

Tema 1:

Introducción a la informática

Objetivos del tema: este tema pretende proporcionar al alumno conocimientos básicos sobre el funcionamiento de un computador u ordenador. Introducirá por un lado el funcionamiento de la parte física (hardware), y por otro la parte lógica (software).

Si bien el resto de los temas se centrarán sobre la parte lógica, esto es, el desarrollo de software, hardware y software son dos componentes imprescindibles para que un sistema informático preste servicio a sus usuarios y, desde un punto de vista práctico, el uno sin el otro es completamente inservible. Por ello, para desarrollar software de calidad es imprescindible tener unos conocimientos mínimos acerca del hardware y su funcionamiento.



CEU

Índice

Índice	2
1 Definición y origen del término informática	3
2 Breve historia de la informática	4
2.1 Predicciones fallidas en la historia de la informática	10
3 Estructura de un computador	11
3.1 El hardware	12
3.1.1 La CPU	13
3.1.2 La memoria RAM.....	15
3.1.3 Dispositivos de entrada y salida	16
3.1.4 El bus del sistema	17
3.2 El software	18
3.2.1 Software de sistema y de aplicación.....	20
3.2.2 Lenguajes de programación.....	22
3.2.2.1 Lenguaje Máquina.....	22
3.2.2.2 Lenguaje Ensamblador.....	23
3.2.2.3 Lenguajes de más alto nivel	23
3.2.2.4 Compilación de lenguajes de alto nivel.....	24
3.2.2.5 Interpretación de lenguajes de alto nivel.....	27

1 Definición y origen del término informática

La informática surgió de la necesidad de transmitir y tratar información de manera automática. Su propósito inicial era ayudar al hombre en aquellos trabajos rutinarios y repetitivos, generalmente de cálculo y de gestión, donde es frecuente la repetición de tareas.

El término informática se acuñó en 1962 en Francia y procede de las palabras "información automática": INFORMATICA= INFORmación + autoMATICA. En los países anglosajones se conoce más comúnmente como *Computer Science*.

Según la Real Academia de la Lengua, informática es el "conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento automático de la información por medio de computadores electrónicos."

Según la wikipedia "La Informática o computación es la ciencia del tratamiento automático de la información mediante un computador (llamado también ordenador o computadora). Entre las tareas más populares que ha facilitado esta tecnología se encuentran: elaborar documentos, enviar y recibir correo electrónico, dibujar, crear efectos visuales y sonoros, maquetar folletos y libros, manejar la información contable en una empresa, reproducir música, controlar procesos industriales y jugar."

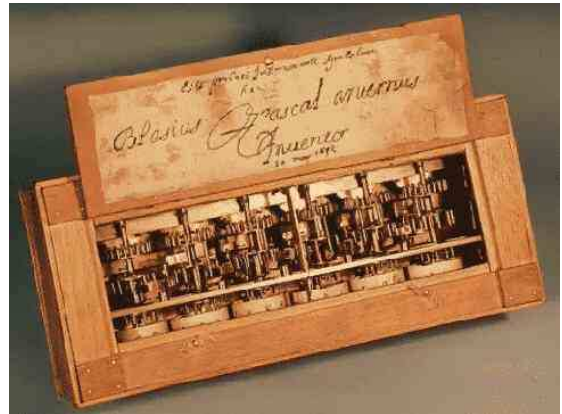
Desde la aparición de las primeras máquinas de cálculo (grandes calculadoras) no se ha parado de investigar para obtener máquinas cada vez más potentes y rápidas. También se trabajó en la consecución de nuevos métodos de procesar datos, novedosas formas de explotación de las máquinas e innovadores modos de compartir los recursos.

2 Breve historia de la informática

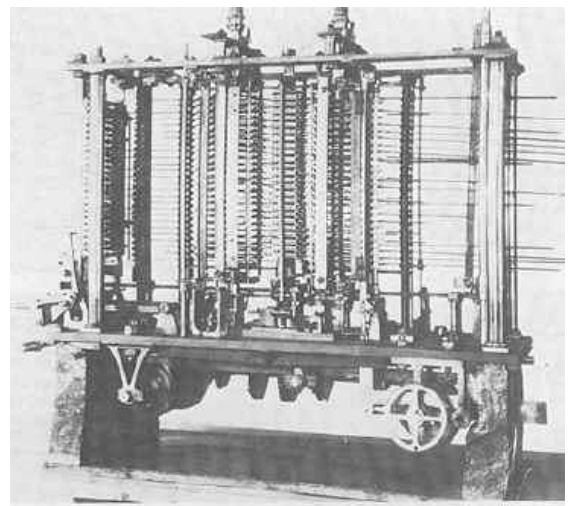


El primer dispositivo manual de cálculo fue el **ábaco**, empleado en Japón desde el año 3.500 a.C. El ábaco es un instrumento compuesto por filas y columnas de cuentas; las columnas representan los números y las filas se usan para calcular los acarreos.

Blaise Pascal (matemático y filósofo francés, 1623-1662) inventó la primera máquina automática de calcular empleando ruedas dentadas. Sumaba y restaba y los resultados aparecían por una ventanilla. Su nombre era **pascalina** o **máquina aritmética de Pascal**.



Charles Babbage (matemático inglés, 1792-1871) diseñó la **máquina de diferencias** que podía resolver funciones (ej. x^2). Esta máquina usaba tecnología mecánica (ruedas y palancas) para realizar los cálculos. Lamentablemente la tecnología disponible en la época no permitió construir la máquina. Babbage, no contento con las capacidades de su invención, diseñó una nueva máquina: la **máquina analítica**, que podía realizar diseño del primer ordenador todas las operaciones matemáticas y podía ser programada con tarjetas de cartón perforado. Por ello, a pesar de que tampoco fue posible construir esta máquina, es considerado el **padre de la Informática**, por haber definido los fundamentos teóricos de las computadoras actuales.



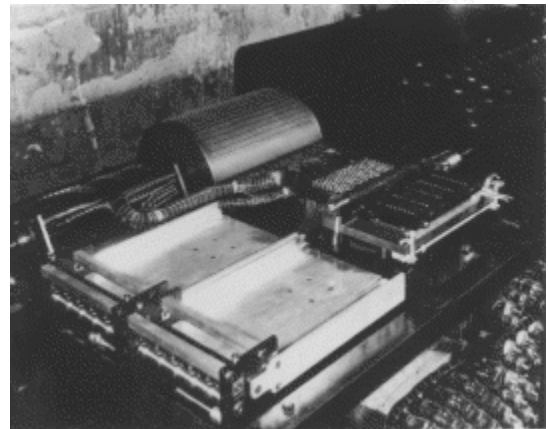
Augusta Ada Byron (1788-1824) escribió el primer programa para la máquina analítica de Babbage. Por ello, aunque el programa nunca se llegó a ejecutar, se la considera **la primera programadora de la historia**. Uno de los primeros lenguajes de programación creados se denominó, en honor a ella, Ada.

George Boole (matemático inglés, 1815-1864) desarrolló el **álgebra de Boole** que permitió a sus sucesores el desarrollo matemático del álgebra binaria y la representación de circuitos de conmutación.

Thomas J. Watson funda IBM en 1924. En breve la empresa se convertiría en el mayor gigante mundial de la industria informática.



En 1938, **John Vincent Atanasoff y Clifford Berry** construyeron la primera máquina de calcular digital. Operaba en binario siguiendo la idea de Babbage. Se llamó **ABC**. Esta máquina no puede considerarse un computador de propósito general ya que sólo era capaz de realizar una única tarea: resolver ecuaciones lineales.

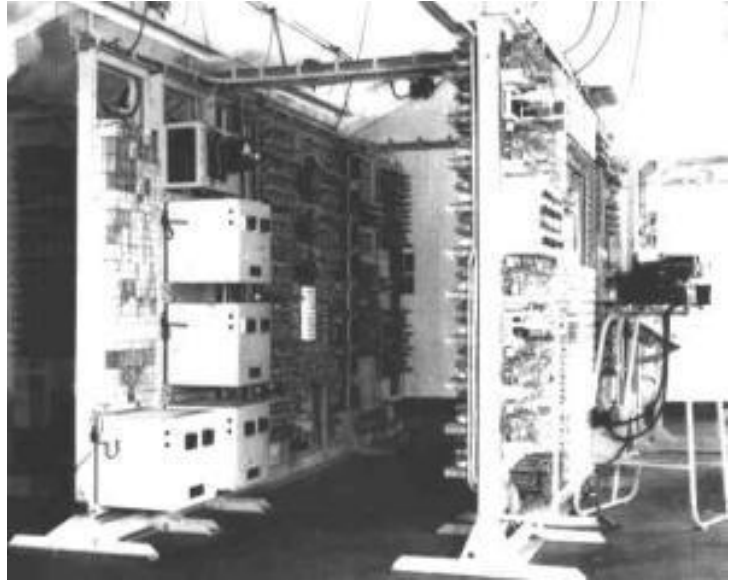


Alan M. Turing (matemático inglés, 1912-1954) desarrolló los fundamentos teóricos de una máquina capaz de resolver todo tipo de problemas con solución algorítmica. Una **máquina de Turing** es una forma de representar un proceso a partir de su descripción. Esto dio lugar a la teoría matemática de la computación y se demostró la existencia de problemas sin solución algorítmica. Turing fue una de las personas clave a la hora de romper las comunicaciones cifradas de los nazis durante la Segunda Guerra



Mundial, para lo cual diseñó la Máquina de Bombas; esta máquina nuevamente no es una máquina de propósito específico y no puede considerarse un ordenador. Por su trabajo durante la Segunda Guerra Mundial la reina de Inglaterra lo nombró caballero.

