

# 1. Introducción

## 1.1. Evaluación del rendimiento de un sistema informático

Se podría definir la evaluación del rendimiento de un sistema informático como *la medida de cómo un software determinado está utilizando el hardware con una determinada combinación de programas, que constituyen lo que se denomina carga del sistema* (figura 1.1).

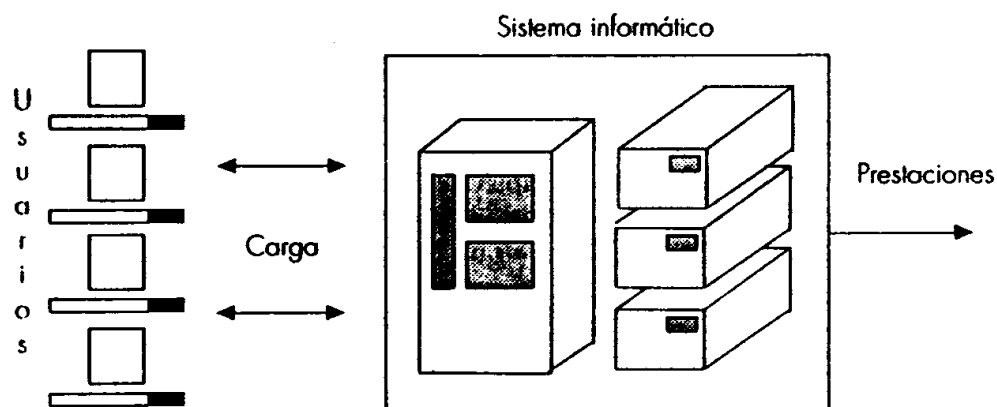


FIGURA 1.1.

Tal vez la mayor dificultad de la evaluación de prestaciones proviene del hecho que la carga real de un sistema informático cambia continuamente y las mediciones no pueden repetirse a no ser que se trabaje en un entorno controlado de carga, técnica que más adelante se describirá. Por otro lado, las variables o índices significativos de la medición de las prestaciones pueden variar de una instalación a otra, en función de la configuración y uso del sistema. Asimismo, los valores de esas variables o índices que permiten considerar el buen o mal funcionamiento del sistema también varían de una instalación a otra.

Los usuarios o las necesidades de servicio exigen que un sistema informático deba cumplir una serie de requisitos o especificaciones de ejecución para considerar que su funcionamiento es aceptable.

Actualmente, todos los usuarios de los sistemas informáticos, diseñadores, administradores, etc. están interesados en los estudios de evaluación del comportamiento de su sistema, con el objetivo de obtener las mejores prestaciones al mínimo costo.

Las actividades que forman parte del estudio del comportamiento de un sistema se denominan de *evaluación de sus prestaciones*.

Las prestaciones de un sistema son el criterio clave en el diseño, selección, compra y uso de un sistema informático.

### 1.2. Necesidad de la evaluación de prestaciones

Esta necesidad ha aparecido como una consecuencia natural del aumento de la potencia y de la complejidad de los sistemas informáticos. Los primeros computadores estaban concebidos para que los utilizara el propio programador que, personalmente, controlaba lo que sucedía en el sistema mientras se ejecutaba el programa. Eran los tiempos en que prácticamente no existía el software y en que los elementos fundamentales para la evaluación del comportamiento consistían en la longitud de la palabra, el conjunto de instrucciones y su implantación, el ciclo de base de la CPU, el tiempo de ejecución de una instrucción característica (normalmente la instrucción de sumar), etc.

La aparición de la ayudas software, la de los periféricos cada vez más sofisticados, y la de las unidades centrales más complejas (multiprocesadores, *pipelines*, memorias cache, etc.) con sistemas de interrupciones muy sofisticados, el aumento de la dimensión de las memorias, etc., han hecho que la evaluación del comportamiento se haya convertido en un cuerpo de doctrina en el que no sólo se ha de considerar el hardware, sino también las facilidades proporcionadas por el software al acercar la máquina a los usuarios, provocando entonces la aparición del *overhead* (es decir, de los gastos generales de la máquina para repartir los recursos entre los distintos usuarios) que lleva asociado el software. Además, la evaluación del comportamiento de un sistema hay que hacerla teniendo presente que ha de satisfacer a unos usuarios locales o remotos que, en general, se ven a sí mismos como únicos usuarios del sistema en la ejecución de sus programas. Estas consideraciones hacen comprender que la evaluación del comportamiento no es una tarea sencilla, ya que ha de tener en cuenta muchos y variados aspectos del hardware, del software y de las aplicaciones que se han de llevar a cabo en el sistema informático.

En consecuencia, es necesario evaluar un sistema, para comprobar que su funcionamiento es correcto, es decir, el esperado.

También va a influir esta evaluación en las decisiones de diseño, implantación, compra y modificación de los sistemas informáticos.

