

1. Introducción

Los ordenadores¹ están presentes en muy diversos ámbitos de nuestras vidas (Figura 1). Cuando se pide a un usuario que describa su ordenador lo hace según lo percibe, con pantallas llenas de ventanas, comandos, etc. más que describiendo el sistema físico con el que trabaja. Esta imagen la proporciona el sistema operativo. Sin embargo, pocos usuarios serían capaces de dar una definición correcta del mismo.

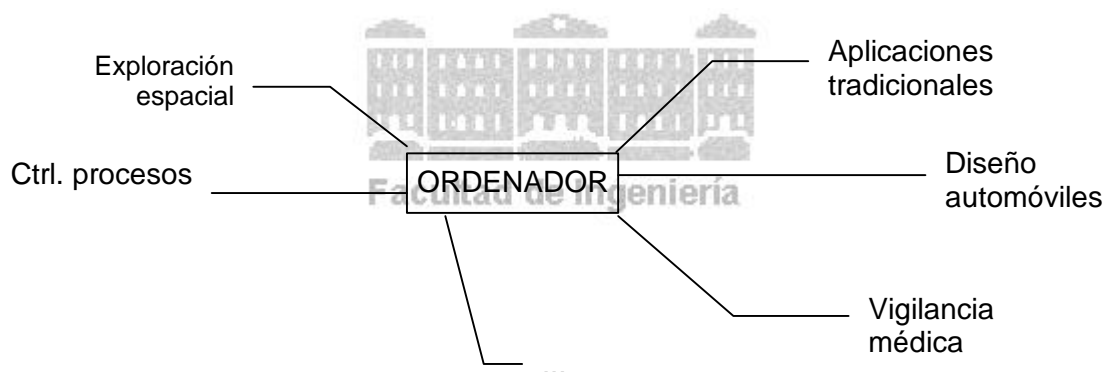


Figura 1: Ámbitos de nuestro mundo en los que se emplean ordenadores

1.1 Definición de Sistema Operativo

Para que un ordenador sea verdaderamente útil y ayude a aumentar el rendimiento, a cada tipo de usuario – si no el de cada usuario concreto – debe ofrecerse una visión personalizada del sistema; es decir, un mismo ordenador como máquina física, debe mostrarse a cada uno en la forma en que lo necesita (diferente para un programador de sistemas, un operador, etc.). Para ello, a partir de un hardware común, es necesario un interfaz entre el ordenador y los usuarios que sea capaz de presentar distintas visiones de la parte física del sistema de computación.

Este interfaz es el sistema operativo y hace de intermediario entre los programas de aplicación y usuarios, y el hardware. Es el único componente que se encuentra en contacto directo con este último y en consecuencia, el único con capacidad para gestionar los recursos físicos de la máquina, supervisando además las ejecuciones².

¹ El término “ordenador” y el término “computador” son sinónimos y hacen referencia a la máquina. El término “sistema de computación” engloba hardware, software – el sistema operativo, programas de aplicación, etc. – y recursos humanos.

² No obstante, el S.O. podría no estar presente. De hecho, los primeros ordenadores carecían de sistema operativo, y el hardware era manipulado directamente por los distintos usuarios.

