



EL ECOSISTEMA DE LOS SISTEMAS OPERATIVOS Y SUS APLICACIONES

**INNOVACIÓN Y TRANSFORMACIÓN PEDAGÓGICA
EN LA GESTIÓN DE ENTORNOS INFORMÁTICOS MODERNOS**



FRANKLIN MARCELO ALVARADO CANALES

ISBN: 978-9907-818-05-5



9 789907 818055



El Ecosistema de los Sistemas Operativos y sus Aplicaciones

Innovación y Transformación Pedagógica en la Gestión de Entornos Informáticos Modernos

Autor:

Franklin Marcelo, Alvarado Canales

Orcid: <https://orcid.org/0009-0000-4397-230X>

Correo: franklinalvaradoc21@gmail.com

Filial: Unidad Educativa Fiscal Guayaquil

Editorial “ACACFESA SAS”

La presente obra fue revisada por 2 pares académicos externos ciegos conforme al proceso editorial de ACACFESA SAS.

Los rigurosos procedimientos editoriales de ACACFESA SAS garantizan la selección de manuscritos por sus aportes significativos al conocimiento y cualidades científicas.

Editorial: ACACFESA SAS.

Sello editorial: 978-9907-818-05

Teléfono: (+593) 998955446

Web: <https://acacfesa.com/editorial/index.php/1>

ISBN: 978-9907-818-05-5

Doi: <https://doi.org/10.70577/9cafhw16/ACACFESA.EDITORIAL>

Año: Abril 2026

Aviso Legal

El contenido de esta obra, que incluye ilustraciones, textos, tablas, gráficos, cuadros y referencias bibliográficas, es responsabilidad única del autor o autores. Lo que se ha dicho, los criterios y los datos no reflejan necesariamente la posición institucional ni el pensamiento de la Editorial ACACFESA SAS.

Derechos de Autor ©

Este documento se publica conforme los términos y condiciones de la EDITORIAL ACACFESA SAS

Esta obra está bajo una licencia internacional



[Creative Commons Atribución-NoComercial-CompartirIgual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

La "Editorial ACACFESA SAS " y todos sus autores tienen la propiedad única de los derechos de autor y de propiedad intelectual e industrial relacionados con el contenido de esta publicación. La reproducción total o parcial de esta obra, su almacenamiento en sistemas informáticos, su tratamiento digital y cualquier tipo de distribución, transmisión o comunicación pública por medios electrónicos, ópticos, mecánicos, químicos, fotográficos o de grabación están prohibidos. Esto se

encuentra bajo las sanciones que establece la legislación vigente a menos que se cuente con la autorización previa y por escrito de los titulares del copyright.

Queda excluido únicamente el uso con fines académicos o de investigación científica, siempre que no busque objetivos comerciales y se ejecute sin remuneración, debiendo mencionarse en todo momento a la fuente editorial correspondiente. Los autores son los únicos responsables de las opiniones expresadas en los diferentes capítulos, las cuales no necesariamente representan el punto de vista institucional de la editorial.

Constancia de Arbitraje

La Editorial ACACFESA SAS, certifica que este libro es el resultado de un estudio llevado a cabo por los autores, el cual fue evaluado por jurados expertos bajo el sistema de doble ciego, tanto en contenido como en forma. Asimismo, se llevó a cabo un análisis del método de investigación, paradigma y enfoque; a partir de la matriz epistémica adoptada por los autores, utilizando las normas APA, Séptima Edición y el proceso de antiplagio en línea turnitinedu para asegurar que la obra fuera científica.

DEDICATORIA

A mi esposa, Shirley Malusín e hija, Rebeca Alejandra por su amor incondicional

A mis padres, Don José Román Alvarado Maruri (+2025) y Doña Lucía Canales Mera, por darme la vida y su amor incondicional, sacrificio y apoyo constante a lo largo de mi formación académica, quienes inspiraron a no rendirme, incluso cuando el camino parecía difícil, por ser mi refugio en los momentos de las adversidades, y por recordarme que la meta final es tan importante como el camino recorrido.

AGRADECIMIENTO

Al Ser Supremo, por ser mi guía espiritual y mi fortaleza en cada paso del camino.

A todos los docentes, investigadores y estudiantes que han inspirado este trabajo con sus experiencias, preguntas y aportes en torno a la relación entre tecnología y aprendizaje. También a quienes, desde la práctica cotidiana, demuestran que la innovación no depende solo de las herramientas, sino de la capacidad de pensar críticamente sobre su uso y sentido.

AVISO DE DESCARGO DE RESPONSABILIDAD (DISCLAIMER)

Finalidad del contenido: El presente libro, El Ecosistema de los Sistemas Operativos y sus Aplicaciones: Innovación y Transformación Pedagógica en la Gestión de Entornos Informáticos Modernos, tiene un propósito estrictamente educativo, pedagógico y profesional. Su contenido ha sido desarrollado para proporcionar una comprensión integral del ecosistema de los sistemas operativos, sus aplicaciones y su papel dentro de la gestión de infraestructuras tecnológicas contemporáneas, con especial énfasis en el contexto ecuatoriano.

Responsabilidad legal: El autor no se responsabiliza por el uso indebido, ilegal o negligente de la información, metodologías, herramientas o configuraciones descritas en esta obra. La implementación de técnicas relacionadas con la administración de sistemas, auditoría informática o ciberseguridad fuera de entornos controlados y sin la debida autorización del propietario de los activos tecnológicos puede constituir una infracción legal, conforme a lo establecido en la normativa vigente, incluido el Código Orgánico Integral Penal (COIP).

Actualización técnica: Dado el carácter dinámico de los sistemas operativos, las tecnologías de información y comunicación y sus aplicaciones en el entorno digital, el autor no garantiza la ausencia total de errores u omisiones, ni asume responsabilidad por posibles desactualizaciones que puedan surgir con posterioridad a la publicación de la obra.

NOTIFICACIÓN DE PROPIEDAD INTELECTUAL Y USO ACADÉMICO

Marcas registradas: Todos los nombres de sistemas operativos, plataformas tecnológicas, aplicaciones, logotipos y marcas comerciales mencionados en este libro forman parte del ecosistema tecnológico global y son propiedad de sus respectivos titulares. Su uso en esta obra responde exclusivamente a fines educativos, analíticos y de referencia técnica, en el marco del derecho de cita, sin intención de vulnerar derechos de propiedad intelectual ni de atribuirse su titularidad.

Derechos de autor del material visual: Los esquemas, diagramas, modelos conceptuales y representaciones gráficas del ecosistema de sistemas operativos han sido desarrollados por el autor con fines académicos. Se prohíbe su reproducción total o parcial con fines comerciales sin autorización expresa. Estos recursos tienen como objetivo facilitar la comprensión de la interacción entre sistemas, aplicaciones y entornos informáticos modernos.

Originalidad de la obra: La presente obra es original y se centra en el análisis estructurado del ecosistema de los sistemas operativos y sus aplicaciones. No reproduce contenidos de terceros de forma indebida. Las coincidencias conceptuales con otras publicaciones responden a la naturaleza compartida del conocimiento en áreas de los sistemas operativos y sus aplicaciones.

FICHA METODOLÓGICA DEL LIBRO

La obra ha sido diseñada bajo un enfoque educativo integral que articula conocimientos técnicos, criterios normativos y competencias profesionales necesarias para comprender y gestionar el ecosistema de los sistemas operativos en entornos informáticos modernos.

LAS METODOLOGÍAS APLICADAS SON:

Aprendizaje autónomo: Se promueve la exploración independiente del ecosistema tecnológico mediante el análisis de documentación técnica, estándares internacionales y plataformas reconocidas en el ámbito de la ciberseguridad y administración de sistemas.

Aprendizaje invertido (Flipped Learning): El lector accede primero a los fundamentos teóricos del funcionamiento, requerimientos y aplicaciones de los sistemas operativos, para posteriormente aplicarlos en entornos prácticos y escenarios reales.

Aprendizaje autorreflexivo: Se integran análisis críticos y dilemas éticos relacionados con la gestión de sistemas, la protección de datos y el uso responsable de tecnologías dentro del ecosistema digital.

Aprendizaje basado en proyectos (ABP): El contenido guía al lector en la construcción de soluciones prácticas, como la implementación de entornos operativos seguros, políticas de gestión y protocolos de buenas prácticas en infraestructura tecnológica.

Aprendizaje en contextos reales: Los casos presentados reflejan situaciones reales en la administración de sistemas operativos, soporte técnico y gestión de infraestructuras en el contexto ecuatoriano y global.

Simulación de roles (Role Playing): Se plantean escenarios donde el lector asume distintos roles dentro del ecosistema tecnológico (administrador de sistemas, usuario final, auditor, especialista en soporte), fortaleciendo habilidades técnicas y comunicativas.

Metodología STEAM: El libro integra Ciencia, Tecnología, Ingeniería y Matemática como base del análisis del ecosistema de sistemas operativos, incorporando además un enfoque transversal en ética digital, normativa legal y responsabilidad profesional.

EPIGRAFE

"Una computadora merecería ser llamada inteligente si pudiera engañar a un humano haciéndole creer que es humana."

Alan Mathison Turing (1912–1954)

Índice de contenido

DEDICATORIA	vi
AGRADECIMIENTO	vii
EPÍGRAFE.....	xi
ÍNDICE DE TABLAS	xv
ÍNDICE DE FIGURAS.....	xvi
PROLOGO	xviii
INTRODUCCIÓN	1
CAPÍTULO I.....	3
COMPRENSIÓN DEL ENTORNO TECNOLÓGICO	3
FUNDAMENTOS DEL HARDWARE, SOFTWARE Y LA INFORMÁTICA	3
1.1. FUNDAMENTO DE INFORMÁTICA	3
1.2. TERMINOLOGÍA DE LA PARTE CONCEPTUAL DE LA INFORMÁTICA.....	4
1.3. FUNDAMENTOS DEL COMPUTADOR.....	5
1.3.1. FUNCIONES BÁSICAS DEL COMPUTADOR.....	6
1.4. MODELO CONCEPTUAL DEL COMPUTADOR DE JOHN VON NEUMANN.....	6
1.5. COMPONENTES DEL HARDWARE.....	7
1.5.1. CLASIFICACIÓN DEL HARDWARE	8
1.6. COMPONENTES DEL SOFTWARE	10
1.6.1. CLASIFICACIÓN DEL SOFTWARE	11
1.7. RELACIÓN ENTRE EL HARDWARE Y EL SOFTWARE	12
1.8. IMPORTANCIA EN EL ENTORNO INFORMATICO MODERNO.....	12
1.9. USO BÁSICO DE UN SISTEMA INFORMÁTICO.....	13
1.10. DIFERENCIA ENTRE DATO E INFORMACIÓN	13
1.11. USO INCORRECTO DEL SISTEMA INFORMÁTICO	14
CAPÍTULO II.....	16
PARÁMETROS TÉCNICOS Y COMPATIBILIDAD	16
LA BASE DE LA TOMA DE DECISIONES TECNOLÓGICAS	16
2.1. DECISIONES TECNOLÓGICAS: HARDWARE & SOFTWARE	16
2.2. COMPATIBILIDAD: EL PRINCIPIO DE INTEGRACIÓN TECNOLÓGICA.....	16
2.2.1. TIPOS DE COMPATIBILIDAD	16
2.3. ARQUITECTURA DEL SISTEMA	17
2.3.1. TIPOS DE ARQUITECTURA	17
2.4. RENDIMIENTO DEL SISTEMA	18
2.5. COMPARACIÓN DEL RENDIMIENTO DEL SISTEMA.....	19

2.6.	CAPACIDAD DEL SISTEMA.....	19
2.6.1.	TIPOS DE CAPACIDADES DEL SISTEMA.....	19
2.7.	EVALUACIÓN DE RESTRICCIONES TÉCNICAS.....	20
2.8.	INTEGRACIÓN DE PARÁMETROS PARA LA TOMA DE DECISIONES.....	21
2.8.1.	IMPORTANCIA ESTRATÉGICA.....	21
2.9.	ANÁLISIS DE REQUERIMIENTOS DEL ENTORNO Y DEL USUARIO.....	24
2.10.	IDENTIFICACIÓN DE NECESIDADES DEL USUARIO.....	24
2.10.1.	TIPOS DE USUARIOS.....	25
2.11.	SOLUCIÓN PRIMARIA NO AVANZADA:.....	29
2.12.	ANÁLISIS DEL ENTORNO ORGANIZACIONAL.....	31
2.13.	ENTORNO ORGANIZACIONAL FACTORES QUE INFLUYEN.....	32
2.14.	CRITERIOS PARA SELECCIONAR COMBINACIONES ÓPTIMAS DE HARDWARE Y SOFTWARE.....	35
2.15.	ARQUITECTURA Y FUNCIONAMIENTO GENERAL DE LOS SISTEMAS OPERATIVOS.....	40
2.15.1.	CLASIFICACIÓN FUNCIONAL DE LOS SISTEMAS OPERATIVOS.....	41
2.15.2.	CLASIFICACIÓN GENERAL DE LOS SISTEMAS OPERATIVOS.....	44
2.16.	LICENCIAMIENTO Y MODELOS DE DISTRIBUCIÓN DE SOFTWARE.....	47
2.16.1.	ASPECTOS LEGALES, TÉCNICOS Y ESTRATÉGICOS EN EL USO DEL SOFTWARE.....	47
2.16.2.	CLASIFICACIÓN DE LOS TIPOS DE LICENCIAS DE SOFTWARE.....	51
2.17.	ESQUEMA INTEGRAL DE LOS SISTEMAS OPERATIVOS.....	55
CAPÍTULO III.....		57
ARQUITECTURA Y TIPOS DE SISTEMAS OPERATIVOS.....		57
ESTRUCTURA INTERNA Y CLASIFICACIÓN FUNCIONAL DE LOS SISTEMAS OPERATIVOS.....		57
3.1.	FUNCIONES FUNDAMENTALES DEL SISTEMA OPERATIVO.....	58
3.2.	ESQUEMA FUNCIONAL DE LOS SISTEMAS OPERATIVOS.....	66
3.3.	ARQUITECTURA DE LOS SISTEMAS OPERATIVOS.....	66
3.3.1.	TIPOS DE ARQUITECTURA DE LOS SISTEMAS OPERATIVOS.....	67
3.4.	EL NÚCLEO KERNEL.....	71
3.5.	REPRESENTACIÓN CONCEPTUAL DEL KERNEL.....	72
CAPÍTULO IV.....		73
INSTALACIÓN DE SISTEMAS OPERATIVOS Y SUS APLICACIONES.....		73
4.1.	PROCEDIMIENTOS, MÉTODOS Y BUENAS PRÁCTICAS PARA UNA IMPLEMENTACIÓN EFICIENTE.....	73

4.2.	REQUISITOS TÉCNICOS Y DE COMPATIBILIDAD	74
4.2.1.	TIPOS DE REQUISITOS.....	74
4.3.	MÉTODOS DE INSTALACIÓN DE LOS SISTEMAS OPERATIVOS	75
4.4.	SISTEMAS DE ARCHIVOS Y GESTIÓN DEL ALMACENAMIENTO.....	89
4.4.1.	CONCEPTO DE SISTEMA DE ARCHIVOS	89
4.5.	ESTRUCTURA LÓGICA DEL ALMACENAMIENTO	90
4.6.	TIPOS DE SISTEMAS DE ARCHIVOS	91
CAPÍTULO V		98
ACTUALIZACIÓN Y MANTENIMIENTO DE SISTEMAS		98
5.1.	GESTIÓN ESTRATÉGICA PARA LA CONTINUIDAD OPERATIVA Y LA SEGURIDAD INFORMÁTICA.....	98
5.2.	CONCEPTO DE ACTUALIZACIÓN Y MANTENIMIENTO	98
5.2.1.	TIPOS DE ACTUALIZACIONES	99
5.3.	GESTIÓN DE RIESGOS EN ENTORNOS INFORMÁTICO MODERNOS.....	111
5.3.1.	TIPOS DE RIESGOS	111
CAPÍTULO VI.....		114
SISTEMAS EN RED.....		114
INTERCONECTIVIDAD, RECURSOS COMPARTIDOS Y GESTIÓN DISTRIBUIDA EN ENTORNOS INFORMÁTICOS.....		114
6.1.	SISTEMAS EN RED	114
6.1.1.	TIPOS DE REDES.....	116
6.2.	REDES POR ALCANCE	116
6.3.	REDES ESPECIALIZADAS.....	120
6.4.	REDES POR TIPO DE CONEXIÓN	122
6.5.	REDES POR RELACIÓN FUNCIONAL.....	123
6.6.	REDES POR DIRECCIONALIDAD	124
6.7.	REDES POR GRADO DE AUTENTIFICACIÓN	126
6.8.	REDES POR GRADO DE DIFUSIÓN	128
6.9.	REDES POR TOPOLOGÍA	129
REFERENCIAS BIBLIOGRÁFICAS		135
GLOSARIO DEL ECOSISTEMA DE LOS SISTEMAS OPERATIVOS Y SUS APLICACIONES		137
ACERCA DEL AUTOR.....		146
SINOPSIS		148

ÍNDICE DE TABLAS

Tabla 1. Componentes	19
Tabla 2. Capacidad de expansión	38
Tabla 3. Tipos de requisitos	74

ÍNDICE DE FIGURAS

Figura 1 Informática	4
Figura 2 Computador.....	5
Figura 3 Funciones Básicas del Computador	6
Figura 4 Modelo Conceptual del Computador de John Von Neumann	7
Figura 5 Componentes del Hardware.....	7
Figura 6 Clasificación del Hardware	8
Figura 7 Unidad Central de Procesamiento (CPU).....	9
Figura 8 Dispositivos de Almacenamiento	9
Figura 9 Dispositivos de Entrada.....	10
Figura 10 Dispositivos de Salida.....	10
Figura 11 Clasificación del Software.....	11
Figura 12 Relación entre el Hardware y el Software	12
Figura 13 Enfoque Integral del Entorno Informático Moderno	13
Figura 14 Tipos de Compatibilidad	17
Figura 15 Arquitectura del Sistema Representación Conceptual	18
Figura 16 Tipos De Capacidades Del Sistema.....	20
Figura 17 Integración de Parámetros para la toma de decisiones.....	22
Figura 18 Esquema de Decisión	23
Figura 19 Análisis de Requerimientos del Entorno y del Usuario.....	24
Figura 20 Tipos de Usuarios	25
Figura 21 Solución Primaria	28
Figura 22 Análisis del Entorno Organizacional	31
Figura 23 Combinaciones Óptimas de Hardware y Software	35
Figura 23 Modelo de Evaluación Simplificado.....	39
Figura 24 Funcionamiento general de los Sistemas Operativos	40
Figura 25 Clasificación Funcional de los Sistemas Operativos.....	41
Figura 26 Clasificación General de los Sistemas Operativos	44
Figura 27 Licenciamiento y modelos de distribución de software	47
Figura 28 Clasificación de los Tipos de Licencias de Software.....	51
Figura 29 Esquema Integral De Los Sistemas Operativos.....	55
Figura 30 Funciones Fundamentales del Sistema Operativo	58
Figura 31 Esquema Funcional de los Sistemas Operativos	66
Figura 32 Tipos de Arquitectura de los Sistemas Operativos	67

Figura 33 El núcleo KERNEL engranaje del Sistemas.....	71
Figura 34 Representación Conceptual del Kernel.....	72
Figura 35 Métodos de Instalación de Sistemas Operativos.....	75
Figura 36 Instalación Manual de Sistema Operativo.....	77
Figura 37 Instalación Automatizada de Sistema Operativo	79
Figura 38 Instalación por red	81
Figura 39 Instalación mediante imágenes (clonación).....	83
Figura 40 Instalación de aplicaciones.....	85
Figura 41 Sistemas de archivos y gestión del almacenamiento.....	89
Figura 42 Estructura lógica del almacenamiento	90
Figura 43 Tipos de sistemas de archivos	91
Figura 44 Tipos de actualizaciones.....	99
Figura 45 Tipos de riesgos.....	111
Figura 46 Sistemas en red	115
Figura 47 Tipos de redes.....	116
Figura 48 Redes por alcance.....	117
Figura 49 Redes especializadas	121
Figura 50 Redes por tipos de conexión.....	123
Figura 51 Redes por relación funcional	124
Figura 52 Redes por direccionalidad.....	125
Figura 53 Redes por grado de autenticación	127
Figura 54 Redes por grado de difusión.....	128
Figura 55 Redes por topología.....	129
Figura 56 Topología física	132
Figura 57 Topología lógica.....	134

PROLOGO

Vivimos en una época donde la tecnología ha dejado de ser una herramienta opcional para convertirse en el eje que articula gran parte de nuestra vida cotidiana. Desde las actividades más simples hasta los procesos más complejos dentro de organizaciones e industrias, los sistemas informáticos están presentes, operando silenciosamente y sosteniendo la dinámica del mundo moderno.

En este escenario, los sistemas operativos y sus aplicaciones no solo cumplen una función técnica, sino que forman parte de un ecosistema amplio, interconectado y en constante evolución. Comprender este ecosistema ya no es exclusivo de especialistas; es una necesidad para todo aquel que desee desenvolverse con criterio y eficiencia en un entorno digital cada vez más exigente.

“El Ecosistema de los Sistemas Operativos y sus Aplicaciones” surge precisamente como una respuesta a esa necesidad. Esta obra no se limita a enseñar el uso de herramientas, sino que invita al lector a desarrollar una visión integral de la informática, entendiendo cómo interactúan el hardware, el software y los usuarios en contextos reales. Aquí no solo se aprende a instalar o configurar sistemas, sino a interpretarlos, analizarlos y gestionarlos de manera estratégica.

A lo largo de sus páginas, el lector encontrará un recorrido cuidadosamente estructurado que parte desde los fundamentos básicos de la informática hasta llegar a aspectos más complejos como la gestión de sistemas operativos, el mantenimiento, la toma de decisiones técnicas y el cumplimiento de normativas. Todo ello acompañado de ejemplos prácticos que reflejan situaciones reales, permitiendo conectar el conocimiento con la experiencia cotidiana.

Uno de los grandes aciertos de esta obra es su enfoque humano. Nos recuerda que, detrás de cada sistema, cada configuración y cada línea de código, existe una intención, una decisión y una mente que guía el proceso. La tecnología, por sí sola, no resuelve problemas; es el ser humano quien le da sentido, dirección y propósito.

En un mundo donde muchos proyectos tecnológicos fracasan no por falta de herramientas, sino por una comprensión limitada de su entorno, este libro se convierte en una guía valiosa. No forma únicamente operadores, sino profesionales capaces de observar, cuestionar y transformar su realidad mediante el uso adecuado de la tecnología.

Este no es un libro para memorizar, sino para comprender. No es un punto de llegada, sino un compañero en el camino del aprendizaje continuo. Porque en el ámbito tecnológico, cada avance abre nuevas preguntas, y cada respuesta construye una oportunidad.

Finalmente, estimado lector, recuerda que las máquinas ejecutan instrucciones, pero las ideas nacen en la mente humana. Este libro es una invitación a fortalecer esa capacidad de pensar, decidir y crear en un entorno donde la tecnología es el medio, pero el verdadero protagonista siempre serás tú.

INTRODUCCIÓN

En estos días, la informática, es el estudio de muchas personas, especialista y novatos que buscan cada día estar en el auge de los sistemas operativos informáticos y sus aplicaciones en el entorno global, ha dejado de ser un campo exclusivo para convertirse en un componente esencial de la vida diaria de cada uno de nosotros, está inmersa en la educación, en la industria, en el comercio y en cada una de las ramas del estudio y campo en general

En este sentido existe un ecosistema de los sistemas operativos y sus aplicaciones bien estructurado, donde cada componente cumple una función específica y cada decisión técnica puede marcar la diferencia entre la eficiencia y el colapso en el entorno de los sistemas operativos informáticos modernos.

Este texto nace precisamente de la necesidad de comprender ese ecosistema de los sistemas operativos y sus aplicaciones en profundidad para conocer cada uno de sus componentes, no es únicamente manipular los sistemas operativos si no que desarrollar una visión general integral de conocimiento que permitan y puedan interactuar entre el hardware y el software, además para que sepan instalar y configurarla los sistemas operativos contemporáneos

En esta trayectoria de conocimiento amigo lector usted podrá adquirir los conocimientos básico e intermedio de los conocimientos técnicos y apropiados, donde tendrá la capacidad de analizar la toma de decisiones en las gestiones del entorno de los sistemas operativos informáticos modernos

Los mayores desafíos en estos momentos contemporáneos es el manejo de todos los instrumentos tecnológico moderno y una correcta implementación en las áreas técnicas específicas

El libro contiene fundamentos integrados teóricos y prácticos con ejemplo de la vida diaria en su entorno de trabajo, lo cual establecerá los conocimientos solidos cuando sean de realizar sus trabajos a ellos encomendados

Amigo lector aquí encontrará desde los principios básicos de la ciencia informática hasta la estructura completa de un computador, además contendrá ejemplos claros y precisos de las instalaciones, configuraciones, actualizaciones, y el correcto mantenimiento preventivo, correctivo,

con el manejo de la gestión y Las normativas legales de los diferentes tipos de software, que cada uno tiene su configuración y requerimientos propios.

Tengo que indicar que las tecnologías modernas, no es un principio, ni un fin, si no son estructuras que van resolver los diferentes tipos de problemas, y así podrán optimizar todos los procesos y recursos para mejorar la calidad de vida de los usuarios. Por consiguiente, este texto, no busca formar solamente simples operadores de sistemas informáticos modernos, sino profesionales capaces de interpretar, analizar y transformar su entorno mediante el uso adecuado de las herramientas informáticas modernas en el entorno digital que nos rodea.

