# Políticas Corporativas para el Desarrollo de Software

*Alineadas con ISO/IEC 25010, ISO/IEC 25000 y ISO/IEC/IEEE 42010*

## 1. Introducción y Alcance

### 1.1 Objetivo del Documento

Este documento define el marco normativo, obligatorio y riguroso para el desarrollo, mantenimiento y gestión de software en la organización, asegurando la calidad y alineación con los estándares internacionales ISO/IEC 25010, ISO/IEC 25000 (SQuaRE) e ISO/IEC/IEEE 42010. Su adopción es obligatoria para equipos internos, proveedores y todos los involucrados en el ciclo de vida del software, siendo aplicable a proyectos críticos en entornos regulados como banca, finanzas, seguros, salud y cumplimiento normativo internacional.

### 1.2 Ámbito de Aplicación

* Aplicaciones internas, externas, móviles, web, APIs, microservicios, integraciones y sistemas de control.
* Proyectos gestionados por equipos de arquitectura, desarrollo, QA, DevSecOps, producto, soporte y auditoría.
* Incorporación en contratos, licitaciones, procesos de adquisiciones y proyectos tercerizados.
* Ambientes de desarrollo, pruebas, homologación, producción y continuidad de negocio.
* Tecnologías soportadas: infraestructuras físicas, virtuales y nativas en la nube; arquitecturas monolíticas, distribuidas, SOA, microservicios, eventos y contenedores.

### 1.3 Referencias Normativas

* ISO/IEC 25010:2011 – Quality model for software product
* ISO/IEC 25000:2014 – SQuaRE (Software Product Quality Requirements and Evaluation)
* ISO/IEC/IEEE 42010:2011 – Systems and Software Engineering – Architecture Description
* PCI DSS, GDPR, HIPAA, SOX, y regulaciones locales sectoriales donde sea aplicable

## 2. Políticas Fundamentales de Calidad (ISO/IEC 25010)

### 2.1 Funcionalidad

#### Generalidades

La funcionalidad es la capacidad del software para cumplir con los requisitos explícitos e implícitos en cada contexto de uso, así como los requerimientos legales o regulatorios. La validación funcional debe ser sistemática y documentada.

#### Políticas Específicas

* Todas las funcionalidades entregadas deben estar trazadas mediante matriz de requisitos.
* Pruebas de validación funcional; criterios de aceptación alineados a casos de uso o historias de usuario.
* Implementación de controles automáticos, validadores y gestión de excepciones para evitar desviaciones.
* Revisión y aprobación formal de la conformidad funcional antes de cada liberación.
* Documentar todo cambio de alcance funcional.
* Cumplimiento obligatorio de normativas sectoriales que afectan la funcionalidad del software.

#### Subcaracterísticas y requisitos

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Adecuación funcional | Matriz de trazabilidad por requisito | 4.2.1.1 |
| Exactitud | Pruebas unitarias y de integración asociadas | 4.2.1.1 |
| Interoperabilidad | Pruebas programadas de integración y gestión de APIs | 4.2.1.2 |
| Seguridad funcional | Uso de validadores, manejo de errores, roles | 4.2.1.3 |

### 2.2 Fiabilidad (Confiabilidad)

#### Generalidades

La confiabilidad refiere a la capacidad del software para mantener su nivel de desempeño bajo condiciones establecidas, tolerar fallos, recuperarse y operar continuamente sin incidentes.

#### Políticas Específicas

* Pruebas de estrés y estabilidad obligatorias, simular escenarios de carga extrema y contingencia.
* Implementar monitorización activa en producción, detección temprana de incidentes.
* Planes automáticos para recuperación ante fallos, con procedimientos de respaldo y restauración validados.
* Registro y análisis de incidencias para retroalimentación de diseño y mejora continua.
* Documentación de esquemas de mantenimiento preventivo y predictivo.

#### Subcaracterísticas y requisitos

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Madurez | Reportes de fallas y gestión de incidentes | 4.2.2.1 |
| Disponibilidad | SLA explícitos, pruebas de alta disponibilidad | 4.2.2.2 |
| Tolerancia a fallos | Redundancia y recuperación automática | 4.2.2.2 |
| Recuperabilidad | Procedimientos documentados y simulaciones regulares | 4.2.2.3 |

### 2.3 Usabilidad

#### Generalidades

Incluye todos los aspectos que influyen en la experiencia del usuario: operabilidad, facilidad de aprendizaje, estética, personalización y accesibilidad.

#### Políticas Específicas

* Pruebas de experiencia de usuario (UX) obligatorias en cada entrega mayor.
* Cumplimiento de normas WCAG 2.1 AA para aplicaciones web y móviles.
* Capacitación, documentación de usuario y asistencias integradas.
* Monitoreo de tiempos de aprendizaje y tasa de error de usuarios reales.
* Rediseño continuo basado en métricas de satisfacción y accesibilidad auditadas.

