

ISO/IEC/IEEE 12207:2026 Systems and software engineering, Software life cycle processes (2.ª edición, abril de 2026)

Marco de procesos para el ciclo de vida del software

63 requisitos. Para esta norma no existe un certificado acreditado. Versión de 07/2026.

ISO/IEC/IEEE 12207 define un marco de 30 procesos que cubre todo el ciclo de vida del software, desde el acuerdo hasta el desarrollo, la operación y la baja. No existe un esquema de certificación acreditado: la norma trata explícitamente la conformidad como una autodeclaración en tres variantes (conformidad total con los resultados, conformidad total con las tareas, o conformidad adaptada con un registro documentado del ajuste), de modo que el resultado de una evaluación es una declaración de conformidad y no un certificado. Las organizaciones que necesitan un juicio externo encargan una evaluación de procesos basada en la serie ISO/IEC 33000, que entrega un perfil de capacidad en forma de informe.

Empresa	Fecha	Elaborado por
---------	-------	---------------

6.1 Procesos de acuerdo (Agreement Processes)

6

6.1.1 El proceso de Adquisición está implantado: se identifica la necesidad de software o servicios obtenidos externamente, se seleccionan proveedores y se concluye y supervisa un acuerdo.

Documento

Cómo se demuestra: La evidencia incluye documentos de licitación o de requisitos, comparativas de ofertas, contratos u órdenes de compra firmados, y registros de aceptación. En equipos pequeños basta con un ticket breve de adquisición por compra en el sistema de seguimiento de incidencias, que recoja la necesidad, la comparación de alternativas y la aprobación.

Abierto	En curso	Cumplido	No aplicable
---------	----------	----------	--------------

Evidencia / justificación

6.1.1.2 Se logran los resultados de la adquisición: se definen los requisitos del elemento adquirido, se concluye el acuerdo y se confirma su cumplimiento.

Cómo se demuestra: Criterios de aceptación en el contrato o en el ticket, más la prueba de aceptación o el registro de aceptación. Para dependencias de código abierto, el archivo de bloqueo (lockfile) con un criterio de selección documentado y el análisis de licencias sirven como evidencia de aceptación.

Abierto	En curso	Cumplido	No aplicable
---------	----------	----------	--------------

Evidencia / justificación

6.1.1.3 Se llevan a cabo las actividades de adquisición: la identificación de la necesidad, la selección de proveedores, la conclusión del acuerdo, el seguimiento del progreso y la aceptación se realizan de forma trazable.

Cómo se demuestra: Registros del sistema de seguimiento de incidencias y de la carpeta de contratos, complementados con informes de estado de los proveedores. Para organizaciones pequeñas basta con una lista de proveedores con la fecha de la última evaluación.

Abierto	En curso	Cumplido	No aplicable
---------	----------	----------	--------------

Evidencia / justificación

6.1.2 El proceso de Suministro está implantado: la organización revisa las solicitudes, presenta una oferta, concluye un acuerdo, entrega el producto o servicio y cierra formalmente el acuerdo.

Documento

Cómo se demuestra: Ofertas, confirmaciones de pedido, notas de entrega y de versión, registros de traspaso y facturas. Las notas de versión por cada versión son la evidencia de entrega más práctica para equipos de software.

Abierto	En curso	Cumplido	No aplicable
---------	----------	----------	--------------

Evidencia / justificación

6.1.2.2 Se logran los resultados del suministro: se acuerda el alcance de la entrega, se traspasa el producto y se resuelve la responsabilidad sobre él.

Cómo se demuestra: Artefactos de versión firmados con número de versión, registro de cambios y un mensaje de traspaso al cliente. Además, los acuerdos de soporte y garantía en las condiciones del contrato.

Abierto	En curso	Cumplido	No aplicable
---------	----------	----------	--------------

Evidencia / justificación

6.1.2.3 Se llevan a cabo las actividades de suministro: se evalúa la viabilidad, se ejecuta la entrega y el acuerdo se cierra una vez cumplido.

Cómo se demuestra: Un registro de la estimación de viabilidad o de esfuerzo en la oferta, el registro de despliegue del pipeline de integración continua como evidencia de entrega, y un breve registro de cierre por pedido.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2 Procesos organizativos de habilitación de proyectos (Organizational Project-Enabling Processes)

9

6.2.1 La Gestión del Modelo de Ciclo de Vida está implantada: la organización define, mantiene y pone a disposición los modelos de ciclo de vida, procesos y procedimientos que utilizan los proyectos.

[Documento](#)

Cómo se demuestra: Una descripción de proceso sencilla en el repositorio, por ejemplo un CONTRIBUTING.md más una página con el flujo de trabajo de desarrollo, el modelo de ramas, la definición de terminado y la cadencia de versiones. Sus cambios se pueden fechar a través del historial de Git.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2.1.2 Se logran los resultados: los modelos de ciclo de vida en uso están definidos, los proyectos los conocen y se revisan periódicamente en cuanto a su eficacia.

Cómo se demuestra: Notas de retrospectiva o una entrada de revisión anual en el repositorio que explique el motivo de adaptar el flujo de trabajo. Las solicitudes de incorporación de cambios (pull requests) sobre los documentos de proceso son la evidencia de eficacia.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2.2 La Gestión de la Infraestructura está implantada: se define, se provee y se mantiene la infraestructura necesaria para desarrollo, prueba y operación.

Cómo se demuestra: Infraestructura como código en el repositorio (Terraform, Docker Compose, configuración de ejecutores de integración continua), más una lista de los servicios en uso con sus responsables. Para equipos pequeños basta con una página con entornos, accesos y reglas de copia de seguridad.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2.2.3 Se llevan a cabo las actividades de infraestructura: se identifican las necesidades, se adquieren los recursos, se operan y se retiran cuando ya no se necesitan.