Para Finalizar, solamente me queda decirle, que este libro no es, ni será, un punto de partida o un de llegada, si no será la meta que buscas en el proceso tecnológico moderno, pero recuerda quien da las instrucciones es el ser humano, la maquina solo obedece lo que tu mente crea y tus dedos escriben

CAPÍTULO I

COMPRENSIÓN DEL ENTORNO TECNOLÓGICO

FUNDAMENTOS DEL HARDWARE, SOFTWARE Y LA INFORMÁTICA



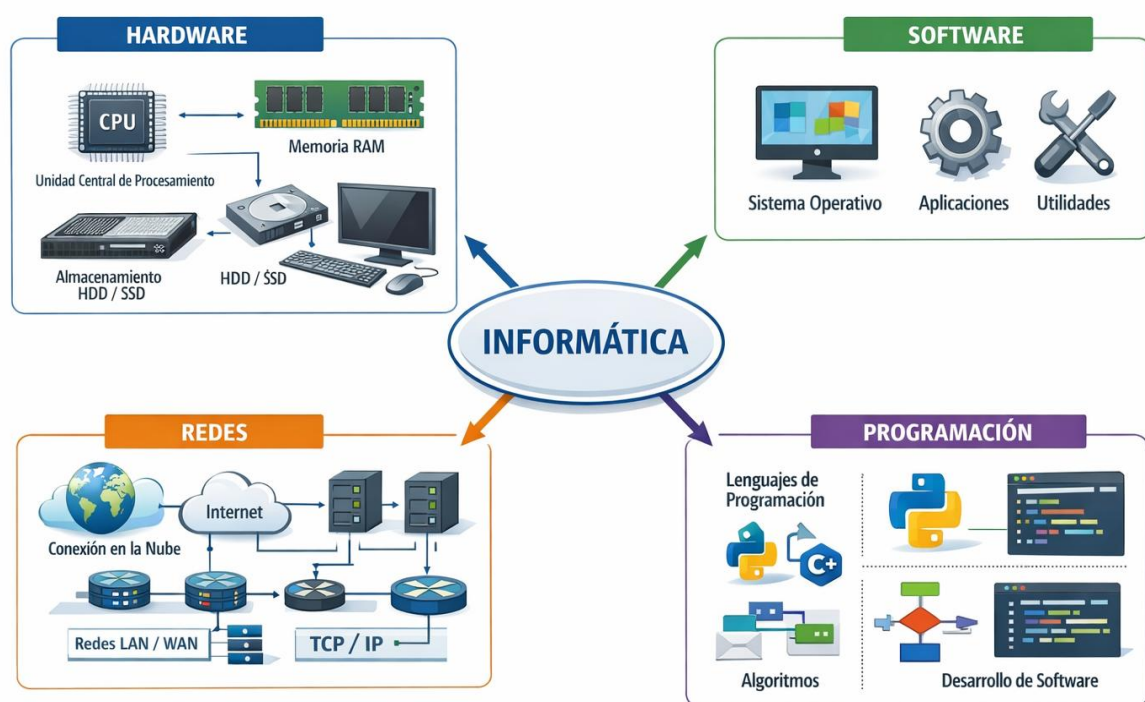
1.1. FUNDAMENTO DE INFORMÁTICA

La informática es la ciencia que constituye la disciplina la cual estudia el tratamiento automático de la información a través de los ordenadores, mediante los sistemas computacionales, el desarrollo de la ciencia informática ha estado vinculado desde la década de los años 60, con la evolución tecnológica y a la necesidad del ser humano de procesar grandes cantidades de datos de manera correcta y eficiente.

El término Informática fue acuñado en Francia en el año de 1962 por el Ingeniero Francés Philippe Dreyfus, la palabra informática se deriva de la combinación de las palabras griegas **INFO**=información y **MÁTICA**=automática, lo que significa información automática. Desde esa época, hasta estos días, la informática ha ido evolucionando y hoy por hoy sea convertido en un pilar fundamental de la cultura y sociedad digital.

Por consiguiente, para comprender y entender el ecosistema de los sistemas operativos y sus aplicaciones, se hace necesario establecer conceptos claros y preciso de lo que es el término informática

Figura 1
Informática



1.2. TERMINOLOGÍA DE LA PARTE CONCEPTUAL DE LA INFORMÁTICA

- 1). - **DATO:** Es la representación simbólica de un hecho o entidad sin procesar.
- 2). - **INFORMACIÓN:** Es el resultado de los procesamientos de datos con significado y utilidad real.
- 3). - **SISTEMA:** Es un conjunto de varios elementos interrelacionados entre sí, que funcionan como un todo.
- 4). - **SISTEMA INFORMÁTICO:** Es la integración entre el hardware y el software y los usuarios para procesar todo tipo de información.

Para reforzar a parte conceptual se puede decir que la informática permite comprender y entender, que la tecnología no se limita, sino que se involucra en los procesos, en las personas y en todas las estructuras organizativas tecnológicas.

1.3. FUNDAMENTOS DEL COMPUTADOR

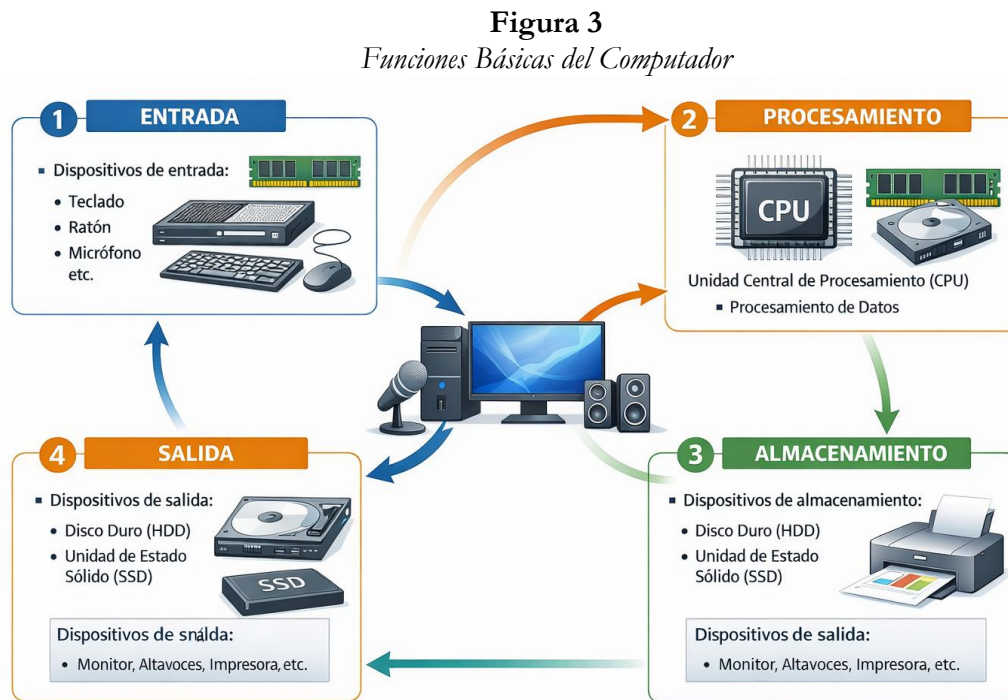
Un computador es una máquina electrónica capaz de recibir, procesar y almacenar información, dadas por el usuario

Figura 2
Computador



1.3.1. FUNCIONES BÁSICAS DEL COMPUTADOR

Un sistema computacional opera bajo cuatro básicas:



- 1).- **ENTRADA:** Recepción de datos
- 2).- **PROCESAMIENTO:** Transformación de datos
- 3).- **ALMACENAMIENTO:** Conservación de información
- 4).- **SALIDA:** Presentación de resultados

1.4. MODELO CONCEPTUAL DEL COMPUTADOR DE JOHN VON NEUMANN

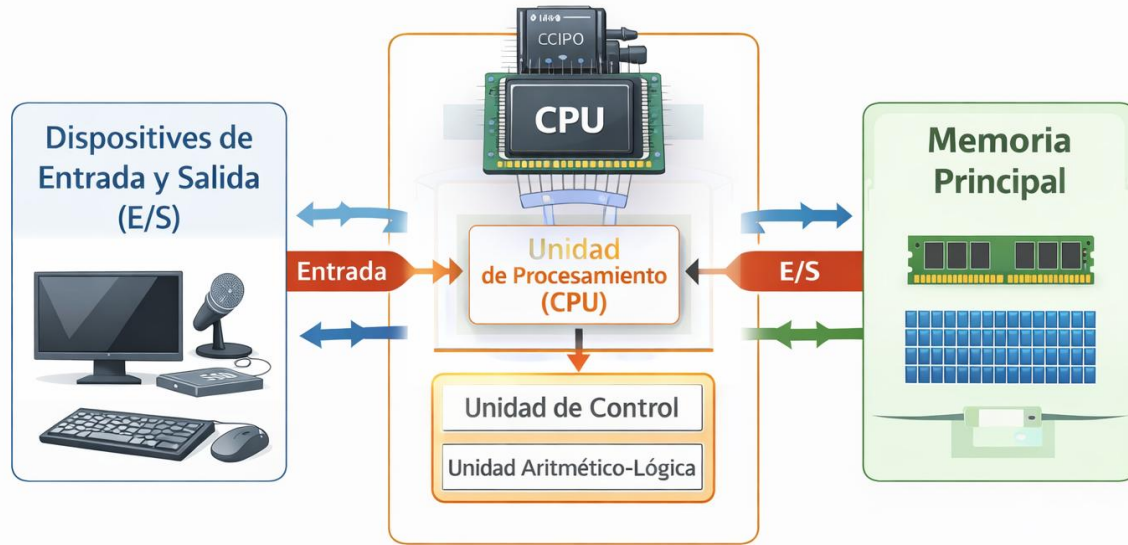
Dentro de los modelos conceptuales del computador, el más influyente en la arquitectura de computadores fue el modelo de John Von Neumann, el cual fue propuesto en año de 1945.

Este modelo establece que un computador está compuesto por:

- 1).- **Unidad de Procesamiento (CPU)**
- 2).- **Memoria Principal**
- 3).- **Dispositivos de Entrada y Salida (E/S)**

Este modelo se fundamenta principalmente, porque radica, en que tanto los datos como las instrucciones se almacenan en la misma memoria, lo que permite una ejecución secuencial de programas.

Figura 4
Modelo Conceptual del Computador de John Von Neumann



1.5. COMPONENTES DEL HARDWARE

El hardware que es la parte dura del computador, lo tangible, lo que se puede ver y tocar, corresponde a todos los elementos físicos que conforman un sistema informático.

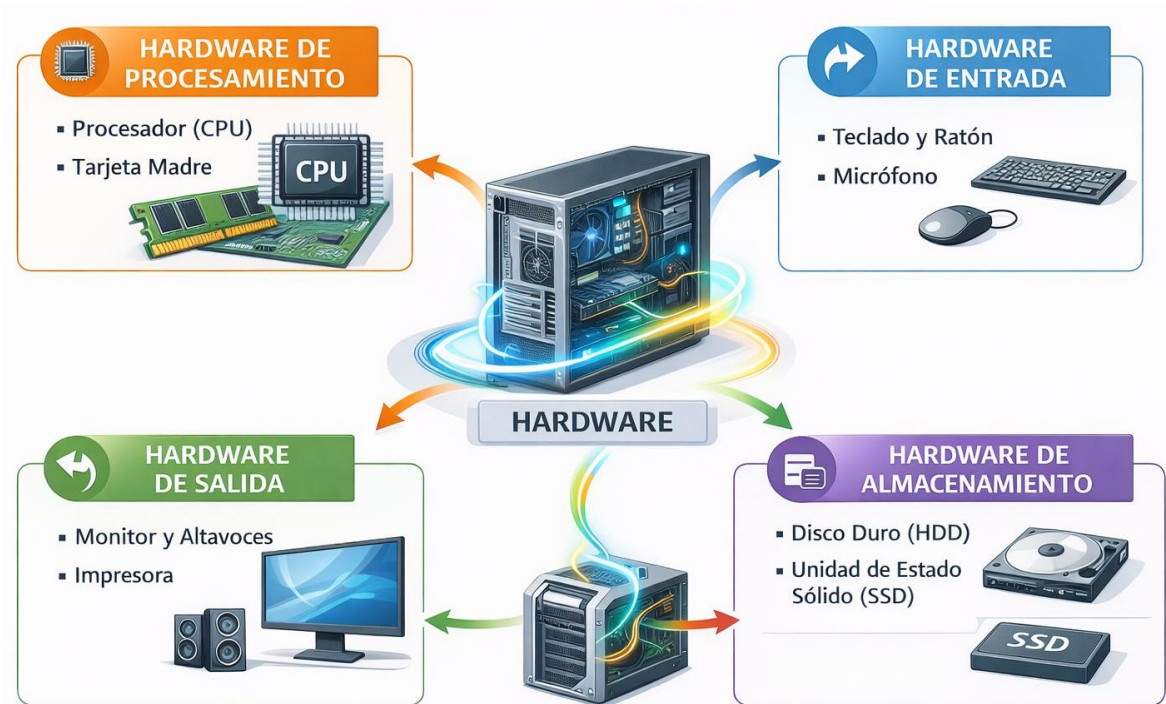
Figura 5
Componentes del Hardware



1.5.1. CLASIFICACIÓN DEL HARDWARE

El Hardware se Clasifica en:

Figura 6
Clasificación del Hardware



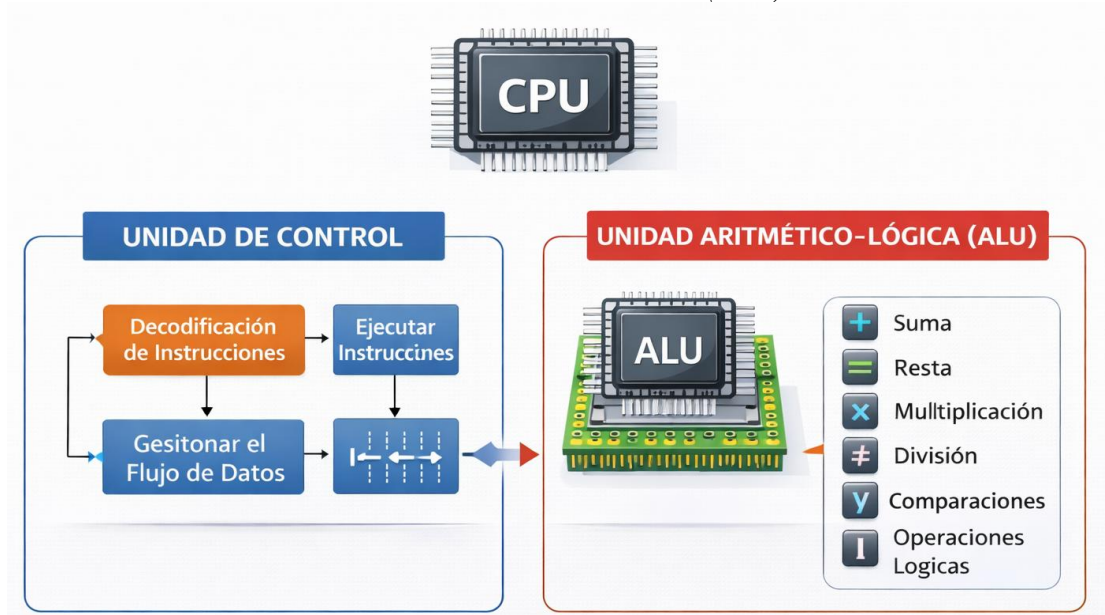
1).- UNIDAD CENTRAL DE PROCESAMIENTO (CPU)

Encargada de ejecutar todas las instrucciones, matemática, físicas y/o lógicas

Está Compuesta por:

- a). - Unidad de Control
- b). - Unidad Aritmético-Lógica (ALU)

Figura 7
Unidad Central de Procesamiento (CPU)



2).- DISPOSITIVOS DE ALMACENAMIENTO

- a). - Discos Duros (HDD)
- b). - Unidades de Estado Sólido (SSD)

Figura 8
Dispositivos de Almacenamiento



3).- DISPOSITIVOS DE ENTRADA

- a). - Teclado
- b). - Ratón
- c). - Escáner

Figura 9
Dispositivos de Entrada



4).- DISPOSITIVOS DE SALIDA

- a). - Monitor
- b). - Impresora

Figura 10
Dispositivos de Salida



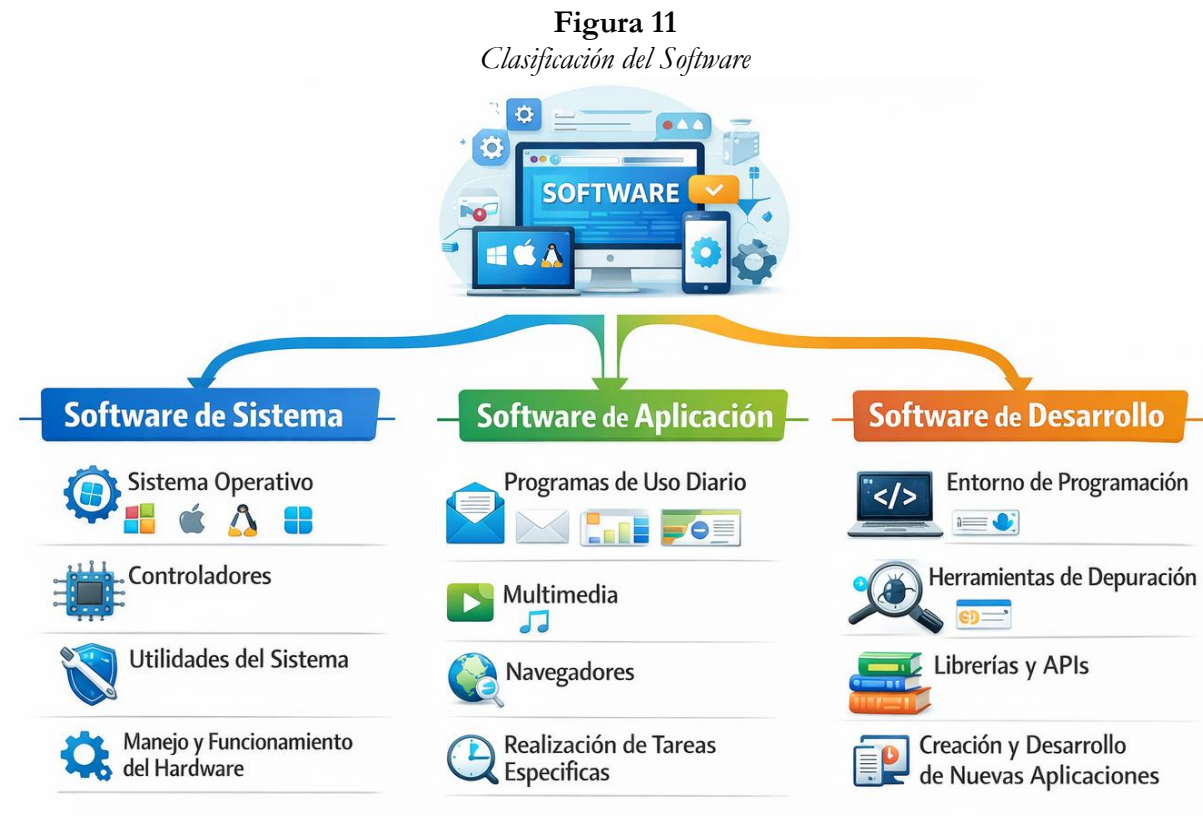
1.6. COMPONENTES DEL SOFTWARE

Conceptualización del Software

Software es el conjunto de programas que permiten al hardware ejecutar tareas específicas.

1.6.1. CLASIFICACIÓN DEL SOFTWARE

El Software se Clasifica en:



1).- SOFTWARE DE SISTEMA

- a). - Sistemas Operativos
- b). - Controladores (drivers)
- c). - Utilidades del Sistema
- d). - Manejo y funcionamiento del Hardware

2).- SOFTWARE DE APLICACIÓN

- a). - Aplicaciones de Oficina programas de uso diario
- b). - Navegadores Web
- c). - Software Especializado en tareas específicas
- d). – Multimedia

3).- SOFTWARE DE DESARROLLO

- a). – Entorno de Programación
- b). – Herramientas de Depuración
- c). – Librerías APIs
- d). – Creación y desarrollo de nuevas aplicaciones

1.7. RELACIÓN ENTRE EL HARDWARE Y EL SOFTWARE

El sistema operativo actúa como intermediario, gestionando recursos y facilitando la interacción entre usuario y máquina para una mayor comprensión

El hardware y el software funcionan de manera interdependiente:

- 1).- El hardware proporciona la infraestructura física
- 2).- El software define las instrucciones y procesos

Figura 12
Relación entre el Hardware y el Software



1.8. IMPORTANCIA EN EL ENTORNO INFORMÁTICO MODERNO

Hoy en día, los sistemas operativos y sus aplicaciones en el entorno tecnológico no solo implican conocer los componentes de un sistema informático, sino también entender su integración dentro del entorno informático moderno.

Este enfoque integral del entorno informático moderno permite tomar las decisiones más acertadas en la selección, implementación y gestión de los sistemas.

A continuación, detallamos cuatro aspectos importantes:

- 1)- Entornos Empresariales
- 2).- Infraestructuras Digitales
- 3).- Sistemas Distribuidos
- 4).- Computación en la Nube

Figura 13
Enfoque Integral del Entorno Informático Moderno



1.9. USO BÁSICO DE UN SISTEMA INFORMÁTICO

Un usuario enciende su computador para realizar una tarea académica:

- 1).- **HARDWARE:** Laptop (CPU, RAM, disco) y otros componentes
- 2).- **SOFTWARE:** Sistema operativo, procesador de texto, hoja de cálculos, presentaciones digitales
- 3).- **USUARIO:** Estudiante, Profesor, y demás profesionales

Análisis:

En este sentido se evidencia la interacción completa del sistema informático entre: El usuario con los componentes como es el Hardware y el Software, que dan como resultado final el Procesamiento de información.

1.10. DIFERENCIA ENTRE DATO E INFORMACIÓN

- 1).- Un **Dato** es: “85, 90, 78”
- 2).- La **Información** es: “Promedio del estudiante: 84.3”

Interpretación:

Dentro de la interpretación se puede decir que, el valor surge del procesamiento de la información, no de los datos aislados.

Teoría de caso

FALTA DE CONOCIMIENTO DEL HARDWARE

Una empresa adquiere software de alto rendimiento, pero:

- 1).- Los equipos informáticos tienen poca RAM
- 2).- Los procesadores son limitados

Resultado final de los equipos informáticos

- 1.- Lentitud
- 2.- Fallos constantes
- 3.- Baja productividad

En consecuencia: El desconocimiento de los componentes físicos de los equipos informáticos afecta directamente el rendimiento.

1.11. USO INCORRECTO DEL SISTEMA INFORMÁTICO

Un usuario elimina varios archivos del sistema sin conocer las funciones de los mismos:

Consecuencia:

- 1).- Fallos en el sistema operativo
- 2).- Necesidad de reinstalación

Lección:

Comprender la estructura de los equipos informáticos evita cometer errores críticos.

Teoría del Caso:

Una institución educativa de la localidad cuenta con:

- 1).- Computadores antiguos
- 2).- Software actualizado

Preguntas que se deben de realizar:

- 1).- ¿Existe compatibilidad?
- 2).- ¿Qué componente limita el sistema?
- 3).- ¿Qué solución propondría?

Análisis y respuesta técnica

Los conocimientos de los fundamentos de la informática, tanto del software como del hardware no es solamente es teórico, la parte primordial es que debe de ser práctico.

Cada acción que es realizada por un usuario, desde la parte más sencilla como es, abrir un archivo hasta lo más complejo como instalar un sistema operativo, está basado en procesos sistemáticos que deben de ser comprendidos, la comprensión de estos fundamentos teóricos prácticos permite:

- 1).- Evitar errores
- 2).- Optimizar recursos
- 3).-Tomar mejores decisiones

CAPÍTULO II

PARÁMETROS TÉCNICOS Y COMPATIBILIDAD

LA BASE DE LA TOMA DE DECISIONES TECNOLÓGICAS



2.1. DECISIONES TECNOLÓGICAS: HARDWARE & SOFTWARE

En cualquier entorno informático moderno, la correcta selección entre el hardware y el software no depende únicamente de preferencias o disponibilidad, sino de un análisis fuerte con rigurosos parámetros de seguimiento técnicos que determinen su funcionamiento correcto.

En la gestión de entornos informáticos modernos es fundamental tener ciertos criterios técnicos como es la compatibilidad de los equipos, la arquitectura, el correcto rendimiento y la capacidad, para que constituyan los pilares fundamentales sobre los cuales se construyen un sistemas informático eficientes y estables.

2.2. COMPATIBILIDAD: EL PRINCIPIO DE INTEGRACIÓN TECNOLÓGICA

La compatibilidad hace referencia a la capacidad de distintos componentes entre el hardware y el software que sirven para funcionar de manera conjunta sin conflictos.

2.2.1. TIPOS DE COMPATIBILIDAD

1).- **COMPATIBILIDAD DE HARDWARE:** Es la relación entre componentes físicos, ejemplo: placa madre y procesador

2).- **COMPATIBILIDAD DE SOFTWARE:** Es la relación entre aplicaciones y sistemas operativos

3).- **COMPATIBILIDAD ENTRE SISTEMAS:** Es la interoperabilidad entre diferentes plataformas informáticas

Figura 14
Tipos de Compatibilidad



Ejemplo:

Un usuario intenta instalar una versión reciente de un sistema operativo en un equipo antiguo:

- 1.- Procesador: 32 bits
- 2.- Sistema operativo requerido: 64 bits

Consecuencia encontrada: existe una incompatibilidad total, el sistema operativo no puede instalarse.

