**ARQUITECTURA Y SISTEMAS OPERATIVOS**

### ****SISTEMA OPERATIVO (SO)****

Es un **programa que controla la ejecución de aplicaciones** y **actúa como interfaz** entre estas y el **hardware (HW)**.

### ****Objetivos del Sistema Operativo****

El SO tiene **tres objetivos principales**:

1. **Facilidad de uso**: Facilita el uso del computador.
2. **Eficiencia**: Permite utilizar los recursos del sistema de manera eficiente.
3. **Capacidad para evolucionar**: Permite desarrollar, probar e introducir nuevas funciones sin interrumpir el servicio.

### ****SO como Máquina Extendida****

* Presenta una **interfaz amigable para el programador u operador**.
* **Oculta las complejidades del hardware**, como interrupciones, gestión de memoria, temporizadores, y otras funciones de bajo nivel.

### ****SO como Controlador de Recursos****

* **Administra y reparte** de forma ordenada los **procesadores, memoria y dispositivos de E/S** entre los distintos programas.
* **Monitorea el uso de recursos**, gestiona solicitudes y **media en los conflictos** entre programas y usuarios que compiten por los mismos recursos.

# **DESARROLLO HISTORICO DE LOS SISTEMAS OPERATIVOS**

## ****GENERACIONES DE LAS COMPUTADORAS****

### ****Primera Generación (1945–1955): Tubos de vacío y tableros****

* Computadoras **enormes y lentas**.
* Utilizaban **miles de tubos de vacío**.
* Realizaban **operaciones numéricas simples**.
* La programación se hacía en **lenguaje de máquina absoluto**, muchas veces **alambrando tableros de conexión**.

### ****Segunda Generación (1955–1965): Transistores y sistema por lotes****

* Uso de **transistores**, lo que permitió mayor fiabilidad y menor tamaño.
* Surgieron los **mainframes**, que se alojaban en grandes salas y eran manejados por **operadores profesionales**.
* Se implementó el **sistema por lotes** para **reducir costos operativos**.

### ****Tercera Generación (1965–1980): Circuitos integrados y multiprogramación****

* Incorporación de **circuitos integrados**, lo que mejoró el rendimiento y redujo el tamaño.
* Las computadoras comenzaron a realizar **cómputos científicos y comerciales**.
* Apareció el concepto de **familia de computadoras compatibles**.
* Implementación de la **multiprogramación**, que dividía la memoria en partes para ejecutar **varios trabajos simultáneamente**.
* Introducción del **spooling** y los **sistemas de tiempo compartido**.

### ****Cuarta Generación (1980–presente): Computadoras personales****

* Nace la **computadora personal (PC)** gracias al desarrollo de **chips** y **microprocesadores**.
* Se desarrollaron **interfaces gráficas** amigables para el usuario.
* Surgieron los **sistemas operativos de red** y los **sistemas operativos distribuidos**.

## ****CONCEPTOS BÁSICOS****

**Procesos – Archivos – Llamadas al Sistema – Shell – Kernel**

### ****PROCESOS****

Un **proceso** es básicamente un **programa en ejecución**. Está compuesto por:

* El **programa ejecutable**.
* Sus **datos**, **pila**, **contador de programa** y otros **registros**.

Los procesos pueden ser **interrumpidos o liberados temporalmente** para que el CPU atienda a otros procesos.  
Cuando esto ocurre, **su estado se guarda** completamente en memoria para que, al retomarlo, **continúe desde donde fue interrumpido**.

### ****LLAMADAS AL SISTEMA****

Los procesos de usuario **se comunican con el sistema operativo** a través de las **llamadas al sistema**.  
Cada llamada tiene asociado un **procedimiento de biblioteca** que se encarga de:

* Colocar los **parámetros en lugares específicos** para el sistema.
* Realizar tareas como:
  + **Crear procesos**
  + **Controlar la memoria**
  + **Leer y escribir archivos**
  + **Operaciones de E/S** (Entrada/Salida)

👉 **E/S (Entrada/Salida)**: se refiere a toda interacción con **dispositivos externos**, como teclado, pantalla, discos, impresoras, etc.

### ****ARCHIVOS****

Los **archivos** son estructuras donde se **almacena información** de manera persistente.  
El sistema operativo, mediante llamadas al sistema, permite:

* **Crear, eliminar, abrir y cerrar archivos**.

Los archivos están organizados en **directorios**.  
Se accede a ellos mediante una **ruta de acceso** (por ejemplo: /home/usuario/documento.txt), desde el **directorio raíz**.

