

**UNIVERSIDAD NACIONAL DEL**  
**NORDESTE**

*FACULTAD DE CIENCIAS NATURALES Y EXACTAS Y  
AGRIMENSURA*



**“Sistemas Operativos”**

**ANALISIS COMPARATIVO DEL**  
**RENDIMIENTO**

**MONOGRAFÍA DE ADSCRIPCIÓN**

**PROF.: Dr. David L. La Red Martínez**

**PUA: Bruno Benjamín Vergara**

**2011**

## CONTENIDO

- *Introducción*
- *Necesidad de la evaluación de prestaciones*
- *Objetivos de una evaluación*
- *Técnicas de evaluación de un sistema informático*
- *Medidas del rendimiento*
- *Estrategia de comparación: ratios*
- *Magnitudes que se deben medir*
- *Otras magnitudes relativas al comportamiento*
- *Caracterización de la carga*
- *Representatividad de la carga*
- *Magnitudes que caracterizan la carga*
- *Concepto de Benchmark*
- *Tipos de Benchmark*
- *Niveles de Benchmark*
- *Magnitudes para controlar el comportamiento*
- *Conclusión*
- *Bibliografía*

# INTRODUCCION

Antes que nada es indispensable aclarar lo que significa “rendimiento” o “desempeño” de un sistema informático, ya que es un concepto de gran importancia para tratar el tema en cuestión, y es un criterio que expresa la manera o la eficiencia con la que un sistema de computación cumple con sus metas.

Por ello es tan sustancial llevar a cabo la evaluación del rendimiento de los sistemas debido a lo expresado con anterioridad, puesto que permite mejorarlo o compararlo con otros sistemas informáticos, de aquí la conveniencia hoy en día de llevar a cabo la evaluación del sistema durante todo su ciclo de vida, esto quiere decir que, es necesaria la evaluación en cada etapa del sistema informático, ya sea en su desarrollo, en su diseño, o en su implementación, etc. Existen muchas instalaciones que realizan muy poco o ningún control y evaluación, o bien, realizando los controles generan grandes cantidades de datos que muchas veces no se sabe cómo interpretar o qué hacer con ellos, debido a que rara vez cuentan con personal versado en las técnicas de análisis de rendimiento.

Durante los primeros años del desarrollo de las computadoras, el hardware representaba el costo dominante de los sistemas y debido a ello los estudios de rendimiento se concentraban en el hardware, actualmente y según la tendencia apreciable el software representa una porción cada vez mayor de los presupuestos informáticos, incluye el S. O. de multiprogramación / multiproceso, sistemas de comunicaciones de datos, sistemas de administración de bases de datos, sistemas de apoyo a varias aplicaciones, etc.

El campo de la evaluación actualmente tiende a usar índices que tienen en cuenta el tiempo de ejecución, de un conjunto de programas de pruebas (*benchmarks*) en un computador como la unidad de medición más confiable. Estos programas benchmarks tienen distintos caracteres y naturalezas, ello depende de qué se quiere evaluar de un sistema informático. Puede variar desde un nivel más bajo ligado a componentes (*procesador + sistema de memoria + compilador*), hasta un nivel más elevado representado por el sistema completo (*servidor web o servidor de correo electrónico*).

Además a la hora de hablar de Análisis Comparativo del Rendimiento hay que tener presente otro concepto clave, la “carga”, es decir, las demandas de servicios que realizan los usuarios de un sistema en un intervalo de tiempo. No se puede hablar de prestaciones sin hablar de la carga con la que se han tomado las medidas, y para poder comparar diferentes sistemas y configuraciones, será imprescindible aplicarles una carga. La mayor dificultad en la evaluación comparativa de las prestaciones precisamente proviene del hecho que la carga real de un sistema cambia continuamente y las mediciones no pueden repetirse a no ser que se trabaje en un entorno controlado de carga.

# NECESIDAD DE LA EVALUACIÓN DE PRESTACIONES

Es una consecuencia natural del aumento de la potencia y de la complejidad de los sistemas.

En los primeros tiempos los ordenadores eran concebidos para que sean utilizados en su totalidad por el programador (prácticamente no existía el software), y los elementos fundamentales para la medición eran la longitud de la palabra del ordenador, el conjunto de instrucciones y su implementación, el ciclo de base de la CPU, el tiempo de ejecución de una instrucción característica (ej.: la instrucción sumar), etc.

La aparición del software, la de los periféricos cada vez más sofisticados, y 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, 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 todo software.

Además la evaluación del comportamiento de un sistema hay que hacerla teniendo presente que ha de satisfacer a usuarios locales y usuarios remotos que, en general, se ven a sí mismos como únicos usuarios del sistema en la ejecución de sus programas.

Todas estas consideraciones hacen comprender que la evaluación del comportamiento no es 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, evaluamos un sistema, para comprobar que su funcionamiento es el correcto, es decir, el esperado.