En el contexto internacional muchas organizaciones enfrentaron problemas graves al migrar a un sistema operativo de 64 bits, especialmente con aplicaciones heredadas (equipos obsoletos) que solo podían funcionar en entornos de 32 bits, obligándolo a implementar soluciones como máquinas virtuales o emuladores para que continúen operativos.

2.3. ARQUITECTURA DEL SISTEMA

La arquitectura del sistema define la estructura interna del sistema operativo y cómo se procesan los datos.

2.3.1. TIPOS DE ARQUITECTURA

a). - 32 bits

Manejo limitado de memoria (hasta 4 GB)

Compatible con software antiguo

b). - **64 bits**

Mayor capacidad de procesamiento

Soporte para grandes volúmenes de memoria

ARQUITECTURAS MODERNAS (ARM, x86, x64)

ARM: eficiencia energética (móviles, tablets)

x86/x64: alto rendimiento (PC, servidores)

REPRESENTACIÓN CONCEPTUAL

- 1).- **Arquitectura 32 bits** → Menor capacidad → Equipos antiguos
- 2).- **Arquitectura 64 bits** → Mayor capacidad → Equipos modernos
- 3).- **Arquitectura ARM** → Bajo consumo → Dispositivos móviles

Figura 15

Arquitectura del Sistema Representación Conceptual



Ejemplo:

- 1).- **Laptop Básica** → Procesador ARM → ideal para tareas ligeras
- 2).- **Servidor Empresarial** → Arquitectura x64 → manejo de múltiples procesos simultáneos

2.4. RENDIMIENTO DEL SISTEMA

El rendimiento del sistema, mide la capacidad para ejecutar tareas de manera eficiente.

Factores que influyen

- 1).- Velocidad del procesador (GHz)
- 2).- Cantidad de memoria RAM
- 3).- Tipo de almacenamiento (HDD vs SSD)
- 4).- Optimización del sistema operativo

Estudio de caso

Una empresa mejora el rendimiento de sus equipos sin cambiar el procesador:

- 1).- Antes: HDD + 4 GB RAM
- 2).- Después: SSD + 8 GB RAM

Resultado: el incremento significativo en velocidad de arranque y ejecución de aplicaciones del equipo informático

2.5. COMPARACIÓN DEL RENDIMIENTO DEL SISTEMA

Tabla 1.
Componentes

COMPONENTE	BAJO RENDIMIENTO	ALTO RENDIMIENTO
Almacenamiento	HDD	SSD
RAM	4 GB	16 GB
CPU	2 núcleos	8 núcleos

2.6. CAPACIDAD DEL SISTEMA

Es la capacidad que tiene el sistema en recursos disponibles para poder operar correctamente.

Ejemplo:

Un diseñador gráfico requiere:

- a). - 16–32 GB de RAM
- b). - SSD de alta velocidad
- c). - GPU dedicada

Mientras que un usuario de oficina básica necesita:

- a). - 4–8 GB de RAM
- b). - Almacenamiento estándar

2.6.1. TIPOS DE CAPACIDADES DEL SISTEMA

- 1).- Capacidad de Almacenamiento (GB, TB)
- 2).- Capacidad de Memoria (RAM)
- 3).- Capacidad de Procesamiento (núcleos, hilos)

Figura 16
Tipos De Capacidades Del Sistema

1. Capacidad de Almacenamiento (GB, TB)



Gigabytes / Terabytes

2. Capacidad de Memoria (RAM)



Memoria de Acceso Aleatorio

3. Capacidad de Procesamiento (Núcleos, Hilos)



Núcleos y Procesos

2.7. EVALUACIÓN DE RESTRICCIONES TÉCNICAS

No todos los entornos permiten implementar cualquier solución tecnológica.

Factores Limitantes:

- 1).- Presupuesto
- 2).- Infraestructura existente
- 3).- Licencias de software
- 4).- Capacitación del personal

Teórica de caso

Una institución educativa con equipos antiguos decide implementar software moderno:

Problema:

Equipos informáticos antiguos, no soportan nuevas versiones

Solución:

Instalar sistemas operativos ligeros (Linux)

Implementación de software de bajo consumo para que los equipos funcionen correctamente

2.8. INTEGRACIÓN DE PARÁMETROS PARA LA TOMA DE DECISIONES

La selección adecuada de un sistema requiere integrar todos los parámetros analizados:

- 1).- **Compatibilidad:** para evitar fallos
- 2).- **Arquitectura:** para definir la capacidad técnica
- 3).- **Rendimiento:** para garantizar la eficiencia
- 4).- **Capacidad:** para asegurar el manejo de cantidad crecientes

2.8.1. IMPORTANCIA ESTRATÉGICA

La correcta evaluación de parámetros técnicos permite:

- 1.- Reducir costos a largo plazo
- 2.- Evitar fallos de implementación
- 3.- Garantizar continuidad operativa
- 4.- Mejorar la experiencia del usuario

Dentro de los entornos empresariales, estas decisiones impactan directamente en la productividad y la competitividad organizacional.

Figura 17
Integración de Parámetros para la toma de decisiones

Integración de Parámetros para la Toma de Decisiones

Factores importantes al seleccionar componentes informáticos



Figura 18
Esquema de Decisión



2.9. ANÁLISIS DE REQUERIMIENTOS DEL ENTORNO Y DEL USUARIO

Cuando vas a implementar un sistema operativo y sus aplicaciones, no depende únicamente de la tecnología disponible, sino de la capacidad de interpretar correctamente las necesidades del usuario y las condiciones del entorno tecnológico para que el equipo o los equipos informáticos funcionen adecuadamente sin complicaciones, así no tendrás inconveniente en el futuro con tus equipos, dentro del análisis de requerimientos se constituye en una fase importante que permite alinear los recursos tecnológicos con objetivos funcionales, operativos y organizacionales para que los sistemas instalados funcionen correctamente y no produzcan falla posteriores

Figura 19
Análisis de Requerimientos del Entorno y del Usuario



2.10. IDENTIFICACIÓN DE NECESIDADES DEL USUARIO

Todo sistema informático debe diseñarse o desarrollarse en función del usuario final.

2.10.1. TIPOS DE USUARIOS

Cuando hablamos de tipos de usuarios, normalmente se piensa en una clasificación simple. Pero en realidad, estamos hablando de algo más interesante como, por ejemplo:

- 1).- Diferentes formas de relacionarse con la tecnología.
- 2).- No todos usan un sistema operativo de la misma manera.
- 3). - Algunos solo interactúan con la superficie, mientras otros exploran lo que hay debajo.

Figura 20
Tipos de Usuarios



USUARIO BÁSICO: Es la persona que utiliza herramientas sencillas de uso general como, por ejemplo: ofimática, navegación, multimedia

El usuario básico no necesita entender cómo funciona el sistema, solo necesita que funcione.

Su relación con la computadora es práctica, directa y orientada a resultados.

1).- **No le interesa:**

- a). - Cómo se administra la memoria
- b). - Cómo se ejecutan los procesos

2).- **Le interesa:**

- a). - Escribir
- b). - Buscar información
- c). - Ver contenido

Ejemplo:

UN USUARIO BÁSICO USA:

- 1).- Procesadores de texto
- 2).- Navegadores web
- 3).- Reproductores multimedia

Procedimiento que realiza

Hace clic, escribe, consume, edita contenido, espera que todo responda correctamente y obtiene su resultado

Teórica de caso

Una persona que usa la computadora para estudiar, ver videos y enviar correos, esta interactuando solo con la capa visible del sistema operativo

Aquí ocurre algo interesante:

Aunque no lo sepa, el sistema operativo está tomando cientos de decisiones por él. El usuario básico confía completamente en el sistema de todo lo que realiza.

USUARIO TÉCNICO: Es la persona que desarrolla programas, da soporte a los equipos informáticas, administra sistemas y realiza demás actividades en la parte técnica del equipo, aquí te detallo cierto criterio que el usa:

- 1).- El usuario técnico ya no se conforma con usar el sistema.
- 2).- Quiere entenderlo, modificarlo y, en algunos casos, repararlo.
- 3).- Su relación con la computadora cambia:
 - a). - Ya no solo ejecuta acciones
 - b). - Empieza a cuestionarlas

Ejemplo

Un usuario técnico puede:

- 1).- Instalar sistemas operativos
- 2).- Configurar redes
- 3).- Diagnosticar fallos
- 4).- Escribir programas

Teoría de Caso

Un técnico que administra equipos con Windows y Linux puede:

- 1).- Detectar por qué un sistema está lento
- 2).- Optimizar recursos
- 3).- Resolver errores que un usuario básico no entendería

USUARIO ESPECIALIZADO: Es la persona que tiene especificada sus funciones como, por ejemplo: un Diseñador gráfico, profesional que maneja programas de ingeniería y análisis de datos

Ejemplo:

Un estudiante universitario requiere:

- 1).- Navegador web
- 2).- Suite ofimática
- 3).- Plataforma de videoconferencia

El usuario especializado no se enfoca en el sistema en sí, usa la computadora como medio para algo más complejo.

Su objetivo no es el sistema operativo, sino:

- 1).- Crear
- 2).- Analizar
- 3).- Diseñar
- 4).- Resolver problemas específicos

Ejemplo:

Este tipo de usuario trabaja con herramientas avanzadas como:

- 1).- Software de diseño gráfico
- 2).- Programas de ingeniería
- 3).- Sistemas de análisis de datos

Teoría de Caso

Un diseñador gráfico, no necesita saber cómo funciona el KERNEL, pero sí necesita que el sistema sea estable y rápido

Un ingeniero, ejecuta simulaciones complejas, requiere precisión y rendimiento; En ambos casos, el sistema operativo se convierte en una plataforma de alto rendimiento

Figura 21
Solución Primaria

SOLUCIÓN PRIMARIA

Pasos básicos para resolver problemas comunes en tu computadora



DEFINICIÓN

El **Sistema Operativo** es el software que permite que el hardware y los programas funcionen juntos.

Cuando algo falla en una computadora, la reacción común es:

⚠️ ¡Se dañó!

¡Necesito un técnico!

Pero en realidad, muchos **problemas son pequeños desajustes** en el **sistema**:



Procesos saturados



Programas bloqueados



Recursos mal distribuidos

La solución primaria busca restablecer el equilibrio, no reparar desde cero.



Reiniciar



Cerrar procesos



Actualizar



Liberar recursos

La solución **primaria** busca restablecer el **equilibrio**, no reparar desde cero.

2.11. SOLUCIÓN PRIMARIA NO AVANZADA:

Es el conjunto de acciones básicas que un usuario puede aplicar para resolver problemas sin conocimientos técnicos profundos.

No todo problema necesita una solución compleja, muchas veces, lo simple es suficiente.

Cuando algo falla en una computadora, la reacción común, es decir se dañó necesito un técnico, pero en realidad, muchos problemas son pequeños, como, por ejemplo:

- 1).- Procesos saturados
- 2).- Programas bloqueados
- 3).- Recursos mal distribuidos

La solución primaria busca restablecer el equilibrio, no reparar desde cero.

Ejemplos

REINICIAR EL SISTEMA

Reiniciar no es apagar y encender, es obligar al sistema operativo a:

- 1).- Cerrar procesos
- 2).- Liberar memoria
- 3).- Empezar desde un estado limpio

Teoría de Caso

Cuando una computadora se congela lo que sucede es que los programas dejan de responder y todo se vuelve lento

Solución: Reiniciar el sistema va a ser que se elimine conflictos internos acumulados en el equipo informático

CERRAR PROGRAMAS INNECESARIOS

Cada programa abierto consume demasiado Memoria (RAM) conjuntamente con el Procesador (CPU) aunque no lo estés usando activamente

Ejemplo

Tienes abiertos:

- 1).- Navegador con muchas pestañas
- 2).- Música
- 3).- Aplicaciones en segundo plano

Que sucede en ese momento que el sistema empieza a saturarse y la maquina se vuelve lenta

Teoría de Caso

Cerrar aplicaciones mejora:

- 1).- Velocidad
- 2).- Estabilidad
- 3).- Tiempo de respuesta

VERIFICAR CONEXIONES BÁSICAS

No todos los errores son del sistema, algunos son externos como, por ejemplo:

- 1).- Internet desconectado
- 2).- Cables sueltos
- 3).- Dispositivos mal conectados

Teoría de Caso

Un usuario cree que no funciona el internet, pero el problema era el Wi-Fi desactivado y/o desconectado, antes de llamar al técnico es conveniente revisar todas las instalaciones y conexiones de los equipos informativos

ACTUALIZAR O REINICIAR APLICACIONES

Un programa puede fallar sin que el sistema esté dañado, que se debe de hacer:

- 1).- Reiniciar la aplicación
- 2).- Eliminar archivos temporales

Solución: el sistema restablece su funcionamiento

Teoría de Caso

Una aplicación se queda congelada, no responde, no carga, se realizar lo siguiente:

Cerrarla y abrirla nuevamente, esto suele resolver el problema

LIBERAR ESPACIO EN EL SISTEMA

El almacenamiento está lleno y afecta el rendimiento, el sistema necesita espacio para:

- 1).- Archivos temporales
- 2).- Procesos internos

Teoría de Caso

Cuando el disco duro está casi lleno, pasa lo siguiente:

- 1).- El sistema se vuelve lento
- 2).- Algunas funciones fallan

Solución: Eliminar archivos innecesarios mejora el rendimiento

2.12. ANÁLISIS DEL ENTORNO ORGANIZACIONAL

El entorno organizacional condiciona fuertemente las decisiones tecnológicas.

Factores que influyen

- a). - Infraestructura existente
- b). - Presupuesto disponible
- c). - Políticas institucionales
- d). - Nivel de capacitación del personal
- e). - Escalabilidad requerida

Figura 22
Análisis del Entorno Organizacional



Cuando se habla de implementar sistemas, muchas personas piensan primero en:

- 1).-Software moderno
- 2).- Equipos potentes
- 3).-Soluciones avanzadas

Pero hay una verdad que pocas veces se dice que la tecnología no falla por sí sola, falla cuando no encaja en su entorno, es decir que no saben o no pueden manipular equipos con tecnología de punta.

Ejemplo:

Una empresa decide instalar un sistema muy avanzado de gestión.

Pero:

- 1).- Los empleados no saben usarlo
- 2).- Los equipos son antiguos
- 3).- No hay soporte técnico

Resultado: el sistema falla, no por ser malo, sino por no adaptarse al entorno informático moderno

Teoría de Caso

Muchas instituciones compran software costoso pensando que mejorará todo, pero ocurre lo siguiente:

- 1).- Bajo conocimientos de los empleados en manejar dicho sistema
- 2).- Producen errores constantes
- 3).- Resistencia del personal a usar esas nuevas herramientas

Solución: El problema no era la herramienta, el problema es el entorno que no está preparado

2.13. ENTORNO ORGANIZACIONAL FACTORES QUE INFLUYEN

1).- INFRAESTRUCTURA EXISTENTE

La infraestructura es el punto de partida real, no el ideal, no puedes construir algo nuevo ignorando lo que ya existe, esto incluye:

- a). - Equipos (computadoras, servidores, etc.)
- b). - Redes
- c). - Sistemas ya instalados

Ejemplo

Quieres implementar un sistema moderno, pero:

- a). - Las computadoras tienen poca RAM
- b). - La red es lenta
- c). - El sistema funcionará mal, aunque sea excelente

Teoría de Caso

Empresas que instalan software pesado en equipos antiguos:

- a). - Lentitud constante
- b). - Bloqueos
- c). - Frustración del usuario

La infraestructura limita el potencial

2).- PRESUPUESTO DISPONIBLE

El presupuesto no solo define qué comprar, también define qué es sostenible en el tiempo

Incluye:

- a). - Compra inicial
- b). - Mantenimiento
- c). - Actualizaciones
- d). - Soporte

Ejemplo

Una organización compra software caro, pero no puede pagar:

- a). - Licencias futuras
- c). - Soporte técnico

El sistema queda abandonado

Teoría de Caso

Implementaciones tecnológicas que funcionan al inicio, pero fallan meses después por falta de recursos

No era una solución real, era una solución momentánea

3).- **POLÍTICAS INSTITUCIONALES**

Las políticas son las reglas invisibles de la organización, que define lo que se puede y no se puede hacer, estas Incluyen:

- a). - Seguridad
- b). - Acceso a información
- c). - Uso de software

Ejemplo

Quieres usar una herramienta en la nube, pero, la organización prohíbe almacenamiento externo, no puedes implementarla, aunque sea útil

4).- **NIVEL DE CAPACITACIÓN DEL PERSONAL**

La tecnología solo es tan útil como las personas que la usan.

Un sistema complejo en manos no preparadas se convierte en un problema

Ejemplo

Se implementa un sistema nuevo, pero nadie sabe usarlo correctamente y los empleados cometen muchos errores constantes; El sistema parece malo pero el problema es lo empleados

Teoría de Caso

Se encuentran organizaciones donde el personal evita usar nuevas herramientas, ellos prefieren métodos antiguos, no por resistencia, sino por falta de formación

5).- **ESCALABILIDAD REQUERIDA**

Escalabilidad significa pensar en el futuro; No solo en lo que necesitas hoy, sino en lo que necesitarás mañana

Ejemplo

Una empresa pequeña implementa un sistema básico, pero crece rápidamente, que ocurre que con el tiempo el sistema ya no soporta la demanda

Teoría de Caso

Negocios que crecen en usuarios o datos, pero su sistema no crece con ellos, que ocurre que terminan migrando o reemplazando todo

2.14. CRITERIOS PARA SELECCIONAR COMBINACIONES ÓPTIMAS DE HARDWARE Y SOFTWARE

Figura 23
Combinaciones Óptimas de Hardware y Software



Para tener una adecuada selección de combinaciones óptimas, esto implica evaluar múltiples variables simultáneamente como, por ejemplo

CRITERIOS OPERATIVOS

- 1).- Facilidad de uso
- 2).- Soporte técnico
- 3).- Disponibilidad de actualizaciones

ELEGIR NO ES COMPRAR, ES EQUILIBRAR

Cuando una persona u organización selecciona hardware y software, suele pensar:

- 1).- ¿Cuál es el mejor?
- 2).- ¿Cuál es más rápido?
- 3).- ¿Cuál es más moderno?

Pero hay una verdad más importante:

- 1).- No existe la mejor combinación en general.
- 2).- Existe la mejor combinación para una situación específica.

Elegir bien no es acumular potencia, es lograr equilibrio entre lo que necesitas, lo que tienes y lo que puedes mantener.

PENSAR EN COMBINACIONES, NO EN PIEZAS

Un error muy común es analizar por separado:

- a). - El hardware
- b). - El software

Cuando en realidad, el equipo funciona como un sistema único

Un software exigente en un hardware limitado genera:

- a). - Lentitud
- b). - Fallos
- c). - Frustración

Un hardware potente con software mal optimizado genera:

- a). - Desperdicio de recursos
- b). - Costos innecesarios

La eficiencia no está en cada componente, está en cómo trabajan juntos.

CRITERIOS TÉCNICOS

- 1).- Compatibilidad
- 2).- Rendimiento requerido
- 3).- Capacidad de expansión
- 4).- Seguridad

COMPATIBILIDAD

Compatibilidad no es solo que funcione, es que hardware y software puedan comunicarse sin crear ningún conflicto.

Incluye:

- a). - Sistema operativo compatible
- b). - Controladores (drivers) adecuados
- c). - Arquitectura (32/64 bits, etc.)

Ejemplo

Instalas un programa moderno en un equipo antiguo:

- a). - Puede no abrir
- b). - Puede fallar o funcionar incorrectamente

Teoría de Caso

Un dispositivo nuevo conectado a un sistema antiguo, no funciona porque no existen drivers compatibles, cual es el problema no es el equipo, si no, es la falta de diálogo entre componentes

Rendimiento requerido

El rendimiento no se mide en qué tan rápido es, se mide en qué tan bien responde a lo que necesitas hacer.

Ejemplo

- a). - Para navegar es un bajo requerimiento
- b). - Para diseño o videojuegos es un alto requerimiento

Teoría de Caso

Tienes una computadora con pocos recursos ejecutando software pesado, que ocurre:

- a). - Se vuelve lenta
- b). - Se bloquea
- c). - Reduce productividad

El sistema no falla, lo que pasa es que está sobre exigido

CAPACIDAD DE EXPANSIÓN

La expansión es la capacidad de crecer sin reemplazar todo, es pensar en el futuro desde el presente

Ejemplo

Un equipo que permite:

- a). - Aumentar RAM
- b). - Cambiar disco
- c). - Mejorar componentes

Este equipo tendrá más años de duración

Teoría de Caso

Empresas que compran equipos cerrados:

- a). - No pueden mejorarlos
- b). - Deben reemplazarlos completamente

Esto aumenta costos a largo plazo

SEGURIDAD

La seguridad no es un complemento, es una condición básica de funcionamiento de equipo

Incluye:

- a). - Protección contra accesos no autorizados
- b). - Integridad de datos
- c). - Control de permisos

Ejemplo

Un sistema sin actualizaciones:

- a). - Vulnerable a ataques
- b). - Está expuesto a fallos

Teoría de Caso

Organizaciones que usan software desactualizado, son más propensas a ataques o pérdida de información

Tabla 2.
Capacidad de expansión

ENTORNO	HARDWARE RECOMENDADO	SOFTWARE RECOMENDADO
Oficina	CPU media, 8GB RAM	Ofimática, Programas básicos
Diseño	CPU alta, GPU dedicada	Software de diseño
Servidor	CPU multi núcleo	Sistema operativo servidor

Figura 23
Modelo de Evaluación Simplificado



2.15. ARQUITECTURA Y FUNCIONAMIENTO GENERAL DE LOS SISTEMAS OPERATIVOS

Figura 24
Funcionamiento general de los Sistemas Operativos



Cuando escuchas hablar de arquitectura de un sistema operativo, suena a algo frío, técnico, casi mecánico, pero en realidad, estás frente a algo mucho más interesante:

- 1).- Un sistema de decisiones que nunca duerme.
- 2).- Tu computadora no funciona sola.

Está siendo constantemente administrada por un sistema que:

- 1).- Observa lo que haces
- 2).- Interpreta tus acciones
- 3).- Decide si ejecutarlas
- 4).- Controla cómo se ejecutan

Es como un asistente invisible que no te pregunta nada, pero evalúa todo.

Ejemplo

Imagina tu computadora como una oficina

- 1).- El usuario → el cliente
- 2).- El sistema operativo → el gerente

- 3).- El hardware → los trabajadores
- 4).- Los programas → las tareas

Teoría de Caso

En computadoras modernas, si la arquitectura está bien diseñada:

- 1).- Puedes abrir muchos programas sin problemas
- 2).- El sistema no se bloquea fácilmente

Pero si está mal optimizada:

- 1).- Todo se vuelve lento
- 2).- Aparecen errores
- 3).- El sistema puede colapsar

2.15.1. CLASIFICACIÓN FUNCIONAL DE LOS SISTEMAS OPERATIVOS

Figura 25
Clasificación Funcional de los Sistemas Operativos



1).- MONOTAREA

Ejecuta una sola tarea a la vez.

Un sistema monotarea es como una persona que solo puede concentrarse en una cosa.

Si empieza algo, no puede hacer nada más hasta terminarlo.

Ejemplo

Quieres escribir un documento y al mismo tiempo escuchar música, en un sistema monotarea, no puedes tener que terminar uno primero

Teoría de Caso

El sistema **MS-DOS** funcionaba así:

1).- No existía multitarea

2).- Todo era secuencial

Era simple, pero poco flexible

2).- MULTITAREA

Ejecuta múltiples procesos simultáneamente

Un sistema multitarea es como una persona que intenta hacer varias cosas a la vez, pero en realidad, no hace todo al mismo tiempo, cambia rápidamente entre tareas

Ejemplo

Estás:

1).- En una videollamada

2).- Escribiendo

3).- Descargando archivos

El sistema hace micro operaciones en ese instante

Teoría de Caso

Sistemas como:

1).- Windows

2).- Linux

Funcionan así.

Pero cuando hay demasiadas tareas:

1).- Todo se vuelve lento

2).- Las prioridades cambian

3).- Algunas tareas “esperan”

3).- MONOUSUARIO

Un solo usuario accede al sistema

Es como una casa donde vive una sola persona, todo está organizado para un único dueño

Ejemplo

- 1).- Solo hay un perfil activo
- 2).- No hay separación real entre usuarios

Teoría de Caso

Muchas computadoras personales funcionan así:

Todo está diseñado para un solo usuario principal

4.- MULTIUSUARIO

Varios usuarios operan simultáneamente

Es como un edificio donde viven muchas personas:

- 1).- Cada uno tiene su espacio
- 2).- Pero comparten recursos

Ejemplo

- 1).- Varios usuarios conectados a un servidor
- 2).- Cada uno con sus archivos y permisos

Teoría de Caso

Sistemas como:

- 1).- Unix
- 2).- Linux

Permiten:

- 1).- Múltiples usuarios simultáneos
- 2).- Trabajo remoto
- 3).- Separación segura

Son la base de internet y servidores modernos

2.15.2. CLASIFICACIÓN GENERAL DE LOS SISTEMAS OPERATIVOS

Figura 26
Clasificación General de los Sistemas Operativos



1).- SISTEMAS OPERATIVOS PROPIETARIOS

Un sistema operativo propietario es aquel cuyo código fuente no es accesible al público y cuyo uso, modificación y distribución están restringidos legalmente por una empresa o entidad.

Esto implica que el usuario no posee el control total del sistema, sino únicamente una licencia de uso bajo ciertas condiciones.

LA EMPRESA MANTIENE CONTROL ABSOLUTO SOBRE:

- 1).- El diseño del sistema
- 2).- Las actualizaciones
- 3).- La compatibilidad de software y hardware
- 4).- Esto permite una integración más estable, pero limita la libertad del usuario.

