

Sistemas Informáticos

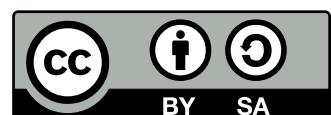
**Libro de texto para el módulo
«Sistemas Informáticos (0483)»
de Formación Profesional**

José J. Durán

Para acceder a los contenidos de este libro, y otros títulos, visita:
<https://github.com/cavefish-dev/libros>

Edición: 10 de septiembre de 2025

Esta obra está bajo una licencia Creative Commons
«Atribución-CompartirIgual 4.0 Internacional».



Índice general

A Introducción

1 Presentación del libro	9
2 Cómo usar este libro	11
2.1 Estructura del libro	12

B Contenidos

3 Explotación de sistemas microinformáticos	15
3.1 Placas base	15
3.2 Estructura y componentes	16
3.3 Adaptadores para la conexión de dispositivos	19
3.4 Normas de seguridad y prevención de riesgos laborales	23
3.5 Redes informáticas	25
3.6 Componentes de una red informática	27
3.7 Topologías de red	29
3.8 Tipos de cableado y conectores	32
3.9 Mapa físico y lógico de una red local	36
4 Instalación de sistemas operativos	41
4.1 Evolución histórica y clasificación	41
4.2 Funciones de un sistema operativo	43
4.3 Tipos de sistemas operativos	44
4.4 Tipos de aplicaciones	46
4.5 Licencias y tipos de licencias	49
4.6 Procedimiento de instalación	51
4.7 Gestores de arranque	54
4.8 Tecnologías de virtualización. Tipos	56

Índice general

4.9 Consideraciones previas a la instalación de sistemas operativos	58
4.10 Instalación de sistemas operativos	60
4.11 Instalación y desinstalación de aplicaciones	60
4.12 Actualización y recuperación de sistemas operativos y aplicaciones	64
4.13 Documentación de la instalación y de las incidencias detectadas	68
5 Gestión de la información	73
5.1 Gestión de sistemas de archivos	73
5.2 Estructura de directorios de sistemas operativos	81
5.3 Búsqueda de información del sistema	83
5.4 Identificación del software instalado	84
5.5 Realización y restauración de copias de seguridad	85
5.6 Herramientas de administración de discos	87
5.7 Tareas automáticas	92
6 Configuración de sistemas operativos	99
6.1 Configuración de usuarios y grupos	99
6.2 Seguridad de cuentas de usuario	102
6.3 Seguridad de contraseñas	104
6.4 Acceso a recursos	106
6.5 Permisos locales	108
6.6 Listas de control de acceso	111
6.7 Servicios y procesos	114
6.8 Comandos de sistemas operativos	118
6.9 Herramientas de monitorización del sistema	123
6.10 Registros y logs	127
7 Conexión de sistemas en red	133
7.1 Configuración del protocolo TCP/IP en un cliente de red	133
7.2 Ficheros de configuración de red	136
7.3 Gestión de puertos	138
7.4 Resolución de problemas de conectividad en sistemas operativos en red	141
7.5 Herramientas gráficas utilizadas en sistemas operativos	143
7.6 Monitorización de redes	144
7.7 Protocolos TCP/IP	147
7.8 Configuración de los adaptadores de red	150
7.9 Interconexión de redes	153
7.10 Redes cableadas	154
7.11 Redes inalámbricas	156
7.12 Seguridad de comunicaciones	158

7.13	Tecnologías de acceso a redes de área extensa	160
8	Gestión de recursos en una red	163
8.1	Permisos y derechos	163
8.2	Configuración de recursos compartidos	166
8.3	Requisitos de seguridad del sistema y de los datos	168
8.4	Servidores de ficheros	169
8.5	Servidores de impresión	173
8.6	Servidores de aplicaciones	175
8.7	Técnicas de conexión remota	179
8.8	Cortafuegos	181
8.9	Implantación y explotación de dominios	182
9	Explotación de aplicaciones informáticas de propósito general	187
9.1	Software: tipos, requisitos, y licencias	187
9.2	Herramientas ofimáticas y de trabajo colaborativo	190
9.3	Utilidades de propósito general	195

A

Introducción

1

Presentación del libro

Este libro está diseñado como un recurso educativo para el módulo de «Sistemas Informáticos (0483)» de Formación Profesional. Su objetivo es proporcionar a los estudiantes una comprensión sólida de los conceptos fundamentales y las prácticas esenciales en el ámbito de los sistemas informáticos, abarcando tanto aspectos teóricos como prácticos. A lo largo del libro, se explorarán temas clave como la arquitectura de los sistemas informáticos, la gestión de hardware y software, las redes de computadoras, la seguridad informática, y la administración de sistemas operativos. Además, se incluirán ejercicios prácticos y ejemplos reales para facilitar el aprendizaje y la aplicación de los conocimientos adquiridos. Este libro está dirigido a estudiantes de Formación Profesional que buscan desarrollar habilidades técnicas y competencias necesarias para desempeñarse eficazmente en el campo de los sistemas informáticos.

2

Cómo usar este libro

A lo largo de este libro encontrarás la información organizada en capítulos y secciones, cada una dedicada a un tema específico. Dentro de cada capítulo, se presentarán ejemplos de código usando el siguiente formato:

```
1 ls -l  
2 cd /home/user/proyecto  
3 git status  
4 echo ":(){ :|:&};"
```

También se incluyen cajas de información destacada, como consejos, advertencias y ejercicios, para resaltar puntos importantes y facilitar la comprensión de los conceptos. Estas cajas están diseñadas para ser visualmente atractivas y fáciles de identificar.

Consejo



A lo largo del libro, encontrarás consejos útiles para mejorar tu comprensión y habilidades en el manejo de los sistemas informáticos. Estos consejos están diseñados para ayudarte a evitar errores comunes y optimizar tu proceso de aprendizaje.

Importante



Es fundamental prestar atención a las advertencias y notas importantes que se presentan en el libro. Estas secciones destacan aspectos críticos que pueden afectar tu comprensión o implementación de los conceptos, o que incluyan información adicional relevante para el tema tratado, pero que sean demasiado avanzados para el nivel del lector.

Ejercicio 2.1



Se incluirán ejercicios prácticos al final de cada capítulo para que puedas aplicar lo aprendido. Estos ejercicios están diseñados para ser desafiantes y fomentar la práctica activa. Los ejercicios se presentan en un formato claro

y conciso, permitiendo al lector enfocarse en la resolución de problemas específicos.

2.1. Estructura del libro

Este libro se divide en varias partes, cada una de las cuales aborda un aspecto diferente de la programación. Se recomienda seguir el orden de los capítulos para construir una comprensión progresiva de los conceptos.

La parte A (*Introducción*) proporciona una visión general de los contenidos de este libro, y como poder ejecutar los ejemplos y ejercicios propuestos.

La parte B (*Contenidos*) cubre los temas principales del libro, incluyendo ejercicios prácticos y ejemplos detallados, para facilitar el aprendizaje y la aplicación de los conceptos.



Contenidos

Explotación de sistemas microinformáticos

La explotación de sistemas microinformáticos abarca el estudio y la gestión de los componentes físicos y lógicos que conforman un ordenador personal y su integración en redes. Este capítulo proporciona una visión detallada de los elementos fundamentales, su funcionamiento y las mejores prácticas para su uso seguro y eficiente.

3.1. Placas base

La placa base es el componente principal de un sistema microinformático, donde se conectan todos los elementos esenciales: procesador, memoria, almacenamiento y periféricos. Los formatos más comunes son ATX, microATX, Mini-ITX y Nano-ITX, cada uno con dimensiones y características específicas que determinan la compatibilidad con cajas y componentes.

Formatos de placas base

- **ATX (Advanced Technology eXtended):** Es el formato estándar más utilizado en equipos de escritorio. Sus dimensiones típicas son 305 x 244 mm. Ofrece múltiples ranuras de expansión, gran cantidad de puertos y buena gestión del flujo de aire. Permite configuraciones avanzadas y es compatible con la mayoría de cajas y fuentes de alimentación.
- **microATX:** Más compacto que ATX, mide aproximadamente 244 x 244 mm. Mantiene muchas de las características del ATX, pero con menos ranuras de expansión y puertos. Es ideal para equipos de tamaño medio, donde se busca un equilibrio entre funcionalidad y espacio.
- **Mini-ITX:** Formato muy pequeño, de 170 x 170 mm. Está diseñado para sistemas compactos y de bajo consumo, como HTPCs y equipos de oficina.



Fuente: Marcin Wieclaw (pcsite.co.uk), CC BY-SA 4.0, via Wikimedia Commons

Figura 3.1: Placa base de un ordenador

na. Ofrece una sola ranura de expansión y menos puertos, pero permite construir ordenadores silenciosos y de tamaño reducido.

- **Nano-ITX:** Aún más pequeño, con dimensiones de 120 x 120 mm. Se utiliza en aplicaciones embebidas, dispositivos industriales y sistemas donde el espacio es crítico. Su capacidad de expansión y conectividad es limitada, pero es ideal para proyectos que requieren máxima miniaturización.

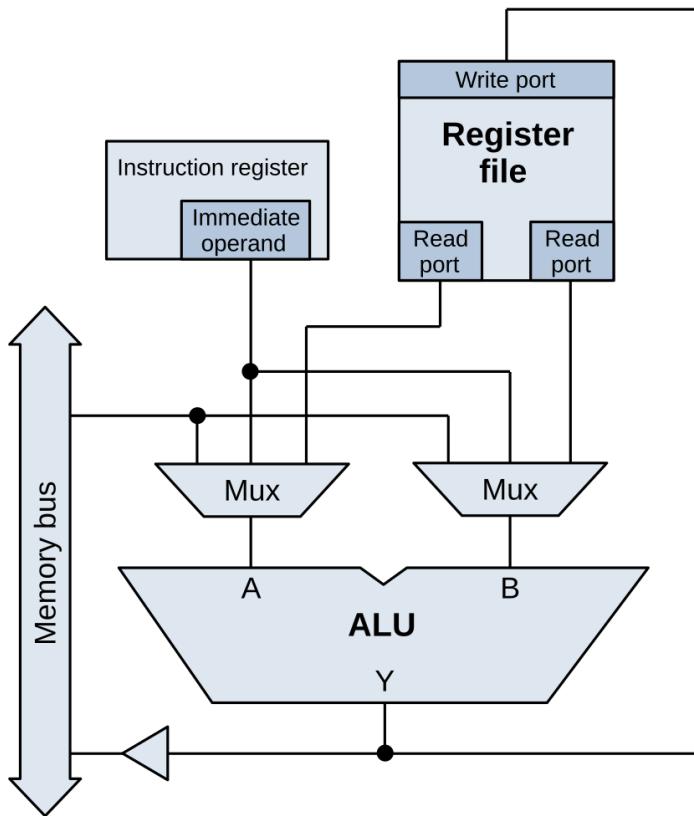
3.2. Estructura y componentes

La estructura de un sistema microinformático está formada por diversos componentes que trabajan de manera conjunta para garantizar el funcionamiento del equipo. Cada elemento cumple una función específica y su correcta integración es fundamental para el rendimiento y la estabilidad del sistema. A continuación se describen los principales componentes y su papel dentro del conjunto.

3.2.1. Procesador (CPU)

El procesador es el núcleo de cualquier sistema informático. Se encarga de ejecutar las instrucciones de los programas y coordinar el funcionamiento de todos los componentes. Sus elementos principales incluyen:

- **Set de instrucciones:** Conjunto de operaciones que el procesador puede realizar, como suma, resta, comparación, etc.



Fuente: Lambtron, CC BY-SA 4.0, via Wikimedia Commons

Figura 3.2: Diagrama de un procesador

- **Registros:** Pequeñas áreas de almacenamiento interno que permiten guardar datos temporales y direcciones.
- **Contador de programa:** Registro que indica la dirección de la siguiente instrucción a ejecutar.
- **Unidad aritmético-lógica (ALU):** Realiza operaciones matemáticas y lógicas.
- **Gestión de interrupciones:** Permite al procesador responder a eventos externos, como la llegada de datos desde un periférico.

La velocidad del procesador se mide en GHz y su arquitectura puede ser de varios núcleos, lo que permite ejecutar múltiples tareas simultáneamente.

3.2.2. Memoria interna

La memoria interna es fundamental para el funcionamiento del sistema, ya que almacena datos e instrucciones de manera temporal o permanente.

- **RAM (memoria volátil):** Almacena datos e instrucciones mientras el ordenador está encendido. Su capacidad y velocidad influyen directamente en el rendimiento.
- **ROM/xPROM (memoria no volátil):** Contiene información esencial para el arranque y funcionamiento básico del sistema, como el firmware.
- **Tipos de memoria:** DDR4, DDR5 y otras tecnologías ofrecen diferentes velocidades, capacidades y consumos energéticos.

La elección de la memoria depende de las necesidades del usuario y la compatibilidad con la placa base.

3.2.3. Interfaces de entrada/salida

Las interfaces de entrada/salida permiten la comunicación entre el sistema y los dispositivos externos e internos.

- **Puertos USB:** Utilizados para conectar periféricos como teclados, ratones, impresoras y discos externos.
- **SATA:** Interfaz para conectar discos duros y unidades ópticas.
- **PCIe:** Ranuras de expansión para tarjetas gráficas, de red y otros adaptadores.
- **Otros puertos:** HDMI, DisplayPort, audio, etc.

La cantidad y tipo de interfaces disponibles varía según el modelo de placa base.

3.2.4. Discos periféricos

Los discos periféricos son dispositivos de almacenamiento que permiten guardar datos de forma permanente.

- **HDD (disco duro mecánico):** Ofrece gran capacidad a bajo coste, pero menor velocidad que otras tecnologías.
- **SSD (unidad de estado sólido):** Mucho más rápido que los HDD, sin partes móviles, ideal para mejorar el rendimiento general.

- **NVMe:** Tecnología de almacenamiento basada en PCIe, proporciona velocidades de transferencia muy superiores a los SSD tradicionales.
- **Unidades ópticas:** Como DVD y Blu-ray, cada vez menos comunes pero útiles para ciertas aplicaciones.

La combinación de diferentes tipos de discos permite optimizar el rendimiento y la capacidad de almacenamiento del sistema.

3.3. Adaptadores para la conexión de dispositivos

Los adaptadores permiten conectar dispositivos adicionales, como tarjetas gráficas, de sonido, de red o controladoras de almacenamiento. Se instalan en ranuras PCI, PCIe o mediante puertos USB. La elección depende de las necesidades del usuario y la compatibilidad con la placa base.

Muchas placas base incluyen adaptadores integrados, pero en algunos casos es necesario añadir tarjetas dedicadas para mejorar el rendimiento o añadir funcionalidades específicas.

3.3.1. Tarjetas gráficas

Las tarjetas gráficas se encargan de procesar y generar imágenes para mostrar en la pantalla. Existen dos tipos principales:

- **Integradas:** Incluidas en el procesador o la placa base, ofrecen rendimiento básico para tareas cotidianas.
- **Dedicadas:** Instaladas en ranuras PCIe, cuentan con memoria y procesadores propios, ideales para juegos, diseño gráfico y edición de vídeo.

La elección depende del uso previsto y del presupuesto. Actualmente, existe una alta demanda en tarjetas gráficas de gama alta, debido a sus capacidades de cálculo paralelo y rendimiento en tareas intensivas como el renderizado 3D y el aprendizaje automático.

3.3.2. Tarjetas de sonido

Permiten la reproducción y grabación de audio. Las tarjetas integradas son suficientes para la mayoría de usuarios, pero las dedicadas ofrecen mejor calidad y funciones avanzadas para profesionales del audio. Dentro de las tarjetas de sonido existen diferentes formatos, como PCI, PCIe y USB, cada uno con sus propias ventajas y desventajas en términos de rendimiento, compatibilidad y facilidad de instalación. También existen distintos propósitos, como la producción musical, el diseño de sonido y el gaming.



Fuente: Cloudschatze, Public domain, via Wikimedia Commons

Figura 3.3: Ejemplo de tarjeta de sonido externa

3.3.3. Tarjetas de red

Facilitan la conexión del equipo a redes cableadas (Ethernet) o inalámbricas (Wi-Fi). Las tarjetas de red pueden estar integradas en la placa base o instalarse como adaptadores adicionales, mejorando la velocidad y la estabilidad de la conexión.

Más adelante, entraremos en detalle en las diferentes conexiones de red, y sus principales características.

3.3.4. Controladoras de almacenamiento

Permiten conectar discos duros, SSDs y otros dispositivos de almacenamiento adicionales. Las controladoras pueden ofrecer soporte para tecnologías RAID, aumentando la seguridad y el rendimiento del almacenamiento. Existen varios tipos de controladoras, entre las que se incluyen:

SATA (Serial ATA) Es el tipo de controladora más común en equipos domésticos y de oficina. Permite conectar discos duros y SSDs mediante cables SATA. Soporta configuraciones RAID básicas y es fácil de instalar y configurar.

NVMe (Non-Volatile Memory Express) Controladoras diseñadas para unidades SSD que utilizan el bus PCIe, ofreciendo velocidades de transferencia muy superiores a SATA. Son ideales para sistemas que requieren alto rendimiento, como estaciones de trabajo y servidores.

RAID (Redundant Array of Independent Disks) Controladoras especializadas que permiten combinar varios discos en arreglos RAID para mejorar el rendimiento, la capacidad o la seguridad de los datos. Existen diferentes niveles de RAID (0, 1, 5, 10, etc.), cada uno con sus propias ventajas y desventajas.

SCSI (Small Computer System Interface) Utilizadas principalmente en servidores y estaciones de trabajo profesionales. Permiten conectar múltiples dispositivos de almacenamiento con alta fiabilidad y rendimiento. Las variantes modernas incluyen SAS (Serial Attached SCSI).

SAS (Serial Attached SCSI) Evolución del SCSI tradicional, ofrece mayor velocidad y flexibilidad. Es común en entornos empresariales y servidores, donde se requiere alta disponibilidad y rendimiento.

IDE (Integrated Drive Electronics) Un estándar más antiguo, actualmente en desuso, que permitía conectar discos duros y unidades ópticas. Ha sido reemplazado por SATA y otras tecnologías más modernas.

Controladoras externas Se conectan mediante USB, Thunderbolt o eSATA y permiten añadir discos duros externos o sistemas de almacenamiento portátiles. Son útiles para copias de seguridad, ampliación de capacidad o transferencia de datos entre equipos.

3.3.5. Adaptadores USB y otros

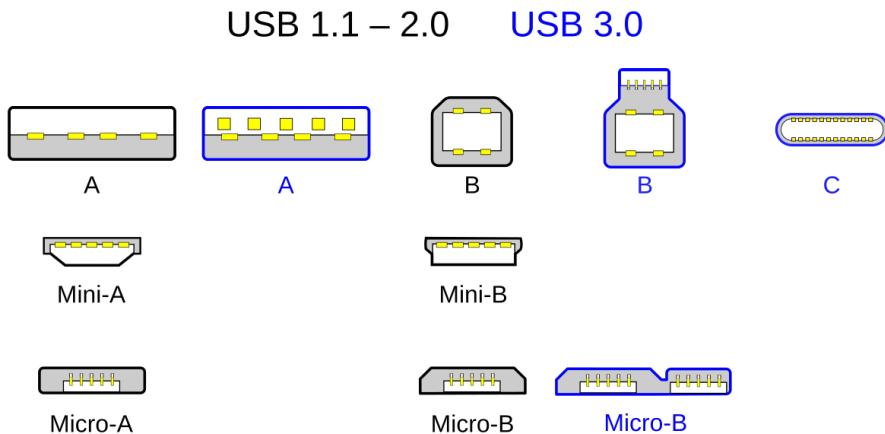
Existen adaptadores para ampliar el número de puertos USB, añadir conectividad Bluetooth, puertos serie, paralelos, o lectores de tarjetas. Estos dispositivos se conectan fácilmente y permiten adaptar el sistema a nuevas necesidades.

Cada tipo de conector USB tiene sus propias características y usos específicos, lo que permite una amplia variedad de dispositivos y configuraciones. Además, cada conector puede usarse en diferentes versiones de USB, lo que garantiza la compatibilidad hacia atrás.

Estándares USB y sus principales características

A lo largo de los años, el estándar USB (Universal Serial Bus) ha evolucionado para ofrecer mayores velocidades de transferencia y nuevas funcionalidades. Los principales estándares USB y sus características son:

USB 1.0/1.1 Primeras versiones, con una tasa de transferencia máxima de 1.5 Mbps (Low Speed) y 12 Mbps (Full Speed). Utilizadas principalmente para teclados, ratones y dispositivos de baja velocidad.



Fuente: Milos.bmx and Andreas Pietzowski, CC BY-SA 3.0, via Wikimedia Commons

Figura 3.4: Distintos tipos de conectores USB

USB 2.0 Introducido en el año 2000, permite hasta 480 Mbps (High Speed). Es compatible con versiones anteriores y sigue siendo común en muchos dispositivos.

USB 3.0 Lanzado en 2008, ofrece hasta 5 Gbps (SuperSpeed). Se reconoce por el color azul en los conectores y mejora significativamente la velocidad de transferencia.

USB 3.1 Presentado en 2013, alcanza hasta 10 Gbps (SuperSpeed+). Introduce el conector USB-C y mejora la eficiencia energética.

USB 3.2 Permite hasta 20 Gbps utilizando dos canales de 10 Gbps sobre USB-C. Ofrece mayor rendimiento en dispositivos compatibles.

USB4 Última generación, soporta hasta 40 Gbps y es compatible con Thunderbolt 3. Permite transmisión de datos, vídeo y energía a través de un solo cable USB-C.

La elección del estándar USB influye directamente en la velocidad de transferencia, la compatibilidad y las capacidades de los dispositivos conectados.

Ejercicio 3.1

Crea un esquema con los conectores de que dispone tu ordenador. ¿Puedes identificar los diferentes tipos de USB?

3.4. Normas de seguridad y prevención de riesgos laborales

La manipulación y el mantenimiento de sistemas microinformáticos requieren la adopción de normas de seguridad para proteger tanto a los usuarios como a los equipos. A continuación se detallan las principales recomendaciones y prácticas para garantizar un entorno de trabajo seguro.

3.4.1. Normas de seguridad en la manipulación de equipos informáticos

La seguridad en el entorno de trabajo informático es esencial para prevenir accidentes y proteger tanto a las personas como a los equipos. A continuación se detallan las principales normas y recomendaciones:

- **Pulseras y alfombrillas antiestáticas:** El uso de pulseras y alfombrillas antiestáticas evita la acumulación de electricidad estática, que puede dañar componentes electrónicos sensibles. Antes de manipular hardware, es recomendable descargar la electricidad estática tocando una superficie metálica conectada a tierra.
- **Desconexión de la corriente eléctrica:** Siempre se debe apagar y desconectar el equipo de la red eléctrica antes de abrir la carcasa o realizar cualquier intervención. Esto reduce el riesgo de descarga eléctrica y protege los componentes.
- **Herramientas adecuadas:** Utilizar destornilladores, pinzas y otros instrumentos específicos para electrónica. Las herramientas deben estar limpias y en buen estado para evitar daños accidentales.
- **Organización del espacio de trabajo:** Mantener el área despejada y bien iluminada facilita la manipulación de piezas pequeñas y reduce el riesgo de accidentes. Es importante identificar y separar tornillos y componentes para evitar pérdidas o confusiones.
- **Identificación de componentes:** Antes de retirar o instalar piezas, identificar correctamente cada componente y su ubicación. Consultar manuales y esquemas si es necesario.
- **Gestión de residuos electrónicos:** Los componentes obsoletos o dañados deben ser gestionados como residuos electrónicos, siguiendo la normativa vigente para su reciclaje y evitando la contaminación ambiental.

- **Protección personal:** En entornos profesionales, es recomendable el uso de guantes aislantes, gafas de protección y ropa adecuada para evitar cortes, quemaduras o exposición a sustancias peligrosas.
- **Ventilación y control ambiental:** Trabajar en espacios bien ventilados y con temperatura controlada ayuda a evitar el sobrecalentamiento de los equipos y la acumulación de polvo.
- **Prevención de incendios:** Disponer de extintores adecuados para equipos eléctricos y conocer los procedimientos de evacuación en caso de emergencia.

3.4.2. Prevención de riesgos laborales específicos

Además de las normas generales, existen riesgos laborales específicos en la explotación de sistemas microinformáticos:

- **Riesgo eléctrico:** Manipular fuentes de alimentación y circuitos puede provocar descargas. Es fundamental comprobar que no hay corriente antes de intervenir y evitar el contacto con partes metálicas expuestas.
- **Riesgo ergonómico:** La postura durante la manipulación y el montaje de equipos debe ser adecuada para evitar lesiones musculares. Utilizar mesas a la altura correcta y realizar pausas periódicas.
- **Riesgo químico:** Algunos componentes contienen sustancias peligrosas (baterías, condensadores, refrigerantes). Manipularlos con precaución y seguir las indicaciones del fabricante para su eliminación.
- **Riesgo de corte y pinchazo:** Los bordes de las carcasa y algunos componentes pueden ser afilados. Manipular con cuidado y utilizar guantes si es necesario.
- **Riesgo de incendio:** Un mal montaje o cortocircuito puede provocar incendios. Revisar conexiones y evitar sobrecargas eléctricas.

Importante

Las leyes que protegen a los trabajadores frente a accidentes laborales son fundamentales para garantizar la seguridad y el bienestar en el entorno profesional. En España, la Ley 31/1995 de Prevención de Riesgos Laborales establece las obligaciones de empresas y trabajadores en materia de seguridad, incluyendo la evaluación de riesgos, la formación, la vigilancia de la salud y la adopción de medidas preventivas. Además, el Real Decreto

486/1997 regula las condiciones mínimas de seguridad y salud en los lugares de trabajo. Estas normativas obligan a los empleadores a proporcionar equipos de protección, informar sobre los riesgos y actuar ante cualquier incidente, asegurando la protección legal y el acceso a indemnizaciones en caso de accidente.

3.4.3. Buenas prácticas en el mantenimiento y reparación

Para garantizar la seguridad y prolongar la vida útil de los equipos, se recomienda:

- Realizar limpiezas periódicas de los componentes internos, utilizando aire comprimido y evitando el uso de líquidos.
- Comprobar el estado de los cables y conectores, sustituyendo aquellos que presenten desgaste o daños.
- Documentar todas las intervenciones realizadas en los equipos, facilitando el seguimiento y la trazabilidad.
- Mantener actualizados los manuales de seguridad y formación del personal encargado de la manipulación y mantenimiento.

Estas normas y recomendaciones contribuyen a crear un entorno de trabajo seguro, eficiente y respetuoso con el medio ambiente, minimizando los riesgos asociados a la explotación de sistemas microinformáticos.

3.5. Redes informáticas

Las redes informáticas se pueden clasificar según su alcance geográfico, propósito y tecnología utilizada. Esta clasificación ayuda a entender las necesidades de cada entorno y a seleccionar la infraestructura adecuada.

3.5.1. LAN (Local Area Network)

Las LAN son redes de área local que conectan dispositivos en un espacio reducido, como una oficina, aula o edificio. Permiten compartir recursos (archivos, impresoras, acceso a Internet) y suelen utilizar cableado Ethernet y Wi-Fi. Las LAN ofrecen alta velocidad y baja latencia, facilitando la colaboración y el trabajo en grupo.

3.5.2. MAN (Metropolitan Area Network)

Las MAN cubren áreas metropolitanas, como ciudades o campus universitarios. Interconectan varias LAN mediante enlaces de alta velocidad, como fibra óptica o radioenlaces. Son gestionadas por empresas, instituciones o proveedores de servicios, y permiten compartir recursos entre diferentes ubicaciones cercanas.

3.5.3. WAN (Wide Area Network)

Las WAN abarcan grandes distancias, conectando redes locales y metropolitanas en diferentes ciudades, países o continentes. Utilizan infraestructuras públicas y privadas, como líneas telefónicas, satélites y enlaces dedicados. Internet es el ejemplo más conocido de WAN. Las WAN requieren protocolos y dispositivos especializados para gestionar la transmisión de datos y garantizar la seguridad.

3.5.4. PAN (Personal Area Network)

Las PAN conectan dispositivos personales en un área muy limitada, como alrededor de una persona. Utilizan tecnologías inalámbricas como Bluetooth, Zigbee o infrarrojos. Ejemplos de PAN son la conexión entre un teléfono móvil y unos auriculares inalámbricos, o entre un ordenador y un smartwatch.

3.5.5. Otras redes especializadas

Existen redes diseñadas para propósitos específicos:

- **SAN (Storage Area Network):** Red dedicada al almacenamiento de datos, utilizada en centros de datos y servidores.
- **CAN (Campus Area Network):** Red que conecta edificios dentro de un campus universitario o empresarial.
- **VPN (Virtual Private Network):** Red virtual que permite conectar dispositivos de forma segura a través de Internet, simulando una red privada.

Comparativa de redes

Cada tipo de red responde a necesidades diferentes en cuanto a cobertura, velocidad, coste y complejidad de gestión. La correcta elección y configuración de la red es fundamental para garantizar la eficiencia y la seguridad en la transmisión de datos.

Tipo	Alcance	Velocidad	Coste	Ejemplo
LAN	Local (edificio)	Alta	Bajo	Oficina, aula
MAN	Ciudad/campus	Media/alta	Medio	Universidad, empresa
WAN	Global	Variable	Alto	Internet, sucursales
PAN	Personal	Baja	Bajo	Bluetooth, wearable

Cuadro 3.2: Comparativa de los principales tipos de redes

3.6. Componentes de una red informática

Una red informática es un conjunto de dispositivos conectados entre sí para compartir información y recursos, como archivos, impresoras y acceso a Internet. El diseño y la implementación de una red requieren la integración de diversos componentes, cada uno con funciones específicas que permiten la comunicación eficiente y segura entre los usuarios y los sistemas. En esta sección se describen los elementos fundamentales que conforman una red informática, sus características y el papel que desempeñan en el funcionamiento global de la infraestructura de red.

3.6.1. Dispositivos finales

Los dispositivos finales son aquellos que utilizan los usuarios para interactuar con la red y acceder a los recursos compartidos. Incluyen ordenadores de escritorio, portátiles, impresoras, teléfonos inteligentes, tabletas y otros equipos conectados. Estos dispositivos pueden funcionar como emisores o receptores de información y suelen estar equipados con adaptadores de red para conectarse mediante cable o de forma inalámbrica. La correcta configuración y protección de los dispositivos finales es esencial para la seguridad y el rendimiento de la red.

3.6.2. Dispositivos de red

Los dispositivos de red gestionan el tráfico de datos y permiten la interconexión entre los distintos elementos de la red. Los principales son:

Routers Dirigen el tráfico entre diferentes redes, gestionando el acceso a Internet y la comunicación entre subredes.

Switches Conectan múltiples dispositivos dentro de una misma red local, optimizando el flujo de datos y evitando colisiones.



Fuente: Evan-Amos, Public domain, via Wikimedia Commons

Figura 3.5: Router WiFi

Puntos de acceso (Access Points) Permiten la conexión inalámbrica de dispositivos, ampliando la cobertura de la red Wi-Fi.

Firewalls Filtran el tráfico y protegen la red frente a accesos no autorizados y amenazas externas.

La elección y configuración de estos dispositivos determina la capacidad, seguridad y escalabilidad de la red.

3.6.3. Medios de transmisión

Los medios de transmisión son los canales físicos o inalámbricos por los que viajan los datos entre los dispositivos de la red. Los principales tipos son:

Cables de cobre Incluyen el par trenzado (UTP/STP) y el cable coaxial, utilizados en redes Ethernet y conexiones de televisión.

Fibra óptica Utiliza hilos de vidrio o plástico para transmitir datos mediante pulsos de luz, ofreciendo alta velocidad y gran alcance.

Ondas radioeléctricas Permiten la transmisión inalámbrica de datos mediante tecnologías como Wi-Fi, Bluetooth y Zigbee.

La elección del medio depende de la distancia, velocidad requerida, coste y entorno físico.

3.6.4. Software de gestión y seguridad

El software de gestión y seguridad permite controlar, supervisar y proteger la red informática. Incluye:

Sistemas operativos de red Gestionan los recursos y servicios compartidos, como archivos, impresoras y aplicaciones.

Herramientas de monitorización Permiten analizar el tráfico, detectar incidencias y optimizar el rendimiento.

Antivirus y sistemas de detección de intrusos (IDS) Protegen los dispositivos y la red frente a malware y ataques externos.

Software de configuración Facilita la administración de dispositivos de red, la gestión de usuarios y la aplicación de políticas de seguridad.

Una gestión adecuada del software es clave para mantener la integridad, disponibilidad y confidencialidad de los datos en la red.

3.7. Topologías de red

La topología de red se refiere a la disposición física o lógica de los dispositivos y conexiones en una red informática. La elección de una topología adecuada es crucial para optimizar el rendimiento, la escalabilidad y la fiabilidad de la red. A continuación se describen las principales topologías utilizadas en redes informáticas, junto con sus características, ventajas y desventajas.

3.7.1. Topología en bus

En la topología en bus, todos los dispositivos están conectados a un único cable central (bus) que actúa como canal de comunicación. Los datos viajan en ambas direcciones y cada dispositivo escucha el tráfico para identificar los mensajes dirigidos a él. Es sencilla y económica, pero si el cable principal falla, toda la red se ve afectada. Además, el rendimiento disminuye a medida que aumenta el número de dispositivos.

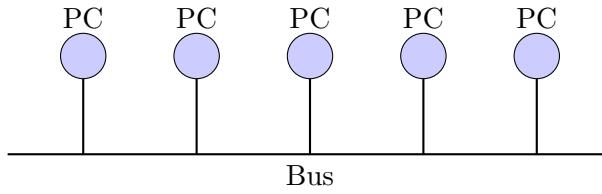


Figura 3.6: Ejemplo de topología en bus

3.7.2. Topología en estrella

En la topología en estrella, todos los dispositivos se conectan a un nodo central (normalmente un switch o hub). Este nodo gestiona el tráfico y facilita la administración de la red. Si un cable individual falla, solo se ve afectado el dispositivo correspondiente, pero si el nodo central falla, toda la red queda inoperativa. Es la topología más utilizada en redes LAN modernas por su facilidad de expansión y mantenimiento.

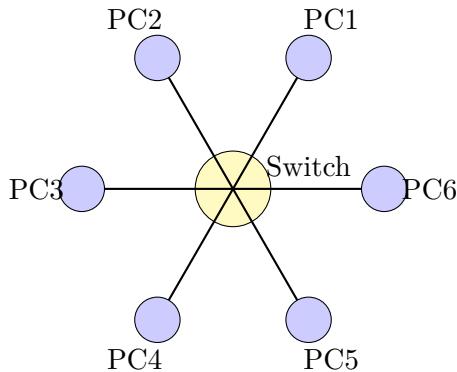


Figura 3.7: Ejemplo de topología en estrella

3.7.3. Topología en anillo

En la topología en anillo, cada dispositivo está conectado al siguiente formando un círculo cerrado. Los datos circulan en una dirección y pasan por cada dispositivo hasta llegar al destino. Es eficiente para redes pequeñas, pero si un dispositivo o enlace falla, puede interrumpir toda la comunicación. Algunas variantes incluyen mecanismos para tolerancia a fallos, como el doble anillo.

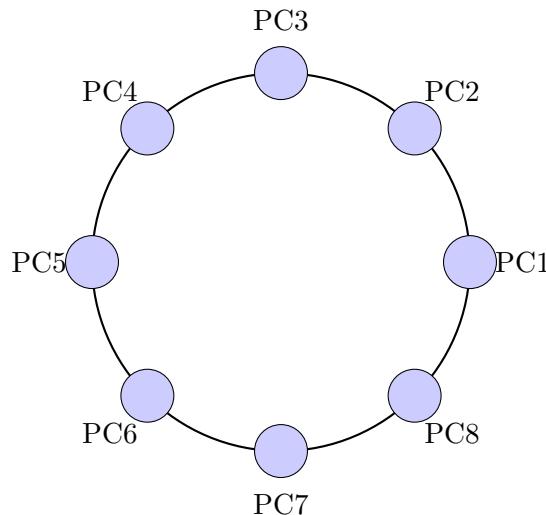


Figura 3.8: Ejemplo de topología en anillo

3.7.4. Topología en malla

La topología en malla conecta cada dispositivo con varios otros, creando múltiples rutas para los datos. Esto proporciona alta redundancia y tolerancia a fallos, ya que si un enlace falla, los datos pueden tomar rutas alternativas. Es común en redes críticas y de gran tamaño, aunque su coste y complejidad de instalación son elevados.

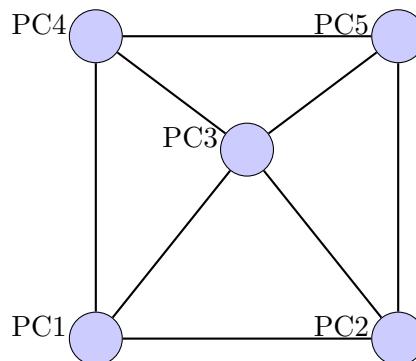


Figura 3.9: Ejemplo de topología en malla

3.7.5. Topología en árbol

La topología en árbol combina características de la estrella y el bus, formando una estructura jerárquica. Los dispositivos se agrupan en segmentos conectados

a nodos centrales, que a su vez se conectan a un nodo principal. Es escalable y facilita la segmentación de la red, pero depende de los nodos centrales para el funcionamiento global.

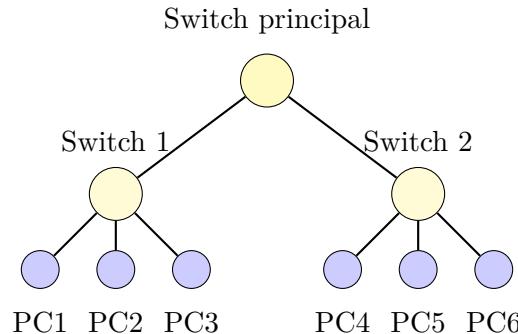


Figura 3.10: Ejemplo de topología en árbol

Importante

Saber elegir la topología apropiada es crucial para el rendimiento y la escalabilidad de la red. Además, es importante considerar el tipo de tráfico y las aplicaciones que se ejecutarán en la red.

3.8. Tipos de cableado y conectores

El cableado y los conectores son elementos fundamentales en la infraestructura de redes informáticas, ya que determinan la velocidad, fiabilidad y alcance de la comunicación entre dispositivos. A continuación se detallan los principales tipos de cables y conectores utilizados en redes, sus características y aplicaciones.

3.8.1. Par trenzado (UTP y STP)

El par trenzado es el tipo de cable más común en redes Ethernet. Consiste en pares de hilos de cobre trenzados entre sí para reducir interferencias electromagnéticas.

UTP (Unshielded Twisted Pair) No tiene apantallamiento, es económico y fácil de instalar. Se utiliza en la mayoría de redes domésticas y de oficina.

STP (Shielded Twisted Pair) Incluye una capa de apantallamiento que protege contra interferencias externas, ideal para entornos industriales o con alta interferencia eléctrica.



