

Tema 5

Software en Sistemas Informáticos

Fundamentos del Hardware
Administración de Sistemas Informáticos en Red

Introducción al Software

Introducción

▶ Definición

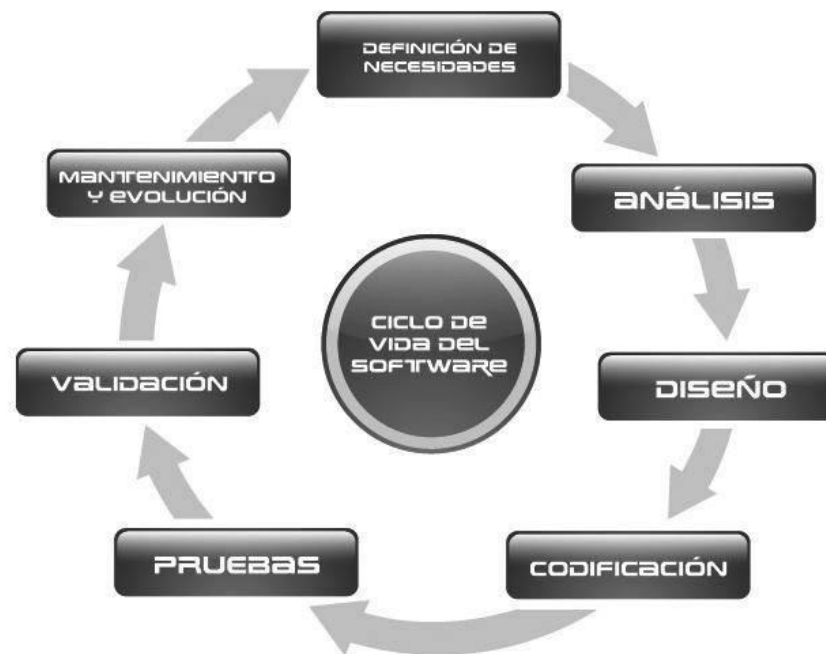
- ▶ Equipamiento o soporte lógico de un sistema informático
- ▶ Conjunto de componentes lógicos (no tangibles y no físicos) necesarios para llevar a cabo una tarea específica en un sistema
- ▶ Conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación (Estándar IEEE 729)

▶ Características

- ▶ Es lógico, no físico
 - ▶ Se desarrolla, no se fabrica
 - ▶ No se estropea
 - ▶ Se puede construir a medida
- ▶ Elemento clave de la evolución de los sistemas y productos informáticos

Introducción

- ▶ Ciclo de vida del software (Ingeniería del Software)
 - ▶ Desarrollo del software desde la fase inicial hasta la fase final
 - ▶ Validación del desarrollo de la aplicación
 - ▶ Verificación de los procedimientos de desarrollo

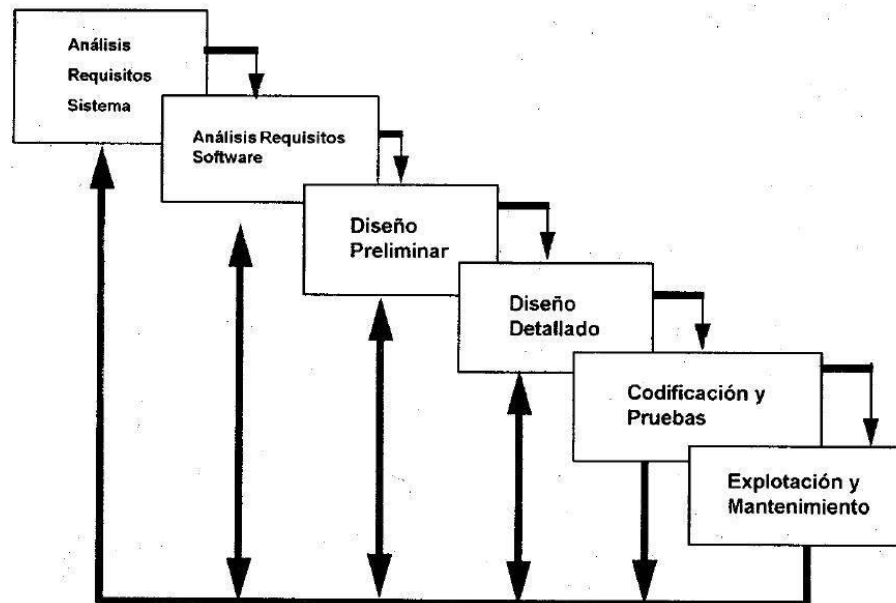


Introducción

- ▶ Ciclo de vida del software – Etapas
 - ▶ Definición de objetivos
 - ▶ Definir el resultado del proyecto y su papel en la estrategia global
 - ▶ Análisis de requisitos y viabilidad
 - ▶ Recopilar, examinar y formular los requisitos del cliente
 - ▶ Examinar cualquier restricción que se pueda aplicar
 - ▶ Diseño general
 - ▶ Determinar los requisitos generales de la arquitectura de la aplicación
 - ▶ Diseño en detalle
 - ▶ Dar una definición precisa de cada subconjunto de la aplicación
 - ▶ Programación (programación e implementación)
 - ▶ Implementación del software en un lenguaje de programación para crear las funciones definidas durante la etapa de diseño
 - ▶ Prueba de unidad
 - ▶ Llevar a cabo una prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones
 - ▶ Integración
 - ▶ Garantizar que los diferentes módulos se integren con la aplicación
 - ▶ Prueba beta o de validación
 - ▶ Garantizar que el software cumple con las especificaciones originales
 - ▶ Documentación
 - ▶ Documentar información necesaria para los usuarios del software y para desarrollos futuros
 - ▶ Implantación
 - ▶ Instalación del software en el entorno real de uso
 - ▶ Mantenimiento
 - ▶ Mantenimiento correctivo (procedimientos correctivos)
 - ▶ Mantenimiento evolutivo o continuo (actualizaciones secundarias)

Introducción

- ▶ Ciclo de vida del software – Modelos
 - ▶ Definen el orden y presencia de cada una de las etapas en el ciclo de vida de una aplicación
 - ▶ Acordado entre el cliente y el equipo de desarrolladores
 - ▶ Modelo en espiral, ciclo de vida prototipado, ciclo de vida incremental, ...