Ejemplo: Windows

2).- SISTEMAS OPERATIVOS LIBRES

Un sistema operativo libre es aquel que otorga al usuario acceso a su código fuente y la capacidad legal de modificarlo, estudiarlo y redistribuirlo.

A diferencia de los sistemas cerrados, no existe una única entidad que controle completamente su evolución. En su lugar, el sistema se desarrolla de manera abierta, permitiendo que múltiples personas o comunidades contribuyan a su mejora.

Ejemplos

- 1).- Linux
- 2).- Ubuntu

3.- SISTEMAS EMBEBIDOS

Un sistema operativo embebido es aquel diseñado para funcionar dentro de un dispositivo específico, cumpliendo una función concreta y generalmente limitada.

A diferencia de los sistemas operativos de propósito general, no están pensados para ejecutar múltiples tareas complejas para un usuario, sino para controlar directamente un hardware específico de manera eficiente y confiable.

Ejemplos

- 1).- **FreeRTOS**: utilizado en microcontroladores y dispositivos electrónicos pequeños
- 2).- **VxWorks**: usado en sistemas críticos como aeronáutica y equipos industriales
- 3).- **QNX**: presente en automóviles y sistemas industriales
- 4).- **Embedded Linux**: versiones adaptadas de Linux para dispositivos específicos

Estos sistemas operativos están presentes en dispositivos como:

- 1).- Electrodomésticos
- 2).- Automóviles
- 3).- Relojes inteligentes
- 4).- Equipos médicos
- 5).- Sistemas industriales

Aunque muchas veces pasan desapercibidos, son fundamentales en el funcionamiento del mundo tecnológico actual.

4.- SISTEMAS MÓVILES

Un sistema operativo móvil es aquel diseñado específicamente para dispositivos portátiles como

teléfonos inteligentes y tabletas, donde las condiciones de uso, interacción y recursos son diferentes a las de una computadora tradicional.

Estos sistemas no solo gestionan el hardware, sino que están optimizados para:

- 1).- Interacción táctil
- 2).- Conectividad constante (redes móviles, Wi-Fi, Bluetooth)
- 3).- Consumo eficiente de energía
- 4).- Ejecución de aplicaciones en entornos controlados

Ejemplos

Estos sistemas dominan el entorno móvil actual y representan dos enfoques distintos:

- 1).- Android:** mayor flexibilidad y adaptación a distintos fabricantes
- 2).- iOS:** mayor control e integración dentro de un ecosistema cerrado

Ambos están diseñados para optimizar la experiencia del usuario en dispositivos portátiles, manteniendo un equilibrio entre rendimiento, seguridad y usabilidad.

5.- SISTEMAS VIRTUALIZADOS

Un sistema operativo virtualizado es aquel que no se ejecuta directamente sobre el hardware físico, sino sobre una capa intermedia llamada hipervisor, que permite crear y gestionar múltiples entornos virtuales dentro de una misma máquina.

En este modelo, un solo equipo físico puede alojar varios sistemas operativos independientes, cada uno funcionando como si tuviera su propio hardware.

El objetivo principal no es reemplazar al sistema base, sino multiplicar su capacidad, permitiendo aislar, probar y ejecutar diferentes sistemas en paralelo, así lo detallamos:

- 1).- El hardware real se abstrae
- 2).- Los recursos (CPU, memoria, almacenamiento) se distribuyen
- 3).- Cada sistema virtual opera de forma independiente

Esto introduce una separación clara entre:

- 1).- Máquina física (real)
- 2).- Máquina virtual (simulada)

Ejemplo

Una empresa necesita probar software en distintos entornos:

Solución: Usar un sistema operativo Virtualizado

Beneficio: mayor reducción de costos en hardware y mayor flexibilidad

2.16. LICENCIAMIENTO Y MODELOS DE DISTRIBUCIÓN DE SOFTWARE

El uso del software no solo implica instalar programas y utilizarlos. También involucra responsabilidades legales, decisiones técnicas y estrategias que afectan tanto a personas como a empresas.

Ejemplo

Usar software correctamente significa cumplir normas, elegir bien las herramientas y aprovecharlas de forma inteligente.

Figura 27

Licenciamiento y modelos de distribución de software



2.16.1. ASPECTOS LEGALES, TÉCNICOS Y ESTRATÉGICOS EN EL USO DEL SOFTWARE

1).- ASPECTOS LEGALES

Son las normas y leyes que regulan el uso del software.

Esto define:

- Qué puedes usar
- Cómo puedes usarlo
- Qué está prohibido

Elementos

- a). - Licencias de software
- b). - Uso ilegal (piratería)
- c). - Protección de datos
- d). - Derechos de autor

Ejemplo

Instalar un programa sin pagar su licencia.

Teoría de Caso

Una empresa usa copias ilegales de Windows 11 o programas de oficina.

Consecuencia:

- a). - Multas legales
- b). - Problemas judiciales
- c). - Pérdida de reputación

2).- ASPECTOS TÉCNICOS

Son las características y requisitos que permiten que el software funcione correctamente.

Elementos

- a). - Compatibilidad (hardware y sistema operativo)
- b). - Rendimiento
- c). - Actualizaciones
- d). - Seguridad

Ejemplo

Un programa puede no funcionar si tu computadora no tiene suficiente memoria RAM.

Teoría de Caso

Una empresa instala un sistema en computadoras antiguas.

Resultado:

- a). - El sistema se vuelve lento
- b). - Los empleados pierden tiempo
- c). - Baja la productividad

3).- ASPECTOS ESTRATÉGICOS

Son las decisiones que se toman para usar el software de manera eficiente y alineada con objetivos.

Elementos

- a). - Elección del software adecuado
- b). - Costos vs beneficios
- c). - Escalabilidad (crecimiento)
- d). - Adaptación a la empresa

Ejemplo

Elegir entre software libre o propietario según el presupuesto y necesidades.

Teoría de Caso

Una empresa decide usar:

- a). - Software libre como Ubuntu para ahorrar costos
- b). - Software propietario por compatibilidad

Esto no es solo técnico, es una decisión estratégica.

LICENCIA DE SOFTWARE

Una licencia de software es un acuerdo legal entre el desarrollador y el usuario que establece los términos de uso del programa.

Define los siguientes aspectos:

- a). - Uso permitido
- b). - Restricciones
- c). - Derechos de modificación
- d). - Distribución

Ejemplo

Una licencia de software es un permiso legal que indica cómo puedes usar un programa informático, es decir que es un acuerdo entre, el creador del software y el usuario

Es como un contrato que te dice qué puedes hacer y qué no puedes hacer con un programa.

Ejemplo

Instalas un programa en tu computadora y aceptas los términos sin leerlos.

Ahí estás aceptando una licencia que puede decir:

- a). - Solo puedes usarlo en un equipo
- b). - No puedes copiarlo
- c). - No puedes modificarlo

Teoría de Casos

1).- USO PERSONAL DE SOFTWARE PROPIETARIO

Cuando instalas Windows 11, aceptas una licencia.

Esto significa:

- a). - Puedes usar el sistema
- b). - Pero no puedes modificar su código
- c). - Ni distribuirlo libremente

2).- USO EMPRESARIAL

Una empresa compra licencias para programas.

Caso real:

- a). - Compra 20 licencias de un software
- b). - Solo 20 computadoras pueden usarlo legalmente

Si lo instalan en más equipos:

Están violando la licencia

2.16.2. CLASIFICACIÓN DE LOS TIPOS DE LICENCIAS DE SOFTWARE

Figura 28
Clasificación de los Tipos de Licencias de Software



1).- FREWARE

El freeware es un tipo de software que se distribuye de manera gratuita, permitiendo al usuario utilizarlo sin costo, pero sin acceso a su código fuente ni libertad para modificarlo.

A diferencia del software libre, el freeware no otorga control sobre su estructura interna; únicamente concede el derecho de uso bajo condiciones establecidas por el desarrollador.

Esto significa que, aunque el usuario puede instalar y utilizar el programa libremente, no puede:

- 1).- Estudiar cómo funciona internamente
- 2).- Modificar su código
- 3).- Redistribuir versiones alteradas

En términos conceptuales, el freeware se sitúa en un punto intermedio entre el software propietario de pago y el software libre.

Ejemplos

- 1).- Google Chrome
- 2).- Adobe Acrobat Reader
- 3).- Skype

Estos programas pueden utilizarse sin costo, pero siguen siendo propiedad de sus respectivas empresas y no permiten modificaciones por parte del usuario.

2).- SHAREWARE

El shareware es un modelo de distribución de software en el cual un programa se ofrece de manera gratuita solo de forma temporal o con funcionalidades limitadas, con el objetivo de que el usuario lo pruebe antes de adquirir la versión completa.

A diferencia del freeware, el shareware no está pensado como uso gratuito permanente, sino como una estrategia de evaluación previa a la compra.

Este modelo permite al usuario experimentar el programa en condiciones reales, reduciendo la incertidumbre antes de invertir en él.

Ejemplos

- 1).- WinRAR
- 2).- WinZip
- 3).- Internet Download Manager

Estos programas permiten su uso inicial sin costo, pero requieren una licencia para acceder a todas sus funciones o eliminar restricciones.

3).- SOFTWARE LIBRE

El software libre es aquel que garantiza al usuario la libertad de usar, estudiar, modificar y distribuir el programa sin restricciones que limiten su control sobre el mismo.

Este tipo de software se basa en un principio fundamental, que el conocimiento debe ser accesible y modificable, permitiendo que cualquier persona pueda comprender cómo funciona el programa y adaptarlo a sus necesidades.

Para que un software sea considerado libre, debe cumplir con libertades esenciales como:

- 1).- Ejecutarlo para cualquier propósito
- 2).- Estudiar su funcionamiento (acceso al código fuente)
- 3).- Modificarlo según necesidades
- 4).- Redistribuir copias, con o sin cambios

Estas libertades están protegidas por licencias específicas que aseguran que el software permanezca abierto y accesible.

Ejemplos

- 1).- GNU/Linux
- 2).- LibreOffice
- 3).- GIMP

Estos programas representan el modelo de software libre, donde el acceso al código y la posibilidad de modificación forman parte esencial de su naturaleza.

4).- SOFTWARE PROPIETARIO

El software propietario es aquel cuyo código fuente está restringido y cuyo uso, modificación y distribución están controlados por una persona, empresa u organización.

En este modelo, el usuario no adquiere el software como propiedad, sino una licencia de uso bajo condiciones específicas definidas por el propietario.

Esto implica que el desarrollo, mantenimiento y evolución del software dependen exclusivamente de su creador, limitando la intervención externa.

Ejemplos

- 1).- Microsoft Windows
- 2).- Microsoft Office
- 3).- Adobe Photoshop

Estos programas son desarrollados y distribuidos bajo un modelo propietario, donde el control del software permanece en manos de la empresa creadora.

IMPLICACIÓN LEGAL

El uso incorrecto de licencias puede generar:

- 1).- Problemas legales
- 2).- Sanciones económicas
- 3).- Riesgos de seguridad

Ejemplo

Una organización utiliza software sin licencia:

Consecuencias:

- 1.- Auditorías
- 2.- Multas
- 3.- Daño reputacional

IMPORTANCIA ESTRATÉGICA EN ENTORNOS REALES

El análisis de requerimientos no es solo una etapa técnica, sino estratégica:

- 1).- Define el éxito o fracaso de un proyecto
- 2).- Impacta directamente en costos
- 3).- Determina la sostenibilidad del sistema

EL ANÁLISIS DE REQUERIMIENTOS: DONDE REALMENTE EMPIEZA TODO

Muchas veces se cree que un proyecto tecnológico empieza cuando:

- 1).- Se compra el software
- 2).- Se instalan equipos
- 3).- Se contrata personal

Pero en realidad, todo comienza mucho antes, realmente empieza cuando alguien se pregunta: ¿Qué necesitamos realmente?, es justamente que en ese momento es el análisis de requerimientos.

PENSAR ANTES DE CONSTRUIR

El análisis de requerimientos es como diseñar un puente antes de construirlo.

Si te equivocas en el diseño esto ocurre:

- 1).- El puente no funciona
- 2).- Se cae con el tiempo

En la gestión de entornos informático moderno pasa exactamente lo mismo

2.17. ESQUEMA INTEGRAL DE LOS SISTEMAS OPERATIVOS

Figura 29
Esquema Integral De Los Sistemas Operativos



Ejemplo

Una empresa necesita mejorar su sistema de gestión, pero no analiza bien, estos parámetros:

- 1).- Qué procesos tiene
- 2).- Qué problemas reales existen
- 3).- Qué necesita el usuario

Implementa una solución incorrecta

Resultado:

- 1).- No resuelve el problema
- 2).- Genera nuevos errores

Caso real

Proyectos tecnológicos que fracasan no lo hacen por mala programación, si no que fallan porque resolvieron el problema equivocado.

CAPÍTULO III

ARQUITECTURA Y TIPOS DE SISTEMAS OPERATIVOS

ESTRUCTURA INTERNA Y CLASIFICACIÓN FUNCIONAL DE LOS SISTEMAS OPERATIVOS



El sistema operativo constituye el eje central de todo sistema informático moderno. Su arquitectura define la forma en que se gestionan los recursos, además se ejecutan los procesos y se establece la comunicación entre el hardware y el usuario de una manera directa.

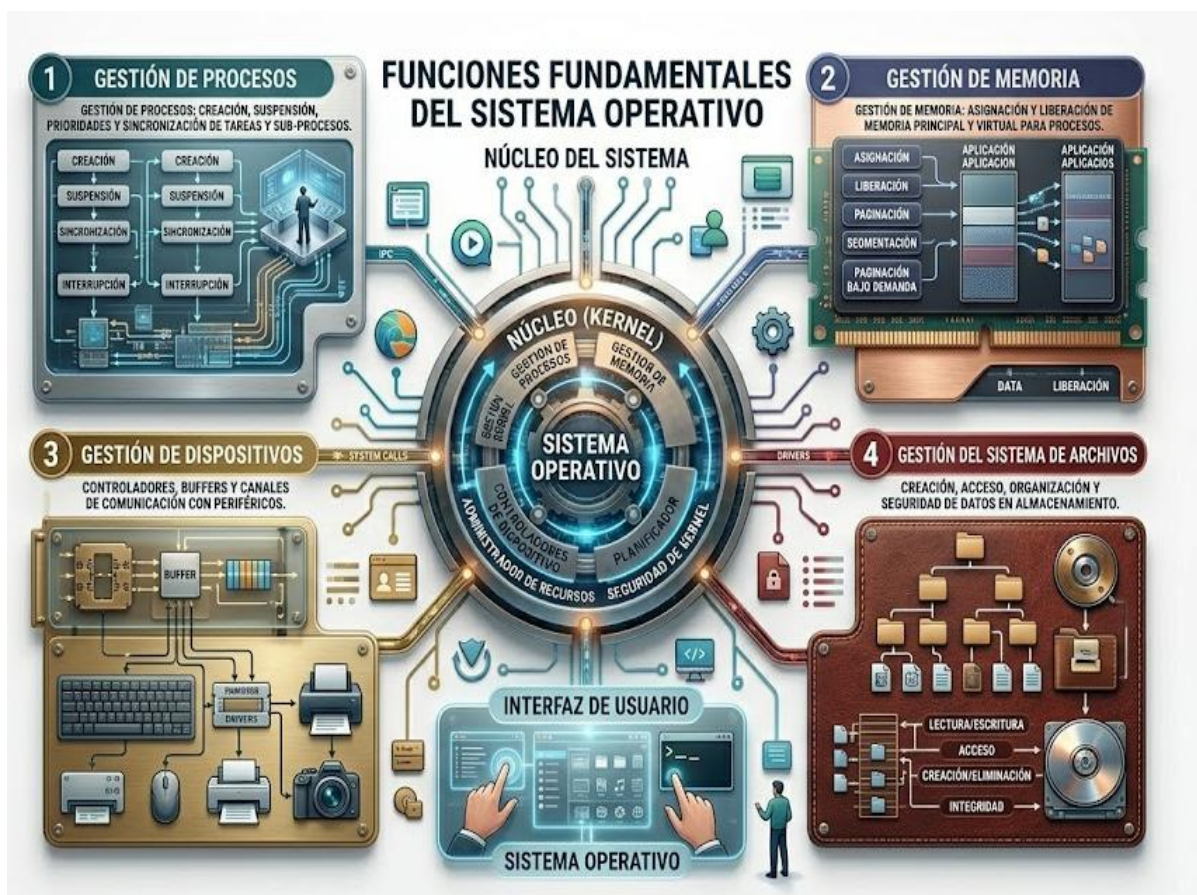
Comprender su estructura interna permite no solo utilizarlo correctamente, sino también tomar decisiones técnicas más acertadas en entornos reales.

3.1. FUNCIONES FUNDAMENTALES DEL SISTEMA OPERATIVO

Cuando hablamos de arquitectura en sistemas operativos, no estamos describiendo solo su estructura técnica, sino que también estamos describiendo la forma en que el sistema organiza, controla y ejecuta todo lo que ocurre dentro de una computadora.

Un sistema operativo cumple funciones específicas que garantizan el funcionamiento correcto del sistema, aquellas funciones se detallan a continuación:

Figura 30
Funciones Fundamentales del Sistema Operativo



1).- GESTIÓN DE PROCESOS

La gestión de procesos es la función del sistema operativo encargada de organizar, controlar y supervisar la ejecución de los programas, asegurando que cada uno utilice los recursos del sistema de forma ordenada, eficiente y sin interferir negativamente con otros.

Es quien controla la ejecución de los programas, no se limita a ejecutar programas; Implica un conjunto de decisiones continuas, que se detallan de esta manera:

- a). - Cuándo inicia un proceso
- b). - Cuánto tiempo usa el procesador

- c). - Cuándo debe esperar
- d). - Cuándo debe finalizar

En esencia, la gestión de procesos convierte múltiples solicitudes simultáneas en un flujo controlado de ejecución.

Ejemplo 1: Videollamada con múltiples aplicaciones

Estás en una videollamada mientras:

- a). - Compartes pantalla
- b). - Tienes un navegador abierto
- c). - Recibes mensajes

Lo que ocurre internamente, es que:

- a). - El sistema prioriza la videollamada (audio y video en tiempo real)
- b). - Reduce recursos de otras aplicaciones
- c). - Mantiene sincronización entre procesos

Si no existiera gestión de procesos, pasaría esto:

- a). - El audio se cortaría
- b). - El video se congelaría

Ejemplo 2: Teléfono con muchas apps abiertas

En un celular, puedes abrir:

- a). - Redes sociales
- b). - Música
- c). - Cámara
- d). - Juegos

Llega un momento y pasa que:

- a). - El sistema cierra aplicaciones en segundo plano

¿Por qué?

Porque la gestión de procesos decide que:

- a). - Mantener todo activo afectaría el rendimiento
- b). - Es mejor liberar recursos

Ejemplo 3: Computadora lenta al iniciar

Cuando enciendes tu computadora pasa lo siguiente:

- a). - Muchos programas se ejecutan automáticamente

El sistema operativo:

- a). - No ejecuta todo al mismo tiempo
- b). - Organiza el inicio por prioridades

Si hay demasiados procesos, el arranque se vuelve lento

Ejemplo 4: Impresión mientras trabajas

Estás escribiendo un documento y envías a imprimir.

Que pasa ahora que el sistema:

- a). - No detiene tu trabajo
- b). - Crea un proceso independiente para la impresión
- c). - Coordina ambos procesos

Esto es gestión de procesos en acción

2).-GESTIÓN DE MEMORIA

La gestión de memoria es la función del sistema operativo encargada de administrar, asignar, controlar y optimizar el uso de la memoria RAM, garantizando que los procesos tengan el espacio necesario para ejecutarse sin interferir entre sí.

No se limita a guardar datos temporalmente, lo que hace es administrar el uso de la memoria RAM

Esto implica decir que:

- a). - Qué proceso usa memoria
- b). - Cuánta memoria se le asigna
- c). - Cuándo se libera
- d). - Qué hacer cuando la memoria se agota

IMAGINA LA MEMORIA COMO UNA MESA DE TRABAJO:

- a). - Cada proceso necesita espacio para trabajar
- b). - La mesa tiene tamaño limitado

Si hay demasiadas tareas, pasa lo siguiente:

- a). - Algunas deben esperar
- b). - otras deben de trabajar en un espacio alternativo

El sistema operativo actúa como quien organiza esa mesa:

- a). - Decide quién trabaja
- b). - Quién espera
- c). - Quién debe retirarse

Ejemplo 1: Muchas pestañas abiertas en el navegador

Tienes abierto:

- a). - 15 o más pestañas
- b). - Algunas con videos
- c). - Otras con contenido pesado

Que ocurre en esos momentos:

- a). - Cada pestaña consume memoria
- b). - El sistema empieza a distribuir la RAM

Si no es suficiente:

- a). - Usa memoria virtual (disco)

Resultado:

- a). - El navegador se vuelve lento
- b). - Cambiarte entre pestañas tarda más

Ejemplo 2: Computadora lenta sin razón aparente

No estás haciendo nada pesado

Pero la maquina esta lenta y responde con retraso:

Posible causa:

- a). - Muchos procesos en segundo plano
- b). - Uso excesivo de RAM

El sistema empieza a usar memoria virtual, esto reduce el rendimiento

Ejemplo 3: Programa que se bloquea

Un software intenta usar más memoria de la disponible, que ocurre en ese instante:

- a). - Limitarlo
- b). - Forzar su cierre
- c). - Provocar un error

Esto evita que afecte a todo el sistema

3.-GESTIÓN DE DISPOSITIVOS

La gestión de dispositivos es la función del sistema operativo encargada de controlar, coordinar y facilitar la comunicación entre el sistema y los dispositivos físicos (periféricos), utilizando programas especializados llamados controladores.

No se limita a conectar dispositivos, si no que controla periféricos mediante controladores
Internamente realiza esto:

- a). - Reconocer el dispositivo
- b). - Interpretar cómo funciona
- c). - Traducir órdenes del sistema
- d). - Coordinar su uso con otros procesos

Los controladores son programas que actúan como intérpretes entre:

- a). - El sistema operativo
- b). - El dispositivo físico

Ejemplo 1: Conectar una impresora nueva Conectas una impresora a tu computadora.

Lo que ocurre internamente:

- a). - El sistema detecta el dispositivo
- b). - Busca o instala el controlador
- c). - Habilita su uso

Si el controlador no existe:

La impresora no funciona correctamente, aunque esté conectada

Ejemplo 2: USB que no se reconoce Conectas una memoria USB y no aparece.

Posibles razones:

- a). - Falta de controlador
- b). -Error en la comunicación
- c). - Problema del dispositivo

El sistema no puede interactuar con ella

Ejemplo 3: Uso simultáneo de un dispositivo

Dos programas intentan usar la impresora al mismo tiempo.

El sistema operativo:

- a). - Organiza una cola de impresión
- b). - Atiende solicitudes en orden

Resultado: Evita conflictos

Ejemplo 4: Cámara en videollamada

Estás en una videollamada usando la cámara.

El sistema:

- a). - Asigna el dispositivo a la aplicación
- b). - Controla su uso en tiempo real
- c). - Evita que otra aplicación interfiera

4.-GESTIÓN DEL SISTEMA DE ARCHIVOS

La gestión del sistema de archivos es la función del sistema operativo encargada de organizar, almacenar, estructurar y controlar el acceso a la información dentro de los dispositivos de almacenamiento, permitiendo que los datos sean localizables, seguros y utilizables.

No se limita a guardar archivos, si no que organiza y administra la información

Esto Implica:

- a). - Definir cómo se almacenan los datos
- b). - Organizar la información de forma lógica
- c). - Controlar quién accede y cómo accede
- d). - Mantener la integridad de los datos

Ejemplo 1: Buscar un archivo rápidamente

Necesitas un documento guardado hace semanas.

Gracias al sistema de archivos:

- a). - Tiene nombre
- b). - Está en una carpeta específica
- c). - Puedes encontrarlo en segundos

Sin esa organización, tendrías que revisar todo el disco manualmente

Ejemplo 2: Guardar archivos con estructura

Un estudiante organiza su información:

- a). - Carpeta “Clases”
- b). - Subcarpetas por materia
- c). - Archivos por tema

Esto refleja cómo el sistema de archivos permite organizar la información de forma lógica

Ejemplo 3: Acceso restringido

En una empresa:

- a). - Algunos archivos solo pueden ser vistos por ciertos usuarios
- b). - El sistema de archivos controla permisos

Ejemplo 4: Disco lleno y desordenado

Cuando el almacenamiento está casi lleno:

- a). - Guardar archivos se vuelve lento
- b). - El sistema tiene dificultad para organizar espacio, esto afecta el rendimiento

5).- INTERFAZ DE USUARIO

La interfaz de usuario es la función del sistema operativo que permite la comunicación directa entre la persona y el sistema, traduciendo las acciones humanas en instrucciones comprensibles para la máquina y mostrando respuestas de forma interpretable.

No es solo, lo que ves en pantalla, si no que permite la interacción con el sistema

Es el punto donde dos mundos se encuentran:

- a). - El humano (intuitivo, visual, abstracto)
- b). - El sistema (lógico, estructurado, binario)

Ejemplo 1: Abrir una aplicación

Haces clic en un icono.

Lo que ocurre:

- a). - La interfaz detecta la acción
- b). - La traduce en una orden
- c). - El sistema ejecuta el programa
- d). - La interfaz muestra la ventana

Todo ocurre en segundos, pero es un proceso de traducción completo

Ejemplo 2: Escribir un documento

Presionas teclas en el teclado.