Fuente: www.heimnetzwerke.net, CC BY 4.0, via Wikimedia Commons

Figura 3.11: Kit de herramientas para cableado

Existen diferentes categorías de par trenzado (Cat 5e, Cat 6, Cat 6a, Cat 7, Cat 8), cada una con capacidades de velocidad y distancia específicas.

3.8.2. Cable coaxial

El cable coaxial está formado por un conductor central rodeado de un aislante, una malla metálica y una cubierta exterior. Se utilizó ampliamente en redes antiguas (Ethernet 10Base2 y 10Base5) y sigue siendo común en sistemas de televisión por cable y CCTV.

- **Ventajas:** Buena protección contra interferencias y capacidad para transmitir señales a largas distancias.
- **Desventajas:** Menor flexibilidad y velocidad comparado con el par trenzado y la fibra óptica.

3.8.3. Fibra óptica

La fibra óptica transmite datos mediante pulsos de luz a través de hilos de vidrio o plástico. Es la opción preferida para redes de alta velocidad y largas distancias.

- **Monomodo:** Utilizada en enlaces de larga distancia, como redes metropolitanas y troncales de Internet.

3 Explotación de sistemas microinformáticos

- **Multimodo:** Adecuada para distancias cortas, como redes dentro de edificios.

Ventajas: Alta velocidad, inmunidad a interferencias electromagnéticas y gran capacidad de transmisión.

3.8.4. Conectores comunes

Los conectores permiten la unión física entre cables y dispositivos de red. Los más utilizados son:

RJ45 Conector estándar para cables de par trenzado en redes Ethernet.

BNC Utilizado en cables coaxiales, especialmente en redes antiguas y sistemas de vídeo.

SC, LC, ST Conectores para fibra óptica, cada uno con características específicas de tamaño y mecanismo de conexión.

3.8.5. Criterios de elección

La selección del tipo de cable y conector depende de varios factores:

- **Velocidad de transmisión:** La fibra óptica y los cables de categoría alta permiten mayores velocidades.
- **Distancia:** La fibra óptica es ideal para largas distancias; el par trenzado para distancias cortas y medias.
- **Entorno:** En ambientes con alta interferencia eléctrica, se recomienda STP o fibra óptica.
- **Coste y facilidad de instalación:** El par trenzado UTP es el más económico y sencillo de instalar.

La correcta elección e instalación del cableado y los conectores garantiza el rendimiento y la fiabilidad de la red informática.

3.8.6. Creación de cables de red con conectores RJ45

Para crear un cable de red con conectores RJ45, se deben seguir los siguientes pasos:

1. **Preparar el cable:** Corte el cable de par trenzado (UTP o STP) a la longitud deseada utilizando una herramienta de corte.
2. **Pelar el extremo:** Retire aproximadamente 2 cm de la cubierta exterior del cable para exponer los pares de hilos.
3. **Ordenar los hilos:** Separe y ordene los hilos según el estándar de cableado elegido (T568A o T568B). El estándar más común es T568B. Fíjate en la figura 3.12.
4. **Cortar los hilos:** Igualar la longitud de los hilos y cortarlos para que queden alineados.
5. **Insertar en el conector RJ45:** Introduzca los hilos en el conector RJ45, asegurándose de que cada hilo llegue hasta el fondo y siga el orden correcto.
6. **Crimpar el conector:** Utilice una herramienta de crimpado para fijar el conector RJ45 al cable, asegurando una conexión firme. El resultado final debe parecerse a la figura 3.13.
7. **Comprobar la conexión:** Utilice un comprobador de cables de red para verificar que todos los hilos están correctamente conectados y no hay cortocircuitos.

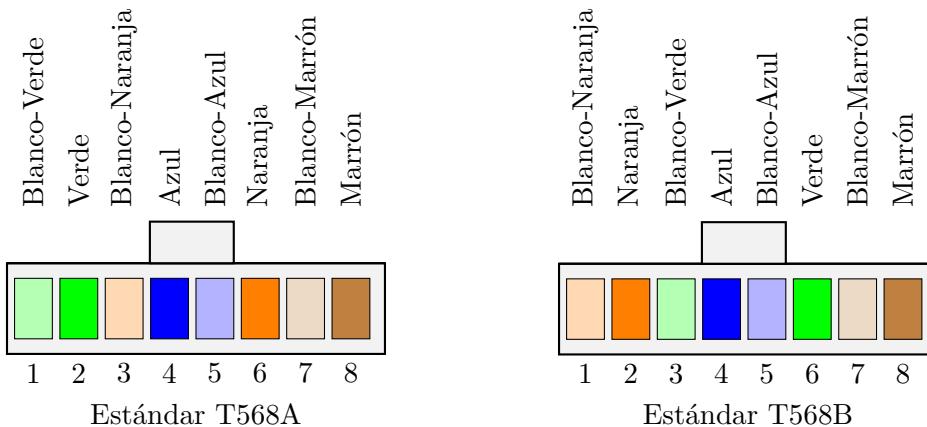


Figura 3.12: Distribución de hilos en un conector RJ45

Consejo: Es importante seguir el mismo estándar en ambos extremos del cable para garantizar la conectividad. Los cables directos se utilizan para conectar dispositivos diferentes (por ejemplo, PC a switch), mientras que los cables cruzados se emplean para conectar dispositivos iguales (por ejemplo, PC a PC).



Figura 3.13: Ejemplo de un cable que usa el estándar T568B

Ejercicio 3.2

Busca un cable de conexión RJ45, y observa los colores de los hilos en su interior. Compara lo que ves con los estándares T568A y T568B.



3.9. Mapa físico y lógico de una red local

El mapa físico representa la disposición real de los dispositivos y el cableado. El mapa lógico muestra la estructura de la red desde el punto de vista de la comunicación y los protocolos utilizados.

3.9.1. Ejemplo de mapa físico y lógico de una red local

Para comprender mejor la estructura y funcionamiento de una red local, es fundamental distinguir entre el mapa físico y el mapa lógico:

- **Mapa físico:** Representa la ubicación real de los dispositivos, el cableado, los puntos de acceso y los elementos de infraestructura (racks, armarios, canalizaciones). Incluye la disposición de los equipos en el espacio, la longitud y tipo de cables, y la posición de los conectores y paneles de parcheo.
- **Mapa lógico:** Muestra cómo se comunican los dispositivos entre sí, la segmentación en subredes, la asignación de direcciones IP, los protocolos utilizados y la relación jerárquica entre los elementos de la red (por ejemplo, qué dispositivos actúan como servidores, clientes, routers, switches, etc.).

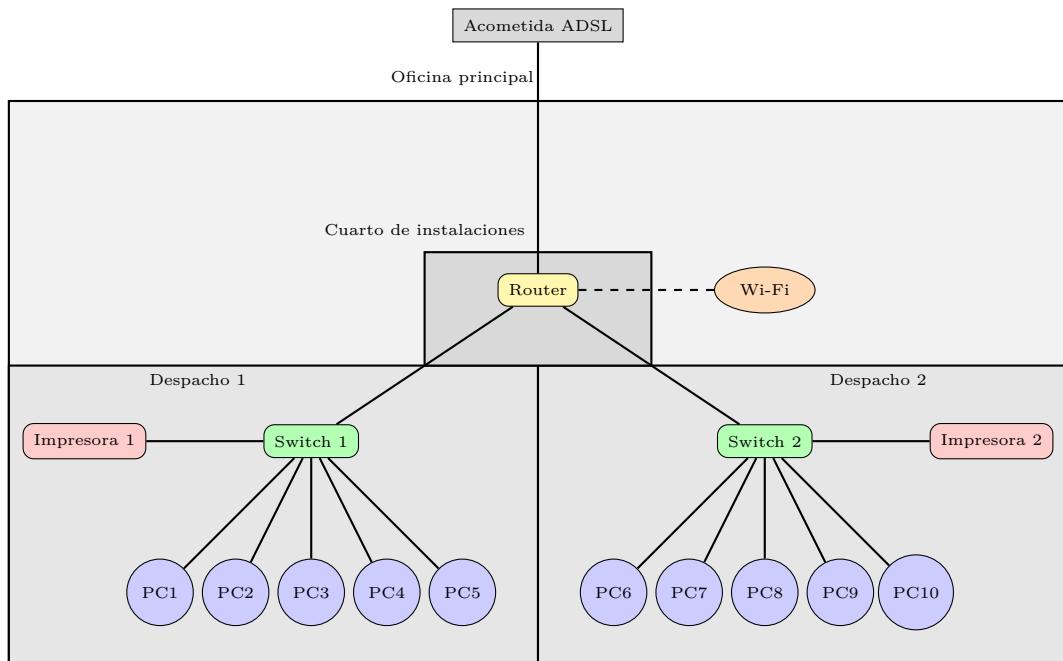
Ejemplo práctico

Supongamos una red local en una pequeña oficina con los siguientes elementos:

- 1 router de acceso a Internet.
- 2 switches para distribuir la conexión.
- 10 ordenadores de sobremesa.
- 2 impresoras de red.
- 1 punto de acceso Wi-Fi.
- Cableado estructurado con panel de parcheo y tomas RJ45 en cada puesto.

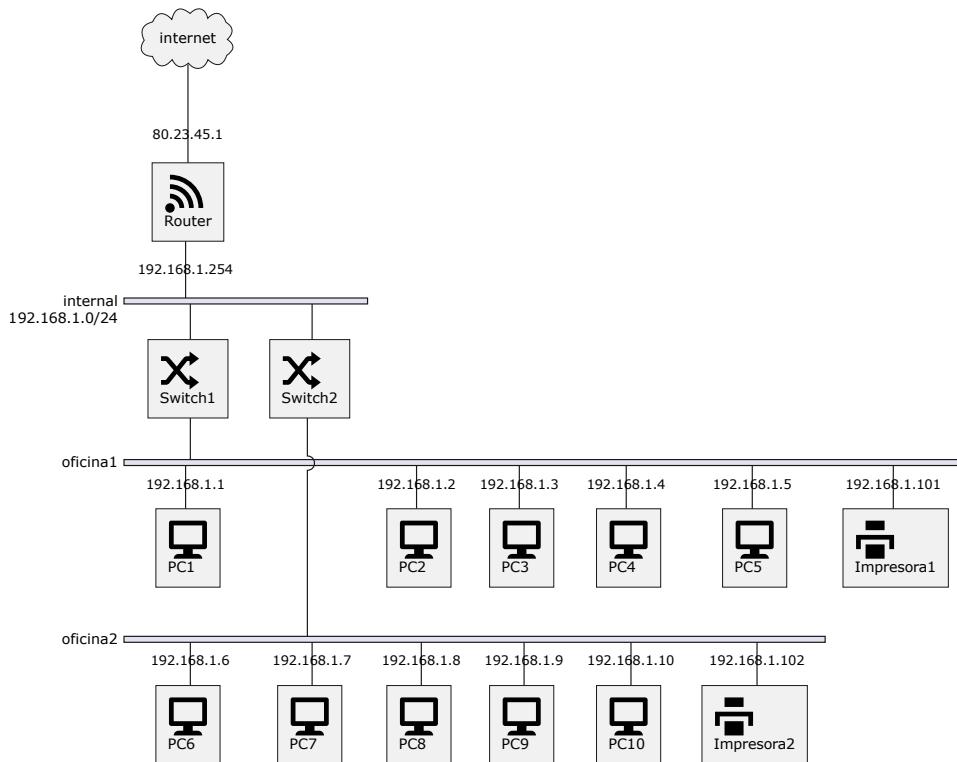
Mapa físico El mapa físico se dibuja sobre el plano de la oficina, indicando:

- La ubicación del armario de comunicaciones, donde se instala el router, los switches y el panel de parcheo.
- El recorrido de los cables de red desde el panel de parcheo hasta cada puesto de trabajo.
- La posición de los ordenadores, impresoras y el punto de acceso Wi-Fi.
- Las conexiones físicas entre los dispositivos (por ejemplo, qué puerto del switch conecta a cada ordenador).



Mapa lógico El mapa lógico se representa mediante un diagrama de red, mostrando:

- La segmentación en subredes (por ejemplo, una subred para los PCs y otra para impresoras).
- La asignación de direcciones IP (por ejemplo, rango 192.168.1.0/24 para los PCs, 192.168.1.100-109 para los ordenadores, 192.168.1.200-201 para las impresoras).
- El flujo de comunicación: los PCs acceden a Internet a través del router, comparten archivos entre sí y pueden imprimir en las impresoras de red.
- Los protocolos utilizados (por ejemplo, TCP/IP, DHCP para asignación automática de IPs, DNS para resolución de nombres).
- La existencia de VLANs si se segmenta la red por departamentos o funciones.



Interpretación y utilidad

- El mapa físico es esencial para tareas de instalación, mantenimiento y resolución de problemas, ya que permite localizar rápidamente los dispositivos y el cableado.
- El mapa lógico facilita la administración de la red, la configuración de servicios y la planificación de ampliaciones o cambios en la infraestructura.
- Ambos mapas deben mantenerse actualizados y documentados, especialmente en entornos profesionales, para garantizar la eficiencia y la seguridad de la red.

Importante

La elaboración de mapas físicos y lógicos es una práctica recomendada en cualquier proyecto de redes, ya que ayuda a identificar posibles cuellos de botella, optimizar el rendimiento y anticipar necesidades futuras.

Ejercicio 3.3



Crea un mapa físico, y otro lógico, de la red local en tu casa.
Es posible que dispongas de los siguientes elementos:

- Router
- Descodificador TV
- PCs
- Impresoras
- Acceso a Internet
- Dispositivos móviles (tabletas, teléfonos)

Resumen

En este capítulo se ha abordado la explotación de sistemas microinformáticos, analizando sus componentes principales, desde la placa base y el procesador hasta los dispositivos de almacenamiento y adaptadores. Se han descrito las normas de seguridad y prevención de riesgos laborales, fundamentales para la manipulación y el mantenimiento de equipos informáticos. Además, se han explicado los tipos de redes, sus componentes, topologías y el cableado necesario para su

3 Explotación de sistemas microinformáticos

implementación, así como la importancia de los mapas físicos y lógicos para la gestión eficiente de una red local. El conocimiento y la correcta aplicación de estos conceptos son esenciales para garantizar el funcionamiento seguro, eficiente y escalable de los sistemas microinformáticos y sus redes asociadas.

4

Instalación de sistemas operativos

La instalación de sistemas operativos es un proceso fundamental en la administración de equipos informáticos. Un sistema operativo (SO) es el software que gestiona los recursos del hardware y proporciona servicios a las aplicaciones y usuarios. Este capítulo aborda la evolución, clasificación, funciones, tipos y procedimientos de instalación de sistemas operativos, así como aspectos legales y técnicos relevantes.

4.1. Evolución histórica y clasificación

Los sistemas operativos han evolucionado desde los primeros sistemas monousuario y monotarea (como MS-DOS) hasta los actuales sistemas multiusuario y multitarea (como Linux, Windows y macOS).

Cuadro 4.1: Hitos históricos en la evolución de los sistemas operativos

Año	Hito histórico
1945	Se construye ENIAC, uno de los primeros ordenadores electrónicos, que ejecutaba programas sin sistema operativo.
1950	Aparecen los primeros sistemas de tarjetas perforadas para automatizar tareas en computadoras.
1956	IBM desarrolla el primer sistema de procesamiento por lotes para mainframes, permitiendo la ejecución secuencial de tareas.
1964	IBM lanza el sistema operativo OS/360, pionero en la gestión de recursos y multitarea en mainframes.
1969	Se crea UNIX en los laboratorios Bell, estableciendo las bases de los sistemas multiusuario y multitarea modernos.
1970	Se desarrolla CP/M, uno de los primeros sistemas operativos para microcomputadoras.
1973	UNIX se reescribe en lenguaje C, facilitando su portabilidad y expansión a diferentes plataformas.

Año	Hito histórico
1981	Microsoft lanza MS-DOS, popularizando los sistemas monousuario en computadoras personales.
1984	Apple presenta Macintosh System Software, pionero en interfaces gráficas de usuario (GUI) accesibles.
1985	Microsoft lanza Windows 1.0, introduciendo la interfaz gráfica en sistemas personales.
1991	Linus Torvalds publica la primera versión de Linux, impulsando el software libre y los sistemas operativos abiertos.
1993	Microsoft lanza Windows NT, introduciendo arquitectura de 32 bits y capacidades avanzadas de red y seguridad.
1995	Microsoft presenta Windows 95, integrando la multitarea y el soporte para redes domésticas.
2001	Aparece Windows XP, consolidando la interfaz gráfica y la estabilidad en sistemas de escritorio.
2007	Apple lanza iOS, revolucionando los sistemas operativos móviles y la interacción táctil.
2008	Google lanza Android, estableciendo un estándar abierto para dispositivos móviles.
2013	Docker populariza los contenedores, cambiando el paradigma de virtualización y despliegue de aplicaciones.
2015	Microsoft lanza Windows 10, unificando plataformas de escritorio, móvil y nube.
2020	Apple introduce macOS Big Sur, con transición a procesadores ARM y rediseño de la interfaz.
2021	Windows 11 se lanza con mejoras en la interfaz y soporte para aplicaciones Android.

4.1.1. Monousuario vs. multiusuario

Los sistemas monousuario están diseñados para ser utilizados por una sola persona a la vez, como MS-DOS. Los sistemas multiusuario permiten que varios usuarios accedan y utilicen los recursos del sistema simultáneamente, como Linux y UNIX, gestionando permisos y sesiones independientes.

4.1.2. Monotarea vs. multitarea

Un sistema monotarea solo puede ejecutar una tarea o proceso en cada momento, lo que limita la eficiencia y la productividad. Los sistemas multitarea pueden ejecutar varios procesos de manera concurrente, asignando recursos y

tiempo de CPU a cada uno, como ocurre en Windows, Linux y macOS.

4.1.3. Centralizados vs. distribuidos

Los sistemas operativos centralizados gestionan todos los recursos en un único equipo, facilitando la administración pero limitando la escalabilidad. Los sistemas distribuidos coordinan recursos y tareas entre varios equipos conectados en red, permitiendo compartir procesamiento, almacenamiento y servicios, como ocurre en clústeres y sistemas en la nube.

4.1.4. Interactivos vs. por lotes

Los sistemas interactivos permiten la comunicación directa entre el usuario y el sistema, respondiendo en tiempo real a comandos e instrucciones, como los sistemas con interfaz gráfica o de línea de comandos. Los sistemas por lotes procesan tareas agrupadas sin intervención directa del usuario, ejecutando trabajos de forma automática y secuencial, como en los primeros sistemas mainframe.

4.2. Funciones de un sistema operativo

Las funciones principales de un sistema operativo se pueden clasificar en los siguientes apartados.

4.2.1. Gestión de procesos

El sistema operativo controla la ejecución de los programas, gestionando los procesos activos en el sistema. Esto incluye la creación, planificación y finalización de procesos, la asignación de recursos como tiempo de CPU, y la coordinación entre procesos mediante mecanismos de comunicación y sincronización. La gestión eficiente de procesos permite la multitarea y la estabilidad del sistema.

4.2.2. Gestión de memoria

Administra la memoria RAM y la memoria virtual, asignando espacio a los procesos y garantizando que no se produzcan conflictos o accesos indebidos. El sistema operativo utiliza técnicas como la paginación y la segmentación para optimizar el uso de la memoria y proteger los datos de cada proceso, evitando que interfieran entre sí.

4.2.3. Gestión de archivos

Organiza, almacena y protege los datos en los dispositivos de almacenamiento. El sistema operativo proporciona sistemas de archivos que permiten crear, leer, modificar y eliminar archivos y carpetas, así como establecer permisos de acceso y mantener la integridad de la información.

4.2.4. Gestión de dispositivos

Controla el acceso y uso de los periféricos conectados al equipo, como discos duros, impresoras, tarjetas de red y otros dispositivos de entrada/salida. El sistema operativo utiliza controladores (drivers) para comunicarse con el hardware y garantizar su funcionamiento correcto y eficiente.

4.2.5. Interfaz de usuario

Proporciona entornos gráficos (GUI) o de línea de comandos (CLI) que permiten al usuario interactuar con el sistema. La interfaz facilita la ejecución de programas, la gestión de archivos y la configuración del sistema, adaptándose a diferentes niveles de experiencia y necesidades.

4.2.6. Seguridad y protección

Controla el acceso al sistema y protege la integridad de los datos y recursos. El sistema operativo implementa mecanismos de autenticación, autorización y cifrado, así como políticas de permisos y auditoría para prevenir accesos no autorizados y garantizar la confidencialidad y disponibilidad de la información.

4.3. Tipos de sistemas operativos

A continuación se describen los principales tipos de sistemas operativos según su uso y características.

4.3.1. Sistemas operativos de escritorio

Son aquellos diseñados para ser utilizados por usuarios finales en computadoras personales, portátiles y estaciones de trabajo. Ofrecen interfaces gráficas amigables, soporte para aplicaciones de oficina, multimedia y herramientas de productividad. Ejemplos destacados incluyen Windows, macOS y las diferentes

distribuciones de Linux. Estos sistemas priorizan la facilidad de uso, la compatibilidad con hardware diverso y la seguridad básica para el usuario doméstico y profesional.

4.3.2. Sistemas operativos de servidor

Están optimizados para gestionar recursos y servicios en redes, como servidores web, bases de datos, correo electrónico y virtualización. Suelen ofrecer mayor estabilidad, escalabilidad y herramientas de administración remota. Ejemplos: Windows Server, Ubuntu Server, Red Hat Enterprise Linux y CentOS. Estos sistemas soportan múltiples usuarios simultáneos, configuraciones avanzadas de seguridad y redundancia, y están diseñados para funcionar de manera continua y fiable.

4.3.3. Sistemas operativos empotrados

Se utilizan en dispositivos específicos como teléfonos móviles, electrodomésticos, automóviles, sistemas industriales y equipos médicos. Están adaptados para funcionar con recursos limitados y cumplir funciones concretas. Ejemplos: Android, iOS, y sistemas propietarios en dispositivos IoT. Los sistemas empotrados suelen ser altamente optimizados, con tiempos de arranque rápidos y bajo consumo energético, y pueden estar integrados en hardware dedicado.

4.3.4. Sistemas operativos de tiempo real

Están diseñados para aplicaciones donde la respuesta debe ser inmediata y predecible, como en sistemas industriales, robótica, telecomunicaciones y control de procesos. Ejemplos: QNX, VxWorks, FreeRTOS. Estos sistemas garantizan que las tareas críticas se ejecuten en un tiempo determinado, minimizando la latencia y asegurando la fiabilidad en entornos donde los retrasos pueden causar fallos graves o riesgos para la seguridad.

Ejercicio 4.1



Investiga los siguientes sistemas operativos, y determina qué tipo de sistema operativo son, y cómo se clasificarían. P.ej.: Windows 10 - Sistema operativo de escritorio, multiusuario, multitarea, centralizado, interactivo.

1. Arch Linux
2. Ubuntu
3. OS/2

4. FreeBSD
5. TempleOS
6. Wear OS
7. NetWare

4.4. Tipos de aplicaciones

Las aplicaciones informáticas se pueden clasificar según su función y el entorno en el que se utilizan.

4.4.1. Aplicaciones de sistema

Las aplicaciones de sistema son programas diseñados para facilitar la administración y el mantenimiento del sistema operativo y el hardware. Incluyen herramientas como utilidades de disco para gestionar particiones y realizar copias de seguridad, programas antivirus para proteger contra malware, gestores de tareas, herramientas de monitorización de recursos, y utilidades de configuración del sistema. Estas aplicaciones suelen requerir permisos elevados y son esenciales para el funcionamiento seguro y eficiente del equipo.

Ejemplos de aplicaciones de sistema

Algunos ejemplos comunes de aplicaciones de sistema incluyen:

Utilidades de disco Programas como GParted (Linux) o Disk Management (Windows) permiten crear, modificar y eliminar particiones, así como formatear discos duros y unidades extraíbles.

Antivirus y antimalware Herramientas como Windows Defender, Avast o ClamAV protegen el sistema contra virus, troyanos y otras amenazas.

Gestores de tareas Aplicaciones como el Administrador de tareas de Windows o el Monitor de sistema de Linux permiten supervisar y gestionar los procesos y el uso de recursos.

Herramientas de copia de seguridad Programas como Time Machine (macOS), Deja Dup (Linux) o Acronis True Image (Windows) facilitan la creación y restauración de copias de seguridad.

Utilidades de configuración Paneles de control, configuradores de red, y herramientas para la gestión de usuarios y permisos.

Actualizadores de sistema Aplicaciones que gestionan la descarga e instalación de actualizaciones de seguridad y mejoras del sistema operativo.

Estas aplicaciones suelen estar integradas en el sistema operativo o disponibles como software adicional, y son esenciales para mantener el equipo funcionando correctamente y seguro.

4.4.2. Aplicaciones de usuario

Las aplicaciones de usuario están orientadas a satisfacer las necesidades y actividades cotidianas de los usuarios finales. Ejemplos incluyen procesadores de texto (como Microsoft Word o LibreOffice Writer), navegadores web (Google Chrome, Mozilla Firefox), programas de diseño gráfico (Adobe Photoshop, GIMP), reproductores multimedia, y juegos. Estas aplicaciones permiten crear, editar y consumir contenido, y su variedad depende del entorno y los intereses del usuario.

Ejemplos de aplicaciones de usuario

Algunos ejemplos habituales de aplicaciones de usuario son:

Procesadores de texto Permiten crear y editar documentos, como Microsoft Word, LibreOffice Writer o Google Docs.

Hojas de cálculo Para gestionar datos numéricos y realizar cálculos, como Microsoft Excel, LibreOffice Calc o Google Sheets.

Navegadores web Facilitan el acceso a Internet, como Google Chrome, Mozilla Firefox, Microsoft Edge o Safari.

Reproductores multimedia Permiten visualizar vídeos y escuchar música, como VLC Media Player, Windows Media Player o Spotify.

Programas de diseño gráfico Para editar imágenes y gráficos, como Adobe Photoshop, GIMP o CorelDRAW.

Clientes de correo electrónico Gestionan mensajes y cuentas de correo, como Microsoft Outlook, Thunderbird o Mail (macOS).

Juegos Aplicaciones de entretenimiento, desde títulos sencillos hasta videojuegos avanzados.

Aplicaciones educativas Herramientas para el aprendizaje, como simuladores, diccionarios, plataformas de cursos en línea, etc.

Utilidades de productividad Calendarios, gestores de tareas, aplicaciones de notas y organización personal.

Estas aplicaciones suelen instalarse según las necesidades del usuario y pueden actualizarse o eliminarse de manera independiente al sistema operativo.

4.4.3. Aplicaciones de red

Las aplicaciones de red permiten la comunicación y colaboración entre equipos y usuarios a través de redes locales o Internet. Entre ellas se encuentran los clientes de correo electrónico (Outlook, Thunderbird), navegadores web, herramientas de mensajería instantánea, aplicaciones de videoconferencia (Zoom, Teams), y plataformas de colaboración en línea (Google Drive, Slack). Estas aplicaciones son fundamentales para el trabajo en equipo, el acceso remoto y la gestión de información compartida.

Ejemplos de aplicaciones de red

Algunos ejemplos habituales de aplicaciones de red incluyen:

Clientes de correo electrónico Programas como Microsoft Outlook, Mozilla Thunderbird y Apple Mail permiten enviar, recibir y gestionar correos electrónicos a través de Internet o redes locales.

Navegadores web Aplicaciones como Google Chrome, Mozilla Firefox, Microsoft Edge y Safari facilitan el acceso y la navegación por páginas web.

Mensajería instantánea Herramientas como WhatsApp Web, Telegram Desktop, Slack y Discord permiten la comunicación en tiempo real entre usuarios.

Videoconferencia Aplicaciones como Zoom, Microsoft Teams, Google Meet y Skype posibilitan reuniones virtuales con audio, vídeo y chat.

Plataformas de colaboración en línea Servicios como Google Drive, Dropbox, OneDrive y SharePoint permiten compartir y editar documentos de forma colaborativa.

FTP/SFTP Clientes como FileZilla y WinSCP facilitan la transferencia de archivos entre equipos a través de la red.

Herramientas de acceso remoto Programas como TeamViewer, AnyDesk y VNC permiten controlar equipos a distancia.

Aplicaciones de gestión de redes Utilidades como Wireshark (análisis de tráfico), PuTTY (conexión SSH/Telnet) y Remote Desktop (escritorio remoto).

Estas aplicaciones son esenciales para la comunicación, el trabajo colaborativo y la administración de sistemas en entornos conectados.

Ejercicio 4.2



Investiga y explica las diferencias entre una aplicación de sistema, una aplicación de usuario y una aplicación de red. Da un ejemplo de cada tipo y describe para qué se utiliza en la práctica.

4.5. Licencias y tipos de licencias

Las licencias de software son acuerdos legales que regulan el uso, distribución y modificación del software. Existen varios tipos de licencias, cada una con sus propias características y restricciones.

4.5.1. Propietarias

Las licencias propietarias son aquellas en las que el software es propiedad de una empresa o desarrollador, y su uso está restringido por términos legales. Normalmente requieren el pago de una licencia y no permiten modificar ni distribuir el código fuente. Los usuarios solo pueden utilizar el software según lo establecido en el contrato de licencia. Ejemplos de software propietario incluyen Microsoft Windows y macOS. Estas licencias suelen ofrecer soporte oficial y actualizaciones periódicas, pero limitan la personalización y el acceso al código.

Ejemplo de software propietario

Un ejemplo de software bajo licencia propietaria es **Microsoft Office**. Este paquete de aplicaciones de productividad (Word, Excel, PowerPoint, Outlook, etc.) requiere la compra de una licencia para su uso, no permite acceder ni modificar el código fuente, y su distribución está restringida por los términos establecidos por Microsoft.

4.5.2. Libres

Las licencias libres permiten a los usuarios utilizar, modificar y distribuir el software sin restricciones significativas. El código fuente está disponible y puede ser adaptado a las necesidades del usuario o la comunidad. Ejemplos de licencias libres son la GNU General Public License (GPL) y la licencia BSD. Sistemas operativos como Linux y FreeBSD se distribuyen bajo este tipo de licencias. El software libre fomenta la colaboración, la transparencia y la innovación, aunque el soporte puede depender de la comunidad.

Ejemplo de software libre

Un ejemplo de software bajo licencia libre es **LibreOffice**. Este paquete de oficina incluye procesador de textos, hoja de cálculo, presentaciones y más, y se distribuye bajo la licencia GNU LGPL, permitiendo su uso, modificación y redistribución libremente por cualquier usuario.

4.5.3. Open Source

El software de código abierto (Open Source) pone a disposición el código fuente, permitiendo su estudio y modificación. Sin embargo, puede incluir ciertas restricciones sobre la redistribución o el uso comercial. Las licencias open source más conocidas son la Apache License, MIT License y Mozilla Public License. Aunque muchos sistemas libres son también open source, no todos los proyectos open source permiten la redistribución sin condiciones. Este modelo facilita la auditoría de seguridad y la adaptación del software, manteniendo ciertos derechos para los autores.

Ejemplo de software open source

Un ejemplo de software bajo licencia open source es **Doom**. El código fuente original de este popular videojuego fue liberado por id Software bajo la licencia GNU GPL, permitiendo a la comunidad estudiar, modificar y crear nuevas versiones y adaptaciones del juego para diferentes plataformas.

4.5.4. Shareware/Freeware

El shareware es software que se distribuye de forma gratuita para su prueba, pero puede requerir pago tras un periodo de uso o para acceder a todas sus funciones. El freeware, por otro lado, es software completamente gratuito, aunque no necesariamente permite modificar o redistribuir el código fuente. Ejemplos de shareware incluyen WinRAR y algunos programas de edición de imágenes;

ejemplos de freeware son Skype y Adobe Acrobat Reader. Estas licencias permiten a los usuarios probar el software antes de comprarlo o utilizarlo sin coste, aunque suelen tener limitaciones en cuanto a soporte y actualizaciones.

Ejemplo de software shareware/freeware

Un ejemplo de software bajo licencia shareware es **WinRAR**. Este programa de compresión y descompresión de archivos permite su uso gratuito durante un periodo de prueba, tras el cual solicita la compra de una licencia para continuar utilizándolo legalmente. Por otro lado, un ejemplo de software freeware es **Skype**, que se puede descargar y utilizar gratuitamente para realizar videollamadas y mensajes, aunque no permite modificar ni redistribuir su código fuente.

Ejercicio 4.3



Investiga y explica las diferencias entre los siguientes tipos de licencias de software: GPL, MIT, Apache y BSD. Para cada una, indica si permiten modificar, redistribuir y usar el software con fines comerciales, y da un ejemplo de software conocido que use cada licencia.

4.6. Procedimiento de instalación

El procedimiento de instalación de un sistema operativo implica una serie de pasos que pueden variar según el tipo de sistema y el entorno (físico o virtual). A continuación se describen las fases generales:

1. **Preparación:** Reunir los requisitos de hardware y software, obtener la imagen de instalación (ISO, DVD, USB), y realizar copias de seguridad de los datos importantes.
2. **Configuración del medio de instalación:** Crear un medio de arranque (USB, DVD) y configurar la BIOS/UEFI para iniciar desde dicho medio.
3. **Inicio del instalador:** Arrancar el equipo desde el medio de instalación y seleccionar las opciones iniciales (idioma, teclado, zona horaria).
4. **Particionado y selección de disco:** Elegir el disco de destino, crear o modificar particiones según las necesidades (sistema, datos, recuperación).
5. **Copia de archivos y configuración básica:** El instalador copia los archivos del sistema operativo y solicita información básica (usuario, contraseña, nombre del equipo).

6. **Instalación de controladores y actualizaciones:** Se instalan los controladores necesarios para el hardware y se descargan actualizaciones recomendadas.
7. **Configuración final:** Ajustar preferencias regionales, privacidad, red y cuentas de usuario.
8. **Verificación y pruebas:** Comprobar que el sistema funciona correctamente, instalar aplicaciones esenciales y restaurar datos si es necesario.

Cada sistema operativo puede tener asistentes y herramientas específicas, pero estos pasos proporcionan una guía general aplicable a la mayoría de instalaciones. A continuación, se detallan los procedimientos para instalar tres sistemas operativos populares: Windows 11, Ubuntu y macOS.

Importante

Se recomienda el uso de máquinas virtuales, como VirtualBox o VMware, para probar sistemas operativos sin afectar el sistema principal. Esto permite realizar prácticas y pruebas de instalación de forma segura y reversible.

4.6.1. Instalación de Windows 11

Los pasos principales para instalar Windows 11 son:

1. Verificar los requisitos mínimos de hardware (procesador compatible, TPM 2.0, 4 GB RAM, 64 GB almacenamiento).
2. Descargar la imagen ISO oficial desde el sitio de Microsoft o crear un USB de instalación con la herramienta Media Creation Tool.
3. Configurar el arranque del equipo desde el USB en la BIOS/UEFI.
4. Iniciar el asistente de instalación, seleccionar idioma y edición.
5. Introducir la clave de producto (puede omitirse temporalmente).
6. Elegir el tipo de instalación (actualización o personalizada).
7. Particionar el disco si es necesario y seleccionar la unidad de destino.
8. Esperar la copia de archivos y reinicios automáticos.
9. Configurar la cuenta de usuario, opciones de privacidad y conectividad.
10. Instalar actualizaciones y controladores recomendados.

4.6.2. Instalación de Ubuntu

Los pasos principales para instalar Ubuntu son:

1. Descargar la imagen ISO oficial desde el sitio de Ubuntu.
2. Crear un USB de arranque con herramientas como Rufus o balenaEtcher.
3. Configurar el arranque del equipo desde el USB en la BIOS/UEFI.
4. Seleccionar «Instalar Ubuntu» en el menú de inicio.
5. Elegir idioma y preferencias regionales.
6. Seleccionar el tipo de instalación (instalación limpia, junto a otro SO, o personalizada).
7. Particionar el disco según las necesidades (automático o manual).
8. Configurar usuario, contraseña y nombre del equipo.
9. Esperar la copia de archivos y reinicios automáticos.
10. Instalar actualizaciones y software adicional recomendado.

4.6.3. Instalación de macOS

Los pasos principales para instalar macOS (ejemplo: macOS Ventura) son:

1. Verificar compatibilidad del equipo y realizar copia de seguridad con Time Machine.
2. Descargar el instalador desde la App Store o usar el modo de recuperación (Comando + R al arrancar).
3. Si es una instalación limpia, borrar y formatear el disco con la Utilidad de Discos.
4. Ejecutar el instalador y seguir las instrucciones en pantalla.
5. Seleccionar el disco de destino y confirmar la instalación.
6. Esperar la copia de archivos y reinicios automáticos.
7. Configurar la cuenta de usuario, Apple ID y preferencias regionales.
8. Instalar actualizaciones y restaurar datos si es necesario.

4.7. Gestores de arranque

El gestor de arranque (bootloader) permite seleccionar el sistema operativo al iniciar el equipo. Algunos gestores populares son GRUB (Linux), BCD/Boot Manager (Windows) y rEFInd (macOS y sistemas EFI). A continuación se describen sus características, configuración y métodos de reparación.

4.7.1. GRUB (Linux)

GRUB (Grand Unified Bootloader) es el gestor de arranque más utilizado en sistemas Linux. Permite seleccionar entre varios sistemas operativos y configuraciones al iniciar el equipo. Sus principales características son la flexibilidad, soporte para múltiples sistemas de archivos y facilidad de personalización.

Configuración: El archivo principal de configuración es `/etc/default/grub` y los scripts en `/etc/grub.d/`. Para aplicar cambios, se ejecuta `sudo update-grub`.

Reparación: Si GRUB falla, puede restaurarse arrancando desde un Live CD/USB y ejecutando:

```
1 sudo grub-install /dev/sda
2 sudo update-grub
```

4.7.2. BCD/Boot Manager (Windows)

BCD (Boot Configuration Data) es el sistema de gestión de arranque moderno en Windows (desde Vista en adelante), reemplazando a NTLDR. Permite gestionar opciones de arranque, sistemas instalados y parámetros avanzados.

Configuración: Se utiliza la herramienta `bcdedit` desde la línea de comandos para modificar entradas, añadir sistemas o cambiar opciones.

Reparación: Ante fallos de arranque, se puede usar el entorno de recuperación y ejecutar:

```
1 bootrec /fixmbr
2 bootrec /fixboot
3 bootrec /rebuildbcd
```

4.7.3. rEFInd (macOS y sistemas EFI)

rEFInd es un gestor de arranque para sistemas con firmware UEFI, muy usado en equipos Apple y en instalaciones duales con Linux. Ofrece una interfaz gráfica y detección automática de sistemas instalados.

Configuración: El archivo de configuración principal es `refind.conf`, donde se pueden personalizar iconos, entradas y parámetros de arranque.

Reparación: Si rEFInd no aparece, puede reinstalarse desde un USB o desde macOS con:

```
1| /Volumes/refind-install.sh
```

4.7.4. LILO (Linux)

LILO (Linux Loader) fue uno de los primeros gestores de arranque para Linux. Aunque está en desuso, aún se encuentra en sistemas antiguos. Es sencillo y rápido, pero menos flexible que GRUB.

Configuración: El archivo principal es `/etc/lilo.conf`. Tras modificarlo, se debe ejecutar `sudo lilo` para aplicar los cambios.

