

SERVICIO NACIONAL DE APRENDIZAJE - SENA

Estándares y Normativas de Calidad en Software

Nicolas Ignacio Parra Silva

Ficha: 3147259

Nombre del Instructor

Fredy Ardila

Programa de Formación

ANÁLISIS Y DESARROLLO DE SOFTWARE

Centro de Formación

CENTRO DE GESTIÓN DE MERCADOS, LOGÍSTICA Y TECNOLOGÍAS DE LA

INFORMACIÓN

Bogotá D.C, Colombia

Octubre 16, 2025

**Actividad 1:** La Torre Incompleta

Instrucción: Reflexione sobre el siguiente escenario y responda de manera individual:

Imagine que su equipo está creando una aplicación de salud que gestiona datos sensibles de pacientes (historiales clínicos). El código es funcional, pero el proyecto no sigue ninguna normativa de seguridad o calidad externa.

- **Riesgo Legal y Ético:** ¿Qué consecuencias legales y de reputación enfrentaría la empresa si, debido a la falta de un estándar de calidad riguroso, los datos de los pacientes se filtran o se pierden?
- **El Caso del "Buenos Días":** Si el equipo cambia de personal, ¿cómo garantiza la empresa que el nuevo desarrollador siga los mismos procesos y produzca la misma calidad que el anterior sin tener un estándar de proceso documentado?
- **Conclusión:** ¿Por qué un estándar o normativa externa (como una ISO) es una base de confianza que va más allá del simple código y se convierte en un activo de negocio?

Solución

1. ¿Qué consecuencias legales y de reputación enfrentaría la empresa si, debido a la falta de un estándar de calidad riguroso, los datos de los pacientes se filtran o se pierden?

**Respuesta:**

La empresa enfrentaría graves sanciones legales por violar leyes de protección de datos personales, como la Ley 1581 de 2012 en Colombia. Además, podría perder la confianza de los pacientes y del público, afectando su reputación y credibilidad. Éticamente, la filtración de información médica es una falta grave al deber de confidencialidad y respeto hacia los pacientes.

2. Si el equipo cambia de personal, ¿cómo garantiza la empresa que el nuevo desarrollador siga los mismos procesos y produzca la misma calidad que el anterior sin tener un estándar de proceso documentado?

**Respuesta:**

Sin un estándar documentado, no hay una guía clara para mantener la calidad ni uniformidad en el trabajo. Cada nuevo desarrollador podría aplicar sus propios métodos, generando inconsistencias, errores y pérdida de tiempo. La empresa no podría asegurar continuidad ni calidad en el desarrollo del sistema.

3. ¿Por qué un estándar o normativa externa (como una ISO) es una base de confianza que va más allá del simple código y se convierte en un activo de negocio?

**Respuesta:**

Porque un estándar como la ISO garantiza que la empresa sigue buenas prácticas reconocidas

internacionalmente, asegurando calidad, seguridad y confiabilidad. Esto genera confianza en los clientes, inversionistas y usuarios, y convierte los procesos de calidad en un activo que respalda la estabilidad y el crecimiento del negocio.

### Actividad 2: Descifrando la Jerarquía

Instrucción: Los términos Modelo, Estándar y Normativa se usan a menudo indistintamente, pero tienen roles conceptuales distintos. Investiga la función principal de cada uno en el contexto de la calidad del software y completa la tabla.

#### Solución

Concepto	Función Principal y Alcance	Ejemplo de la Industria
<b>Modelo (Capability Model)</b>	Evalúa el nivel de madurez y capacidad de los procesos de desarrollo de software. Sirve para identificar debilidades y establecer rutas de mejora continua.	<b>CMMI (Capability Maturity Model Integration)</b>
<b>Estándar (Process Standard)</b>	Define buenas prácticas y procedimientos específicos que deben seguirse durante el desarrollo del software para garantizar calidad y consistencia.	<b>ISO/IEC 12207</b> (Procesos del ciclo de vida del software)
<b>Normativa (Quality Management System)</b>	Establece un marco general para gestionar la calidad en toda la organización, no solo en el desarrollo de software. Se enfoca en la mejora continua y la satisfacción del cliente.	<b>ISO 9001</b> (Sistema de Gestión de la Calidad)

Análisis Conceptual: Con base en la tabla, si una empresa de desarrollo quiere saber cómo mejorar su proceso de desarrollo (madurez), ¿a cuál de los tres conceptos debería recurrir primero? ¿Por qué?

#### Solución

La empresa debería usar primero el **Modelo (Capability Model)** porque ayuda a saber qué tan buenos son sus procesos y en qué nivel de madurez están. Así puede mejorar paso a paso antes de aplicar otros estándares o normas.