La interfaz:

- a). - Interpreta cada tecla
- b). - La convierte en caracteres visibles
- c). - Los muestra en la pantalla

Sin la interfaz, escribir sería imposible

Ejemplo 3: Error del sistema

Aparece el mensaje: Archivo no encontrado

La interfaz está haciendo lo siguiente:

- a). - Traducir un problema técnico
- b). - Es un mensaje comprensible

Ejemplo 4: Pantalla táctil

Tocas un botón en un celular.

La interfaz:

- a). - Interpreta el toque
- b). - Identifica la acción
- c). - Ejecuta la orden correspondiente

3.2. ESQUEMA FUNCIONAL DE LOS SISTEMAS OPERATIVOS

Figura 31
Esquema Funcional de los Sistemas Operativos



3.3. ARQUITECTURA DE LOS SISTEMAS OPERATIVOS

La arquitectura de los sistemas operativos es la forma en que se organizan, estructuran y coordinan internamente todos sus componentes para controlar el hardware y gestionar la ejecución de programas.

No se trata solo de cómo está construido el sistema, sino de cómo toma decisiones, cómo distribuye el control y cómo mantiene el orden interno.

La arquitectura define cómo se organizan internamente los componentes del sistema operativo.

3.3.1. TIPOS DE ARQUITECTURA DE LOS SISTEMAS OPERATIVOS

Figura 32
Tipos de Arquitectura de los Sistemas Operativos



1).- ARQUITECTURA MONOLÍTICA

Una arquitectura monolítica se caracteriza porque:

- 1).- Todo el sistema está unificado en un solo bloque de código.
- 2).- Los distintos módulos están estrechamente acoplados.
- 3).- La aplicación se despliega como una sola pieza.

En otras palabras, no hay separación independiente de servicios; todo funciona dentro del mismo entorno y proceso.

Todos los servicios del sistema operativo funcionan como una sola unidad, alta eficiencia, pero menor modularidad

Ejemplo

Sistemas tipo UNIX tradicionales utilizan este enfoque, lo que les permite gran rendimiento, pero con mayor complejidad en el mantenimiento.

Teoría de Caso

SISTEMA DE COMERCIO ELECTRÓNICO

Para comprender mejor la arquitectura monolítica, imaginemos el desarrollo de un sistema para una tienda en línea. Este sistema permite a los usuarios realizar diversas acciones, tales como:

- 1).- Visualizar productos
- 2).- Registrarse e iniciar sesión
- 3).- Realizar compras
- 4).- Gestionar pedidos
- 5).- Administrar inventario

En este caso, todas estas funcionalidades forman parte de una única aplicación, característica principal de la arquitectura monolítica.

FUNCIONAMIENTO EN LA PRÁCTICA

Aunque el sistema esté dividido conceptualmente en diferentes módulos, todos ellos están integrados dentro de un mismo proyecto. Entre los principales componentes se encuentran:

- 1).- **Módulo de usuarios:** encargado del registro y la autenticación
- 2).- **Módulo de productos:** responsable de mostrar el catálogo
- 3).- **Módulo de pedidos:** gestiona las compras realizadas
- 4).- **Módulo de pagos:** procesa las transacciones

SITUACIÓN REAL DE MANTENIMIENTO

En un entorno real, si un desarrollador necesita realizar un cambio en el módulo de pagos, no puede hacerlo de forma aislada. En su lugar, debe:

- 1).- Modificar el proyecto completo
- 2).- Volver a compilar toda la aplicación
- 3).- Desplegar nuevamente el sistema en su totalidad

Esto ocurre incluso si el cambio afecta únicamente a una pequeña parte del sistema.

INTEGRACIÓN DEL SISTEMA

En una arquitectura monolítica, todos los elementos fundamentales del sistema están directamente conectados dentro de una misma aplicación:

- 1).- La interfaz de usuario
- 2).- La lógica de negocio
- 3).- El acceso a la base de datos

Esta integración simplifica el desarrollo inicial, pero incrementa la dependencia entre los componentes.

2).- ARQUITECTURA EN CAPAS

La arquitectura en capas es un modelo de diseño de software que organiza una aplicación en niveles o capas jerárquicas, donde cada una cumple una función específica dentro del sistema.

Este enfoque busca separar responsabilidades, facilitando el desarrollo, mantenimiento y comprensión del software.

En la arquitectura en capas, el sistema se divide en diferentes niveles, donde:

- 1).- Cada capa tiene una responsabilidad definida
- 2).- Las capas se comunican entre sí de forma ordenada
- 3).- Generalmente, una capa solo interactúa con la capa inmediatamente inferior o superior

Esto permite una estructura más organizada y modular dentro de una aplicación.

VENTAJA

Mayor organización y facilidad de mantenimiento

3).- ARQUITECTURA CLIENTE-SERVIDOR (MICROKERNEL)

La arquitectura cliente-servidor es un modelo en el cual las aplicaciones se dividen en dos partes principales; Los clientes, que solicitan servicios, y servidores, que los proporcionan.

Por otro lado, la arquitectura MICROKERNEL (también conocida como arquitectura de núcleo mínimo) es un enfoque donde el sistema se construye alrededor de un núcleo central muy pequeño, al que se le agregan funcionalidades mediante módulos independientes.

Aunque son conceptos distintos, pueden relacionarse porque ambos buscan organizar el sistema de forma modular y eficiente.

En este modelo:

- 1).- El cliente es quien inicia la comunicación (por ejemplo, una aplicación o navegador)
- 2).- El servidor responde proporcionando recursos o servicios (datos, procesamiento, etc.)

Ejemplo

Un usuario accede a una página web desde su navegador (cliente), y un servidor procesa la solicitud y devuelve la información.

Resultado: Sistemas modernos buscan este enfoque para mejorar la seguridad y estabilidad.

4).- ARQUITECTURA HÍBRIDA

La arquitectura híbrida es un enfoque de diseño de software que combina dos o más estilos arquitectónicos dentro de un mismo sistema, su objetivo es aprovechar las ventajas de diferentes modelos para construir soluciones más flexibles, escalables y adaptables.

Este tipo de arquitectura surge como respuesta a las limitaciones de usar un solo modelo, especialmente en sistemas complejos.

En una arquitectura híbrida:

- 1).- No existe un único patrón dominante
- 2).- Se integran diferentes arquitecturas según las necesidades del sistema
- 3).- Cada parte del sistema puede utilizar el enfoque más adecuado

Ejemplo

Muchos sistemas actuales utilizan arquitecturas híbridas para equilibrar rendimiento y modularidad.

3.4. EL NÚCLEO KERNEL

El núcleo (KERNEL) es el componente central de un sistema operativo, es quien actúa como el intermediario fundamental entre el hardware de la computadora y el software que se ejecuta en ella.

Sin el núcleo KERNEL, todas las aplicaciones no podrían comunicarse directamente con los recursos físicos del sistema, como el procesador, la memoria o los dispositivos de entrada y salida.

Figura 33
El núcleo KERNEL engranaje del Sistemas



El núcleo KERNEL es la parte del sistema operativo que:

- 1).- Controla y gestiona los recursos del hardware
- 2).- Permite la ejecución de programas
- 3).- Facilita la comunicación entre software y hardware

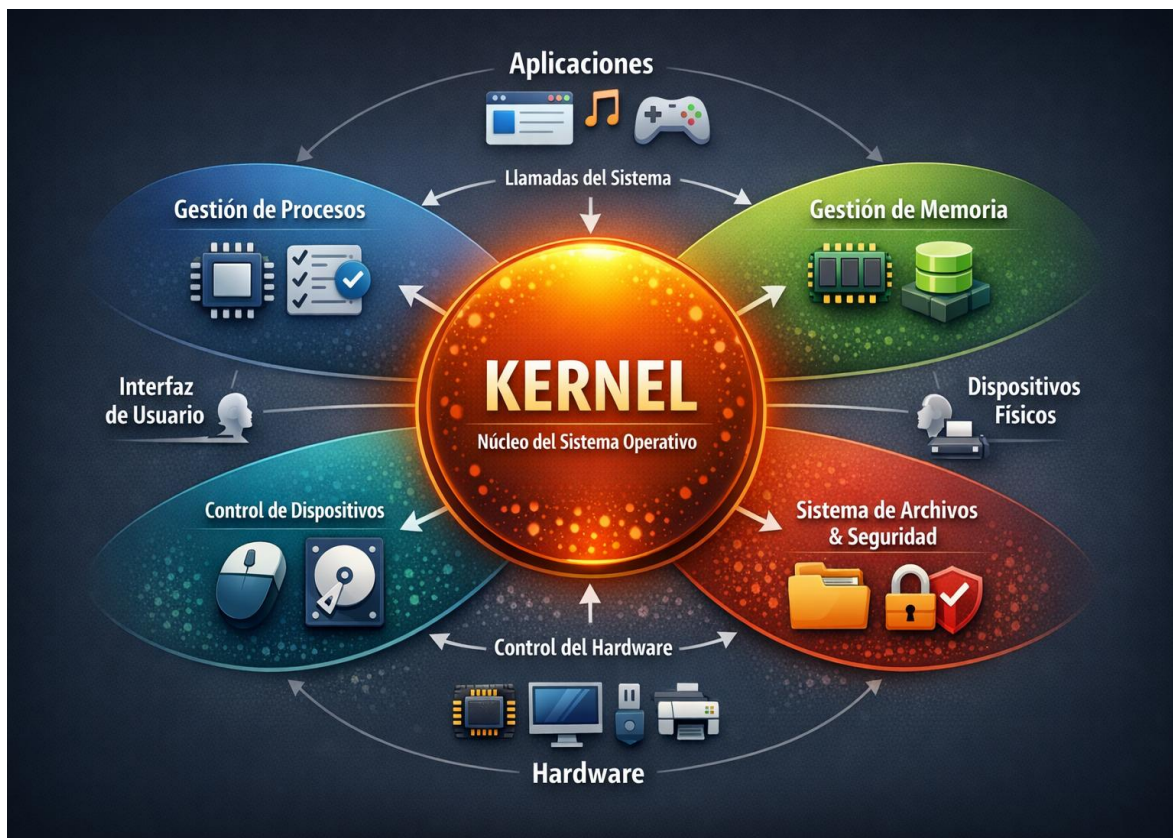
En términos simples, el núcleo KERNEL es el cerebro operativo que coordina todo lo que ocurre en el sistema.

FUNCIONES PRINCIPALES:

- 1).- Control de procesos
- 2).- Gestión de memoria
- 3).- Comunicación entre hardware y software
- 4).- Manejo de interrupciones

3.5. REPRESENTACIÓN CONCEPTUAL DEL KERNEL

Figura 34
Representación Conceptual del Kernel



CAPÍTULO IV

INSTALACIÓN DE SISTEMAS OPERATIVOS Y SUS APLICACIONES



4.1. PROCEDIMIENTOS, MÉTODOS Y BUENAS PRÁCTICAS PARA UNA IMPLEMENTACIÓN EFICIENTE

La instalación de sistemas operativos y sus aplicaciones constituye una de las fases más críticas en la gestión de entornos informáticos moderno. Ya que una implementación adecuada no solo garantiza el correcto funcionamiento del sistema, sino que también previene errores futuros, mejora el rendimiento y fortalece la seguridad y el equipo trabaja correctamente, adicionalmente en este proceso se requiere planificación, conocimiento técnico y aplicación de buenas prácticas.

4.2. REQUISITOS TÉCNICOS Y DE COMPATIBILIDAD

Antes de instalar cualquier sistema operativo o alguna aplicación, es indispensable verificar que el equipo cumpla con los requisitos necesarios.

4.2.1. TIPOS DE REQUISITOS

1).- Requisitos mínimos:

Permiten ejecutar el sistema de forma básica

2).- Requisitos recomendados:

Garantizan un rendimiento óptimo

Tabla 3.

Tipos de requisitos

REQUISITO	MÍNIMO	RECOMENDADO
RAM	4 GB	8–16 GB
Procesador	2 núcleos	4+ núcleos
Almacenamiento	50 GB	SSD 100+ GB

Teoría de caso

Un usuario instala un sistema operativo con requisitos mínimos:

Resultado:

- a). - Funcionamiento lento
- b). - Fallos en aplicaciones
- c). - Mala experiencia de usuario

Siempre es recomendable considerar los requisitos.

PREPARACIÓN DEL ENTORNO DE INSTALACIÓN

Antes de iniciar la instalación, se deben realizar acciones previas:

- a). - Respaldo de información (backup)
- b). - Verificación del hardware
- c). - Obtención de medios de instalación
- d). - Configuración de arranque (BIOS/UEFI)

Error común que hacen al momento de instalar un sistema operativo

No realizar copia de seguridad:

Resultado: Pérdida irreversible de los datos

4.3. MÉTODOS DE INSTALACIÓN DE LOS SISTEMAS OPERATIVOS

Los métodos de instalación de los sistemas operativos constituyen el conjunto de procedimientos, técnicas y estrategias mediante los cuales un sistema operativo es transferido, configurado e integrado en un dispositivo informático, garantizando su correcto funcionamiento sobre el hardware disponible.

No se trata únicamente de instalar un programa, sino de un proceso más complejo que implica:

- a). - Preparar el entorno del sistema
- b). - Adaptar el software al hardware
- c). - Definir configuraciones iniciales
- d). - Establecer la base operativa del equipo

En otra palabra el método de instalación es el proceso mediante el cual una computadora pasa de ser una máquina sin funcionalidad definida a convertirse en un sistema capaz de ejecutar tareas, interactuar con el usuario y gestionar recursos.

Figura 35
Métodos de Instalación de Sistemas Operativos



Ejemplo

Imagina tres situaciones diferentes:

- a). - Un usuario instala Windows 11 desde una USB en su casa
- b). - Una empresa instala Linux en 100 computadoras desde un servidor
- c). - Un estudiante instala Ubuntu en una máquina virtual

Aunque el objetivo es el mismo (tener un sistema operativo), los métodos cambian completamente según el contexto.

ENFOQUE EN LA REALIDAD ACTUAL

En la actualidad, los métodos de instalación han evolucionado junto con la tecnología:

- a). - **Antes:** discos físicos y procesos manuales
- b). - **Ahora:** instalaciones automatizadas, virtualizadas y en red

Esto implica que responde a nuevas necesidades como:

- 1).- Rapidez
- 2).- Escalabilidad
- 3).- Seguridad
- 4).- Flexibilidad

INSTALACIÓN MANUAL

La instalación manual es un método en el cual el usuario participa activamente en todo el proceso de instalación del sistema operativo, tomando decisiones en cada etapa, esto a su vez implica que el sistema no se instala de forma automática, sino que el usuario debe:

- 1).- Seleccionar opciones
- 2).- Configurar parámetros
- 4).- Confirmar cada paso del proceso

Es decir, una instalación guiada, donde el sistema te orienta, pero tú tomas las decisiones.

CARACTERÍSTICAS PRINCIPALES

- 1).- Proceso guiado paso a paso
- 2).- Intervención directa del usuario
- 3).- Configuración personalizada
- 4).- Mayor control sobre el sistema
- 5).- Ideal para aprendizaje y formación técnica

Figura 36
Instalación Manual de Sistema Operativo



Ejemplo

Instalación desde USB booteable

Un caso típico de instalación manual es cuando una persona instala un sistema operativo desde una memoria USB.

Procedimiento:

- 1).- Se crea una USB con el sistema operativo
- 2).- Se reinicia la computadora
- 3).- Se inicia desde la USB
- 4).- El sistema muestra un asistente de instalación

El usuario elige:

- 1).- Idioma
- 2).- Tipo de instalación
- 3).- Particiones del disco
- 4).- Se completa la instalación paso a paso

Teoría de Caso

Un técnico instala Windows 11 en una computadora nueva.

Durante el proceso:

- 1).- Decide cómo dividir el disco (particiones)
- 2).- Configura el idioma y región
- 3).- Define usuario y contraseña

Resultado:

Un sistema adaptado exactamente a las necesidades del usuario

VENTAJAS Y DESVENTAJAS

Ventajas

- 1).- Control total del proceso
- 2).- Permite personalización
- 3).- Ayuda a entender el sistema
- 4).- Adaptable a diferentes escenarios

Desventajas

- 1).- Requiere conocimientos básicos
- 2).- Puede tomar más tiempo
- 3).- Riesgo de errores si no se sabe qué elegir

INSTALACIÓN AUTOMATIZADA

La instalación automatizada es un método en el que el sistema operativo se instala de forma automática, utilizando configuraciones previamente definidas, sin necesidad de que el usuario intervenga en cada paso.

Esto se logra mediante:

- 1).- Scripts (archivos con instrucciones)
- 2).- Archivos de configuración
- 3).- Respuestas predefinidas

Una instalación automática se hace sola, porque ya sabe qué decisiones tomar.

Figura 37
Instalación Automatizada de Sistema Operativo

Instalación Automatizada de Sistema Operativo



CARACTERÍSTICAS PRINCIPALES

- 1).- Uso de scripts o archivos de configuración
- 2).- Reduce la intervención humana
- 3).- Instalación rápida y eficiente
- 4).- Configuración estandarizada
- 5).- Repetible en múltiples equipos

Ejemplo

Instalación con archivo de respuestas

Un técnico crea un archivo donde define:

- 1).- Idioma
- 2).- Zona horaria
- 3).- Tipo de partición
- 4).- Usuario y contraseña

Luego, al iniciar la instalación:

- 1).- El sistema lee ese archivo
- 2).- Ejecuta todo automáticamente

Resultado: El usuario no tiene que hacer clic en cada paso.

Una empresa necesita instalar Windows 11 en 50 computadoras.

En lugar de hacerlo manualmente una por una:

- 1).- Se crea una instalación automatizada
- 2).- Todas las computadoras reciben la misma configuración
- 3).- El proceso se ejecuta sin intervención constante

Resultado:

- 1).- Ahorro de tiempo
- 2).- Menos errores
- 3).- Mayor eficiencia

VENTAJAS Y DESVENTAJAS

Ventajas

- 1).- Ahorra tiempo
- 2).- Reduce errores humanos
- 3).- Permite instalar en muchos equipos
- 4).- Configuración uniforme

Desventajas

- 1).- Requiere preparación previa
- 2).- Menor flexibilidad durante la instalación
- 3).- Necesita conocimientos técnicos
- 4).- Si hay un error en el script, se repite en todos los equipos

INSTALACIÓN POR RED

La instalación por red es un método en el cual el sistema operativo se instala desde un servidor central a través de una red, sin necesidad de utilizar medios físicos como USB o DVD.

La computadora descarga e instala el sistema operativo directamente desde otra máquina conectada en la red.

Figura 38
Instalación por red



CARACTERÍSTICAS PRINCIPALES

- 1).- El sistema se instala desde un servidor
- 2).- No requiere USB ni discos físicos
- 3).- Ideal para entornos corporativos
- 4).- Permite instalar en varios equipos simultáneamente

Ejemplo

Imagina un laboratorio de computación con 30 equipos.

En lugar de instalar el sistema uno por uno:

- 1).- Se configura un servidor con el sistema operativo
- 2).- Las computadoras se conectan a la red
- 3).- Inician desde la red (boot por red)
- 4).- El sistema se instala automáticamente desde el servidor

Todo el proceso se realiza sin usar memorias USB.

Teoría de Caso

Una institución educativa necesita instalar el sistema operativo Linux en todas las computadoras de un laboratorio.

Solución:

- 1).- Configuran un servidor central
- 2).- Todas las máquinas se conectan a él
- 3).- Se instala el sistema en todas al mismo tiempo

Resultado:

- 1).- Ahorro de tiempo
- 2).- Instalación uniforme
- 3).- Control centralizado

VENTAJAS Y DESVENTAJAS

Ventajas

- 1).- Instalación masiva (muchos equipos a la vez)
- 2).- No requiere dispositivos físicos
- 3).- Ahorra tiempo en grandes entornos
- 4).- Control centralizado

Desventajas

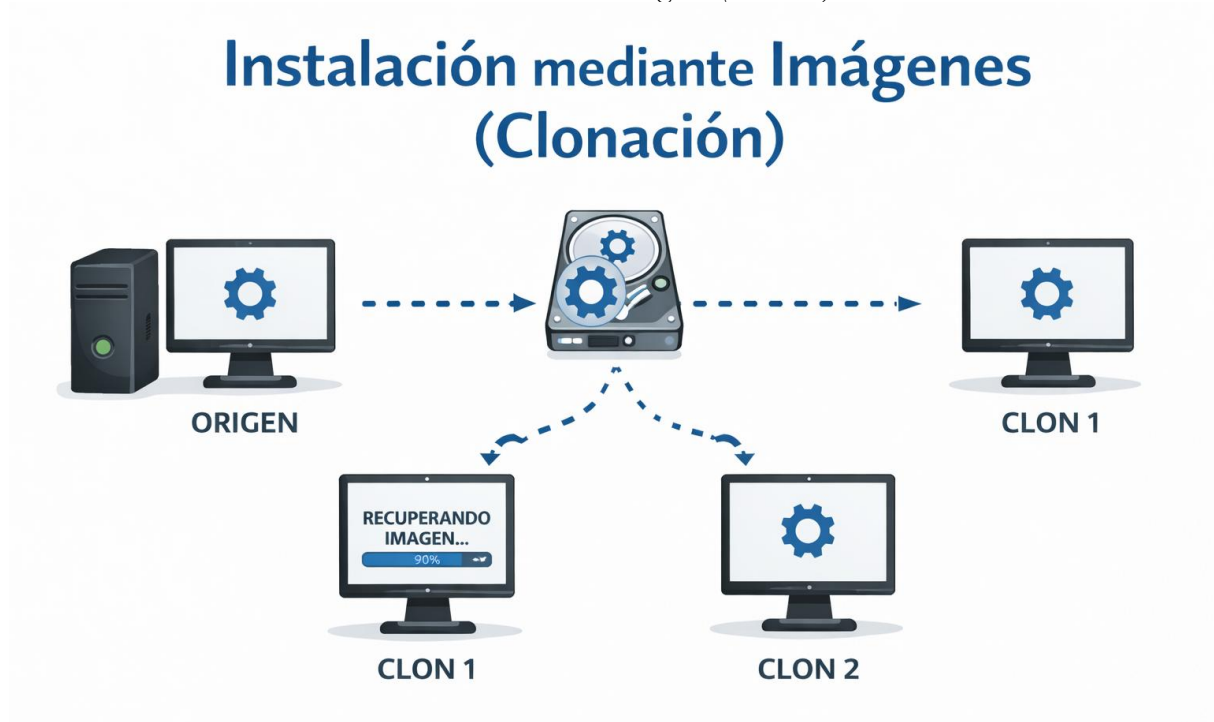
- 1).- Requiere infraestructura de red
- 2).- Configuración inicial más compleja
- 3).- Dependencia del servidor
- 4).- Puede fallar si hay problemas de red

INSTALACIÓN MEDIANTE IMÁGENES (CLONACIÓN)

La instalación mediante imágenes, también conocida como clonación, es un método que consiste en copiar exactamente un sistema operativo ya instalado y configurado para replicarlo en otros equipos.

Es como hacer una copia exacta de una computadora y ponerla en otra.

Figura 39
Instalación mediante imágenes (clonación)



CARACTERÍSTICAS PRINCIPALES

- 1).- Copia exacta de un sistema ya configurado
- 2).- Instalación rápida
- 3).- Configuración idéntica en todos los equipos
- 4).- Muy usado en empresas e instituciones
- 5).- Proceso repetible

Ejemplo

Un técnico configura una computadora con:

- 1).- Sistema operativo
- 2).- Programas necesarios
- 3).- Configuración de red
- 4).- Ajustes personalizados

Luego crea una imagen de ese sistema, esa imagen se copia en otras computadoras, dejando todas exactamente iguales.

Teoría de Caso

Una empresa instala Windows 11 en una computadora base con:

- 1).- Programas de oficina
- 2).- Antivirus
- 3).- Configuración empresarial

Luego clona ese sistema a 50 computadoras.

Resultado:

- 1).- Todas las máquinas tienen la misma configuración
- 2).- Se reduce el tiempo de instalación
- 3).- Se evitan errores humanos

VENTAJAS Y DESVENTAJAS

Ventajas

- 1).- Muy rápido
- 2).- Configuración uniforme
- 3).- Reduce errores
- 4).- Ideal para implementaciones masivas

Desventajas

- 1).- Poca flexibilidad (todo queda igual)
- 2).- Puede copiar errores si el sistema base no está bien configurado
- 3).- Requiere herramientas especiales
- 4).- Problemas si el hardware es diferente

INSTALACIÓN DE APLICACIONES

La instalación de aplicaciones es el proceso mediante el cual un programa es incorporado a un sistema operativo para que pueda ejecutarse correctamente y cumplir una función específica.

Este proceso no solo implica copiar archivos, sino también:

- 1).- Integrar la aplicación con el sistema
- 2).- Configurar dependencias necesarias
- 3).- Garantizar su correcto funcionamiento

Instalar una aplicación es preparar un programa para que funcione de manera adecuada dentro de tu computadora o dispositivo.

Figura 40
Instalación de aplicaciones

Instalación de Aplicaciones



1).- **Compatibilidad**

Se refiere a que la aplicación debe ser compatible con el sistema operativo y el hardware del equipo.

Ejemplo

Un programa diseñado para una versión moderna puede no funcionar en un sistema antiguo.

Teoría de Caso

Intentar instalar una aplicación avanzada en un equipo con pocos recursos puede provocar:

- 1).- Lentitud
- 2).- Errores
- 3).- Fallos del sistema

Ejemplo

Instalar un software pesado en un equipo con poca memoria RAM afecta el rendimiento.

2).- NECESIDAD

Implica instalar solo las aplicaciones que realmente que se necesitan.

Ejemplo

Instalar muchos programas por si acaso, esto puede saturar el sistema.

Teoría de Caso

Un usuario instala múltiples aplicaciones que no utiliza.

