



FUNDAMENTOS INFORMÁTICOS INGENIERÍA DE EDIFICACIÓN

TEMA 1 ELEMENTOS DE UN ORDENADOR

BLOQUE 2
SOFTWARE

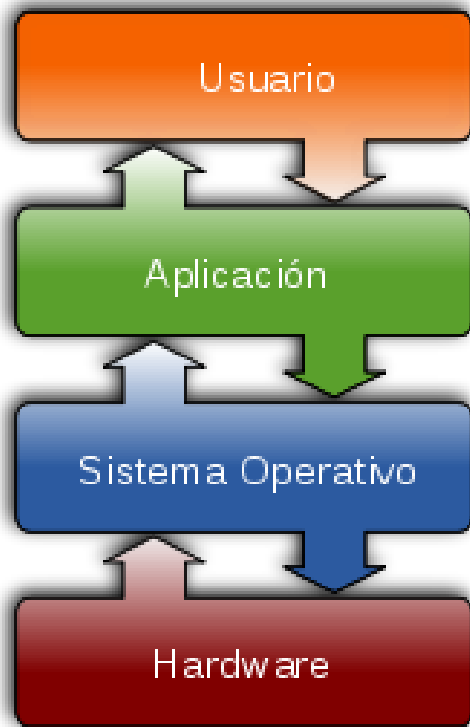


ÍNDICE

1. SISTEMAS OPERATIVOS
2. APLICACIONES INFORMÁTICAS
3. LENGUAJES DE PROGRAMACIÓN

Software

1. SISTEMA OPERATIVO



- Procesos informáticos que actúan de interfaz entre los dispositivos *hardware* y las aplicaciones de usuario.
- El SO se inicia al encender el ordenador. Es la única forma de dar utilidad al *hardware* y de interactuar con él.
- Gestiona y coordina:
 - El arranque del ordenador
 - Todas las tareas (procesos, programas) que se llevan a cabo en el ordenador,
 - los sistemas de seguridad y protección, los sistemas de archivos y de comunicaciones,
 - los recursos: procesadores, memoria RAM, dispositivos de almacenamiento y de entrada/salida.

Software

1. SISTEMA OPERATIVO. COMPONENTES PRINCIPALES

- Interfaz de usuario
 - Gráfico o línea de comando
- Núcleo del sistema (*kernel*)
 - Manejo de interrupciones, de trabajo de la CPU y de comunicación
- Administrador de memoria
 - Asigna posiciones en la RAM a los programas
- Sistema de entrada/salida (E/S)
 - Muestra al usuario la E/S de datos independiente del dispositivo
- Administrador de archivos
 - Supervisa la creación, actualización y eliminación de archivos
- Administrador de procesos
 - Asigna tiempos y organiza los procesos evitando conflictos entre ellos

Software

1. SISTEMA OPERATIVO. ASPECTOS IMPORTANTES

- Seguridad
 - Integridad, disponibilidad y confidencialidad de los datos.
- Eficiencia
 - Rendimiento incluso con grandes cantidades de información
- Estabilidad
 - Tolerancia a fallos de componentes *hardware* o *software*
- Administración
 - Políticas de uso de los recursos
- Flexibilidad
 - Plataformas múltiples (arquitecturas, procesadores, periféricos,...)
- Concurrencia
 - Acceso simultáneo de múltiples usuarios y ejecución de múltiples tareas

Software

1. SISTEMA OPERATIVO. FAMILIAS

- Windows
- Unix
- Mac OS
- ¿Hay más?
 - <http://marketshare.hitslink.com/operating-system-market-share.aspx>
 - <http://www.top500.org>



Microsoft[®]



Software

1. SISTEMA OPERATIVO: WINDOWS

- Surge como un entorno gráfico que trabaja sobre MS-DOS.
- Interfaz gráfica formada por ventanas independientes.
- El más utilizado
- Múltiples versiones
 - Uso: servidor, escritorio
 - Arquitectura: 32 bits, 64 bits