Cómo se demuestra: Historial de cambios de la configuración de infraestructura en Git, paneles de monitorización y tickets para cambios de entorno. Los entornos dados de baja se documentan en el ticket con una fecha.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2.3 La Gestión de la Cartera está implantada: los proyectos se inician, se priorizan, se dotan de recursos, se evalúan de forma continua y se finalizan cuando dejan de aportar valor.

Cómo se demuestra: Una hoja de ruta de producto o un tablero de cartera con prioridades y fechas de decisión, más registros de las iniciativas que se detuvieron. Un documento de hoja de ruta con revisión trimestral basta para organizaciones pequeñas.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2.4 La Gestión de Recursos Humanos está implantada: se determinan las competencias necesarias, el personal se cualifica y el conocimiento se genera y se conserva.

Cómo se demuestra: Una matriz de competencias por rol, registros de formación y certificados, una lista de verificación de incorporación. En equipos pequeños basta una tabla de habilidades que muestre las carencias y la formación planificada, junto con evidencia de la formación completada.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2.5	La Gestión de la Calidad está implantada: se definen los objetivos de calidad, la política de calidad y las responsabilidades, y se evalúa el logro de los objetivos.				Documento
	Cómo se demuestra: Una política de calidad breve en el repositorio, objetivos de calidad definidos (cobertura de pruebas, densidad de defectos, tiempo de respuesta ante incidencias) y las métricas correspondientes procedentes de la integración continua y del sistema de seguimiento de incidencias.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					
6.2.5.2	Se logran los resultados de la gestión de la calidad: los objetivos de calidad son medibles, las desviaciones se detectan y se hace seguimiento de las acciones correctivas.				
	Cómo se demuestra: Informes de tendencia del pipeline de integración continua (cobertura, compilaciones fallidas), estadísticas de defectos del sistema de seguimiento de incidencias, y tickets de las acciones resultantes con una fecha de cierre.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					
6.2.6	La Gestión del Conocimiento está implantada: el conocimiento, las habilidades y los activos reutilizables se capturan, se comparten y se mantienen disponibles para proyectos posteriores.				
	Cómo se demuestra: Una base de conocimiento interna o wiki, registros de decisiones de arquitectura en el repositorio, bibliotecas reutilizables con documentación. Las notas de análisis posterior a incidentes (postmortem) son una evidencia sólida.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					
6.3 Procesos técnicos de gestión (Technical Management Processes)					16
6.3.1	La Planificación del Proyecto está implantada: se planifican el alcance, los entregables, el cronograma, el esfuerzo, los roles y las interfaces, y el plan se mantiene actualizado.				Documento
	Cómo se demuestra: Un plan de proyecto o backlog con hitos, estimaciones y responsables. Un tablero de sprint o de hitos mantenido en el sistema de seguimiento de incidencias cumple el requisito siempre que el alcance y las fechas sean visibles.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					
6.3.1.2	Se logran los resultados de la planificación: se asignan tareas y recursos, se nombran los roles y los planes se acuerdan con los implicados.				
	Cómo se demuestra: Incidencias asignadas con un responsable y una versión objetivo, más el resumen de hitos. Los supuestos de capacidad se anotan brevemente en el ticket de planificación.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					
6.3.1.3	Se llevan a cabo las actividades de planificación: se define el alcance del proyecto, se elaboran los planes y se actualizan cada vez que se producen cambios.				
	Cómo se demuestra: Historial de versiones del documento de plan o de hoja de ruta en Git, y comentarios de cambio en los hitos. Esto hace que cada actualización sea fechable.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					
6.3.2	La Evaluación y Control del Proyecto está implantada: se evalúa el avance frente al plan, se detectan desviaciones y se toman acciones de control.				Documento
	Cómo se demuestra: Informes de estado, revisiones de sprint, métricas de burndown o de rendimiento del sistema de seguimiento de incidencias, y registros de las decisiones tomadas cuando se produjeron desviaciones. Basta con una breve nota de estado semanal.				
	Abierto	En curso	Cumplido	No aplicable	
Evidencia / justificación					

6.3.3	La Gestión de Decisiones está implantada: las decisiones se preparan de forma estructurada, se evalúan alternativas, y la selección se justifica y se registra.	Documento
Cómo se demuestra: Registros de decisiones de arquitectura en el repositorio, que cubran contexto, opciones, decisión y consecuencias. Para decisiones menores basta un comentario razonado en la solicitud de incorporación de cambios.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		
6.3.3.2	Se logran los resultados de la gestión de decisiones: se reconoce la necesidad de una decisión, se evalúan alternativas y la decisión tomada es trazable junto con su justificación.	
Cómo se demuestra: La carpeta de registros de decisiones de arquitectura con numeración secuencial y estado (propuesto, aceptado, sustituido). El historial de Git evidencia la fecha y el autor.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		
6.3.4	La Gestión de Riesgos está implantada: los riesgos se identifican, se analizan, se tratan y se supervisan de forma continua.	Documento
Cómo se demuestra: Un registro de riesgos con probabilidad, impacto, tratamiento y responsable. Los riesgos técnicos se pueden seguir como incidencias con una etiqueta de riesgo, junto con los resultados de los análisis de dependencias y de vulnerabilidades.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		
6.3.4.2	Se logran los resultados de la gestión de riesgos: los riesgos se priorizan, se definen umbrales de actuación y se hace seguimiento de la eficacia del tratamiento.	
Cómo se demuestra: Historial de actualización del registro de riesgos, tickets de riesgo cerrados con su justificación, e informes del análisis automatizado de dependencias que muestren la fecha de corrección.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		
6.3.5	La Gestión de la Configuración está implantada: se identifican los elementos de configuración, se definen líneas base, los cambios se controlan y el estado de cada elemento se puede determinar en todo momento.	Documento
Cómo se demuestra: Git como control de versiones con etiquetas para las versiones publicadas, ramas principales protegidas, archivos de bloqueo (lockfiles) para las dependencias y una lista de materiales de software (SBOM) por versión. El historial de etiquetas es el registro de líneas base.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		
6.3.5.2	Se logran los resultados de la gestión de la configuración: se establecen líneas base, los cambios son trazables y las versiones son reproducibles.	
Cómo se demuestra: Una compilación reproducible a partir de una etiqueta de Git a través del pipeline de integración continua, más los registros de compilación y un repositorio de artefactos con sumas de verificación.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		
6.3.5.3	Se llevan a cabo las actividades de gestión de la configuración: las solicitudes de cambio se evalúan, se aprueban o se rechazan, y su incorporación se verifica.	
Cómo se demuestra: Historial de solicitudes de incorporación de cambios con revisión obligatoria y una incidencia enlazada, y reglas de fusión en el repositorio (protección de ramas) como evidencia del control de aprobación.		
Abierto En curso Cumplido No aplicable		
Evidencia / justificación		