Resultado:

- 1).- Menor rendimiento
- 2).- Mayor consumo de espacio
- 3).- Posibles riesgos de seguridad

Ejemplo

Un usuario instala un navegador en su computadora.

Antes de hacerlo debería considerar:

- 1).- Si su sistema lo soporta
- 2).- Si realmente lo necesita
- 3).- Si es seguro y confiable

VALIDACIÓN POSTERIOR A LA INSTALACIÓN

La validación posterior a la instalación es el conjunto de verificaciones que se realizan después de instalar un sistema operativo, con el objetivo de asegurar que todo funciona correctamente, no solo basta con que el sistema arranque; si no que es necesario comprobar que:

- 1).- Todos los componentes funcionan
- 2).- El sistema responde adecuadamente
- 3).- No existen errores ocultos

Es como revisar que todo esté bien después de armar algo, para asegurarse de que realmente funciona.

ELEMENTOS DE VALIDACIÓN

COMPROBACIÓN DE CONTROLADORES

Los controladores (drivers) permiten que el sistema operativo se comunice con el hardware.

Ejemplo

Verificar si funcionan correctamente:

- 1).- Audio
- 2).- Video
- 3).- Teclado
- 4).- Red

Teoría de Caso

Después de instalar Windows 11, una computadora no tiene sonido.

Problema:

- 1).- Falta el controlador de audio

Solución:

- 1).- Instalar el driver correcto

VERIFICACIÓN DE CONECTIVIDAD

Consiste en comprobar que el equipo puede conectarse a redes (internet o red local).

Ejemplo

- 1).- Conectarse a WiFi
- 2).- Navegar en internet
- 3).- Acceder a recursos en red

Teoría de Caso

Una computadora recién instalada no detecta redes WiFi.

Esto puede indicar:

- 1).- Problema de configuración
- 2).- Falta de controlador de red

PRUEBAS DE RENDIMIENTO

Evalúan si el sistema funciona de manera rápida y eficiente.

Ejemplo

- 1).- Abrir programas
- 2).- Ejecutar tareas
- 3).- Medir tiempos de respuesta

Teoría de Caso

Un sistema recién instalado funciona muy lento.

Posibles causas:

- 1).- Hardware insuficiente
- 2).- Configuración incorrecta
- 3).- Procesos innecesarios

REVISIÓN DE ESTABILIDAD

Consiste en comprobar que el sistema no presenta fallos, errores o bloqueos.

Ejemplo

- 1).- Uso continuo del sistema
- 2.- Ejecución de varias aplicaciones
- 3.- Reinicios de prueba

Teoría de Caso real

Una computadora se reinicia sola después de la instalación.

Esto puede indicar que tiene:

- 1).- Problemas de drivers
- 2).- Fallos de hardware
- 3).- Errores en la instalación

4.4. SISTEMAS DE ARCHIVOS Y GESTIÓN DEL ALMACENAMIENTO

El almacenamiento de la información es uno de los pilares fundamentales de cualquier sistema informático moderno. No solo basta con disponer de dispositivos físicos; es necesario contar con mecanismos que permitan organizar, acceder y proteger los datos de manera eficiente, para que funcionen correctamente y no causen inconveniente en el futuro, en cambio los sistemas de archivos cumplen esta función, actuando como una capa lógica que estructura la información dentro de los dispositivos de almacenamiento.

Figura 41

Sistemas de archivos y gestión del almacenamiento



4.4.1. CONCEPTO DE SISTEMA DE ARCHIVOS

Un sistema de archivos es el método y estructura que utiliza un sistema operativo para organizar y gestionar datos en dispositivos de almacenamiento.

Estos a su vez permiten:

- 1).- Crear, modificar y eliminar archivos
- 2).- Organizar la información en directorios
- 3).- Controlar accesos y permisos
- 4).- Optimizar el uso del espacio

4.5. ESTRUCTURA LÓGICA DEL ALMACENAMIENTO

Figura 42
Estructura lógica del almacenamiento



Antes de aplicar un sistema de archivos, el almacenamiento debe organizarse en particiones.

PARTICIONES

Una partición es una división lógica de un dispositivo físico.

TIPOS PRINCIPALES:

1).- PARTICIÓN PRIMARIA

Permite instalar sistemas operativos

2).- PARTICIÓN EXTENDIDA

Contiene múltiples particiones lógicas

3).- PARTICIONES LÓGICAS

Subdivisiones dentro de la extendida

Teoría de Caso

Un equipo tiene:

- 1).- Una partición para el sistema operativo
- 2).- Otra para datos personales

Beneficio:

- 1).- Mayor seguridad ante fallos
- 2).- Facilidad de respaldo

4.6. TIPOS DE SISTEMAS DE ARCHIVOS

Existen diversos sistemas de archivos, cada uno con características específicas.



FAT (File Allocation Table) es un sistema de archivos utilizado por los sistemas operativos para organizar, almacenar y gestionar la información dentro de un dispositivo de almacenamiento (como discos duros o memorias USB).

FAT es como un índice que le dice al sistema dónde está guardado cada archivo.

CARACTERÍSTICAS PRINCIPALES

1).- ESTRUCTURA SIMPLE

FAT tiene un diseño sencillo, lo que facilita su uso y comprensión.

Esto significa:

- a). - Fácil implementación
- b). - Bajo consumo de recursos
- c). - Ideal para dispositivos básicos

2).- ALTA COMPATIBILIDAD

Uno de los puntos más fuertes de FAT es que puede ser utilizado por muchos sistemas operativos.

Es compatible con:

- a). - Windows
- b). - Linux
- c). - macOS

Ejemplo

Tienes una memoria USB formateada en FAT.

Cuando guardas un archivo:

- 1).- FAT registra en qué partes de la memoria está guardado
- 2).- Cuando lo abres, el sistema usa esa información para reconstruirlo

Teoría de caso

1).- Memorias USB

Muchas memorias USB utilizan FAT (especialmente FAT32) por su compatibilidad.

Puedes conectarlas en diferentes dispositivos sin problemas.

2).- Cámaras digitales

Las cámaras suelen usar FAT para guardar fotos y videos.

Esto permite que las tarjetas funcionen en distintos equipos.

3).- Intercambio de archivos

Cuando necesitas mover archivos entre diferentes sistemas operativos, FAT es una opción práctica.

VENTAJAS Y DESVENTAJAS

VENTAJAS

- 1).- Fácil de usar
- 2).- Alta compatibilidad
- 3).- Ligero y eficiente
- 4).- Ideal para dispositivos portátiles

DESVENTAJAS

- 1).- Limitación en tamaño de archivos
- 2).- Baja seguridad
- 3).- No adecuado para sistemas modernos complejos
- 4).- Mayor riesgo de fragmentación

NTFS (New Technology File System) es un sistema de archivos avanzado desarrollado por Microsoft, diseñado para gestionar, organizar y proteger la información en dispositivos de almacenamiento dentro de los sistemas informáticos modernos.

NTFS es una versión más inteligente y segura de organizar archivos en un disco.

CARACTERÍSTICAS PRINCIPALES

1).- SOPORTE PARA PERMISOS DE SEGURIDAD

NTFS permite definir quién puede acceder, modificar o eliminar archivos.

Ejemplo

En una computadora compartida:

- 1).- Un usuario puede ver un archivo
- 2).- Otro puede editarlo
- 3).- Otro no puede acceder

Teoría de caso

En una empresa con equipos con Windows 11:

El administrador restringe el acceso a archivos importantes para evitar modificaciones indebidas.

2).- **MAYOR ESTABILIDAD**

NTFS incluye mecanismos que reducen errores y evitan la pérdida de datos.

Ejemplo

Si ocurre un apagón inesperado:

- 1).- **NTFS** puede recuperar información
- 2).- Reduce el riesgo de corrupción de archivos

Teoría de caso

Un equipo se apaga repentinamente mientras se guarda un documento.

Con **NTFS**:

- 1).- Es más probable que el archivo no se pierda completamente

3).- **USO EN SISTEMAS MODERNOS**

NTFS es el sistema de archivos estándar en sistemas operativos modernos.

Ejemplo

La mayoría de computadoras actuales utilizan **NTFS** por defecto.

Teoría de caso

Un equipo con Windows 11 utiliza NTFS para:

- 1).- Manejar grandes cantidades de datos
- 2).- Ejecutar aplicaciones modernas
- 3).- Garantizar seguridad y rendimiento

VENTAJAS Y DESVENTAJAS

VENTAJAS

- 1).- Alta seguridad (permisos y control de acceso)
- 2).- Mayor estabilidad
- 3).- Soporte para archivos grandes
- 4).- Mejor gestión del almacenamiento

DESVENTAJAS

- 1).- Menor compatibilidad con algunos sistemas (comparado con FAT)
- 2).- Más complejo
- 3).- Puede consumir más recursos

EXT (Extended File System) es una familia de sistemas de archivos utilizada principalmente en sistemas operativos basados en Linux, diseñada para gestionar, organizar y optimizar el almacenamiento de datos en discos duros y otros dispositivos.

CARACTERÍSTICAS PRINCIPALES

1).- USO EN SISTEMAS BASADOS EN LINUX

EXT es el sistema de archivos más utilizado en distribuciones Linux.

Ejemplo

Cuando instalas una distribución como:

- 1).- Ubuntu
- 2).- Debian

Teoría de caso

Un servidor web que funciona con Linux usa EXT para:

- 1).- Almacenar páginas web
- 2).- Gestionar bases de datos
- 3).- Mantener estabilidad en operaciones continuas

2).- Versiones: **EXT2, EXT3, EXT4**

EXT ha evolucionado con el tiempo para mejorar su rendimiento y seguridad.

Ejemplo

EXT2 → versión básica, sin registro de cambios

EXT3 → añade JOURNALING (registro de eventos para evitar pérdidas)

EXT4 → versión moderna, más rápida, eficiente y estable

Teoría de Caso

Hoy en día, la mayoría de sistemas Linux utilizan EXT4 porque:

- 1).- Maneja grandes volúmenes de datos
- 2).- Reduce errores
- 3).- Mejora el rendimiento

3).- ALTA EFICIENCIA Y RENDIMIENTO

EXT está diseñado para aprovechar mejor los recursos del sistema.

Ejemplo

- 1).- Acceso rápido a archivos
- 2).- Menor fragmentación
- 3).- Mejor uso del espacio

Teoría de caso

Un servidor que maneja miles de usuarios utiliza EXT4 para garantizar:

- 1).- Respuesta rápida
- 2).- Estabilidad continua
- 3).- Bajo riesgo de fallos

VENTAJAS Y DESVENTAJAS

VENTAJAS

- 1).- Alto rendimiento
- 2).- Gran estabilidad
- 3).- Optimizado para Linux
- 4).- Manejo eficiente del almacenamiento

DESVENTAJAS

- 1).- Menor compatibilidad con otros sistemas (como Windows)
- 2).- Requiere conocimientos técnicos
- 3).- No es común en entornos domésticos tradicionales

GESTIÓN DEL ALMACENAMIENTO

La administración eficiente del almacenamiento implica:

- 1).- Control del espacio disponible
- 2).- Optimización del uso del disco
- 3).- Eliminación de datos innecesarios

TÉCNICAS DE OPTIMIZACIÓN

- 1).- Desfragmentación (en HDD)
- 2).- Uso de SSD
- 3).- Compresión de archivos
- 4).- Limpieza del sistema

Teoría de caso

Un equipo presenta lentitud:

Problema:

Disco lleno y fragmentado

Solución:

Limpieza + optimización

Resultado:

Mejora del rendimiento

SEGURIDAD EN EL ALMACENAMIENTO

Los sistemas de archivos modernos incorporan mecanismos de seguridad que realiza lo siguiente:

- 1).- Permisos de acceso
- 2).- Cifrado de datos
- 3).- Control de usuarios

Teoría de caso

Falta de control de permisos:

Consecuencia:

Acceso no autorizado a información sensible

CAPÍTULO V

ACTUALIZACIÓN Y MANTENIMIENTO DE SISTEMAS



5.1. GESTIÓN ESTRATÉGICA PARA LA CONTINUIDAD OPERATIVA Y LA SEGURIDAD INFORMÁTICA

Los sistemas informáticos no son estáticos; evolucionan constantemente debido a nuevas necesidades, mejoras tecnológicas y amenazas de seguridad. Por ello, la actualización y el mantenimiento se convierten en procesos esenciales para garantizar la estabilidad, el rendimiento y la protección de la información.

Una gestión inadecuada puede provocar fallos críticos, vulnerabilidades y pérdida de productividad.

5.2. CONCEPTO DE ACTUALIZACIÓN Y MANTENIMIENTO

ACTUALIZACIÓN:

Proceso de incorporación de mejoras, correcciones o nuevas funcionalidades en un sistema.

MANTENIMIENTO:

Conjunto de acciones destinadas a preservar el correcto funcionamiento del sistema a lo largo del tiempo.

5.2.1. TIPOS DE ACTUALIZACIONES

Figura 44
Tipos de actualizaciones



1).- ACTUALIZACIONES CORRECTIVAS

- a). - Corrigen errores o fallos del sistema
- b). - Solucionan vulnerabilidades

2).- ACTUALIZACIONES EVOLUTIVAS

- a). - Incorporan nuevas funcionalidades
- b). - Mejoran la experiencia del usuario

3).- ACTUALIZACIONES DE SEGURIDAD

- a). - Protegen contra amenazas externas
- b). - Corrigen vulnerabilidades críticas

Teoría de Caso

Una organización no aplica actualizaciones de seguridad:

Resultado:

- 1).- Exposición a ataques
- 2).- Pérdida de información

Las actualizaciones de seguridad son prioritarias.

FUENTES Y MECANISMOS DE ACTUALIZACIÓN

1).- FUENTES CONFIABLES

- a). - Sitios oficiales
- b). - Repositorios autorizados
- c). - Plataformas certificadas

2).- MECANISMOS DE ACTUALIZACIÓN

- a). - Actualizaciones automáticas
- b). - Actualizaciones manuales
- c). - Actualización centralizada (en empresas)

Teoría de caso

Empresa con múltiples equipos:

Implementa servidor de actualizaciones centralizado

Beneficio:

- a). - Control total
- b). - Reducción de consumo de red
- c). - Estandarización

EVALUACIÓN DE COMPATIBILIDAD, IMPACTO Y VIABILIDAD

Antes de aplicar una actualización, es necesario analizar los siguientes aspectos fundamentales:

1).- COMPATIBILIDAD

2).- IMPACTO

3).- VIABILIDAD

COMPATIBILIDAD

- a). - Hardware soportado
- b). - Software existente
- c). - Controladores

IMPACTO

- a). - Cambios en el sistema
- b). - Posibles fallos
- c). - Interrupciones del servicio

VIABILIDAD

- a). - Recursos disponibles
- b). - Tiempo de implementación
- c). - Costos asociados

PROCEDIMIENTOS SEGUROS DE ACTUALIZACIÓN

Para garantizar la estabilidad del sistema, se deben seguir las siguientes prácticas:

- 1).- Realizar copias de seguridad
- 2).- Probar en entornos controlados
- 3).- Aplicar actualizaciones en horarios estratégicos
- 4).- Documentar el proceso

Error común

Actualizar directamente en producción sin pruebas:

Riesgo

- 1).- Fallos críticos
- 2).- Pérdida de información

REGISTRO TÉCNICO Y TRAZABILIDAD

La documentación permite controlar los cambios realizados en el sistema.

ELEMENTOS

- 1).- Versión instalada
- 2).- Fecha de actualización
- 3).- Cambios realizados
- 4).- Responsable

IMPORTANCIA

- 1).- Auditorías
- 2).- Diagnóstico de fallos
- 3).- Control de versiones

NORMATIVAS Y PROTOCOLOS ORGANIZACIONALES

Las organizaciones establecen políticas para regular:

- 1).- Instalación de software
- 2).- Actualización de sistemas
- 3).- Uso de licencias
- 4).- Seguridad informática

Teoría de caso

Empresa con políticas definidas:

Resultado:

- 1).- Mayor control
- 2).- Reducción de errores
- 3).- Cumplimiento legal

ESTRATEGIAS DE MANTENIMIENTO

Las estrategias de mantenimiento son el conjunto de acciones planificadas que se aplican a un sistema informático con el objetivo de garantizar su correcto funcionamiento, prolongar su vida útil y prevenir fallos.

MANTENIMIENTO PREVENTIVO

Consiste en realizar acciones antes de que ocurran fallos, para evitarlos.

Ejemplos

- 1).- Actualizar el sistema operativo
- 2).- Limpiar archivos innecesarios
- 3).- Revisar el estado del hardware

Teoría de caso

Un usuario actualiza periódicamente su sistema en Windows 11.

Resultado:

- 1).- Menos errores
- 2).- Mayor seguridad
- 3).- Mejor rendimiento

MANTENIMIENTO CORRECTIVO

Se aplica cuando ya existe un problema y se necesita solucionarlo.

Ejemplo

- 1).- Reparar errores del sistema
- 2).- Eliminar virus
- 3).- Reinstalar programas dañados

Teoría de caso

Una computadora comienza a fallar y mostrar errores.

El técnico:

- 1).- Detecta el problema
- 2).- Repara el sistema
- 3).- Restaura su funcionamiento

MANTENIMIENTO EVOLUTIVO

Consiste en mejorar el sistema para adaptarlo a nuevas necesidades.

Ejemplo

- 1).- Actualizar software
- 2).- Instalar nuevas aplicaciones
- 3).- Mejorar hardware

Teoría de caso

Una empresa actualiza sus equipos para soportar nuevas aplicaciones.

Resultado:

- 1).- Mayor productividad
- 2).- Mejor rendimiento

IMPORTANCIA DE LAS ESTRATEGIAS DE MANTENIMIENTO

Aplicar estas estrategias permite:

- 1).- Evitar fallos inesperados
- 2).- Prolongar la vida útil del equipo
- 3).- Mejorar el rendimiento
- 4).- Garantizar la seguridad
- 5).- Reducir costos a largo plazo

ENFOQUE PROFESIONAL

En entornos modernos, la gestión de actualizaciones debe ser:

- 1).- Planificada
- 2).- Controlada
- 3).- Documentada
- 4).- Evaluada constantemente

Este enfoque profesional transforma el mantenimiento en un proceso estratégico dentro de la administración de sistemas.

NORMATIVAS Y GESTIÓN ORGANIZACIONAL DEL SOFTWARE

En los entornos informáticos modernos, la gestión del software no puede limitarse a aspectos técnicos, sino que es necesario incorporar marcos normativos, políticas organizacionales y prácticas que aseguren el uso adecuado, seguro y eficiente de los recursos tecnológicos.

La gobernanza del software permite alinear la tecnología con los objetivos estratégicos de la organización, garantizando control, cumplimiento y sostenibilidad.

CONCEPTO DE NORMATIVAS EN TECNOLOGÍA

Las normativas en tecnológicas son conjuntos de reglas, estándares y lineamientos que regulan el uso, implementación y gestión de sistemas informáticos moderno dentro de una organización.

Estas pueden ser:

- 1).- **Internas:** políticas propias de la organización
- 2).- **Externas:** estándares internacionales o regulaciones legales

IMPORTANCIA DE LAS NORMATIVAS

La aplicación de normativas permite:

- 1).- Garantizar seguridad de la información
- 2).- Cumplir requisitos legales
- 3).- Estandarizar procesos
- 4).- Reducir riesgos operativos

Teoría de caso

Una organización sin políticas claras de software:

Resultado:

- 1).- Uso de software no autorizado
- 2).- Vulnerabilidades de seguridad
- 3).- Falta de control

Las normativas son esenciales para el orden tecnológico.

POLÍTICAS ORGANIZACIONALES DE SOFTWARE

Las políticas organizacionales de software son un conjunto de normas, lineamientos y reglas que establecen cómo debe ser gestionado el software dentro de una institución.

Estas políticas definen:

- 1).- Qué software se puede usar
- 2).- Cómo debe instalarse
- 3).- Cómo se mantiene
- 4).- Cómo se protege
- 5).- Bajo qué condiciones legales se utiliza

TIPOS DE POLÍTICAS

1).- POLÍTICA DE INSTALACIÓN DE SOFTWARE

Las políticas de instalaciones de software definen quién puede instalar software, cómo hacerlo y qué programas están permitidos.

Ejemplo

- 1).- Solo el área de sistemas puede instalar programas
- 2).- Se permite únicamente software autorizado

Teoría de caso

En una empresa con equipos con Windows 11

Los usuarios no pueden instalar programas sin autorización del administrador.

Resultado:

- 1).- Se evita software malicioso
- 2).- Se mantiene el control del sistema

2).- POLÍTICA DE ACTUALIZACIÓN Y MANTENIMIENTO

Establece cómo y cuándo deben actualizarse los sistemas y aplicaciones.

Ejemplo

- 1).- Actualizaciones automáticas programadas
- 2).- Revisiones periódicas del sistema

Teoría de caso

Una organización programa actualizaciones mensuales de sus sistemas.

Resultado:

- 1).- Mayor seguridad
- 2).- Menos fallos
- 3).- Mejor rendimiento

3).- POLÍTICA DE SEGURIDAD INFORMÁTICA

Define las medidas para proteger la información y los sistemas.

Ejemplos

- 1).- Uso de contraseñas seguras
- 2).- Instalación de antivirus
- 3).- Restricción de accesos

Teoría de caso

Una empresa implementa controles de acceso a archivos sensibles.

Resultado:

- 1).- Protección de datos
- 2).- Prevención de ataques
- 3).- Reducción de riesgos

4).- POLÍTICA DE LICENCIAMIENTO

Regula el uso legal del software dentro de la organización.

Ejemplo

- 1).- Uso de software con licencia válida
- 2).- Prohibición de programas piratas

Teoría de caso

Una empresa utiliza únicamente software con licencias oficiales.

Resultado:

- 1).- Cumplimiento legal
- 2).- Evita sanciones
- 3).- Garantiza soporte técnico

IMPORTANCIA DE LAS POLÍTICAS

Las políticas organizacionales permiten:

- 1).- Orden en el uso del software
- 2).- Seguridad de la información
- 3).- Cumplimiento legal
- 4).- Mejor gestión tecnológica
- 5).- Reducción de riesgos

PROTOCOLOS DE INSTALACIÓN Y MANTENIMIENTO

Los protocolos establecen procedimientos estandarizados para:

- 1).- Instalación de sistemas operativos
- 2).- Configuración de aplicaciones
- 3).- Actualización de software
- 4).- Mantenimiento preventivo

Ejemplo

Antes de instalar software:

- 1).- Verificar licencia
- 2).- Validar compatibilidad
- 3).- Autorizar instalación
- 4).- Documentar proceso

CUMPLIMIENTO NORMATIVO (COMPLIANCE)

El cumplimiento normativo implica adherirse a leyes y estándares relacionados con tecnología y datos.

Aspectos importantes que hay que seguir

- 1).- Protección de datos
- 2).- Seguridad de la información
- 3).- Uso legal del software

Teoría de caso

Empresa auditada:

Problema:

Uso de software sin licencia

Consecuencia:

- 1).- Multas
- 2).- Obligación de regularización

GESTIÓN ORGANIZACIONAL DEL SOFTWARE

La gestión del software implica:

- 1).- Planificación
- 2).- Implementación
- 3).- Control
- 4).- Evaluación

ROLES Y RESPONSABILIDADES

Para una correcta gestión, es necesario definir responsabilidades, tales como:

- 1).- Administrador de sistemas
- 2).- Equipo de soporte técnico
- 3).- Usuarios finales
- 4).- Responsables de seguridad

Teoría de caso

Falta de roles definidos:

Resultado:

- 1).- Confusión
- 2).- Errores operativos
- 3).- Falta de control

AUDITORÍA Y CONTROL

La auditoría y control en entornos tecnológicos informáticos modernos es el proceso mediante el cual se evalúa el uso, gestión y cumplimiento de las normativas relacionadas con el software y los sistemas informáticos dentro de una organización, es una revisión para comprobar si todo se está haciendo correctamente.

ENFOQUE CONCEPTUAL

Las auditorías no solo buscan errores, sino también:

- 1).- Verificar buenas prácticas
- 2).- Detectar riesgos
- 3).- Mejorar procesos
- 4).- Asegurar el cumplimiento de normas

Son una herramienta importante para mantener el orden y la confiabilidad en los sistemas.

TIPOS DE AUDITORÍA

1).- AUDITORÍA INTERNA

Es realizada por personal de la misma organización.

Ejemplo

El departamento de sistemas revisa:

- 1).- Uso del software
- 2).- Cumplimiento de políticas
- 3).- Estado de los equipos

Teoría de Caso

Una empresa revisa internamente sus equipos con Windows 11 para verificar que:

- 1).- No haya software no autorizado
- 2).- Todo esté correctamente configurado

2).- AUDITORÍA EXTERNA

Es realizada por entidades o profesionales externos a la organización.

Ejemplo

Una empresa contrata especialistas para evaluar:

- 1).- Seguridad informática
- 2).- Cumplimiento legal
- 3).- Uso de licencias

Teoría de caso

Una organización recibe una auditoría externa para verificar que todo su software sea legal.

Resultado:

- 1).- Validación independiente
- 2).- Mayor credibilidad

3).- AUDITORÍA TÉCNICA

Se enfoca en el análisis profundo del funcionamiento técnico de los sistemas.

Ejemplo

- 1).- Evaluación de rendimiento
- 2).- Revisión de configuraciones
- 3).- Análisis de vulnerabilidades

Teoría de Caso

Un equipo técnico analiza servidores para detectar fallos o debilidades.

Resultado:

- 1).- Mejora del sistema
- 2).- Reducción de riesgos

5.3. GESTIÓN DE RIESGOS EN ENTORNOS INFORMÁTICO MODERNOS

La gestión de riesgos es el proceso de identificar, analizar y controlar posibles amenazas que pueden afectar los sistemas informáticos de una organización.

