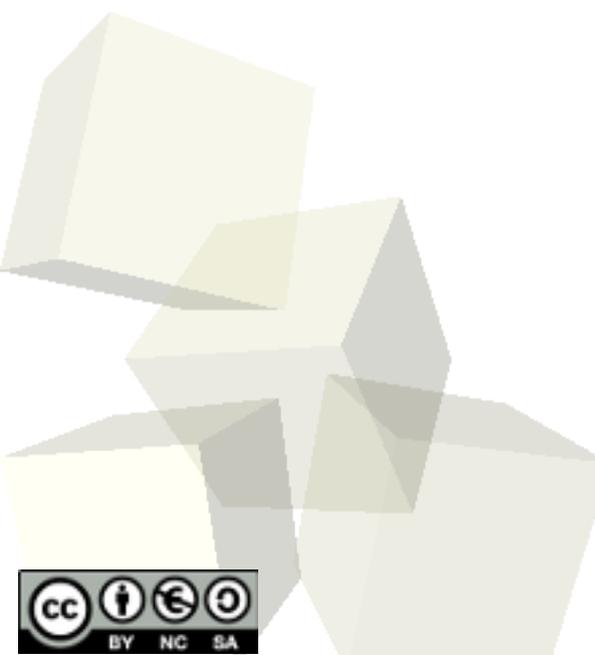


Tema 1

Introducción

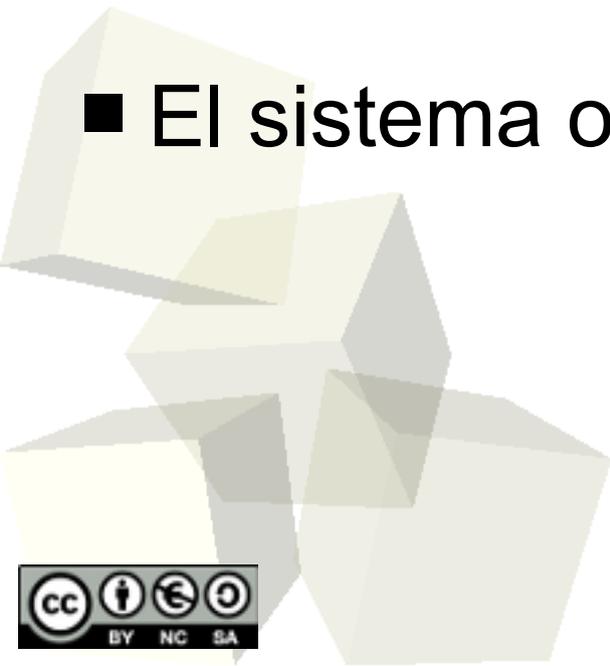
Informàtica
Grado en Física
Universitat de València

Ariadna.Fuertes@uv.es
Francisco.Grimaldo@uv.es

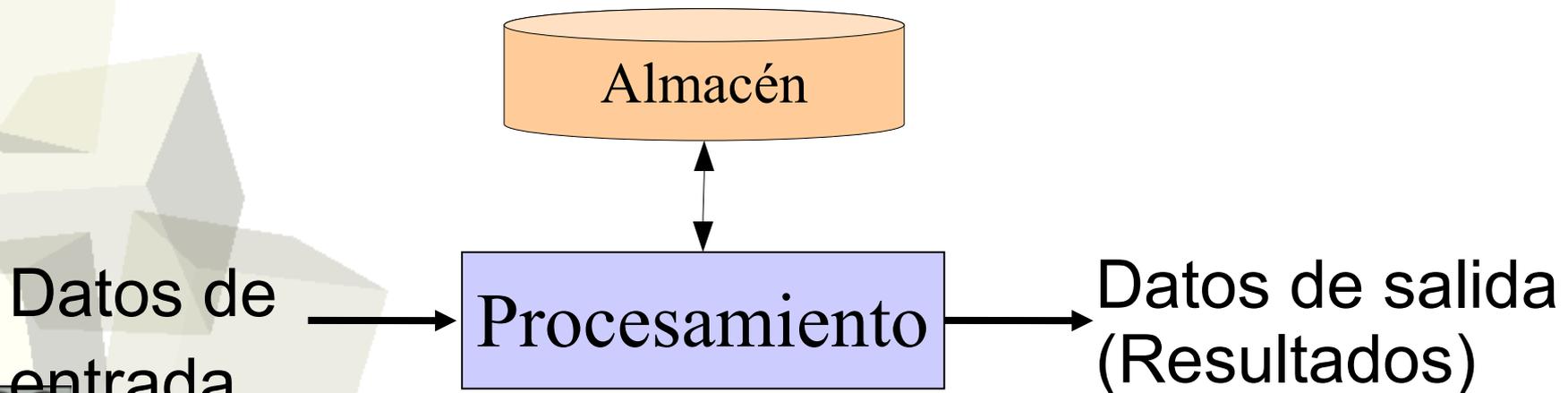




- **Conceptos básicos**
- Antecedentes
- Estructura interna del computador
- Lenguajes de programación
- El sistema operativo



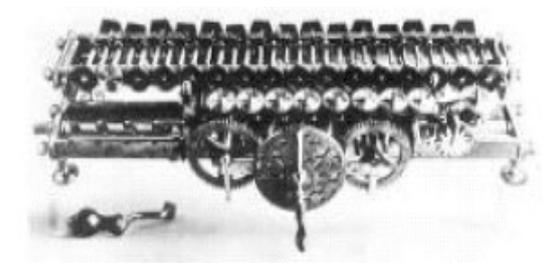
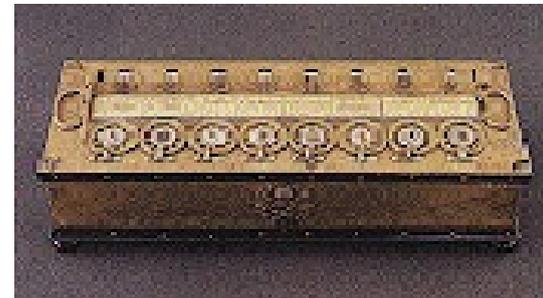
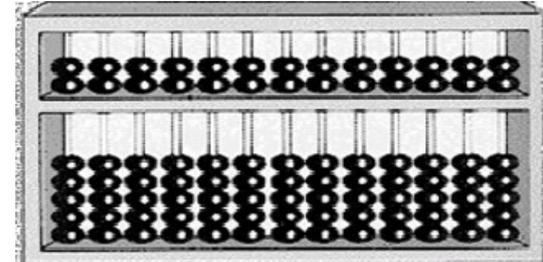
- Informática = **INFOR**mación + auto**MÁTICA**.
- **Informática**: disciplina científica encargada del tratamiento de la información mediante métodos automáticos.
- **Ordenador**: dispositivo electrónico programable capaz de almacenar, recuperar y procesar información.





- Primero fueron las máquinas de calcular...
- Antecedentes **mecánicos**:

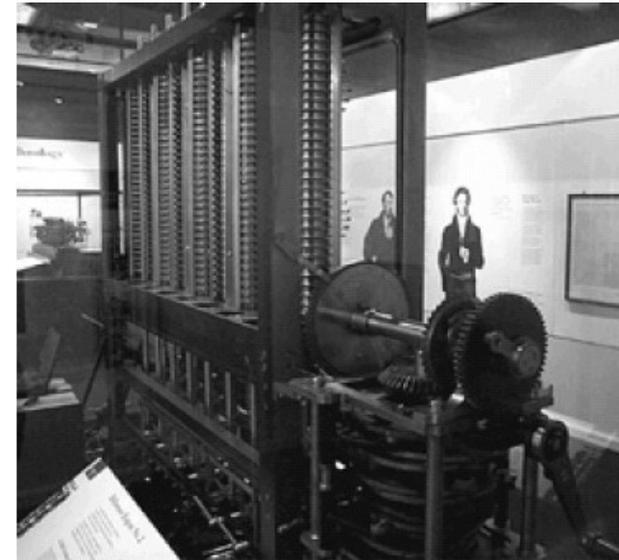
- **Siglo XII**: **el ábaco chino** es un dispositivo manual para la asistencia al cálculo (sumas y restas). Permite almacenar cantidades y operar sobre las cantidades representadas. El proceso de la información representada lo realiza el usuario.
- **Siglo XVII**: El científico francés **Blas Pascal** inventó la “**machina arithmetica**” en 1.642. Parece que Whilem Schickard inventó una más potente en 1623, año de nacimiento de Pascal.
- **Siglo XVII-XVIII**: El alemán **Leibnitz**, entre 1.671 y 1.694, desarrolla la “**calculadora universal**” que, además de realizar operaciones de **suma y resta**, podía efectuar **productos, cocientes y raíces cuadradas**.
- **Siglo XIX**: Comienza la comercialización industrial de las primeras máquinas de calcular.





