

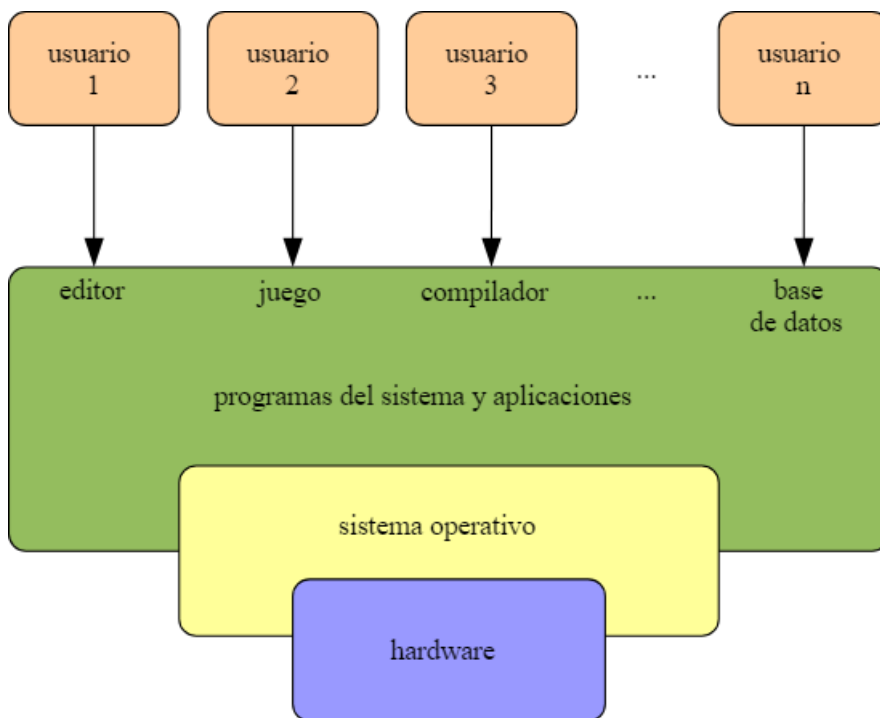
Sistemas Operativos

Definición de sistema operativo

Programa que actúa de intermediario entre los programas de aplicación de un ordenador y el hardware. También gestiona los recursos hardware y software entre los procesos y usuarios que compiten por ellos.

Funciones del sistema operativo

Un **sistema informático** puede ser dividido, en cuatro componentes: el hardware, los usuarios, los programas de aplicación y el sistema operativo



- **Programas de aplicación:** Programas que usan los usuarios para resolver sus problemas, como procesadores de texto, hojas de cálculo, etc.
- **Hardware:** Proporciona los recursos computacionales del sistema informático. Los programas de aplicación necesitan usar estos recursos computacionales para resolver los problemas informáticos de los usuarios.
- **Sistema operativo:** En un sistema informático, las aplicaciones necesitan realizar operaciones comunes. En lugar de que cada aplicación lo intente por su cuenta, el sistema operativo las centraliza y gestiona.

Por lo tanto, el sistema operativo controla, coordina el acceso y asigna los recursos computacionales del hardware a los distintos programas de aplicación.

Perspectiva del sistema informático

Un sistema informático tiene múltiples recursos, tanto hardware como software. Estos recursos los necesitan los programas de aplicación para resolver los problemas de los usuarios.

En resumen, desde la perspectiva del sistema informático, las funciones del **sistema operativo** son:

- Gestionar los recursos computacionales del sistema operativo.
- Controlar la ejecución de los programas de usuario y el acceso a los dispositivos de E/S.

Un sistema operativo **no** hace un trabajo directamente útil para los usuarios, pero proporciona un entorno adecuado para que los programas de aplicación lo hagan.

Perspectiva del usuario

La definición de las funciones del sistema operativo varía según el tipo de usuario y el sistema, pero el sistema operativo siempre tiene la responsabilidad de gestionar los recursos computacionales y controlar los programas. Estas funciones se mantienen constantes, aunque los objetivos varíen según el tipo de sistema.

1. *Sistema de escritorio:*

- Diseñado para maximizar la productividad de un solo usuario.
- Enfocado en la **facilidad de uso**, con algo de atención en el rendimiento y sin preocupación por el uso eficiente de los recursos.