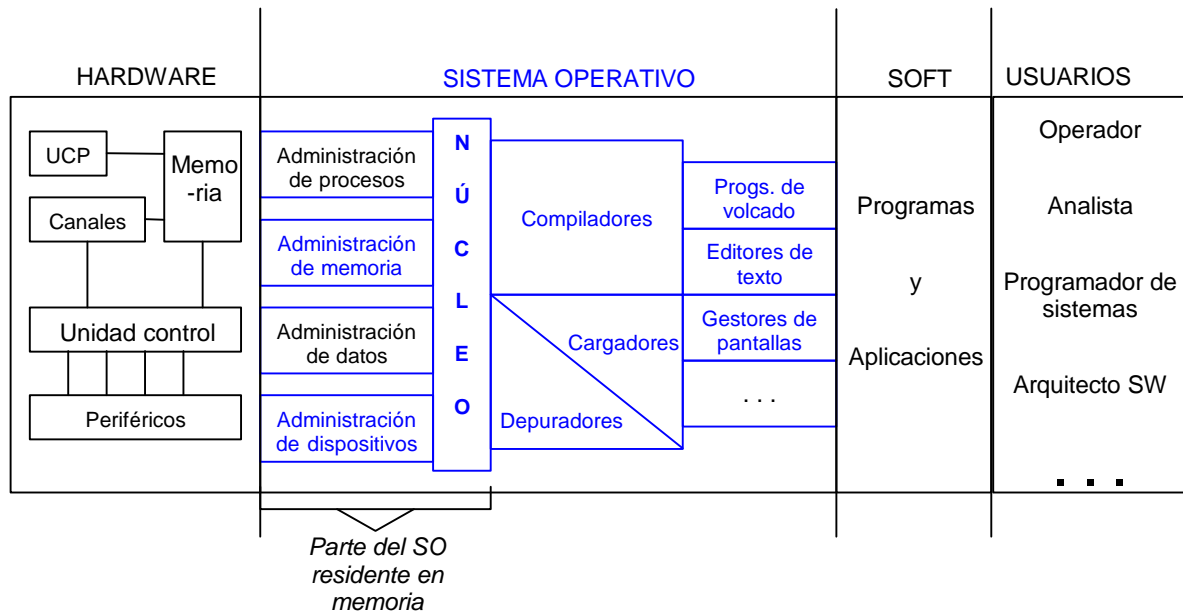


Figura 2: Componentes de un sistema de computación.

En los años 60, se definía “sistema operativo” como *software que gestiona³ el hardware*. No obstante, muchas de sus rutinas – cada vez en mayor cantidad – se implementan en microcódigo o *firmware*, lo cual aporta entre otras cosas mayor seguridad y velocidad al sistema operativo. Así, actualmente una definición más adecuada sería *conjunto de programas implementados⁴ tanto en software como en firmware que hacen asequible el hardware*. Entendiendo por “hacer asequible” que el hardware esté disponible de la forma más adecuada a cada usuario, y que aumente el rendimiento total, que engloba el tiempo de respuesta y el rendimiento específico o *throughput*.

El sistema operativo como gestor de recursos		El sistema operativo como interfaz
<ul style="list-style-type: none"> Gestiona los recursos: <ol style="list-style-type: none"> Supervisa ejecuciones. Evita uso inadecuado. Detecta y da respuesta a fallos y errores. 	<ul style="list-style-type: none"> Los asigna a distintos usuarios y/o aplicaciones: <ol style="list-style-type: none"> Se encarga de su conmutación. 	<ul style="list-style-type: none"> Proporciona un entorno dentro del cuál es posible la realización de trabajo útil

Tabla 1: Misiones del Sistema Operativo

³ Englobando el término “gestión” tareas de control, asignación y supervisión.

⁴ La implementación es la realización práctica, cómo se ha llevado a cabo.

1.2 Funciones del Sistema Operativo

El sistema operativo es el encargado de crear de un entorno que haga posible la ejecución de procesos, ofreciendo a los mismos y a los usuarios una serie de funciones o servicios. No obstante, estos servicios pueden variar sustancialmente de unos sistemas a otros.

Todo sistema operativo de propósito general se caracteriza por la existencia de varios usuarios trabajando simultáneamente sobre un amplio abanico de aplicaciones, proporcionarles un amplio espectro de facilidades (compiladores, depuradores, editores, procesadores de texto,...) y asegurar un flujo continuo de trabajos en forma de tareas – por lotes o interactivas.

Dichas prestaciones implican la existencia de una serie de funciones, como son:

- Capacidad multiacceso.
- Gestión de memoria.
- Secuenciación de trabajos.
- Control de procesos.
- Interpretación de comandos.
- Gestión de operaciones de E/S.
- Gestión de dispositivos.
- Gestión de interrupciones.
- Planificación.
- Control de errores y protección.
- Gestión de ficheros.
- Contabilidad del sistema.
- ...

Todas ellas se pueden agrupar en dos: funciones *desde el punto de vista del usuario*, es decir aquellas funciones encaminadas a transformar el ordenador en una máquina más tratable – virtual –, y *funciones internas del sistema operativo*, que dan soporte a la concurrencia.

1.2.1 FUNCIONES DESDE EL PUNTO DE VISTA DEL USUARIO.

Gestionar los recursos

Un gran número de funciones van encaminadas a la ejecución y control de programas, incluyendo su carga, ejecución y control de errores, asegurando en este último caso que cada tipo de error – electromecánico, de protección, de operación no válida, etc. – tenga la respuesta adecuada.

El Sistema Operativo se encarga de controlar y gestionar todos los recursos – tanto hardware como software – del sistema de computación: la memoria principal, la UCP, los dispositivos periféricos, la información almacenada, etc. de acuerdo con su disponibilidad y de acuerdo con ciertas características propias de los programas en ejecución o procesos.

