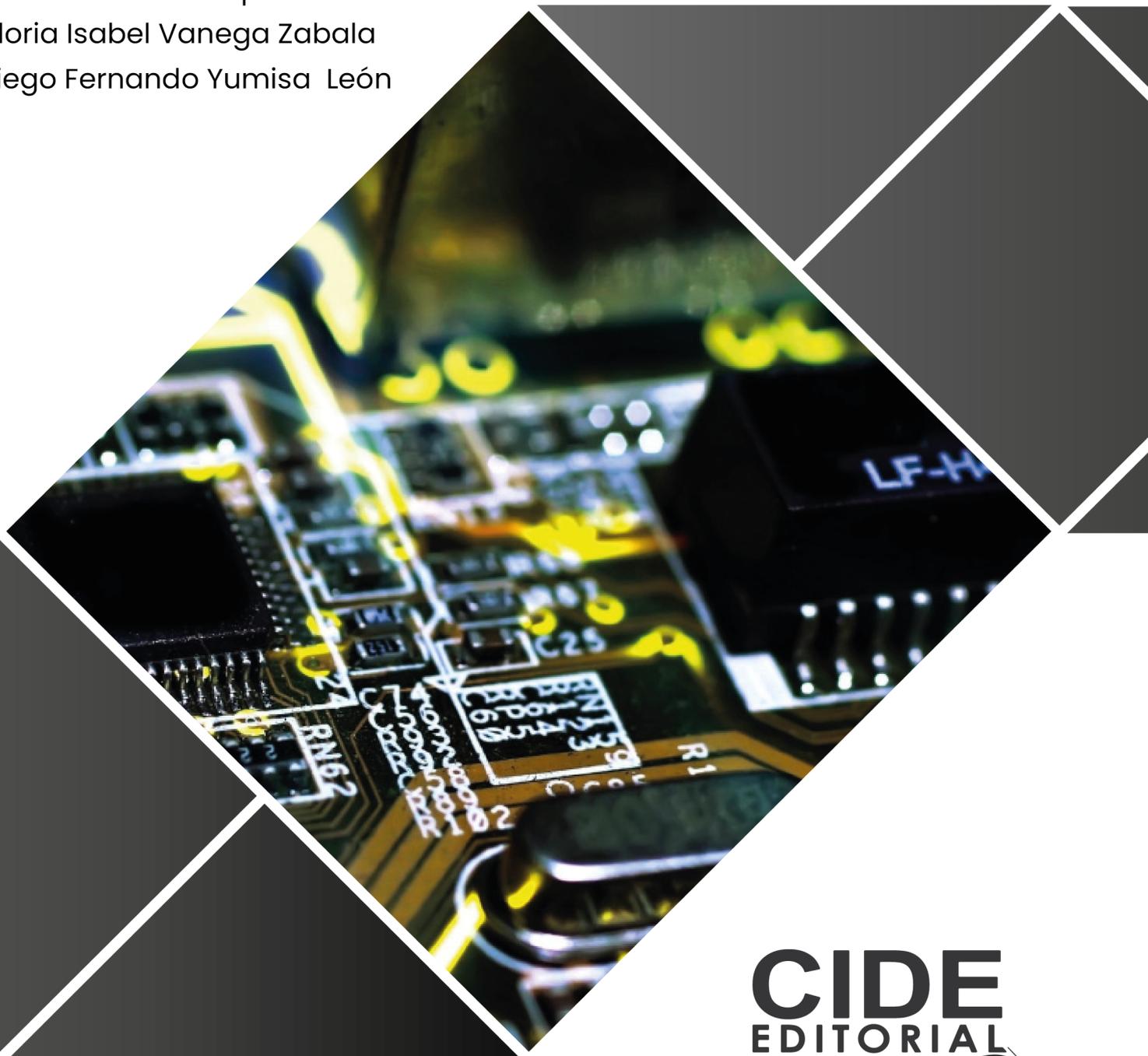


# ARQUITECTURA

# DE COMPUTADORAS

David Patricio López Carrillo  
Angel Marcelo Acuña Félix  
Rolando Fabián Tipán Tisalema  
Gloria Isabel Vanega Zabala  
Diego Fernando Yumisa León



**CIDE**  
EDITORIAL



# ARQUITECTURA DE COMPUTADORAS

# ARQUITECTURA DE COMPUTADORAS

## **Autores:**

David Patricio López Carrillo  
Ángel Marcelo Acuña Félix  
Rolando Fabián Tipán Tisalema  
Gloria Isabel Vanega Zabala  
Diego Fernando Yumisa León

## Arquitectura de computadoras

Reservados todos los derechos. Está prohibido, bajo las sanciones penales y el resarcimiento civil previstos en las leyes, reproducir, registrar o transmitir esta publicación, íntegra o parcialmente, por cualquier sistema de recuperación y por cualquier medio, sea mecánico, electrónico, magnético, electroóptico, por fotocopia o por cualquiera otro, sin la autorización previa por escrito al Centro de Investigación y Desarrollo Ecuador (CIDE).

Copyright © 2025

Centro de Investigación y Desarrollo Ecuador

Tel.: + (593) 04 2037524

[http. :/www.cidecuador.org](http://www.cidecuador.org)

ISBN: 978-9942-679-37-6

[doi.org/10.33996/cide.ecuador.AC2679376](https://doi.org/10.33996/cide.ecuador.AC2679376)

**Dirección editorial:** Lic. Pedro Misacc Naranjo, Msc.

**Coordinación técnica:** Lic. María J. Delgado

**Diseño gráfico:** Lic. Danissa Colmenares

**Diagramación:** Lic. Alba Gil

**Fecha de publicación:** febrero, 2025



Guayaquil – Ecuador

La presente obra fue evaluada por pares académicos  
experimentados en el área

### **Catalogación en la Fuente**

Arquitectura de computadoras / David Patricio López Carrillo, Ángel Marcelo Acuña Félix, Rolando Fabián Tipán Tisalema, Gloria Isabel Vanega Zabala, Diego Fernando Yumisa León. – Ecuador: Editorial CIDE, 2025.

171 p.: incluye tablas, figuras; 21,6 x 29,7 cm.

ISBN: 978-9942-679-37-6

1. Computación

# *Dedicatoria*



La presente obra se dedica a toda la comunidad educativa del Instituto Superior Bolívar de la ciudad de Ambato

# *Agradecimiento*



Los autores expresamos el agradecimiento a la Editorial CIDE por permitir la publicación del presente trabajo.

## *Semblanza de los autores*



David Patricio López Carrillo

<https://orcid.org/0000-0002-0398-8881>

[haslotuyo@hotmail.com](mailto:haslotuyo@hotmail.com)

Ingeniero en Electrónica y Computación, además de magíster en Ciencias de la Educación por la Pontificia Universidad Católica del Ecuador. Su trayectoria profesional abarca más de una década de experiencia en el ámbito de las telecomunicaciones, educación superior y tecnología. Ha trabajado en importantes empresas y centros educativos, desempeñándose como docente, coordinador de carrera y jefe técnico en proyectos relacionados con redes, fibra óptica y telecomunicaciones. Su pasión por la educación lo ha llevado a dictar cátedras en áreas como electrónica, electricidad y telecomunicaciones, contribuyendo a la formación de futuros profesionales en el Instituto Superior Tecnológico Bolívar y otras instituciones. También ha participado en múltiples jornadas y seminarios sobre metodología de investigación, gestión de proyectos, inteligencia artificial y docencia universitaria. Con una sólida formación en telecomunicaciones y docencia, David Patricio López Carrillo combina su experiencia técnica con su vocación por la enseñanza, destacándose como un profesional comprometido con la innovación y el desarrollo académico.



Ángel Marcelo Acuña Félix

<https://orcid.org/0000-0002-0844-877X>

[amarcelo.af@gmail.com](mailto:amarcelo.af@gmail.com)

Profesional destacado en el ámbito académico y laboral. Ingeniero en Sistemas y Magíster en Informática Educativa, posee una sólida formación que respalda su labor en el campo de la informática educativa y las Ciencias de la Educación. Con 14 años de experiencia en instituciones de educación superior, ha desempeñado diversos roles dentro de la academia. Su trayectoria como docente ha sido fundamental en la formación de generaciones de estudiantes en el área tecnológica, brindando su conocimiento y compromiso con la educación de calidad. Además de su labor como educador, el Ing. Acuña ha ejercido importantes funciones de liderazgo, entre las que destacan su rol como Representante docente al órgano colegiado superior, Coordinador de carrera, Gestor de campus, Gestor de Titulación y presidente del Consejo Electoral Institucional. A lo largo de su carrera, ha sido reconocido por su dedicación, puntualidad, y por su capacidad de liderazgo y gestión en diferentes áreas de la institución. Su enfoque en la mejora continua y la innovación educativa le ha permitido contribuir al desarrollo y consolidación de procesos académicos eficientes y efectivos.



## Rolando Fabián Tipan Tisalema

[rolo\\_762@hotmail.com](mailto:rolo_762@hotmail.com)

<https://orcid.org/0000-0002-4723-6446>

Ingeniero en Electrónica Control y Redes Industriales graduado en la ESPOCH, al momento me encuentro cursando la Maestría en Energías Renovables en la Universidad Internacional del Ecuador, Experiencia en el campo de docencia 7 años en instituciones públicas y privadas. Amplio mi formación profesional en diferentes áreas como Electrónica, Robótica, Redes, Domótica, Automatización, Sistemas de seguridad Guiados por IA, e realizado publicaciones en aérea de tecnología, fui coordinador de Carrera de Redes y Telecomunicaciones del IST. Bolívar de la ciudad de Ambato.



## Gloria Isabel Vanegas Zabala

[ivanegas@institutos.gob.ec](mailto:ivanegas@institutos.gob.ec)

<https://orcid.org/0000-0003-0490-996X>

Destacada profesional con una sólida trayectoria en los ámbitos académico e investigativo. Es Doctora en Automática, Robótica e Informática Industrial, con mención *Cum Laude* por la Universitat Politècnica de València. Posee una

Maestría en Robótica y Automatización por la Universidad Carlos III de Madrid. Además, es Ingeniera en Electrónica y Computación y Tecnóloga en Informática Aplicada por la Escuela Superior Politécnica de Chimborazo. Su formación académica se complementa con un Diplomado en Docencia Universitaria por la Universidad Estatal de Bolívar y la certificación en Formación de Formadores, consolidando así su experiencia en Electrónica, Automática y Ciencias de la Computación. En el ámbito académico, ha ejercido la docencia en diversas instituciones de prestigio, entre ellas la Escuela Superior Politécnica de Chimborazo (ESPOCH - Riobamba), donde se desempeñó como Coordinadora de la Maestría en Sistemas de Control y Automatización Industrial, y la Universidad Internacional del Ecuador (UIDE - Quito). Actualmente, es Coordinadora de la Carrera de Redes y Telecomunicaciones en el Instituto Superior Tecnológico Bolívar (ISTB - Ambato). Su producción intelectual incluye artículos en revistas indexadas, capítulos de libros y ponencias en congresos nacionales e internacionales, consolidándola como un referente en su disciplina. Ha participado en proyectos de investigación tanto a nivel nacional como internacional, entre ellos el Proyecto DPI2016-81002-R, *Control Avanzado y Aprendizaje de Robots en Operaciones de Transporte*, en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación de España, desarrollado en la Universitat Politècnica de València. También ha trabajado en el proyecto *Algoritmos de Planificación de Trayectoria Suave 3D para Vehículos Aéreos no Tripulados (UAVs) de Ala Fija mediante Metodologías Heurísticas*, en colaboración con la Universidad Nacional de Chimborazo. A lo largo de su carrera, ha sido reconocida con múltiples méritos académicos y profesionales, destacándose su Doctorado *Cum Laude*. Su compromiso con la excelencia y la innovación ha contribuido significativamente al avance del conocimiento en su campo.



Diego Fernando Yumisa León

[diefer2266@gmail.com](mailto:diefer2266@gmail.com)

<https://orcid.org/0000-0002-4515-1019>

Destacado profesional con una amplia trayectoria en el ámbito académico y Técnico. Es Ingeniero en Electrónica y Computación, de la Escuela Superior Politécnica de Chimborazo del Ecuador, egresado de la maestría en Tecnologías para la Gestión y Práctica Docente, títulos que consolidan su formación en el área de Tecnologías, también estudio el CCNA como Instructor CISCO en la Universidad San Francisco de Quito, extensión Riobamba y se actualizo como Instructor cisco en Instituto Superior Tecnológico Bolívar, lo cual fortalecieron sus conocimientos en Redes de Sistemas Informáticos. En el ámbito técnico a ejercido por 5 años y medio en el Instituto Nacional de Patrimonio Cultural como director del área de Sistemas Técnico Zonal 3, en el área de TIC's Tecnologías de Información y Comunicación donde desempeño con éxito el cargo donde consolido su experiencia en Redes de Sistemas Informáticos y otras áreas afines. En el ámbito académico a trabajado en el Colegio Nacional Juan de Velasco de la ciudad de Riobamba en el área de Informática, también trabajo como Docente en el Instituto Tecnológico Superior Juan de Velasco en la Carrera de Ensamblaje, Mantenimiento y Reparación de Computadores por el lapso de 3 años, En el Instituto

Superior Tecnológico Luis A. Martínez desempeño cargos como Coordinador Académico, Líder de Infraestructura, más tarde en Instituto Superior Tecnológico Bolívar, fue Líder de Titulación, Gestor de Titulación y actualmente desempeña el cargo de Líder de Admisiones. En lo intelectual incluye Artículos Científicos en revistas indexadas, Ponencias en Universidades, participación en Congresos Nacionales, participación en Concursos de Robótica y Electrónica, consolidando su experiencia profesional en lo Técnico y en lo Académico por más de 15 años de experiencia.

|                               |   |
|-------------------------------|---|
| Dedicatoria.....              | 5 |
| Agradecimiento.....           | 6 |
| Semblanza de los autores..... | 7 |

## Capítulo 1 Arquitectura de Computador

|   |    |
|---|----|
| 1.1. Introducción.....                                      | 17 |
| 1.2. Conceptos básicos.....                                 | 18 |
| 1.2.1 Definición de arquitectura de computador.....         | 18 |
| 1.2.2 Historia y evolución.....                             | 19 |
| 1.2.3 Principios fundamentales.....                         | 22 |
| 1.3. Arquitectura de un computador.....                     | 25 |
| 1.3.1 Estructura general.....                               | 25 |
| 1.3.2. Modelo de Von Neumann.....                           | 27 |
| 1.3.3. Modelo de Harvard.....                               | 28 |
| 1.4. Componentes de la arquitectura de un computador.....   | 29 |
| 1.4.1 Unidad Central de Procesamiento (CPU).....            | 29 |
| 1.4.2. Memoria.....   | 31 |
| 1.4.3. Dispositivos de entrada y salida.....                | 32 |
| 1.4.4 Bus de datos.....                                     | 33 |
| 1.4.5 Placa Base (Motherboard).....                         | 35 |
| 1.5. Representación de la información en un computador..... | 36 |
| 1.5.1 Representación binaria.....                           | 36 |
| 1.5.2 Codificación de datos.....                            | 38 |
| 1.5.3 Sistema de numeración.....                            | 40 |
| 1.5.4 Operaciones con datos.....                            | 41 |
| 1.6. Aplicaciones y tendencias futuras.....                 | 42 |
| 1.6.1. Evolución tecnológica.....                           | 43 |

## Capítulo 2 Unidad Aritmético-Lógica (ALU) y Unidad de Control

|  |    |
|--|----|
| 2.1. Unidad Aritmético-Lógica (ALU) y Unidad de Control..... | 54 |
|--|----|

## Capítulo 3 Introducción a los Sistemas Operativos

|  |    |
|--|----|
| 3.1 Definición y propósito de un sistema operativo.....      | 75 |
| 3.2 Historia y evolución de los sistemas operativos.....     | 76 |
| 3.3 Funciones principales de un sistema operativo.....       | 78 |
| 3.2. Arquitectura de los sistemas operativos.....            | 80 |
| 3.2.1 Componentes de un sistema operativo.....               | 80 |
| 3.2.2 Estructura de un sistema operativo.....                | 82 |
| 3.2.3 Ejemplos de sistemas operativos.....                   | 85 |
| 3.3. Gestión de procesos.....                                | 87 |
| 3.3.1 Concepto de proceso y subprocesso (Hilos).....         | 87 |
| 3.3.2 Estados de un proceso.....                             | 88 |
| 3.3.3 Planificación de procesos: Algoritmos y políticas..... | 89 |
| 3.3.4 Sincronización y comunicación entre procesos.....      | 91 |
| 3.4. Gestión de memoria.....                                 | 93 |

|  |     |
|--|-----|
| 3.4.1 Memoria primaria y secundaria.....                           | 93  |
| 3.4.2 Paginación y segmentación.....                               | 94  |
| 3.4.3 Memoria virtual.....   | 96  |
| 3.4.4 Algoritmos de gestión de memoria.....                        | 97  |
| 3.5. Gestión de almacenamiento.....                                | 99  |
| 3.5.1 Sistemas de archivos.....                                    | 99  |
| 3.5.2 Estructuras de almacenamiento y acceso a datos.....          | 101 |
| 3.5.3 Seguridad y protección de datos.....                         | 103 |
| 3.5.4 Gestión de dispositivos de almacenamiento.....               | 104 |
| 3.6. Gestión de entrada/salida (E/S).....                          | 106 |
| 3.6.1 Controladores de dispositivos.....                           | 106 |
| 3.6.2 Interrupciones y DMA.....                                    | 107 |
| 3.6.3 Optimización de la E/S.....                                  | 108 |
| 3.7. Gestión de recursos.....                                      | 112 |
| 3.7.1 Recursos del sistema: CPU, Memoria, dispositivos de E/S..... | 112 |
| 3.7.2 Asignación y liberación de recursos.....                     | 113 |
| 3.7.3 Prevención y manejo de Deadlocks (Interbloqueos).....        | 115 |

**Capítulo 4**  
**Dispositivos de un sistema informático**

|  |     |
|--|-----|
| 4.1. Conceptos básicos.....                | 119 |
| 4.2. Dispositivos – Hardware.....          | 120 |
| 4.2.1. Dispositivos de entrada.....        | 120 |
| 4.2.2. Dispositivos de salida.....         | 121 |
| 4.2.3. Dispositivos de almacenamiento..... | 121 |
| 4.2.4. Dispositivos de procesamiento.....  | 122 |
| 4.3. Dispositivos de red.....              | 122 |
| 4.3.1. Ensamblaje de computadores.....     | 123 |
| 4.3.1.1. Planificación.....                | 123 |
| 4.3.1.2. Montaje del hardware.....         | 123 |
| 4.3.1.3. Conexión de periféricos.....      | 123 |
| 4.3.1.4. Verificación y pruebas.....       | 124 |
| 4.4.2. Mantenimiento correctivo.....       | 129 |
| 4.4.3. Mantenimiento predictivo.....       | 129 |

**Capítulo 5**  
**Redes y comunicaciones**

|   |     |
|---|-----|
| 5.1. Definición y conceptos básicos.....                              | 132 |
| 5.1.2 Evolución histórica de las redes y comunicaciones.....          | 133 |
| 5.1.3 Importancia en el mundo moderno.....                            | 134 |
| 5.2 Fundamentos de las redes.....                                     | 134 |
| 5.2.1 Tipos de redes (LAN, WAN, MAN, PAN) .....                       | 134 |
| 5.2.2 Topologías de red.....  | 135 |
| 5.2.3 Modelos de referencia (OSI y TCP/IP).....                       | 136 |
| 5.3 Medios de transmisión.....  | 137 |
| 5.3.1 Medios guiados (cable coaxial, par trenzado, fibra óptica)..... | 137 |
| 5.3.2 Medios no guiados (ondas de radio, microondas, infrarrojo)..... | 138 |
| 5.4 Protocolos de comunicación.....                                   | 138 |
| 5.4.1 Protocolos de capa física.....                                  | 138 |
| 5.4.2 Protocolos de enlace de datos.....                              | 138 |
| 5.4.3 Protocolos de red (IP, ICMP).....                               | 139 |
| 5.4.4 Protocolos de transporte (TCP, UDP).....                        | 139 |
| 5.4.5 Protocolos de aplicación (HTTP, FTP, SMTP).....                 | 139 |
| 5.5 Dispositivos de red.....  | 140 |
| 5.5.1 Hubs y switches.....  | 140 |
| 5.5.2 Routers.....  | 140 |
| 5.5.3 Firewalls.....  | 140 |
| 5.5.4 Servidores y clientes.....                                      | 141 |
| 5.6 Tecnologías de Red Inalámbrica.....                               | 141 |

|  |     |
|--|-----|
| 5.6.1 Wi-Fi (IEEE 802.11).....                     | 141 |
| 5.6.2 Bluetooth.....                               | 141 |
| 5.6.3 Redes celulares (3G, 4G, 5G).....            | 142 |
| 5.6.4 Satélites de comunicación.....               | 142 |
| 5.7 Seguridad en Redes.....                        | 142 |
| 5.7.1 Amenazas comunes.....                        | 142 |
| 5.7.2 Criptografía y autenticación.....            | 146 |
| 5.7.3 VPNs (Redes Privadas Virtuales).....         | 151 |
| 5.7.4 Políticas de seguridad.....                  | 152 |
| 5.8 Internet y la World Wide Web.....              | 152 |
| 5.8.1 Estructura y funcionamiento de Internet..... | 152 |
| 5.8.2 Servicios de internet.....                   | 153 |
| 5.8.3 Protocolos web (HTTP, HTTPS).....            | 153 |
| 5.8.4 Tecnologías web modernas.....                | 153 |
| 5.9 Redes emergentes y tendencias futuras.....     | 154 |
| 5.9.1 Internet de las cosas (IoT) .....            | 154 |
| 5.9.2 Redes definidas por software (SDN).....      | 154 |
| 5.9.3 5G y más allá.....                           | 154 |
| 5.9.4 Computación en la nube y edge computing..... | 155 |
| 5.10 Gestión y monitoreo de redes.....             | 155 |
| 5.10.1 Herramientas de gestión de red.....         | 155 |
| 5.10.2 Análisis de tráfico.....                    | 155 |
| 5.10.3 Resolución de problemas.....                | 155 |
| 5.10.4 Optimización del rendimiento.....           | 156 |

**Capítulo 6**  
**Arquitecturas avanzadas y tendencias emergentes**

|   |     |
|---|-----|
| 6.1   | 158 |
| Introducción.....   |     |
| 6.2 Conceptos de Procesamiento Paralelo.....                    | 159 |
| 6.2.1 Principios Básicos del Procesamiento Paralelo.....        | 159 |
| 6.2.2 Tipos de Paralelismo .....                                | 161 |
| 6.3 Arquitecturas de multiprocesadores y multicomputadoras..... | 164 |
| 6.3.1 Procesadores Multi-Core.....                              | 164 |
| 6.3.2 Unidades de procesamiento gráfico (GPU).....              | 165 |
| 6.3.4 Arquitecturas de computación cuántica.....                | 166 |
| 6.3.5 Computación neuromórfica.....                             | 167 |
| 6.4 Arquitecturas avanzadas.....                                | 168 |
| 6.4.1 Computación cuántica.....                                 | 168 |
| 6.4.2 Computación neuromórfica.....                             | 169 |
| 6.4.3 Computación heterogénea.....                              | 169 |
| Referencias.....  | 170 |

# UNIDAD 1

Arquitectura de computador



1



# 1

## ARQUITECTURA DE COMPUTADOR

### 1.1. Introducción

La arquitectura de computadores es un campo fundamental dentro de la informática que se enfoca en el diseño y la organización de los sistemas computacionales. Esta disciplina abarca tanto el diseño físico del hardware como la estructura funcional del software, definiendo cómo se ensamblan y operan los distintos componentes dentro de un computador. La arquitectura de un computador describe cómo están organizados y cómo interactúan los elementos esenciales de un sistema computacional. Este campo cubre desde los circuitos electrónicos básicos hasta los sistemas operativos y las aplicaciones que se ejecutan sobre el hardware.

Entender cómo funciona un computador ayuda a optimizar el rendimiento de los sistemas y las aplicaciones. Los ingenieros de hardware tienen la capacidad de diseñar componentes para incrementar la velocidad y la eficiencia, mientras que los desarrolladores de software tienen la capacidad de crear código que utilice las habilidades del hardware. Problemas técnicos se pueden encontrar y resolver con un

conocimiento profundo de la arquitectura. Facilita a los desarrolladores y técnicos detectar y solucionar fallas en el software y el hardware.

La estructura de los ordenadores está en constante cambio. Los profesionales pueden estar al tanto de las innovaciones y tendencias más recientes al comprender su estructura; esto es fundamental para mejorar las tecnologías actuales y desarrollar nuevas. La arquitectura es esencial para crear nuevos sistemas computacionales, ya sean de hardware o software. Esto incluye la adaptación a las necesidades cambiantes de los usuarios y la incorporación de nuevos componentes.

## **1.2. Conceptos básicos**

### **1.2.1 Definición de arquitectura de computador**

El diseño conceptual y la estructura operacional fundamental de un sistema de computación se denominan arquitectura de computador. En otras palabras, incluye las relaciones e interacciones entre los componentes fundamentales de un sistema informático, así como su organización y diseño.

Aspectos claves de la arquitectura del computador:

**Estructura:** La disposición y conexión de los componentes hardware, que incluyen buses de comunicación, dispositivos de entrada/salida, memoria y unidad central de procesamiento (CPU).

**Funcionalidad:** El conjunto de instrucciones del procesador (ISA - Instruction Set Architecture) determina las operaciones que el hardware puede realizar.

**Rendimiento:** Los factores como la arquitectura del bus, la caché, la memoria y la velocidad del reloj del procesador afectan la eficiencia con la que se realizan las operaciones.

**Interfaz:** La forma en que las aplicaciones, el sistema operativo y los controladores interactúan con el hardware.

### **1.2.2 Historia y evolución**

La arquitectura de los computadores ha experimentado una evolución notable desde sus comienzos, destacándose por importantes hitos y avances tecnológicos que han revolucionado la manera en que usamos y comprendemos las computadoras hoy en día. A continuación, exploraremos los principales momentos y avances tecnológicos en la historia de la arquitectura de los computadores.

#### **Primera generación (1940-1956) - Tubos de vacío**

- **ENIAC (1945):** Uno de los primeros computadores electrónicos de propósito general, usaba tubos de vacío y relés electromecánicos.
- **UNIVAC I (1951):** Primer computador comercial, diseñado para aplicaciones empresariales.

#### **Segunda generación (1956-1963) – Transistores**

- **Transistores:** Reemplazaron los tubos de vacío, haciendo los computadores más compactos, rápidos y eficientes energéticamente.
- **IBM 1401 (1959):** Un computador de uso general que ganó popularidad por su flexibilidad y facilidad de programación.

### **Tercera generación (1964-1971) - Circuitos integrados**

- **Circuitos Integrados (ICs):** Permitieron la integración de múltiples transistores en un solo chip, reduciendo aún más el tamaño y el costo de los computadores.
- **IBM System/360 (1964):** Introdujo la idea de una familia de computadores compatibles, permitiendo a las empresas escalar sus sistemas sin reescribir software.

### **Cuarta generación (1971-presente) – Microprocesadores**

- **Intel 4004 (1971):** Primer microprocesador comercial, un chip de 4 bits que marcó el comienzo de los microprocesadores.
- **Altair 8800 (1975):** Considerado el primer microcomputador personal, impulsando el desarrollo de la industria del PC.
- **Apple II (1977):** Popularizó el uso de computadores personales en hogares y pequeñas empresas.