6.3.6	La Gestión de la Información está implantada: se determina la información que debe crearse y conservarse, y se mantiene, se protege y se pone a disposición de quienes tienen derecho a ella.	Documento		
	Cómo se demuestra: Un concepto de registros que cubra el lugar de almacenamiento, el responsable, el periodo de conservación y los derechos de acceso. Los permisos del repositorio y los registros de copia de seguridad evidencian la protección y la disponibilidad.			
	Abierto	En curso	Cumplido	No aplicable

Evidencia / justificación

6.3.7	El proceso de Medición está implantado: se determinan las necesidades de información, se definen medidas adecuadas, se recopilan, se analizan y se utilizan para tomar decisiones.			
	Cómo se demuestra: Métricas de la integración continua y del sistema de seguimiento de incidencias, como el tiempo de entrega, la tasa de defectos, la cobertura de pruebas o la frecuencia de despliegue. Lo que importa es el camino evidenciado desde la medida hasta la decisión, por ejemplo en el registro de la retrospectiva.			
	Abierto	En curso	Cumplido	No aplicable

Evidencia / justificación

6.3.7.2	Se logran los resultados de la medición: los datos recopilados son válidos, el análisis está documentado y los resultados se comunican.			
	Cómo se demuestra: Informes de pipeline generados automáticamente con marca de tiempo, y un resumen periódico de métricas publicado en el canal del equipo o en la wiki.			
	Abierto	En curso	Cumplido	No aplicable

Evidencia / justificación

6.3.8	El Aseguramiento de la Calidad está implantado: los productos y los procesos se comprueban de forma independiente frente a los criterios definidos, y las desviaciones se registran y se resuelven.	Documento		
	Cómo se demuestra: Revisión obligatoria de cuatro ojos en la solicitud de incorporación de cambios, ejecuciones de linter y de pruebas como condición de fusión, y registro de las desviaciones encontradas como incidencias. Un equipo pequeño logra la independencia mediante revisores rotativos en lugar de un departamento de calidad dedicado.			
	Abierto	En curso	Cumplido	No aplicable

Evidencia / justificación

6.3.8.2	Se logran los resultados del aseguramiento de la calidad: se evalúan la conformidad del proceso y la conformidad del producto, y las acciones correctivas se siguen hasta su cierre.			
	Cómo se demuestra: Comentarios de revisión con estado de resolución, tickets de errores cerrados que hacen referencia al commit de corrección, y una ejecución de pipeline en verde para la etiqueta de versión.			
	Abierto	En curso	Cumplido	No aplicable

Evidencia / justificación

6.4 Procesos técnicos (Technical Processes) 26

6.4.1	Se realiza el Análisis de Negocio o de Misión: se determinan el problema por resolver, el espacio de soluciones y las restricciones del proyecto.			
	Cómo se demuestra: Una visión de producto, un planteamiento del problema o un caso de negocio en la wiki, complementado con análisis de mercado o entrevistas a usuarios. Para proyectos pequeños basta con una épica con una descripción del problema y un objetivo en el sistema de seguimiento de incidencias.			
	Abierto	En curso	Cumplido	No aplicable

Evidencia / justificación

6.4.2	Se realiza la Definición de Necesidades y Requisitos de las Partes Interesadas: se identifican las partes interesadas, se recogen sus necesidades y se transforman en requisitos verificables de las partes interesadas.	Documento
Cómo se demuestra: Una lista de partes interesadas, notas de entrevistas, historias de usuario con criterios de aceptación. Los criterios de aceptación del ticket son la forma verificable del requisito.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		
6.4.2.2	Se logran los resultados: los requisitos de las partes interesadas se acuerdan, se priorizan y son trazables hasta la necesidad subyacente.	
Cómo se demuestra: Un backlog priorizado que enlaza la historia con la épica, más el comentario de aprobación del cliente en el ticket.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		
6.4.3	Se realiza la Definición de Requisitos del Sistema: los requisitos técnicos del sistema de software se derivan de los requisitos de las partes interesadas, incluyendo características de calidad y restricciones.	Documento
Cómo se demuestra: Un documento de requisitos o incidencias estructuradas que cubran requisitos funcionales y no funcionales (rendimiento, seguridad, accesibilidad). Enlazar cada requisito con un caso de prueba hace explícita la verificabilidad.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		
6.4.3.2	Se logran los resultados de la definición de requisitos: los requisitos son inequívocos, están libres de contradicciones, son verificables y trazables hasta los requisitos de las partes interesadas.	
Cómo se demuestra: Una matriz de trazabilidad, o alternatively enlaces entre la historia, la prueba y el commit. Las pruebas de aceptación automatizadas son a la vez la evidencia de verificabilidad.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		
6.4.4	Se realiza la Definición de la Arquitectura del Sistema: se define y se evalúa frente a los requisitos una arquitectura con vistas, bloques de construcción e interfaces.	Documento
Cómo se demuestra: Una vista general de arquitectura en el repositorio (por ejemplo diagramas arc42 o C4 como imagen más texto), junto con registros de decisiones de arquitectura que recojan las alternativas evaluadas.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		
6.4.4.2	Se logran los resultados de la definición de la arquitectura del sistema: la arquitectura se evalúa frente a los requisitos, se sopesan alternativas y la selección se justifica.	
Cómo se demuestra: Un registro de decisión de arquitectura que sopesa opciones y consecuencias, más los resultados de una revisión de arquitectura en la solicitud de incorporación de cambios o en un acta de reunión.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		
6.4.5	Se realiza la Definición del Diseño: existen diseños de los bloques de construcción de la arquitectura suficientes para la implementación, incluyendo el modelo de datos y los contratos de interfaz.	
Cómo se demuestra: Especificaciones de interfaz (OpenAPI, esquema GraphQL, definiciones de tipos), scripts de migración de base de datos y notas de diseño en la solicitud de incorporación de cambios. El propio código, con sus tipos y límites de módulo, es evidencia de diseño aceptable siempre que las decisiones estén documentadas.		
AbiertoEn cursoCumplidoNo aplicable		
Evidencia / justificación		