2. *Sistemas con múltiples usuarios (grandes ordenadores centrales):*

- Los recursos son compartidos entre varios usuarios.
- El objetivo es maximizar el **aprovechamiento eficiente de los recursos**, garantizando que la CPU, memoria y E/S se utilicen equitativamente.

3. *Sistemas empujados:*

- Electrodomésticos u otros dispositivos con poca o ninguna interacción directa con usuarios.
- Aquí el sistema operativo está diseñado para el control automatizado de las tareas con un enfoque específico en la **eficiencia del sistema** sin priorizar la interacción humana.

En resumen, aunque los objetivos y las interfaces del sistema operativo difieren según el tipo de sistema, el **control de recursos y programas** sigue siendo su función clave.

Tipos de sistemas operativos

Mainframe

Los **ordenadores centrales** o **mainframes** fueron los primeros computadores utilizados en muchas aplicaciones comerciales y científicas. Se caracterizan por su gran capacidad de memoria, gran capacidad de almacenamiento secundario, gran cantidad de dispositivos E/S y rapidez de estos y alta fiabilidad.

Los **mainframes** pueden funcionar durante años sin problemas ni interrupciones y las reparaciones se realizan sin detener su funcionamiento.

Sistemas de procesamiento por lotes

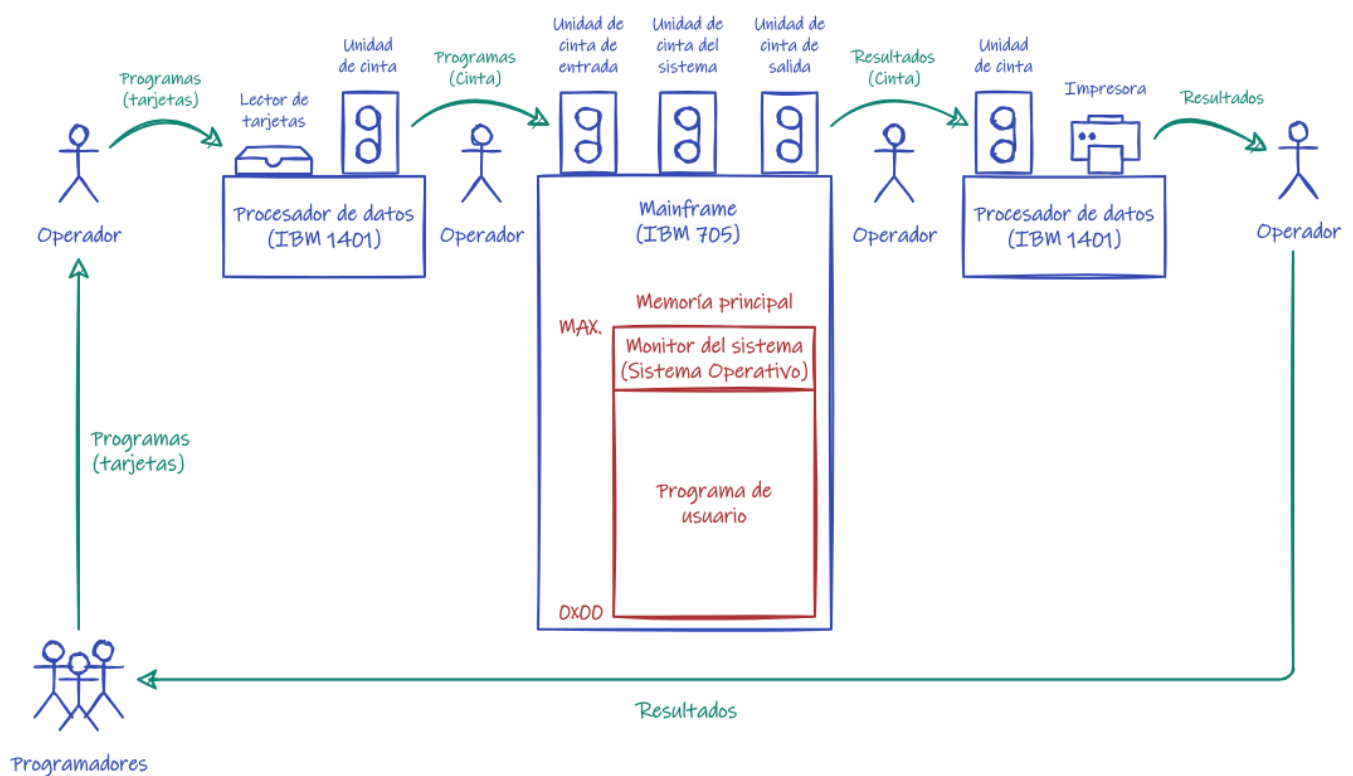
Los primeros **mainframes** eran enormes máquinas operadas desde una consola y conectados a lectores de tarjetas perforadas.

