

# Gestión del software, actualizaciones y auditorias.

## Índice de contenidos:

1. Instalación de software. Actualización y recuperación del sistema.
2. Gestión de paquetes. Repositorios.
3. Auditorías. Análisis forense.
4. Gestión de incidencias. Copias de seguridad.

## Objetivos de Aprendizaje:

1. Estudiar los procedimientos de instalación de software en Ubuntu GNU/Linux y los mecanismos de actualización y recuperación del sistema.
2. Analizar la gestión de paquetes y repositorios del sistema.
3. Estudiar los conceptos de auditoría informática y análisis forense, sus implicaciones y usos.
4. Reconocer la importancia de la gestión de incidencias en el sistema informático y la realización de copias de seguridad.

## Contenido

Gestión del software, actualizaciones y auditorias .....	1
1.- Instalación de software. Actualización y recuperación del sistema.....	2
1.1 Mecanismos de instalación de software.....	3
1.2 Gestión del software. Actualizaciones.....	5
1.3 Recuperación del sistema. ....	6
2.- Gestión de paquetes: repositorios. ....	7
3.- Auditorías. Análisis forense.....	11
3.1 Análisis forense digital.....	13
4.- Gestión de incidencias y copias de seguridad.....	15
4.1 Gestión de incidencias.....	15
4.2 Copias de seguridad. ....	17
4.3 Recuperación de la información. ....	20

## **1.- Instalación de software. Actualización y recuperación del sistema.**

La instalación de la distribución GNU/Linux elegida comprende, además del sistema operativo propiamente dicho, una serie de aplicaciones de uso común y frecuente por parte de los usuarios, tales como el navegador web, un procesador de textos y toda serie de utilidades.

Pero la tarea del administrador requiere la instalación de otras herramientas que no vienen en la imagen ISO instalada, tanto en la versión de servidor como en la de escritorio. Ya se ha comprobado en las unidades anteriores, en las que continuamente hemos necesitado de software complementario para cada una de las funcionalidades explicadas.

En este apartado se explican los diferentes procedimientos de adquisición, instalación y configuración de software adicional, así como las actualizaciones tanto de este software como del propio sistema operativo.

En primer lugar se van a explicar diferentes procedimientos para la instalación de software en Ubuntu GNU/Linux. A continuación se explica la gestión del software, su actualización y recuperación del sistema.

En un principio para instalar software en GNU/Linux se hacia siempre a partir del código fuente y se compilaba en la misma máquina en la que se quería instalar. De esta forma se obtenía el archivo binario, es decir, el ejecutable.

En la actualidad las diferentes distribuciones GNU/Linux realizan estas tareas previamente y lo que suministran son paquetes con una extensión que dependerá de la distribución a la que va dirigido. Pero la opción de compilar 'in situ' sigue existiendo para aquellos administradores que lo prefieran.

## 1.1 Mecanismos de instalación de software.

### A. Mediante archivos con extensión .bin

Archivos con extensión .bin son archivos ejecutables en GNU/Linux, es decir, están en binario. No es el formato habitual pero mucho software de tipo comercial viene en .bin, y generalmente no son software libre.

Por defecto los archivos en Linux no tienen permiso de ejecución y los archivos .bin que se descargan tampoco. Para poder ejecutarlos hay que asignar dicho permiso.

Desde el entorno gráfico se puede hacer. Hay que situarse sobre la referencia del .bin descargado y en el menú flotante mostrado pulsando el botón derecho del ratón ir a *Propiedades > Permisos*, marcar la opción *Permitir ejecutar el archivo como un programa*.

Luego, para ejecutar hacer doble clic sobre el archivo.

Desde terminal también se puede ejecutar la orden chmod para añadir ejecución:

```
$ sudo chmod +x nombre_archivo.bin
```

E instalar el archivo .bin, estando en el mismo directorio, ejecutando:

```
$ sudo ./[nombre_archivo].bin
```

Si no se está en el mismo directorio que el archivo .bin hay que indicar el camino relativo o absoluto (completo).

### B. Mediante archivos con extensión .sh

Los archivos con extensión .sh son guiones o scripts. Su contenido suele ser instrucciones que ejecutará el shell para realizar, la mayoría de veces, gestiones de administración del sistema.

La forma más general y común a todas las distribuciones GNU/Linux es ejecutar los archivos .sh desde una terminal, y, estando en el directorio donde se encuentra el .sh, mediante los comandos siguientes:

Primero asignar permiso de ejecución:

```
$ chmod +x nombre_script.sh
```

### **C. Mediante archivos de extensión .run**

Los archivos .run generalmente ayudan en la instalación de software. Si no tienen permiso de ejecución hay que asignarlo o se pueden ejecutar directamente desde terminal con el comando siguiente:

```
$ sh ./archivo.run
```

Si requiere permisos de root añadir delante sudo.

```
$ sudo sh ./archivo.run
```

### **D. Mediante compilación del código fuente**

Normalmente estos archivos vienen comprimidos con extensiones tar.gz o tar.bz2. Se suelen utilizar cuando el software que se quiere instalar no está disponible como paquete específico para esa distribución.