Ejercicio

- ▶ Investiga sobre otros modelos de ciclos de vida del software aparte del ciclo de vida clásico que se ha visto en esta unidad.
- ▶ Analiza las diferencias que encuentres entre los mismos

Introducción

► Clasificación del software

► Software de Sistema

- Conjunto de programas o rutinas con el objetivo de facilitar el uso del ordenador (administrar y asignar recursos del sistema)
- Sistema operativo, controladoras de dispositivos, herramientas de diagnóstico y reparación, herramientas de optimización, ...
 - Entornos operativos

► Software de Aplicación

- Permiten a los usuarios llevar a cabo una o varias tareas específicas en diversos campos

► Software de Programación

- Conjunto de herramientas que permiten al programador desarrollar programas informáticos
 - Entorno de Lenguaje Integrado (IDE)
 - Lenguaje de programación
 - Lenguaje de bajo nivel
 - Lenguaje de alto nivel



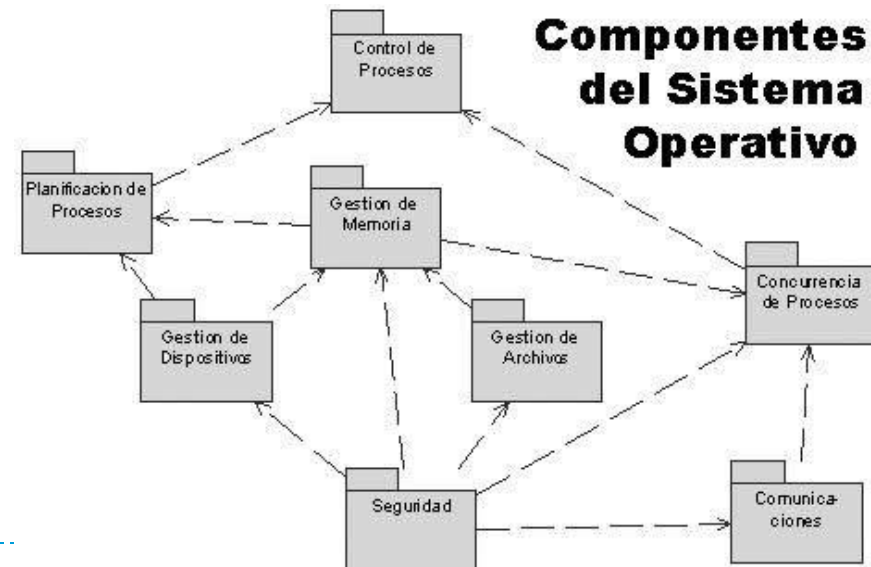
Entornos Operativos

Funciones de los Sistemas Operativos

- ▶ Control de recursos
 - ▶ Coordinar y manipular el hardware del sistema informático
- ▶ Control y manejo de dispositivos de E/S
- ▶ Ejecución y secuenciación de tareas
- ▶ Base estándar sobre la que ejecutar otros programas
- ▶ Administración y mantenimiento de sistemas de archivo de disco
- ▶ Interacción entre sistema y usuarios
 - ▶ Interfaz gráfica (GUI)
- ▶ Detección e informe de errores

Componentes de un Sistema Operativo

- ▶ Gestión de procesos
 - ▶ Proceso: programa en ejecución que necesita recursos (tiempo de CPU, memoria, archivos, dispositivos E/S) para realizar su tarea
- ▶ Gestión de la Memoria Principal
- ▶ Gestión del almacenamiento secundario
- ▶ Gestión de la E/S
- ▶ Gestión de archivos
- ▶ Mecanismos de protección
- ▶ Gestión de las comunicaciones
- ▶ Utilidades de sistema



Entornos operativos actuales

- ▶ Características básicas comunes
 - ▶ Interfaz en modo gráfico
 - ▶ Multiusuario y multitarea
 - ▶ Forman parte de una familia o plataforma compatibles entre sí
 - ▶ Soportan uno o varios sistemas de archivos
 - ▶ Ofrecen herramientas similares con distintos nombres



Entornos operativos actuales

- ▶ Entornos operativos para computadores
 - ▶ Plataforma Windows
 - ▶ Plataforma GNU/Linux
 - ▶ Plataforma Mac OS
 - ▶ Otras: Google Chrome OS, eyeOS, Jolicloud, ...
- ▶ Entornos operativos para dispositivos móviles
 - ▶ Android (baso en núcleo Linux)
 - ▶ iOS (anteriormente iPhone OS)
 - ▶ Windows Mobile
 - ▶ HP WebOS
 - ▶ Blackberry RIM



Entornos operativos actuales

▶ Plataforma Windows

- ▶ Alta implantación en entorno doméstico
- ▶ Gran implantación en pequeña y mediana empresa
- ▶ MS DOS / IBM PC-DOS 1.0 (1981) → MS DOS 6.22 (1994)
 - ▶ Versiones posteriores (hasta 8.01) incluidas en Windows (95-Me)
- ▶ Windows 3.0 (1990) → Windows 7 (2009)
- ▶ Distribuciones para hogar, empresa y servidor
- ▶ Amplia gama de aplicaciones disponibles

Ejercicio

- ▶ Enumera las diferentes versiones de Windows por las que se ha pasado tratando de identificar el periodo de vigencia y las características más relevantes

Entornos operativos actuales

► Plataforma GNU/Linux

- Núcleo del S.O. Linux + Herramientas de sistema GNU
- Proyecto GNU (Richard Stallman, 1983)
- Linux (Linux Torvalds, 1991)
- Distribuciones del sistema (Slackware, Debian, RedHat/Fedora, SuSE/OpenSUSE, Gentoo, Ubuntu, ArchLinux,...)
- Características
 - **Software libre**
 - Amplia gama de aplicaciones disponibles
 - Herramientas avanzadas de seguridad y administración de redes
 - Prácticamente inmune a virus y otras amenazas
- Varias empresas comercializan soluciones basadas en GNU/Linux



Ejercicio

- ▶ Describe las principales distribuciones Linux a nivel mundial así como las diferentes distribuciones Linux que se ofrecen en España por comunidades

Entornos operativos actuales

▶ Plataforma Mac OS

- ▶ Desarrollado por Apple para sus equipos

- ▶ Primer entorno con GUI

 - ▶ Interacción ratón

 - ▶ Uso de ventanas, iconos y menús

- ▶ Primera versión en 1984

 - ▶ A partir de Mac OS X (2002) basado en núcleo BSD Unix

- ▶ Mac OS X 10.6 “Snow Leopard” (2009)

- ▶ Características

 - ▶ Versiones cliente y servidor

 - ▶ Plataforma estable y segura (multitarea preventiva, memoria protegida, ...)

 - ▶ Interfaz gráfica Aqua, tecnología antialiasing, ...