■ Antecedentes **mecánicos**:

- **Siglo XIX:** El matemático inglés **Charles Babbage** diseñó la “**màquina analítica**”, que se podía programar con tarjetas perforadas para ejecutar **cualquier operación matemática**. Además disponía de una **memoria** que podía almacenar 1000 números de 50 cifras. **Lady Lovelace** concibió los primeros programas. **No se completó!**
- Con el desarrollo de la electrónica, se comienzan a reemplazar los sistemas de engranajes y varillas por **impulsos electromagnéticos**.
- Se establece la representación binaria: ***el paso de corriente eléctrica será representado como un 1 y cuando no haya paso de corriente eléctrica se representará como un 0.***
- **Siglos XIX y XX:** El cántabro **Leonardo Torres Quevedo** desarrollará diversas máquinas analógicas de cálculo basadas en relés.





■ Generaciones de computadores:

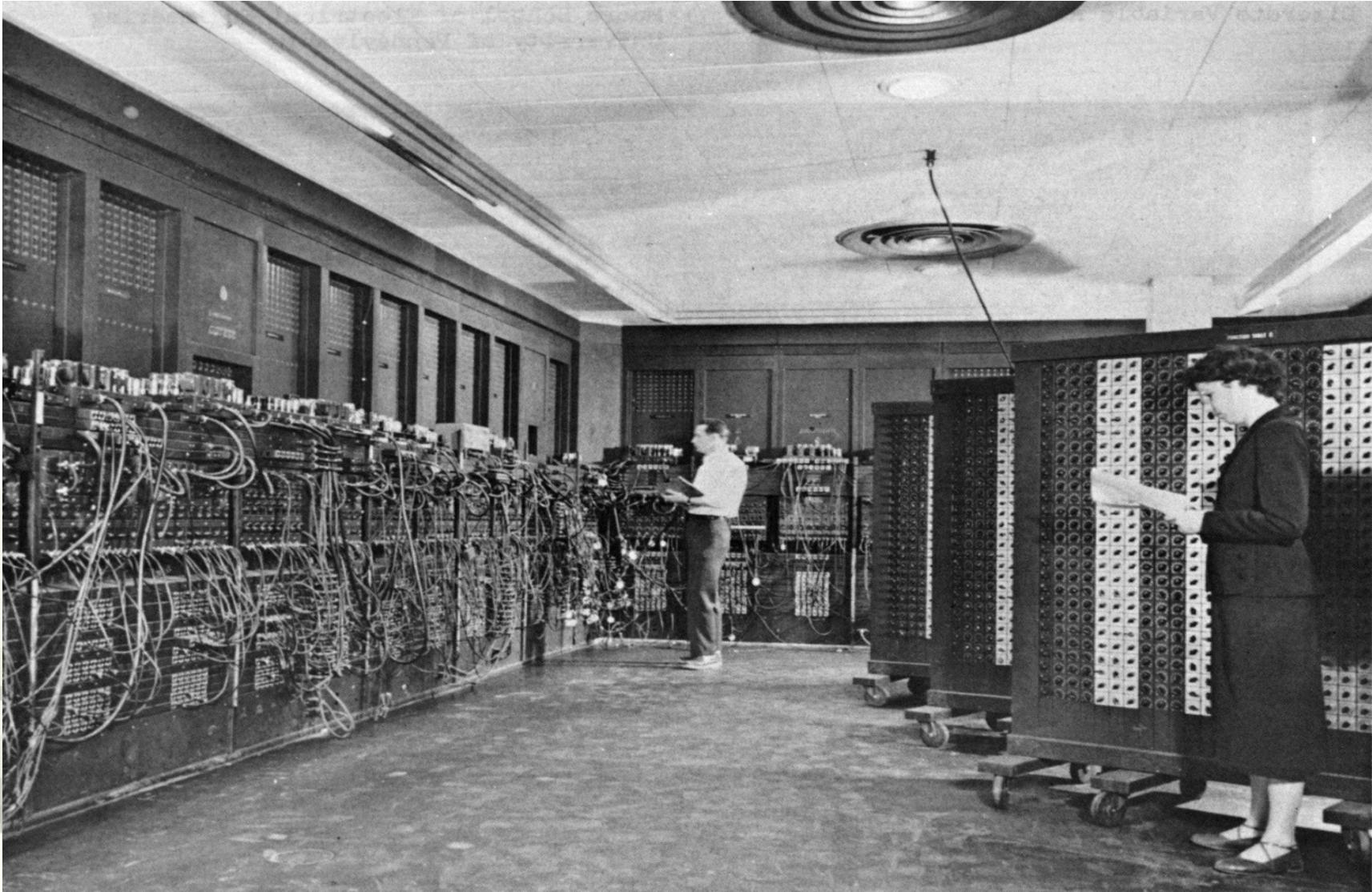
- ♦ **Primera generación (1940-1958):** Uso de **válvulas o tubos de vacío**. Aplicación en el ámbito científico, militar y gubernamental.
- ♦ Primeros ordenadores (relés):
 - **Z3** (1940) realizado per Konrad Zuse.
 - **Mark I** (1944) de la Univ. Harvard + IBM.
- ♦ **ENIAC** (1945) creado por Mauchly y Eckert (EEUU) con finalidad militar, pero llegaron tarde...
- ♦ **UNIVAC** (1950) realizado por Sperry tuvo mucho éxito y se utilizó en oficinas del censo de EEUU.
- ♦ **IBM 705** (1952) fue la contraoferta de IBM.
- ♦ **UNIVAC II** (1958) no triunfó porque el mercado era de IBM.





Antecedentes (4/8)

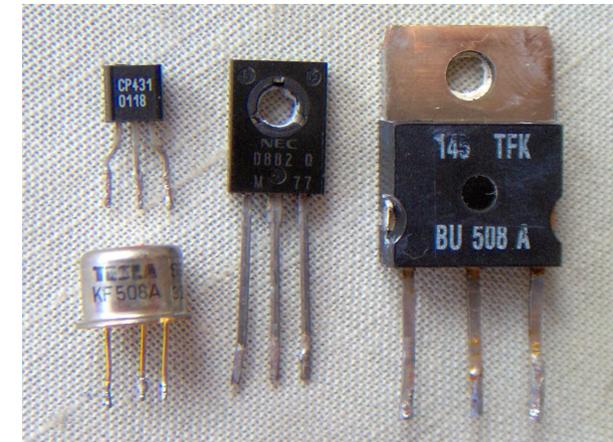
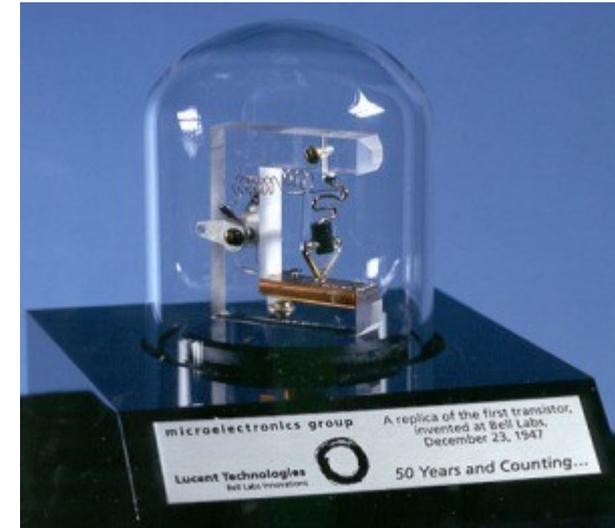
- Grandes, caros y propensos a fallos.

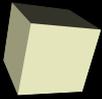




■ Generaciones de computadores:

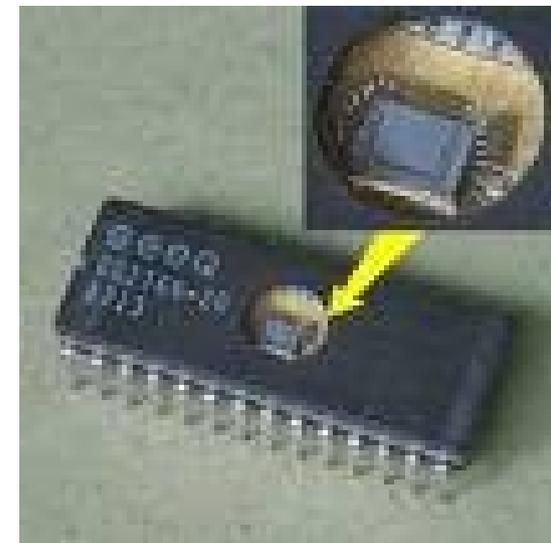
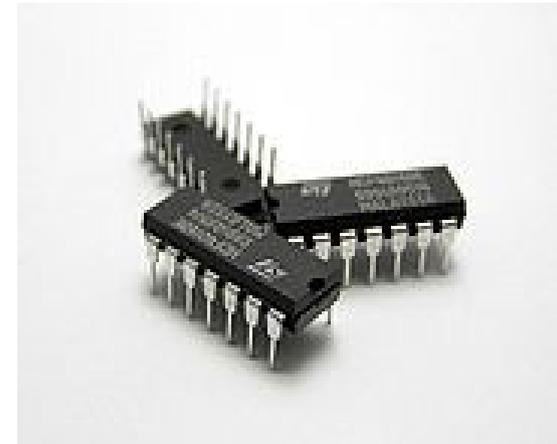
- **Segunda generación (1958-1964):** Aparece el **transistor**. Aplicación al ámbito comercial.
- **IBM 7070** (1960) fabricado por IBM.
- **UNIVAC 1107** (1962) de Sperry Rand Co.
- Dotados de una programación previa (**Sistema operativo**).
- **Lenguajes de programación:** FORTRAN (1954), COBOL (1959), LISP (1962).
- Programa → Lenguaje máquina → Ejecución.
- También surgen: unidades de **cinta**, **discos magnéticos**, **lectores de tarjetas** perforadas, **impresoras** de alta velocidad...





