



Unidad 1

Instalación de
software libre
y propietario

Implantación de
sistemas operativos



Índice



1.1. Estructura de un sistema informático

- 1.1.1. Monolítica
- 1.1.2. Capas o anillos (ring)

1.2. Máquinas virtuales. Cliente-servidor

1.3. Arquitectura de un sistema operativo

- 1.3.1. Sistemas por lotes (batch)
- 1.3.2. Sistemas distribuidos

1.4. Funciones de un sistema operativo

- 1.4.1. Controlar y gestionar el uso del hardware del ordenador
- 1.4.2. Administrar la ejecución de los procesos
- 1.4.3. Controlar el proceso de organización de la información
- 1.4.4. Controlar el acceso de los programas o los usuarios a los recursos del sistema

1.5. Tipos de sistemas operativos

- 1.5.1. Monousuario o multiusuario
- 1.5.2. Centralizado o distribuido
- 1.5.3. Monotarea o multitarea
- 1.5.4. Monoprocesador o multiprocesador

1.6. Sistemas operativos libres

- 1.6.1. Características del software libre
- 1.6.2. Las 4 libertades del software libre

1.7. Sistemas operativos propietarios

1.8. Tipos de aplicaciones

- 1.8.1. Tipos de aplicaciones especializadas en ofimática

1.9. Licencias y tipos de licencias

- 1.9.1. Copyright, copyleft y creative commons
- 1.9.2. Otros tipos de licencias

1.10. Instalación de sistemas operativos

- 1.10.1 Requisitos hardware, versiones y licencias
- 1.10.2. Medios de instalación

1.11. Gestión de varios sistemas operativos en un ordenador

1.12. Gestores de arranque

- 1.12.1 MBR (Master Boot Record)
- 1.12.2. GPT (GUID Partition Table)
- 1.12.3. Gestor de arranque Windows
- 1.12.4. Gestor de arranque Debian

1.13. Instalación / desinstalación de aplicaciones

- 1.13.1. Aplicaciones y características de Windows
- 1.13.2. Aplicaciones y características en Debian

1.14. Actualización de sistemas operativos y aplicaciones

- 1.14.1. Windows
- 1.14.2. Linux

1.15. Registros (logs) del sistema

- 1.15.1. Windows
- 1.15.2. Linux

1.16. Actualización y mantenimiento de controladores de dispositivos

- 1.16.1. Windows
- 1.16.2. Linux



Introducción

Cuando hablamos de sistemas operativos no se puede dar una correcta definición porque son varias, y su razón de existir es debida a que es una de las mejores alternativas para resolver los problemas de la vida moderna ya que el hardware por sí solo no es capaz de solucionarlos.

Si cogemos otro punto de vista debemos de tener en cuenta que **el sistema operativo debe de ser capaz de asegurarnos un correcto funcionamiento del sistema informático a la misma vez que debe de ser lo más eficiente posible**, de ahí su complejidad a la hora del desarrollo.

Como veremos en esta unidad, el sistema realizará múltiples funciones y además hay dos puntos de vista acerca del sistema operativo, su visión funcional que nos hará verlo como una especie de máquina virtual y su visión como una implementación de software al hardware

Al finalizar esta unidad

- + Sabremos de que elementos se compone un sistema informático.
- + Conoceremos las principales características del software libre.
- + Conoceremos los principales aspectos de las máquinas virtuales.
- + Identificaremos los tipos de software propietario y sus tipos de licencias.
- + Diferenciaremos los tipos de arquitecturas de los sistemas operativos según su estructura.
- + Sabremos como gestionar el arranque de un equipo.
- + Habremos aprendido todas las funciones que debe de realizar un sistema operativo.
- + Sabremos instalar distintos sistemas operativos y sus aplicaciones y actualizaciones.
- + Podremos identificar los distintos tipos de sistemas operativos.
- + Seremos capaces de gestionar los registros de nuestros sistemas



1.1.

Estructura de un sistema informático

Cuando hablamos de un sistema operativo, debemos tener en cuenta que su estructura es totalmente transparente al usuario, es decir, no se ve que corre por dentro de dicho sistema, y es por esto por lo que vamos a indicar aquí cuales son las estructuras más típicas o conocidas.

1.1.1. Monolítica

Este tipo de estructura se caracteriza porque todas las funciones del sistema operativo las realiza un solo programa de este sistema ejecutado en modo kernel (como "director" del sistema operativo). A su vez, los procedimientos que sigue el sistema si que se ven ayudados los unos por los otros, llamándose entre ellos en cuanto lo necesiten.

1.1.2. Capas o anillos (ring)

Este tipo de sistema se caracteriza por que el sistema se organiza en una serie de capas jerárquicamente otorgando a cada una de las capas una función específica.

Esta estructura la podemos orientar de dos maneras diferentes:

- > Se pueden clasificar como un sistema operativo que se compone de distintos niveles superpuestos uno inmediatamente después del anterior. El primer sistema que usó esta estructura fue el sistema THE (Technische Hogeschool Eindhoven).
- > Se puede presentar también el sistema como anillos que se van cerrando cada vez más hasta llegar al núcleo del sistema, como es el caso del sistema MULTICS.

Lógicamente, cuanto más inferior sea la capa o anillo, más protegida estará de los agentes de software externos y su acercamiento con el hardware también es mayor.

Para terminar, habría que destacar que los sistemas más modernos tratan de que el código base se mueva a sistemas superiores para así tener el mínimo núcleo o kernel posible, lo que lo haría estar más aislado y por tanto ser más ágil y seguro.



1.2.

Máquinas virtuales. Cliente-servidor

Hay ocasiones en la que es necesario ejecutar un software específico, ya sea un sistema operativo o una aplicación que podría no ser compatible con nuestro sistema físico o simplemente ser perjudicial.

La virtualización como herramienta, nos permite gracias a un cierto software, simular un entorno hardware aislado.

Esto es posible gracias a un mecanismo llamado hipervisor o monitor de máquina virtual (VMM).

Dependiendo de donde se ejecute el hipervisor, hay dos tipos:

- > Si se ejecutan sobre hardware son de *tipo 1, nativo, un-hosted o bare metal*.
- > Si se ejecutan sobre el sistema operativo anfitrión son *tipo 2 o hosted*.

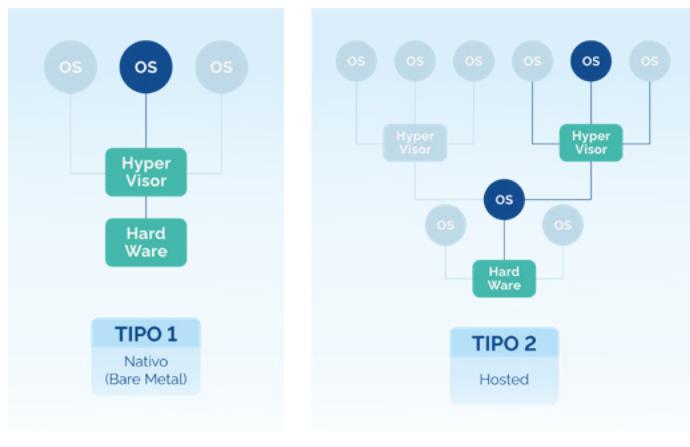


Imagen 9. Tipos de hipervisores.

Los dos conceptos principales de la virtualización son:

- > **Anfitrión (host):** aquí nos referimos a la máquina real donde vamos a instalar el programa que nos ayudará a virtualizar.
- > **Invitado o huésped (guest):** la propia máquina virtual creada en el programa de virtualización

Hoy en día la virtualización no solo se usa para realizar pruebas o ensayos de desarrollo de aplicaciones o diversos sistemas. Se ha convertido en una de las opciones más solicitadas para albergar servidores, existiendo empresas que se dedican exclusivamente a los siguiente: se tiene una infraestructura física con varios entornos de virtualización montados que a su vez albergan distintos servidores de otras empresas.

Esta técnica permite una mayor seguridad de los datos, ya que es muy fácil tener la información replicada.

Por tanto, podríamos decir que las máquinas virtuales tienen unas ventajas muy claras:

- > Es posible tener varios sistemas operativos en una misma máquina física de manera aislada, de modo que, si tenemos un fallo en el sistema o de seguridad en una máquina virtual, este no afectará a nuestro sistema anfitrión ni al resto de máquinas.
- > Hace posible que además de convivir varios sistemas juntos, estos se puedan comunicar entre sí.
- > Se puede guardar el estado de la máquina en un momento específico mediante *snapshots o instantáneas*.
- > Se puede emular un hardware que sea distinto al de nuestro sistema físico.
- > Es muy sencillo modificar los recursos de hardware asignados a las máquinas.
- > Tiene un consumo de costes y recursos reducidos
- > Se consume menos energía que con varias máquinas físicas.

Si hablamos de inconvenientes, el mayor de estos es que tanto el sistema invitado como el anfitrión pueden sufrir ralentizaciones debido a la complejidad de la virtualización.



1.3.

Arquitectura de un sistema operativo

Un sistema informático cuenta con multitud de recursos y actúa como administrador de estos además de asignarle a cada programa y usuario los recursos necesarios específicos para que realicen las tareas necesarias.

El sistema operativo consiste en primera instancia en un conjunto de programas que trabajan al unísono para poder interactuar con el sistema físico y cogiendo el control sobre el hardware del sistema.

Además, el sistema operativo realiza la función de soporte para los programas que se alojan en este, el llamado software de aplicación.

Las tareas que el sistema operativo realiza además de las anteriormente descritas son: administrar los periféricos, controlar la temperatura del procesador, transferir la información entre la memoria principal y los dispositivos de almacenamiento.

En resumen, el sistema operativo tiene la función de intermedio entre el hardware físico de la máquina y el usuario que usa dicha máquina.

Hoy en día hay que tener en cuenta que el sistema operativo ya no es exclusivo de un ordenador, ya que los encontramos en cualquier aparato electrónico que disponga de un microprocesador, o al menos en su mayoría.

El S.O. se podría describir como el primer programa que se carga en el equipo para que sirva de base a todos los demás, ya que el hardware se comunicará con los otros programas a través de este dependiendo del tipo de sistema de un modo u otro.

La manera en la que el usuario se comunica con el sistema operativo es gracias a la interfaz, que puede ser de texto o gráfica; con la función designada de que el usuario pueda transmitir sus necesidades al sistema.

1.3.1. Sistemas por lotes (batch)

Este tipo de arquitectura tiene varios nombres, uno de ellos puede ser Workload Automation & Job Scheduling.

Como la mayoría de las cosas referentes en la informática, este tipo de arquitectura ha ido evolucionando a lo largo de los tiempos hasta que se llegue al punto en el que, para muchas empresas, el tipo de procesamiento batch, es uno de los componentes indispensables de cara al éxito.

Esto último se debe a que el sistema por lotes se caracteriza por su poca interacción con el usuario, haciéndolo muy ágil y eficiente.



1.3.2. Sistemas distribuidos

Estos sistemas son una "reciente" incorporación a la informática que se está propagando a un ritmo asombroso. Su funcionamiento se basa en que haya varios equipos independientes conectados en una pequeña red entre sí. Lógicamente, cada uno de los equipos posee sus propios recursos y se comparten entre todos los demás de la red.

