

# Sistemas Informáticos – 1º DAM

## 1. Finalidad de la asignatura

El módulo tiene como objetivo que el alumnado **conozca, instale, configure y mantenga sistemas informáticos** (hardware, software y redes básicas) para dar soporte al desarrollo de aplicaciones. Se busca que adquieran autonomía para **preparar entornos de trabajo fiables y seguros** en los que posteriormente desplegarán y probarán aplicaciones.

---

## 2. Resultados de aprendizaje (RA)

El alumnado será capaz de:

1. **Reconocer la estructura de un ordenador**, sus componentes y funciones.
2. **Instalar, configurar y verificar software de base** (sistemas operativos, máquinas virtuales).
3. **Gestionar la información y los recursos del sistema** (ficheros, procesos, usuarios, permisos).
4. **Configurar redes de área local básicas** y verificar su funcionamiento.
5. **Aplicar medidas de seguridad** en el sistema informático (antivirus, copias de seguridad, actualizaciones).
6. **Automatizar tareas** mediante scripts sencillos.

### **3. Contenidos básicos**

#### **Bloque 1: Arquitectura y hardware del sistema**

- Componentes físicos del ordenador.
- Placa base, CPU, memoria, buses y dispositivos de almacenamiento.
- Periféricos de entrada, salida y comunicación.
- Configuración básica de BIOS/UEFI.

#### **Bloque 2: Instalación y configuración de sistemas operativos**

- Instalación de Windows y Linux en equipos físicos y virtuales.
- Configuración de controladores y utilidades básicas.
- Virtualización: creación y gestión de máquinas virtuales.
- Herramientas de gestión de discos y particiones.

#### **Bloque 3: Gestión de sistemas operativos**

- Sistemas de archivos: NTFS, EXT4, FAT32, etc.
- Gestión de usuarios, grupos y permisos.
- Procesos y servicios: monitorización y control.
- Configuración de software de uso general (ofimática, utilidades, entornos de desarrollo).

#### **Bloque 4: Redes básicas**

- Conceptos de redes locales.
- Dirección IP, máscara de subred y puerta de enlace.
- Configuración de adaptadores de red.
- Comprobación de conectividad (ping, tracert, ipconfig/ifconfig).

## **Bloque 5: Seguridad en sistemas**

- Políticas de actualización y parches.
- Antivirus, cortafuegos y medidas de protección.
- Estrategias de copia de seguridad y restauración.
- Control de acceso al sistema.

## **Bloque 6: Automatización y scripting**

- Introducción a la línea de comandos (Windows y Linux).
- Scripts básicos en Bash y/o PowerShell.
- Automatización de tareas y ejecución programada.

# Sistemas Informáticos (1º DAM)

## Objetivos de la sesión

- Comprender **qué es un sistema informático** y sus componentes.
- Analizar la **arquitectura de Von Neumann** como modelo de referencia.
- Conocer los **principales subsistemas de hardware** (CPU, memoria, buses, almacenamiento).
- Introducir la relación **hardware ↔ sistema operativo ↔ software de aplicación**.

## Contenido teórico (nivel técnico)

### 1. Definición de sistema informático

Un **sistema informático** es un conjunto de elementos interrelacionados que permiten el **tratamiento automático de la información**.

- **Hardware** → parte física (circuitos, dispositivos).
- **Software** → instrucciones que gestionan el hardware.
- **Usuarios** → operan el sistema.
- **Datos** → la materia prima que circula por el sistema.

## 2. Arquitectura de Von Neumann

Base de casi todos los sistemas actuales.

- **Componentes principales:**
  - **CPU** (Unidad Central de Proceso): controla y ejecuta instrucciones.
    - **ALU (Unidad Aritmético-Lógica)** → operaciones matemáticas y lógicas.
    - **UC (Unidad de Control)** → interpreta instrucciones y coordina al resto.
    - **Registros** → memoria ultrarrápida interna (contador de programa, acumulador, registros de propósito general).
  - **Memoria principal (RAM)**: almacena instrucciones y datos en ejecución.
  - **Dispositivos de E/S**: permiten la comunicación con el exterior.
  - **Bus del sistema**: canal de comunicación (direcciones, datos, control).

Idea clave: **programa e instrucciones se almacenan en memoria junto con los datos.**

## 3. Jerarquía de memoria

- **Registros** (nanosegundos).
- **Caché (L1, L2, L3).**
- **Memoria principal (RAM).**
- **Almacenamiento secundario** (SSD, HDD).
- **Almacenamiento terciario** (copias de seguridad en cintas, nube).