Ejercicio

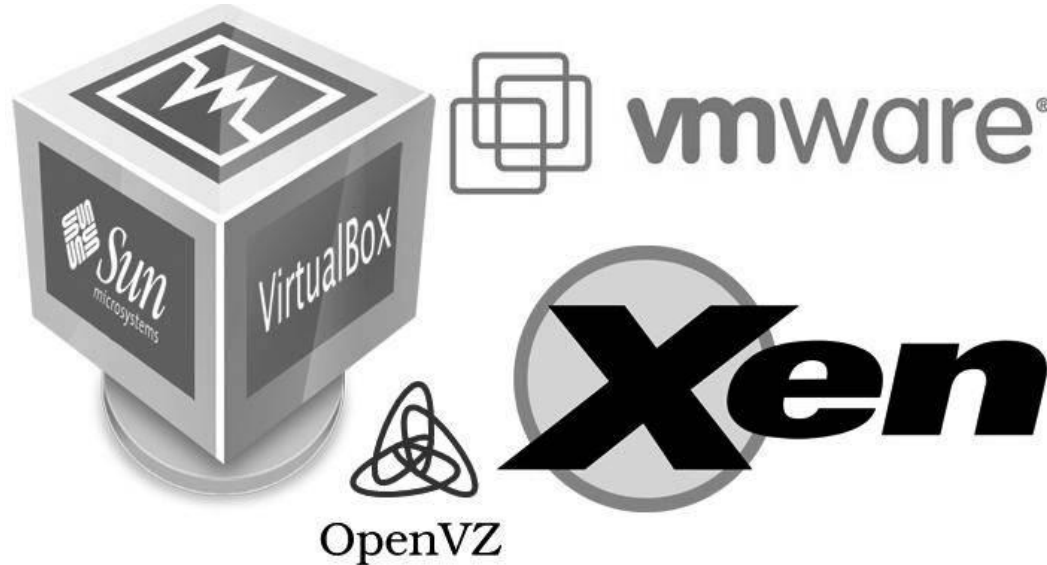
- ▶ Estudia los requerimientos y prestaciones de las principales distribuciones en las diferentes plataformas (Windows 7, Ubuntu 10.4-10.10, Mac OS X)

Virtualización de entornos operativos

- ▶ Permite ejecutar distintos sistemas operativos de forma virtual sobre uno solo
- ▶ Virtualización de plataforma
 - ▶ Se lleva a cabo en una plataforma de hardware mediante un software anfitrión –programa de control que simula un entorno computacional (máquina virtual) – para su software invitado
 - ▶ El software invitado suele ser un SO completo que se ejecuta como si estuviera instalado en una plataforma de hardware autónoma



Virtualización de entornos operativos



- ▶ Software de virtualización
 - ▶ VMWare (versiones de pago y gratuitas)
 - ▶ Parallels (entorno Mac OS X)
 - ▶ VirtualPC (gratuito, entorno Windows)
 - ▶ Software libre: Xen, OpenVZ, VirtualBox, ...

Ejercicio

- ▶ Explica qué utilidad tiene la virtualización
- ▶ Utilizando la herramienta de virtualización VirtualBox, indica los pasos dados para su instalación y puesta en marcha en un sistema anfitrión. Explica los pasos y opciones de configuración para la creación de un entorno computacional. Documenta la instalación de un sistema invitado GNU/Linux en el entorno computacional que acabas de crear.

Tipos de aplicaciones

Tipos de aplicaciones

- ▶ Software de aplicación
 - ▶ Aquel tipo de software diseñado como una herramienta que permita al usuario realizar una tarea específica
- ▶ Una aplicación...
 - ▶ Programa compilado o interpretado escrito en algún lenguaje de programación
 - ▶ Necesita de un sistema operativo soporte para poder funcionar
 - ▶ Requiere unas condiciones determinadas para su instalación y ejecución,
 - ▶ Espacio en disco, procesador y memoria RAM
- ▶ Solucionan ciertas tareas humanas complicadas
 - ▶ Gestión de almacén, redacción de documentos, diseño de planos, ...



- ▶ Desarrollado de forma estándar y de uso generalizado / desarrollado a medida
- ▶ Aplicaciones verticales (campo de empleo específico) / aplicaciones horizontales (amplio sector de usuarios)
- ▶ Paquete integrado o *suite*
 - ▶ Agrupación de diversos programas de uso similar o complementario entre sí
- ▶ Toda aplicación está desarrollada para ser ejecutada en una única plataforma o S.O.

Tipos de aplicaciones

- ▶ **Aplicaciones de productividad empresarial**
 - ▶ Empleadas para fines empresariales de mejora de productividad en diversos sectores
 - ▶ Aplicaciones ofimáticas, aplicaciones gráficas, gestión de proyectos, ...
- ▶ **Aplicaciones de ámbito doméstico**
 - ▶ Utilizadas sin un fin profesional para el entretenimiento o la formación
 - ▶ Juegos, enciclopedias multimedia, ...
- ▶ **Aplicaciones profesionales horizontales**
 - ▶ Diseñadas para gestionar y ejecutar un proceso empresarial de forma estándar
 - ▶ Aplicaciones de contabilidad profesional, gestión de recursos, diseño web, ...
- ▶ **Aplicaciones verticales**
 - ▶ Diseñadas para la realización de unas determinadas funciones dentro de un sector determinado
- ▶ **Aplicaciones utilitarias**
 - ▶ Realizan una tarea muy específica (menor tamaño, coste y complejidad que las anteriores)
 - ▶ Aplicaciones de rendimiento y diagnóstico, compresores, antivirus, programas de grabación, ...

Tipos de licencias de software

Licencias de software

- ▶ Definición
 - ▶ Contrato en donde se especifican todas las normas que rigen el uso de un determinado programa
 - ▶ De una parte, el propietario de la licencia
 - ▶ Puede ser el autor o titular de los derechos de explotación o distribución del producto
 - ▶ De otra parte, quien adquiere la licencia, usuario o empresa
 - ▶ Empleará la aplicación cumpliendo una serie de términos y condiciones establecidos en el contrato
- ▶ Las diferentes cláusulas determinan, entre otras cosas
 - ▶ Plazo de la cesión de los derechos
 - ▶ Ámbito geográfico de validez del contrato
 - ▶ Límites de responsabilidad por fallos
 - ▶ Tipo de mantenimiento que le da soporte
 - ▶ Compromisos que debe adquirir el propietario
 - ▶ Número de copias licitadas
 - ▶ No cesión del programa a otros
 - ▶ No reinstalación de la aplicación en equipos distintos del original,
 - ▶ ...

