

# Clase 7 - Sistemas Operativos

▼ Status

Partes de una computadora

## Funcionalidades y administración

### Que son?

Un SO es un software que comprende un conjunto de programas. Es el soporte lógico que controla el funcionamiento del equipo físico. Es un conjunto de programas y funciones que ocultan los detalles de hardware, ofreciendo al usuario una vía sencilla y flexible de acceso al mismo.

La computadora requiere de un sistema operativo para funcionar.

### Funcionalidades

Administra los recursos ofrecidos por el hardware y actúa como intermediario entre computadora y usuario.

Comienza a funcionar al momento que encendemos el dispositivo y deja de funcionar cuando lo apagamos.

Es quien le da las ordenes al procesador para que lo ejecute.

Administra los dispositivos de entrada, de salida, los recursos del dispositivo y la cola de procesos.

### Recursos gestionados por el SO

- Gestiona la **RAM** y ejecuta aplicaciones, designando los recursos necesarios.
- Administra la CPU. El **procesador**.
- Direcciona la entrada y salida de datos mediante drivers.
- Administra la **información**.
- Dirige las autorizaciones de uso para el usuario.

- Administra los archivos.

## Actualizaciones

Actualizaciones: los SO son softwares diseñados para mejorar y facilitar nuestra experiencia como usuarios. Por eso se encuentran en constante mejora.

## Servidores

Los servidores también tienen sistemas operativos especiales derivados de UNIX, como Red Hat, Windows Server y Debian.

Estos últimos SO son multiusuario, es decir que varios usuarios están conectados al mismo tiempo trabajando sobre el mismo núcleo. En cambio en las computadoras domésticas son monousuario.

## Clasificación

Los tipos de sistema operativo varían según el hardware y la función de cada dispositivo. Se clasifican según su tipo de licencia:

1. **Open source:** permiten modificar, usar y adaptar un SO a voluntad del usuario. Ubuntu y Red Hat, Debian. Cualquier persona puede ver el código fuente y modificarlo en su propio equipo.
2. **Proprietary software:** son de propietarios y no permiten modificaciones, como Windows e iOS.

No todos los hardware están en capacidad de soportar o requerir el mismo SO. Ej. Una computadora con menos de 4GB de RAM no va a tolerar un SO de 64bits.

**También se pueden clasificar por otras cualidades:**

1. **Usuario**
  - Multiusuario: permite que varios usuarios ejecuten simultáneamente sus programas, ya sea por medio de terminales conectadas a la computadora o por sesiones remotas en una red de comunicación. Ejemplos: Windows, Linux, Mac OSX, Unix, Solaris.
  - Monousuario: solo permite ejecutar los programas de un usuario a la vez. No importa la cantidad de procesadores, solo podrá atender un usuario a

la vez. Windows (hasta Millenium Edition), DOS (Disk Operating Sistem).

## 2. Gestión de tareas

- Multitarea: puede ejecutar varios procesos al mismo tiempo. Son los mas comunes y modernos. Ejemplos: Windows, Linux, Mac OSX, Unix.
- Monotarea: solo ejecuta un proceso a la vez, sin que se pueda interrumpir. Son los SO mas primitivos, como Windows ME, Windows Vista y DOS. Por ejemplo, si queremos imprimir algo, no podremos ejecutar ninguna otra tarea hasta que se termine la impresión.

## 3. Gestión de recursos

- Centralizado: solo permite usar los recursos de 1 solo ordenador. Ejemplos: Windows, Linux, Mac OSX, Unix.
- Distribuido: permite ejecutar los procesos de mas de un ordenador al mismo tiempo. Ejemplos, Linux, Unix, Cisco IOS, Windows Server, Novell Netware.

## 4. Estructura Interna

- Monolítica: Constituido por un solo programa compuesto por una sola serie de **rutinas** entrelazadas entre sí. Suelen estar hechos a medida, muy rápidos pero tienen poca flexibilidad para soportar diferentes tipos de aplicaciones. Ejemplos: VMS, Linux, Multics, Windows (hasta ME).

Tiene todas sus gestiones cerca.

- Jerárquica: Se crearon porque a medida que fueron creciendo las necesidades de los usuarios y los SO crecieron, se hizo necesaria una mayor organización de SO. Conformado por partes y subpartes organizados en niveles. Se subdivide en capas o anillos bien definidos y con clara interfaz con respecto al resto de los recursos. Ejemplos: Unix, Multics.

Cada nivel gestiona una tarea.

- Máquina virtual: Separan dos conceptos que generalmente están unidos en el resto de los sistemas. La multiprogramación y la máquina extendida.  
**Objetivo: integrar distintos SO dando la sensación de ser varias máquinas**

**distintas.** Ejemplos: Microsoft Hyper-V, VMware, VirtualBox, QEMU, Kernel-Based Virtual machine.

Se instalan en un hardware y se pueden instalar muchos sistemas operativos en una máquina

- Cliente - Servidor: el más reciente. Sirve para todas las apps, es de propósito general y cumple con las mismas actividades de los SO convencionales. La idea es tener equipos por separado que tengan acceso a los recursos del servidor mediante una red. Desventaja: no resuelven los problemas de compartir información, lo que dificulta el trabajo en grupo.