Para evaluar un sistema debemos partir de unos datos y se deben fijar unos objetivos. Estos deben 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 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.

## ***Diseñar una máquina***

Durante el diseño de un ordenador 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 implementación para ver cuál de las posibles opciones de diseño disponibles es la mejor teniendo en cuenta el entorno en el que se implementará.

## ***Diseñar un sistema informático***

Durante las fases de diseño de una aplicación informática, especialmente si se trata de un sistema en tiempo real, hay que tomar una serie de decisiones que pueden tener influencia en el comportamiento del mismo. Por lo tanto, es necesario estudiar el comportamiento del sistema antes de su implementación para ver cuál de las posibles opciones de diseño disponibles es la mejor teniendo en cuenta el entorno de trabajo establecido.

## ***Seleccionar y configurar un 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.

## ***Sintonizar o ajustar un sistema informático***

Debido al crecimiento vegetativo de la carga de un sistema, 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).

## ***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.

En todos los casos mencionados, un problema en el rendimiento de un sistema, es seleccionar una adecuada configuración o diseñar una máquina y/o un sistema, es necesario proceder a evaluar sus prestaciones.

# OBJETIVOS DE UNA EVALUACIÓN

Los objetivos de la evaluación del rendimiento son los siguientes:

- Comparar distintas alternativas;
- Determinar el impacto de un nuevo elemento o característica en el sistema (por ejemplo: la adición de un disco duro nuevo);
- Sintonizar el sistema, es decir, hacer que funciones mejor desde algún punto de vista;
- Medir prestaciones relativas entre diferentes sistemas;
- Depuración de prestaciones, es decir, identificar los fallos del sistema que hacen que vaya más lento (cuello de botellas).

Un ordenador se compone de muchos subsistemas diferentes, tanto software, como hardware, y todos interaccionan entre sí para dar el resultado que observa un usuario. El procesador, los diferentes elementos de la jerarquía de memoria, el sistema operativo, compiladores, la cantidad de usuarios, todos tienen un impacto en las prestaciones del sistema. En concreto, los sistemas con los que trabaja el ordenador se pueden dividir de la forma siguiente:

Hardware: la parte física del ordenador, compuesta a su vez del subsistema CPU-memoria, entrada/salida, y conectada a este subsistema, la red, el sistema de almacenamiento rápido, y los gráficos, principalmente, aparte de otros subsistemas, que no suelen influir tanto en las prestaciones (por ejemplo, sistemas de copias de respaldo, impresoras, CDs, etc.).

Software: también el sistema operativo y los programas de usuario hacen uso de diferentes *capas*, que van desde la más baja llamada *capa de abstracción de hardware*, para pasar al *kernel* o *microkernel*. Más arriba están los diferentes subsistemas del sistema operativo: entrada salida, seguridad, gestión de objetos, gestor de procesos, memoria virtual y demás.

En el caso de un sistema monousuario, también es necesaria la evaluación de prestaciones con el objeto de aprovechar al máximo las posibilidades del hardware; la mayoría de los sistemas operativos serios ofrecen una serie de posibilidades de medición y sintonización; en el caso de un sistema informático compuesto por una red con varias decenas de recursos, periféricos, usuarios, y que se use las 24 horas del día, y en el cual la adquisición de un nuevo componente sea un proceso largo y costoso, será imprescindible evaluar los problemas del sistema y sus posibles soluciones.

Una primera aproximación a la medición de prestaciones son las ofrecidas por el fabricante de cada uno de los componentes del ordenador. Un ordenador con una tarjeta de vídeo rápida será habitualmente más rápido que otro con la tarjeta de vídeo más lenta, o con más memoria (al tener más memoria, hay que transferir más información en cada instante, y por tanto hace más lento el sistema), si todos los demás componentes son iguales. Pero ¿habría mucha diferencia de velocidad, si la velocidad es lo que importa, entre una tarjeta rápida con mucha memoria y otra lenta con menos memoria?. ¿Cómo influiría en eso la velocidad del microprocesador?. ¿Y el tipo de bus al que se conecta la tarjeta?. Todas estas preguntas son habituales cuando hablamos de evaluación.

La mayor dificultad en la evaluación de prestaciones proviene del hecho que la carga real en un sistema cambia continuamente y las mediciones no pueden repetirse a no ser que se trabaje en un entorno controlado de carga.

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 el uso del sistema. Asimismo los valores de las 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 exigen que el sistema deba cumplir una serie de requisitos o especificaciones de ejecución para considerar que su funcionamiento es aceptable. Hoy en día, todos los usuarios del sistema, administradores, analistas, diseñadores, programadores, etc., están interesados en los estudios de evaluación del comportamiento de un sistema, con el objetivo de obtener las mejores prestaciones al mínimo costo. Además la predicción del comportamiento del sistema con nuevas cargas (planificación de capacidad) constituye otro objetivo de la evaluación de prestaciones. Otros de los objetivos que se pretende al evaluar un sistema informático es encontrar el o los factores que impiden un funcionamiento adecuado, como pueden ser los cuellos de botella.