Licencias de software

- ▶ Tipos básicos de licencias software
 - ▶ **Licencia propietaria**
 - ▶ Uso en una o varias máquinas por el pago de un precio
 - ▶ **Licencia Shareware**
 - ▶ Uso limitado en tiempo o capacidades tras el cual habrá que pagar un precio
 - ▶ **Licencia Freeware**
 - ▶ Uso y copia ilimitado con precio cero
 - ▶ **Licencia libre**
 - ▶ Distintas combinaciones de uso, copia, modificación y distribución libre, con acceso al código fuente

| Tipo de licencias de software | Limitaciones | Precio cero | Permiso de copia y redistribución | Código Fuente y permiso para modificarlo (sin ánimo de lucro) | Código Fuente y permiso para modificarlo (con fines comerciales) |
|----------------------------------|----------------------------------------------|-------------|-----------------------------------|---------------------------------------------------------------|------------------------------------------------------------------|
| <i>Propietario</i> | Ninguna | No | No | No | No |
| <i>Shareware o de Evaluación</i> | No 100% funcional, o uso por tiempo limitado | Sí | Sí | No | No |
| <i>Freeware</i> | Ninguna | Sí | Sí | No | No |
| <i>Semilibre</i> | Ninguna | Sí | Sí | Sí | No |
| <i>Libre / Open Source / GPL</i> | Ninguna | Sí* | Sí | Sí | Sí |

* Hay veces que dentro de un mismo producto se mezcla algunas partes libres y otras no (sobre todo distribuciones de Linux). Es habitual que estos productos tengan un precio no cero debido a las partes que no son libres.

Clasificación de las licencias de software

- ▶ Según a quien va dirigida
 - ▶ Licencia de usuario final
 - ▶ Licencia de distribuidor
- ▶ Según los derechos que cada autor reserva
 - ▶ Licencia de software de código abierto con permisos
 - ▶ Permite crear una aplicación derivada sin que requiera protección alguna
 - ▶ Licencia de software de código abierto robusta (con restricciones)
 - ▶ Robustas o con restricciones fuertes, *CopyLeft* fuerte
 - ▶ Débiles o con restricciones débiles, *CopyLeft* débil, suave o híbrido
 - ▶ Licencias de código cerrado (software propietario o privativo)
 - ▶ Licencias de dominio público (sin licencia)

Clasificación de las licencias de software

| TIPOS DE LICENCIAS | | | | | |
|-------------------------------------------------------------------------------------|-------------------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | | Copiar y distribuir | Uso comercial | Trabajos derivados | Autoría |
|  | Todos los derechos reservados | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
|  | Dominio público | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
|  | Creative Commons Reconocimiento | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | Creative Commons Reconoc.-Compartir igual | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | Creative Commons Reconoc.-No derivada | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | Creative Commons Reconoc.-No comercial | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | Creative Commons Rec.-No comerc.-Comp. igual | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | Creative Commons Rec.-No comerc.-No derivada | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | GNU GPL | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | GNU LGPL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|  | GNU GFDL | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

☒ SI
 ☒ NO
 ☐ SI, CON RESTRICCIONES
 ☐ SI, Y SE HEREDA

Ejercicio

- ▶ Ofrece varios ejemplos de licencias para la clasificación según los derechos reservados por el autor del software
- ▶ Define GPL, LGPL y BSD enumerando diferencias así como ventajas e inconvenientes de unas y otras
- ▶ Define el concepto *copyleft* y evalúa posible su conveniencia frente a las licencias con derechos reservados

Componentes de Aplicaciones Arquitecturas del Software

Arquitectura del software

- ▶ La arquitectura del software...
 - ▶ Se basa en la creación de aplicaciones basadas en módulos y componentes según una metodología
 - ▶ Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del S.I. cubriendo todas las necesidades
 - ▶ Es el diseño de más alto nivel de la estructura de una aplicación
 - ▶ Define de manera abstracta los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos
- ▶ Vistas fundamentales
 - ▶ Visión estática
 - ▶ Describe los componentes que tiene la arquitectura
 - ▶ Visión dinámica
 - ▶ Describe cómo se comportan los componentes a lo largo del tiempo y cómo interactúan entre sí
 - ▶ Visión funcional
 - ▶ Describe qué hace cada componente
- ▶ Estas vistas (modelos) se expresan mediante lenguaje natural, diagramas de estado, diagramas de flujo de datos,...
 - ▶ UML (lenguaje unificado de modelado) permite implementar todos los modelos

Arquitectura del software

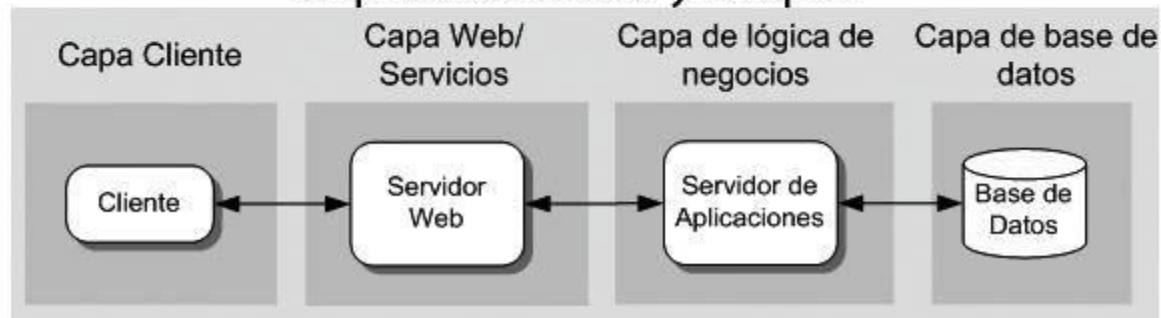
- ▶ Clasificación en base a la organización de los componentes
 - ▶ **Arquitectura Monolítica**
 - ▶ El software se estructura en componentes funcionales fuertemente acoplados
 - ▶ **Arquitectura Cliente-Servidor**
 - ▶ La carga de cómputo se reparte en dos partes independientes pero sin reparto claro de funciones
 - ▶ **Arquitectura de tres niveles**
 - ▶ Caso concreto de arquitectura C/S
 - ▶ La carga se divide en 3 partes (niveles o capas), que se relacionan únicamente con la inmediata, con un reparto claro de funciones
 - Capa de presentación (interfaz de usuario)
 - Capa de procesamiento o cálculo (lógica de negocio)
 - Capa de persistencia o almacenamiento

Arquitectura del software

Arquitectura Monolitica (2 capas)



Arquitecturas de 3 y 4 capas



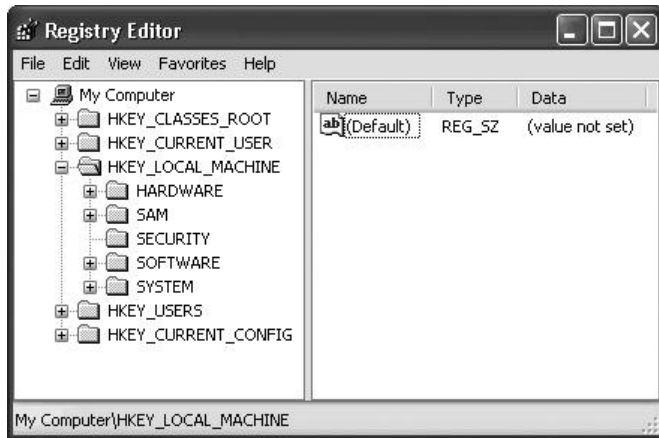
Instalación, configuración y eliminación de aplicaciones