Suelen llevar un archivo README o INSTALLATION que es muy importante leer, ya que en él se explican detalles de instalación para diferentes distribuciones.

Este procedimiento requiere varios pasos:

1. Instalar el meta-paquete build-essential.

```
$ sudo apt update  
$ sudo apt install build-essential
```

2. Descargar el archivo con el código fuente (.tar.gz o .tar.bz2).

3. Ahora hay que descomprimir el archivo descargado. Se puede realizar desde el entorno gráfico. Estando sobre el archivo pulsar el botón derecho y seleccionar *Extraer aquí*. Al descomprimir se crea un directorio con todo el contenido del archivo tar.

4. Ir al directorio creado y ejecutar el archivo *configure*. Este archivo es un script que realiza las comprobaciones del sistema que pueden influir en la compilación. Con todas ellas se configura la futura compilación y genera un archivo *makefile*.

```
$ cd directorio_creado_descompresion  
$ ./configure
```

5. La compilación propiamente la realiza el comando *make*.

```
$ sudo make
```

6. Ahora es el momento de la instalación del software propiamente.

```
$ sudo make install
```

## 1.2 Gestión del software. Actualizaciones.

Todo el software, independientemente del formato en el que se suministre, tiene asignada una versión. El objetivo de la versión es indicar el punto en el que se encuentra en su evolución o desarrollo. Consiste en asignar un código numérico, que normalmente es compuesto. Recordar la nomenclatura del kernel.

Existen diferentes técnicas de numeración de versiones, pero en el ámbito del software libre la más extendida es la siguiente. El código numérico consta de tres números y otro más que es opcional separados por puntos.

`major.minor.revision[-fase]`

Cuyo significado es:

- major number: versión principal del software. Funcionalidad principal del software.
- minor number: funcionalidad menor proporcionada en la versión de software entregada.
- revision: revisiones de código cuando hay fallos del software.
- fase: indica si se encuentra en una fase de desarrollo que no sea la final o estable, es decir, una fase inestable o de pruebas. Se suele indicar con un guion seguido de la fase correspondiente en minúsculas, o un espacio seguido de la fase.

Estos números se incrementan cuando:

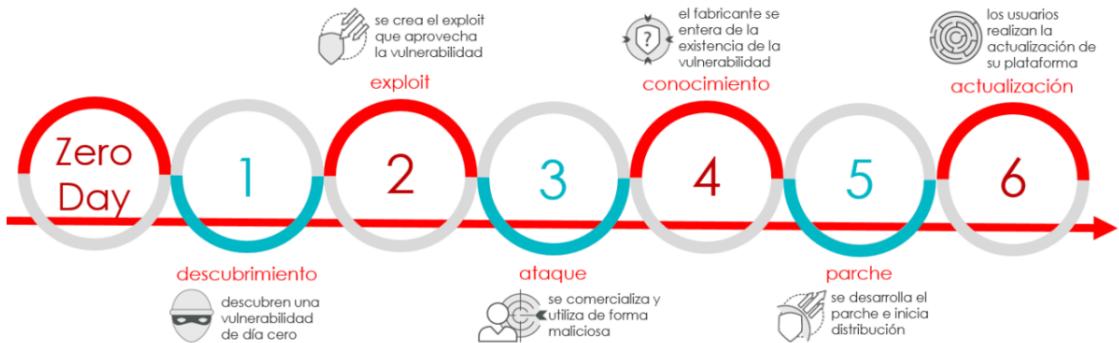
- major: el software sufre grandes cambios y mejoras.
- minor: el software sufre pequeños cambios.
- revision: se aplica una corrección al software, y además hay pocos cambios.
- fase: puede haber varias versiones de una misma fase, para indicar el avance en el desarrollo del software pero manteniendo la fase para indicar que todavía es inestable, indicándose añadiendo un número al final del nombre de la fase que va incrementando conforme se publiquen nuevas versiones de esta fase.

Ejemplo: VNC-Viewer-6.21.1109-Linux-x86.deb, en este caso el major es 6, el minor es 21 y la revisión es 1109. No se indica fase.

Mantener el sistema actualizado con las últimas versiones estables de los paquetes es una medida de seguridad muy importante. Los errores y fallos detectados mediante la explotación del sistema deben ser eliminados, sobre todo cuando estamos en entornos de servidores que ofrecen funcionalidad crítica. Y, en general, se deben tener los equipos siempre actualizados aunque se trate de un equipo doméstico, aparentemente sin importancia. Actualizar los sistemas debe ser una rutina integrada en el proceso de administración, no importa el nivel de relevancia.

Por lo tanto, mantener TODOS los equipos actualizados es una tarea de los administradores de sistemas, ya que de lo contrario y sobre todo si están expuestos en Internet se puede comprometer la información almacenada.

La siguiente figura ilustra las diferentes fases por las que pasa la detección y resolución de una vulnerabilidad tipo 'día cero' (vulnerabilidad que acaba de ser descubierta y aún no hay un parche que la solucione).



Se define parche como una porción de código que modifica un software ya instalado en el equipo, solucionando un error, ampliando la funcionalidad del software, o modificando su comportamiento.