Las actividades que forman parte del estudio del comportamiento de un sistema se denomina **evaluación de sus prestaciones** y son el criterio clave en el diseño, selección, compra y uso de un sistema informático.

# TÉCNICAS DE EVALUACIÓN DE UN SISTEMA INFORMÁTICO

Las técnicas más habituales usadas para evaluar un sistema son:

- a) **La medición.**
- b) **El modelado.**
- c) **La simulación.**
- d) **El benchmarking.**

**a) Medición (monitorización):** Consiste en tomar medidas directamente sobre el sistema en el que uno está interesado, usando también la carga adecuada, o bien una parte de la misma, que es lo que se suele denominar, en general, carga sintética.

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. Puede 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.

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.

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. Dado que la toma de medidas por un monitor software (el monitor hardware produce menos sobrecarga) implicará la ejecución de un programa, y para ello se usará el procesador del computador, el simple hecho de monitorizar implicará la perturbación del sistema sobre el que se realiza la medida. El grado de distorsión en la medida suele estimarse mediante una variable denominada sobrecarga del monitor (overhead), que viene definida de la siguiente manera:

$$\text{Sobrecarga} = \text{tiempo de ejecución del monitor} / \text{intervalo de medida}$$

**b) Modelado:** Cuando se trata de evaluar un sistema incompleto, o que no se ha construido aún, hace falta construir un modelo analítico del mismo, es decir, usando fórmulas y ecuaciones diferenciales, tratar de hallar a partir de los valores conocidos o estimados de ciertos parámetros, los valores de los que nos van a interesar.

En general se fundamenta 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 métodos analíticos que proporcionan las teorías de colas y de redes de colas, o por medio de la simulación.

**c) 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 formulas cerradas o algoritmos aproximados de las ecuaciones matemáticas que representan el equilibrio que existe entre los eventos o transiciones de estados discontinuas que se producen en el sistema.

La limitación de los métodos analíticos es su incapacidad para tratar de determinadas estructuras y comportamientos de colas que existen en los sistemas informáticos. Los métodos de simulación no tienen estas limitaciones, pero, en general, son muchos más caros que los analíticos tanto en tiempo de cálculo con el 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.

Para **simular** el sistema se puede usar algún lenguaje de simulación, como el SIMULA, o cualquier otro lenguaje orientado a objetos con las herramientas gráficas adecuadas.

Generalmente se usa simulación antes de construir un sistema, especialmente cuando se construyen nuevos microprocesadores, y se basa su estudio en las versiones anteriores de los microprocesadores. Para ello se suelen usar máquinas masivamente paralelas o superordenadores.

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

En cuanto al *tiempo* que se tarda en obtener resultados, lo más rápido es usar un modelo analítico: simplemente se aplican ecuaciones; las mediciones tardan un poco más (sobre todo, teniendo en cuenta la variabilidad de las cargas de trabajo durante el tiempo); por último, la simulación es lo más lento, pues hay que hacer un programa y evaluar los resultados.

En cuanto a la *exactitud*, por supuesto mediciones sobre el propio sistema es lo más exacto, seguido por la simulación, ya que en ella se ponen casi todos los elementos del sistema real, y por último, el modelo analítico, porque requiere gran cantidad de suposiciones. También habría que tener en cuenta el *coste* (normalmente

la medición es bastante cara), y, por supuesto, lo *vendibles* que son los resultados (en este caso, lo mejor son mediciones).

## MEDIDAS DEL RENDIMIENTO

De entre todas las magnitudes medibles de un sistema informático susceptible de ser utilizadas como índices de prestaciones, el tiempo en llevar a cabo una actividad determinada representa la más intuitiva y, desde el punto de vista de la manipulación matemática, la menos susceptible de incorporar falsedad.

Ha sido ampliamente aceptado que el tiempo de ejecución de un programa representa la medida exacta del rendimiento de un computador: aquel que ejecute la misma cantidad de trabajo en el menor tiempo posible será el más rápido.

Pero también existen otros índices de rendimiento que, aun presentando algunos inconvenientes, se siguen utilizando, con mayor o menor acierto, en determinadas áreas de la informática. Nos referimos a los índices MIPS, MFLOPS, MHz y CPI, todos ellos empleados excesivamente por los diseñadores de procesadores.

Hasta hace poco, el procesador se ha venido considerando como la unidad funcional que representa de manera esencial el rendimiento del sistema completo. Tanto es así que, se suele equiparar el rendimiento del procesador con el rendimiento del computador en su conjunto. De hecho, la mayoría de los índices clásicos de prestaciones tienen como objetivo la medida de rendimiento del procesador.