Formas de instalación

- ▶ Instalación y configuración del software es condición necesaria para su uso
- ▶ Si falla el proceso puede que la aplicación no funcione o lo haga de forma incorrecta
- ▶ Instalación por copia directa (Mac Os)
 - ▶ Fácil e intuitivo
 - ▶ Aplicaciones organizadas en un único punto (uso de librerías compartidas, lista de preferencias)
 - ▶ Ventajas
 - ▶ El funcionamiento general del S.O. no se ve alterado
 - ▶ Evita conflictos de incompatibilidad entre diferentes aplicaciones
 - ▶ Inconvenientes
 - ▶ Se puede “olvidar” versiones antiguas, ocupando espacio en disco



Formas de instalación



- ▶ Instalación mediante instalador (Windows)
 - ▶ Archivo ejecutable (instalador) en un medio de almacenamiento o bien descargado de Internet
 - ▶ Copia el software en las ubicaciones definidas
 - ▶ Repartidos por varios subdirectorios
 - ▶ Deja constancia de la instalación en el S.O.
 - ▶ Registro del Sistema
 - ▶ Librerías DLL en directorios de sistema

▶ Ventajas

- ▶ Hace el proceso de instalación transparente al usuario
- ▶ La ubicación raíz de los programas instalados es seleccionable

▶ Inconvenientes

- ▶ Hace necesario el uso de desinstaladores para no dejar rastro de la aplicación al eliminarla y que no haya ningún elemento que afecte al rendimiento

Formas de instalación

- ▶ Instalación usando un gestor de paquetes (GNU-Linux)
 - ▶ Colección de herramientas del entorno operativo destinadas a automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software
 - ▶ Repositorios de paquetes software
 - ▶ Software + meta información
 - ▶ Sistemas basados en paquetes binarios
 - ▶ dpkg (APT) → utilizado por Debian y derivados; formato de paquete deb; primer sistema en emplear gestión de paquetes
 - ▶ fink → utilizado por MacOS en la instalación de software libre
 - ▶ rpm → desarrollado por RedHat, utilizado en gran número de distribuciones mediante herramientas urpmi, Yast, yum, ...
 - ▶ tgz → archivo tar comprimido con gzip; usado por Slackware
 - ▶ pacman → gestiona archivos tgz; utilizada por ArchLinux
 - ▶ portage → basado en el sistema de distribución de FreeBSD; utilizado por Gentoo

Pasos básicos de una instalación

1. Comprobar si se cumplen los requisitos básicos de instalación (hardware y software)
 - ▶ Puede requerir la desinstalación de versiones antiguas
 2. Verificación de que el software es original
 3. Creación de los directorios
 4. Creación de usuarios y permisos necesarios
 - ▶ Por petición explícita de la aplicación o como recomendación
 5. Copia, desempaquetado y descompresión
 6. Compilación y enlace con las bibliotecas
 7. Configuración de la aplicación
 - ▶ Definición de variables de entorno
 8. Registro de la instalación en caso necesario
-
- ▶ ¿Qué pasa si el software ha sido desarrollado a medida?

Pasos básicos de una instalación

- ▶ Puesta en marcha de un sistema software en un sistema informático
 - ▶ Es el primer sistema implantado en la empresa
 - ▶ Hay traspaso de información de un sistema antiguo
 - ▶ Proceso encadenado
 - Continúa funcionando el antiguo de forma que los resultados del nuevo se contrastan y comparan con el antiguo
 - ▶ Proceso directo
 - Desactivación del antiguo y activación directa del nuevo
 - ▶ Proceso en paralelo
 - Ambos conviven a la vez hasta comprobar la fiabilidad del nuevo
 - ▶ Proceso por subsistemas
 - Ambos sistemas, el antiguo y el nuevo, se reparten el trabajo

Configuración de aplicaciones

- ▶ Conjunto de acciones que determina el valor de algunas variables de una aplicación en su ejecución
 - ▶ Comportamiento de la aplicación
- ▶ Configuraciones típicas:
 - ▶ Configuración predeterminada
 - ▶ Cuando no se ha definido ninguna configuración
 - ▶ Genérica → no optimizada al entorno concreto
 - ▶ Configuración personalizada
 - ▶ Determinada de forma específica por el usuario
- ▶ Errores en el proceso de configuración
 - ▶ Pueden provocar la imposibilidad de ejecución o una ejecución defectuosa
 - ▶ Preventivo:
 - ▶ Comprobar requerimientos de la aplicación
 - ▶ Seguir recomendaciones de la guía de instalación

Eliminación de aplicaciones

- ▶ Plataforma Windows
 - ▶ Uso de software específico desinstalador
 - ▶ Eliminación de los archivos que constituyen la aplicación
 - ▶ Ajustes necesarios de configuración del SO
 - ▶ Eliminación de iconos y accesos directos
 - ▶ Nunca utilizar eliminación directa
- ▶ Plataforma Mac OS
 - ▶ Eliminación directa (arrastrar icono de la aplicación a la papelera)
 - ▶ Posible necesidad de eliminación de archivos de configuración
- ▶ Plataforma GNU/Linux
 - ▶ Sistema automatizado desde el gestor de paquetes
- ▶ Software para limpiar. Cleaners
 - ▶ Pequeñas aplicaciones encargadas de analizar el sistema en busca de rastros en forma de librerías y archivos de configuración de aplicaciones ya desinstaladas que pueden afectar al rendimiento del sistema

Ejercicio

- ▶ Basándote en tu experiencia, indica las ventajas e inconvenientes que encuentras entre los métodos de instalación y desinstalación basado en instalador o basado en gestor de paquetes.