Howard H. Aiken (1900-1973) siguió la idea de Babbage y construyó la primera computadora electromecánica con relés electromagnéticos, ruedas dentadas y embragues electromecánicos. El proyecto se desarrolló en la Universidad de Harvard y fue patrocinado por la marina de los Estados Unidos e IBM. El computador que construyeron se denominó **Mark-I** y se finalizó en 1944. Tenía 16 metros de largo, 2.6 m. de alto, pesaba 70 toneladas y contaba con más de 800000 metros de cables. Sumaba dos números en menos de un segundo y multiplicaba en tres segundos.

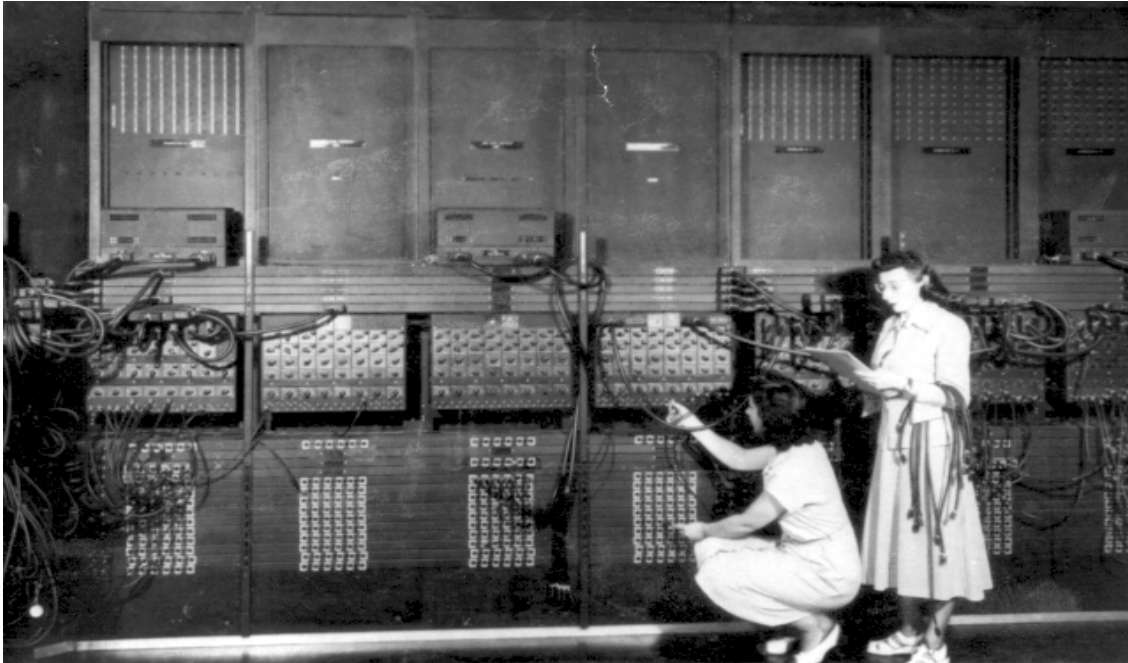


En cuanto la máquina estuvo operativa la marina de los Estados Unidos requisó tanto la máquina como a su inventor para emplearlos en la Segunda Guerra Mundial. Mark-I fue empleado para calcular las trayectorias de los proyectiles que se lanzaban desde los barcos y permitió descifrar las comunicaciones alemanas encriptadas mediante la máquina ENIGMA.

La tecnología electromecánica tenía dos grandes problemas: era lenta y muy difícil de programar. Por ello en la Universidad de Pennsylvania se llevó a cabo un proyecto de investigación en el que se usaron tubos de vacío eléctricos en lugar de relés. En 1945, **John W. Mauchly** y **John Presper Eckert** construyeron el ENIAC (Electronic Numerical Integrator and Computer), la primera computadora electrónica. Para su construcción emplearon tubos de vacío; esta computadora puede considerarse el primer miembro de la primera generación de ordenadores, caracterizados por el uso de tubos de vacío.



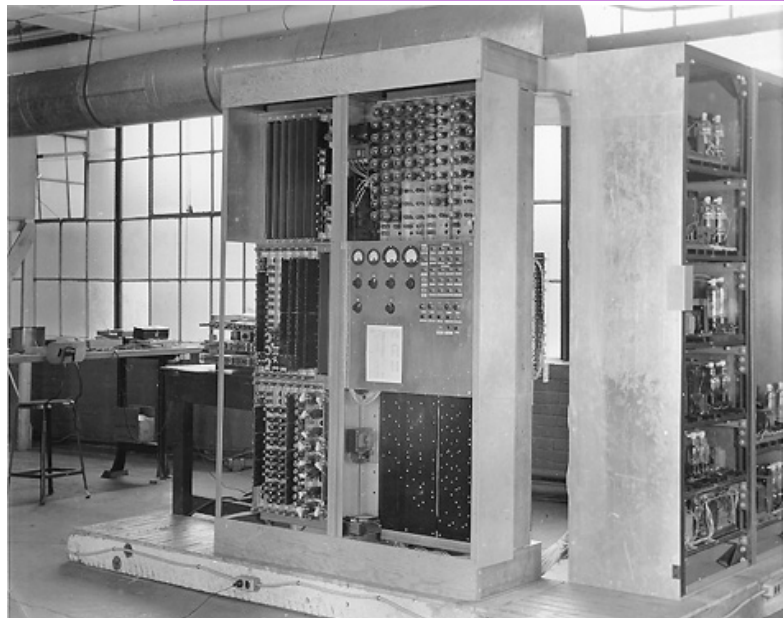
ENIAC podía ser programable. Para ello había que cambiar la posición de un conjunto de cables en un panel de control (las dos mujeres de la imagen se encuentran enfrente al panel de control programando la computadora). ENIAC ocupó toda la planta baja de un edificio de grandes dimensiones y consumía una gran cantidad de electricidad.



En 1944, **John von Neumann** (1903-1957) desarrolló la idea de programa interno y describe

el fundamento teórico de construcción de una computadora electrónica denominada arquitectura de

von Neumann. Su idea consistía en la coexistencia en el tiempo de datos e instrucciones en la computadora y la posibilidad de ser programada, no estando las órdenes cableadas

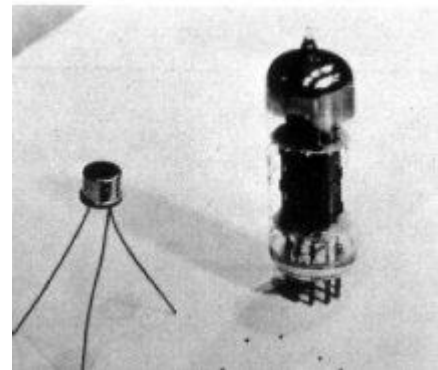


en los circuitos de la máquina. El **EDVAC**, construido en 1952, fue la primera máquina construida siguiendo las ideas de von Neumann. EDVAC pesaba aproximadamente 7850 kg, tenía una superficie de 150 m² y estaba organizado en seis partes:

sigue considerandose de la primera generacion

- Unidad de lectura-grabadora, que era la encargada de la lectura, grabación y borrado de las cintas magnéticas.
- Unidad de control, que contenía los botones de operación, las lámparas indicadoras, los interruptores de control y un osciloscopio para el mantenimiento de la computadora.
- Unidad de "reparto", que se encargaba de decodificar las instrucciones, emitía señales de control hacia el resto de unidades y almacenaba la instrucción que se debía ejecutar en ^{texto} cada momento.
- Memoria de alta velocidad, que consistía en dos unidades iguales, cada una contenía 64 líneas de 8 palabras cada una.
- Computadora, la unidad que realizaba las operaciones básicas aritméticas. La unidad aritmética estaba por duplicado, las operaciones se hacían en ambas unidades y se comparaban los resultados, interrumpiéndose la ejecución si no eran idénticos.
- Reloj, que emitía pulsos de reloj a intervalos de 1 μ segundo.

En la década de los 50 se produjo un desarrollo tecnológico que contribuyó enormemente al avance de los computadores: el transistor. Un transistor es un dispositivo electrónico que puede hallarse en dos estados diferentes (conduciendo corriente o no conduciéndola), de bajo consumo y con una transición muy rápida entre ambos estados. La sustitución de los tubos de vacío por transistores permitió construir máquinas más rápidas, con un menor consumo eléctrico, menor disipación de calor y menor tamaño. Estos nuevos computadores se denominan *computadores de segunda generación*.



pequeños chips onde se empaquetaban varios transistores



En la década de los 60 la tecnología electrónica da un paso más gracias al avance de los **circuitos integrados**; estos circuitos permiten **empaquetar varios transistores en un chip**, lo que redujo todavía más el consumo y el espacio que requerirán los computadores y aumentó más sus prestaciones. Este **avance tecnológico** dio

lugar a los **computadores de tercera generación**, que además fueron los primeros en introducir terminales de rayos catódicos y teclados similares a los actuales. Estos computadores tenían toda las características básicas de los actuales, si bien seguía siendo muy grandes, con muy poca memoria, excesivamente caros y requerían de instalaciones especiales, por lo que sólo podían acceder a ellos las empresas más grandes. En la imagen podemos ver uno de los computadores más célebres de la época, los IBM serie 360. Tenían CPUs de 1 MHz, 128 kB de RAM y unos enormes discos duros con una capacidad de unos pocos megabytes.