Reparación: Si LILO falla, puede reinstalarse desde un Live CD con:

```
1| sudo lilo -M /dev/sda
```

4.7.5. autoexec.bat y config.sys (MS-DOS y Windows 9x)

Estos archivos eran fundamentales en los sistemas MS-DOS y Windows 9x para la configuración y el arranque del sistema.

autoexec.bat: Es un archivo de procesamiento por lotes que se ejecuta automáticamente al iniciar el sistema. Permite definir variables de entorno, ejecutar programas y cargar controladores residentes. Ejemplo de contenido:

```
1| @echo off
2| SET PATH=C:\DOS;C:\WINDOWS
3| LH SMARTDRV.EXE
4| PROMPT $P$G
```

config.sys: Archivo de configuración que se procesa antes de `autoexec.bat`. Permite cargar controladores de dispositivos y definir parámetros de memoria y sistema. Ejemplo de contenido:

```
1| DEVICE=C:\DOS\HIMEM.SYS
2| DEVICE=C:\DOS\EMM386.EXE RAM
3| FILES=40
4| BUFFERS=20
```

Configuración y reparación: Para modificar el comportamiento del arranque, se editan estos archivos con un editor de texto. Si el sistema no arranca correctamente, se puede iniciar en modo seguro (presionando F8) y corregir errores en estos archivos. También es posible restaurar versiones anteriores desde un disco de inicio.

Estos métodos han sido reemplazados por sistemas más avanzados en versiones modernas de Windows, pero siguen siendo relevantes en sistemas antiguos y en la comprensión de la evolución de los gestores de arranque.

4.8. Tecnologías de virtualización. Tipos

La virtualización permite ejecutar varios sistemas operativos en una misma máquina física. Tipos:

- **Virtualización completa:** El SO invitado no sabe que está virtualizado (VMware, VirtualBox).
- **Paravirtualización:** El SO invitado es consciente y optimizado para la virtualización (Xen).
- **Contenedores:** Ejecutan aplicaciones aisladas (Docker, LXC).

Ventajas: ahorro de costes, flexibilidad, pruebas y desarrollo seguro.

4.8.1. Virtualización completa

La virtualización completa permite ejecutar sistemas operativos invitados sin que estos sean conscientes de que están funcionando en un entorno virtual. El hipervisor emula todo el hardware necesario, permitiendo que el sistema operativo invitado funcione como si estuviera instalado en una máquina física real. Ejemplos de software que implementan virtualización completa son VMware Workstation, Oracle VirtualBox y Microsoft Hyper-V. Esta tecnología es ideal para pruebas, desarrollo, consolidación de servidores y aislamiento de entornos, aunque puede requerir más recursos de hardware debido a la emulación.

Ejercicio 4.4



Instala Ubuntu en una máquina virtual usando VirtualBox. Documenta los pasos realizados, incluyendo la configuración de la máquina virtual (memoria, disco, red), el proceso de instalación y la comprobación de funcionamiento del sistema operativo instalado. Adjunta capturas de pantalla de cada fase y explica cualquier incidencia encontrada y cómo la resolviste.

4.8.2. Paravirtualización

En la paravirtualización, el sistema operativo invitado está modificado para ser consciente de que se ejecuta en un entorno virtual. El hipervisor y el sistema

operativo colaboran para optimizar el acceso a los recursos, lo que reduce la sobrecarga y mejora el rendimiento respecto a la virtualización completa. Xen es uno de los hipervisores más conocidos que utiliza paravirtualización. Esta técnica es útil en entornos donde se requiere alto rendimiento y los sistemas operativos pueden ser adaptados para trabajar con el hipervisor.

Ejercicio 4.5



Investiga qué es el Subsistema de Windows para Linux (WSL). Explica sus ventajas y limitaciones respecto a una máquina virtual tradicional. Instala una distribución Linux usando WSL en Windows 10 o 11, documenta los pasos realizados y muestra ejemplos de comandos ejecutados en el entorno Linux. Comenta para qué tipo de tareas resulta útil WSL y cuándo sería preferible usar una máquina virtual completa.

4.8.3. Contenedores

Los contenedores permiten ejecutar aplicaciones y servicios de forma aislada, compartiendo el mismo núcleo del sistema operativo pero manteniendo entornos independientes. A diferencia de la virtualización tradicional, los contenedores no requieren un sistema operativo completo por cada instancia, lo que los hace más ligeros y eficientes. Docker y LXC son ejemplos populares de tecnologías de contenedores. Los contenedores son ideales para despliegue rápido de aplicaciones, escalabilidad, microservicios y desarrollo continuo, facilitando la portabilidad entre diferentes entornos.

Ejercicio 4.6



Realiza el siguiente ejercicio práctico sobre contenedores Docker:

1. Instala Docker en tu sistema (consulta la documentación oficial para tu sistema operativo).
2. Descarga y ejecuta una imagen de Linux (por ejemplo, `ubuntu`) en modo interactivo con el siguiente comando:

```
1| docker run -it ubuntu /bin/bash
```
3. Una vez dentro del contenedor, prueba los siguientes comandos y anota el resultado:
 - `ls` (listar archivos y carpetas)
 - `pwd` (mostrar el directorio actual)

- `cat /etc/os-release` (ver información del sistema operativo)
 - `apt update` (actualizar la lista de paquetes)
 - `whoami` (mostrar el usuario actual)
4. Sal del contenedor con el comando `exit`.
 5. Explica brevemente qué ventajas ofrece el uso de contenedores frente a máquinas virtuales tradicionales.

4.8.4. Ventajas de la virtualización

La virtualización ofrece múltiples ventajas en la administración de sistemas informáticos:

- **Ahorro de costes:** Permite consolidar varios servidores virtuales en una sola máquina física, reduciendo el gasto en hardware y energía.
- **Flexibilidad:** Facilita la creación, modificación y eliminación de máquinas virtuales según las necesidades del usuario o la empresa.
- **Pruebas y desarrollo seguro:** Permite probar sistemas operativos, aplicaciones y configuraciones en entornos aislados, sin afectar el sistema principal.
- **Recuperación ante fallos:** Las máquinas virtuales pueden ser respaldadas y restauradas fácilmente, mejorando la disponibilidad y la recuperación ante desastres.
- **Escalabilidad:** Es posible aumentar o disminuir los recursos asignados a cada máquina virtual de forma dinámica.

4.9. Consideraciones previas a la instalación de sistemas operativos

Antes de instalar un sistema operativo, es importante analizar las necesidades del usuario, el tipo de licencia, la compatibilidad del hardware y la disponibilidad de soporte y actualizaciones, para asegurar una instalación exitosa y adecuada al entorno.

4.9.1. Compatibilidad de hardware y software

Antes de instalar un sistema operativo, es esencial comprobar que el hardware del equipo cumple con los requisitos mínimos y recomendados del sistema. Esto incluye procesador, memoria RAM, espacio en disco, tarjeta gráfica y periféricos. Además, se debe verificar la compatibilidad de los controladores y la existencia de versiones adecuadas para el sistema operativo elegido. En el caso de software, es importante asegurarse de que las aplicaciones necesarias funcionarán correctamente en el nuevo entorno.

4.9.2. Copias de seguridad

Realizar copias de seguridad previas protege los datos ante posibles errores durante la instalación. Se recomienda respaldar documentos, configuraciones y cualquier información importante en dispositivos externos o servicios en la nube. Así, si ocurre algún problema, se podrá restaurar la información sin pérdidas.

4.9.3. Tipo de licencia y versión

La elección entre sistemas operativos libres y propietarios implica considerar el tipo de licencia, las restricciones legales y el soporte disponible. Es importante seleccionar la versión adecuada según el uso previsto (personal, profesional, servidor) y asegurarse de contar con una licencia válida para evitar problemas legales y recibir actualizaciones y soporte.

4.9.4. Particionado y gestión de datos

Planificar el particionado del disco permite organizar el almacenamiento, separar el sistema de los datos y facilitar futuras actualizaciones o reinstalaciones. Es recomendable decidir si se instalará el sistema junto a otros sistemas operativos (dual boot), crear particiones para recuperación y definir el sistema de archivos más adecuado.

4.9.5. Consulta de documentación y soporte

Antes de comenzar la instalación, consultar la documentación oficial y los foros de soporte ayuda a resolver dudas y anticipar posibles incidencias. La información proporcionada por los desarrolladores y la comunidad puede incluir guías paso a paso, soluciones a problemas frecuentes y recomendaciones específicas para cada sistema operativo y hardware.

4.10. Instalación de sistemas operativos

La instalación de sistemas operativos, ya sean libres o propietarios, requiere una planificación cuidadosa y la consideración de varios aspectos clave para asegurar una implementación exitosa y adecuada a las necesidades del usuario.

4.10.1. Requisitos

Antes de instalar un sistema operativo, es imprescindible verificar los requisitos mínimos y recomendados de hardware y software. Esto incluye procesador, memoria RAM, espacio en disco, tarjeta gráfica y compatibilidad de periféricos. Además, se debe comprobar que el equipo soporta la versión del sistema operativo elegida y que existen controladores disponibles para todos los componentes. En sistemas virtualizados, también es importante asignar recursos suficientes a la máquina virtual.

4.10.2. Versiones

La elección de la versión del sistema operativo depende del uso previsto (personal, profesional, servidor, educativo). Las versiones pueden diferir en funcionalidades, soporte, estabilidad y requisitos. Por ejemplo, Windows ofrece ediciones Home, Pro y Enterprise; Ubuntu tiene versiones Desktop y Server; macOS publica actualizaciones periódicas con nuevas características. Es recomendable consultar la documentación oficial para seleccionar la versión más adecuada y asegurarse de que recibirá soporte y actualizaciones.

4.10.3. Licencias

El tipo de licencia determina cómo se puede usar, modificar y distribuir el sistema operativo. Las licencias libres (GPL, BSD, MIT) permiten modificar y redistribuir el software, mientras que las propietarias (Windows, macOS) imponen restricciones legales y suelen requerir pago. Es fundamental leer y aceptar los términos de la licencia antes de la instalación, y asegurarse de contar con una licencia válida para evitar problemas legales y recibir soporte oficial. En entornos empresariales, la gestión de licencias es clave para cumplir con la normativa y evitar sanciones.

4.11. Instalación y desinstalación de aplicaciones

La instalación y desinstalación de aplicaciones es una tarea esencial para adaptar el sistema operativo a las necesidades del usuario, permitiendo añadir nuevas

funcionalidades o eliminar software innecesario de forma sencilla y segura.

4.11.1. Instalación de aplicaciones en sistemas operativos

La instalación de aplicaciones varía según el sistema operativo y el tipo de licencia del software. A continuación se detallan los procedimientos más habituales en sistemas operativos libres (Linux) y propietarios (Windows, macOS), así como consideraciones sobre requisitos, versiones y licencias.

Instalación en sistemas operativos libres (Linux)

En Linux, la instalación de aplicaciones suele realizarse mediante gestores de paquetes, que automatizan la descarga, instalación y actualización de software. Los gestores más comunes son `apt` (Debian, Ubuntu), `dnf / yum` (Fedora, CentOS, RHEL), y `pacman` (Arch Linux).

- **Repositorios oficiales:** Los sistemas Linux disponen de repositorios oficiales que contienen aplicaciones verificadas y actualizadas. Para instalar una aplicación, basta con ejecutar el comando correspondiente, por ejemplo:

```
1| sudo apt update
2| sudo apt install libreoffice
```

- **Repositorios de terceros y PPAs:** Algunas aplicaciones no están en los repositorios oficiales y requieren añadir repositorios externos (PPAs en Ubuntu) o descargar paquetes manualmente (`.deb`, `.rpm`).
 - **Instalación manual:** En ocasiones, es necesario descargar el código fuente y compilar la aplicación. Esto requiere instalar dependencias y seguir las instrucciones del desarrollador.
 - **Gestores universales:** Herramientas como `snap`, `flatpak` y `AppImage` permiten instalar aplicaciones de forma universal en distintas distribuciones.
 - **Desinstalación:** Se realiza con el mismo gestor de paquetes, por ejemplo:
- ```
1| sudo apt remove libreoffice
```
- **Licencias:** La mayoría de aplicaciones en Linux son libres o de código abierto, pero es importante revisar la licencia antes de instalar software de terceros.

## Instalación en sistemas operativos propietarios (Windows, macOS)

En sistemas propietarios, la instalación de aplicaciones suele realizarse mediante asistentes gráficos o instaladores descargados desde sitios oficiales.

### ■ Windows:

- **Instaladores ejecutables:** La mayoría de aplicaciones se distribuyen como archivos `.exe` o `.msi`. El usuario debe ejecutar el instalador y seguir los pasos del asistente.
- **Microsoft Store:** Permite instalar aplicaciones de forma segura y automática.
- **Gestores de paquetes:** Herramientas como `winget`, `chocolatey` y `scoop` permiten instalar aplicaciones desde la línea de comandos.
- **Desinstalación:** Se realiza desde el Panel de control, Configuración o mediante el gestor de paquetes.
- **Licencias:** Es fundamental aceptar los términos de la licencia durante la instalación. El software puede ser propietario, shareware, freeware o open source.

### ■ macOS:

- **Archivos .dmg:** Las aplicaciones suelen distribuirse como imágenes de disco. El usuario debe montar el archivo y arrastrar la aplicación a la carpeta **Aplicaciones**.
- **Mac App Store:** Permite instalar y actualizar aplicaciones de forma centralizada.
- **Gestores de paquetes:** Herramientas como Homebrew facilitan la instalación de software desde la terminal.
- **Desinstalación:** Basta con mover la aplicación a la papelera, aunque algunas requieren herramientas específicas para eliminar todos los componentes.
- **Licencias:** Es importante revisar los términos de uso, especialmente en aplicaciones comerciales o de pago.

## Requisitos, versiones y licencias

Antes de instalar cualquier aplicación, se deben considerar los siguientes aspectos:

- **Compatibilidad:** Verificar que la aplicación es compatible con la versión del sistema operativo y la arquitectura (32/64 bits).

- **Requisitos mínimos:** Comprobar los requisitos de hardware (memoria RAM, espacio en disco, procesador, tarjeta gráfica) y software (dependencias, librerías).
- **Versiones:** Seleccionar la versión adecuada según las necesidades (estable, beta, portable, profesional).
- **Licencia:** Leer y aceptar los términos de la licencia. Determinar si el software es libre, propietario, shareware, freeware u open source.
- **Fuente de descarga:** Descargar siempre desde sitios oficiales o repositorios verificados para evitar malware y garantizar actualizaciones.

### **Ejemplo práctico: Instalación de una aplicación en Linux y Windows**

**Linux (Ubuntu):** Para instalar el navegador Firefox:

```
1 | sudo apt update
2 | sudo apt install firefox
```

**Windows:** Para instalar VLC Media Player:

1. Descargar el instalador desde el sitio oficial (<https://www.videolan.org/vlc/>).
2. Ejecutar el archivo `vlc-setup.exe`.
3. Seguir los pasos del asistente y aceptar la licencia.
4. Finalizar la instalación y ejecutar la aplicación.

### **Desinstalación y actualización**

La desinstalación de aplicaciones debe realizarse mediante los métodos recomendados por el sistema operativo o el desarrollador para evitar dejar archivos residuales. La actualización de aplicaciones es fundamental para mantener la seguridad y el rendimiento; puede hacerse de forma automática, manual o mediante gestores de paquetes.

### **Documentación y registro**

Es recomendable documentar las aplicaciones instaladas, sus versiones y licencias, especialmente en entornos profesionales o educativos, para facilitar la gestión, el soporte y el cumplimiento legal.

### Ejercicio 4.7



Instala la aplicación GIMP (editor de imágenes) en Ubuntu utilizando el gestor de paquetes `apt`. Documenta los pasos realizados, incluyendo los comandos ejecutados, la comprobación de la instalación y una breve descripción de para qué sirve GIMP. Adjunta una captura de pantalla mostrando GIMP abierto en tu sistema.

## 4.12. Actualización y recuperación de sistemas operativos y aplicaciones

La actualización y recuperación de sistemas operativos y aplicaciones son procesos esenciales para mantener la seguridad, estabilidad y funcionalidad de los equipos informáticos, permitiendo corregir errores, proteger frente a amenazas y restaurar el sistema ante incidencias.

### 4.12.1. Actualización de sistemas operativos

La actualización de sistemas operativos es un proceso esencial para mantener la seguridad, estabilidad y compatibilidad del equipo. Las actualizaciones pueden incluir parches de seguridad, correcciones de errores, mejoras de rendimiento y nuevas funcionalidades. Existen diferentes métodos de actualización según el sistema operativo:

**Actualización automática** Muchos sistemas operativos permiten configurar la descarga e instalación automática de actualizaciones. Por ejemplo, Windows Update en Windows, Software Updater en Ubuntu y Actualización de software en macOS. Esta opción garantiza que el sistema esté protegido frente a vulnerabilidades conocidas y que se reciban mejoras sin intervención manual.

**Actualización manual** El usuario puede buscar e instalar actualizaciones manualmente desde las herramientas del sistema. Esto es útil para controlar cuándo se aplican los cambios, especialmente en entornos donde la estabilidad es crítica.

**Actualización incremental** Consiste en instalar solo los cambios recientes, minimizando el impacto en el sistema y reduciendo el tiempo de actualización.

**Actualización mayor (upgrade)** Implica pasar de una versión principal a otra (por ejemplo, de Windows 10 a Windows 11, o de Ubuntu 22.04 a 24.04). Este proceso puede requerir más tiempo y planificación, incluyendo copias de seguridad y comprobación de compatibilidad.

### Procedimiento de actualización en diferentes sistemas

#### Windows:

1. Acceder a Configuración → Actualización y seguridad → Windows Update.
2. Buscar actualizaciones y seleccionar las que se desean instalar.
3. Reiniciar el equipo si es necesario para completar el proceso.
4. En actualizaciones mayores, seguir el asistente de actualización y comprobar la compatibilidad de hardware y software.

#### Linux (Ubuntu):

1. Ejecutar en la terminal:

```
1| sudo apt update
2| sudo apt upgrade
```

2. Para actualizar a una nueva versión:

```
1| sudo do-release-upgrade
```

3. Revisar los cambios y reiniciar si es necesario.

#### macOS:

1. Acceder a Preferencias del sistema → Actualización de software.
2. Descargar e instalar las actualizaciones disponibles.
3. Para actualizaciones mayores, seguir el asistente y realizar copia de seguridad previa con Time Machine.

## 4.12.2. Actualización de aplicaciones

Las aplicaciones también requieren actualizaciones periódicas para corregir errores, mejorar la seguridad y añadir nuevas funciones. Los métodos más comunes son:

- **Gestores de paquetes (Linux):** Permiten actualizar todas las aplicaciones instaladas con un solo comando, por ejemplo:

```
1 | sudo apt update
2 | sudo apt upgrade
```

- **Microsoft Store y Mac App Store:** Facilitan la actualización automática o manual de aplicaciones instaladas desde la tienda oficial.
- **Actualizadores integrados:** Muchas aplicaciones incluyen su propio sistema de actualización, que notifica al usuario cuando hay nuevas versiones disponibles.
- **Descarga manual:** En algunos casos, es necesario descargar la nueva versión desde el sitio web oficial y realizar la instalación manualmente.

## 4.12.3. Recuperación de sistemas operativos

La recuperación de sistemas operativos es fundamental ante fallos, corrupción de archivos o problemas de arranque. Los métodos más utilizados son:

- **Restauración del sistema:** Permite volver a un punto anterior en el tiempo, restaurando la configuración y los archivos del sistema sin afectar los documentos personales. En Windows, se accede desde Panel de control → Recuperación. En macOS, se utiliza Time Machine.
- **Recuperación desde discos de rescate:** Los discos de rescate o Live CDs permiten arrancar el equipo desde un medio externo y reparar el sistema, restaurar archivos o reinstalar el gestor de arranque.
- **Herramientas de recuperación:** Existen utilidades específicas para reparar el arranque (como bootrec en Windows o grub-install en Linux), recuperar archivos borrados y solucionar problemas de configuración.
- **Reinstalación del sistema operativo:** Si la recuperación no es posible, puede ser necesario reinstalar el sistema operativo, restaurando los datos desde copias de seguridad previas.

### Ejemplo de recuperación en Windows

1. Acceder al entorno de recuperación (WinRE) presionando F8 o desde un USB de instalación.
2. Seleccionar Solucionar problemas → Opciones avanzadas.
3. Usar Restaurar sistema, Reparación de inicio o Símbolo del sistema para ejecutar comandos como:

```
1 bootrec /fixmbr
2 bootrec /fixboot
3 bootrec /scanos
4 bootrec /rebuildbcd
```

### Ejemplo de recuperación en Linux

1. Arrancar desde un Live CD/USB.
2. Montar la partición raíz y reinstalar el gestor de arranque GRUB:

```
1 sudo mount /dev/sda1 /mnt
2 sudo grub-install --root-directory=/mnt /dev/sda
3 sudo update-grub
```
3. Reiniciar el sistema y comprobar el funcionamiento.

#### 4.12.4. Buenas prácticas en actualización y recuperación

- Realizar copias de seguridad periódicas antes de actualizar o modificar el sistema.
- Leer la documentación oficial y las notas de la versión antes de aplicar actualizaciones mayores.
- Mantener el sistema y las aplicaciones actualizados para reducir riesgos de seguridad.
- Documentar los procedimientos realizados y las incidencias detectadas para facilitar futuras recuperaciones.
- Utilizar herramientas de recuperación y restauración recomendadas por el fabricante o la comunidad.

## 4.13. Documentación de la instalación y de las incidencias detectadas

La documentación de la instalación y de las incidencias detectadas es una parte fundamental en la administración de sistemas informáticos. Un registro detallado permite analizar el proceso, identificar errores, facilitar la resolución de problemas y servir como referencia para futuras instalaciones o auditorías.

### 4.13.1. Documentación de la instalación

La documentación debe incluir todos los pasos realizados durante la instalación del sistema operativo y las aplicaciones, así como las configuraciones aplicadas. Los elementos recomendados son:

- **Datos del equipo:** Marca, modelo, características técnicas (CPU, RAM, disco, tarjeta gráfica, periféricos).
- **Tipo y versión del sistema operativo:** Indicar la edición, versión y arquitectura (32/64 bits).
- **Medio de instalación:** Especificar si se utilizó USB, DVD, red, imagen ISO, etc.
- **Particionado del disco:** Describir el esquema de particiones, sistemas de archivos y tamaños asignados.
- **Configuraciones iniciales:** Idioma, zona horaria, usuario y contraseña, nombre del equipo, configuración de red.
- **Controladores instalados:** Listar los drivers instalados manualmente y su versión.
- **Aplicaciones instaladas:** Enumerar las aplicaciones principales, su versión y método de instalación (gestor de paquetes, instalador, etc.).
- **Actualizaciones aplicadas:** Registrar las actualizaciones del sistema y aplicaciones instaladas tras la instalación inicial.
- **Copias de seguridad:** Indicar si se realizaron respaldos previos y cómo se restauraron los datos.
- **Capturas de pantalla:** Adjuntar imágenes de las fases clave del proceso (particionado, configuración, instalación finalizada).

### 4.13.2. Documentación de incidencias

Registrar las incidencias detectadas durante la instalación y su resolución es esencial para mejorar procesos y evitar errores recurrentes. La documentación debe incluir:

- **Descripción de la incidencia:** Explicar el problema encontrado (error de instalación, incompatibilidad, fallo de hardware, etc.).
- **Momento de aparición:** Indicar en qué fase del proceso ocurrió la incidencia.
- **Mensajes de error:** Copiar los mensajes mostrados por el sistema o la aplicación.
- **Acciones realizadas:** Detallar los pasos seguidos para intentar resolver el problema (comandos ejecutados, cambios de configuración, reinstalación, consulta de documentación).
- **Resultado:** Indicar si la incidencia fue resuelta, cómo se solucionó o si persiste.
- **Referencias consultadas:** Documentación oficial, foros, artículos, soporte técnico.
- **Lecciones aprendidas:** Sugerencias para evitar el problema en futuras instalaciones.

### 4.13.3. Ejemplo de registro de instalación e incidencias

- **Equipo:** Lenovo ThinkPad E15, Intel i5, 16 GB RAM, SSD 512 GB.
- **SO instalado:** Ubuntu 24.04 LTS, 64 bits.
- **Medio:** USB creado con balenaEtcher.
- **Particiones:** / (ext4, 100 GB), /home (ext4, 400 GB), swap ↗ (12 GB).
- **Configuración:** Idioma español, zona horaria Madrid, usuario «admin», contraseña segura.
- **Aplicaciones:** LibreOffice, GIMP, Firefox, VLC, instalado con apt.
- **Actualizaciones:** Ejecutado sudo apt update && sudo apt upgrade ↗ tras la instalación.

### *3 Explotación de sistemas microinformáticos*

- **Incidencia:** Error al instalar el driver Wi-Fi (`rtl8821ce`). Mensaje: «No se detecta adaptador inalámbrico».
- **Acciones:** Instalado driver manualmente desde repositorio GitHub, ejecutando `dkms install`.
- **Resultado:** Wi-Fi funcional tras reinicio.
- **Lección:** Comprobar compatibilidad de hardware antes de instalar, tener acceso a Internet por cable para descargar drivers.

#### **4.13.4. Importancia de la documentación**

Una buena documentación permite:

- Replicar instalaciones exitosas en otros equipos.
- Facilitar el soporte técnico y la resolución de problemas.
- Cumplir requisitos legales y de auditoría en entornos profesionales.
- Mejorar la formación y el aprendizaje en prácticas de laboratorio.
- Mantener un historial de cambios y actualizaciones en los sistemas.

Se recomienda utilizar plantillas, hojas de cálculo o sistemas de gestión de incidencias para organizar la información y compartirla con el equipo o la comunidad.

## **Resumen**

En este capítulo se ha abordado de manera integral el proceso de instalación de sistemas operativos, desde su evolución histórica y clasificación, hasta los procedimientos prácticos y consideraciones técnicas y legales. Se han explicado las funciones esenciales de los sistemas operativos, los diferentes tipos existentes (de escritorio, servidor, empotrados y de tiempo real), así como la importancia de los gestores de arranque y las tecnologías de virtualización.

Se ha detallado el procedimiento general de instalación, incluyendo la preparación, configuración de medios, particionado, instalación de controladores y configuración final, con ejemplos específicos para Windows 11, Ubuntu y macOS. Además, se han analizado los distintos tipos de aplicaciones (de sistema, usuario y red), los métodos de instalación y desinstalación en sistemas libres

#### *4.13 Documentación de la instalación y de las incidencias detectadas*

y propietarios, y la relevancia de las licencias de software (propietarias, libres, open source, shareware y freeware).

El capítulo también ha destacado la importancia de mantener los sistemas y aplicaciones actualizados, así como los métodos de recuperación ante incidencias, proporcionando ejemplos prácticos y buenas prácticas. Finalmente, se ha subrayado el valor de la documentación detallada de la instalación y de las incidencias detectadas, como herramienta fundamental para la gestión eficiente, el soporte técnico y la mejora continua en la administración de sistemas informáticos.

En resumen, la correcta instalación, actualización y documentación de sistemas operativos y aplicaciones es clave para garantizar la seguridad, estabilidad y funcionalidad de los equipos, adaptándose a las necesidades del usuario y del entorno profesional o educativo.



# 5

# Gestión de la información

La gestión de la información en sistemas operativos implica el manejo eficiente de archivos, directorios, copias de seguridad y tareas automatizadas. Este capítulo aborda los conceptos y herramientas fundamentales para administrar la información en Windows y Linux, con ejemplos prácticos y ejercicios para ambos sistemas.

**Nota:** Como macOS se basa en Unix, muchos comandos y herramientas son similares a los de Linux, por lo que no se entra en detalle específico sobre macOS.

## 5.1. Gestión de sistemas de archivos

Los sistemas operativos ofrecen diferentes métodos para gestionar archivos y carpetas:

### 5.1.1. Windows

El sistema de ficheros de Windows presenta particularidades importantes respecto a otros sistemas operativos. Utiliza principalmente los sistemas de archivos FAT32 y NTFS.

- **FAT32 (File Allocation Table 32):** Es un sistema de archivos antiguo, compatible con muchos dispositivos y sistemas operativos. Permite particiones de hasta 2 TB y archivos de hasta 4 GB. Es ideal para memorias USB y discos externos, pero carece de funciones avanzadas de seguridad y recuperación ante fallos.
- **NTFS (New Technology File System):** Es el sistema de archivos moderno de Windows. Soporta archivos y particiones de gran tamaño, ofrece permisos avanzados, cifrado, compresión, registro de cambios y recuperación ante errores. Es el sistema recomendado para discos internos y sistemas donde se requiere seguridad y fiabilidad.

## 5 Gestión de la información

La elección entre FAT32 y NTFS depende del uso: FAT32 para compatibilidad y dispositivos extraíbles, NTFS para seguridad y rendimiento en sistemas Windows.

Adicionalmente, Windows organiza los sistemas de almacenamiento en letras de unidad, tal como se puede ver en la figura 5.1. Cada letra de unidad puede representar un tipo distinto de almacenamiento:

- **A:** y **B:** Históricamente reservadas para disqueteras, hoy en día raramente usadas.
- **C:** Unidad principal del sistema operativo, donde se instalan Windows y la mayoría de las aplicaciones.
- **D:** Usualmente utilizada para discos duros adicionales o particiones de recuperación.
- **E:** A menudo utilizada para unidades ópticas (CD/DVD) o unidades extraíbles.
- **Z:** A menudo utilizada para unidades de red o recursos compartidos.

**Nota:** A medida que se conectan nuevas unidades, por ejemplo memorias USB, estas se asignan automáticamente a letras de unidad disponibles.

### Gestión mediante entornos gráficos

El Explorador de archivos permite copiar, mover, renombrar y eliminar archivos de forma gráfica.

Las operaciones básicas en el Explorador de archivos incluyen:

- **Copiar y mover archivos:** Selecciona los archivos o carpetas, haz clic derecho y elige **Copiar** o **Cortar**, luego navega a la ubicación deseada y selecciona **Pegar**<sup>1</sup>.
- **Renombrar:** Haz clic derecho sobre el archivo o carpeta y selecciona **Renombrar**, o pulsa la tecla F2.
- **Eliminar:** Selecciona el archivo o carpeta y pulsa la tecla **Supr** o haz clic derecho y elige **Eliminar**. Los elementos eliminados se envían a la Papelera de reciclaje, desde donde pueden restaurarse o eliminarse definitivamente.
- **Crear carpetas:** Haz clic derecho en el área deseada y selecciona **Nuevo → Carpeta**.

---

<sup>1</sup>Podrás hacer uso de los atajos de teclado **Ctrl+C**, **Ctrl+X** y **Ctrl+V** respectivamente.

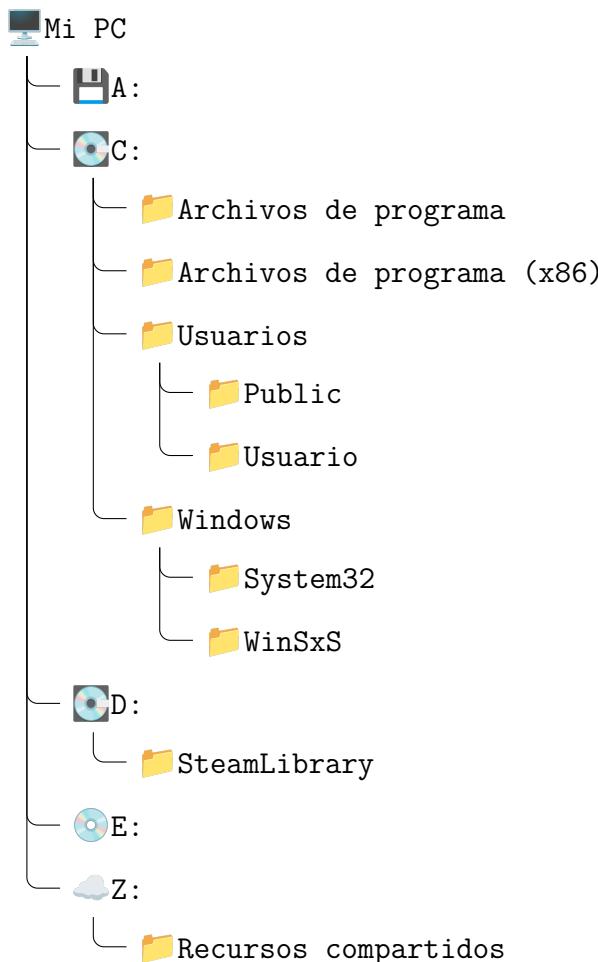


Figura 5.1: Estructura de archivos en Windows

- **Crear ficheros:** Haz clic derecho en el área deseada y selecciona Nuevo → Documento de texto (o el tipo de archivo que desees crear).
- **Propiedades:** Haz clic derecho y selecciona Propiedades para ver información sobre tamaño, permisos y ubicación.

El Explorador de archivos también permite buscar archivos, organizar la vista por tipo, fecha o tamaño, y acceder a ubicaciones de red y dispositivos externos.

### Ejercicio 5.1

Crea las carpetas y ficheros que aparecen en la figura 5.2.



## 5 Gestión de la información

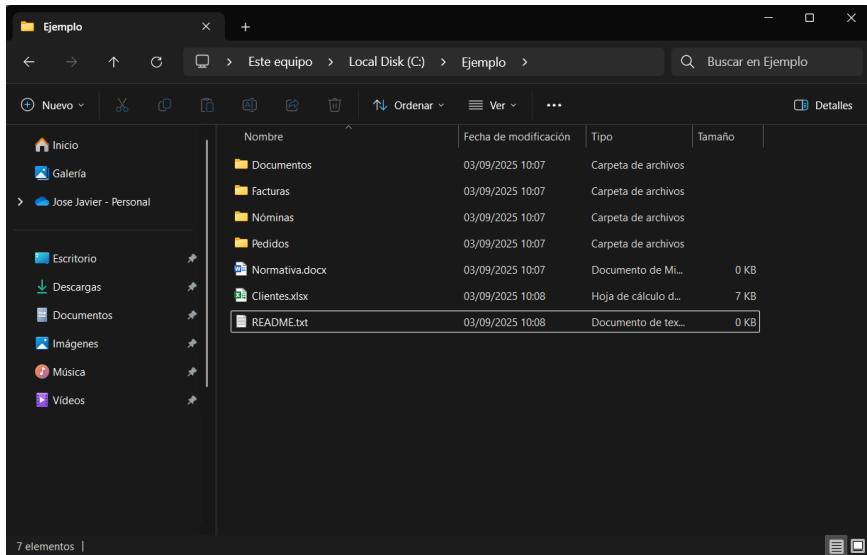


Figura 5.2: Explorador de archivos en Windows

### Gestión mediante línea de comandos

La línea de comandos (`cmd`) y PowerShell ofrecen comandos como `copy`, `move`, `del`, `ren`, `mkdir`, `rmdir`.

A continuación se muestran algunos comandos básicos para la gestión de archivos y carpetas en Windows desde la línea de comandos:

- `dir`: Lista el contenido de la carpeta actual.
- `tree`: Muestra la estructura de directorios de forma jerárquica.
- `cd <carpeta>`: Cambia el directorio actual.
- `copy <origen> <destino>`: Copia archivos de una ubicación a otra.
- `move <origen> <destino>`: Mueve archivos o carpetas.
- `del <archivo>`: Elimina archivos.
- `ren <archivo_viejo> <archivo_nuevo>`: Renombra archivos.
- `mkdir <carpeta>`: Crea una nueva carpeta.
- `rmdir <carpeta>`: Elimina una carpeta vacía.

Por ejemplo, para crear una carpeta llamada `prueba`, copiar un archivo, renombrarlo y eliminarlo, se pueden usar los siguientes comandos:

```

1 mkdir prueba
2 copy archivo.txt prueba\
3 ren prueba\archivo.txt ejemplo.txt
4 del prueba\ejemplo.txt
5 rmdir prueba

```

PowerShell ofrece comandos similares y más avanzados, como `Copy-Item`, `Move-Item`, `Remove-Item`, `Rename-Item`, y permite automatizar tareas mediante scripts.

### Ejercicio 5.2



Ve a la carpeta que creaste en el ejercicio anterior, y realiza las siguientes acciones:

1. Crea una carpeta llamada `prueba`.
2. Copia el archivo `README.txt` y nómbralolo `copia.txt`.
3. Renombra `copia.txt` a `final.txt`.
4. Mueve el fichero `final.txt` a la carpeta `prueba`.
5. Lista el contenido de la carpeta raíz usando el comando `tree`.
6. Elimina el archivo `final.txt`.
7. Elimina la carpeta creada.

Indica los comandos utilizados en la línea de comandos y adjunta capturas de pantalla del proceso.

## 5.1.2. Linux

El sistema de ficheros de Linux presenta varias particularidades que lo diferencian de otros sistemas operativos. Su estructura jerárquica parte de un único directorio raíz (`/`), desde el cual se organizan todos los archivos y directorios del sistema, sin necesidad de letras de unidad. Linux permite montar diferentes sistemas de archivos en cualquier punto del árbol de directorios, lo que facilita la gestión flexible de particiones y dispositivos.

Además, el sistema de ficheros de Linux destaca por su robusto sistema de permisos, que controla el acceso de usuarios y grupos a archivos y carpetas, y por el soporte de enlaces simbólicos y duros, que permiten crear referencias flexibles a archivos existentes. Linux soporta múltiples tipos de sistemas de ficheros, entre los que destacan `ext2`, `ext3`, `ext4`, `XFS`, `Btrfs`, entre otros.

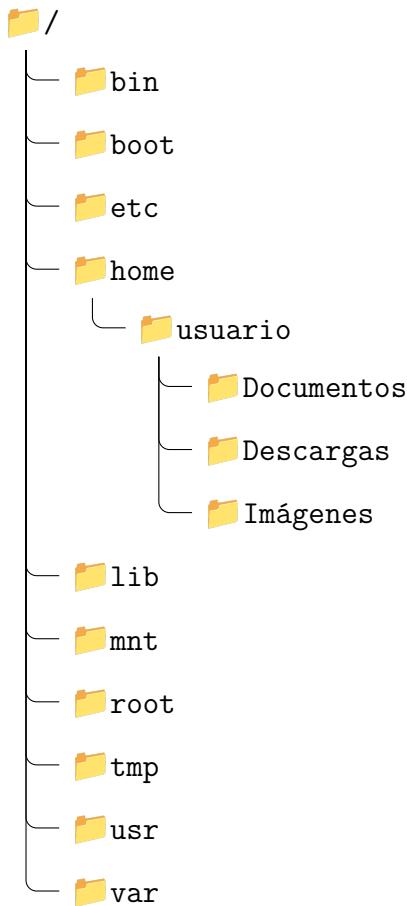


Figura 5.3: Estructura de archivos en Linux

En cuanto a ext3, es una evolución del sistema de ficheros ext2 que incorpora journaling, una técnica que registra los cambios pendientes en un diario antes de aplicarlos, lo que mejora la integridad y recuperación ante fallos del sistema. Ext3 mantiene compatibilidad con ext2, permitiendo la conversión entre ambos sin pérdida de datos. Ofrece tres modos de journaling (journal, ordered, writeback) que permiten equilibrar el rendimiento y la seguridad según las necesidades del usuario. Gracias a estas características, ext3 fue ampliamente utilizado en sistemas Linux por su fiabilidad y facilidad de recuperación tras apagados inesperados o errores.