6.4.5.2 Se logran los resultados de la definición del diseño: las características del diseño son trazables hasta la arquitectura y la viabilidad está confirmada.

Cómo se demuestra: Referencias desde la solicitud de incorporación de cambios al registro de decisión de arquitectura y al requisito, y prototipos o pruebas de concepto (spikes) con una nota de resultado como evidencia de viabilidad.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.6 Se realiza el Análisis del Sistema: se investigan cuestiones técnicas con métodos analíticos para fundamentar decisiones sobre arquitectura, diseño o riesgos.

Cómo se demuestra: Mediciones de carga y resultados de pruebas de rendimiento (benchmarks), análisis de costes o de compensación de alternativas, notas de modelado de amenazas. Los resultados se archivan junto con el registro de decisión de arquitectura relacionado como una breve nota de análisis.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.7 Se realiza la Implementación: los elementos de software se materializan de acuerdo con el diseño y las reglas de codificación, y los resultados se someten a control de configuración.

Cómo se demuestra: Historial de Git con commits que hacen referencia a incidencias, reglas de linter y de formato como condición de fusión, y compilaciones exitosas en el pipeline de integración continua.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.7.3 Se llevan a cabo las actividades de implementación: se definen los estándares de codificación, la realización se revisa y los elementos se ponen a disposición para la integración.

Cómo se demuestra: Archivos de configuración de linter y de formateador en el repositorio, revisiones de solicitudes de incorporación de cambios aprobadas, y artefactos de compilación publicados en el repositorio de artefactos.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.8 Se realiza la Integración: los elementos de software se ensamblan progresivamente en un sistema funcional y se comprueban las interfaces.

Cómo se demuestra: Ejecuciones de integración continua en cada fusión, pruebas de integración y pruebas de contrato frente a las interfaces, más los registros de compilación con marca de tiempo.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.8.2 Se logran los resultados de la integración: se define la estrategia de integración, las interfaces son coherentes y los defectos de integración se registran y se resuelven.

Cómo se demuestra: La definición del pipeline en el repositorio como estrategia de integración, ejecuciones fallidas seguidas de ejecuciones en verde como evidencia de resolución de defectos, y tickets de errores con el commit de corrección.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.9 Se realiza la Verificación: para cada requisito especificado existe evidencia verificable y trazable de que el software construido lo satisface realmente.[Documento](#)

Cómo se demuestra: Pruebas unitarias, de integración y de regresión automatizadas con un informe de pruebas del pipeline de integración continua, análisis estático y revisiones de código. El informe de pruebas por cada etiqueta de versión es la evidencia central.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.9.2 Se logran los resultados de la verificación: se definen criterios de verificación, las comprobaciones se ejecutan y las desviaciones se registran.

Cómo se demuestra: Umbrales de aceptación definidos (por ejemplo una cobertura de pruebas mínima, ausencia de hallazgos críticos), informes de pruebas archivados, y una incidencia por cada desviación.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.9.3 Se llevan a cabo las actividades de verificación: se planifican el alcance y los métodos de verificación, se ejecutan y los resultados se analizan.

Cómo se demuestra: Un plan de pruebas o una estrategia de pruebas como una página breve en el repositorio, la configuración del pipeline con sus etapas de verificación, y el análisis en la aprobación de la versión.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.10 Se realiza el proceso de Transición: el sistema se instala en su entorno de destino, se entrega listo para su uso, y se confirma su capacidad operativa.[Documento](#)

Cómo se demuestra: Un manual de despliegue (runbook), registros de despliegue del pipeline, pruebas de humo (smoke tests) tras la puesta en producción, y un plan de reversión documentado. Las notas de versión evidencian la entrega a los usuarios y a operaciones.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.11 Se realiza la Validación: en las condiciones de uso previstas se demuestra que el software satisface realmente las necesidades originales de las partes interesadas.[Documento](#)

Cómo se demuestra: Pruebas de aceptación con el cliente, una fase beta o piloto con comentarios de usuarios, y criterios de aceptación en el ticket marcados como cumplidos. Los datos de uso procedentes de la operación aportan un respaldo adicional a la validación.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.11.2 Se logran los resultados de la validación: se definen criterios de validación, la validación tiene lugar en un entorno representativo y las desviaciones se registran.

Cómo se demuestra: Un registro de aceptación con fecha y participantes, resultados del entorno de preproducción o piloto, y tickets de comentarios con su estado de resolución.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.12 El proceso de Operación está implantado: el sistema se opera según lo previsto, se supervisa y se da soporte a los usuarios.

