

Unidad 1 – Introducción a los sistemas hardware y arquitectura básica

1. Concepto de sistema informático

Un **sistema informático** es un conjunto de elementos que cooperan para procesar información:

- **Hardware** → la parte física (CPU, memoria, placas, discos, periféricos).
- **Software** → programas que gestionan el hardware y proporcionan servicios.
- **Firmware** → código de bajo nivel que controla el hardware (BIOS/UEFI, microcódigo de CPU).
- **Usuarios/procesos** → quienes usan o ejecutan el sistema.

En ASIR nos centraremos en la **capa hardware**, pero siempre relacionándola con el software que administra el sistema.

2. Arquitectura de Von Neumann

La mayoría de los sistemas actuales siguen el **modelo de Von Neumann**, que define:

- **CPU (Unidad Central de Proceso)**
 - **Unidad de Control (UC):** interpreta instrucciones y genera señales de control.
 - **Unidad Aritmético-Lógica (ALU):** realiza operaciones matemáticas y lógicas.
 - **Registros:**
 - **Generales:** operandos de operaciones.
 - **PC (Program Counter):** dirección de la próxima instrucción.
 - **IR (Instruction Register):** instrucción en ejecución.
 - **SP (Stack Pointer):** controla la pila.
 - **FLAGS/PSW:** indicadores de estado (overflow, cero, negativo...).
- **Memoria principal (RAM):** contiene instrucciones y datos en ejecución.
- **Buses:**
 - **De direcciones** → dónde leer/escribir.
 - **De datos** → bits a transferir.
 - **De control** → señales de sincronización (lectura, escritura, interrupciones).
- **E/S (entrada/salida):** periféricos como teclado, pantalla, red, discos.

3. Ciclo de instrucción

El funcionamiento de la CPU se basa en un ciclo repetitivo:

1. **Fetch:** el PC indica la dirección de la instrucción; se lee de memoria.
2. **Decode:** la UC interpreta el opcode.
3. **Execute:** la ALU u otra unidad realiza la operación.
4. **Write-back:** el resultado se almacena en un registro o en memoria.
5. El PC se incrementa (salvo en saltos o interrupciones).

Un procesador moderno puede realizar miles de millones de ciclos por segundo (3 GHz = 3.000 Mhz \approx 3.000 millones de instrucciones/s).

4. Jerarquía de memoria

Las memorias se organizan en niveles, equilibrando **velocidad, capacidad y coste**:

Nivel	Latencia aprox.	Tamaño típico	Ejemplo
Registros	1 ciclo CPU	bytes	EAX, R0...
Caché L1	1-2 ns	32–64 KB	Integrada en núcleo CPU
Caché L2	4–12 ns	256 KB–1 MB	Compartida por núcleo
Caché L3	20–40 ns	4–32 MB	Compartida multicore
RAM	80–120 ns	8–64 GB	DDR4, DDR5
SSD NVMe	~100 µs	TB	PCIe 4.0/5.0
HDD	5–10 ms	TB	SATA

Acceder a RAM puede ser **100 veces más lento** que a caché, y un HDD es **100.000 veces más lento** que un registro.

5. Subsistemas de almacenamiento

- **HDD (discos magnéticos):** cabezal + platos → tiempo de acceso = latencia rotacional + búsqueda.
- **SSD (SATA/NVMe):** celdas NAND Flash → accesos en microsegundos.
- **NVMe:** múltiples colas de comandos, aprovecha el bus PCIe → >3 GB/s.
- **Sistemas RAID:**
 - **RAID 0:** striping, velocidad sin redundancia.
 - **RAID 1:** mirroring, seguridad duplicada.
 - **RAID 5/10:** compromiso entre velocidad y redundancia.

6. Mecanismos de entrada/salida

- **Polling:** CPU consulta dispositivo → ineficiente.
- **Interrupciones (IRQ):** dispositivo avisa a la CPU → más eficiente.
- **DMA (Direct Memory Access):** controlador mueve datos de dispositivo a RAM sin pasar por CPU.

Ejemplo: tarjeta de red descarga paquetes directamente en memoria, CPU los procesa después.

7. Sistema operativo como gestor de hardware

El SO actúa como **capa de abstracción**:

- **Gestión de procesos:** planificación y multitarea.
- **Gestión de memoria:** segmentación, paginación, TLB, protección.
- **Gestión de archivos:** sistemas de ficheros (NTFS, ext4, XFS).
- **Gestión de dispositivos:** drivers como interfaz estándar.
- **Seguridad:** separación modo kernel/usuario, permisos, auditoría.

8. Relación con ASIR

- Un administrador debe **conocer los límites y capacidades del hardware** para configurar correctamente servidores, SO y servicios.
- Saber diagnosticar **fallos de RAM, disco, fuente de alimentación o CPU** es esencial para el mantenimiento.
- Entender la **jerarquía de memoria y almacenamiento** ayuda a optimizar rendimiento de sistemas y redes.

Los sistemas informáticos se construyen sobre una arquitectura base donde la CPU, memoria, buses y E/S interactúan de manera orquestada. La eficiencia y fiabilidad de un sistema dependen de entender esa arquitectura y de cómo el sistema operativo la gestiona.