Así pues, la evaluación de un sistema es necesaria en todas las etapas de su ciclo de vida; por tanto, los evaluadores de un sistema deben ser sus diseñadores, fabricantes, vendedores, administradores y usuarios según la etapa que se considere.

La evaluación va a ser necesaria cuando un diseñador de sistemas quiere comparar un cierto número de diseños alternativos y encontrar el mejor; lo mismo le ocurre al administrador de un sistema, que quiere saber qué sistema es el más adecuado para ejecutar un determinado conjunto de aplicaciones.

Otro de los objetivos que se pretende al evaluar un sistema informático es encontrar los factores que impiden un funcionamiento adecuado, como pueden ser los cuellos de botella. También la predicción del comportamiento del sistema con nuevas cargas (planificación de la capacidad) constituye otro objetivo de la evaluación de prestaciones.

Para realizar la evaluación de un sistema se debe de partir de unos datos y se deben fijar unos objetivos. Estos deben de estar muy claros, es decir, hay que saber con antelación el problema que se va a resolver.

Por lo tanto es necesario evaluar un sistema informático cuando se quiere:

- Diseñar una máquina.
- Diseñar un sistema informático.
- Seleccionar y configurar un sistema informático.
- Planificar la capacidad del sistema informático.
- Sintonizar o ajustar un sistema informático.

En todos los casos expuestos se ha visto que el comportamiento del sistema no es algo independiente del entorno de trabajo sino que está íntimamente relacionado con el uso a que se destinará. Por ello, dentro del ámbito de la evaluación de prestaciones se incluye un apartado destinado a:

- Caracterizar y predecir la carga.

A continuación se describen estas distintas tareas.

### 1.2.1. Diseñar una máquina

Durante las fases de diseño de un computador hay que tomar una serie de decisiones que pueden tener una gran influencia en el comportamiento del mismo. Por lo tanto, es necesario estudiar el comportamiento del sistema antes de su implantación para ver cuál de las posibles opciones de diseño disponibles es la mejor teniendo en cuenta el entorno en el que se implantará.

### 1.2.2. Diseñar un sistema informático

Durante las fases de diseño de una aplicación informática, especialmente si se trata de un sistema de tiempo real o acoplado, hay que tomar una serie de decisiones que pueden tener una gran influencia en el comportamiento del mismo. Por lo tanto, es necesario estudiar el comportamiento del sistema antes

de su implantación para ver cuál de las posibles opciones de diseño disponibles es la mejor teniendo en cuenta el entorno de trabajo establecido.

### 1.2.3. Seleccionar y configurar un sistema informático

Cuando se desea seleccionar una nueva máquina, hay que ver qué configuración es la más adecuada para atender un determinado conjunto de trabajos y qué fabricante ofrece mejor relación precio/prestaciones para esa situación.

### 1.2.4. Planificar la capacidad del sistema informático

A lo largo de la vida de un sistema, por diversas razones, como el aumento vegetativo de la carga o la incorporación de nuevas aplicaciones, es necesario aumentar su capacidad y, en consecuencia, es necesario planificar la capacidad del sistema con posibles nuevas configuraciones.

### 1.2.5. Sintonizar o ajustar un sistema informático

Debido al crecimiento vegetativo de la carga de un sistema informático, cuando las prestaciones del mismo decrecen, es necesario ajustar o cambiar alguno de los parámetros del sistema operativo, normalmente para mejorar el comportamiento del sistema. En algunos casos si el sistema (hardware más software básico) no se puede variar, hay que intentar mejorar el comportamiento del sistema modificando la carga (programas).

### 1.2.6. Caracterizar y predecir la carga

El comportamiento de un sistema es muy dependiente de la carga aplicada al mismo; por lo tanto conocer o caracterizar la carga que tiene un sistema y predecir su carga futura es fundamental para poder estudiar el comportamiento del mismo.

### 1.2.7. Conclusión

Siempre que se observe un problema en el rendimiento de un sistema informático o queramos seleccionar una configuración o un sistema o diseñar una máquina es necesario proceder a evaluar sus prestaciones.

## 1.3. Magnitudes que se deben medir

Para evaluar el comportamiento de un sistema hay que hacer referencia a unas medidas cuantitativas o unos parámetros que van a caracterizar el comportamiento del hardware y software del ordenador o que van a hacer referencia a cómo el usuario (visión externa) y el responsable del sistema (visión interna) ven su comportamiento.

Estas magnitudes y parámetros están relacionados con tres tipos de medidas correspondientes a:

- Consumo de tiempos.

- Utilización de recursos o dispositivos.
- Trabajo realizado por el sistema o componentes del mismo.