Prueba o testing de aplicaciones

Pruebas de software

- ▶ Conjunto de procesos que permiten verificar y validar la calidad de un producto software identificando errores de diseño e implementación
- ▶ Integradas dentro de las diferentes fases del ciclo del software
 - ▶ Es habitual que se inicie en el mismo momento que empieza el desarrollo y continúe hasta que finaliza el mismo
- ▶ Ejecutar el software bajo condiciones contraladas
 - ▶ Condiciones normales
 - ▶ Condiciones anormales
- ▶ Objetivo
 - ▶ Detectar errores de programación y defectos de forma

Fallos del software

▶ Causas

- ▶ Poca o falta de comunicación entre los usuarios que intervienen en su desarrollo
- ▶ Complejidad del software
 - ▶ Poca reutilización de código
- ▶ Errores de programación
- ▶ Cambios continuos durante su desarrollo
 - ▶ Cambios de requerimientos una vez comenzado el proyecto conlleva constantes rediseños y replanificaciones
- ▶ Presiones de tiempo
 - ▶ Omitir fases de pruebas y control para cumplir con los plazos
- ▶ Pobre documentación del código
 - ▶ Dificulta la modificación del código

Fallos del software

- ▶ El **testing** puede **probar** la **presencia** de errores, pero **no** la **ausencia** de ellos (*Edsger Dijkstra*)
- ▶ Consideraciones generales:
 - ▶ El *tester* (probador, persona que realiza las pruebas) es habitualmente un profesional de altos conocimientos en lenguajes de programación y métodos, técnicas y herramientas especializadas de pruebas
 - ▶ Nunca se debe someter a prueba el software en un entorno de explotación → se deberá preparar un entorno de pruebas físicamente separado del de producción y en condiciones lo más parecidas posible (o idénticas)
 - ▶ El grupo de probadores debería ser diferente e independiente del de desarrolladores

Tipos de pruebas

▶ **Verificación**

- ▶ ¿Está el producto correctamente construido?
- ▶ Demostrar que un programa cumple con sus especificaciones
- ▶ Comprobación de las distintas fases antes de pasar a la siguiente

▶ **Validación**

- ▶ ¿El producto construido es correcto?
- ▶ Comprobar que el programa da la respuesta que espera el usuario
- ▶ Comprobación de los requerimientos del software
- ▶ Incluye las pruebas del software
- ▶ Comienza después que la verificación esté completada

▶ **Tipos de pruebas**

- ▶ Prueba de caja negra o caja blanca, unidad de testeo, integración incremental, prueba de integración, prueba funcional, prueba de sistema, prueba de fin a fin, prueba de sanidad, prueba de regresión, prueba de aceptación, prueba de carga, prueba de estrés, prueba de performance, prueba de instalación y desinstalación, prueba de recuperación, prueba de seguridad, prueba de compatibilidad, prueba de explotación, prueba de comparación, prueba alfa, prueba beta, prueba de mutación

Tipos de software

- ▶ **Versión beta**
 - ▶ Versión no final del producto para que la prueben los usuarios
- ▶ **Versión alpha**
 - ▶ Versión anterior a versión beta en el proceso de desarrollo
 - ▶ Programa incompleto
 - ▶ Funcionalidad básica
- ▶ **Versión RTM (Release To Market)**
 - ▶ Programa completo
 - ▶ Se comprueba cada funcionalidad del aplicativo en entornos de producción

Ejercicio

- ▶ Explica las diferencias entre las pruebas de tipo alfa y las pruebas de tipo beta.
- ▶ ¿Conoces aplicaciones que lancen este tipo de distribuciones? Enuméralas.

Evaluación y rendimiento de aplicaciones. Comparativas

Evaluación y rendimiento de aplicaciones

- ▶ Objetivo del desarrollo de cualquier software → satisfacer una necesidad planteada por un cliente
- ▶ ¿Cómo puede saber un desarrollador si el producto construido corresponde exactamente con lo que el cliente pidió?
- ▶ ¿Cómo puede un desarrollador estar seguro de que el producto construido funcionará correctamente?
- ▶ El producto debe ser evaluado a medida que se construye

Evaluación y rendimiento de aplicaciones

- ▶ **Control de la calidad**
 - ▶ Adaptación a las necesidades del cliente (usuario)
 - ▶ Investigar sus necesidades (análisis de mercados)
 - ▶ Definición de esas necesidades (especificación de requisitos)
- ▶ **Crisis del software de los 90**
 - ▶ Calidad insuficiente del producto final
 - ▶ Estimaciones de duración de proyectos y asignación de recursos inexactas
 - ▶ Escasez de personal cualificado en un mercado laboral de alta demanda
 - ▶ Tendencia al crecimiento del volumen y complejidad de los productos
- ▶ La calidad del software no se mide únicamente por unos parámetros de funcionamiento sino que hay otros aspectos que también son importantes (p.e. el soporte)

Evaluación y rendimiento de aplicaciones

▶ Pruebas de rendimiento

- ▶ Destinadas a evaluar la calidad del software
- ▶ Orientadas hacia el usuario (p.e. tiempo de respuesta)
- ▶ Orientadas hacia el sistema (p.e. uso de CPU)
- ▶ Medidas típicas de rendimiento
 - ▶ Variables de tiempo (tiempo de retorno, tiempo de respuesta, tiempo de ejecución)
 - ▶ Capacidad de ejecución
 - ▶ Carga de trabajo
 - ▶ Utilización
 - ▶ ...
- ▶ Ayudan a evaluar la escalabilidad, fiabilidad y uso de recursos del software

Evaluación y rendimiento de aplicaciones. Comparativas

- ▶ **Benchmark**
 - ▶ Aplicación o conjunto de aplicaciones cuya finalidad es evaluar el rendimiento de un sistema informático
- ▶ **Categorías generales de pruebas de comparación**
 - ▶ **Pruebas de aplicación-base**
 - ▶ Se encargan de ejecutar y cronometrar los tiempos de las mismas
 - ▶ **Pruebas playback**
 - ▶ Usan llamadas al sistema durante actividades específicas de una aplicación como uso de disco o llamadas a rutinas de gráficos, ejecutándolas aisladamente
 - ▶ **Pruebas sintéticas**
 - ▶ Enlazan actividades de la aplicación en subsistemas específicos
 - ▶ **Pruebas de inspección**
 - ▶ No intentan imitar la actividad sino que las ejecuta directamente en su entorno productivo

Ejercicio

- ▶ Phoronix Test Suite es un conjunto de aplicaciones de benchmark orientada tanto a sistemas con entorno operativo GNU/Linux como a otros entornos
 - ▶ Enumera las principales características de la suite
 - ▶ Realiza una prueba de rendimiento de tu sistema. Compara los resultados obtenidos con los resultados obtenidos por el resto de compañeros

Software de propósito general

Software de propósito general

- ▶ Software de propósito general
 - ▶ Todos aquellos programas diseñados para un uso común y generalizado por parte de un gran número de usuarios de diferentes perfiles y ámbitos
- ▶ Software de propósito específico
 - ▶ Aquel que está diseñado para una tarea especial, muy específica y concreta (p.e. software de un TPV, cajero automático de un banco o software de gestión de un cibercafé)

Ofimática y documentación electrónica

- ▶ Tipo de aplicaciones más extendido y usado

- ▶ Procesadores de texto.

