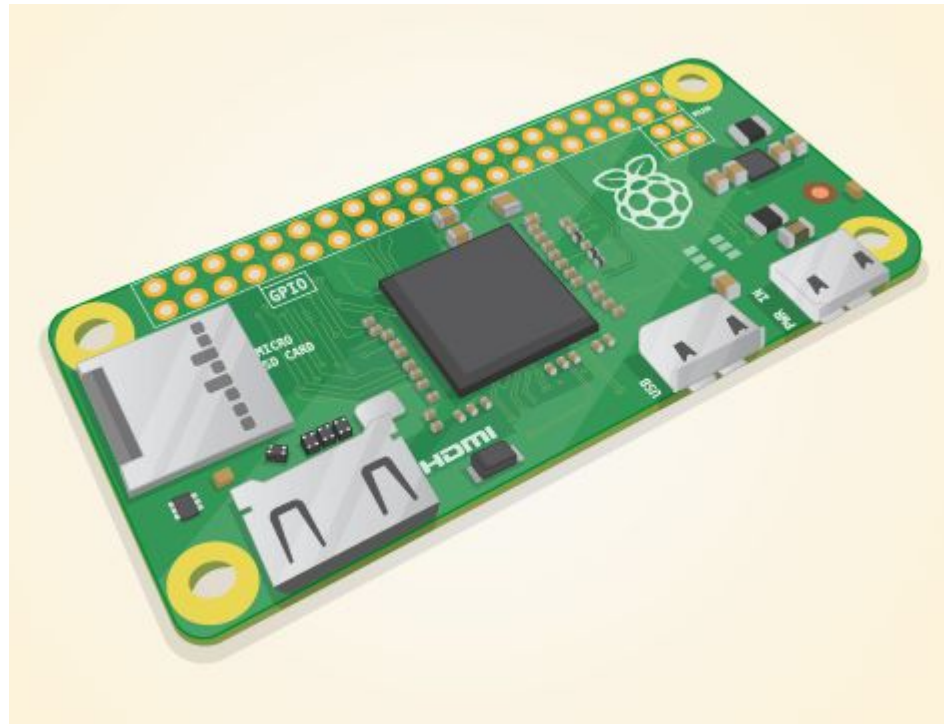
- Serie: Una línea conductora.
- Paralelo (varios hilos conductores en paralelo: 8, 16, 32, 64, 128).



