

1. CONCEPTO DE COMPUTADOR

1. CONCEPTO DE COMPUTADOR.....	1
INTRODUCCIÓN.....	1
Definición de ALGORITMO.....	2
Proceso para resolución de un problema con el ordenador.....	2
ANTECEDENTES.....	2
LA MÁQUINA DE VON NEUMANN.....	3
Buses y señales de control.....	6
Parámetros básicos de un ordenador.....	6
LENGUAJES DE PROGRAMACIÓN.....	6
Evolución histórica.....	7
Escritura y traducción de programas escritos en lenguajes de alto nivel.....	7
Elementos básicos de un lenguaje de alto nivel.....	8
Tipos de lenguajes de alto nivel.....	8
TIPOS DE COMPUTADORES.....	9
SOPORTE LÓGICO.....	10
Software del sistema.....	10
Software de aplicación.....	11
EL SISTEMA OPERATIVO.....	11
Ejemplos de sistemas operativos.....	12

Introducción

Un computador es una máquina programable para el procesamiento de información.

Para que la información sea entendida por el ordenador debe de estar debidamente codificada. La codificación se realiza en BITS (*Binary digiT*).

Procesar información es extraer unos ciertos resultados o conclusiones a partir de unos ciertos datos de entrada.



Ejemplo:

Resolución de una ecuación de segundo grado. $2 \cdot x^2 - 3 \cdot x + 1 = 0$

Datos: 2 / -3 / 1

$$\text{Procesamiento: } x = \frac{3 \pm \sqrt{3^2 - 4 \cdot 2 \cdot 1}}{2 \cdot 2} = \frac{3 \pm \sqrt{9 - 8}}{4} = \frac{3 \pm 1}{4}$$

$$\text{Resultados: } \begin{cases} x_1 = 1 \\ x_2 = \frac{1}{2} \end{cases}$$

Para realizar un procesamiento de información el ordenador, necesariamente deberá poder realizar las siguientes tareas:

- Entrada de datos.
- Salida de resultados
- Procesamiento/Tratamiento de la información:
 - Almacenamiento de datos y programas.
 - Recuperación de datos y programas.*(Los programas son los procedimientos correctos para tratar adecuadamente la información)*

Definición de ALGORITMO

Un algoritmo es un conjunto **finito** de pasos y acciones que especifican de forma **clara y concisa (sin ambigüedades)** la **secuencia** de operaciones a realizar para procesar adecuadamente unos datos con un determinado objetivo.

Ejemplos:

1. Receta de cocina: Secuencia de pasos para realizar un determinado plato.
2. Algoritmo para obtener el resultado de dividir un número por otro.

Llamaremos **programa** a la plasmación de un algoritmo de forma que pueda ser entendido por un ordenador.

En principio el ordenador sólo es capaz de entender lenguaje binario (lenguaje que sólo contiene 2 posibles caracteres, habitualmente **0** y **1**), pero para hacer más fácil la escritura de programas existen unos lenguajes de programación a los que se asocian traductores.

Un **lenguaje de programación** es un conjunto de palabras con una cierta sintaxis y una semántica asociadas.

Las características son:

- Un vocabulario limitado (*palabras reservadas*).
- Sintaxis rígida y sin excepciones y con pocas variaciones.
- Semántica estricta y sin ambigüedades.

Proceso para resolución de un problema con el ordenador.

1. Planteamiento del problema y propuesta de un método de resolución en lenguaje natural (castellano/valenciano normal)
2. Plasmación del método de resolución en un algoritmo.
3. Traducción del algoritmo a lenguaje de programación (generación del programa).
- (4. Compilación del programa a lenguaje binario (traducción automática))*

Antecedentes

(<http://video.dlib.vt.edu/~history/>)

(<http://video.dlib.vt.edu/cgi-bin/Lobby?Method=Timeline>)

