Tema 1

Introducción y Conceptos Generales

Introducción

- Definición, elementos y aplicaciones de la informática
- Antecedentes
- La máquina Von Neumann
- Lenguajes de programación
- Tipos de ordenadores
- Soporte lógico
- El sistema operativo

Concepto de informática y ordenador

La palabra *Informática* proviene de la fusión de los términos *información* y *automática*. Definiremos *Informática* como la disciplina científica encargada del tratamiento de la información mediante métodos automáticos.

Con ello, definiremos *ordenador* como una máquina programable para el procesado de información. Más formalmente, podremos definir ordenador como un dispositivo electrónico programable capaz de almacenar, recuperar y procesar información.

Máquina procesadora de información

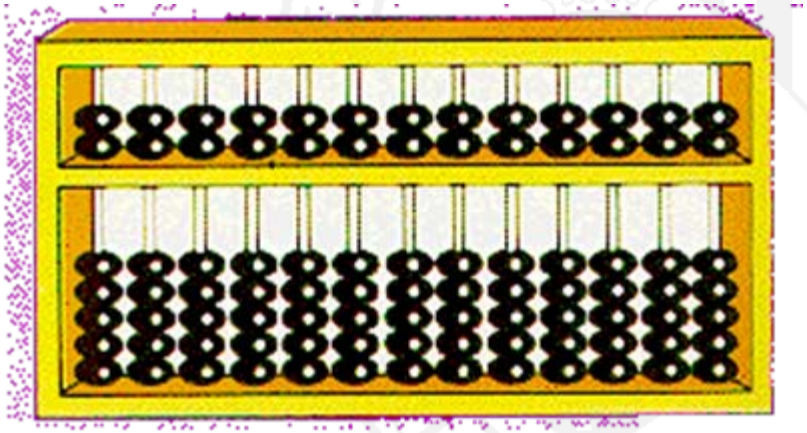
Funciones que debe realizar un ordenador

- Entrada de datos
- Salida de resultados
- Procesamiento/ Tratamiento de la Información
  - Almacenamiento de datos y programas
  - Recuperación de datos y programas
- Transmisión/Recepción de información

Antecedentes históricos

- Las distintas funciones antes citadas tienen precedentes históricos que en muchos casos no tenían como objetivo el desarrollo de una máquina electrónica programable procesadora de información

Antecedentes históricos: el ábaco



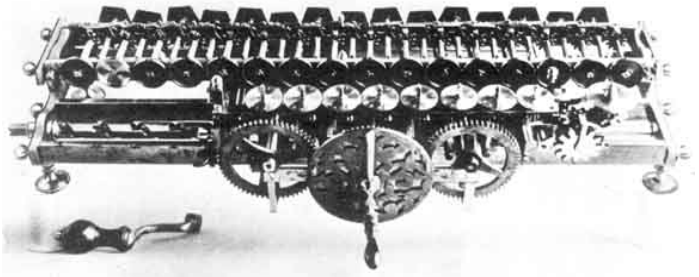
- Origen posiblemente chino, de unos 2500 años.
- Dispositivo manual para la asistencia al cálculo.
- Permite almacenar cantidades y operar sobre las cantidades representadas.
- El procesado de la información representada lo realiza el usuario.

Las primeras calculadoras mecánicas

- Blaise Pascal inventó en 1642 la primera sumadora mecánica, conocida como la *Pascalina,* que *r*ealizaba sumas y restas

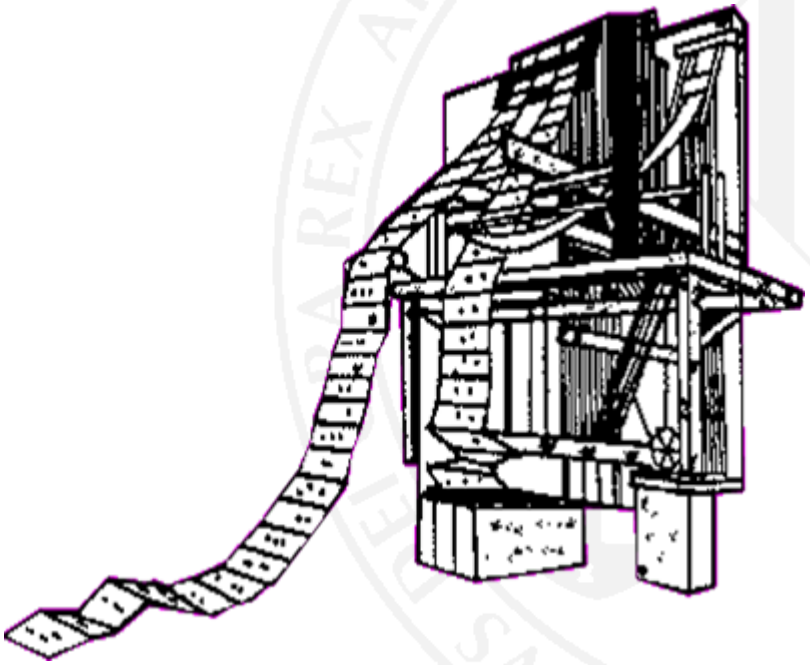
Sumadora de Blaise Pascal

- A finales del siglo XVII Leibnitz inventó otra máquina capaz además de multiplicar, dividir y calcular raíces cuadradas.
- Basadas en ruedas, engranajes y ventanas.
- Capacidad de cálculo limitada, pero falta de capacidad de programación