El progreso de la tecnología integrada permitió introducir **cada vez más transistores dentro de un chip de silicio**: a principios de los años 80 ya era posible introducir toda la **arquitectura Von Neumann dentro de un chip**, surgiendo así el concepto de microprocesador y dando así lugar a los computadores de **cuarta generación**. Estos equipos ocupaban menos tamaño que sus predecesores, consumía menos potencia eléctrica, ya no requerían de unas instalaciones especiales y su precio se redujo sensiblemente. Este fue el principio de la expansión de los ordenadores por pequeñas empresas, oficinas e incluso hogares, a la vez que echó por tierra la previsión que KEN OLSON, presidente, chairman y fundador de Digital. Equipment Corp., había hecho en 1977: "No existe una sola razón por la cual alguien quisiera tener un ordenador en su casa."

En este momento llegaron los "ordenadores personales" o PCs (Personal Computer), ordenadores muy pequeños y baratos que cabían en una mesa de un despacho. El primero en

salir al mercado, en agosto de 1981, fue el 5150 de IBM, con un microprocesador Intel 8088 (que contaba con 29.000 transistores en un área de 33 mm²) y con un precio de \$5000. El cliente podía elegir el sistema operativo entre CP/M por \$400 o MS-Dos (de una empresa por aquel entonces desconocida: Microsoft) por \$100 (lo que llevó a que se implantara más el sistema operativo de Microsoft).



2.1 Predicciones fallidas en la historia de la informática

El desarrollo de las tecnologías relacionadas con la computación ha sido tan vertiginoso que incluso aquellas personas que, al menos en su tiempo, podían considerarse grandes expertos mundiales del campo erraron totalmente al intentar vislumbrar cómo sería su evolución. Casi de modo continuado, y en muy pocos años, en el campo de la informática se pasa de lo que se considera ciencia ficción a la realidad cotidiana. A continuación recogemos unas cuantas frases que demuestran lo difícil que es intentar hacer predicciones en este campo:

- "Creo que hay un mercado mundial de quizás unos cinco ordenadores". THOMAS WATSON, chairman de IBM, 1943.
- "Los ordenadores del futuro no pesarán más de 1,5 toneladas". POPULAR MECHANICS, adelantando el vertiginoso avance científico, 1949.
- "He viajado por este país de arriba a abajo, hablado con los ejecutivos más importantes y con los mejores técnicos, y os puedo asegurar que el proceso automático de datos es una chapuza que no va a durar mas de un año". El editor de libros sobre empresa de Prentice Hall, 1957.
- "Pero... ¿para que nos van a servir?". Ingeniero de la Advanced Computing Systems Division de IBM, 1968, hablando de los microchips.

- "No existe una sola razón por la cual alguien quisiera tener un ordenador en su casa". KEN OLSON, presidente, chairman y fundador de Digital. Equipment Corp. 1977.
- "640K debería ser suficiente para cualquiera." BILL GATES, 1981
- "Nadie quiere ver video en un iPod", STEVE JOBS, 2003.

3 Estructura de un computador

ordenador: dispositivo electrónico que procesa inf. para obtener ciertos resultados

Podemos definir **computador u ordenador** como 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 humana y bajo el control de un programa de instrucciones previamente almacenado en la propia computadora.

En la definición se puede ver que un ordenador consta de dos partes bien diferenciadas: por un lado la parte física, la máquina que da el soporte a las operaciones que se realizarán en ella (el hardware) y por otro lado la parte lógica, el programa que determina el conjunto de operaciones que debe de ejecutar el hardware (el software). Definamos ambos elementos:

Hardware: elemento físico de un computador. Incluye todos los materiales como la propia computadora, los dispositivos externos, los cables, los soportes de la información, etc.

Software: conjunto de elementos lógicos necesarios para que los computadores puedan realizar las tareas encomendadas. Dota al equipo físico de capacidad para prestar servicios a los usuarios.

El hardware es completamente inútil sin el software, ya que no permite realizar ninguna operación. En un principio, el hardware tenía más peso específico que el software: había grandes y caras computadoras dotadas de pocos y pequeños programas. Hoy en día el software tiene mayor peso específico que el hardware.

En ocasiones el hardware lleva "empotrado" software; que se denomina **Firmware**. Éste puede definirse como el software que las computadoras tienen pregrabado desde su fabricación y que suele estar en memorias de sólo lectura (ROM- *Read Only Memory*).

acepta datos de entrada

procesa esos datos realizando ciertas acciones (operaciones lógicas y analógicas)

proporciona datos de salida

Los ordenadores son máquinas que realizan lo que se denomina **tratamiento de la información**, y los programas determinan el tratamiento que debe de realizarse sobre la información. El tratamiento de la información puede estructurarse en:

1. Entrada de datos (se toman los datos del exterior y se envían a la computadora)
2. Almacenamiento de datos
3. Proceso Aritmético-Lógico
4. Salida (proporcionar los resultados a las personas o entidades correspondientes)
5. Almacenamiento de resultados

La gran versatilidad de los ordenadores consiste en que la misma máquina (mismo hardware) puede efectuar tareas muy distintas dependiendo de los programas que ejecute.

3.1 El hardware

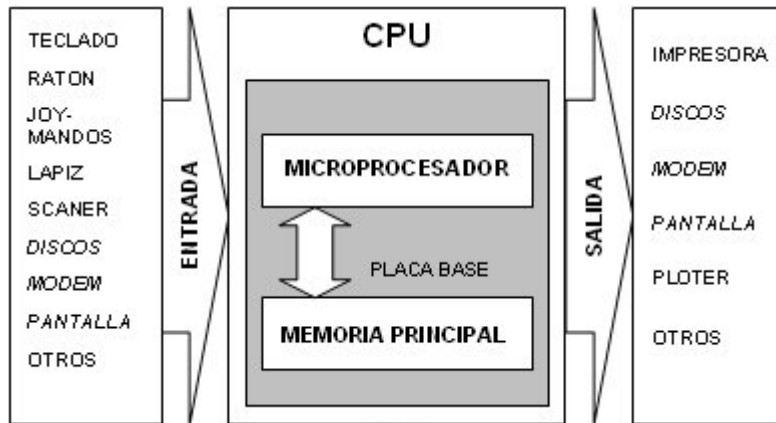
El hardware es la parte física del equipo. Sus componentes principales son:

CPU (*Central Process Unit* - Unidad Central de Proceso) o **procesador**: es un circuito integrado con capacidad de ejecutar instrucciones de un lenguaje elemental llamado *lenguaje máquina* o *código máquina*. Es el elemento principal del ordenador.

Memoria Principal o memoria RAM: Es un espacio de almacenamiento donde se guardan:

- Las instrucciones del programa a ejecutar.
- Las variables donde se almacenan los datos iniciales, intermedios y finales.

Dispositivos de E/S: permiten que el ordenador se comunice con el exterior. Los más comunes son la salida estándar (pantalla) y la entrada estándar (teclado).



Para comunicarse entre sí estos dispositivos emplean el **bus** de datos. Veamos en más detalle cada uno de estos componentes.

3.1.1 La CPU

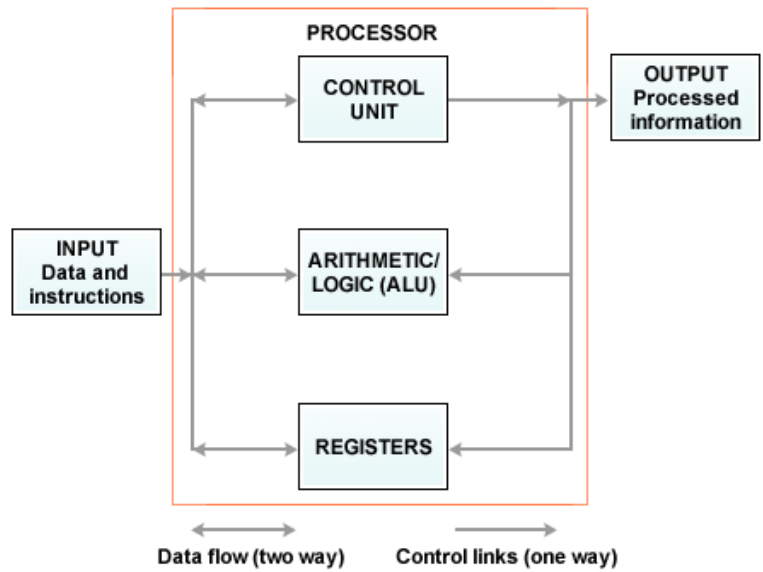
La **CPU** es el elemento principal, el centro neurálgico del ordenador. Su misión consiste en coordinar y realizar todas las operaciones del ordenador. Para ello extrae una a una las instrucciones del programa que está ejecutando, las analiza y emite las órdenes necesarias para su ejecución.