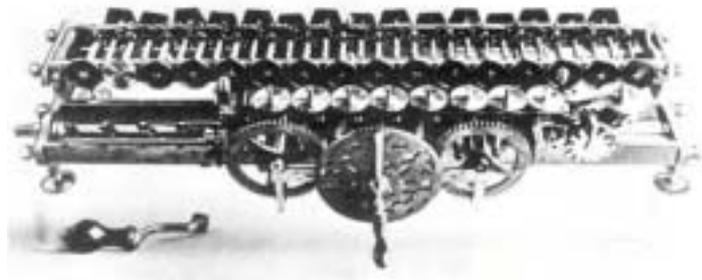
(<http://goldenink.com/computersandnetworks.shtml>)

Ábaco: Herramienta mecánica útil para sumar, restar, multiplicar y dividir. El programa es la mente humana.

Calculadoras mecánicas:

Blaise Pascal (1623-1662) desarrolló en 1642 una máquina aritmética con engranajes, ruedas y ventanas que realizaba sumas y restas.

Gottfried Leibnitz (1646-1716) desarrolla en 1673 una máquina mecánica, llamada la rueda de Leibnitz, que realizaba sumas, restas, multiplicaciones, divisiones y raíces cuadradas.



Máquina de Leibniz

También podemos considerar como antecedente de las actuales computadoras los telares mecánicos desarrollados durante la revolución industrial, que cambiaban los dibujos de las telas a partir de la información extraída de tarjetas perforadas.

En 1833, Charles Babbage, desarrolla una máquina de propósito general (máquina analítica) que puede ser considerada como la base de los computadores modernos. Tanto los programas como los datos se guardaban en tarjetas perforadas, y los programas podían utilizar los datos de salida como nuevas entradas.

Computadores electromecánicos:

Tras diferentes máquinas electromecánicas desarrolladas durante la década de los 30, a finales de los 30 se desarrolla el Harvard MARK I (la primera calculadora digital automática, de propósito general, a gran escala, electromecánica) concebida por Howard Aiken a finales de los 30 y construida por Hamilton, Lake y Durfee de IBM. Esta máquina realizaba 1 operación cada 3 segundos. Los programas no se guardaban en la máquina sino que eran leídos e interpretados desde una cinta de papel, mientras que los datos eran proporcionados en tarjetas perforadas.

Entre 1940 y 1944 un grupo de investigadores de Bletchley Park, entre Oxford y Cambridge, incluyendo a Alan Turing desarrollaron una serie de máquinas que culminaron en el desarrollo y la implementación de Colossus en 1943. Estas máquinas fueron desarrolladas especialmente para la interpretación, codificación y decodificación de códigos secretos.

Computadores electrónicos:

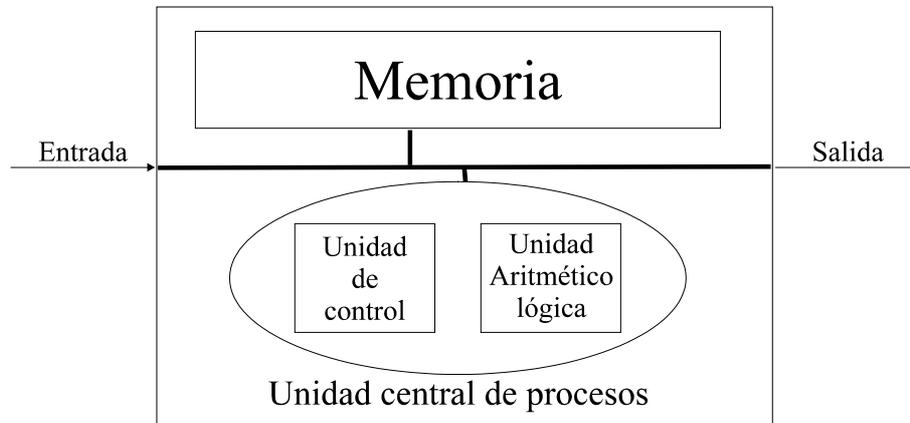
En 1946 se desarrolla el ENIAC (*Electronic Numerical Integrator Analyzer and Computer*). Construido con válvulas de vacío era mucho más veloz que los ordenadores de su época (unas 5000 operaciones por segundo). Tiene como característica especial que los programas eran introducidos en la máquina mediante el cambio de posición de cables que cambiaban la forma en que se conectaban las diferentes partes del ordenador. Este 'problema' se soluciona en versiones posteriores de la máquina. Para hacer la máquina más impresionante a los ojos de los reporteros de la época, un miembro del equipo (posiblemente Eckert) puso unas esferas translúcidas (medias pelotas de ping-pong) sobre las luces. La oficina de patentes de Estados Unidos, más tarde, reconoció esta como la primera computadora.