Uno de los principales motivos por los que se elige este sistema es porque se puede acceder a los recursos (archivos, software) de los distintos equipos que lo forman, liberando al equipo de presión y otorgando una mayor eficiencia.

Ventajas

- > Como los sistemas son independientes, no hay problema para todos en caso de que uno falle.
- > Al estar los recursos compartidos, los algoritmos se ejecutan a una mayor velocidad.
- > La carga en el equipo se reduce.
- > Los sistemas se pueden ir agregando a la red de manera sencilla, lo que los hace unos sistemas escalables.
- > El procesamiento de datos es mucho más rápido.

Desventajas

- > Si la red principal se cae, sufriremos la caída de toda la comunicación al completo.
- > A la hora de establecer un sistema distribuido, todavía no hay ningún lenguaje bien definido.
- > Como se trata de un software bastante complejo y sin mucha comprensión además de ser bastante caro, no se implementan al 100% todavía.

Ejemplos de sistemas operativos distribuidos: LOCUS, etc.



1.4.

Funciones de un sistema operativo

Un sistema operativo se encuentra formado por un conjunto de rutinas o módulos que nos ayudan a que las tareas de manejo del equipo se vean simplificadas.

Además, hay una serie de funciones que desarrolla el sistema operativo que resultan esenciales para la correcta gestión del sistema informático en general.

Un Sistema Operativo, cuando se diseña, debe de aportar las siguientes posibilidades:

> **Interfaces de usuario:** se trata del medio de comunicación entre el sistema y el usuario del equipo. En esta interfaz se cargan problemas, visualizan archivos y se realizan multitud de tareas distintas a estas anteriores, por lo que aporta una comodidad muy alta en el uso del equipo. Hay tres tipos básicos de interfaces de usuario:

- » Basadas en comandos.
- » Con uso de menús.
- » Interfaces gráficas de usuario.

> **Administración de recursos:** el sistema operativo se encarga también de gestionar los recursos del equipo y de las redes de un sistema informático.

Por ejemplo, se encargará de la administración de la CPU, de la memoria, etc. De hecho, dos de las funciones más importantes de un sistema operativo, que se engloban aquí dentro son:

- » La administración de periféricos.
- » La administración de memoria.

> **Administración de archivos:** de manera nativa en el sistema operativo viene un programa secundario que se usa para poder escribir, borrar, crear y acceder a los distintos archivos que se contengan en el equipo.

Esto también se lleva a cabo gracias a que el propio sistema tiene siempre almacenada la dirección física de los archivos independientemente del medio de almacenamiento que se encuentren.

> **Administración de tareas o control de la ejecución de programas:** controla el flujo de programas que se ejecutan en el sistema, otorgándoles distintas prioridades y recursos a medida que se necesite.

Esto se traduce en que el sistema operativo gestiona el llamado control de concurrencia. Por ejemplo, puede asignar memoria a un programa, pero pararlo drásticamente porque otro con una prioridad mucho más alta lo necesita.

> **Servicios de soporte o actualización del sistema:** cada tipo de sistema operativo tendrá un servicio de soporte distinto dependiendo de su fabricante que nos aportarán una serie de ayudas.

Las principales ayudas que se nos ofrece desde soporte independientemente del sistema que sea son:

- » Actualización de versiones.
- » Mejoras de seguridad
- » Inclusión de alguna versión nueva,
- » Controladores para manejar nuevos periféricos (a no ser que sea muy específico).
- » Corrección de errores de software.

> **Control de seguridad:** también es tarea del sistema operativo controlar que la información que se almacena no se pierda haciendo una gestión eficiente de los permisos de acceso a las aplicaciones.

Eso sí, debemos tener en cuenta que no todas las aplicaciones y/o utilidades que se usan para gestionar el sistema forman parte de este. Por ejemplo, los programas de administración de bases de datos hacen una tarea muy importante pero no forman parte del mismo sistema operativo.



1.4.1. Controlar y gestionar el uso del hardware del ordenador

Controlar y gestionar la memoria

El administrador de memoria de un sistema operativo tiene la función de gestionar el espacio de la memoria principal del equipo para que se puedan seguir alojando procesos de manera que no colapse el sistema operativo.

Este administrador de memoria por ejemplo tiene la capacidad de matar procesos en memoria para que se puedan cargar unos nuevos con mayor importancia, pero, además, da protección y seguridad a la integridad de los datos.

Algunas de sus cualidades son la capacidad de almacenamiento de datos, la velocidad de transmisión de datos y el tiempo que tardan las operaciones de lectura o escritura.

Como se verá más adelante, todos y cada uno de los procesos que se realizan en el equipo necesitan de espacio en memoria para que el código se almacene durante su ejecución.

Hay varios tipos de memoria, y se pueden clasificar del siguiente modo:

Según su función

- > **Memoria interna;** que a su vez se puede clasificar en:
 - » **Memoria principal o central.** Es la memoria donde se van a alojar los datos y los programas a ejecutar. Su velocidad de acceso es muy alta debido a la necesidad recurrente de esta, pero su capacidad de almacenamiento es mucho menor.
 - » **Memoria caché.** Al igual que la memoria central, tiene una capacidad de procesamiento muy alto, pero muy inferior a la del microprocesador, por lo que se puede sufrir un retraso cuando se ejecuten los procesos debido al tiempo que el microprocesador espera hasta que le llegue la información.
- Para esto surge la memoria caché, que también tiene una velocidad de procesamiento alta y se compagina con el microprocesador.
- » **Memoria de registros.** Son pequeñas direcciones de memoria con un fin en el tiempo que almacenan datos temporales en el momento en el que se están procesando. Son el tipo de memoria más rápido, pero su capacidad de almacenamiento es mínima.
- » **Memoria externa o secundaria;** se trata de los medios de almacenamiento como discos duros, CD etc....



Según su posibilidad de acceso

» **RAM (Random Acces Memory)**: es la memoria de acceso aleatorio. Es una memoria volátil, es decir, **cada vez que deja de recibir corriente los datos que están dentro se pierden.**

Según el funcionamiento de la memoria, tenemos dos tipos

» **SRAM o RAM Estática**. Siempre que la memoria reciba alimentación el contenido está almacenado. Aunque este tipo de memoria es de las más rápidas, también es una de las más caras.

Por ejemplo, la memoria caché se encuentra formada por este tipo de memoria RAM.

» **DRAM o RAM Dinámica**. Este tipo de memoria RAM, aunque tenga suministro eléctrico perderá el contenido con el paso del tiempo. Para que este contenido almacenado no se pierda se debería de ir reescribiendo cada cierto tiempo, proceso al que llamamos refresco de la memoria.

El rendimiento de estos tipos de memoria es menor que el de los anteriores, pero también tienen un precio menor. Para terminar, hay que saber que cada uno de los módulos de RAM que van en equipos acoplados a la placa base son de este tipo.

» **ROM: Read Only Memory**, memoria de solo lectura. Este tipo de memoria se caracteriza porque se puede acceder a los datos, pero solo leerlos, no se puede modificar.

Esto es debido a que estos datos se encuentran grabados en el dispositivo cuando se fabrica y una vez que se ha hecho ya no hay posibilidad de modificación.

Controlar los dispositivos periféricos

Un **periférico** es un dispositivo electrónico hardware que nos facilita el intercambio de información entre este y el equipo.

Una de las principales funciones del SO es controlar estos periféricos para saber en qué momento tiene el procesador que prestarles atención y permitir la entrada o salida de información.

Se conectan al equipo de manera física mediante slots o puertos y en la mayoría de los casos van directos a la placa base,

La comunicación con el periférico se realiza del siguiente modo:

1. Se conecta el aparato al equipo.
2. Dentro del dispositivo, se activa la parte electrónica, que es la denominada controladora física.
3. La controladora física manda la información al equipo.
4. Esta información la recibe el driver o controlador lógico, que es un software específico que sirve para traducir de manera bidireccional las ordenes entre periférico y sistema operativo.
5. El driver traduce la información y se realiza la acción.



Mientras que el controlador físico de un periférico viene ya dado por el mismo, el driver hay muchas veces que tendremos que instalarlo nosotros, o bien porque sea de un fabricante específico o bien porque el que tenga nuestro sistema se encuentre desactualizado.

Muchas veces los controladores son los mismos independientemente del sistema, funcionaría el mismo tanto en Windows como en Linux.

Los periféricos los podemos clasificar según su función de uso:

- > **De entrada.** Se trata de los dispositivos mediante los cuales se transmite una información al equipo. Por ejemplo, un ratón.
- > **De salida.** Se trata de los dispositivos mediante los cuales el equipo transmite información al usuario final. Por ejemplo, un amplificador de sonido.
- > **De entrada/salida.** Se trata de los dispositivos que envían información en ambas direcciones simultáneamente entre ordenador y usuario. Un ejemplo podría ser unos cascos con micrófono.

1.4.2. Administrar la ejecución de los procesos

Un proceso se puede definir del siguiente modo:

"Se trata un proceso como un conjunto de recursos necesarios para que se lleve a cabo la ejecución de un programa. Si nos basamos en el Sistema Operativo, un proceso es el objeto del cual el SO tiene que hacerse cargo, gestionarlo y darle servicio."

Un programa se trata de una serie de instrucciones que junto con unos datos almacenados se transforma en un ejecutable pasivo.

La diferencia es que un proceso es el programa en ejecución, es decir, la misma acción que realiza el programa, lo que los hace dinámicos frente a la pasividad de los programas.

Este dinamismo surge porque los procesos necesitan de recursos para su ejecución, y cada vez que haya que realizar una acción buscará esos recursos necesarios, los que puede que estén bloqueados y el proceso se tenga que mantener a la espera.

Como es el caso de los programas, cada proceso incluye un contador de programa que nos indicará la dirección de la siguiente instrucción a ejecutar en cuanto a los mismos procesos y registros de CPU se refiere.

Esto se explica para entender mejor, que, de manera lógica, los procesos se manejan mejor en un sistema operativo multiproceso, pero no porque pueda haber múltiples procesos, sino porque en este tipo de sistema operativo, cada proceso independiente de otro, lo que nos ayuda a que no se bloqueen entre ellos, ya sea porque van ligados de la mano o simplemente porque no se pueden ejecutar al mismo tiempo.



En la ejecución de procesos en los sistemas operativos multiproceso, juega el papel principal la CPU porque todos los procesos se ejecutan de manera simultánea en este componente. Es el uso de algún recurso adicional lo que hace que pueda o no ejecutarse en ese momento.

Esto es lo que en un principio se hablaba, el sistema operativo tiene la tarea de administrar esos procesos, es decir, de decidir su prioridad y de otorgar los recursos necesarios para su ejecución a cada uno conforme los vayan necesitando.

Por último, debemos saber que cada proceso se representa en el sistema operativo como un bloque de control de proceso, PCB (Process Control Block).

Dicho bloque contiene la información de un proceso necesaria para su ejecución, que son los siguientes datos: el identificador del proceso (para que lo distinga el SO), el estado en el que se encuentra el proceso, los registros de la CPU en juego, la información de ejecución en la CPU (prioridad principalmente) e información del resto de recursos como pueden ser memoria, porcentaje de disco, etc.

