Unidad 3

Instalación de software de utilidad y propósito general para un sistema informático

Fundamentos de Hardware

Índice

Introducción

Los sistemas operativos van estrictamente ligados al hardware del sistema informático y por eso es necesario que se comprenda como se puede organizar esta relación.

Para que se haga una correcta gestión de los recursos del sistema y como organizar sus aplicaciones y procesos para el mínimo gasto de recursos manteniendo la eficiencia los sistemas operativos nos proveen de distintas herramientas y utilidades que varían dependiendo de su plataforma y su tipo.

En esta unidad veremos cómo funcionan los distintos tipos de sistemas y que se nos plantea en relación con la eficiencia en el trabajo y la gestión de este, es el llamado software de utilidad y de propósito general.

Al finalizar esta unidad

- Conoceremos los distintos tipos de malware que pueden afectar a un sistema informático y cuáles son sus características.
- Estableceremos el funcionamiento de los antivirus.
- Sabremos analizar los distintos entornos operativos.
- Manejaremos utilidades de administración, mantenimiento y reparación de un sistema informático.
- Seremos capaces de identificar los distintos tipos de aplicaciones existentes en el mercado.
- Valoraremos el funcionamiento interno de un compresor de archivos.

3.1. Entornos operativos

Podemos clasificar el software y las aplicaciones según su uso:

- **Software de sistema:** referencia a cualquier programa que este en contacto con el hardware y actúa de intermediario entre el usuario y el hardware.

  El más conocido es el sistema operativo, aunque la mayoría de los dispositivos disponen de un software que actúa de interfaz entre el hardware y el usuario.

- **Conjunto de software necesario en el diseño, implementación y desarrollo de software de sistema o aplicación:** En este tipo se incluyen editores, compiladores, depuradores de código y entornos de desarrollo integrados (IDE).

- **Software de aplicación:** tipo de software orientado a realizar tareas concretas y uso cotidiano por parte del usuario. Como, por ejemplo: aplicaciones ofimáticas, juegos, etc.

El sistema operativo

Podemos definir sistema operativo como un conjunto de módulos que interactúan entre sí para realizar gran cantidad de tareas. Se encuentra en continua evolución, ya que se actualiza para corregir problemas de seguridad y poder adaptarse a la evolución de los recursos hardware y software. Algunas de las características más importantes de los sistemas operativos son las siguientes:

- Actúa de interfaz entre el usuario y el hardware.
- Gestiona los recursos de hardware y software del sistema de forma imperceptible para el usuario.
- Se encarga de facilitar la interacción del usuario con la máquina, aprovechando todos los recursos del equipo.

Existe una gran variedad de sistemas operativos. Cada uno tiene características especiales y atiende a necesidades diferentes.

Algunos ejemplos de sistemas operativos son: Windows de Microsoft, macOS de Apple o Android de Google.

3.1.1. Firmware

Firmware hace referencia al software que contiene un hardware específico. Confundido muchas veces con el sistema operativo, el propósito general del firmware es la realización de ciertas tareas específicas al mismo tiempo que la optimización completa del hardware. El sistema operativo en cambio es de propósito general y muchísimo más complejo.

Por ejemplo, un aparato muy de moda en estos días, como puede ser ChromeCast o Fire Stick TV, para la reproducción de contenido creados por Google y Amazon respectivamente, llevan instalado un firmware que es la capa visible.

Además, este tipo de software puede ser actualizado muchas veces para que no quede obsoleto o inseguro.

3.1.2. Estructura de un sistema operativo móvil

Cada sistema operativo difiere de otro dependiendo de que compañía lo desarrolle y dentro de los de la misma compañía también difieren por sus tipos, pero todos conservan una estructura en gran medida similar, ordenada por capas.

Cada una de estas capas suele realizar una función concreta para el correcto funcionamiento del sistema informático y en constante comunicación con las capas con las que limita. En general hay cuatro capas bien definidas:

- **Interfaz de usuario y aplicaciones nativas.** En todos los dispositivos móviles que usamos en el día a día se pueden encontrar una serie de programas o aplicaciones que vienen de serie instaladas en nuestro sistema y que no se pueden/deben eliminar, porque son las llamadas *aplicaciones nativas*.

  Estas aplicaciones normalmente funcionan de intermediario con otras aplicaciones mediante una serie de *API, Application System Interface*.

- **Interfaz de aplicaciones.** Esta interfaz permite que las *apps* instaladas tengan un modo de contacto con el usuario y que haya un control de las notificaciones o los servicios de telefonía del dispositivo.

- **Librerías/middleware.** Estas librerías permiten que se puedan controlar ciertas funciones del sistema como por ejemplo librerías gráficas o servicios de bases de datos para el correcto desarrollo y funcionamiento de las aplicaciones.

- **Núcleo o Kernel.** Al igual que en un sistema operativo de ordenador, los SO de los dispositivos móviles tienen un núcleo que controla todos los procesos sobre el sistema. En los dispositivos móviles este núcleo suele estar basado en Unix, por ejemplo, en Android se basa en el núcleo de Linux y en iOS en el de MAC OS X.

Estructura de un sistema operativo movil
Interfaz de usuario y aplicaciones nativas
Interfaz de aplicaciones
Librerías de middleware
Kernel

3.2. Tipos de aplicaciones

El termino *software* se usa tanto para el sistema operativo como para las aplicaciones que se desarrollan para la interacción con el sistema. Dentro de este último grupo tenemos las aplicaciones que nos ayudan a realizar funciones concretas y las aplicaciones que específicamente nos ayudan a desarrollar nuevas apps.

Hace ya un tiempo las aplicaciones eran programas que estaban compilados y se ejecutaban en un equipo en concreto, pero con el paso del tiempo esto ha evolucionado surgiendo además los siguientes tipos:

- **Aplicaciones web.** Son las aplicaciones que están alojadas en un servidor en la web y se ejecutan de manera directa en el navegador del cliente.

  Una parte del código se ejecuta directamente en el servidor, como puede ser el lenguaje PHP y otra se ejecuta en el navegador cliente directamente, como el lenguaje *JavaScript*. La mayor ventaja de estas aplicaciones es que son multiplataforma en su gran mayoría.

- **Apps móviles.** Son las aplicaciones que corren directamente sobre el sistema del dispositivo móvil en cuestión. Dentro de estas hay otros dos tipos, las nativas o las híbridas.