Cómo se demuestra: Monitorización y alertas con umbrales definidos, tickets de incidentes con tiempos de respuesta, y un buzón de soporte o canal de tickets. Un manual de operaciones de pocas páginas basta para organizaciones pequeñas.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.12.2 Se logran los resultados de la operación: se definen la estrategia de operación y sus medidas, los incidentes se registran, y el rendimiento se evalúa frente a lo esperado.

Cómo se demuestra: Métricas de disponibilidad y de errores procedentes de la monitorización, registros de incidentes, y análisis posteriores a incidentes (postmortems) con las acciones derivadas de ellos.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.13

El proceso de Mantenimiento está implantado: las correcciones de defectos, las adaptaciones y las mejoras se registran, se evalúan, se implementan y se publican sin comprometer la integridad del sistema.

[Documento](#)

Cómo se demuestra: Tickets de errores y de cambios con prioridad y causa raíz, ramas de corrección urgente (hotfix), pruebas de regresión antes de la publicación, y notas de versión por cada versión de mantenimiento. Las actualizaciones periódicas de dependencias se evidencian mediante el historial del archivo de bloqueo (lockfile).

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.13.2

Se logran los resultados del mantenimiento: los cambios se analizan, se evalúa su impacto y se confirma la eficacia de la corrección.

Cómo se demuestra: Un análisis de impacto en el ticket, una ejecución de pruebas de regresión tras la corrección, y la confirmación por parte de quien reportó el problema. Las actualizaciones de seguridad se evidencian mediante informes de análisis de vulnerabilidades que muestren la fecha de corrección.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.14

El proceso de Baja está implantado: se planifica la retirada de sistemas o de partes de sistemas, los datos se migran o se eliminan, y se informa a las partes afectadas.

[Documento](#)

Cómo se demuestra: Un plan de retirada con fechas, evidencia de migración o eliminación de datos (incluidas las copias de seguridad), un registro de apagado de la infraestructura, y el aviso a los usuarios o clientes.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4.14.2

Se logran los resultados de la baja: se cumplen las obligaciones de conservación, los entornos se apagan y termina la responsabilidad sobre el sistema.

Cómo se demuestra: Un archivo de los datos sujetos a obligaciones de conservación con el periodo de conservación indicado, un ticket para el apagado de recursos con una fecha, y una confirmación de eliminación por parte del proveedor de hospedaje.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

A Conformidad y ajuste (Tailoring)

6

4.1

Se define el alcance de aplicación (apartado 4.1, General): se determina si la norma se aplica a un proyecto, a una organización o a un acuerdo, y qué procesos quedan cubiertos.

[Documento](#)

Cómo se demuestra: Una breve declaración de alcance en la documentación de proceso que nombre el proyecto, el producto y los grupos de procesos aplicados. Una página en el repositorio es suficiente.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.2

Se declara la variante de conformidad elegida: conformidad total con todos los resultados de los procesos reclamados, o conformidad total con todas las tareas de los procesos reclamados.

[Documento](#)

Cómo se demuestra: Una declaración de conformidad como documento independiente que nombre la variante y remita a la evidencia de cada proceso. Esto produce una autodeclaración, no un certificado.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.3

Cuando se reclama conformidad adaptada, el ajuste se documenta: se indica qué procesos, tareas o resultados se han omitido, modificado o añadido.

[Documento](#)

Cómo se demuestra: Una tabla de ajuste con las columnas proceso, decisión (adoptado, adaptado, omitido), justificación y riesgo. Para equipos pequeños, este documento es la evidencia decisiva.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

5 Los conceptos clave se aplican a la organización: la relación entre el modelo de ciclo de vida, la estructura del proyecto y los procesos en uso se presenta de forma comprensible.

Cómo se demuestra: Una página de resumen que relacione el enfoque elegido (por ejemplo iterativo, entrega continua) con los 30 procesos. Un diagrama más una tabla son suficientes.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

A.1 El ajuste se trata como una decisión deliberada: el ajuste está permitido, pero solo se aplica cuando las restricciones del proyecto lo justifican, y siempre se justifica.[Documento](#)

Cómo se demuestra: Una columna de justificación en la tabla de ajuste que se refiera al tamaño del proyecto, la clase de riesgo, la situación contractual o los requisitos legales. Las eliminaciones generales sin justificación no constituyen un ajuste válido.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

A.2 El proceso de ajuste se lleva a cabo: se identifican los factores que influyen, se involucra a las partes interesadas, y las decisiones de ajuste se toman y se documentan.[Documento](#)

Cómo se demuestra: Un registro o ticket de la decisión de ajuste que nombre a los roles involucrados, más la tabla de ajuste versionada en Git. Los efectos sobre los riesgos se anotan en el registro de riesgos.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

ISO/IEC 25010:2023, Ingeniería de sistemas y software, SQuaRE, Modelo de calidad del producto

Modelo de calidad para productos de software

40 requisitos. Para esta norma no existe un certificado acreditado. Versión de 07/2026.

ISO/IEC 25010:2023 es un modelo de calidad, no una norma de requisitos con cláusulas auditables, por lo que no puede certificarse mediante acreditación. El modelo sirve como marco de referencia para formular de manera completa los requisitos de calidad. La medición se rige por la norma ISO/IEC 25020 y la evaluación por la norma ISO/IEC 25040, y el resultado es un informe de evaluación trazable, no un certificado. No está permitido publicitar una supuesta certificación ISO 25010.

Empresa	Fecha	Elaborado por

1 Adecuación funcional3

1.1 El conjunto de funciones cubre todas las tareas especificadas y los objetivos de los usuarios para los que se emplea el producto (Compleitud funcional). Las brechas entre los requisitos documentados y las funciones implementadas son conocidas y están evaluadas.