Cuando se detecta una vulnerabilidad los desarrolladores del software generan el parche adecuado que le debe ser aplicado solucionando así el problema.

El parche puede liberarse como:

- Código fuente: habrá que compilar e incluir en el software.
- Revisión de un software (Versión menor). Hay que actualizar.

Como resumen diremos que:



#### IMPORTANTE

Hay que mantener los sistemas actualizados.

### 1.3 Recuperación del sistema.

Ubuntu proporciona una opción de arranque del sistema en 'modo seguro' que permite algunas tareas de recuperación, por ejemplo de claves, o ejecutar una terminal root para permitir el acceso completo al equipo y comprobar su estado.

Otra situación en la que se puede requerir la recuperación del sistema es cuando se intenta instalar software, no se realiza correctamente, y el sistema queda inestable. En estos casos puede ser de interés recuperar el estado del sistema a la situación previa a la instalación de este software.

La herramienta `systemback` permite la creación de copias de seguridad del sistema y los archivos de configuración de los usuarios. Si hay necesidad por cualquier circunstancia, se puede restaurar el estado anterior del sistema.

El **Caso práctico extendido del apartado 1** aborda la instalación, configuración y uso de esta herramienta.

## **2.- Gestión de paquetes: repositorios.**

Pasamos ahora a estudiar la gestión de paquetes en Ubuntu, la agrupación de los mismos en repositorios y el concepto de dependencia.

Todos ellos son pilares fundamentales de cualquier distribución GNU/Linux, pero cada una de ellas utiliza formatos de paquetes diferentes, impidiendo de esa forma la interoperabilidad entre ellas.

Hacemos especial hincapié en las dependencias, ya que constituyen un aspecto fundamental en la instalación y funcionamiento del software instalado.

En una distribución GNU/Linux en general y Ubuntu en particular...

... Se define paquete como una agrupación de elementos (scripts, bibliotecas, archivos de texto, licencias, etcétera) que permiten la instalación, de forma sencilla y eficiente, de un software específico en el sistema.

El término paquete hace referencia a que todos estos elementos están empaquetados, reunidos y comprimidos. La extensión de los paquetes para Ubuntu y Debian es .deb. Existen otros tipos de paquetes disponibles para Ubuntu y otras distribuciones que utilizan extensiones diferentes para identificar sus paquetes.

El elemento del sistema encargado de la gestión de los paquetes en Ubuntu es APT. APT es el responsable de la correcta instalación y actualización de los paquetes. En el entorno gráfico existe la aplicación Ubuntu Software cuya función es la gestión de paquetes.

Los paquetes se agrupan en **repositorios** (pools).

Un repositorio APT es un servidor de red (ftp o http) o un directorio local (incluido un CD-ROM) que contiene paquetes .deb y archivos de metadatos accesibles mediante el comando apt.

Cuando se instala una aplicación el sistema busca en los repositorios los paquetes necesarios para ello: imágenes, bibliotecas, código fuente, documentación, traducciones, etcétera, además del programa propiamente. Este software se almacena en repositorios diferenciados según sean sus características o procedencia. De esta forma se controla el acceso a determinado software por su carácter privativo.

Además de los servidores principales existen servidores espejo repartidos por el mundo para no saturarlos y además se gana en eficiencia, ya que, al poder acceder a alguno más cercano la descarga será más rápida.

Los repositorios en Ubuntu se pueden agrupar en tres tipos básicos:

1. Repositorios tradicionales
2. Archivos personales de paquetes (PPA)
3. Tienda de Snap

Los repositorios tradicionales son el mecanismo principal de instalación de paquetes en Ubuntu. Y en general todos ellos (paquetes .deb) tienen dependencias.

Definimos las dependencias como software requerido por la instalación de otro software y que no lo lleva incluido. Es decir, que los paquetes pueden necesitar para su correcto funcionamiento un conjunto de bibliotecas (bibliotecas) o programas de apoyo que no incorpora en ellos. Recordar que una biblioteca es un archivo cuyo contenido es un conjunto de funciones.

Tener resueltas las dependencias marcará la diferencia entre que el software instalado funcione correctamente o no.

Pueden ser estáticas y dinámicas.

- **Biblioteca estática:** queda incorporada al propio programa. Por lo tanto si es requerida por varios programas existirá una copia por cada uno de ellos.
- **Biblioteca dinámica:** se ejecuta cuando se necesita, es decir, se comparte con todos los programas que la solicitan.

La siguiente infografía muestra el concepto de dependencia de paquetes de software en general para GNU/Linux.

Volvemos a insistir en la importancia de leer los archivos README con instrucciones de instalación incorporados en los paquetes de aplicaciones, ya que en ellos se especifican las dependencias. También pueden ser consultadas mediante los gestores de paquetes.

A su vez los repositorios tradicionales se clasifican en cinco categorías:

1. Main
2. Universe
3. Multiverse
4. Restricted
5. Asociados de Canonical

Veamos cada uno de ellos.

