



Sistema Autónomo de información automotriz

PLAN DE CALIDAD DE SOFTWARE

Sergio Fernando Limas

Sergio Andres Leon

Brayan Stiven Chavarro

Ficha del documento

Fecha	Autor	Revisión	Revisor	Verificado dep. calidad.
24/08/2022	Sergio Limas Brayan Chavarro Sergio León	08	Edwin Albeiro Ramos	

Tabla de Contenido

1. Introducción
 - 1.1.1. Objetivo del Documento
 - 1.1.2. Alcance
 - 1.1.3. Objetivos de Calidad
2. Implementación de la ISO/IEC 9126
 - 2.1.1. Funcionalidad
 - 2.1.2. Fiabilidad
 - 2.1.3. Usabilidad
 - 2.1.4. Eficiencia
 - 2.1.5. Facilidad de mantenimiento
 - 2.1.6. Portabilidad
3. Implementación de la ISO/IEC 25010
 - 3.1.1. Calidad en el producto
 - 3.1.2. Calidad en el proceso
 - 3.1.3. Calidad en el proceso
 - 3.1.4. Calidad en el sistema
 - 3.1.5. Calidad en el uso
4. Modelo de calidad McCall
 - 4.1.1. Funcionalidad
 - 4.1.2. Fiabilidad
 - 4.1.3. Usabilidad
 - 4.1.4. Eficiencia en el rendimiento
 - 4.1.5. Facilidad de mantenimiento
 - 4.1.6. Integridad
 - 4.1.7. Adaptabilidad
 - 4.1.8. Cumplimiento de estándares
5. Procedimientos de Calidad
 - 5.1.1. Análisis y Diseño de Requisitos
 - 5.1.2. Análisis Exhaustivo de Requisitos
 - 5.1.3. Diseño Integral
 - 5.1.4. Participación de las Partes Interesadas
 - 5.1.5. Desarrollo
 - 5.1.6. Mejores Prácticas de Codificación
 - 5.1.7. Manejo de Errores y Excepciones
 - 5.1.8. Revisiones e Inspecciones de Código
 - 5.1.9. Pruebas de Componentes y Módulos
 - 5.1.10. Pruebas
 - 5.1.11. Planes de Prueba Integrales

- 5.1.12. Pruebas Unitarias, de Integración, de Sistema y de Aceptación
- 5.1.13. Pruebas de Rendimiento
- 5.1.14. Pruebas de Seguridad
- 5.1.15. Implementación y Mantenimiento
- 5.1.16. Procedimientos de Implementación
- 5.1.17. Supervisión en el Entorno de Producción
- 5.1.18. Mantenimiento y Soporte
- 5.1.19. Mejora Continua

6. Listas de Verificación de Calidad

- 6.1.1. Lista de Verificación de Requisitos
- 6.1.2. Lista de Verificación de Diseño
- 6.1.3. Lista de Verificación de Desarrollo
- 6.1.4. Lista de Verificación de Pruebas
- 6.1.5. Lista de Verificación de Implementación y Mantenimiento
- 6.1.6. Lista de chequeo de requisitos del sistema

7. Conclusión

Introducción

Este documento de Calidad del Software tiene como objetivo delinear los estándares y lineamientos para asegurar la calidad de la aplicación web denominada SAIA, desarrollada para la gestión de los siguientes módulos principales:

Usuarios, Inventarios, Compras y Ventas.

Este documento destaca los atributos de calidad del software que se abordarán, proporcionando estrategias como: Procedimientos de calidad y listas de verificación de calidad con el fin de garantizar que se cumplan estos atributos de calidad.

Alcance

Este documento aplicará los lineamientos para el desarrollo y mantenimiento de la aplicación web denominada SAIA, desarrollada para la gestión de los módulos: Usuarios, Inventarios, Compras y Ventas. Cumpliendo o abarcando en todas las fases del ciclo de vida del desarrollo del software, incluido el análisis de requisitos, el diseño, el desarrollo, las pruebas, la implementación y el mantenimiento de la misma.