La relación que se expone a continuación no pretende ser una lista exhaustiva de las magnitudes que siempre deben medirse en todas las instalaciones sino una lista de variables que normalmente se acostumbran a medir, pero sin que ello sea óbice en cada instalación concreta para añadir variables no consideradas aquí o para eliminar algunas de ellas por carecer de significado en un contexto dado.

#### 1.3.1. Variables externas o perceptibles por el usuario

##### *a) Productividad o throughput*

Es la cantidad de trabajo útil ejecutado por unidad de tiempo en un entorno de carga determinado (normalmente se mide en trabajos por hora o en transacciones por segundo).

##### *b) Capacidad*

Es la máxima cantidad de trabajo útil que se puede realizar por unidad de tiempo en un entorno de carga determinado.

##### *c) Tiempo de respuesta*

Es el tiempo transcurrido entre la entrega de un trabajo o una transacción al sistema y la recepción del resultado o la respuesta.

#### 1.3.2. Variables internas o del sistema

##### *a) Factor de utilización de un componente*

Es el porcentaje de tiempo durante el cual un componente del sistema informático (CPU, canal, dispositivo de E/S, etc.) ha sido realmente utilizado.

##### *b) Solapamiento de componentes*

Es el porcentaje de tiempo durante el cual dos o más componentes del sistema han sido utilizados simultáneamente.

##### *c) Overhead*

Es el porcentaje de tiempo que los distintos dispositivos del sistema (CPU, discos, memoria, etc.) han sido utilizados en tareas del sistema no directamente imputables a ninguno de los trabajos en curso.

##### *d) Factor de carga de multiprogramación*

Es la relación entre el tiempo de respuesta de un trabajo en un determinado entorno de multiprogramación y su tiempo de respuesta en monoprogramación.

*e) Factor de ganancia de multiprogramación*

Es la relación entre el tiempo total necesario para ejecutar un conjunto de programas secuencialmente en monoprogramación y en multiprogramación.

*f) Frecuencia de fallo de página*

Es el número de fallos de página que se producen por unidad de tiempo en un sistema de memoria virtual paginada.

*g) Frecuencia de swapping*

Es el número de programas expulsados de memoria por unidad de tiempo a causa de falta de espacio o con el fin de permitir su reorganización para recuperar espacio en ella o para disminuir la paginación.

### 1.3.3. Otras magnitudes relativas al comportamiento

Hay otras medidas relacionadas con el comportamiento del sistema pero no directamente con las prestaciones que en muchos casos también es importante tenerlas en cuenta, como:

- *Fiabilidad:* Es la función del tiempo definida como la probabilidad condicional de que el sistema trabajará correctamente a lo largo del intervalo de tiempo  $[t_0, t]$ , si trabajaba correctamente en el instante  $t_0$ . En otras palabras es la probabilidad que el sistema trabaje correctamente a lo largo de un intervalo de tiempo dado. Se mide por la probabilidad de fallos por unidad de tiempo, o por el tiempo medio entre fallos.
- *Disponibilidad:* Es la función del tiempo definida como la probabilidad de que el sistema esté trabajando correctamente y que esté disponible para realizar sus funciones en el instante considerado  $t$ . Incluye la posibilidad de que el sistema pueda haber estado averiado y posteriormente reparado.
- *Seguridad:* Es la probabilidad de que el sistema esté realizando correctamente sus funciones o parado de forma tal que no perturbe el funcionamiento de otros sistemas ni comprometa la seguridad de las personas relacionadas con él.
- *Performabilidad:* En muchos casos, es posible diseñar un sistema que continúe trabajando correctamente después de la aparición de fallos de hardware o software, pero con un nivel de prestaciones de alguna forma disminuido. Es pues una función del tiempo definida como la probabilidad de que las prestaciones del sistema estarán por encima de un cierto nivel en un instante determinado.
- *Mantenibilidad:* Es la medida de la facilidad con que un sistema puede ser reparado después de un fallo. En términos cuantitativos es la probabilidad de que un sistema averiado pueda ser reparado y devuelto al estado operacional dentro de un período de tiempo determinado.

## 1.4. Magnitudes que caracterizan la carga

Tanto en el caso de tomar medidas de un sistema existente como cuando se recaban datos para predecir el comportamiento de uno inexistente, se plantea el problema de representar la carga de manera que se pueda considerar que el sistema se comporta de forma característica para proceder entonces a efectuar las mediciones. En consecuencia, para evaluar correctamente las prestaciones de un sistema informático, la carga debe de ser seleccionada cuidadosamente.

El término *carga de prueba* se utiliza para denominar la carga usada en el estudio de prestaciones. Esta carga puede ser *real* o *sintética*.

La carga *real* se observa en un sistema durante su funcionamiento normal. Su inconveniente es que no permite repeticiones para eliminar los errores de medición, y por ello es difícilmente utilizable como *carga de prueba*.