■ Generaciones de computadores:

- **Tercera generación (1964-1971):** Uso del **circuito integrado (chip)**.
- Reducción de **costes y tamaño físico** y aumento de la **capacidad** de procesamiento.
- **IBM 360 (1964) y 370 (1970)** fabricados por IBM.
- **Serie 1100 (1964)** de Sperry Rand Co.
- **Estandarización** de los lenguajes de programación: FORTRAN (1966), COBOL (1970)...
- **Nuevos lenguajes:** BASIC (1964), PASCAL (1971).
- A finales de los 60 aparecen los **miniordenadores**, con circuitos integrados de mediana escala usados por empresas medianas y grandes (p. ej. PDP 11 de DEC).





■ Generaciones de computadores:

- ♦ **Cuarta generación (1971-1981):** Integración de los componentes electrónicos en un circuito integrado.
- ♦ **Microprocesador:** Integración de todos los componentes básicos de un ordenador en un chip.
- ♦ **Intel 4004** (1971) fue el primer microprocesador.
- ♦ Aparecen los **ordenadores personales (PC):**
 - **Kentelek 8** (1973) de Distesa (editorial Anaya).
 - **Altair 8800** (1974) de MITS (EEUU).
 - **PET 2001** de Commodore.
 - **Apple II** de Steven Jobs y Stephen Wozniak.
 - **Sinclair ZX80** de Spectrum → Amstrad.

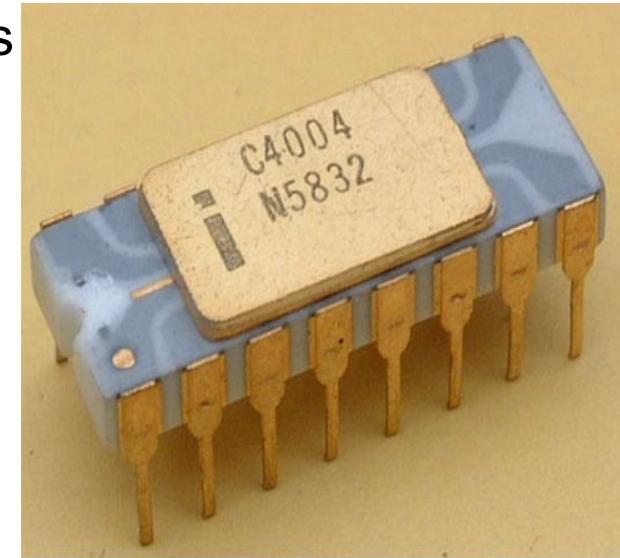
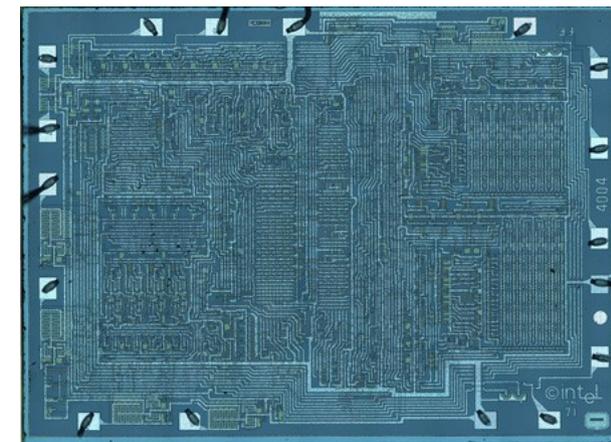


Image courtesy of CPU-Zone.com. Used with permission.





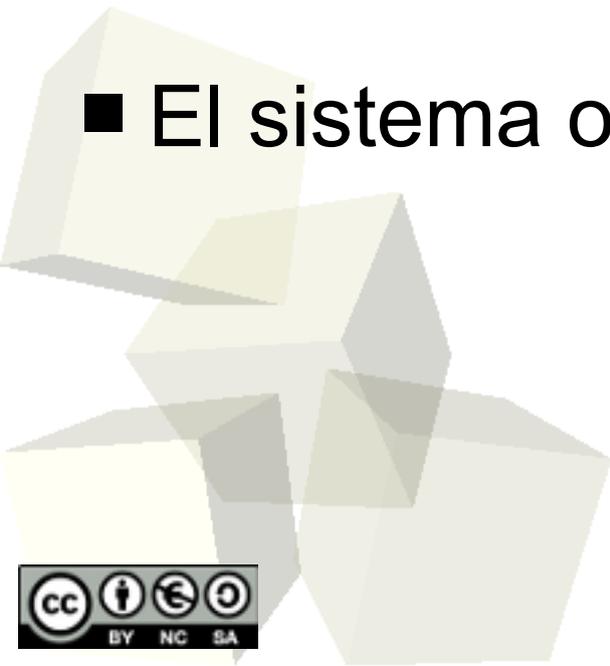
■ Generaciones de computadores:

- ♦ **Quinta generación (1981-?)**: Desde los primeros microprocesadores hasta los de última generación.
- ♦ Chips con **múltiples procesadores**:
 - Intel Core 2 Duo.
 - Dual Core AMD Opteron.
 - Intel Core i7
- ♦ **Comparad:**
 - Calculadora → 1r computador (3 siglos)
 - Mark I → 1r microordinador (28 años)





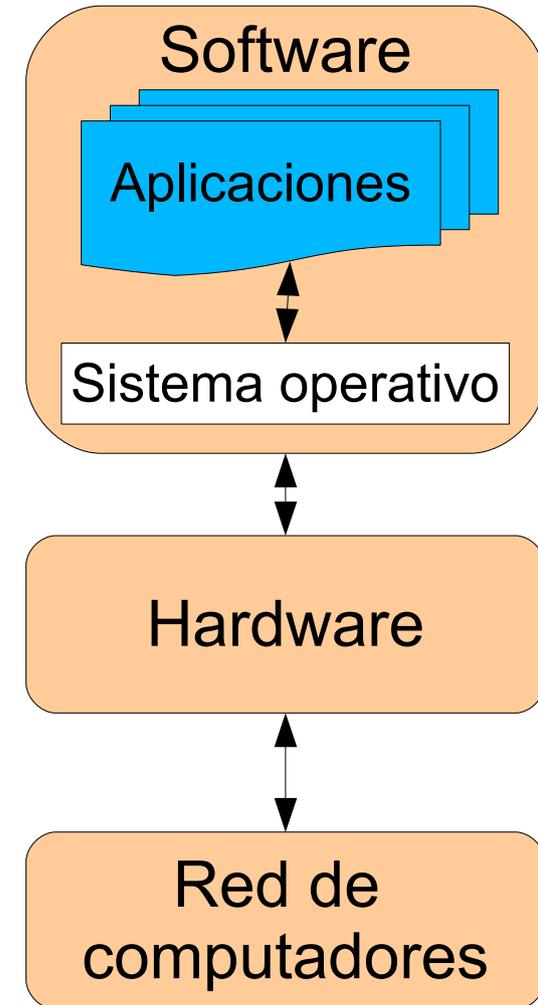
- Conceptos básicos
- Antecedentes
- **Estructura interna del computador**
- Lenguajes de programación
- El sistema operativo





Conceptos básicos

- **Hardware:** Conjunto de elementos físicos de un sistema informático, que generalmente se compone de una unidad central y de periféricos.
- **Software:** Soporte lógico de un ordenador. Incluye el software de sistema (sistema operativo) y el de aplicación (p. ej. procesador de textos...)
- **Red de computadores:** Conjunto de dispositivos interconectados que comparten información (archivos), recursos (impresoras) y servicios (acceso a Internet, e-mail...)



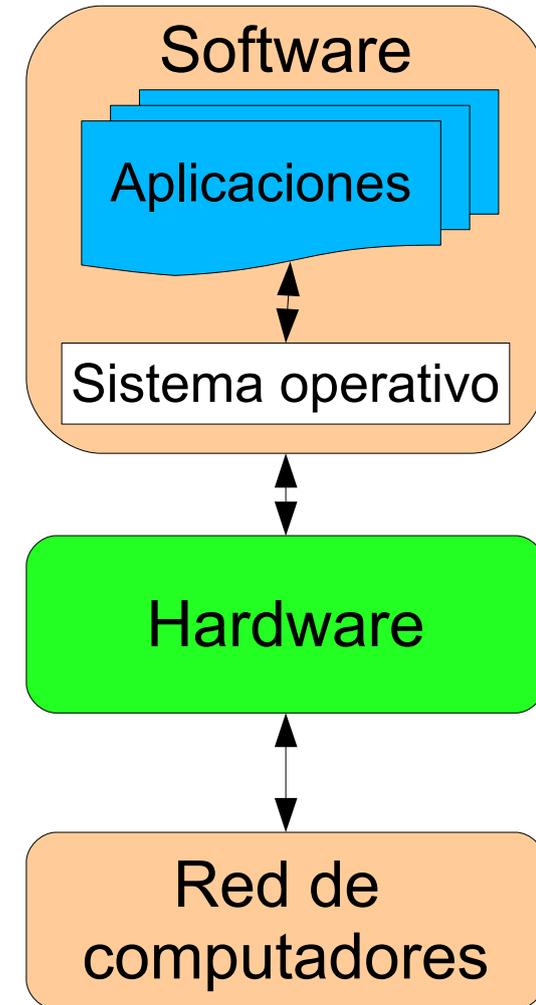
Estructura interna del computador

■ Arquitectura de Von Newman (1945):

- ♦ Se trata de un diseño conceptual que especifica cómo debería funcionar una máquina programable capaz de procesar información.