  Las primeras ya se han mencionado anteriormente y las segundas usan un *WebView* que hace que se *renderice* su visionado desde una página web, conectando directamente, pero de manera transparente con su URL.

  Estos *WebViews* son bastante potentes al nivel de poder gestionar la navegación o los datos del usuario como son las *cookies*.

  Hay algunos *frameworks* en la actualidad que permiten que se desarrolle de manera muy sencilla estas aplicaciones basadas en cualquier tecnología web de las vistas en otros módulos.

- **Widgets.** Son las aplicaciones que tienen os teléfonos móviles, por ejemplo, y que aplican distintas funciones complementarias al sistema. Se encuentran también en los sistemas operativos como Windows o Mac OS.

- **Plugins.** Como los *widgets* en algunos aspectos, pero estos corren sobre otros programas o aplicaciones directamente con la intención de aumentar su funcionalidad. También reciben el nombre de *add-on*.

3.2.1. Software de propósito general

Como indica su propio nombre, son aplicaciones que tienen unas funciones más generales y de uso muy común, estas se contraponen a otro tipo de aplicaciones que solo pueden desarrollar una función para un puesto o sitio específico.

Las aplicaciones más comunes de este tipo de *software* son:

- **Aplicaciones de oficina u ofimática:**
  - Hojas de cálculo
  - Editores de presentaciones
  - Visores de fotos
  - Gestores de bases de datos
  - Procesadores de texto
  - Visores de documentos
- **Productividad y negocios**
  - Agendas de contactos
  - Calculadoras
  - Utilidades de contabilidad.
- **Juegos**
- **Navegadores**
- **Aplicaciones P2P**
- **Educación**
  - Enciclopedias
  - Diccionarios
- **Multimedia**
  - Reproductores
  - Creación de video
- **Antimalware**
- **Imagen y diseño**
- **Programación**

3.2.2. Instalación y prueba de aplicaciones

Aunque hoy en día muchas aplicaciones son aplicaciones web que no necesitan de una instalación, sigue habiendo un cierto grupo que no poseen aplicación web (la mayoría poseen ambas).

Para las aplicaciones que deben de ser instaladas sí o sí, los fabricantes de estas y sus desarrolladores han creado unos ciertos repositorios o *stores* donde almacenar las últimas versiones de estas con el fin de evitar así el *malware*.

Para poder descargar estas aplicaciones habrá que cumplir con unos ciertos requisitos como pagar una tasa o cumplir con unas características concretas, estos requisitos los marca el fabricante o desarrollador.

Las dos mayores tiendas de aplicaciones móviles son:

- **AppStore.** Es la tienda de Apple en la cual se pueden descargar aplicaciones para todos los dispositivos Apple y son aplicaciones en su mayoría específicas.

  Para poder publicar aplicaciones en esta tienda es necesario ser un desarrollador únicamente de Apple y además las apps pasarán un exhaustivo control de calidad.

- **Google Play.** Se trata de la tienda de aplicaciones creada por Google para dispositivos Android.

Hay tres formas de instalar aplicaciones y las vamos a ver a continuación, que es mediante copia directa, mediante instalador o mediante un gestor de paquetes.

Copia directa

Es una de las opciones más usadas para instalar aplicaciones en algunos sistemas operativos como Mac OS porque los programas usan librerías del sistema operativo. Esto repercute en que su instalación consiste en llevar simplemente el programa hasta la carpeta '*Aplicaciones*' y el programa está listo para su uso.

Esta instalación tiene la ventaja que el desarrollador del sistema sabrá que la instalación de estas aplicaciones no va a alterar el sistema. Además, no suelen necesitar de privilegios de administración para su instalación.

Instalación mediante instalador

Es la manera más común de instalar un programa en Windows, y después en el *Panel de control,* nos dejará eliminarlo.

EL mayor problema que presentaba esta opción es que el propio sistema Windows almacenaba toda la información relativa a instalaciones y desinstalaciones en el llamado *Registro del sistema*, que es una gran base de datos con los registros de todo el sistema (de ahí su nombre). Esto hacía que cada cierto tiempo fuera necesaria una limpieza del registro e incluso del propio sistema.

Con el paso de los años cada vez los registros se llenan menos, pero sigue siendo un dolor de cabeza para los usuarios que usen este sistema a gran nivel.

Instalación mediante un gestor de paquetes

Esta opción es la más usada en los sistemas Linux, donde las aplicaciones se almacenan en paquetes que el sistema instala y administra cuando lo solicitemos. Estos paquetes se almacenan en *repositorios*, algunos de los cuales son comunes del sistema y otros más específicos para ciertas aplicaciones de terceros. Dos sistemas de paquetes ampliamente usados son:

- **APT (Advanced Packcaging Tool):** es la herramienta usada por el proyecto Debian y Ubuntu principalmente.
- **RPM o Red HatPackcge Manager:** creada por la distribución Red Hat es característica por ejemplo además en Madndriva u openSUSE entre otros.

3.2.3. Comparación de aplicaciones. Evaluación y rendimiento

Cuando un desarrollador se encuentra desarrollando un *software* en concreto necesitan evaluar este durante el desarrollo para poder estar seguros de que funcionará según expectativas.

En paralelo al desarrollo habrá un grupo de personas que se dediquen al testeo de los programas para verificar que el *software* tendrá las funciones solicitadas para el usuario final.

El objetivo del desarrollo de *software* no es solo que esté libre de fallos y errores, sino que además cumpla con una serie de características de calidad mínimas para que el rendimiento sea el óptimo en la medida de lo posible.

Para poder asegurarse de esto también existen pruebas de rendimiento que nos dirán que velocidad de procesamiento tiene el *software* y si es la suficiente para que se pueda trabajar eficientemente con él.

3.3. Compresión y descompresión de archivos

El objetivo de la compresión es que el tamaño de un fichero se reduzca de manera considerable para facilitar su gestión y almacenamiento. El *ratio o razón de compresión, RC,* nos va a indicar cuánto están comprimidos los datos.

Por ejemplo, si unos datos tienen un ratio de compresión de 5:1, se quiere decir que cada 5 bits del fichero de origen ocupan un solo bit en el fichero comprimido.