En un sistema de computación debe ser posible la ejecución simultánea de varios programas, que comparten y compiten por los recursos disponibles. El sistema operativo es el que se encarga de gestionarlos de acuerdo con su disponibilidad y de acuerdo con ciertas características propias de los programas en ejecución.

Uno de los elementos más críticos que controla es la UCP; su utilización debe ser repartida entre los distintos programas en ejecución, puesto que sólo uno de ellos puede ser atendido en cada momento. Además, la ejecución simultánea de procesos⁵ suele requerir la existencia de mecanismos de comunicación o interacción entre ellos, mecanismos que deben estar disponibles incluso para los procesos del propio sistema operativo y, por tanto, tienen que estar implementados lo más próximos posible al hardware.

Otro recurso esencial es la Memoria. Los diferentes tipos de memoria del ordenador se organizan en una jerarquía de niveles – registros máquina, memoria caché, memoria principal, memoria secundaria y memoria terciaria – que el sistema operativo controla, supervisa y gestiona como un todo para aumentar el rendimiento, presentándola con la forma y tamaño más adecuados para cada proceso y usuario, y evitando que los distintos programas en ejecución, cada uno con distintas necesidades de memoria, se mezclen entre sí. También coordina la utilización de los periféricos, regulando por ejemplo los accesos a discos o el orden en que serán atendidas las tareas de impresión, y liberando consiguientemente al usuario de tener que programar los mecanismos *hardware* de E/S.

Proporcionar un interfaz de usuario

El sistema operativo hace factible la comunicación entre usuarios y/o programas de aplicación en ejecución y el computador. En este sentido, los sistemas operativos proporcionan un conjunto de órdenes o *comandos* que constituyen un *interfaz de usuario dirigido por comandos*. Las órdenes introducidas por el usuario son “traducidas” por el intérprete de comandos, perteneciente al sistema operativo, para

⁵ Aunque el término proceso es más amplio, por el momento se asociará a la ejecución de un programa.

posteriormente ejecutar la acción o acciones asociadas a esa orden. El problema radica en la dificultad a la hora de aprender, recordar y utilizar estos comandos (su sintaxis, su semántica y sus posibles parámetros).

Una alternativa para hacer frente a esta dificultad la constituyen los interfaces *de usuario orientados a menús*, que presentan las posibles opciones de forma que los usuarios puedan a través de entradas de menús y submenús para indicar al sistema operativo los servicios requeridos.

La tendencia actual es la formada por los interfaces *gráficas de usuario*⁶ (IGU), constituida por elementos gráficos como iconos que hacen posible el acceso a la mayoría de funciones o acciones del sistema operativo. Estas “metáforas visuales” empleadas para introducir las distintas órdenes y acceder a las funciones del sistema operativo, redundan en una gran facilidad de aprendizaje y manejo.

1.2.2 FUNCIONES INTERNAS DEL SISTEMA OPERATIVO

Las funciones denominadas internas del sistema operativo son las orientadas principalmente a dar soporte a la ejecución concurrente de procesos, y por tanto, a la compartición de recursos entre los procesos. Esta compartición es recomendable, cuando no necesaria, por varios motivos como son:

- Económico: compartir recursos físicos entre varios usuarios y/o procesos supone un ahorro considerable. Por ejemplo, un servidor de impresión instalado en una red puede hacer innecesaria la adquisición de una impresora para cada usuario.
- Ahorro de tiempo y trabajo, gracias a la posibilidad de reutilización de fragmentos de código o rutinas desarrolladas por otros.
- Eliminación de redundancia de recursos (no sólo físicos, sino también información).
- La naturaleza, de por sí compartida, de la información y por tanto de recursos, como ficheros, bases de datos, etc.

Gestión del hardware

Desde el punto de vista interno del sistema operativo, la gestión de recursos tiene como objetivo el aumento de su disponibilidad y utilización por parte de los procesos,

⁶ GUI: Graphic User Interface.

que llevan a cabo las funciones requeridas por los usuarios. Así, existen recursos, que precisan de programas especiales para su gestión: asignación, liberación, requisamiento cuando es necesario, etc. Entre estos recursos destacan la UCP y la memoria, de cuya gestión se encargan el planificador y el gestor de memoria respectivamente.

Otros, como los dispositivos de E/S, siguen políticas de asignación y liberación generales, siendo su gestión labor de programas más o menos generales o comunes que se encargan incluso de la gestión de varias unidades o varios recursos simultáneamente.

Protección

Posibilitar la ejecución concurrente de varios procesos y/o usuarios en un sistema no solamente conlleva problemas de compartición de recursos. Cuantos más sean los procesos y/o usuarios que comparten los mismos elementos, mayor será la posibilidad de potenciales interferencias entre ellos, y por tanto más complicado el sistema de protección.