El trabajo era preparado por cada programador y entregado al operador del sistema, que era quién tenía acceso al sistema y la responsabilidad de ejecutar los programas y devolver los resultados.

No había sistema operativo y el operador debía cargar y ejecutar cada programa de uno en uno.



Estos sistemas se convirtieron en **sistemas de procesamiento por lotes** o **sistemas en batch** cuando se comenzó a utilizar un pequeño programa, cuya función era cargar y ejecutar sin interrupción un conjunto de programas.



El **monitor del sistema** es un predecesor de los sistemas operativos y tenía las siguientes características:

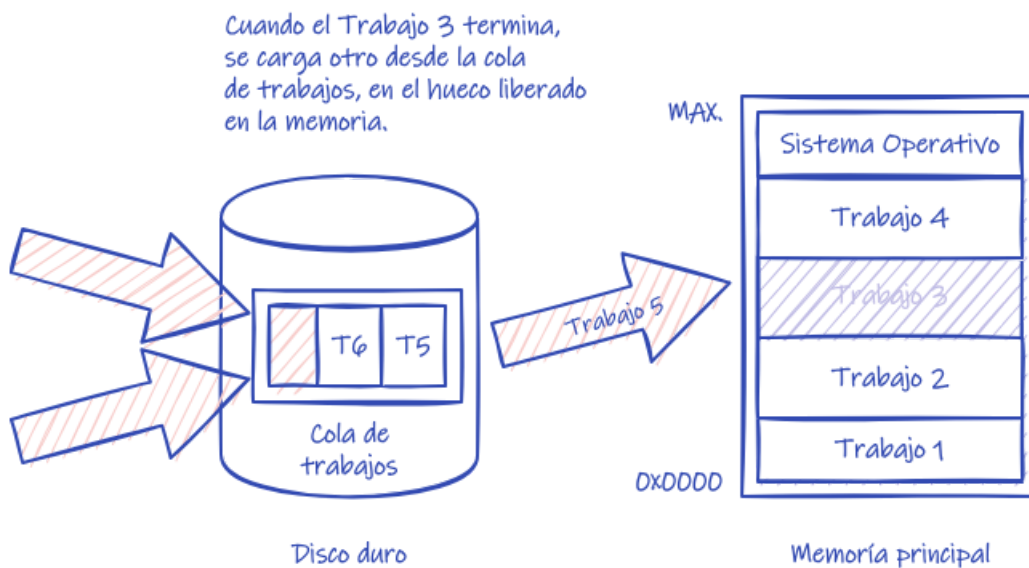
- Permanecía cargado durante todo el tiempo en la memoria del sistema.
- Su única tarea era cargar y transferir automáticamente la ejecución de un programa al siguiente cuando el anterior terminaba.
- El mayor inconveniente de este tipo de sistemas era que la CPU permanecía mucho tiempo desocupada porque era varios órdenes de magnitud más rápida que los dispositivos de E/S.

Sistemas multiprogramados

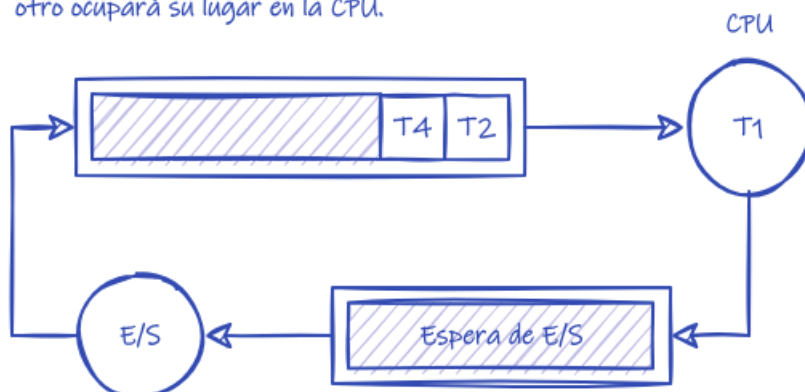
La solución al inconveniente de los sistemas de procesamiento por lotes con la E/S fue que los programas no accedieran directamente al dispositivo E/S, sino que, en su lugar, solicitaran la operación al **monitor del sistema** para que este la solicitara al hardware. Así el sistema operativo tiene la oportunidad de sustituir el programa en la CPU por otro, mientras la operación de E/S se completa.