- ▶ Crear documentos de texto con gran riqueza tipográfica.
- ▶ Microsoft Word, OpenOffice.org Writer

L^AT_EX LaTeX **no** es un procesador de textos sino un **sistema de composición de textos** de alta calidad

- ▶ Hojas de cálculo

- ▶ Mediante una estructura de celdas organizadas en hojas, filas y columnas permiten realizar gran cantidad de cálculos de forma rápida y sencilla y presentarlos adecuadamente con gran facilidad y acompañados de gráficos
- ▶ Microsoft Excel, OpenOffice.org Calc, Lotus 123

- ▶ Gestores de bases de datos

- ▶ Permiten el almacenamiento y la consulta de datos organizados de forma estructurada, almacenándose para ellos los datos y las relaciones entre ellos
- ▶ Microsoft Access, OpenOffice.org Base, MySQL, Oracle

Ofimática y documentación electrónica

- ▶ Editores de presentaciones.
 - ▶ Permiten crear presentaciones con textos, gráficos, sonios, animaciones y vídeos, para ser vistas en ordenadores o mediante video proyectores
 - ▶ Microsoft Powerpoint, OpenOffice.org Impress
- ▶ Agendas y organizadores
 - ▶ Agendas electrónicas que almacenan citas, fechas, contactos, direcciones o teléfonos. Incluyen funciones como aviso de citas, organización de contactos en grupos, ...
 - ▶ Microsoft Outlook, Mozilla Sunbird, Evolution
- ▶ Visualizadores de Documentos
 - ▶ Formatos electrónicos PDF y PS
 - ▶ Adobe Reader, GhostView
- ▶ Suites Ofimáticas
 - ▶ Paquetes de software que incluyen una aplicación de cada uno de los tipos citados con anterioridad
 - ▶ Microsoft Office, OpenOffice.org

Imagen, sonido y software multimedia



- ▶ Software dedicado a la creación, edición y reproducción de contenidos multimedia (gráficos, sonidos, vídeo, animaciones, ...)
- ▶ Gráficos
 - ▶ Visualizadores
 - ▶ AcdSee, SlowView, XnView, IrfanView
 - ▶ Editores de imágenes de mapas de bits. Retoque fotográfico
 - ▶ Adobe Photoshop, Paint Shop Pro, GIMP, Corel Photo Paint

- ▶ Editores de imágenes vectoriales. Maquetación
 - ▶ Corel Draw, Macromedia Freehand, Inkscape, QuarkXpress
- ▶ Programas CAD (Computer Aided Design, diseño asistido por ordenador)
 - ▶ Autodesk Autocad, Orcad, Qcad
- ▶ Diseño 3D
 - ▶ 3D Studio MAX, SoftImage XSI, Maya, Blender
- ▶ Escaneo y OCR

Imagen, sonido y software multimedia

▶ Sonido

▶ Reproductores de sonido

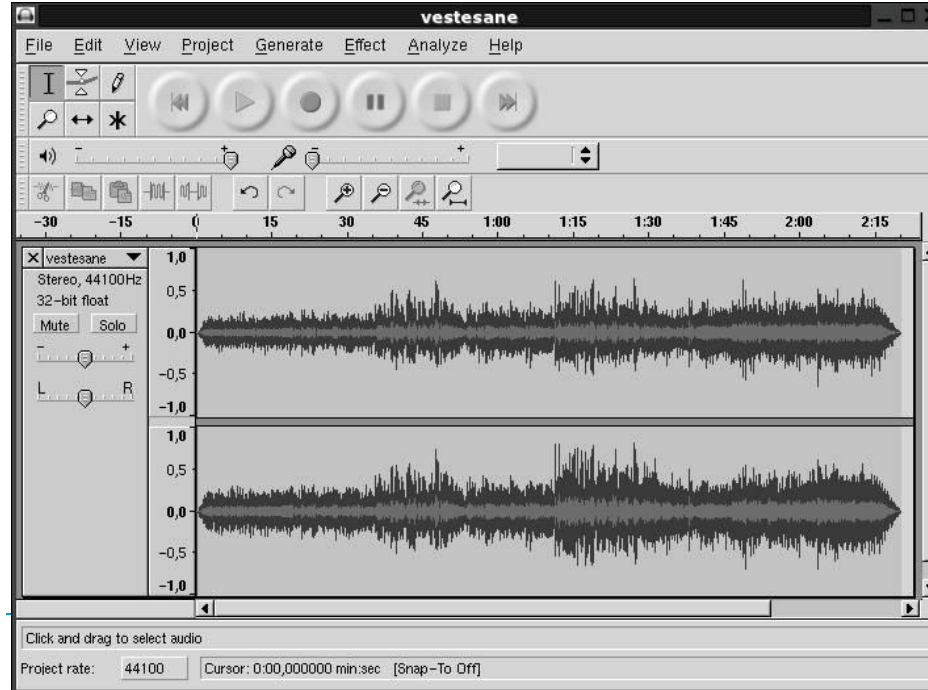
- ▶ Winamp, Sonique, Windows Media Player, VideoLan

▶ Editores de sonido

- ▶ CoolEdit, Adobe Audition, Audacity

▶ Otras utilidades de sonido

- ▶ de “ripeno”, de conversión de formatos, ...



Imagen, sonido y software multimedia

▶ Vídeo

▶ Reproductores de vídeo

- ▶ DivXPlayer, VLC (VideoLan Client), SMPlayer (mplayer front-end)

▶ Editores de vídeo

- ▶ Adobe Premiere, Pinnacle Studio, Ulead Media Studio

▶ Reproductores DVD

- ▶ PowerDVD, WinDVD

▶ Otras utilidades (ripeno,...)

- ▶ DVDShrink,...

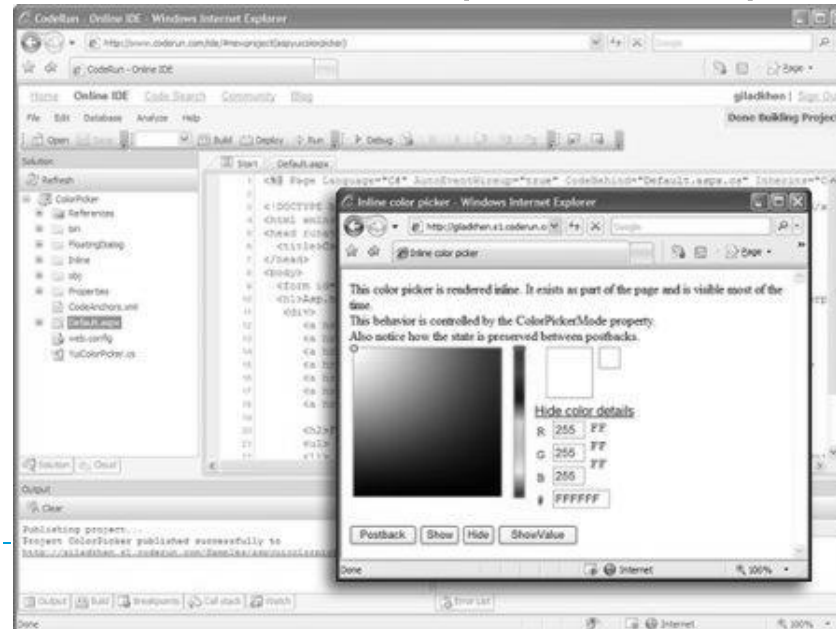


▶ Software para Media Center (centro multimedia)