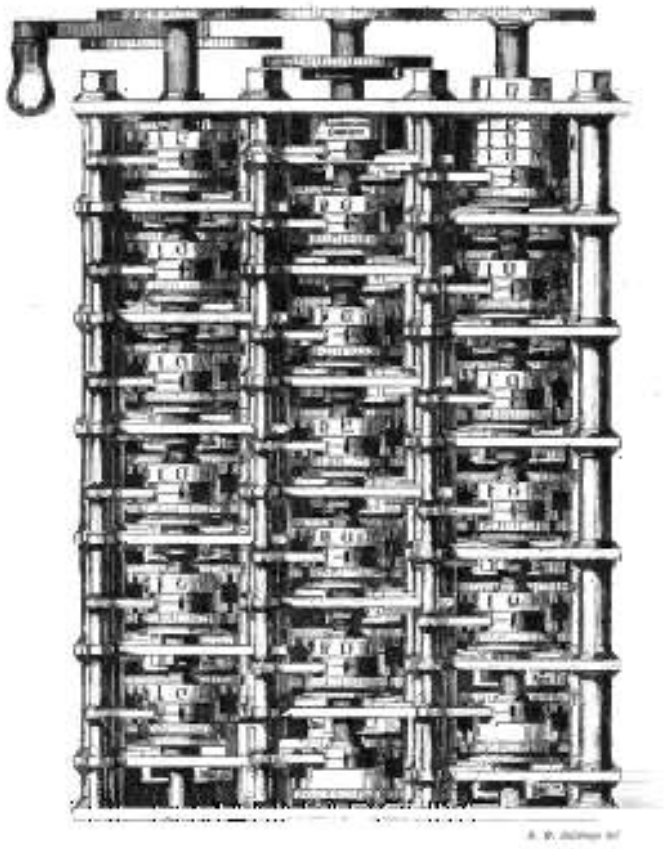
Rueda de Leibnitz

Las tarjetas perforadas



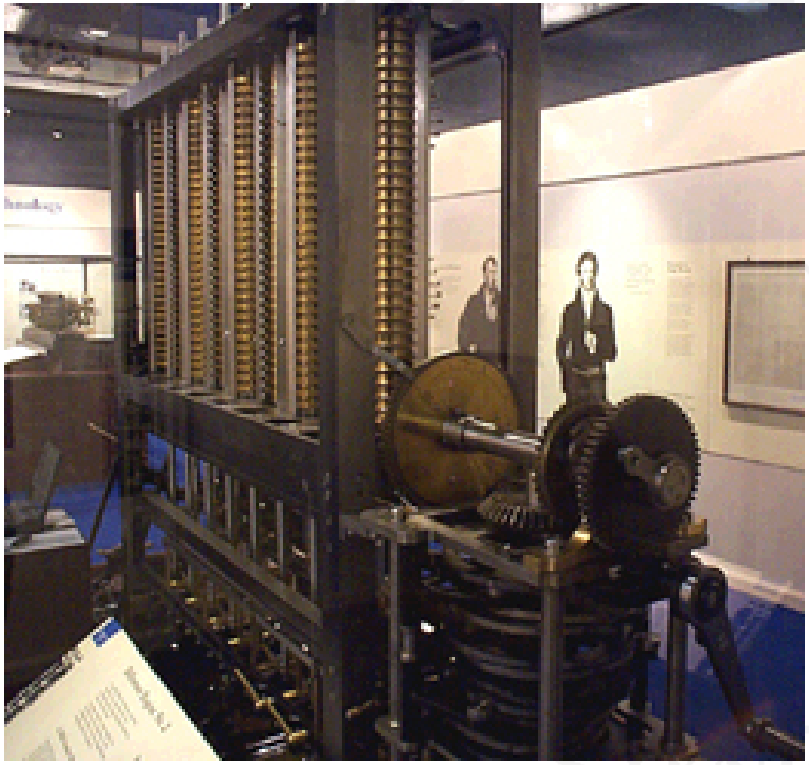
- En 1801, Joseph-Marie Jacquard inventa un telar capaz de cambiar los dibujos de las telas basándose en un perfil codificado en tarjetas perforadas
- La máquina era por tanto programable, ya que bastaba con cambiar las tarjetas para conseguir un dibujo diferente.