La máquina de Von Neumann

En 1945, el matemático y químico Jonh (Janos) von Neumann presenta los principios generales que debe seguir una máquina de propósito general.

- La máquina debe ser controlada por un conjunto de instrucciones con un pequeño número de elementos centrales de proceso.
- Tanto la información (datos) como el programa (método de procesado de la información) deben almacenarse en el interior del computador en formato binario (con un alfabeto compuesto exclusivamente de ceros y unos).

La máquina de Von Neumann es una máquina ideal que seguiría el siguiente esquema:



Las unidades funcionales de la máquina de Von Neumann son:

- Unidad de Entrada (*Input (I)*):
- Unidad de Salida (*Output (O)*):
- Memoria:
- Unidad central de procesos (U.C.P.) (*Central Process Unit (C.P.U.)*):
 - **Unidad de control (U.C.) (Control Unit):**
 - **Unidad lógico-aritmética (U.L.A.) (Arithmetic and Logic Unit (A.L.U.)):**
- **Unidad de Entrada (*Input (I)*):**

Dispositivos por donde se introducen los datos e instrucciones.

❖ Teclado, ratón, lector de barras,...

- **Unidad de Salida (*Output (O)*):**

Dispositivos por donde se obtienen los resultados.

❖ Monitor, impresora, paneles de información de tráfico, plotter, ...

- **Unidad central de procesos (U.C.P.) (*Central Process Unit (C.P.U.)*):**
 - **Unidad de control (U.C.) (Control Unit):**

Es la encargada de administrar y coordinar los recursos y actividades del ordenador.

- ❖ Interpretar el código y generar las señales de control que lo ejecutan.
- ❖ Controlar la secuencia en que se ejecutan las operaciones.
- ❖ Controlar el acceso a memoria.
- ❖ Enviar y recibir señales de control relacionadas con las operaciones que se ejecutan en la A.L.U.
- ❖ Regular la E/S (Entrada/Salida.)

- **Unidad lógico-aritmética (U.L.A.) (Arithmetic and Logic Unit (A.L.U.)):**

Unidad que contiene los registros básicos para realizar las operaciones aritméticas y lógicas.

- **Memoria:**

Encargada de almacenar de forma eficiente tanto los datos como los programas.

- ❖ Memoria principal (RAM/ROM)
- ❖ Memoria secundaria (discos/cintas/CDRom)

Información almacenada en bytes (8 bits)

- ❖ 1 Kilobyte (1 Kb) = 2^{10} bytes = 1.024 bytes \approx 1.000 bytes
- ❖ 1 Megabyte (1 Mb) = 2^{20} bytes = 1,048.576 bytes \approx 1,000.000 bytes
- ❖ 1 Gigabyte (1 Gb) = 2^{30} bytes = 1.073,741.824 bytes \approx 1.000,000.000 bytes
- ❖ 1 Terabyte (1 Tb) = 2^{40} bytes \approx 1,200.000,000.000 bytes

En relación de tamaños y rapidez de memorias contenidas en los ordenadores actuales tendríamos:

- *Registros:* Pequeñas memorias que están en la A.L.U. y sobre las que se realizan las operaciones directamente.
- *Cache:* Memoria intermedia contenida en la C.P.U. de manera que no hay que atravesar los buses para acceder a la información contenida en la caché.
- *Memoria principal (RAM):* Región de memoria a la que la C.P.U. puede acceder normalmente.
- *Memoria secundaria (discos):* Memoria adicional en donde se puede guardar información de forma permanente, aunque para poder trabajar con esa información, primero la C.P.U. deberá pasarla a la memoria principal.