Estados de un proceso

Cuando usamos un sistema operativo multiproceso en el que tenemos una serie de procesos ejecutándose al mismo tiempo en un único procesador es muy probable que muchos de ellos estén bloqueados por falta de recursos o porque simplemente no haya ocurrido la operación necesaria para que el proceso se reactive.

Los procesos tienen distintos estados que se suelen explicar mediante un diagrama de proceso que muestra cómo van cambiando los procesos según las necesidades que el Sistema operativo impone.

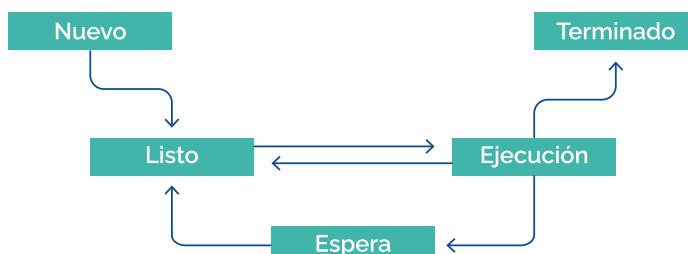


Imagen 2. Estados de un proceso.

Como se ve en el diagrama anterior, un proceso puede pasar por múltiples estados:

- > **Nuevo.** Este estado es el primer estado que puede tener un proceso, y se acaba de crear ya sea porque se ha abierto un programa o porque se ha abierto un fichero. Un proceso nuevo no es válido por sí solo ya que carece de los recursos necesarios.
- > **Listo o preparado (ready).** Este estado es cuando el proceso ya está listo, pero a falta de que el procesador comience su ejecución. Cuando un proceso está en este estado puede que haya dos vertientes, o que se ponga en ejecución, o que directamente termine porque al final no sea necesario su ejecución o porque sea necesaria su finalización para que otro proceso suceda.



- > **Ejecución (run)**. En estado, el proceso se encuentra trabajando en la CPU y ejecutando instrucciones. Hay tres vertientes: el proceso puede ejecutar todas y cada una de las instrucciones que se le han dado hasta que termine y finalice; también puede que el proceso se bloquee por necesidades del usuario o del propio sistema; por último, puede que el proceso vuelva directamente a estar en preparado porque su tiempo de ejecución se ha excedido y no ha empezado.
- > **Espera o bloqueado (wait)**. El proceso se encuentra a la espera de que suceda cualquier cambio que lo haga volver a estar preparado, o entrar en ejecución directamente e incluso finalizar si fuese necesario.
- > **Terminado**. Este es el estado en el cual se encuentra el proceso una vez que ha pasado por los demás estados y es cuando desaparece, ya no tiene retorno.

El sistema operativo gestiona los procesos mediante colas, pero cada una es diferente debido a que puede haber una por ejemplo para los procesos nuevos y otra para los que están en espera.

Es tarea del planificador del procesador el revisar estas colas y decidir qué proceso se ejecuta además de cuándo, cómo, con qué recursos y dónde.

Planificación de un proceso

El sistema operativo debe de encargarse de decidir qué procesos son los que entran en la CPU y cuáles, una vez en ella, se ejecutan, así como cuando sale el proceso. Todo esto no es simple intuición o suerte, todo esto se realiza debido a que el sistema operativo cuenta con una política de planificación de procesos.

Hay muchísimas maneras de definir las políticas de planificación de procesos, desde por orden de llegada del proceso hasta por orden de prioridad, una prioridad que es dada por el propio sistema operativo (esta suele ser la más usada). Al final, las políticas de planificación de procesos tienen la función de establecer un orden lógico a la hora de ejecutar procesos para que no haya problemas de rendimiento, funcionamiento, control, seguridad, etc.

Como hemos dicho antes, hay múltiples políticas de planificación, pero ninguna va a ser la perfecta para todos los sistemas operativos. Puede que por ejemplo en un sistema tengamos una política que se basa en la prioridad y va muy bien mientras que en otro sistema esta política cause una caída del rendimiento en el equipo.

Esto viene dado por las características que tenga cada proceso porque puede que necesiten de una prioridad muy alta porque son procesos en tiempo real, o que necesiten una carga de CPU altamente importante como puede ser el caso de un proceso de un juego arrancando.



Tenemos diferentes planificadores para nuestro sistema operativo:

- > En primera instancia nos encontramos el **planificador a largo plazo**: este se encarga de que haya un control sobre los procesos múltiples en el sistema, buscando una equivalencia en trabajos de CPU y de E/S. Además, también se encarga de suministrar procesos a la cola de planificación a corto plazo.
- > Despues tenemos el **planificador a medio plazo**: este planificador es el encargado de suspender o reactivar procesos que no tienen una mayor importancia e ir pasándolos de la memoria principal a las memorias secundarias o incluso algún dispositivo de almacenamiento externo. Este es el famoso swapping y solo se ejecuta si hay una escasez en los recursos. Como curiosidad, Linux crea una partición dedicada al swapping cuando se instala, consiguiendo un rendimiento óptimo.
- > Por último, el **planificador a corto plazo**: como hemos dicho antes, los procesos vienen dados por el planificador a largo plazo y su función es la de asignar y desasignar procesos con la CPU. Vamos a explicarlo, este planificador coge un proceso que se encuentra en estado preparado, y le asigna una CPU para que este se ponga en ejecución, después lo saca y entra el siguiente, así de manera sucesiva.

Nos encontramos con **dos planificadores a corto plazo**:

- » **El no expulsivo**: en este caso el proceso abandona la CPU directamente al terminar o si lo motiva otro proceso externo.
- » **El expulsivo**: en este caso los procesos pueden pasar al estado listado por una orden del sistema operativo sin ningún problema, es el más común en los sistemas en tiempo real, que son los más usados.

Las principales políticas de planificación de procesos son las siguientes:

- > **Primero en llegar, primero en salir (FIFO)**. Es similar a una carrera, el proceso que llegue antes a la cola de preparados será el primero en ser planificado y posteriormente ejecutado. Además, es no expulsiva, lo que no la hace una política usable en los sistemas operativos de tiempo compartido como Debian o Windows.
- > **Primero el proceso más corto (SJF)**, Cada vez que un proceso termine su ejecución en la CPU, se cogerá el siguiente por ser el que menor tiempo de ejecución tenga. Vuelve a ser no expulsiva en su mayoría, aunque hay algunos casos con procesos muy pequeños en los que se usa SRTF, que sí es expulsiva.

- > **Prioridades**. Como hemos comentado antes, un proceso tiene una serie de prioridades asignadas ya sea por parte del sistema operativo o porque el propio usuario las haya dividido así.

En este caso, todos los procesos están divididos en colas dependiendo de su prioridad y el planificador elegirá los procesos de la primera cola siguiendo a su vez también el método FIFO. Una vez la primera cola se haya acabado pasará a la siguiente, y así sucesivamente hasta que terminen todas las colas.

Esta técnica puede ser tanto expulsiva como no expulsiva, lo que se traduce en que habrá procesos que no se ejecuten debido a los cambios, pero su solución pasará por aumentar de manera progresiva la prioridad.

- > **Turno rotatorio (Round-Robin)**. Esta es la mejor técnica a la hora de usar un sistema de tiempo compartido. Su funcionamiento consiste en que cada cierto tiempo se genera una interrupción de reloj periódica y cada proceso cuenta con un cuanto de tiempo máximo al que llamamos quatum.

Cuando este tiempo termina el proceso que esté en ejecución pasa a estar listo y se sigue entonces la técnica FIFO para que entre el siguiente.

- > **Retroalimentación**. El último tipo de planificación que vamos a ver es el que consiste en el trabajo de diferentes colas de procesos en estado de listo o preparado, pero con distintas políticas.

Esto hace que los procesos vayan pasando por las colas dependiendo del estado u otra política hasta que lleguen a la última cola y terminen.

En cuestión de sistemas operativos los procesos también se pueden dividir o agrupar, Por ejemplo, en Linux, hay tres categorías: interactivos, tiempo real o por lotes.

En Windows, por ejemplo, la planificación de los procesos usa colas múltiples ordenadas por prioridades.

Para MAC OS se usa el mismo modo que en Windows.



1.4.3. Controlar el proceso de organización de la información

Según la RAE, un fichero o archivo es:

"Conjunto de datos almacenados en la memoria de una computadora u otro dispositivo electrónico, que puede manejarse con una instrucción única."

El Sistema Operativo tiene la utilidad llamada sistema de ficheros mediante la cual se manejan, organizan y almacenan los ficheros de manera permanente en el equipo o en un dispositivo externo.

Los sistemas de ficheros manejan dos tipos de objetos fundamentales:

- > **Los ficheros regulares (file):** se trata de una unidad lógica de memoria donde se almacena cualquier tipo de información que suele tener un indicativo de su contenido, por lo general un nombre.

Las características de cada fichero dependerán del sistema de ficheros que se use, y por la misma razón, estos varían dependiendo del sistema operativo, por ejemplo, Windows usa NTFS.

- > **Los directorios (directory):** son objetos de memoria que sirven para almacenar y organizar ficheros. Incluso, dentro de unos directorios puede haber más directorios.

En la mayoría de los sistemas operativos existentes existe un directorio raíz o root, que es el directorio central del que cuelgan todos los demás archivos y directorios.

Tanto ficheros como directorios tienen una serie de atributos que nos indican el tipo de archivo o directorio, los usuarios que pueden operar sobre ellos y que operaciones pueden realizar estos usuarios sobre los ficheros o directorios.

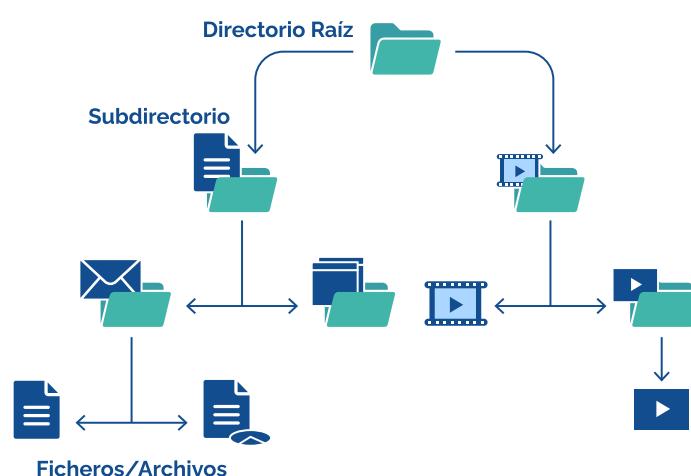


Imagen 3. Esquema de la estructura de directorios.



Los directorios suelen tener una estructura jerárquica dividida en ramas, 'en árbol'. En estos sistemas jerárquicos normalmente la ubicación de un directorio o archivo viene dada por una ruta o path que suele escribirse del siguiente modo, una sucesión de nombres de directorios y subdirectorios que terminan con el nombre del archivo o directorio final.