Las primeras calculadoras mecánicas (2)



- En 1835 Charles Babbage desarrolla una calculadora mecánica llamada *máquina diferencial*
- Era capaz de sumar, restar, multiplicar y dividir.
- La tecnología disponible en aquel momento, la mecánica, limitó un diseño conceptual muy avanzado.

Las primeras calculadoras mecánicas (3)

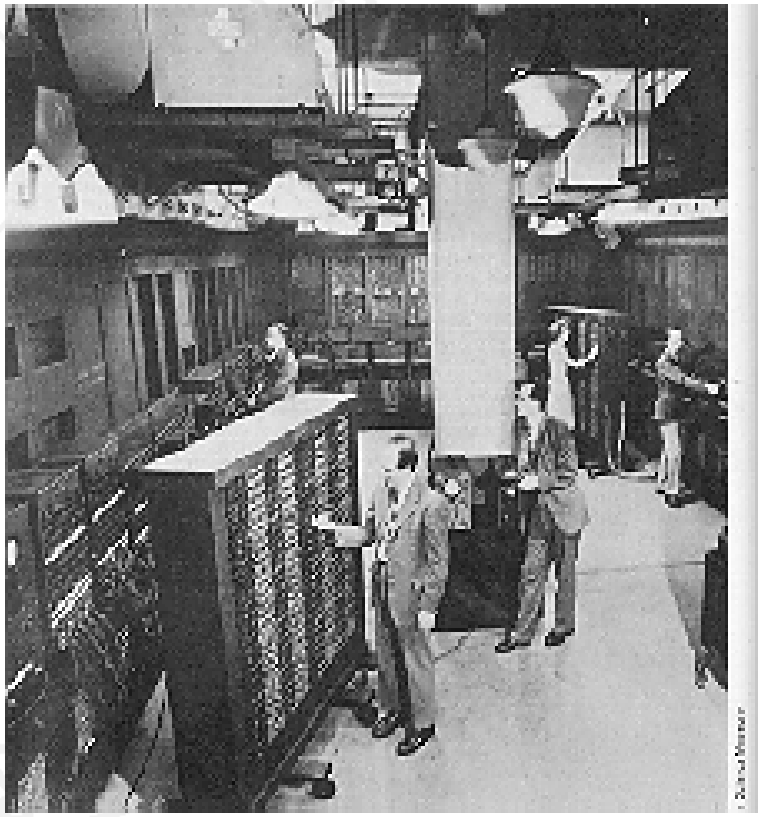


Máquina diferencial operativa

- Años más tarde Babbage diseñó la *máquina analítica*, con características avanzadas:
  - Mayor capacidad de cálculo
  - Programa definido en tarjetas perforadas.
  - Capacidad de “control de programa”: bucles, bifurcaciones condicionales...
  - Uso de resultados previos como datos de entrada para pasos posteriores.
- La máquina analítica no fue construida pero sentó las bases de los actuales ordenadores.