La carga *sintética* está constituida por un conjunto de programas extraídos o no de la carga real del sistema informático que la reproduce de forma compacta. Se utiliza más porque puede utilizarse repetidamente y también porque puede modificarse sin afectar a la operatividad del sistema.

En muchos sistemas la evaluación se suele realizar en un sistema aparte del real, es decir, creándose dos sistemas paralelos. Uno será el sistema real y el otro el de test. En el primero trabajarán, por así decirlo, aquellos usuarios que explotan el sistema. En el segundo, trabajarán aquellos técnicos que desarrollan el sistema o que están encargados de su buen funcionamiento.

El común denominador de todos estos problemas reside en la determinación de las magnitudes que caracterizan la carga del sistema. Dependerán del tipo y modo de trabajo del sistema. Algunas de estas magnitudes características se exponen a continuación.

### 1.4.1. Para cada componente de la carga

#### a) *Tiempo de CPU por trabajo*

Es el tiempo total de CPU necesario para ejecutar un trabajo (programa, transacción, etc.) en un sistema determinado. Evidentemente es función directa del número de instrucciones que se ejecutan para realizar ese trabajo, del volumen de datos procesados y de la velocidad del procesador.

#### b) *Número de operaciones de E/S por trabajo*

Es el número total de operaciones de entrada/salida que requiere la ejecución de un trabajo. Evidentemente dicho número total conviene desglosarlo según el dispositivo, el archivo, etc. sobre el que se realizan.

#### c) *Características de las operaciones de E/S por trabajo*

Hacen referencia al soporte (cinta, disco, etc.) y, en el caso de discos, a la posición que ocupa el archivo sobre el que se efectúan. Todo ello tiene una influencia directa en el tiempo necesario para realizar una operación de E/S.

*d) Prioridad*

Es la que el usuario asigna a cada uno de los trabajos que somete al sistema.

*e) Memoria*

Es la que requiere ocupar, para su ejecución, un trabajo determinado. Puede ser constante –memoria real– o variable –memoria virtual paginada (conjunto de trabajo) o segmentada–, según la gestión que el sistema operativo haga de la memoria.

*f) Localidad de las referencias*

Es el tiempo en el que todas las referencias a memoria hechas por un trabajo permanecen dentro de una página (segmento) o conjunto de páginas (segmentos).

Si se considera la ejecución de un programa como una sucesión de referencias a memoria, la localidad del programa será tanto mayor cuanto más tiempo esté dentro de la página (segmento) o conjunto de páginas (segmentos) considerados. Es la aparente contradicción de medir una magnitud ligada al espacio (localidad) mediante un tiempo.

1.4.2. Para el conjunto de la carga

*a) Tiempo entre llegadas*

Es el tiempo entre dos requerimientos sucesivos para un servicio (ejecución de un trabajo o transacción) del sistema.

*b) Frecuencia de llegada*

Es el número medio de llegadas de nuevas peticiones de ejecución que se producen por unidad de tiempo. Evidentemente es la inversa del tiempo medio entre llegadas.

*c) Distribución de trabajos*

Define la proporción existente entre las ejecuciones de los distintos trabajos que constituyen la carga.

1.4.3. Para cargas conversacionales

*a) Tiempo de reflexión del usuario*

Es el tiempo que el usuario de un terminal de un sistema interactivo necesita para generar una nueva petición al sistema (es decir, es el tiempo de leer la respuesta previa, de pensar en la nueva acción que se vaya a tomar, función de la respuesta recibida y de teclearla).



**b) Número de usuarios simultáneos**

Es el número de usuarios interactivos que trabajan simultáneamente sobre el mismo sistema en un instante dado.

**c) Intensidad del usuario**

Es la relación entre el tiempo de respuesta de una petición y el tiempo de reflexión del usuario.

**1.5. Magnitudes para controlar el comportamiento**

Hasta ahora se ha estudiado lo que hay que medir y cómo se escoge cuándo hay que medir; pero, ¿qué hay que hacer si el comportamiento del sistema no es satisfactorio? ¿Cuáles son los resortes que hay que mover para mejorar el comportamiento del sistema?

Las modificaciones que se pueden introducir en un sistema para mejorar su comportamiento pueden hacerse en todos los niveles que influyen en el comportamiento del mismo mediante:

- Ajuste de los parámetros del sistema operativo.
- Modificación de las políticas de gestión del sistema operativo.
- Equilibrado de la distribución de cargas.
- Modificación o sustitución de componentes hardware del sistema.
- Modificación de los programas.

A continuación se analizan con algún detalle el significado de estas distintas opciones.

**1.5.1. Ajuste de los parámetros del sistema operativo**

La relación que sigue no pretende ser exhaustiva, ni pretende que en todos los sistemas operativos puedan modificarse con la misma facilidad los parámetros expuestos, pero sí es una lista de parámetros de un sistema operativo que pueden modificarse con facilidad y cuya variación puede influir en el comportamiento del sistema.