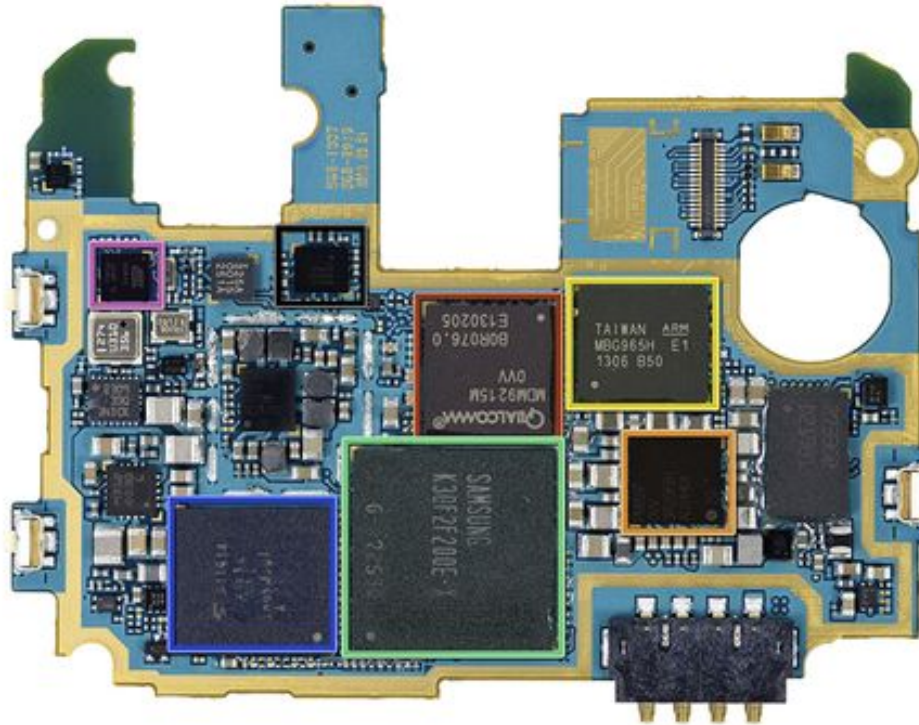


Raspberry Pi Zero

- 1Ghz, Single-core CPU.
- 512MB RAM.
- Mini HDMI and USB On-The-Go ports.
- Micro USB power.
- HAT-compatible 40-pin header.
- Composite video and reset headers.



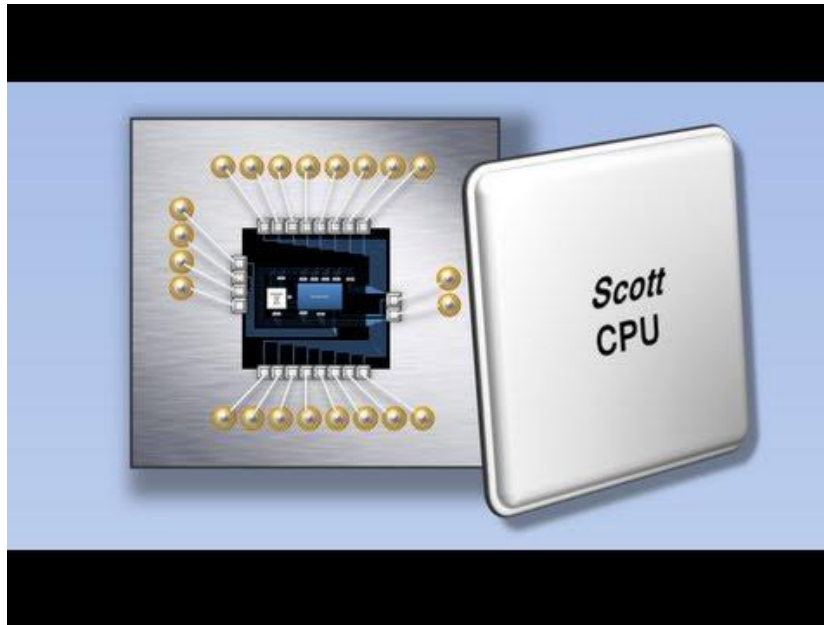
Smartphone



<http://gizmodo.com/5995449/samsung-galaxy-s4-teardown-easy-to-fix-by-design>

1. Fundamentos de Informática

¿Cómo funciona el procesador de un ordenador?



https://www.youtube.com/watch?v=cNN_tTXABUA

1.3. Prestaciones de un ordenador

- Memoria
- Velocidad del procesador

Memoria

- Capacidad: mide en Bytes.
 - Principal (del orden de GB).
 - Masiva (del orden de Gb o TB).
- Tiempo de Acceso (de lectura y de escritura):
 - Memoria Principal (del orden de nanos: 10^{-9} s.).
 - Disco magnético (del orden de micros: 10^{-6} s.).



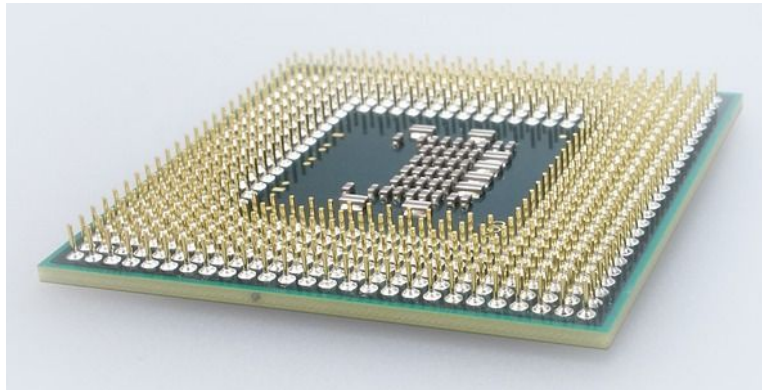
Disco Duro SSD - Unidad de Estado Sólido:

- Más rápidos y robustos

Velocidad del procesador

- Frecuencia de reloj (MHz o GHz).
- MIPS: millones de instrucciones máquina por segundo.

$$\text{MIPS} = \text{NI} / (\text{tiempo ejecución} * 10^6)$$



1.4. Tipos de Computadoras

- Se pueden clasificar según diferentes criterios:
 - Paralelismo.
 - Generalidad de uso.
 - Potencia de procesamiento, ...