Los primeros ordenadores

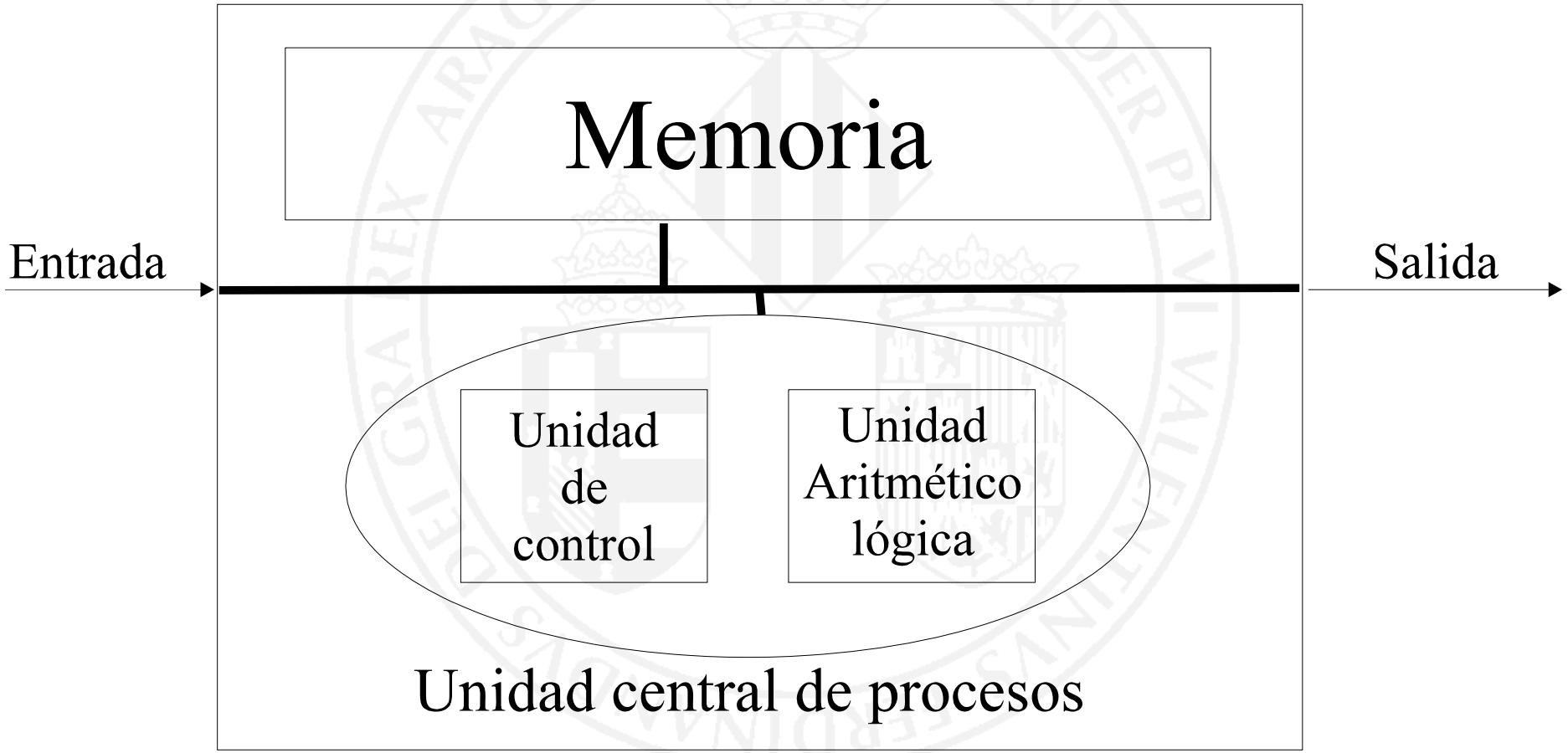
- A partir de la segunda guerra mundial aparecen los primeros ordenadores programables.
- Los primeros fueron electromecánicos, basados en relés telefónicos. La programación se realizaba mediante cambios en el cableado entre las unidades funcionales.
- La velocidad y la miniaturización han ido aumentando a medida que se han ido desarrollando tecnologías más avanzadas en cuanto a las unidades funcionales.
- Así, podemos hablar de varias generaciones basadas en:
  - Relés (1944)
  - Válvulas de vacío (1946)
  - Transistores (1947)
  - Circuitos integrados (1960)
  - Circuitos de alta integración.



La máquina Von Neumann

- Se trata de un diseño conceptual que especifica cómo debería funcionar una máquina programable capaz de procesar información.
- La máquina Von Neumann deberá tener las siguientes características:
  - 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 Von Neumann



Unidades funcionales de la máquina Von Neumann

- Unidad de entrada (input): Dispositivos que permiten introducir información en el sistema.
- Unidad de salida (output): Dispositivos que permiten mostrar información.
- Unidad Central de procesos (CPU): Se puede dividir en dos componentes:
  - Unidad de control (UC)
  - Unidad aritmético-lógica (ALU)
- Memoria: Almacena los datos y programas.

Unidades funcionales de la máquina Von Neumann: La unidad central de procesos

- La CPU puede dividirse en dos subunidades con funciones claramente diferenciadas:
  - La unidad de control (U.C.), cuyas tareas son:
    - 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 entrada/salida.
  - La unidad aritmético-lógica (A.L.U.), que se encarga de ejecutar las instrucciones aritméticas y lógicas con los datos según el programa almacenado en la memoria.

Unidades funcionales de la máquina Von Neumann: La memoria

- En una máquina Von Neumann, tanto los datos como las instrucciones para procesarlos se almacenan internamente.
- La memoria es el dispositivo que se encarga de almacenar toda la información.
- La memoria podrá clasificarse, según el grado de accesibilidad de sus contenidos, en:
  - Memoria primaria (principal)
  - Memoria secundaria
- El contenido de la memoria se almacenará codificado en forma de bits (0 ó 1)
- Denominaremos *byte* a un conjunto de 8 bits.

Unidades de medida de almacenamiento digital

- BIT (BInary digiT): unidad mínima de información que sólo puede contener 0 ó 1.
- Denominaremos *byte* a un conjunto de 8 bits.
- A partir de ahí:
  - 1 Kilobyte (1 Kb) = 2^{10}bytes = 1.024 bytes ≈ 1.000 bytes
  - 1 Megabyte (1 Mb) = 2^{20}bytes = $1_1$048.576 bytes ≈ $1_1$000.000 bytes
  - 1 Gigabyte (1 Gb) = 2^{30}bytes = 1.073$_1$741.824 bytes ≈ 1.000$_1$000.000 bytes
  - 1 Terabyte (1 Tb) = 2^{40}bytes ≈ $1_2$000.000$_1$000.000 bytes

Tipos de memoria

- 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 directamente.
  - *Memoria secundaria:* 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.

