

**Eficiencia** Esto está relacionado con la velocidad de ejecución y la utilización eficiente de la memoria, o sea, con la búsqueda del punto óptimo de la relación Tiempo/Espacio. Este es uno de los objetivos importantes, aunque no se debe conseguir a expensas de la pérdida de la claridad y la sencillez. Muchos programas complicados conducen a un enfrentamiento de estas características. En estos casos es necesario recurrir a la experiencia y al sentido común. Al programar deben utilizarse las instrucciones más adecuadas, las estructuras de datos más apropiadas y los métodos de solución más rápidos y directos. Se incluye en este punto la *optimización de código* (reducción del número de variables y eliminación de sentencias redundantes) y la compactación y simplificación de código sin olvidar la claridad y la sencillez.

**Integridad** Se refiere a la exactitud de los cálculos y a la confiabilidad de los resultados. Los programas deben ser lo más exactos posible. Ningún programa tendrá aceptación si los cálculos que efectúa y los resultados que produce no son correctos. Una conversión de tipos inadvertida, el redondeo o truncamiento desafortunado de valiosos decimales, la no validación de los datos de entrada o la no verificación de los resultados calculados pueden hacer que un programa pierda credibilidad, utilidad y méritos. Esta característica es absolutamente necesaria en cualquier programa.

**Flexibilidad** Los programas deben poderse adaptar a nuevas necesidades con tan sólo ligeros cambios. Esto implica diseñar los programas de una forma clara y limpia, imprimirle generalidad, utilizar funciones de bibliotecas, usar constantes e identificadores correctos y una lógica sencilla a lo largo de todo el programa.

**Confiabilidad** Es la probabilidad de operación libre de fallas de un programa de computadora en un entorno determinado y durante un tiempo específico.

El fallo es cualquier no concordancia con los requerimientos del software. Hay distintos grados de fallos, estos pueden ser simplemente desconcertantes o catastróficos.

La confiabilidad del software se encuentra en un etapa de formación de desarrollo y es la característica de rendimiento más costosa de conseguir y difícil de conseguir y de difícil de garantizar. La naturaleza del proyecto ayuda para la formulación de estimaciones de costo y el esfuerzo que asegure la confiabilidad requerida.

Los modelos de confiabilidad del software se usan para caracterizar y predecir el comportamiento importante para directores e ingenieros.

La generación de fallos depende del código desarrollado, tales como tamaño y las características del proceso de desarrollado tales como las tecnologías y herramientas de ingeniería de software usadas.

**Mantenibilidad** La ingeniería de fiabilidad es el estudio de la longevidad y el fallo de los equipos. Para la investigación de las causas por las que los dispositivos envejecen y fallan se aplican principios científicos y matemáticos. El objetivo estriba en que una mayor comprensión de los fallos de los dispositivos ayudará en la identificación de las mejoras que pueden introducirse en los diseños de los productos para aumentar su vida o por lo menos para limitar las consecuencias adversas de los fallos. Por tanto, se concede mucha importancia al diseño de los productos o a su rediseño, con anterioridad a la fabricación o a la venta.

**Portabilidad** La **portabilidad** de un software se define como su dependencia de la plataforma en la que corre. La portabilidad es mayor cuanto menor es su dependencia del software de plataforma.

Si un software puede ser compilado en plataformas diversas (x86, IA64, amd64, etc.), dicho software se dice que es multiplataforma.

En algunos casos el software es "independiente" de la plataforma y puede ejecutarse en plataformas diversas sin necesidad de ser compilado específicamente para cada una de ellas, a este tipo de software se le llama interpretado, porque necesita un intérprete para ser ejecutado en las diferentes plataformas.

**Usabilidad** La usabilidad es la cualidad que tiene un sistema por la que permite a sus usuarios alcanzar objetivos específicos (como escribir una carta o enviar un mensaje SMS) con efectividad, eficiencia y satisfacción. Es decir, que cuanto mejor permita hacer algo un sistema mayor usabilidad tendrá. Si el sistema ayuda a que el usuario cometa los menos errores o se recupera de ellos fácilmente, si permite hacer la tarea lo más rápidamente posible y además el usuario queda satisfecho con la labor realizada, el sistema tiene una buena usabilidad.

El concepto en torno al cual gravita la usabilidad es la calidad de uso. No se trata de "pelearse" con el ordenador para conseguir que haga lo que uno quiere sino todo lo contrario; el objetivo es que el sistema esté diseñado como una herramienta que encaja con la forma de trabajar del usuario y le permite realizar aquello que éste pretende hacer lo mejor posible.

**Seguridad** La seguridad de software aplica los principios de la seguridad de información al desarrollo de software. **Information security** (La seguridad de información) se refiere a la seguridad de información comúnmente como la protección de sistemas de información contra el acceso desautorizado o la modificación de información, si esta en una fase de almacenamiento, procesamiento o tránsito. También la protege contra la negación de servicios a usuarios desautorizados y la provisión de servicio a usuarios desautorizados, incluyendo las medidas necesarias para detectar, documentar, y contrarrestar tales amenazas.

Muchas preguntas con respecto a la seguridad, son relacionadas al ciclo vital de software. En particular, la seguridad del código y el proceso de software; deben de ser considerados durante la fase del diseño y desarrollo. Además, la seguridad debe de ser preservada durante la operación y el mantenimiento para asegurar la integridad de una parte (pedazo) de software.

**Corrección** Poder garantizar la corrección del software que construimos es una tarea deseable.

En algunas aplicaciones, es, sin duda, crucial:

- Software para equipamiento médico
- Software para el control de vehículos
- Software para el control de armamento
- (Para algunos, algunas aplicaciones financieras)

A estos sistemas, cuyas fallas pueden ocasionar daños de importancia (pérdida de vidas humanas, grandes pérdidas financieras, catástrofes nucleares, etc.) Se denominan *sistemas críticos*.