Estos nombres se separan por caracteres especiales que cambian dependiendo del sistema operativo, las principales son '\' para Windows y '/' para Linux.

Por ejemplo, un archivo de música que se llame informática.mp3 y se encuentre en la carpeta de escritorio de Juan, en Linux sería del siguiente modo:

`/home/juan/desktop/informática.mp3`

En cambio, en Windows, si está en el disco C: sería:

`C:\Users\Juan\Escritorio\informática.mp3`

En un fichero o directorio se pueden realizar de manera común y normal las siguientes operaciones:

- | | |
|-------------|------------|
| > Crear | > Leer |
| > Renombrar | > Escribir |
| > Abrir | > Cerrar |
| > Copiar | > Borrar |
| > Buscar | |

Todas estas operaciones van sujetas a que los usuarios tengan permiso para realizarlas, si no los tienen será imposible que puedan llevarse a cabo cualquiera de las operaciones anteriores.

Por último, hay que saber que para que un sistema de ficheros pueda ser instalado en el equipo o creado, es necesario que se haga en una partición específica de un disco.

La partición de un disco es una división lógica de una unidad física de almacenamiento y funciona de manera independiente a las demás particiones, porque se podría hasta instalar dos sistemas operativos en dos particiones diferentes.

Eso sí, en una misma partición solo puede haber un único sistema de ficheros.



1.4.4. Controlar el acceso de los programas o los usuarios a los recursos del sistema

En todos los sistemas operativos debe de haber un control para el acceso a los datos y a la ejecución en el sistema de ciertos programas con la intención de preservar la seguridad e integridad del equipo y los datos, más aún en los equipos que tienen un sistema multiusuario.

Es por esto por lo que los sistemas cuentan con unos mecanismos que nos ayudan a gestionar permisos y privilegios. Es tarea del usuario con ayuda del sistema la gestión principal de estos permisos.

Elementos bajo control. Sujetos y Objetos del sistema

Los principales elementos sobre los que se aplicarán los controles de acceso que se dispongan por parte del sistema o del usuario van a ser de manera frecuente o el sujeto o el objeto:

- > **Sujetos.** Estos son todos los usuarios, procesos, programas, etc., que operan en el sistema sobre los recursos del propio equipo.
- > **Objetos.** Estos son los ficheros, directorios, servicios, dispositivos de E/S etc., que no operan directamente sobre los recursos del sistema.

Tipos de control de Acceso

Si nos fijamos en los tipos de control de acceso que hay en los sistemas nos encontramos con que hay tres tipos principales: Control de Acceso discrecional (DAC), Control de Acceso Basado en Roles (RBAC) y Control de Acceso Obligatorio (Mandatory Access Control, MAC).

- > **Control de acceso discrecional (DAC).** Este método de control de acceso se basa en la identidad de los usuarios para controlar el acceso a los diferentes sujetos u objetos del sistema.

Suele ser el dueño del objeto/sujeto el que determina que usuarios pueden acceder a sus elementos y que operaciones pueden realizar.

Normalmente, son los atributos comunes, lectura, escritura, ejecución, etc...., los que se establecen para aplicar los permisos.

No obstante, hay multitud de opciones para compartir ficheros u otros elementos con usuarios sin permisos.

> **Control de acceso basado en roles.** Este método de control de acceso se basa en la estructuración por roles o perfiles preasignados con ciertos privilegios y opciones de acceso a los archivos y programas del sistema.

Este sistema es muy usado en empresas con gran tamaño en la que el control de acceso discrecional resulta poco eficiente.

Normalmente tiene una estructura jerárquica y también se pueden otorgar los permisos necesarios por grupos, por ejemplo, los departamentos de una empresa: marketing, ventas, recursos humanos, etc.

> **Control de acceso obligatorio.** Este método es una implementación a los otros dos anteriores para añadir un punto superior de seguridad sobre los datos.

Este sistema MAC funciona con una predisposición de las políticas de acceso, es decir, queda predefinido que accesos están disponibles para cada usuarios o grupos y que operaciones se pueden realizar.

Estas 'etiquetas' preestablecidas no se pueden alterar bajo ningún concepto con ninguna de las anteriores políticas de control de acceso y es tarea del administrador configurar y aplicar las políticas centrales MAC.

Como hemos dicho, realmente, ninguna de las anteriores políticas es excluyente una de la otra, sino que se implementan y compenetran para otorgar la mayor seguridad posible a los sistemas.

Solo debemos tener en cuenta lo explicado en el control de acceso obligatorio.



1.5.

Tipos de sistemas operativos

1.5.1. Monousuario o multiusuario

Si nos centramos en la cantidad de usuarios que pueden trabajar al mismo tiempo sobre el mismo sistema, tenemos dos tipos:

Monousuario

Los recursos del sistema están disponibles para un solo usuario al mismo tiempo, sin importar la cantidad de recursos que se tengan ni el número de tareas simultáneas que se puedan ejecutar al mismo tiempo.

Multiusuario

En estos sistemas los recursos del equipo se encuentran repartidos entre varios usuarios que pueden trabajar al mismo tiempo sobre el mismo equipo y sistema.

Esto puede ser por conexiones por red, por terminales gráficas o por otro modo distinto.

De nuevo, es independiente del número de recursos y de tareas que se puedan ejecutar, pero cuantos más usuarios se conecten, más recursos se consumen, y cuantos menos recursos, menor es el rendimiento.

1.5.2. Centralizado o distribuido

Si nos fijamos en la manera de ofrecer los servicios:

Sistemas centralizados

Un gran equipo o servidor central denominado mainframe se encarga de todo el procesamiento de los datos y cada usuario se conecta de manera remota mediante estaciones cliente, comúnmente llamados terminales tontos o bobos, que no tienen capacidad de procesamiento y carecen de memoria y procesador.

Sistemas distribuidos

Es como un sistema en red en el que los recursos se comparten en red creando un gran sistema, pero con una máquina virtual. Estos sistemas funcionan totalmente transparente al usuario, que no sabe dónde se alojan esos recursos.



1.5.3. Monotarea o multitarea

Si los clasificamos por el número de tareas que pueden procesar al mismo tiempo:

Monotarea

Solo se puede ejecutar una tarea o proceso al mismo tiempo porque cuando el proceso entra en ejecución, todos los recursos del sistema van destinados a este.

Multitarea

El Sistema Operativo es capaz de realizar multitud de tareas al mismo tiempo porque se encarga de gestionar y repartir los recursos en función de las necesidades de cada proceso, principalmente el consumo de CPU. Todos los Sistemas operativos multiusuario son multitarea.

1.5.4. Monoprocesador o multiprocesador

Si evaluamos el número de procesadores que el sistema puede manejar al mismo tiempo:

Monoprocesador

La CPU se encuentra compuesta por un solo procesador para atender todo el funcionamiento del equipo.

Multiprocesador

La CPU se encuentra compuesta por varios procesadores que ayudan a distribuir la cantidad de trabajo y por lo tanto mejora el rendimiento del equipo.

Pueden trabajar de forma simétrica, es decir, enviando la misma carga de trabajo a cada uno de los procesadores o de forma asimétrica, donde se enviarán los procesos independientemente de la carga a distintos procesadores.



1.6.

Sistemas operativos libres

El software libre nació como término promovido por el proyecto GNU y apoyada por la Free Software Foundation. Este tipo de software se caracteriza porque los usuarios pueden visualizar libremente el código fuente y operar con él sin restricciones algunas.

Hay que tener claro que software libre no tiene por qué ser sinónimo de software gratuito.

Y también al revés, es decir, puede que haya un software gratuito pero que no se pueda distribuir o modificar.

Hay diferentes licencias para el software libre, como por ejemplo Apache.

1.6.1. Características del software libre

El Software libre tiene como principales características las siguientes:

Desarrollo distribuido basado en la comunidad

Prácticamente todas las distribuciones de software libre han empezado del mismo modo, con unas pocas personas o incluso una sola que, de manera voluntaria comienzan a desarrollar un proyecto.

Conforme dicho proyecto va cogiendo forma y obteniendo éxito, se crea la denominada comunidad de proyecto donde varios miembros que pertenezcan a dicha comunidad conforme van creciendo aportan sus ideas con respecto al desarrollo de dicho software. Para que el proyecto siga adelante sin errores se suele usar el sistema CVS.

Este CVS se trata de un sistema de control de revisiones de software cliente-servidor. Mientras que permite que varios desarrolladores trabajen conjuntamente al mismo tiempo, va realizando un seguimiento a los archivos que se crean para comprobar que están correctos y que no hay fallos según lo que previamente se establezca. Además, es software libre y gratuito.

Hoy en día ya se trabaja todo desarrollo en servidores virtuales, lo que ofrece una mayor flexibilidad a la hora de trabajar, así como unos costes más reducidos, más facilidad de asegurar los datos y capacidad de mejorar los recursos casi instantáneamente.

Diseño modular

Como el software libre y de código abierto va recibiendo cambios de manera continua, es necesario que haya un desarrollo modular.

Es decir, el software se irá dividiendo en diferentes módulos pequeños sobre los que se operará y se establecerá el diseño.

Esto se realiza con el fin de poder ir añadiendo los módulos conforme estén terminados sin que deje de funcionar el software principal, así no hay que parar el diseño o desarrollo en ningún momento.



Reutilización

La reutilización consiste en usar un mismo segmento o trozo del código fuente del software para agregar nuevas funciones o características más adelante con poca modificación. Este proceso es algo que ayuda mucho en la producción de software libre donde tantas personas desarrollan al mismo tiempo.

Como, además, el software libre tiene una distribución no acotada, esta misma reutilización se puede usar para varios proyectos distintos sin ningún problema.

Existen grandes repositorios en los que la gente pública trozos de código reutilizable para que los desarrolladores inserten en sus proyectos.

Distribución y licencias

Toda la distribución del software libre se basa en Internet. Ya sea mediante sitios web, repositorios, listas de correo, todos los medios son válidos a la hora de la ampliación y distribución del código abierto. Y hay varios tipos de licenciamientos dentro del software libre, como GPL, LGPL, licencia artística, etc.

Mecanismos de recompensa

Aunque siempre se ha caracterizado por ser en casi todo su ámbito gratuito, el software libre ha ido consiguiendo un reconocimiento que hace que se haya establecido un modelo de negocio.

Por ejemplo, hay casos en los que, aunque software libre, no es gratuito, como puede ser Red Hat o Fedora. También hay ocasiones en las que el software gratuito es el principal y luego hay packs de expansión o complementos con coste adicional.

Así, hoy en día el software libre es casi tan rentable como el software propietario.

1.6.2. Las 4 libertades del software libre

El Software libre garantiza de manera estricta 4 libertades fundamentales, y en caso de que alguna de estas falte, pasaría a ser entonces software propietario.

Estas 4 libertades son:

1. **Utilizar.** Se puede usar el software libre para cualquier propósito que veamos conveniente dentro de la legalidad estipulada sin limitaciones geográficas o de licencias.
2. **Compartir.** Como sabemos, es de libre distribución para cualquier ámbito.
3. **Estudiar.** El software libre puede estudiarse por cualquiera que quiera ya que tampoco hay reglas de no divulgación al respecto.
4. **Mejorar.** Al poder ser editado por cualquiera, también se puede mejorar y compartir de manera pública con la comunidad.