Tenemos dos tipos principales de algoritmos de compresión:

- **Algoritmos de comprensión con pérdida:** cada vez que descomprimamos los datos no se obtendrán todos los datos que teníamos en el origen porque al usar el algoritmo algunos se pierden. Esto pasa porque este algoritmo comprime eliminando información ue considera innecesaria para la gestión del fichero.
- **Algoritmo de compresión sin pérdida:** en este tipo de algoritmo de compresión, cuando descomprimamos los datos tendremos todo igual que en el origen, El algoritmo más conocido de este tipo es el *Huffman* que se usa en el formato ZIP.

3.4. Utilidades para el mantenimiento de y reparación de los sistemas informáticos

3.4.1. Recuperación del arranque (cargador)

Hay multitud de ocasiones en las que nos encontraremos con que un equipo no arranca y tendremos la sospecha de que el problema viene derivado del cargador de arranque, independientemente del que sea.

Para solucionar esto proponemos usar *Rescatux*, que se encuentra basado en Debian y nos permite recupera el cargador de arranque.

Además, esta herramienta también da otras muchas opciones para la administración del sistema.

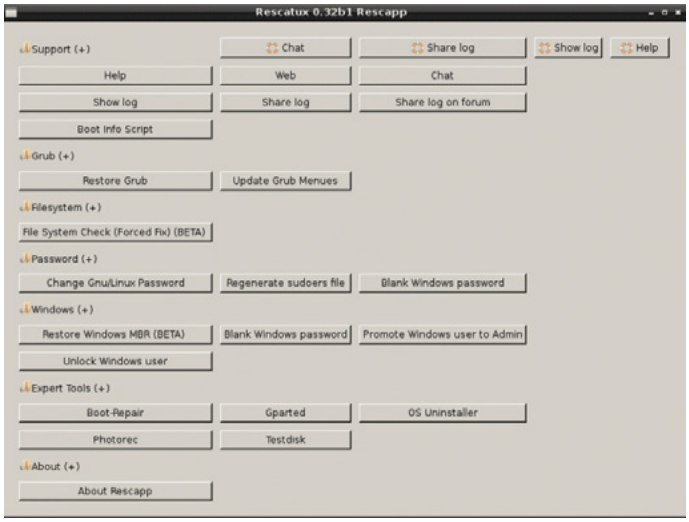


Imagen 1. Panel de control de Rescatux.

3.4.2. Utilidades para la recuperación de ficheros

Cuando borramos un fichero, el sistema operativo no lo borra físicamente del disco duro, simplemente se recupera el espacio que ocupaba para su uso en el sistema, por eso si se usa por ejemplo la técnica *file carving* o alguna utilidad forense de archivos se podría recuperar el contenido, siempre que no haya sido dañado.

Una de las utilidades más usadas en Windows para la recuperación de archivos es *Recuva*. Una de sus versiones es portable, que quiere decir que su instalación el sistema no es obligatorio.

Funciona de manera sencilla e intuitiva, primero realiza un escaneo de la unidad y clasificar los ficheros por colores debido a como se encuentran.

El código de colores que sigue es el siguiente:

- **Rojo.** Irrecuperable
- **Verde.** Recuperable
- **Amarillo.** Se puede recuperar de manera parcial porque el archivo está incompleto.

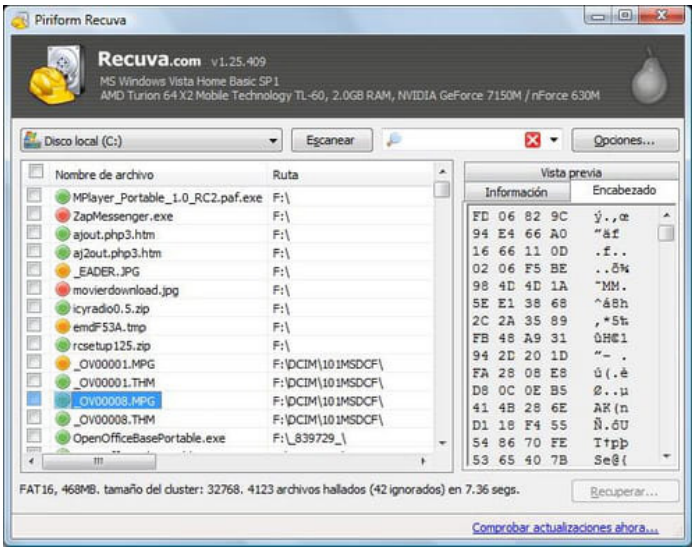


Imagen 2. Recuva Portable.

3.4.3. Multiherramienta para Mac OS-X Onyx

Para realizar el mantenimiento de un equipo con Mac OS-X instalado, la herramienta más usada es *Onyx*. Esta es una multiherramienta que comprueba que la estructura del sistema de ficheros esté correcta en primera instancia y luego nos permite administrar dicho sistema.

Además, con ella también se pueden reorganizar los índices del sistema y muchas otras tareas de administración del propio equipo. Su instalación es por **Copia Directa**.

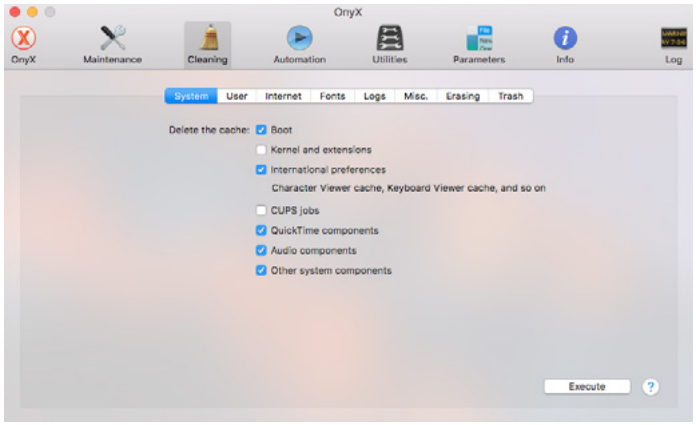


Imagen 3. Onyx.

3.5. Malware y antivirus

Hoy en día el termino de *malware* no solo hace referencia a los virus informáticos como pasaba antes, sino que se ha ampliado mucho el espectro de posibilidades de un programa malicioso.