👉 Compromiso entre **velocidad, capacidad y coste.**

---

## 4. Subsistemas de almacenamiento

- **Discos magnéticos (HDD):** platos, cabezales, tiempo de búsqueda, latencia rotacional.
- **Discos sólidos (SSD, NVMe):** sin partes móviles, gran velocidad, interfaz SATA vs PCIe.
- **Sistemas RAID:**
  - RAID 0 (striping, rendimiento, sin redundancia).
  - RAID 1 (mirroring, redundancia total).
  - RAID 5 (paridad distribuida).
  - RAID 10 (rendimiento + redundancia).

## 5. Periféricos y controladores

- **Periféricos de entrada** (teclado, ratón, escáner).
- **Periféricos de salida** (monitor, impresora).
- **Dispositivos mixtos** (discos, tarjetas de red).
- **Drivers** → software que permite al sistema operativo gestionar un dispositivo.

## 6. Interacción hardware ↔ software

- **Sistema operativo (SO):** interfaz entre hardware y aplicaciones.
- Funciones básicas del SO:
  - Gestión de procesos.
  - Gestión de memoria.
  - Gestión de archivos.
  - Gestión de dispositivos de E/S.
  - Seguridad y control de acceso.

Esquema simplificado:

Usuario



Aplicaciones



Sistema operativo



Hardware

## Tarea

- Investigar el hardware de su propio PC (CPU, RAM, disco, placa base, GPU).
- Presentar en la siguiente clase un breve informe técnico (modelo, especificaciones, rendimiento relativo).



# 1. Concepto de sistema informático

Un **sistema informático** no es solo hardware + software:

- **Hardware** → recurso físico, finito y limitado.
- **Firmware** → capa intermedia que controla el hardware a bajo nivel (BIOS/UEFI, microcódigo).
- **Sistema Operativo (SO)** → kernel que abstrae el hardware y ofrece servicios.
- **Software de aplicación** → programas que consumen servicios del SO.
- **Usuario** → humano o proceso que interactúa.

En entornos DAM es fundamental entender que **cada aplicación que programen competirá por CPU, memoria y E/S**, gestionados por el SO.

## 2. Arquitectura de Von Neumann

### Elementos

- **CPU (Central Processing Unit):**
  - **UC (Unidad de Control):** interpreta la instrucción binaria, decodifica la operación y activa señales de control en buses.
  - **ALU (Unidad Aritmético-Lógica):** ejecuta sumas, restas, desplazamientos, comparaciones binarias.
  - **Registros:**
    - **Registros de propósito general** (R0, R1... en arquitecturas RISC; EAX, EBX... en x86).
    - **Registros especiales:**
      - **PC (Program Counter)** → apunta a la próxima instrucción en memoria.
      - **IR (Instruction Register)** → almacena la instrucción en ejecución.

- **SP (Stack Pointer)** → referencia al tope de la pila.
- **FLAGS/PSW** → banderas de estado (overflow, zero, carry).
- **Bus del sistema:**
  - **Bus de direcciones** → indica posición de memoria a leer/escribir.
  - **Bus de datos** → transporte de bits entre CPU, RAM y dispositivos.
  - **Bus de control** → señales de sincronización (lectura/escritura, interrupciones, clock).

### **Ciclo de instrucción (fetch–decode–execute):**

1. **Fetch:** CPU lee instrucción en dirección apuntada por PC.
2. **Decode:** UC interpreta opcode y operandos.
3. **Execute:** ALU/procesadores especializados ejecutan la instrucción.
4. **Write back:** resultado se guarda en registros o memoria.
5. PC se incrementa (salvo saltos).

Esto es clave para entender por qué un **bucle infinito** satura la CPU: nunca se libera el ciclo fetch-decode-execute.

### 3. Jerarquía y latencia de memoria

La memoria se organiza por **niveles de velocidad y coste**.

Nivel	Latencia aprox.	Ejemplo
Registros CPU	1 ciclo	eax, r0
Caché L1	1–2 ns	32KB, asociativa
Caché L2	4–12 ns	256KB–1MB
Caché L3	20–40 ns	compartida multicore
RAM DDR4/DDR5	80–120 ns	8–64 GB
SSD NVMe	100 µs	1–4 TB
HDD	5–10 ms	500 GB–2 TB

Impacto directo en programación:

- Algoritmos con **acceso secuencial** aprovechan la localidad espacial.
- Algoritmos con **acceso aleatorio** penalizan caches → peor rendimiento.