El sistema operativo es el encargado de garantizar el control de la información en cuanto a usos y accesos, de manera que sólo los usuarios o procesos autorizados puedan acceder a la información, y únicamente en los modos o para las operaciones autorizadas.

Es también labor del sistema operativo el mantenimiento de la integridad de la información, especialmente comprometida cuando los distintos procesos acceden simultáneamente a las mismas variables. Además, es necesario garantizar que no se den interferencias no deseadas entre programas en ejecución que si bien se ejecutan concurrentemente pueden tener áreas de memoria con acceso exclusivo.

Auditoría interna.

El sistema operativo realiza labores de contabilidad del sistema o *system accounting* llevando un registro del uso del sistema por parte de los distintos usuarios y procesos. De esta manera, es posible acceder a informaciones relativas al porcentaje de utilización de la UCP por parte de cada usuario y/o proceso, al tiempo que ha tenido el control de la misma, al tiempo que ha estado esperando por ella y el tiempo que ha estado en espera de cierto recurso; al número de operaciones de E/S realizadas, etc.

Estas funciones se encaminan tanto a la facturación por utilización del sistema como a la optimización del sistema o *Tuning*.

1.2.3 ACCESO A LOS SERVICIOS OFRECIDOS POR EL S. O. ESTRUCTURA.

Un sistema operativo se compone de diferentes elementos que pueden agruparse formando una jerarquía de capas software. En la ifugra 3 se muestra un sistema opeativo estructurado por niveles o capas, distinguiendo tres niveles generales.

Nivel de núcleo o kernel.

El núcleo es el nivel más interno del sistema operativo y suele implementarse en software y/o en firmware, y es – junto con los drivers, que no siempre están incluidos en el mismo – el único elemento que interactúa directamente con el hardware, proporcionando al resto de niveles cierta independencia respecto de éste a la vez que libera a los usuarios de la obligación de conocer directamente las especificidades del hardware. Se trata del nivel que proporciona la mayoría de las funciones básicas del sistema operativo, incluyendo la ejecución de programas, la asignación y “reparto” de muchos recursos hardware a cada programa individual, etc.

La modificación del núcleo de un sistema operativo podría ser suficiente para que éste pudiera interactuar con distintos hardwares, es decir, el sistema operativo podría funcionar en diversas plataformas.

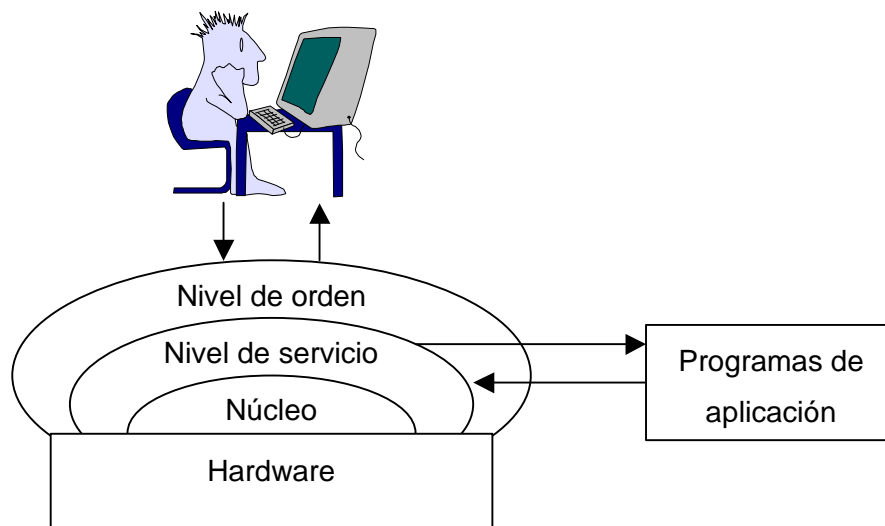


Figura 3: Sistema operativo genérico, de tres niveles

Nivel de servicio o de llamadas al sistema.

Acepta peticiones de servicios provenientes del nivel de orden o de los programas de aplicación, traduciéndolas a un conjunto de instrucciones que el núcleo entiende (en

lenguaje ensamblador). Cuando el núcleo recibe estas instrucciones, lleva a cabo las acciones correspondientes. Si como consecuencia se genera algún resultado, este nivel los devolvería al nivel de orden o al programa de aplicación que solicitó el servicio. Existen prácticamente tantas llamadas al sistema distintas como servicios pueda prestar el sistema operativo.