En la figura 5.3 se puede observar la estructura de directorios típica de un sistema Linux. A continuación, enumeramos los principales directorios y su función:

- /: Directorio raíz, punto de partida del sistema de archivos.

- **/bin:** Ejecutables esenciales del sistema.
- **/boot:** Archivos de arranque del sistema.
- **/etc:** Archivos de configuración del sistema.
- **/home:** Directorio personal de los usuarios.
- **/lib:** Bibliotecas esenciales del sistema.
- **/mnt:** Puntos de montaje para sistemas de archivos.
- **/root:** Directorio personal del usuario root (administrador).
- **/tmp:** Archivos temporales.
- **/usr:** Aplicaciones y archivos de usuario.
- **/var:** Archivos variables, como logs y bases de datos.

## Gestión mediante entornos gráficos

Los entornos gráficos (Nautilus, Dolphin) facilitan la gestión visual de archivos y carpetas.

Los principales entornos gráficos para la gestión de archivos en Linux son Nautilus (GNOME), Dolphin (KDE), Thunar (XFCE) y otros. Estas aplicaciones permiten realizar operaciones como copiar, mover, renombrar, eliminar archivos y carpetas, crear nuevos elementos, y acceder a ubicaciones de red o dispositivos externos de forma visual e intuitiva.

Las acciones básicas incluyen:

- **Copiar y mover archivos:** Selecciona los archivos o carpetas, haz clic derecho y elige **Copiar** o **Cortar**, luego navega a la ubicación deseada y selecciona **Pegar**.
- **Renombrar:** Haz clic derecho sobre el archivo o carpeta y selecciona **Renombrar**, o pulsa la tecla F2.
- **Eliminar:** Selecciona el archivo o carpeta y pulsa la tecla Supr o haz clic derecho y elige **Eliminar**. Los elementos eliminados suelen enviarse a la papelera, desde donde pueden restaurarse o eliminarse definitivamente.
- **Crear carpetas y archivos:** Haz clic derecho en el área deseada y selecciona **Crear nueva carpeta** o **Crear nuevo documento**.

## 5 Gestión de la información

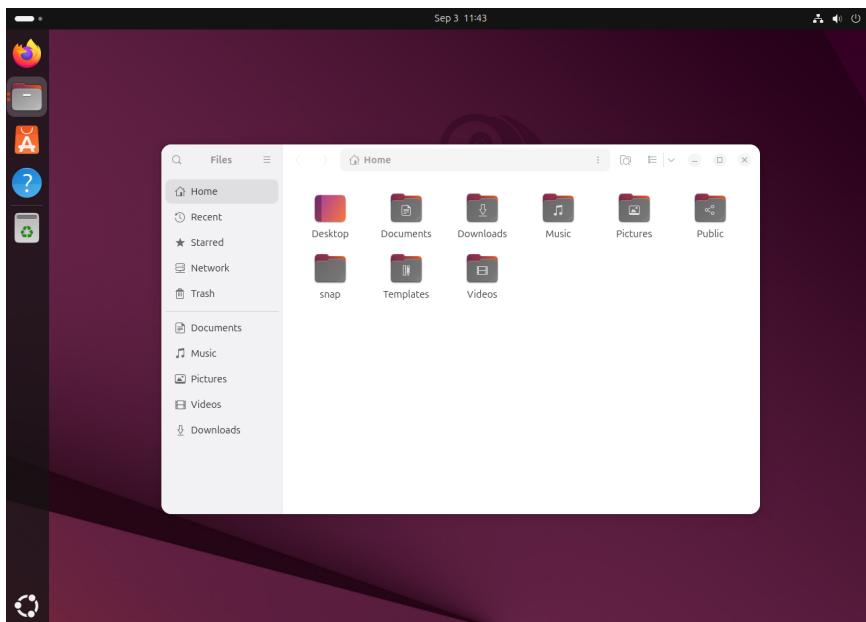


Figura 5.4: Gestor de archivos en Linux (Ubuntu)

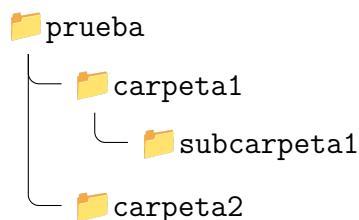
- **Propiedades:** Haz clic derecho y selecciona **Propiedades** para ver información sobre tamaño, permisos y ubicación.

Estos gestores permiten también buscar archivos, organizar la vista por tipo, fecha o tamaño, y gestionar permisos de acceso de forma sencilla.

### Ejercicio 5.3



Crea la siguiente estructura de carpetas y archivos en tu gestor de archivos de Linux:



## Gestión mediante línea de comandos

En terminal, se usan comandos como `cp`, `mv`, `rm`, `mv`, `mkdir`, `rmdir`, `ls`.

A continuación se muestran los comandos básicos para gestionar archivos y directorios en Linux desde la terminal:

- **ls**: Lista el contenido del directorio actual.
- **cd <directorio>**: Cambia al directorio indicado.
- **cp <origen> <destino>**: Copia archivos o carpetas.
- **mv <origen> <destino>**: Mueve o renombra archivos o carpetas.
- **rm <archivo>**: Elimina archivos.
- **rm -r <carpeta>**: Elimina carpetas y su contenido.
- **mkdir <carpeta>**: Crea una nueva carpeta.
- **rmdir <carpeta>**: Elimina una carpeta vacía.
- **touch <archivo>**: Crea un archivo vacío.

Ejemplo de uso para crear una estructura de carpetas y archivos:

```

1 mkdir -p prueba/carpeta1/subcarpeta1
2 mkdir prueba/carpeta2
3 touch prueba/carpeta1/archivo1.txt
4 touch prueba/carpeta2/archivo2.txt
5 ls -R prueba

```

Puedes combinar estos comandos para copiar, mover, renombrar y eliminar archivos y carpetas según tus necesidades. Recuerda que para operaciones en directorios protegidos o archivos de otros usuarios puede ser necesario usar `sudo`.

### Ejercicio 5.4



Utiliza la terminal para crear la estructura de carpetas y archivos mostrada en el ejercicio anterior.

Además, crea un fichero en la carpeta1.

Indica los comandos utilizados y adjunta capturas de pantalla del proceso.

## 5.2. Estructura de directorios de sistemas operativos

En los sistemas operativos, los directorios organizan los archivos en una estructura jerárquica que facilita su gestión y acceso. Tanto Windows como Linux utilizan rutas (paths) para localizar archivos y carpetas, pero existen diferencias en su funcionamiento y sintaxis.

### 5.2.1. Windows

En Windows, los directorios se representan mediante letras de unidad seguidas de rutas separadas por barras invertidas ( \ ). Por ejemplo, `C:\Usuarios\Usuario\Documentos\archivo.txt` es una ruta absoluta, ya que comienza desde la raíz de la unidad C:. Las rutas relativas, en cambio, se definen respecto al directorio current. Por ejemplo, si estamos en `C:\Usuarios\Usuario`, la ruta relativa `Documentos\archivo.txt` apunta al archivo dentro de la carpeta `Documentos`.

Windows permite el uso de los símbolos . (directorio actual) y .. (directorio padre) en rutas relativas, por ejemplo: ..\Imágenes\foto.jpg.

Aunque Windows no utiliza enlaces simbólicos de forma tradicional, desde Windows Vista es posible crear enlaces simbólicos (symlinks) y enlaces duros mediante el comando `mklink`. Los enlaces simbólicos permiten crear accesos directos a archivos o carpetas ubicados en otras rutas, facilitando la organización y el acceso.

### 5.2.2. Linux

En Linux, la estructura de directorios parte del directorio raíz /, y las rutas se separan por barras ( / ). Una ruta absoluta comienza siempre por /, por ejemplo: `/home/usuario/Documentos/archivo.txt`. Las rutas relativas se definen respecto al directorio actual, por ejemplo: `Documentos/archivo.txt` si estamos en `/home/usuario`.

Linux utiliza los símbolos . y .. para referirse al directorio actual y al directorio parent, respectivamente. Por ejemplo, `../Imágenes/foto.jpg` accede a un archivo en el directorio superior. Además, podemos referirnos al directorio de usuario con ~, que representa la ruta `/home/usuario`, p. ej. `~/Documentos/archivo.txt` es análogo a `/home/usuario/Documentos/archivo.txt`.

Una característica destacada de Linux es el uso de enlaces simbólicos ( symlinks ) y enlaces duros. Los enlaces simbólicos se crean con el comando `ln -s origen destino` y permiten que un archivo o carpeta apunte a otro, incluso en diferentes ubicaciones del sistema de archivos. Los enlaces duros ( `ln -r origen destino` ) crean una referencia adicional al mismo archivo físico, pero sólo pueden usarse dentro del mismo sistema de archivos.

El uso de rutas absolutas y relativas, junto con los enlaces simbólicos, facilita la organización, el acceso y la gestión flexible de archivos en ambos sistemas operativos.

**Importante**

En los sistemas operativos basados en UNIX, se distingue entre mayúsculas y minúsculas en los nombres de archivos y directorios. Esto significa que `archivo.txt` y `Archivo.txt` se consideran archivos diferentes.

## 5.3. Búsqueda de información del sistema

La búsqueda de información en el sistema de ficheros permite localizar archivos y carpetas de forma rápida y eficiente, tanto en Windows como en Linux. Existen métodos gráficos y comandos específicos para realizar estas búsquedas.

### 5.3.1. Windows

#### Interfaz gráfica

El Explorador de archivos de Windows incluye una barra de búsqueda en la parte superior derecha. Al escribir el nombre (o parte del nombre) de un archivo o carpeta, el sistema muestra los resultados en la ubicación actual y sus subcarpetas. Es posible filtrar por tipo de archivo, fecha de modificación y otros criterios. Además, el menú de búsqueda avanzada permite refinar los resultados.

#### Línea de comandos

En la línea de comandos (`cmd`), se puede buscar archivos usando el comando `dir` con opciones de búsqueda. Por ejemplo:

```
1| dir /s /b C:\ruta*.txt
```

Este comando busca todos los archivos `.txt` en la ruta indicada y sus subcarpetas. En PowerShell, se puede usar `Get-ChildItem` junto con `Where-Object` para búsquedas más avanzadas:

```
1| Get-ChildItem -Path C:\ruta -Recurse -Filter *.txt
```

### 5.3.2. Linux

#### Interfaz gráfica

Los gestores de archivos como Nautilus, Dolphin o Thunar incluyen una barra de búsqueda que permite localizar archivos y carpetas por nombre, tipo o contenido. Al escribir el término de búsqueda, el gestor muestra los resultados en la carpeta actual y, según la configuración, en subcarpetas.

## Línea de comandos

En terminal, los comandos más utilizados son `find` y `grep`. Para buscar archivos por nombre:

```
1| find /ruta -name "*.txt"
```

Para buscar texto dentro de archivos:

```
1| grep -r "palabra" /ruta
```

El comando `locate` también permite búsquedas rápidas si la base de datos está actualizada:

```
1| locate archivo.txt
```

### Ejercicio 5.5



Realiza una búsqueda de archivos de texto en una carpeta específica usando la interfaz gráfica y la línea de comandos en Windows y Linux. Adjunta capturas de pantalla y los comandos utilizados.

## 5.4. Identificación del software instalado

La identificación del software instalado es fundamental para gestionar aplicaciones, solucionar problemas y mantener el sistema actualizado. Tanto Windows como Linux ofrecen métodos gráficos y comandos para consultar los programas instalados.

### 5.4.1. Windows

#### Interfaz gráfica

En Windows, puedes ver el software instalado accediendo a **Panel de control** → **Programas y características**, donde se muestra una lista de aplicaciones instaladas, su editor y fecha de instalación. En Windows 10/11, también puedes ir a **Configuración** → **Aplicaciones**, que permite buscar, desinstalar y gestionar aplicaciones.

#### Línea de comandos

Desde la línea de comandos (`cmd`), puedes listar programas instalados usando el comando:

```
1| wmic product get name,version
```

En PowerShell, puedes obtener información más detallada con:

```
1| Get-WmiObject -Class Win32_Product | Select-Object Name, Version
```

También puedes consultar aplicaciones instaladas desde la Microsoft Store con:

```
1| Get-AppxPackage
```

## 5.4.2. Linux

### Interfaz gráfica

En Linux, los entornos de escritorio suelen incluir gestores de software como **Centro de Software de Ubuntu**, **Discover** (KDE), o **GNOME Software**, que muestran las aplicaciones instaladas y permiten buscar, instalar o eliminar programas fácilmente.

### Línea de comandos

La gestión de software depende del sistema de paquetes. En distribuciones basadas en Debian/Ubuntu, puedes listar los paquetes instalados con:

```
1| dpkg --get-selections
```

O filtrar por nombre:

```
1| dpkg -l | grep <nombre>
```

En sistemas basados en Red Hat/Fedora:

```
1| rpm -qa
```

En distribuciones con **pacman** (Arch Linux):

```
1| pacman -Q
```

Estos comandos permiten consultar la lista de paquetes instalados, sus versiones y detalles adicionales.

### Ejercicio 5.6



Identifica el software instalado en tu sistema usando la interfaz gráfica y la línea de comandos en Windows y Linux. Adjunta capturas de pantalla y los comandos utilizados.

## 5.5. Realización y restauración de copias de seguridad

Las copias de seguridad permiten proteger la información ante fallos, pérdidas o ataques. Tanto Windows como Linux ofrecen métodos gráficos y comandos para realizar y restaurar copias de seguridad.

### 5.5.1. Windows

#### Interfaz gráfica

Windows incluye herramientas como **Historial de archivos** y **Copia de seguridad y restauración** (Panel de control). Estas permiten seleccionar carpetas, programar copias automáticas y restaurar versiones anteriores de archivos. El proceso suele implicar elegir una unidad de destino (disco externo, red) y configurar la frecuencia de las copias.

#### Consejo



En Windows, existen aplicaciones como **Synkron** que permiten realizar copias de seguridad de manera sencilla y programada.

#### Línea de comandos

Desde la línea de comandos, se pueden usar utilidades como **robocopy** y **xcopy** para copiar archivos y carpetas de forma avanzada. Ejemplo:

```
1| robocopy C:\Origen D:\Destino /MIR
```

Este comando realiza una copia espejo de la carpeta origen en el destino. Para restaurar, basta con copiar los archivos de la copia de seguridad al lugar original.

### 5.5.2. Linux

#### Interfaz gráfica

En Linux, existen aplicaciones como **Deja Dup** (GNOME), **Timeshift** y **KBackup** que permiten seleccionar carpetas, programar copias automáticas y restaurar archivos fácilmente. El usuario puede elegir el destino (disco externo, red) y gestionar versiones de las copias.

#### Línea de comandos

La terminal ofrece herramientas como **rsync**, **tar** y **cp** para realizar copias de seguridad. Ejemplo con **rsync**:

```
1| rsync -av --delete /home/usuario/ /media/backup/
```

Este comando sincroniza la carpeta de usuario con la ubicación de respaldo. Para restaurar, se copian los archivos desde la copia al directorio original.

### Ejercicio 5.7



Realiza una copia de seguridad de una carpeta en Windows y Linux usando la interfaz gráfica y la línea de comandos. Restaura algún archivo desde la copia y documenta el proceso con capturas de pantalla y comandos utilizados.

## 5.6. Herramientas de administración de discos

Las herramientas de administración de discos permiten gestionar particiones, volúmenes, realizar desfragmentación, chequeo y cifrado de datos. A continuación se describen las principales opciones en Windows y Linux, tanto en interfaz gráfica como en línea de comandos.

### 5.6.1. Windows

#### Interfaz gráfica

Windows incluye la herramienta **Administración de discos** (`diskmgmt.msc`), accesible desde el menú de inicio o ejecutando el comando `diskmgmt.msc`. Permite crear, eliminar, redimensionar particiones, asignar letras de unidad y formatear discos. También se puede convertir discos entre formatos básicos y dinámicos, y gestionar volúmenes RAID.

Para desfragmentar discos en Windows, se utiliza la aplicación **Desfragmentar y optimizar unidades**, accesible desde el menú de inicio o buscando «desfragmentar» en la barra de búsqueda. Esta herramienta analiza el estado de fragmentación de los discos duros y permite optimizarlos reorganizando los archivos para que ocupen espacios contiguos, lo que mejora la velocidad de acceso y el rendimiento general del sistema. El proceso puede programarse para ejecutarse automáticamente en intervalos regulares, y es especialmente recomendable en discos mecánicos (HDD), ya que en unidades de estado sólido (SSD) la desfragmentación no es necesaria y el sistema realiza optimizaciones específicas para este tipo de almacenamiento. La aplicación muestra el porcentaje de fragmentación y el estado de cada unidad, permitiendo al usuario iniciar la optimización manualmente o configurar la programación automática según sus necesidades.

El cifrado de discos en Windows puede realizarse mediante **BitLocker**, una herramienta integrada en las ediciones profesionales y empresariales del sistema operativo. BitLocker permite cifrar volúmenes completos, incluidos discos internos y externos, protegiendo la información ante accesos no autorizados, robos o pérdidas del dispositivo. El proceso de cifrado utiliza algoritmos avanzados (AES) y puede requerir el uso de un módulo TPM (Trusted Platform Module)

para almacenar de forma segura las claves de cifrado. BitLocker ofrece opciones para establecer contraseñas, utilizar autenticación mediante PIN o integrar la protección con cuentas de Microsoft y Active Directory. Además, permite configurar el cifrado automático de nuevas unidades y gestionar la recuperación mediante claves de respaldo, que pueden guardarse en archivos, impresiones o cuentas en la nube. Una vez activado, el acceso a los datos cifrados requiere la autenticación configurada, y el sistema realiza el cifrado y descifrado de forma transparente para el usuario.

### Línea de comandos

Desde la línea de comandos, se pueden usar herramientas como `diskpart` para gestionar discos y particiones. **Diskpart** es una utilidad avanzada que permite crear, eliminar, redimensionar y formatear particiones, así como asignar letras de unidad y gestionar discos dinámicos. Es especialmente útil para tareas que requieren mayor control que la interfaz gráfica, como la automatización de procesos o la gestión de discos en servidores.

A continuación se muestra un ejemplo de uso típico de `diskpart` para crear y formatear una partición:

```
1 diskpart
2 list disk
3 select disk 0
4 create partition primary size=10240
5 format fs=ntfs quick
6 assign letter=E
7 exit
```

- `list disk` Muestra todos los discos conectados al sistema.
- `select disk 0` Selecciona el disco sobre el que se va a trabajar (en este caso, el disco 0).
- `create partition primary size=10240` Crea una partición primaria de 10 GB.
- `format fs=ntfs quick` Formatea la partición en NTFS de forma rápida.
- `assign letter=E` Asigna la letra de unidad E: a la nueva partición.
- `exit` Sale de la utilidad diskpart.

**Nota:** Diskpart permite también convertir discos entre formatos básicos y dinámicos, limpiar discos, gestionar volúmenes RAID y trabajar con discos virtuales (VHD).

Para desfragmentar, se puede usar el comando `defrag`, que reorganiza los archivos en el disco para mejorar el rendimiento. El parámetro `/O` optimiza el disco tras la desfragmentación:

```
1| defrag C: /O
```

La desfragmentación es especialmente importante en discos duros mecánicos (HDD), ya que los archivos pueden quedar dispersos físicamente, ralentizando el acceso. En unidades SSD, el sistema operativo realiza optimizaciones específicas y no es necesario desfragmentar.

El chequeo de discos se realiza con `chkdsk`, que analiza el sistema de archivos en busca de errores y los repara si es posible. El parámetro `/F` indica que se deben corregir los errores encontrados:

```
1| chkdsk C: /F
```

- `chkdsk C:` Analiza la unidad C: en busca de errores.
- `/F` Corrige automáticamente los errores detectados.
- `/R` Localiza sectores defectuosos y recupera información legible.

El cifrado de volúmenes mediante BitLocker se puede gestionar con `manage-bde`, una herramienta de línea de comandos que permite activar, desactivar y administrar el cifrado de discos. Por ejemplo, para cifrar la unidad E: se utiliza:

```
1| manage-bde -on E:
```

- `manage-bde -on E:` Activa BitLocker en la unidad E:.
- `manage-bde -status E:` Consulta el estado del cifrado en la unidad E:.
- `manage-bde -off E:` Desactiva BitLocker y descifra la unidad.
- `manage-bde -protectors -add E: -RecoveryPassword` Añade una clave de recuperación.

**BitLocker** utiliza cifrado AES y puede requerir un módulo TPM para almacenar las claves de forma segura. Permite proteger discos internos, externos y unidades USB, y ofrece opciones de recuperación en caso de pérdida de la contraseña o fallo del sistema.

## 5.6.2. Linux

### Interfaz gráfica

En Linux, existen herramientas gráficas como **GParted** y **KDE Partition Manager** para crear, eliminar, redimensionar y formatear particiones. Estas aplicaciones permiten gestionar discos de manera visual y sencilla, mostrando información detallada sobre cada dispositivo, tipo de sistema de archivos, espacio libre y ocupado, y permitiendo operaciones avanzadas como mover, copiar, cambiar el tamaño y establecer etiquetas en las particiones. GParted soporta una amplia variedad de sistemas de archivos (ext2, ext3, ext4, NTFS, FAT32, XFS, Btrfs, entre otros) y puede trabajar con discos internos, externos y memorias USB. Además, permite gestionar particiones LVM y realizar operaciones sin perder datos, aunque siempre se recomienda hacer copias de seguridad antes de modificar particiones.

KDE Partition Manager ofrece funcionalidades similares, integrándose con el entorno KDE y proporcionando una interfaz intuitiva para usuarios de ese escritorio. Ambas herramientas permiten visualizar la estructura del disco mediante gráficos y tablas, facilitando la identificación de particiones y el estado de cada una.

Para desfragmentar, generalmente no es necesario en sistemas de archivos modernos como ext4, ya que implementan técnicas de asignación eficiente que minimizan la fragmentación. Sin embargo, existen utilidades como **e4defrag** para casos específicos donde se detecta fragmentación excesiva, especialmente en sistemas con uso intensivo de archivos grandes o muchas operaciones de escritura. El comando **e4defrag** analiza y optimiza la estructura interna del sistema de archivos ext4, mejorando el rendimiento en situaciones concretas. Ejemplo de uso:

```
1| sudo e4defrag /home/usuario
```

El cifrado de discos puede realizarse mediante aplicaciones como **Discos** (GNOME), que permite crear y gestionar particiones cifradas de forma gráfica, o configurando cifrado al instalar el sistema (por ejemplo, seleccionando la opción de cifrado de disco completo en el instalador de Ubuntu). El cifrado en Linux suele implementarse mediante LUKS (Linux Unified Key Setup), que proporciona un estándar seguro y flexible para proteger datos en discos y particiones. El usuario puede establecer contraseñas, gestionar claves de recuperación y elegir algoritmos de cifrado. Además, **cryptsetup** permite administrar volúmenes cifrados desde la terminal, ofreciendo opciones avanzadas para usuarios experimentados.

## Línea de comandos

La gestión de particiones y volúmenes se realiza con herramientas como **fdisk** →, **parted**, **lsblk** y **mkfs**. Estas utilidades permiten crear, modificar, eliminar y visualizar particiones, así como formatear y montar sistemas de archivos en discos duros, SSDs y dispositivos extraíbles.

**fdisk** Herramienta clásica para gestionar particiones en discos con formato MBR. Permite listar particiones, crear nuevas, eliminarlas y cambiar sus tipos. Ejemplo de uso:

```
1| sudo fdisk /dev/sda
```

Dentro de fdisk, se pueden usar comandos como **m** (ayuda), **p** (listar particiones), **n** (nueva partición), **d** (eliminar), **w** (guardar cambios).

**parted** Similar a fdisk, pero soporta discos con formato GPT y operaciones avanzadas como redimensionar particiones. Ejemplo:

```
1| sudo parted /dev/sdb
```

Permite crear, eliminar, redimensionar y mover particiones, así como cambiar el tipo de sistema de archivos.

**lsblk** Muestra una lista jerárquica de todos los dispositivos de bloques (discos, particiones, volúmenes) conectados al sistema, facilitando la identificación de discos y particiones antes de operar sobre ellos.

**mkfs** Permite formatear una partición con el sistema de archivos deseado (ext4, xfs, btrfs, etc.). Ejemplo:

```
1| sudo mkfs.ext4 /dev/sda1
```

Tras formatear, se puede montar la partición en un punto de montaje:

```
1| sudo mount /dev/sda1 /mnt
```

Para comprobar y reparar sistemas de archivos, se utiliza **fsck** (File System Check), que analiza la integridad del sistema de archivos y repara errores:

```
1| sudo fsck /dev/sda1
```

**fsck** puede detectar bloques dañados, corregir errores de estructura y recuperar archivos huérfanos. Es recomendable ejecutar fsck en particiones desmontadas para evitar daños.

Para cifrado, se utiliza **cryptsetup** con LUKS (Linux Unified Key Setup), el estándar para cifrado de discos en Linux. Permite crear volúmenes cifrados, abrirlos y gestionarlos:

## 5 Gestión de la información

```
1| sudo cryptsetup luksFormat /dev/sda2
2| sudo cryptsetup open /dev/sda2 secure_disk
```

El primer comando inicializa el cifrado en la partición, solicitando una contraseña segura. El segundo comando abre el volumen cifrado y lo asocia al nombre `secure_disk`, permitiendo su uso como cualquier otra partición tras formatearla y montarla.

Para desfragmentar ext4, se utiliza `e4defrag`, que analiza y optimiza la estructura interna del sistema de archivos ext4, mejorando el rendimiento en casos de fragmentación elevada:

```
1| sudo e4defrag /home/usuario
```

El comando muestra un informe del estado de fragmentación y realiza la optimización si es necesario.

Estas herramientas permiten administrar discos, particiones, volúmenes, realizar chequeos, desfragmentación y cifrado de datos en sistemas Linux, proporcionando flexibilidad y control avanzado sobre el almacenamiento.

## 5.7. Tareas automáticas

La planificación de tareas automáticas permite ejecutar acciones programadas en el sistema, como copias de seguridad, actualizaciones o limpieza de archivos, sin intervención manual. Tanto Windows como Linux ofrecen herramientas gráficas y de línea de comandos para gestionar estas tareas.

### 5.7.1. Windows

#### Interfaz gráfica

Windows incluye el **Programador de tareas** (Task Scheduler), accesible desde el menú de inicio o buscando «Programador de tareas». Esta herramienta permite crear, modificar y eliminar tareas programadas, especificando el desencadenante (hora, evento, inicio de sesión, etc.), la acción (ejecutar programa, enviar correo, mostrar mensaje) y las condiciones adicionales (repetición, expiración, etc.). El asistente guía al usuario en la configuración de tareas periódicas o puntuales, facilitando la automatización de procesos como copias de seguridad, limpieza de archivos temporales o ejecución de scripts.

#### Línea de comandos

Desde la línea de comandos, se puede utilizar el comando `schtasks` para crear, listar, modificar y eliminar tareas programadas. Ejemplo para crear una tarea que ejecuta un script cada día a las 18:00:

```

1| schtasks /create /tn "CopiaDiaria" /tr "C:\scripts\backup.bat" /→
 sc daily /st 18:00
2| schtasks /query

```

El comando `schtasks` es una herramienta muy potente para la gestión de tareas programadas en Windows desde la línea de comandos o scripts. Permite automatizar procesos, ejecutar scripts, lanzar aplicaciones y realizar tareas administrativas de forma periódica o bajo demanda.

#### Parámetros principales:

- `/create` Crea una nueva tarea.
- `/tn` Nombre de la tarea ( `Task Name`).
- `/tr` Ruta del programa o script a ejecutar ( `Task Run`).
- `/sc` Frecuencia ( `Schedule`), por ejemplo: `daily`, `weekly`, `monthly` → , `onstart`, `onlogon`.
- `/st` Hora de inicio ( `Start Time`), en formato HH:MM (24h).
- `/ru` Usuario bajo el que se ejecuta la tarea (por defecto, el usuario actual).
- `/rp` Contraseña del usuario (si es necesario).
- `/delete` Elimina una tarea existente.
- `/query` Lista las tareas programadas.
- `/change` Modifica una tarea existente.
- `/enable` y `/disable`: Activa o desactiva tareas.

**Ejemplo:** Crear una tarea semanal que ejecuta un script los lunes y viernes a las 20:30 bajo el usuario `Administrador`:

```

1| schtasks /create /tn "InformeSemanal" /tr "C:\scripts\informe.←
 bat" /sc weekly /d MON,FRI /st 20:30 /ru Administrador /rp ←
 MiContraseñaSegura

```

#### Listar todas las tareas programadas:

```

1| schtasks /query /fo LIST /v

```

El parámetro `/fo LIST` muestra la información en formato lista, y `/v` (verbose) añade detalles como el estado, última ejecución, próxima ejecución, usuario, etc.

#### Eliminar una tarea programada:

```

1| schtasks /delete /tn "CopiaDiaria" /f

```

## 5 Gestión de la información

El parámetro `/f` fuerza la eliminación sin pedir confirmación.

### Modificar una tarea existente:

```
1| schtasks /change /tn "CopiaDiaria" /enable
```

Permite activar o desactivar tareas, cambiar el usuario, etc.

### Automatización y scripting:

```
1| REM Crear varias tareas desde un script por lotes
2| for %%D in (LUN MAR MIE JUE VIE) do (
3| schtasks /create /tn "Backup_%%D" /tr "C:\scripts\backup_%%D.bat" /sc weekly /d %%D /st 19:00
4)
```

Este ejemplo crea una tarea de copia de seguridad para cada día de la semana laboral.

### Seguridad y buenas prácticas:

- Utiliza rutas absolutas y verifica los permisos del usuario que ejecuta la tarea.
- Protege las contraseñas y evita incluirlas en scripts públicos.
- Revisa periódicamente el estado de las tareas programadas y los logs de ejecución.
- Documenta cada tarea y su propósito para facilitar el mantenimiento.

**Visualización y gestión avanzada:** Las tareas programadas pueden visualizarse y gestionarse también desde la interfaz gráfica del Programador de tareas, donde se pueden ver los desencadenantes, acciones, historial y condiciones de cada tarea.

## 5.7.2. Linux

### Interfaz gráfica

En Linux, existen aplicaciones como **GNOME Schedule** (en escritorios GNOME) o **KDE Task Scheduler** que ofrecen una interfaz visual para programar tareas automáticas. Estas herramientas permiten crear, editar y eliminar tareas cron y at, especificando comandos, horarios y frecuencia de ejecución. El usuario puede programar tareas recurrentes (diarias, semanales, mensuales) o puntuales, y gestionar fácilmente la automatización de procesos habituales.

## Línea de comandos

La planificación de tareas en Linux se realiza principalmente mediante `cron` y `at`. Para tareas recurrentes, se edita el archivo `crontab`:

```
1 crontab -e
2 0 18 * * * /home/usuario/scripts/backup.sh
3 echo "/home/usuario/scripts/backup.sh" | at 18:00
```

La planificación de tareas en Linux se realiza principalmente mediante `cron` (para tareas recurrentes) y `at` (para tareas puntuales). Estas herramientas permiten automatizar procesos, ejecutar scripts, realizar copias de seguridad, actualizaciones y tareas de mantenimiento sin intervención manual.

### Cron:

```
1 crontab -e
```

Este comando abre el editor de tareas programadas del usuario actual. Cada línea del archivo `crontab` define una tarea con la siguiente sintaxis:

```
1 minuto hora día_mes mes día_semana comando
```

Por ejemplo, para ejecutar un script cada día a las 18:00:

```
1 0 18 * * * /home/usuario/scripts/backup.sh
```

Se pueden usar comas para varios valores, guiones para rangos y asteriscos para «cualquier valor». Ejemplo: ejecutar un script los lunes y viernes a las 20:30:

```
1 30 20 * * 1,5 /home/usuario/scripts/informe.sh
```

**Variables de entorno y logs:** Por defecto, cron ejecuta los comandos con un entorno mínimo. Es recomendable definir variables como `PATH` al inicio del crontab:

```
1 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Para registrar la salida de los comandos, se puede redirigir a un archivo de log:

```
1 0 18 * * * /home/usuario/scripts/backup.sh >> /home/usuario/backup.log 2>&1
```

### Gestión de tareas:

- `crontab -l` Lista las tareas programadas del usuario actual.
- `crontab -r` Elimina todas las tareas del usuario.
- `sudo crontab -e -u usuario` Edita el crontab de otro usuario.
- Los archivos de crontab de todos los usuarios se almacenan en `/var/spool/cron/crontabs`.

**Automatización avanzada:** Se pueden programar scripts que realicen tareas complejas, como copias de seguridad, limpieza de archivos temporales, sincronización de datos, etc. Ejemplo de script semanal:

```
1| 0 3 * * 0 /home/usuario/scripts/backup_semanal.sh
```

Este comando ejecuta el script todos los domingos a las 3:00 AM.

**Seguridad y buenas prácticas:**

- Verifica los permisos de los scripts y archivos involucrados.
- Usa rutas absolutas para evitar errores.
- Documenta cada tarea y su propósito.
- Revisa los logs periódicamente para detectar fallos.
- Evita programar tareas que puedan consumir muchos recursos en horarios de alta actividad.

**At:** at permite programar tareas puntuales para ejecutarse una sola vez en el futuro. Ejemplo:

```
1| echo "/home/usuario/scripts/backup.sh" | at 18:00
```

Se puede especificar la fecha y hora exacta, por ejemplo:

```
1| echo "reboot" | at 23:55 03.09.2025
```

Para listar las tareas pendientes:

```
1| atq
```

Para eliminar una tarea programada:

```
1| atrm <número_tarea>
```

**Automatización y scripting:** Se pueden crear scripts que programen tareas con at desde otros procesos, por ejemplo para ejecutar una acción tras finalizar una copia de seguridad.

**Ejercicio práctico:**

**Ejercicio 5.8**

Programa una tarea en Linux que realice una copia de seguridad diaria de una carpeta usando cron, y otra tarea puntual que envíe un mensaje al usuario usando at. Documenta los comandos utilizados, la configuración del crontab y adjunta capturas de pantalla del proceso y los logs generados.



## Resumen

En este capítulo se han abordado los conceptos y herramientas fundamentales para la gestión de la información en sistemas operativos Windows y Linux. Se ha explicado la estructura de los sistemas de archivos, la organización de directorios y las diferencias entre sistemas propietarios y libres. Se han presentado métodos gráficos y comandos para gestionar archivos, buscar información, identificar software instalado, realizar copias de seguridad y administrar discos, incluyendo operaciones de desfragmentación, chequeo y cifrado. Además, se ha detallado la planificación de tareas automáticas mediante herramientas específicas en ambos sistemas. El capítulo proporciona ejemplos prácticos y ejercicios para afianzar los conocimientos adquiridos, facilitando la administración eficiente y segura de la información en distintos entornos operativos.



# 6

# Configuración de sistemas operativos

La configuración de sistemas operativos es fundamental para garantizar la seguridad, el rendimiento y la correcta gestión de los recursos. Este capítulo aborda los conceptos clave y las herramientas prácticas para configurar usuarios, permisos, servicios y monitorización en Windows y Linux, con ejemplos detallados para ambos sistemas.

## 6.1. Configuración de usuarios y grupos

La gestión de usuarios y grupos permite controlar quién accede al sistema y qué permisos tiene. Los usuarios representan cuentas individuales, mientras que los grupos facilitan la administración colectiva de permisos. Una buena organización de usuarios y grupos mejora la seguridad y simplifica la gestión de recursos.

### 6.1.1. Windows

La gestión de usuarios y grupos en Windows puede realizarse tanto de forma gráfica como por consola, permitiendo crear, modificar y eliminar cuentas, así como asignar permisos y pertenencia a grupos. Los usuarios individuales tienen configuraciones y privilegios propios, mientras que los grupos facilitan la administración colectiva de permisos. Es fundamental organizar correctamente los usuarios y grupos para mantener la seguridad y facilitar la gestión del sistema.

#### Gestión gráfica

Desde el Panel de control, en «Cuentas de usuario», puedes crear, modificar y eliminar cuentas. En ediciones profesionales, la herramienta `lusrmgr.msc` permite una gestión avanzada de usuarios y grupos locales.

## 6 Configuración de sistemas operativos

- Asignar imágenes, cambiar el tipo de cuenta (estándar/administrador) y definir contraseñas.
- Los grupos permiten asignar permisos colectivos, por ejemplo, el grupo «Administradores» tiene privilegios elevados sobre el sistema.

### Gestión por consola

- Crear usuario y añadir a grupo:

```
1| net user juan MiContraseña /add
2| net localgroup Administradores juan /add
```

- Listar usuarios:

```
1| net user
```

- Eliminar usuario:

```
1| net user juan /delete
```

- Crear y gestionar grupos:

```
1| net localgroup profesores /add
2| net localgroup profesores juan /add
```

### Buenas prácticas

- Asigna contraseñas seguras y cambia las credenciales predeterminadas.
- Limita el número de administradores y revisa periódicamente los miembros de cada grupo.
- Documenta los cambios realizados en la gestión de usuarios y grupos.

#### 6.1.2. Linux

La gestión de usuarios y grupos en Linux es esencial para controlar el acceso y los permisos sobre los recursos del sistema. Cada usuario tiene un identificador único (UID) y pertenece a uno o varios grupos (GID), lo que permite asignar privilegios de forma individual o colectiva. La administración puede realizarse tanto de forma gráfica como por consola, siendo esta última la más utilizada en entornos profesionales y servidores.

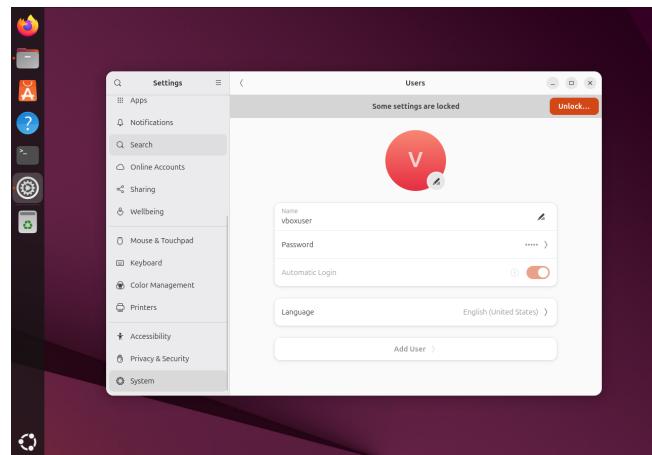


Figura 6.1: Gestión de usuarios en Ubuntu

## Gestión gráfica

En entornos como GNOME y KDE, puedes gestionar usuarios desde «Configuración del sistema» → «Usuarios». Permite crear cuentas, asignar grupos, definir contraseñas y eliminar usuarios de forma visual.

## Gestión por consola

- Crear usuario y añadir a grupo (por ejemplo, sudo):