## **Quinta generación y más allá - Inteligencia artificial y computación cuántica**

- **Computadores personales (1980s):** La introducción de sistemas operativos gráficos como Windows y Mac OS hizo los computadores más accesibles.
- **Internet y redes (1990s):** La explosión de Internet transformó la manera en que los computadores se utilizaban y comunicaban.
- **Smartphones y dispositivos móviles (2000s):** La convergencia de la computación y las telecomunicaciones llevó a la creación de dispositivos portátiles y altamente integrados.
- **Inteligencia artificial y machine learning:** Computadores modernos con capacidades de aprendizaje y procesamiento masivo de datos, habilitados por arquitecturas de procesamiento paralelo y GPU avanzadas.
- **Computación cuántica:** Aunque en sus primeras etapas, representa una posible revolución en la capacidad de procesamiento y solución de problemas complejos.

### **Avances tecnológicos clave**

- **Miniaturización:** La evolución de los componentes, desde tubos de vacío a transistores, luego a circuitos integrados y microprocesadores, ha reducido su tamaño significativamente mientras aumenta su potencia.

- **Multiprocesamiento y arquitecturas paralelas:** El desarrollo de arquitecturas multicore y GPUs ha facilitado el procesamiento paralelo a gran escala, esencial para las aplicaciones modernas de inteligencia artificial y análisis de datos.
- **Tecnologías de almacenamiento:** La evolución de la tecnología de almacenamiento, desde cintas magnéticas hasta discos duros, SSDs y almacenamiento en la nube, ha mejorado notablemente la capacidad y velocidad de acceso a los datos.
- **Conectividad y redes:** La transformación de las redes, desde ARPANET hasta la Internet moderna, ha habilitado una conectividad global y ha impulsado el desarrollo de la computación en la nube.

La arquitectura de los computadores ha avanzado enormemente, desde las grandes y lentas máquinas de la primera generación hasta los dispositivos potentes y compactos de hoy. Cada generación ha aportado avances tecnológicos que han ampliado las posibilidades y aplicaciones, no solo transformando la tecnología, sino también impactando significativamente en la sociedad.

### **1.2.3 Principios fundamentales**

Estos principios han orientado el desarrollo de la arquitectura de los computadores, estableciendo una base sólida para la evolución de los sistemas computacionales desde sus orígenes hasta la tecnología avanzada actual.

## Principio de la máquina de turing

- **Introducción:** Propuesto por Alan Turing en 1936, la máquina de Turing es una abstracción teórica que define una máquina capaz de realizar cualquier cálculo lógico.
- **Componentes clave:**
  - **Cinta infinita:** Utilizada para almacenar datos de manera secuencial.
  - **Cabezal de lectura/escritura:** Puede leer y escribir símbolos en la cinta y moverla a izquierda o derecha.
  - **Estado interno:** Guarda la memoria y controla el comportamiento de la máquina.
  - **Tabla de transiciones:** Establece las reglas que determinan las acciones del cabezal basándose en el estado actual y el símbolo leído.
- **Implicaciones:** La máquina de Turing es esencial para la teoría de la computación, demostrando que cualquier problema computacional "calculable" puede ser resuelto por una máquina de Turing, sentando así las bases de la computación moderna.

## Modularidad

- **Definición:** El modularidad es el principio que sugiere que un sistema puede dividirse en componentes más pequeños y manejables, cada uno con una función específica, pero

trabajando de manera integrada para formar el sistema completo.

- **Aplicación en la arquitectura de computadores:**

- **Unidades funcionales:** Como la CPU, la memoria y los dispositivos de entrada/salida, diseñados para realizar tareas específicas.
- **Interconexión:** Uso de buses y enlaces de comunicación para permitir la interacción y el intercambio de datos entre módulos.
- **Beneficios:** Facilita la escalabilidad y el mantenimiento, permitiendo la actualización y modificación de componentes sin afectar el sistema en su totalidad, mejorando la flexibilidad y reduciendo la complejidad del diseño.

Otros principios fundamentales

- **Principio de localidad:**

- **Temporal:** Los datos o instrucciones recientemente utilizados son probables de ser usados nuevamente pronto.
- **Espacial:** Las instrucciones o datos cercanos en la memoria se utilizarán en proximidad.

- **Impacto:** Es crucial para la optimización de la memoria caché y el diseño de la arquitectura de la memoria.
- **Principio de escalabilidad:** La capacidad de aumentar la capacidad del sistema sin perder rendimiento o funcionalidad, permitiendo la expansión del procesamiento y almacenamiento.
- **Principio de concurrencia:** La habilidad de realizar múltiples operaciones simultáneamente, esencial para la arquitectura de multiprocesadores y sistemas paralelos.
- **Principio de abstracción:** Facilita ocultar los detalles complejos del funcionamiento interno, proporcionando una interfaz simplificada para desarrolladores y usuarios.

### 1.3. Arquitectura de un computador

#### 1.3.1 Estructura general

El computador funciona como un sistema integrado, donde cada parte desempeña un papel específico y la comunicación entre ellos permite realizar de manera efectiva tareas complejas, aquí el detalle:

**Procesador central (CPU)** La CPU actúa como el núcleo del computador, responsable de llevar a cabo las instrucciones de los programas y realizar cálculos. Está compuesta por:

- **Unidad aritmético-lógica (ALU):** Se encarga de ejecutar operaciones matemáticas y lógicas.

- **Unidad de control (CU):** Se ocupa de interpretar las instrucciones del software y coordinar el funcionamiento de otros componentes.
- **Registros:** Son pequeñas áreas de almacenamiento dentro de la CPU para guardar datos temporales y resultados de las operaciones realizadas.

**Memoria** La memoria se clasifica en diferentes tipos:

- **Memoria RAM (Memoria de Acceso Aleatorio):** Es de naturaleza volátil y se encarga de almacenar temporalmente datos e instrucciones que la CPU requiere mientras está en uso.
- **Memoria caché:** Es una memoria más rápida y de menor tamaño que se encuentra integrada en la CPU o cerca de ella, destinada a guardar datos de acceso frecuente para mejorar la velocidad de acceso.

**Unidad de almacenamiento** Esta unidad tiene la función de conservar datos de forma permanente. Los tipos más comunes son:

- **Disco duro (HDD):** Emplea discos giratorios y cabezales para leer y escribir información.
- **Unidad de estado sólido (SSD):** Utiliza memoria flash para el almacenamiento de datos, ofreciendo mayor velocidad y durabilidad en comparación con un HDD.

### 1.3.2. Modelo de Von Neumann

El modelo de Von Neumann ha sido crucial para el avance de la computación moderna al ofrecer una estructura definida y flexible para el diseño de computadoras. A pesar de que se han creado arquitecturas y mejoras adicionales, este modelo continúa siendo una base esencial en la teoría y la práctica de la informática.

**Procesador central (CPU)** El procesador central actúa como el núcleo del computador, responsable de ejecutar las instrucciones del software y llevar a cabo cálculos. Está compuesto por:

- **Unidad aritmético-lógica (ALU):** Encargada de realizar cálculos matemáticos y operaciones lógicas.
- **Unidad de control (CU):** Se ocupa de interpretar las instrucciones del programa y coordinar el funcionamiento de los demás componentes.
- **Registros:** Son pequeñas zonas de almacenamiento dentro de la CPU que guardan datos temporales y resultados de las operaciones realizadas.

### Memoria

- **Memoria principal (RAM):** Guarda tanto los datos como las instrucciones que la CPU necesita para operar en tiempo real. Esta memoria es de acceso aleatorio y volátil, lo que significa que su contenido se borra cuando el computador se apaga.

## **Bus de datos**

- Es un sistema de comunicación que transporta datos entre la CPU, la memoria y otros componentes del sistema.

## **Dispositivos de entrada/salida (E/S):**

- **Dispositivos de Entrada:** Permiten al usuario introducir datos en el computador, como el teclado y el ratón.
- **Dispositivos de Salida:** Permiten al usuario recibir información del computador, como el monitor y la impresora.

### **1.3.3. Modelo de Harvard**

El modelo de Harvard es un modelo de computadora que se diferencia del modelo de Von Neumann principalmente en cómo maneja la memoria y el almacenamiento de datos e instrucciones. Te explico el modelo de Harvard y las diferencias fundamentales con el modelo de Von Neumann a continuación:

#### **Memoria aparte**

**Memoria de instrucciones:** Guarda el código (instrucciones) del programa.

**Memoria de Datos:** Guarda los datos que utiliza el programa. Se pueden acceder simultáneamente a los datos y las instrucciones porque ambas memorias tienen caminos de acceso separados.

## **Memoria separada**

- **Memoria de Instrucciones:** Guarda el código del programa (instrucciones).
- **Memoria de Datos:** Guarda los datos que utiliza el programa.
- Ambas memorias tienen caminos de acceso distintos, lo que posibilita el acceso simultáneo a instrucciones y datos sin que interfieran entre sí.

## **Bus de datos y bus de instrucciones separados:**

- Emplea buses distintos para las instrucciones y para los datos, evitando los cuellos de botella que pueden surgir en el modelo de Von Neumann, donde ambos compiten por un único bus.

## **Ejecución paralela**

- Permite la lectura de instrucciones y la lectura/escritura de datos de manera simultánea, lo que mejora el rendimiento general al reducir el tiempo necesario para acceder a instrucciones y datos.

## **1.4. Componentes de la arquitectura de un computador**

### **1.4.1 Unidad Central de Procesamiento (CPU)**

El centro del procesamiento de datos de un computador es la CPU; sus componentes trabajan juntos para realizar cálculos, administrar el flujo de información en el sistema y ejecutar instrucciones. Su función es fundamental para garantizar una ejecución

efectiva de las tareas y para que el computador funcione correctamente.

**Procesador:** El núcleo de la CPU es el procesador, que es el componente que lleva a cabo las tareas de procesamiento de datos y ejecución de instrucciones. "CPU" y "procesador" con frecuencia se utilizan de manera equivalente en el uso cotidiano. La velocidad es la cantidad de operaciones que puede realizar por segundo se calcula en gigahercios (GHz). La arquitectura puede afectar su eficiencia y rendimiento.

**Núcleos:** Un procesador tiene una unidad de procesamiento llamada núcleo. Cada núcleo tiene la capacidad de ejecutar instrucciones de manera independiente, lo que posibilita la ejecución simultánea de varias tareas. Permite que cada núcleo maneje múltiples tareas o procesos al mismo tiempo, lo que mejora el rendimiento general del sistema. Por ejemplo, la capacidad del sistema para realizar múltiples tareas se incrementa cuando un procesador con cuatro núcleos puede ejecutar simultáneamente cuatro procesos diferentes.

**Hilos:** Es una unidad fundamental de ejecución en un proceso. Los hilos hacen posible dividir un proceso en partes más pequeñas que se pueden realizar en simultáneo. Al permitir que cada núcleo maneje más de una tarea a la vez, los hilos optimizan el uso de los núcleos del procesador. Esto es particularmente ventajoso para aplicaciones que pueden dividir su carga de trabajo en dos partes. Por ejemplo, un procesador con hiperhilo puede administrar dos hilos por núcleo, lo que le permite a cada núcleo realizar dos tareas al mismo tiempo.

**Caché:** La caché es una memoria de alta velocidad que se integra en el procesador (o cerca de él) y almacena instrucciones y datos que se

emplean con frecuencia. Comparado con la memoria principal (RAM), su propósito es acelerar el acceso a la información.

#### **1.4.2. Memoria**

**Memoria principal (RAM):** La memoria RAM, también conocida como memoria de acceso aleatorio, es una forma de almacenamiento volátil que el sistema emplea para almacenar los datos y las instrucciones que la CPU requiere durante el funcionamiento del computador. Cuando el computador se apaga o se reinicia, la información almacenada en la RAM se pierde. Permite acceder simultáneamente a cualquier ubicación de memoria, lo que facilita la lectura y escritura rápida de datos. Almacena instrucciones y datos que la CPU requiere de inmediato para realizar procesos y programas. Al disminuir la necesidad de acceder a la memoria secundaria (más lenta), una mayor cantidad de RAM permite ejecutar más aplicaciones al mismo tiempo y mejorar el rendimiento general del sistema.

**Memoria Secundaria (Discos Duros, SSD):** Los dispositivos de almacenamiento que mantienen los datos de manera no volátil, es decir, que los mantienen incluso cuando el computador está apagado, se denominan memoria secundaria. El Disco Duro para almacenar y recuperar datos, se utilizan cabezales de lectura/escritura y discos giratorios. Tiene un precio relativamente bajo por gigabyte y una gran capacidad. Debido a los mecanismos mecánicos involucrados, es más lento que los SSD.

**Memoria Caché:** La memoria caché es una memoria de alta velocidad que almacena instrucciones y datos y se integra en o cerca del procesador con el propósito de acelerar el acceso a la información. Los niveles de caché son caché L1, caché L2 y caché L3.

### 1.4.3. Dispositivos de entrada y salida

**Dispositivos de Entrada:** El usuario puede introducir datos e instrucciones al computador con los dispositivos de entrada; estos dispositivos transmiten información al sistema para que pueda ser procesada.

- Teclado: La presión de teclas permite al usuario introducir comandos y texto.
- Ratón (Mouse): dispositivo que, mediante movimientos y clics, permite al usuario interactuar con la interfaz gráfica del sistema.
- Escáner: Transforma imágenes o documentos físicos en formatos digitales que el computador puede procesar.
- Micrófono: Puede convertir el audio en datos digitales para que el computador pueda procesarlo.
- Cámara web: Transmite datos visuales al computador mediante imágenes y video.

**Dispositivos de Salida:** Al mostrar o reproducir datos procesados, los dispositivos de salida permiten que el computador comunique información al usuario.

- Monitor: Ofrece información visual como gráficos, video o texto. Es uno de los dispositivos de salida más importantes para la visualización de datos.

Impresora: Convierte datos digitales en una copia física en papel, como imágenes o documentos.

-

Altavoces: Son capaces de reproducir sonido digital, lo que permite escuchar voz, efectos de sonido o música.

- Auriculares: brindan una salida de audio individual que permite escuchar sonidos sin interferir con otros.

- Proyector: Es útil para visualizar en grupo o para proyectar imágenes o video en una superficie o pantalla grande.

#### **1.4.4 Bus de datos**

Un sistema de comunicación que transmite datos entre los componentes de un computador, como la memoria, la unidad central de procesamiento (CPU) y los periféricos, se conoce como bus de datos. Su función principal es llevar datos, direcciones y instrucciones dentro del sistema informático, lo que permite que los componentes interactúen de manera coordinada y efectiva.

#### **Las funciones del bus de datos**

- Transferencia de datos: Los datos se mueven entre la CPU y otros componentes del sistema, como los dispositivos de almacenamiento, la memoria RAM y los dispositivos de entrada/salida.

- Transferencia de entrenamiento: La CPU puede enviar y recibir instrucciones de y hacia la memoria, lo cual es esencial para la ejecución de programas.
- Transferencia de responsabilidades: Para determinar la ubicación de la memoria donde se deben escribir o leer datos, se utilizan líneas de dirección.

### **Tipos de buses**

- Bus de datos: Transfiere datos reales entre los componentes.
- Bus de control: Contiene señales de control que regulan el acceso al bus de datos y al bus de dirección y coordinan una variedad de operaciones.

### **Comunicación entre componentes**

Al brindar un canal común por el cual se pueden enviar y recibir datos, el bus de datos facilita la comunicación entre los diversos componentes del computador. La siguiente es la forma en que esto se logra:

- Sincronización: Para que los componentes sepan cuándo pueden enviar o recibir datos, las líneas de control manejan las señales que sincronizan la transferencia de datos.
- Arbitraje: Para evitar conflictos cuando varias partes intentan usar el bus al mismo tiempo, los controladores de bus controlan el acceso.

- Ancho de banda: El ancho del bus de datos, como los de 32 bits o 64 bits, determina la cantidad de datos que se pueden enviar al mismo tiempo; esto tiene un impacto en la velocidad y el rendimiento del sistema.

#### **1.4.5 Placa Base (Motherboard)**

El componente fundamental de la arquitectura de un computador es la placa base, también conocida como Motherboard o tarjeta madre; permite la comunicación entre todos los demás componentes del sistema. El procesador, la memoria, las tarjetas de expansión, los dispositivos de almacenamiento y otros periféricos están conectados como una plataforma.

#### **Soporte eléctrico y físico**

- Físico: Para montar y conectar componentes como la CPU, la memoria RAM, las tarjetas gráficas y otros dispositivos periféricos, la placa base brinda el apoyo físico necesario.
- Eléctrico: Ofrece las conexiones eléctricas que los componentes necesitan para poder comunicarse entre sí.

#### **Interacción entre componentes**

- El zócalo de la CPU: facilita la conexión con otros componentes al permitir la instalación del procesador en la placa base.
- Slots de memoria (DIMM/SIMM): Son conectores para módulos de memoria RAM que facilitan una comunicación rápida entre el procesador y la memoria.

- Conectores de almacenamiento: Para conectar unidades de almacenamiento, como SSDs y discos duros: se utilizan interfaces como SATA o NVMe.
- Puertos de expansión (PCI/PCIe): Puertos que mejoran las capacidades del sistema con tarjetas de expansión, como tarjetas de red, de sonido y gráficas.

**Distribución energética:** Para distribuir energía eléctrica a los distintos componentes del sistema, la placa base utiliza conectores de energía.

### **Control y sincronización**

- Chipset: El conjunto de chips (chipset) maneja funciones importantes, como la administración de los buses de datos y de control; además, regula la comunicación entre los dispositivos periféricos, la memoria y la CPU.
- BIOS/UEFI: Durante el arranque, el firmware de la placa base (BIOS o UEFI) inicia y prueba el hardware y ofrece una interfaz para la configuración del sistema.

## **1.5. Representación de la información en un computador**

### **1.5.1 Representación binaria**

La representación en formato binario es fundamental para la informática moderna debido a su facilidad, eficacia y universalidad. Las computadoras y otros dispositivos digitales pueden realizar una variedad de tareas complejas de manera confiable y rápida.

**Sistema binario:** Los bits, o dígitos binarios, son los únicos dígitos utilizados en el sistema binario. Cada bit representa un estado de apagado (0) o encendido (1). Para representar información más compleja, estas piezas se combinan en combinaciones más grandes.

### **Sistema jerárquico**

- Bit: La unidad de información binaria más fundamental.
- Byte: Ocho bits componen un byte. Para evaluar la capacidad de almacenamiento y la velocidad de transferencia de datos en los sistemas informáticos, es la unidad estándar.
- Palabra: Un conjunto de bytes que representan una unidad de datos que maneja la CPU de una computadora, generalmente en sistemas modernos de 32 bits o 64 bits, se conoce como palabra.

### **La codificación de los datos**

- Números: Las combinaciones de bits se utilizan para representar los números en binario. Por ejemplo, en binario, el número decimal 5 se representa con "101".
- Texto: Los estándares de codificación como Unicode o ASCII (American Standard Code for Information Interchange) se utilizan para asignar caracteres, números y símbolos a códigos binarios.
- Imágenes, audio y video: También se crean secuencias binarias que representan cuadros de video, píxeles o muestras de audio.

### 1.5.2 Codificación de datos

Para codificar datos en un computador, se emplean variados esquemas de codificación, cada uno creado para manejar distintos tipos de datos y usos específicos. A continuación, se detallan algunos de los esquemas de codificación más usuales:

**Codificación binaria** El sistema binario es esencial para la representación de datos en los computadores, utilizando solo los dígitos 0 y 1, conocidos como bits. Su representación puede ser en números enteros o en datos de control.

**Codificación ASCII (American Standard Code for Information Interchange)** ASCII es un esquema de codificación de caracteres que utiliza 7 bits para representar caracteres, permitiendo hasta 128 combinaciones distintas, se aplica el texto en computadores y dispositivos de comunicación también en la transferencia de datos entre dispositivos y sistemas.

**Codificación unicode:** es un estándar de codificación que usa 8, 16 o 32 bits para representar caracteres, permitiendo la representación de miles de caracteres de casi todos los sistemas de escritura del mundo. Se puede aplicar en la representación de texto en aplicaciones y sistemas globales, también en la codificación de caracteres especiales y símbolos.

**Codificación UTF (Unicode Transformation Format)** UTF es una familia de esquemas de codificación de Unicode que incluye UTF-8, UTF-16 y UTF-32. UTF-8 es especialmente popular porque es compatible con ASCII y eficiente en términos de almacenamiento. Se aplican en las

páginas web y correos electrónicos (UTF-8). También en el procesamiento y almacenamiento de texto multilingüe.

**BCD (Binary-Coded Decimal)** BCD es una forma de representar números decimales en un formato binario, donde cada dígito decimal se representa mediante un grupo de cuatro bits. Se aplica en calculadoras y dispositivos financieros. También en sistemas donde se requiere precisión decimal.

**Codificación gray** El código Gray es una forma de codificación binaria en la cual dos valores consecutivos difieren en solo un bit. Se aplica en sistemas de codificación en redes y telecomunicaciones. También en la minimización de errores en conversiones analógicas a digitales.

**Codificación base64** Base64 es un método de codificación que convierte datos binarios en una representación de texto utilizando 64 caracteres ASCII. Se aplica en la codificación de datos en correos electrónicos. También en la transmisión segura de datos en aplicaciones web.

**Codificación de imágenes y video** Formatos específicos como JPEG, PNG, MPEG y H.264 utilizan técnicas de compresión y codificación para representar imágenes y video de manera eficiente. Se aplica en el almacenamiento y transmisión de imágenes y videos. También en la compresión de medios para reducir el tamaño del archivo.

**Codificación de audio** Formatos como MP3, AAC y WAV utilizan diferentes técnicas de compresión y codificación para representar datos de audio. Se aplica en el almacenamiento y transmisión de audio. También en la reproducción de música y voz.

### 1.5.3 Sistema de numeración

Cada uno de estos sistemas de numeración tiene su propio uso y aplicación particular en el ámbito de la informática y la computación, adaptándose a diversas necesidades de representación y procesamiento de datos.

#### Sistema binario (Base 2)

El sistema binario emplea únicamente dos dígitos: 0 y 1. Cada posición en un número binario representa una potencia de 2. Por ejemplo, el número binario 1011 se convierte a decimal de la siguiente manera:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$$

Es Base para la representación interna de datos en computadores y circuitos digitales, también usado en operaciones lógicas y aritméticas dentro del hardware de los computadores.

#### Sistema decimal (Base 10)

El sistema decimal, el más común para nosotros, utiliza diez dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Cada posición en un número decimal representa una potencia de 10. Por ejemplo, el número decimal 456 se calcula así:

$$4 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 = 400 + 50 + 6 = 456$$

Este sistema es usado en la mayoría de las operaciones matemáticas diarias y en la interacción con los usuarios. Tiene una representación

estándar para la mayoría de los sistemas numéricos fuera del ámbito computacional.

### **Sistema hexadecimal (Base 16)**

El sistema hexadecimal utiliza dieciséis dígitos: 0-9 y A-F, donde A=10, B=11, C=12, D=13, E=14 y F=15. Cada posición en un número hexadecimal representa una potencia de 16. Por ejemplo, el número hexadecimal 1A31A31A3 se convierte a decimal de la siguiente manera:

$$1 \times 16^2 + A \times 16^1 + 3 \times 16^0 = 1 \times 256 + 10 \times 16 + 3 \times 1 = 419_{10}$$

Utilizado para representar direcciones de memoria y valores binarios de manera más compacta y legible. Ampliamente utilizado en programación y en la representación de colores en gráficos digitales.

#### **1.5.4 Operaciones con datos**

El procesador sigue un ciclo constante de obtención, decodificación, ejecución y almacenamiento para cada instrucción, ya sea aritmética o lógica. Este proceso ocurre a una velocidad muy alta, dictada por la frecuencia de reloj del procesador, y es esencial para todas las funciones de un computador, desde cálculos simples hasta operaciones complejas de manejo de datos.

**Operaciones aritméticas** Las operaciones aritméticas básicas (suma, resta, multiplicación, división) se ejecutan utilizando circuitos especializados dentro del procesador. El proceso comienza con la obtención de instrucciones y datos desde la memoria principal hacia el registro del procesador. Luego, el procesador decodifica la instrucción para determinar la operación aritmética específica que se realizará y

sobre qué datos. Posteriormente, la operación aritmética se lleva a cabo utilizando las unidades aritméticas y lógicas (ALU) dentro del procesador. Por ejemplo, la suma implica la adición de valores binarios utilizando la lógica de suma binaria, mientras que la resta se realiza mediante el método de complemento a dos. Operaciones más complejas como la multiplicación y la división utilizan algoritmos y circuitos especializados. Finalmente, el resultado se almacena de nuevo en la memoria o se transfiere a otro registro dentro del procesador para su uso futuro.

**Operaciones lógicas** Las operaciones lógicas en un computador involucran la manipulación de valores binarios (0 y 1) según las reglas de la lógica booleana. Esto incluye operaciones como AND, OR, NOT y XOR. El proceso inicia con la obtención y decodificación de instrucciones, similar al proceso de las operaciones aritméticas. Luego, la unidad aritmética y lógica (ALU) dentro del procesador ejecuta la operación lógica específica. Por ejemplo, AND realiza una operación bit a bit donde el resultado es 1 solo si ambos bits son 1, mientras que OR resulta en 1 si al menos uno de los bits es 1. La operación NOT invierte el valor de cada bit (0 se convierte en 1 y viceversa), y XOR resulta en 1 si los bits son diferentes. Al igual que en las operaciones aritméticas, el resultado final de las operaciones lógicas se almacena o se utiliza en operaciones subsiguientes.

## **1.6. Aplicaciones y tendencias futuras**

En los años venideros, los avances en la arquitectura de los computadores tienen el propósito de transformar la forma en que interactuamos con la tecnología y aumentar significativamente las habilidades de procesamiento y análisis de datos.

### **1.6.1. Evolución tecnológica**

#### **Arquitectura de las computadoras – evolución**

##### Computer First Generation (1940-1950s)

- Técnica: Tuberías de vacío.
- Características: Velocidad y confiabilidad limitadas, alto consumo de energía y gran tamaño. Por ejemplo: UNIVAC y ENIAC.

##### Computer Second Generation (1950s-1960s)

- Científicamente: Transistores.
- Características: mayor confiabilidad, eficiencia en el uso de energía y menor tamaño. Se incorporaron lenguajes de programación avanzados.

##### Computer Third Generation (1960s-1970s)

- Ciencia: Circuitos integrados (IC)
- Características: Un aumento notable en la miniaturización y la capacidad de procesamiento de componentes. La programación multitarea y la creación de sistemas operativos fueron factibles.

##### Computing of the Fourth Generation (1970s to present)

- Aplicación: Microprocesadores.
- Características: Un chip que integra todos los componentes del computador. Los PC se crearon con un aumento en la capacidad de procesamiento y una notable disminución en el tamaño y el costo. Por ejemplo, Intel 4004 y Pentium.

##### Computadores de quinta generación (futuro y presente)

- Tecnología: Sistemas basados en inteligencia artificial, arquitectura de 64 bits y procesadores multicore.

- Características: Computación en la nube, desarrollo de sistemas inteligentes y mayor capacidad de procesamiento paralelo. Se destaca la incorporación del aprendizaje automático y la IA en la arquitectura.