Por ejemplo:

- El índice MIPS (millón de instrucciones por segundo) para un programa determinado se define como:

$$\frac{\text{Número de instrucciones}}{\text{Tiempo de ejecución} * 10^6}$$

Aunque los MIPS han sido empleados tradicionalmente como índice para comparar prestaciones de computadores con un mismo juego de instrucciones, presenta varios e importantes inconvenientes: no tiene en cuenta el tipo de instrucciones, puede variar según el programa ejecutado en un mismo computador. Más importante aún, el que un computador obtenga más MIPS que otro no implica necesariamente que proporcione un tiempo de ejecución menor.

- Otro de los índices clásicos empleado para medir el rendimiento del procesador en el tratamiento de números en coma flotante, especialmente en entornos de grandes computadores, es el MFLOPS (millón de operaciones de coma flotante por segundo), que computa el rendimiento en aritmética de coma flotante y se define como:

$$\frac{\text{Número de operaciones de coma flotante}}{\text{Tiempo de ejecución} * 10^6}$$

El problema que plantea es que el conjunto de operaciones de coma flotante suele variar de una arquitectura de procesador a otra. Por otro lado, no es lo mismo, hacer una suma que una división en coma flotante. Estos problemas se resuelven parcialmente con el establecimiento de los denominados MFLOPS normalizados, que permiten ponderar de una manera más justa la complejidad de cada operación en coma flotante.

Una métrica muy empleada en el ámbito comercial consiste en promocionar los computadores a partir de la frecuencia de reloj del procesador. Esta estrategia consigue captar la atención del comprador ingenuo porque resulta muy fácil asociar la idea de mayor frecuencia con la de menor tiempo de ejecución. Pero esta asociación resulta errónea porque deja de lado una gran cantidad de detalles tecnológicos y arquitectónicos tanto del procesador como del computador en su conjunto.

La mayoría de los textos académicos actuales que tratan sobre el diseño del procesador tienen en cuenta varios factores que intervienen en el tiempo de ejecución de un programa. Este tiempo se expresa en función del número  $I$  de instrucciones ejecutadas, del número medio de ciclos de reloj necesarios para ejecutar una instrucción (CPI, ciclos de reloj por instrucción), y del tiempo del ciclo de reloj  $T_c$ :

$$\text{Tiempo de ejecución} = I * CPI * T_c$$

La variable  $I$  depende tanto del repertorio de instrucciones como del compilador empleado en traducir el programa a lenguaje máquina. La variable CPI es un valor promedio y depende de la implementación que se haga del juego de instrucciones. Y el tiempo de ciclo o periodo de reloj  $T_c$  depende de la frecuencia de reloj a la que funciona el procesador.

## ESTRATEGIA DE COMPARACION: RATIOS

El uso de ratios como índices de prestaciones ofrece un amplio abanico de posibilidades para comparar rendimientos. Un ratio consiste en un numerador y un denominador. Muchos de los índices utilizados en el ámbito informático son ratios (por ejemplo, CPI o MFLOPS) o bien se basan en el uso previo de éstos. Sin embargo, las comparaciones llevadas a cabo utilizando ratios pueden resultar confusas.

Supongamos que queremos comparar el rendimiento de dos computadores, A y B. Si por ejemplo, los tiempos de ejecución de A son siempre menores que los de B, el uso de ratios no puede ayudar a proclamar que B es más rápido que A. Sin embargo, si algunos programas se ejecutan más rápidamente en un computador que en otro, y viceversa; el ratio que define la normalización de tiempos puede ayudar a decantar la balanza para un lado.

Supongamos que tenemos los siguientes tiempos de ejecución:

<u>Programa</u>	<u>A</u>	<u>B</u>
1	50	100
2	100	150
Suma	150	250
Media aritmética	75	125

Atendiendo a la suma de los dos tiempos de ejecución, la máquina A resulta  $250/150 = 1,67$  veces más rápida que B.

A igual conclusión llegamos si comparamos las medias aritméticas:  $125/75 = 1,67$ . Pero, podemos usar ratios y alterar el resultado de la comparación normalizando los tiempos anteriores y tomando como máquina de referencia uno de los dos computadores. La normalización nos ofrece los siguientes valores: Atendiendo a la suma de los dos tiempos de ejecución, la máquina A resulta  $250/150 = 1,67$  veces más rápida que B.

<u>Programa</u>	<u>A (A)</u>	<u>B (B)</u>	<u>A (B)</u>	<u>B (B)</u>
1	1,00	2,00	0,50	1,00
2	1,00	1,50	0,67	1,00
Suma	2,00	3,50	1,17	2,00
Media aritmética	1,00	1,75	0,58	1,00

Por ejemplo, los dos primeros valores de la columna B (A) se obtienen tras dividir los tiempos de ejecución de B entre los tiempos de ejecución en A:  $100/50 = 2,00$  y  $150/100 = 1,50$ .