Objetivos de calidad

Los objetivos de este Documento de Calidad del Software son los siguientes:

1. **Funcionalidad:** La aplicación web debe cumplir con los requisitos funcionales especificados para cada módulo, incluida la gestión de usuarios, la gestión de inventario y la gestión de compras y ventas.
2. **Confiabilidad:** La aplicación web debe ser confiable, garantizando que funcione como se espera y no encuentre errores o fallas inesperadas.
3. **Usabilidad:** La aplicación web debe ser fácil de usar y proporcionar una experiencia de usuario intuitiva y fluida.
4. **Rendimiento:** La aplicación web debe ser eficiente y receptiva, lo que garantiza tiempos de carga rápidos y retrasos mínimos en el procesamiento.
5. **Seguridad:** La aplicación web debe implementar medidas de seguridad adecuadas para proteger los datos del usuario y evitar el acceso no autorizado.
6. **Mantenibilidad:** La aplicación web debe ser fácilmente mantenible, lo que permitirá futuras mejoras, correcciones de errores y actualizaciones.
7. **Escalabilidad:** La aplicación web debe ser escalable, capaz de manejar una mayor carga de usuarios y adaptarse al crecimiento futuro.

Implementación de la Norma ISO/IEC 9126

Describe el modelo de calidad del producto de Software especificando 6 características Internas y Externas. Se manifiestan cuando el software es utilizado como parte de un sistema. La calidad interna evalúa los atributos a satisfacer, medible a partir de características que lo conforman. La calidad externa evalúa que el software satisfaga las necesidades del usuario, medible en el comportamiento del producto. Nos permite especificar y evaluar la calidad del Software desde distintas perspectivas, asociadas con la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoría del software.

1. Funcionalidad:

- Adecuación: La aplicación debe ser adecuada para los requisitos funcionales de los módulos de Usuarios, Inventarios, Compras y Ventas.
- Exactitud: Debe funcionar de manera precisa y entregar resultados correctos.
- Interoperabilidad: Debe ser capaz de interactuar con otros sistemas o aplicaciones según sea necesario.

2. Fiabilidad:

- Madurez: La aplicación debe ser madura, lo que significa que no debe tener errores graves ni fallas.
- Tolerancia a fallos: Debe ser capaz de manejar situaciones inesperadas o errores sin fallar por completo.
- Capacidad de recuperación: Debe poder recuperarse de fallos y volver a un estado operativo normal.

3. Usabilidad:

- Inteligibilidad: La interfaz de usuario debe ser clara y fácil de entender.
- Aprendizaje: Los usuarios deben poder aprender a utilizar la aplicación de manera rápida y sencilla.
- Operabilidad: Debe ser fácil de operar y eficiente en su uso diario.

4. Eficiencia:

- Comportamiento en el tiempo: Debe tener un rendimiento rápido y eficiente en términos de tiempos de respuesta y recursos utilizados.
- Uso de recursos: Debe utilizar los recursos del sistema de manera eficiente.

5. Facilidad de Mantenimiento:

- Analizabilidad: Debe ser fácil de analizar para identificar problemas y realizar mejoras.
- Modificabilidad: Debe ser fácil de modificar para futuras actualizaciones y correcciones de errores.
- Estabilidad: Las modificaciones no deben introducir nuevos errores ni afectar negativamente el sistema existente.

6. Portabilidad:

- Adaptabilidad: Debe ser adaptable a diferentes entornos y plataformas según sea necesario.

