

Tema 2. Software.

Fundamentos de Informática
Grado en Ingeniería Mecánica



Definición de software

«Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.»

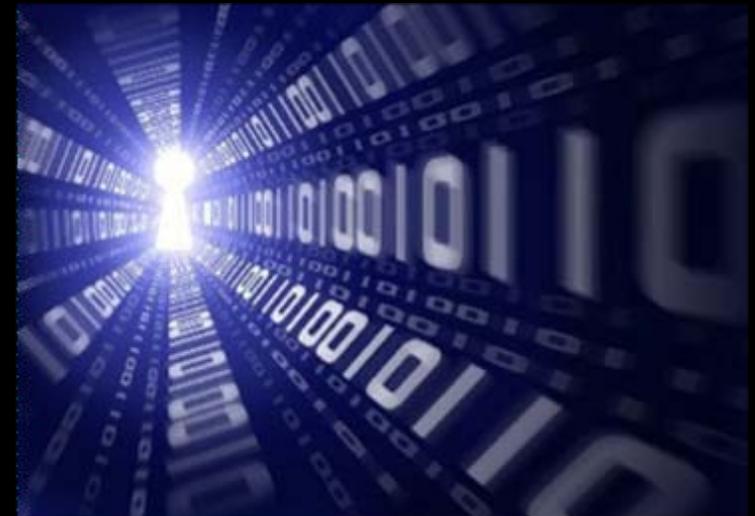
Extraído del estándar 729 del IEEE

- Computador = Hardware + Software
- El software determina el comportamiento del hardware
- Cada programa se desarrolla para un propósito específico
Recordar que un computador era una máquina de propósito general
- Cuando el computador usa un programa se dice que está ejecutando ese programa
- El primer paso para ejecutar un programa es copiarlo de la memoria secundaria a la memoria RAM

Programa fuente vs ejecutable

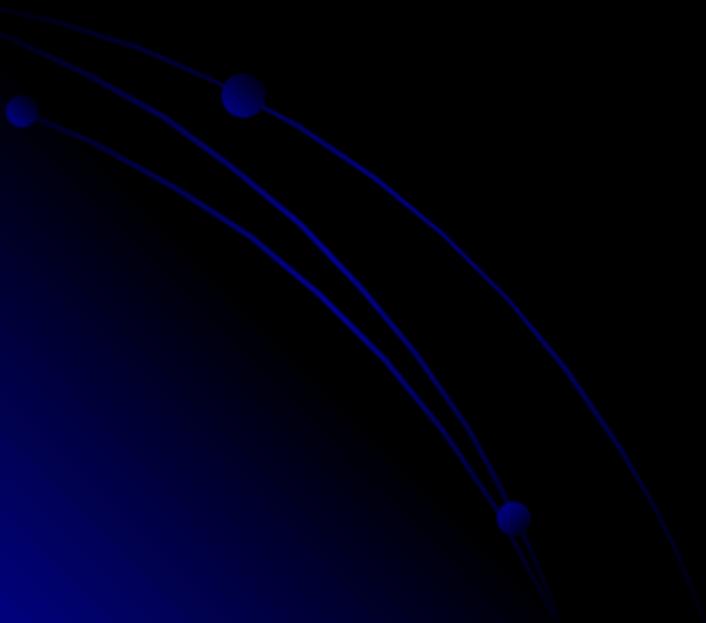
- Código fuente de un programa
 - Fichero de texto que contiene las instrucciones escritas en un lenguaje de programación que debe seguir la computadora para ejecutar dicho programa
 - Dicho fichero ha de ser compilado para generar un código ejecutable
- Código ejecutable o programa ejecutable
 - Fichero binario que contiene las instrucciones escritas en lenguaje máquina que debe seguir la computadora para ejecutar dicho programa

```
import java.io.*;
import java.util.*; //CARGAR LAS CLASES
class Programa {
    //PEQUEÑO EJEMPLO DEL LENGUAJE, PARA PODER VER SU FUNCIONAMIENTO
    static Scanner teclado = new Scanner(System.in).useLocale(Locale.US);
    //MENÚ DEL PROGRAMA
    public static void main(String[] args) {
        //CARGAR MENÚ POR PANTALLA
        int a = 0;
        System.out.println("BIENVENIDA");
        System.out.println("1.- Calcular el volumen del tubo.");
        System.out.println("2.- Calcular el volumen de la esfera.");
        System.out.println("Introduzca la opción deseada:");
        a = teclado.nextInt();
        //COMPROBAR DEL OPCION A (1) O OPCION B (2) O TAMBIEN SALIR
        switch(a) {
            case 1: VolumenTubo(volumen); //CALCULO VOLUMEN TUBO
            case 2: VolumenEsfera(volumen); //CALCULO VOLUMEN ESFERA
            default: System.out.println("La opción no es la elegida"); break;
        }
        //FIN DEL PROGRAMA
        public static void VolumenTubo(v) {
            System.out.println("Dime el radio del tubo:");
            double radio = teclado.nextDouble();
            double volumen = 1.5708*radio*radio;
            System.out.println("El volumen del tubo es: " + volumen);
        }
        public static void VolumenEsfera(v) {
            System.out.println("Dime el radio de la esfera:");
            double radio = teclado.nextDouble();
            double volumen = 4/3*Math.PI*(radio*radio*radio);
            System.out.println("El volumen de la esfera es: " + volumen);
        }
    }
}
```



