



UF2 - Introducción a los Sistemas Operativos

SISTEMAS INFORMÁTICOS

Índice

Introducción

Características de un SO

Recursos del sistema informático

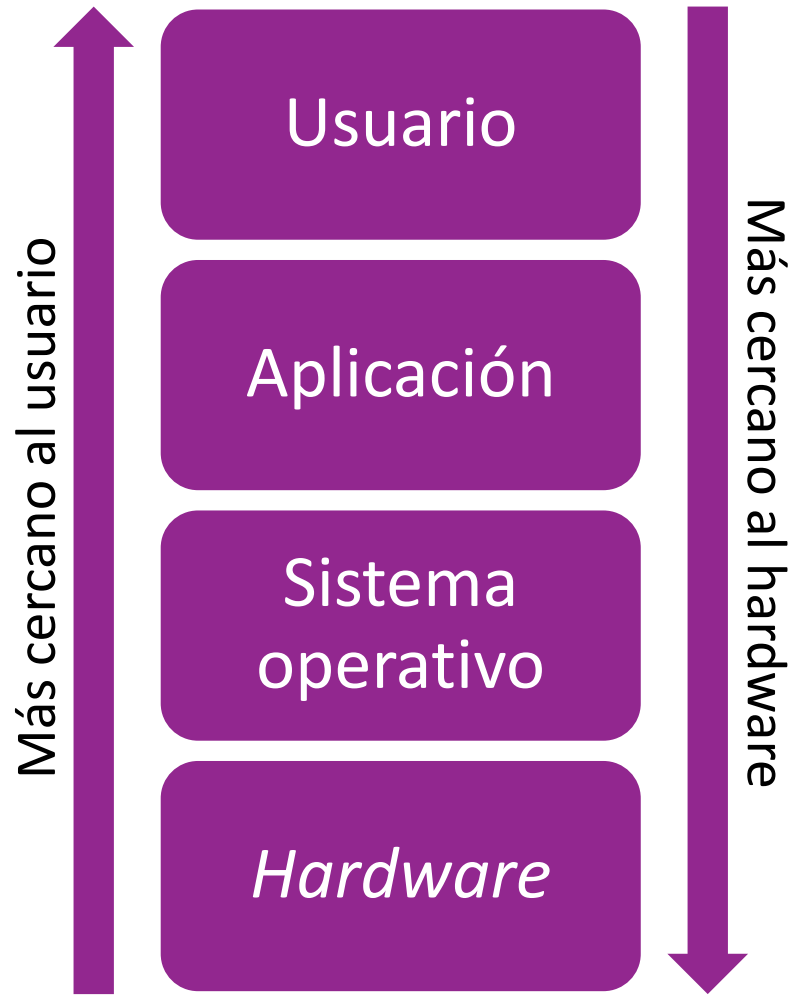
Funciones del SO

Tipos de sistemas operativos

Familias de sistemas operativos

Instalación y arranque de un sistema operativo

Actualizaciones, parches y drivers



Introducción

Cuando trabajamos con un **sistema operativo (SO)** en un sistema informático, el usuario no se tiene que preocupar de las direcciones de memoria RAM usadas, la gestión de las interrupciones, etc.

Hoy en día los SOs están compuestos por software muy avanzado que trata de facilitar el uso del dispositivo al usuario lo máximo posible gestionando el hardware lo más eficientemente posible.

Características de un SO

Las funciones básicas de un sistema operativo son:

- Actuar de **interfaz** entre el usuario y el *hardware* de manera transparente para el usuario, de forma que este no tenga que ser un especialista en *software* o *hardware* para usar el SI.
- **Gestionar los recursos** *hardware* y *software* del equipo para un uso eficiente.

Por ello, un SO debe tener las siguientes características:

- **Adaptabilidad:** hacer uso de nuevas características y mejoras del nuevo *software* y el nuevo *hardware*.
- **Facilidad de uso:** la facilidad de manejo es primordial, aunque una mayor comodidad puede implicar mayor gasto de recursos.
- **Eficiencia:** los recursos (memoria principal, CPUs, acceso a discos, red, etc.) son limitados, y el SO debe atender todas los accesos de los usuarios, programas y el propio SO a esos recursos, lo cual es una tarea muy compleja de realizar eficientemente.