- ▶ Windows Media Center, MediaPortal, XBMC, MythTV, Freevo, Moovida (ant. Elisa)

Programación

- ▶ Conjunto de herramientas que permiten al programador desarrollar programas informáticos empleando lenguajes de programación
 - ▶ Editores de texto
 - ▶ Compiladores e intérpretes
 - ▶ Enlazadores
 - ▶ Depuradores
 - ▶ Optimizadores
 - ▶ Entornos de desarrollo integrado (IDE)
- ▶ Microsoft Visual Studio, NetBeans, Eclipse, KDevelop, ...



Educación

- ▶ Software destinado a favorecer los procesos de enseñanza-aprendizaje
- ▶ Tipos de programas
 - ▶ Enciclopedias y diccionarios multimedia
 - ▶ Programas interactivos temáticos
 - ▶ Software destinado al aprendizaje del uso de un software especial
 - ▶ Plataformas para la gestión de cursos on-line
 - ▶ Distribuciones GNU/Linux orientadas a la enseñanza
- ▶ Ejemplos
 - ▶ Enciclopedia multimedia Encarta
 - ▶ Colección Aprende con Pipo
 - ▶ JClic
 - ▶ Moodle
 - ▶ Edubuntu
 - ▶ ...

Hogar, ocio y entretenimiento

- ▶ Una de las categorías más populares en la actualidad y de mayor desarrollo y difusión.
- ▶ Engloba desde los tradicionales juegos de mesa hasta enormes y costosos proyectos de producción en forma de aventuras gráficas, juegos de rol, etc.

Productividad y negocios

- ▶ Software que ayuda a mejorar nuestra productividad
 - ▶ Llevar las finanzas de un negocio, registro de artículos vendidos, obtener gráficas y estadísticas de los beneficios, ...

- ▶ Categorías:
 - ▶ Agenda de contactos
 - ▶ ContactKeeper, Outlook Express
 - ▶ Calculadoras
 - ▶ Contabilidad
 - ▶ ContaHogar, GNU Cash, ContaPlus
 - ▶ Gestión de proyectos
 - ▶ Microsoft Project, GNOME Planner
 - ▶ Terminales de punto de venta o TPV
 - ▶ Rellenar formularios
 - ▶ FormMax Filler

Cientes para servicios de Internet

- ▶ Navegadores
 - ▶ Internet Explorer, Mozilla Firefox, Google Chrome, Opera
- ▶ Clientes de correo electrónico
 - ▶ Microsoft Outlook Express, Apple Mail, Mozilla Thunderbird
- ▶ Programas de chat o mensajería instantánea
 - ▶ Windows Live Messenger, ICQ, Pidgin, Google Talk
- ▶ Reproductores multimedia de internet
 - ▶ Flash Player, QuickTime, Real Player, Windows Media Player
- ▶ Aplicaciones P2P
 - ▶ emule, BitTorrent
- ▶ Software cortafuegos
 - ▶ Firestarter (iptables front-end), Gufw (Ubuntu), ZoneAlarm, ...

Software a medida y software enlatado

- ▶ **Software a medida**
 - ▶ Aplicaciones que se realizan de acuerdo a los requerimientos de las instituciones o empresas que necesitan un programa adecuado al ámbito en que desarrollan sus actividades
 - ▶ Características:
 - ▶ Necesita un tiempo de desarrollo
 - ▶ Se adapta a las necesidades específicas de la empresa
 - ▶ Puede contener errores y ser mejorable
 - ▶ Suele ser más costoso que el estándar
- ▶ **Software enlatado o estándar.**
 - ▶ Software genérico, que resuelve múltiples necesidades, y la empresa probablemente solo empleará algunas
 - ▶ Características:
 - ▶ Ya fue desarrollado, se compra ya hecho
 - ▶ Tiene menos errores
 - ▶ Es más barato
 - ▶ Puede no ajustarse exactamente a lo que se quiere

Otras categorías de interés

- ▶ Software de tiempo real
 - ▶ Íntimamente relacionado con el mundo exterior, debiendo responder a un problema en éste en un tiempo determinado crítico en muchas ocasiones
- ▶ Software científico y de ingeniería
 - ▶ Uso extensivo de algoritmos numéricos
- ▶ Software empujado
 - ▶ Reside en memorias de solo lectura
 - ▶ Se utiliza para controlar productos y sistemas en los mercados industriales y de consumo
- ▶ Software de inteligencia artificial
 - ▶ Algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo

Tipos de archivos

Tipos de archivos

▶ Archivo

- ▶ Conjunto organizado de información del mismo tipo, almacenado en una unidad de almacenamiento
- ▶ Identificado por el nombre y el directorio (carpeta) que lo contiene
- ▶ Puede ser accedido simultáneamente por varios programas
 - ▶ El programa que acceda deberá entender la estructura del archivo (formato de archivo) y tratarla de forma adecuada

▶ Extensión

- ▶ Cadena de caracteres anexada al final del nombre de un archivo, usualmente precedida por un punto, que ayuda en algunos entornos operativos a distinguir y reconocer el tipo de información que contiene el archivo

Extensiones de archivos

- ▶ Documentos
 - ▶ doc, docx, pdf, odt, txt
- ▶ Hojas de cálculo
 - ▶ xls, xlsx, ods
- ▶ Presentaciones
 - ▶ ppt, pptx, pps, odp
- ▶ Imágenes de mapa de bits
 - ▶ bmp, jpg, gif, png, tif, raw, pcx
- ▶ Imágenes vectoriales
 - ▶ svg, cdr, wmf
- ▶ Sonidos
 - ▶ wav, mp3, wma, mid, snd
- ▶ Vídeo
 - ▶ avi, mpg, wmv, mov, 3gp
- ▶ Páginas web
 - ▶ htm, html, php
- ▶ Programas ejecutables
 - ▶ exe, com, bat
- ▶ Archivos del sistema
 - ▶ dll, inf, sys
- ▶ Paquetes comprimidos
 - ▶ zip, rar, tar, gz

Ejercicio

- ▶ Elabora un historial de los diversos sistemas operativos que has manejado así como las distintas aplicaciones que hayas utilizado desde que empezaste a hacer uso del software. ¿Cuáles han sido las diferencias más significativas que has encontrado con respecto al software actual?