<i>Tipo</i>	<i>Tamaño</i>	<i>Rapidez</i>
Registro	10-50 Bytes	+
Caché	256-512 Kb	↑
R.A.M.	128 Mb-4 Gb	↓
Disco duro	10-80 Gb	-

Una de las características de la máquina de Von Neumann es que tanto los datos como las instrucciones se deben guardar dentro de la máquina (en la memoria), esto es en forma de sucesión de unos y ceros que deben ser interpretados de una forma determinada.

Un programa es una sucesión de órdenes que la C.P.U. va leyendo de la memoria y ejecutando de forma consecutiva.

El proceso básico de ejecución de un programa en la máquina de Von Neumann es el siguiente:

- a.- Se lleva una instrucción desde la memoria a la U.C. (y se incrementa el contador del programa)
- b.- La unidad de control interpreta la instrucción y la ejecuta
 - i.- Buscar información de la memoria (si hace falta) y llevarla a los registros de la A.L.U.
 - ii.- Ejecutar la instrucción (sumas, restar, mirar entrada, escribir en salida,...)
 - iii.- Poner la información resultante en la memoria (si es necesario)
- c.- Mientras no se llegue al final del programa volver al paso (a)

Los ordenadores actuales siguen principalmente las características enunciadas de la máquina de Von Neumann pero ampliada con más dispositivos.

Buses y señales de control

La manera en que se intercambian información las distintas unidades funcionales de la máquina de Von Neumann es mediante buses. Existen básicamente dos buses: el bus de datos, por donde circula la información entre la entrada, la salida, la memoria y el microprocesador (U.C.P.), y el bus de direcciones (mediante el cual se fija la dirección de memoria donde se está trabajando (guardando/leyendo información)).

La manera de controlar el acceso a este bus, y dar el control a las diferentes unidades funcionales de la máquina de Von Neumann es mediante las señales de control, generadas por la unidad de control.

Parámetros básicos de un ordenador

- **Tiempo de ciclo o frecuencia de reloj:** Determina en gran medida la velocidad con que funciona un ordenador. Es la cantidad de instrucciones básicas que se ejecutan por segundo en el ordenador. Se mide en megahercios (MHz)
- **Longitud de la palabra:** es la cantidad de bits que se mueven simultáneamente por el bus de datos del ordenador y determina en parte la precisión de los cálculos del ordenador.
- **Tamaño del bus de datos y del bus de direcciones:** El tamaño del bus de datos determina la cantidad de bits que se mueven simultáneamente entre las diferentes partes del ordenador, de manera que ayudará a que la velocidad del ordenador sea mayor. El tamaño del bus de direcciones nos limitará la cantidad de direcciones de memoria a las que podemos referirnos, limitando el tamaño de la memoria con el que podemos trabajar (con un bus de direcciones de una sola línea sólo podríamos acceder a dos casillas distintas de la memoria. Con un bus de dos líneas podríamos acceder hasta a 4 casillas distintas. Con un bus de n líneas podríamos acceder a 2^n casillas distintas.)

Lenguajes de programación

Un programa es una sucesión de pasos que debe seguir el ordenador para llevar a término un trabajo o proceso de información.

Un usuario que desee que un ordenador realice un cierto procesado de información deberá escribir un programa en un lenguaje que pueda entender el ordenador.

El único lenguaje que realmente entienden los ordenadores es el lenguaje binario (que es el único que puede estar guardado en la memoria del ordenador.). El lenguaje binario tienen como característica que sólo contiene dos símbolos (habitualmente el cero y el uno.)

Como es difícil escribir programas en ese lenguaje binario se han desarrollado otro tipo de lenguajes que deben ser fácilmente traducibles al lenguaje binario:

- Lenguajes de bajo nivel (ensamblador): Lenguajes en los que las instrucciones son directamente traducibles, una por una, a lenguaje binario.
- Lenguajes de alto nivel: Lenguajes más elaborados, fáciles de leer y entender por un humano, y que deben de ser fácilmente traducibles a lenguaje máquina.