■ Características:

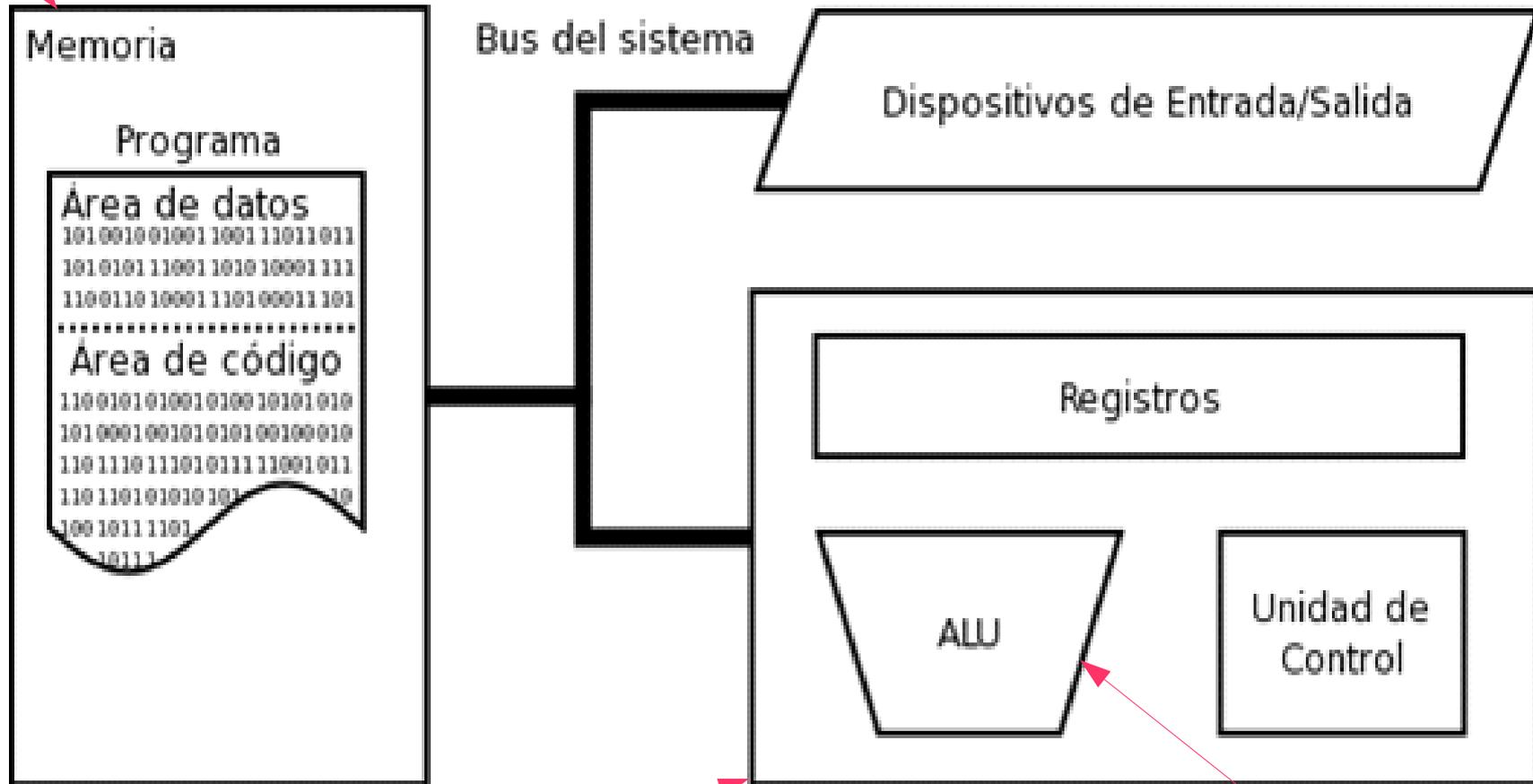
- ♦ La máquina debe ser controlada por un conjunto de instrucciones con un pequeño número de elementos centrales de proceso.
- ♦ Tanto la información (datos) como el programa (método de procesado de la información) deben almacenarse en el interior del computador en formato binario (con un alfabeto compuesto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





Unidades funcionales

Datos + Instrucciones



Unidad central de procesos

Unidad aritmético-lógica





La unidad central de procesos

- Dos subunidades con funciones 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.



- La memoria es el dispositivo que se encarga de **almacenar** toda la información.
- En la arquitectura de Von Neumann, **tanto los datos como las instrucciones** para procesarlos se almacenan internamente.
- El contenido de la memoria se almacenará codificado en forma de **bits (0 ó 1)**.

| | |
|---|---|
|  |  |
| VALOR 1 | VALOR 0 |



0 0



0 1



1 0



1 1



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 \approx 1.000 bytes
 - ♦ 1 **Megabyte** (1 Mb) = 2^{20} bytes = 10.48.576 bytes \approx 1₁.000.000 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 1.000.000.000.000 bytes

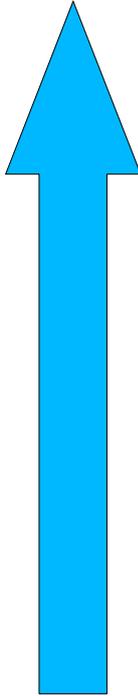


Tipos de memoria (1/2)

- Según su tamaño y rapidez, distinguimos:
 - ♦ **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 (2/2)

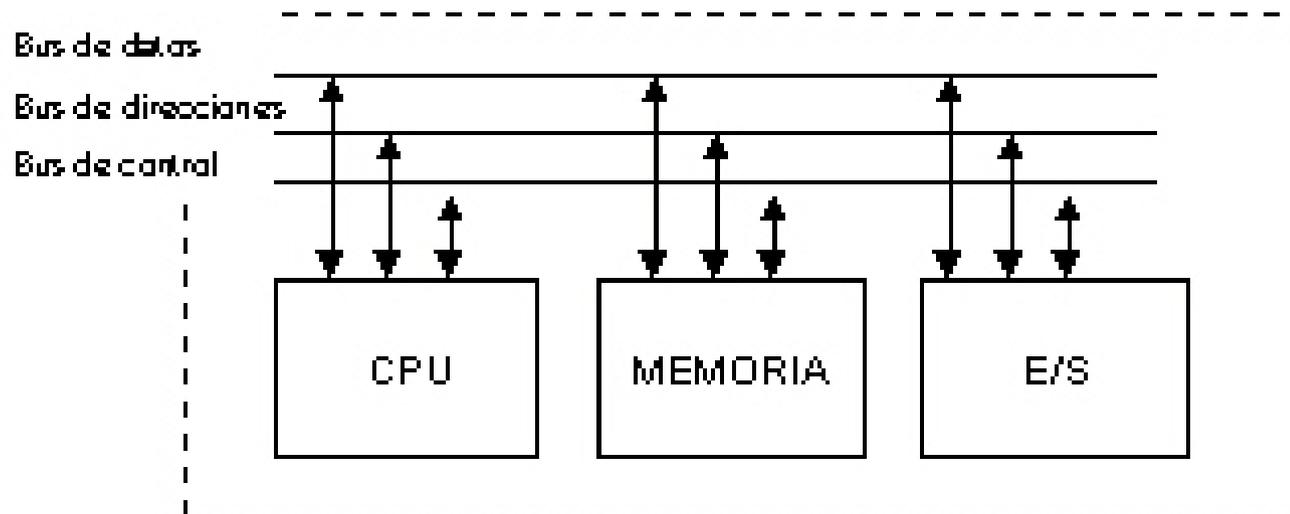
| Tipo | Orden de magnitud | Rapidez |
|------------|-------------------|--|
| Registro | <i>Bytes</i> | + |
| Caché | <i>Kb</i> |  |
| R.A.M. | <i>Mb</i> | |
| Disco duro | <i>Gb</i> | |
| | | |

- A) Se **carga 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**



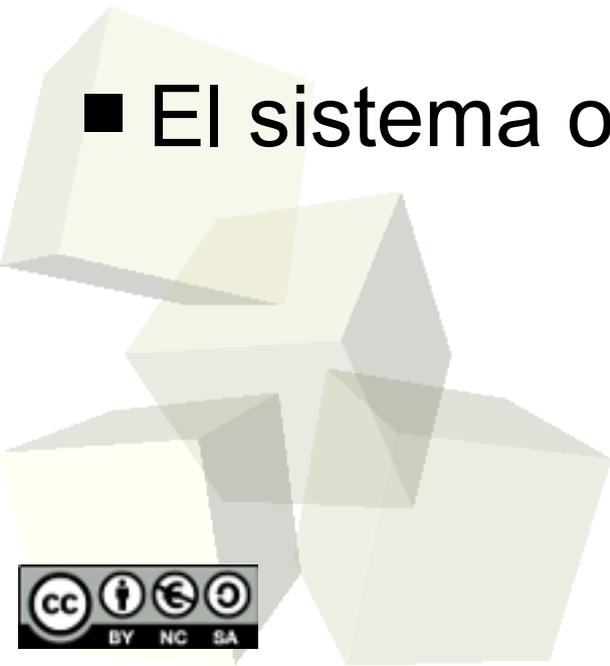
Buses y señales de control

- Las unidades funcionales se comunican mediante 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





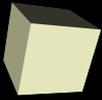
- Conceptos básicos
- Antecedentes
- Estructura interna del computador
- **Lenguajes de programación**
- El sistema operativo



Concepto de algoritmo y programa

- 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.