### Main

Repositorio principal. Contiene paquetes de software libre básicos o imprescindibles que se pueden redistribuir libremente. Desde Canonical se mantienen libres de errores y permanentemente actualizados. Esta activado por defecto.

### Universe

Contiene paquetes de software de código abierto sin soporte garantizado por Canonical. Pero se añaden las actualizaciones y correcciones de seguridad en cuanto están disponibles. Suele estar activado por defecto.

### Multiverse

Contiene paquetes de software con licencias que restringen su modificación y redistribución, siendo los usuarios los responsables de garantizar su cumplimiento. Los desarrolladores de Ubuntu no son responsables del mantenimiento o actualización. Es el caso de ciertos paquetes para la reproducción de contenido multimedia.

No está activado por defecto.

### Restricted

Contiene los controladores para dispositivos que no tienen licencia libre (privativos) y no pueden ser modificados ni redistribuidos sin permiso de los desarrolladores originales. Este tipo de software se puede instalar/desinstalar sin dañar el sistema operativo, pero puede afectar a su funcionalidad. Por ejemplo, los controladores para determinadas tarjetas gráficas o de red.

### Asociados de Canonical (Partner)

Contiene software con licencias restrictivas pero con acuerdos con Canonical que permiten su distribución en Ubuntu. No se garantiza nada: ni actualización ni corrección de errores ni se da solución a los problemas de seguridad.

Mantenerlo desactivado por defecto.

En el grupo 2 están los **archivos personales de paquetes (PPA)**

Ubuntu tiene líneas de desarrollo de software 'oficiales', pero otros usuarios desarrolladores pueden publicar sus aplicaciones sin tener que pasar los duros testeos de Ubuntu. Son los archivos personales de paquetes o PPA.

Este tipo de repositorio no está físicamente disponible en los mismos servidores oficiales. Utilizan el servicio Launchpad (de Canonical) y la instalación de este tipo de paquetes PPA requiere agregar el repositorio correspondiente.

Los archivos PPA son específicos para una versión de Ubuntu en concreto.

En el grupo 3 está la tienda de **Snap**.

Como se ha dicho, cada distribución trabaja con un tipo de paquete diferente y eso determina la incompatibilidad entre ellas. Para intentar unificar todos los tipos de paquetes se ha creado un formato 'universal' y es el formato snap.

Existen otros proyectos en este sentido, pero el más avanzado es snap. Tanto es así que desde Ubuntu se tiene la intención de, a la larga, suprimir el formato .deb y solo trabajar con snap. Como ejemplo, decir que el centro de software (Ubuntu Software) ya gestiona este tipo de paquetes.

El paquete `snap` contiene todo lo necesario para su funcionamiento (paquete autocontenido) y se puede actualizar o eliminar sin afectar el resto del sistema operativo.

El archivo de configuración de los repositorios es `/etc/apt/sources.list` y desde el entorno gráfico se puede acceder a los repositorios desde la herramienta *Software y Actualizaciones* (buscar en *Mostrar aplicaciones*).

Ambas herramientas requieren privilegios de administrador y están conectadas, es decir, cualquier cambio en modo terminal actualiza el contenido mostrado desde la herramienta gráfica y viceversa.

Muy importante es hacer una copia de seguridad del archivo `sources.list` si se va a acceder a él. Si se hace algún cambio incorrecto el sistema ya no se actualizará.



### **3.- Auditorías. Análisis forense.**

En las unidades anteriores relativas a GNU/Linux, se han ido describiendo comandos básicos para la gestión del sistema y se han instalado aplicaciones que complementan esta gestión en función de las necesidades de los administradores o usuarios particulares.

Pero, tan importante es dar servicio mediante la instalación y configuración de estas herramientas, como asegurar que su funcionamiento es el esperado.

El objetivo de este apartado es introducir conceptos y herramientas que garanticen el correcto funcionamiento del sistema, detectando problemas e intentando solucionarlos.

Si en el ámbito informático existe alguna certeza es la relativa a la inseguridad de los sistemas informáticos. Ningún sistema informático es completamente seguro, siempre existen vulnerabilidades que pueden transformarse en amenazas si nuestro sistema es sensible a dicha vulnerabilidad.

El uso de herramientas de auditoría ayuda a detectar posibles fallos, errores o debilidades en el sistema informático en base a unas políticas y directrices establecidas.

Incluimos el concepto de auditoría informática dado por Alsina.

Conjunto de procedimientos y técnicas para evaluar y controlar, total o parcialmente, un sistema informático, con el fin de proteger sus activos y recursos, verificar si sus actividades se desarrollan eficientemente y de acuerdo con la normativa informática y general existentes en cada empresa y para conseguir la eficacia exigida en el marco de la organización correspondiente.

En consecuencia una tarea muy importante en relación a la gestión de los sistemas es realizar periódicamente auditorías que ayuden en su funcionamiento seguro y eficaz.

La auditoría debe cubrir dos aspectos o áreas:

1. Área no técnica. Se basa en la comprobación, por parte de terceros, del grado de cumplimiento de las políticas y mecanismos de seguridad de la propia empresa u organización.
2. Área técnica. Se basa en la realización de pruebas independientes para conocer si los mecanismos hardware y software que utiliza la empresa para asegurar el sistema son eficaces.