De igual forma, la columna A (B) se obtiene dividiendo los tiempos de ejecución en A entre los tiempos correspondientes en B:  $50/100 = 0,50$  y  $100/150 = 0,67$ .

Tomemos, por ejemplo, la media aritmética de los tiempos de ejecución normalizados como índice de rendimiento. Si se emplea la máquina A como referencia se concluye que ésta es  $1,75/1,00 = 1,75$  veces más rápida que B, mientras que si se usa B como referencia entonces la mejora es ligeramente menor:  $1,00/0,58 = 1,72$ . Obviamente, los diseñadores de la máquina A optarán por la primera opción a la hora de comparar.

Este ejemplo pone de manifiesto que los datos de partida pueden manipularse de diferentes maneras para obtener conclusiones más favorables. En cualquier caso, hay que tener en cuenta que promediar ratios no es una estrategia válida para comparar rendimientos, y menos aún cuando se emplea la media aritmética y una de las máquinas que se compara actúa como referencia en el proceso de normalización.

## **MAGNITUDES QUE SE DEBEN MEDIR**

Para evaluar tenemos que hacer referencia a medidas cuantitativas o a parámetros que van a caracterizar el comportamiento del hard y soft 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 están relacionadas con tres tipos de medida:

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

Se expone una lista de magnitudes que normalmente se acostumbra medir.

### **Variables externas o perceptibles por el usuario:**

**a) Productividad (throughput):** Cantidad de trabajo útil ejecutado por unidad de tiempo en un entorno de carga determinado.

**b) Capacidad:** 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.

**d) Eficiencia** es la tasa del throughput máximo al throughput que se consigue de forma efectiva.

**e) Ancho de banda:** bits por segundo que es capaz de procesar el sistema. No confundir con baudios.

### **Variables internas o del sistema:**

**a) Factor de utilización de un componente:** Es el porcentaje de tiempo durante el cual el 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.

**h) Factores relacionados con la memoria caché de CPU:** similares a lo dicho con la memoria virtual, y aparte veces que se vacían los buffers TLB, por ejemplo.

**i) Otros subsistemas:** red, gráficos (que tienen mucha importancia últimamente).

## **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 que el sistema trabajará correctamente a lo largo del intervalo de tiempo  $[t_0, t]$ . 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 factibilidad 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 periodo de tiempo determinado.

## CARACTERIZACION DE LA CARGA

Se denomina carga a todas las demandas que realizan los usuarios de un sistema informático (programa, datos, órdenes, etc.) durante un intervalo de tiempo.

Para realizar las mediciones para evaluar las prestaciones de un ordenador es necesario someter al sistema informático a una carga determinada. El resultado de la medición es función de la carga bajo la cual fue determinado.

Aspectos que debemos considerar:

1. Así, cuando se da un índice de prestación, debe especificarse la carga bajo la cual se obtuvo. Esto implica que la comparación entre sistemas sólo tiene sentido cuando las cargas son las mismas. Caso contrario, la diferencia de los índices reflejarán, también, las diferencias de las cargas.
2. Otro aspecto a tener en cuenta es la variación de la carga en el tiempo que dificulta su modelación.
3. Otro aspecto es la interacción de la carga con el propio sistema en el que se está evaluando, estableciéndose distintas realimentaciones (interna y externa).

Una carga está correctamente caracterizada si, y solo si, su resultado es un conjunto de parámetros cuantitativos relacionados de acuerdo con los objetivos de la operación de caracterización.

El uso del modelo de carga que se va a caracterizar es el primer paso de cualquier estudio de evaluación del comportamiento. Una vez establecidos los objetivos, deben determinarse las herramientas que manipularán (modelos analíticos, simulación, sistema real, etc.) el modelo de la carga.

Algunos de los parámetros que se utilizaran en la caracterización de la carga son los siguientes:

- Tiempo total de la CPU.
- Número de operaciones de E/S.
- Espacio de la memoria.
- Ficheros en disco.
- Cintas leídas.

**Carga de prueba:** Independientemente de la forma como se represente la carga de prueba (modelo), también llamada Test, se define como la carga que debe procesar un sistema mientras se está realizando un experimento de medición. Uno de los principales problemas de la carga es que no es fácilmente reproducible. Aún cuando se ejecutan los mismos programas en el mismo sistema. Esto nos lleva a concluir que la repetitividad o la reproductibilidad es una característica importante de las cargas de prueba.

La carga que procesa el sistema (durante la evaluación) puede ser la carga real o un modelo de la carga.

En la siguiente tabla se resume los principales tipos:

Carga de prueba	Real		
	Sintética	Natural	
		Híbrida	
	Artificial	Ejecutables	Mix
			Kernels
			Programa Sintético
			Secuencia Transaccionales
			Secuencia Conversacionales
No Ejecutables		Distribuciones Estadísticas	
	Trazas por Simulación		

**Carga Real:** Es la más representativa y la más barata de implementar. La única elección que debe realizar el encargado de modelar la carga es decidir qué porción de tiempo tendrá la ejecución de la carga.