Indice

- 1) Clasificaciones del software
- 2) Sistemas Operativos
- 3) Software de aplicación
- 4) Programación de software



Clasificaciones del software

Según la función desarrollada en el computador

- Software de sistema
 - Utilizar los componentes del computador
 - Software de control – Sistema Operativo
 - Utilidades de diagnóstico y mantenimiento
- Software de aplicación
 - Realizar tareas específicas para los usuarios
- Software de programación – un programador es un usuario avanzado

Clasificaciones del software

Según la licencia de uso

- **Software propietario**
 - El software pertenece a una persona o empresa
 - Sujeto a derechos de autor
 - Su distribución, reproducción, modificación y comercialización está controlada por el propietario
 - Frecuentemente el usuario sólo tiene acceso al código ejecutable
- **Software libre (open source)**
 - El software pertenece al usuario
 - El autor conserva la propiedad intelectual pero suele renunciar a cobrar por su distribución
 - El autor permite su distribución, reproducción, modificación y comercialización normalmente en los mismos términos
 - Licencia GPL
 - El código fuente está disponible

Clasificaciones del software

Según su forma de distribución

- Adquisición de licencia
 - La mayoría del software propietario tiene este tipo de distribución
- Freeware
 - Software liberado por el autor para su uso gratuito
 - Debe ser utilizado en las formas expresamente permitidas por el autor
 - La mayoría del software libre es gratuito
- Shareware y demo
 - Software sujeto a derechos de autor
 - Distribuido sin cargo como versión de evaluación
 - Prototipos o versiones incompletas (beta)
- Abandonware
 - Software sujeto a derechos de autor aunque cedido por el autor para su uso gratuito
 - Debido a su antigüedad no se comercializa