Para que un lenguaje sea fácilmente traducible a lenguaje máquina debe cumplir una serie de características, que ya comentamos al comienzo del tema:

- Un vocabulario limitado (*palabras reservadas*).
- Sintaxis rígida y sin excepciones y con pocas variaciones.
- Semántica estricta y sin ambigüedades.

Evolución histórica

El primer lenguaje de alto nivel desarrollado fue el **FORTRAN** (*FORmula TRANslation*) en 1957. Fue desarrollado por John Backus en IBM. Es un lenguaje de alto nivel especialmente pensado en la resolución de problemas matemáticos. Pero tenía como gran inconveniente la dificultad de trabajo con ficheros y la falta de recursividad (posibilidad de definir un cierto proceso refiriéndose a sí mismo.) Estos inconvenientes se han ido mejorando a lo largo de las diferentes versiones (FORTRAN 66, FORTRAN 70, FORTRAN 90 y FORTRAN 95).

Poco después del FORTRAN apareció el **BASIC** (*Begginers All-purpose Symbolic Instruction Code*), desarrollado por John Kemeny y Thomas Kuntz especialmente para enseñar a programar a usuarios poco especializados.

El uso de grandes ordenadores a aplicaciones empresariales llevó a desarrollar lenguajes específicos como el **COBOL** (en 1959.) Los lenguajes empresariales se caracterizan por desarrollar fácilmente informes elaborados, maneras precisas de describir y guardar números decimales y caracteres y la habilidad para especificar operaciones decimales.

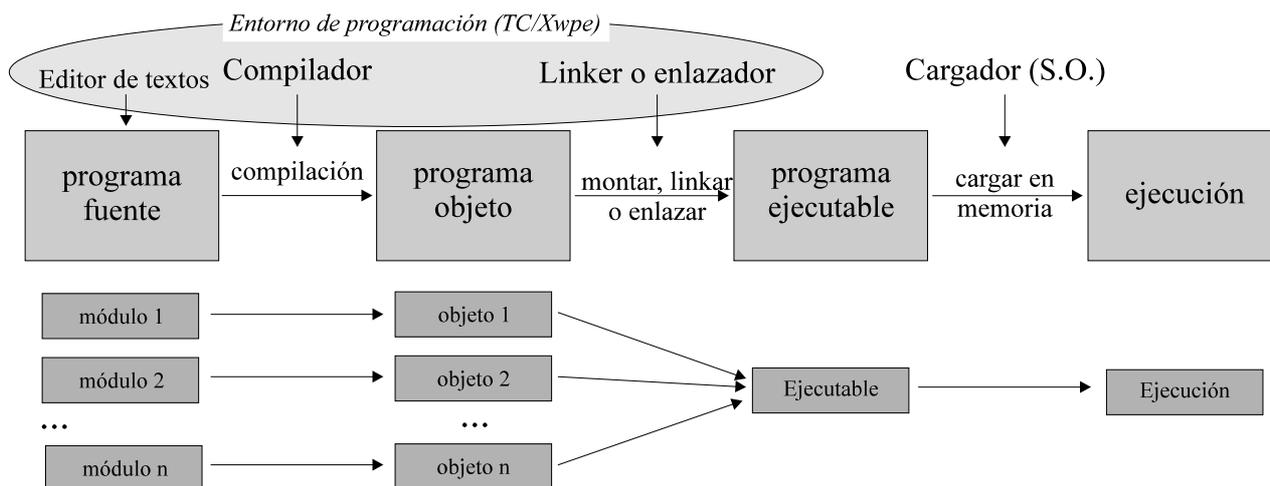
En los 70, con la llamada ‘crisis del software’, aparecen los conceptos de programación estructurada, y Niklaus Wirth plasma estos conceptos en el lenguaje **PASCAL** (1971) desarrollado especialmente para enseñar programación estructurada.