**a) Tamaño del quantum**

Es la cantidad de tiempo de uso ininterrumpido de la CPU que un sistema de tiempo compartido asigna a los diferentes trabajos. En ciertos sistemas, no existe uno sino varios quanta para las distintas prioridades internas de los distintos trabajos por lo que no es un solo quantum el que hay que cambiar sino buscar un equilibrio adecuado entre todos ellos.

Es evidente que si el quantum es demasiado grande se favorece a los trabajos con mucho uso de CPU y si es demasiado pequeño se puede introducir un *over-*

*head* importante con los continuos cambios de contexto de un programa a otro cada vez que se agota el quantum.

*b) Prioridad interna*

Es el nivel inicial de prioridad interna que recibe un programa en función de la prioridad externa asignada.

*c) Factor de multiprogramación*

Es el número máximo de trabajos que están simultáneamente en memoria principal y, por lo tanto, que tienen opción a utilizar la CPU y los demás recursos activos del sistema.

Cuanto mayor sea este valor tanto mejor aprovechamiento se puede tener de todos los recursos del sistema, aunque también aumentará el *overhead*.

*d) Tamaño de la partición de memoria*

Es la cantidad fija de memoria principal asignada a una cola de trabajos.

Hay que adecuar esas particiones a los tamaños de los programas de la instalación y a su frecuencia de ejecución.

*e) Tamaño de la ventana*

Es el intervalo de tiempo durante el cual el sistema toma medidas para determinar el conjunto de trabajo de un programa en un entorno de memoria virtual paginada que use esa política.

Evidentemente, según el período de tiempo durante el que se tomen medidas para determinar el conjunto de trabajo, éste varía y, por lo tanto, el valor medio del conjunto de trabajo estará afectado por los valores que intervengan en su cálculo.

*f) Máxima frecuencia de fallo de página*

Es la frecuencia máxima de fallo de página permitida. A partir del instante en que se alcanza, se efectúa la suspensión o *swapping* de alguno de los trabajos en curso para evitar el excesivo *overhead* que se estaba generando.

*g) Índice de supervivencia de las páginas*

Es el número de ráfagas de CPU recibidas por un programa antes de que saque de la memoria principal una página que no haya sido referenciada durante ese período.

*h) Número de usuarios simultáneos*

Es el máximo número de usuarios de terminal permitidos por un sistema de tiempo compartido.

### 1.5.2. Modificación de las políticas de gestión del sistema operativo

Teniendo en cuenta que las políticas propuestas por los fabricantes en sus sistemas operativos son para atender eficientemente a usuarios con una carga promedio, es posible que, en determinados sistemas, aquéllas no sean las más adecuadas en un caso concreto, por lo que puede ser conveniente y recomendable la sustitución de la rutina encargada de la gestión de un determinado recurso por otra que realice una política más idónea a unas necesidades concretas.

Este cambio lleva asociado (evidentemente, el riesgo de problemas en el sistema operativo durante el período de depuración y puesta a punto de la nueva rutina. Además hay que tener en cuenta que habrá que adaptar el cambio a las sucesivas versiones del sistema operativo.

### 1.5.3. Equilibrado de la distribución de cargas

El ideal, inalcanzable, de utilizar por igual todos los dispositivos del sistema informático debe sustituirse por el de utilizarlos de la forma más uniforme posible. No obstante, con frecuencia, el uso de recursos es notablemente desequilibrado y se deben disponer los cambios (de asignación de dispositivos periféricos a los canales, de ubicación de archivos en disco, de ejecución simultánea de programas que requieren los mismos recursos, de distribución de componentes software en la jerarquía de memoria, de políticas de explotación, etc.) necesarios para tender a lograr el equilibrio deseado.

Este tipo de corrección acostumbra, en muchos casos, a proporcionar mejoras espectaculares en el comportamiento del sistema.

### 1.5.4. Sustitución o ampliación de los componentes del sistema

Cuando el recurso a los métodos anteriores resulta ineficaz o inaplicable, se está abocado a la modificación de la configuración del sistema, bien sea sustituyendo determinados elementos por otros de mayor capacidad o rapidez, bien sea por aumento del número de dispositivos que constituyen la configuración del sistema. No obstante, la ampliación de la configuración debe hacerse de forma tal que se despeje el cuello de botella (elemento cuya saturación impide la mejora del comportamiento) que se pueda haber detectado, pues, de lo contrario, el comportamiento conjunto del sistema ampliado no varía de forma significativa.