Software

1. SISTEMA OPERATIVO: WINDOWS. Versiones



- v.1.01: la primera versión comercializada en 1985.
- v.2.0. 1987. Mejora la interfaz de usuario, la presentación y la interacción con el teclado y el ratón compitiendo con OS/2 Presentation Manager.
- v.3.0. 1990. Mejor soporte para el color y para aplicaciones MS-DOS.
- v.3.1. 1992. Inclusión del estándar OLE y el gestor de fuentes TrueType.
- Windows NT. 1993. No depende de MS-DOS. Para servidores de red.
- Windows para Trabajo en Grupo y Windows 3.11 incorporaba el soporte de red y funciones que mejoran la seguridad en la red y la velocidad.
- Windows 95. 1995. Promesas incumplidas
- Windows NT 4.0. SO más estable. Sin plug&play. Altos requerimientos hardware. Versiones servidor (Server) y estación de trabajo (WorkStation).
- Windows 98. 1998. Evolución de Windows 95. Nueva FAT32.
- Windows ME. El gran fracaso de Windows
- Windows 2000 y XP. Características de Windows NT y Windows 98.
- Windows Vista: lanzamiento en 2007. Otro gran fracaso
- Windows 7: Último lanzamiento en 2010
- ¿El futuro?: windows 8 en 2012... (por confirmar)
- Otras versiones...

Software

1. SISTEMA OPERATIVO: UNIX

- El más veterano.
 - Diseñado en los laboratorios Bell de la empresa AT&T, la primera versión de UNIX aparece en 1971.
- SO de propósito general, multitarea y multiusuario.
- Tradicionalmente asociado al tratamiento de grandes volúmenes de información y/o usuarios.
- Amplia difusión en ambientes universitarios y científicos.
- Existen multitud de ramificaciones de este sistema operativo.
 - Una de ellas es el Minix, origen de uno de los SO más populares: **GNU/LINUX.**

Software

1. SISTEMA OPERATIVO: UNIX. GNU/Linux



- Creado por Linus Torvalds y Richard Stallman en 1991.
- Su origen reside en el proyecto GNU (1983)
 - “...volver al espíritu de cooperación que prevaleció en los tiempos iniciales...” (Richard Stallman)
 - Su filosofía se basa en ofrecer gratuitamente (GNU GPL), cualquier tipo de mejora, por lo que cuenta con una gran cantidad de seguidores.
- Éxito en servidores y superordenadores
 - Soportado por un elevado número de arquitecturas hardware y con un buen acceso a los ficheros de otros sistemas operativos, es el más utilizado en servidores y superordenadores.

Software

1. SISTEMA OPERATIVO: UNIX. GNU/Linux

- Oferta de programas
 - Muchas de las grandes empresas de desarrollo software disponen de versiones para Linux de sus programas
 - *excepto Microsoft, naturalmente.*
- Distribuciones
 - Se especializan o se dirigen a un usuario o aplicación concretos
 - Multitud de distribuciones: Ubuntu, Debian, Suse, Red Hat, ...
- Derivaciones
 - Otros sistemas operativos actuales como Android, Google Chrome, etc... nacen a partir de Linux



Software

1. SISTEMA OPERATIVO: MAC-OS X



- Desarrollado y comercializado por Apple
- Origen en los sistemas operativos MAC-OS (1984)
- La primera versión X data de 2002, a partir de la cual soporta arquitecturas PC (Intel/AMD i386).
- Las versiones actuales se basan en Unix
 - Nombre de grandes felinos (Leopard, Tiger, Panther, Jaguar, ...)
 - La última versión (2010) es la Lion (v10.7)
- Goza de una excelente reputación en los entornos de trabajo multimedia y especialmente en las áreas gráficas.
- iOS: versión específica para otros dispositivos (iphone, ipad, ...)