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.



Lenguaje de programación

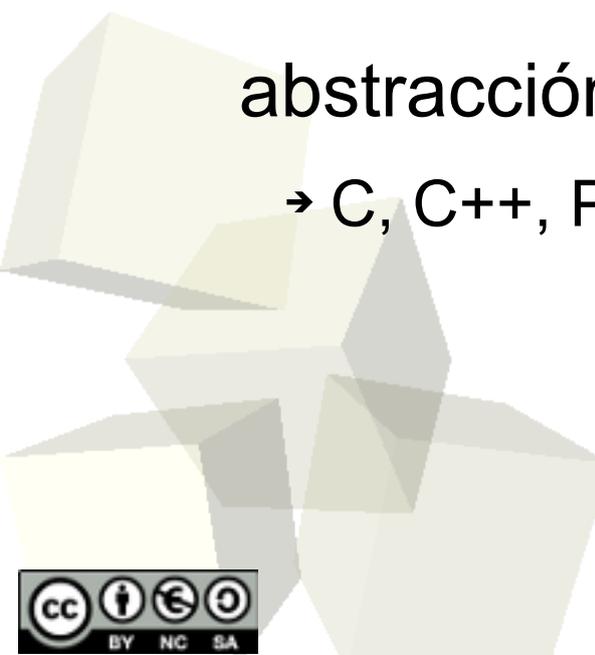
- Un **lenguaje de programación** es un conjunto controlado de palabras con una sintaxis y una semántica asociadas que permiten describir algoritmos de forma que puedan ser ejecutados por un ordenador.
- Todo lenguaje de programación deberá tener:
 - ♦ Un **vocabulario** limitado (palabras reservadas).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



Tipos de lenguajes (1/5)

■ 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 y tienen un alto nivel de abstracción.
 - C, C++, Pascal, Fortran, etc.





■ 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 intérprete. 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 (3/5)

■ Según su propósito:

- ♦ De **propósito general**: BASIC, C, Pascal, Java,...
- ♦ De **enseñanza**: Logo, Pascal...
- ♦ De **cálculo científico**: Algol, Fortran, Matlab...
- ♦ De **gestión**: Cobol, Modula.
- ♦ De **gestión de bases de datos**: Informix, SQL, dBASE, Oracle.
- ♦ Usados en **Inteligencia Artificial**: Prolog, Lisp...



Tipos de lenguajes (4/5)

■ Según su paradigma de programación:

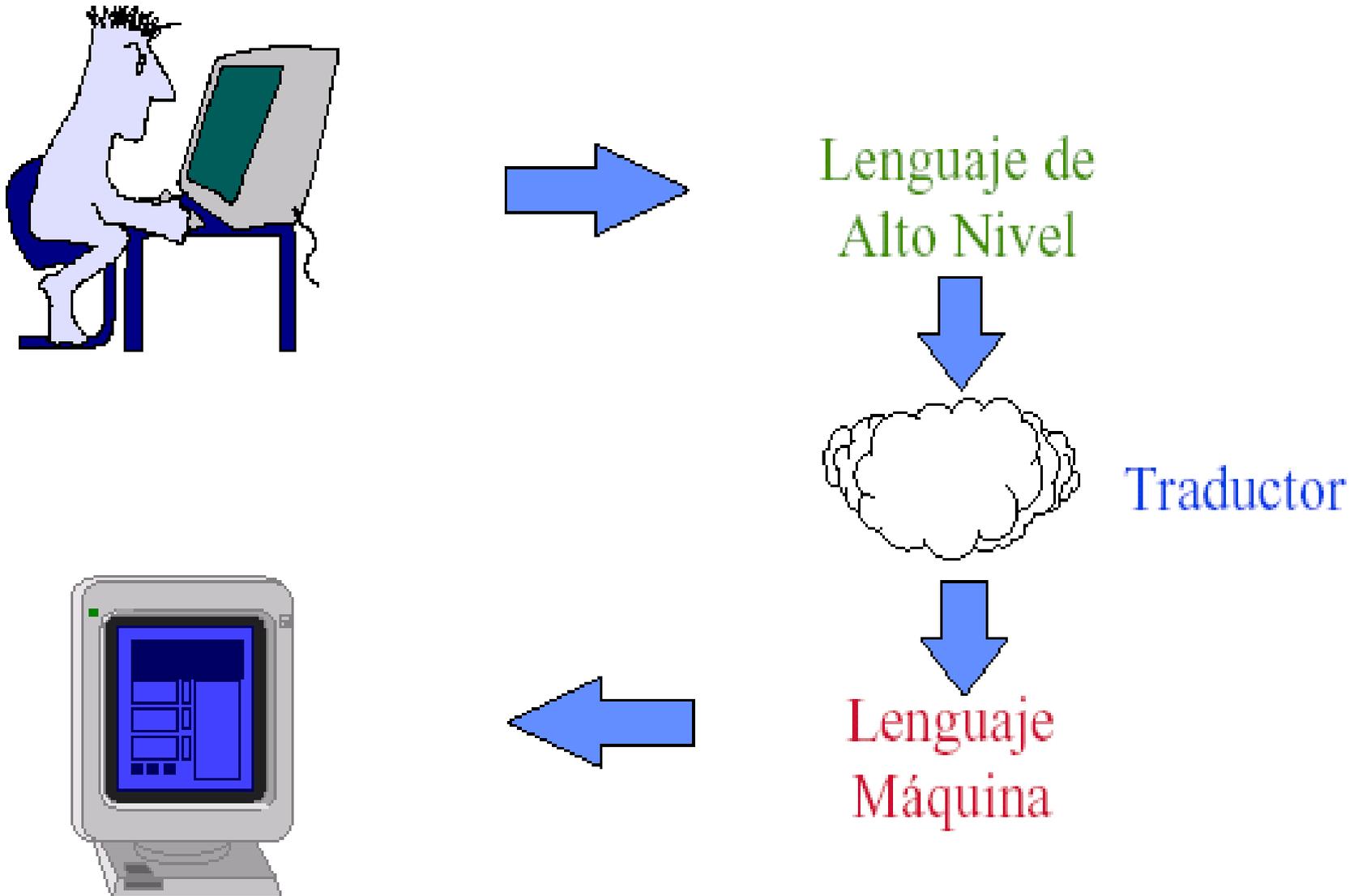
- ♦ **Lenguajes procedurales/procedimentales o imperativos** (p. ej. C, Pascal, BASIC):
 - Un programa es una secuencia de acciones que se ejecutan en un orden controlado.
 - PROGRAMA = PROCEDIMIENTOS + ALGORITMOS.
- ♦ **Lenguajes declarativos** (p. ej. Prolog):
 - Un programa es un conjunto de hechos y relaciones entre éstos. No existe una secuencia definida de ejecución.
 - PROGRAMA = DEF_ELEMENTOS + RELACIONES.
- ♦ **Lenguajes orientados a objetos** (p. ej. C++, Java):
 - Un programa está constituido por un conjunto de objetos 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
 - PROGRAMA = OBJETOS + MÉTODOS + MENSAJES.