Tipos de memoria y órdenes de magnitud

Tipo	***Orden***	***Rapidez***
Registro	Bytes	+ ↑ | ↓ -
Caché	Kb	
R.A.M.	Mb/ Gb	
Disco duro	Gb/ Tb	

Proceso de ejecución de un programa en una máquina Von Neumann

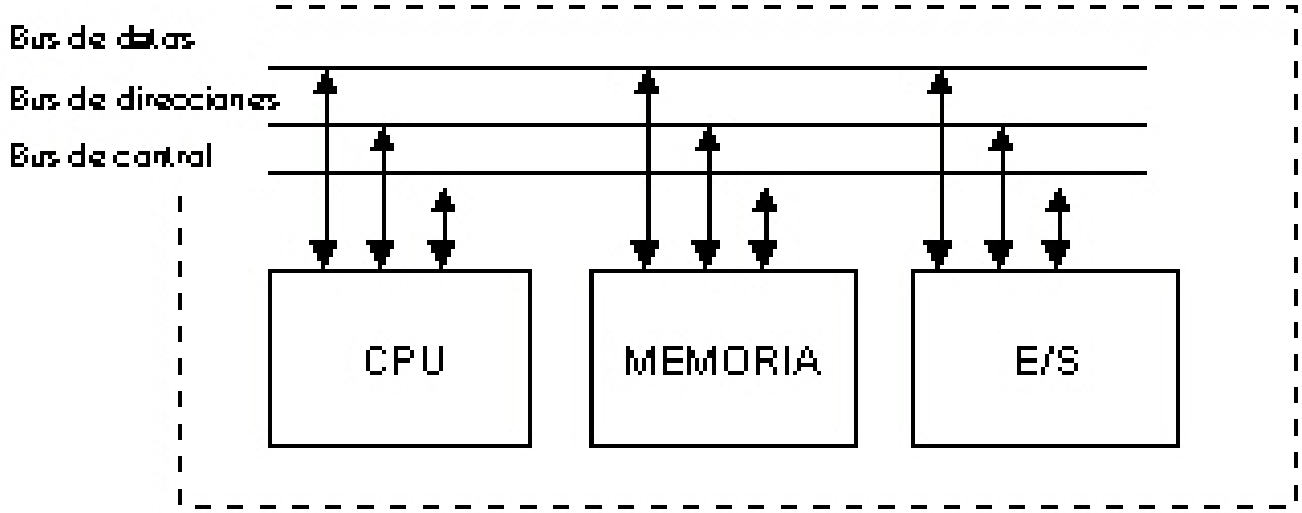
A- Se lleva una instrucción desde la memoria a la U.C. (y se incrementa el contador del programa)
B- La unidad de control interpreta la instrucción y la ejecuta:

1- Busca información en la memoria y la lleva a la CPU
2- Ejecuta la instrucción
3- Poner el resultado de la instrucción en la memoria

C- Si no ha terminado el programa, volver a A

Máquina Von Neumann: Buses y señales de control

- Las unidades de una máquina Von Neumann se comunican mediante buses. Al menos habrá dos buses:
  - Bus de datos
  - Bus de direcciones
- Habitualmente existe un tercer bus de control, que se encarga de transmitir las señales de control y sincronización generadas por la unidad de control



Concepto de algoritmo, programa y lenguaje de programación

- Definiremos *algoritmo* como 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.
- Llamaremos **programa** a la plasmación (implementación) de un algoritmo de forma que pueda ser entendido y ejecutado por un ordenador, mediante un lenguaje comprendido por éste.

Lenguaje de programación

- Un lenguaje de programación es un conjunto controlado de palabras con una sintaxis y una semántica asociadas que permiten describir algoritmos de forma que puedan ser ejecutados por un ordenador.
- Todo lenguaje de programación deberá tener:
  - Un vocabulario limitado (*palabras reservadas*).
  - Una sintaxis rígida y sin excepciones y con pocas variaciones.
  - Una semántica estricta y sin ambigüedades.

Proceso de resolución de un problema con un ordenador

1. Planteamiento del problema y propuesta de un método de resolución en lenguaje natural.
2. Plasmación del método de resolución en un algoritmo.
3. Traducción del algoritmo a un lenguaje de programación (generación del programa o implementación).
4. *Interpretación o compilación del programa a lenguaje binario.*

Tipos de lenguajes de programación

- Según su nivel de abstracción:
  - Lenguajes de bajo nivel:
    - Son lenguajes máquina o cercanos a la máquina.
    - Código máquina o ensamblador.
  - Lenguajes de alto nivel:
    - Son lenguajes cercanos al usuario o al problema.
    - Tienen un alto nivel de abstracción.
    - C, C++, Pascal, Fortran, etc.

