

Sistemas Informáticos

UT2_1 – Sistemas operativos



César Fernández Zapico

cesarfz@educastur.org

Índice de contenido

- Sistemas operativos:
 1. Concepto de sistema operativo
 2. Componentes principales de un sistema operativo
 3. Tipos y evolución de los sistemas operativos
 4. Funciones del sistema operativo
 5. Gestores de arranque

Debate

¿Qué es un sistema operativo?



1. Concepto de sistema operativo

- Un sistema operativo es como...

*Cerebro o
Director de orquesta*

Da ordenes al HW

*Coordina todas las operaciones
pedidas optimizando resultados*

*Dependiente o
Intermediario*

*Recibe peticiones del usuario
(final o programa) y las
gestiona*

1. Concepto de sistema operativo

- Definiciones...

“Un sistema operativo es un programa cuyo objetivo es **simplificar el manejo** y la utilización de la computadora, **haciéndolo seguro y eficiente**” *Jesús Carretero*

El sistema operativo es como el director. El **responsable de coordinar todos los componentes individuales del computador**, de forma que operan juntos siguiendo un solo plan. El sistema operativo asigna recursos del computador a los diversos programas, sincroniza sus actividades y proporciona los mecanismos adecuados para que los programas se ejecuten en perfecta armonía. *Gary Nutt*

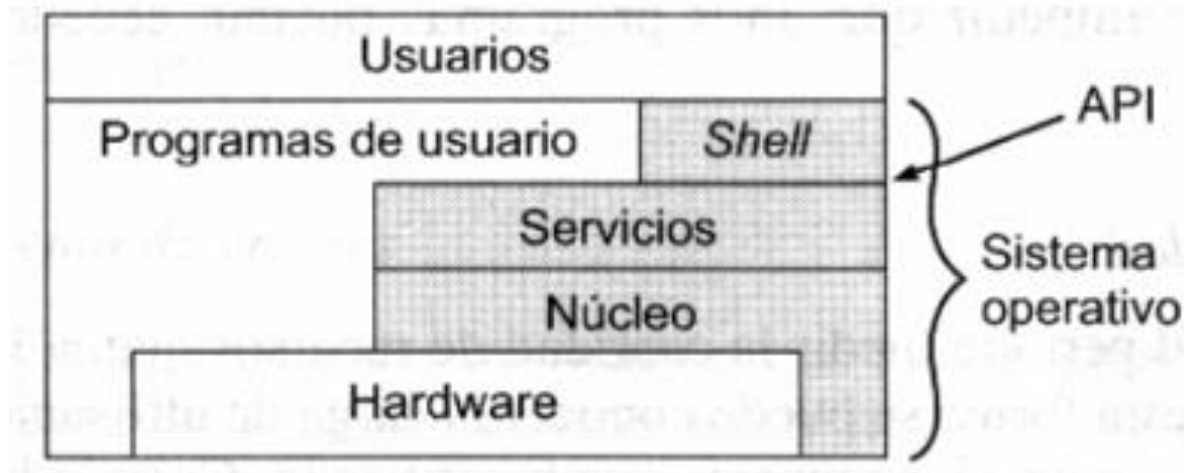
Un sistema operativo es un programa que **controla la ejecución de los programas** de aplicación y que **actúa como interfaz** entre las aplicaciones del usuario y el hardware de un computador. Puede considerarse que un sistema operativo tiene **3 objetivos: Comodidad, eficiencia y capacidad de evolución**. *William Stallings*.

1. Concepto de sistema operativo

- Un sistema operativo es un conjunto de programas de control cuya tarea principal es **administrar los recursos del sistema**.
- También debe **servir de interfaz entre el usuario y la máquina**, teniendo dos objetivos fundamentales:
 - Seguridad. El sistema operativo debe actuar contra cualquier manipulación extraña, ya sea accidental o premeditada.
 - Abstracción. Se trata de ocultar lo más posible los detalles de bajo nivel.

2. Componentes principales de un SO

- Componentes de un sistema operativo



2. Componentes principales de un SO

- El núcleo
- La API del núcleo
- El sistema de archivos
- El intérprete de comandos (Shell)

2.1. EL núcleo

- El núcleo o kernel es el componente del sistema operativo que sirve de interfaz entre el software y el hardware.
- Puede definirse como un programa que controla:
 - Los accesos al procesador y a la memoria.
 - Es responsable de los drivers más importantes.
 - Puede acceder directamente al hardware.

2.2. La API del núcleo

- La API (*Application Programming Interface*) o Interfaz de programación de aplicaciones es el **conjunto de servicios** que ofrece un sistema a las aplicaciones usuarias del mismo.
- Las aplicaciones invocan estos servicios a través de llamadas a procedimientos, y la llamada a uno de estos procedimientos recibe el nombre de **llamada al sistema**.