Implementación de la ISO/IEC 25010

A continuación, se utilizarán las características definidas en ISO/IEC 25010 para evaluar los requisitos del proyecto SAIA:

1. Calidad en el producto:

a. Funcionalidad:

- Adecuación: La aplicación debe ser adecuada para los requisitos funcionales de los módulos de Usuarios, Inventarios, Compras y Ventas.
- Exactitud: Debe funcionar de manera precisa y entregar resultados correctos.
- Interoperabilidad: Debe ser capaz de interactuar con otros sistemas o aplicaciones según sea necesario.

b. Eficiencia en el desempeño:

- Comportamiento en el tiempo: La aplicación debe tener un rendimiento rápido y eficiente en términos de tiempos de respuesta y recursos utilizados.
- Capacidad: Debe ser capaz de manejar una carga de usuarios y datos adecuada sin degradación significativa del rendimiento.

c. Compatibilidad:

- Coexistencia: La aplicación debe poder coexistir con otros sistemas o software en el mismo entorno.

d. Usabilidad:

- Reconocimiento de idoneidad: Los usuarios deben sentir que la aplicación es adecuada para sus necesidades.
- Aprendizaje: Los usuarios deben poder aprender a utilizar la aplicación de manera rápida y sencilla.
- Operabilidad: Debe ser fácil de operar y eficiente en su uso diario.

2. Calidad en el proceso:

a. Capacidad de mantenimiento:

- Analizabilidad: Debe ser fácil de analizar para identificar problemas y realizar mejoras.
- Modificabilidad: Debe ser fácil de modificar para futuras actualizaciones y correcciones de errores.

b. Calidad en la transición:

- Instalabilidad: Debe ser fácil de instalar en diferentes entornos.

3. Calidad en el sistema:

a. Seguridad:

- Confidencialidad: La aplicación debe proteger la información confidencial de los usuarios.
- Integridad: Debe garantizar la integridad de los datos y la seguridad de las transacciones.
- No repudio: Debe proporcionar evidencia de la autenticidad de las transacciones y actividades realizadas por los usuarios.

b. Madurez:

- Tolerancia a fallos: Debe ser capaz de manejar situaciones inesperadas o errores sin fallar por completo.
- Disponibilidad: Debe estar disponible y funcional cuando los usuarios la necesiten.

c. Facilidad de uso:

- Comportamiento humano: La aplicación debe ser intuitiva y fácil de usar para los usuarios finales.

4. Calidad en el uso:

a. Eficacia:

- Eficiencia en el uso: Optimiza la eficiencia de las operaciones comunes, de modo que los usuarios puedan realizar tareas con la menor cantidad de esfuerzo posible.
- Cumplimiento de tareas: Los usuarios deben poder realizar sus tareas de manera efectiva utilizando la aplicación.

b. Satisfacción del usuario:

- Satisfacción del usuario: Recopila y considera la retroalimentación de los usuarios para mejorar constantemente la usabilidad de la aplicación.

ISO 9126 Implementada para establecer los requisitos

ISO 25010 Implementada para evaluar los Requisitos

Modelo de calidad McCall

Es un marco de calidad del software que se utiliza para evaluar y medir la calidad de un sistema de software en función de diversos atributos clave. Estos atributos incluyen la funcionalidad, la fiabilidad, la usabilidad, la eficiencia, la facilidad de mantenimiento, la integridad, la adaptabilidad y el cumplimiento de estándares. Implementar el modelo McCall en el proyecto SAIA es esencial porque proporciona una estructura sólida para medir y mejorar la calidad del software en todos estos aspectos clave.

1. Funcionalidad (MF - Functionality):

- Completitud: Asegúrate de que todos los requisitos funcionales especificados para los módulos de Usuarios, Inventarios, Compras y Ventas se implementen correctamente.
- Corrección: Verifica que la aplicación web funcione sin errores y que las funciones realizadas sean precisas.
- Concisión: Evita la redundancia en las funciones y garantiza que la aplicación realice tareas de manera eficiente y sin exceso de características innecesarias.

2. Fiabilidad (MR - Reliability):

- Tolerancia a fallos: Implementa mecanismos de recuperación de fallos para que la aplicación pueda manejar situaciones inesperadas sin interrupciones graves.
- Disponibilidad: Asegura que la aplicación esté disponible y funcional cuando los usuarios la necesitan, minimizando el tiempo de inactividad no planificado.