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

Proceso de programación... (1/3)

- ... en un lenguaje de alto nivel:

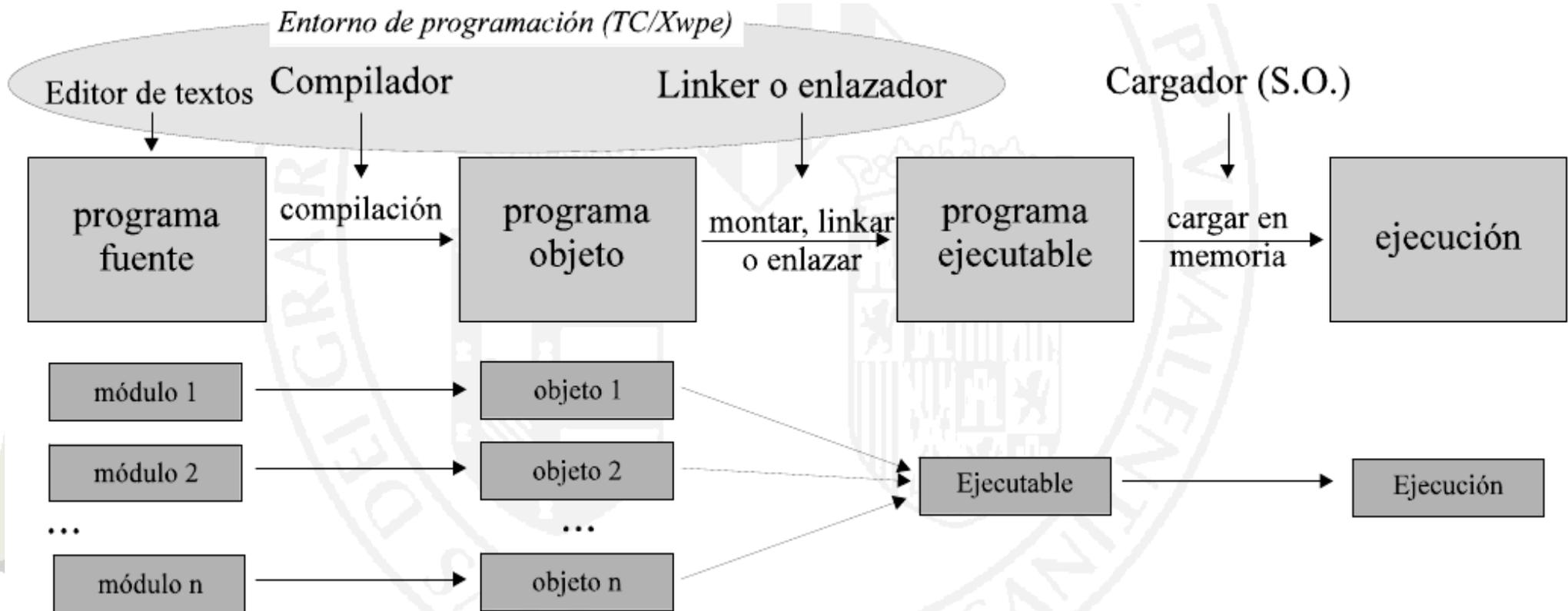


- ... en un lenguaje de alto nivel compilado:
 - ♦ **Edición**: Escribir el programa fuente en un editor de textos. Este programa puede constar de uno o más módulos.
 - ♦ **Compilación**: Usar un compilador para realizar un análisis léxico y sintáctico del programa fuente. El resultado será un módulo 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... (3/3)

- ... en un lenguaje de alto nivel compilado:



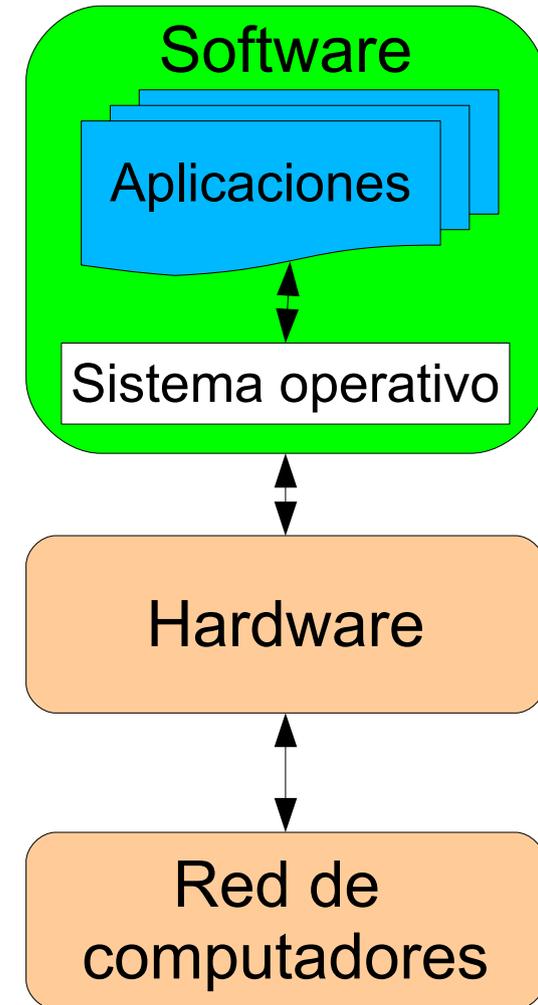


- Conceptos básicos
- Antecedentes
- Estructura interna del computador
- Lenguajes de programación
- **El sistema operativo**



Definición de Sistema Operativo

- El **Sistema Operativo** (SO) es el conjunto de programas que administran los recursos del ordenador y permiten la comunicación con el usuario mediante una determinada **interfaz**.
- **Interfaz de usuario**: Mecanismo mediante el que el usuario establece la comunicación con el ordenador:
 - ♦ Interfaz **textual**: introducción de comandos de texto en la “línea de comandos” (p. ej. MS-DOS, UNIX).
 - ♦ Interfaz **gráfica**: entornos de ventanas (p. ej. Windows, Mac i Linux).





Funciones del SO (1/2)

- Las **funciones fundamentales** del SO son:
 - ♦ Comunicar los diferentes elementos del ordenador.
 - ♦ Relacionar al usuario con la máquina.
 - ♦ Optimizar al máximo los recursos del sistema.
- **Funciones concretas:**
 - ♦ Administración y control de los dispositivos periféricos.
Ej: monitores, impresoras, discos...
→ **Drivers** o controladores.
 - ♦ Gestión de múltiples tareas o procesos (**multitarea** o **multiproceso**).
 - ♦ Administración de **memoria**: Un mundo de procesos...
→ Memoria virtual.



- **Funciones concretas:**
 - ♦ Planificación, inicio y supervisión de los **programas**.
 - ♦ Gestión de **almacenamiento**: Mantenimiento del sistema de archivos.
 - ♦ Control de **errores** del sistema.
 - ♦ Coordinación de las **comunicaciones** en red.
 - ♦ Gestión de múltiples usuarios (**multiusuario**).
- El SO no descansa **nunca!**



Características deseables

- **Eficiencia:** Rapidez en las tareas.
- **Fiabilidad:** Ejecución sin fallos ni errores.
- **Interfaz** hombre-máquina.
- **Facilidad** de uso y de mantenimiento.
- **Compatibilidad** con otros sistemas.
- **Tamaño** reducido:
 - ♦ Últimamente es más importante la facilidad de uso por parte del usuario.



SSOO: MS-DOS (1/2)

- **Microsoft** Disk Operating System (1981).
- Origen: QDOS (Quick and Dirty Operating System) de Tim Paterson.
- SO de los IBM **PC y compatibles**.
- **Procesadores**: 80806, 80286, 80386, 80486, Pentium...
- **Monousuario**.
- **Monoproceso**.
- **No implementa medidas de seguridad**.





SSOO: MS-DOS (2/2)

```
C:\>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5840-6173

Directorio de C:\

17/06/2005  19:02    <DIR>          3ImpactWork
09/10/2005  22:56    <DIR>          Archivos de programa
03/05/2005  20:50    <DIR>          ATI
10/08/2005  16:35    <DIR>          audio
22/03/2005  22:03    <DIR>          @ AUTOEXEC.BAT
02/09/2005  21:08    <DIR>          bb
21/00/2005  11:39    <DIR>          COBUILD
22/03/2005  22:03    <DIR>          @ CONFIG.SYS
03/09/2005  23:00    <DIR>          D72
24/08/2005  13:52    <DIR>          Dev-Cpp
08/05/2005  16:13    <DIR>          ? 741.336 Bio8521MP2K.exe
22/03/2005  22:49    <DIR>          Documents and Settings
15/06/2005  21:24    <DIR>          ENCORE
11/05/2005  21:59    <DIR>          localtxmf
14/08/2005  00:53    <DIR>          MPEGSuite
22/05/2005  23:24    <DIR>          Program Files
03/05/2005  22:22    <DIR>          Programas
11/08/2005  16:30    <DIR>          Slap
27/07/2005  15:44    <DIR>          sunset
08/07/2005  21:46    <DIR>          txmf
15/07/2005  10:52    <DIR>          UnrealTournament
03/09/2005  23:09    <DIR>          WINDOWS
03/05/2005  21:36    <DIR>          WUTemp
          3 archivos      7.741.336 bytes
          20 dirs    19.520.974.040 bytes libres

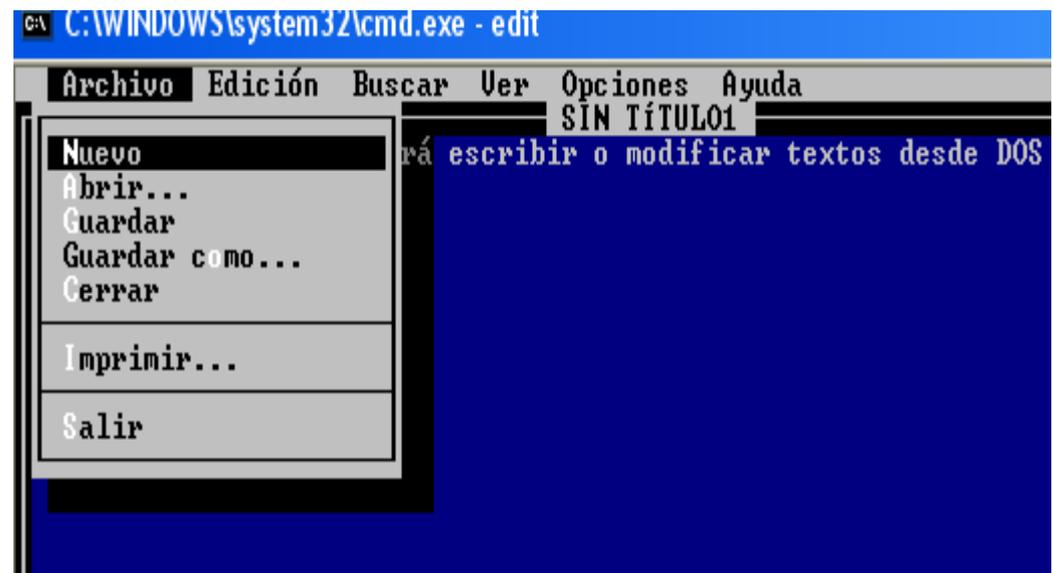
C:\>_
```

■ Interfaz de línea de comandos:

- ♦ El *prompt* indica al usuario que el sistema puede aceptar una nueva orden o instrucción.