```
1| sudo adduser juan
2| sudo usermod -aG sudo juan
```

- Listar todos los usuarios del sistema:

```
1| cut -d: -f1 /etc/passwd
```

- Eliminar usuario:

```
1| sudo deluser juan
```

- Crear y gestionar grupos:

```
1| sudo addgroup profesores
2| sudo adduser juan profesores
```

## Archivos relevantes

- */etc/passwd*: Lista de usuarios.
- */etc/group*: Lista de grupos.

- `/etc/shadow`: Contraseñas cifradas.

### Buenas prácticas

- Usa contraseñas robustas y cambia las credenciales predeterminadas.
- Limita el uso de la cuenta root y utiliza sudo para tareas administrativas.
- Revisa periódicamente los usuarios y grupos existentes, eliminando cuentas innecesarias.

#### Importante

Podrás ejecutar comandos en Linux con `sudo` para obtener permisos de superusuario temporalmente. Asegúrate de entender los riesgos asociados al uso de estos privilegios.

## 6.2. Seguridad de cuentas de usuario

La seguridad de cuentas de usuario implica proteger las credenciales, restringir el acceso no autorizado y auditar las acciones realizadas por cada usuario. Es fundamental aplicar políticas de bloqueo, expiración y auditoría, así como limitar los privilegios y eliminar cuentas innecesarias. Tanto en Windows como en Linux, existen herramientas y configuraciones específicas para fortalecer la seguridad de las cuentas, prevenir ataques y garantizar la trazabilidad de las acciones. Una gestión adecuada reduce el riesgo de accesos indebidos y mejora la protección global del sistema.

### 6.2.1. Windows

La seguridad de cuentas de usuario en Windows se basa en la aplicación de políticas que regulan el acceso, la protección ante intentos de acceso no autorizados y la auditoría de acciones.

- **Políticas de bloqueo:** Permiten bloquear cuentas tras varios intentos fallidos de inicio de sesión, evitando ataques de fuerza bruta. Se configuran en «Política de seguridad local» (`secpol.msc`) → «Política de bloqueo de cuenta».
- **Expiración y deshabilitación:** Las cuentas pueden configurarse para expirar tras un periodo de inactividad o deshabilitarse manualmente.

- **Auditoría de acceso:** El sistema puede registrar intentos de inicio de sesión exitosos y fallidos, cambios de contraseña y modificaciones de cuentas en el Visor de eventos.

- **Ejemplo avanzado:** Bloquear una cuenta tras 3 intentos fallidos y definir el tiempo de desbloqueo:

```
1| net accounts /lockoutthreshold:3
2| net accounts /lockoutduration:30
3| net accounts /lockoutwindow:30
```

- **Deshabilitar una cuenta temporalmente:**

```
1| net user juan /active:no
```

- **Auditoría:** Configura la auditoría en «Política de seguridad local» → «Políticas de auditoría» para registrar eventos relevantes.

extbfBuenas prácticas:

- Deshabilita o elimina cuentas que no se utilicen.
- Revisa periódicamente los registros de auditoría y los intentos fallidos.
- Limita el número de cuentas con privilegios elevados.
- Configura el bloqueo automático tras varios intentos fallidos y define un tiempo de desbloqueo razonable.

## 6.2.2. Linux

En Linux, la seguridad de cuentas de usuario se gestiona mediante archivos de configuración, módulos PAM y herramientas de auditoría.

- **Bloqueo tras intentos fallidos:** Se configura en `/etc/pam.d/common-auth` con módulos como `pam_tally2` o `pam_faillock`. Ejemplo para bloquear tras 3 intentos:

```
1| sudo faillog -u juan -m 3
```

- **Deshabilitar cuentas:** Puedes bloquear una cuenta manualmente:

```
1| sudo usermod -L juan
```

- **Expiración de cuentas:** Configura la fecha de expiración con `chage`:

```
1| sudo chage -E 2025-12-31 juan
```

- **Auditoría:** Revisa los logs de autenticación en `/var/log/auth.log` y utiliza herramientas como `auditd` para monitorizar cambios y accesos.

- **Ejemplo para desbloquear una cuenta tras bloqueo:**

```
1| sudo pam_tally2 --user juan --reset
```

### Buenas prácticas:

- Elimina o bloquea cuentas inactivas o innecesarias.
- Configura el bloqueo tras intentos fallidos y revisa los logs periódicamente.
- Limita el uso de cuentas con privilegios de sudo/root.
- Audita los cambios y accesos con `auditd` y revisa los informes regularmente.

## 6.3. Seguridad de contraseñas

La seguridad de contraseñas es esencial para proteger el acceso a los sistemas operativos. Una contraseña segura debe ser larga, compleja y difícil de adivinar. Los sistemas operativos permiten configurar políticas que obligan a los usuarios a cumplir ciertos requisitos de seguridad.

- **Longitud mínima:** Establece el número mínimo de caracteres requeridos.
- **Complejidad:** Exige el uso de mayúsculas, minúsculas, números y símbolos.
- **Caducidad:** Obliga a cambiar la contraseña periódicamente.
- **Historial:** Impide reutilizar contraseñas recientes.
- **Bloqueo tras intentos fallidos:** Protege contra ataques de fuerza bruta.

### Importante

Utiliza gestores de contraseñas para almacenar credenciales de forma segura y nunca compartas tus contraseñas. Considera activar la autenticación en dos pasos (2FA) cuando esté disponible.

### 6.3.1. Windows

La seguridad de contraseñas en Windows se gestiona principalmente mediante políticas locales (`secpol.msc`) o directivas de grupo en entornos empresariales. Las opciones más relevantes incluyen:

- **Longitud mínima:** Define el número mínimo de caracteres. Ejemplo:

```
1| net accounts /minpwlen:8
```

- **Complejidad:** Requiere el uso de mayúsculas, minúsculas, números y símbolos. Se activa en «Política de seguridad local» → «Política de contraseñas».

- **Caducidad:** Permite forzar el cambio de contraseña cada cierto tiempo. Ejemplo:

```
1| net accounts /maxpwage:30
```

- **Historial:** Impide reutilizar las últimas contraseñas. Ejemplo:

```
1| net accounts /uniquepw:5
```

- **Bloqueo tras intentos fallidos:** Protege contra ataques de fuerza bruta. Ejemplo:

```
1| net accounts /lockoutthreshold:3
```

extbfBuenas prácticas:

- Utiliza contraseñas de al menos 12 caracteres y activa la complejidad.
- Cambia las contraseñas periódicamente y evita reutilizarlas.
- No compartas contraseñas y utiliza autenticación multifactor si está disponible.
- Audita los cambios y accesos mediante el visor de eventos.

### 6.3.2. Linux

En Linux, la seguridad de contraseñas se gestiona mediante archivos de configuración y módulos PAM. Las opciones más relevantes incluyen:

- **Longitud mínima y complejidad:** Configurable en `/etc/login.defs` y mediante `pam_pwquality` o `pam_cracklib`. Ejemplo de configuración en `/etc/security/pwquality.conf`:

```
1 minlen = 10
2 dcredit = -1
3 ucredit = -1
4 ocredit = -1
5 lcredit = -1
```

- **Caducidad y expiración:** Se gestiona con `chage`. Ejemplo para forzar cambio cada 90 días:

```
1 sudo chage -M 90 juan
```

- **Historial de contraseñas:** Se puede activar con el módulo `pam_pwhistory` para evitar reutilización.
- **Bloqueo tras intentos fallidos:** Se configura con `pam_tally2` o `faillog`. Ejemplo:

```
1 sudo faillog -u juan -m 3
```

### Buenas prácticas

- Usa contraseñas largas y complejas, y cambia las credenciales predeterminadas.
- Configura la caducidad y el historial para evitar reutilización.
- Revisa los logs de autenticación en `/var/log/auth.log`.
- Considera el uso de autenticación en dos pasos (2FA) si el sistema lo permite.

## 6.4. Acceso a recursos

El acceso a recursos consiste en definir y controlar quién puede utilizar archivos, carpetas, impresoras y otros elementos del sistema operativo. Los permisos pueden ser de lectura, escritura, modificación y ejecución, y se asignan a usuarios y grupos según las necesidades de seguridad y operatividad. Una gestión adecuada de los accesos previene el uso indebido de la información y protege los datos frente a modificaciones no autorizadas. Los sistemas operativos ofrecen herramientas gráficas y de consola para compartir recursos y establecer permisos, tanto localmente como en red.

## 6.4.1. Windows

El acceso a recursos en Windows abarca la gestión de carpetas, archivos, impresoras y unidades de red. Los permisos pueden ser de lectura, escritura, modificación y control total, y se asignan tanto a usuarios como a grupos.

### Gestión gráfica

- **Compartir carpetas y archivos:** Clic derecho sobre el recurso → «Propiedades» → «Compartir» → Seleccionar usuarios y definir permisos.
- **Compartir impresoras:** «Panel de control» → «Dispositivos e impresoras» → «Propiedades» → «Compartir».
- **Acceso a unidades de red:** «Conectar a unidad de red» permite mapear recursos compartidos en otros equipos.

### Gestión por consola

- **Compartir carpeta y otorgar permisos completos:**

```
1| net share Recursos=C:\Recursos /grant:juan,full
```

- **Listar recursos compartidos:**

```
1| net share
```

- **Eliminar recurso compartido:**

```
1| net share Recursos /delete
```

### Buenas prácticas

- Limita el acceso solo a los usuarios/grupos necesarios.
- Revisa y audita periódicamente los recursos compartidos y sus permisos.
- Utiliza nombres descriptivos para los recursos y documenta su propósito.
- Configura el cifrado de datos en recursos sensibles.

## 6.4.2. Linux

En Linux, el acceso a recursos se gestiona principalmente mediante permisos de archivos y carpetas, y mediante servicios de red como Samba y NFS para compartir con otros sistemas.

## Gestión gráfica

- **Compartir carpetas:** En entornos como GNOME y KDE, clic derecho sobre la carpeta → «Opciones de compartición» → «Configurar usuarios y permisos».
- **Acceso a recursos de red:** «Conectar a servidor» permite acceder a recursos compartidos en otros equipos.

## Gestión por consola

- **Compartir carpeta con Samba:** Edita `/etc/samba/smb.conf` para definir el recurso y los permisos.

```
1 sudo nano /etc/samba/smb.conf
2 # Añadir al final del archivo:
3 [Ejercicio]
4 path = /home/usuario/Ejercicio
5 valid users = juan
6 read only = no
```

- **Reiniciar el servicio Samba:**

```
1 sudo systemctl restart smbd
```

- **Listar recursos compartidos:**

```
1 smbclient -L localhost -U juan
```

## Buenas prácticas

- Limita el acceso solo a los usuarios y grupos necesarios.
- Revisa los permisos de archivos y carpetas antes de compartirlos.
- Audita los logs de Samba y NFS para detectar accesos no autorizados.
- Utiliza cifrado en recursos compartidos sensibles y restringe el acceso desde redes externas.

## 6.5. Permisos locales

Los permisos locales son reglas que determinan el acceso de los usuarios y grupos a los archivos y carpetas dentro de un sistema operativo. Una correcta configuración de estos permisos es esencial para proteger la información y evitar modificaciones no autorizadas. Los sistemas operativos modernos permiten

gestionar estos permisos tanto de forma gráfica como mediante comandos de consola, ofreciendo flexibilidad y control granular sobre los recursos locales.

- **Tipos de permisos:** Los más habituales son lectura, escritura, ejecución y modificación. Estos pueden combinarse para definir el nivel de acceso de cada usuario o grupo.
- **Propietario y grupo:** Cada archivo o carpeta tiene un propietario y un grupo asociado, lo que facilita la asignación colectiva de permisos.
- **Permisos predeterminados:** Al crear nuevos archivos o carpetas, el sistema asigna permisos por defecto que pueden ser modificados posteriormente.
- **Gestión avanzada:** Además de los permisos básicos, existen mecanismos como las listas de control de acceso (ACL) que permiten una configuración más detallada.

Una gestión adecuada de los permisos locales ayuda a cumplir el principio de mínimo privilegio, reduce el riesgo de accesos indebidos y facilita la auditoría de cambios y accesos en el sistema.

### 6.5.1. Windows

En Windows, los permisos locales determinan qué usuarios y grupos pueden acceder, modificar o ejecutar archivos y carpetas. El sistema de archivos NTFS permite una gestión granular de permisos, que incluyen:

- **Lectura (Read):** Permite ver el contenido de archivos y carpetas.
- **Escritura (Write):** Permite modificar archivos y crear nuevos elementos.
- **Modificación (Modify):** Incluye lectura, escritura y eliminación de archivos.
- **Ejecución (Execute):** Permite ejecutar archivos y programas.
- **Control total (Full control):** Permite realizar cualquier acción, incluyendo cambiar permisos y tomar posesión.

#### Gestión gráfica

Haz clic derecho sobre el archivo o carpeta → «Propiedades» → «Seguridad» para ver y modificar los permisos asignados a cada usuario o grupo.

### Gestión por consola

Utiliza el comando `icacls` para consultar y modificar permisos NTFS:

```
1 # Otorga control total a juan
2 icacls C:\Recursos /grant juan:F
3 # Otorga permisos de modificación al grupo profesores
4 icacls C:\Recursos /grant profesores:M
5 # Elimina los permisos de juan
6 icacls C:\Recursos /remove juan
7 # Muestra los permisos actuales
8 icacls C:\Recursos
```

### Buenas prácticas

- Asigna el menor nivel de permisos necesario para cada usuario o grupo (principio de mínimo privilegio).
- Revisa y audita periódicamente los permisos de archivos y carpetas sensibles.
- Evita otorgar «Control total» salvo que sea imprescindible.
- Documenta los cambios realizados en la configuración de permisos.

### 6.5.2. Linux

En Linux, los permisos locales se gestionan mediante atributos de archivos y carpetas, y determinan quién puede leer, escribir o ejecutar cada elemento. Los permisos tradicionales se representan con tres grupos: propietario (user), grupo (group) y otros (others).

- **Lectura (r):** Permite ver el contenido.
- **Escritura (w):** Permite modificar o eliminar.
- **Ejecución (x):** Permite ejecutar archivos o acceder a carpetas.

### Gestión por consola

```
1 # Ver permisos actuales
2 ls -l /home/juan/archivo.txt
3 # Cambia propietario y grupo
4 sudo chown juan:profesores /home/juan/archivo.txt
5 # Da lectura y escritura a propietario y grupo
6 sudo chmod 660 /home/juan/archivo.txt
7 # Da permiso de ejecución al propietario
8 sudo chmod u+x /home/juan/script.sh
```

## Permisos numéricos

Los permisos pueden representarse con números (por ejemplo, 660 = `rw--rw----`).

## Gestión avanzada

Para permisos más detallados, Linux permite el uso de ACLs (Listas de Control de Acceso) con `setfacl` y `getfacl`.

```
1| setfacl -m u:ana:r /home/juan/archivo.txt # Da solo lectura a ↪
 ana
2| getfacl /home/juan/archivo.txt # Muestra los permisos ACL
```

## Buenas prácticas

- Asigna permisos restrictivos por defecto y amplíalos solo cuando sea necesario.
- Revisa periódicamente los permisos de archivos y carpetas importantes.
- Evita el uso de permisos 777 (todos los permisos para todos) salvo en casos excepcionales.
- Documenta los cambios y utiliza grupos para facilitar la gestión colectiva de permisos.

### Consejo



Podrás ver los permisos de un archivo o carpeta en Linux utilizando el comando `ls -l`. La salida mostrará los permisos en formato simbólico y numérico, así como el propietario y grupo asociados.

## 6.6. Listas de control de acceso

Las listas de control de acceso (ACL, por sus siglas en inglés) son mecanismos avanzados que permiten definir permisos detallados sobre archivos y carpetas, superando las limitaciones de los permisos tradicionales. Una ACL especifica qué usuarios o grupos pueden acceder a un recurso y qué acciones pueden realizar (lectura, escritura, ejecución, etc.), permitiendo una gestión granular y flexible de la seguridad.

- Ventajas de las ACLs:

- Permiten asignar permisos específicos a múltiples usuarios y grupos.
- Facilitan la gestión en entornos colaborativos o con necesidades de acceso diferenciadas.
- Mejoran la trazabilidad y el control sobre los recursos sensibles.

- **Tipos de ACL:**

- **Discretionary ACL (DACL):** Controla los permisos de acceso.
- **System ACL (SACL):** Define qué acciones deben ser auditadas.

- **Gestión:** Las ACLs pueden configurarse tanto de forma gráfica como mediante comandos de consola, dependiendo del sistema operativo.

El uso adecuado de las ACLs permite cumplir el principio de mínimo privilegio y proteger la información frente a accesos no autorizados, siendo una herramienta fundamental en la administración avanzada de sistemas operativos.

### 6.6.1. Windows

Las Listas de Control de Acceso (ACLs) en Windows permiten definir permisos avanzados y detallados sobre archivos y carpetas, más allá de los permisos básicos. Una ACL es un conjunto de reglas que especifica qué usuarios o grupos pueden acceder a un recurso y qué acciones pueden realizar.

- **Tipos de ACL:**

- **DACL (Discretionary ACL):** Controla quién puede acceder y con qué permisos (lectura, escritura, ejecución, etc.).
- **SACL (System ACL):** Define qué acciones deben ser auditadas (por ejemplo, acceso o modificación).

- **Gestión gráfica:** Haz clic derecho sobre el archivo o carpeta, selecciona “Propiedades”, luego “Seguridad” y “Opciones avanzadas” para ver y editar las entradas de la ACL. Aquí puedes agregar usuarios, grupos y definir permisos específicos, como herencia, acceso condicional, y auditoría.

- **Gestión por consola:** El comando `icacls` permite consultar y modificar ACLs desde la línea de comandos. Ejemplo:

```
1 # Otorga control total a juan, incluyendo herencia a subcarpetas→
2 # y archivos
2 icacls C:\Recursos /grant juan:(OI)(CI)F
3 # Muestra la ACL completa de un recurso
4 icacls C:\Recursos
```

```

5 # Elimina una entrada de la ACL
6 icacls C:\Recursos /remove juan
7 # Copia la ACL de un recurso a otro
8 icacls C:\Recursos /save acl.txt
9 icacls C:\NuevoRecurso /restore acl.txt

```

## Buenas prácticas

- Revisa y ajusta las ACLs periódicamente, especialmente en recursos sensibles.
- Evita otorgar permisos excesivos o innecesarios.
- Documenta los cambios y utiliza auditoría para registrar accesos importantes.
- Utiliza grupos para simplificar la gestión de permisos.

### 6.6.2. Linux

En Linux, las ACLs permiten asignar permisos más detallados que los tradicionales (propietario, grupo, otros), facilitando la gestión en entornos colaborativos o con necesidades específicas.

- **Concepto:** Una ACL define permisos para múltiples usuarios y grupos sobre un mismo archivo o carpeta, sin limitarse al propietario y grupo principal.
- **Gestión por consola:**
  - **setfacl:** Permite añadir, modificar o eliminar entradas en la ACL de un archivo o carpeta.
  - **getfacl:** Muestra la ACL actual de un recurso.
- **Ejemplos:**

```

1 # Da permisos de lectura al grupo profesores
2 setfacl -m g:profesores:r /home/juan/archivo.txt
3 # Da permisos de escritura a usuario ana
4 setfacl -m u:ana:w /home/juan/archivo.txt
5 # Elimina los permisos ACL de un usuario
6 setfacl -x u:ana /home/juan/archivo.txt
7 # Muestra la ACL completa
8 getfacl /home/juan/archivo.txt
9 # Da permisos recursivos a una carpeta y su contenido
10 setfacl -R -m g:profesores:rw /home/juan/Ejercicios

```

## Buenas prácticas

- Utiliza ACLs solo cuando los permisos tradicionales no sean suficientes.
- Revisa y limpia las ACLs periódicamente para evitar configuraciones confusas o inseguras.
- Documenta los cambios y utiliza grupos para facilitar la gestión colectiva.
- Audita los accesos y cambios en archivos importantes.

### Ejercicio 6.1



Realiza las siguientes tareas en un sistema Windows y en un sistema Linux:

1. Crea un usuario llamado `practica` y asígnalo al grupo de administradores (Windows) o al grupo `sudo` (Linux).
2. Crea una carpeta llamada `Ejercicio` y comparte el acceso con el usuario `practica` otorgándole permisos completos.
3. Configura la política de contraseñas para que la longitud mínima sea de 8 caracteres.
4. Verifica que el usuario `practica` puede acceder y modificar archivos en la carpeta `Ejercicio`.
5. Elimina el usuario `practica` y comprueba que ya no tiene acceso a la carpeta compartida.

*Incluye los comandos utilizados y captura de pantalla (si es posible) de cada paso.*

## 6.7. Servicios y procesos

Los servicios y procesos son componentes fundamentales en la administración de sistemas operativos. Un servicio es una aplicación que se ejecuta en segundo plano para proporcionar funcionalidades esenciales (como red, impresión, seguridad, etc.), mientras que un proceso es cualquier programa o tarea activa en ejecución. La correcta gestión de servicios y procesos permite optimizar el rendimiento, garantizar la disponibilidad de recursos y mantener la seguridad del sistema.

- **Servicios:** Programas que se ejecutan automáticamente y gestionan tareas críticas del sistema.
- **Procesos:** Instancias de programas en ejecución, que pueden ser servicios, aplicaciones o scripts.
- **Gestión:** Los sistemas operativos ofrecen herramientas gráficas y de consola para iniciar, detener, monitorizar y configurar servicios y procesos.
- **Monitorización:** Es importante revisar el consumo de recursos, detectar procesos problemáticos y auditar los servicios activos.
- **Automatización:** Los servicios pueden configurarse para iniciar automáticamente, manualmente o permanecer deshabilitados según las necesidades del sistema.

Una administración adecuada de servicios y procesos contribuye a la estabilidad, seguridad y eficiencia del sistema operativo, permitiendo detectar y resolver incidencias de forma proactiva.

### 6.7.1. Windows

En Windows, los servicios son programas que se ejecutan en segundo plano y proporcionan funcionalidades esenciales (como impresión, red, actualizaciones, etc.), mientras que los procesos representan cualquier programa o tarea activa en el sistema.

- **Gestión gráfica:**
  - **Administrador de tareas:** Permite ver y gestionar procesos activos, su consumo de recursos y finalizar tareas problemáticas.
  - **Servicios ( `services.msc` ):** Permite iniciar, detener, pausar y configurar servicios del sistema, así como establecer el tipo de inicio (automático, manual, deshabilitado).
- **Gestión por consola:**
  - `sc query`: Lista todos los servicios y su estado.
  - `sc start <servicio>`: Inicia un servicio (ejemplo: `sc start → spooler`).
  - `sc stop <servicio>`: Detiene un servicio.
  - `tasklist`: Muestra todos los procesos activos y su PID.
  - `taskkill /PID <pid> /F`: Finaliza un proceso por su identificador.

## 6 Configuración de sistemas operativos

- **Get-Service** y **Get-Process** (PowerShell): Comandos avanzados para consultar y gestionar servicios y procesos.

- **Ejemplo de comandos:**

```
1 # Consultar servicios
2 sc query
3 # Iniciar el servicio de impresión
4 sc start spooler
5 # Detener el servicio de impresión
6 sc stop spooler
7 # Listar procesos activos
8 tasklist
9 # Finalizar un proceso por PID
10 taskkill /PID 1234 /F
11 # Consultar servicios con PowerShell
12 Get-Service
13 # Consultar procesos con PowerShell
14 Get-Process
```

### Buenas prácticas

- No detengas servicios críticos sin conocer su función.
- Revisa el consumo de recursos y finaliza procesos solo si afectan el rendimiento o la estabilidad.
- Configura los servicios para que inicien automáticamente solo si son necesarios.
- Documenta los cambios realizados en la configuración de servicios y procesos.

### Ejercicio 6.2



Utiliza los comandos adecuados en Windows para listar todos los servicios activos y encontrar el servicio de «Spooler de impresión». Luego, detén el servicio y verifica que ya no está en ejecución. Finalmente, vuelve a iniciar el servicio y comprueba que se ha reiniciado correctamente.

*Consejo: Puedes usar `sc query` para listar los servicios y `sc start <servicio>` y `sc stop <servicio>` para gestionar el servicio de impresión.*

## 6.7.2. Linux

En Linux, los servicios (también llamados «daemons»<sup>1</sup>) son programas que se ejecutan en segundo plano y gestionan tareas como red, impresión, web, etc. Los procesos incluyen tanto servicios como aplicaciones y scripts en ejecución.

- Gestión de servicios:

- **systemctl**: Herramienta principal para administrar servicios en sistemas con systemd. Permite iniciar, detener, reiniciar, habilitar/deshabilitar servicios y consultar su estado.
- **service**: Comando tradicional para gestionar servicios en sistemas sin systemd.

- Gestión de procesos:

- **ps aux**: Muestra todos los procesos activos con detalles como usuario, PID, uso de CPU y memoria.
- **top** y **htop**: Herramientas interactivas para monitorizar procesos y recursos en tiempo real.
- **kill <pid>**: Finaliza un proceso por su identificador. El parámetro **-9** fuerza la terminación.

- Ejemplo de comandos:

```

1 # Consultar el estado del servicio SSH
2 sudo systemctl status ssh
3 # Iniciar el servicio SSH
4 sudo systemctl start ssh
5 # Detener el servicio SSH
6 sudo systemctl stop ssh
7 # Habilitar el servicio para que inicie automáticamente
8 sudo systemctl enable ssh
9 # Listar todos los procesos activos
10 ps aux
11 # Monitorizar procesos en tiempo real
12 top
13 # Finalizar un proceso con PID 1234
14 kill -9 1234

```

---

<sup>1</sup>En castellano, se usa el término «demonios». Sin embargo, esto es un *false friend*, ya que la traducción correcta sería «duendes».

## Buenas prácticas

- No detengas servicios esenciales para el sistema o la red sin justificación.
- Utiliza `systemctl` para una gestión moderna y centralizada de servicios.
- Monitoriza el uso de recursos y finaliza procesos solo si afectan la estabilidad o seguridad.
- Documenta los cambios y revisa los logs de servicios para detectar problemas.

### Ejercicio 6.3



Ejecuta el siguiente comando en Linux, para crear un proceso en segundo plano que escriba la fecha y hora actual en consola cada 10 segundos:

```
1| while true; do date; sleep 10; done &
```

Luego, utiliza los comandos adecuados para listar los procesos y encontrar el PID del proceso que acabas de crear. Finalmente, termina el proceso utilizando su PID.

*Consejo: Puedes usar `ps --forest` para ver los procesos en forma de árbol y localizar más fácilmente el proceso que has creado.*

## 6.8. Comandos de sistemas operativos

En esta sección se presentan los comandos más importantes y utilizados en la administración de sistemas operativos, tanto en Windows como en Linux. El dominio de estos comandos permite realizar tareas de gestión, diagnóstico, automatización y resolución de problemas de manera eficiente. Se agrupan por categorías y se incluyen ejemplos prácticos para facilitar su comprensión y aplicación en entornos reales.

### 6.8.1. Windows

En Windows, existen numerosos comandos que permiten gestionar el sistema operativo desde la consola, facilitando tareas administrativas, de diagnóstico y automatización. A continuación se agrupan los comandos más relevantes por categorías y se explica su utilidad:

- Gestión de usuarios y grupos:
  - `net user`: Gestiona cuentas de usuario (crear, modificar, eliminar, listar).

- **net localgroup:** Gestiona grupos locales y añade usuarios a grupos.
- **Permisos y seguridad:**
  - **icacls:** Consulta y modifica permisos NTFS y ACLs en archivos y carpetas.
- **Servicios y procesos:**
  - **sc:** Gestiona servicios (iniciar, detener, consultar estado).
  - **tasklist:** Lista los procesos activos.
  - **taskkill:** Finaliza procesos por nombre o PID.
- **Diagnóstico y administración avanzada:**
  - **wmic:** Interfaz para consultar y modificar información del sistema (hardware, software, procesos, etc.).
  - **shutdown:** Apaga o reinicia el sistema.
  - **gpupdate:** Actualiza las directivas de grupo.
  - **regedit:** Editor gráfico del registro de Windows.

## Ejemplo de comandos útiles

```

1 # Reiniciar el sistema inmediatamente
2 shutdown /r /t 0
3 # Actualizar directivas de grupo
4 gpupdate /force
5 # Abrir el editor del registro
6 regedit
7 # Listar todos los usuarios
8 net user
9 # Consultar permisos de una carpeta
10 icacls C:\Recursos
11 # Detener el servicio de impresión
12 sc stop spooler
13 # Finalizar el proceso notepad.exe
14 taskkill /IM notepad.exe /F

```

## Buenas prácticas

- Utiliza la consola con privilegios de administrador para tareas críticas.
- Documenta los comandos utilizados y sus efectos, especialmente en entornos de producción.

- Antes de modificar el registro o los permisos, realiza copias de seguridad.
- Consulta la ayuda de cada comando (por ejemplo, `net /?`) para conocer todas sus opciones.

### Ficheros por lotes

En Windows, los ficheros por lotes (batch files) son archivos de texto con extensión `.bat` o `.cmd` que contienen una secuencia de comandos que se ejecutan automáticamente en la consola de Windows (CMD). Estos scripts permiten automatizar tareas administrativas como la gestión de usuarios, copias de seguridad, instalación de programas o configuración de permisos.

Un fichero por lotes básico puede incluir comandos, variables, bucles y condicionales. Por ejemplo:

```
1 @echo off
2 echo Listado de usuarios:
3 net user
4 pause
```

Para ejecutar un fichero por lotes, basta con hacer doble clic sobre el archivo o ejecutarlo desde la consola. Los scripts de Windows son útiles para tareas repetitivas y para facilitar la administración de sistemas en entornos Windows.

#### Consejo



Puedes crear ficheros por lotes con cualquier editor de texto y aprovechar variables (`\%USERNAME\%`), bucles (`for`) y condicionales (`if`) para scripts más avanzados.

#### Ejercicio 6.4



Crea un fichero por lotes llamado `listar\_servicios.bat` que realice las siguientes acciones:

1. Muestra un mensaje de bienvenida.
2. Lista todos los servicios activos en el sistema.
3. Pausa la ejecución para que el usuario pueda ver la lista antes de cerrar la ventana.

Guarda el fichero y ejecútalo para verificar que funciona correctamente.

*Consejo: Utiliza los comandos `echo` para mostrar mensajes y `pause` para pausar la ejecución.*

## 6.8.2. Linux

En Linux, la consola es la herramienta principal para la administración y diagnóstico del sistema. Los comandos se agrupan por funcionalidad y permiten automatizar tareas, gestionar usuarios, permisos, procesos y recursos. A continuación se explican los comandos más relevantes:

- **Gestión de usuarios y contraseñas:**

- `useradd`, `adduser`: Crear usuarios.
- `passwd`: Cambiar contraseñas.
- `deluser`, `userdel`: Eliminar usuarios.

- **Permisos y propiedad de archivos:**

- `chmod`: Cambia los permisos de archivos y carpetas.
- `chown`: Cambia el propietario y grupo de archivos.
- `chgrp`: Cambia el grupo de archivos.
- `setfacl`, `getfacl`: Gestiona ACLs avanzadas.

- **Procesos y servicios:**

- `ps`, `top`, `htop`: Monitorizan procesos activos.
- `kill`: Finaliza procesos por PID.
- `systemctl`, `service`: Gestionan servicios del sistema.

- **Monitorización y logs:**

- `journalctl`: Consulta logs del sistema (systemd).
- `dmesg`: Muestra mensajes del kernel.
- `df`, `du`: Consultan uso de disco y espacio ocupado por carpetas.
- `free`: Muestra uso de memoria RAM.

- **Sistema y apagado:**

- `shutdown`, `reboot`: Apagan o reinician el sistema.

### Ejemplo de comandos útiles

```

1 # Reiniciar el sistema
2 sudo shutdown -r now
3 # Consultar logs de errores
4 sudo journalctl -xe
5 # Ver uso de disco

```

## 6 Configuración de sistemas operativos

```
6 df -h
7 # Ver espacio ocupado por la carpeta de usuario
8 du -sh /home/juan
9 # Cambiar permisos de un archivo
10 chmod 644 archivo.txt
11 # Cambiar propietario de un archivo
12 chown juan:profesores archivo.txt
13 # Listar procesos activos
14 ps aux
15 # Finalizar el proceso con PID 1234
16 kill -9 1234
17 # Consultar el estado del servicio SSH
18 sudo systemctl status ssh
```

### Buenas prácticas

- Utiliza `sudo` solo cuando sea necesario para tareas administrativas.
- Documenta los comandos y scripts utilizados para facilitar la gestión y auditoría.
- Antes de eliminar usuarios o modificar permisos, verifica el impacto en el sistema.

#### Consejo



Podrás usar `man` para consultar la ayuda de cualquier comando en Linux, proporcionando detalles sobre su uso, opciones y ejemplos. Por ejemplo, `man ls` mostrará la documentación del comando `ls`.

### Ficheros de lotes y scripts

En Linux, los scripts de Bash son archivos de texto que contienen una secuencia de comandos que se ejecutan de forma automática. Permiten automatizar tareas administrativas, como la gestión de usuarios, permisos, copias de seguridad o monitorización del sistema. Un script básico comienza con la línea `#!/bin/bash` y puede incluir comandos, estructuras de control (if, for, while), variables y funciones.

```
1 #!/bin/bash
2 echo "Listado de usuarios:"
3 cut -d: -f1 /etc/passwd
```

Para ejecutar un script, primero se le otorgan permisos de ejecución (`chmod +x script.sh`) y luego se ejecuta con `./script.sh`. Los scripts de Bash son

una herramienta fundamental para la administración eficiente y repetible de sistemas Linux.

### Consejo



La primera línea de un script de Bash, conocida como «shebang» (`#!/bin/bash`), indica al sistema qué intérprete utilizar para ejecutar el script. Es esencial para asegurar que el script se ejecute correctamente en el entorno adecuado.

*Nota: Es posible usar otros intérpretes, como `#!/usr/bin/env python3` para scripts en Python.*

### Ejercicio 6.5



Crea un script de Bash llamado `listar_usuarios.sh` que realice las siguientes acciones:

1. Muestra un mensaje de bienvenida.
2. Lista todos los usuarios del sistema.
3. Pausa la ejecución para que el usuario pueda ver la lista antes de cerrar la terminal.

Guarda el script, otórgale permisos de ejecución y ejecútalo para verificar que funciona correctamente.

*Consejo: Utiliza los comandos `echo` para mostrar mensajes y `read -p "Presiona Enter para continuar..."` para pausar la ejecución.*

## 6.9. Herramientas de monitorización del sistema

La monitorización del sistema es el proceso de observar y analizar el estado, rendimiento y actividad de los recursos del sistema operativo (CPU, memoria, disco, red, procesos, etc.). Permite detectar problemas, optimizar el uso de recursos y anticipar incidencias antes de que afecten a los usuarios. Existen herramientas gráficas y de consola que facilitan la monitorización en tiempo real y el análisis histórico, tanto en Windows como en Linux.

- **Objetivos de la monitorización:**

- Identificar procesos que consumen excesivos recursos.
- Detectar cuellos de botella en CPU, memoria, disco o red.

- Auditar el uso de servicios y aplicaciones.
- Prevenir fallos mediante alertas y análisis de tendencias.
- Facilitar la resolución de incidencias y la toma de decisiones para mejoras.

- **Tipos de monitorización:**

- **En tiempo real:** Permite observar el estado actual del sistema y reaccionar ante problemas inmediatos.
- **Histórica:** Registra datos a lo largo del tiempo para analizar tendencias y planificar mejoras.
- **Herramientas:** Los sistemas operativos incluyen utilidades integradas y permiten instalar herramientas avanzadas para monitorizar recursos, generar informes y configurar alertas.

Una monitorización adecuada ayuda a mantener la estabilidad, seguridad y rendimiento del sistema operativo, permitiendo una administración proactiva y eficiente.

### 6.9.1. Windows

En Windows, existen varias herramientas para monitorizar el estado y rendimiento del sistema, tanto gráficas como por consola:

- **Monitor de recursos (Resource Monitor):** Permite analizar en detalle el uso de CPU, memoria RAM, disco y red. Es útil para identificar procesos que consumen muchos recursos o detectar cuellos de botella.
- **Administrador de tareas (Task Manager):** Ofrece una vista rápida de los procesos activos, el rendimiento general, los servicios y las aplicaciones en ejecución. Permite finalizar procesos y ver el consumo de recursos por aplicación.
- **Monitor de rendimiento (Performance Monitor, perfmon):** Herramienta avanzada para crear gráficos, establecer alertas y registrar datos históricos sobre el rendimiento del sistema. Permite monitorizar contadores específicos (CPU, disco, red, memoria, etc.) y guardar informes.
- **Comandos de consola:**
  - `tasklist`: Lista todos los procesos activos y su consumo de memoria.
  - `typeperf`: Permite monitorizar contadores de rendimiento desde la línea de comandos.

- **systeminfo**: Muestra información detallada del sistema y hardware.

### Ejemplo de comandos útiles

```

1 # Abrir el Monitor de recursos
2 perfmon
3 # Listar procesos activos
4 tasklist
5 # Monitorizar el uso de CPU desde consola
6 typeperf "\Processor(_Total)\% Processor Time"
7 # Ver información del sistema
8 systeminfo

```

### Buenas prácticas

- Utiliza el Monitor de recursos y el Administrador de tareas para identificar procesos que afectan el rendimiento.
- Configura alertas en el Monitor de rendimiento para detectar problemas antes de que impacten al usuario.
- Revisa periódicamente el uso de recursos y elimina aplicaciones innecesarias.
- Documenta los hallazgos y acciones tomadas para facilitar la resolución de incidencias futuras.

#### Ejercicio 6.6



Utiliza el Monitor de recursos en Windows para identificar el proceso que consume más CPU en tu sistema. Luego, utiliza el comando **tasklist** en la consola para verificar el PID (Identificador de Proceso) de ese proceso. *Consejo: Puedes abrir el Monitor de recursos desde el Administrador de tareas o ejecutando **resmon.exe**.*

### 6.9.2. Linux

En Linux, la monitorización del sistema se realiza principalmente desde la consola, aunque existen herramientas gráficas y web. Las más utilizadas son:

- **top**: Muestra en tiempo real los procesos activos, uso de CPU, memoria y carga del sistema. Permite ordenar y filtrar procesos.
- **htop**: Versión mejorada de **top**, con interfaz interactiva y visual, permite gestionar procesos fácilmente.

## 6 Configuración de sistemas operativos

- **vmstat**: Muestra estadísticas de memoria virtual, procesos, paginación y uso de CPU.
- **iostat**: Informa sobre el uso de dispositivos de almacenamiento y el rendimiento de entrada/salida.
- **free**: Muestra el uso de memoria RAM y swap.
- **sar**: Permite recolectar y analizar datos históricos de rendimiento (requiere instalar el paquete **sysstat**).
- **glances**: Herramienta avanzada que muestra información resumida de CPU, memoria, disco, red y procesos en una sola pantalla.
- **Comandos adicionales:**
  - **ps aux**: Lista todos los procesos activos con detalles.
  - **df -h**: Muestra el uso de espacio en disco.
  - **du -sh <carpeta>**: Muestra el espacio ocupado por una carpeta.

### Ejemplo de comandos útiles