2.3. El sistema de archivos

- El sistema de archivos es el software que provee las funciones para operar con archivos y directorios almacenados en disco, proporcionando mecanismos de protección y seguridad.
- Es un componente del sistema operativo cuyas principales funciones son:
 - La asignación de espacio a los archivos.
 - La administración del espacio libre.
 - El acceso a los datos guardados
- [Sistemas de archivos – web de Xataka](#)

2.4. El intérprete de comandos (Shell)

- El intérprete de comandos (Shell) es el programa que se encarga de leer las órdenes interactivas que el usuario escribe en la terminal y lo convierte en instrucciones para el sistema operativo.

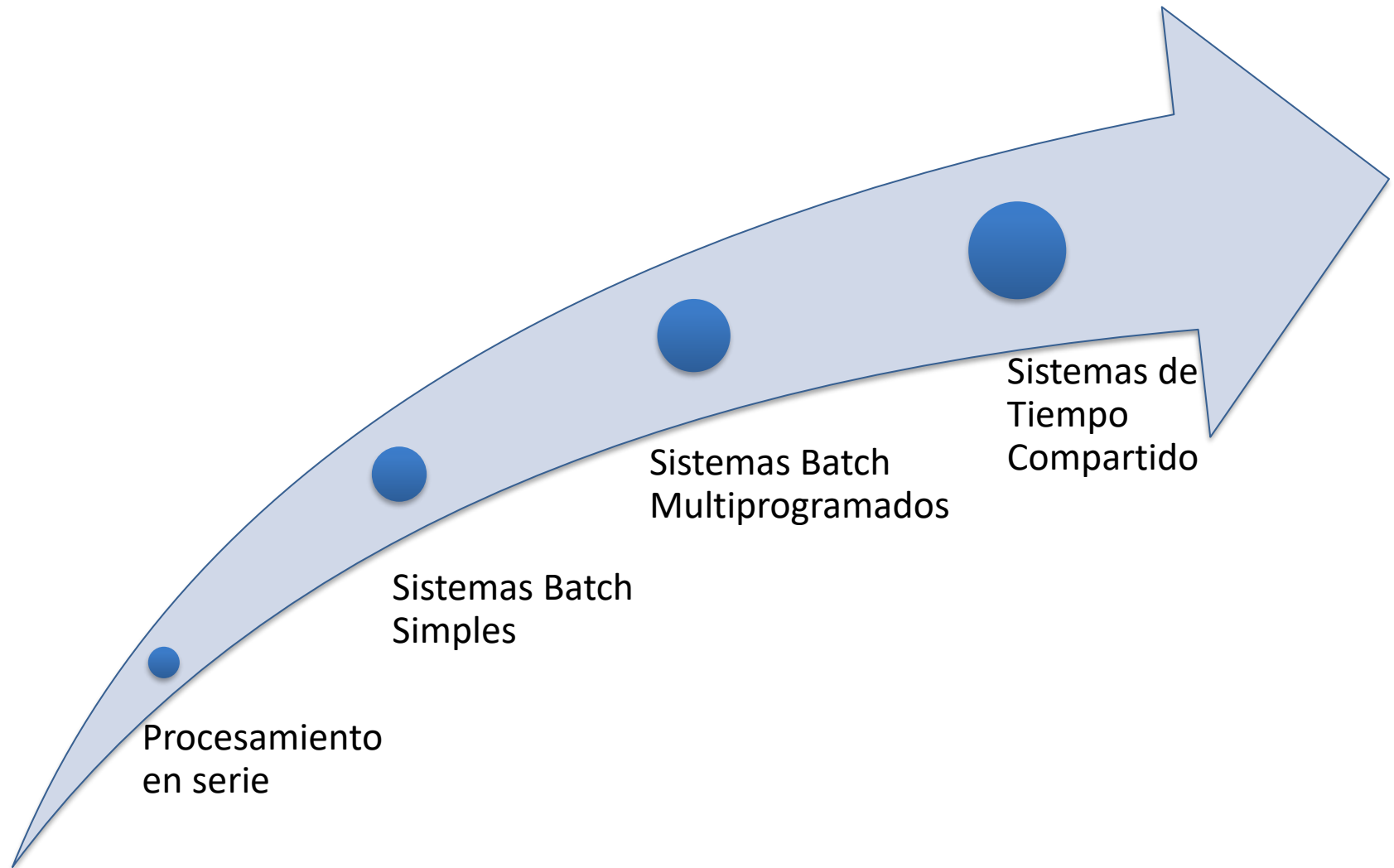
3. Tipos de sistemas operativos

- Según su funcionalidad:
 - Monousuario / Multiusuario
 - Monoprogramación o monotarea / Multiprogramación o multitarea
 - De tiempo real
 - Interactivos
 - Por lotes (batch)
 - De propósito general
 - De propósito específico
 - Para móviles (Android, iOS, Ubuntu Touch, etc.)