Tipos de computadoras según la generalidad de uso

- Computadores de **uso general**.
- Computadores de **uso especializado**, son los que más abundan. Computadores embebidos o empotrados:
 - Hogar: lavadora, lavavajillas, neveras, TV, ...
 - Coche: carputers
 - “Weareables”: pulseras, gafas, ...
 - Smartphone, ..



Tipos de computadoras según su potencia

- Supercomputadores
- Servidores
- Computadores personales
- Computadores móviles: tabletas, teléfonos inteligentes, ...

Supercomputadores

HPC: **Computación de Altas Prestaciones.**

Supercomputador o superordenador:
“es aquel con capacidades de cálculo muy superiores a las computadoras comunes y de escritorio y que son usadas con fines específicos”.

“conjunto de poderosos ordenadores unidos entre sí para aumentar su potencia de trabajo y rendimiento”.

MareNostrum	
	
Supercomputador MareNostrum	
Ubicación	BSC  Barcelona,  España
Instalación	2004
Características	
Arquitectura	AMD64 (Intel Xeon E5 2670)
Nº procesadores	6.112 (48.896 núcleos)
Rendimiento	1,1 PetaFLOPS ¹
Memoria	95,5 TiB
Sistema operativo	GNU/Linux (SUSE)

Supercomputadores

BSC-CNS: Supercomputación y eCiencia



<https://www.youtube.com/watch?v=b5bQdTL0wAg>

Servidor

Es un ordenador o sistema informático conectado a una red y que da servicio compartido a otras máquinas, ordenadores o usuarios (llamadas clientes). Tipos:

- **Servidores de aplicaciones:** los usuarios pueden ejecutar programas, acceder a servicios remotos, ...
- **Servidores de archivos:** unidad NAS
- **Servidores de impresión**
- **Servidor Web, ...**



1.5. Software de una Computadora

Soporte lógico o **Software** de un ordenador es el conjunto de programas ejecutables por el ordenador o sistema informático:

- Sistemas operativos.
- Aplicaciones.
- Firmware.
- Apps, ...



Clasificación del software

- ***Software de Control***
- ***Utilidades***
- ***Aplicaciones***



1.5.1. Software de Control

- Módulos que controlan el funcionamiento de los programas que se ejecutan, y administran los recursos hardware, facilitando el uso del computador de la forma más eficiente posible → **Sistema Operativo**

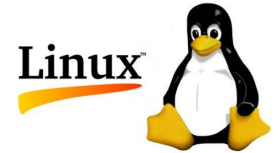


El sistema operativo

“Es un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación” (Wikipedia).

Es un programa de control que tiene como objetivo facilitar el uso del ordenador y conseguir que este se utilice eficientemente.

Es un programa que siempre se está ejecutando y que facilita el trabajo al resto de programas.



El sistema operativo

- **Gestión de procesos:** permite que varios programas se ejecuten a la vez.
- **Abstracción hardware:** trabaja con los diferentes modelos de hardware (p.ej: distintos discos duros) de forma que los programas no tengan que preocuparse de los detalles. Cada fabricante programa un “driver”.
- **Gestión de la Entrada/Salida:** permite leer y escribir información en los dispositivos de E/S. Incluye la gestión de archivos.
- **Gestión de memoria:** permite que los programas usen la memoria RAM sin notar que hay otros programas usándola (piensan que tienen toda la memoria para ellos).

1.5.2. Utilidades

- Programas genéricos de servicio que amplían la funcionalidades ofrecidas por el sistema operativo. Ejemplos: compresores, copias de seguridad, antivirus, ...



- Herramientas para construir/desarrollar aplicaciones. Ejemplos: editores de código, compiladores, IDEs, ...