```
1 # Monitorizar procesos en tiempo real
2 top
3 # Interfaz avanzada para monitorización
4 htop
5 # Ver uso de memoria RAM
6 free -m
7 # Ver estadísticas de disco
8 iostat
9 # Ver estadísticas de memoria virtual
10 vmstat 2 5
11 # Monitorización avanzada (si está instalado)
12 glances
13 # Ver uso de disco
14 df -h
15 # Ver espacio ocupado por la carpeta de usuario
16 du -sh /home/juan
```

### Buenas prácticas

- Utiliza **top** o **htop** para identificar procesos que consumen muchos recursos.
- Revisa periódicamente el uso de disco y memoria para evitar saturaciones.

- Instala y configura `glances` o `sar` para monitorización avanzada y generación de informes.
- Documenta los problemas detectados y las acciones correctivas realizadas.

### Ejercicio 6.7



Utiliza el comando `top` en Linux para identificar el proceso que consume más memoria en tu sistema. Luego, utiliza el comando `ps aux` para verificar el PID (Identificador de Proceso) de ese proceso.

*Consejo: Puedes ordenar los procesos por uso de memoria presionando la tecla M mientras estás en la interfaz de `top`.*

## 6.10. Registros y logs

Los registros y logs son archivos o bases de datos donde el sistema operativo y las aplicaciones almacenan información sobre eventos, errores, accesos y actividades relevantes. Su análisis es fundamental para la auditoría, la monitorización y la resolución de problemas. Los logs permiten detectar incidencias, analizar el comportamiento del sistema, identificar intentos de acceso no autorizado y documentar cambios realizados.

- **Tipos de registros:** Incluyen logs de sistema, seguridad, aplicaciones, servicios y hardware.
- **Ubicación:** Los sistemas operativos almacenan los logs en rutas específicas y ofrecen herramientas para consultarlos.
- **Gestión:** Es posible filtrar, exportar, rotar y eliminar logs según las necesidades de administración y cumplimiento normativo.
- **Auditoría:** La revisión periódica de los registros ayuda a anticipar problemas, mejorar la seguridad y cumplir con políticas de trazabilidad.

Una gestión adecuada de los registros y logs contribuye a la estabilidad, seguridad y transparencia del sistema operativo.

### 6.10.1. Windows

En Windows, los registros y logs son fundamentales para la auditoría, el diagnóstico y la resolución de problemas. El sistema almacena información sobre eventos de sistema, seguridad, aplicaciones y hardware.

## Tipos de logs

- **Sistema:** Registra eventos relacionados con el funcionamiento del sistema operativo y los controladores.
- **Seguridad:** Almacena intentos de inicio de sesión, cambios de permisos y auditoría de accesos.
- **Aplicaciones:** Guarda eventos generados por programas instalados.
- **Hardware:** Incidencias y cambios en dispositivos físicos.

## Herramientas gráficas

- **Visor de eventos (eventvwr.msc):** Permite consultar, filtrar y exportar logs. Se accede desde el menú de inicio o ejecutando `eventvwr.msc`. Es posible buscar eventos por tipo, fecha, usuario o gravedad.
- **Monitor de confiabilidad:** Ofrece una vista cronológica de errores y advertencias, facilitando la identificación de problemas recurrentes.

## Herramientas de consola

- `wevtutil`: Permite consultar, exportar y limpiar registros desde la línea de comandos. Ejemplo:

```
1| wevtutil qe System /c:5 /f:text
```

- `Get-EventLog` (PowerShell): Consulta eventos de forma avanzada. Ejemplo:

```
1| Get-EventLog -LogName System -Newest 10
```

## Ubicación de los logs

- Los registros se almacenan en archivos con extensión `.evtx` en la ruta `C:\Windows\System32\winevt\Logs`.

## Buenas prácticas

- Revisa periódicamente los logs para detectar intentos de acceso no autorizado, errores y advertencias.
- Configura alertas para eventos críticos (por ejemplo, fallos de hardware o accesos sospechosos).

- Exporta y respalda los registros antes de realizar cambios importantes en el sistema.
- Utiliza filtros para localizar rápidamente eventos relevantes.
- Documenta las incidencias y las acciones correctivas tomadas.

### Ejercicio 6.8



Utiliza el Visor de eventos en Windows para buscar y listar los últimos 10 eventos registrados en el log de «Sistema». Luego, identifica cualquier evento con nivel de «Error» o «Advertencia» y describe brevemente su posible causa.

*Consejo: Puedes abrir el Visor de eventos desde el menú de inicio o ejecutando `eventvwr.msc`.*

## 6.10.2. Linux

En Linux, los registros y logs son esenciales para la monitorización, auditoría y solución de problemas. El sistema almacena información sobre el kernel, servicios, aplicaciones y accesos.

### Tipos de logs

- **Kernel:** Mensajes generados por el núcleo del sistema, accesibles con `dmesg`.
- **Sistema:** Eventos generales, arranque, apagado y errores en `/var/←log/syslog` o `/var/log/messages`.
- **Autenticación:** Intentos de acceso y cambios de usuario en `/var/←log/auth.log`.
- **Servicios:** Cada servicio puede tener su propio log (por ejemplo, `/←var/log/apache2/` para Apache).
- **Aplicaciones:** Programas específicos pueden generar logs en `/var/←log/`.

### Herramientas de consulta

- **journalctl:** Consulta el registro del sistema en distribuciones con systemd. Permite filtrar por fecha, prioridad, servicio, etc. Ejemplo:

```
1| sudo journalctl -p err -b
```

- **dmesg**: Muestra mensajes del kernel, útil para diagnosticar problemas de hardware.
- **tail, less, cat**: Permiten visualizar logs en tiempo real o examinar archivos grandes. Ejemplo:

```
1| dmesg | tail
2| tail -n 50 /var/log/syslog
```

- **grep**: Busca patrones específicos dentro de los logs para localizar errores o eventos concretos.

### Ubicación de los logs

- **/var/log/**: Carpeta principal de logs del sistema y servicios.
- Archivos destacados: **syslog**, **auth.log**, **kern.log**, **dmesg**, entre otros.

### Buenas prácticas

- Revisa los logs tras incidencias, actualizaciones o cambios de configuración.
- Configura la rotación de logs (**logrotate**) para evitar que ocupen demasiado espacio.
- Limita el acceso a los archivos de logs para proteger información sensible.
- Automatiza la monitorización y generación de alertas ante eventos críticos.
- Documenta los errores detectados y las soluciones aplicadas.

#### Ejercicio 6.9

Utiliza el comando **journalctl** en Linux para listar los últimos 10 eventos registrados en el log del sistema. Luego, identifica cualquier evento con nivel de «error» y describe brevemente su posible causa.

#### Ejercicio 6.10

Utiliza el comando **tail** en Linux para mostrar las últimas 20 líneas del archivo de log **/var/log/syslog**. Explica qué tipo de información aparece en esas líneas y cómo podrías utilizar este comando para monitorizar eventos recientes en el sistema.

## Resumen

Este capítulo ha presentado los conceptos y herramientas fundamentales para la configuración y administración de sistemas operativos, tanto en Windows como en Linux. Se han abordado la gestión de usuarios y grupos, la seguridad de cuentas y contraseñas, el control de acceso a recursos y permisos locales, así como el uso de listas de control de acceso (ACL) para una administración avanzada. Además, se han explicado la gestión de servicios y procesos, los comandos esenciales para la administración, la monitorización del sistema y el análisis de registros y logs. La correcta aplicación de estas prácticas y herramientas permite mejorar la seguridad, el rendimiento y la trazabilidad de los sistemas, facilitando una administración eficiente y proactiva.



# Conexión de sistemas en red

La conexión de sistemas en red es un pilar fundamental en la informática moderna, permitiendo la comunicación y el intercambio de información entre dispositivos, usuarios y servicios. Comprender cómo se configuran, gestionan y protegen las redes es esencial para garantizar el funcionamiento eficiente y seguro de cualquier entorno tecnológico, ya sea doméstico, empresarial o educativo.

En este capítulo se exploran los conceptos clave relacionados con la interconexión de sistemas, los protocolos de comunicación, la configuración de adaptadores y dispositivos de red, así como las tecnologías empleadas en redes cableadas e inalámbricas. Además, se abordan las buenas prácticas y herramientas necesarias para la monitorización, resolución de problemas y protección de las comunicaciones, proporcionando una base sólida para la administración y optimización de infraestructuras de red.

## 7.1. Configuración del protocolo TCP/IP en un cliente de red

La configuración del protocolo TCP/IP es esencial para que un sistema pueda comunicarse en una red local o acceder a Internet. A continuación se explican los conceptos clave y los métodos de configuración más habituales en sistemas operativos libres y propietarios.

### 7.1.1. Direcciones IP

La dirección IP identifica de forma única a cada dispositivo en una red. Existen dos versiones principales:

- **IPv4:** Formato de 32 bits, representado como cuatro números decimales separados por puntos (ejemplo: 192.168.1.10). Es el más utilizado actualmente.

- **IPv6:** Formato de 128 bits, representado como ocho grupos de cuatro dígitos hexadecimales separados por dos puntos (ejemplo: 2001:0db8:85a3:0000:0000:8a2e:0370:7334). Permite una mayor cantidad de direcciones y está en expansión.

### 7.1.2. Máscaras de subred

La máscara de subred determina qué parte de la dirección IP corresponde a la red y cuál al host. En IPv4, se representa como otra dirección (ejemplo: 255.255.255.0) o en notación CIDR (ejemplo: /24). Permite dividir redes en subredes más pequeñas y organizar mejor los dispositivos.

### 7.1.3. Puertas de enlace (Gateway)

La puerta de enlace es el dispositivo (normalmente un router) que conecta la red local con otras redes, como Internet. Los paquetes destinados a direcciones fuera de la red local se envían a la puerta de enlace para su reenvío.

### 7.1.4. Servidores de nombres de dominio (DNS)

Los servidores DNS traducen nombres de dominio (como www.ejemplo.com) en direcciones IP. Es fundamental para la navegación web y el acceso a servicios en red. Se pueden configurar servidores DNS públicos (como 8.8.8.8 de Google) o privados.

### 7.1.5. Configuración estática

Consiste en asignar manualmente todos los parámetros de red (IP, máscara, puerta de enlace, DNS) en el sistema operativo. Es útil en redes pequeñas o cuando se requiere un control preciso sobre las direcciones.

- **Windows:** Se realiza desde el Panel de control → Centro de redes → Cambiar configuración del adaptador → Propiedades → Protocolo TCP/IPv4 o TCP/IPv6.
- **Linux:** Se puede configurar editando archivos como `/etc/network/interfaces` (Debian/Ubuntu) o usando `nmcli` y `nmtui` en NetworkManager. Ejemplo por consola:

```
1 | sudo ip addr add 192.168.1.20/24 dev eth0
2 | sudo ip route add default via 192.168.1.1
```

## 7.1.6. Configuración dinámica automática (DHCP)

El protocolo DHCP permite que los dispositivos obtengan automáticamente su configuración de red (IP, máscara, puerta de enlace, DNS) desde un servidor. Es el método más común en redes domésticas y empresariales, ya que simplifica la administración.

- **Windows:** Por defecto, los adaptadores de red están configurados para obtener la IP automáticamente. Se puede verificar y modificar en las propiedades del adaptador.
- **Linux:** NetworkManager gestiona la configuración dinámica en la mayoría de distribuciones modernas. También se puede usar `dhclient`:

```
1| sudo dhclient eth0
```

## 7.1.7. Ejemplo de configuración

- **Windows (configuración estática):**

```
1| netsh interface ip set address "Ethernet" static ↪
 192.168.1.50 255.255.255.0 192.168.1.1
2| netsh interface ip set dns "Ethernet" static 8.8.8.8
```

- **Linux (configuración estática):**

```
1| sudo ip addr add 192.168.1.50/24 dev eth0
2| sudo ip route add default via 192.168.1.1
3| echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf
```

## 7.1.8. Buenas prácticas

- Documenta la configuración de red de cada dispositivo para facilitar la gestión y resolución de problemas.
- Utiliza DHCP en redes grandes para evitar conflictos y simplificar la administración.
- Asigna direcciones estáticas solo a servidores, impresoras y dispositivos críticos.
- Configura servidores DNS alternativos para mejorar la disponibilidad.
- Revisa periódicamente la configuración y realiza pruebas de conectividad (`ping`, `ipconfig`, `ifconfig`).

### Ejercicio 7.1



Cambia la configuración de tu adaptador de red para que use dirección estática en lugar de DHCP. Asigna una dirección IP dentro del rango de tu red local, una máscara de subred adecuada, la puerta de enlace predefinida y un servidor DNS. Luego, verifica la conectividad a Internet utilizando el comando `ping` hacia un sitio web conocido (por ejemplo, `ping www.google.com`).

- **Windows:** Utiliza el Panel de control o el comando `netsh` para realizar la configuración.
- **Linux:** Edita los archivos de configuración de red o utiliza `nmcli` para gestionar la conexión.

## 7.2. Ficheros de configuración de red

La configuración de red mediante ficheros permite definir de forma precisa los parámetros de las interfaces, direcciones IP, máscaras de subred, puertas de enlace y servidores DNS. El método varía según el sistema operativo y la distribución utilizada.

### 7.2.1. Linux

En Linux, la configuración de red suele gestionarse a través de archivos de texto ubicados en el sistema. Los más relevantes son:

- **/etc/network/interfaces:** Utilizado en Debian y derivados. Permite definir interfaces, direcciones IP estáticas o dinámicas, máscaras, puerta de enlace y DNS. Ejemplo:

```

1 auto eth0
2 iface eth0 inet static
3 address 192.168.1.100
4 netmask 255.255.255.0
5 gateway 192.168.1.1
6 dns-nameservers 8.8.8.8 1.1.1.1

```

- **/etc/netplan/:** En Ubuntu moderno, los ficheros YAML de Netplan permiten configurar interfaces y parámetros de red. Ejemplo:

```

1 network:
2 version: 2
3 ethernets:

```

```

4 eth0:
5 dhcp4: no
6 addresses: [192.168.1.100/24]
7 gateway4: 192.168.1.1
8 nameservers:
9 addresses: [8.8.8.8, 1.1.1.1]

```

- **/etc/hostname:** Define el nombre del equipo en la red.
- **/etc/hosts:** Permite asociar nombres de host a direcciones IP localmente.
- **/etc/resolv.conf:** Especifica los servidores DNS que utilizará el sistema. Ejemplo:

```

1 nameserver 8.8.8.8
2 nameserver 1.1.1.1

```

En distribuciones como Red Hat y CentOS, la configuración se realiza en **/etc/sysconfig/network-scripts/ifcfg-<interfaz>**.

## 7.2.2. Windows

En Windows, la configuración de red mediante ficheros es menos habitual, ya que se gestiona principalmente desde el registro y herramientas gráficas. Sin embargo, existen archivos relevantes:

- **Fichero hosts:** Permite asociar nombres de dominio a direcciones IP de forma local. Ruta: C:\Windows\System32\drivers\etc\hosts. Ejemplo:
 

```

1 192.168.1.100 servidor.local
2 8.8.8.8 google-public-dns

```
- **Fichero lmhosts:** Para resolución de nombres NetBIOS. Ruta: C:\Windows\System32\drivers\etc\lmhosts.

La configuración IP, máscara, puerta de enlace y DNS se realiza habitualmente desde el Panel de control o con comandos como `netsh`, pero los ficheros mencionados permiten personalizar la resolución de nombres local.

## 7.2.3. Buenas prácticas

- Realiza copias de seguridad de los ficheros antes de modificarlos.
- Documenta los cambios y la configuración aplicada.

- Verifica la sintaxis y el contenido tras cada cambio y reinicia los servicios de red si es necesario.
- Limita el acceso a los ficheros sensibles para proteger la seguridad de la red.

### Ejercicio 7.2



Edita el fichero de configuración de red en tu sistema operativo para asignar una dirección IP estática a tu adaptador de red. Asegúrate de incluir la máscara de subred, la puerta de enlace y los servidores DNS. Luego, reinicia el servicio de red o el equipo para aplicar los cambios y verifica la conectividad utilizando el comando `ping`.

- **Linux:** Edita `/etc/network/interfaces` o utiliza Netplan según tu distribución.
- **Windows:** Modifica el fichero `hosts` para añadir una entrada personalizada y verifica su funcionamiento.

## 7.3. Gestión de puertos

La gestión de puertos es fundamental para el funcionamiento y la seguridad de los sistemas en red. Los puertos permiten que los servicios y aplicaciones se comuniquen a través de la red, y su correcta administración ayuda a prevenir accesos no autorizados y a optimizar el tráfico.

### 7.3.1. Concepto de puerto

Un puerto es un identificador numérico (de 0 a 65535) que permite distinguir diferentes servicios en un mismo dispositivo. Por ejemplo, el puerto 80 se utiliza para HTTP y el 443 para HTTPS.

### 7.3.2. Tipos de puertos

- **Puertos bien conocidos (0-1023):** Reservados para servicios estándar (HTTP, FTP, SSH, DNS, etc.).
- **Puertos registrados (1024-49151):** Asignados a aplicaciones y servicios específicos.
- **Puertos dinámicos o privados (49152-65535):** Utilizados temporalmente por aplicaciones cliente.

### 7.3.3. Gestión de puertos en sistemas operativos

#### Windows

- **Visualización de puertos abiertos:** Utiliza el comando `netstat -an` para listar conexiones y puertos en uso.
- **Firewall de Windows:** Permite bloquear o permitir puertos y aplicaciones desde la interfaz gráfica o con comandos como `netsh advfirewall`.
- **Gestión avanzada:** Herramientas como TCPView o Resource Monitor permiten analizar el uso de puertos en tiempo real.

#### Linux

- **Visualización de puertos abiertos:** Comandos como `netstat -tuln`, `ss -tuln` y `lsof -i` muestran los puertos y servicios activos.
- **Firewall (iptables, ufw, firewalld):** Permiten definir reglas para permitir o bloquear puertos y servicios.
- **Gestión avanzada:** Herramientas como `nmap` permiten escanear puertos en equipos locales o remotos.

### 7.3.4. Ejemplos de comandos

#### Windows

- **Visualizar puertos abiertos:** `netstat -an` muestra todas las conexiones y puertos en uso.

- **Abrir un puerto en el firewall:**

```
1| netsh advfirewall firewall add rule name="Abrir puerto 8080" dir=in action=allow protocol=TCP localport=8080
```

- **Mostrar reglas del firewall:**

```
1| netsh advfirewall firewall show rule name=all
```

- **Cerrar un puerto:**

```
1| netsh advfirewall firewall delete rule name="Abrir puerto 8080"
```

## Linux

- Visualizar puertos abiertos:

```
1 ss -tuln
2 netstat -tuln
3 lsof -i
```

- Abrir un puerto con UFW (Uncomplicated Firewall):

```
1 sudo ufw allow 8080/tcp
```

- Mostrar reglas de UFW:

```
1 sudo ufw status numbered
```

- Cerrar un puerto con UFW:

```
1 sudo ufw delete allow 8080/tcp
```

- Abrir un puerto con iptables:

```
1 sudo iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
```

- Cerrar un puerto con iptables:

```
1 sudo iptables -D INPUT -p tcp --dport 8080 -j ACCEPT
```

- Listar reglas de iptables:

```
1 sudo iptables -L -n
```

### 7.3.5. Buenas prácticas

- Mantén cerrados los puertos que no sean necesarios para reducir la superficie de ataque.
- Revisa periódicamente los puertos abiertos y los servicios asociados.
- Utiliza firewalls para controlar el acceso y registrar intentos de conexión sospechosos.
- Documenta los cambios realizados en la configuración de puertos y servicios.
- Realiza auditorías de seguridad con herramientas como `nmap` para detectar vulnerabilidades.

### Ejercicio 7.3



Realiza las siguientes tareas en tu sistema operativo:

1. Lista todos los puertos abiertos y los servicios asociados utilizando los comandos adecuados.
2. Abre un puerto específico (por ejemplo, el 8080) y verifica que está accesible desde otro equipo de la red.
3. Cierra el puerto abierto y comprueba que ya no está disponible.
4. Documenta los comandos utilizados y captura la salida de cada paso.

*Incluye ejemplos para Windows (netstat, netsh) y Linux (ss, ufw, iptables).*

## 7.4. Resolución de problemas de conectividad en sistemas operativos en red

La resolución de problemas de conectividad es esencial para garantizar el correcto funcionamiento de los sistemas en red. Los fallos pueden deberse a configuraciones incorrectas, problemas físicos, errores de software o restricciones de seguridad.

### 7.4.1. Herramientas de diagnóstico

Existen diversas herramientas y comandos que permiten identificar y solucionar problemas de red:

**ping** Comprueba la conectividad entre el equipo local y otro dispositivo o servidor. Permite verificar si una dirección IP o dominio responde correctamente.

Muestra la ruta que siguen los paquetes hasta el destino, identificando posibles puntos de fallo en la red.

**traceroute** (~~tracert~~  
**Windows**) / **ifconfig** (Windows) / **ip** (Linux) Muestran la configuración de red actual, incluyendo direcciones IP, máscaras, puertas de enlace y estado de las interfaces.

**netstat** Permite visualizar las conexiones de red activas, puertos abiertos y estadísticas de red.

## 7 Conexión de sistemas en red

**nslookup / dig** Diagnóstican problemas de resolución de nombres DNS.

**arp** Muestra y gestiona la caché ARP, útil para detectar conflictos de direcciones IP en la red local.

**route** Permite visualizar y modificar la tabla de rutas del sistema.

**nmap** Escanea puertos y servicios en equipos remotos para detectar accesibilidad y posibles vulnerabilidades.

### Ejemplos de comandos

#### Windows:

```
1 ipconfig /all # Muestra la configuración completa de red
2 ping 8.8.8.8 # Prueba la conectividad con un servidor DNS ↵
 público
3 tracert www.google.com # Rastrea la ruta hasta Google
4 nslookup www.google.com # Consulta DNS
5 netstat -an # Muestra conexiones y puertos
6 arp -a # Muestra la caché ARP
7 route print # Muestra la tabla de rutas
```

#### Linux:

```
1 ip a # Muestra la configuración de red
2 ping 8.8.8.8 # Prueba la conectividad con un servidor DNS ↵
 público
3 traceroute www.google.com # Rastrea la ruta hasta Google
4 dig www.google.com # Consulta DNS
5 netstat -tuln # Muestra conexiones y puertos
6 arp -n # Muestra la caché ARP
7 route -n # Muestra la tabla de rutas
8 nmap -p 80,443 8.8.8.8 # Escanea puertos
```

### 7.4.2. Pasos para la resolución de problemas

1. Verifica la conexión física (cables, adaptadores, estado de la interfaz).
2. Comprueba la configuración IP, máscara de subred, puerta de enlace y DNS.
3. Utiliza ping para probar la conectividad con la puerta de enlace y servidores externos.
4. Revisa las reglas del firewall y los puertos abiertos.
5. Analiza la tabla de rutas y la resolución de nombres.

6. Consulta los registros del sistema y los logs de red para identificar errores.
7. Reinicia los servicios de red o el equipo si es necesario.

### Ejercicio 7.4



Supón que tu equipo no tiene acceso a Internet. Realiza los siguientes pasos para diagnosticar y resolver el problema:

- Verifica la configuración IP con `ipconfig` (Windows) o `ip a` (Linux).
- Comprueba la conectividad con la puerta de enlace usando `pingng`.
- Utiliza `tracert` o `traceroute` para identificar dónde se interrumpe la conexión.
- Revisa la configuración de DNS con `nslookup` o `dig`.
- Consulta las reglas del firewall y los puertos abiertos con `netstat` o `ufw status`.
- Documenta los comandos utilizados y las soluciones aplicadas.

## 7.5. Herramientas gráficas utilizadas en sistemas operativos

En los sistemas operativos, tanto libres como propietarios, existen herramientas gráficas y comandos que facilitan la gestión y diagnóstico de la red. A continuación se describen algunas de las más comunes y su uso.

### 7.5.1. Windows

**Centro de redes y recursos compartidos** Permite visualizar el estado de la red, configurar adaptadores, cambiar opciones de uso compartido y solucionar problemas.

**Administrador de dispositivos** Muestra los adaptadores de red instalados y permite actualizar, deshabilitar o desinstalar controladores.

**Firewall de Windows** Interfaz para gestionar reglas de entrada y salida, así como permitir o bloquear aplicaciones.

**Solucionador de problemas de red** Asistente que detecta y repara automáticamente problemas comunes de conectividad.

### 7.5.2. Linux

**NetworkManager** Utilidad presente en la mayoría de entornos gráficos (GNOME, KDE) para gestionar conexiones cableadas, inalámbricas, VPN y más.

**nm-connection-editor** Editor avanzado de conexiones de red.

**nmtui** Interfaz de texto interactiva para NetworkManager.

**firewall-config** Herramienta gráfica para gestionar firewalld en distribuciones como Fedora y CentOS.

### 7.5.3. Buenas prácticas

- Utiliza herramientas gráficas para tareas habituales y comandos para diagnósticos avanzados.
- Documenta los cambios realizados en la configuración de red.
- Realiza pruebas de conectividad tras modificar parámetros.
- Mantén actualizadas las herramientas y utilidades de red.

#### Ejercicio 7.5



Utiliza una herramienta gráfica y un comando en tu sistema operativo para visualizar la configuración de red y comprobar la conectividad. Documenta los pasos realizados y la información obtenida.

- **Windows:** Usa el Centro de redes y recursos compartidos y el comando `ipconfig`.
- **Linux:** Usa NetworkManager (o nmtui) y el comando `ip a a`.

## 7.6. Monitorización de redes

La monitorización de redes permite supervisar el estado, el rendimiento y la seguridad de los sistemas conectados, facilitando la detección temprana de incidencias y la optimización de recursos. La monitorización puede ser proactiva

(anticipando problemas antes de que afecten a los usuarios) o reactiva (respondiendo a incidencias detectadas). Es fundamental en entornos empresariales y educativos para garantizar la disponibilidad y el rendimiento de los servicios de red.

Entre los aspectos clave a monitorizar se encuentran:

- **Disponibilidad de dispositivos y servicios:** Comprobación periódica de que los equipos y servicios de red están accesibles.
- **Consumo de ancho de banda:** Identificación de picos de tráfico y posibles saturaciones.
- **Latencia y pérdida de paquetes:** Medición de tiempos de respuesta y detección de problemas de calidad de servicio.
- **Seguridad:** Detección de accesos no autorizados, intentos de intrusión y tráfico sospechoso.
- **Estado de los enlaces y hardware:** Supervisión de cables, switches, routers y adaptadores para anticipar fallos físicos.

La elección de la herramienta de monitorización depende del tamaño de la red, los requisitos de seguridad y el nivel de detalle necesario. En redes pequeñas pueden bastar utilidades básicas, mientras que en redes empresariales se requieren soluciones avanzadas con alertas y reportes automáticos.

### 7.6.1. Herramientas de monitorización

**Wireshark** Analizador de protocolos que captura y muestra paquetes en tiempo real, permitiendo analizar el tráfico y detectar anomalías.

**Nagios** Plataforma de monitorización de red y sistemas que alerta sobre fallos y problemas de rendimiento.

**Zabbix** Solución integral para monitorizar redes, servidores y aplicaciones, con gráficos y alertas personalizables.

**Netdata** Herramienta ligera para monitorizar en tiempo real el rendimiento de sistemas y redes.

**nload, iftop, iptraf** Utilidades en consola para visualizar el tráfico de red en tiempo real.

**Task Manager / Resource Monitor (Windows)** Permiten ver el uso de red por procesos y aplicaciones.

## 7 Conexión de sistemas en red

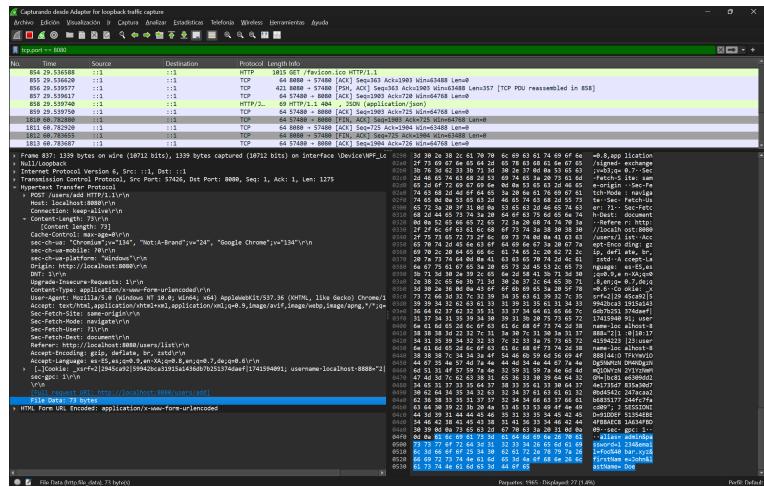


Figura 7.1: Wireshark, una herramienta popular para la captura y análisis de tráfico de red.

### 7.6.2. Ejemplos de uso

**iftop:**

```
1| sudo iftop -i eth0
```

Muestra el tráfico entrante y saliente en tiempo real por pares de direcciones IP.  
**Netdata:**

```
1| sudo apt install netdata
2| sudo systemctl start netdata
```

Accede a la interfaz web para visualizar métricas de red y sistema.

**Resource Monitor (Windows):** Ejecuta `resmon` desde el menú inicio para ver el uso de red por procesos.

### 7.6.3. Buenas prácticas

- Monitoriza de forma continua los sistemas críticos y el tráfico de red.
- Configura alertas para detectar caídas de servicios, saturación de ancho de banda o accesos no autorizados.
- Analiza los registros y las tendencias para anticipar problemas y planificar mejoras.
- Protege los datos de monitorización y limita el acceso a las herramientas.
- Documenta los procedimientos y las incidencias detectadas.

## Ejercicio 7.6



Instala y utiliza una herramienta de monitorización de red en tu sistema operativo. Realiza una captura de tráfico, identifica los principales protocolos utilizados y documenta los resultados obtenidos.

- **Windows:** Usa Resource Monitor o Wireshark.
- **Linux:** Usa iftop, nload o Wireshark.

## 7.7. Protocolos TCP/IP

El conjunto de protocolos TCP/IP constituye la base de las comunicaciones en redes modernas, permitiendo la transmisión de datos entre dispositivos de manera fiable y eficiente. TCP/IP es un modelo en capas que define cómo se estructuran y gestionan las comunicaciones en red.

### 7.7.1. Modelo de capas TCP/IP

El modelo TCP/IP se compone de cuatro capas principales:

- **Capa de acceso a la red (Enlace):** Gestiona la transmisión física de datos entre dispositivos en la misma red local. Incluye protocolos como Ethernet, Wi-Fi y PPP.
- **Capa de red:** Encargada del direccionamiento y encaminamiento de los paquetes entre redes. El protocolo principal es IP (Internet Protocol), junto con ICMP y ARP.
- **Capa de transporte:** Proporciona la comunicación fiable o no fiable entre aplicaciones. Los protocolos más importantes son TCP (Transmission Control Protocol) y UDP (User Datagram Protocol).
- **Capa de aplicación:** Incluye los protocolos utilizados por las aplicaciones para intercambiar datos, como HTTP, FTP, SMTP, DNS, SSH, entre otros.

### 7.7.2. Principales protocolos TCP/IP

- **IP (Internet Protocol):** Protocolo de red que se encarga de direccionar y enviar paquetes entre dispositivos. Existen dos versiones: IPv4 e IPv6.

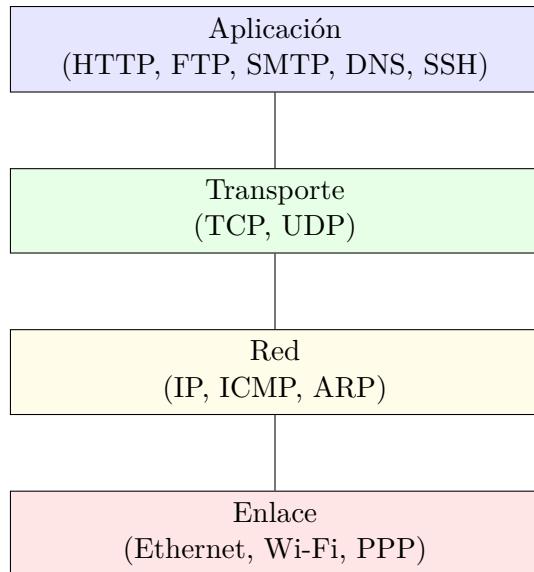


Figura 7.2: Modelo de capas TCP/IP y ejemplos de protocolos en cada capa.

- **TCP (Transmission Control Protocol):** Protocolo orientado a la conexión que garantiza la entrega fiable y ordenada de los datos.
- **UDP (User Datagram Protocol):** Protocolo sin conexión, más rápido pero menos fiable, utilizado en aplicaciones donde la velocidad es prioritaria (streaming, VoIP).
- **ICMP (Internet Control Message Protocol):** Utilizado para enviar mensajes de control y error, como los generados por el comando `ping`.
- **ARP (Address Resolution Protocol):** Permite resolver direcciones IP en direcciones físicas (MAC) dentro de la red local.
- **DHCP (Dynamic Host Configuration Protocol):** Asigna automáticamente direcciones IP y otros parámetros de red a los dispositivos.
- **DNS (Domain Name System):** Traduce nombres de dominio en direcciones IP.
- **HTTP/HTTPS, FTP, SMTP, SSH, etc.:** Protocolos de aplicación para servicios web, transferencia de archivos, correo electrónico y acceso remoto seguro.

### 7.7.3. Funcionamiento básico

Cuando un dispositivo envía datos a través de la red, estos se encapsulan sucesivamente en cada capa del modelo TCP/IP. Por ejemplo, al acceder a una página web:

1. La aplicación (navegador) utiliza HTTP para solicitar la página.
2. Los datos se envían mediante TCP, que garantiza la entrega fiable.
3. TCP utiliza IP para direccionar los paquetes al servidor.
4. IP utiliza Ethernet o Wi-Fi para transmitir los paquetes físicamente.

### 7.7.4. Ejemplo de comunicación

```

1 # Comando para consultar una web usando curl (HTTP sobre TCP/IP)
2 curl http://www.ejemplo.com
3
4 # Comando para consultar la resolución DNS
5 nslookup www.ejemplo.com
6
7 # Comando para comprobar la conectividad con ICMP
8 ping 8.8.8.8

```

### 7.7.5. Buenas prácticas

- Mantén actualizados los protocolos y servicios para evitar vulnerabilidades.
- Utiliza protocolos seguros (HTTPS, SSH) para proteger la información.
- Configura correctamente los parámetros de red (IP, máscara, puerta de enlace, DNS).
- Monitoriza el tráfico y los servicios para detectar anomalías.

#### Ejercicio 7.7

Investiga y describe el funcionamiento de los protocolos TCP y UDP. Realiza una prueba de transferencia de archivos utilizando ambos protocolos (por ejemplo, con `scp` para TCP y `netcat` para UDP) y compara los resultados en cuanto a fiabilidad y velocidad.



## 7.8. Configuración de los adaptadores de red

La configuración de los adaptadores de red permite que los dispositivos se conecten correctamente a la red y accedan a los servicios disponibles. El proceso varía según el sistema operativo y el tipo de adaptador (cableado, inalámbrico, virtual).

### 7.8.1. Windows

En Windows, la gestión de adaptadores de red se realiza principalmente desde el Panel de control y el Administrador de dispositivos. Las tareas habituales incluyen:

- **Visualizar adaptadores:** Accede a *Centro de redes y recursos compartidos* → *Cambiar configuración del adaptador* para ver todos los adaptadores instalados.
- **Configurar parámetros:** Haz clic derecho sobre el adaptador y selecciona *Propiedades* para modificar la configuración TCP/IP, habilitar o deshabilitar el adaptador, y gestionar protocolos.
- **Actualizar controladores:** Desde el *Administrador de dispositivos*, puedes actualizar, deshabilitar o desinstalar los controladores del adaptador de red.
- **Diagnóstico y solución de problemas:** Utiliza el solucionador de problemas de red para detectar y reparar incidencias comunes.

### 7.8.2. Linux

En Linux, los adaptadores de red se gestionan mediante herramientas gráficas (NetworkManager) y comandos en consola. Las tareas principales son:

- **Listar adaptadores:** Usa `ip link` o `ifconfig -a` para ver los adaptadores disponibles.
- **Configurar adaptadores:** Edita archivos como `/etc/network/interfaces` o utiliza Netplan (`/etc/netplan/n/`) en Ubuntu moderno. NetworkManager permite gestionar conexiones desde la interfaz gráfica o con `nmcli` y `nmtui`.
- **Activar/desactivar adaptadores:** Comandos como `ip link set ↪ eth0 up/down` permiten habilitar o deshabilitar interfaces.

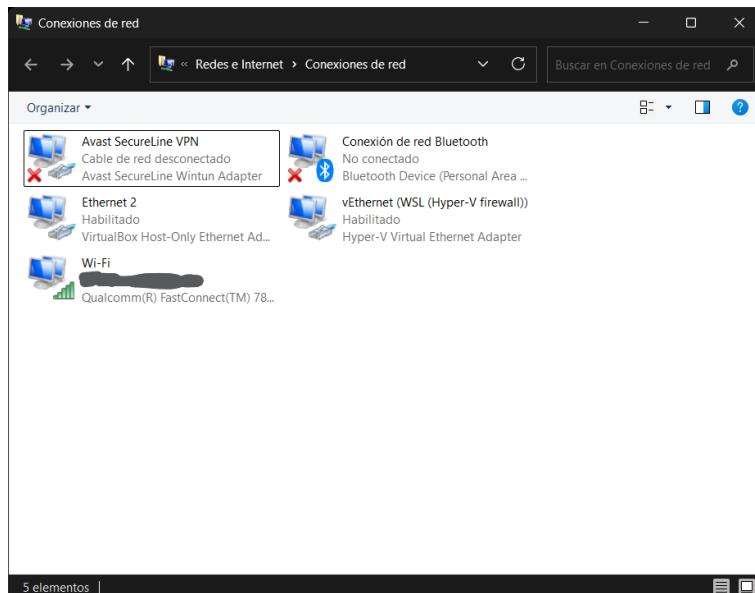


Figura 7.3: Configuración de adaptadores de red en Windows

- **Actualizar controladores:** La instalación y actualización de drivers suele realizarse mediante el gestor de paquetes de la distribución.

### 7.8.3. Configuración de adaptadores inalámbricos

La configuración de adaptadores Wi-Fi implica seleccionar la red, introducir la clave de acceso y definir parámetros de seguridad (WPA2, WPA3). En Windows y Linux existen asistentes gráficos para facilitar la conexión, aunque también puede realizarse por consola.

### 7.8.4. Adaptadores virtuales

Los adaptadores virtuales permiten crear interfaces de red adicionales para máquinas virtuales, VPN o redes internas. Se configuran desde el software de virtualización (VirtualBox, VMware) o mediante utilidades específicas en el sistema operativo.

### 7.8.5. Buenas prácticas

- Mantén los controladores de los adaptadores actualizados para garantizar compatibilidad y seguridad.

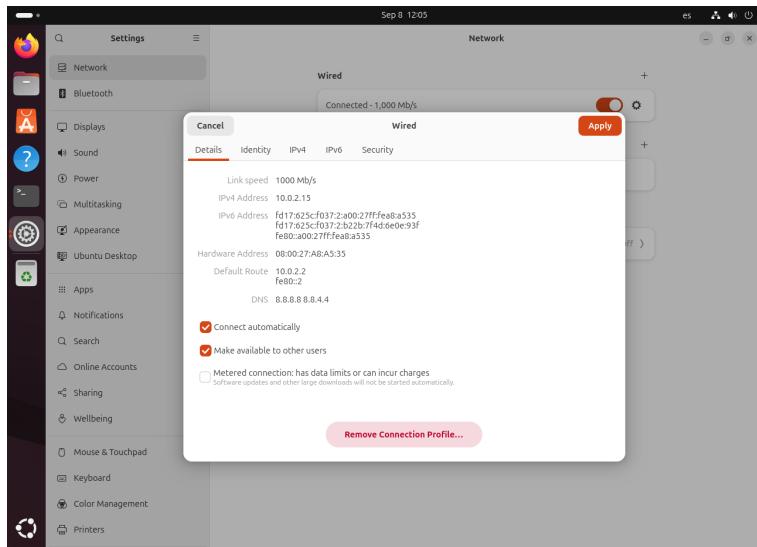


Figura 7.4: Configuración de adaptadores de red en Ubuntu

- Deshabilita los adaptadores que no utilices para reducir riesgos de seguridad.
- Documenta la configuración aplicada y realiza pruebas de conectividad tras cada cambio.
- Utiliza nombres descriptivos para las interfaces en sistemas con múltiples adaptadores.

### Ejercicio 7.8

Accede a la configuración de los adaptadores de red en tu sistema operativo. Cambia la configuración de uno de ellos (por ejemplo, asigna una IP estática o modifica la red Wi-Fi utilizada), verifica la conectividad y documenta los pasos realizados.

- **Windows:** Usa el Panel de control o el Administrador de dispositivos.
- **Linux:** Utiliza NetworkManager, `nmcli`, `nmtui` o edita los archivos de configuración.

## 7.9. Interconexión de redes

La interconexión de redes permite que diferentes segmentos de red se comuniquen entre sí, ya sea dentro de una organización o a través de Internet. Para lograrlo, se utilizan adaptadores de red y dispositivos de interconexión, cada uno con funciones específicas.

### 7.9.1. Adaptadores de red

Los adaptadores de red (tarjetas de red) son dispositivos físicos o virtuales que permiten a los equipos conectarse a una red cableada (Ethernet) o inalámbrica (Wi-Fi). Cada adaptador tiene una dirección MAC única y puede configurarse con parámetros de red (IP, máscara, puerta de enlace).

En los siguientes puntos de este capítulo, entraremos en más detalle sobre los tipos de redes, dispositivos de interconexión y protocolos utilizados.

### 7.9.2. Enrutamiento

El enrutamiento es el proceso mediante el cual los routers determinan el camino que deben seguir los paquetes de datos para llegar a su destino. Los routers utilizan tablas de enrutamiento, que pueden configurarse de forma estática o dinámica mediante protocolos como RIP, OSPF o BGP.

- **Enrutamiento estático:** Las rutas se configuran manualmente. Es sencillo pero poco flexible.
- **Enrutamiento dinámico:** Los routers intercambian información y actualizan sus tablas automáticamente, adaptándose a cambios en la red.

### 7.9.3. Ejemplo de configuración de enrutamiento

**Linux (añadir una ruta estática):**