## **Avances esperados en el futuro**

### Computación cuántica

- *Descripción:* Para resolver problemas particulares, se emplean qubits en lugar de bits convencionales para realizar cálculos a velocidades exponencialmente más rápidas.
- *Impacto Esperado:* La criptografía, la optimización de problemas complejos y la simulación de materiales y moléculas serán transformadas.

### Arquitectura neuromórfica

- *Descripción:* Inspirada en la estructura y funcionamiento del cerebro humano, utiliza redes de neuronas artificiales para procesar información.
- *Impacto Esperado:* Mejorará la eficiencia en el procesamiento de tareas complejas relacionadas con el reconocimiento de patrones y la inteligencia artificial.

### Computación en la Nube y Edge Computing

- *Descripción:* La computación en la nube facilita el acceso a los recursos de computación a través de internet, mientras que el procesamiento de datos se lleva a cabo cerca de la fuente.
- *Impacto esperado:* La eficiencia en el manejo de grandes cantidades de datos y aplicaciones distribuidas se incrementará y la latencia se reducirá.

### Memoria y Almacenamiento Avanzados

- *Descripción:* El desarrollo de tecnologías de almacenamiento como la memoria óptica y de tecnologías de memoria no volátil, como la memoria 3D NAND.  
*Efecto esperado:* Una mejor integración y eficiencia energética, así como una mayor capacidad de almacenamiento y velocidad de acceso.

### Tecnología de Circuitos Integrados Avanzados

- *Descripción:* El progreso hacia la miniaturización de los transistores y la creación de arquitecturas de circuitos integrados nuevas.  
*Impacto esperado:* Permite la creación de dispositivos más rápidos y con menor consumo de energía, lo que mejora la eficiencia energética y el rendimiento.

### Computación Basada en ADN y Biotecnología

- *Descripción:* Para almacenar y procesar datos, emplea moléculas biológicas.  
*Efecto esperado:* La posibilidad de crear sistemas de procesamiento de datos a nivel molecular y de almacenamiento masivo.

#### **1.6.2 Tendencias actuales**

Al brindar nuevas habilidades y desafíos, la inteligencia artificial y la computación en la nube están transformando la arquitectura de los computadores. La automatización y el procesamiento de datos están impulsados por la IA, mientras que la computación en la nube permite una mayor escalabilidad y flexibilidad. La forma en que se diseñan y manejan los sistemas computacionales está evolucionando

continuamente debido a estas tendencias, adaptándose a las crecientes exigencias de funcionalidad, eficiencia y rendimiento.

## **Computación en la Nube (Cloud Computing)**

El uso de recursos de computación (software, almacenamiento, bases de datos, redes y servidores) a través de internet en lugar de depender de hardware local se conoce como computación en la nube. Los proveedores de servicios en la nube brindan estos recursos bajo demanda, lo que permite a las empresas escalar y administrar sus recursos de manera adaptable.

### Tendencias y avances:

Tres categorías principales de servicios en la nube son la infraestructura como servicio (IaaS), la plataforma como servicio (PaaS) y el software como servicio (SaaS). Cada uno brinda diversos niveles de flexibilidad y control. Para evitar depender de un solo proveedor, las organizaciones están adoptando enfoques híbridos que combinan servicios en la nube pública y privada, así como estrategias multinube que combinan múltiples proveedores de servicios en la nube. Al acercar el procesamiento de datos a la fuente de generación (como los dispositivos IoT), la computación en el borde (edge computing) mejora la eficiencia y la latencia y complementa la computación en la nube.

### Impacto en la Arquitectura:

- *Elasticidad y escalabilidad:* La capacidad de ajustar los recursos según la demanda y la escalabilidad rápida son posibles gracias a la arquitectura de la nube, lo que mejora la eficiencia operativa y reduce los costos.

- *Descentralización:* Una arquitectura distribuida, en la que los datos y las aplicaciones se ubican en varias ubicaciones geográficas, mejora la disponibilidad y la redundancia.

## **Inteligencia Artificial (IA)**

La simulación de procesos inteligentes por parte de máquinas, en particular sistemas computacionales, se conoce como inteligencia artificial (IA). El aprendizaje automático (machine learning), las redes neuronales, la visión por computadora y el procesamiento del lenguaje natural (NLP) son ejemplos.

### Tendencias y avances:

- *Procesadores especializados:* Las Unidades de Procesamiento de Tensor (TPU) de Google y las Unidades de Procesamiento de Datos (DPU) mejoran el rendimiento en tareas de procesamiento de grandes cantidades de datos y en tareas de aprendizaje automático.
- *Arquitectura de redes neuronales:* Las aplicaciones de IA avanzan gracias al uso de redes neuronales profundas (deep learning) para tareas complejas como el reconocimiento de patrones, imágenes y voz.
- *IA en el borde:* El procesamiento de datos localmente es posible gracias a la incorporación de IA en dispositivos de computación en el borde, lo que mejora la eficiencia y la latencia y reduce la necesidad de enviar datos a la nube.

### Impacto en la arquitectura:

- *Optimización del procesamiento:* El diseño de arquitecturas de procesadores y sistemas de memoria optimizados para realizar tareas de análisis de datos y aprendizaje automático es impulsado por la inteligencia artificial.
- *Integración de IA en sistemas:* Los sistemas de IA están siendo incorporados en varias aplicaciones, como los sistemas de recomendación y los asistentes virtuales; esto tiene un impacto en cómo se diseñan y utilizan los computadores.

### **1.6.3 Impacto de la arquitectura en el rendimiento**

Numerosos factores, como la CPU, la memoria cache, la memoria principal, el bus de datos y las tecnologías de aceleración, influyen en el rendimiento y la eficiencia de un computador. Las características particulares de cada uno de estos componentes son esenciales para la forma en que se procesan los datos y se ejecutan las instrucciones; en última instancia, esto determina la habilidad del sistema para administrar tareas y aplicaciones.

### **Unidad Central de Procesamiento (CPU)**

#### Número de núcleos y hilos:

*Impacto:* La habilidad del procesador para realizar varias tareas al mismo tiempo depende de la cantidad de núcleos y hilos presentes en una CPU. El rendimiento en aplicaciones multihilo y multitarea mejora gracias a la capacidad de ejecutar en paralelo múltiples procesos con más núcleos. Por ejemplo, un procesador de 4 núcleos no puede

realizar más tareas simultáneamente que un procesador de 8 núcleos con 16 hilos.

#### Frecuencia de reloj:

*Impacto:* La velocidad a la que el procesador puede ejecutar instrucciones se muestra en la frecuencia del reloj, que se calcula en GHz. Aunque el rendimiento del procesador depende de su arquitectura interna, una mayor frecuencia generalmente implica un procesamiento más rápido. Por ejemplo, un procesador de 3.5 GHz tiene la capacidad de ejecutar más instrucciones por segundo que uno de 2.5 GHz.

#### Arquitectura de pipeline:

*Efecto:* Las etapas de procesamiento, como la decodificación, la obtención y la ejecución de instrucciones, se pueden realizar en simultáneo gracias a la arquitectura de pipeline. La velocidad y la eficiencia del procesamiento se pueden incrementar con un pipeline más avanzado. Por ejemplo, es posible decodificar y obtener la siguiente instrucción mientras se ejecuta una instrucción mediante un pipeline de múltiples etapas.

### **Memoria cache**

#### Niveles de cache:

*Impacto:* La memoria cache (L1, L2 y L3) funciona como una memoria intermedia rápida entre la memoria principal y la CPU. Al almacenar con frecuencia datos e instrucciones, un mayor tamaño y más niveles de cache pueden mejorar el rendimiento y disminuir la latencia. Se puede

acceder a datos más rápidamente con un procesador con una cache L3 grande, por ejemplo.

#### Velocidad de acceso:

*Impacto:* La capacidad de la CPU para acceder a la memoria principal y a la cache tiene un impacto directo en el rendimiento. La eficiencia y los tiempos de espera pueden disminuir con una cache más rápida y efectiva, por ejemplo, la cache L1 es esencial para la rapidez de procesamiento y tiene tiempos de acceso más rápidos que la cache L2 y L3.

### **Memoria Principal (RAM)**

#### Capacidad de memoria:

*Efecto:* La habilidad de un sistema para manejar múltiples aplicaciones y grandes cantidades de datos está influenciada por la cantidad de memoria RAM disponible. El sistema puede manejar archivos más grandes sin utilizar el almacenamiento en disco, que es mucho más lento, y ejecutar más aplicaciones al mismo tiempo con más RAM. Por ejemplo, un sistema con 8 GB de RAM no es capaz de manejar aplicaciones y archivos grandes de manera más efectiva.

#### Velocidad de la RAM:

*Efecto:* La rapidez con que se pueden leer y escribir datos está influenciada por la velocidad de la memoria RAM, que se mide en MHz. El rendimiento general del sistema se puede mejorar con una RAM más rápida. Por ejemplo, la RAM a 3200 MHz funciona más rápido que la

RAM a 2400 MHz, lo que puede conducir a un acceso a datos más eficiente.

## **Bus de datos y ancho de banda**

### Ancho de Banda del Bus:

*Impacto:* El bus de datos conecta la CPU con los dispositivos de entrada/salida, la memoria y otras partes del sistema. El rendimiento general del sistema se mejora cuando un bus con mayor ancho de banda puede transmitir más datos simultáneamente. Un bus de 64 bits, por ejemplo, tiene la capacidad de transmitir datos más rápidamente que un bus de 32 bits.

### Latencia del bus:

*Impacto:* La latencia del bus es la duración de la transmisión de datos desde el origen hasta el destino. La transferencia de datos es más efectiva con una menor latencia.

Por ejemplo, la comunicación más rápida entre la memoria y la CPU es posible gracias a un bus de alta velocidad con baja latencia.

## **Arquitectura de instrucciones**

### Juego de Instrucciones (ISA):

*Impacto:* El conjunto de operaciones que el procesador puede realizar está definido por el juego de instrucciones. Un ISA más sofisticado puede permitir instrucciones más complejas y efectivas, lo que mejora el rendimiento en tareas específicas.

Las instrucciones SIMD (Single Instruction, Multiple Data) mejoran la eficiencia de las aplicaciones multimedia al permitir el procesamiento simultáneo de datos.

## **Tecnologías de aceleración**

### Unidades de Procesamiento Gráfico (GPU):

*Efecto:* Las GPUs son ideales para aplicaciones gráficas y cálculos científicos debido a su capacidad para realizar operaciones paralelas masivas. El rendimiento en tareas de procesamiento paralelo puede verse significativamente mejorado con la integración de GPUs. Se emplean las GPUs para la aceleración de los cálculos en aplicaciones de inteligencia artificial y para el procesamiento de gráficos en juegos.

### Aceleradores de IA y TPU:

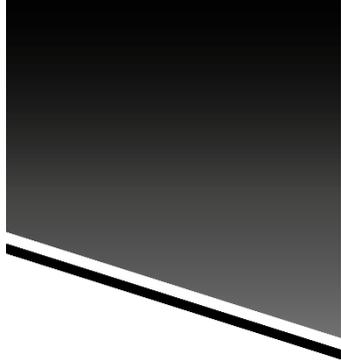
*Efecto:* Para tareas de aprendizaje automático, los aceleradores de inteligencia artificial, como las TPU (Tensor Processing Units), tienen el potencial de mejorar la eficacia en el procesamiento de modelos de inteligencia artificial. Por ejemplo, las TPUs tienen la capacidad de realizar operaciones de redes neuronales mucho más rápido que las CPUs convencionales.

# UNIDAD 2

Unidad Aritmético-Lógica (ALU)  
y unidad de control



2



## Capítulo

# 2

### **UNIDAD ARITMÉTICO-LÓGICA (ALU) Y UNIDAD DE CONTROL**

#### **2.1. Unidad Aritmético-Lógica (ALU) y Unidad de Control**

La Unidad Aritmética Lógica (ALU) es un componente fundamental de la Unidad Central de Proceso (CPU) de un ordenador. Su función principal es realizar operaciones aritméticas (suma, resta, multiplicación, división) y lógicas (NOT, AND, OR, XOR, corrimiento, rotación, etc.) sobre los datos procesados por el sistema. La ALU es esencial para la ejecución de instrucciones y es clave para el rendimiento de la CPU. Siendo, las operaciones lógicas más o menos directas, pero instrumentar operaciones aritméticas requiere conocimiento de aritmética computacional, es decir, requieren de diseño de algoritmos para manipular números mediante circuitos de hardware o rutinas de software. Lo que implica aritmética computacional y la representación de números, entonces, es necesaria una forma de representación estándar de números en los medios electrónicos, limitados por su hardware y software. Finalmente, es importante destacar que existe una amplia diversidad de formatos no convencionales para la representación de números, dentro de la máquina y que son invisibles para el usuario.

## Arquitectura de la unidad aritmético-lógico

En el año 1903 en Budapest nació el matemático John Von Neumann (Figura ), quien propuso el concepto de la ALU en 1945, a través de un informe sobre el nuevo computador llamado Computador Automático Variable Discreto Electrónico (EDVAC).

### Figura 1

*John von Neumann, 1940.*



**Nota:** <https://es.wikipedia.org/wiki/Archivo:JohnvonNeumann-LosAlamos.gif>

La arquitectura de John Von Neumann se caracteriza porque los procesadores que poseen el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos. Entonces, se almacenan en un mismo formato dentro de la memoria, y utilizan un único bus de datos para mantenerse en contacto con la CPU. Lo que significa un aumento en la eficiencia de la utilización de memoria. Todo ordenador que utiliza este modelo recurre a la unidad aritmética lógica (ALU).

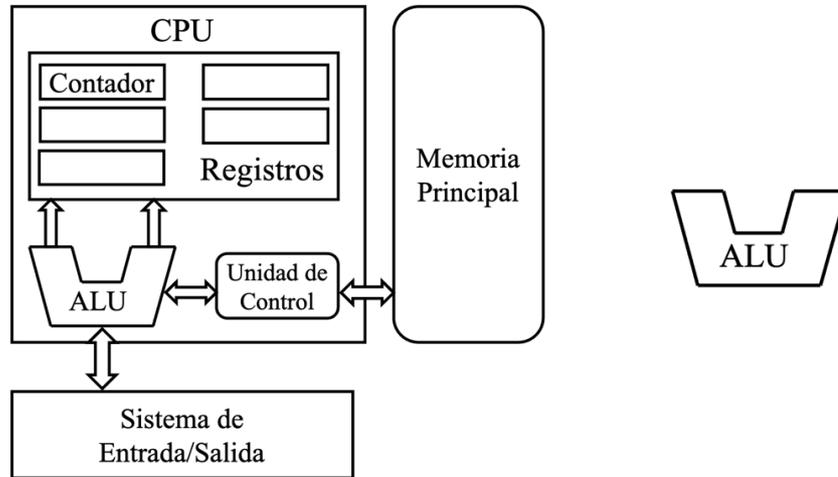
En computación, la unidad aritmético lógica (ALU) es representada como un circuito digital que calcula diferentes operaciones aritméticas (como suma, resta, multiplicación, etc.), además,

operaciones lógicas como (si, no, y "and", o "or"), entre dos números. La capacidad de la ALU para realizar operaciones aritméticas y lógicas esencial para el funcionamiento y el rendimiento de la Unidad Central de Proceso (CPU). Por tanto, con el objetivo de comprender los procesos internos de un ordenador y cómo se ejecutan las instrucciones a nivel de hardware es necesario estudiar el funcionamiento de la ALU.

De forma general, un gran número de circuitos electrónicos requieren realizar algún tipo de operación aritmética, por tanto, un circuito dentro de un reloj digital posee una ALU, y su representación clásica se puede ver en la Figura 2.

**Figura 2**

(a). *Arquitectura Von Newman.* (b) *Símbolo esquemático para una ALU.* A y B representan los operandos, R la salida; F la entrada de la unidad de control y D es un estado de la salida.



(a) Arquitectura Von Neumann

(b) Unidad Aritmética Lógica

**Representación del sistema decimal**

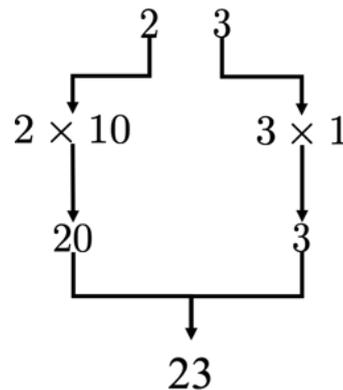
En el sistema de numeración decimal cada uno de los diez dígitos

(0 – 9), representa una determinada cantidad. Los diez símbolos (dígitos) no se limitan a expresar solamente diez cantidades diferentes, ya que el uso de varios dígitos en las posiciones adecuadas dentro de un número indica la magnitud de la cantidad.

Un número decimal se diferencia de sus representaciones, a través de la posición de cada dígito dentro del número, donde se indica la magnitud que representa (es decir, los numerales). Por ejemplo, el número "23" se describe en la Figura

**Figura 3**

*Representación de un número decimal.*



Los pesos para los números enteros son representados por las potencias positivas de diez, que aumentan de derecha a izquierda, iniciando por  $10^0 = 1$ , es decir,

$$\dots 10^5 \ 10^4 \ 10^3 \ 10^2 \ 10^1 \ 10^0 \tag{0.1}$$

En consecuencia, para números con fracciones, los pesos se representan con potencias negativas de diez que decrecen de izquierda a derecha comenzando por  $10^{-1}$ .

$$, 10^{-1} 10^{-2} 10^{-3} \dots \tag{0.2}$$

**EJEMPLO:**

Expresar el número decimal 47 como una suma de valores de cada dígito.

$$\begin{aligned}47 &= (4 \times 10^1) + (7 \times 10^0) \\ &= (4 \times 10) + (7 \times 1) \\ &= 40 + 7\end{aligned}$$

Sin embargo, no siempre se hace la diferencia entre números y numerales y con frecuencia se usan bases distintas a representación en base 10. Las bases distintas de 10, como los números con base 2 (binario) se volvió popular con la aparición de las computadoras debido a su uso de dígitos binarios, o bits, que tienen sólo dos posibles valores, 0 y 1; que, además, es compatible con las señales electrónicas. Los números con base 8 (octal) y base 16 (hexadecimal) se usan como notación abreviada para los números binarios.

En un sistema numérico posicional general de base  $r$ , con ancho de palabra fijo  $k$ , un número  $x$  se representa mediante una cadena de  $k$  dígitos  $x_i$ , con  $0 \leq x_i \leq r - 1$ .

$$x = \sum_{i=0}^{k-1} x_i r^i = (x_{k-1} x_{k-2} \cdots x_1 x_0) r \quad (0.3)$$

**EJEMPLO:**

Expresar el número decimal 27 en binario.

$$\begin{aligned}27 &= (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= (11011)_2\end{aligned}$$

De esta forma, en un sistema numérico de base  $r$  y  $k$  dígitos, se pueden representar los números naturales desde 0 hasta  $r^k - 1$ . Por otra parte, dado un rango de representación  $[0, P]$ , se requiere de  $k$  de dígitos en base  $r$ , siendo:

$$k = \lceil \log(P + 1) \rceil = \lceil \log_r P \rceil + 1 \quad (0.4)$$

**EJEMPLO:**

La representación numérica convencional de base  $x$ , use los valores digitales desde 0 hasta  $r - 1$ , determinar el número más grande "máx" que sea representativo con el número indicado  $k$  de dígitos y el número mínimo  $K$  de dígitos requeridos para representar todos los números naturales menores que un millón.

El número natural más grande a representar para la segunda parte de la pregunta es  $P = 999\,999$

a)  $r = 2, k = 16$

$$\max = r^k - 1 = 2^{16} - 1 = 65,535$$

$$K = \lceil \log_r(P + 1) \rceil = \lceil \log_2(10^6) \rceil = 20 (2^{19} < 10^6)$$

b)  $r = 3, k = 10$

$$\max = 3^{10} - 1 = 59048$$

$$K = \lceil \log_3(10^6) \rceil = 13 (3^{12} < 10^6)$$

Por otra parte, el desbordamiento en aritmética de enteros se produce cuando una operación aritmética produce un resultado que excede el rango de valores que pueden representarse con el número de bits disponibles en un sistema digital. Esto puede ocurrir en sumas, restas, multiplicaciones y otras operaciones aritméticas.

En la suma ocurre cuando dos números enteros producen un resultado que es mayor que el valor máximo representable en el

sistema, por ejemplo, en binario (8 bits):  $11001000_2 + 01100100_2 = 100101100_2$ , siendo, el resultado correcto en binario de 9 bits, no se puede representar en 8 bits.

En la resta ocurre cuando dos números enteros producen un resultado que es menor que el valor mínimo representable en el sistema, por ejemplo, en un sistema de 8 bits con números sin signo, restar 100 a 50, lo que produce un de valor negativo, que no se puede representar.  $50_{10} - 100_{1-} = -50_{10}$ , así, como en binario (8 bits sin signo), no se puede representar un valor negativo.

En la multiplicación se produce cuando el producto de dos números enteros devuelve un resultado mayor que el valor máximo representable en el sistema, por ejemplo, en un sistema de 8 bits con números sin signo, la multiplicación de 20 y 15 devuelve como resultado 300, que supera el máximo representable (255), (es decir,  $20_{10} \times 15_{10} = 300_{10}$  En binario (8 bits): No se puede representar en 8 bits).

Finalmente, el desbordamiento de signo se produce cuando una operación aritmética cambia el signo de un número, por ejemplo, en un sistema de 8 bits con números con signo (-128 a 127), la suma de 127 y 1 daría como resultado -128, un cambio de signo inesperado, es decir,  $27_{10} + 1_{10} = 128_{10}$ , en binario con signo (8 bits):  $01111111_2 + 00000001_2 = 10000000_2$  (representa -128 en complemento a dos).

No obstante, se puede gestionar el desbordamiento en las diferentes operaciones por medio de diversos lenguajes de programación, como C o Python.

**EJEMPLO:**

Para cada uno de los siguientes sistemas convencionales de representación numérica en base  $r$ , determine si evaluar la expresión aritmética mostrada conduce a desbordamiento. Todos los operandos se dan en base 10.

a)  $r = 2, k = 16; 10^2 \times 10^3$

El resultado  $10^5$  es mayor que  $máx = 65\,535$ , de modo que ocurrirá desbordamiento.

b)  $r = 3, k = 10; 15\,000 + 20\,000 + 25\,000$

El resultado  $60\,000$  es mayor que  $máx = 59\,048$ , de modo que ocurrirá desbordamiento.

c)  $r = 8, k = 6; 555 \times 444$

El resultado  $246420$  no es mayor que  $máx = 262143$ , de modo que no se presentará desbordamiento.

**Conversión de base numérica**

Si se asume un número  $x$  representado en base  $r$ , su representación en base  $R$  puede ser expresado de dos formas.

**• Regla de Horner**

La nueva base  $R$ , se evalúa por medio de un polinomio en  $r$  cuyos coeficientes son los dígitos  $x_i$ , expresados en la a regla de Horner, la cual plantea pasos alternados de multiplicación por  $r$  y sumas, tal que:

$$(x_{k-1}x_{k-2} \cdots x_1x_0)_r = (\cdots ((0 + x_{k-1})r + x_{k-2})r + \cdots + x_1)r + x_0 \quad (0.5)$$

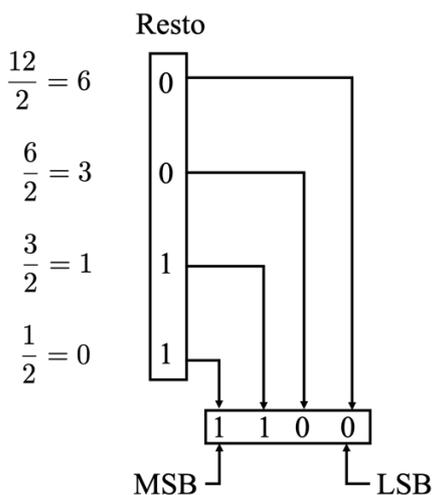
Procedimiento adecuado para la conversión de una base arbitraria  $r$  a la base 10. Por tanto, la complejidad de cálculo aritmético en base 10 es baja por lo que es adecuado para este tipo de conversión.

- **Divisiones sucesivas**

Un método sistemático para convertir números enteros decimales a una base  $r$  es el proceso de la división sucesiva por la base  $r$ . Es decir, para convertir el número decimal 12 a binario ( $r = 2$ ), se inicia dividiendo 12 entre 2. A continuación, cada cociente resultante se divide entre 2 hasta obtener un cociente cuya parte entera sea igual a 0. Los restos generados en cada división forman el número binario. El primer resto es el bit menos significativo (LSB) del número binario y el último resto es el bit más significativo (MSB).

**EJEMPLO:**

Número decimal 23, base de conversión binario ( $r = 2$ ).



- **Conversión de fracciones decimales**

La parte fraccional de un número se representa por medio de un punto base (punto decimal en base 10, punto binario en base 2, etc.). La posición del punto base casi siempre es implícita y el punto no se muestra explícitamente. Si un número de punto fijo tiene  $k$  dígitos enteros y dígitos fraccionales, su valor se puede expresar como:

$$x = \sum_{i=-1}^{k-1} x_i r^i = (x_{k-1} x_{k-2} \cdots x_1 x_0 \cdot x_{-1} x_{-2} \cdots x_{-l})_r \quad (0.6)$$

En específico, los dígitos a la derecha del punto base se les asignan índices negativos y sus pesos son potencias negativas de la base.

**EJEMPLO:**

Convertir el número decimal 0,625 en binario

$$0,625 = 0,5 + 0,125$$

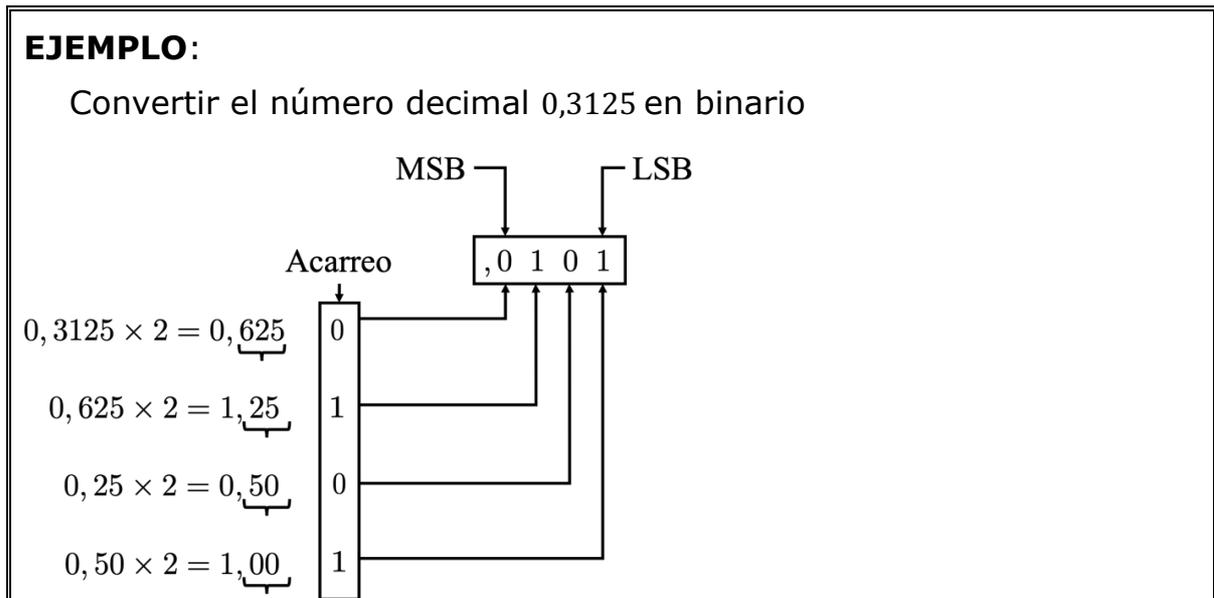
$$= 2^{-1} + 2^{-3} = 0,101$$

Entonces, un sistema numérico de fracción decimal en base  $r$  y  $(k + l)$  dígitos, con  $k$  dígitos enteros, puede representar los números desde 0 hasta  $r^k - r^{-l}$ , en incrementos de  $r^{-l}$ . Siendo el tamaño de paso o resolución  $r^{-l}$ .