En este nivel se encuentran los programas que proporcionan, en respuesta a las llamadas, los servicios correspondientes. Éstos son:

1. **Control de procesos**, que incluye su ejecución, creación y terminación, su finalización, la obtención y establecimiento de sus atributos, su suspensión, la espera de un suceso, la asignación o liberación de memoria, etc.
2. **Acceso a dispositivos de almacenamiento o de E/S (drivers)**, que engloba la asignación y liberación de dispositivos, el movimiento de datos desde una aplicación a una impresora, un disco magnético o una unidad de cinta, etc.
3. **Manipulación de archivos (sistema de ficheros)** abarcando el mantenimiento de la estructura de directorios, la apertura y cierre de archivos, la lectura / escritura sobre los mismos, su creación y eliminación, etc.
4. **Coordinación y control de la memoria (gestor de memoria)** que incluye el almacenamiento de datos y el posterior acceso a los mismos, y la compartición de datos entre varios programas de aplicación.
5. **Mantenimiento de información del sistema**, como, por ejemplo, la obtención o establecimiento de la fecha y hora del sistema, atributos de un proceso, archivo o dispositivo, la obtención o modificación de parámetros del sistema, etc.
6. **Otros servicios**, como la gestión de elementos del IGU (ventanas, iconos, etc.), el acceso a redes de comunicaciones o servicios básicos de bases de datos, incluyendo la creación y eliminación de conexiones, el envío y recepción de mensajes, la transferencia de información de estado, etc.

Nivel de orden o de comandos del sistema.

Constituye el nivel más externo. Proporciona el interfaz de usuario y es la única parte del sistema operativo con la que los usuarios pueden interactuar directamente. Es un conjunto de programas del sistema que engloba órdenes o comandos y varía de unos sistemas operativos a otros.

El intérprete de comandos, ubicado en este nivel, recibe las ordenes o comandos introducidos por el usuario, traduciéndolas o iniciando la ejecución de una o más llamadas al sistema por parte del nivel de servicio.

La idea que el usuario tiene de un sistema operativo es la que los programas del sistema le proporcionan. Sin embargo, es difícil diseñar un buen interfaz para un sistema operativo. De hecho existen sistemas cuya interfaz no es excesivamente “amigable” y que ocultan una funcionalidad muy robusta como es el caso de los sistemas tipo *Unix*. Por otro lado, un mismo sistema operativo puede presentar varios interfaces, como es el caso de *X-Windows*.

Categoría	Programas al sistema
Manejo y manipulación de ficheros.	Copia, eliminación, creación,... de ficheros.
Modificación de ficheros.	Editor de texto.
Modificación de información de estado.	Manipulación de fecha, hora, colas del sistema,...
Carga / ejecución de programas.	Cargador, montador, depurador, etc.
Soporte de lenguajes.	Compiladores, Intérpretes, etc.
Comunicaciones	Correo electrónico.
Programas de aplicación.	Procesadores de texto, editores de imágenes, etc.
Interpretación de comandos.	Intérprete de comandos.

Tabla 2: Categorización de los programas del sistema

En la tabla 2 se muestra una clasificación en categorías de los programas del sistema más habituales de los sistemas operativos de propósito general. De ellos, uno de los principales es el intérprete de comandos, puesto que es el programa del sistema a través del cual se accede a todos los demás. Su función de hecho es capturar comandos e iniciar su ejecución, cuando no ejecutarlos.

A la hora de implementar el intérprete de comandos, hay básicamente dos enfoques. El primero consiste en incluir dentro del propio intérprete de comandos el código correspondiente a todos y cada uno de los comandos. Con él se logra una ejecución muy rápida de los comandos pero presenta dos desventajas: una gran falta de flexibilidad y un gran tamaño (mayor cuantos más comandos implemente).

El segundo enfoque, que es el que se utiliza en *Unix*, consiste en la implementación de cada comando como un programa del sistema. En este caso, el intérprete de comandos no ejecuta el comando sino que lo utiliza para buscar un programa del sistema correspondiente – aquel con igual nombre que el comando introducido para que sea cargado en memoria y ejecutado. Evidentemente, es más lento y es necesario

implementar algún mecanismo de paso de parámetros entre los programas, pero presenta varias ventajas como son su reducido tamaño, su independencia respecto del número de comandos que reconoce y su gran flexibilidad, puesto que la adición de un nuevo comando al sistema únicamente requiere añadir un nuevo programa que tenga por nombre el de dicho comando.