Físicamente está formada por circuitos electrónicos que se encuentran integrados en un chip de silicio. Los distintos circuitos del chip se construyen sobre una oblea de silicio empleando técnicas similares a las que emplea una fotocopidora para crear copias de una hoja. En la actualidad en una CPU hay del orden de decenas de millones de transistores.

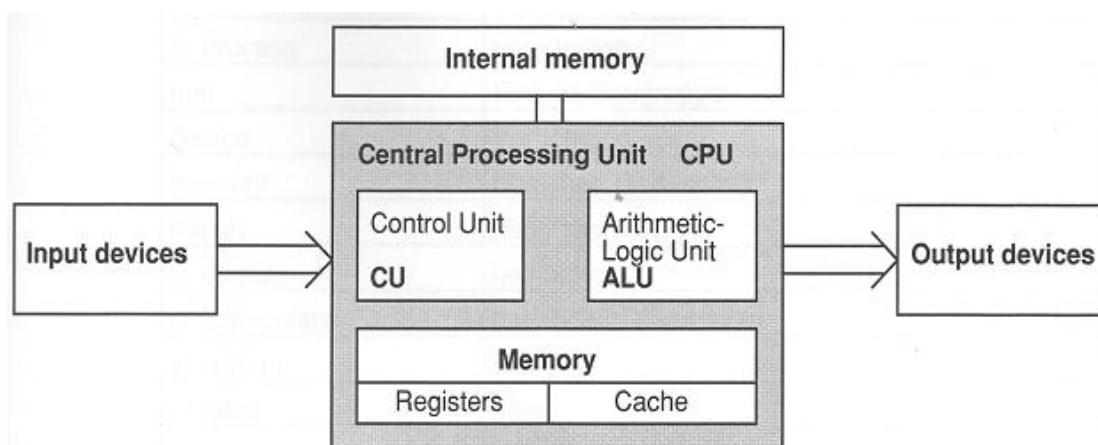
La CPU consta de:

- **Unidad de Control** (*Control Unit*, CU). Se encarga de gobernar al resto de las unidades. Interpreta y ejecuta las instrucciones y controla su secuencia. Contiene un reloj o generador de pulsos que sincroniza todas las operaciones elementales. La frecuencia de reloj (que suele darse en ciclos/segundo, habitualmente MegaHercios, MHz o GigaHercios, GHz) que determina la velocidad de funcionamiento del ordenador. En 1974 la velocidad típica de una CPU era de 2 MHz, hoy en día está entre los 2 y los 3 GHz.

- Unidad Aritmético-lógica** (*Arithmetic-Logic Unit, ALU*). Se encarga de realizar todas las operaciones elementales de tipo aritmético (sumas, restas, productos y divisiones) y lógico (ej. comparaciones). La longitud de la **palabra de CPU** es el número de bits máximo de los datos con los que opera la ALU. En los ordenadores de propósito general las longitudes de palabras típicas hoy en día son de 32 bits y 64 bits. Los procesadores de propósito específico orientados al procesamiento de imágenes (como las CPU de las videoconsolas) tienen longitudes de palabra superiores (128 bits).



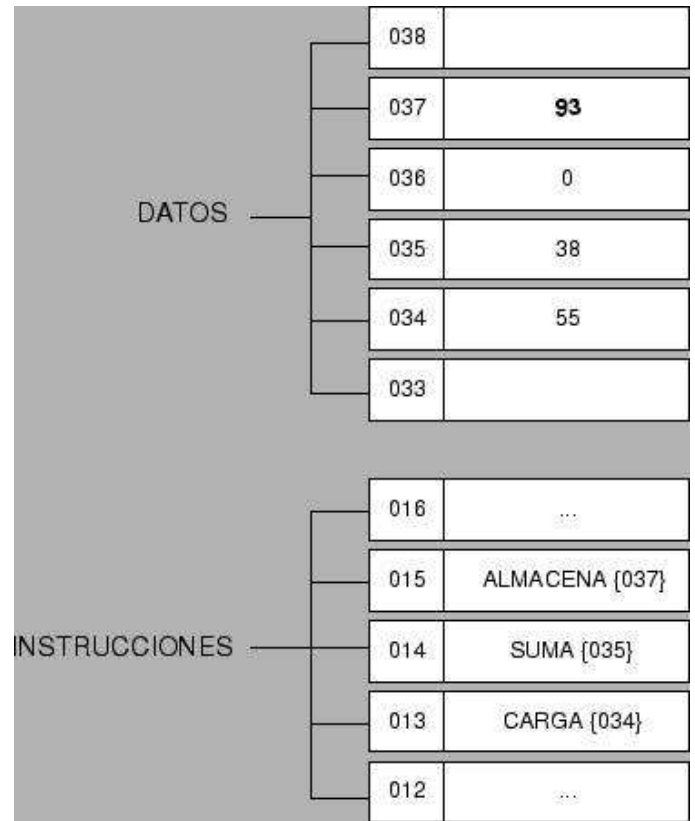
- Banco de registros** (pequeñas unidades de almacenamiento de memoria). Son dispositivos de alta velocidad para almacenar datos o instrucciones temporalmente. Son la zona más alta en la jerarquía de memoria. Hay una serie de registros especiales como el contador de programa, que almacena la dirección de la siguiente instrucción ejecutar, registros que almacenan el estado del ordenador, etc.



Hoy en día, gracias a la gran capacidad de integración de los circuitos de silicio, también suele incluirse cierta cantidad de memoria RAM dentro de la CPU, cuyo tamaño es de unos pocos Megabytes.

3.1.2 La memoria RAM

La **memoria central, memoria principal o memoria RAM** (*Random Access Memory*) se encarga de almacenar los programas y datos necesarios para el funcionamiento del sistema informático. Para que un programa pueda ser ejecutado tanto sus instrucciones como los datos que haya que procesar tienen que estar en esta memoria. Es *volátil*, esto es, se borra al apagar el ordenador ya que requiere de alimentación de corriente continuada para conservar la información que almacena.



Está constituida por multitud de **celdas o posiciones de memoria** idénticas numeradas de forma consecutiva. La numeración de cada celda se denomina **dirección de memoria** y mediante esta dirección se puede acceder de forma directa a cualquiera de sus celdas. Por ello se dice que el acceso a esta memoria es **directo o aleatorio**. La longitud de **palabra de memoria** es la cantidad de información (expresada en bits) que puede introducirse o extraerse de la memoria central de una sola vez. Los tamaños de palabra de memoria típicos actuales son 32 y 64 bits.

Normalmente el tamaño de la memoria se expresa como algún múltiplo de $2^{10}=1024$ bytes, esto es 1K. Actualmente el

		Factor binario
Bytes	B	$2^0 = 1$
KiloBytes	Kb	$2^{10} = 1024$
MegaBytes	Mb	$2^{20} = 1\ 048\ 576$
GigaBytes	Gb	$2^{30} = 1\ 073\ 741\ 824$
TeraBytes	Tb	$2^{40} = 1\ 099\ 511\ 627\ 776$
PetaBytes	Pb	$2^{50} = 1\ 125\ 899\ 906\ 842\ 624$
ExaBytes	Eb	$2^{60} = 1\ 152\ 921\ 504\ 606\ 846\ 976$
ZettaBytes	Zb	$2^{70} = 11\ 805\ 916\ 207\ 174\ 113\ 034\ 24$
YottaBytes	Yb	$2^{80} = 12\ 08\ 925\ 819\ 614\ 629\ 174\ 706\ 176$

tamaño de la memoria suele ascender a cientos de Megabytes ($1 \text{ M} = 2^{10} \text{ K}$) o Gigabyte ($1 \text{ G} = 1024 \text{ M}$). A principios de la década de los 80 la capacidad de almacenamiento típica de un ordenador personal era del orden de unos pocos cientos de Kilobytes.

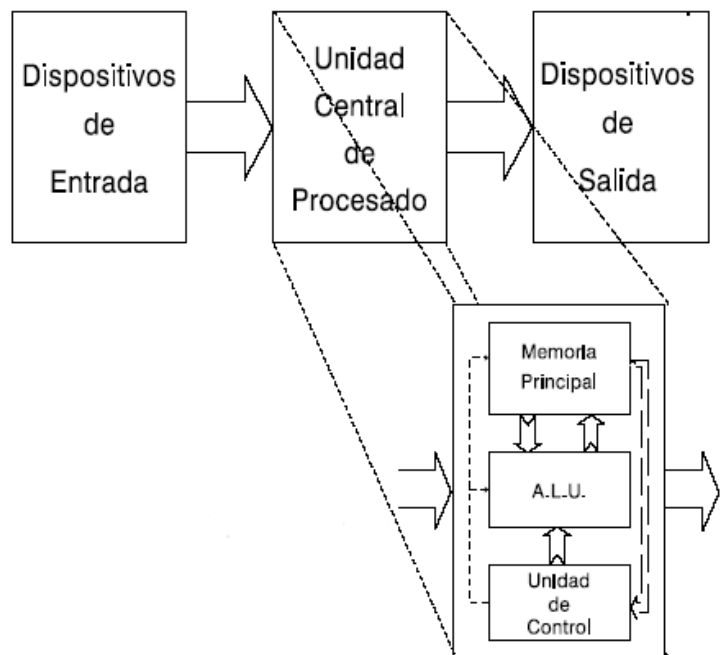
Otro tipo de memoria que merece mención es la memoria **ROM** (*Read Only Memory*). El contenido de esta permanece al desconectar el ordenador. Normalmente contiene instrucciones necesarias para que el hardware del equipo comience a funcionar. Es una memoria donde no se puede escribir nada, sólo se puede leer su contenido.