En específico para el ejemplo anterior, se indica que hay un 1 en la posición  $2^{-1}$ , un 0 en la posición  $2^{-2}$  y un 1 en la posición  $2^{-3}$ . Es decir, los pesos binarios fraccionarios inician desde significativo es 0,5 que es el más significativo, es decir  $2^{-1}$ , y que dividiendo entre dos cualquier peso se obtiene el siguiente peso menor, entonces, los primeros cuatro pesos binarios fraccionarios se definen como 0,5; 0,25; 0,125; 0,0625.

Por otra parte, de manera análoga a la conversión de números decimales enteros a binario, se realiza dividiendo de forma sucesiva entre 2. Los números decimales fraccionarios pueden convertirse en números binarios multiplicando sucesivamente por 2. Por ejemplo, para convertir a binario el número decimal fraccionario 0,3125, el inicio del proceso se da por la multiplicación de 0,3125 por 2 y después se

multiplica cada parte fraccional resultante del producto por 2 hasta que el producto fraccionario sea cero o hasta que se alcance el número deseado de posiciones decimales. De esta forma, los dígitos de acarreo, generados por las multiplicaciones dan lugar al número binario, donde, el primer acarreo define el MSB y el último acarreo define el LSB, como se puede apreciar en el siguiente ejemplo.



## Sumadores y ALU simples

Un computador digital realiza el conjunto de operaciones digitales, no obstante, la operación aritmética de mayor relevancia es la suma, así, el diseño de sumadores de bits (medio sumadores y sumadores completos), es una tarea de relevancia significativa dentro de los procedimientos que implican contadores, operaciones de corrimiento y lógicas, dentro de la ALU, en específico, a continuación, se cubre la suma y resta binaria.

- **Semi-Sumador o Sumador simple**

A partir de las reglas básicas de la suma binaria

**Tabla 1***Suma binaria de 2 bits.*

| A | + | B | = | $C_{out}$ | $\Sigma$ |
|---|---|---|---|-----------|----------|
| 0 | + | 0 | = | 0         | 0        |
| 0 | + | 1 | = | 0         | 1        |
| 1 | + | 0 | = | 0         | 1        |
| 1 | + | 1 | = | 1         | 0        |

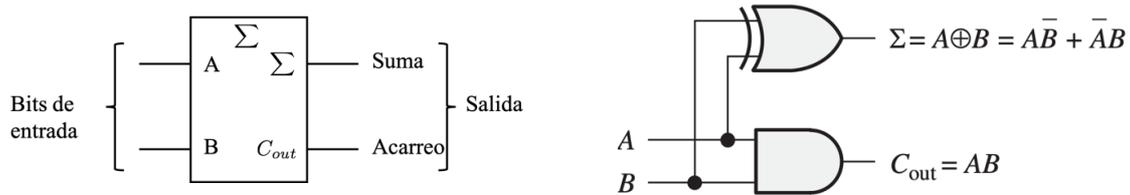
 $\Sigma$  = suma $C_{out}$  = acarreo de salida

A y B = variables de entrada.

Entonces, el semi-sumador suma dígitos binarios en sus entradas y genera dos dígitos binarios en sus salidas, definidos como un bit de suma (S) y un bit de acarreo (C). Esquema que se puede representar en forma de símbolo lógico y de circuito lógico como se puede ver en el bloque de la Figura 4

**Figura 4**

a) Símbolo lógico de un semi-sumador. b) Circuito lógico de un semi-sumador.



Las expresiones correspondientes a la suma y al acarreo de la salida se pueden obtener como funciones de las entradas. Entonces, la salida de acarreo ( $C_{out}$ ) es 1 sólo cuando A y B son 1 (valor expresable por medio de una compuesta AND), es decir,  $C_{out} = AB$ . Mientras, la salida correspondiente a la suma ( $\Sigma$ ) es 1 sólo si las variables A y B son distintas (valor expresable por medio de una operación OR-exclusiva de las variables de entrada), es decir,  $\Sigma = A \oplus B$ .

- **Sumador Completo**

Un sumador completo acepta un bit adicional de acarreo en las entradas. Las operaciones de implicadas en el sumador completo se definen como:

**Tabla 2**

*Suma binaria de 2 bits.*

| A | + | B | + | $C_{in}$ | = | $C_{out}$ | $\Sigma$ |
|---|---|---|---|----------|---|-----------|----------|
| 0 | + | 0 | + | 0        | = | 0         | 0        |
| 0 | + | 0 | + | 1        | = | 0         | 1        |
| 0 | + | 1 | + | 0        | = | 0         | 1        |
| 0 | + | 1 | + | 1        | = | 1         | 0        |
| 1 | + | 0 | + | 0        | = | 0         | 1        |
| 1 | + | 0 | + | 1        | = | 1         | 0        |
| 1 | + | 1 | + | 0        | = | 1         | 0        |
| 1 | + | 1 | + | 1        | = | 1         | 1        |

$C_{in}$  = Acarreo de entrada.

$C_{out}$  = Acarreo de salida

$\Sigma$  = suma

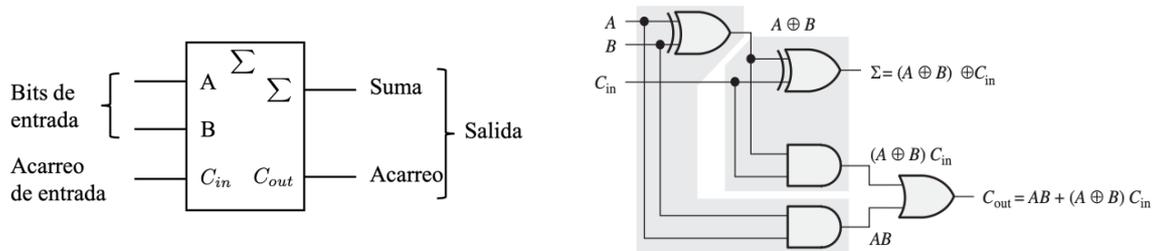
A y B = variables de entrada (operandos)

Mientras, el símbolo lógico y el circuito lógico se muestran en la Figura 5.

**Figura 5**

a) *Símbolo lógico de un semi-sumador. (Sumador completo)*

b) *Circuito lógico de un semi-sumador. (sumador completo)*



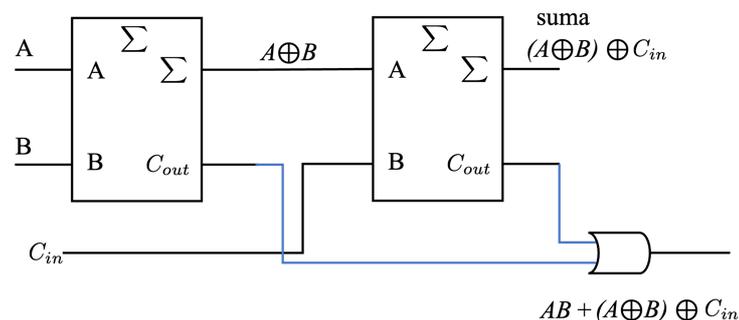
Como se puede ver en la Figura 5, el acarreo de salida es 1, sí las

dos entradas de la primera puerta XOR están a 1, o sí las dos entradas de la segunda puerta XOR están a 1. Es decir, el acarreo de salida del sumador completo se obtiene a partir del producto lógico (AND) de las entradas  $A$  y  $B$ , y del producto lógico de  $A \oplus B$  y  $C_{in}$ . Finalmente, la operación OR sobre estos resultados indica la salida del acarreo.

Por otra parte, ya que se ha alcanzado resultados de un semi-sumador, la implementación de un sumador completo también es posible, la Figura 6, muestra un esquema de configuración de un sumador completo, a partir de 2 semi-sumadores.

**Figura 6**

a) Símbolo lógico de un sumador completo, construidos por 2 semi-sumador.



- **Redes de propagación de acarreo**

Como se ha podido apreciar, cuando se suman dos números, se generan acarreos en ciertas posiciones. Entonces, para que en la suma binaria exista acarreo, es necesario que ambos bits operando sean 1. Por tanto, se define una señal binaria auxiliar  $g_i$  como 1 para las posiciones en las que se genera un acarreo y 0. Para la suma binaria,  $g_i = x_i y_i$ ; siendo  $g_i$  una operación la AND lógica de los bits operandos  $x_i$  y  $y_i$ . Quedando posiciones de dígito en las que se propaga un acarreo entrante.

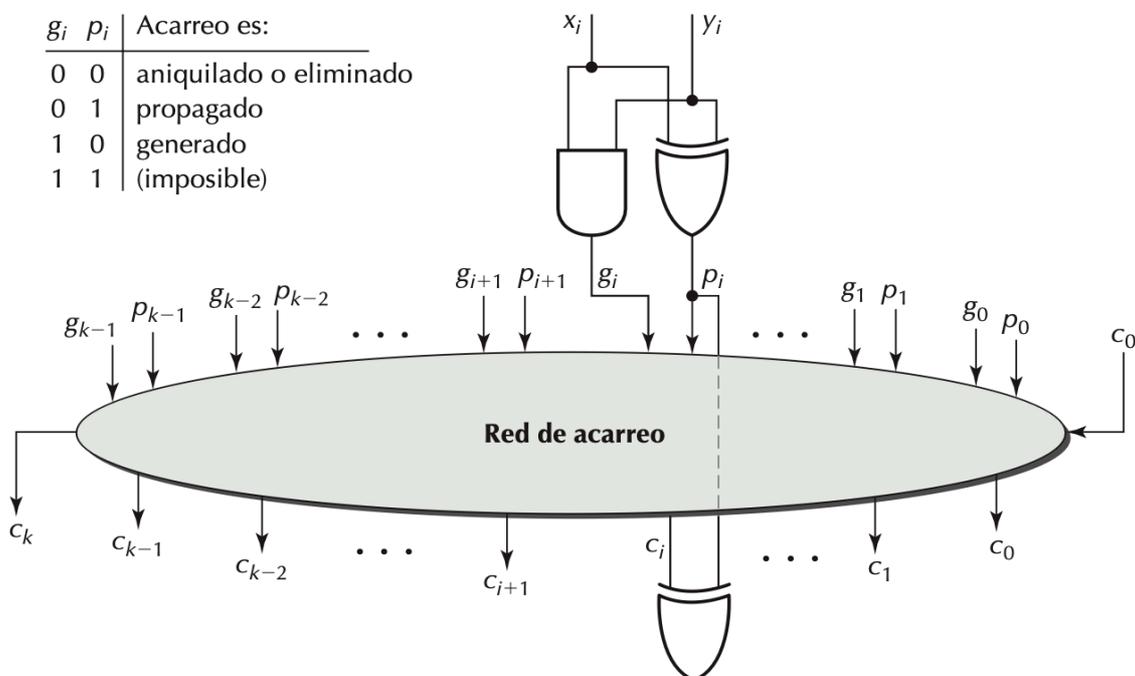
En la suma binaria, la propagación de acarreo requiere que un bit operando sea 0 y el otro sea 1. La señal binaria auxiliar  $p_i$ , expresada como  $p_i = x_i \oplus y_i$  para suma binaria, definida como 1 sí la posición del dígito  $i$  propaga un acarreo en la entrada.

La Figura 7 muestra la estructura general de un sumador binario, donde, el acarreo entrante  $c_0$  del sumador y de las señales auxiliares  $g_i$  y  $p_i$  para todas las posiciones de dígito  $k$ , obtenidos, por medio de los acarreos intermedios  $c_i$  y del acarreo saliente  $c_k$  independientemente de los valores en la entrada, expresados como:

$$s_i = x_i \oplus y_i \oplus c_i = p_i \oplus c_i \quad (0.7)$$

**Figura 7**

*Sumador es la red de acarreo y configuración de compuertas que generan los bits suma y las señales g y p*



## **Sumadores de acarreo serie y de acarreo anticipado**

Los sumadores paralelo pueden clasificarse en dos categorías, siendo, acarreo serie y acarreo anticipado. En específico estos sumador no difieren en términos de entradas y salidas. La diferencia se centra en la velocidad con la que suman los números (el sumador de acarreo anticipado es más rápido que el sumador de acarreo serie).

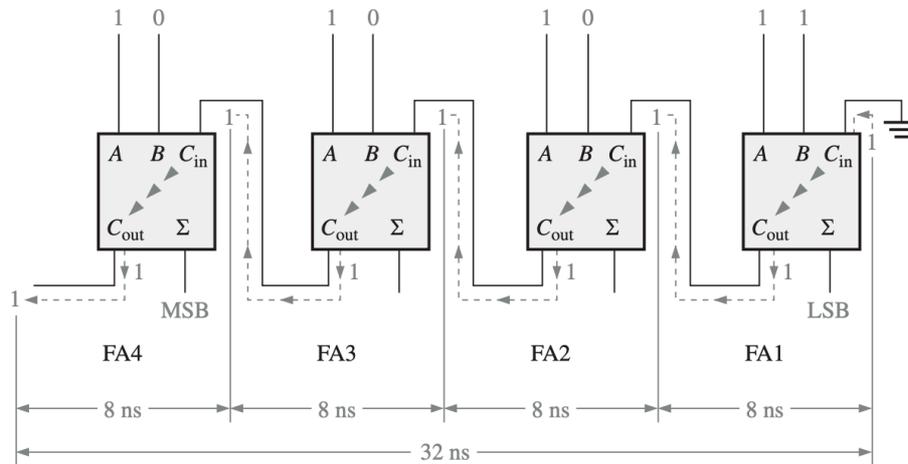
- **Sumador de acarreo serie**

En la salida de acarreo de cada sumador completo se conecta a la entrada de acarreo del sumador completo anterior. La suma y el acarreo de salida de cualquier sumador no se pueden generar hasta obtener el acarreo de entrada, lo que da lugar a un retardo temporal en el proceso, como se muestra en la Figura

De esta forma, se puede apreciar que el retardo de propagación del acarreo para cada sumador completo se debe al acarreo de entrada hasta el acarreo de salida, lo que produce un tiempo de retardo, asumiendo que las entradas  $A$  y  $B$  ya existan. Por tanto, el retardo acumulado por medio de las diferentes etapas de sumador, se traduce como el tiempo de suma.

## Figura 8

Un sumador paralelo de 4 bits con acarreo serie



- **Sumador de acarreo anticipado**

Opera por medio de las etapas de un sumador paralelo. Este método permite acelerar el proceso de adición eliminando el retardo del acarreo serie. El sumador con acarreo anticipado se anticipa el acarreo de salida de sumador y, en función de los bits de entrada, genera el acarreo de salida mediante la generación de acarreo o la propagación de acarreo.

De esta forma, si el sumador completo genera un acarreo de salida, solo cuando ambos bits de entrada son 1, entonces, se genera un acarreo. El acarreo generado,  $C_g$ , se expresa como la función AND de los 2 bits de entrada,  $A$  y  $B$ . Mientras que, la propagación de acarreo se transmite como acarreo de salida. Un acarreo de entrada puede ser propagado por el sumador completo cuando uno o ambos bits de entrada son igual a 1. El acarreo propagado,  $C_p$ , se define como la función OR de los bits de entrada, tal que:

$$C_g = AB \quad (0.8)$$

$$C_p = A + B \quad (0.9)$$

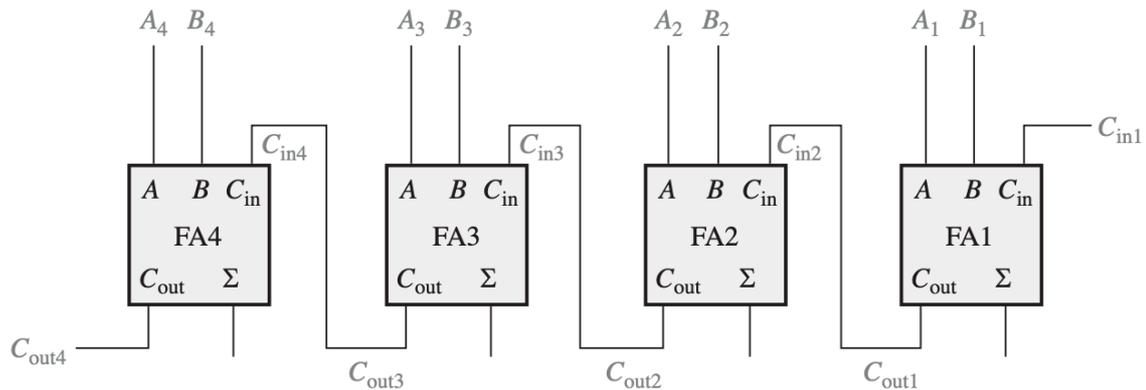
El acarreo de salida de un sumador completo  $C_g$  y el acarreo propagado  $C_p$ . El acarreo de salida  $C_{out}$  es un 1 si el acarreo generado es 1, ó si el acarreo propagado es 1, y el acarreo de entrada  $C_{in}$  es 1, es decir:

$$C_{out} = C_g + C_p C_{in} \quad (0.10)$$

En consecuencia, un sumador completo, devuelve un acarreo de salida que depende de  $C_g$ ,  $C_p$  y  $C_{in}$ . Entonces, los bits de las funciones  $C_g$  y  $C_p$  para cada sumador están disponibles para la entrada  $A$  y  $B$  y el acarreo de entrada del sumador menos significativo (LSB).

**Figura 9**

*Acarreo y propagación de acarreo en un sumador de 4 bits.*



Sumador completo 4      Sumador completo 3      Sumador completo 2      Sumador completo 1

$$C_{g4} = A_4 B_4$$

$$C_{g3} = A_3 B_3$$

$$C_{g2} = A_2 B_2$$

$$C_{g1} = A_1 B_1$$

$$C_{p4} = A_4 + B_4$$

$$C_{p3} = A_3 + B_3$$

$$C_{p2} = A_2 + B_2$$

$$C_{p1} = A_1 + B_1$$

Por lo tanto, a partir del desarrollo descrito, las expresiones de salida de los sumadores se definen como:

Sumador completo 1

$$C_{out1} = C_{g1} + C_{p1} C_{in1} \quad (0.11)$$

### Sumador completo 2

$$\begin{aligned}C_{in2} &= C_{out1} \\C_{out2} &= C_{g2} + C_{p2}C_{in2} \\&= C_{g2} + C_{p2}C_{out1} \\&= C_{g2} + C_{p2}(C_{g1} + C_{p1}C_{in1})\end{aligned}\tag{0.12}$$

### Sumador completo 3

$$\begin{aligned}C_{out3} &= C_{out2} \\C_{out3} &= C_{g3} + C_{p3}C_{in3} \\&= C_{g3} + C_{p3}(C_{g2} + C_{p2}C_{g1} + C_{p2}C_{in1} + C_{p1}C_{in1}) \\&= C_{g3} + C_{p3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{in1}\end{aligned}\tag{0.13}$$

### Sumador completo 4

$$\begin{aligned}C_{out4} &= C_{out3} \\C_{out4} &= C_{g4} + C_{p4}C_{in4} \\&= C_{g4} + C_{p4}C_{out3} \\&= C_{g4} + C_{p4}(C_{g3} + C_{p3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{in1}) \\&= C_{g4} + C_{p4}C_{g3} + C_{p4}C_{p3}C_{g2} + C_{p4}C_{p3}C_{p2}C_{g1} + C_{p4}C_{p3}C_{p2}C_{p1}C_{in1}\end{aligned}\tag{0.14}$$

Entonces, se puede apreciar que todos los acarrees de salida están disponibles y no es necesario esperar a que se propague un acarreo a través de todos los sumadores antes de obtener el resultado final, en consecuencia, el acarreo anticipado acelera se realiza en menor tiempo.

### **ALU multifunción**

La ALU multifunción no sólo gestiona sumas y restas, sino que también realiza diversas operaciones lógicas y aritméticas, como AND, OR, XOR, NOT, desplazamientos, etc. Representa un componente versátil y esencial en los sistemas y procesadores digitales. Su

capacidad para realizar diversas operaciones aritméticas y lógicas lo convierte en un componente fundamental para la ejecución de programas y la manipulación de datos. Comprender su diseño y funcionamiento es crucial para el desarrollo eficiente del hardware y la optimización de los procesos computacionales.

La unidad lógica realiza una de las cuatro operaciones lógicas de bits, entonces, el diseño de la unidad lógica consiste de cuatro arreglos de compuertas que calculan tales funciones lógicas y un multiplexor de cuatro entradas que permite elegir el conjunto de resultados, entonces, una ALU multifunción se puede implementar utilizando una combinación de sumadores, puertas lógicas y multiplexores.

# UNIDAD 3

Introducción a los sistemas operativos



3

# Capítulo 3

## INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

### 3.1 Definición y propósito de un sistema operativo

Un **sistema operativo (SO)** es un software fundamental que actúa como intermediario entre los usuarios y el hardware de una computadora. Su propósito principal es gestionar y coordinar los recursos de hardware y software del sistema, proporcionando un entorno estable y consistente para que las aplicaciones se ejecuten de manera eficiente.

#### Figura 10

Sistemas Operativos vigentes



## Propósitos del sistema operativo

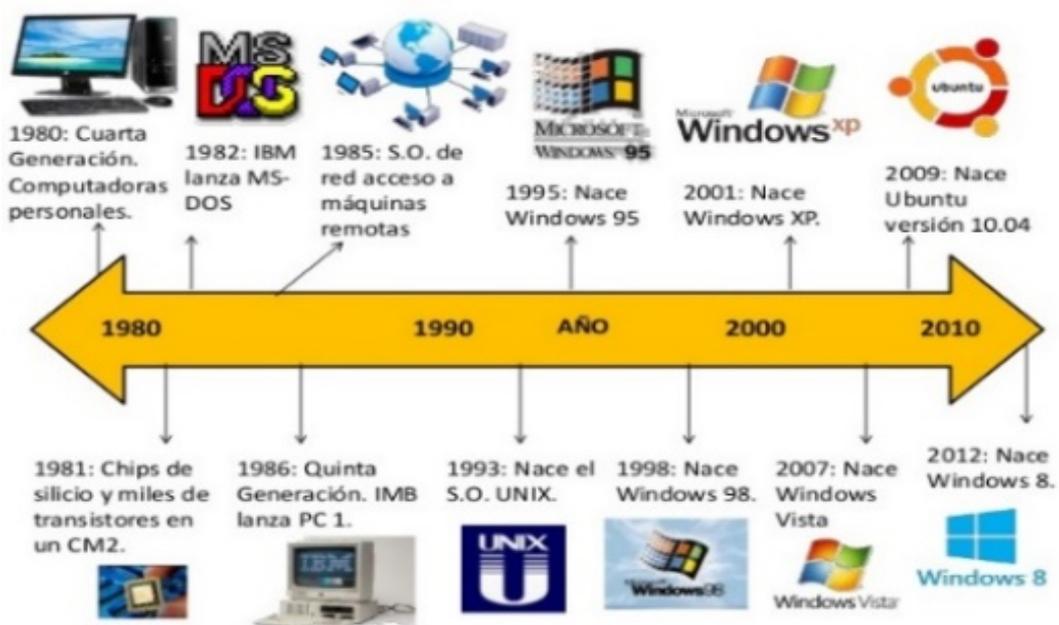
- **Gestión de recursos:** Asignar y monitorear el uso de la CPU, memoria, dispositivos de entrada/salida y almacenamiento.
- **Interfaz de usuario:** Proveer una interfaz, ya sea gráfica o de línea de comandos, para interactuar con el sistema.
- **Ejecución de aplicaciones:** Cargar y ejecutar aplicaciones de manera eficiente, gestionando los procesos y la memoria.
- **Seguridad:** Proteger el sistema y los datos contra accesos no autorizados y ataques.
- **Comunicación:** Facilitar la comunicación entre diferentes partes del sistema y entre usuarios y dispositivos.

## 3.2 Historia y evolución de los sistemas operativos

La evolución de los sistemas operativos ha pasado por varias etapas clave:

**Figura 11**

Evolución de los sistemas operativos



- **Década de 1950:** Las primeras computadoras no tenían sistemas operativos. Los programas eran cargados manualmente en la memoria y ejecutados secuencialmente.
- **Década de 1960:** Aparecen los primeros sistemas operativos, como el **Compatible Time-Sharing System (CTSS)** y **MULTICS**. Estos sistemas introdujeron conceptos de multitarea y tiempo compartido, permitiendo a múltiples usuarios ejecutar programas simultáneamente.
- **Década de 1970:** Se desarrollan sistemas operativos más avanzados como **UNIX**, que introdujo una arquitectura modular y conceptos de multitarea y multiusuario. Microsoft también desarrolló **MS-DOS** para computadoras personales.
- **Década de 1980:** Aparecen las interfaces gráficas de usuario (GUI) con sistemas como **Macintosh System Software** y **Microsoft Windows**. UNIX se expande y se crean variantes como **BSD**.
- **Década de 1990:** La popularidad de Windows crece con el lanzamiento de **Windows 95** y **Windows NT**. También surge **Linux**, un sistema operativo de código abierto basado en UNIX.
- **Década de 2000:** Los sistemas operativos móviles ganan prominencia con el lanzamiento de **iOS** y **Android**. Windows y macOS siguen evolucionando, y Linux se convierte en una opción popular para servidores y desarrolladores.
- **Década de 2010 y en adelante:** La virtualización y los contenedores se vuelven populares, con tecnologías como **Docker** y **Kubernetes**. Los sistemas operativos distribuidos y para Internet de las cosas (IoT) también ganan importancia.

### 3.3 Funciones principales de un sistema operativo

Las principales funciones de un sistema operativo se pueden agrupar en las siguientes categorías:

**Figura 12**

*Funciones principales de un sistema operativo*



#### **Gestión de procesos**

- Creación y eliminación de procesos.
- Planificación de la CPU y asignación de tiempo de procesador.
- Sincronización y comunicación entre procesos.

#### **Gestión de memoria**

- Asignación y liberación de memoria a procesos.
- Implementación de memoria virtual para ampliar la capacidad de la memoria física.
- Control de acceso y protección de la memoria.

## **Gestión de almacenamiento**

- Organización y control del acceso a los archivos en dispositivos de almacenamiento.
- Mantenimiento del sistema de archivos y manejo de operaciones de entrada/salida.
- Seguridad y protección de los datos almacenados.

## **Gestión de dispositivos**

- Control de los dispositivos de entrada y salida (E/S).
- Implementación de controladores de dispositivos y manejo de interrupciones.
- Optimización del uso de dispositivos de E/S para mejorar el rendimiento.

## **Interfaz de usuario**

- Provisión de una interfaz gráfica (GUI) o de línea de comandos (CLI) para la interacción del usuario.
- Gestión de ventanas, menús, y otros elementos de interfaz.

## **Seguridad y protección**

- Implementación de mecanismos de autenticación y autorización.
- Protección de recursos del sistema contra accesos no autorizados.
- Monitoreo y auditoría de actividades para detectar y prevenir amenazas.