1.7.

Sistemas operativos propietarios

El software privativo o propietario recibe su nombre debido a que se caracteriza porque tienen su origen en una entidad privada y no se puede divulgar ni distribuir de manera libre y se restringe a una serie de licencias a menudo muy restrictivas.

Para saber si un software es propietario o no, no basta con verlo, hay que probar si se puede acceder a su código fuente. Si el acceso al código fuente es restrictivo quiere decir que se trata de software propietario.

Este código fuente solo puede ser visto y tratado por los desarrolladores que trabajen con este software, ya que a veces, el acceso por parte de terceros incluso quebranta la ley.

En el caso del software privativo, hay que tener en cuenta que si que se podría usar el código por parte de terceros siempre que el usuario que tenga los derechos de autor lo exprese claramente.

Vamos a aclarar esta situación, el propietario de los derechos de autor de un software propietario no tiene por qué ser simplemente una sola persona, de hecho, suele ser una entidad o una organización más grande; y dentro de esos derechos de autor se puede expresar que se puede hacer y no con el software desarrollado, sin importar las restricciones que se apliquen.

En definitiva, podemos decir que realmente el software propietario nace como lo opuesto al software libre porque oprime alguna de las 4 libertades antes mencionadas.

Los sistemas operativos propietarios por excelencia son los desarrollados para la familia Microsoft Windows.

1.8.

Tipos de aplicaciones

Un sistema operativo por sí solo no ofrece herramientas para propósitos específicos. Las aplicaciones nos ofrecen esa posibilidad de realizar funciones específicas. Existen muchas empresas dedicadas a crear aplicaciones y venderlas en forma de paquete, conocidos como suites, como por ejemplo Microsoft Office.

Dentro de las aplicaciones existen dos principales tipos:

5. **Aplicaciones multiplataforma:** Es el tipo de aplicaciones que pueden funcionar en dos o más sistemas operativos. Se pueden denominar también aplicaciones multientorno.
6. **Aplicaciones multiusuario:** Son las aplicaciones que pueden ser gestionadas por varios usuarios con permisos para crear, modificar y borrar archivos que generará la aplicación. Los permisos estarán administrados por un administrador.



Imagen 4. Logotipo de Microsoft Office



1.8.1. Tipos de aplicaciones especializadas en ofimática

Hoy en día existen infinidad de tipos de aplicaciones dedicadas a funciones más generales que están presentes en ordenadores de entorno doméstico. Las más relevantes las veremos a continuación:

- > **Procesadores de texto.** Este tipo de aplicaciones permiten crear y manipular documentos de texto, permitiendo ajustar el formato del texto incorporando en él gráficos u otros elementos. El resultado que obtenemos al procesar el texto es literalmente lo que se ve en la pantalla de nuestro ordenador. Un ejemplo de procesador de texto es **Microsoft Word**.
- > **Hojas de cálculo.** Estas aplicaciones permiten realizar cálculos más o menos complejos. Una de sus características más importantes es la actualización automática de los cálculos cuando modificamos los datos de partida. El claro ejemplo de hoja de cálculo es **Microsoft Excel**.
- > **Bases de datos.** Son herramientas para almacenar, ordenar y consultar información de forma muy eficiente. Uno de los mejores programas para gestionar una base de datos es **MySQL**.
- > **Manipulación de gráficos e imágenes.** Permiten manipular y retocar gran cantidad de imágenes y gráficos realizando todo tipo de cambios sobre el original. Un ejemplo de este tipo de aplicación sería **Photoshop**.
- > **Presentaciones gráficas.** Es un conjunto de gráficos, imágenes y textos destinados a presentar de forma esquemática una cuestión en concreto. Por ejemplo, **Canva**.
- > **Clientes de correo.** Permite redactar mensajes de correo electrónico para su envío y recepción, así como almacenar los correos electrónicos entrantes junto a sus archivos, etc. Uno de los más utilizados es **Microsoft Outlook**.
- > **Agenda electrónica.** Son útiles para gestionar citas, reuniones y tareas. La mayoría de las veces viene integrada en el propio cliente de correo o en el sistema operativo.
- > **Otras aplicaciones ofimáticas.** Son menos usadas y de menor relevancia como las calculadoras, las herramientas de reconocimiento fácil y del habla, etc.

También existen gran cantidad de aplicaciones específicas diseñadas para ámbitos profesionales, como, por ejemplo:

- > Aplicaciones de contabilidad, facturación, etc.
- > Aplicaciones para la gestión de almacén.
- > Aplicaciones médicas.
- > Aplicaciones para el desarrollo de programas informáticos.
- > Aplicaciones de dibujo técnico.
- > Aplicaciones dedicadas a la ingeniería.



1.9.

Licencias y tipos de licencias

Nos referimos a licencia de software como la autorización expresa del propietario de un software mediante la que se regulan que actividades expresas se pueden realizar con dicho material.

De manera general hay que aceptar una serie de condiciones que sirven a modo de contrato para que ante la ley queden dispuestos estos requerimientos.

Existen varios tipos de licenciamiento para las aplicaciones en las que se expone de qué manera son de más a menos restrictivas y las veremos a continuación.

1.9.1. Copyright, copyleft y creative commons

En internet hay mucha desinformación sobre los derechos de autor. Simplemente publicando un enlace de un video de YouTube en tu propio blog, estarías infringiendo una de las reglas de copyright.

A continuación, veremos las diferencias entre los tres tipos de derechos de autor:

> **Copyright:** se define como un conjunto de normas donde todos los derechos de la obra son del propietario de esta. No puede ser utilizada ni modificada sin permiso. Es la ley más restrictiva.

Con el paso del tiempo los derechos de dicha obra pueden pasar a ser de dominio público. Debemos tener en cuenta dos aspectos:

- » Registrar una obra no es obligatorio, ya que con su creación se le asignan automáticamente sus derechos.
- » En el caso de que en una obra no haya ningún aviso legal se considera que la obra posee Copyright.

> **Copyleft:** es el tipo de licencia que garantiza el derecho a cualquier usuario de utilizar, modificar y redistribuir una obra.

Deben mantenerse siempre las mismas condiciones de utilización y difusión. Puede ser desde un programa informático a una obra de arte.

> **Creative Commons:** en este tipo de licencias los autores pueden decidir ceder a otros usuarios alguno de los derechos de sus obras y mantener otros.

Estas licencias son gratuitas. Al igual que el Copyright, su registro no es obligatorio.

1.9.2. Otros tipos de licencias

> **Software comercial.** Es cualquier tipo de software que se comercialice y que haya que pagar por él. Puede ser tanto software libre como propietario.

> **Software de dominio público.** Se trata del software que usa la licencia copyright.

> **Freeware.** Son programas en los que la redistribución está permitida, pero nada más, no se puede modificar. Suelen tener su código fuente disponible.

> **Shareware.** También se puede redistribuir, pero con limitaciones en tiempo y forma y suele necesitar un pago adicional para poder usarse de manera completa. Suelen no tener el código fuente disponible.

> **GPL.** La licencia por excelencia del código abierto, de hecho, se la considera la licencia de software libre con protección heredada.

Como es para software libre se puede pensar que no tiene mucho sentido que haya un licenciamiento, pero es que esta licencia lo que trata es proteger las 4 libertades del software libre, para que ningún usuario pueda apropiarse del programa en cuestión y coartar alguna de estas libertades.

Dentro de esta hay varias licencias, de la que destaca la Licencia Pública General de GNU (GNU GPL). Esta última licencia otorga los derechos de autor al autor del software para que establezca como se puede distribuir dicho software.

> **DFSG.** Se trata de la licencia que va vinculada entre Debian y la comunidad de usuarios.

> **BSD.** Esta es la Licencia de Software libre sin protección heredada, esta licencia hace que se pueda manipular el software a gusto del consumidor, es realmente como si no hubiera licencia porque permite que se haga con el software tanto venta y dinero como usarlo de manera personal.



1.10.

Instalación de sistemas operativos

A la hora de instalar un sistema operativo hay que tener en cuenta ciertos aspectos ya que este es el software base del ordenador. **El sistema operativo se encarga de la comunicación entre los distintos programas y el usuario que maneja el equipo.**

Aunque como veremos más adelante el sistema operativo suele guiarnos a la hora de la instalación, es conveniente realizar una planificación de la instalación de los sistemas operativos. Para esta planificación es necesario que se revisen los siguientes aspectos:

- > Elegir qué sistema operativo es el más adecuado para cada uno de nuestros propósitos, pues podemos usar más de uno al mismo tiempo.
- > Debemos de saber que elementos de hardware son los necesarios para que el sistema se pueda instalar sin problemas y funcione correctamente.
- > Decidir, como hemos dicho antes si se instalarán varios sistemas operativos. En caso afirmativo deberemos de tener en cuenta el espacio en disco disponible y poner en práctica la creación de particiones específicas para cada sistema como veremos más adelante.
- > Seleccionar que sistema de archivos dentro de los soportados usaremos en los discos.
- > Saber si se va a instalar de 0 o solo se actualiza el sistema a una versión más moderna.
- > Conocer el medio de instalación que se usará, live CD, por red, sistema virtual, etc.
- > Realizar un estudio previo de la instalación en base a los manuales que el fabricante aporte para saber que debemos instalar y si hace falta algún requisito previo.
- > Por último, intentar tener todos los dispositivos electrónicos que tengamos conectados al equipo para que el sistema compruebe su compatibilidad, quizás las compatibilidades nos hagan cambiar de elección de sistema operativo.

También deberíamos de tener en cuenta que hoy en día la mayoría de las empresas u organizaciones implementan los llamados sistemas en red. Estos sistemas se basan en una arquitectura LAN o WAN, dependiendo del tamaño de la empresa.

Es por esto por lo que hoy en día todos los sistemas operativos suelen contar con una arquitectura de red implementada en el propio sistema que nos ayuda a que haya una conexión latente.

Es decir, actualmente cualquier sistema operativo debe de poder implementarse a un sistema en red para que se habilite la compartición de recursos entre los distintos equipos.

En los entornos de red podemos distinguir dos tipos principales de sistemas operativos que trabajarán de distinto modo dependiendo de su objetivo:

- > **El software que se instala en un servidor:** este software se instala en un equipo por lo general con unos recursos ampliados y sirve de conexión entre los distintos equipos en red.

Este **suele ser el controlador de dominio y sirve de maestro para los equipos clientes que se conectan a él para obtener información y recursos.** Estos servidores se incluyen en la arquitectura de red cliente/servidor.

- > **El software que se instala en un equipo cliente:** es un software totalmente normal y corriente que permite una conexión a internet y pone sus recursos a disposición del equipo maestro para que este gestione toda la red.

Se les suele denominar **estaciones o equipos cliente** y suele estar en coordinación con las demás estaciones.

Lógicamente, se necesita de una conexión de red para poder trabajar con estos sistemas.



