

## 2. Clasificación y tipos



### IMPORTANTE

Aunque los términos **programa** y **proceso** están relacionados, no son equivalentes y conviene entender bien sus diferencias. Un **programa es un conjunto de instrucciones** que debemos seguir para realizar una tarea. Los programas se ubican en el disco duro. Por su parte, **un proceso es un programa en ejecución** que reside en la memoria principal, es decir, que tendrá un estado, tareas parcialmente realizadas, tareas terminadas o incluso tareas sin empezar. Si utilizáramos, con fines ilustrativos, un símil culinario, el programa sería la receta en la que se detallan los pasos por seguir y los ingredientes necesarios para elaborar un plato, mientras que el proceso sería el punto exacto en el que nos encontramos cuando tenemos una receta a medio hacer. Por poner un ejemplo, esto es lo que ocurre cuando tenemos el sofrito listo, pero aún debemos hervir la pasta.

Vamos a clasificar los sistemas operativos siguiendo estos criterios:

### 2.1. Según su utilización de los recursos

**Monoprogramados:** solo admiten un programa cargado en el sistema. El programa se carga en la memoria y se queda allí hasta que termina su ejecución. Durante este tiempo, no puede realizar ninguna otra tarea.

**Multiprogramados:** son capaces de realizar más de una tarea. Disponen de diferentes programas cargados en la memoria. Es importante diferenciar esta noción del concepto de multitarea, en el que el sistema operativo es capaz, además, de realizar estas tareas simulando que las ejecuta simultáneamente (para tener una multitarea real necesitamos disponer de tantos procesadores como tareas queramos ejecutar a la vez).

- **Multitarea apropiativa:** es el sistema operativo el que controla, en cada momento, qué proceso hace uso del procesador.
- **Multitarea cooperativa:** es el propio proceso el que decide si debe seguir usando el procesador o si se lo cede a otro proceso.

### 2.2. Según su interactividad

Cuando hablamos de interactividad, nos referimos a la forma en la que se produce la comunicación entre el sistema operativo y el usuario.

- **Sistemas de procesamiento por lotes (*batch*):** son aquellos en los que introducimos un lote de acciones o de pasos que el sistema operativo debe seguir. El sistema operativo los procesa en el orden en que los ha recibido y genera la salida correspondiente. Los primeros sistemas operativos que procesaban los lotes de tarjetas perforadas funcionaban con este tipo de sistemas operativos.

- **Sistemas de tiempo compartido** (*time sharing*): este tipo de sistemas operativos aceptan la ejecución de diferentes tareas al mismo tiempo (de manera simulada, puesto que, en la mayoría de los casos, el número de procesos en ejecución es superior al número de procesadores). Esto implica que el procesador estará asignado a un determinado proceso durante un tiempo limitado, llamado generalmente *quantum*. La mayor parte de los sistemas operativos actuales son sistemas de tiempo compartido, incluso aunque nuestro sistema tenga más de un procesador físico, pues siempre se está llevando a cabo un número de procesos superior al número total de procesadores físicos disponibles.
- **Sistemas en tiempo real** (*real time*): se trata de sistemas operativos destinados a realizar tareas críticas que dependen del tiempo de respuesta. En ellos, el tiempo máximo de respuesta está limitado y acotado; es decir, antes de ejecutar una tarea determinada, el usuario sabe cuál es el tiempo máximo que va a necesitar el sistema para devolver un resultado. Por ejemplo, los ordenadores de control de vuelo de una aeronave utilizan este tipo de sistemas operativos. Y es que no tendría sentido que, al girar la palanca del vehículo, este no respondiera debido a que el sistema operativo se encuentra realizando cualquier otra tarea menos importante y vital. Cuando el tiempo de respuesta no es crítico ni es conocido a priori, los llamamos **sistemas en tiempo diferido**. Este tipo de sistemas son los más abundantes.

### 2.3. Según el número de usuarios

Según el número de usuarios que pueden usar el sistema de manera simultánea, encontramos:

- **Sistemas monousuario**, en los que un solo usuario puede hacer uso del sistema en un determinado momento. Puede haber otros usuarios en el sistema, pero solo uno será capaz de usarlo en un momento dado.
- Por el contrario, los **sistemas multiusuario** permiten, a través de diferentes técnicas de multiprogramación, que diferentes usuarios accedan al sistema (de manera directa o a través de conexiones remotas) con el fin de ejecutar tareas sobre el sistema.

### 2.4. Según el tipo de aplicación

En función del tipo de aplicaciones informáticas que será capaz de ejecutar el sistema operativo, podemos encontrar:

- **Sistemas de propósito general**: pueden ejecutar cualquier tipo de aplicación informática. Los sistemas operativos más usuales, como Windows, GNU/Linux, Mac OS, Android, iOS, etcétera, serían un buen ejemplo de esta clase.
- **Sistemas de propósito específico**, diseñados para ejecutar específicamente un tipo de aplicaciones informáticas o de tareas concretas. Estos son, por ejemplo, los sistemas operativos que incluyen las máquinas industriales o los electrodomésticos que tenemos en nuestros hogares.

## 2.5. Según el número de procesadores

Al principio de la informática, los sistemas informáticos contaban con un solo procesador, pero hoy en día es habitual que muchos de ellos cuenten con varios de estos procesadores. En este sentido, tenemos:

- **Sistemas monoprocesadores.** En ellos, el sistema cuenta con un único procesador y, por tanto, este es el responsable de ejecutar todos los procesos que se requieran.
- **Sistemas multiprocesadores o multiproceso.** En estos, podemos encontrar más de un procesador, lo que permite ejecutar más de una tarea a la vez. Del mismo modo, si una única tarea es divisible, permiten realizar un mismo proceso en menos tiempo.