## **Comunicación y Red**

- Facilitar la comunicación entre dispositivos y redes.
- Implementación de protocolos de red y manejo de conexiones.

Estas funciones permiten que el sistema operativo proporcione un entorno estable y eficiente para la ejecución de aplicaciones, asegurando que los recursos del sistema se utilicen de manera óptima y segura.

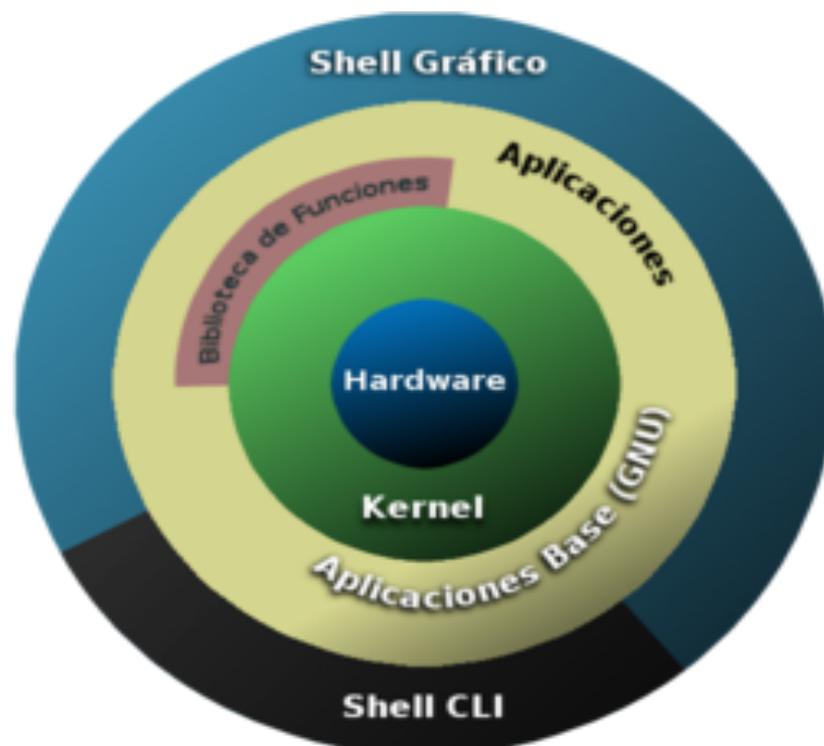
## 3.2. Arquitectura de los sistemas operativos

### 3.2.1 Componentes de un sistema operativo

Un sistema operativo está compuesto por varios componentes fundamentales que trabajan en conjunto para gestionar los recursos del sistema y proporcionar servicios a las aplicaciones y usuarios. Los componentes principales incluyen:

#### Figura 13

*Arquitectura de los sistemas operativos*



## **Núcleo (Kernel)**

- El núcleo es el corazón del sistema operativo. Controla todas las operaciones del hardware y gestiona recursos como la CPU, la memoria y los dispositivos de entrada/salida.
- Se encarga de la planificación de procesos, la gestión de memoria, el manejo de interrupciones y la comunicación entre procesos.

## **Gestor de procesos**

- Controla la creación, ejecución y terminación de los procesos.
- Planifica el uso de la CPU para maximizar la eficiencia y garantizar la equidad en la asignación de recursos.

## **Gestor de memoria**

- Administra la memoria principal del sistema.
- Se encarga de la asignación y liberación de memoria, la paginación y la segmentación, y la gestión de la memoria virtual.

## **Sistema de archivos**

- Proporciona una estructura lógica para almacenar y organizar datos en dispositivos de almacenamiento.
- Maneja operaciones de creación, lectura, escritura y eliminación de archivos.

## **Gestor de dispositivos**

- Controla y coordina el uso de los dispositivos de entrada y salida.

- Implementa controladores de dispositivos y maneja las interrupciones y el acceso a los dispositivos.

### **Interfaz de usuario**

- Proporciona medios para que los usuarios interactúen con el sistema operativo.
- Puede ser una interfaz gráfica (GUI) o una interfaz de línea de comandos (CLI).

### **Gestor de redes**

- Facilita la comunicación entre dispositivos y redes.
- Implementa protocolos de red y maneja conexiones y transferencias de datos.

### **Seguridad y protección**

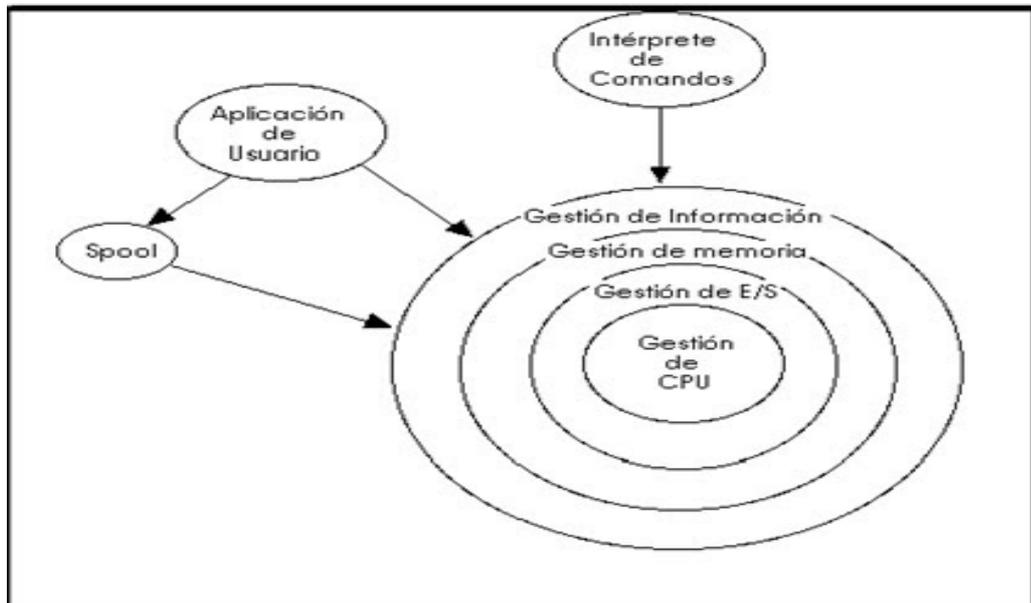
- Implementa mecanismos para proteger el sistema y los datos contra accesos no autorizados.
- Controla la autenticación y autorización de usuarios y aplicaciones.

### **3.2.2 Estructura de un sistema operativo**

Los sistemas operativos pueden estructurarse de diferentes maneras, dependiendo de cómo se organizan sus componentes y cómo interactúan entre sí. Las estructuras más comunes incluyen:

**Figura 14**

*Estructura de un Sistema Operativo*



### **Monolítico**

- En un sistema operativo monolítico, todos los componentes principales (núcleo, gestores de memoria, dispositivos, archivos, etc.) están integrados en un solo bloque de código.
- Ejemplos: **Unix, Linux.**
- Ventajas: alto rendimiento y eficiencia.
- Desventajas: mayor complejidad y dificultad para depurar y mantener.

### **Microkernel**

- En un sistema operativo de microkernel, solo las funciones básicas del núcleo (como la gestión de procesos y la comunicación interprocesos) se ejecutan en el núcleo. Otras funciones se ejecutan en el espacio de usuario como servicios.
- Ejemplos: **Mach, QNX.**

- Ventajas: mayor modularidad y seguridad.
- Desventajas: posible pérdida de rendimiento debido a la sobrecarga de comunicación entre componentes.

### **Modular**

- Un sistema operativo modular combina características de los sistemas monolíticos y de microkernel. El núcleo puede cargar y descargar módulos según sea necesario.
- Ejemplos: **Linux** (con módulos de kernel), **Solaris**.
- Ventajas: flexibilidad y capacidad de actualización dinámica.
- Desventajas: puede ser más complejo de implementar y gestionar.

### **Híbrido**

- Los sistemas operativos híbridos combinan características de diferentes estructuras para aprovechar sus ventajas y mitigar sus desventajas.
- Ejemplos: **Windows NT** (combina características de microkernel y monolítico), **macOS** (basado en un núcleo híbrido de Mach y BSD).
- Ventajas: equilibrio entre rendimiento, modularidad y seguridad.
- Desventajas: mayor complejidad en diseño e implementación.

### 3.2.3 Ejemplos de sistemas operativos

**Figura 15**

*Ejemplos de sistemas operativos*



#### **Windows**

- Desarrollado por Microsoft, Windows es uno de los sistemas operativos más populares para computadoras personales.
- Utiliza una arquitectura híbrida y ofrece una interfaz gráfica de usuario (GUI).
- Es conocido por su compatibilidad con una amplia variedad de aplicaciones y hardware.

#### **Linux**

- Linux es un sistema operativo de código abierto basado en Unix.
- Utiliza una arquitectura modular y es conocido por su flexibilidad y estabilidad.
- Se utiliza ampliamente en servidores, sistemas embebidos y dispositivos móviles (a través de Android).

#### **macOS:**

- Desarrollado por Apple, macOS es el sistema operativo utilizado en las computadoras Mac.
- Utiliza una arquitectura híbrida basada en Mach y BSD.

- Ofrece una interfaz gráfica de usuario avanzada y es conocido por su diseño y facilidad de uso.

## **Android**

- Android es un sistema operativo móvil desarrollado por Google, basado en el kernel de Linux.
- Utiliza una arquitectura modular y es conocido por su flexibilidad y personalización.
- Es el sistema operativo más utilizado en dispositivos móviles como teléfonos inteligentes y tabletas.

## **iOS**

- Desarrollado por Apple, iOS es el sistema operativo utilizado en dispositivos móviles como el iPhone y el iPad.
- Utiliza una arquitectura similar a macOS, con un núcleo basado en Mach y BSD.
- Es conocido por su rendimiento, seguridad y integración con el ecosistema de Apple.

Esta diversidad de arquitecturas y ejemplos demuestra cómo los sistemas operativos se adaptan a diferentes necesidades y entornos, proporcionando las bases para el funcionamiento eficiente y seguro de los dispositivos informáticos.

### 3.3. Gestión de procesos

#### 3.3.1 Concepto de proceso y subprocesso (Hilos)

**Figura 16**

*Gestión de procesos*



#### **Proceso**

- Un proceso es una instancia de un programa en ejecución. Es la unidad básica de trabajo en un sistema operativo, y contiene el código del programa, los datos que manipula, y el estado de ejecución.
- Cada proceso tiene su propio espacio de direcciones en la memoria, lo que incluye el código ejecutable, datos, pila, y otras áreas de memoria asignadas.
- Los procesos son gestionados por el núcleo del sistema operativo, que se encarga de su creación, ejecución, y terminación.

#### **Subproceso (Hilo)**

- Un hilo, o subprocesso, es la unidad más pequeña de procesamiento que puede ser ejecutada por un núcleo del

sistema operativo. Un proceso puede tener múltiples hilos ejecutándose concurrentemente.

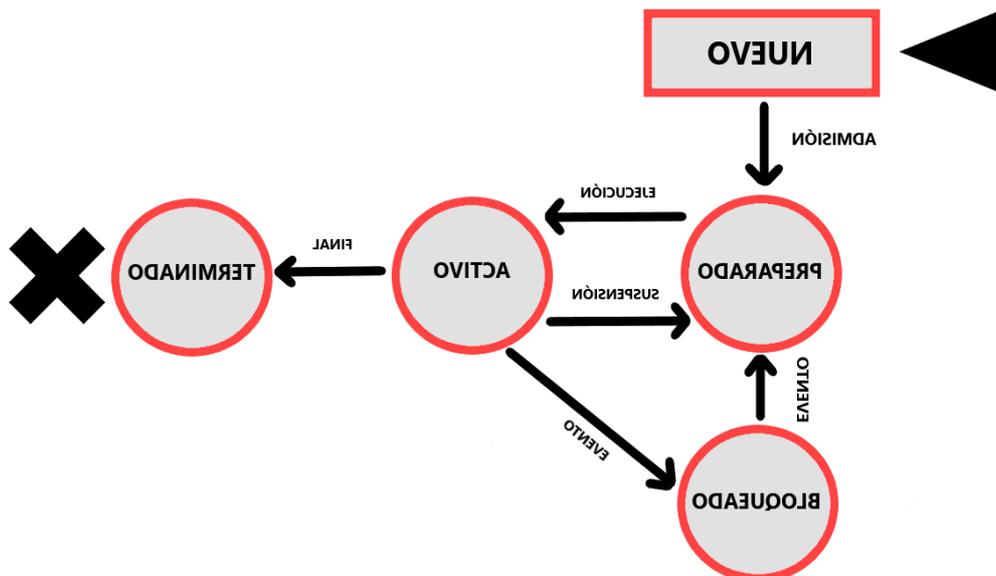
- Todos los hilos de un mismo proceso comparten el mismo espacio de direcciones, lo que incluye el código del programa y los datos. Sin embargo, cada hilo tiene su propia pila y registros.
- La utilización de hilos permite la ejecución concurrente de tareas dentro del mismo proceso, mejorando la eficiencia y la capacidad de respuesta.

### 3.3.2 Estados de un proceso

Un proceso pasa por varios estados durante su ciclo de vida. Los estados principales son:

**Figura 17**

*Estados de un proceso*



### **Nuevo (New)**

- El proceso se ha creado, pero aún no está listo para ejecutarse. Se están asignando recursos y configurando su entorno.

### **Listo (Ready)**

- El proceso está en espera de ser asignado a la CPU. Está listo para ejecutarse en cuanto se le asigne el tiempo de procesador.

### **Ejecutando (Running)**

- El proceso está siendo ejecutado por la CPU. Solo un proceso puede estar en este estado en un núcleo de CPU en un momento dado.

### **Bloqueado (Blocked) o esperando (Waiting)**

- El proceso no puede continuar su ejecución hasta que ocurra algún evento, como la finalización de una operación de entrada/salida o la recepción de un recurso.

### **Terminado (Terminated)**

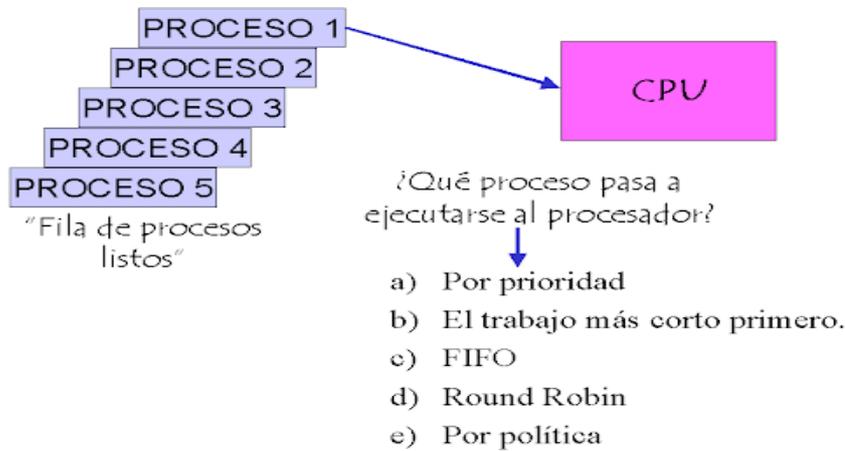
- El proceso ha completado su ejecución o ha sido finalizado por el sistema operativo. Sus recursos están siendo liberados.

### **3.3.3 Planificación de procesos: Algoritmos y políticas**

La planificación de procesos es el mecanismo mediante el cual el sistema operativo decide qué proceso debe ejecutarse en un momento dado. Los principales algoritmos de planificación incluyen:

## Figura 18

Planificación de procesos: algoritmos y políticas



### Primero en llegar, primero en ser atendido (FCFS):

- Los procesos se ejecutan en el orden en que llegan. Simple pero puede llevar a problemas de "convoy effect" donde procesos largos bloquean a procesos cortos.

### Por prioridad

- Cada proceso se le asigna una prioridad, y el proceso con la mayor prioridad se ejecuta primero. Puede ser estática o dinámica.

### Round Robin (RR)

- Cada proceso recibe un tiempo de CPU fijo (quantum) de manera circular. Es justo y adecuado para sistemas interactivos.

### **Shortest Job Next (SJN) o Shortest Job First (SJF)**

- El proceso con el tiempo de ejecución más corto se ejecuta primero. Puede ser no apropiativo o apropiativo (Shortest Remaining Time First, SRTF).

### **Multinivel por Colas (MLQ)**

- Los procesos se dividen en diferentes colas basadas en categorías como prioridad o tipo de proceso, y cada cola puede tener su propio algoritmo de planificación.

### **Multinivel por Colas con Retroalimentación (MLFQ)**

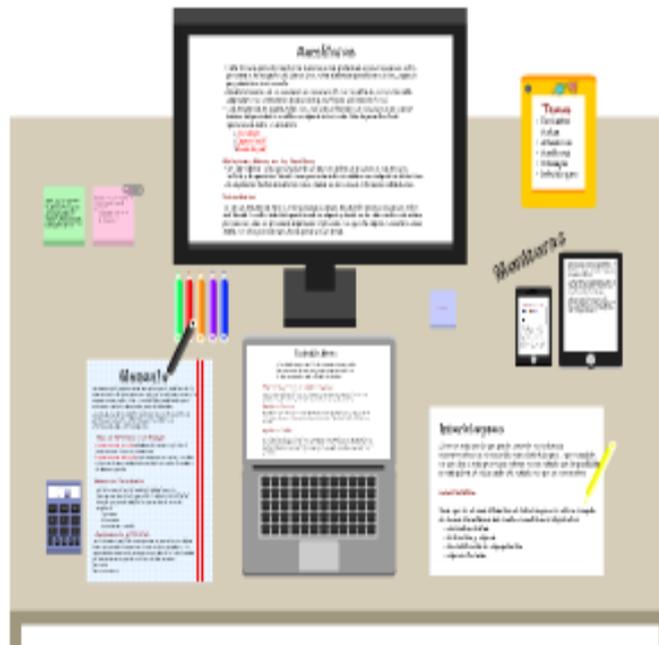
- Similar a MLQ, pero los procesos pueden moverse entre colas basadas en su comportamiento y tiempo de ejecución. Proporciona una mayor equidad y eficiencia.

### **3.3.4 Sincronización y comunicación entre procesos**

La sincronización y comunicación entre procesos son fundamentales para garantizar la correcta ejecución concurrente y evitar problemas como condiciones de carrera. Los mecanismos incluyen:

## Figura 19

### Sincronización y comunicación entre procesos



## Semáforos

- Variables especiales utilizadas para controlar el acceso a recursos compartidos. Pueden ser binarios (0 o 1) o contar (cualquier entero no negativo).

## Monitores

- Estructuras de alto nivel que proporcionan un mecanismo de exclusión mutua y permiten la sincronización mediante variables de condición.

## Bloqueos (Locks)

- Permiten que un proceso o hilo adquiera el acceso exclusivo a un recurso. Los bloqueos pueden ser simples, de lectura/escritura, recursivos, etc.

## Comunicación entre procesos (IPC)

- Mecanismos como tuberías (pipes), colas de mensajes, memoria compartida y sockets permiten que los procesos se comuniquen y coordinen entre sí.

## Condiciones de carrera

- Ocurren cuando dos o más procesos acceden a recursos compartidos y tratan de cambiar su estado al mismo tiempo. Es crucial utilizar mecanismos de sincronización para prevenir estas situaciones.

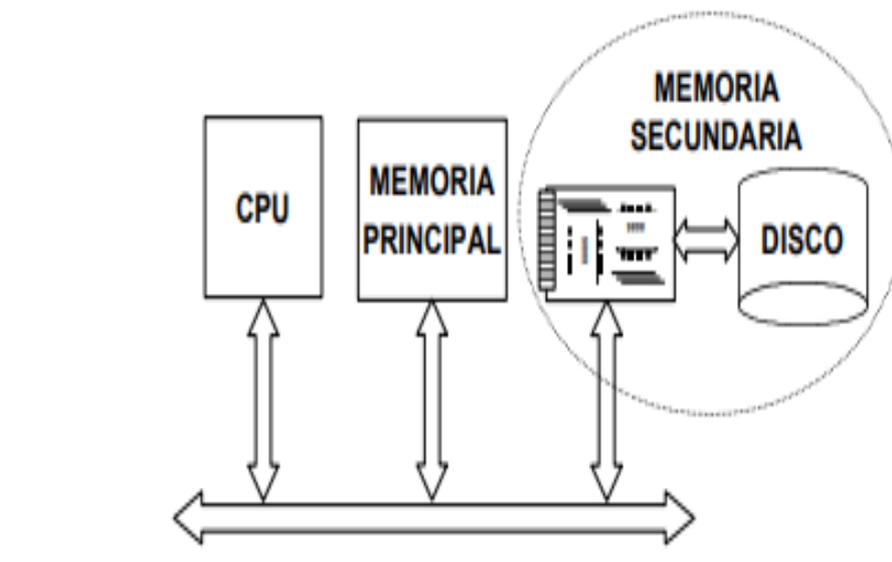
Estos mecanismos aseguran que los procesos puedan trabajar juntos de manera efectiva y segura, manteniendo la integridad de los datos y optimizando el rendimiento del sistema.

## 3.4. Gestión de memoria

### 3.4.1 Memoria primaria y secundaria

#### Figura 20

*Gestión de memoria*



## Memoria primaria

- También conocida como memoria principal o RAM (Random Access Memory), es la memoria de acceso rápido utilizada por la CPU para almacenar datos y programas que están en uso activo.
- La memoria primaria es volátil, lo que significa que pierde su contenido cuando se apaga el sistema.
- Es crucial para el rendimiento del sistema, ya que permite el acceso rápido a los datos necesarios para la ejecución de programas.

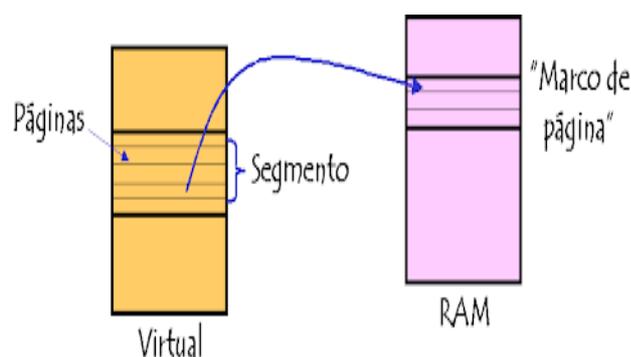
## Memoria secundaria

- La memoria secundaria se refiere a dispositivos de almacenamiento no volátiles, como discos duros (HDD), unidades de estado sólido (SSD), CD, DVD, y cintas.
- Se utiliza para almacenar datos y programas de manera permanente.
- La memoria secundaria es más lenta que la memoria primaria, pero proporciona una gran capacidad de almacenamiento.

### 3.4.2 Paginación y segmentación

#### Figura 21

*Paginación y segmentación*



## **Paginación**

- La paginación es una técnica de gestión de memoria que divide el espacio de direcciones de un proceso en bloques de tamaño fijo llamados páginas. La memoria física se divide en marcos de página del mismo tamaño.
- Cuando un proceso necesita una página, el sistema operativo la carga en un marco de página disponible en la memoria física.
- La paginación permite la memoria virtual, donde el espacio de direcciones virtuales puede ser mayor que la memoria física disponible.
- Las tablas de páginas se utilizan para traducir direcciones virtuales a direcciones físicas.

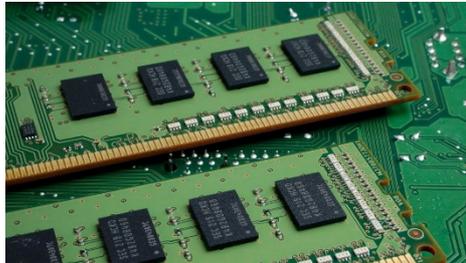
## **Segmentación**

- La segmentación divide el espacio de direcciones de un proceso en segmentos de tamaño variable, que pueden corresponder a estructuras lógicas como funciones, objetos, o datos.
- Cada segmento tiene una base y un límite, y las direcciones virtuales se traducen utilizando estos valores.
- La segmentación facilita la protección y el aislamiento de los segmentos, permitiendo que diferentes segmentos tengan diferentes permisos de acceso.
- Es útil para la organización lógica de programas y para la implementación de políticas de protección y compartición de memoria.

### 3.4.3 Memoria virtual

#### Figura 22

*Memoria virtual*



#### Memoria virtual

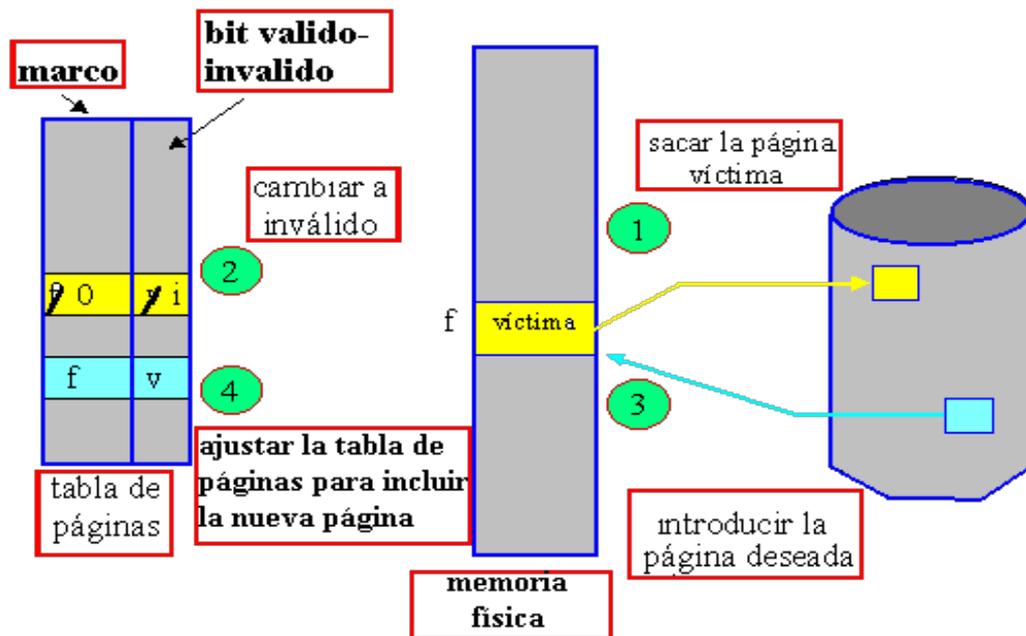
- La memoria virtual es una técnica que permite que el espacio de direcciones virtual de un proceso sea mayor que la memoria física disponible.
- Utiliza la paginación para cargar solo las partes necesarias de un programa en la memoria física, manteniendo el resto en la memoria secundaria.
- Beneficios de la memoria virtual:
  - **Expansión de la memoria:** Permite que los programas utilicen más memoria de la que físicamente está disponible.
  - **Protección:** Cada proceso tiene su propio espacio de direcciones virtuales, aislado de otros procesos.
  - **Flexibilidad:** Facilita la ejecución de programas grandes y complejos y mejora el rendimiento mediante la gestión eficiente de la memoria.
- Los mecanismos como la paginación por demanda y los algoritmos de reemplazo de páginas son utilizados para gestionar la memoria virtual.
-

### 3.4.4 Algoritmos de gestión de memoria

#### Algoritmos de reemplazo de páginas

**Figura 23**

*Algoritmos de gestión de memoria*



#### **FIFO (First-In-First-Out)**

- Las páginas se reemplazan en el orden en que se cargaron en la memoria. La página que ha estado más tiempo en la memoria es la primera en ser reemplazada.
- Sencillo de implementar, pero puede llevar al problema de anomalías FIFO (anomalía de Belady).