3.1.3 Dispositivos de entrada y salida

Los dispositivos de entrada y salida (E/S) permiten la comunicación entre el ordenador y el resto del mundo. Actualmente los ordenadores incluyen tres tipos de dispositivos típicos para llevar a cabo las operaciones de entrada y salida:

- *Interfaz con el usuario.* Permiten al ordenador comunicarse directamente con el ser humano.

Los dispositivos de entrada más comunes son el ratón y teclado; otros son lápices ópticos, el micrófono, escáneres, cámaras de vídeo, lectores de códigos de barras, etc. Los dispositivos más típicos de salida son el monitor y la impresora; otros son altavoces, ploters, proyectores, etc.



- *Conexión con los dispositivos de almacenamiento.* Los dispositivos de almacenamiento solventan el problema de la volatilidad de la memoria RAM y proporcionan mayor capacidad de almacenamiento que ésta; el precio a pagar por estas ventajas es una velocidad de acceso mucho menor. Actualmente se usan distintas tecnologías para los dispositivos de almacenamiento: dispositivos de

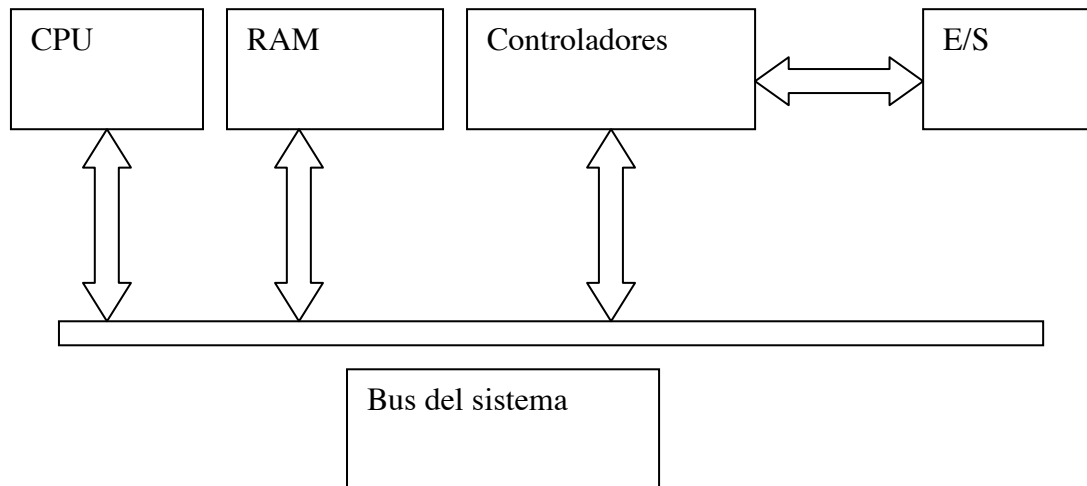
almacenamiento magnéticos (discos duros, diskettes y cintas), ópticos (CDROM y DVD) y microelectrónicos (lápices USB). El **disco duro** constituye el principal almacenamiento del ordenador, siendo el dispositivo que permite un acceso más rápido y un mayor espacio de almacenamiento (también tienen el precio más alto). Es por ello que cuando la memoria RAM no resulta suficiente para almacenar toda la información se vuelca una parte del contenido de ésta al disco duro; a esta información se la denomina **memoria virtual**. El tamaño típico del disco duro actualmente ronda los 1000 Gigabytes.

- *Conexiones a redes.* Hay dos formas básicas de conexión a la red: conexión a red telefónica y conexión a la red de área local. La primera se lleva a cabo conectando al PC un dispositivo denominado módem y la segunda se realiza a través de una tarjeta de red.

3.1.4 El bus del sistema

El bus es el mecanismo de comunicación entre los elementos de un ordenador. Está constituido por múltiples líneas que permiten indicar la dirección de memoria donde están los datos a tratar, transmitir señales de control a las líneas de datos y dirección, y transmitir los propios datos sobre los cuales opera el ordenador. Toda esta información circula en paralelo por el bus, por lo que cuando un dispositivo lo está usando ninguno más puede acceder a él. La mayor parte del tráfico del bus lo causan los accesos de la CPU a la memoria RAM.

Los dispositivos de entrada y salida no se conectan directamente al bus del sistema; se conectan mediante un **controlador**, esto es, un circuito impreso que conoce el funcionamiento del dispositivo de entrada y salida y hace las veces de mediador para el intercambio de información entre el dispositivo y el resto del sistema.

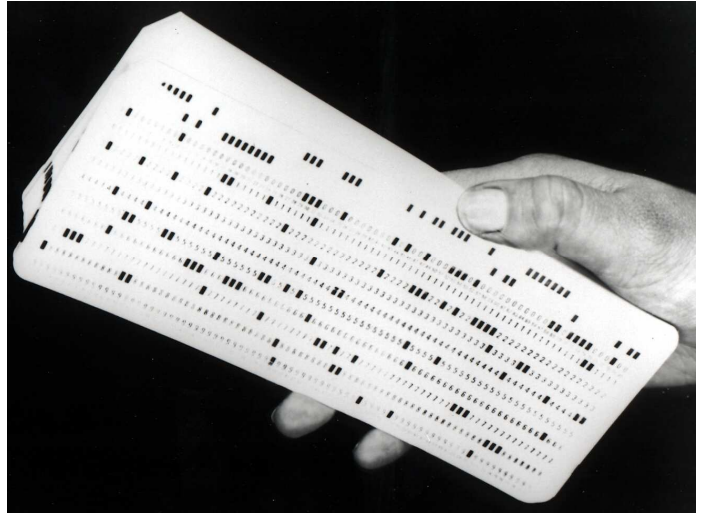


3.2 El software

Los programas o software están compuestos por un conjunto de instrucciones que operan sobre un conjunto de datos de entrada para generar un conjunto de datos de salida. Una **instrucción** es un conjunto de símbolos que representa una orden de operación o tratamiento para la computadora. Podemos definir **programa** como un conjunto de instrucciones que ejecuta un ordenador para realizar un proceso determinado. El concepto de programa está íntimamente ligado al concepto de ordenador, ya que un ordenador es una máquina que ejecuta las instrucciones codificadas en el programa.

El ordenador está compuesto por un conjunto de circuitos electrónicos que tienen una serie de señales eléctricas de entrada y como resultado de su operación genera una serie de señales eléctricas de salida. En función de la intensidad de las señales se codifica la información: si las señales llevan intensidad (o se les ha aplicado un voltaje) se consideran 1s. Si las señales no llevan intensidad se consideran como 0s. Existen dos tipos de señales claramente diferenciadas. Por un lado, las que están orientadas al control operación de los componentes electrónicos del ordenador que, por tanto, sirven para controlar el funcionamiento interno. Las otras señales están ligadas a las instrucciones que debe ejecutar el ordenador y a los datos que se van a procesar.