Aunque la auditoría la puede realizar la propia empresa, siempre se aconseja que sea una entidad independiente la que la lleve a cabo. En cualquiera de los casos se obtienen beneficios, como son la comprobación de la eficacia de las políticas de seguridad, mantener la integridad, seguridad y confidencialidad de los datos, salvaguardar los activos o recursos del sistema, la detección de amenazas internas y externas y la identificación de deficiencias en el sistema y/o vulnerabilidades.

El primer paso para realizar una auditoría es plantear y planificar una serie de preguntas relacionadas con los servicios instalados, los recursos del sistema, la estructura de la red utilizada, los perfiles de usuarios, la utilización de cortafuegos y/o proxys, entre otras. Y a continuación habrá que decidir qué tipo de herramientas se van a utilizar para hacer una monitorización que permita dar respuesta a estas cuestiones con el objetivo de detectar fallos o plantear mejoras en el funcionamiento del sistema.

Pero, a su vez, el mismo proceso de auditoría debe ser monitorizado para tener la certeza de que se está realizando correctamente.

Es importante clasificar y guardar los resultados de las auditorías realizadas ya que sirven de punto de comparación para auditorías posteriores.

Estas herramientas se pueden agrupar en dos grupos:

- Herramientas que auditán el sistema operativo instalado en el equipo. Son más generales, pero permiten auditar determinados servicios dentro del sistema operativo que atienden peticiones a través de un puerto concreto. Es el caso de Nessus y Nagios, que se estudia en el módulo de Seguridad del 2º curso, cuya función principal es la detección de vulnerabilidades en sistemas GNU/Linux.

- Herramientas que auditán servidores de aplicaciones instalados en el sistema operativo. Es el caso de la herramienta sarg, ya estudiada en la unidad 9.

Otra herramienta muy utilizada para realizar auditorías específicas es Lynis (<https://cisofy.com>). Lynis es una herramienta de auditoría de seguridad de código abierto (Licencia GPL) para sistemas GNU/Linux y MacOS. Tiene como objetivo realizar un análisis del estado del sistema, detecta problemas de configuración, obtiene información del sistema y de los paquetes instalados, entre otras muchas funciones.

En el caso práctico del apartado se utiliza Lynis.

Por último decir que existen distribuciones GNU/Linux preparadas para realizar auténticas auditorías informáticas, de software libre y con todas las aplicaciones necesarias incluidas para ello. Es el caso de Kali Linux.

### 3.1 Análisis forense digital.

**Análisis Forense Digital** se define como conjunto de principios y técnicas que comprenden el proceso de adquisición, conservación, documentación, análisis y presentación de evidencias digitales y que, llegado el caso, puedan ser aceptadas legalmente en un proceso judicial.

Definimos ahora el concepto de **evidencia digital**.

Conjunto de datos en formato binario, como archivos, su contenido o referencias a éstos (metadatos), que se encuentren en los soportes físicos o lógicos del sistema atacado.

El análisis forense en sistemas informáticos se encarga de adquirir, preservar, obtener y presentar datos que han sido procesados electrónicamente y guardados en un sistema informático.

Se fundamenta en el principio de transferencia de Locard, que dice que: 'cualquier persona u objeto que entra en la escena de un crimen deja un rastro en la escena o en la propia víctima y, viceversa, también se lleva consigo algún rastro de la escena del crimen'. En nuestro caso el crimen será un delito informático.

Cuando en una investigación criminal son intervenidos elementos informáticos se aplican estas técnicas. Son utilizadas por los investigadores para reconstruir eventos y generar pistas de cómo se hizo el delito.

La validez ante la ley de la evidencia depende de la rigurosidad de los procedimientos utilizados para su recolección, manipulación y conservación.

La computación o investigación forense requiere formación interdisciplinaria: es decir expertos en derecho, criminalistas, tecnologías de información, psicología, expertos en seguridad informática.

Etapas o fases en el análisis forense:

1. Identificación del incidente
2. Captura de las evidencias
3. Preservación de las evidencias

4. Análisis de la información obtenida
5. Elaboración de un informe con las conclusiones del análisis forense

En España existen organizaciones como es CERT (equipo de respuesta para emergencias informáticas), creado por el Ministerio de Ciencia y Tecnología, que realizan análisis forenses para organismos privados y públicos. Estos servicios se prestan a aquellas organizaciones que han sufrido ataques informáticos y también se utilizan para colaborar en la resolución de investigaciones donde existan datos digitales involucrados.

El análisis forense informático siempre está presente en:

- Delitos donde se actúa directamente contra los equipos informáticos.
- Delitos donde los equipos informáticos contienen las evidencias.
- Delitos donde los equipos informáticos son utilizados para cometer el crimen.

En todos los países las fuerzas de seguridad poseen unidades dedicadas al análisis forense informático, tanto para delitos informáticos, como para delitos donde se ha utilizado el equipo informático para cometerlos. En España:

- Brigada de Investigación Tecnológica del Cuerpo Nacional de Policía.
- Grupo de Delitos Telemáticos de la Guardia Civil.