1.3 Características de los sistemas operativos de propósito general.

Un sistema operativo de propósito general debe cumplir una serie de requisitos o características que lo hagan útil para su explotación, independientemente de las decisiones ulteriores del mercado. Entre éstas cabe destacar las siguientes:

- **Fiabilidad:** un sistema operativo debe estar libre de errores o “bugs” de manera que pueda responder adecuadamente a cualquier evento.
- **Determinismo e indeterminismo:** un sistema operativo debe ser determinista si bien es necesario que presente cierto grado de indeterminismo. El determinismo del mismo implica que a partir de unas mismas entradas, se deben generar siempre las mismas salidas, mientras que el indeterminismo conlleva responder a cualquier situación (producida generalmente al azar).
- **Flexibilidad y generalidad:** un sistema operativo debe adaptarse a las diferentes necesidades de los distintos usuarios, y además debe ser – en la medida de lo posible – portable.
- **Facilidad de uso:** una de las tendencias que ha marcado la evolución de los sistemas operativos es el logro de productos cada vez más fáciles de acceder, aprender y manejar.
- **Seguridad e integridad:** el sistema operativo es el encargado de proteger tanto datos como programas de cualquier intento de acceso o utilización no permitidos, evitando en lo posible la corrupción de datos y programas.
- **Visibilidad u opacidad:** el sistema operativo solamente debe dejar ver a cada usuario aquello que necesita, y en la manera que pueda obtener un mayor rendimiento.
- **Eficiencia y disponibilidad:** un sistema operativo debe funcionar correctamente y proporcionar el mayor número posible de servicios a los usuarios.

- **Mantenibilidad y extensibilidad:** el sistema operativo debe ser fácil de mantener y ampliar, para lo cual es recomendable que esté diseñado de manera modular.
- **Tamaño no excesivo:** un sistema operativo de tamaño considerable disminuiría sobremanera la productividad del sistema.

1.4 Tendencias actuales de los sistemas operativos.

El diseño de los sistemas operativos modernos, viene marcado por las siguientes tendencias: amigabilidad, máquina virtual, procesamiento distribuido y procesamiento paralelo.

Implementando muchas funciones de los sistemas operativos en microcódigo y con la paulatina aplicación a su diseño de los avances del área de la Ingeniería del Software, se ha logrado incrementar la fiabilidad de los sistemas operativos, además de su facilidad de uso, mantenimiento y extensión, y todo ello sin afectar de manera negativa a su rendimiento, sino logrando que aumente.

1.4.1 AMIGABILIDAD

Deriva del término inglés *user friendliness*, y hace referencia a la facilidad tanto de acceso a los servicios del sistema operativo como de su manejo. Se persigue que no sea necesario que los usuarios – o al menos algunos grupos de usuarios – tengan conocimientos y/o experiencia en computadores. De hecho, uno de los objetivos en el diseño de sistemas operativos ha sido mejorar el interfaz con el usuario y los medios (iconográficos, auditivos, etc.) utilizados en la interacción usuario-ordenador, para hacer un S.O. lo más sencillo posible.

Tras este objetivo subyace el deseo de expandir el uso de la informática a todas las áreas de conocimiento, de manera que los ordenadores se conviertan en una herramienta imprescindible. Si bien antes un ordenador se consideraba un recurso (se otorgaba gran trascendencia al hecho de disponer o no del mismo) actualmente, y dada su omnipresencia, se hace mayor hincapié en su disponibilidad, flexibilidad y fácil utilización.

1.4.2 MÁQUINA VIRTUAL

Se trata de un término íntimamente ligado con el anterior. La mejor manera de lograr un sistema amigable para un usuario – incluso distinto para cada usuario – es presentarle un sistema como el que desea ver – y por tanto, distinto del físico o real.

Dicho de otra manera, se presenta a cada usuario un sistema *virtual*, se le ocultan las complejidades y detalles físicos de la máquina real; el usuario ve solamente la máquina que le presenta el sistema operativo.

Indirectamente, se persigue que todos los usuarios – incluidos los programadores, diseñadores, desarrolladores, etc. – trabajen o contemplen únicamente sistemas virtuales, *simulados* sobre cualquier plataforma. De esta manera, cualquier programa desarrollado para el sistema virtual podría implementarse en cualquier plataforma en que éste fuera simulado, logrando así “alejar” o independizar el software del hardware.

Un ejemplo, eso sí, muy particular, de aplicación de esta tendencia la encontramos en la *Java Virtual Machine (JVM)* o Máquina Virtual Java, un elemento software que puede ser instalado sobre prácticamente todas las plataformas hardware del mercado, de manera que el sistema presentado por ella es siempre el mismo, independientemente de la plataforma subyacente. Así, el código Java puede ser ejecutado sobre cualquier plataforma: la portabilidad es absoluta.