Además, con la aparición de la tecnología de los discos magnéticos, los trabajos de los programadores comenzaron a ser almacenados en discos, desde donde eran escogidos por el sistema operativo para su ejecución.

A estos sistemas se los llamó **multiprogramados**, porque permitían tener varios programas en memoria al mismo tiempo e intercalar su ejecución en la CPU. A la cantidad de programas cargados en memoria en un instante dado se la denominaba **grado de multiprogramación**.



Mientras Trabajo 1 se ejecuta en la CPU, el resto de trabajos en la memoria esperan su oportunidad. Si Trabajo 1 termina o solicita una operación de E/S, otro ocupará su lugar en la CPU.



En los **sistemas multiprogramados** la ejecución de los trabajos funcionaba de la siguiente manera:

1. En el disco magnético se almacenaba una cola donde se iban colocando todos los trabajos que tenían que ser ejecutados.
2. El sistema operativo cargaba varios trabajos en memoria del conjunto de trabajos en la cola en el disco magnético.
3. El sistema operativo cede la CPU a uno de los trabajos en memoria.

4. Cuando el trabajo en la CPU requería usar la E/S se lo pedía al sistema operativo. En lugar de mantener a la CPU ocupada inútilmente, el sistema operativo programaba la operación E/S, pero escogía otro trabajo de entre los que estaban en memoria y lo ejecutaba en la CPU. Cuando la operación de E/S del anterior trabajo terminaba, el programa que ocupaba la CPU no era interrumpido, sino que debía esperar a una nueva oportunidad de ser escogido para ejecutarse en la CPU.
5. Cuando un programa en la CPU terminaba, sus recursos se liberaban, dejando memoria libre. Por lo tanto, el sistema operativo escogía un nuevo trabajo de la cola de trabajos en el disco magnético y lo cargaba en la memoria.

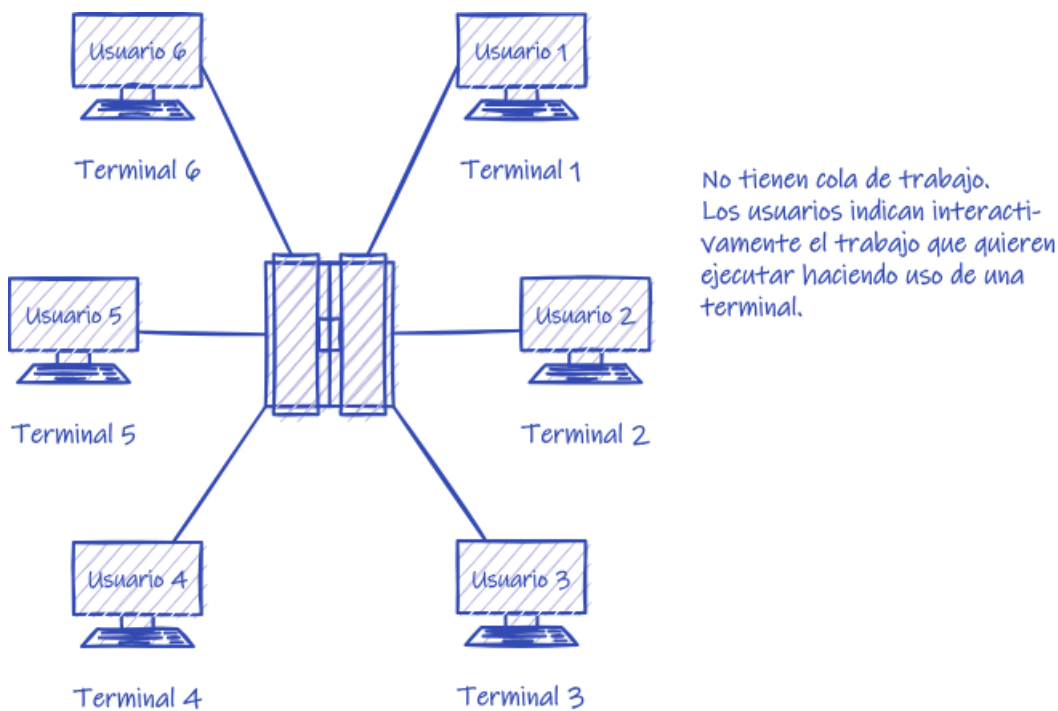
Todo este proceso se repetía mientras hubiera trabajos que ejecutar en la cola de trabajos en el disco.