### Actividad 3: Identificación de Aplicación de Normas

Instrucción: Utilizando los conceptos de ISO/IEC 25010 y CMMI, identifique qué estándar o modelo se aplicaría para resolver cada uno de los siguientes problemas dentro de un proyecto de software:

Problema a Solucionar	Estándar o Modelo Aplicable	Justificación (¿Por qué ese estándar/modelo ayuda?)
<b>P1:</b> El cliente se queja de que la aplicación es lenta y no puede manejar muchos usuarios a la vez.	<b>ISO/IEC 25010</b>	Porque esta norma evalúa la calidad del software, incluyendo el rendimiento y la eficiencia, ayudando a mejorar la velocidad y capacidad del sistema.
<b>P2:</b> El equipo de desarrollo quiere dejar de ser "caótico" y establecer pasos predecibles y medibles para la mejora continua de sus procesos.	<b>CMMI</b>	Porque el modelo CMMI permite organizar y mejorar los procesos de desarrollo para que sean más estables, medibles y controlados.
<b>P3:</b> El líder quiere verificar si la base de código es fácil de modificar y de arreglar en el futuro.	<b>ISO/IEC 25010</b>	Porque esta norma analiza la mantenibilidad del software, es decir, qué tan fácil es actualizar o corregir el código.
<b>P4:</b> Queremos asegurar que nuestra documentación sea compatible con la de nuestros proveedores en el extranjero.	<b>ISO/IEC 25010</b>	Porque busca que el software y su documentación cumplan estándares internacionales de compatibilidad y calidad.

#### Actividad 4: Mapeo de Atributos al Código

Instrucción: Utilizando los atributos de calidad de ISO/IEC 25010, demuestre cómo se traducirían en requisitos concretos para el desarrollo de su propio proyecto productivo

Atributo de Calidad (ISO/IEC 25010)	Requisito concreto para su proyecto	Buena práctica para asegurarlo	Métrica para demostrar el cumplimiento
<b>Seguridad</b>	El sistema debe cifrar la contraseña con un algoritmo fuerte antes de almacenarla.	- Usar algoritmos de cifrado como <b>bcrypt</b> o <b>Argon2</b> .- Implementar conexión segura <b>HTTPS</b> .- Validar entradas del usuario.- Mantener actualizado el sistema.	- Contraseñas cifradas verificadas en la base de datos.- Certificado SSL activo.- Sin vulnerabilidades críticas detectadas en pruebas.
<b>Mantenibilidad</b>	Cualquier módulo o clase del código debe poder ser modificado, probado e implementado de forma independiente sin afectar la funcionalidad principal de los otros 7 módulos críticos del sistema de pagos.	- Aplicar <b>arquitectura modular</b> o <b>MVC</b> .- Documentar el código.- Implementar <b>pruebas unitarias</b> y usar <b>control de versiones</b> .- Mantener un entorno de integración continua (CI/CD).	- Cobertura de pruebas $\geq 80\%$ .- Tiempo promedio de corrección $< 1$ día.- Errores cruzados entre módulos $< 5\%$ .
<b>Rendimiento / Eficiencia</b>	El tiempo de carga de la página principal del catálogo de productos no debe exceder los 2 segundos bajo una carga de 1,000 usuarios concurrentes realizando búsquedas.	- Implementar <b>caché</b> y optimizar consultas.- Comprimir imágenes y recursos.- Usar <b>CDN</b> .- Pruebas de carga periódicas.	- Tiempo de respuesta $\leq 2$ segundos.- Soporta 1,000 usuarios sin fallos.- Uso de CPU y RAM $< 70\%$ .
<b>Usabilidad</b>	El tiempo promedio que tarda un usuario nuevo en completar el proceso de registro y realizar su primera compra no debe exceder los 3 minutos sin requerir el acceso a la sección de ayuda.	- Interfaz intuitiva (UX/UI clara).- Reducir pasos en registro y compra.- Mensajes de validación claros.- Pruebas de usabilidad con usuarios reales.	- Tiempo promedio $\leq 3$ minutos.- $\geq 90\%$ de usuarios completan sin ayuda.- Satisfacción $\geq 4/5$ .

<b>Fiabilidad</b>	El sistema debe garantizar que las transacciones y operaciones críticas (como pagos o registros) se completen correctamente incluso en caso de fallos de red o errores temporales.	- Implementar <b>copias de seguridad automáticas</b> .- Manejo de errores y recuperación de sesión.- Monitoreo del sistema 24/7.- Uso de servicios en la nube con redundancia.	- Disponibilidad del sistema $\geq 99\%$ .- Tiempo medio entre fallos (MTBF) $\geq 500$ h.- Tasa de fallos críticos $< 1\%$ .
-------------------	--	--	---

#### Actividad 5: La Propuesta de Madurez (CMMI)

Instrucción: Imagine que su equipo de proyecto se encuentra actualmente en el Nivel 1 (Inicial) de CMMI (los procesos son ad-hoc y dependen de los héroes). Prepara una propuesta de mejora para alcanzar el Nivel 2 (Gestionado).

La propuesta debe incluir tres (3) acciones concretas y explicar cómo cada una de ellas introduce el concepto de gestión y planificación para salir del caos del Nivel 1.

<b>Acción Propuesta</b>	<b>Concepto de Gestión (Nivel 2) que Implementa</b>	<b>Impacto en el Equipo</b>
<b>Acción 1:</b> Crear un plan de trabajo con tareas, fechas y responsables.	<b>Gestión de Proyectos:</b> Permite planear y controlar las actividades del proyecto.	El equipo sabe qué debe hacer, cuándo y quién es responsable de cada tarea.
<b>Acción 2:</b> Usar un repositorio compartido (como GitHub) para guardar versiones del código.	<b>Gestión de Configuración:</b> Controla los cambios y versiones del software.	Evita confusiones y pérdidas de código, manteniendo el trabajo ordenado.
<b>Acción 3:</b> Realizar reuniones semanales de seguimiento del avance.	<b>Monitoreo y Control del Proyecto:</b> Supervisa el progreso y ajusta el plan si es necesario.	Mejora la comunicación y permite detectar problemas a tiempo.