Tipos de lenguajes de programación

- Según el tipo de ejecución:
  - Interpretados: Cada instrucción se traduce al lenguaje máquina durante el tiempo de ejecución. El programa que realiza esta tarea se denomina *íntérprete.* Ej: Basic o Perl.
    - Más fácil de escribir los programas y encontrar errores.
    - Ejecución más lenta y hay que tener disponible el traductor.
  - Compilados: El programa fuente completo se traduce a código máquina directamente ejecutable por parte de la máquina. Ej: C, Pascal.
    - Una vez traducido, la ejecución es más rápida.

Tipos de lenguajes de programación

- Según su propósito:
  - De propósito general: BASIC, C, Pascal, Java,...
  - De enseñanza: Logo, Pascal…
  - Cálculo Científico: Algol, Fortran, Matlab…
  - De gestión: Cobol, Modula.
  - De gestión de bases de datos: Informix, SQL, dBASE, Oracle.
  - Inteligencia Artificial: Prolog, Lisp…

Tipos de lenguajes de programación

- Según su paradigma de programación:
  - Lenguajes procedurales/procedimentales o imperativos:
    - Un programa es una secuencia de acciones que se ejecutan en un orden controlado (PROGRAMA = PROCEDIMIENTOS + ALGORITMOS)
    - Ej: C, Pascal, BASIC
  - Lenguajes declarativos: (PROGRAMA = DEF_ELEMENTOS +RELACIONES)
    - Un programa es un conjunto de hechos y relaciones entre éstos. No existe una secuencia definida de ejecución.
    - Ej: Prolog.
  - Lenguajes orientados a objetos:

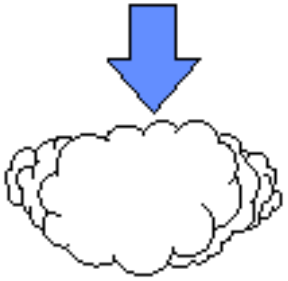
    (PROGRAMA = OBJETOS + MÉTODOS + MENSAJES)
    - Un programa está constituido por un conjunto de objetos con características de uso y comportamiento definidas que intercambian mensajes entre sí.
    - Cada objeto es una entidad que agrupa una información, su *estado*, y una serie de mecanismos para consultar o manipular dicho estado, sus *métodos*
    - *Ej: C++, Eiffel, Java.*

Tipos de lenguajes de programación

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

Proceso de programación en un lenguaje de alto nivel

Lenguaje de
Alto Nivel



Traductor

Lenguaje
Máquina

Proceso de programación en un lenguaje de alto nivel compilado

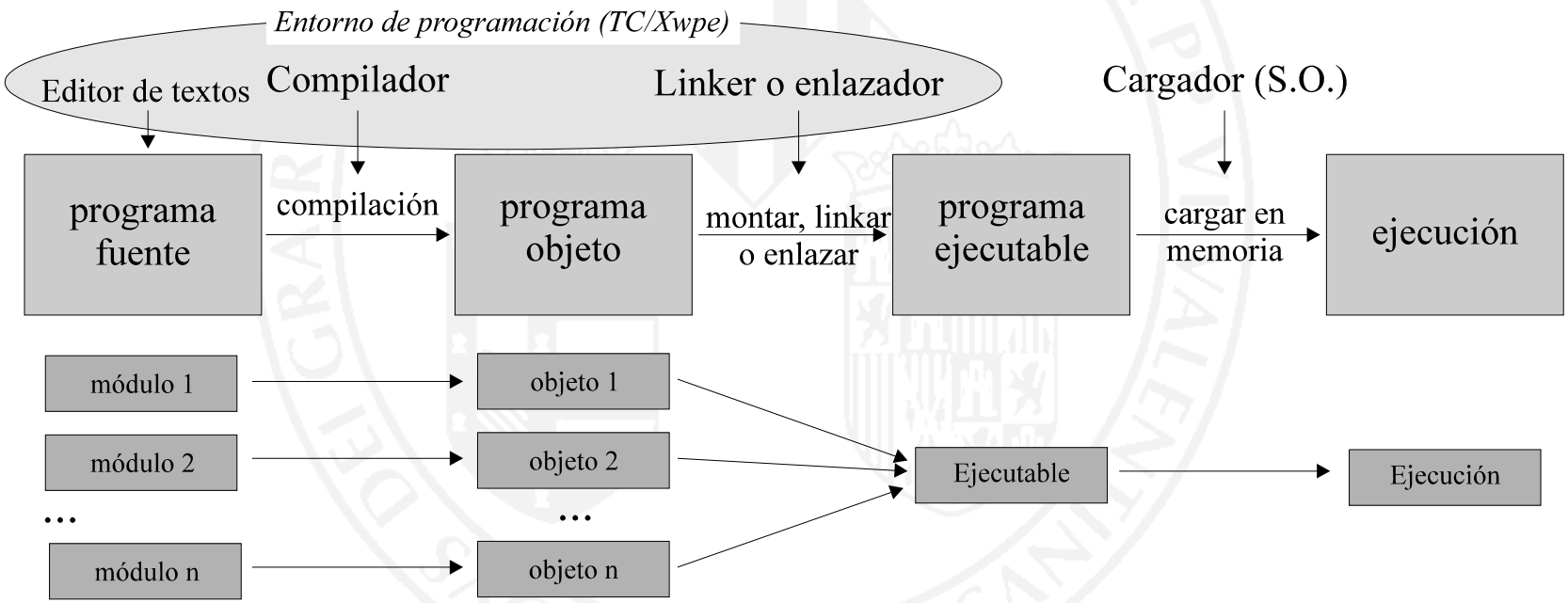
• Edición: Escribir el programa fuente en un editor de textos. Este programa puede constar de uno o más módulos.