Para operar de la forma descrita es necesario que el sistema operativo realice tres tareas esenciales:

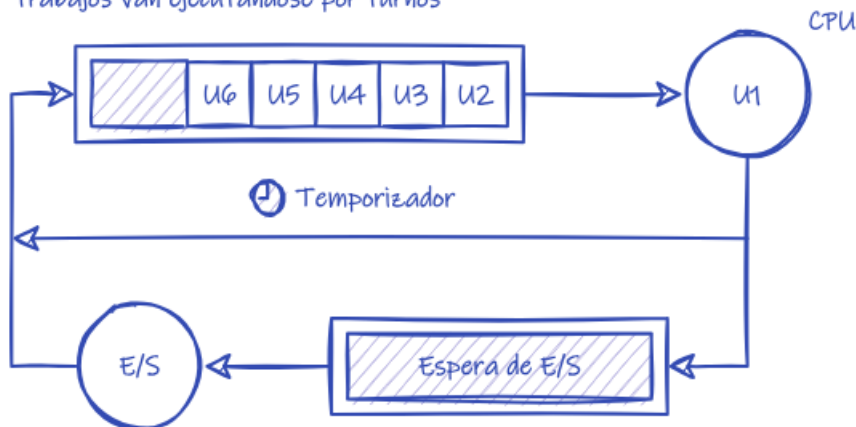
- La **planificación de trabajos**, cuya responsabilidad es seleccionar el siguiente trabajo que será cargado en la memoria principal para mantenerla llena.
- La **planificación de la CPU**, cuya responsabilidad es elegir el siguiente trabajo que será ejecutado en la CPU, de entre los disponibles en la memoria principal.
- La **gestión de la memoria**, cuya responsabilidad es repartir la memoria principal entre los trabajos alojados en la misma.

Sistemas de tiempo compartido

Los **sistemas de tiempo compartido** se desarrollaron tras observar que al dar acceso a un grupo de usuarios se podía conseguir un uso más eficiente del sistema, en comparación a cuando solo podía ser utilizado por un usuario a la vez. Esto es debido a que, generalmente, un usuario introduce información de forma continua para luego detenerse durante largos periodos de tiempo, mientras que en un grupo de usuarios, las pausas de uno de ellos se pueden llenar con la actividad de los otros.



Mientras el trabajo del Usuario 1 se ejecuta en la CPU, el resto de trabajos en la memoria esperan su oportunidad. Si el trabajo del usuario 1 termina o solicita una operación de E/S, otro ocupará su lugar en la CPU. Si acapara demasiado tiempo la CPU, se le expulsará para que otro pase a ejecutarse, de forma que los trabajos van ejecutándose por turnos



Los **sistemas de tiempo compartido** se caracterizaban por:

- Tener **terminales**, hardware especializado en hacer de interfaz directa entre los usuarios y el sistema. A través de estas terminales los usuarios podían enviar comandos al sistema e interactuar con sus trabajos. Podía haber múltiples usuarios al mismo tiempo, pero cada uno solo podía tener un trabajo en ejecución a la vez.
- Usar la **multiprogramación** para tener varios trabajos en la memoria principal al mismo tiempo e intercambiar el trabajo en la CPU cuando este solicitaba una operación de E/S, como ya se venía haciendo en los **sistemas multiprogramados** para hacer un uso más eficiente de la CPU.
- Repartir el tiempo de CPU entre usuarios. El sistema operativo asignaba un tiempo de CPU a cada usuario, denominado **ventana de tiempo** o **cuanto** de CPU. Cuando este tiempo se agotaba, el sistema intercambiaba el trabajo en la CPU por el de otro usuario en el sistema. La ventana de tiempo era extremadamente pequeña, dando a cada usuario la impresión de que su trabajo nunca se detenía, como si dispusiera de la CPU en exclusiva.