Cómo se demuestra: La evidencia procede de una matriz de trazabilidad que vincula cada requisito con un caso de prueba, medible como porcentaje de cobertura de requisitos. En equipos pequeños basta con un gestor de incidencias con una etiqueta por requisito y una prueba de aceptación por objetivo de usuario.

AbiertoEn cursoCumplidoNo aplicable

Evidencia / justificación

1.2 Para las entradas definidas, el producto entrega resultados con el grado de precisión requerido (Corrección funcional). Las tolerancias y desviaciones admisibles están especificadas.

Cómo se demuestra: La cifra medible es la densidad de defectos por cada mil líneas de código o por función, respaldada por pruebas unitarias y de integración que comparan resultados esperados con resultados reales, además de las estadísticas de errores del gestor de incidencias. Para la lógica de cálculo, conserve conjuntos de datos de referencia con los resultados esperados.

AbiertoEn cursoCumplidoNo aplicable

Evidencia / justificación

1.3 Las funciones proporcionadas apoyan las tareas de los usuarios sin pasos innecesarios y se ajustan al uso previsto (Pertinencia funcional).

Cómo se demuestra: La evidencia procede de la tasa de éxito de tareas y del número de pasos por tarea de usuario en una prueba de usabilidad reducida con cinco participantes, registrada por tarea. Los datos de uso o las solicitudes de soporte que revelen rodeos completan el panorama.

AbiertoEn cursoCumplidoNo aplicable

Evidencia / justificación

2 Eficiencia de desempeño3

2.1 Los tiempos de respuesta, procesamiento y rendimiento se mantienen por debajo de los umbrales definidos (Comportamiento temporal). Los umbrales están definidos para los escenarios de uso típicos.

Cómo se demuestra: Es medible mediante el tiempo de carga y la latencia de interacción capturados con Lighthouse o Web Vitals (LCP, INP), junto con los tiempos de respuesta del servidor tomados de los registros en el percentil 95. Documente los umbrales por escenario en un breve presupuesto de rendimiento.

AbiertoEn cursoCumplidoNo aplicable

Evidencia / justificación

2.2 El consumo de tiempo de procesador, memoria, almacenamiento y ancho de banda de red se mantiene dentro de los límites definidos (Utilización de recursos).

Cómo se demuestra: La evidencia procede de valores de monitorización de CPU, RAM y volumen de datos bajo carga, junto con el tamaño del paquete distribuido tomado del informe de compilación. Para pequeñas empresas basta un registro de medición de una prueba de carga contra los límites del hardware objetivo.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

2.3 Se conoce y se respeta el número máximo de usuarios, transacciones o registros concurrentes (Capacidad).

Cómo se demuestra: La evidencia procede de una prueba de carga (por ejemplo con k6 o JMeter) con un punto de saturación documentado y el punto en el que las tasas de error comienzan a aumentar. Resuma el resultado en un breve informe de capacidad que indique el límite real y la demanda prevista.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

3 Compatibilidad

2

3.1 El producto funciona en un entorno compartido sin afectar a otros productos (Coexistencia).

Cómo se demuestra: La evidencia procede de una prueba realizada en el entorno objetivo con otro software ejecutándose en paralelo, registrando conflictos de puertos, uso de memoria y accesos a archivos. La contenedorización con límites de recursos definidos es la vía práctica para equipos pequeños.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

3.2 El intercambio de datos con otros sistemas funciona a través de interfaces y formatos definidos (Interoperabilidad).

Cómo se demuestra: La evidencia procede de una descripción de interfaz con control de versiones (OpenAPI o archivo de esquema) y de pruebas de contrato frente a esa descripción. Verifique los formatos de importación y exportación con archivos de muestra reales y conserve los resultados.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4 Capacidad de interacción

8

4.1 Los usuarios pueden reconocer sin ayuda externa si el producto se ajusta a sus necesidades (Reconocibilidad de la pertinencia). El propósito y el beneficio son evidentes en la interfaz y en los textos del nivel de entrada.

Cómo se demuestra: La evidencia procede de una breve prueba de primera impresión: los participantes describen el propósito del producto tras 30 segundos y se registra la tasa de acierto. Las capturas de pantalla de la página inicial y de las pantallas de incorporación respaldan el resultado.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.2 Los usuarios nuevos alcanzan las tareas principales dentro de un tiempo definido y con un número limitado de errores (Capacidad de aprendizaje).

Cómo se demuestra: Es medible como el tiempo hasta la primera tarea principal completada con éxito y los intentos fallidos por tarea, capturado en una prueba con cinco usuarios nuevos. Documente la incorporación, la visita guiada o los datos de demostración como medidas de apoyo.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.3 El manejo es eficiente y controlable, y las tareas recurrentes pueden llevarse a cabo sin rodeos (Operabilidad).

Cómo se demuestra: La evidencia procede de un análisis de recorrido de clics por tarea estándar (pasos previstos frente a pasos reales) y del manejo exclusivamente con teclado, sin ratón. El registro de la prueba de usabilidad con la tasa de abandono por tarea completa el expediente.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.4 El producto previene errores de manejo o los hace reversibles (Protección contra errores de usuario). Las acciones críticas requieren confirmación o pueden deshacerse.

Cómo se demuestra: La evidencia procede de casos de prueba con entradas inválidas y valores límite, junto con la comprobación de validación, diálogo de confirmación y función de deshacer. Evalúe las entradas erróneas del registro de producción y siga su frecuencia como métrica.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.5 La interacción está diseñada para resultar atractiva y mantiene a los usuarios motivados a continuar con la tarea (Compromiso del usuario).