Software

2. APLICACIONES INFORMÁTICAS

- Las aplicaciones informáticas abarcan cualquier campo posible, existiendo un programa diferente para cada tarea a realizar.
- Una clasificación (como cualquier otra):
 - Aplicaciones de oficina (ofimática)
 - Científicos
 - Gráficos
 - Comunicaciones e Internet
 - Seguridad
 - Multimedia
 - Juegos y Educativos
 - Utilidades de sistema
 - Aplicaciones a medida

Software

2. APLICACIONES INFORMÁTICAS. LICENCIAS

- Libre
 - Dependiendo de la licencia marcada por el creador, el usuario puede tener la libertad para ejecutar, copias, distribuir, estudiar, cambiar, mejorar o incluso cobrar el software.
 - No necesariamente gratuito
 - General Public License (GPL), Berkeley Software Distribution (BSD), ...
- Propietario
 - El creador establece a través de diferentes tipos de licencia (shareware, freeware, ...) las limitaciones que el usuario tiene en el uso, modificación y distribución del software.
 - No necesariamente de pago
 - Shareware, Freeware, ...

Software

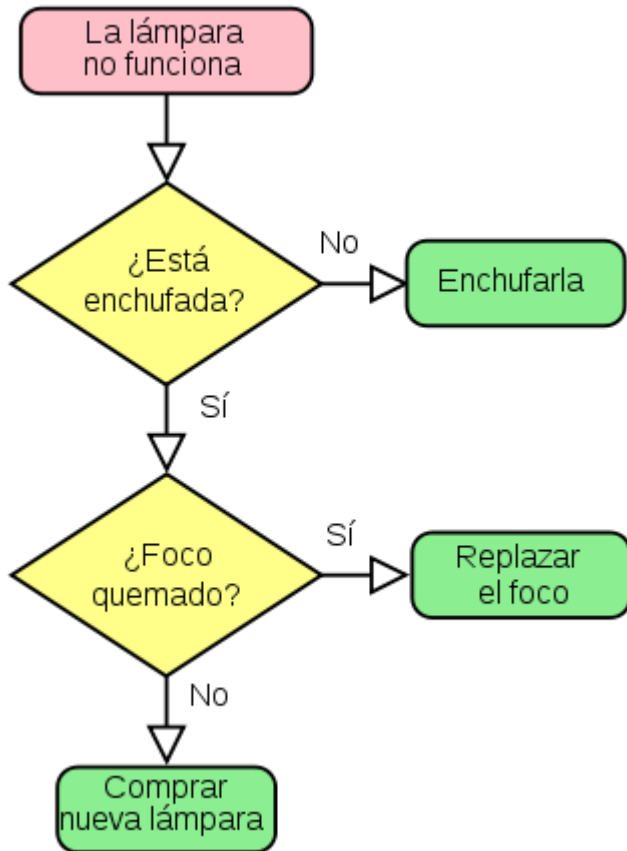
3. LENGUAJES DE PROGRAMACIÓN

- **Algoritmo**
 - Secuencia finita de reglas que, ejecutadas de forma secuencial, permite la realización de una tarea concreta, siendo válido para cualquier posibilidad de estado de los valores de entrada.
- **Programa**
 - La escritura de un algoritmo (o conjuntos de algoritmos) en un lenguaje inteligible para la máquina se denomina programa (instrucción como unidad básica).
- **Lenguaje de programación**
 - Los programas se escriben en un lenguaje concreto
 - posee un alfabeto, un vocabulario y un conjunto de reglas que definen la sintaxis y la semántica de las instrucciones.
 - Se distingue entre lenguajes de bajo nivel y de alto nivel.