5.3.1. TIPOS DE RIESGOS

Figura 45
Tipos de riesgos



1).- RIESGOS TÉCNICOS

Relacionados con fallos en hardware o software.

Ejemplo

- 1).- Fallo del disco duro
- 2).- Errores del sistema operativo
- 3).- Problemas de compatibilidad

Teoría de Caso

Un servidor falla por un error del sistema.

Consecuencia:

Interrupción del servicio

2).- RIESGOS OPERATIVOS

Derivan de errores humanos o procesos mal definidos.

Ejemplo

Eliminación accidental de archivos

- 1).- Mala configuración del sistema

Teoría de Caso

Un empleado borra información importante por error.

Consecuencia: Pérdida de datos

3).- RIESGOS LEGALES

Relacionados con el incumplimiento de leyes o licencias.

Ejemplo

- 1).- Uso de software ilegal
- 2).- Incumplimiento de normativas

Teoría de Caso

Una empresa usa software sin licencia.

Consecuencia: Sanciones legales

4).- RIESGOS DE SEGURIDAD

Amenazas que afectan la confidencialidad, integridad o disponibilidad de la información.

Ejemplo

- 1).- Virus
- 2).- Hackeos
- 3).- Accesos no autorizados

Teoría de Caso

Un sistema es atacado por malware.

Consecuencia: Robo o pérdida de información

CAPÍTULO VI

SISTEMAS EN RED

INTERCONECTIVIDAD, RECURSOS COMPARTIDOS Y GESTIÓN DISTRIBUIDA EN ENTORNOS INFORMÁTICOS



6.1. SISTEMAS EN RED

Los sistemas en red representan la evolución natural de los entornos informáticos, ya no se trata de equipos aislados, sino de estructuras interconectadas donde la información fluye, los recursos se comparten y las decisiones técnicas impactan a múltiples usuarios simultáneamente.

Comprender las redes no solo amplía el conocimiento técnico, sino que permite gestionar la tecnología de manera más eficiente, segura y estratégica.

Un sistema en red es la interconexión de dos o más equipos informáticos mediante medios físicos o inalámbricos, con el propósito de compartir información, recursos y servicios.

En este entorno, cada dispositivo puede operar de manera autónoma, pero al mismo tiempo forma parte de una estructura integrada que permite la comunicación y colaboración entre usuarios y sistemas.



Ejemplo

En una oficina:

- 1).- Varios computadores están conectados
- 2).- Comparten una impresora
- 3).- Acceden a archivos en común

Esto constituye un sistema en red funcional.

Teoría de Caso

Una institución educativa conecta sus laboratorios:

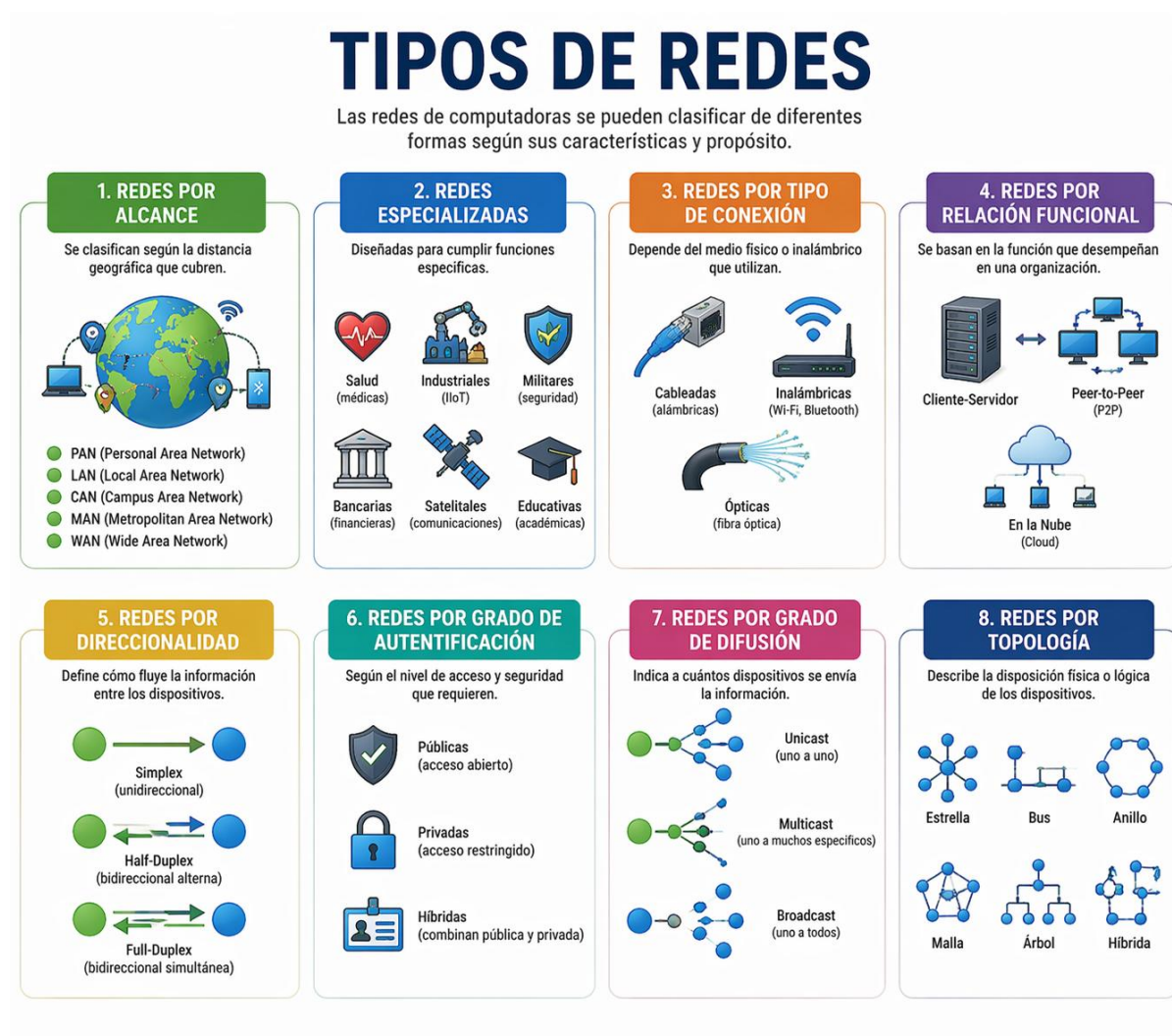
- 1).- Servidor central con contenidos académicos
- 2).- Estudiantes acceden desde diferentes equipos

Resultado

Optimización de recursos y acceso controlado a la información.

6.1.1. TIPOS DE REDES

Figura 47
 Tipos de redes



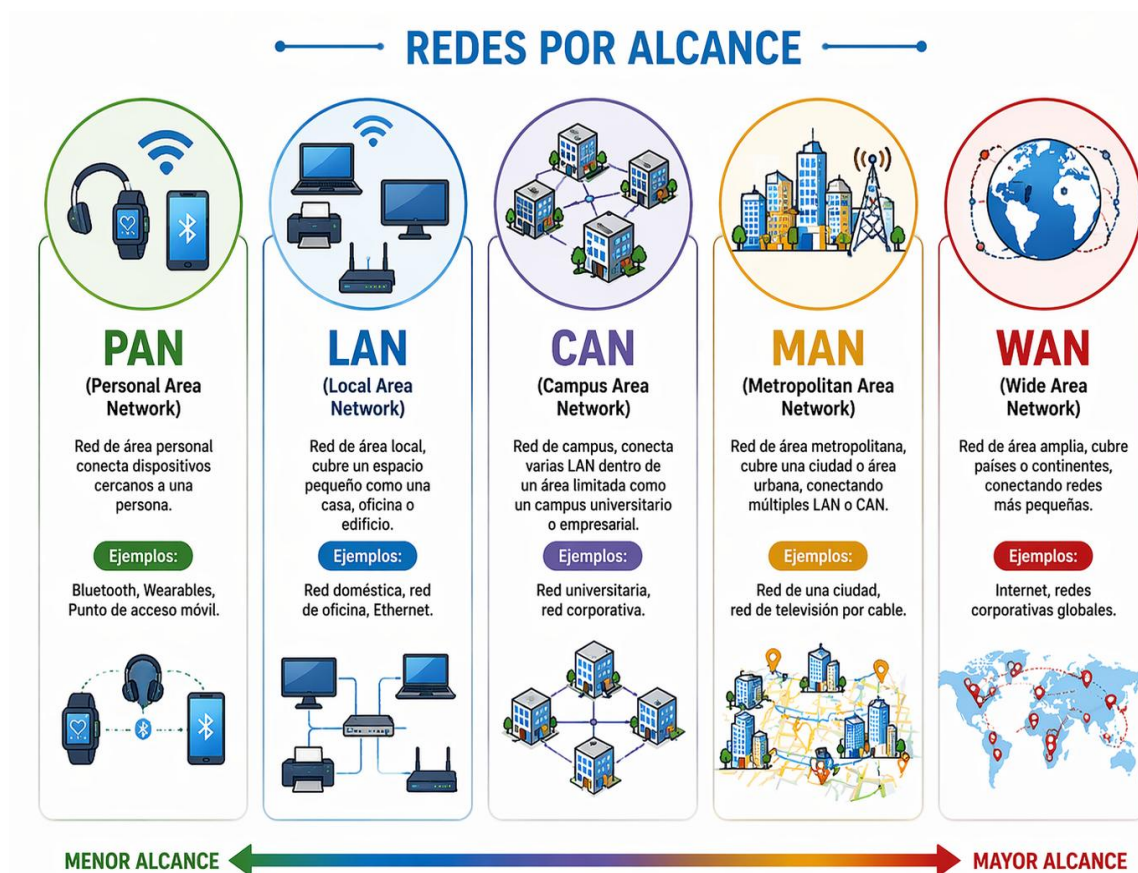
6.2. REDES POR ALCANCE

Las redes por alcance se clasifican según la extensión geográfica que cubren y el nivel de interconexión que permiten entre dispositivos.

Sin embargo, este criterio no solo implica distancia, sino también:

- 1).- Nivel de complejidad técnica
- 2).- Tipo de infraestructura requerida
- 3).- Cantidad de usuarios involucrados
- 4).- Forma de administración

Figura 48
Redes por alcance



PAN (Personal Area Network)

La PAN es la red más cercana al usuario, ya que representa el nivel más básico de conectividad, donde los dispositivos personales interactúan de forma directa e inmediata.

Es el primer nivel de integración tecnológica en la vida cotidiana.

Ejemplo

- 1).- Audífonos Bluetooth conectados al celular
- 2).- Smartwatch sincronizado
- 3).- Transferencia de archivos entre dispositivos cercanos

Teoría de caso

- 1).- Una persona utiliza:
- 2).- Celular
- 3).- Reloj inteligente
- 4).- Audífonos inalámbricos

Todos conectados entre sí.

Resultado:

- 1).- Experiencia integrada
- 2).- Automatización de tareas
- 3).- Mayor comodidad

LAN (Local Area Network)

La LAN es el núcleo de la conectividad en espacios reducidos, donde múltiples dispositivos comparten recursos dentro de un entorno controlado.

Es la base de la infraestructura tecnológica en hogares y empresas.

Ejemplo:

- 1).- Red de una casa con varios dispositivos conectados al router.
- 2).- Red doméstica con computadoras, celulares y Smart TV
- 3).- Oficina con impresoras y servidores compartidos

Teoría de caso

Una pequeña empresa implementa una LAN para:

- 1).- Compartir archivos
- 2).- Utilizar software interno
- 3).- Gestionar impresiones

Problema

si no se gestiona bien la red, causa los siguientes incidentes:

- 1).- Lentitud
- 2).- Conflictos de acceso
- 3).- Vulnerabilidades

CAN (Campus Area Network)

La CAN conecta múltiples redes LAN dentro de una organización más grande, como universidades o complejos empresariales, introduce una nueva dimensión que coordina múltiples redes locales como, por ejemplo:

- 1).- Red de una universidad
- 2).- Red de un parque tecnológico
- 3).- Infraestructura de una empresa grande con varios edificios

Teoría de caso

Una universidad conecta:

- 1).- Facultades
- 2).- Laboratorios
- 3).- Bibliotecas

Desafíos:

- 1).- Control de acceso por áreas
- 2).- Gestión del tráfico
- 3).- Estabilidad de conexión simultánea

Resultado: Acceso centralizado, pero con alta exigencia técnica.

MAN (Metropolitan Area Network)

La MAN cubre una ciudad o área metropolitana, conectando múltiples redes de gran tamaño, aquí la red deja de ser organizacional y se vuelve infraestructura urbana.

Ejemplo

- 1).- Red de servicios de internet en una ciudad.
- 2).- Redes de proveedores de internet
- 3).- Sistemas de videovigilancia urbana
- 4).- Redes de transporte inteligente

Teoría de caso

Una ciudad implementa una red para:

- 1).- Semáforos inteligentes
- 2).- Cámaras de seguridad
- 3).- Servicios públicos digitales

Problema: Si falla la red, afecta a miles de personas

Conclusión: Las MAN son críticas para el funcionamiento de ciudades modernas.

WAN (Wide Area Network)

La WAN es el nivel más alto de interconexión, donde redes de diferentes regiones o países se conectan entre sí.

Es la base de la comunicación global.

Ejemplo

- 1).- Internet
- 2).- Redes corporativas internacionales
- 3).- Sistemas bancarios globales

Teoría de caso

Una empresa multinacional conecta:

- 1).- Oficinas en diferentes países
- 2).- Servidores distribuidos
- 3).- Sistemas en la nube

Desafíos:

- 1).- Latencia
- 2).- Seguridad
- 3).- Dependencia de proveedores

Resultado: Operaciones globales en tiempo real.

6.3. REDES ESPECIALIZADAS

Las redes especializadas son estructuras diseñadas para cumplir funciones específicas dentro de un entorno informático moderno, a diferencia de las redes tradicionales que buscan conectividad general.

Estas redes no solo conectan dispositivos, sino que están orientadas a:

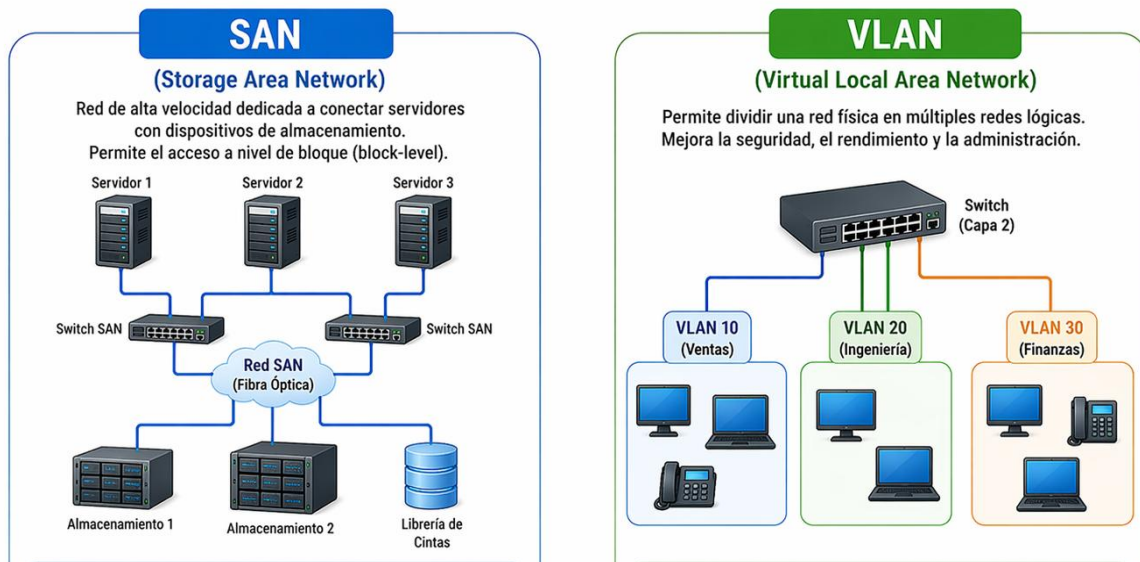
- 1).- Optimizar procesos concretos
- 2).- Mejorar el rendimiento en tareas específicas
- 3).- Incrementar la seguridad
- 4).- Gestionar recursos de forma más eficiente

En otras palabras: Son redes creadas con un propósito claro, donde la eficiencia y el control son prioritarios.

Figura 49
Redes especializadas

REDES ESPECIALIZADAS

Diseñadas para cumplir funciones específicas dentro de una infraestructura de TI.



SAN (Storage Area Network):

Una **SAN** es una red dedicada exclusivamente al almacenamiento de datos, diseñada para permitir que múltiples servidores accedan a grandes volúmenes de información de manera rápida, segura y centralizada.

A diferencia de una red común:

- 1).- No transporta todo tipo de datos
- 2).- Está enfocada únicamente en almacenamiento

Esto la convierte en una infraestructura crítica.

Ejemplo

Un centro de datos donde múltiples servidores acceden a discos de almacenamiento centralizado.

Ventaja: Alta velocidad y eficiencia en gestión de información.

Teoría de caso

Una empresa financiera utiliza SAN para:

- 1).- Almacenar bases de datos críticas
- 2).- Garantizar acceso rápido y seguro

VLAN (Virtual Local Area Network)

Una **VLAN** es una red lógica que permite dividir una red física en múltiples redes independientes, sin necesidad de cambiar la infraestructura física.

Esta red introduce una parte importante que la separación virtual dentro de una misma red.

Ejemplo

Separar la red de:

- 1).- Administración
- 2).- Estudiantes
- 3).- Invitados

Aunque usen el mismo cableado.

Ventaja: Mejora la seguridad y organización de la red.

Teoría de caso

En una empresa:

VLAN 1 → área administrativa

VLAN 2 → área técnica

Evita accesos no autorizados entre departamentos.

6.4. REDES POR TIPO DE CONEXIÓN

Las redes por tipo de conexión se clasifican según el medio físico o tecnológico utilizado para transmitir la información entre los dispositivos.

Este criterio no solo define cómo se conectan los equipos, sino también estos parámetros:

- 1).- La calidad de la comunicación
- 2).- La velocidad de transmisión
- 3).- La estabilidad de la red
- 4).- El nivel de seguridad

En otras palabras, el tipo de conexión determina cómo viajan los datos y qué tan confiable es ese viaje.

Figura 50
Redes por tipos de conexión

REDES POR TIPO DE CONEXIÓN

Se clasifican según el medio físico o inalámbrico que utilizan para transmitir los datos.



Ejemplo

Según el medio utilizado:

- 1).- **Cableadas:** usan cables (Ethernet)
- 2).- **Inalámbricas:** usan señales (Wi-Fi)

Teoría de caso

Una empresa migra de red cableada a inalámbrica:

Resultado:

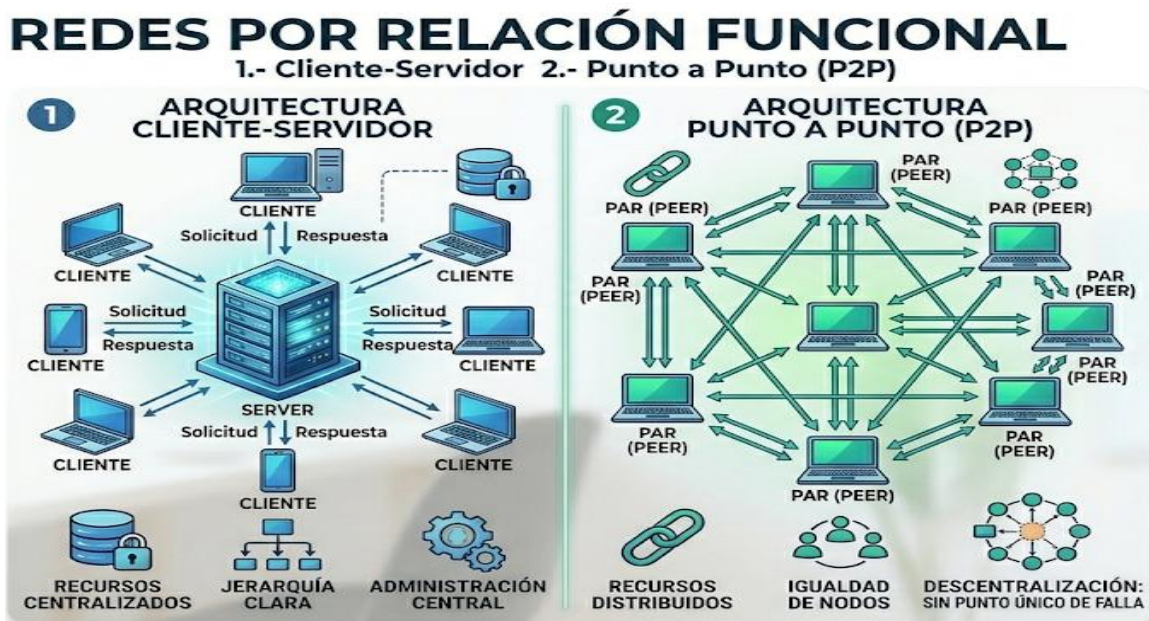
- 1).- Mayor movilidad
- 2).- Menor uso de infraestructura física

6.5. REDES POR RELACIÓN FUNCIONAL

Las redes por relación funcional se clasifican según la forma en que los dispositivos interactúan entre sí y comparten los recursos dentro de la red.

Esta clasificación no se basa en la distancia ni en el tipo de conexión, sino en algo más importante, en quién controla la red y cómo se distribuyen los recursos

Figura 51
Redes por relación funcional



Ejemplo

1).-Cliente-Servidor:

Un equipo central administra recursos

2).-Punto a punto(P2P):

Todos los equipos comparten recursos directamente

Ejemplo

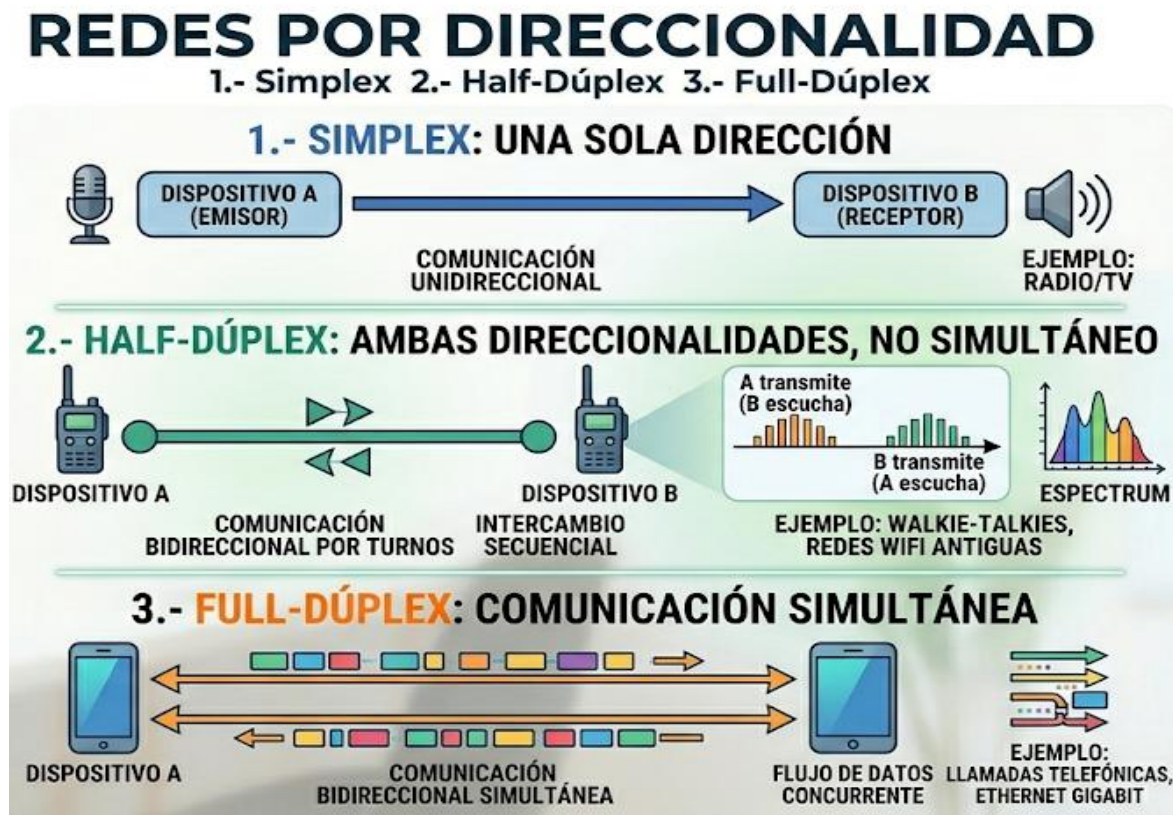
1).- Servidor de archivos → cliente-servidor

2).- Compartir archivos entre laptops → P2P

6.6. REDES POR DIRECCIONALIDAD

Las redes por direccionalidad se clasifican según la forma en que fluye la información entre los dispositivos dentro de un sistema de comunicación, no se trata de qué se transmite, sino de cómo se transmite.

Figura 52
Redes por direccionalidad



Ejemplo 1

- 1).- **Simplex**: una sola dirección
- 2).- **Half-Dúplex**: ambas direcciones, no simultáneo
- 3).- **Full-Dúplex**: comunicación simultánea

Ejemplo 2

- 1).- Radio → simplex
- 2).- Walkie-talkie (radio bidireccionales) → half-duplex
- 3).- Videollamada → full-dúplex

Teoría de caso

Un sistema de señalización digital en un aeropuerto:

- 1).- Envía información de vuelos

- 2).- Los usuarios solo reciben datos

Resultado:

- 1).- Comunicación masiva

- 2).- Sin interacción

6.7. REDES POR GRADO DE AUTENTIFICACIÓN

Las redes por grado de autenticación se clasifican según el nivel de control que existe para permitir el acceso de usuarios y dispositivos.