Los objetivos principales de todo análisis forense son:

- Recrear lo que ha ocurrido en un dispositivo digital durante el incidente de seguridad.
- Analizar las incidencias para impedir que se repitan en el futuro.
- Custodiar las evidencias digitales obtenidas.

Los objetivos se deben conseguir sin dañar la información que se investiga, en la medida de lo posible. Si la investigación obliga a duplicar la información, se debe asegurar una destrucción de la información duplicada al final del proceso.

Los objetivos se han cumplido cuando:

- Se sabe cómo se produjo el incidente de seguridad.
- Se conocen las circunstancias bajo las que ocurrió.
- Se conoce la identidad de los atacantes.
- Se sabe cuando ocurrió el incidente.
- Se conocen los objetivos de los atacantes.
- Se tienen las evidencias que lo demuestran.
- Se ha destruido totalmente cualquier información duplicada para realizar la investigación.

La normativa ISO 27037 establece una serie de directrices que sirven para la identificación, recogida, adquisición y protección de evidencias electrónicas.

Este estándar tiene el objetivo de promover las buenas prácticas en cuanto a análisis forense de cualquier dispositivo electrónico. En especial, está pensada para facilitar el trabajo a los forenses informáticos, así como a las administraciones jurídicas que intervienen en este proceso.

## 4.- Gestión de incidencias y copias de seguridad.

### 4.1 Gestión de incidencias.

**Incidencia** es cualquier evento que no forma parte del desarrollo habitual del servicio y que causa, o puede causar una interrupción del mismo o una reducción de la calidad de dicho servicio. (según ITIL, <https://www.itlibrary.org/>).

Existen sistemas de gestión de incidencias que permiten hacer un control y supervisión adecuado a los incidentes notificados por los usuarios/clientes para que sean procesados de forma eficiente y minimizar su impacto negativo en el ámbito correspondiente.



El sistema de gestión de incidencias debe poder registrar las incidencias, abrir lo que se llama un ticket, además de clasificarlas, establecer prioridades, realizar su análisis y diagnóstico, resolverlas y ejecutar el cierre de la incidencia.

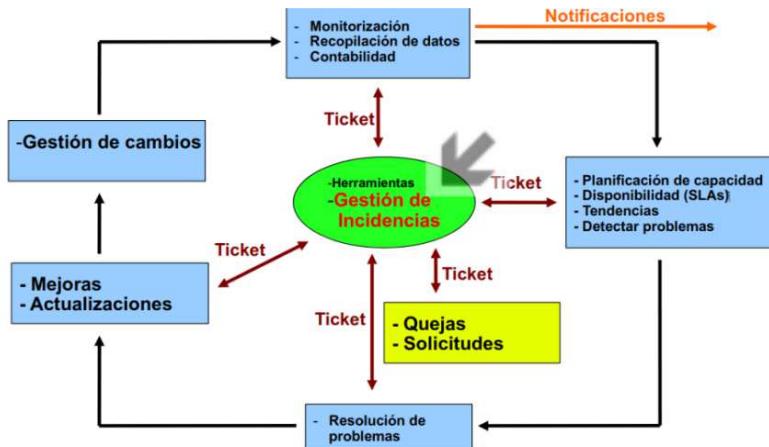
Normalmente los gestores de incidencias trabajan en combinación con otras herramientas de seguridad tipo antimalware y cortafuegos y así poder establecer alertas ante posibles ataques o situaciones de riesgo. Y para llevar a cabo su tarea, los gestores intentan detectar actividades sospechosas, violaciones de seguridad y anomalías en general desde diferentes áreas para reunir información que permita evaluar el sistema.

Es responsabilidad de la organización o empresa establecer, en el marco de las políticas de seguridad, mecanismos de registro, análisis y solución de las diferentes incidencias que se presenten en el sistema informático.

El proceso que generalmente se sigue para la gestión de un incidente es el siguiente:

1. Detección y registro del incidente.
2. Clasificación y soporte inicial.
3. Investigación y diagnóstico.
4. Escalamiento.
5. Solución y restablecimiento del servicio.
6. Cierre del incidente.
7. Monitorización, seguimiento y comunicación del incidente.

La figura siguiente muestra el esquema de funcionamiento de un gestor de incidencias en base al uso de tickets para realizar las diferentes tareas asignadas.



Existe una amplia gama de gestores de incidencias, con capacidades mas o menos similares. En nuestro caso hemos elegido GLPi por ser software libre y tener un procedimiento de instalación y uso muy sencillo.

GLPi es una herramienta de gestión de servicios TIC y un centro de atención que realiza las siguientes tareas:

1. Inventario de equipos, impresoras y otros componentes informáticos.
2. Gestión de incidencias con creación de tickets y seguimiento de éstos últimos.
3. Control financiero.
4. Gestión de la documentación de los equipos.
5. Preguntas frecuentes (F.A.Q).

Para el caso concreto sobre incidentes de seguridad informática recomendamos acceder a la página del INCIBE <https://www.incibe.es/protege-tu-empresa/tematicas/gestion-incidentes-seguridad>.