Debo elegirlo en base a nuestras necesidades y capacidades del hardware.

## Kernel y llamadas al sistema

El kernel es el cerebro del sistema operativo, es una parte del SO encargado de interactuar entre las diferentes aplicaciones y sus necesidades con los recursos que posee el dispositivo para ejecutarlas.

Relaciona la CPU, la memoria y los dispositivos.

Asigna y prioriza recursos, de las aplicaciones que se ejecutan

Asigna prioridades según las necesidades del SO.

Estas interacciones se llevan a cabo a través de las **llamadas al sistema**. Son el método de las apps para solicitar un servicio o recurso. Ej: Solicitar a la impresora la impresión del doc.

El kernel esta dentro del SO y es una pt esencial del mismo. Cada SO tiene su propio kernel. Y otros se basan en kernels ya creados como Android y Linux.

Cuando algo falla, la función del kernel es interrumpir todos los procesos de la computadora para evitar el daño del SO. Todas las acciones pasan a través de un kernel, por lo que la performance del SO depende de el.

## Tipos de Kernel

Existen varios modelos de kernel que varían según el creador del SO. pero hay dos ramas:

1. **Monolítico:** Linux. Código de muchas líneas que esta alojado en un solo espacio de memoria y posee todos los drives, servicios y métodos de administración de recursos. Desventaja: desperdicio de espacio en memoria, porque al cargar el kernel se carga todo para todos los tipos de dispositivos.
2. **Microkernel:** solo posee las instrucciones básicas de la administración en un pequeño espacio de memoria y deja a cargo de los distintos dispositivos su propio manejo. Se encarga de las tareas mas básicas. Desventaja: pertenece únicamente a un dispositivo, hay que diseñar un SO por cada dispositivo con su microkernel determinado. Requiere mas lógica y programación. Hace las cosas de forma mas compleja porque no tiene los recursos que tiene el monolítico.
3. **Híbrido:** microkernel con muchas líneas de código. Windows.

Monolítico	Microkernel
El más veloz ya que se comunica con llamadas al sistema.	Más lento debido a que se comunica con paso de mensajes.
El 70% del kernel no es utilizado.	Más fácil agregar nuevas funcionalidades.
Si un sistema falla, todo el núcleo falla	Requiere más líneas de código.

Hay otras variantes que buscan explotar aun mas sus fortalezas y mejorar sus habilidades:

Kernel híbrido	Nanokernel
En esencia es un microkernel con más código "no esencial", pero menor al de un monolítico puro.	El código es aún más reducido que en microkernel, pero más difícil crear.
Agiliza la velocidad de un microkernel.	Todos los servicios se comunican con paso de mensajes
Compatible para gran variedad de dispositivos.	Fácil modificación del sistema operativo.

## Llamadas al sistema

Son la manera en la que un programa solicita una acción al sistema operativo.

Permite que las aplicaciones usen recursos del hardware.

Esta acción es el punto de enlace entre el modo usuario y el modo privilegiado (modo administrador).

## Objetivos

- **Autoriza y administra todas las acciones potencialmente riesgosas.**
- Diferenciar que acciones puede generar un usuario (modo usuario) y cuales no.
- Se restringen las acciones porque algunas pueden ser muy dañinas para el SO. Solo se pueden realizar en modo privilegiado.

## Tipos de llamadas al sistema

Nombre	Descripción
Gestión de control	Supervisa el inicio, creación, detención y finalización de los procesos.
Gestión de archivos	Incluyen la creación, eliminación, apertura, cierre, escritura y lectura de archivos.
Gestión de dispositivos	Administra los recursos disponibles, como ser el almacenamiento.
Gestión de información	Asegura la puntualidad e integridad de la información.
Comunicación entre procesos	Coordina la interacción entre los distintos procesos y aplicaciones.

## Ejemplos

### Windows

Es el sistema operativo más utilizado en todo el mundo. Fue lanzado en 1985. Al principio funcionó como una interfaz gráfica encima de MS-DOS. Luego las características de MS-DOS se integraron en Windows 95.

Posee el 88,14% del mercado.

Tiene uno de los mejores antivirus: Windows Defender.

Ideal para gaming

### Pros

- 1. Integración:** casi todas las aplicaciones del mercado son compatibles con Windows.
- 2. Soporte:** existe mucho soporte, tanto oficial como en diferentes foros y webs.
- 3. Facilidad de uso:** es muy fácil de usar, sumado a eso, la mayoría de nosotros creció utilizando alguna versión de Windows.

### Contras

- 1. Seguridad:** al ser el S.O. más utilizado, es el que más malware atrae.
- 2. Costo:** es un sistema operativo costoso.
- 3. Tamaño:** algunas versiones de Windows ocupan mucho espacio de disco.

## Mac

Este sistema operativo es más antiguo que Windows, fue lanzado en 1984. Comenzó a utilizar una interfaz gráfica desde su inicio. Todas sus actualizaciones o versiones reciben un nombre en clave para los usuarios, por ejemplo: Mojave para la del 2018, Catalina para la del 2019 y Big Sur para la del 2020.