• Compilación: Usar un compilador para realizar un análisis léxico y sintáctico del programa fuente. El resultado será un módulo objeto por cada módulo fuente.

• Enlazado (*linkado*): Los distintos módulos objeto obtenidos en la compilación se enlazan para formar un programa binario ejecutable.

•Ejecución: Carga del programa ejecutable en memoria y puesta en funcionamiento.

Proceso de programación en un lenguaje de alto nivel compilado



Componentes de un sistema informático

- Hardware: Elementos físicos del sistema.
- Software: Elementos lógicos. Podremos dividirlo en varios grupos dependiendo de su función o de su cercanía al usuario:
  - Software de sistema: el encargado de administrar los recursos de la máquina.
  - Software de aplicación: programas que manejará el usuario para realizar cualquier tarea con el sistema informático.

Software del sistema

•Programa de arranque: Conjunto de tareas que se realizan en el ordenador en el momento de encenderlo.

-Chequeo y comprobación de dispositivos hardware.

-Lectura de parámetros de la BIOS y configuración de algunos dispositivos (memoria, discos...)

-Búsqueda del sistema operativo y carga del mismo en memoria.

•Sistema operativo (S.O.): Conjunto de programas que relacionan al usuario con el hardware y gestiona eficientemente los recursos.

•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

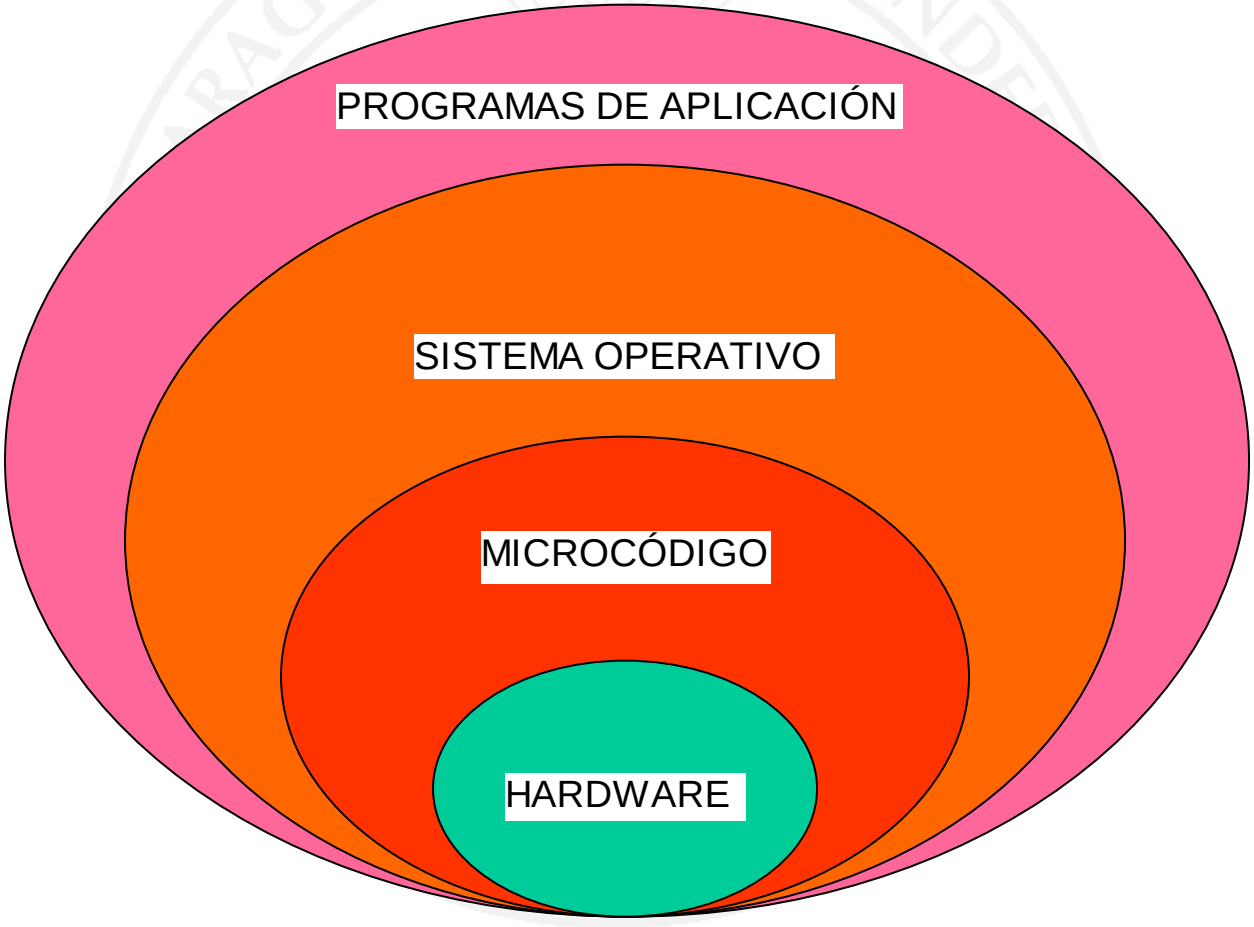
Programas de edición de textos.
Gestión de bases de datos (gestión de bibliotecas, de matriculados en la universidad, etc.)
Programas de administración y contabilidad.
Herramientas de diseño y edición gráfica (CAD-CAM)