1.5.3. Aplicaciones

- Navegadores: Firefox, Google Chrome, Safari, ...
- Ofimática: Microsoft Office, Open Office, Libre Office,...
- Imágenes: Photoshop, Gimp, ...
- Vídeo, música, juegos, ...



2.1. Instrucciones y Programas

Instrucción: conjunto de símbolos que representa una orden de operación o tratamiento para el ordenador. Se realizan con datos.

Programa: conjunto ordenado de instrucciones que se dan al computador indicando las operaciones o tareas que se quiere que realice.

Lenguaje de programación: símbolos y reglas para construir o redactar programas (sintácticas y semánticas). Es un lenguaje formal para especificar **algoritmos** que puedan ser ejecutados en una computadora (o sistema informático).

2.2. Algoritmos

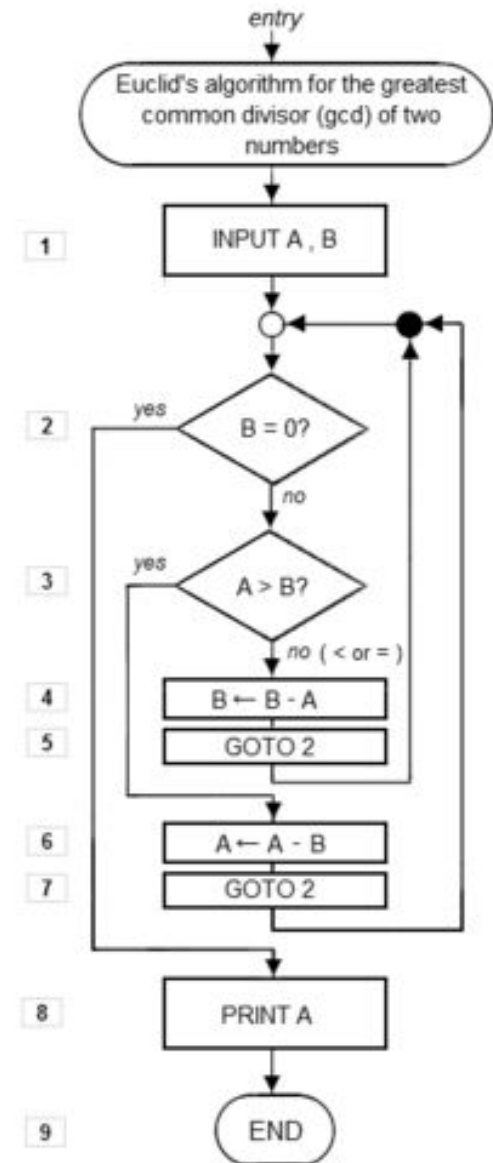
Algoritmo: concepto matemático y de ciencias de la computación) define un procedimiento para llevar a cabo una tarea (Mohammed al-KhoWârizmi, s. IX).

Los “programas de ordenador” o software están formados por algoritmos ejecutables.

Antes de escribir el programa, debemos pensar el algoritmo.

Debe describir:

- Entrada.
- Proceso.
- Salida.



Algoritmos

Método para resolver un problema mediante pasos precisos, definidos y finitos.

- **Preciso**: indica orden de realización en cada paso.
- **Definido**: si se sigue dos veces, se obtiene el mismo resultado cada vez.
- **Finito**: tiene un número determinado de pasos.

Algoritmos

- Escritura de un algoritmo implica, generalmente, su descomposición en pasos más simples: refinamiento sucesivo cada acción se puede descomponer en otras más simples.
- Aspectos a considerar:
 - Palabras reservadas: describen estructuras de control fundamentales y procesos de toma de decisión.
 - Indentación: facilidad de interpretación posterior del algoritmo.
- Se pueden representar por fórmulas, diagramas de flujo y pseudocódigo.

2. Conceptos Básicos de Programación

Algoritmo para resolver cubo de Rubik en 1'



2.3. Lenguajes de programación

Lenguaje máquina: es el único que entienden el procesador del ordenador (CPU). Instrucciones en código binario. Depende del modelo del procesador. Repertorio de instrucciones muy reducido. Tarea de programación compleja.