3. Usabilidad (MH - Usability):

- Facilidad de aprendizaje: Diseña la interfaz de usuario de manera que los usuarios puedan aprender a utilizar la aplicación de manera rápida y sencilla.
- Eficiencia de uso: Optimiza la eficiencia de las operaciones comunes, de modo que los usuarios puedan realizar tareas con la menor cantidad de esfuerzo posible.
- Satisfacción del usuario: Recopila y considera la retroalimentación de los usuarios para mejorar constantemente la usabilidad de la aplicación.

4. Eficiencia en el rendimiento (ME - Efficiency):

- Tiempo de respuesta: Asegúrate de que la aplicación tenga tiempos de carga rápidos y respuestas ágiles a las acciones del usuario.
- Uso de recursos: Optimiza el uso de recursos, como la memoria y el ancho de banda, para garantizar que la aplicación funcione de manera eficiente.

5. Facilidad de mantenimiento (MM - Maintainability):

- Modularidad: Diseña la aplicación de manera modular para que sea más fácil de mantener y actualizar en el futuro.
- Documentación: Proporciona una documentación clara y completa del código fuente y los procedimientos, lo que facilita el mantenimiento por parte de otros miembros del equipo o desarrolladores externos.

6. Integridad (MI - Integrity):

- Seguridad de datos: Implementa medidas de seguridad sólidas para proteger los datos del usuario y garantizar que no sean accesibles de manera no autorizada.
- Control de acceso: Gestiona adecuadamente quién tiene acceso a las funciones y datos críticos de la aplicación.

7. Adaptabilidad (MA - Adaptability):

- Escalabilidad: Diseña la aplicación de manera que pueda manejar un aumento en la carga de usuarios sin degradación significativa del rendimiento.
- Flexibilidad: Permite que la aplicación se adapte a cambios en los requisitos o en el entorno sin requerir una revisión importante.

8. Cumplimiento de estándares (MP - Portability):

- Cumplimiento de estándares de desarrollo: Asegúrate de que el código cumpla con los estándares de codificación y las mejores prácticas de desarrollo reconocidas.
- Documentación de estándares: Documenta los estándares y prácticas utilizadas para que otros desarrolladores puedan comprender y seguir el mismo enfoque.

Procedimientos de Calidad

1. Analizar y diseñar los requisitos.

- Realizar un análisis exhaustivo de los requisitos de cada módulo y asegúrese de que sean claros, completos e inequívocos.
- Realizar una fase de diseño integral, considerando la usabilidad, escalabilidad y mantenibilidad de la aplicación web.
- Involucrar a las partes interesadas, incluidos los usuarios finales, en el proceso de análisis y diseño de requisitos para obtener sus aportes y comentarios.

2. Desarrollar código fuente.

- Seguir las mejores prácticas de codificación, cumpliendo con los estándares de codificación para garantizar un código coherente y fácil de mantener.
- Implementar mecanismos adecuados de manejo de errores y manejo de excepciones.
- Realizar revisiones e inspecciones de código para identificar cualquier problema o vulnerabilidad de codificación.
- Probar componentes y módulos individuales durante el proceso de desarrollo para identificar y solucionar cualquier defecto o problema de manera temprana.

3. Realizar Pruebas

- Desarrollar planes de prueba integrales y casos de prueba para cubrir todas las funcionalidades y escenarios.
- Realizar pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación para garantizar que la aplicación web cumpla con los requisitos especificados.
- Realizar pruebas de rendimiento para evaluar el tiempo de respuesta, el uso de recursos y la escalabilidad de la aplicación.
- Realizar pruebas de seguridad para identificar y mitigar posibles vulnerabilidades de seguridad.