Los sistemas que, como los de tiempo compartido, pueden ser utilizados por varios usuarios simultáneamente se denominan sistemas **multiusuario**.

Los sistemas de tiempo compartido significaron un salto importante en complejidad por diversas razones:

- Como varios trabajos están en la memoria principal al mismo tiempo, el sistema operativo requiere mecanismos de **gestión de memoria** y **protección**.
- Para tener un tiempo de respuesta razonable, los trabajos deben estar cargados en la memoria principal. Para que quepan más trabajos de los usuarios en la memoria, el sistema operativo debe utilizar técnicas de **memoria virtual** para ejecutar trabajos que no están completamente cargados en la memoria principal.
- Como la CPU debe ser compartida entre los trabajos, el sistema operativo necesita mecanismos de **planificación de la CPU**.
- Como varios trabajos pueden tener la necesidad de cooperar y que su ejecución siga cierto orden, el sistema operativo debe proporcionar mecanismos de **sincronización** y **comunicación**.
- Como el sistema debe disponer de un **sistema de archivos** para repartir el espacio en disco y facilitar a los usuarios el acceso y gestión de sus datos, el sistema operativo necesita un componente de **gestión de discos**.

Sistemas de escritorio

Los primeros **sistemas operativos de escritorio** eran muy básicos. Carecían de otros mecanismos de protección y no eran ni multiusuario ni multitarea.

Los **sistemas operativos de escritorio** se han beneficiado del desarrollo de los sistemas operativos para *mainframes*. Los sistemas de escritorio actuales son **multiusuario** y **multitarea**; incluyen sistemas de archivos con permisos, autenticación y mecanismos de protección de la memoria, y han incorporado muchas otras características de los sistemas operativos para *mainframe*.

Aunque con el tiempo los **sistemas operativos de escritorio** han ido incorporando características que originalmente se desarrollaron en los *mainframes*, es importante recordar que ambos tipos de sistemas aún se diseñan con objetivos distintos. En los *mainframes*, el objetivo es maximizar la fiabilidad y la utilización eficiente de los recursos, mientras que en los sistemas de escritorio se busca priorizar la facilidad de uso y reducir el tiempo de respuesta para el usuario, prestando solo algo de atención al rendimiento.

Los **sistemas operativos de escritorio** modernos ya no son "solo de escritorio" ni se utilizan únicamente en ordenadores personales. Hoy en día, un alto porcentaje de ellos se emplea en servidores, superordenadores e incluso en dispositivos móviles. Por eso, en la actualidad el término "sistema operativo de propósito general" es mucho más adecuado.

Sistemas de mano

Con el nombre genérico de **sistemas de mano** hacemos referencia a las *tables*, *smartphones*, lectores de libros electrónicos y otros sistemas móviles y portátiles. Los desarrolladores de aplicaciones y sistemas de mano se enfrentan a problemas relacionados con el limitado tamaño. Debido a esas limitaciones, muchos sistemas de mano tienen poca cantidad de memoria, procesadores lentos y pantallas más pequeñas.

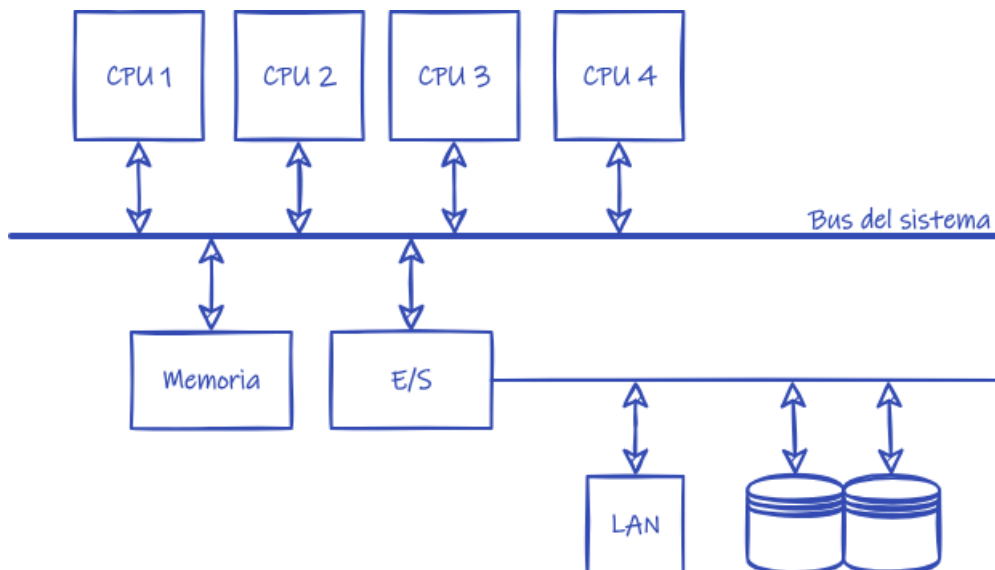
El diseño del sistema operativo suele primar la facilidad de uso y buscar un buen equilibrio entre rendimiento y tiempo de vida de la batería.