Los nuevos sistemas operativos se desarrollan con metodologías **orientadas a objetos**. Todo en el sistema operativo es una colección de objetos. El núcleo se encarga de mantener las definiciones de los objetos y el control relativo al acceso a los mismos.

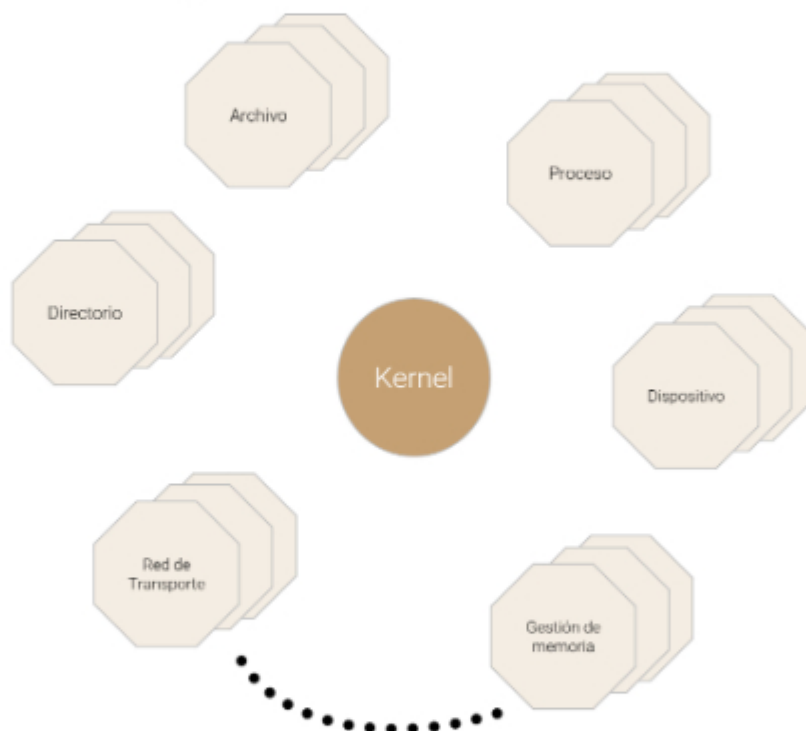


Figura 2.1 Estructura orientada a objetos.

Ahora que los sistemas son cada vez más rápidos y que contamos con infraestructuras de red también muy veloces, se acostumbra a combinar varios procesadores en una **estructura multiprocesador** para conseguir una solución económica y escalar acorde con las necesidades de cada momento. De este modo, teniendo en cuenta la relación que existe entre la memoria y los procesadores, tenemos:

- **Multiprocesadores fuertemente acoplados o sistema de memoria compartida.** En este caso, cada procesador tiene acceso a la totalidad de la memoria del sistema.

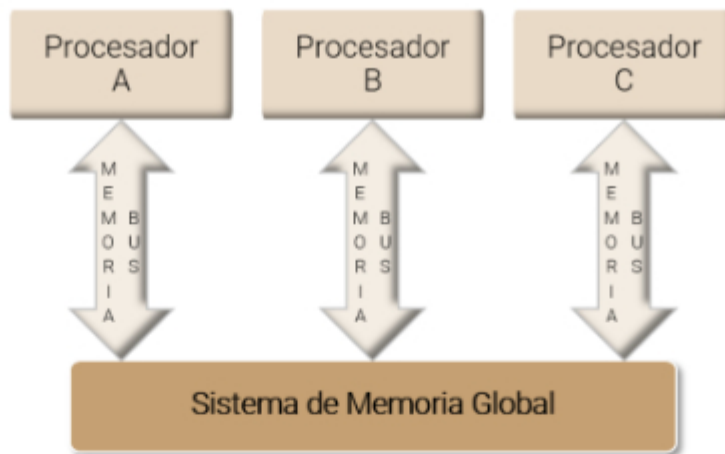


Figura 2.2 Multiprocesadores fuertemente acoplados.

- **Multiprocesadores débilmente acoplados o sistema de memoria distribuida.** En este sistema, cada procesador tiene acceso exclusivo a una memoria privada, y los procesos se comunican a través del paso de mensajes.

## 2.6. Según la distribución de las tareas del sistema

Con el avance de las tecnologías de comunicación, cada vez es más usual que los sistemas operativos cooperen entre sí para obtener mejores resultados. Así pues, podemos encontrar:

- **Sistemas distribuidos.** Existe un conjunto de máquinas —con diferentes capacidades y dotadas de diversos tipos de procesadores y arquitecturas— que se comunican entre sí mediante redes de comunicaciones. De esta forma, se reparten el trabajo llevando a cabo una subdivisión de las tareas, lo que permite mejorar el rendimiento final del sistema en su conjunto. A día de hoy, los grandes sistemas de información basan su estructura en este tipo de sistemas, ya que les permiten mejorar su rendimiento, gozar de independencia con respecto a la ubicación del usuario y presentar una mejor tolerancia frente a los fallos.
- **Sistemas centralizados:** en ellos, una sola máquina es la responsable de realizar todas las tareas que exijan al sistema operativo.

