

## CARRERA DE COMPUTACIÓN

### 1. Datos Informativos

- 1.1. **Módulo:**
- 1.2. **Nivel:** 7mo Nivel
- 1.3. **Apellidos y Nombres:** Paguay Ijujes Almer Rodrigo
- 1.4. **Tema:** Normativa 9126
- 1.5. **Fecha:** 23 de octubre de 2025

### 2. Objetivo

- Analizar la norma ISO/IEC 9126 y comprender cómo sus características y métricas permiten evaluar y mejorar la calidad del software, garantizando su fiabilidad, mantenibilidad y satisfacción del usuario final.
- Valorar la aplicación de esta norma en la industria del desarrollo de software como parte esencial de los procesos de evaluación y aseguramiento de la calidad.

### 3. Contenido

#### Normativa 9126

El video comienza con una introducción del expositor (David Fernando Rosero Guerrero) quien anuncia que pertenece al SENA de Pasto, carrera de Ingeniería en Sistemas y que presentará la norma ISO/IEC 9126 junto con una explicación de indicadores de calidad del software. Se dice que el estándar ISO/IEC 9126 es un modelo de calidad de software adaptado del estándar ISO/IEC25000 (o serie ISO 25000) de 2005, que evalúa la "bondad" de un programa en términos de su calidad y usabilidad interna y externa. Caracterización del modelo de calidad. El estándar define seis características principales de calidad de un producto de software, cada una de las cuales tiene sus propias subcaracterísticas.

La misma lista se puede ver en el vídeo:

- Funcionalidad
- Fiabilidad

- Eficiencia
- Mantenimiento
- Portabilidad

Este esquema es consistente con lo que indican fuentes externas: según ISO/IEC 9126-1, estos seis parámetros están definidos para la calidad interna/externa del software.

### **Funcionalidad:**

Se define como un conjunto de atributos asociados a la existencia de un determinado conjunto de funciones y propiedades: es decir, para qué está destinado el software, si corresponde a lo implícito/directo. Estas subfunciones incluyen:

- Idoneidad (idoneidad del conjunto de funciones para las tareas especificadas).
- Precisión (resultados correctos como se esperaba).
- Interoperabilidad (capacidad de conectarse a otros sistemas).
- Cumplimiento operativo (hace lo que se supone que debe hacer).
- Seguridad (capacidad de impedir el acceso no autorizado, proteger los datos)

La inclusión de "seguridad" en la funcionalidad es interesante porque, si bien el estándar ISO/IEC 9126 original no especificaba la seguridad como una característica separada (luego lo hizo en ISO/IEC 25010), el video la considera una subcaracterísticas de la funcionalidad. Fiabilidad

Se refiere a la capacidad del software para mantener su rendimiento bajo ciertas condiciones durante un período de tiempo determinado.

### **Subfunciones explicadas:**

- Vida útil (cuántos errores tiene el software).
- Resiliencia (restaurar la funcionalidad y los datos después de una falla).
- Cumplimiento (adherencia a normas o estándares de confiabilidad).
- Tolerancia a fallos (el programa continúa ejecutándose incluso si se producen errores)

### **Facilidad de uso**

Relacionado con el esfuerzo del usuario por utilizar el software y la evaluación individual del uso final. Subfunciones:

- Capacidad de aprendizaje (el usuario puede aprender a usarlo).
- Comprensión (el usuario comprende conceptos y aplicaciones lógicas).
- Operabilidad (el usuario opera y controla adecuadamente el software).
- Atractivo (el software es atractivo para el usuario, deseable)

### **Eficiencia**

Se refiere a la ejecución de un programa en relación con los recursos requeridos bajo ciertas condiciones. Dos indicadores principales:

- Tiempo de respuesta, procesamiento, inicio, apagado.
- Uso de recursos (memoria, espacio, hardware)

### **Mantenimiento**

La capacidad del software para modificarse, corregirse, ampliarse o adaptarse a lo largo del tiempo. Subfunciones:

- Estabilidad (no se vuelve inestable debido a efectos inesperados).
- Análisis simple (identificar errores, evaluar debilidades).
- Fácil modificación (adaptación al nuevo entorno).
- Fácilmente verificable (verificabilidad)

### **Portabilidad**

Capacidad de transferir software de un entorno/plataforma a otro.