Sistemas multiprocesador

Un **sistema multiprocesador** es aquel ordenador con procesadores interconectados que comparten el bus del sistema, el reloj y, en ocasiones la memoria, y los periféricos.

Las principales ventajas de estos sistemas son:

- **Aumentan la cantidad de trabajo realizado:** A mayor número de procesadores, mayor cantidad de trabajo puede realizar el sistema. Sin embargo, un sistema con N CPUs **no** es un sistema N veces más rápido. Cuando varios procesadores cooperan para realizar una tarea, existe cierta pérdida de rendimiento debida a los mecanismos de sincronización requeridos para controlar el acceso a los recursos compartidos por los procesadores.
- **Economía de escala:** Un sistema multiprocesador puede costar menos que múltiples sistemas monoprocesadores conectados para hacer un trabajo equivalente, porque comparten periféricos, almacenamiento, alimentación, etc.
- **Alta disponibilidad:** Con el hardware adecuado, el sistema puede ser tolerante al fallo de uno de los procesadores. En caso de fallo, el sistema no se detendría, pero si trabajaría más despacio.



Existen dos tipos de sistemas multiprocesador:

- En los **sistemas de multiprocesamiento simétrico** o **SMP**, todos los procesadores son iguales. Todos comparten los mismos recursos, pueden acceder a los mismos dispositivos y cada uno ejecuta una copia del núcleo del sistema operativo. El sistema operativo debe haber sido diseñado para saber repartir el trabajo entre los procesadores y compartir adecuadamente entre tareas y procesadores el resto de recursos del sistema. Casi todos los sistemas multiprocesador modernos son de este tipo.
- En los **sistemas de multiprocesamiento asimétrico** o **AMP** hay un procesador principal y varios secundarios a quienes el principal planifica y entrega las tareas que deben ejecutar. En ocasiones, los procesadores secundarios se distinguen del principal por haber sido diseñados para realizar algún tipo concreto de tareas de forma muy eficiente o por estar conectadas a hardware especial. Ejemplos de esto son las **GPU**, que no son sino procesadores diseñados específicamente para el procesamiento de gráficos.

Sistemas distribuidos

Los sistemas distribuidos pueden ser clasificados en **sistemas cliente-servidos** y **sistemas de redes entre iguales**.

Sistemas cliente-servidor

En los **sistemas cliente-servidor** existen ordenadores que actúan como **servidores** encargados de satisfacer las peticiones generadas por otros ordenadores que actúan como **clientes**.

Este tipo de sistema han sustituido a los terminales conectados a *mainframes*, debido a que los sistemas de escritorio son cada vez más potentes y baratos.

- Los terminales han sido sustituidos por sistemas de escritorio que son capaces de realizar muchas de las funcionalidades que anteriormente eran manejadas directamente por los *mainframes*.
- Al mismo tiempo, estos *mainframes* se han reemplazado por servidores, no muy diferentes a los sistemas de escritorios, pero por preparados para atender las peticiones de sus clientes.

Ejemplos de este tipo de sistemas son los servidores de base de datos, que responden a las consultas SQL de los clientes.

Sistemas de redes entre iguales

En los **sistemas de redes entre iguales** o **P2P**, clientes y servidores no se distinguen los unos de los otros. Todos los nodos del sistema son iguales y cada uno puede actuar como cliente o servidor, dependiendo de cuándo piden o proporcionan un servicio.