3. Tipos de sistemas operativos

- Según su estructura:
 - Sistemas operativos monolíticos
 - Funcionalidad en un único programa
 - Difícil modificación
 - Sistemas operativos estructurados
 - Por capas
 - Micronúcleos (Cliente/servidor)
 - Sistemas operativos Híbridos
 - Modular
 - Micronúcleo no puro
 - Sistemas operativos distribuidos
 - **Middleware** (Software para que varias aplicaciones se comuniquen entre sí)

3. Evolución de los sistemas operativos



3. Evolución de los sistemas operativos

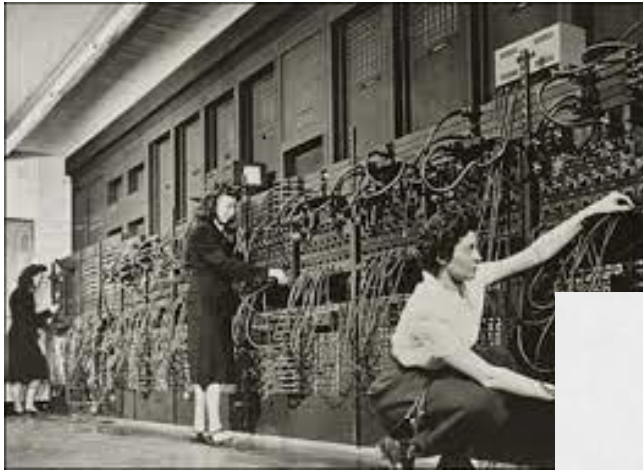
Época	Características del Sistema Operativo	HW/ SO/lenguajes
Años 40 Prehistoria	Inexistente. Procesamiento en serie	Válvulas Arq. Von Neumann EDVAC ENIAC (1943) Ensamblador
Años 50 1ª Generación	Carga y ejecución de programas Procesamiento de trabajos por lotes (Batch) Rutinas de E/S (manejadores de dispositivos) Recuperación de errores Lenguaje de control	Transistores IBSYS (IBM) FMS (IBM) FORTRAN, COBOL

3. Evolución de los sistemas operativos

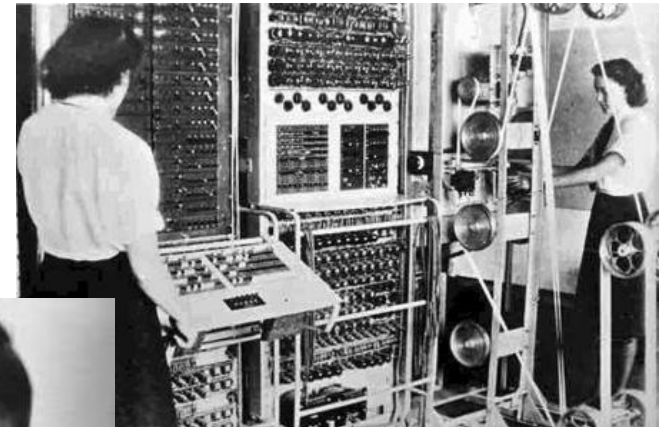
COLOSUS fue usada por Alan Turing para descifrar mensajes de los alemanes en la Segunda Guerra Mundial

ENIAC fue desarrollada por el departamento de balística de los EEUU

ENIAC (*Electronic Numerical Integrator And Computer*)



COLOSUS



3. Evolución de los sistemas operativos

Época	Características del Sistema Operativo	HW/SO/lenguajes
Años 60 2ª Generación	Incrementar rendimiento: <ul style="list-style-type: none">• Multiprogramación• Multiusuario interactivo: Tiempo compartido• Tiempo Real• Multiprocesador Grandes y costosos Complejo lenguaje de control	Circuitos integrados OS/360(IBM 360) MULTICS – UNIX BASIC, ALGOL

3. Evolución de los sistemas operativos

Época	Características del Sistema Operativo	HW/SO/lenguajes
Años 70	<p>SO de propósito general Difusión de multiusuario y tiempo compartido</p> <p><u>Unix</u></p> <ul style="list-style-type: none">•Laboratorios Bell ATT•Ken Thompson, Dennis Ritchie•Implementación en Lenguaje C (Dennis Ritchie) (1973)•Difusión a lab. y universidades (cod. Fuente)•Distintas distribuciones:<ul style="list-style-type: none">•Aparición de BSD (Uni. Berkeley)•Aparición de System V (Lab Bell)•Otros fabricantes (Sun, HP, IBM)...	<p>Apple II Intel 8008</p> <p>UNIX (Bell 1976)</p>