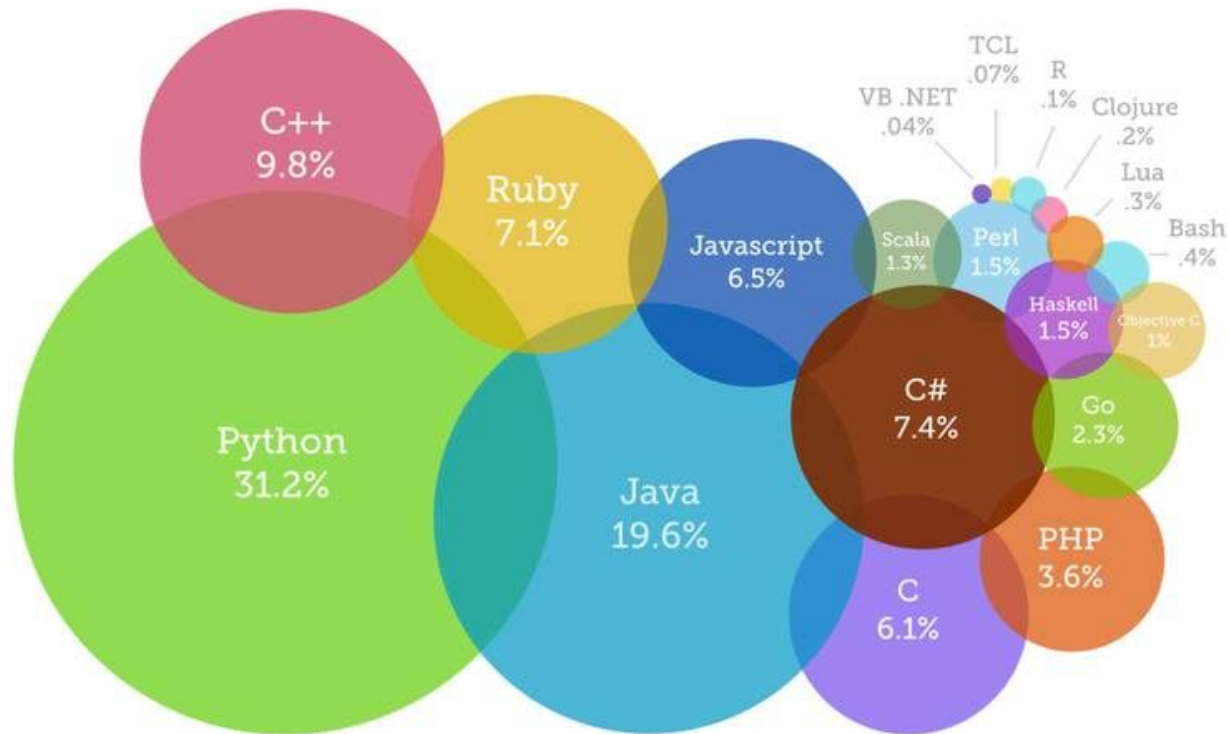
Lenguajes de alto nivel: contruidos para facilitar la tarea de programación. No dependen de la computadora específica.

Ejemplos: Fortran, Basic, Cobol, Pascal, Ada, C, C#, Java, Python, ...

2. Conceptos Básicos de Programación

Lenguajes de programación

Most Popular Coding Languages of 2015



@codeeval

<code*eval>

www.codeeval.com

Traductor

- Se encarga de traducir un programa escrito en un lenguaje de programación de alto nivel a un lenguaje que entienda el ordenador (lenguaje máquina).
- Tipos:
 - **Compiladores**: traducen programa fuente de forma global y generan programa objeto.
 - **Intérpretes**: analizan, traducen y ejecutan una a una las instrucciones del programa fuente. No generan programa objeto.

Paradigmas/Lenguajes de Programación

- **Procedural**: divide programa en partes más pequeñas o subprogramas (subrutinas, funciones, procedimientos) que se invocan entre sí. Ejemplos: C, Pascal, modula,...
- **Orientado a objetos**: utilizan como abstracción el concepto de clase. Un programa está formado por un conjunto de objetos, instancias de una clase, que llevan a cabo acciones y se comunican con otros objetos utilizando mensajes. Ejemplos: C++, Java, ...
- **Funcional**: programación en base a definición de funciones, que se invocan de manera recursiva. Ejemplo: Lisp.
- **Lógica**: programación basada en **representación del conocimiento y lógica de predicados**. Ejemplo: Prolog.