4. Asegurar la Implementación y el mantenimiento

- Asegurar que se sigan los procedimientos de implementación adecuados, incluido el control de versiones, la gestión de versiones y la gestión de configuración.
- Supervisar periódicamente la aplicación web en el entorno de producción para identificar y resolver cualquier problema o cuello de botella en el rendimiento.
- Proporcionar mantenimiento y soporte oportunos para correcciones de errores, mejoras y actualizaciones.
- Evaluar y mejorar continuamente la aplicación web en función de los comentarios de los usuarios y los requisitos cambiantes.

Listas de verificación de calidad

Se deben utilizar las siguientes listas de verificación durante las fases de desarrollo, prueba e implementación para garantizar la calidad de la aplicación web:

1. Lista de verificación de requisitos:
 - ¿Los requisitos son claros, completos e inequívocos?
 - ¿Han revisado y aprobado todos los requisitos?
2. Lista de verificación de diseño:
 - ¿El diseño aborda la usabilidad, la escalabilidad y la mantenibilidad?
 - ¿Todos los aspectos del diseño han sido revisados y aprobados por las partes interesadas relevantes?
3. Lista de verificación de desarrollo:
 - ¿El estilo de codificación es consistente y cumple con los estándares de codificación?
 - ¿Se implementa correctamente el manejo de errores y el manejo de excepciones?
 - ¿Se han realizado revisiones e inspecciones del código?
4. Lista de verificación de pruebas:
 - ¿Están todas las funcionalidades y escenarios cubiertos por planes de prueba y casos de prueba?
 - ¿Se han realizado pruebas unitarias, pruebas de integración, pruebas de sistemas y pruebas de aceptación?
 - ¿Se han realizado pruebas de rendimiento y seguridad?
5. Lista de verificación de implementación y mantenimiento:
 - ¿Se siguen los procedimientos de implementación adecuados, incluido el control de versiones y la gestión de lanzamientos?
 - ¿Se monitorea adecuadamente la aplicación web en el entorno de producción?
 - ¿Existe algún mecanismo para corregir errores y actualizarlos oportunamente?
6. Lista de chequeo de requisitos del sistema

Basada en una lista de comprobación, como mecanismo primario de control de la calidad de las especificaciones del software. Su objetivo es encontrar defectos en la definición de las necesidades del usuario para el sistema o componente, una mala descripción de los requisitos de usuario, funcionalidades incorrectas o incompatibles, interfaces de usuario no claras, elementos de software y/o hardware incorrectos en los puntos de interacción

del sistema con el entorno. Basado en el modelo de calidad internacional ISO 25010 donde se establecen las características de calidad que se evalúan en un producto software, se construye la plantilla para el aseguramiento de calidad de los requisitos del sistema.

Funcionalidad	Respuesta
¿Se han especificado todas las tareas que debe realizar el sistema/software?	
Para cada tarea especificada, ¿se ha detallado el contenido de datos/información utilizado por la tarea	
¿Se ha especificado el contenido de datos/información que se obtendrá como resultado de la misma?	
¿El sistema de información cuenta con componentes que actúen como “compresores”, es decir, proyectados para recibir más información de la que transmiten? (Compresión)	
¿Se ofrecen las herramientas necesarias para añadir, borrar, mantener, exhibir, imprimir, buscar y actualizar datos?	
¿Se cumple con la función de procesamiento, que implica la modificación de la base de datos para mantenerla actualizada? (Procesamiento)	
¿Permite la recuperación de datos en tiempo real?	
¿Se han especificado las pre-condiciones para cada una de las funcionalidades del sistema/software?	
EFICIENCIA / DESEMPEÑO	RESPUESTA
¿El sistema tiene configurado un tiempo máximo de sesión de usuario?	
¿Se ha especificado el tiempo de respuesta esperado de todas las operaciones especificadas?	
¿Se han especificado otras consideraciones temporales, tales como el tiempo de procesamiento, el de transferencia de datos o la tasa de transferencia?	
¿Se ha establecido el tiempo promedio de movimiento entre pantallas?	
¿Se ha establecido el tiempo máximo de movimiento entre pantallas?	
¿Se ha establecido, para el uso de la memoria del servidor, un porcentaje que no exceda el uso de la memoria disponible?	
¿Se ha especificado la fiabilidad del sistema/software, incluyendo las consecuencias en el caso de que falle?	
PORTABILIDAD	RESPUESTA