## 4.2 Copias de seguridad.

Copia de seguridad o **backup** es un duplicado que se realiza sobre archivos o aplicaciones contenidas en un ordenador con la finalidad de poder recuperarlos en el caso de que el sistema de información sufra daños o pérdidas accidentales de los datos almacenados.

El objetivo de las copias de seguridad no es evitar esos problemas, sino poder recuperar los datos en el caso de que ocurran.

Las causas que pueden provocar la pérdida de información son muy variadas, desde el mal funcionamiento de una aplicación hasta una rotura de un disco, pasando por todo tipo de programas maliciosos, motivo por el cual resulta imprescindible planificar y llevar a cabo las tareas de prevención correspondientes.

Todo plan de contingencia de una empresa o usuario particular requiere contar con una planificación adecuada de las copias de seguridad.

Algunos de las condiciones que debe cumplir la planificación de copias de seguridad son:

1. Identificar los datos que deben ser preservados.
2. Establecer la frecuencia con la que se van a realizar los procesos de copia. Esta frecuencia influye en la cantidad de información que se puede perder con respecto a la fuente original. Este parámetro es de suma importancia y requiere de un análisis exhaustivo. Por ejemplo, si se realiza una copia cada noche y el soporte se estropea a las 12h toda la información generada desde la noche anterior hasta las 12h no se encontrará en la copia de seguridad.
3. Disponer del almacén físico para las copias. Este almacén se determina en función de la seguridad que requiere la información entre almacenes en el mismo edificio o remotos en edificios externos. Por ejemplo, si se produce un incendio en el edificio de la empresa, la información almacenada en un edificio externo sigue estando disponible.
4. Buscar una probabilidad de error mínima. Asegurarse que los datos son copiados íntegramente del original y en unos soportes fiables y en buen estado. No se deben utilizar soportes que estén cerca de cumplir su vida útil para evitar que fallen cuando vaya a recuperarse la información que contienen.
5. Controlar los soportes que contienen las copias. Guardarlos en un lugar seguro y restringiendo su acceso sólo a las personas autorizadas.
6. Planificar la restauración de las copias:
  - Formando a los técnicos encargados de realizarlas.
  - Disponiendo de soportes para restaurar la copia, diferentes de los de producción.

- Estableciendo los medios para disponer de dicha copia en el menor tiempo posible.
7. Probar el sistema de forma exhaustiva. Comprobar su correcta planificación y la eficacia de los medios dispuestos.
  8. Definir la vigencia de las copias. Establecer un período en el que dicha copia deja de tener validez y puede sustituirse por una copia más actualizada de la información.

Controlar la obsolescencia de los dispositivos de almacenamiento. Para el caso de aquellas copias que almacenan información histórica de la organización, por ejemplo proyectos ya cerrados, se debe tener en cuenta el tipo de dispositivo en el que se ha realizado la copia, para evitar que en el momento que se requiera la restauración de dicha información no existan ya lectores adecuados para dicho dispositivo.

Cuando se desechen los soportes de almacenamiento, porque hayan llegado al límite de vida útil fijado en la política de copias de seguridad, es importante realizar un proceso de borrado seguro o destrucción para asegurar que la información que contiene no podrá ser recuperada posteriormente.

Existen formas diferentes de hacer las copias de seguridad y esto determina su clasificación en tipos de copias de seguridad. Esto dependerá de cada sistema operativo, pero en general, soportan al menos tres tipos de copia de seguridad.

Los tres tipos de copia de seguridad diferentes son:

**Copia completa (Nivel 0):** copia de todo un sistema o partición, reseteando el bit de archivo.

- Técnicamente son sencillas de realizar.

- Gran consumo de recursos (tiempo, dispositivos. . . ).
- En ocasiones, la restauración es costosa (pero sencilla).
- Recomendables de forma periódica y antes de grandes cambios.

**Copia incremental o progresiva:** el programa examina el bit de archivo y hace una copia de seguridad sólo de los archivos que han cambiado desde la última copia de seguridad incremental o normal.

- Borra el bit de archivo que copia.
- Más rápidas que las completas.
- Muy frecuentes en organizaciones medias o grandes.
- Muchos datos para una completa diaria.
- Recomendables al menos cada dos días.
- Restauración costosa (varios juegos de cintas, si se usan las cintas DAT como soporte, , en caso contrario, este problema no existe.). Para realizar la restauración de archivos ante un desastre debemos disponer de todas las cintas anteriores hasta llegar a la primera copia normal.

**Copia diferencial:** se guardan los archivos modificados desde la ultima copia (normal o incremental) sin resetear el bit de archivo.

- Requiere menos espacio que la copia normal.
- El proceso de restauración únicamente necesita la última cinta con la copia normal.
- El proceso de restauración únicamente necesita la última cinta con la copia normal.
- Consume más tiempo en realizar la copia y más espacio que en la incremental.
- Frecuente en organizaciones medias o grandes.
- Misma información, registrada varias veces.