1.10.1 Requisitos hardware, versiones y licencias

Windows

Son los sistemas operativos por excelencia de Microsoft y actualmente el más usado es Windows 10, que tiene las siguientes versiones:

- > **Windows 10 Home.** Se trata de la edición más básica de todas y la que se instala por defecto en cientos de PCS y portátiles de uso doméstico. Esta versión incluye todas y cada una de las características necesarias para que se trabaje de manera fluida, pero sin llegar a ciertas características que nos ayudan a enfocar el sistema a un entorno más avanzado y orientado a la empresa.
- > **Windows 10 Pro.** Es igual que el sistema anterior, con las necesidades básicas para el desempeño de tareas, pero añade unas características más particulares para empresas o PYMES.

Por ejemplo, se añade el sistema de cifrado mediante bitlocker, que nos ayuda a proteger nuestro disco y también la característica de Escritorio Remoto. Si hubiese algo que destacar además de los anteriormente nombrados sería que en este modelo ya se aplican las políticas de grupo que el controlador nos dictamine.
- > **Windows 10 Enterprise.** Aún más orientado a la empresa que el anterior, este tipo de licencia solo se puede obtener como licencia por volumen por parte de Microsoft. Entre sus características más aclamadas está el llamado DirectAccess que crea una VPN para que los usuarios accedan a una red interna y el AppLocker, que permite que el administrador restrinja el acceso al usuario final a ciertas aplicaciones.
- > **Windows 10 Education.** Esta edición nos puede llevar a la confusión por su nombre, porque no está orientada a los alumnos, sino a las instituciones educativas en sí. Es exactamente igual que Windows Enterprise con la diferencia de que en esta compilación, Cortana está deshabilitada.
- > **Windows 10 Pro Education.** Exactamente igual que la anterior, pero con unas pequeñas mejoras y lanzada en junio de 2016 en principio con el propósito de que aceptase más licencias especiales en cuanto a educación se refería. Solo tuvo efecto verdadero en Estados Unidos y Australia.
- > **Windows 10 Mobile.** Es la edición que se orienta a dispositivos móviles. De momento sigue a la sombra de Android e iOS, pero se espera que con los años vaya creciendo exponencialmente.

Por último, a finales de 2021 surgió Windows 11, que pretende mejorar ciertos aspectos de Windows 10 y hacer una interfaz gráfica más intuitiva y estética.



Imagen 5. Interfaz gráfica de Windows 11

Requisitos de instalación de Windows 10

- > **Procesador:** mínimo a 1 GHz o un sistema en un chip.
- > **RAM:** 1 GB para sistemas de 32 bits o 2 GB para sistemas de 64 bits.
- > **Espacio en disco duro:** 16 GB para sistemas de 32 bits o 32 GB para sistemas de 64 bits.
- > **Tarjeta gráfica:** DirectX 9 mínimo.
- > **Pantalla:** mínimo una resolución de 800x600.
- > **Conexión a internet:** cualquier conexión simplemente con la necesidad de descargar actualizaciones y parches de seguridad.

Requisitos de instalación de Windows 11

- > **Procesador:** mínimo a 1 GHz, pero con dos núcleos y un procesador de 64 bits o un sistema en un chip.
- > **RAM:** mínimo 4GB de RAM.
- > **Almacenamiento:** mínimo 64 GB de almacenamiento.
- > **Tarjeta gráfica:** se necesita una DirectX 12 o posterior.
- > **Pantalla:** mínimo una pantalla de 720 p con más de 9 pulgadas en diagonales y con 8 bits por color.
- > **Conexión a internet:** necesaria para actualizarse además de tener que estar en continuo contacto con la cuenta de Microsoft.
- > **Nuevas características necesarias:** Firmware del sistema. Es necesario contar con un sistema por UEFI y un módulo de plataforma segura o TPM en su interior.

Hay que tener en cuenta que todos estos requisitos son los mínimos, y que siempre se recomienda el doble de estos para que el sistema nos dé un rendimiento óptimo.

A la hora de instalar Windows también es recomendable saber que por lo general las licencias son de un uso por ordenador y en la mayoría de los casos se ligan a algún hardware del equipo que sea inamovible como la BIOS, pero es un aspecto importante el guardar la licencia a la hora de cambiar de equipo de disco mismamente.



Linux

Linux es una de las variantes del sistema operativo UNIX más usada en la actualidad. Puede ser ejecutada en la mayoría de las arquitecturas de ordenadores (S.O. multiplataforma).

Parte de su software se desarrolla bajo el proyecto GNU, y, por lo tanto, es libremente distribuible (cualquier usuario puede desarrollar nuevos módulos bajo las condiciones GLP de licencia pública).

Existen muchas distribuciones que trabajan con la última versión estable del núcleo. Entre las distribuciones más usadas de Linux podemos encontrar:

- > **Red Hat Enterprise Linux** (<http://www.redhat.com/>) también conocido por sus siglas RHEL es una distribución comercial de Linux desarrollada por Red. Su software está empaquetado en formato RPM.
- > **OpenSUSE** (<http://www.opensuse.org/es/>): es el nombre de la distribución y proyecto libre desarrollado por Novell y AMD para el desarrollo y mantenimiento de un sistema operativo basado en Linux.
- > **Debian, Proyecto debian** (en inglés debian Project, <http://www.es.debian.org/>): es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo GNU basado en software libre precompilado y empaquetado en un formato sencillo en múltiples arquitecturas de computador y en varios núcleos.
- > **LinuxMint** (<https://www.linuxmint.com>) o **Ubuntu** (<http://www.ubuntu.com/>): proporciona un sistema operativo actualizado y estable para el usuario promedio, con un fuerte enfoque en la facilidad de uso y de instalación del sistema.

Al igual que otras distribuciones se compone de múltiples paquetes de software normalmente distribuidos bajo una licencia libre o de código abierto.

- > **Mandriva** o **Mandriva Linux** (<http://www.mandriva.com/>): es una distribución Linux publicada por la compañía francesa Mandriva destinada tanto para principiantes como para usuarios experimentados.

Existen distribuciones nacionales en las que las comunidades autónomas han creado su propia distribución promovida por el aporte de software gratuito en las administraciones, principalmente en educación. Están basadas en Debian y podemos encontrar Guadalinex, Llurex, Molinux, Linuka, etc.



Algunas consideraciones previas antes de realizar la instalación son:

- > El proceso de instalación es responsabilidad del usuario Administrador del sistema llamado root.
- > Antes de realizar la instalación en un equipo que contenga datos y programas es necesario realizar una copia de seguridad de toda la información. Creando una imagen de las particiones o de todo el disco con alguna utilidad disponible en el mercado (Ghost, Acronis, System-Rescue, etc.), podemos realizar backups en algún soporte auxiliar de la información con la propia herramienta disponible en el sistema operativo instalado.
- > Hay que asegurar de que se dispone de todos los programas que se desea instalar y los que se emplean habitualmente.
- > Recopilar todos los controladores de hardware que necesita el ordenador y comprobar su compatibilidad con el sistema a instalar (consultar la página de la distribución elegida).
- > Recopilación de datos o parámetros referentes a la configuración de la red de ordenadores, en el caso de que el ordenador formara parte de ella.
- > Lo primero que debemos considerar antes de iniciar la instalación de Linux es el tipo de distribución que vamos a utilizar dependiendo de las necesidades y del hardware disponible
- > Decidir el tipo de instalación según el trabajo o función que realizará en el entorno de red:
 - » Servidor
 - » Terminal o estación de trabajo.
- > Debemos decidir si va a trabajar sólo con Linux o compartir el disco duro con otros sistemas operativos, con lo que se deberán gestionar las particiones y el espacio de disco disponible.

Algunos administradores de sistemas antes de realizar la instalación analizan y gestionan las particiones del disco con alguna herramienta que permita crear, realizar copias de seguridad, formatear, eliminar y modificar particiones como son el Eaesus Partition Manager, Partition Magig, Gparted, etc., de manera que al llegar al apartado del proceso de instalación de decidir dónde alojar el sistema seleccionan la partición ya creada por este método. Las distribuciones Linux, en el proceso de instalación ejecutan un módulo que permite gestionar las particiones (algunas son más completas que otras).



- > Habrá que recoger algunos datos referentes al hardware del PC en el que se instala el sistema para solucionar posibles problemas en el proceso de instalación como son: la marca y el modelo de la tarjeta gráfica, la tarjeta de sonido y las tarjetas de red.; la marca de la pantalla y los modelos que soporta.

Además, habrá que comprobar que nuestra distribución es compatible y soporta el hardware del PC como la placa base y el microprocesador, para ello se puede consultar la página web oficial de la distribución.

- > Antes de iniciar el proceso de instalación es conveniente leer toda la documentación referente a dicho proceso en el manual de la distribución.
- > Si queremos instalar más de un S.O en un ordenador es recomendable instalar en último lugar LINUX (actualizará el menú de acceso a cualquier sistema; no así los Windows que no reconocen el S.O Linux).

1.10.2. Medios de instalación

Tenemos varios medios posibles para realizar la instalación de un sistema operativo actualmente gracias a la evolución que ha experimentado la informática.

Hoy en día hay tres medios de instalación principales:

- > **CD/DVD:** los CDs o DVDs lives contienen una imagen del sistema desde la cual arrancaremos el equipo y comenzará el proceso de instalación. Hoy en día son los menos usados puesto que muchos equipos ya no cuentan con lector de CDs.
- > **USB Botteable:** exactamente igual que el anterior, pero esta vez se realiza con un USB que ha sido modificado por parte de un software específico para almacenar el sistema operativo en él.
- > **El propio sistema:** muchos sistemas operativos como por ejemplo Windows nos dan la opción de reinstalar nuestro sistema.

Además, por ejemplo, Linux no tiene un instalador de cero, sino que al insertar un CD live o un USB, nos muestra una imagen del sistema que se puede usar, pero es su versión live donde estará la opción para la instalación permanente del sistema.



1.11.

Gestión de varios sistemas operativos en un ordenador

A la hora de instalar varios sistemas operativos, debemos tener en cuenta una serie de temas:

- > El primero que instalaremos siempre, será Windows, ya que Linux crea su propia partición a la hora de la instalación.
- > La BIOS debe soportar este recurso en los equipos. La mayoría de BIOS actuales soportan varios sistemas a la vez.
- > Debemos de contar con suficientes recursos en el equipo para almacenar ambos sistemas.
- > Si, como hemos indicado, instalamos primero Windows, a la hora de instalar Linux, habrá que modificar el gestor de arranque del primero para que reconozca el gestor de arranque del último y detecte el multiboot.

1.12.

Gestores de arranque

Para que el sistema pueda empezar a funcionar se usa un gestor de arranque, que es un software específico que decide o elige que código se va a ejecutar en el siguiente arranque.

En muchas ocasiones el gestor de arranque irá de forma implícita en el cargador de arranque del sistema.

Este es por ejemplo el caso de GRUB en los sistemas Linux.

También puede darse que haya un cargador de arranque sin gestor de arranque o un gestor de arranque independiente sin necesidad de que exista un cargador de arranque.



1.12.1 MBR (Master Boot Record)

Se utiliza principalmente en sistemas que poseen BIOS, es el primer sector con información de la unidad de almacenamiento del disco.