También a principios de los 70 Kernighan y Ritchie desarrollan el lenguaje de programación **C** (1972), que admite los conceptos de programación estructurada, tal como Pascal, pero que permite una mayor libertad a la hora de realizar ciertas tareas que no pueden llegar a hacerse en la mayoría de los lenguajes de alto nivel. De hecho el C es un lenguaje de alto nivel, que es capaz de bajar de nivel y con el que es posible programar incluso sistemas operativos, que ya comentaremos más adelante.

A principios de los 80, se empiezan a desarrollar nuevos conceptos de programación que terminarían en la programación orientada a objetos y lenguajes como **C++**, **Eiffel** o **Smalltalk**.

Escritura y traducción de programas escritos en lenguajes de alto nivel

El proceso entre la escritura de un programa por parte del usuario y que el programa pueda ser ejecutado por el ordenador es el siguiente:



El primer paso es escribir el programa en un editor de textos. Una vez escrito tendremos el llamado programa fuente (*source*) escrito en un lenguaje de alto nivel.

A continuación mediante el uso de un compilador pasaremos el programa fuente a un lenguaje intermedio y obtenemos el programa objeto. En este paso se realiza el análisis léxico (que sólo se hayan escrito símbolos y palabras correspondientes al lenguaje de alto nivel que estemos utilizando) y

sintáctico (que los diferentes símbolos y palabras escritos estén en la sucesión correcta) del programa fuente y ciertas comprobaciones semánticas (que nos aseguran el significado correcto de las expresiones).

El programa fuente puede estar escrito en varios módulos. La compilación traduce cada uno de los módulos a un programa objeto. En los diferentes objetos puede haber referencias a elementos que estén en otros objetos.

El último paso de la traducción es el *linkado* (del inglés *link*, enlazar) o montado de los diferentes objetos en un solo programa ya escrito en lenguaje binario y que se suele denominar programa ejecutable.

Finalmente, hay que cargar el programa ejecutable en memoria y ejecutarlo para obtener los resultados esperados.

Aunque en principio los lenguajes de programación son únicos, cada compilador específico de cada máquina puede incorporar ciertas características que no comparte con otros compiladores del mismo lenguaje de alto nivel.

Para evitar en lo posible la dispersión de un mismo lenguaje la *American National Standards Institute (ANSI)* establecen unos estándares para cada lenguaje de programación. Así podemos hablar de C en general, que puede contener ciertas características especiales o podemos hablar del ANSI C, que cumple estrictamente con los estándares establecidos permitiendo una mayor portabilidad de los programas escritos (mayor facilidad para llevarlos de un ordenador a otro sin tener que modificarlos).

Elementos básicos de un lenguaje de alto nivel

- Palabras reservadas
- Constantes y variables
- Expresiones...
 - ...aritméticas: + / - / * / ... devuelven valores aritméticos
 - ...relacionales: > / < / = / ... devuelven valores booleanos (TRUE/FALSE)
 - ...lógicas: AND / OR / ... devuelven valores booleanos (TRUE/FALSE)
- Etiquetas
- Comentarios
- Asignaciones (←)
- Entrada
- Salida

Tipos de lenguajes de alto nivel

1ª clasificación: Por la forma en que se traducen.

- Lenguajes compilados → C, Pascal, C++,...
Los programas, una vez escritos de forma completa se traducen completamente (se compilan mediante la utilización de un compilador) y se tienen preparados para poder ser ejecutados, de manera que la ejecución, una vez el programa está traducido es mucho más rápida.
- Lenguajes interpretados → BASIC
Los programas se analizan a medida que van siendo escritos, y la ejecución se realiza línea a línea, de manera que es más fácil escribir los programas y encontrar errores en ellos, aunque la ejecución es mucho más lenta y siempre tenemos que tener disponible el traductor (intérprete).

2ª clasificación: Objetivo principal de los programas escritos en el lenguaje.