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Operabilidad | Manuales y ayuda online/documentada | 4.2.3.1 |
| Estética | Guías corporativas de UI y branding | 4.2.3.2 |
| Accesibilidad | Validación por expertos externos | 4.2.3.3 |
| Facilidad de aprendizaje | Medición y reportes periódicos | 4.2.3.4 |

### 2.4 Eficiencia

#### Generalidades

La eficiencia se refiere al uso óptimo de recursos, alta capacidad de respuesta y comportamientos adecuados bajo niveles de carga planificados.

#### Políticas Específicas

* Monitoreo y registro sistemático de uso de CPU, memoria y red en todos los ambientes.
* Pruebas de desempeño: respuesta, escalabilidad y optimización obligatoria.
* Plan de contingencias ante degradación, escalamiento automático.
* Tiempos de respuesta alineados con SLA predefinidos por negocios y clientes.
* Ajustes automáticos según métricas de uso y carga.

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Comportamiento bajo carga | Pruebas de estrés y simulación | 4.2.4.1 |
| Utilización de recursos | Perfilado y optimización | 4.2.4.2 |

### 2.5 Mantenibilidad

#### Generalidades

La capacidad para ser analizado, modificado y evolucionado eficazmente con mínimo riesgo e impacto.

#### Políticas Específicas

* Modularidad estricta, cada módulo con responsabilidad única y documentación.
* Uso obligatorio de herramientas de análisis estático de código en integración continua.
* Gestión y remediación de deuda técnica; registro y visualización en backlog accesible.
* Requisitos de refactorización, validación y aprobación con métricas de calidad (Maintainability Index ≥ 75).
* Pruebas regulares de reemplazo, actualización y mejora de componentes.

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Modularidad | Separación clara de capas y microservicios | 4.2.5.1 |
| Reusabilidad | Registro de componentes reutilizables | 4.2.5.3 |
| Analizabilidad | Automatización CI/CD, reportes y dashboards | 4.2.5.4 |
| Modificabilidad | Validación de cambios: pruebas automáticas | 4.2.5.2 |

### 2.6 Portabilidad

#### Generalidades

Posibilidad de ser transferido y operar en diferentes entornos y plataformas sin rediseño significativo.

#### Políticas Específicas

* Proveer scripts y documentación detallada para despliegue en sistemas soportados.
* Implementar procesos automatizados de migración y conversión de datos.
* Validación y pruebas en entornos multiplataforma, emuladores y contenedores.
* Gestión de compatibilidad y coexistencia con otros sistemas corporativos.
* Planes de desinstalación, migración y contingencia obligatorios.

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Adaptabilidad | Pruebas multicapa, homologación multiplataforma | 4.2.6.1 |
| Instalabilidad | Despliegue automatizado, rollback | 4.2.6.2 |
| Coexistencia | Matriz de compatibilidad, reporte de conflictos | 4.2.6.3 |
| Reemplazabilidad | Planificación de sustitución documentada | 4.2.6.4 |

### 2.7 Seguridad

#### Generalidades

Protección de la información, mecanismos de prevención, detección y recuperación ante amenazas y vulnerabilidades.

#### Políticas Específicas

* Gestión estricta de acceso, autenticación y autorización: integración con IdP, uso de MFA.
* Cifrado robusto de datos en tránsito y reposo: TLS, AES, hashing y controles de integridad.
* Registro de eventos de seguridad, detección proactiva y alarmas continuas.
* Implementar controles de no repudio, firmas digitales y análisis de logs.
* Ejecución continua de análisis SAST (análisis estático de seguridad) y SCA (escaneo de componentes y librerías).
* Pruebas de penetración regulares y simulaciones de ataques.

|  |  |  |
| --- | --- | --- |
| Subcaracterística | Política | Referencia ISO 25010 |
| Confidencialidad | Cifrado y segregación de datos sensibles | 4.2.7.1 |
| Integridad | Validación de autenticidad y completos de datos | 4.2.7.2 |
| No repudio | Registro y auditoría de acciones | 4.2.7.4 |
| Autenticación | MFA, federación, administración de credenciales | 4.2.7.5 |
| Autorización | Roles y mínimos privilegios | 4.2.7.6 |

## 3. Políticas de Arquitectura de Software (ISO/IEC/IEEE 42010)

### 3.1 Identificación de Stakeholders y Gestión de Concerns

* Mapear exhaustivamente stakeholders internos y externos: negocio, usuarios, clientes, auditoría, soporte, seguridad, transformación digital, devops, operación, proveedores, organismos reguladores.
* Documentar las preocupaciones ('concerns') de cada stakeholder: funcionalidad, escalabilidad, disponibilidad, seguridad, mantenimiento, soporte, costo, cumplimiento, interoperabilidad, usabilidad.
* Revisiones formales y revalidación de stakeholders y concerns en cada hito de arquitectura y liberación.