Respecto al bit de archivo, los archivos que residen en disco disponen de un atributo que indica si dicho archivo ha cambiado desde la última vez que se realizó una copia de seguridad. Este atributo es un único bit para cada archivo que el software de copia de seguridad se encarga de marcar o borrar cuando se necesita.

El atributo reseñado está marcado inicialmente a '0', y cambia su valor a '1' cuando un usuario o el sistema modifican el contenido del archivo correspondiente, volviendo a cambiar su valor a '0' cuando se efectúa una copia de seguridad normal o incremental de dicho archivo.

La realización o restauración de una copia de seguridad puede llevarse a cabo por aquellos usuarios que dispongan de los permisos adecuados para ello.

#### 4.3 Recuperación de la información.

Si se ha perdido el acceso a una información almacenada, dicha pérdida puede no ser definitiva si existen medios para restaurar la disponibilidad de acceso, en cuyo caso se puede plantear la recuperación de la información.

El acceso a la información se puede perder por varios motivos, como son:

1. Porque el dispositivo tiene dañado algún componente físico necesario para su funcionamiento.
2. Porque, aún funcionando correctamente, la "lista de archivos" se ha corrompido impidiendo conocer la ubicación de los archivos almacenados.
3. Porque los datos almacenados han sido reemplazados por nuevos datos a través de la sobreescritura.

En los dos primeros casos se puede intentar un proceso de recuperación, en el tercero no puede realizarse ya que esos datos ya no existen y por tanto no pueden recuperarse.

### A. Métodos de recuperación de la información

La información sigue estando almacenada en los dispositivos, pero no está disponible por alguna de las causas señaladas anteriormente. Los métodos de recuperación de datos se agrupan en 2:

#### 1. Métodos de recuperación lógica.

Los métodos de recuperación lógica se utilizan cuando todos los componentes del dispositivo funcionan correctamente y por tanto se puede acceder a todos y cada uno de los sectores donde se almacena la información. Si se ha perdido acceso a los datos podrá ser debido a que:

- Alguna parte de la estructura del sistema de archivos está dañada.
- Algun archivo ha sido borrado con los comandos del sistema y por tanto no aparece en la "lista de archivos".

La recuperación lógica de datos analiza la estructura de archivos que permanece, identifica el daño producido y permite acceder a los datos que aún están en el dispositivo. En algunos casos se pueden recuperar los datos identificativos del archivo, (nombre, extensión, tamaño, fecha de creación, etc.) y en otros sólo el contenido del mismo.

Las herramientas de reparación de sistemas de archivos que disponen los sistemas operativos tales como `chkdsk` en sistemas Windows o `fsck` en GNU/Linux pueden reparar algunos errores de los sistemas de archivos.

#### 2. Métodos de recuperación física.

Si algún componente físico del dispositivo está dañado, pero el soporte de almacenamiento sigue inalterado, se podrá intentar una recuperación física reparando o sustituyendo el componente dañado y accediendo nuevamente a la información guardada.

Los métodos de recuperación física requieren un conocimiento de cada uno de los dispositivos y no es aconsejable el intento de recuperación por personal no experto.

Existen múltiples herramientas software que permiten la recuperación de datos lógica.

En cualquier caso se recomienda utilizar cada herramienta con las debidas precauciones evitando en todo momento escribir sobre los propios discos de los que se quiere recuperar información.

#### Rescatar la información de un sistema inestable

Pasos a seguir:

4. Asegurarse de que no funciona el sistema.

5. Arrancar en modo seguro e intentar sacar, al menos, a un disco duro externo o un lápiz USB, todos los archivos y carpetas que tengamos en el ordenador que sean importantes para nosotros.
6. Si el sistema no arranca crear un CD live de Ubuntu o un USB live y arrancar desde él y seleccionar la opción 'probar Ubuntu'.
7. Cuando termine de iniciarse el sistema aparece una carpeta personal virtual vacía. Tenemos que navegar por Nautilus hasta el disco duro del ordenador y decir que lo monte simplemente haciendo doble click.
8. Una vez montado estarán todos los archivos y carpetas tal como los dejamos la última vez que arrancamos el ordenador. Ahora el objetivo es sacar toda la información importante a un disco duro externo, o mediante dvds, cds, etcétera. Lo mejor es crear una partición /home, separada de /, al instalar el sistema. De esa forma si le ocurre algún incidente de seguridad a la partición raíz no afectará a nuestros datos.
9. Cuando hayamos copiado todos los archivos importantes, procedemos a instalar de nuevo Ubuntu.
10. Cuando lleguemos al paso de las particiones, seleccionar "manual". Sale la tabla de particiones y marcamos formatear la partición / y damos a siguiente. Si tenemos partición /home, no seleccionar para que no formatee la misma. De esta manera, reinstalamos Ubuntu pero tanto los archivos, carpetas, documentos, imágenes etcétera seguirán tal como estaban.

Si no existe la partición /home separada de la partición /, como no queda mas remedio que formatear la partición /, se borrarán también todos los archivos guardados.

Cuando se inicie el sistema conectar el disco duro externo o dispositivo donde hemos guardado antes los archivos y documentos importantes y restauraremos la copia de seguridad.