■ Interfaz de menús:

- ♦ Interfaz pseudo-gráfica en modo texto.





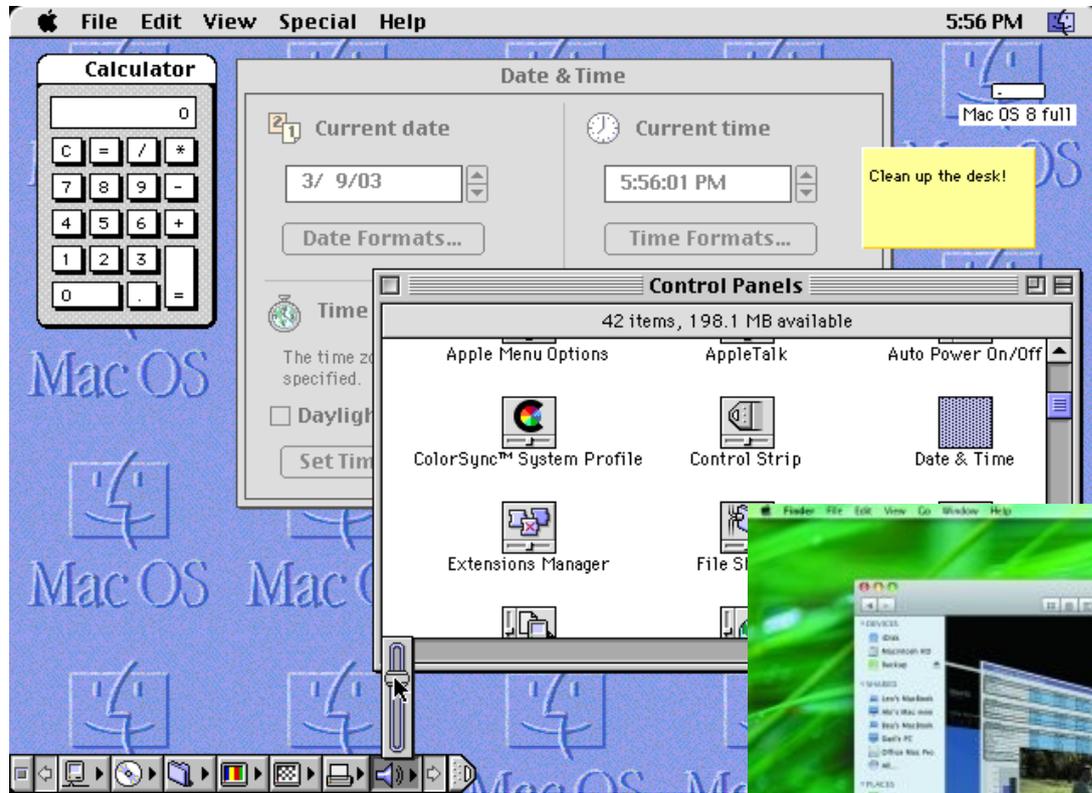
SSOO: MAC OS (1/2)

- Apple **Macintosh** (1984).
- Origen: SO Alto de Xerox (1973).
- **GUI** (*Graphical User Interface*).
- **Paradigma gráfico:** Escritorio, iconos, ventanas, carpetas (directorios), documentos (ficheros)...
- **Procesador:** IBM/Motorola Power PC (Plataforma Mac).
- Debajo de las ventanas hay un SO de tipo **UNIX**.
- Últimas **versiones:** Mac OS 9, Mac OS X (10).





SSOO: MAC OS (2/2)



- Mac OS 8 (1997).



- Mac OS X Leopard.





SSOO: Windows (1/3)

- Desarrollado a partir de 1985.
- Primeras versiones (**Windows 3.11, 95 i 98**):
 - ♦ Shell gráfica para MS-DOS.
 - ♦ Monousuario.
 - ♦ Multiproceso.
 - ♦ Características de seguridad mínimas.
- Se prefiere la **facilidad de uso** frente a la velocidad o capacidad de control del usuario.
- Se sacrifican ventajas posibles para mantener la **compatibilidad** con sistemas anteriores.





SSOO: Windows (2/3)

■ Últimas versiones:

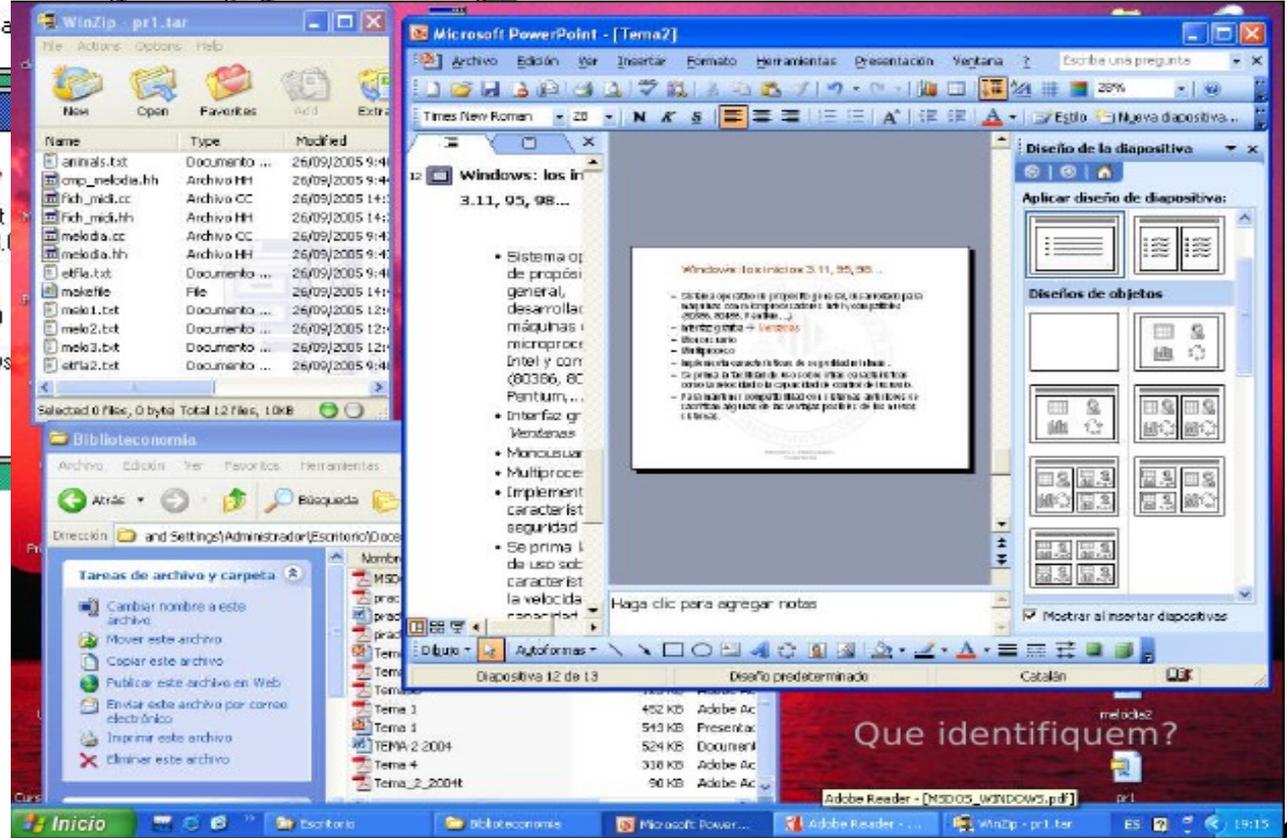
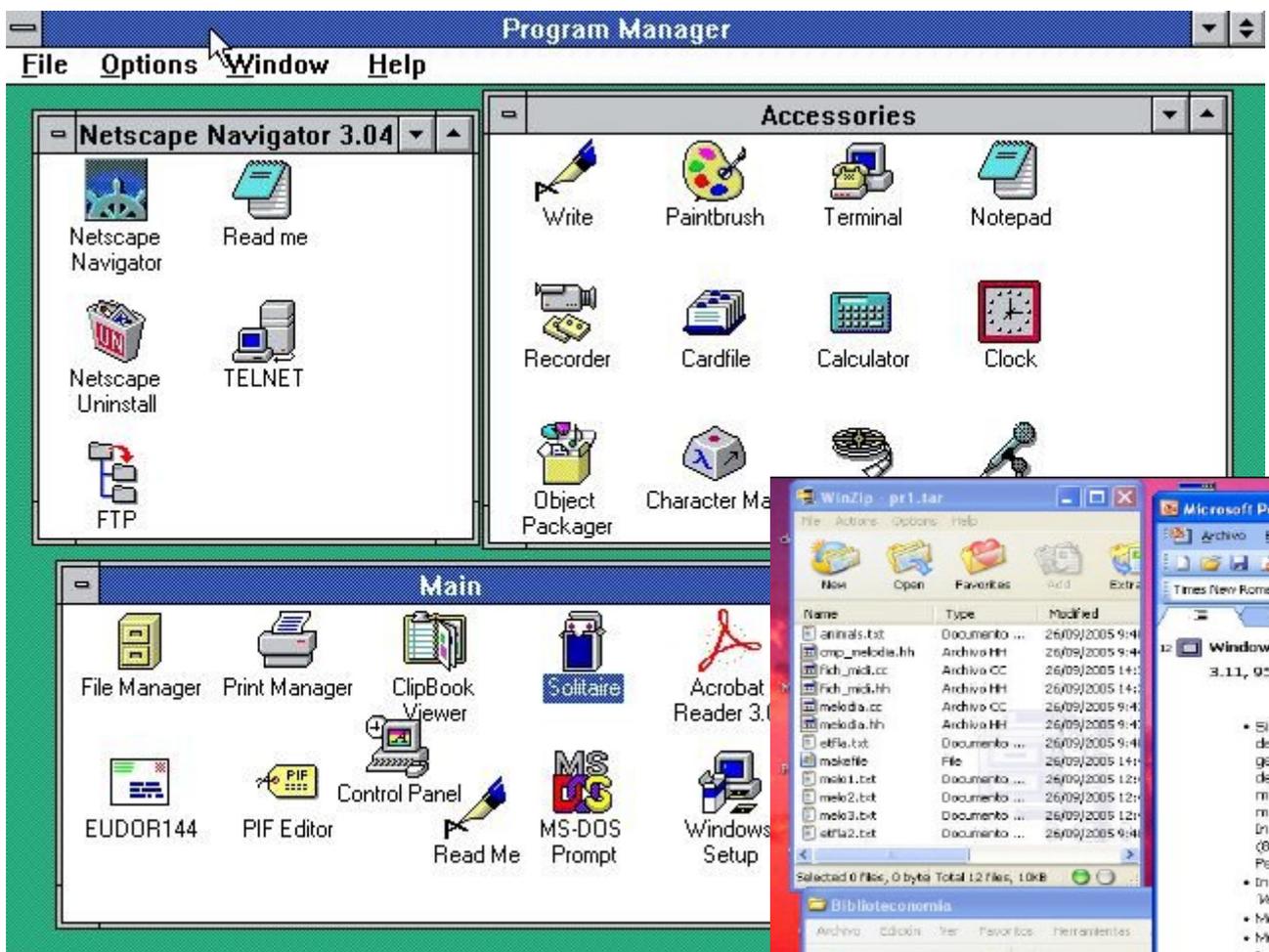