Recursos del sistema informático

El sistema operativo debe gestionar de forma eficiente recursos del sistema informático como:

- **Memoria RAM:** cada vez que se ejecuta un programa, este se carga en memoria, y cuando deja de ejecutarse, se libera ese espacio de la memoria. El núcleo o *kernel* del sistema operativo siempre está en memoria.
- **Procesador:** gran cantidad de procesos van a ejecutarse en un pequeño número de núcleos de CPU.
- **Adaptadores de red:** múltiples procesos pueden hacer uso de la red simultáneamente.
- **Medios de almacenamiento:** múltiples procesos suelen solicitar acceso simultáneo a disco.
- **Colas de impresión, etc.**

Funciones del SO

- **Gestión de procesos:** el SO ha de repartir el tiempo del procesador (de cada núcleo) entre los procesos que desean ejecutarse, por lo que debe gestionar:
 - La asignación de procesos a cada núcleo del procesador.
 - El uso de la multiprogramación sobre núcleos individuales.
 - La sincronización entre procesos.

Estos objetivos son definidos por políticas de planificación con orientaciones diferentes:

- **Planificación orientada a los usuarios (orientado a las entradas y salidas):** intenta agilizar los accesos a discos, entradas de pantallas táctiles o accesos a internet. Prima el tiempo de respuesta a los usuarios.
- **Planificación orientada al sistema (orientada a procesos de cálculo):** su objetivo es la eficiencia y el rendimiento de procesos que realizan cálculos complejos.

Funciones del SO

- **Gestión de memoria:** íntimamente ligada a la gestión de procesos, ya que cada proceso demanda su espacio de memoria (código y datos). Hoy en día los sistemas disponen de memoria RAM suficiente para albergar el sistema operativo y mucho más, pero en caso de necesitar mayor espacio de memoria se emplea espacio del disco duro (denominado **memoria virtual**) mediante un *archivo de paginación* (Windows) o una *partición SWAP* (Linux), que deben ser 1 o 1.5 veces el tamaño de la RAM.
- **Gestión de entradas y salidas:** engloba acciones como tocar una pantalla táctil, imprimir un documento, acceder a un fichero en disco o navegar por internet. El SO debe gestionar el acceso a estos dispositivos.
- **Gestión de ficheros:** un proceso no lee ni escribe archivos y carpetas directamente, sino que son tareas que solicita al SO y es este el que las realiza. La estructura organizativa de los archivos viene determinada por los sistemas de ficheros.

Funciones del SO

- **Gestión de la seguridad:** el SO debe evitar inconsistencias originadas por errores del software o del usuario, o incluso por acciones malintencionadas. Por ello, debe garantizar:
 - El servicio y la disponibilidad de sus recursos.
 - La confidencialidad e integridad del sistema y los datos.
 - El control de accesos.
- **Gestión de los errores:** el SO debe gestionar todo tipo de errores de la forma más liviana posible, informando al usuario y salvaguardando de forma prioritaria la seguridad del sistema y los datos.
- **Gestión de la interfaz de usuario:** el SO es el encargado de ofrecer un soporte que permita una fluida comunicación con los usuarios, de forma que estos puedan realizar todas las acciones necesarias para la gestión, administración o explotación del SI.

Tipos de sistemas operativos

Según su interfaz de usuario:

- **Sistemas operativos de escritorio:** tienen **interfaz gráfica de usuario (GUI)** y aplicaciones que hacen uso de ella. Se usan en equipos que van a ser usados directamente por un usuario con pantalla, ratón y teclado. A su vez, pueden ser de propósito general o específico.
- **Sistemas operativos de servidor:** a menudo no tienen GUI y para interactuar directamente con ellos es necesario usar la **interfaz de línea de comandos (CLI)**, local o remota (SSH), que es una pantalla con solo texto donde el usuario escribe comandos y recibe respuestas textuales.
- **Sistemas operativos por lotes, *batch* o no interactivos:** se les suministra un conjunto de tareas y las ejecutan sin la intervención del usuario. No tienen ninguna clase de interfaz de usuario.