1.4.3 PROCESAMIENTO DISTRIBUIDO

Modo de procesamiento en sistemas en los que cooperan varios ordenadores, independientemente del número de procesadores de que disponga cada uno.

Este tipo de sistemas viene propiciado por un declive en el costo de las comunicaciones unido a un aumento de la velocidad de transmisión.

En los sistemas distribuidos, cada uno de los programas en ejecución puede realizar operaciones y tomar decisiones locales (que afectan únicamente a su entorno) y/o participar de decisiones que pueden afectar a sistemas remotos (“externos” a él), como por ejemplo en el control de un proceso distribuido.

En los sistemas distribuidos, se pueden instalar sistemas operativos distribuidos o dispersos. Es decir, el código del sistema operativo, en lugar de estar al completo en todos y cada uno de los sistemas en red, se reparte entre los ordenadores que forman el sistema, instalando en todos ellos únicamente un núcleo mínimo.

1.4.4 PROCESAMIENTO PARALELO

Se da en aquellos sistemas en los que hay varios procesadores, de manera que todos ellos trabajan en paralelo. El incremento de la escala de integración (componentes potencialmente superiores en aputados cada vez menores) ha favorecido este tipo de sistemas.

La primera consecuencia clara de este avance fue la reducción del tamaño de los ordenadores que hizo posible finalmente la aparición de ordenadores personales e incluso portátiles – cosa impensable en torno a los años 60 (hoy por hoy las memorias y los procesadores son físicamente mucho menores de lo que eran entonces). Otra de las consecuencias fue la inclusión en un mismo sistema de varios procesadores que trabajaban paralelamente. Este tipo de sistemas es requerido para cubrir las necesidades de cálculo cada vez mayores de ciertas aplicaciones, como las de simulación de procesos, métodos numéricos, etc.

A diferencia de los sistemas distribuidos, este tipo de procesamiento no implica la existencia de una red de ordenadores conectados, sino la existencia de varios procesadores que, estando en un mismo ordenador, *trabajan a la vez*, compartiendo el resto de recursos, como por ejemplo la memoria.

1.5 Lectura: Software de sistema vs. Software de aplicaciones

Los sistemas de computación se componen de firmware, software, hardware, datos y recursos humanos, combinados para proporcionar una herramienta capaz de dar solución a diversos problemas. El software se puede clasificar en función del propósito para el cual fue elaborado.

El Sistema Operativo es el software más importante del sistema, al ser el único indispensable para poder hacer uso del computador. Éste se engloba dentro del **software del sistema**, que es el que proporciona un entorno de programación general en el cuál los programadores pueden crear y/o ejecutar aplicaciones específicas que satisfagan sus necesidades, es decir, el **software de aplicación**. Éste es el que hace que el computador sea útil en distintas áreas y actividades.

Los usuarios interactúan directamente con el software de aplicación y/o con el software del sistema directamente, sin tener que conocer ni preocuparse de los detalles del proceso físico. El software del sistema esconde al usuario la gran mayoría de estos detalles físicos de la máquina así como el funcionamiento básico de muchas tareas.

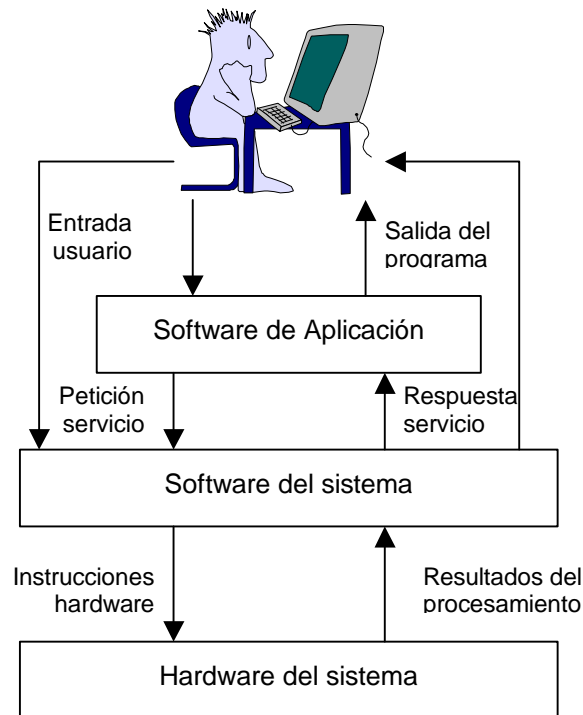


Figura 4: Interacción del usuario con las distintas categorías de software



1.5.1 EL SOFTWARE DEL SISTEMA

El *software del sistema* es un conjunto de programas que gestionan los recursos del sistema presentando la máquina de la manera más adecuada a cada usuario, para que el rendimiento total sea óptimo.