## 4. Subsistema de almacenamiento

- **HDD:** acceso mecánico (latencia  $\approx 10$  ms). Cuellos de botella.
- **SSD SATA:** interfaz limitado a 600 MB/s.
- **SSD NVMe (PCIe):** paralelismo de colas de comandos,  $> 3$  GB/s.
- **RAID:**
  - RAID 0: striping  $\rightarrow$  máximo rendimiento, cero redundancia.
  - RAID 1: duplicación  $\rightarrow$  coste en capacidad, alta seguridad.
  - RAID 5: paridad  $\rightarrow$  compromiso entre rendimiento y seguridad.
  - RAID 10: striping + mirroring  $\rightarrow$  rendimiento + seguridad (muy usado en servidores).

Importante: sistemas de bases de datos **exigen RAID con redundancia**, ya que no toleran pérdida de bloques.

## 5. Mecanismos de E/S

- **Polling**: CPU consulta constantemente al dispositivo (ineficiente).
- **Interrupciones (IRQ)**: el dispositivo interrumpe a la CPU para atender un evento.
- **DMA (Direct Memory Access)**: el controlador de E/S transfiere datos a memoria sin intervención de la CPU.

Ejemplo: una tarjeta de red usa **DMA** para descargar paquetes en memoria → el kernel luego los procesa.

## 6. Sistema operativo como capa de abstracción

El kernel ofrece:

- **Gestión de procesos**: planificación (round-robin, prioridades, colas multinivel).
- **Gestión de memoria**:
  - **Segmentación** (código, datos, pila).
  - **Paginación** (traducción virtual ↔ física con MMU, TLB).
- **Gestión de archivos**: árboles jerárquicos, inodos, journaling (ext4, NTFS).
- **Gestión de dispositivos**: drivers como abstracción estandarizada.

Concepto avanzado: **modo usuario vs modo kernel** → protección del hardware.

# Actividad práctica: “Auditoría Express de tu PC – Windows Forensics”

## Objetivos

- Monitorizar el **rendimiento** del sistema en tiempo real (CPU, RAM, disco, red).
- Detectar **problemas de seguridad** en configuración, usuarios y actualizaciones.
- Elaborar un informe sencillo con **hallazgos y propuestas de mejora**.

## Desarrollo de la actividad

### Fase 1: Rendimiento

#### 1. Visor de recursos (Resource Monitor)

- Abrir con **Win + R** → **resmon**.
- Analizar:
  - **CPU**: procesos que más consumen.
  - **Memoria**: memoria en uso vs disponible.
  - **Disco**: procesos que generan más I/O.
  - **Red**: conexiones activas.
- Preguntas:
  - ¿Qué proceso consume más CPU?
  - ¿Qué aplicación está generando más tráfico de red?

#### 2. Administrador de tareas (Task Manager)

- **Ctrl + Shift + Esc**.
- Ver pestañas: “Rendimiento” y “Inicio”.
- Preguntas:

- ¿Cuántos programas arrancan con Windows? ¿Son necesarios todos?
- ¿Qué impacto tiene esto en el arranque y en la RAM?

### 3. Prueba de disco

- Ejecutar `winsat disk -drive C` en **Símbolo del sistema** como admin.
- Pregunta: ¿qué tasa de transferencia obtiene el disco? ¿Se acerca a lo esperado para HDD o SSD?

## Fase 2: Seguridad

### 1. Usuarios y permisos

- `net user` en Símbolo del sistema.
- Abrir **Panel de control** → **Herramientas administrativas** → **Administración de equipos** → **Usuarios y grupos locales**.
- Pregunta: ¿hay más de un usuario administrador? ¿Es recomendable?

### 2. Servicios y puertos

- Abrir **PowerShell** como admin.

Comando:

```
Get-NetTCPConnection | Where-Object {$_.State -eq "Listen"}
```

- Pregunta: ¿qué servicios están escuchando? ¿Son todos necesarios?

### 3. Actualizaciones

Comando:

```
Get-WindowsUpdateLog
```

- Comprobar en Configuración → Windows Update si hay actualizaciones pendientes.

- Pregunta: ¿qué riesgos existen si no aplicamos parches de seguridad?

#### 4. Antivirus y protección

- Comprobar en **Seguridad de Windows**:
  - Estado de antivirus.
  - Protección contra ransomware.
- Pregunta: ¿qué diferencias hay entre un antivirus actualizado y uno obsoleto?