Tipos de sistemas operativos

Según sus servicios:

- **Sistemas operativos mono-/multi-usuario:** si el SO permite la creación y gestión de usuarios y les permite a estos el uso del SO, es multiusuario. Si no, es monousuario. Hoy en día todos los sistemas operativos son multiusuario.
- **Sistemas operativos mono-/multi-tarea:** si el SO es capaz de ejecutar varios programas y rutinas simultáneamente, es multitarea. Si no, es monotarea.
- **Sistemas operativos mono-/multi-proceso:** si el SO es capaz de ejecutar instrucciones en varias CPUs al mismo tiempo, es multiproceso. Si no, es monoproceso.

Casi todos los SOs de escritorio y servidor modernos son multiusuario, multitarea y multiproceso.

Tipos de sistemas operativos

Según las tareas que se quieren realizar mediante un sistema operativo, este se considera:

- **Sistemas operativos interactivos:** sistemas de propósito general orientados a la participación continua del usuario. Hacen uso de programas interactivos como navegador web, procesador de texto o editor de imágenes. Por ejemplo: Windows, Ubuntu, MacOS X.
- **Sistemas operativos en tiempo real:** diseñados para gestionar tareas que deben cumplir escrupulosamente unos plazos. Siempre deben tener un comportamiento predecible. Por ejemplo, SOs para sistemas de alertas, simulaciones matemáticas, control industrial, robótica, sistemas multimedia, etc.

Tipos de sistemas operativos

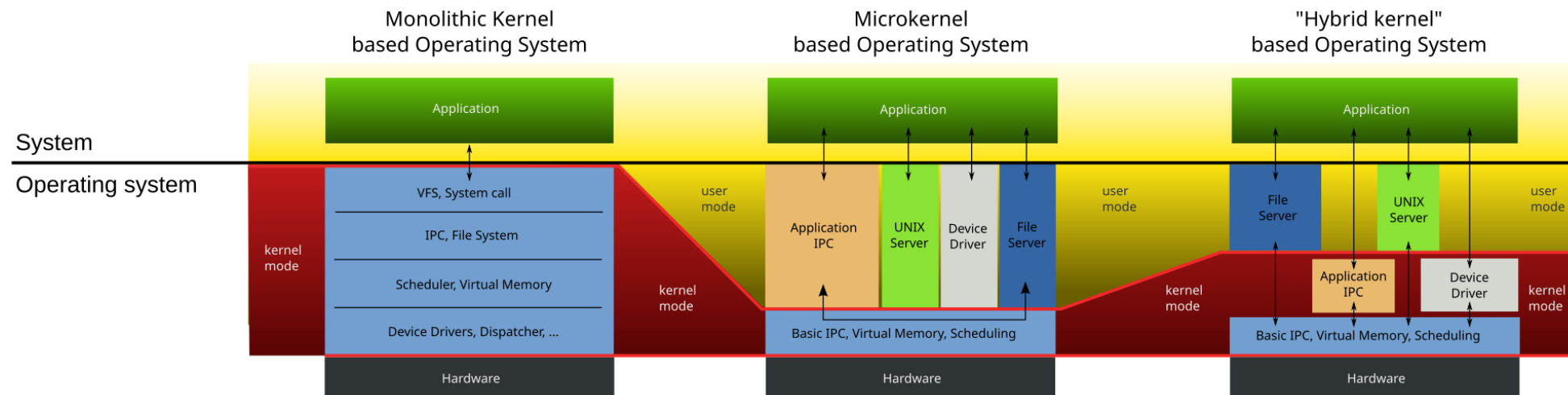
Según su forma de ofrecer los servicios:

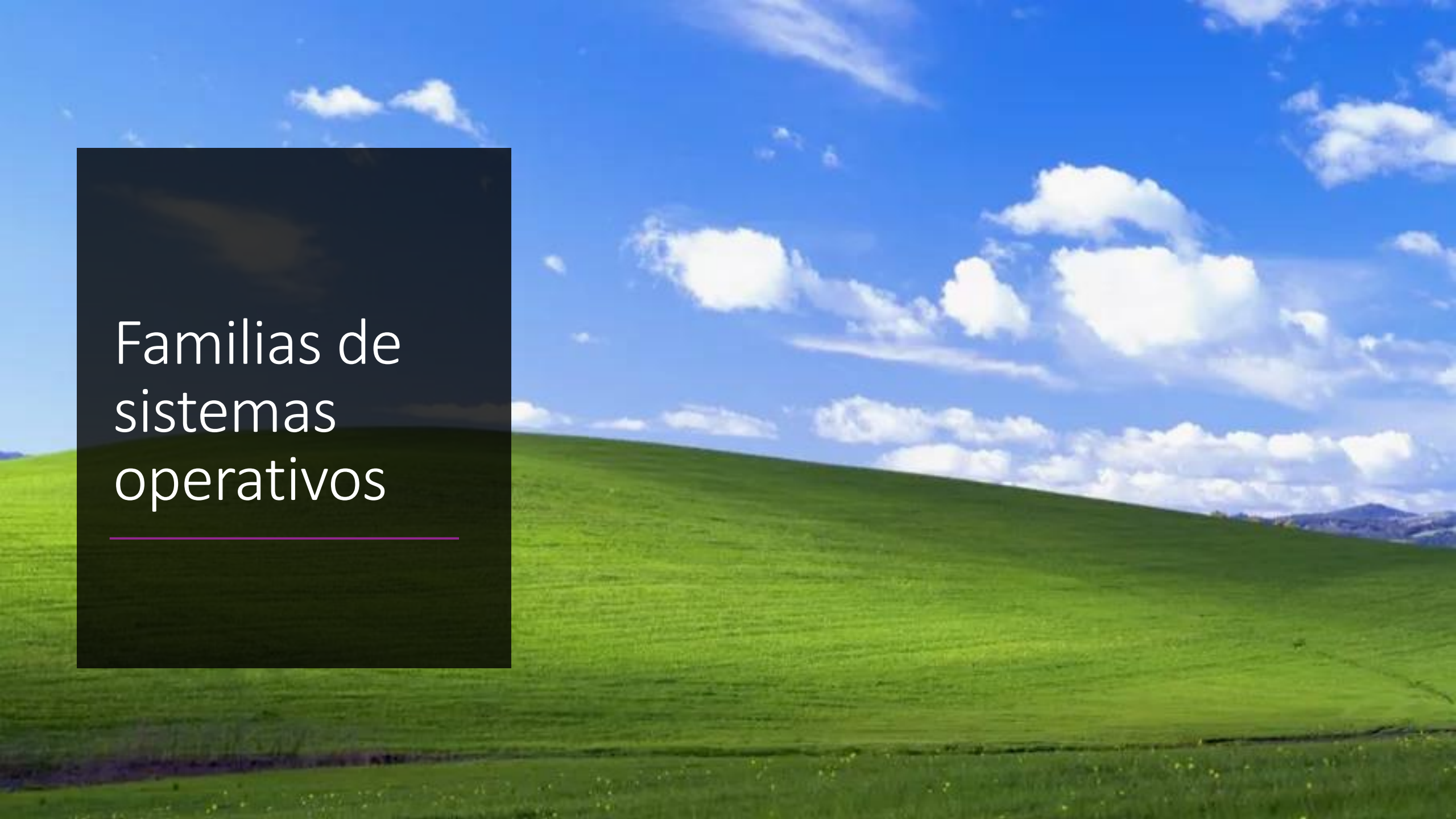
- **Sistemas operativos cliente:** realizan las actividades en una sola computadora, que suele estar conectada en red, pero para realizar accesos externos por parte del usuario. Son normalmente empleados por hogares y pequeñas empresas.
- **Sistemas operativos en red:** gestionan la red, los usuarios y los recursos de una red de ordenadores de forma centralizada mediante uno o varios servidores (réplicas o extensiones). Son estos servidores los que instalan estos sistemas operativos, mientras que el resto de equipos de la red usan sistemas operativos cliente y se conectan a estos servidores. Así, se centraliza el intercambio de información, aunque por eso mismo el/los servidores pueden resultar un cuello de botella. Destacan por su seguridad y robustez.
- **Sistemas operativos distribuidos:** a diferencia de los anteriores, varios computadores actúan de forma transparente al usuario, de forma que da la sensación de que interactúa con uno solo. Permiten emplear los recursos de varias computadoras en paralelo (redundancia y/o balanceo de carga).