Luchas contra estos es una de las principales medidas que debe tomar un técnico microinformático sobre todo en el que caso de que los usuarios que usen los equipos finales no tengan mucha idea acerca de informática.

3.5.1. Malware

A continuación, se van a describir los tipos más conocidos de malware y cómo funciona cada uno además del daño que pueden causar con la intención de poder identificarlos y detenerlos de manera más sencilla.

NOTA

Debemos de tener en cuenta que nos encontraremos algún malware que forme parte de dos tipos a la vez.

Los virus

Los virus son un tipo de programa malicioso que va infectando a distintos archivos del sistema, de ahí su nombre, con el fin de poder pasar después a otros sistemas. Muchas veces el virus ataca directamente al cargador de arranque del equipo para asegurarse de que siempre va a estar cargado en memoria y así no es eliminado al borrar los archivos infectados.

Su objetivo final es terminar con la vida del equipo hasta que ya no funcione.

Los dos virus informáticos que más daño han causado a lo largo de la historia son el conocido Iloveyou, que infectaba los equipos en el año 2000 a través de un archivo .txt, mandado por correo electrónico y el Melissa que solo un año antes atacó a grandes compañías como puede ser Intel.

NOTA

Casi todos los virus que han infectado equipos a lo largo de la historia se han introducido en los equipos por correo electrónico o por un sistema de mensajería, por lo que es recomendable que si desconocemos el remitente de dicho mensaje o el contenido nos resulta extraño, no se abra ese mensaje y se escale a los encargados de ciberseguridad.

Los troyanos

La función de un troyano no es la de destruir el equipo infectado, pero su efecto puede ser peor, porque su intención es la de robar datos de carácter valioso para el usuario como pueden ser contraseñas o incluso simplemente obtener el control de la máquina donde se hospede el troyano.

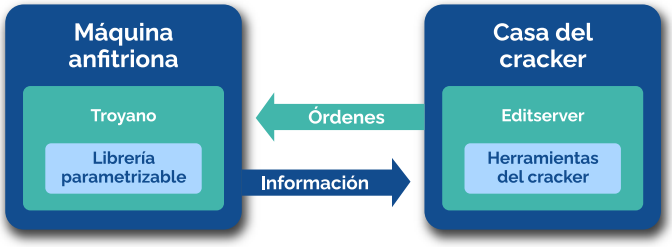


Imagen 4. Partes de un troyano.

Los troyanos muchas veces tienen la opción de parametrizarse, con esto nos referimos a que desde el origen del que se envían, dentro de su estructura contienen ciertas librerías parametrizables que le pueden indicar que opción en concreto deben de llevar a cabo.

Además, desde la ubicación de origen el cracker es capaz de orientar y manejar el troyano a su libre antojo.

NOTA

Muchas veces los troyanos se crean con el objetivo de tomar el control de la máquina para después poder instalar otro tipo distinto de malware

Las botnets

En una botnet nos encontramos con diversos robots informáticos llamados bots que son autónomos y siguen una serie de órdenes de acuerdo con el administrador de la botnet.

Hoy en día los bots reciben muchos usos maliciosos como pueden ser:

- **Realizar ataques DDoS para la Denegación del Servicio,** ya que como se encuentran de manera dispersa, los ataques hacia un servidor o red son difíciles de dictaminar haciendo que al final el servicio atacado se caiga.
- **Enviar spam** es posiblemente el mayor uso para estos bots, porque se pueden enviar multitud de mensajes desde diferentes ubicaciones y de nuevo se hace muy difícil filtrar desde donde vienen los ataques

Si quiero saber si mi equipo ha sido infectado y es un bot, debo tener en cuenta lo siguiente:

- El equipo no se deja actualizar.
- El internet del equipo va lento, pero si se hace un test de velocidad todo indica que va bien.
- Si el equipo se ralentiza al realizar cualquier gestión.
- Si el equipo se encuentra trabajando cuando no debería.
- Si se abre el administrador de tareas y hay ejecuciones que no conocemos.
- Si hay picos de internet en el monitoreo de la red.

Si detectamos cualquiera de estas características en un equipo deberíamos de usar algún servicio antibotnet que nos monitoree la conexión a internet. También se pueden instalar en los navegadores plugins que nos avisan en caso de que el equipo esté infectado.

NOTA

Es importante saber que un bot no es un tipo de malware, sino que se puede usar para hacer daño a otro equipo. La función real de los bots es la automatización de ciertas tareas beneficiosas.

Los keyloggers

Los keyloggers son una serie de dispositivos que registran las pulsaciones que se realizan en un teclado, aunque suelen ser un dispositivo físico, también hay tipos de software que realizan esta función.

Imagen 5. Keylogger Hardware. Fuente: commons.wikimedia.org

Con la implementación de un keylogger en un equipo lo que pretende el atacante es el robo de usuarios o contraseñas, por ejemplo.

Cuando tenemos un keylogger hardware instalado, a veces las pulsaciones se registran en el mismo dispositivo, pero los más modernos lo mandan a otro receptor inalámbrico, para que en caso de ser descubierto no se puedan borrar sus datos.

En el caso de los keylogger implementados por software, estos son de fácil descubrimiento por algún antivirus convencional, por eso se usan mucho más los físicos.

El spyware

Un spyware es un tipo de malware que se basa en la recopilación de información de interés del usuario de un equipo o incluso del mismo equipo.

Además del robo de información que supone para fines que el usuario desconoce, también se sufre en la conexión a internet porque el envío de esta información ocupa ancho de banda.

Si en un equipo vemos que ocurre algo de lo siguiente, puede ser señal de que tenemos instalado un spyware:

- Cambia la página de inicio.
- Cambia el motor de búsqueda.
- Nos salen multitud de anuncios en el navegador.
- Aparecen botones nuevos en la barra de herramientas.
- Hay procesos sospechosos y desconocidos en el administrador de tareas, y aunque se eliminen, vuelven a aparecer.
- Se desinstala un programa que creemos malicioso y "solo", se vuelve a instalar.

NOTA

Para luchar contra un sypware se usan los antispywares, pero no están muy recomendados debido a que solo detecta elimina el programa, pero detectarlo es tarea nuestra.

NOTA

Los spywares van asociados normalmente de manera muy estrecha a los troyanos, pues estos últimos muchas veces se introducen para poder instalar el spyware.

Esto quiere decir que, si detectamos un troyano, revisemos el equipo en busca de un spyware.