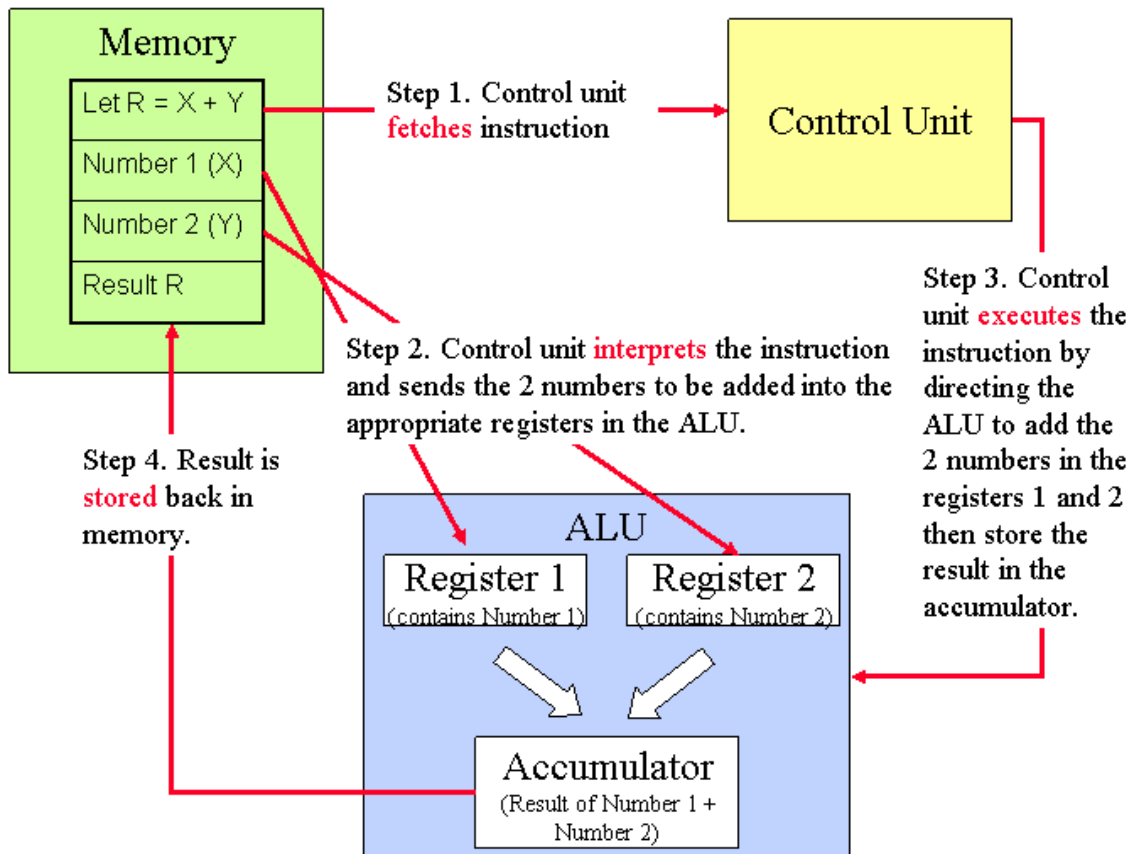
El objetivo de esta asignatura es enseñar al alumno a construir programas que realicen un procesado de cierta información acorde a unas especificaciones dadas. En un principio, la programación de los ordenadores se realizaba mediante tarjetas perforadas donde la ausencia o presencia de agujeros indicaban un 1 o un 0. Afortunadamente, hoy en



día existen formas mucho más simples y elegantes de crear programas: los **lenguajes de programación**, que definen el conjunto de símbolos y reglas para construir o redactar programas.

La CPU es la encargada de ejecutar los programas. Para ello lee de forma ordenada una lista de instrucciones, las interpreta y controla la ejecución de cada una de ellas. Las instrucciones se ejecutan de forma consecutiva una detrás de otra. Para ejecutar cada instrucción la CPU realiza los siguientes pasos:

1. Lee de la memoria la instrucción que tiene que ejecutar y la guarda en un registro en el interior de la CPU.
2. Interpreta la instrucción que acaba de leer (¿qué tipo de instrucción es?), y comprueba si la instrucción necesita utilizar datos que se hallan en la memoria (y no en un registro interno de la CPU). Si es así, determina donde debe ir a buscar estos datos y los trae a la CPU.
3. Ejecuta la instrucción y almacena el resultado de la ejecución en un registro.
4. Almacena los datos nuevamente en la memoria RAM y vuelve al paso 1.

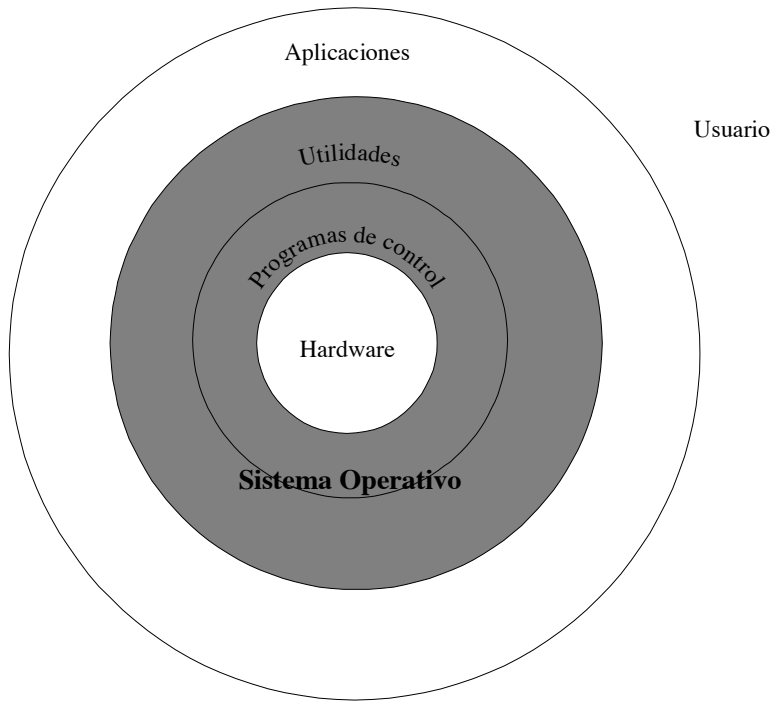


Una **aplicación informática** es un conjunto de uno o varios programas, y su documentación correspondiente, cuyo fin es realizar un determinado trabajo. La documentación permite al usuario de la aplicación comprender su funcionamiento y manejarla. Por **sistema informático** se entiende el conjunto de elementos necesarios (computadoras, terminales, impresoras, etc.) para la realización y explotación de aplicaciones informáticas. Un sistema informático requiere de varios tipos diferentes de software (programas) que veremos a continuación.

3.2.1 Software de sistema y de aplicación

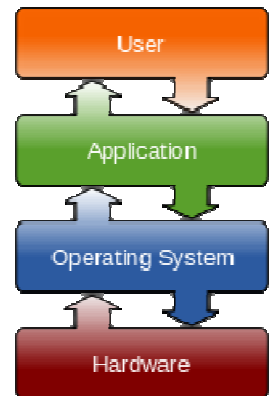
Hay dos tipos diferentes de software: el software de sistema y software de aplicación. El *software de sistema* es aquél que proporciona los mecanismos de gestión del hardware y las utilidades para desarrollar aplicaciones. Sirve como base para desarrollar programas y para que el software de aplicación pueda acceder al hardware, permitiendo así aislar al programador de los detalles de bajo nivel de la máquina e incrementando su productividad.

El **sistema operativo** es, sin duda, el software de sistema más importante de un ordenador. Este software controla y gestiona a los recursos hardware del ordenador, entre ellos la CPU, la memoria RAM y los dispositivos de entrada y salida. El sistema operativo es el único software que puede interactuar directamente con el hardware del ordenador.



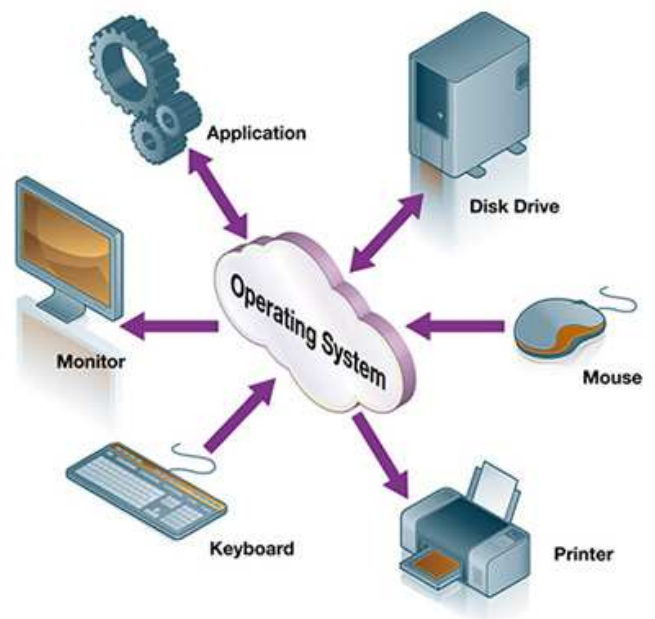
Cualquier otra interacción de cualquier otro software con el

hardware se realiza con la mediación del sistema operativo. Las funciones más importantes del sistema operativo son la **gestión de los programas en ejecución**, permitiendo que varios programas puedan compartir la CPU y los diversos recursos del ordenador; la **asignación de memoria** los programas para que éstos puedan ejecutarse; la **gestión de los controladores de los dispositivos de entrada y salida**; la **gestión del sistema de archivos** proporcionando una organización lógica de los sistemas de almacenamiento a través de volúmenes (o unidades), directorios (o carpetas) y archivos; etc.



Un sistema operativo contiene programas de control, que se ocupan de controlar al equipo físico en todos sus aspectos, y contiene programas de utilidad o aplicaciones, que se encargan de ayudar al usuario en trabajos típicos, como formatear disquetes, manejar ficheros, etc.

Además del sistema operativo, el software de sistema incluye editores para introducir textos y programas en el ordenador, compiladores, intérpretes, etc.



y una serie de herramientas que permiten al programador crear software.