La ventaja fundamental es que la carga se distribuye entre todos los nodos de la red.

Sistemas operativos para sistemas distribuidos

Desde el punto de vista de los sistemas operativos para sistemas distribuidos es posible hacer la siguiente distinción:

- Los **sistemas operativos de red** ofrecen a las aplicaciones que corren sobre ellos servicios de acceso a redes de ordenadores. Implementan algún mecanismo que permita a diferentes procesos en diferentes ordenadores enviar y recibir mensajes. Además, suelen incorporar la opción de proporcionar algunos servicios de red.

Los ordenadores con sistemas operativos de red son autónomos. Simplemente es que gracias al sistema operativo de red, conocen la existencia de la red y saben usarla para comunicarse con otros ordenadores de la misma.

Este tipo de sistemas operativos son los más utilizados en los tipos de sistemas distribuidos comentados anteriormente. En la actualidad, la inmensa mayoría de sistemas de escritorio y dispositivos de mano utilizan sistemas operativos de red.

- Los **sistemas operativos distribuidos** crean en el usuario la ilusión de que está en un único ordenador, dando al usuario acceso transparente a los recursos en todos los equipos de la misma.

Con este tipo de sistemas operativos, el usuario no sabe en qué ordenador se ejecutan sus procesos, donde se almacenan sus archivos, ni qué equipo tiene conectado a los distintos periféricos a los que tiene acceso.

Sistemas en clúster

Como los sistemas distribuidos, los **sistemas en clúster** interconectan ordenadores individuales.

Los **sistemas en clúster** se utilizan para:

- **Obtener servicios con alta disponibilidad:** Para ello, un nodo del clúster puede estar ejecutando un servicio mientras otro nodo lo monitoriza. En caso de fallo en el nodo que da el servicio, el que lo monitoriza lo sustituye.
- **Provisión de varios servicios:** Si se requieren múltiples servicios, se pueden distribuir entre dos o más nodos, y estos se monitorizan entre sí para asegurar el funcionamiento continuo.
- **Computación de alto rendimiento (HPC):** En este caso, todos los nodos trabajan juntos para ofrecer un único servicio. Un nodo especial, llamado balanceador de carga, se encarga de distribuir el trabajo entre los nodos.

Este tipo de clúster es ideal para realizar cálculos intensivos, como simulaciones o para romper sistemas de cifrado.

También se emplea mucho en servidores de Internet o bases de datos, donde deben atender a una gran cantidad de clientes simultáneamente. En estos casos, el balanceador de carga distribuye las conexiones de los usuarios entre los servidores del clúster.

Sistemas de tiempo real

Los **sistemas de tiempo real** se emplean cuando se requiere que ciertas tareas se ejecuten o que ciertos flujos de datos se procesen dentro de tiempo estrictos.

Estos sistemas se usan principalmente en dispositivos de control, donde es necesario recoger datos de uno o varios sensores dentro de límites de tiempo estrictos, analizarlos y luego realizar alguna acción mediante mecanismos de control.

Los **sistemas de tiempo real** están estrechamente vinculados con los **sistemas empujados**, los cuales:

- Están diseñados para realizar tareas muy específicas, a diferencia de los sistemas de propósito general.
- Sus sistemas operativos son muy limitados en características y no necesitan necesariamente una interfaz de usuario.

Tipos de sistemas de tiempo real

1. **Sistemas de tiempo real estricto (hard real-time):** Garantizan que las tareas se ejecutarán dentro de los límites de tiempo especificados. Para asegurar esto, todas las situaciones imprevistas que puedan causar retrasos en el sistema operativo deben estar perfectamente acotadas en términos de tiempo. Debido a ello, estos sistemas suelen prescindir de memoria virtual u otras abstracciones que generen impredecibilidad. Los sistemas de tiempo real estricto no son compatibles con los sistemas de tiempo compartido.
2. **Sistemas de tiempo real flexible (soft real-time):** Son útiles cuando ciertas tareas tienen más importancia que otras y deben ejecutarse con mayor prioridad. Sin embargo, no son adecuados para tareas con límites de tiempo y precisos, ya que no garantizan el cumplimiento de dichas restricciones. Son compatibles con memoria virtual y otras características propias de los sistemas de escritorio.

Por esta razón, la mayoría de los sistemas de escritorio actuales soportan tareas de tiempo real flexible.