### 1.5.5. Modificación de los programas

Para mejorar el comportamiento de un sistema informático puede recurrirse a modificar los programas de forma que su ejecución promedio requiera menos recursos (tiempo de CPU, número de E/S, etc.), bien sea por recodificación de los caminos del programa recorridos con mayor asiduidad, bien sea por un montaje que agrupe en la misma página o segmento aquellos módulos del programa que deben coexistir en memoria para la ejecución del programa, etc. Además es

conveniente verificar periódicamente que las hipótesis que justificaron una determinada concepción de un programa siguen siendo válidas.

Este procedimiento de mejora provoca la modificación de la carga, por lo que, aun cuando es una forma eficaz y frecuente de mejorar el comportamiento del sistema, normalmente se considerará la carga como un dato del problema que no se puede modificar.

## 1.6. Sistemas de referencia

A lo largo del texto se hará constante referencia a tres tipos de sistemas o tipos de funcionamiento de un sistema informático:

- Sistema por lotes o *batch*.
- Sistema interactivo o por demanda.
- Sistema transaccional.

Estos sistemas servirán de referencia para el estudio de las prestaciones y su evaluación.

### 1.6.1. Trabajo en lotes o *batch*

En muchas instalaciones de empresas, bancos, etc., el ordenador se queda por la noche en marcha ejecutando una serie de programas que se dejan almacenados en memoria. En estos sistemas (figura 1.2), el responsable de la explotación decide los trabajos que deben estar en ejecución en cada instante; así pues, la planificación interna del sistema operativo está ayudada por la externa humana.

Estos trabajos realizan ciclos de uso de la CPU y de los discos de forma continua hasta que finalizan.

Índices característicos de las prestaciones de estos sistemas son los siguientes:

- *Turnaround time* (equivalente al tiempo de respuesta en estos sistemas) es el tiempo que transcurre desde que se lanza la ejecución de un trabajo hasta que termina y en el que la generación de cada petición depende de la recepción de la respuesta en la petición.
- Productividad medida en trabajos por unidad de tiempo (segundo u hora).

### 1.6.2. Sistema transaccional

Un sistema transaccional es aquél en que un conjunto de terminales remotos conectados al sistema interaccionan con un conjunto determinado de programas; cada una de las interacciones constituye lo que se denomina una transacción (figura 1.3). Como ejemplos de este tipo de sistemas se pueden citar el sistema informático de un banco o caja de ahorros, o el de un sistema de reserva de billetes, o el que recibe medidas de un satélite. En sistemas de este tipo el sis-

tema de planificación interna del sistema operativo ha de ser capaz de gestionar las peticiones que llegan al sistema y no está ayudado en forma alguna por la planificación humana externa ya que desde cada terminal se tiene la sensación de ser el dueño absoluto del sistema y se tiene la posibilidad de lanzar cualquier transacción en cualquier instante.

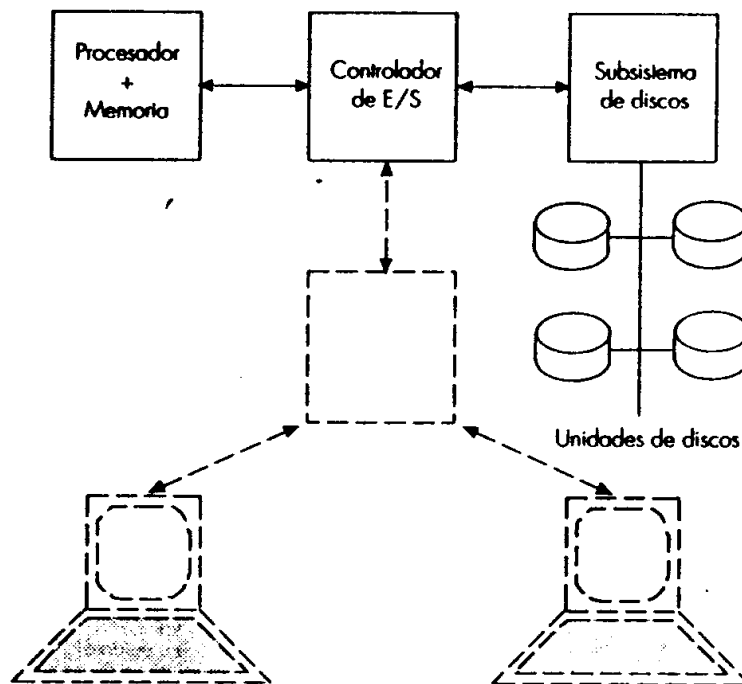


FIGURA 1.2.

El sistema de este tipo queda definido por el flujo de transacciones que le llega y su índice de prestaciones característico es:

- Tiempo de respuesta, que es la suma de los tiempos siguientes:
  - *Tiempo de reacción*, que es el tiempo que transcurre desde que la transacción llega al sistema hasta que comienza su ejecución.
  - *Tiempo de ejecución*, que es el que transcurre desde que el sistema comienza la ejecución de la transacción hasta que termina.
  - *Tiempo de retorno*, que es el que transcurre desde que finaliza la ejecución hasta que, eventualmente, se completa la respuesta hacia el usuario.