Sistemas Operativos

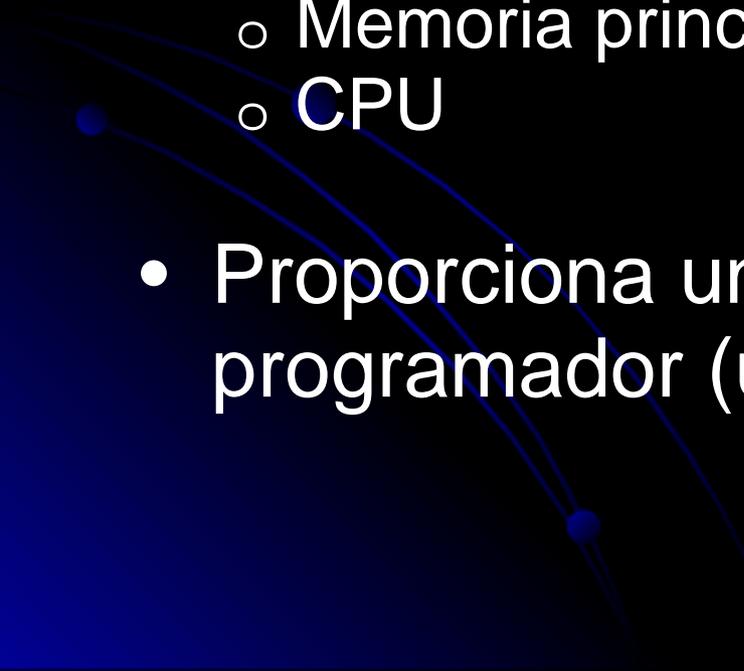
Definición

- Software de control del computador
- Proporciona la plataforma de interacción entre
 - dispositivos de hardware - drivers
 - aplicaciones
 - usuario que utiliza el computador
- Responsable de gestionar y coordinar los programas en ejecución y llevar a cabo la asignación de los recursos



Sistemas Operativos

Funciones

- Posibilita la comunicación usuario-máquina
 - Gestión de recursos
 - Periféricos
 - Memoria secundaria
 - Memoria principal
 - CPU
 - Proporciona un entorno de trabajo al programador (usuario avanzado)
- 

Sistemas Operativos

Funciones: comunicación usuario-máquina

- Proporciona una interfaz de usuario
 - Interfaz gráfica
 - Metáfora del escritorio donde se presentan los recursos disponibles en forma de objetos
 - El ratón es el dispositivo de entrada
 - Ventanas, iconos y menús
 - Interfaz textual
 - Lenguaje de comandos
 - El teclado es el dispositivo de entrada
 - Línea de comandos o SHELL
- Estandarización de la comunicación
 - Dos máquinas diferentes con el mismo S.O. se utilizan igual a nivel de usuario

Sistemas Operativos

Funciones: gestión de periféricos

- Periférico – dispositivo de entrada y/o salida de datos
- Asignación de uso de periféricos a los programas
- Gestión eficiente de los recursos periféricos
- Transformación de señales en datos y viceversa
 - Teclado – pulsaciones en datos
 - Impresora – datos en caracteres imprimibles

Sistemas Operativos

Funciones: gestión de memoria principal

- Copiar en memoria RAM los programas que entran en ejecución
- Copiar en memoria RAM los datos utilizados por los programas en ejecución
- Compartir memoria RAM entre programas
- Registro de zonas de memoria “prohibidas” y “permitidas”
- Registro de la memoria RAM no utilizada
- Recuperar zonas de memoria RAM
- Interacción entre la memoria RAM y el resto de la memoria principal (principalmente Caché)

Sistemas Operativos

Funciones: gestión de memoria secundaria

- Registro de localización de programas y datos
- Registro de zonas “prohibidas” y “permitidas”
- Registro de memoria secundaria no utilizada
- Recuperación de memoria secundaria no utilizada
 - Borrado de ficheros - papelera
- Organización de la información
 - jerarquía de ficheros y directorios en forma de árbol invertido
- Seguridad
- Sistema de ficheros (formatear)
 - FAT, NTFS
 - Ext2, ext3, ext4
 - HFS