Algunas de las razones que limitan su uso: falta de flexibilidad, necesidad de utilizar los datos originales, diferencia de hard y soft de los distintos sistemas, poco reproducibles, etc.

### **Carga Sintética**

***Naturales:*** Cuando tienen un conjunto de programa extraídos de la carga real. Son programas extraídos de la carga real pero no es carga real porque los usuarios no los están utilizando para realizar trabajo útil.

***Híbridas:*** La carga que se pretende modelar no existe completamente. Para ello es necesario adoptar soluciones híbridas que se componen de dos partes: 1) representar la carga conocida por un conjunto de programas extraídos de ella y, 2) la carga no existente representarla mediante algunos elementos artificiales (kernels, programas sintéticos, etc.).

### **Carga Artificial:**

***Ejecutables:*** Un modelo de este tipo consta de componentes diseñados con el propósito de cargar el sistema real o un modelo matemático de él. O dicho de otra forma, ningún componente de la carga real del sistema forma parte de una carga de test artificial. Pueden ser: mix, kernels, programas sintéticos, y demás.

### ***No Ejecutables:***

***(Distribuciones Estadísticas):*** En una carga de test probabilística, los parámetros se consideran usualmente como variables aleatorias independientes cuyas distribuciones pueden estimarse a partir de las distribuciones medidas en la carga real, o aproximarse mediante funciones analíticas.

***(Trazas por Simulación):*** Consiste en una secuencia cronológica, registrada en un soporte adecuado y representando la información referente a determinados tipos de acontecimientos que han accedido en un sistema durante una sesión de medida.

# **REPRESENTATIVIDAD DE UN MODELO DE CARGA**

Es la precisión con que un modelo representa una carga. La carga puede representarse a distintos niveles: nivel físico, nivel virtual, nivel funcional.

El modelo deberá ser representativo atendiendo al nivel al que está asignado.

## ***\*Representatividad a nivel físico***

- Los modelos se caracterizan por el consumo de recursos físicos.
- Orientado al consumo de recursos físicos.
- Muy dependiente del sistema.
- Datos para el modelo fáciles de conseguir (S .O., monitores, hardware).

Las mayores aplicaciones de estos modelos son:

- Estudio de sintonización.
- Planificación de capacidad residual.
- Diseño.

## ***\*Representatividad a nivel virtual***

- Consumos de recursos lógicos.
- Menor dependencia del sistema que en el modelo anterior.
- Más cercanos al usuario.
- Parámetros más difíciles de obtener.
- Se utilizan en estudios de ampliación.
- Ej. de componentes: sentencias de lenguaje de alto nivel, número de ficheros utilizados, órdenes interactivas, espacio de memoria virtual.

## ***\*Representatividad a nivel funcional***

- Se trata de modelos que realizan las mismas aplicaciones que la carga a la que representan.
- Son bastante independientes del sistema.
- Es difícil diseñarlo de forma sistemática. A veces son bastante subjetivos.
- Aplicaciones: selección de equipos, planificación de la capacidad.

## **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 existente, 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 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 lo 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 características las exponemos a continuación:

### **A) Para cada componente de la carga**

**A.1) Tiempo de la 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.

**A.2) 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.

**A.3) 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 en el archivo sobre el que se efectúan. Todo ello tiene una influencia directa en el tiempo necesario para realizar una operación de E/S.

**A.4) Prioridad:** Es la que el usuario asigna a cada uno de los trabajos que somete al sistema.

**A.5) Memoria:** Es la que requiere ocupar, para su ejecución un trabajo determinado. Puede ser constante (memoria real) o variable (memoria virtual paginada o segmentada), según la gestión que el sistema operativo haga de la memoria.

**A.6) Localidades 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 este 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.

**B)** Para el conjunto de la carga

**B.1)** Tiempo entre llegadas: Es el tiempo entre dos requerimientos sucesivos para un servicio (ejecución de un trabajo o transacción) del sistema.

**B.2)** 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.

**B.3)** Distribución de trabajos: Define la proporción existente entre las ejecuciones de los distintos trabajos que constituyen la carga.

**C)** Para cargas conversacionales

**C.1)** 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).

**C.2)** Número de usuarios simultáneos: Es el número de usuarios interactivos que trabajan simultáneamente sobre el mismo sistema en un instante dado.

**C.3)** Intensidad del usuario: Es la relación entre el tiempo de respuesta de una petición y el tiempo de reflexión del usuario.

## CONCEPTO DE BENCHMARK

Puesto que cada usuario tiene unas necesidades diferentes, es decir, va a ejecutar sobre la máquina diferentes tipos de aplicaciones con diferentes requerimientos, necesitamos obtener una métrica fiable del rendimiento de la máquina. Para ello surgen los conjuntos de programas de evaluación o benchmarks.