```
1| sudo ip route add 192.168.2.0/24 via 192.168.1.1
```

**Windows (añadir una ruta estática):**

```
1| route add 192.168.2.0 mask 255.255.255.0 192.168.1.1
```

### 7.9.4. Buenas prácticas

- Documenta la topología y las rutas configuradas en la red.
- Utiliza enrutamiento dinámico en redes grandes para facilitar la gestión.

- Protege los dispositivos de interconexión con contraseñas y actualizaciones periódicas.
- Realiza pruebas de conectividad tras modificar la configuración de enrutamiento.

### Ejercicio 7.9



Configura una ruta estática en tu sistema operativo para permitir la comunicación con una subred diferente. Verifica la conectividad utilizando el comando `ping` y documenta los pasos realizados.

- **Windows:** Usa el comando `route add`.
- **Linux:** Usa el comando `ip route adddd`.

## 7.10. Redes cableadas

Las redes cableadas utilizan medios físicos, como cables de cobre o fibra óptica, para transmitir datos entre dispositivos. Son ampliamente empleadas en entornos domésticos, empresariales y educativos por su fiabilidad, velocidad y seguridad.

### 7.10.1. Tipos y características

**Ethernet** Es el estándar más común para redes cableadas. Utiliza cables de par trenzado (Cat5e, Cat6, Cat7) y conectores RJ-45. Permite velocidades desde 10 Mbps hasta 10 Gbps o más.

**Fibra óptica** Utiliza hilos de vidrio o plástico para transmitir datos mediante pulsos de luz. Ofrece altas velocidades (hasta 100 Gbps), baja latencia y gran alcance, siendo ideal para enlaces troncales y redes de campus.

**Cable coaxial** Antiguamente usado en redes Ethernet (10BASE2, 10BASE5), hoy en día se emplea principalmente en televisión por cable y acceso a Internet.

### 7.10.2. Adaptadores de red

Los adaptadores de red (NIC, Network Interface Card) permiten a los dispositivos conectarse a la red cableada. Pueden ser internos (integrados en la placa base) o externos (USB, PCIe). Cada adaptador tiene una dirección MAC única.

### 7.10.3. Dispositivos de interconexión

**Switch (comutador)** Dispositivo que conecta múltiples equipos en una red local, gestionando el tráfico de datos y optimizando la comunicación.

**Router (enrutador)** Permite la conexión entre diferentes redes y el acceso a Internet. Gestiona el tráfico entre subredes y proporciona funciones de seguridad.

**Bridge (puente)** Une dos segmentos de red, filtrando el tráfico según la dirección MAC.

**Patch panel** Panel de conexiones que facilita la organización y gestión de cables en instalaciones profesionales.

### 7.10.4. Ventajas y desventajas

- **Ventajas:** Alta velocidad, baja latencia, mayor seguridad y estabilidad frente a interferencias.
- **Desventajas:** Requiere cableado físico, instalación más compleja y menor flexibilidad que las redes inalámbricas.

### 7.10.5. Buenas prácticas

- Utiliza cables y conectores de calidad adecuados a la velocidad requerida.
- Etiqueta y organiza el cableado para facilitar el mantenimiento.
- Protege los cables de daños físicos y fuentes de interferencia.
- Documenta la topología y los dispositivos conectados.

#### Ejercicio 7.10

Dibuja la topología de una red cableada típica en un aula o empresa, indicando los dispositivos principales (adaptadores, switch, router) y el tipo de cable utilizado. Explica las ventajas de este tipo de red frente a una red inalámbrica.



## 7.11. Redes inalámbricas

Las redes inalámbricas permiten la conexión de dispositivos sin necesidad de cables físicos, utilizando ondas de radio para transmitir datos. Son ampliamente utilizadas en entornos domésticos, empresariales y públicos por su flexibilidad y facilidad de instalación.

### 7.11.1. Tipos y características

**Wi-Fi (IEEE 802.11)** Es el estándar más común para redes inalámbricas locales. Opera en las bandas de 2.4 GHz y 5 GHz, con velocidades que varían según la versión (802.11n, 802.11ac, 802.11ax).

**Bluetooth** Utilizado para conexiones de corto alcance entre dispositivos personales (teclados, auriculares, teléfonos).

**WiMAX** Tecnología para acceso inalámbrico de banda ancha en áreas extensas.

**Redes móviles (4G/5G)** Permiten la conexión a Internet a través de la infraestructura de telefonía móvil.

### 7.11.2. Adaptadores de red

Los adaptadores inalámbricos (Wi-Fi) pueden ser internos (integrados en la placa base o en portátiles) o externos (USB, PCIe). Cada adaptador tiene una dirección MAC única y soporta diferentes estándares y bandas de frecuencia.

### 7.11.3. Configuración y gestión de adaptadores inalámbricos

La configuración de adaptadores inalámbricos implica seleccionar la red Wi-Fi, introducir la clave de acceso y definir parámetros de seguridad. En Windows, esto se realiza desde el Centro de redes y recursos compartidos o el ícono de red en la barra de tareas. En Linux, se puede utilizar NetworkManager, `nmcli`, `nmtui` o editar archivos de configuración según la distribución.

- **Windows:** Haz clic en el ícono de red, selecciona la red Wi-Fi deseada, introduce la contraseña y verifica la conexión.
- **Linux:** Usa NetworkManager desde la interfaz gráfica o ejecuta comandos como:

```
1| nmcli device wifi list
```

```
2| nmcli device wifi connect SSID password CONTRASEÑA
```

En ambos sistemas, es posible gestionar redes preferidas, olvidar redes antiguas y configurar parámetros avanzados como IP estática, DNS o seguridad adicional.

### Ejercicio 7.11



Configura la conexión a una red Wi-Fi en tu sistema operativo. Cambia la contraseña de acceso, verifica la conectividad y documenta los pasos realizados.

- **Windows:** Utiliza el asistente de conexión Wi-Fi.
- **Linux:** Usa NetworkManager, `nmcli` o `nmtui`.

## 7.11.4. Dispositivos de interconexión

**Punto de acceso (Access Point)** Dispositivo que permite la conexión de equipos inalámbricos a la red local. Puede estar integrado en routers o ser independiente.

**Router inalámbrico** Combina funciones de router y punto de acceso, gestionando la conexión a Internet y la red Wi-Fi.

**Repetidor/extensor** Amplía la cobertura de la red inalámbrica en áreas donde la señal es débil.

## 7.11.5. Seguridad

La seguridad en redes inalámbricas es fundamental para evitar accesos no autorizados y proteger la información transmitida. Las principales medidas incluyen:

- Utilizar protocolos de cifrado robustos (WPA2, WPA3).
- Cambiar la contraseña por defecto del Wi-Fi y del administrador del router.
- Ocultar el SSID o limitar la difusión si es posible.
- Filtrar el acceso por dirección MAC.
- Actualizar el firmware de los dispositivos regularmente.

### 7.11.6. Ventajas y desventajas

- **Ventajas:** Flexibilidad, facilidad de instalación, movilidad de los dispositivos.
- **Desventajas:** Menor velocidad y estabilidad que las redes cableadas, mayor exposición a interferencias y riesgos de seguridad.

#### Ejercicio 7.12



Realiza una búsqueda de redes inalámbricas disponibles en tu entorno y documenta sus características (nombre, tipo de cifrado, intensidad de señal). Conéctate a una red Wi-Fi segura, verifica la conectividad y explica las medidas de seguridad implementadas.

## 7.12. Seguridad de comunicaciones

La seguridad en las comunicaciones de red es esencial para proteger la integridad, confidencialidad y disponibilidad de los datos transmitidos entre dispositivos. Los riesgos incluyen la interceptación de información, ataques de suplantación, acceso no autorizado y manipulación de datos.

### 7.12.1. Principales amenazas

- **Intercepción de datos:** Un atacante puede capturar paquetes en tránsito mediante técnicas como sniffing.
- **Suplantación de identidad (spoofing):** Consiste en falsificar direcciones IP, MAC o credenciales para acceder a recursos restringidos.
- **Ataques de denegación de servicio (DoS):** Buscan saturar la red o los servicios para impedir su funcionamiento.
- **Acceso no autorizado:** Usuarios o dispositivos sin permisos pueden acceder a información sensible.
- **Manipulación de datos:** Alteración de la información transmitida entre origen y destino.

### 7.12.2. Medidas de protección

- **Cifrado de datos:** Utiliza protocolos seguros como HTTPS, SSH, VPN y WPA2/WPA3 en redes inalámbricas para proteger la información.

- **Autenticación:** Implementa mecanismos de autenticación robustos (contraseñas seguras, certificados digitales, autenticación multifactor).
- **Firewalls:** Configura firewalls para controlar el tráfico y bloquear accesos no autorizados.
- **Actualización de sistemas:** Mantén el software y firmware actualizados para corregir vulnerabilidades.
- **Segmentación de red:** Separa redes internas, externas y de invitados para limitar el alcance de posibles ataques.
- **Monitorización y auditoría:** Supervisa el tráfico y los accesos para detectar actividades sospechosas.

### 7.12.3. Protocolos y tecnologías de seguridad

**SSL/TLS** Proporciona cifrado y autenticación en comunicaciones web (HTTPS).

**IPsec** Permite cifrar y autenticar el tráfico IP, utilizado en VPNs.

**SSH** Protocolo seguro para acceso remoto y transferencia de archivos.

**802.1X** Control de acceso a la red mediante autenticación de usuarios y dispositivos.

**VPN (Virtual Private Network)** Crea túneles cifrados para conectar redes de forma segura a través de Internet.

### 7.12.4. Buenas prácticas

- Utiliza contraseñas fuertes y cámbialas periódicamente.
- Configura el cifrado en todas las comunicaciones sensibles.
- Limita los servicios y puertos expuestos a Internet.
- Realiza copias de seguridad y planes de recuperación ante incidentes.
- Educa a los usuarios sobre riesgos y medidas de seguridad.

#### Ejercicio 7.13

Configura una conexión segura en tu sistema operativo utilizando SSH o VPN. Realiza una transferencia de archivos cifrada y verifica que la comu-



nicación está protegida. Documenta los pasos realizados y las herramientas utilizadas.

- **Windows:** Usa una aplicación de VPN o el cliente SSH de PowerShell.
- **Linux:** Utiliza `ssh`, `scp` o configura una VPN con OpenVPN.

Realiza la misma transferencia sin cifrado (por ejemplo, usando FTP) y compara la seguridad de ambas comunicaciones.

*Nota: Asegúrate de realizar estas pruebas en un entorno controlado y seguro para evitar riesgos de seguridad. Utiliza una herramienta de captura de paquetes (como Wireshark) para observar la diferencia entre las comunicaciones cifradas y no cifradas.*

## 7.13. Tecnologías de acceso a redes de área extensa

Las redes de área extensa (WAN, Wide Area Network) permiten la interconexión de redes locales (LAN) situadas en ubicaciones geográficas diferentes, facilitando la comunicación entre sedes de una empresa, campus universitarios o el acceso a Internet. Para ello, existen diversas tecnologías y medios de acceso, cada una con características específicas en cuanto a velocidad, coste y cobertura.

### 7.13.1. Principales tecnologías de acceso WAN

**ADSL/VDSL** Utilizan la infraestructura de líneas telefónicas para ofrecer acceso a Internet de banda ancha. Son comunes en entornos domésticos y pequeñas empresas.

**Fibra óptica** Proporciona altas velocidades y baja latencia. Es la tecnología preferida para conexiones empresariales y enlaces troncales.

**Redes móviles (4G/5G)** Permiten el acceso a Internet mediante la infraestructura de telefonía móvil, ofreciendo movilidad y cobertura en áreas extensas.

**Radioenlaces** Utilizan ondas de radio para conectar ubicaciones distantes, especialmente en zonas rurales o donde no llega el cableado.

**Satélite** Ofrece conectividad en lugares remotos donde no existen otras alternativas, aunque con mayor latencia y coste.

**MPLS (Multiprotocol Label Switching)** Tecnología utilizada por operadores para crear redes privadas virtuales (VPN) de alto rendimiento entre sedes.

**VPN (Virtual Private Network)** Permite conectar redes locales a través de Internet de forma segura, utilizando túneles cifrados.

**RDSI (ISDN) y líneas dedicadas (E1/T1)** Tecnologías tradicionales para enlaces punto a punto, hoy en día en desuso frente a alternativas más modernas.

### 7.13.2. Dispositivos y configuración

Para acceder a una WAN, se emplean dispositivos como routers, módems, antenas y firewalls. La configuración suele implicar:

- Definir parámetros de conexión (usuario, contraseña, dirección IP, VLAN).
- Configurar el tipo de acceso (PPPoE, DHCP, IP estática).
- Establecer reglas de seguridad y segmentación de red.
- Monitorizar el estado del enlace y el tráfico.

### 7.13.3. Buenas prácticas

- Elige la tecnología de acceso adecuada según las necesidades de velocidad, fiabilidad y coste.
- Protege la conexión WAN con firewalls y cifrado.
- Documenta la configuración y los dispositivos utilizados.
- Realiza pruebas de rendimiento y monitoriza la disponibilidad del enlace.
- Mantén actualizados los equipos y el firmware.

#### Ejercicio 7.14



Investiga qué tecnologías de acceso WAN están disponibles en tu entorno (ADSL, fibra, 4G/5G, radioenlace, satélite). Elabora una tabla comparativa con sus características principales (velocidad, latencia, coste, cobertura) y explica cuál sería la más adecuada para una empresa con varias sedes.

### Ejercicio 7.15



Calcula la velocidad de descarga necesaria para que un disco duro de 1 TB se descargue en 2 horas. Expresa el resultado en Mbps (megabits por segundo).

Después, calcula cuál es la velocidad de transferencia, si trasladamos físicamente el disco duro en un vehículo que se desplaza a una media de 90 km/h, hasta la otra sede de la empresa a 30 km de distancia. Expresa el resultado en Mbps (megabits por segundo).

¿Y si el disco duro fuese llevado por una paloma<sup>a</sup>, que recorriese esa distancia a una media de 40 km/h?

¿Qué medio es más eficiente? ¿Y el menos eficiente?

*Nota: considera que 1 kilobyte = 1.000 bytes y 1 megabit = 1.000.000 bits.*

---

<sup>a</sup>Visita [https://en.wikipedia.org/wiki/IP\\_over\\_Avian\\_Carriers](https://en.wikipedia.org/wiki/IP_over_Avian_Carriers) para más información sobre este supuesto.

## Resumen

En este capítulo se han abordado los conceptos fundamentales relacionados con la conexión de sistemas en red, la configuración de protocolos TCP/IP, la gestión de puertos y la resolución de problemas de conectividad en distintos sistemas operativos. Se han presentado los principales ficheros y herramientas de configuración, tanto en Linux como en Windows, así como buenas prácticas para garantizar la seguridad y el correcto funcionamiento de la red.

También se han descrito los modelos y protocolos TCP/IP, la configuración de adaptadores de red, la interconexión de redes y los dispositivos implicados. Se han comparado las características de las redes cableadas e inalámbricas, destacando sus ventajas y desventajas, y se han analizado las tecnologías de acceso a redes de área extensa (WAN).

Por último, se ha hecho hincapié en la importancia de la monitorización y la seguridad de las comunicaciones, presentando herramientas y medidas de protección para prevenir amenazas y garantizar la integridad de los datos. El dominio de estos conocimientos es esencial para administrar, proteger y optimizar cualquier sistema informático conectado a una red.

# Gestión de recursos en una red

La gestión de recursos en una red es fundamental para asegurar el funcionamiento eficiente, seguro y colaborativo de los sistemas informáticos de una organización. En este capítulo se abordan los conceptos y técnicas esenciales para administrar los recursos compartidos, como archivos, impresoras y aplicaciones, así como la configuración de permisos, la protección de datos y la implantación de servidores y dominios. Además, se presentan las mejores prácticas y herramientas disponibles en los principales sistemas operativos, permitiendo al administrador garantizar la seguridad, disponibilidad y control de los recursos en el entorno de red.

## 8.1. Permisos y derechos

Los permisos y derechos en una red determinan qué usuarios o grupos pueden acceder y manipular recursos compartidos, como carpetas, archivos o impresoras. Existen dos tipos principales de permisos:

- **Permisos de red:** Controlan el acceso a recursos compartidos a través de la red. Se configuran en el recurso compartido y afectan a los usuarios que acceden remotamente.
- **Permisos locales:** Regulan el acceso a los recursos desde el propio equipo donde están almacenados. Se configuran en el sistema de archivos.

La **herencia** permite que los permisos establecidos en una carpeta se transmitan automáticamente a sus subcarpetas y archivos, facilitando la administración.

Las **Listas de Control de Acceso (ACL)** especifican de forma detallada los permisos que tiene cada usuario o grupo sobre un recurso, permitiendo una gestión granular de la seguridad.

### 8.1.1. Windows

En sistemas Windows, la gestión de permisos se realiza principalmente a través de las propiedades de los archivos y carpetas. Al hacer clic derecho sobre un recurso y seleccionar «Propiedades» → «Seguridad», se accede a la configuración de permisos NTFS, donde se pueden asignar permisos específicos a usuarios y grupos, como lectura, escritura, modificación o control total.

Además, Windows utiliza las ACL para definir los permisos de acceso. Los administradores pueden modificar estas listas para conceder o denegar permisos detallados. En recursos compartidos de red, también se pueden establecer permisos de compartición, que se suman a los permisos NTFS para determinar el acceso efectivo.

Las políticas de grupo (GPO) permiten aplicar configuraciones de seguridad y permisos de manera centralizada en entornos de dominio, facilitando la administración de múltiples equipos y usuarios.

#### Comandos de utilidad

- **icacls**: Permite ver y modificar permisos y ACLs en archivos y carpetas.  
Ejemplo: `icacls carpeta /grant Juan:(R,W)`
- **net share**: Gestiona recursos compartidos en red. Ejemplo: `net share Documentos="C:\Documentos" /grant:Juan,READ`
- **cacls** (obsoleto, sustituido por **icacls**): Modifica permisos en archivos y carpetas.
- **net user**: Administra cuentas de usuario. Ejemplo: `net user Juan /active:yes`
- **net localgroup**: Administra grupos locales. Ejemplo: `net localgroup Usuarios Juan /add`

#### Ejemplo práctico

Supongamos que queremos compartir una carpeta en Windows y permitir que el usuario **Juan** tenga acceso de lectura y escritura:

1. Haz clic derecho sobre la carpeta **Documentos** y selecciona **Propiedades** → .
2. Ve a la pestaña **Compartir** y pulsa **Compartir**.
3. Añade el usuario **Juan** y selecciona el nivel de permiso (**Lectura/Escritura**).

4. En la pestaña Seguridad, verifica que Juan tiene los permisos NTFS adecuados.

### Ejercicio 8.1



En un entorno Windows, crea una carpeta compartida llamada Proyectos y configura los permisos para que el usuario Ana tenga acceso de solo lectura, mientras que el usuario Carlos tenga acceso de lectura y escritura. Documenta los pasos realizados y verifica los permisos asignados.

## 8.1.2. Linux

En sistemas Linux, la gestión de permisos se basa principalmente en el sistema de archivos y en el modelo de usuarios y grupos. Cada archivo y carpeta tiene asociados tres tipos de permisos: lectura (r), escritura (w) y ejecución (x), que se asignan por separado al propietario, al grupo y a otros usuarios. Estos permisos pueden visualizarse y modificarse mediante los comandos `ls -l`, `chmod`, `chown` y `chgrp`. Estos permisos se codifican en un formato de tres dígitos, donde cada dígito representa los permisos para el propietario, el grupo y otros usuarios, respectivamente. P. ej. `chmod 755 archivo` otorga permisos de lectura, escritura y ejecución al propietario, y permisos de lectura y ejecución al grupo y a otros usuarios. Cada dígito representa los siguientes permisos: 4 (lectura), 2 (escritura) y 1 (ejecución), y es posible combinarlos sumándolos.

Además, Linux soporta ACLs (Access Control Lists) para una gestión más detallada de los permisos, permitiendo asignar permisos específicos a múltiples usuarios y grupos. Los comandos `getfacl` y `setfacl` se utilizan para consultar y modificar estas listas.

En recursos compartidos de red, como los gestionados por Samba o NFS, se pueden definir permisos adicionales en la configuración de los servicios, controlando el acceso remoto a los archivos y carpetas. La combinación de permisos locales y de red permite una administración flexible y segura de los recursos compartidos en entornos Linux.

### Comandos de utilidad

- `ls -l`: Muestra los permisos, propietario y grupo de los archivos y carpetas.
- `chmod`: Modifica los permisos de archivos y carpetas. Ejemplo: `chmod 755 archivo`.

- **chown:** Cambia el propietario y grupo de archivos y carpetas. Ejemplo: `chown usuario:grupo archivo`.
- **chgrp:** Cambia el grupo asociado a un archivo o carpeta. Ejemplo: `chgrp grupo archivo`.
- **getfacl:** Consulta las ACL de un archivo o carpeta. Ejemplo: `getfacl carpeta`.
- **setfacl:** Modifica las ACL de un archivo o carpeta. Ejemplo: `setfacl -m u:usuario:rw carpeta`.

### Ejemplo práctico

Supongamos que queremos compartir la carpeta `/home/juan/documentos` y asignar permisos de lectura y escritura al usuario `juan`:

1. Asigna la propiedad de la carpeta: `sudo chown juan:juan /home/juan/documentos`
2. Establece los permisos: `chmod 770 /home/juan/documentos`
3. Para permisos más detallados, usa ACL: `setfacl -m u:pedro:rw /home/juan/documentos`

Estos pasos permiten gestionar de forma segura los permisos sobre recursos compartidos en ambos sistemas operativos.

### Ejercicio 8.2

En un sistema Linux, crea una carpeta llamada `Compartida` en el directorio `/home`. Configura los permisos para que el usuario `usuario` tenga acceso completo, el grupo `desarrolladores` tenga acceso de lectura y escritura, y otros usuarios no tengan ningún acceso. Utiliza tanto los comandos tradicionales de permisos como las ACL para lograr esto. Documenta los comandos utilizados y verifica los permisos asignados.

## 8.2. Configuración de recursos compartidos

La configuración de recursos compartidos en una red implica definir qué recursos (carpetas, archivos, impresoras, etc.) estarán disponibles para otros usuarios y cómo se accede a ellos. Es fundamental establecer permisos de acceso y aplicar directivas de seguridad para proteger la información y garantizar que solo los usuarios autorizados puedan acceder o modificar los recursos.

### 8.2.1. Permisos de acceso

Los permisos de acceso determinan el nivel de interacción que los usuarios tienen con los recursos compartidos. Se pueden configurar diferentes niveles de acceso, como solo lectura, lectura y escritura, o control total. Es recomendable asignar los permisos siguiendo el principio de mínimo privilegio, otorgando solo los permisos necesarios para cada usuario o grupo.

### 8.2.2. Directivas de seguridad

Las directivas de seguridad permiten definir reglas y restricciones adicionales sobre el uso de los recursos compartidos. Por ejemplo, se pueden establecer políticas de contraseñas, restricciones de acceso por horario, auditoría de accesos y bloqueo de cuentas tras varios intentos fallidos. En entornos Windows, estas directivas se gestionan principalmente mediante las Políticas de Grupo (GPO). En Linux, se pueden aplicar mediante configuraciones en los servicios de compartición (Samba, NFS) y mediante el uso de ACLs.

Una correcta configuración de recursos compartidos y la aplicación de directivas de seguridad son esenciales para proteger los datos y garantizar el funcionamiento seguro de la red.

### 8.2.3. Windows

En Windows, la configuración de recursos compartidos se realiza principalmente a través del Explorador de archivos y las herramientas administrativas. Para compartir una carpeta, haz clic derecho sobre ella, selecciona **Propiedades** y accede a la pestaña **Compartir**. Desde aquí puedes elegir los usuarios o grupos que tendrán acceso y definir el nivel de permiso (lectura, escritura, control total).

Además, es posible gestionar recursos compartidos mediante la consola de administración de equipos (`compmgmt.msc`) y el uso de comandos como `net share`.

Es importante combinar los permisos de compartición con los permisos NTFS para asegurar que los usuarios tengan el acceso adecuado.

Las Políticas de Grupo (GPO) permiten aplicar configuraciones de compartición y seguridad de forma centralizada en entornos de dominio, facilitando la administración de múltiples equipos y usuarios.

#### Ejemplo práctico

Para compartir una carpeta llamada **Recursos** y permitir acceso de lectura al grupo **Usuarios** y acceso de escritura al usuario **Administrador**:

1. Haz clic derecho sobre la carpeta **Recursos** y selecciona **Propiedades**.
2. Ve a la pestaña **Compartir** y pulsa **Compartir**.
3. Añade el grupo **Usuarios** con permiso de lectura y el usuario **Administrador** con permiso de lectura/escritura.
4. En la pestaña **Seguridad**, verifica y ajusta los permisos NTFS para que coincidan con los permisos de compartición.
5. Opcionalmente, utiliza el comando: `net share Recursos="C:\Recursos" /grant:Usuarios,READ /grant:Administrador,CHANGE`

### Ejercicio 8.3



Configura un recurso compartido en un sistema Windows llamado **Datos** ubicado en `C:\Datos`. Otorga permisos de solo lectura al grupo **Usuarios** y permisos de lectura y escritura al usuario **Admin**. Documenta los pasos realizados, incluyendo la configuración de permisos NTFS y el uso del comando `net share`.

#### 8.2.4. Linux

En Linux, la compartición de recursos se realiza principalmente mediante servicios como Samba (para compartir con equipos Windows) y NFS (para compartir entre sistemas Unix/Linux). Entraremos en detalle en la configuración de ambos servicios en la sección sobre servidores de ficheros.

### 8.3. Requisitos de seguridad del sistema y de los datos

La seguridad de los sistemas y de los datos en una red es fundamental para proteger la información frente a accesos no autorizados, pérdida, manipulación o ataques. Los principales requisitos de seguridad incluyen:

- **Confidencialidad:** Garantizar que solo los usuarios autorizados puedan acceder a la información. Se logra mediante el uso de permisos, cifrado y autenticación.
- **Integridad:** Asegurar que los datos no sean modificados de forma no autorizada. Se implementa mediante controles de acceso, registros de auditoría y mecanismos de verificación.

- **Disponibilidad:** Mantener los recursos accesibles para los usuarios legítimos cuando los necesiten. Se consigue mediante redundancia, copias de seguridad y protección frente a ataques de denegación de servicio.
- **Autenticación:** Verificar la identidad de los usuarios antes de permitir el acceso a los recursos. Se emplean contraseñas seguras, autenticación multifactor y certificados digitales.
- **Autorización:** Definir qué acciones puede realizar cada usuario sobre los recursos, aplicando el principio de mínimo privilegio.
- **Auditoría:** Registrar y supervisar las acciones realizadas sobre los sistemas y los datos para detectar posibles incidentes de seguridad.
- **Protección física:** Asegurar el acceso físico restringido a los servidores y dispositivos de red.
- **Copia de seguridad:** Realizar backups periódicos de los datos y sistemas para garantizar la recuperación ante fallos o incidentes.
- **Actualización y parcheo:** Mantener los sistemas operativos y aplicaciones actualizados para corregir vulnerabilidades.

La aplicación de estos requisitos debe ser continua y adaptarse a las necesidades y riesgos específicos de cada organización, combinando medidas técnicas, organizativas y legales.

### Ejercicio 8.4



Investiga y describe al menos tres medidas de seguridad que puedes implementar en un sistema Windows y en un sistema Linux para proteger los datos y garantizar la seguridad del sistema.

Explica cómo cada medida contribuye a la seguridad general.

## 8.4. Servidores de ficheros

Los servidores de ficheros son sistemas dedicados a almacenar, gestionar y compartir archivos dentro de una red. Permiten que múltiples usuarios accedan de forma centralizada a documentos, imágenes, bases de datos y otros tipos de información, facilitando la colaboración y la administración de los datos.

### 8.4.1. Características principales

- **Centralización de datos:** Todos los archivos se almacenan en un único servidor, lo que simplifica la gestión y las copias de seguridad.
- **Acceso controlado:** El servidor permite definir permisos y políticas de acceso para usuarios y grupos, garantizando la seguridad y la privacidad de la información.
- **Compartición de recursos:** Los usuarios pueden acceder a los archivos desde diferentes dispositivos y ubicaciones dentro de la red.
- **Escalabilidad:** Es posible ampliar la capacidad de almacenamiento y el número de usuarios según las necesidades de la organización.
- **Integración con otros servicios:** Los servidores de ficheros pueden integrarse con sistemas de autenticación, auditoría y respaldo.

### 8.4.2. Tipos de servidores de ficheros

- **Windows:** Utiliza servicios como el *Servidor de archivos* y protocolos como SMB/CIFS. La administración se realiza mediante el Explorador de archivos, la consola de administración y comandos como `net share`.
- **Linux/Unix:** Utiliza servicios como Samba (para compartir con equipos Windows) y NFS (para compartir entre sistemas Unix/Linux). La configuración se realiza editando archivos de configuración y estableciendo permisos adecuados.
- **NAS (Network Attached Storage):** Dispositivos dedicados que ofrecen almacenamiento en red y suelen incluir interfaces web para su administración.

### 8.4.3. Windows

En Windows, los servidores de ficheros se configuran habitualmente mediante el rol de «Servidor de archivos», disponible en las ediciones profesionales y de servidor del sistema operativo. Este rol permite compartir carpetas y discos con usuarios de la red utilizando el protocolo SMB/CIFS.

#### Instalación y configuración

1. Instala el rol «Servidor de archivos» desde el Administrador del servidor (**Server Manager**) en Windows Server.

2. Crea una carpeta destinada a ser compartida, por ejemplo `C:\Compartida`
3. Haz clic derecho sobre la carpeta, selecciona **Propiedades** y accede a la pestaña **Compartir**.
4. Configura los usuarios o grupos que tendrán acceso y define los permisos (lectura, escritura, control total).
5. Ajusta los permisos NTFS en la pestaña **Seguridad** para garantizar el acceso adecuado.
6. Opcionalmente, utiliza el comando `net share` para compartir la carpeta desde la línea de comandos:

```
1| net share Compartida="C:\Compartida" /grant:Usuarios,READ /→
 grant:Admin,CHANGE
```

### Gestión y acceso

Los usuarios pueden acceder a los recursos compartidos desde otros equipos de la red mediante el Explorador de archivos, usando rutas UNC como `\\servidor\Compartida`. Es posible auditar el acceso y modificar permisos en cualquier momento para mantener la seguridad.

#### Ejercicio 8.5



Instala el rol de servidor de archivos en un sistema Windows Server. Crea una carpeta compartida llamada **Proyectos** y configura los permisos para que el grupo **Desarrollo** tenga acceso de lectura y escritura, mientras que el grupo **Invitados** solo tenga acceso de lectura. Documenta los pasos realizados y verifica el acceso desde un equipo cliente.

### 8.4.4. Linux

En Linux, los servidores de ficheros suelen implementarse mediante servicios como Samba y NFS, que permiten compartir archivos entre diferentes sistemas operativos y equipos de la red.

#### Samba

Samba es una solución popular para compartir archivos entre sistemas Linux y equipos Windows. Para configurar un servidor de ficheros con Samba:

1. Instala Samba: `sudo apt install samba`
2. Crea la carpeta a compartir, por ejemplo: `sudo mkdir -p /srv/compartida`
3. Asigna el propietario y permisos: `sudo chown usuario:usuario /srv/compartida` y `chmod 770 /srv/compartida`
4. Edita el archivo `/etc/samba/smb.conf` y añade:

```
1 [compartida]
2 path = /srv/compartida
3 read only = no
4 valid users = usuario
```

5. Añade el usuario a Samba: `sudo smbpasswd -a usuario`
6. Reinicia el servicio: `sudo systemctl restart smbd`

Los usuarios pueden acceder al recurso compartido desde equipos Windows o Linux utilizando la ruta de red `\\\servidor\compartida`.

### Ejercicio 8.6



Instala y configura un servidor de ficheros en Linux utilizando Samba. Crea una carpeta compartida llamada `Documentos` y otorga acceso de lectura y escritura al usuario `maria`. Documenta los pasos realizados y verifica el acceso desde un equipo cliente Windows.

## NFS

NFS es utilizado principalmente para compartir archivos entre sistemas Unix/Linux. Para configurar un servidor NFS:

1. Instala NFS: `sudo apt install nfs-kernel-server`
2. Crea la carpeta a compartir: `sudo mkdir -p /srv/compartida`
3. Asigna permisos: `sudo chown usuario:usuario /srv/compartida` y `chmod 770 /srv/compartida`
4. Edita el archivo `/etc/exports` y añade:  
`/srv/compartida 192.168.1.0/24(rw, sync)`
5. Aplica los cambios: `sudo exportfs -ra`

6. Reinicia el servicio: `sudo systemctl restart nfs-kernel-server`

Los clientes pueden montar el recurso compartido con `mount` y acceder a los archivos según los permisos configurados.

## Gestión y acceso

La gestión de permisos se realiza mediante los comandos tradicionales (`chmod`, `chown`) y, si se requiere mayor granularidad, mediante ACLs (`setfacl`). Es recomendable auditar el acceso y mantener el servidor actualizado para garantizar la seguridad.

### Ejercicio 8.7



Instala y configura un servidor de ficheros en Linux utilizando NFS. Crea una carpeta compartida llamada `Proyectos` y otorga acceso de lectura y escritura al grupo `desarrolladores`. Documenta los pasos realizados y verifica el acceso desde un equipo cliente Linux.

### 8.4.5. Buenas prácticas

- Definir permisos siguiendo el principio de mínimo privilegio.
- Realizar copias de seguridad periódicas.
- Monitorizar el acceso y uso de los archivos mediante registros de auditoría.
- Mantener el servidor actualizado y protegido frente a amenazas.

## 8.5. Servidores de impresión

Los servidores de impresión son sistemas dedicados a gestionar y compartir impresoras en una red, permitiendo que varios usuarios envíen trabajos de impresión de forma centralizada y eficiente.

### 8.5.1. Características principales

- **Centralización:** Todas las impresoras se gestionan desde un único servidor, facilitando la administración y el control de los trabajos de impresión.
- **Gestión de colas:** El servidor organiza los trabajos de impresión en colas, permitiendo priorizar, pausar o cancelar tareas.

- **Control de acceso:** Es posible definir qué usuarios o grupos pueden utilizar cada impresora, aplicando permisos y restricciones.
- **Monitorización:** Permite supervisar el estado de las impresoras, el uso y los trabajos realizados.
- **Compatibilidad:** Soporta diferentes modelos y marcas de impresoras, así como diversos protocolos de impresión.

## 8.5.2. Windows

En Windows, la gestión de impresoras en red se realiza principalmente a través de la aplicación «Configuración» y el panel «Dispositivos e impresoras». Permite compartir impresoras conectadas al equipo y controlar los trabajos de impresión de los usuarios de la red.

### Instalación y configuración

1. Conecta la impresora al equipo y asegúrate de que el controlador esté instalado correctamente.
2. Ve a **Configuración** → **Dispositivos** → **Impresoras y escáneres**.
3. Selecciona la impresora, haz clic en **Administrar** y luego en **Propiedades de la impresora**.
4. En la pestaña **Compartir**, marca la opción **Compartir esta impresora** y asigna un nombre de recurso.
5. En la pestaña **Seguridad**, define los permisos de acceso para los usuarios y grupos.
6. Los usuarios pueden agregar la impresora compartida desde otros equipos usando la ruta `\nombre_equipo\impresora`.

### Ejercicio 8.8

Comparte una impresora llamada **LaserJet** en Windows y otorga acceso de impresión al grupo **Usuarios**, mientras que el grupo **Administradores** puede gestionar la cola de impresión. Documenta los pasos realizados y verifica el acceso desde un equipo cliente.

### 8.5.3. Linux

En Linux, la gestión de impresoras en red se realiza principalmente mediante el sistema CUPS (Common UNIX Printing System), que permite compartir impresoras y administrar trabajos de impresión.

#### Instalación y configuración

1. Instala CUPS: `sudo apt install cups`
2. Accede a la interfaz web de administración: `http://localhost:631`
3. Añade la impresora y configúrala como compartida.
4. Define los permisos de acceso y los usuarios autorizados.
5. Los equipos clientes pueden añadir la impresora compartida mediante CUPS o protocolos estándar (IPP, LPD).

#### Ejercicio 8.9



Instala y configura CUPS en un sistema Linux. Comparte una impresora llamada `HP_Officejet` y permite que el usuario `ana` pueda imprimir, mientras que el usuario `admin` pueda administrar la cola de impresión. Documenta los pasos realizados y verifica el acceso desde un equipo cliente Linux.

### 8.5.4. Buenas prácticas

- Definir permisos de impresión y administración según las necesidades.
- Monitorizar el uso y estado de las impresoras.
- Mantener los controladores y el servidor actualizados.
- Realizar mantenimiento preventivo de las impresoras físicas.

## 8.6. Servidores de aplicaciones

Los servidores de aplicaciones son sistemas que alojan y ejecutan aplicaciones empresariales, permitiendo el acceso y uso centralizado por parte de los usuarios de la red. Estos servidores gestionan la lógica de negocio, el acceso a bases de datos y la comunicación entre clientes y otros servicios.

### 8.6.1. Características principales

- **Centralización de aplicaciones:** Las aplicaciones se ejecutan en el servidor, facilitando la administración, actualización y mantenimiento.
- **Acceso remoto:** Los usuarios pueden acceder a las aplicaciones desde diferentes dispositivos y ubicaciones.
- **Gestión de usuarios y permisos:** Permite definir qué usuarios pueden acceder y qué acciones pueden realizar en cada aplicación.
- **Escalabilidad:** Es posible ampliar la capacidad del servidor y el número de usuarios según las necesidades.
- **Integración:** Los servidores de aplicaciones pueden interactuar con bases de datos, servidores web y otros servicios de red.

### 8.6.2. Tipos de servidores de aplicaciones

- **Windows:** Utiliza servicios como IIS (Internet Information Services) para aplicaciones web, y Remote Desktop Services para aplicaciones de escritorio.
- **Linux/Unix:** Utiliza servidores como Apache Tomcat, JBoss, GlassFish, o servicios de escritorio remoto como X2Go y VNC.
- **Virtualización y contenedores:** Plataformas como Docker y Kubernetes permiten desplegar aplicaciones en contenedores, facilitando la gestión y escalabilidad.

### 8.6.3. Windows

En Windows, el servidor de aplicaciones más común es IIS, que permite alojar aplicaciones web basadas en tecnologías como ASP.NET, PHP o Node.js. Para aplicaciones de escritorio, se pueden utilizar los Servicios de Escritorio Remoto (RDS), que permiten a los usuarios ejecutar aplicaciones en el servidor y acceder a ellas desde sus equipos.

#### Ejemplo práctico

Para publicar una aplicación web en IIS:

1. Instala el rol IIS desde el Administrador del servidor.

2. Copia los archivos de la aplicación al directorio C:\inetpub\wwwroot\miapp.
3. Configura el sitio web en IIS y define los permisos de acceso.
4. Los usuarios acceden a la aplicación mediante un navegador web usando la dirección del servidor.

### Ejercicio 8.10



Instala IIS en un sistema Windows Server y publica una aplicación web sencilla. Configura los permisos para que solo el grupo Usuarios pueda acceder a la aplicación. Documenta los pasos realizados y verifica el acceso desde un equipo cliente.

## 8.6.4. Linux

En Linux, los servidores de aplicaciones más utilizados son Apache Tomcat (para aplicaciones Java), Node.js, y servidores de escritorio remoto como X2Go. La instalación y configuración varía según la tecnología utilizada.

### Ejemplo práctico

Para desplegar una aplicación Java en Tomcat:

1. Instala Tomcat: `sudo apt install tomcat9`
2. Copia el archivo .war de la aplicación al directorio /var/lib/tomcat9/webapps/.
3. Configura los permisos y usuarios en Tomcat si es necesario.
4. Accede a la aplicación desde un navegador web usando la dirección del servidor y el nombre de la aplicación.

### Ejercicio 8.11



Instala Apache Tomcat en un sistema Linux y despliega una aplicación web de ejemplo. Configura los permisos para que solo el usuario ana puebla acceder a la aplicación. Documenta los pasos realizados y verifica el acceso desde un equipo cliente.

### 8.6.5. Docker y contenedores

Docker y otras tecnologías de contenedores permiten desplegar aplicaciones de forma aislada y portátil, facilitando la gestión, escalabilidad y seguridad en entornos de red.

#### Características principales

- **Aislamiento:** Cada aplicación se ejecuta en su propio contenedor, separado del resto del sistema y de otros contenedores.
- **Portabilidad:** Los contenedores pueden ejecutarse en cualquier sistema que soporte Docker, independientemente de la distribución o versión.
- **Escalabilidad:** Es sencillo crear, replicar y eliminar instancias de aplicaciones según la demanda.
- **Gestión centralizada:** Herramientas como Docker Compose y Kubernetes permiten orquestar múltiples contenedores y servicios.

#### Principales comandos

**docker run** es el comando básico para ejecutar un contenedor a partir de una imagen. Permite especificar opciones como puertos, volúmenes, variables de entorno y el modo de ejecución. Ejemplo:

```
1 docker run -d -p 8080:80 --name miapp nginx
```

Este comando ejecuta un contenedor en segundo plano ( **-d**), mapea el puerto 80 del contenedor al 8080 del host, y lo nombra **miapp** usando la imagen **nginx**.

**docker compose** es una herramienta para definir y gestionar aplicaciones multicontenedor mediante un archivo **docker-compose.yml**. Permite describir servicios, redes y volúmenes de forma declarativa. Ejemplo de archivo:

```
1 version: '3'
2 services:
3 web:
4 image: nginx
5 ports:
6 - "8080:80"
7 db:
8 image: mysql
9 environment:
10 MYSQL_ROOT_PASSWORD: ejemplo
```

Para iniciar los servicios definidos, se usa:

```
1| docker compose up -d
```

**docker compose** facilita la gestión, escalabilidad y replicación de aplicaciones complejas, permitiendo iniciar, detener y administrar todos los servicios con comandos sencillos.

### 8.6.6. Buenas prácticas

- Mantener las aplicaciones y el servidor actualizados.
- Definir permisos de acceso siguiendo el principio de mínimo privilegio.
- Monitorizar el uso y rendimiento de las aplicaciones.
- Realizar copias de seguridad periódicas de la configuración y los datos.
- Proteger el servidor frente a amenazas mediante cortafuegos y medidas de seguridad adicionales.

## 8.7. Técnicas de conexión remota

Las técnicas de conexión remota permiten acceder y administrar sistemas y recursos de la red desde ubicaciones externas, facilitando la gestión y el soporte técnico. Las principales opciones incluyen:

### 8.7.1. Acceso remoto en Windows

En Windows, la herramienta más utilizada es el **Escrivtorio remoto** (Remote Desktop), que permite conectarse a otro equipo y operar como si estuvieras físicamente presente. Para habilitarlo:

1. Accede a Configuración → Sistema → Escritorio remoto.
2. Activa la opción Permitir conexiones remotas.
3. Configura los usuarios autorizados y verifica la configuración del cortafuegos.
4. Conéctate desde otro equipo usando la aplicación Conexión a Escritorio remoto (mstsc).

También existen herramientas como **PowerShell Remoting** (Enter- PSSession, Invoke-Command) y utilidades de terceros como TeamViewer o AnyDesk.

### 8.7.2. Acceso remoto en Linux

En Linux, el acceso remoto se realiza principalmente mediante **SSH** (Secure Shell), que permite conexiones seguras para administración y transferencia de archivos.

1. Instala el servidor SSH: `sudo apt install openssh-server`
2. Verifica que el servicio esté activo: `sudo systemctl status ssh`
3. Conéctate desde otro equipo con: `ssh usuario@ip_servidor`

Para acceso gráfico, se pueden usar soluciones como **VNC** (`vncserver`, `vncviewer`), **X2Go** o **RDP** (mediante `xrdp`).

#### Consejo



Además de SSH, existe SFTP (SSH File Transfer Protocol), un protocolo seguro para la transferencia de archivos que utiliza el canal cifrado de SSH. SFTP permite subir, descargar y gestionar archivos de forma remota, garantizando la confidencialidad e integridad de los datos. Para usar SFTP, basta con conectarse mediante el comando `sftp` `usuario@ip_servidor` desde el cliente. Es recomendable restringir el acceso SFTP solo a usuarios autorizados y utilizar claves SSH para una autenticación más segura.

### 8.7.3. Buenas prácticas

- Utilizar conexiones cifradas (SSH, RDP con TLS).
- Restringir el acceso remoto solo a usuarios autorizados.
- Cambiar los puertos por defecto y usar autenticación fuerte.
- Monitorizar los accesos y registrar las sesiones remotas.
- Deshabilitar el acceso remoto cuando no sea necesario.

Estas técnicas permiten una administración eficiente y segura de los sistemas en red, facilitando el soporte y la gestión remota.

#### Ejercicio 8.12



Configura el acceso remoto en un sistema Windows utilizando Escritorio remoto y en un sistema Linux utilizando SSH. Asegúrate de aplicar me-

didas de seguridad adecuadas, como la restricción de usuarios y el uso de autenticación fuerte. Documenta los pasos realizados y verifica el acceso desde equipos clientes.

## 8.8. Cortafuegos

Los cortafuegos (firewalls) son sistemas de seguridad que controlan el tráfico de red entre diferentes segmentos, permitiendo o bloqueando conexiones según reglas definidas. Su objetivo principal es proteger los sistemas y datos frente a accesos no autorizados, ataques y amenazas externas.

### 8.8.1. Tipos de cortafuegos

- **Cortafuegos de red:** Dispositivos o software que filtran el tráfico entre redes (por ejemplo, entre la red interna y Internet).
- **Cortafuegos de host:** Software instalado en equipos individuales para controlar el tráfico entrante y saliente.
- **Cortafuegos de próxima generación (NGFW):** Incluyen funciones avanzadas como inspección profunda de paquetes, detección de intrusiones y filtrado de aplicaciones.

### 8.8.2. Configuración en Windows

Windows incluye el **Firewall de Windows**, que permite definir reglas para aplicaciones y puertos. Para configurarlo:

1. Accede a Panel de control → Sistema y seguridad → Firewall de Windows Defender.
2. Define reglas de entrada y salida para permitir o bloquear aplicaciones y puertos.
3. Utiliza el comando `netsh advfirewall` para gestionar el cortafuegos desde la línea de comandos.

### 8.8.3. Configuración en Linux

En Linux, los cortafuegos más comunes son **iptables**, **nftables** y **ufw** (Uncomplicated Firewall). Ejemplo básico con UFW:

1. Instala UFW: `sudo apt install ufw`
2. Activa el cortafuegos: `sudo ufw enable`
3. Permite el acceso SSH: `sudo ufw allow ssh`
4. Deniega otros accesos no necesarios: `sudo ufw deny 23` (por ejemplo, para Telnet)
5. Consulta el estado: `sudo ufw status`

#### 8.8.4. Buenas prácticas

- Definir reglas restrictivas, permitiendo solo el tráfico necesario.
- Actualizar y revisar periódicamente la configuración del cortafuegos.
- Monitorizar los registros de acceso y los intentos de conexión bloqueados.
- Combinar cortafuegos de red y de host para una protección más completa.

Los cortafuegos son una pieza clave en la defensa de los sistemas informáticos, ayudando a prevenir accesos no autorizados y a mitigar riesgos de seguridad en la red.

##### Ejercicio 8.13

Configura un cortafuegos en un sistema Windows y en un sistema Linux. Define reglas para permitir solo el acceso necesario (por ejemplo, Escritorio remoto en Windows y SSH en Linux) y bloquea el resto del tráfico. Documenta los comandos y pasos realizados, y verifica la efectividad de las reglas aplicadas.



#### 8.9. Implementación y explotación de dominios

La implantación y explotación de dominios en una red permite centralizar la gestión de usuarios, equipos y recursos, facilitando la administración y la seguridad. Un dominio es un entorno gestionado por un servidor central (controlador de dominio) que autentica y autoriza a los usuarios y equipos.

### 8.9.1. Conceptos clave

- **Controlador de dominio:** Servidor que gestiona la autenticación, autorización y políticas de seguridad en el dominio.
- **Active Directory (Windows):** Servicio de directorio que almacena información sobre usuarios, equipos y recursos, y permite aplicar políticas de grupo (GPO).
- **LDAP (Linux/Unix):** Protocolo estándar para servicios de directorio, utilizado por soluciones como OpenLDAP y Samba.
- **Unión al dominio:** Proceso por el cual un equipo pasa a formar parte del dominio y es gestionado centralizadamente.

### 8.9.2. Implantación de un dominio en Windows

1. Instala el rol **Servicios de dominio de Active Directory** en Windows Server.
2. Promociona el servidor como controlador de dominio y define el nombre del dominio (por ejemplo, `empresa.local`).
3. Configura usuarios, grupos y unidades organizativas en Active Directory.
4. Une los equipos cliente al dominio desde **Propiedades del sistema → Nombre de equipo → Cambiar → Dominio**.
5. Aplica políticas de grupo (GPO) para gestionar configuraciones, permisos y directivas de seguridad.

### 8.9.3. Implantación de un dominio en Linux

En Linux, la gestión de dominios puede realizarse mediante Samba (como controlador de dominio) o OpenLDAP.

1. Instala Samba y configúralo como controlador de dominio: `sudo apt→ install samba`, edita `/etc/samba/smb.conf` y define el dominio.
2. Añade usuarios y equipos al dominio mediante comandos como `samba→ -tool user add` y `samba-tool computer add`.
3. Configura los clientes Linux o Windows para unirse al dominio.
4. Utiliza LDAP para gestionar usuarios y permisos si se requiere integración con otros servicios.

## Ejemplo práctico

Para implantar un dominio con OpenLDAP en Linux, sigue estos pasos básicos:

1. Instala OpenLDAP y las herramientas asociadas:

```
1| sudo apt install slapd ldap-utils
```

2. Configura el dominio durante la instalación (por ejemplo, dc=empresa,dc=local).

3. Añade usuarios y grupos mediante archivos LDIF. Ejemplo de usuario:

```
1 dn: uid=juan,ou=usuarios,dc=empresa,dc=local
2 objectClass: inetOrgPerson
3 uid: juan
4 sn: Pérez
5 cn: Juan Pérez
6 userPassword: contraseña_segura
```

4. Importa el usuario con el comando:

```
1| ldapadd -x -D "cn=admin,dc=empresa,dc=local" -W -f usuario.ldif
```

5. Configura los clientes Linux para autenticarse contra el servidor LDAP editando los archivos /etc/ldap.conf y /etc/nsswitch.conf.

6. Verifica la autenticación y la consulta de usuarios con:

```
1| ldapsearch -x -b "dc=empresa,dc=local"
```

Este proceso permite centralizar la gestión de usuarios y autenticación en la red mediante OpenLDAP.

### 8.9.4. Explotación y administración

- Centraliza la gestión de usuarios, contraseñas y permisos.
- Aplica políticas de seguridad y configuración mediante GPO (Windows) o scripts/configuraciones centralizadas (Linux).
- Supervisa el acceso y uso de recursos mediante registros y auditoría.
- Realiza copias de seguridad periódicas del servicio de directorio.
- Mantén los controladores de dominio actualizados y protegidos.

**Ejercicio 8.14**

Implanta un dominio en Linux utilizando Samba como controlador de dominio. Crea un usuario llamado `ana` y un grupo llamado `proyectos`. Une un equipo cliente Linux al dominio y verifica que el usuario `ana` puede autenticarse correctamente. Documenta los pasos realizados, incluyendo la configuración de Samba, la creación de usuarios y grupos, y la unión del cliente al dominio.

## Resumen

En este capítulo hemos visto cómo gestionar y proteger los recursos en una red, incluyendo la configuración de permisos y derechos en sistemas Windows y Linux, la compartición de archivos e impresoras, la implantación de servidores de ficheros y aplicaciones, y la aplicación de técnicas de conexión remota y cortafuegos. También hemos abordado la implantación y administración de dominios para centralizar la gestión de usuarios y recursos. La correcta configuración y administración de estos elementos es esencial para garantizar la seguridad, disponibilidad y eficiencia en el entorno de red de cualquier organización.



# Explotación de aplicaciones informáticas de propósito general

Las aplicaciones informáticas de propósito general son herramientas fundamentales en el entorno profesional, educativo y personal. Permiten realizar tareas diversas, desde la gestión de documentos y la comunicación hasta el mantenimiento y la seguridad de los sistemas. El conocimiento sobre los distintos tipos de software, sus requisitos y licencias, así como el manejo de herramientas ofimáticas y colaborativas, resulta esencial para aprovechar al máximo las posibilidades que ofrecen las tecnologías de la información. Este capítulo proporciona una visión global sobre la explotación eficiente y responsable de estas aplicaciones, facilitando su selección, instalación y uso en diferentes contextos.

## 9.1. Software: tipos, requisitos, y licencias

En esta sección se analizarán los principales tipos de software, los requisitos necesarios para su funcionamiento y las distintas licencias que regulan su uso. El objetivo es ofrecer una base sólida para comprender cómo seleccionar, instalar y utilizar aplicaciones informáticas de manera eficiente y conforme a la legalidad vigente.

### 9.1.1. Tipos de software

El software se clasifica principalmente en tres categorías: software de sistema, software de desarrollo y software de aplicación. Cada tipo de software tiene un propósito específico y está diseñado para cumplir diferentes funciones dentro del entorno informático.

#### Software de sistema

El software de sistema es el conjunto de programas que gestionan y controlan el hardware del ordenador, proporcionando una base para que otros progra-

mas funcionen. Los ejemplos más comunes son los sistemas operativos (como Windows, Linux o macOS), los controladores de dispositivos (drivers), y las utilidades de gestión del sistema (herramientas de copia de seguridad, administración de discos, etc.). Este tipo de software es esencial para el funcionamiento del equipo y suele ejecutarse en segundo plano, facilitando la interacción entre el usuario, el hardware y el resto del software.

### **Software de desarrollo**

El software de desarrollo está compuesto por herramientas que permiten crear, editar, depurar y mantener otros programas informáticos. Incluye entornos de desarrollo integrados (IDE), editores de código, compiladores, intérpretes, depuradores y sistemas de control de versiones. Ejemplos de software de desarrollo son Visual Studio Code, Eclipse, NetBeans, Git, GCC, y herramientas de diseño de bases de datos. Este tipo de software es utilizado principalmente por programadores y desarrolladores para construir aplicaciones y sistemas.

### **Software de aplicación**

El software de aplicación está diseñado para ayudar al usuario a realizar tareas específicas, facilitando actividades como la edición de documentos, gestión de datos, comunicación, diseño gráfico, y entretenimiento. Ejemplos comunes incluyen procesadores de texto (Microsoft Word, LibreOffice Writer), hojas de cálculo (Microsoft Excel, Google Sheets), navegadores web (Google Chrome, Mozilla Firefox), programas de correo electrónico (Outlook, Thunderbird), aplicaciones de edición multimedia (Adobe Photoshop, Audacity), y gestores de bases de datos (MySQL, Microsoft Access). Este tipo de software suele tener interfaces gráficas intuitivas y está orientado a satisfacer necesidades concretas de los usuarios finales.

#### **9.1.2. Requisitos de software**

Los requisitos de software pueden ser hardware (memoria RAM, espacio en disco, procesador, etc.) y software (sistema operativo compatible, librerías necesarias, versiones mínimas, etc.). Es fundamental revisar estos requisitos antes de instalar cualquier aplicación para asegurar su correcto funcionamiento.

#### **9.1.3. Licencias de software**

Las licencias de software determinan cómo se puede usar, distribuir y modificar un programa. Es importante conocer la licencia de cada software para cumplir con las condiciones legales de uso.

## Software libre

El software libre otorga a los usuarios la libertad de usar, estudiar, modificar y distribuir el programa sin restricciones. Esto fomenta la colaboración y el desarrollo comunitario. Ejemplos destacados incluyen el sistema operativo GNU/Linux, el navegador Mozilla Firefox, el editor de imágenes GIMP y la suite ofimática LibreOffice. Las licencias más conocidas en este ámbito son la GPL (General Public License), la MIT License y la Apache License.

## Software propietario

El software propietario es desarrollado y distribuido bajo términos que restringen el acceso al código fuente y limitan la modificación, copia y redistribución. Normalmente requiere la compra de una licencia y está sujeto a las condiciones impuestas por el fabricante. Ejemplos de software propietario son Microsoft Windows, Adobe Photoshop, Microsoft Office y AutoCAD.

## Software freeware

El software freeware se distribuye de forma gratuita, permitiendo su uso sin coste alguno. Sin embargo, generalmente no permite modificar el código fuente ni redistribuir versiones modificadas. Ejemplos de freeware son el reproductor multimedia VLC, el programa de compresión WinRAR (en modo gratuito), y Skype. Aunque es gratuito, puede incluir publicidad o limitaciones en las funcionalidades.

## Software shareware

El software shareware se ofrece para su uso gratuito durante un periodo de prueba o con funcionalidades limitadas. Tras el periodo de prueba, el usuario debe adquirir una licencia para continuar utilizando el software o desbloquear todas sus características. Ejemplos de shareware incluyen WinZip, el editor de texto Sublime Text, y algunos programas antivirus como AVG. Este modelo permite evaluar el producto antes de decidir la compra.

### Ejercicio 9.1



Crea una tabla comparativa que tenga por columnas los distintos tipos de software (libre, propietario, freeware y shareware) y por filas el tipo de licencia bajo la que se distribuye. Añade los siguientes software a la tabla: GNU/Linux, Microsoft Windows, VLC, WinZip, LibreOffice, Adobe Photoshop, Skype, AVG, y GIMP.

## 9.2. Herramientas ofimáticas y de trabajo colaborativo

En esta sección se presentan las principales herramientas ofimáticas y de trabajo colaborativo utilizadas en entornos profesionales y educativos. Se analizarán sus características, ventajas y buenas prácticas para optimizar la gestión de documentos, la comunicación y la coordinación entre usuarios. El objetivo es proporcionar una visión general sobre cómo estas aplicaciones contribuyen a mejorar la productividad y facilitar el trabajo en equipo, tanto de forma presencial como remota.

### 9.2.1. Herramientas ofimáticas

Las herramientas ofimáticas son aplicaciones diseñadas para facilitar la creación, edición y gestión de documentos, hojas de cálculo, presentaciones y bases de datos. Entre las suites ofimáticas más populares se encuentran Microsoft Office, LibreOffice y Google Workspace. Estas herramientas permiten automatizar tareas administrativas, mejorar la productividad y garantizar la compatibilidad de formatos entre diferentes plataformas.

#### Procesadores de texto

Permiten crear y editar documentos escritos, como cartas, informes y manuales. Ejemplos: Microsoft Word, LibreOffice Writer y Google Docs. Ofrecen funciones de formato, corrección ortográfica, inserción de imágenes y tablas, y colaboración en tiempo real.

#### Hojas de cálculo

Facilitan el manejo de datos numéricos y la realización de cálculos automáticos. Ejemplos: Microsoft Excel, LibreOffice Calc y Google Sheets. Incluyen herramientas para crear gráficos, aplicar fórmulas y analizar información.

#### Presentaciones

Ayudan a crear diapositivas para exponer ideas de forma visual y estructurada. Ejemplos: Microsoft PowerPoint, LibreOffice Impress y Google Slides. Permiten insertar imágenes, vídeos, animaciones y colaborar en línea.

## Gestores de bases de datos

Permiten almacenar, organizar y consultar grandes volúmenes de información. Ejemplos: Microsoft Access, LibreOffice Base y Google Tables.

## Herramientas de trabajo colaborativo

Facilitan la comunicación y la colaboración entre usuarios, tanto en entornos locales como remotos. Incluyen aplicaciones de mensajería instantánea (Microsoft Teams, Slack), videoconferencia (Zoom, Google Meet), almacenamiento en la nube (Google Drive, OneDrive, Dropbox) y edición colaborativa de documentos.

Estas herramientas son fundamentales en el entorno profesional y educativo, ya que optimizan el trabajo en equipo, la gestión de la información y la productividad.

### 9.2.2. Trabajo colaborativo

El trabajo colaborativo consiste en la realización conjunta de tareas y proyectos por parte de varios usuarios, aprovechando herramientas tecnológicas que permiten compartir información, editar documentos en tiempo real y comunicarse de manera eficiente. Este enfoque fomenta la cooperación, la creatividad y la productividad, ya que facilita la integración de ideas y la resolución de problemas en grupo.

Las principales características del trabajo colaborativo incluyen:

- **Edición simultánea:** Varios usuarios pueden modificar un mismo documento al mismo tiempo, viendo los cambios en tiempo real.
- **Comunicación integrada:** Herramientas como chats, videollamadas y comentarios permiten la interacción directa entre los participantes.
- **Gestión de versiones:** Se mantiene un historial de cambios, permitiendo recuperar versiones anteriores y controlar el progreso del trabajo.
- **Acceso remoto:** Los usuarios pueden colaborar desde diferentes ubicaciones geográficas, accediendo a los recursos a través de Internet.
- **Asignación de tareas:** Es posible distribuir responsabilidades y hacer seguimiento del avance de cada miembro del equipo.

Ejemplos de plataformas de trabajo colaborativo incluyen Google Workspace, Microsoft 365, Slack, Trello, Asana y Notion. Estas herramientas son esenciales en entornos profesionales y educativos, ya que optimizan la gestión de proyectos, la comunicación y el trabajo en equipo.

## Google Workspace

Google Workspace es una suite de aplicaciones en la nube que incluye herramientas como Google Docs, Sheets, Slides, Drive, Gmail, Calendar y Meet. Permite la edición colaborativa en tiempo real, almacenamiento compartido, gestión de usuarios y comunicación integrada. Es ampliamente utilizada en entornos educativos y empresariales por su facilidad de acceso desde cualquier dispositivo y su integración con otros servicios de Google.

## Microsoft 365

Microsoft 365 ofrece aplicaciones como Word, Excel, PowerPoint, Outlook, Teams y OneDrive, tanto en versión de escritorio como en la nube. Destaca por sus potentes funciones de colaboración, integración con Windows y servicios empresariales, seguridad avanzada y gestión centralizada de usuarios. Es una solución muy utilizada en empresas y organizaciones por su versatilidad y compatibilidad con diferentes plataformas.

## Slack

Slack es una plataforma de mensajería y colaboración orientada a equipos de trabajo. Permite la comunicación mediante canales temáticos, mensajes directos, integración con aplicaciones externas (Google Drive, Trello, GitHub, etc.), intercambio de archivos y llamadas de voz y vídeo. Es especialmente útil para la gestión de proyectos y la coordinación de equipos distribuidos.

Existen muchas alternativas a Slack, como Microsoft Teams, Discord o Mattermost, que ofrecen funcionalidades similares para la comunicación y colaboración en equipo.

## Trello

Trello es una herramienta de gestión de proyectos basada en tableros, listas y tarjetas. Facilita la organización visual de tareas, la asignación de responsabilidades, el seguimiento del progreso y la colaboración entre miembros del equipo. Permite integrar otras aplicaciones y automatizar flujos de trabajo.

Alternativas a Trello incluyen Jira, Monday.com y ClickUp, que ofrecen funcionalidades avanzadas para la gestión de proyectos y equipos.

## Asana

Asana es una plataforma de gestión de tareas y proyectos que ayuda a planificar, organizar y supervisar el trabajo en equipo. Ofrece funciones para crear

proyectos, asignar tareas, establecer fechas límite, compartir archivos y comunicarse dentro de cada tarea. Es utilizada para mejorar la productividad y la coordinación en equipos de cualquier tamaño.

## Notion

Notion es una herramienta multifuncional que combina la gestión de notas, bases de datos, tareas y proyectos en una sola plataforma. Permite la colaboración en tiempo real, la creación de wikis, la documentación compartida y la organización flexible de la información. Es popular en entornos educativos y profesionales por su versatilidad y facilidad de uso.

## Ventajas del trabajo colaborativo

Las principales ventajas del trabajo colaborativo son:

- **Mejora de la productividad:** Permite que varios usuarios trabajen simultáneamente en un mismo proyecto, acelerando la finalización de tareas y optimizando el uso de recursos.
- **Fomento de la creatividad:** La colaboración facilita el intercambio de ideas y perspectivas, enriqueciendo el resultado final con aportaciones diversas.
- **Flexibilidad y accesibilidad:** Los participantes pueden acceder a los documentos y herramientas desde cualquier lugar y dispositivo, adaptándose a diferentes horarios y ubicaciones.
- **Comunicación eficiente:** Las herramientas colaborativas integran funciones de mensajería, videollamadas y comentarios, mejorando la coordinación y la toma de decisiones.
- **Gestión centralizada de la información:** Toda la documentación y los avances del proyecto se almacenan en un único lugar, facilitando el seguimiento y la recuperación de datos.
- **Control de versiones:** Se mantiene un historial de cambios, permitiendo revertir modificaciones y garantizar la integridad del trabajo.
- **Asignación clara de responsabilidades:** Es posible distribuir tareas y supervisar el progreso de cada miembro, mejorando la organización y el cumplimiento de objetivos.

Estas ventajas hacen que el trabajo colaborativo sea fundamental en entornos modernos, promoviendo la eficiencia, la innovación y el éxito de los proyectos.

## Buenas prácticas en el uso de herramientas colaborativas

Al utilizar herramientas colaborativas, es importante seguir una serie de buenas prácticas para maximizar su eficacia y evitar problemas de organización o seguridad:

- **Definir roles y permisos:** Asigna claramente los roles de cada usuario y limita los permisos según las necesidades, evitando accesos innecesarios a información sensible.
- **Establecer normas de comunicación:** Utiliza canales adecuados para cada tipo de mensaje y fomenta la comunicación clara y respetuosa entre los miembros del equipo.
- **Organizar la información:** Mantén los documentos y archivos bien estructurados, utilizando carpetas, etiquetas y nombres descriptivos para facilitar su localización.
- **Gestionar las versiones:** Revisa y controla el historial de cambios para evitar conflictos y asegurar la integridad de los documentos.
- **Realizar copias de seguridad:** Programa copias de seguridad periódicas para prevenir la pérdida de información importante.
- **Proteger la privacidad y la seguridad:** Utiliza contraseñas seguras, activa la autenticación en dos pasos y evita compartir información confidencial por canales inseguros.
- **Formar a los usuarios:** Proporciona formación sobre el uso de las herramientas y las políticas de colaboración para garantizar un uso correcto y eficiente.
- **Revisar y actualizar procesos:** Evalúa periódicamente las herramientas y los métodos de trabajo para adaptarlos a las necesidades cambiantes del equipo o proyecto.

Aplicar estas buenas prácticas contribuye a un entorno colaborativo seguro, organizado y productivo, facilitando el logro de los objetivos comunes.

### Ejercicio 9.2

Investiga y elabora un informe comparativo entre Google Workspace y Microsoft 365, destacando sus principales características, ventajas, desventajas y casos de uso recomendados. Incluye aspectos como herramientas disponibles, capacidades de colaboración, integración con otros servicios,



costos y seguridad.

Crea ese informe en una herramienta colaborativa (Google Docs, Microsoft Word Online, Notion, etc.) e invita a un compañero o compañera a revisarlo y añadir comentarios o sugerencias.

## 9.3. Utilidades de propósito general

Las utilidades de propósito general son aplicaciones que permiten realizar tareas esenciales para el funcionamiento, la seguridad y la gestión eficiente de los sistemas informáticos. A continuación se describen las principales categorías y ejemplos de cada una.

### 9.3.1. Antimalware

Las utilidades antimalware están diseñadas para detectar, bloquear y eliminar software malicioso (virus, troyanos, spyware, ransomware, etc.). Son fundamentales para proteger la integridad y la privacidad de los datos.

- **Ejemplos:** Windows Defender, Avast, Malwarebytes, ESET NOD32, ClamAV (Linux).
- **Funciones:** Análisis en tiempo real, escaneo bajo demanda, cuarentena de archivos sospechosos, actualización automática de firmas, protección web y de correo electrónico.
- **Buenas prácticas:** Mantener el software actualizado, realizar análisis periódicos y evitar descargar archivos de fuentes no confiables.

### 9.3.2. Correo electrónico

Las aplicaciones de correo electrónico permiten enviar, recibir, organizar y gestionar mensajes electrónicos. Incluyen funciones de filtrado, organización por carpetas, búsqueda avanzada y protección contra spam.

- **Ejemplos:** Microsoft Outlook, Mozilla Thunderbird, Gmail, Evolution (Linux), Apple Mail.
- **Funciones:** Gestión de múltiples cuentas, integración con calendarios y contactos, cifrado de mensajes, reglas de filtrado y protección antispam.
- **Buenas prácticas:** Utilizar contraseñas seguras, activar la autenticación en dos pasos y no abrir archivos adjuntos sospechosos.

### 9.3.3. Transferencia de ficheros

Las utilidades de transferencia de ficheros facilitan el envío y la recepción de archivos entre dispositivos locales o remotos, utilizando diferentes protocolos y métodos.

- **Ejemplos:** FileZilla (FTP/SFTP), WinSCP, rsync, scp, Dropbox, Google Drive, WeTransfer.
- **Funciones:** Transferencia segura, sincronización de carpetas, gestión de permisos, reanudación de descargas y subida de archivos grandes.
- **Buenas prácticas:** Utilizar protocolos seguros (SFTP, FTPS), cifrar archivos sensibles y verificar la integridad tras la transferencia.

### 9.3.4. Recuperación de datos

Las utilidades de recuperación de datos permiten restaurar archivos borrados accidentalmente, recuperar información de discos dañados o formateados y reparar sistemas de archivos.

- **Ejemplos:** Recuva, TestDisk, PhotoRec, EaseUS Data Recovery, R-Studio, ddrescue (Linux).
- **Funciones:** Escaneo profundo, recuperación selectiva, soporte para múltiples sistemas de archivos (NTFS, FAT, ext4, etc.), clonación de discos.
- **Buenas prácticas:** Actuar rápidamente tras la pérdida de datos, evitar sobrescribir el disco afectado y realizar copias de seguridad periódicas.

### 9.3.5. Mantenimiento del sistema

Las utilidades de mantenimiento ayudan a optimizar el rendimiento, limpiar archivos innecesarios, gestionar el arranque y monitorizar el estado del sistema.

- **Ejemplos:** CCleaner, BleachBit, Glary Utilities, Windows Disk Cleanup, Monitor de recursos, top/htop (Linux).
- **Funciones:** Limpieza de archivos temporales, gestión de programas de inicio, desfragmentación de disco, monitorización de recursos y actualización de controladores.
- **Buenas prácticas:** Realizar mantenimientos periódicos, actualizar el sistema y eliminar software innecesario.

### 9.3.6. Otras utilidades

Existen muchas otras utilidades que cubren necesidades específicas, como la compresión de archivos, la gestión de contraseñas, la virtualización, la edición multimedia y la impresión de documentos.

- **Compresión de archivos:** WinRAR, 7-Zip, PeaZip, tar/gzip (Linux).
- **Gestores de contraseñas:** KeePass, LastPass, Bitwarden.
- **Virtualización:** VirtualBox, VMware Workstation, QEMU.
- **Edición multimedia:** Audacity, GIMP, Adobe Photoshop, VLC Media Player.
- **Impresión y gestión de PDFs:** Adobe Acrobat Reader, Foxit Reader, CUPS (Linux).

#### Ejercicio 9.3



Investiga y elabora un informe sobre las mejores prácticas para mantener la seguridad y el rendimiento de un sistema informático utilizando utilidades de propósito general. Incluye recomendaciones específicas para cada categoría de utilidad (antimalware, correo electrónico, transferencia de ficheros, recuperación de datos, mantenimiento del sistema y otras utilidades). Presenta el informe en un formato claro y estructurado, destacando la importancia de cada práctica recomendada.

## Resumen

En este capítulo se ha revisado la explotación de aplicaciones informáticas de propósito general, abordando los distintos tipos de software, sus requisitos y licencias, así como las principales herramientas ofimáticas y colaborativas empleadas en entornos profesionales y educativos. Se han analizado utilidades de propósito general que contribuyen al mantenimiento, seguridad y gestión eficiente de los sistemas informáticos. El conocimiento y la correcta utilización de estas aplicaciones permiten optimizar el trabajo, mejorar la productividad y garantizar la protección de la información, adaptándose a las necesidades cambiantes de los usuarios y organizaciones.



# Índice de figuras

|      |                                                                                          |     |
|------|------------------------------------------------------------------------------------------|-----|
| 3.1  | Placa base de un ordenador . . . . .                                                     | 16  |
| 3.2  | Diagrama de un procesador . . . . .                                                      | 17  |
| 3.3  | Ejemplo de tarjeta de sonido externa . . . . .                                           | 20  |
| 3.4  | Distintos tipos de conectores USB . . . . .                                              | 22  |
| 3.5  | Router WiFi . . . . .                                                                    | 28  |
| 3.6  | Ejemplo de topología en bus . . . . .                                                    | 30  |
| 3.7  | Ejemplo de topología en estrella . . . . .                                               | 30  |
| 3.8  | Ejemplo de topología en anillo . . . . .                                                 | 31  |
| 3.9  | Ejemplo de topología en malla . . . . .                                                  | 31  |
| 3.10 | Ejemplo de topología en árbol . . . . .                                                  | 32  |
| 3.11 | Kit de herramientas para cableado . . . . .                                              | 33  |
| 3.12 | Distribución de hilos en un conector RJ45 . . . . .                                      | 35  |
| 3.13 | Ejemplo de un cable que usa el estándar T568B . . . . .                                  | 36  |
| 5.1  | Estructura de archivos en Windows . . . . .                                              | 75  |
| 5.2  | Explorador de archivos en Windows . . . . .                                              | 76  |
| 5.3  | Estructura de archivos en Linux . . . . .                                                | 78  |
| 5.4  | Gestor de archivos en Linux (Ubuntu) . . . . .                                           | 80  |
| 6.1  | Gestión de usuarios en Ubuntu . . . . .                                                  | 101 |
| 7.1  | Wireshark, una herramienta popular para la captura y análisis de tráfico de red. . . . . | 146 |
| 7.2  | Modelo de capas TCP/IP y ejemplos de protocolos en cada capa. . . . .                    | 148 |
| 7.3  | Configuración de adaptadores de red en Windows . . . . .                                 | 151 |
| 7.4  | Configuración de adaptadores de red en Ubuntu . . . . .                                  | 152 |



# **Índice de cuadros**

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| 3.2 Comparativa de los principales tipos de redes . . . . .               | 27 |
| 4.1 Hitos históricos en la evolución de los sistemas operativos . . . . . | 41 |