Solo permite tener cuatro particiones primarias. Podemos aumentar el número de particiones utilizando tres particiones primarias y una extendida.

La partición extendida no puede almacenar datos de forma directa, sino que está preparada para contener otras particiones llamadas lógicas; tantas como la unidad de almacenamiento pueda contener.

Una de las particiones primarias se suele identificar como activa, con el fin de albergar el gestor de arranque del sistema operativo.



Imagen 6. Esquema de partición de un disco duro con MBR.

1.12.2. GPT (GUID Partition Table)

Se utiliza en sistemas con UEFI, es una evolución de la anterior descripta, añade características como la redundancia o la posibilidad de crear hasta 128 particiones con una capacidad total de 8 ZB y no tiene la necesidad de distinguir entre particiones primarias o lógicas.



Imagen 7. Esquema de partición de un disco duro GPT.



Los sistemas UEFI son un camino intermedio entre la BIOS y el propio sistema operativo, lo que nos ofrece un amplio abanico de ventajas debido a su gran flexibilidad.

Por lo general la denominación BIOS hace referencia a sistemas BIOS y UEFI, asumiendo el nombre genérico.

Dentro de los sistemas UEFI, tenemos dos modos de arranque principales, los cuales son:

- > **Heredado o Legacy BIOS**: este es el modo necesario para que la compatibilidad con discos particionados con MBR siga funcionando.

Cuando usamos este método de arranque perdemos casi todas las ventajas que nos plantea UEFI y además el esquema de particiones del disco ya será sí o sí **MBR**.

- > **UEFI**: el recomendado por casi todos los sistemas, porque adopta las ventajas de UEFI plenamente. El esquema de particiones del disco se creará de manera automática como **GPT**.

Debemos tener en cuenta que el sistema que usemos ya sea Legacy o UEFI debe de coincidir con el esquema de particiones que queramos para nuestro sistema operativo.

Si realizamos la instalación de Windows, cualquier edición, y usamos el arranque UEFI, se crea de manera automática el siguiente esquema de particiones:

- > **Partición del sistema (System)**: partición del sistema EFI o ESP. Esta es la primera partición que se inicia al encontrarse en el sistema del disco duro de arranque.
- > **Partición reservada de Microsoft (MSR)**: esta partición es a su vez la que gestiona todas las demás particiones, haciéndose imposible que almacene datos de usuario.
- > **Partición de Windows (Windows)**: partición donde tendremos nuestro sistema operativo y los datos de usuario.
- > **Partición de recuperación (Recovery)**: esta es la partición donde se almacenan las herramientas de recuperación del sistema.



Imagen 8. Estructura de particiones de Windows 10



1.12.3. Gestor de arranque Windows

El último paso de una secuencia de arranque se basa en localizar y cargar el sistema operativo desde el gestor de arranque o boot Loader. Si hay varios sistemas operativos instalados, debemos elegir cuál de ellos queremos iniciar.

Dependiendo del sistema operativo, se utiliza un gestor diferente:

> En Windows: el arranque del sistema se realiza mediante Windows Boot Manager, que emplea dos elementos:

- » El fichero **bootmgr**: se encarga de iniciar el sistema operativo.
- » Base de datos BCD (Boot Configuration Data).

BCD nos permite arrancar sistemas operativos no gestionados con BIOS, además de ser un sistema cifrado. Para poder administrarlo es necesario poseer permisos de administrador y usar el comando '**bcdedit**', aplicaciones como EasyBCD o las propias herramientas del sistema como 'Configuración del sistema'.

También podemos ejecutar el comando '**msconfig**' en la barra de inicio y aparecerá una ventana con diferentes pestañas de configuración. Debemos dirigirnos a 'General – Arranque – Servicios – Inicio de Windows – Herramientas'.

Desde la pestaña 'Arranque' podemos gestionar el arranque de los sistemas operativos instalados en el equipo.

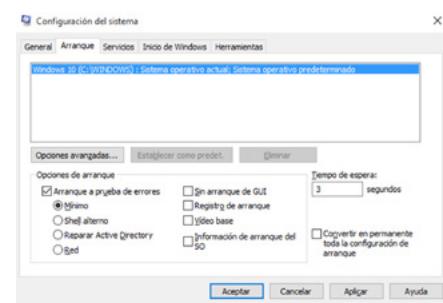


Imagen 9. Selección del sistema operativo mediante el comando 'msconfig.exe'.

1.12.4. Gestor de arranque Debian

En Debian se utiliza GRUB como gestor de arranque en su segunda versión. Este gestor multiarranque posee las siguientes características:

- > Permite arranque a través de red.
- > Soporta gran cantidad de sistemas de archivos como EXT4, NTFS, FAT, etc.
- > Nos permite acceder a los datos de cualquier dispositivo instalado, mientras esté habilitado y reconocido por la BIOS.

Podemos administrar el GRUB mediante una aplicación gráfica, Grub Customizer, que nos permite configurar las entradas del menú de arranque.

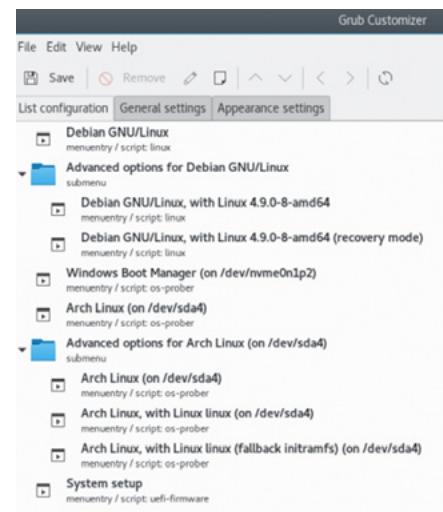


Imagen 10. GRUB Customizer.



1.13.

Instalación / desinstalación de aplicaciones

Para gestionar las aplicaciones Windows utiliza el apartado 'Aplicaciones y características', que podemos encontrar en 'Aplicaciones', dentro de 'Configuración'.

En Debian podemos utilizar la línea de comandos con el comando [apt](#), o si preferimos un entorno gráfico podemos utilizar Synaptics.

1.13.1. Aplicaciones y características de Windows

Si entramos en '[Aplicaciones y características](#)' tendremos un listado con todas las aplicaciones instaladas. Dentro de este listado habrá una serie de aplicaciones que son totalmente nativas de Windows y que por lo tanto no se pueden desinstalar.

Windows también tienen un conjunto de "herramientas" que son opcionales a las cuales llamamos 'Características'. Estas características pueden encontrarse activadas en algunos casos y en otros no. Se accede a las 'Características' siguiendo los siguientes pasos:

'Aplicaciones y Características' → 'Opciones de Configuración relacionadas' → 'Programas y características'

Nos volverán a aparecer todas las aplicaciones instaladas y a la izquierda veremos una opción que pone 'Activar o desactivar las características de Windows'. Si pulsamos dicha opción, nos saldrán todas nuestras características.

Para instalar características y distintos programas necesitaremos disponer de los privilegios de administrador del equipo.

Para desinstalar algunas aplicaciones también necesitaremos de estos privilegios.

1.13.2. Aplicaciones y características en Debian

En Debian, por defecto podemos utilizar el comando [apt-get](#), y también [aptitude](#), para poder instalar aplicaciones .deb.

En modo gráfico, existen distintos gestores de archivos, pero el más conocido es Synaptics.

En la interfaz básica de Synaptics podemos encontrar las siguientes opciones:

1. Índice temático de los programas de nuestro equipo.
2. Paquetes que pertenecen a esa categoría de programas.
3. Descripción del programa y paquetes necesarios para que este funcione.
4. Barra de opciones con un buscador para encontrar más fácilmente un programa o paquete específico.

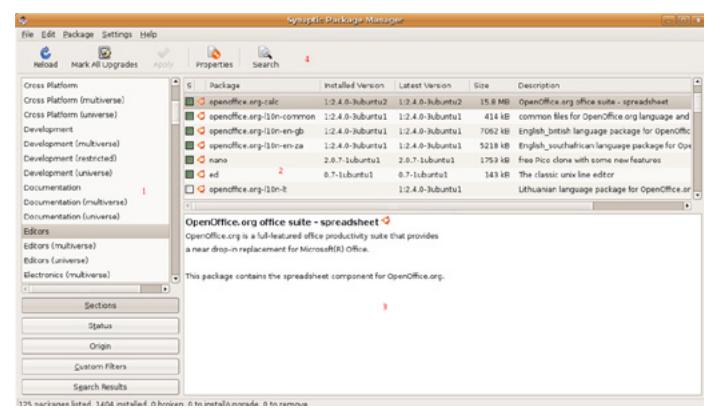


Imagen 11. Gestor de paquetes Synaptics.



1.14.

Actualización de sistemas operativos y aplicaciones

Cuando ya disponemos de un sistema operativo para su uso, lo típico es que tenga alguna carencia o fallo determinado.

Se desarrollan nuevos dispositivos, amenazas o aplicaciones de manera continua que no existían en el momento que se desarrolló el sistema operativo. Para dar solución a estos problemas, recurrimos a las actualizaciones del sistema, que permiten:

- > Optimizar el rendimiento.
- > Aumentar la compatibilidad con el software y hardware.
- > Mejorar la seguridad del sistema.
- > Solucionar errores de funcionamiento.

De forma general, las actualizaciones son gratuitas y podemos instalarlas de forma manual o automática. Debemos administrarlas con cuidado, ante la posibilidad de que surjan problemas como:

- > Actualizaciones con errores.
 - > Gran consumo de ancho de banda.
 - > Incompatibilidad con el hardware y software instalado.
- a. **En Windows:** La aplicación nativa de Windows que se encarga de administrar las actualizaciones del sistema se denomina Windows Update. El principio que sigue es el de actualizaciones silenciosas, de forma general, descargadas desde los servidores de Microsoft e instalándose sin la interacción del usuario.
 - b. **En Debian:** Podemos encargarnos de descargar e instalar las actualizaciones, aunque también podemos automatizar el proceso utilizando algunas herramientas como cron o cron-apt.

1.14.1. Windows

Microsoft contempla ciertas opciones a la hora de sus actualizaciones y las principales son sus llamados parches. Estos parches sirven tanto para solucionar problemas de seguridad (los más críticos), para controladores o para errores de alguna actualización del sistema.

Estos parches se pueden instalar directamente desde la configuración nativa de Windows que nos da esa facilidad o se pueden descargar en forma de archivos llamados Service Pack.

Por lo general, estas actualizaciones se llevarán a cabo desde la aplicación Windows Update.

El proceso de actualización de Windows Update es el siguiente:

1. Vamos a Inicio y a Configuración.
2. Ahora nos dirigimos a Actualización y seguridad.
3. Una vez aquí vemos que lo primero que nos aparece es la opción de Windows Update.

4. Como podemos ver, aquí se nos mostrarán las opciones de actualización disponibles y la opción Buscar Actualizaciones.



Imagen 12. Windows Update

Algunas actualizaciones del sistema nos pedirán reinicio, pero este se puede programar para no interrumpir nuestro trabajo.

Por último, en informática es recomendable no actualizar nunca a la última versión hasta que esta no esté lo suficientemente testeada y nos podamos asegurar de su correcto funcionamiento.