En esencia, responden a una pregunta importante:

¿Quién puede entrar a la red y bajo qué condiciones?

Ejemplo

- 1).- Redes abiertas son de acceso libre

Las redes abiertas son aquellas que permiten el acceso libre, sin necesidad de validación de identidad.

No necesitan:

- 1).- Usuario
- 2).- Contraseña
- 3).- Verificación

Figura 53
Redes por grado de autenticación



2).- Redes seguras son las que requieren autenticación

Las redes seguras requieren que el usuario se autentique antes de acceder, esto implica validar su identidad mediante:

- 1).- Contraseñas
- 2).- Usuarios
- 3).- Certificados
- 4).- Sistemas biométricos (en entornos avanzados)

Teoría de caso

1).- Wi-Fi público vs red corporativa:

Conclusión: La autenticación protege la información.

6.8. REDES POR GRADO DE DIFUSIÓN

Las redes por grado de difusión se clasifican según el nivel de alcance y disponibilidad de acceso a la red, es decir, quién puede usar la red y hasta dónde llega su cobertura de uso

Figura 54
Redes por grado de difusión



1).- **Privadas:** acceso restringido

Las redes privadas son aquellas cuyo acceso está restringido a un grupo definido de usuarios, solo pueden acceder:

- a). - Personas autorizadas
- b). - Dispositivos registrados
- c). - Usuarios con permisos

2).- **Públicas:** acceso general

Las redes públicas son aquellas diseñadas para ser accesibles a cualquier usuario, sin restricciones estrictas.

Ejemplo

- 1).- Red empresarial → privada
- 2).- Internet → pública

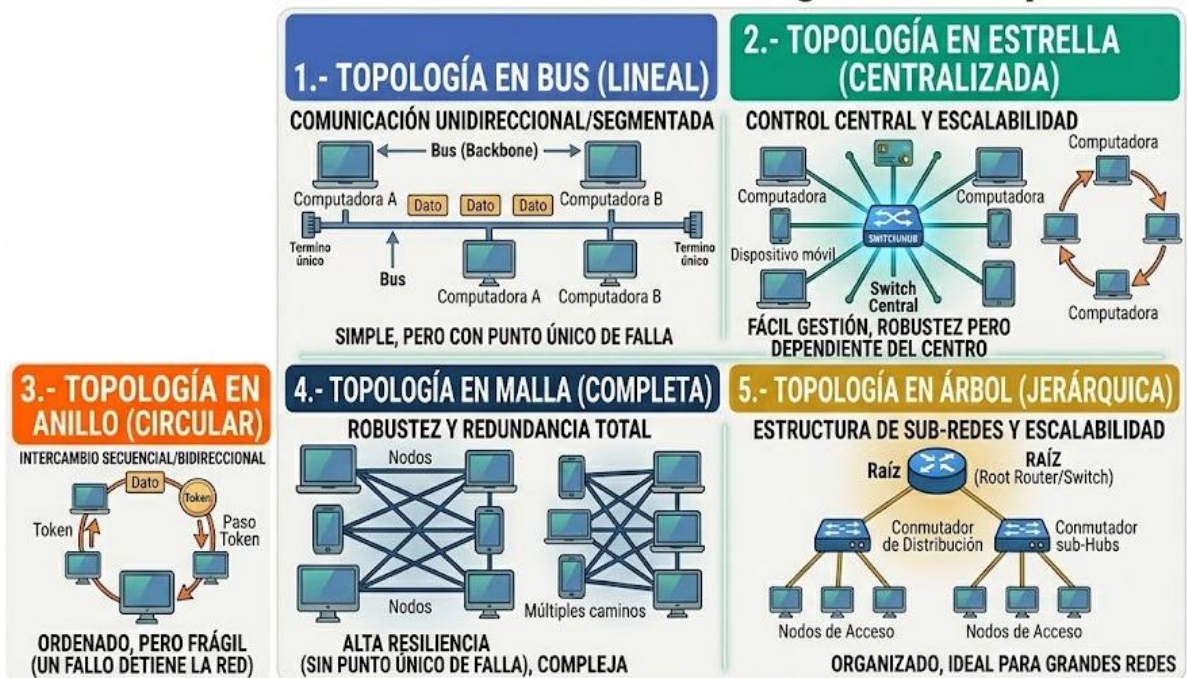
6.9. REDES POR TOPOLOGÍA

Las redes por topología se refieren a la forma en que los dispositivos están organizados e interconectados dentro de una red, no se trata solo de cómo están conectados físicamente, sino también de:

- 1).- Cómo circula la información
- 2).- Cómo se gestionan los fallos
- 3).- Qué tan eficiente es la comunicación

Figura 55
Redes por topología

REDES POR TOPOLOGÍA - Diagramas Comparativos



TOPOLOGÍA EN BUS

En esta topología, todos los dispositivos están conectados a un único canal de comunicación (cable principal), la información viaja por ese canal y todos los equipos la reciben.

Ejemplo:

Red antigua con un solo cable principal.

Limitación: Si el cable falla, toda la red se afecta.

Teoría de caso

Una pequeña empresa en los años 90 utilizaba una red en bus:

1).- Un solo cable conectaba todos los equipos

Problema: Cuando el cable fallaba, toda la red dejaba de funcionar

TOPOLOGÍA EN ESTRELLA

En esta topología, todos los dispositivos se conectan a un nodo central (como un switch o router), toda la comunicación pasa por ese punto central.

Ejemplo

1). - Red doméstica con router.

Ventaja: Fácil gestión y aislamiento de fallos.

Teoría de caso

Una oficina conecta todos sus equipos a un switch:

1).- Si una computadora falla → no afecta a las demás

2).- Si el switch falla → toda la red cae

Resultado: Mayor control, pero dependencia del nodo central

TOPOLOGÍA EN ANILLO

Los dispositivos se conectan formando un círculo cerrado, donde cada uno está conectado al siguiente, la información circula en un solo sentido (o en ambos en versiones modernas).

Ejemplo 1

Sistemas industriales antiguos.

Riesgo:

Una falla puede afectar toda la red.

Ejemplo 2

En una planta industrial:

1).- Los dispositivos estaban en anillo

Un fallo en un equipo interrumpía la comunicación

Resultado: Dependencia total de cada nodo

TOPOLOGÍA EN MALLA

En esta topología, todos los dispositivos están interconectados entre sí, cada nodo tiene múltiples rutas para comunicarse.

Ejemplo

Infraestructura crítica o servidores.

Ventaja: Alta redundancia y confiabilidad.

Teoría de Caso

Un sistema bancario implementa red en malla:

Si un enlace falla → los datos toman otra ruta

Resultado: Alta disponibilidad y continuidad del servicio

TOPOLOGÍA EN ÁRBOL

Es una estructura jerárquica que combina varias topologías (generalmente estrella) en forma de niveles, similar a un organigrama empresarial.

Ejemplo:

Red empresarial por departamentos.

Teoría de caso

Una empresa organiza su red así:

1).- **Nivel 1** → servidor principal

2).- **Nivel 2** → switches por departamento

3).- **Nivel 3** → equipos de usuarios

Resultado: Orden, control y escalabilidad

TOPOLOGÍA FÍSICA

La topología física se refiere a la disposición real y tangible de los elementos de una red, es decir, cómo están ubicados físicamente:

- 1).- Los dispositivos (computadoras, servidores, routers)
- 2).- Los medios de conexión (cables, antenas, switches)

Figura 56
Topología física



Ejemplo

Ubicación de routers y cableado.

Elementos que intervienen

Una topología física considera:

- 1).- Cableado (Ethernet, fibra óptica)
- 2).- Dispositivos de red (switches, routers)
- 3).- Puntos de acceso Wi-Fi
- 4).- Ubicación de usuarios

Teoría de caso

Una empresa diseña su red en un edificio de tres pisos:

- a). - **Piso 1** → área administrativa
- b). - **Piso 2** → área técnica
- c). - **Piso 3** → gerencia

Implementación física:

- 1).- Un switch principal en el cuarto de servidores
- 2).- Cableado estructurado hacia cada piso
- 3).- Puntos de red en cada oficina

Resultado

- 1).- Organización clara
- 2).- Fácil mantenimiento
- 3).- Mejor rendimiento

TOPOLOGÍA LÓGICA

La topología lógica se refiere a la forma en que circula la información dentro de una red, independientemente de cómo estén conectados físicamente los dispositivos, Es decir, describe el comportamiento de los datos, no la estructura visible de la red.

Ejemplo

- 1). - Cómo viajan los datos dentro de la red.

La topología lógica es el flujo invisible de la red, aunque no se vea, es lo que realmente determina:

- 1).- La eficiencia de la comunicación
- 2).- La velocidad de transmisión
- 3).- La organización del tráfico

Ejemplo

En una red doméstica:

- 1).- Todos los dispositivos están conectados al router (topología física en estrella), pero los datos pueden viajar según reglas internas del router, eso corresponde a la topología lógica.

Teoría de caso

Una empresa tiene una red física en estrella:

1).- Todos los equipos conectados a un switch

Figura 57
Topología lógica



Pero implementa VLAN:

- 1).- Área administrativa → red lógica separada
- 2).- Área técnica → otra red lógica

Aunque usan el mismo cableado, los datos no se mezclan

REFERENCIAS BIBLIOGRÁFICAS

- AXELOS. (2019). *ITIL foundation: IT service management*. AXELOS.
- Cisco Networking Academy. (s.f.). *Networking Academy courses*. <https://www.netacad.com>
- Cisco Systems. (2023). *Networking fundamentals*. Cisco Networking Academy.
- Dreyfus, P. (1962). *L'informatique*. Société d'Informatique Appliquée.
- Forouzan, B. A. (2013). *Data communications and networking* (5th ed.). McGraw-Hill Education.
- Hennessy, J. L., & Patterson, D. A. (2019). *Computer architecture: A quantitative approach* (6th ed.). Morgan Kaufmann.
- IEEE. (2020a). *IEEE standards for local and metropolitan area networks*. IEEE.
- IEEE. (s.f.). *IEEE 802 standards*. <https://www.ieee.org>
- IEEE Computer Society. (2020b). *Guide to the software engineering body of knowledge (SWEBOK)*. IEEE.
- IETF. (s.f.). *Request for Comments (RFC)*. <https://www.ietf.org>
- ISO. (s.f.). *Open systems interconnection (OSI) model*. <https://www.iso.org>
- ISO/IEC. (2018a). *ISO/IEC 27001: Information security management systems*. ISO.
- ISO/IEC. (2018b). *ISO/IEC 20000: IT service management*. ISO.
- ISO/IEC. (2018c). *ISO/IEC 7498-1: Open systems interconnection basic reference model*. ISO.
- Kurose, J. F., & Ross, K. W. (2017). *Computer networking: A top-down approach* (7th ed.). Pearson.
- Kurose, J. F., & Ross, K. W. (2021). *Computer networking: A top-down approach* (8th ed.). Pearson.
- Laudon, K. C., & Laudon, J. P. (2020). *Management information systems* (16th ed.). Pearson.
- Microsoft. (2023a). *Windows IT Pro documentation*. Microsoft Docs.
- Microsoft. (2023b). *NTFS technical reference*. Microsoft Docs.
- Microsoft. (2023c). *Windows update management guide*. Microsoft Docs.
- Microsoft. (2023d). *Windows server networking documentation*. Microsoft Docs.
- Microsoft Learn. (s.f.). *Documentación técnica de redes*. <https://learn.microsoft.com>
- Nemeth, E., Snyder, G., Hein, T., & Whaley, B. (2017). *UNIX and Linux system administration handbook* (5th ed.). Pearson.
- OECD. (2019). *Software licensing models and the digital economy*. OECD Publishing.
- Patterson, D. A., & Hennessy, J. L. (2017). *Computer organization and design* (5th ed.). Morgan Kaufmann.

- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.
- Red Hat. (2023). *System update and patch management*. Red Hat Documentation.
- Rosen, L. (2005). *Open source licensing*. Prentice Hall.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating system concepts* (10th ed.). Wiley.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- Stallings, W. (2018a). *Operating systems: Internals and design principles* (9th ed.). Pearson.
- Stallings, W. (2018b). *Computer organization and architecture* (11th ed.). Pearson.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems* (4th ed.). Pearson.
- Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks* (5th ed.). Pearson.
- Tanenbaum, A. S., & Wetherall, D. J. (2021). *Computer networks* (6th ed.). Pearson.
- The Linux Foundation. (2022a). *Linux system administration guide*.
- The Linux Foundation. (2022b). *Linux filesystem hierarchy and management*.
- The Linux Foundation. (2022c). *Open source licensing guide*.
- The Linux Foundation. (2022d). *Introduction to networking*.
- The Linux Foundation. (2023). *Introduction to Linux systems*.
- VMware. (2022). *Virtualization concepts and architecture*.
- Von Neumann, J. (1945). *First draft of a report on the EDVAC*.

GLOSARIO DEL ECOSISTEMA DE LOS SISTEMAS OPERATIVOS Y SUS APLICACIONES

A

Algoritmo: Conjunto ordenado de pasos para resolver un problema.

Ejemplo: Una receta de cocina es un algoritmo.

Arquitectura de computador: Estructura que define cómo funcionan los componentes del sistema.

Ejemplo: Un equipo con SSD y procesador moderno tiene mejor rendimiento.

Autenticación: Proceso de verificación de identidad en un sistema o red.

Ejemplo: Ingresar usuario y contraseña en una red Wi-Fi.

B

Backup (copia de seguridad): Duplicado de información para evitar pérdidas.

Ejemplo: Guardar archivos en la nube.

Bit: Unidad mínima de información (0 o 1).

Ejemplo: Todo archivo digital está formado por bits.

Bus (topología): Estructura de red donde todos los dispositivos comparten un canal.

Ejemplo: Redes antiguas con un solo cable principal.

C

CPU: Componente que ejecuta instrucciones.

Ejemplo: Procesa programas al abrirlos.

Compatibilidad: Capacidad de funcionar correctamente entre sistemas.

Ejemplo: Software que funciona en un sistema operativo específico.

Cliente-Servidor: Modelo donde un equipo central provee servicios a otros.

Ejemplo: Servidor que almacena archivos para varios usuarios.

CAN (Campus Area Network): Red que conecta múltiples edificios dentro de una institución.

Ejemplo: Red de una universidad.

D

Dato: Información sin procesar.

Ejemplo: Números sin analizar.

Disco duro: Dispositivo de almacenamiento permanente.

Ejemplo: Donde se guardan archivos.

E

Sistema Embebido: Sistema integrado en dispositivos específicos.

Ejemplo: Software en un electrodoméstico.

Extensión de archivo: Indica el tipo de archivo.

Ejemplo: .pdf, .jpg

Ethernet: Tecnología de red cableada.

Ejemplo: Conexión por cable en una oficina.

F

Firmware: Software integrado en hardware.

Ejemplo: Control interno de una impresora.

FAT: Sistema de archivos básico.

Ejemplo: USB formateado en FAT32.

Full-Duplex: Comunicación en ambas direcciones simultáneamente.

Ejemplo: Videollamadas.

G

Gigabyte: Unidad de almacenamiento.

Ejemplo: Disco de 500 GB.

Gestión de recursos: Administración del sistema.

Ejemplo: Uso de memoria por aplicaciones.

H

Hardware: Componentes físicos.

Ejemplo: Monitor, teclado.

Hipervisor: Permite ejecutar sistemas virtuales.

Ejemplo: VirtualBox.

Half-Duplex: Comunicación en ambas direcciones, pero no simultánea.

Ejemplo: Walkie-talkie.

I

Informática: Ciencia del procesamiento de información.

Ejemplo: Uso de computadoras en empresas.

Interfaz: Forma de interacción con el sistema.

Ejemplo: Pantalla de un sistema operativo.

Internet: Red global de comunicación.

Ejemplo: Acceso a páginas web.

J

Java: Lenguaje de programación.

Ejemplo: Aplicaciones móviles.

K

Kernel: Núcleo del sistema operativo.

Ejemplo: Controla hardware y procesos.

L

Licencia de software: Permiso de uso.

Ejemplo: Licencia de software comercial.

Linux: Sistema operativo libre.

Ejemplo: Ubuntu.

LAN: Red de área local.

Ejemplo: Red doméstica.

M

Memoria RAM: Memoria temporal.

Ejemplo: Uso al abrir programas.

Multitarea: Ejecución de varias tareas.

Ejemplo: Navegar y escuchar música.

MAN: Red de área metropolitana.

Ejemplo: Red de una ciudad.

N

NTFS: Sistema de archivos avanzado.

Ejemplo: Disco en Windows.

Nube: Almacenamiento en internet.

Ejemplo: Google Drive.

Nodo: Dispositivo conectado a una red.

Ejemplo: Computador dentro de una red.

O

Sistema operativo: Software que gestiona el sistema.

Ejemplo: Windows o Linux.

P

Partición: División del disco.

Ejemplo: Separar sistema y datos.

Procesador: Ejecuta cálculos.

Ejemplo: Intel o AMD.

PAN: Red de área personal.

Ejemplo: Bluetooth entre celular y audífonos.

Q

Query: Consulta de datos.

Ejemplo: Buscar información en base de datos.

R

Rendimiento: Eficiencia del sistema.

Ejemplo: Rapidez de un equipo.

Repositorio: Almacenamiento de software.

Ejemplo: Repositorios Linux.

Router: Dispositivo que conecta redes.

Ejemplo: Router de casa.

S

Software: Programas del sistema.

Ejemplo: Navegador web.

Sistema de archivos: Organiza datos.

Ejemplo: NTFS.

SAN: Red dedicada al almacenamiento.

Ejemplo: Centros de datos empresariales.

Simplex: Comunicación en una sola dirección.

Ejemplo: Radio.

T

Tecnología de la información: Uso de tecnología para gestionar datos.

Ejemplo: Sistemas empresariales.

Trazabilidad: Seguimiento de cambios.

Ejemplo: Registro de actualizaciones.

Topología de red: Forma de conexión de los dispositivos.

Ejemplo: Red en estrella.

U

Usuario: Persona que usa el sistema.

Ejemplo: Estudiante con laptop.

USB: Dispositivo portátil.

Ejemplo: Pendrive.

V

Virtualización: Creación de sistemas virtuales.

Ejemplo: Máquina virtual.

Versión de software: Edición de un programa.

Ejemplo: Windows 10.

VLAN: Red virtual dentro de una red física.

Ejemplo: Separar departamentos en una empresa.

W

Windows: Sistema operativo.

Ejemplo: PC personal.

WAN: Red de gran alcance.

Ejemplo: Internet.

X

XFS: Sistema de archivos.

Ejemplo: Servidores Linux.

Y

Yottabyte: Unidad de almacenamiento.

Ejemplo: Centros de datos masivos.

Z

Zona de almacenamiento: Espacio asignado para datos.

Ejemplo: Carpeta de documentos.

Decálogo del Ecosistema de los Sistemas Operativos y sus Aplicaciones

1.-La tecnología no se utiliza, se comprende: Un sistema operativo no es solo una herramienta que se ejecuta, sino una estructura que se interpreta. Comprender cómo funciona permite tomar decisiones correctas, prevenir errores y optimizar recursos.

2.-Todo sistema es tan eficiente como su entorno lo permita: El rendimiento no depende únicamente del software o del hardware de forma aislada, sino de la relación entre ambos y su adecuación a las necesidades del usuario.

3.-La compatibilidad no es una opción, es una condición: La integración entre componentes es esencial. Un sistema que no es compatible con su entorno está destinado a fallar, sin importar su nivel tecnológico.

4.-Instalar no es suficiente, implementar es estratégico: La instalación de sistemas y aplicaciones debe responder a un análisis previo. Implementar implica planificar, evaluar y adaptar la tecnología al contexto real.

5.-Actualizar es proteger, no solo mejorar: Las actualizaciones no deben verse como simples mejoras funcionales, sino como mecanismos de seguridad y estabilidad que garantizan la continuidad del sistema.

6.-La información es el activo, el sistema es el medio: El verdadero valor no está en el software ni en el hardware, sino en la información que se gestiona.

7.-Sin control de cambios, no hay estabilidad: Modificar un sistema sin control genera incertidumbre. La trazabilidad y la gestión estructurada de cambios son esenciales para mantener la confiabilidad.

8.-La seguridad no es un complemento, es una base: Un sistema funcional sin seguridad es un sistema vulnerable. La protección debe integrarse desde el diseño, no añadirse después.

9.-La tecnología sin normativa es desorden: El uso de sistemas informáticos debe estar regulado por políticas, estándares y procedimientos que garanticen coherencia, legalidad y control organizacional.

10.-El conocimiento transforma al usuario en gestor: El mayor cambio no ocurre en la tecnología, sino en quien la utiliza. Comprender el ecosistema permite pasar de ejecutar acciones a tomar decisiones informadas.

ACERCA DEL AUTOR



Franklin Marcelo Alvarado Canales es un educador, jurista y profesional ecuatoriano, nacido en la aguerrida ciudad de Guayaquil. Con más de 30 años de ejercicio profesional en la docencia, ha dedicado su vida a inspirar a través de la palabra, el ejemplo y el fomento de los valores humanos y familiares.

El autor posee una robusta formación académica que incluye el título de Máster en Educación con énfasis en Administración y Gestión Educativa, además de ser Abogado de los Tribunales y Juzgados de la República, Licenciado en Ciencias de la Educación, Profesor de Segunda Enseñanza, Tecnólogo en Pedagogía y Técnico Superior en Análisis de Sistemas. Cuenta, asimismo, con el título honorífico de Doctor Honoris Causa.

A lo largo de su trayectoria, ha desempeñado múltiples y variados cargos en el sector académico, público y privado. Fue asesor en la Dirección Provincial de Educación del Guayas, y ejerció como Coordinador Académico de Nivelación y Admisión y Profesor del Instituto Tecnológico Superior Guayaquil. Además, fungió como Secretario General de los Institutos Tecnológicos Superiores Guayaquil, Tungurahua y Rita Lecumberry, y se desempeñó como Director Académico del Instituto Aiken.

En el ámbito secundario, fue Miembro del Consejo Directivo y del Consejo Ejecutivo del Colegio Emblemático Nacional Guayaquil, donde actualmente es profesor titular. También dirigió el Proyecto de Reforzamiento de la Educación Técnica (RETEC).

El Dr. Alvarado Canales compartió sus conocimientos como profesor en los colegios Eloy Alfaro de Durán, Ati II Pillahuaso, Batalla de Tarqui, Adolfo H. Simmons, Augusto Mendoza Moreira y Vicente León. En la educación superior, fue profesor de la Universidad de Guayaquil, institución en la cual, durante su época estudiantil, presidió el Comité de Derechos Humanos de Profesionales y Estudiantes y fue Vocal Principal del Directorio de la Asociación Escuela de Derecho.

En el contexto político, terció como candidato a la Asamblea Constituyente (2007) y participó en la candidatura para alcalde de la Ciudad de Guayaquil (2009).

Por su servicio cívico, ha recibido varias condecoraciones y reconocimientos nacionales e internacionales, destacándose la Medalla del Bicentenario, entregada por la M.I. Municipalidad de Guayaquil. Esta medalla es un símbolo de Honor y Orgullo Cívico asociado a un hito histórico de gran trascendencia, que se entrega cada 50 años y deja grabado el nombre en el libro de historia de la ciudad.

SINOPSIS

En un mundo donde la gestión de entorno informáticos modernos define la forma en que trabajamos, aprendemos y nos relacionamos, comprender los sistemas operativos y sus aplicaciones ya no es una opción, sino una necesidad fundamental, sin embargo, más allá del uso cotidiano de dispositivos, existe una realidad menos visible pero profundamente determinante que es un ecosistema tecnológico complejo, dinámico y en constante evolución.

“El Ecosistema de los Sistemas Operativos y sus Aplicaciones” es una obra que invita al lector a ir más allá de la superficie. No se limita a explicar conceptos técnicos, sino que propone una comprensión integral de cómo interactúan el hardware, el software, los usuarios y las organizaciones en entornos reales.

A lo largo de sus capítulos, el libro construye un recorrido progresivo que inicia en los fundamentos de la informática y la arquitectura de los sistemas, y avanza hacia aspectos clave como la compatibilidad, el análisis de requerimientos, la instalación de sistemas operativos, la gestión del almacenamiento, el mantenimiento, el control de cambios y el cumplimiento de normativas.

Lo que distingue a esta obra no es solo su contenido técnico, sino su enfoque práctico y estratégico. Cada tema está desarrollado con una visión aplicada, acompañada de ejemplos reales y situaciones que reflejan los desafíos actuales en la gestión de entornos informáticos. El lector no solo adquiere conocimientos, sino también criterios para tomar decisiones informadas, reducir riesgos y optimizar recursos tecnológicos.

En un contexto donde muchas implementaciones tecnológicas fallan no por falta de herramientas, sino por una gestión inadecuada, este libro se convierte en una guía esencial. Permite entender que la tecnología no funciona de manera aislada, sino dentro de un sistema donde cada elemento influye en el resultado final.

Además, integra aspectos frecuentemente ignorados en la formación técnica, como el impacto del licenciamiento del software, la importancia de la trazabilidad, la gestión de actualizaciones y el cumplimiento de estándares organizacionales. Estos elementos aportan una visión más completa, alineada con las exigencias del entorno profesional actual.

Esta obra está dirigida a estudiantes, profesionales, técnicos y cualquier persona interesada en comprender y gestionar sistemas informáticos con criterio, responsabilidad y visión estratégica. No requiere conocimientos avanzados previos, pero sí una disposición a entender la tecnología más allá de su uso superficial.