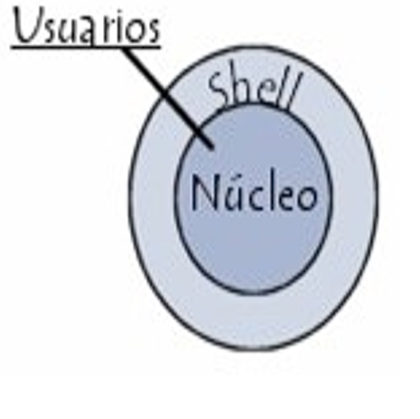
### ****SHELL****

El **shell** o **intérprete de comandos** es la **interfaz principal entre el usuario y el sistema operativo**.  
Aunque **no forma parte del núcleo (kernel)** del sistema, **utiliza muchas de sus funciones**.

Características:

* Toma **comandos del usuario** desde la **terminal** (entrada estándar).
* Ejecuta comandos, lanza programas, administra scripts.

Es como un **traductor entre el humano y el sistema operativo**.

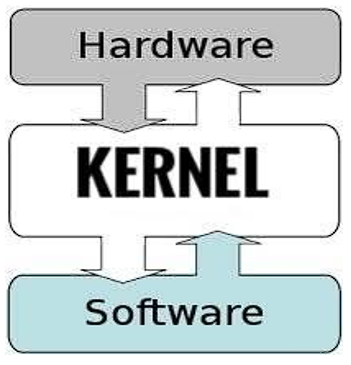


### ****KERNEL****

El **kernel** es el **núcleo del sistema operativo**, un programa **multihilo** que **reside permanentemente en memoria**.  
Se encarga de:

* **Gestión de memoria**: asignar y liberar memoria a los procesos.
* **Gestión de E/S**: controlar la entrada y salida con discos y dispositivos.
* **Gestión de procesos**: iniciar, detener, coordinar y planificar tareas en ejecución.

Es la parte **más esencial y protegida del sistema operativo**, y permite que todo lo demás funcione correctamente.



## ****CLASIFICACIÓN DE LOS SISTEMAS OPERATIVOS****

### Según su ****estructura o arquitectura interna****

### ****1. ESTRUCTURA MONOLÍTICA****

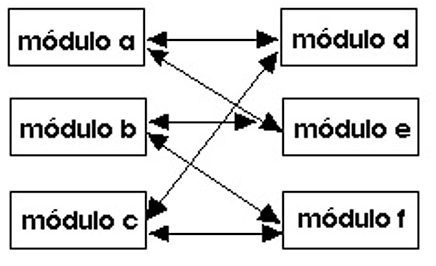
Es la **arquitectura más antigua**, usada en los primeros sistemas operativos.

* Todo el sistema está compuesto por **un único programa grande**, con muchas **rutinas entrelazadas**, donde **cualquier rutina puede llamar a otra**.
* Aunque **los módulos se pueden compilar por separado**, todos se integran en un solo bloque.

**Características:**

* ❖ Módulos **compilados separadamente**.
* ❖ Buena **definición de parámetros de enlace** entre rutinas.
* ❖ **Sin protecciones ni privilegios**: cualquier parte puede acceder a todo.
* ❖ **Hechos a medida** para hardware específico.

Ejemplo: primeros UNIX.



### ****2. ESTRUCTURA JERÁRQUICA (POR CAPAS)****

Se divide el sistema operativo en **niveles bien definidos**, donde **cada capa depende de la anterior** y proporciona servicios a la siguiente.

* Se busca **claridad, modularidad y aislamiento de funciones**.
* La **interfaz entre capas** está claramente definida.

El primer sistema en usar esta arquitectura fue **THE**, desarrollado por **Dijkstra** en Eindhoven, con fines didácticos.

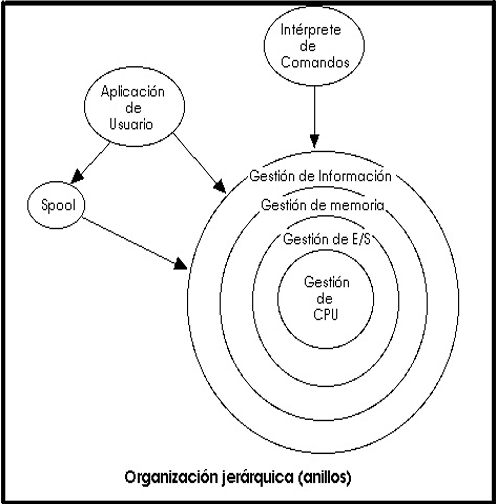


### ****3. ESTRUCTURA EN ANILLOS****

Similar a las capas, pero con un enfoque en **seguridad y protección**.