Originalmente, el software del sistema, y más en concreto los sistemas operativos se desarrollaron para proporcionar los servicios que los programadores precisaban a la hora de implementar software de aplicación. A lo largo del tiempo este propósito fue cambiando. También se hacía necesario lograr que varias aplicaciones software compartieran unos mismos recursos de la forma más eficiente y segura posible.

El programa más importante de esta categoría es el **sistema operativo**. Éste que controla las funciones básicas de un computador, proporciona una plataforma para los programas de aplicación, y hace de intermediario entre la parte física y el usuario.

Otros ejemplos de software del sistema los constituyen los sistemas de ventanas o los sistemas de gestión de base de datos. El primero de ellos es software del sistema que proporciona un terminal virtual a un programa de aplicación, posibilitando por ejemplo que una aplicación desarrollada en Visual Studio pueda hacer uso de las ventanas

estándares de Microsoft Windows (NT o 9x). Se dice que el terminal es virtual porque el programador, de hecho, escribe y lee de las ventanas como si se tratara de un dispositivo.

Un sistema de gestión de base de datos es un sistema completo capaz de almacenar información en los dispositivos de almacenamiento secundario de un ordenador como los discos, de manera que: se controla el almacenamiento de datos redundantes, los datos resultan independientes de los programas que los usan, se almacenan relaciones entre los datos junto con éstos, y es posible el acceso a los mismos de diversas maneras.

De hecho, el software del sistema proporciona un entorno a programas de aplicación y usuarios con servicios que no están disponibles directamente a nivel hardware.

De todas formas no todo software del sistema sirve para todas las aplicaciones y/o dominios. Así, una biblioteca gráfica será específica de cierto dominio de aplicación y puede que resulte totalmente inútil en otros. Probablemente el único totalmente independiente del dominio sea el sistema operativo, que interactúa con el hardware para dar servicio no sólo al software de aplicación sino al resto del software del sistema en todas aquellos casos en que éstos precisan recursos del sistema.

Para facilitar el uso del hardware, el sistema operativo proporciona un modelo abstracto de operación de los elementos hardware, generalizando su comportamiento pero limitando la flexibilidad de manipulación de algunos de ellos, es decir, se facilitan las tareas más comunes pero a cambio se pierde especificidad.

Los programas de aplicación y/o usuarios pueden precisar de muchos tipos de recursos hardware. Cualquier dispositivo particular, como una unidad de disco, tiene un interfaz determinado que delimita la manera de operar con él. Sin embargo, una abstracción puede resultar mucho más sencilla de utilizar puesto que no es necesario conocer ningún tipo de interfaz. Probablemente baste con hacer uso de algún tipo de ventana, función de alto nivel, etc.

1.5.2 EL SOFTWARE DE APLICACIÓN

Este software se diseña y se escribe para resolver problemas específicos y/ o automatizar diversos entornos. Existen numerosos programas de aplicación, casi tantos como necesidades de procesamiento de información.

Éstos pueden dividirse en programas de aplicación genéricos como los procesadores de textos, las hojas de cálculo, los sistemas de generación de nóminas, etc., y específicos.

Cada usuario simplemente debe seleccionar el programa de aplicación que más se ajuste a la tarea que desea realizar. Si no se encuentra un programa genérico bien en su objetivo o en su estructura, deberá recurrirse a un programa de aplicación específico, que no es sino un programa de aplicación diseñado y escrito mediante cualquiera de los lenguajes de programación existentes, para dar respuesta a una necesidad concreta de un usuario dado.

Hoy por hoy, es importante ser conscientes de la importancia que este tipo de software tiene, puesto que el coste de cualquier ordenador se justifica por el valor de su software de aplicación; es el coste del software de aplicación el que más influirá en la decisión final de compra. De hecho, tanto hardware como el software del sistema existen para dar soporte a la creación y/o uso de software de aplicación.

1.5.3 PREGUNTAS

Además el sistema operativo, ¿existe algún otro tipo de software del sistema?

¿Qué relación tiene el usuario del ordenador con el software del sistema? Explícalo.

¿Qué relación tiene el usuario del ordenador con el software de aplicación? Explícalo.

¿Cuál es el software imprescindible hoy en día para el funcionamiento de un ordenador? ¿Qué elementos lo componen?





Facultad de Ingeniería



Facultad de Ingeniería