4. Metodología

Resolución de programa con computadora implica escribir programa y ejecutarlo.

Fases de resolución:

- **Análisis del problema:** se consideran requisitos iniciales del cliente.
- **Diseño del algoritmo:** diseño de solución que conduce a algoritmo para resolver problema.
- **Codificación:** empleando lenguaje de programación elegido.
- **Ejecución, Verificación, Depuración:** comprobación y eliminación de errores (bugs).
- **Mantenimiento:** actualización y modificación si es necesario.
- **Documentación:** escritura de las diferentes fases del ciclo de vida del software.

Análisis del problema

- Requiere definición clara de lo que debe hacer el programa y el resultado deseado.
- Respuestas a:
 - ¿Qué entradas son necesarias?
 - ¿Cuál es la salida deseada?
 - ¿Qué método produce la salida deseada?
 - ¿Qué requisitos adicionales o restricciones de la solución existen?

Diseño del algoritmo

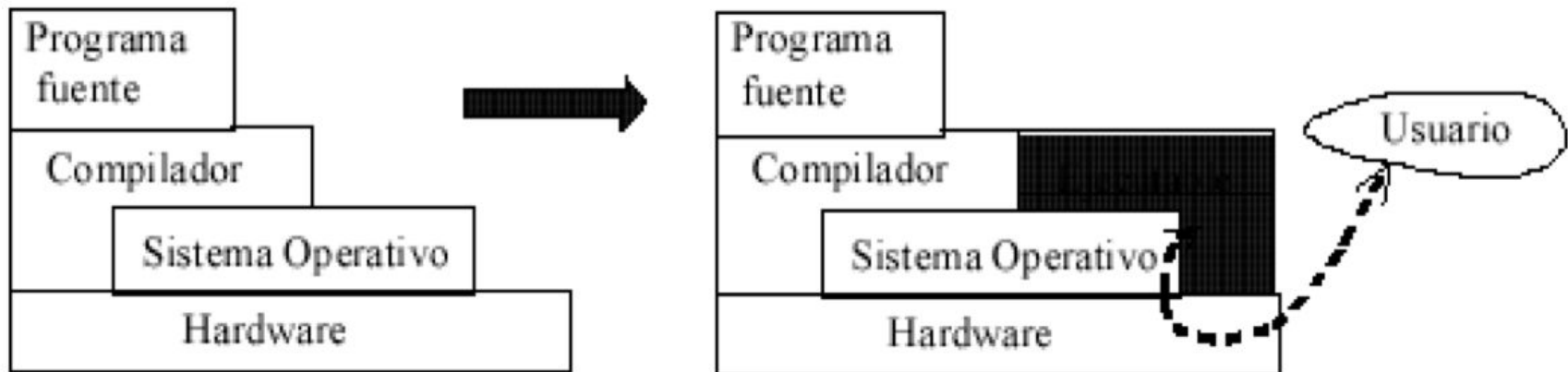
- Conduce a diseño detallado en forma de algoritmo.
- Se determina cómo hace el programa la tarea solicitada.
- Varios métodos: divide y vencerás: dividir problema inicial en otros más pequeños, que son divididos a su vez:
DISEÑO MODULAR.
- **MÓDULO:** subprograma que tiene un único punto de entrada y un único punto de salida.
- Programa bien diseñado consta de:
 - Programa principal (módulo de nivel más alto).
 - Subprogramas (módulos de nivel más bajo).

Codificación

- Escritura en lenguaje de programación de la representación del algoritmo desarrollado.
- Lenguaje de programación: conjunto de símbolos y reglas para combinarlos que se usan para expresar algoritmos.
- Posee:
 - **Léxico**: ¿cuál es el conjunto de símbolos permitidos?
 - **Sintaxis**: ¿cómo se realizan las construcciones del lenguaje?
 - **Semántica**: ¿cuál es el significado de cada construcción concreta?