Tipos de sistemas operativos

Según su arquitectura:

- **Monolítica:** consiste en un programa único, el *kernel*, que realiza todas las funciones que debe realizar el SO en *espacio kernel* (con privilegios de administrador, con acceso al hardware). Es la estructura original de los primeros sistemas operativos. Ejemplo: Linux.
- **Microkernel:** el *kernel* tiene la cantidad mínima de *software* que realiza las funciones básicas del SO. El resto de servicios del SO las realizan programas en *espacio usuario* (sin privilegios de administrador).
- **Kernel híbrido:** algunas funciones del sistema se ejecutan en *espacio kernel* y otras en *espacio usuario*.



The background of the slide is the iconic Windows XP desktop wallpaper, featuring rolling green hills under a bright blue sky with scattered white clouds. A semi-transparent black rectangle is positioned on the left side of the image, serving as a backdrop for the title text.

Familias de sistemas operativos

Sistemas operativos de Microsoft

Para equipos de escritorio: Windows 11. Incluye varias ediciones con menores o mayores prestaciones orientadas a diferentes ámbitos:

- Home: para equipos de sobremesa, tabletas o portátiles de poca potencia a los que se les da un uso básico.
- Pro: orientado a fines de negocio para profesionales o empresas.
- Enterprise: también orientado a fines de negocio, pero de mayor volumen.
- IoT, Education, Pro for Workstations, etc.



Para servidores: Windows Server 2022. Incluye, entre otras, las siguientes ediciones:

- Datacenter: para entornos en la nube o centros de datos altamente virtualizados.
- Standard: para ambientes poco virtualizados.
- Essentials: para pequeños negocios con un número limitado de usuarios y dispositivos.

Sistemas operativos GNU/Linux

Con GNU/Linux nos referimos a una familia de SOs tipo UNIX que se construyen sobre el **kernel de Linux**. A estos SOs los llamamos **distribuciones de Linux**.

Algunas de las más populares para **entorno de escritorio** son:

- Debian
- Ubuntu
- Linux Mint
- Arch Linux
- Kali Linux
- Chromium OS



Algunas de las distribuciones más populares para **entorno servidor**:

- Red Hat Enterprise Linux (RHEL)
- Ubuntu Server
- Debian
- CentOS
- SUSE Linux Enterprise Server



Android, uno de los SOs para smartphones y tablets más populares, también es una distribución Linux.



Sistemas operativos de Apple

Los equipos informáticos del fabricante Apple tienen sus propios sistemas operativos:

- **macOS X:** para ordenadores de sobremesa y portátiles Apple.
- **iOS:** para *smartphones* de Apple (iPhone).



Instalación y arranque de un sistema operativo

Graphical install

Install

Advanced options ...

Accessible dark contrast installer menu ..

Install with speech synthesis

Requisitos *hardware*

Los **requisitos *hardware*** son las características físicas que el proveedor o el desarrollador del *software* considera como mínimas o recomendadas para que el SO funcione de forma correcta.

Esta información suele ser resultado de pruebas de rendimiento que realiza el desarrollador sobre distintas máquinas para que el *software* funcione de forma óptima.

- Los **requisitos mínimos** son los requisitos que debe cumplir el hardware para que el *software* funcione, al menos, de forma ajustada.
- En muchos casos también existen **requisitos recomendados**, que aseguran que el *software* funcione con un rendimiento óptimo en la máquina objetivo.

En el caso de que se superen con creces los requisitos *hardware*, es posible que el software se ejecute correctamente pero que funcione con ciertos errores relacionados con la velocidad del procesador o las características gráficas o sonoras.