El adware

Se puede llamar también advertising software y se basa en la visualización de anuncios publicitarios por parte del usuario sin su solicitud.

Este tipo de software se considera un malware porque puede causar cierta molestia al usuario, pero es verdad que hoy en día muchas aplicaciones solicitan que se acepte el recibir una serie de anuncios a cambio de su uso, por lo que es de los menos dañinos por no decir que su daño es prácticamente nulo.

Las cookies

Las cookies de los navegadores de internet son archivos que contienen información sobre el usuario y que han ido evolucionando a lo largo de los años.

Es normal hoy en día aceptar las cookies de un sitio web para que se recopile información y después enseñarnos un entorno algo más personalizado. El problema real del uso de estas como malware es cuando esa información se usa para fines lucrativos como vender nuestros datos a otras empresas o crearnos perfiles falsos.

NOTA

En un principio las cookies fueron creadas ya que desde un navegador web no se puede acceder a la información local del dispositivo y este era el mejor medio para almacenar información web del cliente.

Las backdoors

Las backdoors o puertas traseras son agujeros de seguridad en el sistema que sirven para que un atacante entre sin la necesidad de una identificación previa y sin la detección del acceso por parte del sistema.

Estas puertas pueden crearse por vulnerabilidades o por algún otro tipo de malware, generalmente un troyano.

El ransomware

El ransomware es posiblemente el más dañino de todos los tipos de malware. Se usa para cifrar todo el sistema y quitarles el acceso a los usuarios ya que no poseen la correspondiente clave.

Este cifrado se suele llevar a cabo para poder pedir a cambio una recompensa económica que suele ser de muy alto precio.

Una solución para este tipo de malware es llevar un sistema de backups al día y poder reestablecer el sistema de cero.

NOTA

En el año 2019, concretamente en noviembre, Everis y la Cadena Ser sufrieron un ataque con un ransomware llamado Ryuk y sus consecuencias fueron devastadoras.

Los web bug, tracking pixel, tracking bug, pixel tag o web beacons

Los cinco realizan prácticamente la misma función, que es la de monitorizar la actividad de red de un usuario, por eso son considerados malware.

Son fáciles de detectar ya que suelen descargar imágenes de un tamaño muy pequeño, pero que están reflejadas.

Los exploit

Los exploit son un tipo de malware que se basa en explotar los fallos o vulnerabilidades de seguridad de un sistema.

Estos exploit también se usan en muchos campos de la ciberseguridad para detectar fallas y poder corregirlas.

NOTA

Para poder protegernos correctamente de los exploit, una buena práctica es tener instaladas todas las actualizaciones y parches de seguridad disponibles para el sistema.

Los rootkits

Son un tipo de herramienta que se usa para obtener el control del sistema con la intención de encontrar ciertas vulnerabilidades o crear exploits con los que acceder al sistema de manera completa.

Leapfrogs o ranas

Como su propio nombre indica es un tipo de malware que salta de un dispositivo a otro replicándose en cada uno de ellos. Para poder pasar de un equipo a otro se basan en el descubrimiento de contraseñas y los mensajes de correo electrónico.

Bulos, jokes o hoaxes

Este tipo de programa malicioso suele tener la intención de engañar al usuario que usa el equipo para que realice alguna acción sobre el equipo, generalmente instalar algún tipo de software con la intención de recopilar información o introducirse en el equipo.

Basura, escoria o scumware

El scumware es un tipo de malware que no permite ser desinstalado de manera normal, ya que se reinstala solo otra vez. Hay que destacar que su intención no es tan dañina ya que suele afectar solo en temas de publicidad vía navegador web.

NOTA

Muchos de los softwares libres que se instalan de páginas no fiables suelen llevar este tipo de malware con ellos.

El spam

Hasta la persona con menos conocimiento informático conoce de oídas el término spam. Este tipo de malware es muy común en el correo electrónico.

Su principal función es la de lanzar publicidad innecesaria y por lo general no deseada por el usuario con el fin de obtener alguna ganancia comercial.

Hoy en día casi todos los servicios de correo electrónico tienen implementado un servicio automático de antispam que hace que estos correos no lleguen o se desvíen a otras bandejas más insignificantes para el usuario.

Imagen 6. Carpeta de Spam que tienen por defecto algunos clientes de correo electrónico.

3.5.2. Los antivirus o antimalware

Para los usuarios que no tienen conocimientos suficientes de informática como para detectar un malware hasta que el daño ya está hecho. Por esto es importante que en mayor o menor medida se tenga instalado un antimalware o antivirus medianamente eficiente en los equipos.

Hay tres cosas que cualquier antivirus debe de cumplir para que lo dejemos instalado en un equipo:

- **Que consuma la mínima memoria posible.** Si el antivirus consume una elevada carga de memoria nos ralentizará el equipo, por lo que su efecto será contraproducente.
- **Que consuma el mínimo de CPU.** Nos sucede lo mismo que con la memoria.
- **Que se actualice de manera frecuente.** Con esto no nos referimos a una subida de versión, que también, sino a actualizaciones en sus bases de datos frente a posibles ataques o infecciones.

NOTA

Dependiendo del nivel de seguridad que necesitemos elegiremos un tipo u otro de antivirus.

Lógicamente, no necesitará un antivirus igual de potente un equipo doméstico que una organización entera.

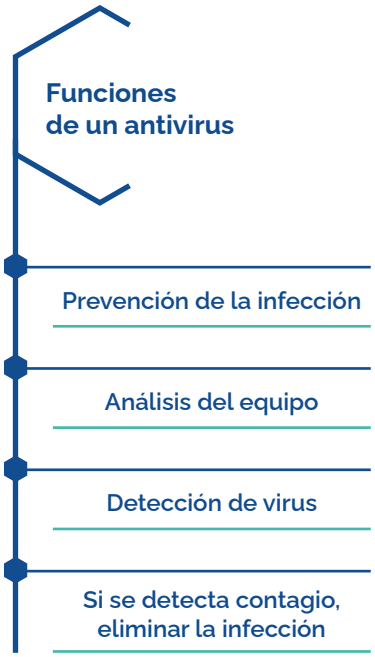


Imagen 7. Funciones de un antivirus.