Cómo se demuestra: Es medible mediante la tasa de retorno, la duración de la sesión y la tasa de finalización de los procesos iniciados, tomadas de estadísticas de uso respetuosas con la privacidad. Una breve encuesta de satisfacción con una escala de valoración basta como complemento, también para pequeñas empresas.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.6 El producto puede ser utilizado por personas con distintas capacidades, idiomas y contextos culturales (Inclusividad).

Cómo se demuestra: La evidencia procede de una comprobación frente a WCAG 2.2 nivel AA: relaciones de contraste, navegación con teclado, etiquetas para lectores de pantalla, verificadas con axe o la auditoría de accesibilidad de Lighthouse más una muestra manual. Demuestre la cobertura de idiomas y la localización mediante la completitud de los archivos de traducción.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.7 Los usuarios reciben ayuda adecuada a la tarea cuando la necesitan, también en situaciones de error y casos límite (Asistencia al usuario).

Cómo se demuestra: La evidencia procede del manual, de los textos de ayuda sensibles al contexto y de mensajes de error comprensibles que sugieren una acción siguiente. Métrica: la proporción de solicitudes de soporte que la ayuda existente podría haber resuelto, tomada del sistema de tickets.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

4.8 La interfaz se explica por sí misma: los estados, los efectos de las acciones y las entradas requeridas son reconocibles sin conocimientos previos (Capacidad de autodescripción).

Cómo se demuestra: La evidencia procede de una revisión de la interfaz frente a una lista de verificación: etiquetas con significado, estado del sistema visible, indicadores de progreso y carga, unidades en los campos numéricos. Un registro de observación de la prueba con usuarios refuerza el expediente.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

5 Fiabilidad

4

5.1 En operación normal el producto funciona sin fallos causados por defectos del software (Ausencia de fallos).

Cómo se demuestra: Es medible mediante la densidad de defectos y el tiempo medio entre fallos (MTBF) tomados de los registros de producción, y mediante la cobertura de pruebas como métrica preventiva. La monitorización de errores, como Sentry, aporta los datos base con poco esfuerzo.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

5.2 El producto es accesible y utilizable durante las franjas horarias acordadas (Disponibilidad). El objetivo de disponibilidad está definido.

Cómo se demuestra: La evidencia procede de la medición del tiempo de actividad mediante monitorización externa, expresada como disponibilidad en porcentaje mensual y comparada con el objetivo acordado. Registre las interrupciones con duración y causa en un registro de incidencias sencillo.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

5.3 Ante fallos de hardware o software, el producto se mantiene en un estado definido y utilizable (Tolerancia a fallos).

Cómo se demuestra: La evidencia procede de casos de prueba con fallos simulados (red caída, base de datos caída, sistema de terceros sin respuesta) y del comportamiento documentado, por ejemplo modo sin conexión, reintento o vista de respaldo. Documente los resultados en un registro de pruebas.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

5.4 Tras un fallo, los datos y el estado operativo pueden restablecerse dentro del tiempo definido (Capacidad de recuperación).

Cómo se demuestra: Las cifras medibles son el tiempo objetivo de recuperación (RTO) y la pérdida máxima de datos tolerable (RPO), evidenciadas mediante una prueba de restauración realizada de forma efectiva, con fecha y duración. Para pequeñas empresas basta con una prueba de restauración semestral con un registro breve.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6 Seguridad**6****6.1 Los datos son accesibles únicamente para personas y sistemas autorizados (Confidencialidad). Los derechos de acceso se conceden según el principio de necesidad de conocer.**[Documento](#)

Cómo se demuestra: La evidencia procede de un concepto de roles y permisos, del cifrado en tránsito y en reposo, y de pruebas que confirmen que el acceso se deniega sin autorización. Una verificación de cabeceras y de TLS de la aplicación completa el expediente.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.2 Se previenen o detectan los cambios no autorizados en programas y datos (Integridad).

Cómo se demuestra: La evidencia procede de sumas de comprobación o firmas de los artefactos, de la validación de entradas y de un análisis de dependencias (npm audit, Dependabot) que muestre que los hallazgos han sido corregidos. Las restricciones de la base de datos y los registros de migración refuerzan este punto.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.3 Las acciones realizadas no pueden negarse posteriormente (No repudio).[Documento](#)

Cómo se demuestra: La evidencia procede de registros protegidos contra manipulación con marca de tiempo y autor, y de firmas cuando la transacción tiene relevancia legal. Un destino de registro central y protegido contra escritura basta para pequeñas empresas.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.4 Las acciones en el sistema pueden atribuirse de forma inequívoca a una persona o servicio responsable (Responsabilidad).[Documento](#)

Cómo se demuestra: La evidencia procede de registros de auditoría que incluyen un identificador de usuario para cada acción relevante para la seguridad, combinados con la prohibición de cuentas compartidas. Verificación puntual: siga una transacción de principio a fin en el registro y documente el resultado.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.5 La identidad de usuarios y sistemas se verifica de forma fiable (Autenticidad).

Cómo se demuestra: La evidencia procede del método de autenticación, de la política de contraseñas y de la autenticación de dos factores para el acceso administrativo, así como de la validación de tokens y certificados. Los casos de prueba con tokens caducados y falsificados demuestran su eficacia.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

6.6 El producto resiste ataques y permanece operativo mientras está siendo atacado (Resistencia).

Cómo se demuestra: La evidencia procede de un análisis de vulnerabilidades, un análisis estático de código (SAST) y, cuando resulte proporcionado, una prueba de penetración con una lista de hallazgos y su estado de corrección. Para pequeñas empresas, un análisis automatizado en el flujo de trabajo más una breve prueba externa anual es la vía práctica.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

7 Mantenibilidad

5

7.1 El producto está formado por componentes separados cuya modificación afecta al menor número posible de otros componentes (Modularidad).