Software

3. LENGUAJES DE PROGRAMACIÓN

Algoritmo



Programa (en un lenguaje)

```
IF NOT funciona(lámpara)
THEN
    IF NOT enchufada(lámpara)
    THEN enchufar(lámpara)
    ELSE IF quemado(lámpara.foco)
    THEN reemplazar(lámpara.foco)
    ELSE comprar(lámpara)
```


Software

3. LENGUAJES DE PROGRAMACIÓN: Lenguajes de bajo nivel

- Lenguaje o código máquina
 - Codificado en binario (presentación en hexadecimal).
 - Instrucciones directamente comprensibles por el ordenador, con lo que no necesita traducción
 - pero escribir el programa no es nada rápido ni sencillo.
 - Velocidad y nivel de optimización altos.
 - Cada procesador tiene su propio código máquina.
- Lenguaje ensamblador
 - Surge en 1945. Lenguaje simbólico en el que cada instrucción corresponde directamente con una o varias instrucciones en código máquina.
 - Mnemotécnicos en lugar de las cadenas de bits, facilita algo la escritura de los programas
 - Necesita traducción
- Actualmente, uso en situaciones muy concretas

Software

3. LENGUAJES DE PROGRAMACIÓN: Lenguajes de alto nivel

- Problemas de los lenguajes de bajo nivel:
 - dificultad de escritura
 - gran número de errores
 - conocimiento de la arquitectura concreta del procesador
- Pretensiones
 - Hacer que el programa sea independiente de la máquina
 - Acercar el lenguaje al lenguaje natural, para minimizar los errores.
 - Incluir un conjunto de procedimientos y rutinas que se utilicen comúnmente (librerías de funciones) para evitar reprogramarlas.

Software

3. LENGUAJES DE PROGRAMACIÓN: de bajo a alto nivel

Nuevo teclado profesional para programadores



```
-u 100 1a
```

```
OCFD:0100 BA0B01
OCFD:0103 B409
OCFD:0105 CD21
OCFD:0107 B400
OCFD:0109 CD21
```

```
-d 10b 13f
```

```
OCFD:0100 48 6F 6C 61 2C
OCFD:0110 20 65 73 74 65 20 65 73-20 75 6E 20 70 72 6F 67
OCFD:0120 72 61 6D 61 20 68 65 63-68 6F 20 65 6E 20 61 73
OCFD:0130 73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20
OCFD:0140 57 69 6B 69 70 65 64 69-61 24
```



```
MOV     DX,010B
MOV     AH,09
INT     21
MOV     AH,00
INT     21
```



```
print("Hola")
```

Software

3. LENGUAJES DE PROGRAMACIÓN: La traducción

- Transformación del código escrito (fuente) a código máquina:
 - Ensambladores
 - Traducen el código ensamblador a código máquina.
 - Compiladores
 - Traducen código en lenguajes de alto nivel a código máquina.
 - Producen ejecutables estáticos
 - *si se cambia algo hay que volver a compilar*
 - Intérpretes:
 - Realizan la traducción y la ejecución instrucción por instrucción.
 - La ejecución es más lenta pero se puede detener y modificar la siguiente instrucción
 - Son muy robustos y fácilmente transportables (lenguajes independientes de la máquina).

Software

3. LENGUAJES DE PROGRAMACIÓN: Algunos lenguajes de alto nivel

- FORTRAN: 1956, propósito general, aplicaciones científicas y técnicas.
- COBOL: 1957-60 aplicaciones comerciales.
- BASIC: 1964. Fines didácticos. Ha sufrido múltiples evoluciones.
- PASCAL: 1970. Aprendizaje de estructuras de programación concretas y su uso en diferentes metodologías. Admite recursividad
- C: Década de los 70. Potente y rápido, modular e independiente de la máquina. Recursividad. Su versión para la programación orientada a objetos es C++ (80s).
- PROLOG: 1971, Lenguaje declarativo. Basado en la lógica de predicados y orientado a aplicaciones de inteligencia artificial
- ADA: 1980. Incluye la multitarea y tiene partes de Algol y Pascal.
- JAVA: 1990-94. Evolución de los LOO. Uso reorientado hacia la web
- PHP: 1994. Programación web con bases de datos.
- JAVASCRIPT: 1995. Lenguaje orientado a la programación web.
- C#: 2001. Ideas de C++ y Java