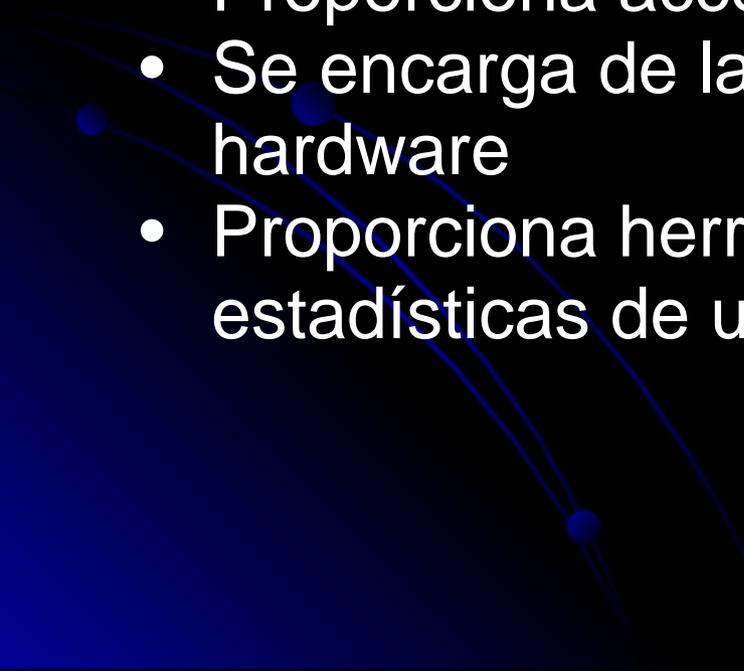
Sistemas Operativos

Funciones: gestión de la CPU

- Los programas se dividen en tareas mas sencillas – procesos
- Cada proceso necesita tiempo de dedicación del microprocesador
- Asignación de tiempos de microprocesador varía entre S.O. - eficiencia
 - Ejemplo: monotarea vs multitarea

Sistemas Operativos

Funciones: entorno de trabajo para el programador

- Proporciona herramientas para la creación de programas
 - Automatiza el mecanismo de ejecución de los programas
 - Facilita el acceso a los dispositivos de hardware
 - Proporciona acceso controlado a los archivos
 - Se encarga de la detección y respuesta ante errores de hardware
 - Proporciona herramientas para la contabilidad y estadísticas de uso y rendimiento
- 

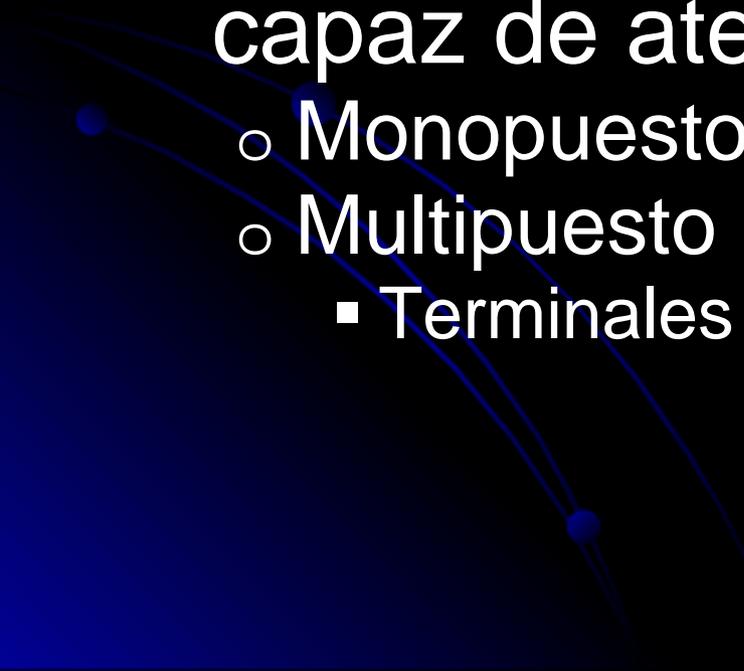
Componentes principales de un Sistema Operativo

- Procesador de mandatos
 - Comunicación con el usuario
- Planificador
- Gestor de archivos
 - Gestión de la memoria secundaria
- Gestor de memoria
 - Gestión de la memoria principal
- Asignador de recursos
 - Gestión de periféricos, memoria principal, memoria secundaria y CPU
- Distribuidor
 - Gestión del tiempo de proceso de la CPU
- Kernel

Clasificaciones de los Sistemas Operativos

- Según el tipo de interacción con el usuario
 - Proceso por lotes
 - Interactivos
 - Tiempo real
- Según el número de usuarios que es capaz de distinguir
 - Monousuario
 - Multiusuario
 - Superusuario - administrador, root