3. Evolución de los sistemas operativos

Época	Características del Sistema Operativo	HW/SO/lenguajes
Años 80	Simplificación de los SO. Importancia usuario Gestión de redes Sistemas operativos de red Interfaces gráficas Diseño interno Orientado a Objetos Diferentes SO para diferentes procesadores de ordenadores personales Sistemas Operativos Distribuidos	UNIX MS-DOS Windows Mc OS, OS/2

3. Evolución de los sistemas operativos

Época	Características del Sistema Operativo	SO ejemplo
Años 90	Sistemas Operativos Libres Sistemas Operativos de tiempo real Arquitectura cliente/servidor Estandarización de Interfaces Seguridad	Linux Windows XP, NT.. POSIX

3. Evolución de los sistemas operativos

Época	Características del Sistema Operativo	SO ejemplo
Presente y futuro	<p>Sistemas operativos empotrados: móviles, PDAs (casi obsoletas), tabletas...</p> <p>Computación distribuida</p> <p>Tolerancia de fallos</p> <p>Desarrollo de nuevas interfaces</p> <p>Personalización y usabilidad</p> <p>Diseños Orientados a Objetos</p> <p>Arquitectura cliente servidor distribuida</p> <p>Incorporación de múltiples utilidades de seguridad y acceso remoto</p> <p>Mejora notable de la seguridad</p>	<p>Android, IOS</p> <p>Linux (RedHat)</p> <p>Windows 8, 10, 11</p>

4. Funciones del sistema operativo

- Funciones principales del sistema operativo:
 - Gestión de procesos
 - Gestión de memoria
 - Gestión de dispositivos de E/S
 - Gestión del sistema de ficheros

4.1. Gestión de procesos (UT2_2)

- Un **proceso** es un programa en ejecución, es decir, debe entenderse como una **entidad dinámica**.
- Tareas para la gestión de procesos:
 - Sincronización y comunicación entre procesos
 - Planificación de procesos ➔ Algoritmos

4.2. Gestión de memoria (UT2_3)

- Para ejecutar un proceso es necesario **asignarle memoria** y cargarlo en ella, liberándola cuando finaliza su ejecución.
- Gestión de memoria ➔ Transparente al usuario:
 - Registro de la memoria libre.
 - Control sobre el intercambio de información entre la memoria y el disco.

4.3. Gestión de dispositivos de E/S

- Técnicas para la gestión de E/S:
 - **Spooling**
 - Multiprogramación
 - Convertir recursos exclusivos en compartidos
 - **Buffering**
 - Buffers de E/S → Espacios de memoria principal que se reservan para el almacenamiento intermedio de datos
 - **Caching**
 - Utilizar memoria caché
 - Bit de caching

4.4. Gestión de archivos y dispositivos

- Gestión de directorios y archivos
 - Directorios ➔ Ruta absoluta y relativa
 - Archivos ➔ Organización lógica y física
- a) Organización lógica de los archivos
 - Organización secuencial
 - Organización directa
 - Organización secuencial indexada
 - Organización secuencial encadenada
 - Organización secuencial indexo-encadenada

4.4. Gestión de archivos y dispositivos

- Gestión de directorios y archivos
 - b) Organización física de los archivos ➔ Sistemas de archivos
 - Windows: FAT – FAT32 – NTFS – exFAT
 - Linux: ext2 – ext3 – ext4 – swap – F2FS – ReiserFS
 - MacOS: HFS – HFS+ – APFS

4.4. Gestión de archivos y dispositivos

- Gestión de dispositivos
 - Técnicas para la gestión de E/S
 - Spooling
 - Buffering
 - Caching
 - Técnicas para la gestión de discos duros HDD
 - Reducción del movimiento del brazo del disco
 - Almacenamiento en sectores vecinos
 - Planificación del movimiento de las cabezas lectoescritoras

5. Gestores de arranque

- El **gestor de arranque (Bootloader)** es un pequeño programa que se ejecuta una vez completado el inicio normal de la BIOS y que permite seleccionar el sistema operativo en caso de disponer de arranque múltiple.
- Los bootloaders sirven como intermediarios entre el hardware y el sistema operativo.

5. Gestores de arranque

- Gestores de arranque populares:
 - **NTLDR** (NT Loader) ➔ Windows XP – Windows Server 2003
 - **Bootmgr** ➔ Windows Vista – Windows 7 – Windows Server 2008
 - **Lilo** (Linux Loader) ➔ Gestor de arranque de Linux
 - **Grub** (Grand Unified Bootloader)
- [Gestores de arranque \(Bootloaders\)](#)