#### **LRU (Least Recently Used)**

- Reemplaza la página que no ha sido utilizada por el mayor tiempo. Se basa en la premisa de que las páginas utilizadas recientemente seguirán siendo utilizadas.

- Requiere un seguimiento preciso del uso de las páginas, lo cual puede ser costoso en términos de recursos.

### **LFU (Least Frequently Used)**

- Reemplaza la página que ha sido utilizada con menor frecuencia. Se basa en la idea de que las páginas menos utilizadas son menos probables de ser necesarias en el futuro.
- Puede llevar a situaciones en las que las páginas antiguas que ya no son necesarias permanezcan en la memoria.

### **NRU (Not Recently Used)**

- Divide las páginas en cuatro clases según su uso y modificación recientes. Reemplaza una página de la clase con la menor prioridad.
- Es una aproximación menos precisa pero más simple que LRU.

### **Algoritmo de reloj (Clock)**

- Utiliza un puntero que recorre las páginas en un orden circular. Si una página no ha sido referenciada recientemente, se reemplaza; de lo contrario, se marca como referenciada y se sigue al siguiente.
- Combina eficiencia y simplicidad, y es una variación de LRU.

## **Algoritmos de asignación de memoria**

### **First Fit:**

- Asigna el primer bloque de memoria libre que es lo suficientemente grande para el proceso.
- Rápido, pero puede llevar a la fragmentación externa.

### **Best Fit:**

- Asigna el bloque libre más pequeño que es lo suficientemente grande para el proceso.
- Minimiza el desperdicio de memoria, pero puede ser más lento y llevar a la fragmentación.

### **Worst Fit:**

- Asigna el bloque libre más grande. La idea es dejar bloques libres grandes para futuros procesos.
- Puede resultar en fragmentación interna.

La gestión eficiente de la memoria es crucial para el rendimiento y la estabilidad de los sistemas operativos. La elección del algoritmo adecuado depende de las necesidades específicas del sistema y de las características de las cargas de trabajo.

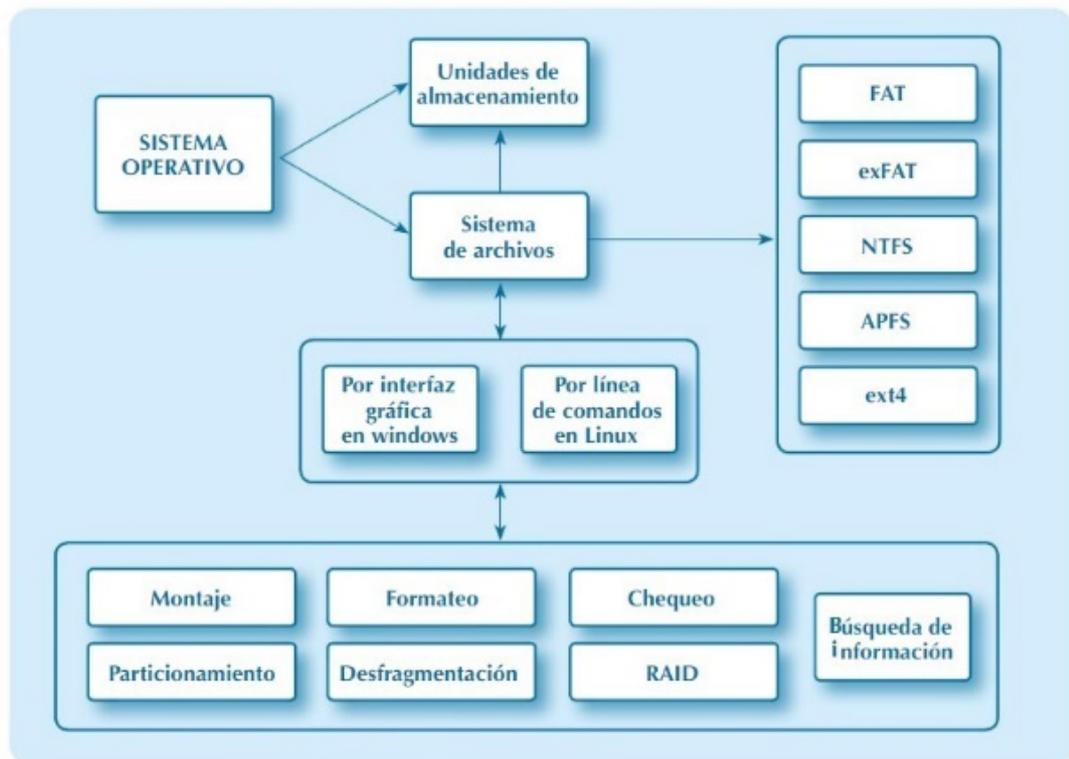
## **3.5. Gestión de almacenamiento**

### **3.5.1 Sistemas de archivos**

Un sistema de archivos organiza y gestiona cómo se almacenan, acceden y gestionan los datos en un dispositivo de almacenamiento. Los sistemas de archivos proporcionan una estructura jerárquica para la organización de los archivos y directorios.

**Figura 24**

*Gestión de almacenamiento*



**Tipos de sistemas de archivos:**

- **FAT (File Allocation Table):** Utilizado en sistemas antiguos y dispositivos de almacenamiento extraíbles. Es sencillo pero limitado en características y tamaño de archivo.
- **NTFS (New Technology File System):** Utilizado por Windows. Soporta grandes volúmenes de almacenamiento, permisos avanzados y características de recuperación.
- **ext (Extended File System):** Utilizado por sistemas Linux. Las versiones incluyen ext2, ext3 y ext4, con mejoras en rendimiento y capacidad.
- **HFS+ (Hierarchical File System Plus):** Utilizado por macOS hasta macOS High Sierra. Reemplazado por APFS (Apple File System) en versiones más recientes.

- **APFS (Apple File System):** Optimizado para almacenamiento flash y SSD, ofrece mayor eficiencia, seguridad y soporte para snapshots.
- **Btrfs (B-tree File System):** Utilizado en Linux, ofrece características avanzadas como snapshots, compresión y balanceo de carga.

### Funciones de un sistema de archivos

- **Almacenamiento y recuperación de archivos:** Manejo de la creación, lectura, escritura y eliminación de archivos.
- **Organización de directorios:** Gestión de estructuras jerárquicas de directorios para organizar archivos.
- **Metadatos:** Almacenamiento de información sobre los archivos, como tamaño, permisos y fechas de modificación.
- **Seguridad:** Implementación de permisos y controles de acceso para proteger archivos y directorios.
- **Integridad:** Mecanismos para garantizar la coherencia y recuperación de datos en caso de fallos.

### 3.5.2 Estructuras de almacenamiento y acceso a datos

**Figura 25**

*Estructura de almacenamiento acceso a datos*



## **Bloques de almacenamiento**

- Los datos se almacenan en bloques de tamaño fijo, que son la unidad básica de almacenamiento en discos.
- La gestión eficiente de bloques es crucial para el rendimiento del sistema de archivos.

## **Inodos:**

- En sistemas como ext de Linux, los inodos almacenan la información sobre archivos y directorios, como metadatos y punteros a bloques de datos.
- Permiten una gestión eficiente y rápida de archivos.

## **Estructuras de índice**

- **Árboles B y B+:** Utilizados para organizar y buscar datos rápidamente en sistemas de archivos y bases de datos.
- **Mapas de Bits:** Utilizados para gestionar bloques libres y ocupados en el sistema de archivos.
- **Tablas de Asignación:** FAT es un ejemplo de tabla de asignación que mapea el uso de bloques en el disco.

## **Journaling**

- Algunos sistemas de archivos como NTFS, ext3 y ext4 utilizan journaling para mantener un registro de cambios pendientes. Esto mejora la recuperación de datos en caso de fallos del sistema.

### 3.5.3 Seguridad y protección de datos

**Figura 26**

*Seguridad y protección de datos*



#### **Permisos de archivos**

- Los sistemas de archivos implementan permisos para controlar quién puede leer, escribir o ejecutar un archivo.
- Los permisos se pueden definir a nivel de usuario, grupo y otros.

#### **Encriptación**

- La encriptación de archivos y sistemas de archivos protege los datos contra accesos no autorizados.
- Ejemplos incluyen BitLocker en Windows y eCryptfs en Linux.

#### **Control de acceso**

- Las listas de control de acceso (ACL) proporcionan un mecanismo detallado para definir permisos de acceso a archivos y directorios.

- Los sistemas como NTFS soportan ACLs avanzadas para una gestión granular de permisos.

### **Copias de seguridad**

- La creación regular de copias de seguridad protege los datos contra pérdidas debido a fallos del sistema, errores humanos o ataques maliciosos.
- Las copias de seguridad pueden ser completas, incrementales o diferenciales.

### **3.5.4 Gestión de dispositivos de almacenamiento**

#### **Figura 27**

*Gestión de dispositivos de almacenamiento*



### **Controladores de dispositivos**

- Los controladores de dispositivos permiten que el sistema operativo se comunice con los dispositivos de almacenamiento, gestionando operaciones de entrada/salida.

- Los controladores son específicos del hardware y proporcionan una interfaz estándar para el sistema operativo.

### **RAID (Redundant Array of Independent Disks)**

- RAID es una tecnología que combina múltiples discos duros en una sola unidad lógica para mejorar el rendimiento, la redundancia o ambas.
- Tipos comunes de RAID incluyen:
  - **RAID 0:** Distribuye los datos entre varios discos para mejorar el rendimiento (sin redundancia).
  - **RAID 1:** Duplica los datos en dos discos para mayor redundancia (mirror).
  - **RAID 5:** Distribuye los datos y la paridad entre tres o más discos para equilibrio entre rendimiento y redundancia.
  - **RAID 6:** Similar a RAID 5, pero con doble paridad para mayor redundancia.

### **Sistemas de almacenamiento en red**

- **NAS (Network Attached Storage):** Dispositivo de almacenamiento dedicado que proporciona acceso a archivos a través de una red.
- **SAN (Storage Area Network):** Red de alta velocidad que conecta múltiples dispositivos de almacenamiento a servidores. Proporciona acceso a bloques de datos en lugar de archivos.

### **Optimización y mantenimiento**

- **Desfragmentación:** Reorganiza los bloques de datos para mejorar el rendimiento de lectura/escritura.

- **Balanceo de Carga:** Distribuye el acceso a datos entre múltiples dispositivos de almacenamiento para evitar cuellos de botella.
- **Monitoreo y Diagnóstico:** Herramientas y técnicas para supervisar la salud y el rendimiento de los dispositivos de almacenamiento y detectar fallos potenciales.

La gestión eficiente del almacenamiento es crucial para el rendimiento, la seguridad y la integridad de los datos en un sistema operativo. Cada componente y técnica contribuye a garantizar que los datos se manejen de manera efectiva y segura.

### 3.6. Gestión de entrada/salida (E/S)

#### 3.6.1 Controladores de dispositivos

Los controladores de dispositivos son programas de software que permiten que el sistema operativo interactúe con los dispositivos de hardware. Actúan como intermediarios entre el hardware y el software del sistema, proporcionando una interfaz estándar para acceder y controlar los dispositivos.

#### Figura 28

*Gestión de entrada/salida(E/S)*



## Funciones de los controladores de dispositivos

- **Comunicación:** Facilitan la comunicación entre el sistema operativo y el hardware.
- **Control:** Gestionan el funcionamiento del hardware, configurando y controlando los dispositivos.
- **Abstracción:** Proporcionan una capa de abstracción que permite a las aplicaciones acceder a los dispositivos sin conocer los detalles específicos del hardware.
- **Manejo de Errores:** Detectan y gestionan los errores en el funcionamiento de los dispositivos.

## Tipos de controladores de dispositivos

- **Controladores de E/S Básicos:** Manejan dispositivos simples como teclados y ratones.
- **Controladores de Almacenamiento:** Gestionan discos duros, SSDs, CD/DVD, etc.
- **Controladores de Red:** Gestionan adaptadores de red y conexiones de red.
- **Controladores de Video:** Controlan tarjetas gráficas y monitores.

### 3.6.2 Interrupciones y DMA

#### Figura 29

#### *Interrupciones y DMA*



## Interrupciones

- Las interrupciones son señales enviadas por dispositivos de hardware o software para alertar a la CPU de que un evento necesita atención inmediata.
- Cuando se produce una interrupción, la CPU pausa la ejecución del programa actual y ejecuta una rutina de servicio de interrupción (ISR) para manejar el evento.
- Las interrupciones permiten un manejo eficiente y rápido de eventos de E/S sin necesidad de que la CPU consulte constantemente el estado de los dispositivos (polling).

## Tipos de interrupciones

- **Interrupciones de Hardware:** Generadas por dispositivos de hardware, como teclados, ratones, y discos duros.
- **Interrupciones de Software:** Generadas por programas, como señales de errores o llamadas al sistema.
- **Interrupciones de Temporizador:** Utilizadas por el sistema operativo para tareas de mantenimiento y planificación.

## DMA (Direct Memory Access)

- DMA es un método que permite a los dispositivos de E/S transferir datos directamente a la memoria principal sin intervención continua de la CPU.
- Un controlador de DMA gestiona la transferencia de datos, liberando a la CPU para realizar otras tareas mientras se completa la transferencia de E/S.

## Ventajas del DMA

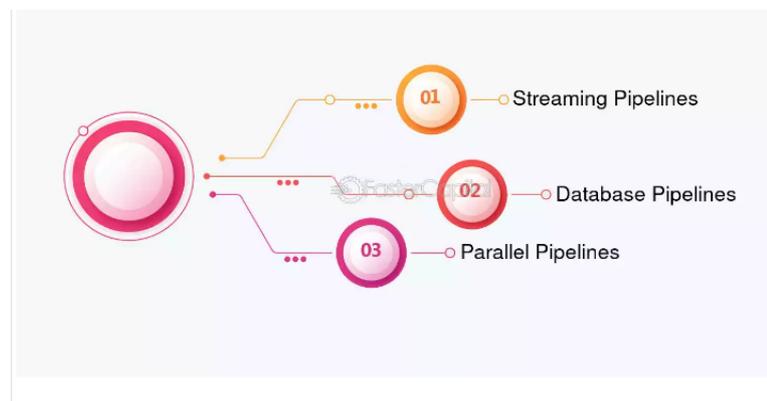
- **Eficiencia:** Reduce la carga de trabajo de la CPU al manejar transferencias de datos de manera autónoma.
- **Velocidad:** Permite transferencias de datos más rápidas entre dispositivos y memoria.
- **Concurrente:** Permite que la CPU y los dispositivos de E/S operen simultáneamente, mejorando el rendimiento del sistema.

### 3.6.3 Optimización de la E/S

La optimización de las operaciones de E/S es crucial para mejorar el rendimiento y la eficiencia del sistema. Existen varias técnicas y estrategias para optimizar la E/S:

#### Figura 30

Optimización de la E/S



#### Buffering

- El uso de buffers permite almacenar temporalmente datos durante la transferencia entre dispositivos y memoria. Esto reduce la latencia y permite una transferencia más eficiente.

- **Doble Buffering:** Utiliza dos buffers para permitir la escritura en uno mientras se lee del otro, aumentando la eficiencia.

### Caching

- El caching utiliza una memoria más rápida para almacenar temporalmente datos frecuentemente accedidos, reduciendo el tiempo de acceso y mejorando el rendimiento.
- **Cache de Disco:** Almacena temporalmente bloques de datos leídos/escritos en disco para acceso rápido.
- **Cache de Páginas:** Almacena páginas de memoria virtual en una memoria más rápida para reducir el tiempo de acceso.

### Spooling

- El spooling (Simultaneous Peripheral Operation On-Line) permite que las operaciones de E/S se manejen en segundo plano, permitiendo que la CPU continúe con otras tareas.
- Comúnmente utilizado en impresoras y dispositivos de salida, donde los datos se almacenan en un buffer o archivo temporal antes de ser procesados por el dispositivo.

### Planificación de E/S

- Los algoritmos de planificación de E/S determinan el orden en que se atienden las solicitudes de E/S para optimizar el rendimiento.
- **FCFS (First-Come, First-Served):** Atiende las solicitudes en el orden en que llegan.
- **SSTF (Shortest Seek Time First):** Atiende las solicitudes que requieren el menor movimiento del cabezal del disco.

- **SCAN y C-SCAN:** El cabezal del disco se mueve de un extremo al otro atendiendo solicitudes en el camino, con C-SCAN regresando al inicio sin atender solicitudes en el retorno.

### **RAID (Redundant Array of Independent Disks)**

- RAID combina múltiples discos duros para mejorar el rendimiento y/o la redundancia. Diferentes niveles de RAID ofrecen beneficios específicos en términos de velocidad y protección de datos.

### **Compresión y descompresión**

- La compresión de datos reduce la cantidad de datos transferidos y almacenados, mejorando la eficiencia de E/S. La descompresión se realiza cuando los datos se leen o utilizan.

### **Prefetching**

- El prefetching carga anticipadamente datos en memoria anticipando su uso futuro, reduciendo la latencia de acceso.

La combinación de estas técnicas y estrategias de optimización permite que los sistemas operativos gestionen las operaciones de E/S de manera más eficiente, mejorando el rendimiento general del sistema y la experiencia del usuario.

### 3.7. Gestión de recursos

#### 3.7.1 Recursos del sistema: CPU, Memoria, dispositivos de E/S

**Figura 31**

*Gestión de recursos*



#### **CPU (Unidad Central de Procesamiento)**

- La CPU es el componente principal del sistema que ejecuta instrucciones de programas. Los recursos de la CPU incluyen los núcleos de procesamiento, el tiempo de CPU, y la caché de la CPU.

- La gestión de la CPU implica la planificación y asignación de tiempo de CPU a los procesos, la gestión de interrupciones, y la optimización del uso de la CPU para maximizar el rendimiento.

### **Memoria**

- La memoria incluye la memoria primaria (RAM) y la memoria secundaria (discos duros, SSDs). La gestión de memoria implica la asignación de espacio de memoria a los procesos, el manejo de la memoria virtual, y la optimización del uso de la memoria para evitar la fragmentación y maximizar la eficiencia.
- Técnicas como la paginación, la segmentación, y el uso de algoritmos de reemplazo de páginas son esenciales para una gestión eficiente de la memoria.

### **Dispositivos de E/S (Entrada/Salida)**

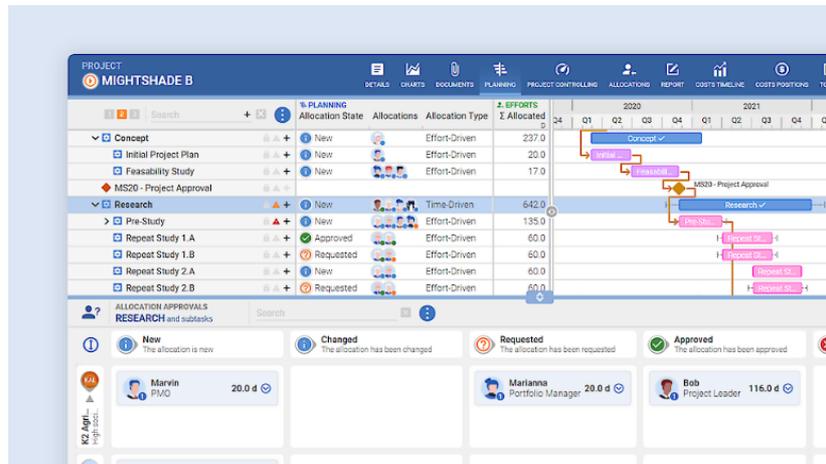
- Los dispositivos de E/S incluyen teclados, ratones, discos duros, impresoras, y dispositivos de red. La gestión de E/S implica la coordinación de las operaciones de entrada y salida, el manejo de interrupciones, y la optimización de la transferencia de datos para minimizar la latencia y maximizar el rendimiento.
- Los controladores de dispositivos, el uso de DMA, y los algoritmos de planificación de E/S son cruciales para una gestión efectiva de los dispositivos de E/S.

### **3.7.2 Asignación y liberación de recursos**

La gestión eficiente de los recursos del sistema es fundamental para garantizar que los procesos tengan acceso a los recursos necesarios cuando los necesiten y que los recursos no se desperdicien.

## Figura 32

### Asignación y liberación de recursos



### Asignación de recursos

- **Planificación de la CPU:** Los algoritmos de planificación (como FCFS, SJF, y Round Robin) determinan el orden en que los procesos reciben tiempo de CPU.
- **Asignación de Memoria:** La memoria se asigna a los procesos utilizando técnicas como la paginación, la segmentación, y la memoria virtual.
- **Asignación de E/S:** Los recursos de E/S se asignan según las solicitudes de los procesos y utilizando algoritmos de planificación de E/S para optimizar el rendimiento.

### Liberación de recursos

- Los recursos deben liberarse cuando ya no son necesarios para que otros procesos puedan utilizarlos.
- **Terminación de Procesos:** Cuando un proceso termina, todos los recursos asignados a él (CPU, memoria, E/S) deben liberarse.
- **Recolección de Basura:** En sistemas con administración automática de memoria, como los entornos de ejecución de

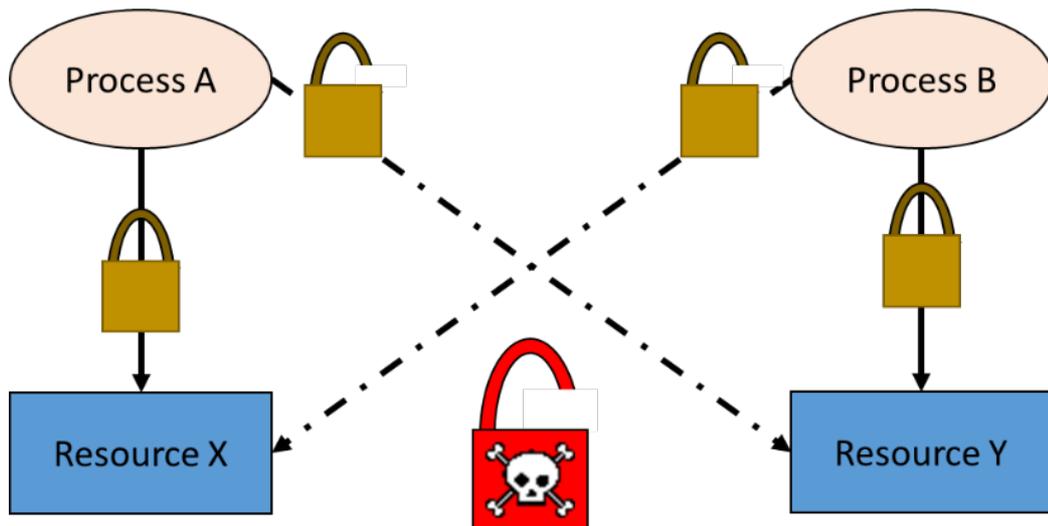
Java, la recolección de basura libera memoria ocupada por objetos que ya no son accesibles.

### 3.7.3 Prevención y manejo de Deadlocks (Interbloqueos)

Un deadlock (interbloqueo) ocurre cuando dos o más procesos se bloquean mutuamente esperando recursos que los otros procesos poseen, lo que impide que cualquiera de ellos pueda continuar.

**Figura 33**

*Prevención y manejo de Deadlocks (Interbloqueos)*



#### Condiciones para un Deadlock

- **Mutua Exclusión:** Al menos un recurso debe estar en modo no compartido, es decir, solo un proceso puede utilizar el recurso en un momento dado.
- **Espera Circular:** Debe existir una cadena de procesos donde cada proceso espera un recurso que está siendo ocupado por el siguiente proceso en la cadena.

- **Espera y Retención:** Un proceso debe estar reteniendo al menos un recurso y esperando por recursos adicionales que están siendo retenidos por otros procesos.
- **No Preempción:** Los recursos no pueden ser forzosamente retirados de los procesos que los poseen; solo pueden ser liberados voluntariamente por el proceso.

### **Prevención de Deadlocks**

- **Eliminación de una de las Condiciones Necesarias:** Diseñar el sistema para que al menos una de las cuatro condiciones necesarias para un deadlock no pueda ocurrir.
  - **Mutua Exclusión:** Utilizar recursos compartidos siempre que sea posible.
  - **Espera Circular:** Imponer un orden total en la asignación de recursos y requerir que los procesos soliciten recursos en un orden predefinido.
  - **Espera y Retención:** Requerir que los procesos soliciten todos los recursos necesarios al inicio y que liberen todos los recursos antes de solicitar nuevos.
  - **No Preempción:** Permitir la preempción de recursos, es decir, que el sistema pueda forzar la liberación de recursos.

### **Evitación de Deadlocks**

- **Algoritmo del Banquero:** Utilizado en sistemas donde los recursos se asignan de manera dinámica. Antes de asignar un recurso, el sistema simula la asignación y verifica si puede llevar al sistema a un estado inseguro.

## **Detección y recuperación de Deadlocks:**

- **Detección:** El sistema puede utilizar algoritmos para detectar la presencia de deadlocks mediante la construcción y análisis de un grafo de asignación de recursos.
- **Recuperación:** Una vez detectado un deadlock, el sistema puede intentar recuperar el control:
  - **Finalización de Procesos:** Terminar uno o más procesos involucrados en el deadlock para liberar recursos.
  - **Preempción de Recursos:** Forzar la liberación de recursos por parte de los procesos involucrados en el deadlock.

La gestión efectiva de recursos y la prevención y manejo de deadlocks son esenciales para garantizar la estabilidad y eficiencia de un sistema operativo, permitiendo que los procesos se ejecuten sin interrupciones y que los recursos del sistema se utilicen de manera óptima.

# CAPÍTULO 4

Dispositivos de un sistema informático

4

# Capítulo 4

## DISPOSITIVOS DE UN SISTEMA INFORMÁTICO

### 4.1. Conceptos básicos

**Sistema Informático:** Un conjunto de componentes interrelacionados que trabajan juntos para procesar datos y producir información. Incluye hardware, software, datos, usuarios y procedimientos.

#### Figura 34

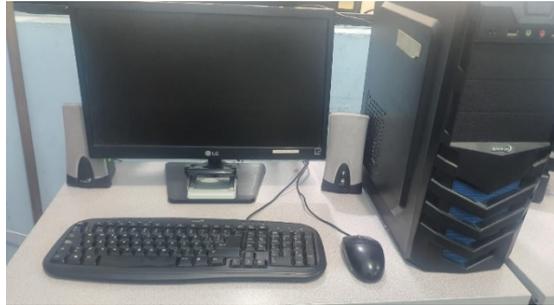
*Sistema básico informático.*



**Hardware:** Los componentes físicos de un sistema informático que se pueden tocar y ver, como el monitor, teclado, disco duro, etc.

## **Figura 35**

*Hardware*



**Software:** Programas y aplicaciones que dirigen al hardware sobre cómo operar y procesar datos.

**Datos:** Información que el sistema informático procesa para generar resultados útiles.

**Usuarios:** Personas que utilizan el sistema informático.

**Procedimientos:** Instrucciones y reglas para usar el hardware y el software correctamente.

## **4.2. Dispositivos – Hardware**

### **4.2.1. Dispositivos de entrada**

## **Figura 36**

*Ratón*



- **Teclado:** Permite al usuario ingresar datos y comandos al sistema.
- **Ratón:** Dispositivo apuntador que permite seleccionar y manipular objetos en la pantalla.
- **Escáner:** Convierte imágenes y documentos físicos en formato digital.
- **Micrófono:** Captura sonido y lo convierte en datos digitales.