- Lenguajes de propósito general → C, BASIC, Delphi, Java...
- Lenguajes para la enseñanza → Logo, Pascal...
- Lenguajes para cálculo científico → Fortran, Matlab, Mathematica, Maple...
- Lenguajes para gestión → Cobol, Modula...
- Lenguajes para la gestión de bases de datos → *System Query Languages (SQL)* Informix...
- Lenguajes de inteligencia artificial → Prolog, Lisp...

3ª clasificación: Por la escritura y concepción de los programas.

- Lenguajes procedurales o procedimentales (Programa = procedimientos + algoritmos) →
→ C, Pascal, Modula...
- Lenguajes declarativos (Programa = definición de elementos + relaciones) →
→ Prolog
- Lenguajes orientados a objetos (Programa = Objetos + métodos + mensajes) →
→ Eiffel, C++, Java...

4ª clasificación: En función de la gestión de la memoria.

- Lenguajes de gestión estática de memoria (se calcula la memoria cuando se realiza la traducción) → Fortran...
- Lenguajes de gestión dinámica de memoria (se reserva la memoria en ejecución, a medida que van apareciendo variables) → Basic, Prolog...
- Memoria en pilas (se guarda la información en pilas, apilando y desapilando información) → C, Pascal...

Tipos de computadores

- **Supercomputadores** (supercomputación): Existen varios procesadores haciendo cálculos (computando) al mismo tiempo (en paralelo) y accediendo a grandes cantidades de información guardada en la memoria o en accesible en dispositivos de entrada. Pensados para procesar grandes cantidades de información en poco tiempo y dedicados a una tarea específica.
 - ❑ Control de órbitas de satélites y naves espaciales.
 - ❑ Aceleradores de partículas.
 - ❑ Control y predicción de procesos biológicos y químicos.
 - ❑ Predicciones termodinámicas (predicciones de tiempo atmosférico.)
- **Macrocomputadores** (o Mainframes): Ordenadores compuestos por distintos procesadores que se encargan de la gestión y el control de diferentes dispositivos (sería un ordenador distribuido.). Controlan cientos de usuarios simultáneamente y cientos de dispositivos de entrada-salida. Son más potentes que las Supercomputadoras dado que pueden tener mayor número de programas ejecutándose simultáneamente pero las Supercomputadoras pueden ejecutar un único programa más rápidamente.
 - ❑ Control de robots en fábricas.
 - ❑ Bancos.
 - ❑ Líneas aéreas.

- **Minicomputadores:** Son ordenadores con varios procesadores. Surgen en 1960 y son una versión más pequeña que las Macrocomputadoras. Están orientados a tareas específicas y no requieren tantos periféricos ni usuarios como las Macrocomputadoras, por lo que redujeron el precio y el mantenimiento.
 - Control de procesos en laboratorios.
 - Automatización industrial.
- **Estaciones de trabajo (WorkStations):** Microprocesadores muy potentes y optimizados para la realización de tareas concretas y diferentes procesos simultáneos (pensados para atender a varios usuarios simultáneamente.)
 - Diseño asistido por ordenador (C.A.D.)
 - Aplicaciones de ingeniería
- **Microcomputadores:** Pensados para el trabajo personal (*Personal Computer (PC)*). Tienen su origen en la creación de los microprocesadores (*la computadora en un chip*).
 - Apple
 - IBM/PC
- **Calculadoras de bolsillo.**

Soporte lógico

Ordenador → Parte física (*Hardware*)
→ Parte lógica (*Software*)

El **soporte lógico** de un ordenador es el conjunto de programas que permiten a un usuario realizar ciertas tareas en él.

Dentro del *software* podemos distinguir entre:

- **Software del sistema:**
Es el encargado de administrar (gestión + mantenimiento) los recursos de la máquina.
- **Software de aplicación:**
Lo forman todos los programas que puede utilizar un usuario para realizar ciertas tareas específicas con el ordenador (edición de textos, contabilidad, diseño,...)