Compilación y ejecución

- Se traduce programa fuente escrito en lenguaje de alto nivel, en programa objeto, escrito en lenguaje máquina.
- Programa objeto se puede almacenar como archivo en memoria masiva.
- Una vez traducido, ejecución no depende de compilador.



Documentación y mantenimiento

- **Documentación:** descripciones de los pasos seguidos en el proceso de resolución del problema planteado.
- Una buena documentación facilita lectura y modificación posterior del programa.
- Puede ser:
 - **Interna:** comentarios.
 - **Externa:** análisis, diagramas de flujo, pseudocódigo,...
- **Mantenimiento:** modificaciones posteriores del programa. Tras ellas, se debe actualizar la documentación.
- Versiones corresponden a cambios: 1.0, 2.0,...

2.5. Pseudocódigo y Diagramas de Flujo

- Representación de algoritmos mediante métodos que permitan independizarlos del lenguaje de programación se podrán codificar en cualquier lenguaje.
- Métodos más usuales:
 - Diagramas de flujo.
 - Pseudocódigo.

Pseudocódigo

- Lenguaje de especificación de algoritmos.
- No puede ser ejecutado por una computadora.
- Debe producirse posteriormente a lenguaje de programación.
- Fácil de modificar.
- Existe terminología en inglés y en español.

Pseudocódigo

- Ejemplo:

Calcular la superficie de un triángulo.

INICIO

LEER (b, a)

$s \leftarrow n * a / 2$

ESCRIBIR (s)




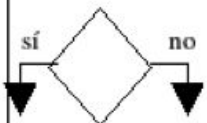





FIN

Diagrama de flujo

- Flowchart.
- Emplea una serie de símbolos (cajas) estándar.
- Tiene los pasos del algoritmo escritos en las cajas unidas por flechas (líneas de flujo).

2. Conceptos Básicos de Programación

Diagrama de flujo

Símbolo	Función
	Terminal. Representa el comienzo o el fin de un programa.
	Entrada / Salida. Indica una introducción de datos desde un dispositivo externo (por defecto, el teclado) o una salida de datos hacia algún dispositivo externo (por defecto, la pantalla)
	Proceso. Representa cualquier operación que se lleve a cabo con los datos del problema.
	Condición. Señala una bifurcación del flujo de instrucciones. La bifurcación está siempre controlada por una operación relacional llamada <i>condición</i> , cuyo resultado puede ser "verdadero" o "falso" (o también "sí" o "no"), dependiendo del valor de los datos de la expresión condicional. En función del resultado de dicha expresión, el flujo de ejecución continúa por una u otra rama (pero nunca por las dos a la vez)
	Condición múltiple. Sirve para indicar una bifurcación del flujo en varias ramas, no sólo en una. En este caso, la condición no puede ser booleana, sino entera.
	Conector. Para enlazar un fragmento del diagrama de flujo con otro fragmento situado en la misma página. Se usa cuando el diagrama es muy grande y no puede dibujarse entero de arriba a abajo.
	Conector. Como el anterior, pero para conectar un fragmento del diagrama con otro fragmento situado en una página diferente.
	Dirección del flujo. Indica el orden de ejecución de los pasos del algoritmo.
	Subrutina. Llamada a un subproceso o módulo independiente (ver apartado de "Programación Modular")