- ◆ Windows NT, 2000, XP, Vista, 7
 - ◆ Arquitectura de micronúcleo (*kernel*) de SO.
 - ◆ Multiusuario.
 - ◆ Multiproceso.
 - ◆ Características de seguridad avanzadas.
- La empresa Microsoft no publica el código fuente.
- Por tanto es un SO **privado y no se puede modificar**, a diferencia de Linux (Software libre).



SSOO: Windows (3/3)

Windows 3.11



Windows XP



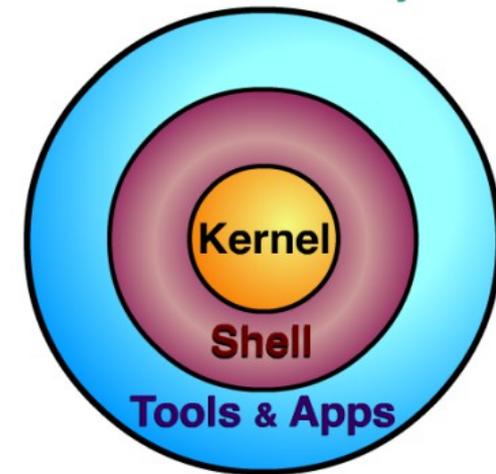


SSOO: UNIX (1/2)

- Creado ~1970 en los laboratorios Bell AT&T.
- SO para *mainframes* y *supercomputadores*.
- **Multiusuario** (SO de red).
- **Multiproceso** (muchas tareas a la vez).
- Características de **seguridad avanzadas**.
- SO de línea de comandos (**núcleo + GUI**).
- **Multiplataforma**: Funciona sobre cualquier procesador.
- **Utilizado por sistemas grandes** con muchos usuarios y programas (Universidades, centros de investigación...)



Parts of the UNIX System



SSOO: UNIX (2/2)

- **Versiones comerciales:** Solaris (Sun), HP-UX (Hewlett Packard), AIX (IBM)... y **MAC OS!!**

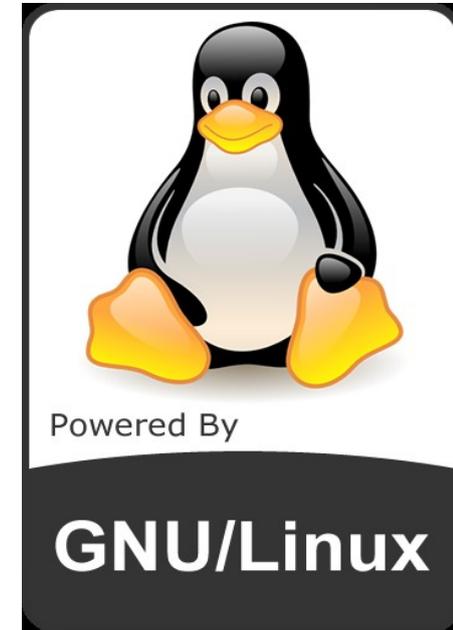


■ **Solaris**

■ **AIX**

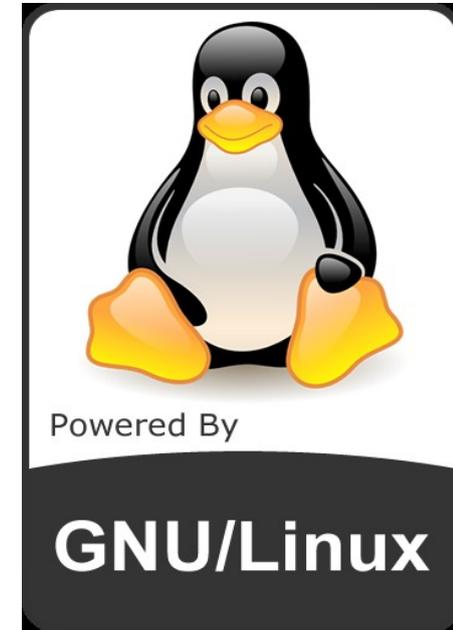


- Creador Linus Torvalds (a partir de 1991).
- SO de proposito general basado en UNIX (Minix) para plataformas de bajo coste (PC).
- **Denominación GNU/Linux:** GNU's not Unix / Linux.
- GNU: Proyecto de SO Unix de software libre.
- El código GNU/Linux se puede leer, modificar, intercambiar y volver a distribuir.
- **GPL** (General Public License).
- Software libre (**Open Source**) vs Gratis.
- Buen ejemplo de informática distribuida.





- **Multiusuario.**
- **Multiproceso.**
- Características de **seguridad avanzadas.**
- **Multiplataforma.**
- Arquitectura de **núcleo + shell gràfica.**
 - Ej: KDE y Gnome.
- GUI similar a Windows: escritorio(s), explorador, ventanas, iconos, botones.
- Incorpora muchas aplicaciones ofimáticas multimedia y de comunicaciones.





■ Palm OS:

- ♦ Palm, Handspring, Sony...
- ♦ Mòbils i PDA's.
- ♦ Facilidades de comunicaci3n.



■ Microsoft Windows CE.NET:

- ♦ Windows reducido.
- ♦ Sistemas incrustados.
- ♦ Tel3fonos m3viles.
- ♦ Computadores de bolsillo.



Bibliografía y ejercicios

- Temas 0, 1 y 4 del libro:
 - ♦ **Introducción a la Informática.** George Beekman. Pearson- Prentice Hall.
- Tema 1 de los **apuntes** de la antigua asignatura de “**Técnicas informáticas**”.
- **Free Software Foundation:**<http://www.fsf.org>

- **Ejercicios:**
 - ♦ Test Verdadero/Falso del Tema 1.
 - ♦ Test multiopción del Tema 1
 - ♦ Test Verdadero/Falso del Tema 4.
 - ♦ Test multiopción del Tema 4.
- Utilizad la hoja de respuestas disponible en el **Aula Virtual.**