Un Benchmark, es un programa que mide las prestaciones de un ordenador, o de una parte del mismo. Estos programas no sólo son capaces de evaluar las prestaciones de un equipo con diferentes configuraciones de Software y Hardware sino que además pueden ayudarnos en la *comparación de diferentes sistemas*. Los benchmarks son pruebas para medir el rendimiento y poder verificar que el hardware funciona de forma óptima o para comparar distintas configuraciones. En general, se trata de programas que se instalan de la misma forma que una aplicación clásica, pero que no le permiten trabajar. En la mayoría de los casos, el benchmark inicia una aplicación o conjunto de aplicaciones (ofimática, cálculo científico, etc.) y mide el tiempo necesario para ejecutar una tarea. La evaluación de la potencia del procesador sirve para tener una idea de la capacidad de cálculo bruto del ordenador.

Cada test Benchmark realiza un trabajo diferente. Así, algunos de estos nos indican lo rápido que es un ordenador generando documentos, otros indican lo veloz que es en los gráficos y rellenos de pantalla, otros determinan la velocidad en operaciones matemáticas, etc. Algunos hacen una mezcla de todos estos test. Para obtener resultados que nos sean útiles deberemos utilizar Benchmarks que reflejen el uso que le daremos al equipo.

Hay cuatro categorías generales de pruebas de comparación:

- *Pruebas aplicaciones-base* (**application-based**) las ejecuta y las cronometra.
- *Pruebas play back* (**play back test**), las cuales usan llamadas al sistema durante actividades específicas de una aplicación (ej.: llamados a gráficos o uso del disco) y las ejecuta aisladamente.
- *Prueba sintética* (**synthetic test**), la cual enlaza actividades de la aplicación en subsistemas específicos.
- *Prueba de inspección* (**inspection tests**), la cual no intenta imitar la actividad de la aplicación, sino que las ejecuta directamente en los subsistemas específicos.

Los “test de aplicaciones base” entregan la mejor forma de medir el rendimiento completo del sistema en el mundo real. El programa **Winstone** de Zdnet, ejecuta más de una docena de las aplicaciones más populares en el ambiente Windows, es un ejemplo de este tipo de comparadores.

Donde sea factible la tecnología “play back” le da la manera más real de medir subsistemas individuales en aislación. El programa **WinBench** de ZDnet utiliza la tecnología play back para probar gráficos, CD-ROM y subsistemas de disco duro, también corre cientos de otras pruebas en áreas específicas del computador.

Los “test sintéticos” continúan en el estado de medición del rendimiento es por eso que winbench usa las pruebas de procesadores.

Los “test de inspección” tienen su lugar verificando el comportamiento libre de fallas y midiendo rendimiento operación por operación, por esto se incluye el test de inspección en el winbench.

## TIPOS DE BENCHMARK

- **Aplicaciones:** miden prestaciones utilizando aplicaciones reales y los segundos, utilizan programas específicos para el testeo de prestaciones.

Ventajas: Utiliza aplicaciones reales como Word, Paradox, etc.  
El uso del ordenador está muy bien reflejado.

Desventajas: Normalmente son largos y difíciles de ejecutar.  
No se pueden emplear para medir futuras necesidades.  
Pueden estar basadas en software ya obsoleto.

- **Sintéticos:** tratan de simular funciones de los programas en un entorno típico de uso. La utilidad de los mismos dependerá directamente de la calidad de su implementación.

Ventajas: Su distribución es más sencilla.  
Son más rápidos y sencillos de ejecutar que los Benchmarks de aplicaciones.

Desventajas: Gran variación en la calidad de implementación.  
Utilización del rendimiento para elegir el sistema correcto.

- **Mezcla de Instrucciones:** es una especificación de varios tipos de instrucciones con su frecuencia de uso. Con el tiempo para cada tipo de instrucción podemos calcular el tiempo medio para una mezcla de instrucciones dada y comparar distintos procesadores. La más famosa es la mezcla de Gibson.

Desventajas: Los ordenadores actuales presentan tipos de instrucciones mucho más complejos.

El tiempo empleado para ejecutar una instrucción es variable, dependiendo de varios factores: cache, modo de direccionamiento.

No reflejan las optimizaciones que tienen los procesadores en la ejecución de instrucciones.

Proporciona un solo valor que permite hacer comparaciones.

La mezcla de instrucciones sólo mide la velocidad del procesador, esto no siempre refleja el rendimiento global del sistema.

- **Núcleos (Kernels):** La introducción de sistemas de optimización de ejecución de instrucciones hace que el tiempo de instrucción sea muy variable. Por lo tanto no podemos considerar las instrucciones de forma aislada.