2. Conceptos Básicos de Programación

Diagrama de flujo




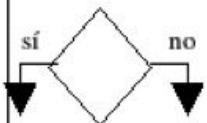





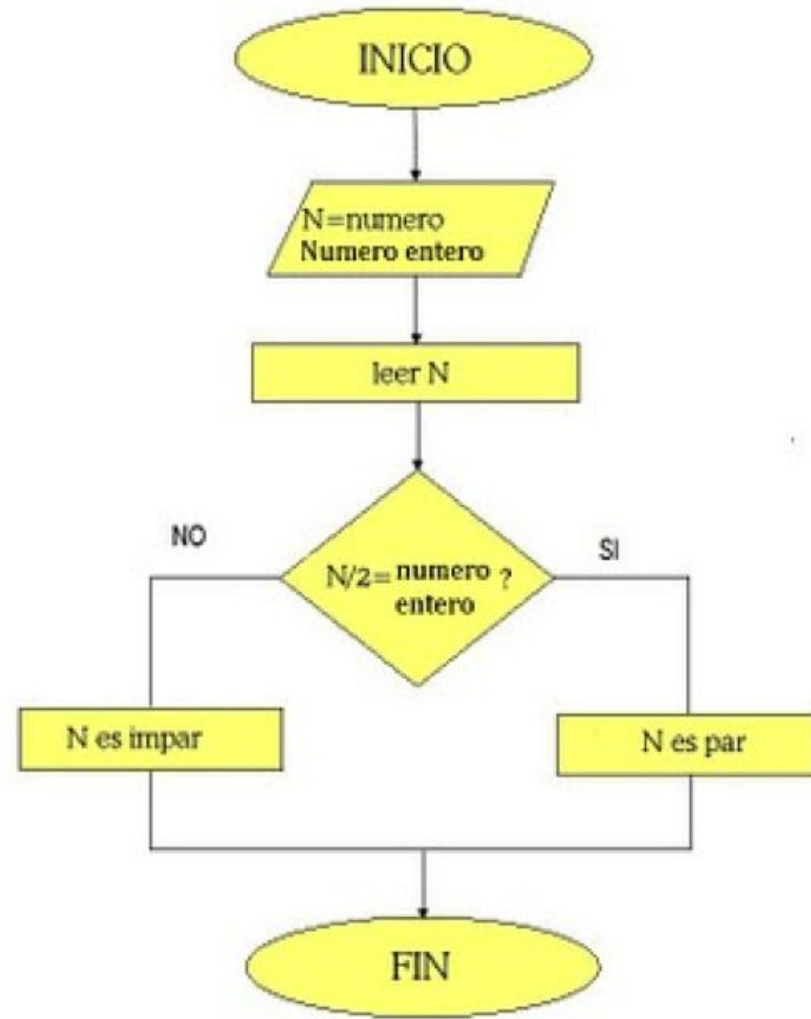
Símbolo	Función
	Terminal. Representa el comienzo o el fin de un programa.
	Entrada / Salida. Indica una introducción de datos desde un dispositivo externo (por defecto, el teclado) o una salida de datos hacia algún dispositivo externo (por defecto, la pantalla)
	Proceso. Representa cualquier operación que se lleve a cabo con los datos del problema.
	Condición. Señala una bifurcación del flujo de instrucciones. La bifurcación está siempre controlada por una operación relacional llamada <i>condición</i> , cuyo resultado puede ser "verdadero" o "falso" (o también "sí" o "no"), dependiendo del valor de los datos de la expresión condicional. En función del resultado de dicha expresión, el flujo de ejecución continúa por una u otra rama (pero nunca por las dos a la vez)
	Condición múltiple. Sirve para indicar una bifurcación del flujo en varias ramas, no sólo en una. En este caso, la condición no puede ser booleana, sino entera.
	Conector. Para enlazar un fragmento del diagrama de flujo con otro fragmento situado en la misma página. Se usa cuando el diagrama es muy grande y no puede dibujarse entero de arriba a abajo.
	Conector. Como el anterior, pero para conectar un fragmento del diagrama con otro fragmento situado en una página diferente.
	Dirección del flujo. Indica el orden de ejecución de los pasos del algoritmo.
	Subrutina. Llamada a un subproceso o módulo independiente (ver apartado de "Programación Modular")

Diagrama de flujo

Ejemplo:

Verificar si un número es par.



Referencias

Prieto, A., Lloris, A., & Torres, J. C. (2006). Introducción a la Informática. McGraw-Hill.

Beekman, G., Martín, J.M.D. (2005). Introducción a la informática. Pearson Prentice Hall.

Videoclases de Fundamentos de Informática, Alberto Prieto Espinosa, Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada: http://atc.ugr.es/APrieto_videoclases

La definición de algunos de los términos y algunas de las imágenes de esta presentación han sido extraídos de la Wikipedia: <https://es.wikipedia.org/>