* El sistema está dividido en **anillos concéntricos**, donde:
  + **Los anillos internos** (más cercanos al núcleo) tienen **mayor privilegio**.
  + **Los anillos externos** tienen menos acceso y deben usar **"puertas" o "trampas" (traps)** para acceder a recursos protegidos.

Este modelo es común en arquitecturas como **Multics** y algunos modelos de seguridad en procesadores.



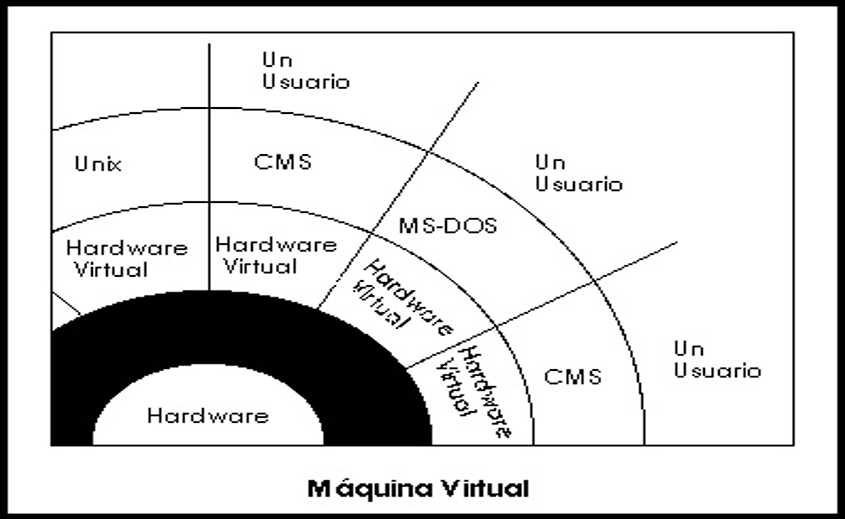
### ****4. MÁQUINA VIRTUAL****

Este sistema operativo **simula varias máquinas reales** sobre una sola máquina física.

* Cada proceso ve una **máquina virtual completa**, aislada de los demás.
* **Separa dos conceptos clave**:
  + **Multiprogramación**
  + **Máquina extendida**

**Objetivo**: Integrar distintos sistemas operativos o entornos, **dando la ilusión de múltiples máquinas independientes**.

Ejemplo: **VMware, VirtualBox, Hyper-V**.

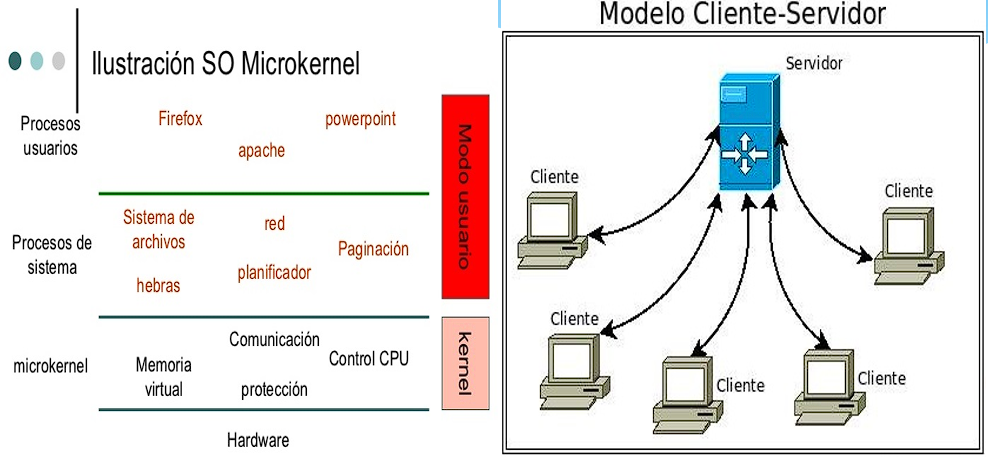


### ****5. MODELO CLIENTE – SERVIDOR (MICROKERNEL)****

En este modelo, el sistema operativo está diseñado como un **conjunto de procesos independientes**.