#### 4.2.2. Dispositivos de salida

##### Figura 37

##### *Impresora*



- **Monitor:** Muestra información y datos procesados en forma visual.
- **Impresora:** Produce una copia física de documentos y gráficos.
- **Altavoces:** Emiten sonido procesado por la computadora.

#### 4.2.3. Dispositivos de almacenamiento

- **Disco Duro:** Principal dispositivo de almacenamiento interno que guarda datos permanentemente.
- **Unidad SSD:** Almacenamiento de estado sólido que ofrece mayor velocidad que los discos duros tradicionales.
- **Memoria USB:** Dispositivo de almacenamiento portátil.
- **CD/DVD:** Medios ópticos para almacenar datos.

#### 4.2.4. Dispositivos de procesamiento

- **CPU (Unidad Central de Procesamiento):** El cerebro del computador que realiza cálculos y ejecuta instrucciones.
- **GPU (Unidad de Procesamiento Gráfico):** Maneja los gráficos y las imágenes.

#### 4.3. Dispositivos de red

**Figura 38**

*Tarjeta de red (NIC).*



- **Tarjeta de Red:** Permite la conexión a una red.
- **Router:** Dispositivo que dirige el tráfico de red entre diferentes dispositivos.
- **Modem:** Convierte señales digitales a analógicas y viceversa para la comunicación con Internet.

### 4.3.1. Ensamblaje de computadores

#### 4.3.1.1. Planificación

- **Identificación de necesidades:** Determinar el propósito del sistema.
- **Selección de componentes:** Elegir los componentes adecuados según las necesidades y el presupuesto.

#### 4.3.1.2. Montaje del hardware

- **Instalación de la placa base:** Colocar la placa base en la carcasa.
- **Montaje de la CPU:** Insertar la CPU en el zócalo correspondiente.
- **Instalación de la memoria RAM:** Colocar los módulos de RAM en los slots adecuados.
- **Montaje de dispositivos de almacenamiento:** Conectar discos duros y SSD.
- **Instalación de la fuente de poder:** Conectar la fuente de alimentación a la placa base y otros componentes.
- **Instalación de la GPU:** Colocar la tarjeta gráfica en su correspondiente slot.

#### 4.3.1.3. Conexión de periféricos

- **Conexión de dispositivos de entrada/salida:** Conectar teclado, ratón, monitor, etc.
- **Configuración de dispositivos de red:** Conectar cables de red o configurar dispositivos inalámbricos.

#### 4.3.1.4. Verificación y pruebas

- **Encender el sistema:** Asegurarse de que todos los componentes funcionen correctamente.
- **BIOS/UEFI:** Configurar las opciones iniciales del sistema.
- **Instalación de Sistema Operativo:** Instalar y configurar el sistema operativo elegido.

#### Pasos para el ensamblaje

##### *Preparación*

- **Área de Trabajo:** Asegúrate de trabajar en un área limpia y libre de electricidad estática. Usa una pulsera antiestática.
- **Comprobación de Componentes:** Verifica que tienes todos los componentes necesarios.

##### *Instalación de la fuente de alimentación (PSU)*

- Coloca la fuente de alimentación en el gabinete y asegúrala con tornillos.
- Asegúrate de que el ventilador de la PSU esté orientado correctamente para una adecuada ventilación.

##### *Instalación de la placa base*

- Coloca los separadores (standoffs) en el gabinete alineados con los agujeros de la placa base.
- Inserta la placa base en el gabinete y atorníllala a los separadores.

### ***Instalación del CPU***

- Abre el socket de la CPU en la placa base.
- Coloca la CPU en el socket, asegurándote de alinear los pines correctamente.
- Cierra el mecanismo de retención del socket.

### ***Aplicación de Pasta Térmica y Disipador de Calor***

- Aplica una pequeña cantidad de pasta térmica en el centro de la CPU.
- Coloca el disipador de calor (o el sistema de enfriamiento) sobre la CPU y asegúralo según las instrucciones del fabricante.

### ***Instalación de la memoria RAM***

- Encuentra las ranuras de RAM en la placa base.
- Abre los clips en las ranuras.
- Inserta los módulos de RAM, presionando firmemente hasta que los clips se cierren automáticamente.

### ***Instalación de almacenamiento (HDD/SSD)***

- Inserta el disco duro o SSD en la bahía correspondiente del gabinete.
- Asegúralo con tornillos.
- Conecta los cables SATA a la unidad y a la placa base.

### ***Instalación de la tarjeta gráfica (si aplica)***

- Inserta la tarjeta gráfica en la ranura PCI-Express de la placa base.
- Asegura la tarjeta gráfica con tornillos al gabinete.
- Conecta los cables de alimentación de la PSU a la tarjeta gráfica, si es necesario.

### ***Conexión de cables***

- Conecta el cable de 24 pines de la PSU a la placa base.
- Conecta el cable de 8 (o 4) pines de la PSU a la CPU.
- Conecta los cables del panel frontal del gabinete (puertos USB, audio, botones de encendido y reinicio) a la placa base.
- Organiza y asegura los cables para mejorar el flujo de aire.

### ***Comprobaciones finales***

- Verifica que todos los componentes estén correctamente instalados y asegurados.
- Revisa todas las conexiones de cables.
- Asegúrate de que no haya tornillos sueltos o componentes moviéndose dentro del gabinete.

### ***Encendido y prueba***

- Conecta el monitor, teclado y ratón.
- Enciende la computadora y entra en el BIOS para verificar que todos los componentes sean reconocidos correctamente.
- Instala el sistema operativo y los controladores necesarios.

### **Consideraciones Finales**

- **Documentación:** Consulta siempre la documentación de cada componente para detalles específicos de instalación y compatibilidad.
- **Compatibilidad:** Asegúrate de que todos los componentes sean compatibles entre sí, especialmente en términos de tipo de socket de CPU, tipo de memoria RAM y puertos de conexión.
- **Seguridad:** Trabaja con cuidado para evitar daños a los componentes sensibles.

El ensamblaje de una computadora puede ser una tarea gratificante que te permite personalizar tu sistema según tus necesidades específicas. Con paciencia y atención al detalle, podrás construir un sistema que funcione eficientemente.

## 4.4. Mantenimiento

### 4.4.1. Mantenimiento preventivo

- **Limpieza:** Mantener el hardware limpio de polvo y suciedad.
- **Actualización de Software:** Mantener el sistema operativo y el software actualizado.
- **Revisión de Componentes:** Verificar regularmente el funcionamiento de los componentes.

#### *Pasos para el mantenimiento preventivo*

##### 1. Limpieza física:

- **Apaga y desconecta:** Apaga la computadora y desconéctala de la fuente de alimentación.
- **Desmontaje del Gabinete:** Abre el gabinete para acceder a los componentes internos.
- **Eliminar el polvo:** Utiliza aire comprimido para soplar el polvo acumulado en los ventiladores, disipadores de calor, y otros componentes. Usa un paño antiestático para limpiar superficies.
- **Limpieza de periféricos:** Limpia el teclado, ratón y monitor con productos adecuados.

## 2. Revisión de componentes:

- **Inspección Visual:** Verifica que no haya cables sueltos o componentes dañados.
- **Verificación de Conexiones:** Asegúrate de que todos los cables y conexiones estén firmemente conectados.

## 3. Actualización de software:

- **Sistema operativo:** Mantén el sistema operativo actualizado con los últimos parches y actualizaciones de seguridad.
- **Controladores (Drivers):** Actualiza los controladores de hardware para asegurar compatibilidad y rendimiento óptimo.
- **Software antivirus:** Asegúrate de que el software antivirus esté actualizado y realiza un escaneo completo del sistema.

## 4. Optimización del sistema:

- **Desfragmentación (HDD):** Desfragmenta el disco duro si es un HDD (esto no es necesario para SSDs).
- **Limpieza de Archivos:** Utiliza herramientas de limpieza de disco para eliminar archivos temporales y no deseados.
- **Gestionar Programas de Inicio:** Revisa y desactiva los programas innecesarios que se inician con el sistema operativo.

## 5. Revisión de temperaturas:

- **Monitoreo:** Usa software de monitoreo para revisar las temperaturas de la CPU, GPU y otros componentes críticos.
- **Reemplazo de Pasta Térmica:** Considera reemplazar la pasta térmica de la CPU si ha pasado mucho tiempo desde la última vez que se hizo.

#### 4.4.2. Mantenimiento correctivo

- **Diagnóstico de fallas:** Identificar y solucionar problemas de hardware y software.
- **Reparación/Reemplazo de Componentes:** Sustituir componentes defectuosos.
- **Restauración del sistema:** Reinstalar el sistema operativo o software en caso de fallos graves.

#### 4.3.3. Mantenimiento predictivo

- **Monitoreo:** Usar software de monitoreo para prever posibles fallos de hardware.
- **Análisis de rendimiento:** Evaluar el rendimiento del sistema y prever actualizaciones necesarias.

### *Pasos para el mantenimiento correctivo*

#### 1. Diagnóstico de Problemas:

- **Síntomas:** Identifica los síntomas del problema (p.ej., ruidos extraños, sobrecalentamiento, fallos de arranque).
- **Herramientas de Diagnóstico:** Usa herramientas de diagnóstico de hardware y software para identificar el componente defectuoso.

#### 2. Reparación o reemplazo de componentes

- **Componentes Dañados:** Reemplaza componentes defectuosos como discos duros, fuentes de alimentación, RAM, etc.
- **Reparación de Software:** Reinstala el sistema operativo si es necesario o repara archivos del sistema usando herramientas de recuperación.

### 3. Revisión de conexiones

- **Verificación de Conexiones:** Asegúrate de que todas las conexiones internas y externas estén firmemente aseguradas.

### 4. Pruebas de funcionamiento

- **Reinicio del Sistema:** Reinicia la computadora y verifica que el problema haya sido solucionado.
- **Pruebas de Rendimiento:** Realiza pruebas de rendimiento para asegurarte de que el sistema funciona correctamente después de la reparación.

Consejos adicionales para el mantenimiento de computadoras

- **Copias de Seguridad Regulares:** Realiza copias de seguridad de tus datos importantes para evitar la pérdida de información.
- **Software de Seguridad:** Utiliza software de seguridad confiable y mantenlo actualizado.
- **Cuidado con el Entorno:** Mantén la computadora en un entorno limpio y bien ventilado para evitar el sobrecalentamiento.
- **Uso Adecuado:** Evita el uso excesivo de la computadora en entornos polvorientos o húmedos.

Realizar un mantenimiento regular y adecuado puede prevenir muchos problemas comunes y asegurar que tu computadora funcione de manera eficiente durante más tiempo.

# UNIDAD 5

Redes y comunicaciones



5



## Capítulo

# 5

## REDES Y COMUNICACIONES

### 5.1. Definición y conceptos básicos

Las redes de comunicaciones son sistemas que permiten la transferencia de datos entre diferentes dispositivos mediante enlaces físicos o inalámbricos y protocolos específicos. Los componentes esenciales de una red incluyen:

**Nodos:** Dispositivos finales como computadoras, impresoras, teléfonos inteligentes, y servidores. Enlaces: Medios físicos (cables) o inalámbricos (ondas de radio) que conectan los nodos.

**Protocolos:** Conjunto de reglas que gobiernan la comunicación y el intercambio de datos entre dispositivos. Ejemplos incluyen TCP/IP, HTTP y FTP. Las redes permiten compartir recursos, como archivos, impresoras y conexiones a Internet, entre múltiples dispositivos. La interconexión eficiente y segura de estos dispositivos es crucial para el funcionamiento de las empresas y las actividades cotidianas.

### 5.1.2 Evolución histórica de las redes y comunicaciones

**Telégrafo (1837):** Inventado por Samuel Morse, permitió la transmisión de mensajes codificados a largas distancias usando cables.

**Teléfono (1876):** Alexander Graham Bell desarrolló el teléfono, facilitando la transmisión de voz a través de cables.

**ARPANET (1969):** Creada por el Departamento de Defensa de EE. UU., fue la primera red de conmutación de paquetes, precursor de Internet.

**Desarrollo de protocolos (1970s-1980s):** Se desarrollaron los protocolos TCP/IP, que son la base de la Internet moderna.

**Expansión de internet (1990s):** Con la popularización de la World Wide Web y los navegadores web como Mosaic y Netscape, Internet se convirtió en una red global accesible al público.

**Redes inalámbricas y móviles (2000s):** La tecnología Wi-Fi y las redes móviles (2G, 3G, 4G y 5G) permitieron la conectividad sin cables, transformando la accesibilidad y el uso de las redes.

Tecnologías emergentes (2010s y más allá): El Internet de las Cosas (IoT), las redes definidas por software (SDN) y la computación en la nube están redefiniendo la infraestructura de red y su uso

### 5.1.3 Importancia en el mundo moderno

**Comunicación:** Las redes permiten la comunicación instantánea a través de correos electrónicos, videoconferencias, y redes sociales.

**Acceso a la información:** Facilitan el acceso a vastos recursos de información y educación en línea.

**Comercio y economía:** Las redes soportan el comercio electrónico, permitiendo a las empresas y consumidores interactuar globalmente.

**Eficiencia y automatización:** En sectores como la manufactura, la salud y la administración pública, mejoran la eficiencia y permiten la automatización de procesos.

**Desarrollo tecnológico:** Son la base de nuevas tecnologías como el IoT, la inteligencia artificial (IA) y el big data.

## 5.2 Fundamentos de las redes

### 5.2.1 Tipos de redes (LAN, WAN, MAN, PAN)

**LAN (Local Area Network):** Redes de área local que conectan dispositivos dentro de un área geográfica limitada, como una oficina o un edificio.

Ejemplo: Una red en una pequeña empresa que conecta computadoras, impresoras y servidores.

**WAN (Wide Area Network):** Redes de área amplia que cubren grandes distancias, conectando múltiples LANs.

Ejemplo: Internet, que conecta redes en todo el mundo.

**MAN (Metropolitan Area Network):** Redes que cubren un área metropolitana o una ciudad. Ejemplo: Una red que conecta diferentes sucursales de una empresa en una ciudad.

**PAN (Personal Area Network):** Redes de área personal que conectan dispositivos en un rango corto, como una conexión Bluetooth entre un teléfono móvil y un auricular.

### 5.2.2 Topologías de red

**Estrella:** Todos los dispositivos están conectados a un nodo central (switch o hub).

Ejemplo: La configuración típica de una red doméstica con un router en el centro.

**Bus:** Todos los dispositivos están conectados a un único cable de comunicación.

Ejemplo: Red antigua de Ethernet con cable coaxial.

**Anillo:** Cada dispositivo está conectado a dos otros dispositivos, formando un círculo cerrado. Ejemplo: Algunas redes token ring.

**Malla:** Cada dispositivo está conectado a varios otros dispositivos, ofreciendo múltiples caminos para los datos.

Ejemplo: Redes de malla en ciudades inteligentes.

### 5.2.3 Modelos de referencia (OSI y TCP/IP)

**OSI (Open Systems Interconnection):** Un modelo de referencia con siete capas que estandariza las funciones de una red:

- **Capa Física:** Transmisión de bits a través de un medio físico.
- **Capa de Enlace de Datos:** Proporciona un enlace fiable entre dos nodos.
- **Capa de Red:** Enrutamiento y direccionamiento de paquetes.
- **Capa de Transporte:** Transferencia de datos fiable y control de flujo.
- **Capa de Sesión:** Gestión de conexiones entre aplicaciones.
- **Capa de Presentación:** Traducción de datos entre el formato de la red y el formato del dispositivo.
- **Capa de Aplicación:** Servicios de red para aplicaciones (HTTP, FTP).

## **TCP/IP (Transmission Control Protocol/Internet Protocol):**

Modelo de referencia utilizado en Internet con cuatro capas

- **Capa de Enlace:** Medios físicos y protocolos de enlace de datos.
- **Capa de Internet:** Direccionamiento y enrutamiento de paquetes (IP, ICMP).
- **Capa de Transporte:** Transferencia de datos fiable o sin conexión (TCP, UDP).
- **Capa de Aplicación:** Protocolos de alto nivel (HTTP, FTP, SMTP).

### **5.3 Medios de transmisión**

#### **5.3.1 Medios guiados (cable coaxial, par trenzado, fibra óptica)**

**Cable coaxial:** Núcleo de cobre rodeado por un aislante, un conductor trenzado y una cubierta exterior. Ejemplo: Utilizado en la transmisión de televisión por cable.

**Par trenzado:** Pares de hilos de cobre trenzados entre sí. Ejemplos: UTP (Unshielded Twisted Pair): Común en redes LAN.

**STP (Shielded Twisted Pair):** Utilizado en ambientes con alta interferencia electromagnética.

**Fibra óptica:** Fibras de vidrio o plástico que transmiten datos mediante pulsos de luz. Ejemplo: Utilizado para conexiones de alta velocidad en redes WAN y backbones de Internet.

### **5.3.2 Medios no guiados (ondas de radio, microondas, infrarrojo)**

**Ondas de radio:** Comunicación inalámbrica utilizada en Wi-Fi, Bluetooth y redes celulares. Ejemplo: Conexión Wi-Fi en un hogar.

**Microondas:** Transmisión de datos a larga distancia con línea de vista directa. Ejemplo: Enlaces satelitales.

**Infrarrojo:** Comunicación a corta distancia sin obstáculos. Ejemplo: Control remoto de televisores.

## **5.4 Protocolos de comunicación**

### **5.4.1 Protocolos de capa física**

Definen las características eléctricas y mecánicas para activar, mantener y desactivar conexiones físicas.

**Ethernet:** Define la transmisión de datos en redes LAN.

USB (Universal Serial Bus): Comunicación de datos y alimentación eléctrica entre dispositivos.

### **5.4.2 Protocolos de enlace de datos**

Proporcionan un enlace fiable entre dos nodos conectados directamente.

**Ethernet:** Utilizado en redes LAN para la transmisión de datos.

PPP (Point-to-Point Protocol): Utilizado en conexiones punto a punto.

### **5.4.3 Protocolos de red (IP, ICMP)**

Proporcionan enrutamiento y direccionamiento de paquetes.

**IP (Internet Protocol):** Direccionamiento y enrutamiento de paquetes en redes IP.

**ICMP (Internet Control Message Protocol):** Gestiona mensajes de control y error en redes IP.

### **5.4.4 Protocolos de transporte (TCP, UDP)**

Proporcionan transferencia de datos fiable o sin conexión.

**TCP (Transmission Control Protocol):** Proporciona transferencia de datos fiable y control de flujo.

**UDP (User Datagram Protocol):** Proporciona transferencia de datos sin conexión y baja sobrecarga.

### **5.4.5 Protocolos de aplicación (HTTP, FTP, SMTP)**

Proporcionan servicios de red para aplicaciones.

**HTTP (Hypertext Transfer Protocol):** Utilizado para la transferencia de páginas web.

**FTP (File Transfer Protocol):** Utilizado para la transferencia de archivos.

**SMTP (Simple Mail Transfer Protocol):** Utilizado para el envío de correos electrónicos.

## **5.5 Dispositivos de red**

### **5.5.1 Hubs y switches**

**Hubs:** Dispositivos de red que retransmiten datos a todos los puertos. Ejemplo: Utilizado en redes pequeñas para conectar dispositivos.

**Switches:** Dispositivos que envían datos solo al puerto de destino correcto, mejorando la eficiencia de la red. Ejemplo: Switch Ethernet en una red LAN.

### **5.5.2 Routers**

Dispositivos que encaminan paquetes de datos entre diferentes redes, determinando la mejor ruta para cada paquete. Ejemplo: Un router en una casa que conecta la red local a Internet.

### **5.5.3 Firewalls**

Dispositivos o programas que protegen las redes de accesos no autorizados y ataques cibernéticos. Ejemplo: Un firewall que bloquea el tráfico no autorizado entre Internet y la red de una empresa.

#### **5.5.4 Servidores y clientes**

**Servidores:** Dispositivos o programas que proporcionan servicios a otros dispositivos o programas (clientes) en la red. Ejemplo: Un servidor web que aloja sitios web y los entrega a los navegadores de los usuarios.

**Clientes:** Dispositivos o programas que solicitan y utilizan los servicios proporcionados por los servidores. Ejemplo: Un navegador web que solicita páginas web de un servidor web.

### **5.6 Tecnologías de Red Inalámbrica**

#### **5.6.1 Wi-Fi (IEEE 802.11)**

Tecnología de red inalámbrica que permite a los dispositivos conectarse a Internet y entre sí sin cables. Utiliza ondas de radio para transmitir datos en frecuencias específicas. Ejemplo: Conexión de dispositivos en una red doméstica o en una red pública en una cafetería.

#### **5.6.2 Bluetooth**

Tecnología de red inalámbrica de corto alcance que permite la comunicación entre dispositivos cercanos, como teléfonos móviles, auriculares y teclados. Utiliza frecuencias de radio en la banda ISM de 2.4 GHz. Ejemplo: Conexión entre un teléfono móvil y un altavoz Bluetooth.

### **5.6.3 Redes celulares (3G, 4G, 5G)**

**3G:** Ofrece velocidades de datos más altas que las redes 2G, permitiendo navegación web, videollamadas y otros servicios de datos móviles.

**4G:** Proporciona velocidades de datos aún mayores que 3G, soportando aplicaciones de datos intensivas como streaming de video HD y juegos en línea.

**5G:** La última generación de redes móviles, ofreciendo velocidades ultrarrápidas, baja latencia y capacidad para conectar una gran cantidad de dispositivos simultáneamente. Ejemplo: Conexiones móviles de alta velocidad para teléfonos inteligentes y dispositivos IoT.

### **5.6.4 Satélites de comunicación**

Satélites en órbita terrestre que permiten la comunicación de datos y voz a largas distancias, especialmente en áreas donde las redes terrestres no son viables. Ejemplo: Comunicación en áreas remotas mediante satélites geoestacionarios.

## **5.7 Seguridad en Redes**

### **5.7.1 Amenazas comunes**

Las amenazas a la seguridad de las redes son variadas y evolucionan constantemente. Algunas de las amenazas más comunes incluyen:

**Malware:** Programas maliciosos como virus, troyanos y ransomware

que pueden dañar sistemas y robar datos.

**Phishing:** Intentos de engañar a los usuarios para que revelen información personal o financiera a través de correos electrónicos falsos.

**Ataques de denegación de servicio (DoS):** Intentos de hacer que un servicio en línea sea inaccesible sobrecargándolo con tráfico.

Intrusiones de red: Accesos no autorizados a redes y sistemas con la intención de robar información o causar daño

### **Ejemplo Práctico: Implementación de Seguridad en una Red Empresarial**

Una empresa mediana con aproximadamente 200 empleados desea mejorar la seguridad de su red interna. La empresa utiliza una red local (LAN) para conectar las computadoras de escritorio, servidores y dispositivos móviles de los empleados. Además, la red está conectada a Internet, lo que permite el acceso remoto a recursos y servicios en la nube.

#### ***Amenazas identificadas***

1. **Malware:** La empresa ha experimentado infecciones de malware en el pasado, afectando la productividad y la integridad de los datos.
2. **Phishing:** Algunos empleados han recibido correos electrónicos de phishing, comprometiendo sus credenciales de acceso.

3. **Accesos no autorizados:** La red no cuenta con una política de control de acceso robusta, lo que podría permitir el acceso no autorizado a datos sensibles.

### ***Soluciones implementadas***

#### **Implementación de un Firewall y un Sistema de Detección de Intrusiones (IDS):**

- **Firewall:** Se instala un firewall de última generación para monitorear y controlar el tráfico entrante y saliente basado en un conjunto de reglas de seguridad. El firewall bloquea el tráfico sospechoso y previene accesos no autorizados desde Internet.
- **IDS:** Un sistema de detección de intrusiones (como Snort) se despliega para identificar y alertar sobre actividades sospechosas o maliciosas dentro de la red. El IDS monitorea el tráfico en tiempo real y genera alertas para el equipo de seguridad.

#### **Uso de Software Antimalware:**

- Se implementa un software antimalware en todos los dispositivos de la empresa, incluidos computadoras de escritorio, laptops y servidores. Este software detecta y elimina malware en tiempo real, proporcionando una capa adicional de protección.

#### **1. Educación y concienciación del personal:**

- Se realiza una capacitación regular para educar a los empleados sobre las amenazas de phishing y la importancia de la seguridad de las contraseñas. Se enseñan mejores

prácticas para identificar correos electrónicos sospechosos y se promueve el uso de autenticación multifactor (MFA).

## 2. **Políticas de seguridad y control de acceso:**

- Se establecen políticas de seguridad claras que definen los derechos de acceso de los empleados a diferentes recursos de la red. Se implementa un sistema de control de acceso basado en roles (RBAC) para restringir el acceso a datos sensibles solo a aquellos empleados que realmente lo necesiten.
- Se utilizan VPNs (Redes Privadas Virtuales) para asegurar las conexiones remotas, permitiendo que los empleados accedan de forma segura a la red corporativa desde ubicaciones externas.

### ***Implementación de criptografía y autenticación***

- **Cifrado de Datos:** Se implementa cifrado de extremo a extremo para proteger la confidencialidad de los datos en tránsito y en reposo. Por ejemplo, se utiliza el protocolo SSL/TLS para asegurar las comunicaciones web y el cifrado AES para proteger los datos almacenados.
- **Autenticación Multifactor (MFA):** Se implementa MFA para todas las cuentas de usuario, añadiendo una capa adicional de seguridad mediante el uso de tokens de autenticación, aplicaciones móviles o códigos enviados por SMS.

Con estas medidas, la empresa logra mejorar significativamente su postura de seguridad. Las incidencias de malware y phishing disminuyen, y se establece una cultura de seguridad más robusta entre

los empleados. La implementación de políticas de control de acceso y cifrado de datos protege la información sensible, y la red se vuelve más resiliente frente a posibles ataques y accesos no autorizados

### **5.7.2 Criptografía y autenticación**

**Criptografía:** Uso de técnicas de cifrado para proteger la confidencialidad e integridad de los datos durante la transmisión. Ejemplo: HTTPS para la seguridad de las conexiones web.

**Autenticación:** Proceso de verificar la identidad de los usuarios y dispositivos antes de permitirles el acceso a recursos de la red. Ejemplo: Uso de contraseñas, tokens de seguridad y biometría.

La criptografía y la autenticación son elementos esenciales en la seguridad de las redes y telecomunicaciones. Estas tecnologías aseguran que la información transmitida sea confidencial, íntegra y que provenga de fuentes verificadas.

La criptografía es el estudio y la práctica de técnicas para asegurar la información y las comunicaciones mediante el uso de códigos para que solo aquellos a quienes va destinada la información puedan leer y procesar los datos.

#### **Tipos de criptografía**

##### **1. Criptografía simétrica:**

- **Descripción:** Utiliza la misma clave para cifrar y descifrar la información. Es eficiente pero presenta el desafío de la distribución segura de las claves.

- **Ejemplo:** Advanced Encryption Standard (AES), utilizado ampliamente en redes inalámbricas y aplicaciones de comunicación segura (Stallings, 2017).

## 2. Criptografía asimétrica:

- **Descripción:** Utiliza un par de claves, una pública y una privada. La clave pública cifra los datos, y la clave privada los descifra.
- **Ejemplo:** Rivest-Shamir-Adleman (RSA), comúnmente usado en protocolos de seguridad de red como SSL/TLS para establecer conexiones seguras (Kaufman, Perlman, & Speciner, 2010).