### 3.2 Documentación de Vistas Arquitectónicas

* Implementar y mantener diagramas y descripciones comprensivas para las siguientes vistas en cada proyecto:
  + Vistas lógicas: componentes, relaciones, dependencias, patrones y diagramas UML.
  + Vistas de desarrollo: estructura del repositorio, organización de carpetas, componentes, dependencias, prácticas de integración continua, estilos de codificación.
  + Vistas de despliegue: infraestructura física y virtual, mapping de contenedores, redes, balanceadores, tolerancia a fallos.
  + Vista de procesos: flujos de ejecución, concurrencia, orquestación, tolerancia a fallos, colas y eventos, automatización.
  + Vista de casos de uso: contexto del negocio, journeys de usuario, flujos críticos y escenarios regulatorios.
* Actualización obligatoria de vistas en ciclo de releases, formalización de artefactos y trazabilidad de cambios.

### 3.3 Uso de Viewpoints Estandarizados

* Adopción del modelo 4+1 view (lógica, desarrollo, proceso, físico, casos de uso) como referencia estructural, extendido según requerimientos del proyecto y regulaciones sectoriales.
* Los viewpoints deben ser estructurados por plantillas institucionales aceptadas y aprobados por Arquitecto Responsable.
* Mantener correlación entre vistas y concerns mediante matrices de trazabilidad.

### 3.4 Gestión de Decisiones Arquitectónicas (ADR)

* Registro obligatorio de toda decisión arquitectónica (Architecture Decision Records — ADR).
* Cada ADR debe contener: contexto, problema, opciones, análisis de pros/contras, decisión tomada y consecuencias esperadas.
* Repositorio centralizado y versionado de ADR, accesible para auditoría y consulta interproyectos.
* Trazabilidad entre ADR y cambios en vistas, requisitos y políticas.
* Notificación, revisión y aprobación formal de las ADR en comités de gobernanza.

## 4. Procesos de Desarrollo y Control de Calidad

### 4.1 Revisión de Código

* Adopción obligatoria de revisiones por pares para todo merge o integración en ramas principales (feature, release, hotfix).
* Uso de checklist de calidad institucional, cubriendo todas las características del modelo ISO 25010 y prácticas de seguridad recomendadas.
* Documentar y reportar observaciones, correcciones y métricas de calidad en cada revisión.
* Validación previa de ambientes y entrega a QA.

### 4.2 Pruebas y Cobertura

* Implementación obligatoria de pruebas unitarias, de integración, regresión y aceptación automatizadas.
* Cobertura mínima de pruebas automatizadas: superior o igual al 80% de líneas y casos críticos del software.
* Uso de herramientas automáticas integradas en CI/CD para medición, reporte y seguimiento de cobertura.
* Pruebas manuales complementarias en escenarios regulatorios o funcionales críticos.

### 4.3 Análisis de Código y Gestión de Dependencias

* Ejecución sistemática de análisis estático de código fuente (SAST) para identificación temprana de defectos, vulnerabilidades y riesgos.
* Integración de escaneo de vulnerabilidades de dependencias externas (SCA), gestionando remediación y actualizaciones proactivamente.
* Reporte detallado y mitigación de vulnerabilidades como prerrequisito de liberación.

### 4.4 Gestión de Deuda Técnica y Métricas de Mantenibilidad

* Mantenimiento de registro proactivo, visualizable y priorizado de deuda técnica.
* Análisis continuo del Maintainability Index (MI) en todos los repositorios principales; ninguna liberación debe tener MI < 75.
* Planes de remediación y refactorización de deuda técnica con compromisos definidos en planning.
* Revisión obligatoria en retrospectivas, auditorías y comités de calidad cada trimestre.

## 5. Gobernanza, Roles y Cumplimiento

### 5.1 Roles, Responsabilidades y Interacción

|  |  |
| --- | --- |
| Rol | Responsabilidad Principal |
| Arquitecto de Software | Liderar definición, documentación y evolución de arquitectura, aprobar vistas y ADRs. |
| QA Lead | Establecer y auditar procesos de pruebas, cobertura, revisión de calidad y reportes QA. |
| DevSecOps Engineer | Configurar pipelines CI/CD, ejecutar y reportar análisis SAST/SCA, veedora de controles y despliegues seguros. |
| Auditor Interno | Realizar auditoría normativa/sectorial, validar matrices de trazabilidad, emitir reportes y recomendaciones. |
| Líder de Producto | Gestionar requerimientos, validación funcional y mapeo de concerns, priorización de entregables. |

### 5.2 Procesos de Auditoría y Cumplimiento

* Auditorías internas y externas cada trimestre: validación de cumplimiento de políticas, controles y procedimientos documentados, revisión de matrices de trazabilidad y anomalías.
* Generación obligatoria de reportes, KPIs y planes de remediación por área responsable.
* Registro anual de conformidad y acciones de mejora continua, con reporte formal a la dirección.
* Revisión, actualización y revalidación obligatoria de las políticas y matrices ante cambios normativos, regulatorios o tecnológicos relevantes.