Requisitos *hardware*

Los requisitos *hardware* a la hora de instalar un SO suelen incluir:

- **Procesador:** es muy importante atender a la arquitectura (x86, x86_64, ARM64, etc.), cantidad de núcleos y la frecuencia del procesador (en GHz).
 - Aunque un mismo SO soporte múltiples arquitecturas, cada arquitectura tiene su propio enlace de descarga en la página de descarga de la ISO, por lo que es fundamental escoger la adecuada.
- **Memoria principal (RAM):** siempre se detalla la capacidad mínima de memoria RAM que se necesita para el funcionamiento estable del SO.
- **Memoria secundaria (almacenamiento):** es necesario que se especifique el espacio de almacenamiento que SO necesita para copiar los archivos del sistema y que este funcione correctamente.
- Otras características secundarias como capacidades gráficas, de sonido, de redes, etc.

Si la instalación del SO es en una máquina virtual, los requisitos fundamentales son la cantidad de memoria RAM y de disco duro (virtual).

Interfaces de *firmware*

El ***firmware* del SI** es instalado por el fabricante y contiene las instrucciones que se deben ejecutar durante el arranque antes de ceder el control del *hardware* a un medio de arranque.

Originalmente se utilizaba la BIOS, pero en la actualidad se trabaja mayoritariamente con UEFI.

- **BIOS (*Basic Input/Output System*)** nació en 1975 y no tuvo interfaz gráfica hasta la década de los 80.
- **UEFI (*Unified Extensible Firmware Interface*)** es el estándar más reciente (2007) que sustituye a la BIOS. Incorpora múltiples ventajas:
 - Puede acceder a la configuración de dispositivos.
 - Puede conectarse a la red.
 - Puede comprobar la originalidad del núcleo del SO antes de cargarlo.
 - Arranque más rápido.
 - Ofrece una GUI más amigable con el usuario.

Es común referirse al *firmware* del SI como BIOS independientemente de si usa la interfaz BIOS o UEFI.

Modos de arranque en sistemas UEFI

Los sistemas UEFI pueden establecer uno de los siguientes modos de arranque:

- **Heredado o *legacy* BIOS**: establece la retrocompatibilidad con discos con esquemas MBR, por lo que no se hacen efectivas las ventajas del estándar UEFI.
- **UEFI**: es el modo que recomienda la mayoría de los sistemas operativos. Usa el formato GPT.

Durante la instalación de los sistemas operativos, se crean por defecto las tablas de partición con el esquema adecuado según el modo de arranque.

Si un equipo con el modo UEFI (*legacy*) inicia un medio de arranque con un esquema de particiones MBR (GPT) que no le corresponde, no lo reconocerá.

Particiones

Los discos se deben particionar para organizar la información convenientemente, almacenar programas y datos e instalar SOs. Cuando adquirimos un disco duro, normalmente se encuentra particionado, pero podemos realizar particiones libremente.

Llamamos **partición** a una división del espacio de almacenamiento de un disco duro.

Para almacenar información en una partición, esta debe tener un sistema de archivos en uso. En cada partición se puede instalar un único **formato o sistema de archivos**. A este proceso se le conoce como ***formatear***.

Ejemplos de sistemas de archivos: ext4, NTFS, FAT32, APFS y ExFAT.

A cada partición se le da un cierto uso, principalmente:

- Alojarse un sistema operativo
- Almacenamiento de archivos
- Memoria de intercambio

Esquemas de particiones

Los discos duros pueden estructurar sus particiones atendiendo a dos estándares o **formatos de tabla de particiones**:

- **MBR (*Master Boot Record*)**: es el esquema de particionado de los sistemas con estándar BIOS. Tiene algunas limitaciones:
 - Solo permite tener 4 particiones primarias.
 - Solo soporta discos duros de hasta 2TB de capacidad.
- **GPT (*GUID Partition Table*)**: es el esquema de particionado de los sistemas con estándar UEFI. A cada partición se le asigna un identificador global único (GUID). Tiene algunas mejoras respecto a MBR:
 - Soporta hasta 128 particiones primarias.
 - Soporta discos de tamaño superior a 2TB.
 - Tiene un gestor de arranque propio, no vinculado a ningún sistema operativo.