3.5.3. Funcionamiento de un antivirus

Los antivirus modernos pueden funcionar generalmente de dos formas:

- **De forma pasiva:** la forma más clásica de actuación de un antivirus. Llamada también técnica de scanning se basa en que el antivirus tiene una base de datos donde va recopilando información acerca de los distintos tipos de malware y sus maneras de actuar para poder detectarlos a tiempo. Esta base de datos se actualiza de manera constante y todos los antivirus mandan información a través de internet para mantenerse al día.

  Cada vez que el antivirus identifique que algún software está siguiendo un patrón similar al de un virus intentará eliminarlo y en caso de no poder pondrá todo lo infectado en cuarentena y avisará al usuario final.

  El mayor problema de este tipo de funcionamiento es que no se evita el contagio previo a la detección del virus.

- **De forma activa o técnicas heurísticas:** es la forma más moderna de trabajar de un antivirus. Se basa en el intento de prevenir la entrada de malwares en el equipo monitorizando los procesos del sistema constantemente con la intención de encontrar movimientos sospechosos.

  El mayor problema ahora mismo de este funcionamiento es que esa monitorización tiene que llevarse a cabo lo más eficientemente posible para que no confunda otros procesos o archivos con malware cuando no lo son.

3.6. Utilidades

Vamos a describir unas ciertas utilidades que está bien que sepa un administrador de sistemas con la intención de facilitar el día a día en su cometido.

3.6.1. Monitorización del sistema Linux mediante comandos

El uso del terminal de comando por parte de un administrador de sistemas es debido a que da una información muy detallada de manera muy rápida.

Los principales comandos para monitorizar un sistema Linux son:

- **uptime:** presenta la siguiente información:
  - Hora del sistema y el tiempo que lleva encendido.
  - Número de usuarios conectados.
  - Valor medio de la carga en:
    1. El último minuto.
    2. Los últimos 5 minutos.
    3. Los últimos 15 minutos.

```
root@debian:/home/miguel# uptime
 10:01:20 up 0 min,  1 user,  load average: 0,97, 0,28, 0,09
root@debian:/home/miguel#
```

Imagen 8. Comando uptime.

- **time programa:** nos permite ver la distribución del tiempo que ha tardado en ejecutar un programa concreto nuestro procesador tanto en modo usuario como en modo supervisor.
- **top:** podemos ver con este comando todos los procesos que hay en ejecución y su consumo de memoria en tiempo real.

```
top - 10:05:38 up 4 min,  1 user,  load average: 0,02, 0,12, 0,07
Tasks: 162 total,   1 running, 161 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,0 us,  0,0 sy,  0,0 ni,100,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :    976,5 total,    110,0 free,    489,2 used,    377,2 buff/cache
MiB Swap:    975,0 total,    899,5 free,     75,5 used.    340,9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  163904   8596   6104 S   0,0   0,9   0:01.16 systemd
      2 root      20   0       0      0      0 S   0,0   0,0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_par+
      5 root      20   0       0      0      0 I   0,0   0,0   0:00.00 kworker+
      6 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker+
      7 root      20   0       0      0      0 I   0,0   0,0   0:00.05 kworker+
      8 root      20   0       0      0      0 I   0,0   0,0   0:00.00 kworker+
      9 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 mm_perc+
     10 root      20   0       0      0      0 S   0,0   0,0   0:00.00 rcu_tas+
     11 root      20   0       0      0      0 S   0,0   0,0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0,0   0,0   0:00.07 ksoftir+
     13 root      20   0       0      0      0 I   0,0   0,0   0:00.06 rcu_sch+
     14 root      rt   0       0      0      0 S   0,0   0,0   0:00.00 migrati+
     15 root      20   0       0      0      0 S   0,0   0,0   0:00.00 cpuhp/0
     17 root      20   0       0      0      0 S   0,0   0,0   0:00.00 kdevtmp+
     18 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 netns
```

Imagen 9. Salida del comando top.

- **ps:** nos muestra los procesos del sistema que han sido lanzados por el usuario que los invoca.

```
root@debian:/home/miguel# ps
    PID TTY          TIME CMD
   1841 pts/0    00:00:00 su
   1842 pts/0    00:00:00 bash
   2135 pts/0    00:00:00 top
   2177 pts/0    00:00:00 ps
```

Imagen 10. ps invocado por el usuario root.

Además de los comandos mencionados los sistemas Linux tienen implementados en su línea de comandos un conjunto de herramientas llamadas *Sysstat*. Estas herramientas se usan para monitorizar el análisis del rendimiento del equipo.

Algunas de las herramientas que presenta este conjunto son:

- **iostat.** Muestra las estadísticas de entrada/salida de los dispositivos, particiones y sistemas de ficheros en la red.

```
root@debian:/home/miguel# iostat
Linux 5.10.0-11-amd64 (debian)  17/02/22        _x86_64_        (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           2,96    0,05    1,05    0,23    0,00   95,71

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_w
rtn    kB_dscd
sda              40,94      1181,15       784,67         0,00    1258464     836
036           0
sr0               0,03         0,07         0,00         0,00         72
  0           0
```

Imagen 11. Comando iostat.

- **mpstat.** Nos genera las estadísticas del procesador.

```
root@debian:/home/miguel# mpstat
_inux 5.10.0-11-amd64 (debian)  17/02/22        _x86_64_        (1 CPU)

10:23:02     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest
  %gnice   %idle
10:23:02     all    2,35    0,04    0,81    0,18    0,00    0,02    0,00    0,00
    0,00   96,59
```

Imagen 12. Comando mpstat.

- **pidstat.** Nos informa de los procesos activos del sistema.

```
root@debian:/home/miguel# pidstat
Linux 5.10.0-11-amd64 (debian)  17/02/22        _x86_64_        (1 CPU)

10:35:04      UID       PID    %usr %system  %guest   %wait    %CPU   CPU  Comma
nd
10:35:04        0         1    0,03    0,06    0,00    0,04    0,09     0  syste
md
10:35:04        0        12    0,00    0,00    0,00    0,02    0,00     0  ksoft
irqd/0
10:35:04        0        13    0,00    0,00    0,00    0,05    0,00     0  rcu_s
ched
10:35:04        0        14    0,00    0,00    0,00    0,00    0,00     0  migra
tion/0
10:35:04        0        23    0,00    0,01    0,00    0,00    0,01     0  kcomp
actd0
10:35:04        0        25    0,00    0,00    0,00    0,00    0,00     0  khuge
paged
10:35:04        0        48    0,00    0,01    0,00    0,01    0,01     0  kwork
er/0:1H-kblockd
10:35:04        0        49    0,00    0,04    0,00    0,15    0,04     0  kswap
d0
10:35:04        0       105    0,00    0,00    0,00    0,00    0,00     0  scsi_
eh_0
10:35:04        0       106    0,00    0,00    0,00    0,00    0,00     0  scsi_
```

Imagen 13. Comando pidstat.