### 1.6.3. Sistema interactivo o por demanda

Un sistema interactivo (figura 1.4) es aquél en que los usuarios acceden a él desde terminales remotos teniendo acceso a la totalidad del sistema operativo. En

estos sistemas, un usuario desde un terminal, después de un tiempo de pensar o de reflexión, da una orden al terminal que pasa a procesarse por el conjunto CPU y discos y que después de un tiempo dado produce una respuesta en el terminal. Se trabajará de modo interactivo de acuerdo con el siguiente ciclo:

ORDEN → RESPUESTA → ORDEN → RESPUESTA → ...

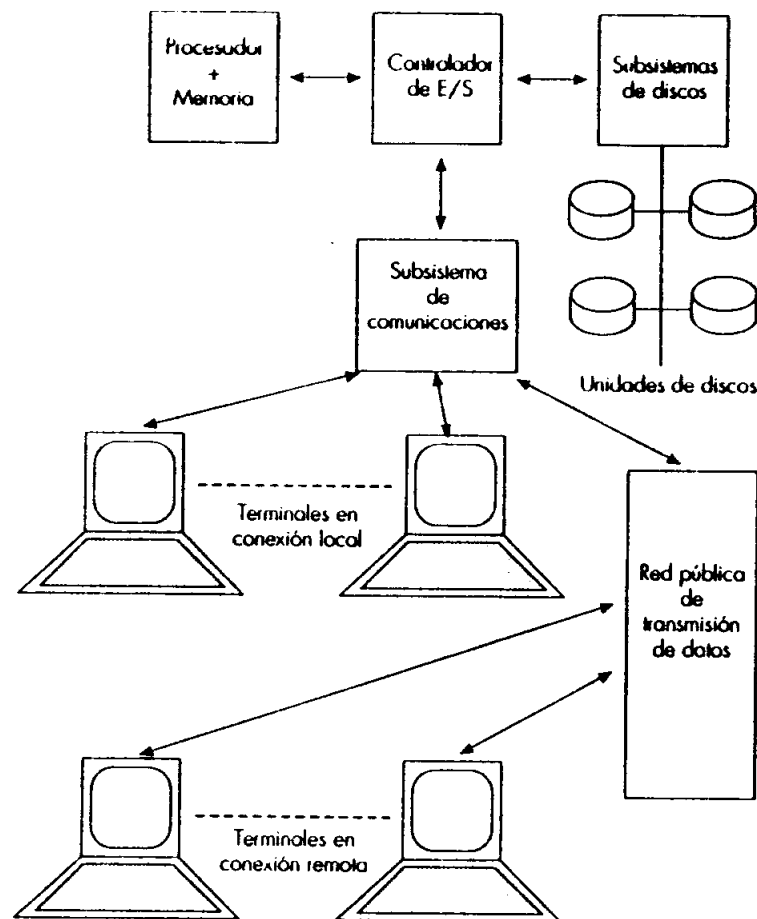


FIGURA 1.3.

En estos sistemas, pues, la generación de una nueva petición depende de la recepción de la respuesta a la petición anterior.

En estos sistemas tampoco la planificación humana ayuda a la planificación del sistema operativo. Un sistema de este tipo queda definido por:

- Número de usuarios que tiene conectados.
- Tiempo de reflexión de los usuarios, que es el tiempo que transcurre desde que el usuario recibe la respuesta hasta que éste finaliza la preparación de la siguiente petición. Incluye, por lo tanto, el tiempo necesario para leer la respuesta del sistema, reflexionar sobre la siguiente acción y teclearla.

Y los índices de prestaciones característicos son:

- Tiempo de respuesta, que será la suma de los tiempos siguientes:
  - *Tiempo de reacción*, que es el tiempo que transcurre desde que el usuario finaliza la petición hasta que el sistema comienza su ejecución.
  - *Tiempo de ejecución*, que es el que transcurre desde que el sistema comienza la ejecución de la petición hasta que termina.
  - *Tiempo de retorno*, que es el que transcurre desde que finaliza la ejecución hasta que se completa la recepción de la respuesta en el usuario.
- Productividad, medida en peticiones por unidad de tiempo, aunque, como se verá más adelante, estas cuatro variables están relacionadas y el conocimiento de tres de ellas permite determinar la cuarta.