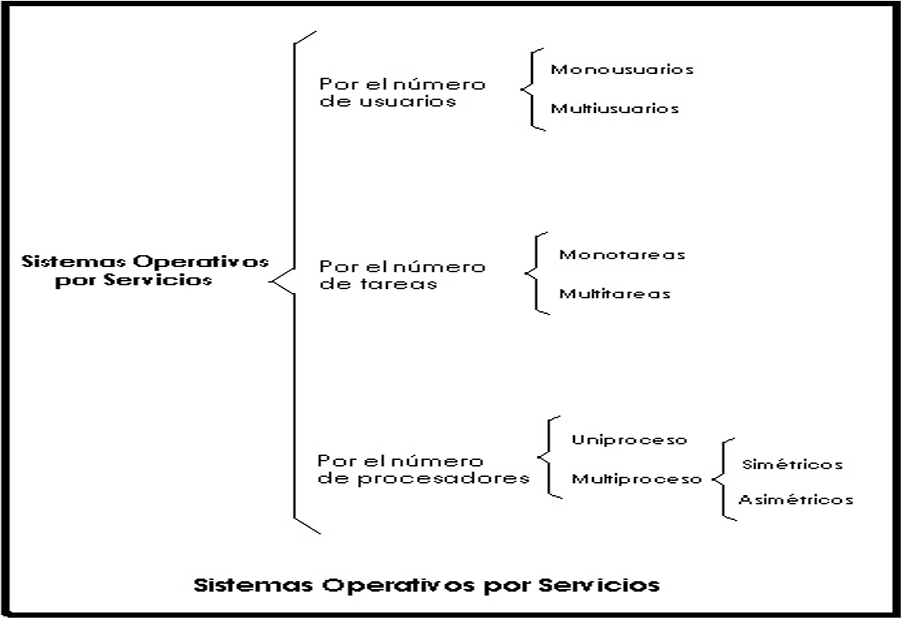
* El **núcleo (microkernel)** tiene funciones mínimas:
  + Comunicación entre procesos
  + Planificación
  + Manejo básico de memoria
* Las demás funciones del sistema (archivos, red, etc.) se implementan como **servidores separados**.
* Los **procesos pueden actuar como clientes o servidores**, comunicándose entre sí.

Ejemplo: **MINIX, QNX, algunos sistemas modernos basados en microkernel**.



# CLASIFICACION DE LOS SISTEMAS OPERATIVOS

POR SERVICIOS



### ****1. POR EL NÚMERO DE USUARIOS****

#### ❖ Monousuario

El sistema operativo **solo admite un usuario a la vez**, sin importar:

* La cantidad de **procesadores** disponibles.
* La cantidad de **procesos** que ese usuario puede ejecutar en simultáneo.

Ejemplo típico: **computadoras personales** tradicionales.

#### ❖ Multiusuario

El sistema operativo puede **atender múltiples usuarios simultáneamente**, ya sea:

* A través de **múltiples terminales físicas** conectadas.
* O mediante **sesiones remotas** a través de una red.

Cada usuario puede ejecutar múltiples procesos, **independientemente del número de procesadores** del sistema.

### ****2. POR EL NÚMERO DE TAREAS****

#### ❖ Monotarea

Solo permite ejecutar **una tarea por usuario** a la vez.

Puede existir un **sistema multiusuario y monotarea**, donde varios usuarios están conectados pero **cada uno solo puede realizar una tarea a la vez**.

#### ❖ Multitarea

Permite que **un usuario realice varias tareas al mismo tiempo**.

Por ejemplo:

* Editar un archivo
* Compilar otro programa
* Recibir emails en segundo plano (**background**)

### ****3. POR EL NÚMERO DE PROCESADORES****

#### ❖ Uniproceso

El sistema solo **aprovecha un procesador**, aunque el equipo tenga más.

Ejemplo: **MS-DOS**

#### ❖ Multiproceso

El sistema puede **utilizar varios procesadores** simultáneamente, **distribuyendo la carga de trabajo** entre ellos.

**Tipos de multiproceso:**

* **Asimétrico (AMP)**:
  + Un procesador actúa como **maestro**, coordinando la ejecución.
  + Los demás actúan como **esclavos**, recibiendo instrucciones del maestro.
* **Simétrico (SMP)**:
  + Todos los procesadores son **iguales**.
  + Los procesos (o sus hilos) pueden ejecutarse en **cualquiera de ellos**, según disponibilidad.