Subcaracterísticas descritas en el vídeo:

- Fácil instalación (instalación en diferentes plataformas).
- Intercambiabilidad (capacidad de utilizar otro programa que realice la misma función).
- Adaptabilidad (adaptabilidad a diferentes plataformas, versiones, parches).

- Convivencia (capacidad de compartir recursos con otros programas)

### **Calidad en uso**

El ponente añade que además de las seis características anteriores, la norma tiene en cuenta la calidad de uso, entendida como un conjunto de atributos relacionados con la aceptación del usuario final. Cuatro características de dicho punto:

- Eficiencia (cuando el usuario realiza el proceso si coincide con lo diseñado).
- Productividad (rendimiento del usuario en las tareas diarias).
- Seguridad (riesgos para el usuario: pérdida de datos, información sensible).
- Satisfacción (el usuario está satisfecho con el uso del software)

### **Indicadores de calidad del software.**

Finalmente, el vídeo analiza qué son los indicadores de calidad del software: una métrica se define como una medida o conjunto de parámetros que nos permite saber qué tan bueno es el software. Uso de métricas: diseñar un programa, personalizarlo, establecer control sobre el proceso de desarrollo, medir la productividad del desarrollador, comparar la calidad con la productividad, justificar nuevas herramientas o capacitación.

Se explican los tipos de métricas:

- a) Métricas de tamaño (por ejemplo, líneas de código).
- b) Métricas de estructura de control (bucles, variables vivas, operaciones lógicas).
- c) Métricas compuestas (combinación de varias métricas).
- d) Métricas de esfuerzo (tiempo, recursos humanos utilizados).
- e) Métricas de calidad/fiabilidad (las más relevantes pues los programas que perduran son los fiables).
- f) Métricas de diseño (modularidad, iconos, comprensibilidad, usabilidad)

Se enfatiza que menos líneas de código para la misma función pueden indicar mayor optimización/calidad. También la estructura lógica, el control de bucles, optimización de variables, etc., entran en el análisis.

### **Normativa 12119**

La norma ISO/IEC 12119 fue publicada en noviembre de 1994 por la International Organization for Standardization y la International Electrotechnical Commission (ISO/IEC JTC 1/SC 7). Su título completo es *Information technology — Software packages — Quality requirements and testing*. Esta norma se aplica a los paquetes de software entregados (como productos finales) y no aborda el proceso de producción del software por parte del proveedor. Su objetivo es establecer requisitos de calidad para los paquetes de software y los procedimientos de prueba para verificar dichos requisitos.

La norma ha sido retirada (withdrawn) y reemplazada por la estándar ISO/IEC 25051 en 2006.

#### **Alcance y definiciones**

- El alcance de la norma cubre paquetes de software “tal como se ofrecen y entregan”, lo que significa que se dirige al producto final que se entrega al usuario o cliente, y no al ciclo de vida de desarrollo del software.
- Contiene un apartado de definiciones que aclara los términos usados.  
Requisitos de calidad (Quality requirements)

La norma se divide en varias secciones para los requisitos de calidad, distribuidos en: descripción del producto, documentación de usuario, programas y datos.

Algunos de los puntos clave:

#### **1. Descripción del producto**

La norma exige que el paquete de software incluya una “descripción de producto” que abarque:

- Contenidos generales: qué incluye el paquete (programas, datos, documentación).
- Identificaciones e indicaciones: versiones, requisitos mínimos del sistema, entorno de instalación.