Software del sistema

- Programa de arranque: Es el conjunto de tareas que se realizan en el ordenador en el momento de encenderlo.
 1. Chequeo y comprobación de dispositivos hardware.
 2. Lectura de parámetros de la BIOS y configuración de algunos dispositivos (memoria, discos...)
 3. Búsqueda del sistema operativo y carga del mismo en memoria.
- Sistema operativo (S.O.): Es un conjunto de programas que relacionan al usuario con el hardware específico del ordenador, de manera que para el usuario no importa el tipo o marca del hardware, sólo el S.O. que este funcionando en el ordenador.

Además se encarga de gestionar eficientemente los recursos de la máquina.
- Programas de ayuda al sistema operativo.

- Programas de diagnóstico: *scandisk*, *ddn*, *speedisk*...
 - Programas de mantenimiento del sistema: Añadir o modificar elementos hardware en el sistema.
- Utilidades generales.
- Traductores (compiladores e intérpretes.)
 - Depuradores (*debuggers*.)
 - Gestores de ficheros.

Software de aplicación.

- Programas de edición de textos.
- Gestión de bases de datos (gestión de bibliotecas, de matriculados en la universidad, etc.)
- Programas de administración y contabilidad.
- Herramientas de diseño y edición gráfica (CAD-CAM)

El sistema operativo

Son **funciones del sistema operativo**:

- Hacer de interfaz entre el usuario y el hardware de la máquina.
- Administrar y controlar los dispositivos hardware.
- Mantener el sistema de archivos.
- Apoyar la ejecución de otros programas.

Para realizar estas funciones de forma correcta, debemos exigir a un **sistema operativo** las siguientes **características**:

1. Eficiencia (rapidez en las tareas.)
2. Fiabilidad (sin fallos ni errores.)
3. Facilidad de mantenimiento.
4. Tamaño reducido (aunque últimamente esta característica está siendo sustituida por facilidad de manejo por parte de un usuario inexperto.)

Las principales **partes del sistema operativo** son: El *Kernel* o núcleo del sistema y el interfaz.

- Núcleo o *Kernel*: Se encarga de las tareas más básicas y próximas al hardware.
 - Asignación de trabajos al microprocesador (μP .)
 - *Swap* o intercambio de tareas en el μP y de datos y programas en memoria.
 - Comunicación entre procesos.
 - Administración del hardware: Gestión de la memoria, control de entrada y salida (*drivers*), administración de archivos, apoyo a la ejecución de programas,...
- Interfaz: Es el elemento a través del que el usuario se comunica con el hardware, y en general está formado por unas instrucciones u órdenes (muchas veces mal llamadas “comandos” del inglés *command*) que permiten al usuario realizar ciertas tareas específicas.

Ejemplos de sistemas operativos

- MS-DOS:
 - Sistema operativo de las máquinas con μ P Intel (8088, 8086, 80286, 386, 486, Pentium,...)
 - Se centra en el control de los aspectos del PC enfatizando en la gestión de sistema de ficheros y la transferencia de información hacia los periféricos.
 - Funciona mediante interrupciones.
 - Para mantener la compatibilidad con versiones anteriores mantiene ciertas características que impiden aprovechar los últimos avances de los PC's.
 - Monousuario y monoproseso.
- UNIX:
 - Sistema operativo de propósito general, desarrollado para cualquier plataforma (en principio cualquier μ P, intentando sacar el máximo partido del hardware aunque no se mantenga la compatibilidad entre versiones.)
 - Esquema básico: Núcleo pequeño + Extensiones propias.
 - Multiusuario y multiproseso.
 - Cualquier sistema de entrada/salida (I/O) se trata exactamente como si fuese un archivo.
 - Al ser multiusuario añade sistemas de protección y seguridad de información y de procesos.
- Windows 95/NT/2000
 - Sistema operativo desarrollado para que el usuario tenga un fácil acceso a los recursos del sistema (sistema de ventanas, iconos y puntero de ratón.)
 - No es necesario ningún conocimiento del funcionamiento de la máquina porque se pretende que todo sea muy visual, muy intuitivo y automático.