Los discos con formato GPT deben tener una partición de sistema EFI (ESP), en formato FAT32 y que suele ser la primera, donde se encuentran los cargadores de arranque de los SOs y otras utilidades.

Proceso de arranque (segunda parte)

Después del POST (visto en la UF1), se ejecutan las instrucciones del firmware de la ROM BIOS. En este momento, aparecerá una pantalla que nos mostrará mensajes para poder entrar en el **menú de configuración BIOS (*BIOS setup utility*)** o en el **menú de arranque (*boot menu*)**, donde podremos elegir manualmente qué dispositivo cargar.

Si se deja pasar la pantalla sin entrar en un menú, la BIOS seleccionará un **medio de arranque** de entre una lista de medios (***boot order***) de donde **escogerá el primero que esté disponible**. Se puede alterar su orden en el *BIOS setup utility*.

- Los medios de arranque pueden ser discos, CD/DVD, unidades flash, la red, etc.
- Si no se encuentra ningún dispositivo disponible en la lista, no se iniciará el SO.

Proceso de arranque (segunda parte)

Al elegir un medio de arranque en la fase anterior, se ejecutará el **cargador de arranque (*boot loader*)** del dispositivo o partición, que son un conjunto de instrucciones que se encarga de:

- **Mostrar un menú** que permite elegir qué SO cargar (si hay más de una partición con SO) y qué configuración para ese SO. En muchas ocasiones no existe el menú porque solo hay un SO que cargar.
- **Iniciar el SO**. En caso de que el SO no fuera compatible con el cargador de arranque, no iniciará el SO directamente, sino que se limitará a lanzar el cargador de arranque de ese SO.

Cargadores de arranque conocidos:

- **BOOTMGR** es el *bootloader* de sistemas Windows desde la versión Vista.
- **Grub 2.0** es el *bootloader* de los sistemas Linux.

Instalación de un SO

Para poder arrancar el equipo con un SO determinado, este debe estar instalado en un dispositivo de almacenamiento (o en una de sus particiones), ya sea en un disco duro interno (como es habitual) o en un dispositivo extraíble.

Para llevar a cabo la instalación de un SO, se deben realizar los siguientes pasos:

- Escoger y **descargar la imagen en formato ISO** de la web oficial del desarrollador del SO. Debe coincidir (o ser compatible) con la arquitectura del SI donde se instalará.
- Configurar una unidad óptica o flash como **medio de instalación**, que se formateará y desplegarán en ella los contenidos de la imagen ISO.
- **Arrancar la máquina con dicho medio de instalación** (se puede seleccionar mediante el *boot menu*).
- **Realizar los pasos de la instalación** del SO, cuyo instalador tendrá su propio cargador de arranque, que en ocasiones nos ofrece opciones como:
 - Arrancar con el Sistema Operativo en **modo live**, para probarlo o para realizar alguna tarea en dicho SO sin instalarlo.
 - Iniciar la instalación con GUI o sin GUI.

Actualizaciones, parches y *drivers*

Cualquier programa, *driver*, librería o SO recibe nuevas funciones y se descubren fallos y vulnerabilidades que el desarrollador corrige creando **parches** y ofreciendo **actualizaciones**.

En general, existen dos clases de actualización de software:

- **Actualizaciones de versiones:** suelen añadir nuevas características. No son prioritarias.
- **Actualizaciones de seguridad:** instalan parches que corrigen errores y vulnerabilidades. Es importante no demorarse en instalarlas.

En Windows, las actualizaciones del sistema operativo y los *drivers* (y algunas aplicaciones) las gestiona Windows Update, y suelen buscarse e instalarse automáticamente. El resto de programas deben actualizarse de otras formas.

En Linux, las actualizaciones de paquetes las gestiona el gestor de paquetes de cada familia de distribuciones. Por ejemplo, Debian usa `apt`, Red Hat usa `yum` y Arch Linux usa `pacman`.