

UNIVERSIDAD AMERICANA

---



## Algoritmos y Estructuras de Datos

---

Eficiencia y eficacia, parámetros que rigen la eficiencia de un programa

---

**Estudiante:**

Gabriel Andrés Juárez Cuevas

**Docente:**

Carlos Argüello

Managua, Nicaragua

19/03/2025

## Eficacia y eficiencia de un programa

La eficiencia y la eficacia tienen en común que las dos buscan cumplir un objetivo o llegar a una meta, la diferencia entre estas dos radica en que la eficacia sólo busca hacer las cosas de manera correcta, mientras que la eficiencia no solo busca eso, sino que busca cumplir con los objetivos optimizando el uso de los recursos disponibles y minimizando los costos.

En informática, la eficiencia de un programa es un factor clave que influye en su rendimiento y usabilidad. En este ámbito los recursos que se buscan optimizar son los del sistema (memoria, tiempo de ejecución, procesamiento, etc.).

La eficacia a como se aclaró anteriormente busca principalmente cumplir con el objetivo, no se centra en la optimización, y para que un programa sea eficaz tiene que cumplir con los siguientes parámetros:

**Tasa de éxito en la realización de tareas:** Mide el porcentaje de tareas que los usuarios pueden completar exitosamente utilizando el software.

**Tasa de errores del usuario:** Evalúa la frecuencia con la que los usuarios cometen errores al interactuar con el software.

**Satisfacción del usuario:** Se refiere al grado de satisfacción de los usuarios al utilizar el software, generalmente medido a través de encuestas o retroalimentación directa.

**Cobertura de código:** Representa el porcentaje del código fuente que ha sido probado mediante pruebas automatizadas.

**Porcentaje de detección de defectos:** Calcula la proporción de defectos identificados antes del lanzamiento del software en comparación con aquellos descubiertos después.

Ya que estamos hablando de eficacia, hay que aclarar que este enfoque solo busca en dar una solución al problema del cliente, que las tareas del software se realicen y que todo trabaje de manera correcta, sin abordar la optimización de recursos y rendimiento del sistema. Por lo tanto se podría decir que el programa es eficaz, pero este podría ser no eficiente debido a un mal uso de recursos o rendimiento subóptimo independientemente de que este cumpla con su objetivo de la manera correcta.

Si bien la eficacia garantiza que el software cumple con su propósito, la eficiencia va un paso más allá al evaluar cómo se utilizan los recursos para lograr ese objetivo. Para medir la eficiencia, se emplean métricas que nos ayudan a saber que tan eficiente es nuestro programa, a continuación se abordarán algunas de estas.

### **Métricas para evaluar la eficiencia de un programa informático:**

1. **Uso de la CPU:** Mide el porcentaje de tiempo que el procesador dedica a ejecutar el programa. Un uso elevado puede indicar que el programa consume muchos recursos, afectando la eficiencia del sistema.
2. **Uso de memoria:** Evalúa la cantidad de memoria RAM que el programa utiliza durante su ejecución. Un consumo excesivo puede llevar a problemas de rendimiento, especialmente en sistemas con recursos limitados.
3. **Tiempo de respuesta:** Mide el tiempo que tarda el programa en responder a una solicitud o completar una tarea específica. Tiempos de respuesta más cortos suelen asociarse con una mayor eficiencia y mejor experiencia del usuario.
4. **Tasa de transferencia de datos:** Calcula la cantidad de datos que el programa puede procesar o transferir en un período determinado. Una mayor tasa indica una mayor eficiencia en la gestión de datos.
5. **Tiempo promedio para corregir defectos:** Mide el tiempo que tarda el equipo de desarrollo en identificar y solucionar errores o defectos en el programa. Un tiempo reducido refleja una mayor eficiencia en el proceso de mantenimiento y mejora del software.
6. **Rotación total de códigos:** Es la cantidad de líneas de código agregadas, modificadas o eliminadas de una base de código durante un periodo de tiempo. Cambiar código con demasiada frecuencia puede indicar problemas de calidad, debido a que cuantos más cambios mayor será la probabilidad de introducir errores en los productos.
7. **Disponibilidad del sistema:** Evalúa la cantidad de tiempo que un sistema está en funcionamiento. Compara el tiempo de actividad del sistema con su equipo de inactividad durante un periodo de tiempo. Un sistema que está disponible es tan rápido como se espera y no falla a menos que ocurran condiciones imprevistas.
8. **Densidad de defectos:** Es la cantidad de defectos en un producto de software en comparación con su tamaño. Este es un número relativo que ayuda a evaluar la calidad del código.
9. **Tiempo medio entre fallas (MTBF):** Es el tiempo promedio entre dos fallas del sistema. Se trata de errores que se encuentran tras el lanzamiento del producto y que se deben, por ejemplo, a un defecto no detectado.

10. **Tiempo medio de resolución (MTTR):** Es el tiempo promedio que le toma a tu equipo resolver un error en un producto de software después de que alguien lo descubre.
11. **Satisfacción del cliente (CSAT):** Es un número que representa la forma en que los clientes experimentan tu producto de software. Se obtiene recopilando y analizando datos de las encuestas de satisfacción de los clientes sobre el funcionamiento del software.
12. **Tiempo medido para remediar una vulnerabilidad:** Es el tiempo promedio que le toma a tu equipo reparar las vulnerabilidades de ciberseguridad en tu producto de software.
13. **Cobertura de código:** Evalúa la cantidad de código fuente de tu producto para el cual existe una prueba unitaria.

Para que nuestros programas no solo sean eficaces y también sean eficientes, existen buenas prácticas que deberíamos implementar si lo que buscamos es la optimización de recursos y un mejor rendimiento.

1. **Hacer uso del código modular y reutilizable.**
2. **Elección de algoritmos y estructuras de datos adecuados.**
3. **Evitar operaciones innecesarias.**
4. **Pruebas de rendimiento.**
5. **Monitoreo continuo y mantenimiento del código.**

## **Bibliografía:**

Computer Weekly. (s.f.). *10 métricas de rendimiento de aplicaciones y cómo medirlas*. Recuperado de <https://www.computerweekly.com/es/consejo/10-metricas-de-rendimiento-de-aplicaciones-y-como-medirlas>

Testing IT. (s.f.). *Métricas en pruebas de software*. Recuperado de <https://www.testingit.com.mx/blog/metricas-pruebas-de-software>

Navarro, A. (2024). *10 Métricas de Calidad del Software para Mejorar la Eficiencia*. Recuperado de <https://innevo.com/blog/metricas-de-calidad-del-software#que-son>

Eficiencia y eficacia de un programa, investigar bajo qué parámetros se dice que tan eficaz es un programa, investigar cuales son las métricas para decir que tan eficiente es un programa