- **sar.** Monitoriza y recoge información acerca de todas las actividades del sistema que tengan relación la CPU, la memoria, las interrupciones o llamadas al sistema, las interfaces, y las tablas del *kernel*.

Todas estas utilidades nos ofrecen información acerca del sistema como puede ser:

- Las tasas de entrada/salida y transferencias
- La carga de la CPU
- El uso de memoria
- La paginación y su carga de memoria y fallos
- La velocidad de generación de nuevos procesos
- Lel número de interrupciones
- La cola de ejecución
- La carga del sistema

Normalmente toda esta información se agrupa en 5 grupos: memoria, red, procesadores, CPU y E/S.

3.6.2. Gestión de recursos (memoria, disco, etc.) mediante comando en Linux

Ahora que ya hemos hablado acerca de cómo los comandos como *top* muestran de manera seguida los procesos que se ejecutan en el equipo y su consumo de recursos, tenemos que hablar de tres comandos que también nos van a facilitar la administración del sistema.

Estos son: `df`, `du` *y* `free`.

df [opciones] [directorio]

Este comando viene de *disk free* y se puede ejecutar con o sin opciones adicionales.

Si lo mostramos solo, sin ninguna opción, nos mostrará la información referente al espacio que tenemos libre en el disco, el que tenemos usado y así, con todos los discos que se encuentran montados en el sistema.

Si especificamos un directorio. Nos mostrará la información de espacio usado justo en la ruta donde se encuentra el directorio.

La opción *-h* hace que se nos muestre el resultado en formato legible para el ser humano, no es que el otro no lo sea, es que este es más sencillo porque usa las medidas convencionales.

```
root@debian:~# df -h
S.ficheros     Tamaño Usados  Disp Uso% Montado en
udev             466M      0  466M   0% /dev
tmpfs             98M   1,1M   97M   2% /run
/dev/sda1        6,9G   5,0G  1,6G  77% /
tmpfs            489M      0  489M   0% /dev/shm
tmpfs            5,0M   4,0K  5,0M   1% /run/lock
tmpfs             98M   116K   98M   1% /run/user/1000
root@debian:~# df
S.ficheros     bloques de 1K  Usados Disponibles Uso% Montado en
udev                  476984       0      476984   0% /dev
tmpfs                  99992    1120       98872   2% /run
/dev/sda1            7173040 5166172     1621128  77% /
tmpfs                 499944       0      499944   0% /dev/shm
tmpfs                   5120       4        5116   1% /run/lock
tmpfs                  99988     116       99872   1% /run/user/1000
root@debian:~#
```

Imagen 14. df -h y df.

du [opciones][path]

El comando *du* nos muestra el espacio total del disco que ocupan los ficheros y subdirectorios de la ruta en la que se lance. Esto lo hace siempre que se lance sin opciones, si por el contrario especificamos una ruta, nos lo mostrará de esta ruta.

Este comando también posee la opción `-h`.

```
root@debian:~# du
8       ./.cache/dconf
4       ./.cache/appstream
16      ./.cache
8       ./.dbus/session-bus
12      ./.dbus
4       ./.config/procps
8       ./.config
8       ./.synaptic
60      .
root@debian:~# du -h
8,0K    ./.cache/dconf
4,0K    ./.cache/appstream
16K     ./.cache
8,0K    ./.dbus/session-bus
12K     ./.dbus
4,0K    ./.config/procps
8,0K    ./.config
8,0K    ./.synaptic
60K     .
```

Imagen 15. du y du -h.

free [opciones]

El comando *free* muestra la información acerca de la RAM y el espacio *swap* usado en el momento en que se lanza el comando. Nos permitirá ver el uso de cada una de las dos categorías, así como el total. Tiene las siguientes opciones:

- **-b:** la salida se muestra en bytes.
- **-k:** la salida se muestra en kilobytes.
- **-m:** la salida se muestra en megabytes.

```
root@debian:~# free
               total        used        free      shared  buff/cache   available
Mem:          999888      461956      135912        6548      402020      387976
Swap:         998396      171720      826676
root@debian:~# free -b
               total        used        free      shared  buff/cache   available
Mem:       1023885312   473042944   139173888     6705152   411668480   397287424
Swap:      1022357504   175841280   846516224
root@debian:~# free -m
               total        used        free      shared  buff/cache   available
Mem:             976         451         132           6         392         378
Swap:            974         167         807
root@debian:~# free -k
               total        used        free      shared  buff/cache   available
Mem:          999888      461956      135912        6548      402020      387976
Swap:         998396      171720      826676
```

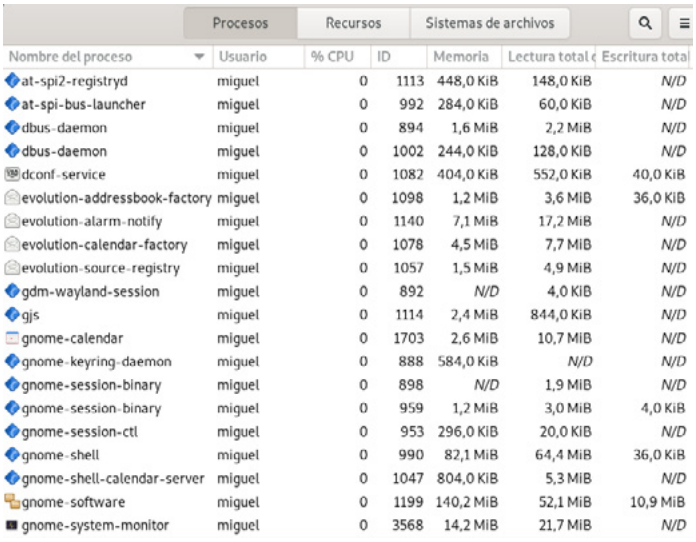
Imagen 16. Comando free sin opciones y con sus tres opciones.