1.14.2. Linux

Al igual que pasaba con Windows, las actualizaciones de Linux ayudan a mejorar el sistema al mismo tiempo que corrigen ciertos errores, pero estas actualizaciones en realidad difieren con las del sistema de Microsoft en que en Linux actúan directamente sobre el kernel.

Es por esto por lo que, si el núcleo sufre una actualización, todas las aplicaciones y paquetes instalados deben de actualizarse también.

En este caso, lo más conveniente es que las versiones de los paquetes no estén desfasadas, pero tampoco en la última versión, sino en la versión estable (así se denominan en Linux), que es la versión que se sabe a ciencia cierta que funciona de manera óptima y sin problemas de compatibilidades.

Por lo tanto, podemos decir que la actualización del sistema de Linux es realmente una actualización de sus paquetes ya que estos en conjunto conforman el sistema operativo.

Para poder **actualizar paquetes** primero tendríamos que actualizar la lista de estos, y esto se realiza mediante la opción:

apt-get update

Lógicamente, como todo en Linux, para poder ejecutar este comando es necesario que tengamos los permisos de superusuario o root.

Hay una opción destacada en Linux que es la de **actualizar los paquetes disponibles a la última versión** siempre que sea posible una vez que la lista esté actualizada, que es:

apt-get upgrade

Por último, apt también cuenta con una opción que además de actualizar paquetes hace un intento de corrección de errores para poder actualizar todos los paquetes, pero hay conflictos que no puede resolver, esta es la opción:

apt-get dist-upgrade

En resumen, las opciones de **apt-get** son:

Opciones apt	
Opción	Descripción
update	Actualiza el estado de paquetes disponibles.
upgrade	Actualiza el sistema con actualizaciones disponibles.
dist-upgrade	Actualiza todos los paquetes del sistema, instalando o desinstalados los paquetes que sean necesarios para resolver las dependencias que puedan generar actualizaciones de otros paquetes.



1.15.

Registros (logs) del sistema

1.15.1. Windows

Los registros del sistema en Windows nos permiten tener una visión completa de la configuración del sistema.

Aquí se registran todas las acciones que se llevan a cabo en el sistema, desde la instalación de un controlador hasta la creación de un nuevo usuario.

Los datos de cada usuario dentro de este registro se almacenan en archivos binarios y el archivo de cada usuario tiene el siguiente nombre: %USERPROFILE%\NTUSER.DAT.

En Windows las variables del sistema se representan entre los símbolos '%'.

En Windows las variables del sistema se representan entre los símbolos '%'.

Si lo que queremos es acceder al registro del sistema de Windows, deberemos de hacer lo siguiente:

1. Nos vamos a Inicio y buscamos regedit.
2. Para acceder nos pedirá permisos de administrador
3. El registro tiene la siguiente estructura:



Imagen 13. Regedit

Cada una de las opciones que tenemos nos sirve para acceder a distintos módulos del registro dependiendo de lo que necesitemos realizar.

1.15.2. Linux

En Linux de manera general tenemos los registros o logs del sistema en la siguiente ruta:

/var/log/

Todos los archivos que se encuentran aquí son archivos terminados en .log como podemos ver a continuación:

alternatives.log	btmp	gdm3	speech-dispatcher
alternatives.log.1	bttmp.1	installer	syslog
apt	daemon.log	journal	syslog.1
auth.log	daemon.log.1	kern.log	unattended-upgrades
auth.log.1	debug	kern.log.1	user.log
boot.log	debug.1	lastlog	user.log.1
boot.log.1	dpkg.log	messages	wtmp
boot.log.2	dpkg.log.1	messages.1	
boot.log.3	faillog	private	
boot.log.4	fontconfig.log	runit	

Imagen 14. /var/log

Aquí se recogen los principales registros del sistema, y, además, hay otros directorios donde se encuentran logs propios de alguna aplicación en concreto.

Por último, cabe destacar que hay muchas aplicaciones que son de tanto peso en el sistema que sus registros o logs se encuentran alojados en la propia ruta de configuración del paquete o aplicación en cuestión.



1.16.

Actualización y mantenimiento de controladores de dispositivos

En los sistemas operativos necesitamos que ciertos dispositivos que no han venido de manera predefinida con el equipo funcionen, y para eso existe el controlador o driver.

Este software se encarga de gestionar que haya una comunicación y una compatibilidad entre el dispositivo y el propio sistema.

Todos los controladores interactúan de manera directa con el núcleo del sistema y pueden ser de dos tipos:

- > Orientados a caracteres.
- > Orientados a bloques.

Estos dos tipos tienen la principal diferencia en que los orientados a caracteres se basan en el intercambio de información carácter a carácter y los orientados a bloques intercambian información en bloques, como indican sus nombres respectivamente.

Para que un controlador se pueda instalar en un sistema operativo este debe de ir firmado, ya que esto nos ayuda a garantizar lo siguiente:

- > **Un sistema más seguro.** Si el editor no es de confianza o el controlador no está firmado, no se permitirá su instalación. La confianza de los editores la decidirá el administrador del sistema.
- > **Una mayor fiabilidad sobre el funcionamiento del dispositivo.** Si la organización en cuestión no ha aprobado ese dispositivo, no irá firmado porque podría no funcionar bien.
- > **Funcionamiento automático.** Una vez que el controlador instalado que esté firmado reconozca el dispositivo, ya no hará falta realizar ninguna acción en las siguientes veces de su uso.

1.16.1. Windows

Cada vez que conectamos un dispositivo a Windows, este lo detecta de manera inmediata debido al sistema que lleva predefinido como Plug-and-play, que consiste en que cada vez que se conecta un dispositivo se puede trabajar de manera automática con él.

Cada vez que un sistema operativo es instalado desde cero conviene que desde la web oficial del fabricante se comprueben todas y cada una de las actualizaciones disponibles del fabricante del hardware para que se pueda trabajar con normalidad, porque puede ser que, si es un hardware separado, Windows no lleve por defecto todos y cada uno de los controladores preinstalados.

Para ver estas actualizaciones hacemos lo siguiente:



1. En Inicio, buscamos 'Administrador de dispositivos' como administrador del equipo.
2. Ahora podemos ver una lista de todos los dispositivos conectados al equipo.

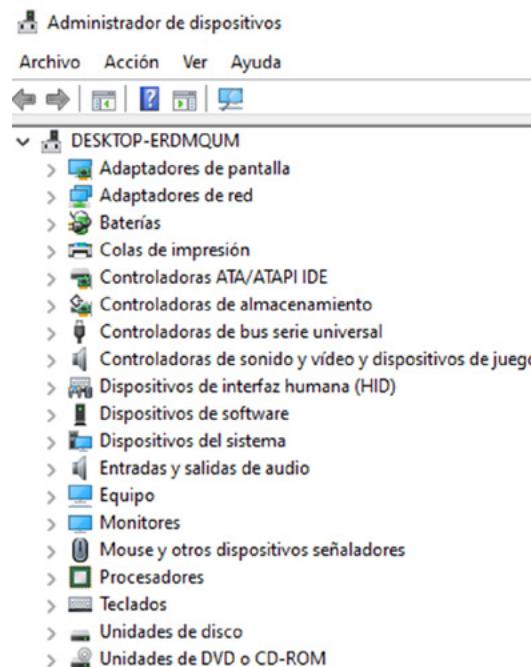


Imagen 15. Administrador de dispositivos

3. Una vez aquí, si pinchamos en cualquiera de ellos, por ejemplo, en el adaptador de red nos saldrá una ventana de sus propiedades.
4. Nos dirigimos a la pestaña 'Controlador'.
5. Ahora elegimos la opción 'Actualizar controlador'.

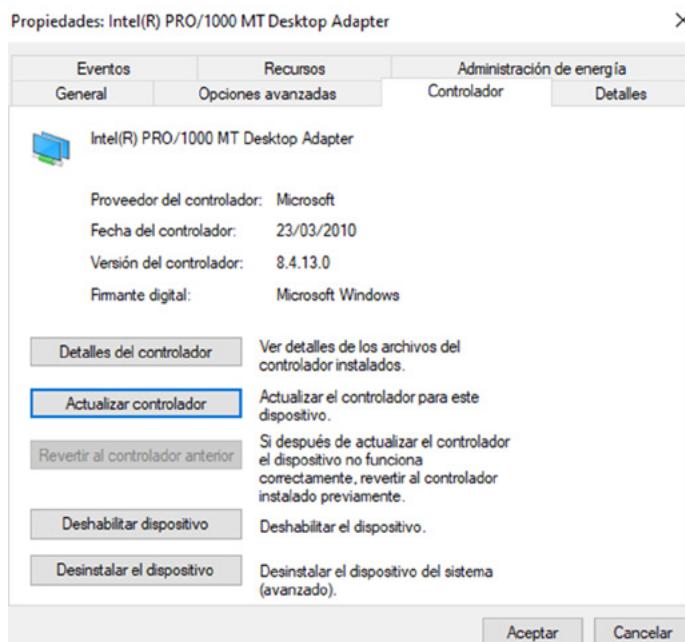


Imagen 15. Opciones de dispositivos



6. En esta opción nos dirá si queremos buscar un controlador en línea o en el equipo, nuestra opción será en línea porque no disponemos de ningún driver específico en el equipo para este adaptador.
7. Si hay alguna actualización pendiente actualizará este controlador, pero si no, nos mostrará el siguiente mensaje:

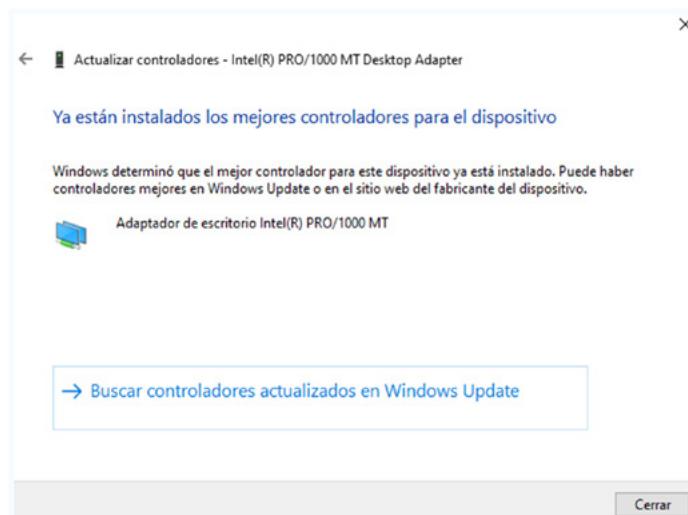


Imagen 16. Mensaje de búsqueda de controladores

1.16.2. Linux

Muchos de los sistemas Linux nos permiten gestionar desde aplicaciones gráficas los controladores del sistema operativo.

Por temas legales, al ser software libre, si un driver es propietario no puede ir integrado directamente en el sistema operativo, lo que hace que en muchos casos deban descargarse a posteriori.

Además, desde Synaptics y desde el fichero /etc/apt/sources.list.d, se pueden gestionar también los repositorios de software, también válidos para controladores.



 www.universae.com