### 5.3 Matrices de Trazabilidad y Control Normativo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Política / Control | ISO 25010 | ISO/IEC/IEEE 42010 | Proceso / Ciclo | Relación Normativa |
| Checklist revisión de código | X |  | Revisión dev | 4.2.5/4.2.6 ISO 25010 |
| Documentación y vistas arquitect. |  | X | Arquitectura | Cláusula 5 y 7.1 ISO 42010 |
| Registro y aprobación de ADRs |  | X | Comité ADR | Cláusula 7.2 ISO 42010 |
| Pruebas automatizadas cobertura >80% | X |  | QA | 4.2.1/4.2.2 ISO 25010 |
| SAST / SCA | X |  | CI/CD, DevSecOps | 4.2.7 ISO 25010 |
| Auditorías trimestrales | X | X | Auditoría/QC | 6.1 ISO 25010, 7.5 ISO 42010 |
| Deuda técnica y MI ≥75 | X |  | QA/Refactor | 4.2.5 ISO 25010 |

### 5.4 Métodos de Comunicación, Notificación y Seguimiento

* Todos los cambios en políticas, controles, matrices y ADR deben ser comunicados vía canales oficiales y repositorios institucionales.
* Programar sesiones de capacitación, actualización normativa y workshops especializados cada semestre.
* Establecer sistemas de notificación, alertas y reporte automático de incumplimientos, con escalamiento a líderes responsables.

## 6. Anexos y Plantillas

### 6.1 Plantilla de Architecture Decision Record (ADR)

# Decisión Arquitectónica [ID]  
## Fecha  
## Estado (Propuesto / Aprobado / Obsoleto)  
## Contexto  
## Problema / Concerns  
## Opciones Consideradas  
## Análisis (pros/contras)  
## Decisión Tomada  
## Consecuencias  
## Stakeholders involucrados  
## Referencias / Documentos Relacionados

### 6.2 Checklist para Revisión de Calidad

|  |  |  |
| --- | --- | --- |
| Ítem Evaluado | Cumple | Observaciones |
| Adecuación funcional |  |  |
| Cobertura de pruebas |  |  |
| Cumplimiento de seguridad |  |  |
| Documentación |  |  |
| Mantenibilidad del código |  |  |
| Portabilidad |  |  |
| Accesibilidad y usabilidad |  |  |
| Trazabilidad |  |  |
| Cumplimiento normativo |  |  |

### 6.3 Tabla de Correspondencia ISO 25010 ↔ Políticas

|  |  |
| --- | --- |
| Característica ISO 25010 | Políticas / Medidas Normativas |
| Funcionalidad | Matriz requisitos, validación, revisiones, controles legales |
| Fiabilidad/Confiabilidad | Pruebas estrés, monitorización, planes de recuperación |
| Usabilidad | WCAG, UX, documentación, métricas de satisfacción |
| Eficiencia | Pruebas carga, monitoreo recursos, optimización CI/CD |
| Mantenibilidad | MI ≥75, modularidad, análisis estático, deuda técnica |
| Portabilidad | Despliegue multiplataforma, scripts, validaciones |
| Seguridad | MFA, cifrado, SAST/SCA, logs, monitoreo proactivo |

### 6.4 Glosario de Términos

* **ADR**: Architecture Decision Record
* **SAST/SCA**: Static Application Security Testing / Software Composition Analysis
* **Maintainability Index (MI)**: Métrica para evaluar la mantenibilidad de software
* **MFA**: Autenticación multifactor
* **Stakeholder**: Parte interesada o afectada
* **CI/CD**: Integración y Entrega Continua
* **SLA**: Acuerdo de nivel de servicio

## 7. Referencias Normativas y Documentales

* ISO/IEC 25010:2011 "Systems and Software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models"
* ISO/IEC 25000:2014 "SQuaRE -- Software Product Quality Requirements and Evaluation"
* ISO/IEC/IEEE 42010:2011 "Systems and software engineering -- Architecture description"
* PCI DSS, GDPR, HIPAA, SOX
* Documentos y guías sectoriales y locales vigentes

**Advertencia:** Este documento es de uso corporativo exclusivo, debe ser revisado y aprobado por la Oficina de Gobernanza de TI y está sujeto a auditoría interna y externa según lo determinen las exigencias regulatorias y del negocio.

**Formato institucional, estructurado para adopción y auditorías en banca, finanzas, seguros, salud y contextos regulados de alta exigencia.**