Debemos considerar conjuntos de instrucciones que formen una función de alto nivel, *un servicio* que proporcione un procesador. Existe lista de las funciones más frecuentes y estas se utilizan como carga de trabajo, reciben el nombre de funciones Kernel. Una operación típica de un Kernel sería la inversión de una matriz.

Desventaja: No suelen hacer uso de los dispositivos de entrada / salida y por lo tanto no refleja el rendimiento global del sistema.

## **NIVELES DE BENCHMARK**

Existen dos niveles de Benchmark, pueden ser:

- **Componentes:**

Evalúan únicamente partes específicas de un ordenador, como por ejemplo el procesador, el disco duro, la tarjeta gráfica, etc. Serán por tanto útiles a la hora de seleccionar componentes específicos para un determinado sistema.

- **Sistemas:**

Evalúan el rendimiento global del sistema, miden las prestaciones del procesador, memoria, vídeo, disco duro, etc., trabajando conjuntamente en el ordenador. Este tipo de pruebas permiten comparar sistemas diferentes. Responde a cuestiones como ¿este equipo X es más rápido que el equipo Y? o ¿puede mi aplicación ejecutarse más rápidamente en el caso de aumentar la velocidad del procesador, o está limitada por otros subsistemas?. Por tanto, estos Benchmark miden prestaciones globales del sistema.

# MAGNITUDES PARA CONTROLAR EL COMPORTAMIENTO

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:

## **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 si 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 interrumpido de la CPU que un sistema de tiempo compartido asigna a diferentes trabajos. En ciertos sistemas, no existe uno sino varios cuanta para las distintas prioridades internas de los distintos trabajos por lo que no es solo un 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 overhead 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 reciba 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 del sistema. Cuanto mayor sea este valor tanto mejor aprovechamiento se puede tener de todos los recursos del sistema, aunque también aumentara 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 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.

## **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, aquellas 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.

## **3) Equilibrado de la distribución de las cargas**

El ideal, inalcanzable, de utilizar por igual todos los dispositivos del sistema informático debe sustituirse por el de utilizarlos de la formas 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 acostumbrada, en muchos casos, a proporcionar mejoras espectaculares en el comportamiento del sistema.

## **4) Modificación o sustitución de componentes hardware 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 el 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 de comportamiento) que se pueda haber detectado, pues, de lo contrario, el comportamiento conjunto del sistema ampliado no varía de forma significativa.

## **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 validas.

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

## CONCLUSIÓN

El rendimiento o desempeño de un sistema informático, expresa la manera o la eficiencia con la que un sistema de computación cumple con sus metas.

Como bien sabemos es de relativa importancia la necesidad de llevar a cabo la evaluación de un sistema informático, debido a que es conveniente ver en qué medida el sistema en sí brinda las prestaciones, si bien ya sean para mejorarlas o bien para compararlas con otros sistemas informáticos.

Las técnicas más utilizadas para evaluar el comportamiento de los sistemas son: la medición, el modelado, y la simulación. El problema en cada caso es qué técnica usar. La mayoría de las veces se recurre a la *medición*: las herramientas existen ya, y sólo hay que aplicarlas a nuestro sistema; sin embargo, si el sistema no existe, la única forma de medir sus prestaciones es mediante *simulación* y *modelización*. Y que el *benchmarking* es una técnica muy utilizada para efectuar las comparaciones con otros sistemas, siempre y cuando se aplique la misma carga para que la comparación tenga sentido, lo que a su vez resulta difícil de llevar a cabo debido a las complejidades propias que presenta la carga.

Una vez llevado a cabo la evaluación de un sistema informático, existen ciertas técnicas como: ajustes de parámetros del sistema operativo o del hardware, un equilibrio adecuado de las cargas, modificación del software, entre otras, y que no son más que medidas que sirven para sintonizar el sistema (mejorar sus prestaciones).

Además debe quedar en claro de que al comparar el rendimiento de sistemas informáticos, el tiempo de ejecución de un programa, o un conjunto de programas es la medida más confiable y la más intuitiva y, desde el punto de vista de la manipulación matemática, la menos susceptible de incorporar falsedad.

A todo lo dicho anteriormente, no se deben dejar al margen cuestiones referidas sobre la carga, es decir, las demandas que realizan los distintos usuarios del sistema por algún servicio o programa. Ya que todo sistema está ligado y estrechamente relacionado a una carga, la cual en el transcurso del tiempo fluctúa y/o varía, convirtiéndose así en su mayor inconveniente de llevar a cabo la evaluación del rendimiento para su futura comparación con otros sistemas.

## **BIBLIOGRAFIA**

- *Sistemas Operativos* - (La Red Martínez David).
- *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos* - (Molero Xavier, Juiz Carlos, Rodeño Miguel).
- *Evaluación de Desempeño de Sistemas Informáticos* – (Curiel Mariela).
- *Evaluación y Explotación de los Sistemas Informáticos* – (Puigjaner Ramón, Serrano Juan José, Rubio Alicia).