Es el segundo SO mas utilizado.

Está basado en unix.

Solo el personal de apple puede acceder al código fuente.

Puede sincronizarse facilmente con dispositivos apple.

### Pros

- 1. Estabilidad:** debido a la minuciosa optimización entre su hardware y software el S.O. casi no presenta fallos.
- 2. Seguridad:** este S.O. es reconocido por su seguridad y menos

### Contras

- 1. Costo:** comprar una MAC suele ser muy costosa.
- 2. Compatibilidad:** a pesar que fue aumentando con el tiempo, todavía existen muchas aplicaciones no compatibles con Mac OS.

vulnerabilidad ante los malware.

**3. Diseño:** tiene un diseño muy estético y fácil de utilizar.

**3. No apto para Gamers:** en estos sistemas no corren la mayoría de los últimos juegos.

## Linux

GNU/Linux hace referencia a los sistemas operativos basados en Unix cuyo núcleo (kernel) se conoce como Linux, creado por el ingeniero de software finlandés Linus Torvalds.

Estos sistemas operativos emplean múltiples componentes y herramientas del proyecto GNU, como un ambiente de escritorio gráfico, editor de imágenes, bibliotecas para lenguajes de programación, compiladores, entre otros.

Tiene 286 distribuciones.

**Linux es software libre.**

1.34% utiliza Linux.

El de las computadoras más rápidas del mundo.

### Pros

**1. Costo:** Cualquiera puede usarlo, modificarlo y redistribuirlo sin costo.

**2. Seguridad:** Los sistemas operativos basados en Linux poseen pocos virus o estos son inexistentes.

**3. Estabilidad:** Es muy estable, tanto que es muy usado en servidores.

### Contras

**1. Complejidad:** a la mayoría se le hace difícil entenderlo y manejarlo, incluso puede volverse problemático instalar un simple programa.

**2. Compatibilidad:** muchos de los programas más utilizados en el mercado no funcionan o necesitan parches y conversiones para funcionar.

**3. Portabilidad:** la portabilidad de las distribuciones de Linux no es prioritaria para muchos desarrolladores de software y de hardware.

## Generaciones de SO



## **Generación Cero - 1940**

Las computadoras electrónicas digitales no tenían SO. Los programas manejaban 1 bit a la vez en columnas de switches mecánicos. Los programas se manejaban con un sistema de tarjetas perforadas.

## **Primera Generación - 1945 a 1955**

### **Tubos de vacío y tableros enchufables**

Se construyeron calculadoras con tubos de vacío, que ocupaban cuartos enteros y eran muy lentas. La programación se realizaba en lenguaje de máquina absoluto.

## **Segunda Generación - 1955 a 1965**

### **Transistores**

Estas máquinas se encerraban en cuartos de computadora con acondicionamiento de aire especial.

Para ejecutar un programa, un programador escribía primero el programa en papel y luego lo perforaba en tarjetas. Luego estas tarjetas se imprimían y se introducían en el lector, a veces se requería un compilador que también debía introducirse en el lector.

Alto costo.

### **Sistemas de lote**

El trabajo se grababa en una cinta magnética usando una computadora pequeña y económica. Luego la cinta se rebobinaba y se montaba en una unidad de cinta. El operador leía el primer trabajo de la cinta y lo ejecutaba. La salida se escribía en una segunda cinta, en lugar de imprimirse. Cada vez que terminaba un trabajo, el sistema operativo leía automáticamente el siguiente trabajo de la cinta y comenzaba a ejecutarlo.

## **Tercera Generación**

### **Circuitos integrados ( CI ) y multiprogramación**

Las máquinas diferían solo en el precio y el rendimiento (memoria máxima, velocidad del procesador, número de dispositivos de E/S permitidos, entre otros).

IBM trató de resolver simultáneamente ambos problemas introduciendo la System/360. Los sistemas operativos de tercera generación lograron satisfacer a sus clientes y popularizaron técnicas nuevas como la multiprogramación.

El problema era el tiempo de espera, la solución a la que se llegó fue dividir la memoria en varias secciones, con un trabajo distinto en cada partición. De esta forma se podía mantener a la CPU ocupada casi todo el tiempo y podían leer trabajos de las tarjetas al disco más rápido

Luego, cada vez que un trabajo terminaba su ejecución, el sistema operativo podía cargar uno nuevo del disco en la partición que había quedado vacía y ejecutarlo.

## **Cuarta Generación**

### **Computadoras personales**

Comienza con la invención de los circuitos integrados a gran escala (LSI), chips que contienen miles de transistores en 1 cm<sup>2</sup> de silicio.

Los dos SO principales fueron:

- MS-DOS de Microsoft: se usaba en la IBM PC, luego en intel 8088, pentium y pentium pro. Las versiones mas avanzadas tienen características de UNIX. Es el predecesor de Windows que aparece en 1995.
- UNIX: se usa en las estaciones de trabajo y otras computadoras del extremo alto, como los servidores de red. UNIX es popular en máquinas basadas en chips RISC de alto rendimiento.