Cómo se demuestra: Es medible mediante el acoplamiento y la cohesión obtenidos de un análisis estático y mediante grafos de dependencias (madge, dependency-cruiser) que muestren la ausencia de ciclos. Documente los límites de los módulos en un breve esquema de arquitectura.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

7.2 Los componentes están diseñados de modo que puedan reutilizarse en más de un contexto (Reutilización).

Cómo se demuestra: La evidencia procede de bibliotecas o componentes compartidos con control de versiones propio y una interfaz documentada. Métrica: la proporción de componentes reutilizados y el nivel de duplicación de código obtenido del análisis estático.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

7.3 Las causas de los fallos y el impacto de los cambios previstos pueden determinarse de forma eficiente (Analizabilidad).

Cómo se demuestra: La evidencia procede del registro estructurado de eventos, de la monitorización de errores y de métricas de código legibles, como la complejidad ciclomática obtenida con SonarQube o ESLint. Métrica: tiempo medio hasta identificar la causa raíz, tomado del sistema de tickets.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

7.4 Los cambios pueden realizarse sin introducir nuevos defectos y sin degradar la calidad (Modificabilidad).

Cómo se demuestra: La evidencia procede de las pruebas de regresión en el flujo de trabajo y de la tasa de fallos por cambio, es decir, la proporción de defectos introducidos por los cambios. La revisión de código obligatoria y el control de versiones con commits trazables respaldan este punto.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

7.5 Pueden establecerse y ejecutarse criterios de prueba y pruebas eficaces para el producto y sus componentes (Capacidad de prueba).

Cómo se demuestra: Es medible mediante la cobertura de pruebas en porcentaje (líneas y ramas) obtenida del informe de cobertura y mediante la duración de ejecución del conjunto de pruebas. Una ejecución automatizada de pruebas en cada commit es la evidencia de que funciona.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

8 Flexibilidad

4

8.1 El producto puede adaptarse a entornos diferentes o cambiantes sin necesidad de reconstruirlo (Adaptabilidad).

Cómo se demuestra: La evidencia procede de la configuración mantenida fuera del código (variables de entorno, archivos de configuración) y de pruebas en varios entornos objetivo, navegadores o sistemas operativos dentro de una matriz de pruebas. Documente el resultado como una tabla de compatibilidad.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

8.2 El producto gestiona volúmenes crecientes o decrecientes de usuarios y datos sin una caída inaceptable del rendimiento (Escalabilidad).

Cómo se demuestra: La evidencia procede de pruebas de carga con un número creciente de usuarios, siguiendo el tiempo de respuesta y la tasa de error, complementadas con la estrategia de escalado (horizontal o vertical). Métrica: tiempo de respuesta con carga simple, doble y quintuple.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

8.3 El producto puede instalarse, actualizarse y desinstalarse de forma fiable en el entorno objetivo (Capacidad de instalación).

Cómo se demuestra: La evidencia procede de una guía de instalación reproducible y de una prueba de instalación automatizada en un sistema limpio, medida como tasa de éxito y tiempo requerido. Ejecute también una vez la desinstalación y la reversión, y registre el resultado.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

8.4 El producto puede sustituir a otro producto utilizado con el mismo propósito conservando los datos y los flujos de trabajo (Capacidad de sustitución).

Cómo se demuestra: La evidencia procede de funciones de importación y exportación en formatos abiertos y de una prueba de migración con datos reales del sistema anterior, verificada en cuanto a completitud y fidelidad de los datos. Mantenga el modelo de datos y los scripts de migración bajo control de versiones.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

9 Seguridad funcional

5

9.1 El rango de funcionamiento seguro está definido, y el producto detecta y gestiona cualquier salida de ese rango (Restricción operacional).

Cómo se demuestra: La evidencia procede de límites operativos documentados (valores umbral, estados admisibles) y de pruebas que verifiquen el comportamiento fuera de esos límites. Si el software no presenta ningún potencial de daño para personas o bienes, este punto normalmente no aplica; esa clasificación debe justificarse y documentarse, no omitirse sin más.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

9.2 Los peligros que pueden surgir del producto o de su uso se identifican y evalúan de forma sistemática (Identificación de riesgos).[Documento](#)

Cómo se demuestra: La evidencia procede de un análisis de peligros y riesgos que indique probabilidad, gravedad y medida mitigadora para cada peligro, práctico en forma de tabla dentro de la carpeta de calidad. No aplica únicamente con una justificación escrita de que no existe potencial de daño para personas o bienes.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

9.3 Ante una anomalía, el producto pasa a un estado seguro (Seguridad ante fallos).

Cómo se demuestra: La evidencia procede de casos de prueba con un fallo forzado y de la transición documentada al estado seguro, por ejemplo apagado, bloqueo o vuelta a un valor por defecto verificado. Para software sin potencial de daño, este punto normalmente no aplica, pero la justificación debe figurar igualmente en la evaluación.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

9.4 El producto advierte con la suficiente antelación de condiciones de peligro inminentes para que sea posible reaccionar (Advertencia de peligro).

Cómo se demuestra: La evidencia procede de un concepto de alarmas y advertencias con umbrales, tiempo de antelación y destinatarios, junto con un registro de pruebas de las advertencias activadas. Sin potencial de daño, este punto normalmente no aplica; la justificación debe quedar documentada.

Abierto En curso Cumplido No aplicable

Evidencia / justificación

9.5 La interacción con otros sistemas no genera nuevos peligros (Integración segura).

Cómo se demuestra: La evidencia procede de una revisión de riesgos de la interfaz y de pruebas de integración con los sistemas conectados, incluido el comportamiento cuando un sistema asociado falla o envía datos erróneos. Para software sin potencial de daño, este punto normalmente no aplica, y ello debe registrarse con una justificación.

Abierto En curso Cumplido No aplicable

Evidencia / justificación