- Declaraciones sobre funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Es decir, se espera que el fabricante o proveedor haga afirmaciones formales (statements) sobre estos atributos.

## 2. Documentación de usuario

La norma también exige que la documentación al usuario (manuales, guías) cumpla con ciertos requisitos de calidad:

- Completitud: toda la información necesaria está presente.
- Corrección: sin errores, sin ambigüedades.
- Consistencia: la documentación no debe contradecirse.
- Facilidad de comprensión (understandability) y facilidad de visión general (ease of overview).

## 3. Programas y datos

Además, en la parte del software (programas) y los datos incluidos, se establecen atributos de producto muy similares a los que aparecen en otras normas de calidad de software. Los atributos principales son: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Por ejemplo:

- Funcionalidad: que las funciones declaradas estén presentes, que el software sea apto para las tareas especificadas.
- Fiabilidad: comportamiento bajo condiciones específicas, que las fallas sean limitadas.
- Usabilidad: el sistema debe poder ser usado por los usuarios para las tareas previstas.
- Eficiencia: en cuanto a tiempo de respuesta, uso de recursos.
- Mantenibilidad: facilidad de modificación, corrección.
- Portabilidad: capacidad de trasladarse de un entorno a otro.

Instrucciones para la prueba (Instructions for testing)

Una parte esencial de la norma es la sección dedicada a la prueba, es decir, cómo verificar que el paquete de software cumple con los requisitos de calidad establecidos. Los puntos relevantes incluyen:

- Pre-requisitos de prueba: el ambiente de prueba debe cumplir determinados criterios, debe contar con los ítems del producto, los componentes del sistema, y entrenamiento del personal de prueba.

- Actividades de prueba: incluir la comprobación de la descripción del producto, la documentación de usuario, los programas y los datos.

- Registro de pruebas: la norma recomienda que para cada prueba se registre información suficiente para permitir la repetición, incluyendo plan de prueba, especificaciones, guía de prueba, resultados, fallas observadas.

Importancia para el desarrollo de software y su relevancia

Aunque esta norma fue retirada, sigue teniendo valor histórico y conceptual como antecedente en el ámbito de la calidad de los paquetes de software. Su foco en producto entregado, más que en el proceso de desarrollo, la hace relevante especialmente cuando se evalúan aplicaciones listas para uso comercial, licencias empaquetadas, etc. Además, la norma se relaciona con la anterior ISO/IEC 9126, en cuanto a los atributos de calidad del software (funcionalidad, fiabilidad, etc.).

Limitaciones y contexto

- La norma no aborda el proceso interno de desarrollo del software (cómo se diseñó, qué metodologías se usaron); solamente aborda el producto final entregado.
- Fue retirada y reemplazada por la norma ISO/IEC 25051 (y luego otras evoluciones), por lo que podría no reflejar los últimos avances en calidad de software o integraciones modernas.
- Al estar enfocada en paquetes de software, no necesariamente se aplica directamente a sistemas grandes a medida, aplicaciones web muy dinámicas u otras arquitecturas modernas sin adaptación.

#### 4. Conclusiones

La norma ISO/IEC 9126 constituye un modelo de referencia fundamental para medir la calidad del software, ya que define criterios claros y objetivos que permiten evaluar el desempeño y la eficiencia de los productos informáticos.

En conclusión, la ISO/IEC 9126 sigue siendo una base teórica y práctica valiosa en la ingeniería de software, aunque haya sido actualizada por la ISO/IEC 25010, ya que conserva los principios esenciales de la calidad del producto y del proceso de desarrollo.

## 5. Referencia Bibliografía

### Bibliografía

International Organization for Standardization. (2001). *ISO/IEC 9126-1:2001 — Software engineering — Product quality — Part 1: Quality model*. International Organization for Standardization. <https://www.iso.org/standard/22749.html>.

## 6. Referencia linkografía

- [¿Video de YouTube?](#)
- [Iso 12119](#)
- [Iso 122119 Requerimientos](#)