Clasificaciones de los Sistemas Operativos

- Según el número de tareas que es capaz de atender simultáneamente
 - Monotarea
 - Multitarea
 - Según el número de usuarios que es capaz de atender simultáneamente
 - Monopuesto
 - Multipuesto
 - Terminales de trabajo
- 

Sistemas Operativos más utilizados

MS DOS - 1981

- Proceso por lotes, monousuario, monotarea, monopuesto
- Interfaz textual

Windows - 1993

- Interactivo, monousuario-multiusuario, multitarea, monopuesto
- Interfaz gráfica
- Interfaz textual

Windows NT - 1995

- Interactivo, multiusuario, multitarea, monopuesto
- Interfaz gráfica
- Interfaz textual
- Sistema operativo para administración de servidores en red



Sistemas Operativos más utilizados

UNIX - 1969

- Interactivo, multiusuario, multitarea, multipuesto
- Interfaz textual
- Interfaz gráfica
- Sistema operativo para administración de servidores en red y supercomputación



Linux - 1991

- Interactivo, multiusuario, multitarea, multiopuesto
- Interfaz textual
- Interfaz grafica
- Sistema operativo para administración de servidores en red y supercomputación
- Software libre



Sistemas Operativos más utilizados

- Mac-OS – 1984
 - Interactivo, monousuario-multiusuario, multitarea, monopuesto
 - Interfaz gráfica
 - Kernel basado en Linux
 - Sistema operativo para diseño gráfico
- Estadísticas de uso
 - <http://gs.statcounter.com/>



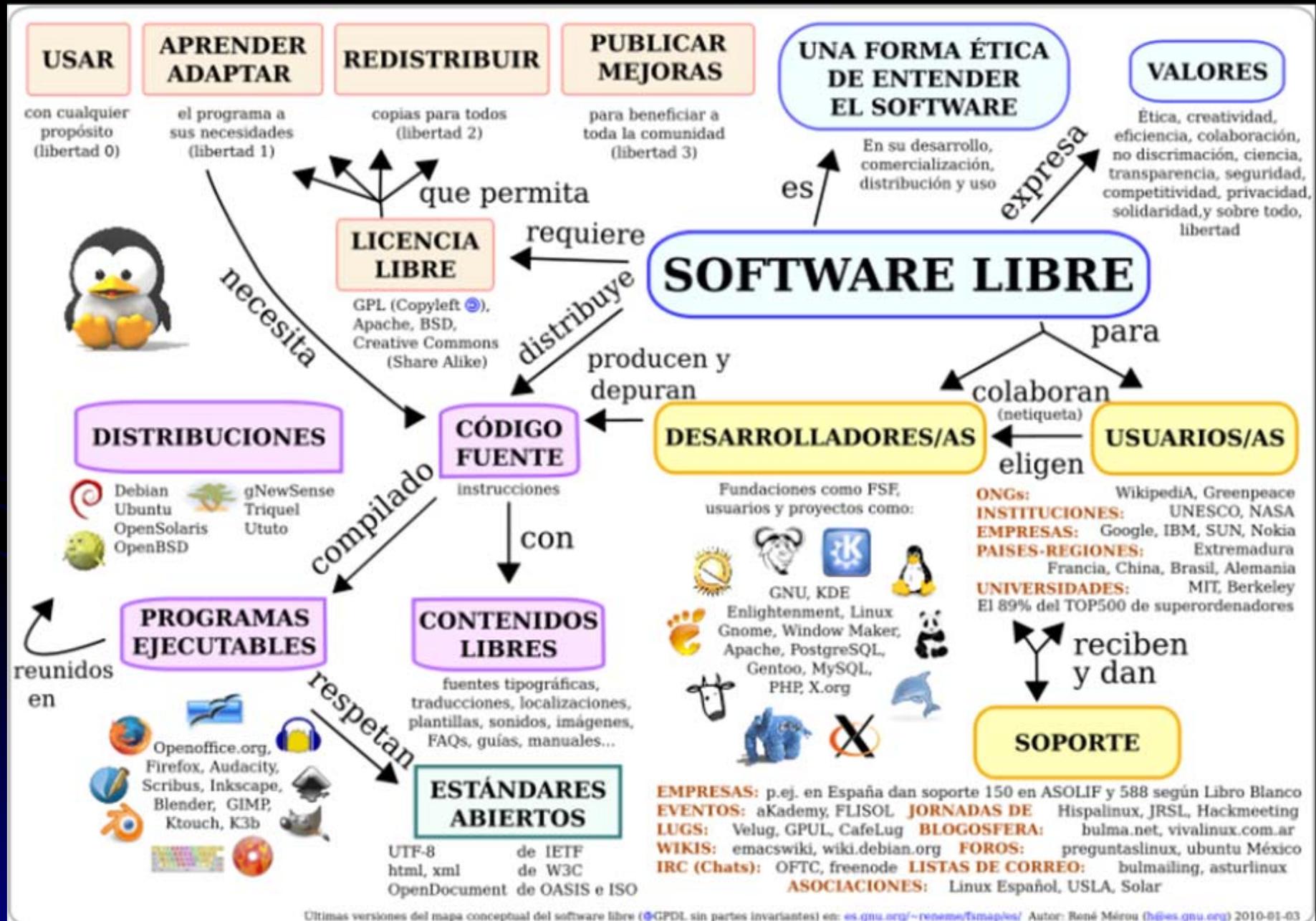
Utilidades de diagnóstico y mantenimiento