NOTA

Si nos fijamos en la imagen anterior podremos ver que, si usamos el comando free sin opciones, por defecto muestra la información en kilobytes.

3.6.3. El monitor del sistema en Linux

El 'Monitor del sistema' de Linux, al igual que con Windows nos dará toda la información que hemos ido mostrando antes, aunque algo más básica, mediante una interfaz gráfica.



Nombre del proceso	Usuario	% CPU	ID	Memoria	Lectura total	Escritura total
at-spi2-registryd	miguel	0	1113	448,0 KiB	148,0 KiB	*N/D*
at-spi-bus-launcher	miguel	0	992	284,0 KiB	60,0 KiB	*N/D*
dbus-daemon	miguel	0	894	1,6 MiB	2,2 MiB	*N/D*
dbus-daemon	miguel	0	1002	244,0 KiB	128,0 KiB	*N/D*
dconf-service	miguel	0	1082	404,0 KiB	552,0 KiB	40,0 KiB
evolution-addressbook-factory	miguel	0	1098	1,2 MiB	3,6 MiB	36,0 KiB
evolution-alarm-notify	miguel	0	1140	7,1 MiB	17,2 MiB	*N/D*
evolution-calendar-factory	miguel	0	1078	4,5 MiB	7,7 MiB	*N/D*
evolution-source-registry	miguel	0	1057	1,5 MiB	4,9 MiB	*N/D*
gdm-wayland-session	miguel	0	892	*N/D*	4,0 KiB	*N/D*
gjs	miguel	0	1114	2,4 MiB	844,0 KiB	*N/D*
gnome-calendar	miguel	0	1703	2,6 MiB	10,7 MiB	*N/D*
gnome-keyring-daemon	miguel	0	888	584,0 KiB	*N/D*	*N/D*
gnome-session-binary	miguel	0	898	*N/D*	1,9 MiB	*N/D*
gnome-session-binary	miguel	0	959	1,2 MiB	3,0 MiB	4,0 KiB
gnome-session-ctl	miguel	0	953	296,0 KiB	20,0 KiB	*N/D*
gnome-shell	miguel	0	990	82,1 MiB	64,4 MiB	36,0 KiB
gnome-shell-calendar-server	miguel	0	1047	804,0 KiB	5,3 MiB	*N/D*
gnome-software	miguel	0	1199	140,2 MiB	52,1 MiB	10,9 MiB
gnome-system-monitor	miguel	0	3568	14,2 MiB	21,7 MiB	*N/D*

Imagen 17. Monitor del sistema en Debian.

Como se puede observar en la imagen anterior, tiene tres pestañas:

- Procesos
- Recursos
- Sistemas de archivos

La pestaña con la que se abre, que es la de 'Procesos' es la que se ve en la imagen de más arriba y nos muestra el ID del proceso y los recursos consumidos.

Si hacemos clic derecho sobre algún proceso podemos:

- Ver sus propiedades
- Ver sus mapas de memoria
- Ver que archivos tiene abierto el proceso
- Modificar la prioridad
- Otras opciones:
  - Detener
  - Continuar
  - Finalizar
  - Matar

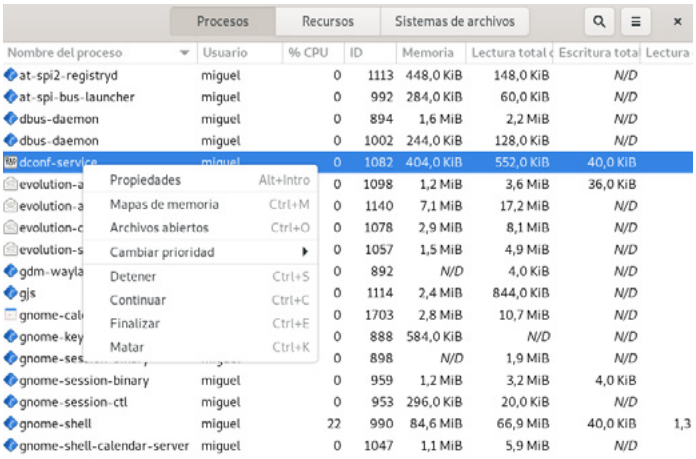


Imagen 18. Opciones sobre procesos.

Si nos dirigimos a la pestaña de 'Recursos', se nos mostrará un panel interactivo sobre el uso de la CPU, la memoria y la red del equipo.

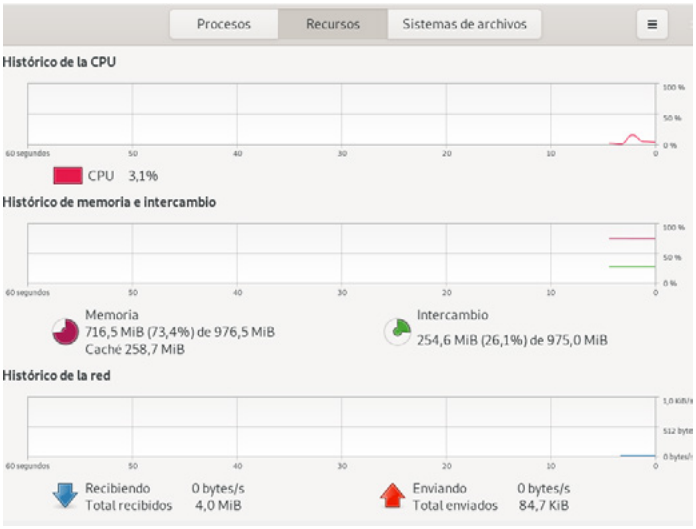


Imagen 19. Pestaña 'Recursos'.

Esta pestaña es importante ya que la información que muestran los gráficos nos mostrará el estado en que se encuentra nuestro sistema en cuanto a recursos principales nos referimos.

Por último, la pestaña 'Sistemas de archivos', nos va a mostrar las particiones del sistema, sus puntos de montaje, los sistemas de archivos que contiene cada una de ellas, su tamaño total, el espacio que queda libre, el espacio que queda disponible y el espacio usado.

Además de todo esto nos mostrará una barra con los porcentajes.

Imagen 20. Pestaña 'Sistemas de archivos'.

www.universae.com