El **software de aplicación o aplicaciones** se compone de programas diseñados para que el usuario interactúe de modo simple con el ordenador con el objeto de realizar una determinada tarea. Es el software que utiliza todo el mundo: procesadores de texto, navegadores web, clientes de correo electrónico, hojas de cálculo, agendas electrónicas etc. Este software es el responsable del éxito de los computadores actuales, ya que proporciona un conjunto de servicios muy atractivos a un usuario que no **tiene necesariamente conocimientos sobre el funcionamiento interno del ordenador ni de los programas.**

3.2.2 Lenguajes de programación

Los lenguajes de programación sirven para **escribir programas que permitan a un ordenador prestar diversos servicios a los usuarios.** Hay tres tipos de lenguajes: **lenguaje máquina, lenguaje ensamblador y lenguajes de alto nivel.**

3.2.2.1 Lenguaje Máquina

Es el lenguaje que entiende directamente el procesador de un ordenador. En la práctica, casi nunca se utiliza para programar porque:

- Requiere un conocer con detalle el diseño del procesador y es distinto para cada modelo.
- Tanto los datos como las instrucciones se representan numéricamente (en binario o hexadecimal). Por ejemplo, 3F2 puede significar "ver si el valor en el registro CX es mayor que 0".
- Las instrucciones son muy elementales: suma, comparación, copia, etc. de números o posiciones de memoria.

Los circuitos electrónicos de la unidad de control de un ordenador sólo pueden interpretar instrucciones en lenguaje máquina. Por tanto, cualquier programa escrito en cualquier otro lenguaje de programación para poder ejecutarse en un ordenador tendrá que traducirse a lenguaje máquina.

3.2.2.2 Lenguaje Ensamblador

Es equivalente al lenguaje máquina, pero las instrucciones se indican con palabras cortas (mnemónicos) en lugar de números. Por ejemplo, la instrucción CMP puede servir para comparar dos valores, ADD para sumar dos valores, etc. A cada instrucción de ensamblador le corresponde una única instrucción de código máquina. La traducción de ensamblador a código máquina la realiza un programa que también se denomina **Ensamblador**.

En la imagen podemos observar a la derecha un programa escrito en código ensamblador y a la izquierda su traducción a código máquina. El programa calcula la suma de los 10 primeros números enteros. Como podemos observar, la programación en lenguaje ensamblador sigue siendo bastante tediosa.

Lenguaje ensamblador	Lenguaje máquina
CAR SUM, 0	1111 0101 0000 0000
CAR i, 0	1111 0111 0000 0000
BUCLE: ADD SUM, i	1110 0101 0000 0000
ADD i, 1	1110 0111 0000 0001
CAR AUX, i	1111 1111 0000 0101
RES AUX, 9	0111 1111 0000 1001
STZ SIGUE	0001 1111 1010 1111
STI BUCLE	0010 1111 1010 0011
SIGUE SAL i	0110 0111 0000 0000

3.2.2.3 Lenguajes de más alto nivel

Los lenguajes de alto nivel son los más utilizados por los programadores. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina y ensamblador. Otra de sus ventajas es que son independientes del procesador. Esto hace que los programas sean *portables* sobre distintos tipos de ordenadores.

Su sintaxis es más fácil de entender y recordar para una persona: do ... while (hacer mientras), open (abrir), y poseen instrucciones potentes (de control de flujo, manejo de ficheros, creación de gráficos), con lo que los programas son más cortos. Ejemplos de estos lenguajes son **Pascal, BASIC, C, C++, Python, COBOL, Java, C#**, etc. Estos lenguajes en la actualidad suman varios cientos sino miles. El motivo de que haya tantos lenguajes es que, habitualmente, cada lenguaje de programación ha sido diseñado para resolver un tipo de

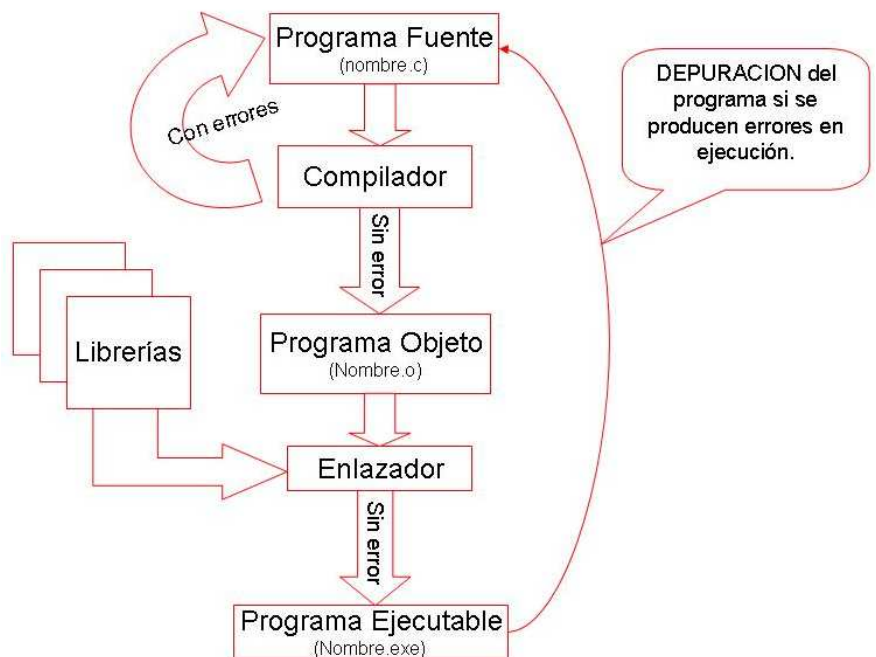
problemas y el abordar desde ese lenguaje de programación problemas diferentes de aquellos para los que fue pensado puede resultar muy tedioso e incluso imposible.

En la figura podemos ver el fragmento de código C que permite calcular el valor de la suma de los primeros 10 valores enteros.

Lenguaje C	Lenguaje ensamblador	Lenguaje máquina
suma=0;	CAR SUM, 0	1111 0101 0000 0000
for (i=0; i<10; i++)	CAR i, 0	1111 0111 0000 0000
suma=suma+i;	BUCLE: ADD SUM, i	1110 0101 0000 0000
printf("%d", i);	ADD i, 1	1110 0111 0000 0001
	CAR AUX, i	1111 1111 0000 0101
	RES AUX, 9	0111 1111 0000 1001
	STZ SIGUE	0001 1111 1010 1111
	STI BUCLE	0010 1111 1010 0011
	SIGUE SAL i	0110 0111 0000 0000

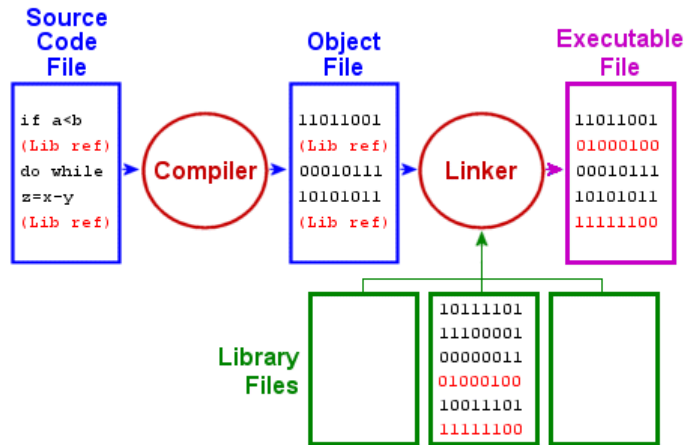
3.2.2.4 Compilación de lenguajes de alto nivel

Como los procesadores sólo "entienden" instrucciones escritas en lenguaje máquina es necesario *traducir* a código máquina los programas escritos en lenguajes de alto nivel. Esta labor de traducción la realizan los **compiladores** y los **intérpretes**. Los compiladores traducen el código fuente generando un programa en lenguaje máquina. Al fichero que genera el compilador se le denomina **fichero objeto**.



Generalmente la compilación produce un programa en código máquina que todavía no es ejecutable, ya que está incompleto. Esto se debe a que casi siempre los programas dependen de recursos externos denominados **librerías**; esto es, ficheros binarios que se han obtenido