- Antivirus – Prevención, detección y eliminación de malware en computadores
- Compresores de archivos – Aprovechamiento de la memoria secundaria
- Defragmentador – Redistribución del espacio en disco asignando ocupaciones contiguas a cada archivo (Windows)
- Software de respaldo – Copias de seguridad
- Software de recuperación – Posibilidad de recuperar archivos borrados por error
- ...

Software de aplicación

- Editores de texto
 - Block de notas, Notepad, Kate, Gedit, VI,...
- Aplicaciones ofimáticas
 - Microsoft office
 - Open office
- Gestores de bases de datos
 - SQL, Access, Oracle...
- Procesado de imagen
 - Photoshop, The Gimp, Inkskape...
- Navegadores
 - Internet explorer, Mozilla Firefox, Google chrome...
- Gestores de correo electrónico
 - Outlook, Eudora, Mozilla Thunderbird...
- Generadores de sitios web
 - Microsoft Frontpage, Dreamweaver, Mozilla SeaMonkey...
- ...

Software de aplicación libre: filosofía



Software de aplicación libre: tres motivos por los que utilizarlo

Software desarrollado pensando en beneficio del usuario vs software desarrollado pensando en beneficios económicos

- Ejemplo: Escritorio de KDE de Linux vs escritorio de Windows
- Ejemplo: Problemas de seguridad de Windows



Proporciona competencia real a software propietario, lo que supone una mejora de la calidad de ambos a largo plazo

- Ejemplo: Competencia entre Mozilla Firefox e Internet Explorer

Legalidad y ética en la utilización del software

Software de aplicación libre: dónde encontrar software libre

- Portal del Software libre

http://es.wikipedia.org/wiki/Portal:Software_libre

- Oficina del Software libre de la Universidad de Zaragoza

<http://osluz.unizar.es/>



Software de aplicación: Malware

- Malicious software – software cuyo objetivo es infiltrarse o dañar un computador sin el consentimiento de su usuario
- Clasificación
 - Malware infeccioso – virus y gusanos
 - Malware oculto – troyanos y puertas traseras
 - Publicidad – spyware y adware
 - Robar información personal – keylogger y stealers
- Malware en Windows vs Linux



Programación de software: Definición

- La programación es el proceso de diseñar, escribir, depurar y mantener el código fuente de programas ejecutables en una computadora.
- El código fuente es escrito en un lenguaje de programación entendible por el programador. Dicho código no se puede ejecutar directamente en una computadora.
- El código fuente debe someterse a un proceso de traducción para convertirlo en lenguaje máquina directamente ejecutable por computador.