## 3. Funciones Hash:

- **Descripción:** Generan un valor de longitud fija a partir de una entrada de cualquier longitud, usadas para verificar la integridad de los datos.
- **Ejemplo:** Secure Hash Algorithm (SHA-256), utilizado para asegurar la integridad de los mensajes y en firmas digitales (Ferguson & Schneier, 2003).

## Protocolos criptográficos

- **SSL/TLS:**
  - **Descripción:** Protocolo que asegura las comunicaciones a través de la red, proporcionando confidencialidad y autenticación.
  - **Aplicación práctica:** Utilizado en los navegadores web para asegurar las conexiones HTTPS, protegiendo la información transmitida entre el navegador y el servidor web (Rescorla, 2001).

- **IPsec:**

- **Descripción:** Conjunto de protocolos que proporciona seguridad a nivel de red para las comunicaciones de Internet.
- **Aplicación práctica:** Utilizado en VPNs para cifrar y autenticar los paquetes de datos transmitidos a través de la red pública

La autenticación es el proceso de verificar la identidad de un usuario o dispositivo antes de permitir el acceso a los recursos de la red.

## **Métodos de autenticación**

### **1. Contraseñas:**

- **Descripción:** Método más común basado en la verificación de un secreto conocido (contraseña).
- **Aplicación práctica:** Acceso a redes Wi-Fi protegidas con WPA2, donde los usuarios deben ingresar una contraseña para conectarse (Stallings, 2017).

### **2. Autenticación multifactor (MFA):**

- **Descripción:** Combina dos o más factores de autenticación para mejorar la seguridad. Los factores pueden ser algo que el usuario sabe (contraseña), algo que el usuario tiene (token de autenticación) y algo que el usuario es (biometría).
- **Aplicación Práctica:** Acceso a servicios en la nube donde los usuarios deben ingresar una contraseña y luego proporcionar un código generado por una aplicación móvil de autenticación.

### 3. **Certificados digitales:**

- **Descripción:** Utilizan criptografía asimétrica para vincular una clave pública con la identidad de una entidad. Los certificados son emitidos por Autoridades de Certificación (CA).
- **Aplicación Práctica:** En una red corporativa, los certificados digitales se utilizan para autenticar dispositivos y usuarios, asegurando que solo los dispositivos certificados puedan acceder a la red (Rescorla, 2001).

### 4. **Biometría:**

- **Descripción:** Autenticación basada en características físicas del usuario, como huellas dactilares, reconocimiento facial o escaneo de iris.
- **Aplicación Práctica:** Acceso a dispositivos móviles y sistemas críticos en entornos corporativos que requieren altos niveles de seguridad (Stallings, 2017).

## **Protocolos de autenticación**

- **Kerberos:**

- **Descripción:** Protocolo de autenticación de red que utiliza criptografía de clave simétrica y un tercero de confianza (servidor de autenticación) para autenticar usuarios y servicios.
- **Aplicación Práctica:** Utilizado en redes Windows y sistemas UNIX para asegurar las sesiones de inicio de sesión de los usuarios (Neuman & Ts'o, 1994).

- **RADIUS:**

- **Descripción:** Protocolo que proporciona autenticación, autorización y contabilización (AAA) para las conexiones remotas y servicios de acceso a la red.
- **Aplicación Práctica:** Implementado en redes Wi-Fi empresariales para autenticar usuarios y dispositivos mediante un servidor central (Rigney et al., 2000).

## **Ejemplo Práctico: Implementación de Criptografía y Autenticación en una Red Corporativa**

### **Contexto**

Una empresa multinacional con oficinas en varias ciudades desea implementar un sistema robusto de criptografía y autenticación para asegurar sus comunicaciones internas y proteger los datos sensibles de sus clientes y empleados.

#### **1. Cifrado de comunicaciones con SSL/TLS:**

- Se implementa SSL/TLS en todos los servidores web de la empresa para asegurar las conexiones HTTPS. Esto garantiza que la información transmitida entre los navegadores de los usuarios y los servidores esté cifrada y protegida contra interceptaciones (Rescorla, 2001).

#### **2. Autenticación Multifactor (MFA):**

Se implementa MFA para todas las cuentas de usuario en la red corporativa. Los empleados deben ingresar su contraseña y luego proporcionar un código de autenticación generado por una aplicación móvil. Esto reduce significativamente el riesgo de

accesos no autorizados incluso si una contraseña es comprometida.

### 3. **Certificados Digitales para Dispositivos:**

- Se emiten certificados digitales para todos los dispositivos que necesitan acceder a la red corporativa. Esto asegura que solo los dispositivos certificados pueden establecer conexiones seguras con los recursos de la red (Rescorla, 2001).
- **Uso de IPsec para VPNs:** Se utilizan túneles IPsec para las conexiones VPN, asegurando que los datos transmitidos entre las oficinas remotas y la sede central estén cifrados y autenticados. Esto protege las comunicaciones de la empresa frente a posibles ataques en la red pública.

### 4. **Implementación de Kerberos:**

- Kerberos se implementa para la autenticación de usuarios y servicios en la red interna, proporcionando un método seguro y eficiente para verificar identidades sin enviar contraseñas en texto claro (Neuman & Ts'o, 1994)

#### **5.7.3 VPNs (Redes Privadas Virtuales)**

Tecnología que permite crear una conexión segura y cifrada sobre una red pública como Internet. Las VPNs protegen la privacidad de los usuarios y permiten el acceso seguro a recursos de la red corporativa desde ubicaciones remotas.

Ejemplo: Empleados que trabajan desde casa accediendo de manera segura a los servidores de la empresa.

#### 5.7.4 Políticas de seguridad

Conjunto de normas y procedimientos que definen cómo proteger los recursos de la red y los datos. Incluyen:

- **Contraseñas seguras:** Uso de contraseñas fuertes y cambios periódicos.
- **Actualización de software:** Aplicación regular de parches de seguridad y actualizaciones.
- **Educación y formación:** Capacitación de los empleados en prácticas de seguridad.

### 5.8 Internet y la World Wide Web

#### 5.8.1 Estructura y funcionamiento de Internet

- Internet es una red global de redes interconectadas que permite la comunicación y el intercambio de datos entre dispositivos en todo el mundo. Utiliza el protocolo TCP/IP para la transmisión de datos.
- **Proveedores de servicios de Internet (ISP):** Empresas que proporcionan acceso a Internet a usuarios y organizaciones.
- **Infraestructura de red:** Incluye cables submarinos, satélites, routers y centros de datos que facilitan la transmisión de datos.

## 5.8.2 Servicios de internet

- **Correo electrónico:** Servicio que permite enviar y recibir mensajes electrónicos.
- **Web:** Conjunto de páginas web interconectadas accesibles a través de navegadores web.
- **Transferencia de archivos:** Servicios como FTP que permiten el intercambio de archivos entre dispositivos.
- **Mensajería instantánea:** Servicios que permiten la comunicación en tiempo real, como WhatsApp y Slack.

## 5.8.3 Protocolos web (HTTP, HTTPS)

HTTP (Hypertext Transfer Protocol): Protocolo utilizado para la transferencia de páginas web.

HTTPS (HTTP Secure): Versión segura de HTTP que utiliza cifrado SSL/TLS para proteger la transmisión de datos.

## 5.8.4 Tecnologías web modernas

**HTML5:** Lenguaje de marcado utilizado para estructurar y presentar contenido en la web.

**CSS3:** Lenguaje de estilo utilizado para describir la presentación de un documento HTML.

**JavaScript:** Lenguaje de programación utilizado para crear contenido web interactivo.

**APIs web:** Interfaces que permiten la integración de servicios y datos de diferentes fuentes en aplicaciones web.

## **5.9 Redes emergentes y tendencias futuras**

### **5.9.1 Internet de las cosas (IoT)**

Conjunto de dispositivos interconectados que pueden recopilar y compartir datos a través de Internet. Ejemplo: Sensores inteligentes en ciudades que monitorizan el tráfico y la calidad del aire.

### **5.9.2 Redes definidas por software (SDN)**

Arquitectura de red que permite la gestión centralizada y programable de los recursos de red mediante software, en lugar de depender de hardware especializado. Ejemplo: Redes en centros de datos que se pueden reconfigurar dinámicamente para optimizar el rendimiento.

### **5.9.3 5G y más allá**

La quinta generación de redes móviles, que ofrece velocidades ultrarrápidas, baja latencia y capacidad para conectar una gran cantidad de dispositivos simultáneamente. Futuras generaciones (6G) prometen aún más mejoras en velocidad y conectividad. Ejemplo: Aplicaciones de realidad aumentada y virtual en tiempo real.

## **5.9.4 Computación en la nube y edge computing**

**Computación en la nube:** Provisión de recursos informáticos (servidores, almacenamiento, bases de datos) a través de Internet. Ejemplo: Servicios como Amazon Web Services (AWS) y Microsoft Azure.

**Edge computing:** Procesamiento de datos cerca del lugar donde se generan, reduciendo la latencia y mejorando la eficiencia. Ejemplo: Análisis de datos en dispositivos IoT en una fábrica inteligente.

## **5.10 Gestión y monitoreo de redes**

### **5.10.1 Herramientas de gestión de red**

Software y dispositivos que permiten la administración, configuración y monitoreo de la infraestructura de red. Ejemplo: Cisco Network Assistant, SolarWinds Network Performance Monitor.

### **5.10.2 Análisis de tráfico**

Proceso de monitorización y análisis del flujo de datos en una red para identificar patrones, problemas y oportunidades de mejora. Ejemplo: Uso de herramientas como Wireshark para capturar y analizar paquetes de datos.

### **5.10.3 Resolución de problemas**

Proceso de identificar, diagnosticar y solucionar problemas en la red. Incluye:

**Detección de fallos:** Identificación de problemas mediante el monitoreo de la red.

**Diagnóstico:** Análisis de los síntomas y la identificación de la causa raíz del problema.

**Corrección:** Implementación de soluciones para resolver el problema y restaurar el funcionamiento normal.

#### **5.10.4 Optimización del rendimiento**

Mejora de la eficiencia y el rendimiento de la red mediante ajustes y actualizaciones. Incluye:

**Balanceo de carga:** Distribución del tráfico de red de manera equitativa entre múltiples servidores.

**Calidad de servicio (QoS):** Priorización de ciertos tipos de tráfico para garantizar un rendimiento óptimo. Ejemplo: Priorización de tráfico de voz sobre IP (VoIP) en una red empresarial

# UNIDAD 6

Arquitecturas avanzadas y  
tendencias emergentes

6

## ARQUITECTURAS AVANZADAS Y TENDENCIAS EMERGENTES

### 6.1 Introducción

La arquitectura de computadores es un campo dinámico y en constante evolución, con innovaciones continuas que buscan mejorar el rendimiento, la eficiencia energética y la capacidad de procesamiento. Las arquitecturas de computación avanzadas están transformando la forma en que se abordan los problemas computacionales y están impulsando el desarrollo de nuevas aplicaciones y tecnologías. Desde los sistemas de multiprocesadores y multicomputadoras hasta la computación en la nube, la computación cuántica y las arquitecturas neuromórficas, cada una de estas arquitecturas ofrece ventajas únicas y enfrenta desafíos específicos.

Las arquitecturas avanzadas representan un segmento crucial en este ámbito, abarcando desde los procesadores multi-core y las unidades de procesamiento gráfico (GPU), hasta las arquitecturas

especializadas como los sistemas en chip (SoC) y las arquitecturas de computación cuántica. Este capítulo explora en detalle las arquitecturas avanzadas, sus principios fundamentales, las innovaciones recientes y su impacto en la informática moderna (Hennessy & Patterson, 2017).

El futuro de la computación dependerá de la capacidad de los investigadores y desarrolladores para superar estos desafíos y aprovechar las oportunidades que brindan estas arquitecturas avanzadas. La colaboración interdisciplinaria y la inversión en investigación y desarrollo serán fundamentales para continuar avanzando en el campo de la computación y para lograr innovaciones que beneficien a la sociedad en su conjunto.

## **6.2 Conceptos de Procesamiento Paralelo**

El procesamiento paralelo es una técnica utilizada en la computación para llevar a cabo múltiples cálculos o procesos de manera simultánea. Esta técnica se ha convertido en esencial para abordar los desafíos de rendimiento y eficiencia en los sistemas informáticos modernos. Con el fin de maximizar la velocidad de procesamiento y optimizar el uso de recursos, el procesamiento paralelo se implementa en diversas arquitecturas de hardware y software. Este texto explora en profundidad los conceptos fundamentales del procesamiento paralelo, incluyendo sus principios básicos, modelos de programación, tipos de paralelismo, y desafíos asociados.

### **6.2.1 Principios Básicos del Procesamiento Paralelo**

El procesamiento paralelo se refiere a la realización de múltiples operaciones o tareas simultáneamente. La motivación principal para utilizar procesamiento paralelo radica en la necesidad de aumentar la

velocidad de ejecución de programas y mejorar el rendimiento general de los sistemas informáticos. A medida que la demanda de procesamiento de datos crece exponencialmente, el procesamiento paralelo ofrece una solución eficaz para manejar grandes volúmenes de información y realizar cálculos complejos en menor tiempo.

## **Arquitectura y Modelos**

Existen varios modelos y arquitecturas de procesamiento paralelo, cada uno con sus propias ventajas y desventajas. Las arquitecturas más comunes incluyen:

- **Arquitectura de Memoria Compartida:** En este modelo, múltiples procesadores acceden a una memoria común. Esto facilita la comunicación y el intercambio de datos entre procesadores, pero puede llevar a problemas de concurrencia y coherencia de caché (Hennessy & Patterson, 2017).
- **Arquitectura de Memoria Distribuida:** Cada procesador tiene su propia memoria local, y la comunicación entre procesadores se realiza a través de una red de interconexión. Este modelo es escalable y adecuado para sistemas de gran tamaño, aunque la comunicación entre nodos puede ser un cuello de botella (Culler & Singh, 1998).
- **Arquitectura Híbrida:** Combina elementos de memoria compartida y distribuida, buscando aprovechar las ventajas de ambos modelos para optimizar el rendimiento y la escalabilidad (Sterling et al., 1999).
- **Programación de Memoria Compartida:** La programación de memoria compartida implica la utilización de múltiples hilos dentro de un único proceso, donde todos los hilos comparten el

mismo espacio de memoria. Las bibliotecas como OpenMP (Open Multi-Processing) facilitan la implementación de programas paralelos en este modelo mediante directivas que permiten al programador indicar qué partes del código deben ejecutarse en paralelo.

- **Programación de Memoria Distribuida:** En la programación de memoria distribuida, cada proceso tiene su propio espacio de memoria, y la comunicación entre procesos se realiza mediante el paso de mensajes. El estándar Message Passing Interface (MPI) es ampliamente utilizado para este tipo de programación, proporcionando una plataforma robusta para la comunicación y sincronización entre procesos en sistemas distribuidos.

- 

### 6.2.2 Tipos de Paralelismo

- **Paralelismo de Datos:** El paralelismo de datos implica la división de grandes conjuntos de datos en partes más pequeñas que se procesan simultáneamente en diferentes unidades de procesamiento. Este enfoque es ideal para aplicaciones como el procesamiento de imágenes, simulaciones científicas y análisis de datos, donde las mismas operaciones se aplican a múltiples elementos de datos.

- **Paralelismo de Tareas:** El paralelismo de tareas se refiere a la ejecución simultánea de diferentes tareas o funciones dentro de un programa. Este tipo de paralelismo es útil en aplicaciones donde se pueden identificar tareas independientes que pueden ejecutarse concurrentemente, como en servidores web y sistemas de bases de datos.

## **Desafíos del procesamiento paralelo**

**Sincronización y Consistencia:** Uno de los mayores desafíos en el procesamiento paralelo es la sincronización adecuada de tareas y la consistencia de los datos. Los problemas de concurrencia, como las condiciones de carrera y los bloqueos, pueden surgir cuando múltiples hilos o procesos intentan acceder y modificar los mismos datos simultáneamente. Las técnicas de sincronización, como los cerrojos, semáforos y barreras, son esenciales para gestionar estos problemas.

**Escalabilidad:** La escalabilidad es la capacidad de un sistema paralelo para mantener o mejorar su rendimiento a medida que se incrementa el número de procesadores. Los sistemas mal diseñados pueden enfrentar cuellos de botella que limitan la eficiencia a medida que se añaden más recursos. Las arquitecturas y algoritmos deben ser diseñados teniendo en cuenta la escalabilidad para maximizar el rendimiento en sistemas grandes.

**Latencia de Comunicación:** En sistemas de memoria distribuida, la latencia de comunicación entre nodos puede ser un factor limitante. La optimización de la comunicación mediante técnicas como la agregación de mensajes y la reducción de la sobrecarga de comunicación es crucial para mejorar el rendimiento de aplicaciones paralelas (Dally & Towles, 2004).

## Aplicaciones del procesamiento paralelo

- **Computación Científica:** El procesamiento paralelo es fundamental en la computación científica, donde se requieren grandes cantidades de cálculos para simular fenómenos físicos, biológicos y químicos. Los supercomputadores y las plataformas de computación de alto rendimiento (HPC) utilizan procesamiento paralelo para resolver problemas complejos en menor tiempo.
- **Inteligencia Artificial y Aprendizaje Profundo:** El aprendizaje profundo y otras técnicas de inteligencia artificial se benefician enormemente del procesamiento paralelo. Las GPUs y TPUs (Tensor Processing Units) están diseñadas específicamente para ejecutar operaciones matemáticas paralelas, acelerando significativamente el entrenamiento de modelos de aprendizaje profundo (LeCun et al., 2015).
- **Análisis de Big Data:** El procesamiento de grandes volúmenes de datos, común en aplicaciones de Big Data, requiere el uso de técnicas de procesamiento paralelo para manejar y analizar datos de manera eficiente. Herramientas como Apache Hadoop y Apache Spark utilizan paradigmas de programación paralela para distribuir el procesamiento de datos entre múltiples nodos (Dean & Ghemawat, 2008).

### 6.3 Arquitecturas de multiprocesadores y multicomputadoras

#### 6.3.1 Procesadores Multi-Core

Surgieron como una respuesta a las limitaciones físicas y térmicas de aumentar la frecuencia de reloj en los procesadores de un solo

núcleo. La Ley de Moore, que predice la duplicación del número de transistores en un chip aproximadamente cada dos años, impulsó la necesidad de encontrar nuevas formas de aprovechar el creciente número de transistores sin incrementar significativamente el consumo de energía y la generación de calor (Moore, 1965; Sutter, 2005).

En un procesador multi-core, varios núcleos independientes están integrados en un solo chip. Cada núcleo puede ejecutar instrucciones de manera independiente, lo que permite el procesamiento paralelo de tareas. Este enfoque mejora el rendimiento en aplicaciones que pueden dividirse en tareas concurrentes, como el procesamiento de imágenes, la simulación científica y las aplicaciones de inteligencia artificial (Hill & Marty, 2008).

Uno de los principales desafíos de los procesadores multi-core es la gestión eficiente de los recursos compartidos, como la memoria caché y el bus de datos. La coherencia de la caché y la sincronización de tareas son aspectos críticos para garantizar un rendimiento óptimo. Las técnicas como la coherencia de la caché basada en directorio y los protocolos de coherencia de caché son fundamentales para abordar estos problemas.

### **6.3.2 Unidades de procesamiento gráfico (GPU)**

#### **Evolución y uso en computación general**

Originalmente diseñadas para el procesamiento gráfico, las GPU han evolucionado hasta convertirse en potentes unidades de procesamiento paralelo utilizadas en una amplia gama de aplicaciones, desde la renderización de gráficos hasta el aprendizaje profundo y la minería de datos (Owens et al., 2008).

## **Arquitectura y principios**

Las GPUs están compuestas por miles de núcleos pequeños y eficientes diseñados para realizar operaciones matemáticas simples de manera simultánea. Este enfoque es ideal para aplicaciones que requieren un procesamiento masivo de datos en paralelo, como la renderización de gráficos y el entrenamiento de modelos de aprendizaje profundo (Kirk & Hwu, 2010).

A pesar de su gran potencia de procesamiento, las GPUs enfrentan desafíos relacionados con la eficiencia energética y la latencia de la memoria. La introducción de memorias de alto ancho de banda (HBM) y técnicas avanzadas de gestión de energía han mejorado significativamente estos aspectos. Además, las plataformas de desarrollo como CUDA y OpenCL han facilitado la programación de aplicaciones que aprovechan el poder de las GPUs (Nickolls et al., 2008).

### **6.3.3 Sistemas en chip (SoC)**

Un SoC integra todos los componentes de un sistema informático en un único chip, incluyendo el procesador, la memoria, las interfaces de entrada/salida y otros periféricos. Esta arquitectura es común en dispositivos móviles, sistemas embebidos y aplicaciones de Internet de las Cosas (IoT) (Wolf, 2014).

Los SoCs están diseñados para ser altamente integrados y eficientes en términos de energía. Utilizan una combinación de núcleos de procesamiento heterogéneos (como CPUs y GPUs) y aceleradores especializados (como unidades de procesamiento neuronal) para optimizar el rendimiento y la eficiencia energética (Keating et al., 2011). (Agregar a referencias)

El diseño de SoCs implica desafíos significativos relacionados con la integración de múltiples componentes en un espacio reducido, la gestión del calor y la interoperabilidad entre diferentes subsistemas. Las técnicas avanzadas de diseño de circuitos y la simulación a nivel de sistema son cruciales para abordar estos problemas. Además, la estandarización de interfaces y protocolos ha facilitado la integración de componentes de diferentes fabricantes (Gajski et al., 2009).

#### **6.3.4 Arquitecturas de computación cuántica**

La computación cuántica se basa en los principios de la mecánica cuántica, utilizando qubits en lugar de bits clásicos. Los qubits pueden representar simultáneamente 0 y 1 gracias al fenómeno de la superposición, y pueden estar entrelazados mediante el entrelazamiento cuántico, lo que permite la realización de cálculos complejos de manera exponencialmente más rápida que las computadoras clásicas en ciertos problemas (Nielsen & Chuang, 2010).

#### **Arquitecturas cuánticas**

Existen varias arquitecturas de computación cuántica, incluyendo los computadores cuánticos basados en trampas de iones, superconductores y fotónica. Cada una de estas arquitecturas tiene sus propias ventajas y desafíos en términos de escalabilidad, estabilidad y corrección de errores cuánticos (Monroe & Kim, 2013).

La computación cuántica enfrenta desafíos significativos relacionados con la decoherencia y la corrección de errores. Las técnicas como el código de corrección de errores cuánticos y los métodos de aislamiento y enfriamiento avanzado han sido desarrolladas para mitigar estos problemas. Además, los algoritmos cuánticos, como el

algoritmo de Shor y el algoritmo de Grover, han demostrado el potencial de la computación cuántica para resolver problemas que son intratables para las computadoras clásicas.

### **6.3.5 Computación neuromórfica**

#### **Inspiración biológica**

La computación neuromórfica se inspira en la estructura y funcionamiento del cerebro humano para desarrollar sistemas informáticos que pueden realizar tareas cognitivas de manera eficiente. Estos sistemas utilizan neuronas artificiales y sinapsis para procesar información de manera paralela y distribuida.

Los chips neuromórficos, como el IBM TrueNorth y el Intel Loihi, están diseñados para emular la eficiencia energética y la capacidad de aprendizaje del cerebro. Utilizan circuitos especializados para implementar redes neuronales spiking, que son más eficientes en términos de energía que las redes neuronales tradicionales (Akopyan et al., 2015; Davies et al., 2018).

La computación neuromórfica tiene aplicaciones prometedoras en el campo de la inteligencia artificial, la robótica y los sistemas embebidos. Sin embargo, el diseño de algoritmos y arquitecturas que puedan aprovechar plenamente el potencial de estos sistemas sigue siendo un desafío. La investigación en este campo está enfocada en desarrollar nuevos modelos de aprendizaje y técnicas de diseño de hardware que puedan mejorar la escalabilidad y eficiencia de los sistemas neuromórficos (Indiveri & Liu, 2015).

Las arquitecturas avanzadas de computadores están transformando la informática moderna, ofreciendo nuevas capacidades y mejorando el rendimiento y la eficiencia energética de los sistemas informáticos. Desde los procesadores multi-core y las GPUs hasta los SoCs y la computación cuántica, estas innovaciones están impulsando el desarrollo de nuevas aplicaciones y tecnologías que están cambiando la forma en que interactuamos con la tecnología. A medida que continuamos explorando y desarrollando estas arquitecturas, es probable que veamos aún más avances que transformarán la informática en los años venideros (Hennessy & Patterson, 2017).

## **6.4 Arquitecturas avanzadas**

### **6.4.1 Computación cuántica**

La computación cuántica representa una nueva frontera en el procesamiento paralelo. Los computadores cuánticos utilizan qubits que pueden estar en múltiples estados simultáneamente, lo que permite la realización de cálculos paralelos a una escala sin precedentes. Aunque aún en desarrollo, la computación cuántica tiene el potencial de revolucionar campos como la criptografía, la optimización y la simulación de materiales (Nielsen & Chuang, 2010).

### **6.4.2 Computación neuromórfica**

La computación neuromórfica, inspirada en la estructura y funcionamiento del cerebro humano, utiliza arquitecturas paralelas para realizar tareas cognitivas de manera eficiente. Los sistemas neuromórficos prometen avances en la inteligencia artificial y la robótica, proporcionando nuevas formas de procesamiento paralelo adaptativas y eficientes en términos de energía.

### **6.4.3 Computación heterogénea**

La computación heterogénea, que combina diferentes tipos de unidades de procesamiento (como CPUs, GPUs y FPGAs) en un solo sistema, está emergiendo como una tendencia importante en el diseño de sistemas paralelos. Esta aproximación puede ofrecer ventajas significativas en términos de rendimiento y eficiencia energética (Keller et al., 2010).

## Referencias

- Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C., & Zhang, J. C. (2014). What Will 5G Be?. *IEEE Journal on Selected Areas in Communications*, 32(6), 1065-1082
- Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). *Mastering Cloud Computing: Foundations and Applications Programming*. Morgan Kaufmann.
- Dahlman, E., Parkvall, S., & Skold, J. (2016). *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press.
- Dierks, T., & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246*.
- Dykstra, T. (2004). *Network Performance Open-Source Toolkit: Using Netperf, Tcptrace, NISTnet, and SSFNet*. Wiley
- Fielding, R., et al. (1999). *Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616*.
- Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press.
- Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1), 14-76
- Matejka, J. (2016). *Virtual Private Networks: A Beginner's Guide*. McGraw-Hill Education
- NN-MZimmermann, H. (1980). OSI Reference Model--The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28(4), 425-432.

- Neuman, B. C., & Ts'o, T. (1994). Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, 32(9), 33-38.
- Postel, J. (1980). *User Datagram Protocol. RFC 768.*
- Postel, J. (1981). *Transmission Control Protocol. RFC 793.*
- Rappaport, T. S. (1996). *Wireless Communications: Principles and Practice.* Prentice Hall.
- Rescorla, E. (2001). *SSL and TLS: Designing and Building Secure Systems.* Addison-Wesley.
- Rigney, C., Willens, S., Rubens, A., & Simpson, W. (2000). *Remote Authentication Dial-In User Service (RADIUS) (RFC 2865).* IETF.
- Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice.* Pearson
- Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637-646
- Spurgeon, C. E., & Zimmerman, J. (2014). *Ethernet Switches: An Introduction to Network Design with Switches.* O'Reilly Media.

ISBN: 978-9942-679-37-6



9789942679376