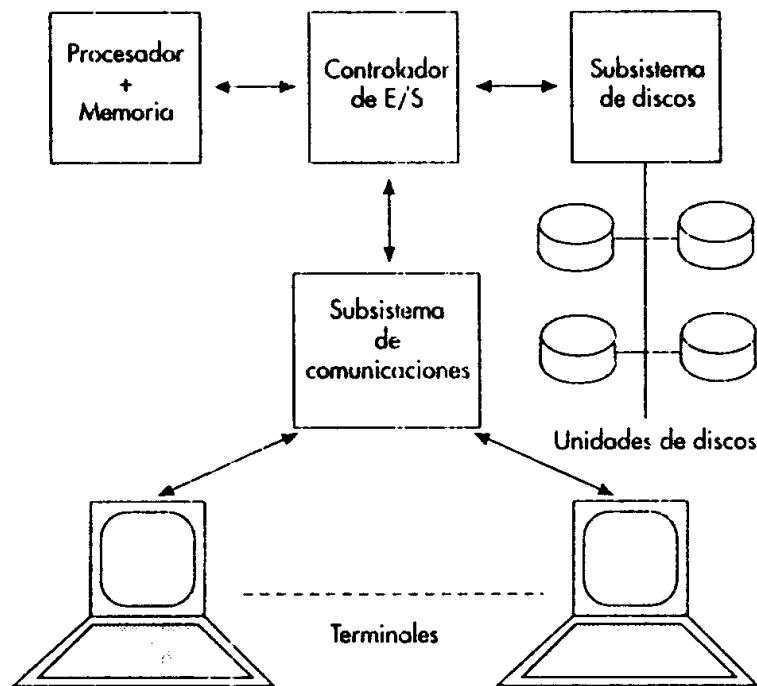


FIGURA 1.4.

### 1.7. Técnicas más comunes a la hora de evaluar un sistema

Se denominan *técnicas de evaluación* a los métodos y herramientas que permiten obtener los índices de prestaciones de un sistema que está ejecutando una carga dada y con unos valores determinados de parámetros del sistema.

Las técnicas pueden ser de tres tipos:

- Monitorización
- Modelado
- *Benchmarking*

### 1.7.1. Monitorización

Los monitores son las herramientas de medición que permiten seguir el comportamiento de los principales elementos de un sistema informático cuando éste se halla sometido a una carga de trabajo determinada. Pueden existir realizaciones hardware, software o mixtas. Aunque su objetivo es la medición de las prestaciones, se les denomina monitores ya que, debido a la imposibilidad de reproducir situaciones con la carga real, estos instrumentos hacen un seguimiento de lo que sucede en el sistema, es decir, lo monitorizan.

Estas herramientas son imprescindibles para evaluar el comportamiento de un sistema existente, a pesar de las perturbaciones que pueden introducir en el sistema cuyo comportamiento se va a evaluar.

Aparte de su utilización directa para tomar medidas de un sistema existente, permiten determinar la aproximación de una carga de *benchmark* a la carga real, obtener datos para la construcción de modelos y su validación posterior, etc.

### 1.7.2. Modelado

Ésta es la herramienta que hay que utilizar cuando se trata de evaluar el comportamiento de un sistema en el que hay algún elemento (hardware o software) que no está instalado.

En general, se fundamentan en la teoría de colas, pudiéndose considerar las colas o bien de forma individual o bien unidas formando redes abiertas o cerradas. Su tratamiento se puede realizar mediante los métodos analíticos que proporcionan las teorías de colas y de redes de colas, o por medio de la simulación.

Las *técnicas de simulación* consisten en la construcción de un programa que reproduce el comportamiento temporal del sistema, basándose en sus estados y sus transiciones. Los resultados se obtienen por extracción de estadísticas del comportamiento simulado del sistema.

Las técnicas analíticas se basan en la resolución mediante fórmulas cerradas o algoritmos aproximados de las ecuaciones matemáticas que representan el equilibrio que existe entre los eventos o transiciones de estado discontinuas que se producen en el sistema.

La limitación de los métodos analíticos es su incapacidad para tratar determinadas estructuras y comportamientos de las colas que existen en los sistemas informáticos. Los métodos de simulación no tienen estas limitaciones, pero, en general, son mucho más caros que los analíticos tanto en tiempo de cálculo como en esfuerzo de puesta a punto.



No obstante, la principal dificultad de esta herramienta reside en la obtención de datos lo suficientemente precisos para ejecutar el modelo y obtener resultados con el grado de aproximación que se exige.

### 1.7.3. *Benchmarking*

Es un método bastante frecuente de comparar sistemas informáticos frente a una carga característica de una instalación concreta, efectuándose la comparación, básicamente, a partir del tiempo necesario para su ejecución. Generalizando se puede considerar como la medición del comportamiento sobre un prototipo. Variantes de este método se usan para evaluar la potencia relativa de un sistema a lo largo de su ciclo de vida, para contrastar monitores y para validar modelos.

Las principales dificultades que se plantean son:

- cómo determinar esa carga característica, de forma que sea suficientemente reducida para ser manejable y suficientemente extensa para ser representativa;
- cómo valorar el aprovechamiento que hacen los programas de las peculiaridades de los distintos sistemas.