El sistema operativo

- El sistema operativo es el conjunto de programas que administran los recursos del ordenador y permiten la comunicación con el usuario mediante un determinado interfaz.

- La función de un sistema operativo es básicamente la de establecer la comunicación lógica entre los diferentes elementos del ordenador para que puedan funcionar las diferentes aplicaciones, intentando mejorar la relación entre el usuario y la máquina y optimizando al máximo los recursos de todo el sistema.

Modelo por capas de sistema informático



Funciones del sistema operativo

- Administrar y controlar los dispositivos hardware: Asignación de memoria y gestión de los dispositivos del sistema.
- Hacer de interfaz entre el usuario y el hardware de la máquina.
- Gestión de almacenamiento: Mantener el sistema de archivos.
- Planificación, iniciación y supervisión de los programas. Apoyar la ejecución de otros programas.
  - Inicialización y control de las operaciones de entrada y salida.
  - Control de errores del sistema.
  - Coordinar la comunicación entre todos los elementos del sistema y entre el usuario y el sistema.
  - Si soporta multiproceso y tiempo compartido, gestionar estas operaciones.

Partes principales del sistema operativo

- 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.

  Generalmente formado por unas instrucciones u órdenes (muchas veces mal llamadas “comandos” del inglés *command*) que permiten al usuario realizar ciertas tareas específicas.

Características a exigir en un sistema operativo

- Eficiencia (rapidez en las tareas.)
- Fiabilidad (sin fallos ni errores.)
- Facilidad de mantenimiento.
- Compatibilidad con otros sistemas.
- Tamaño reducido (aunque últimamente esta característica está siendo sustituida por facilidad de manejo por parte de un usuario inexperto.)

Ejemplos de sistemas operativos

- DOS (Disk Operating System):
  - Sistema operativo para máquinas con microprocesadores Intel y compatibles (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.
  - Monousuario
  - Monoproceso
  - Apenas implementa características de seguridad.
  - Para mantener la compatibilidad con versiones anteriores mantiene ciertas características que impiden aprovechar los últimos avances de los PC's.

Ejemplos de sistemas operativos

- UNIX:
  - Sistema operativo de propósito general, desarrollado para multitud de plataformas hardware (en principio cualquier microprocesador, intentando sacar el máximo partido del hardware aunque no se mantenga la compatibilidad entre versiones.)
  - Esquema básico: Núcleo pequeño + Extensiones propias
  - Multiusuario
  - Multiproceso
  - Implementa características de seguridad
  - Cualquier sistema de entrada/salida (I/O) se trata como si fuese un archivo.

Ejemplos de sistemas operativos

- Windows 95/98.
  - Sistema operativo de propósito general, desarrollado para máquinas con microprocesadores Intel y compatibles (80386, 80486, Pentium,...)
  - Interfaz gráfica nativa.
  - Monousuario
  - Multiproceso
  - Implementa características de seguridad mínimas.
  - Se prima la facilidad de uso sobre otras características como la velocidad o la capacidad de control del usuario.
  - Para mantener compatibilidad con sistemas anteriores se sacrifican algunas de las ventajas posibles de los nuevos sistemas.

Ejemplos de sistemas operativos

- Windows NT/2000/Vista
  - Sistema operativo de propósito general, desarrollado para máquinas con microprocesadores Intel y compatibles (80386, 80486, Pentium,...)
  - Arquitectura de micronúcleo.
  - Interfaz gráfica nativa.
  - Multiusuario
  - Multiproceso
  - Implementa características de seguridad avanzadas.