Programación de software: Clasificación de los lenguajes de programación en función de la cercanía al lenguaje máquina

- Microprogramación
- Lenguaje máquina comercial
- Kernel del Sistema Operativo
- Ensamblador
- Lenguajes de programación de alto nivel
- Aplicaciones

Programación de software: Clasificación de los lenguajes de programación del desarrollo desde la aparición de los primeros computadores

- **Primera generación**
 - Lenguaje máquina y ensamblador.
- **Segunda generación**
 - Lenguajes simbólicos. LISP (1957) y Prolog (1972).
 - Primeros lenguajes de alto nivel. Fortran (1957), Algol (1958), y Cobol (1960).
- **Tercera generación**
 - Lenguajes de alto nivel. Basic (1964), Pascal (1968), C (1971) y Ada (1979).
- **Cuarta generación**
 - Orientados a aplicaciones de gestión y manejo de bases de datos. SQL (1986).
 - Orientados a objetos. C++ (1983), Java (1995), C# (2001)...
- **Quinta generación**
 - Orientados al procesamiento de lenguajes naturales

Programación de software: Clasificación de los lenguajes de programación de alto nivel en función del paradigma de programación

- Declarativos

- Los programas se construyen mediante descripciones de funciones o expresiones lógicas
 - Funcionales - Lisp
 - Lógicos - Prolog

- Imperativos

- Los programas se construyen mediante instrucciones
 - No estructurados – Fortran, Cobol...
 - Estructurados – C, Pascal, Ada...

- Orientación a objetos

- Los programas se construyen mediante los datos y las operaciones que actúan sobre ellos
 - C++, Java, Eiffel...

Lenguajes de programación: Sintaxis y Semántica

- Sintaxis

- Conjunto de reglas que indican cómo realizar la construcción del lenguaje
 - Palabras reservadas – class, int, boolean, if, for, public...
 - Operadores - + * / && || ...
 - Separadores – { } ;

- Semántica

- Conjunto de reglas que permiten determinar el significado de las construcciones del lenguaje

- Un programa se considerará correcto si lo es sintáctica y semánticamente

Lenguajes de programación: Compilación vs interpretación

- Compilador - programa que permite traducir el código fuente de un programa en lenguaje de alto nivel, a otro lenguaje de nivel inferior (típicamente a lenguaje máquina).
- Intérprete – programa que permite analizar y ejecutar el código fuente de un programa en alto nivel. La traducción se realiza instrucción por instrucción y normalmente el resultado de la traducción no se almacena en ninguna forma.

Lenguajes de programación: Proceso de compilación

Código fuente

*lenguaje de programación
(alto nivel)*

Compilador

Código objeto

*lenguaje máquina
(bajo nivel)*

Enlazador o linker

Código ejecutable

*lenguaje máquina
(bajo nivel)*

Lenguajes de programación: Ejecución

- Ejecución del código ejecutable
- Utilización del intérprete
- Máquina virtual
 - La compilación genera un código intermedio que no es directamente ejecutable por el computador
 - La máquina virtual instalada en el computador es capaz de ejecutar el código intermedio
 - En java, el código intermedio recibe el nombre de Bytecode y la máquina virtual recibe el nombre de Máquina Virtual de Java.