¿Se ha definido la adaptabilidad del sistema ya sea hardware, software, operaciones o de uso?	
¿Se ha especificado qué tipo de servidor se va a utilizar?	
¿Se ha especificado la capacidad del servidor?	
¿Se ha definido el motor de base de datos?	
¿Se ha definido la forma para ser instalado o desinstalado de manera exitosa en determinado entorno?	
¿Se ha definido cómo ser reemplazado por otro producto software teniendo en cuenta el mismo propósito y entorno del software?	
¿Se ha definido el sistema operativo que se va a utilizar? *móvil	
¿Se ha definido la versión del sistema operativo? *móvil	
FIABILIDAD	RESPUESTA
¿Se especificaron las consecuencias por causa de fallas en la implementación del requerimiento?	
¿Se definió el plan de contingencia en caso de fallas?	
¿Se definió la estrategia de detección y corrección de errores a causa de las fallas?	
¿Se identifican en los requisitos atributos relacionados con la capacidad del sistema para mantener su nivel de prestación de servicio en las condiciones establecidas durante un período?	
¿La capacidad del sistema para tolerar errores está especificada en los requisitos?	
¿La capacidad del sistema para tolerar sobrecargas en el volumen de información, de usuarios o de procesos está especificada en los requisitos?	
SEGURIDAD	RESPUESTA
¿Se tiene prevista la capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente?	
¿Se considera el control de accesos o modificaciones no autorizados a datos o programas de ordenador?	
¿Se tiene previsto demostrar la identidad de un sujeto o un recurso?	
COMPATIBILIDAD	RESPUESTA
¿Se incluye la respuesta del sistema a los cambios en el entorno operativo, las interfaces, la precisión, el rendimiento, y otras capacidades adicionales predecibles?	

¿Se incluye la manera en que el sistema interactúa con otros software?	
¿Se especifica la comunicación del sistema con otros sistemas software?	
¿Se determina cuáles son los recursos que el sistema compartirá con otros productos software y cómo se hará dicho proceso?	
¿Se especifica el ambiente sobre el cual el sistema funcionará y con cuáles ambientes no lo hará?	
¿Se especifican los requerimientos software y hardware sobre los cuales el sistema funcionará correctamente?	

Los requisitos no funcionales son las restricciones de diseño y de comportamiento que debe cumplir un producto software, entre ellos están:

- Eficiencia de desempeño
- Seguridad
- Portabilidad
- Mantenibilidad
- Compatibilidad
- Usabilidad
- Fiabilidad

En la anterior Checklist se determinan los ítems antes mencionados, dando a conocer si el software cumple o no con cada uno de ellos, para garantizar el aseguramiento de calidad del software.

Conclusión

Podemos documentar de forma más detallada y específica los requerimientos necesarios para garantizar, que el proyecto desarrollado como lo es "SAIA", cumpla con los atributos de calidad establecidos. Bajo los lineamientos especificados durante el ciclo de vida del software, el equipo de desarrollo puede garantizar que la aplicación web sea funcional, confiable, fácil de usar, eficaz, segura, mantenible y escalable, como resultado esperado, dando así cumplimiento a los requerimientos del cliente y cumpliendo sus expectativas.

Historial de Revisiones

Versión	Fecha	Autor	Descripción de la Revisión
1.0	[Fecha]	[Nombre]	Versión Inicial
1.1	[Fecha]	[Nombre]	[Descripción de la revisión]

--

2.0	[Fecha]	[Nombre]	[Descripción de la revisión]
-----	---------	----------	------------------------------