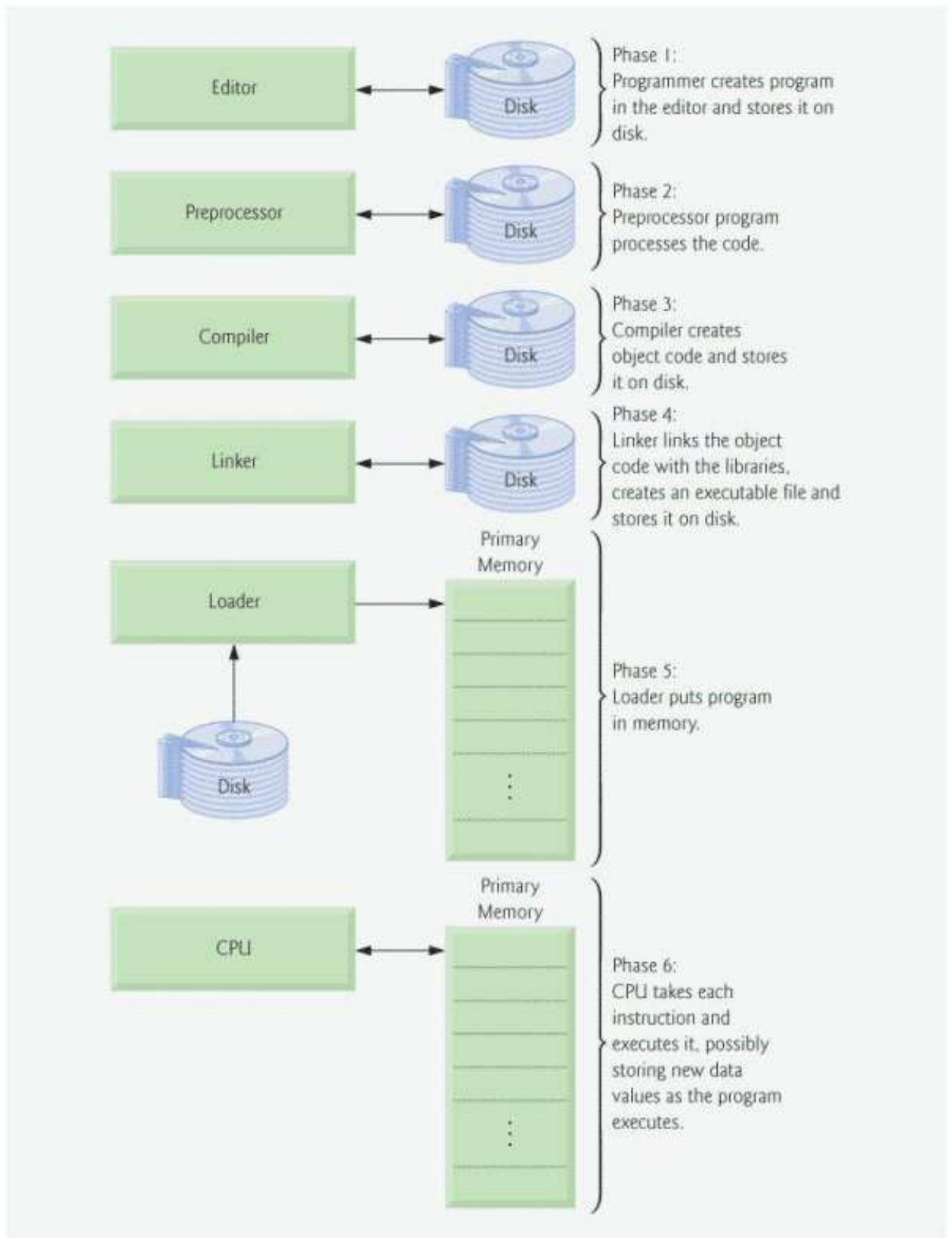
compilando programas, habitualmente escritos en el mismo lenguaje de programación que nuestro programa, y que proporcionan funcionalidad en la que se apoya nuestro software. Por ello existe una etapa posterior llamada **enlace**. En ella, el programa enlazador combina uno o varios ficheros objeto con las librerías para producir el **fichero ejecutable**.



En la figura a la derecha de estas líneas podemos observar el trabajo del enlazador (linker en inglés). El código objeto que ha generado inicialmente el compilador contiene llamadas a las librerías ("Lib ref"). Estas llamadas deben de ser reemplazadas por el código de la librería que el programa está empleando. El enlazador realiza esta operación, dando como resultado final un código máquina "completo" y autocontenido que puede ejecutarse sin dependencia de ninguna librería.

Los compiladores identifican errores sintácticos en el programa fuente y realizan tareas de optimización de código (ej.- eliminan código redundante). En ocasiones el código fuente antes de ser compilado sufre una serie de modificaciones que simplifican el trabajo del compilador como, por ejemplo, eliminar comentarios del código fuente cuya misión es facilitar la comprensión del código por parte de los programadores, y que no contiene ningún tipo de instrucción que se deba ejecutar la CPU ni datos. A estos programas se les denomina **preprocesadores**.

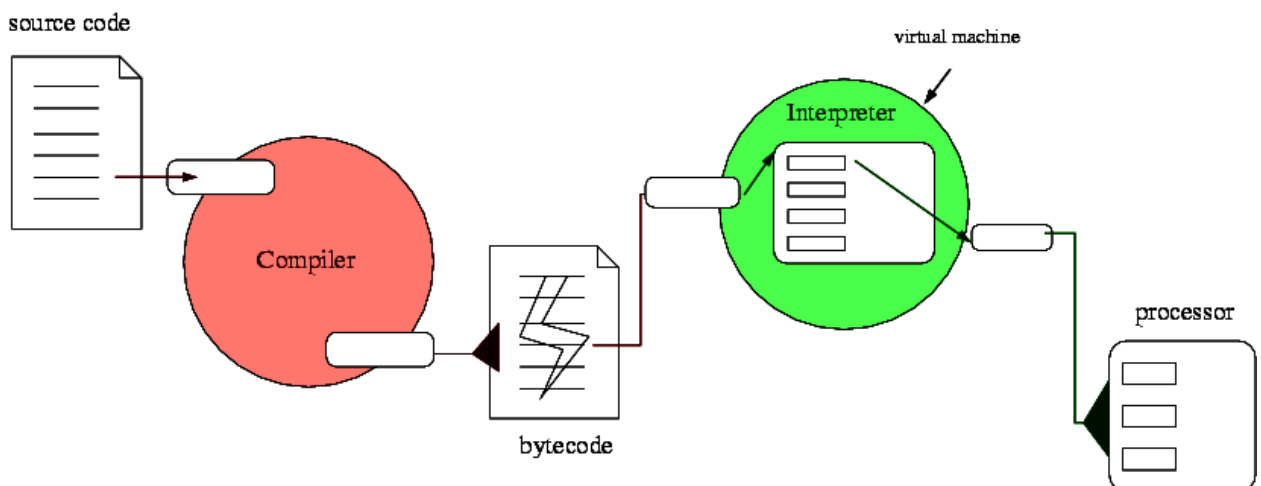
El lenguaje de programación que vamos a usar en esta asignatura, C, tiene además de un compilador un preprocesador. En C, el preprocesador toma como entrada el código fuente que hemos escrito nosotros, y genera a partir de él un código transformado, que será el código que compile el compilador para dar lugar al código objeto. El enlazador completará ese código objeto con las llamadas a las librerías, más un código inicial que permite cargar nuestro programa en memoria y arrancarlo. Cuando el programa se ejecuta, ese código inicial carga nuestras instrucciones en memoria y manda la primera instrucción a ejecutarse a la CPU. Después el programa se ejecuta tal y como hemos visto en el apartado 3.1.1 de este tema. La siguiente imagen muestra cómo es todo el proceso de preprocesamiento, compilación, enlace, carga en memoria y ejecución en el caso del lenguaje C.



3.2.2.5 Interpretación de lenguajes de alto nivel

Una de las desventajas que tiene el proceso de compilación de los lenguajes de alto nivel es que si queremos ejecutar ese programa en varias plataformas con diferentes sistemas operativos (Windows, Linux, Macintosh...) tendremos que compilar ese mismo código fuente en las distintas plataformas, y distribuir un ejecutable diferente para cada plataforma. Además, en ocasiones este proceso es complicado o incluso imposible porque no todas las librerías tienen por qué estar disponibles en todas las plataformas.

Cada vez más los lenguajes han abandonado el paradigma de la compilación y apuestan por la **interpretación**. En un lenguaje de programación interpretado el programador escribe código fuente en archivos de texto, exactamente igual que sucedía con el programa compilado. Después hay un proceso de compilación que no genera código máquina de ninguna plataforma, sino una representación intermedia de ese programa en una especie de "lengua franca de los códigos máquinas" que se suele llamar bytecode. Ese bytecode no es un código máquina que pueda ejecutar ninguna plataforma, sólo lo puede ejecutar el intérprete del lenguajes de programación en cuestión. Los **intérpretes** son programas de traducción que, en lugar de generar código objeto, analizan cada una de las instrucciones del programa fuente y las van ejecutando una a una. De este modo se consiguen programas completamente independientes del hardware. Uno de los lenguajes más populares de este tipo es JavaScript, un lenguaje que se emplea para crear pequeños programas que se ejecutan al cargar una página web en un navegador. La siguiente figura muestra cómo en un lenguaje de programación interpretado el código fuente es compilado a bytecode, que después es interpretado por el intérprete.



En la interpretación no se genera código máquina, por lo que es necesario el intérprete para poder ejecutar el programa. El programa intérprete es un programa que ha sido compilado a código nativo para cada una de las plataformas donde después podemos ejecutar nuestros programas interpretados. La ejecución de los programas interpretados suele ser más lenta que la ejecución de los programas compilados, ya que al mismo tiempo se realiza el análisis del programa y su interpretación. En la compilación se genera código máquina que se guarda en un fichero, una vez obtenido el fichero ejecutable no se necesita el compilador y su ejecución es más rápida. La figura que se encuentran bajo estas líneas hace una comparación entre el proceso de compilación y de interpretación; podemos ver cómo la etapa de "processing" (procesamiento) durante la ejecución del programa en el caso del lenguaje interpretado es más grande, es decir, hay que hacer más cosas y la ejecución del programa es menos eficiente.

En la actualidad los lenguajes interpretados, debido su mayor portabilidad, a la mayor facilidad para el desarrollo de aplicaciones, y a las mayores posibilidades de interacción entre los distintos programas, se están imponiendo sobre los compilados, si bien en ocasiones se realiza una traducción a código máquina (al menos de las partes más críticas del programa) en tiempo de ejecución por motivos de eficiencia. A esta traducción a código máquina en tiempo de ejecución de un lenguaje interpretado se le suele llamar "Just in Time Compiler".