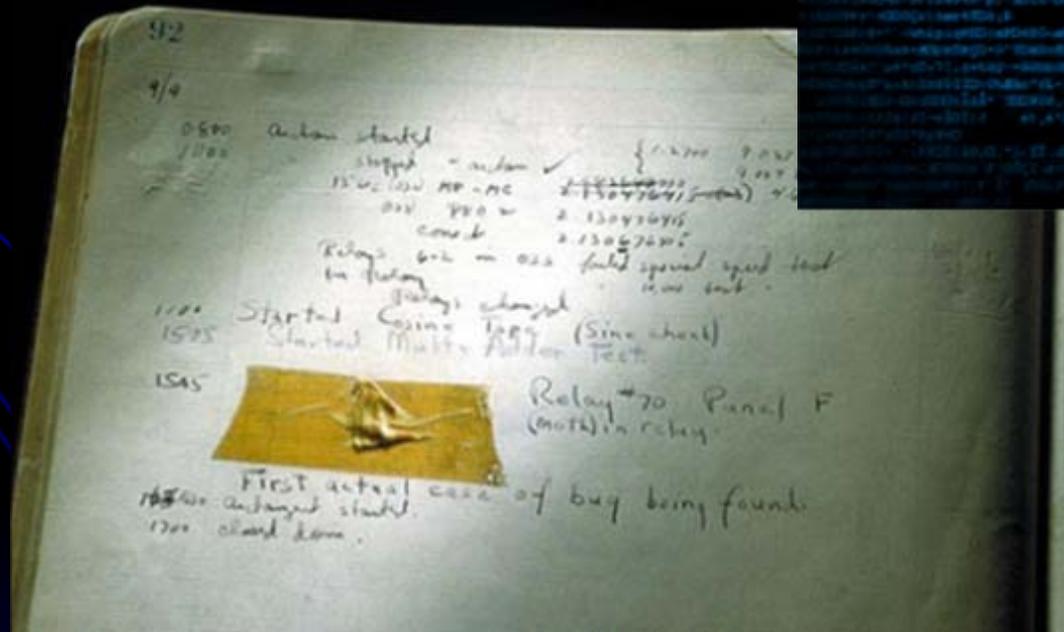
Lenguajes de programación: Depuración

- Tipos de errores
 - Sintácticos
 - El código fuente es sintácticamente incorrecto
 - El compilador es capaz de identificar este tipo de errores
 - En tiempo de ejecución
 - Se produce un error durante la ejecución del programa y aborta
 - División por cero, pantallazo azul de Windows 95
 - Semánticos
 - El programa no se comporta como debería
 - El depurador constituye una herramienta muy útil para ayudar al programador a resolver el problema

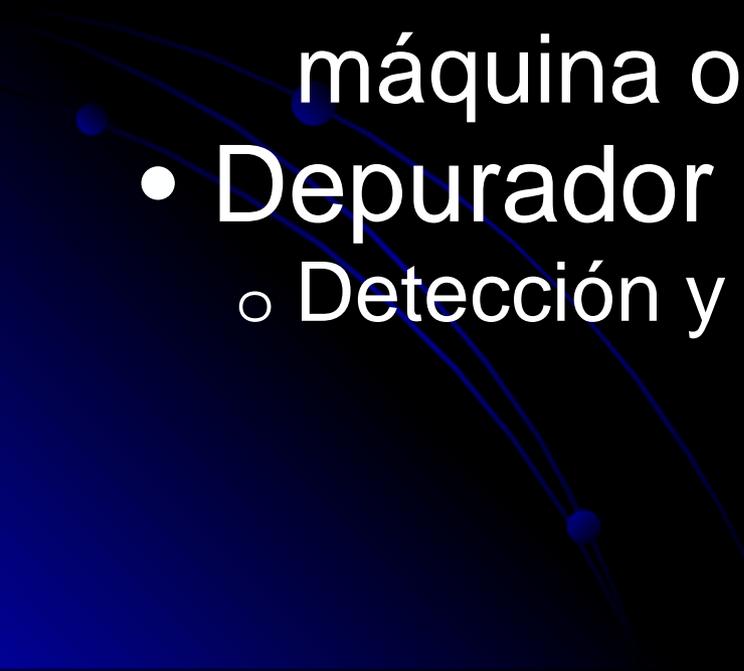


Lenguajes de programación: Bugs

- Depurador
- Debugger
- El primer bug



Lenguajes de programación: Entorno integrado de desarrollo

- Editor de texto
 - Desarrollo de código fuente
 - Compilador y enlazador
 - Traducción de código fuente a código máquina o código intermedio
 - Depurador
 - Detección y corrección de bugs en programas
- 

Blue J tour

