

**SERVICIO NACIONAL DE APRENDIZAJE  
SENA**

PLAN DE ASEGURAMIENTO DE LA  
CALIDAD DEL SOFTWARE  
(Software Quality Assurance Plan - SQA)

Aprendiz: Mateo Algarra

Curso: Calidad del Software – RAP2

**2 de diciembre de 2025**

## **1. PROPÓSITO DEL PLAN DE SQA**

El Plan de SQA establece los mecanismos, procedimientos y responsabilidades para garantizar que el Sistema de Gestión Clínica desarrollado por SoftSena cumpla con los estándares de calidad y requisitos establecidos por el cliente.

### **1.1. Alcance**

Cubre todas las fases del ciclo de vida del proyecto, incluyendo los módulos de: Ingreso/Hospitalización, Información del Paciente, Habitaciones y Camas, Medicamentos y Materiales, Costos de Hospitalización, y Consulta de Disponibilidad.

### **1.2. Objetivos**

1. Asegurar cumplimiento de requisitos y estándares establecidos
2. Verificar adherencia a procesos y metodologías definidos
3. Realizar evaluaciones y auditorías periódicas
4. Garantizar trazabilidad de defectos y su resolución

## 2. REFERENCIAS

El plan se fundamenta en los siguientes estándares internacionales:

- IEEE 730-2002: Planes de Aseguramiento de Calidad del Software
- IEEE 829-1998: Documentación de Pruebas de Software
- IEEE 830-1998: Especificaciones de Requisitos de Software
- IEEE 1012-2004: Verificación y Validación
- ISO 9001: Sistemas de Gestión de Calidad
- ISO/IEC 12207: Procesos del Ciclo de Vida del Software
- ISO/IEC 25000 (SQuaRE): Requisitos y Evaluación de Calidad

## 3. GESTIÓN

### 3.1. Estructura del Equipo SQA

ROL	RESPONSABILIDADES
<b>Líder SQA</b>	Coordinar equipo, asegurar cumplimiento del plan, reportar estado de calidad
<b>Admin. Desarrollo</b>	Controlar avance en diseño y desarrollo, verificar estándares de codificación
<b>Admin. Calidad</b>	Definir métricas, realizar auditorías, gestionar no conformidades

## 4. ACTIVIDADES

### 4.1. Revisión de Productos

Se revisarán: Especificación de Requisitos, Documentos de Diseño, Código Fuente, Casos de Prueba, Documentación de Usuario y Manuales Técnicos.

### 4.2. Revisiones Técnicas Formales

- Revisión de Requisitos (Fin de Análisis)
- Revisión de Diseño Preliminar (Semana 8)
- Revisión de Diseño Crítico (Semana 12)
- Auditoría Funcional (Pre-entrega)
- Auditoría Física (Entrega Final)

### 4.3. Control de Desviaciones

Proceso: Documentar desviación → Notificar al equipo → Plan de acciones correctivas → Escalamiento si no se resuelve → Verificar implementación.

## 5. RESPONSABLES

Líder SQA: Aprobar plan, coordinar actividades, reportar a dirección.  
 Admin. Desarrollo: Supervisar revisiones de código, validar documentación. Admin. Calidad: Definir métricas, realizar auditorías, gestionar defectos.

## 6. DOCUMENTACIÓN

- Requerimientos: SRS (IEEE 830), Casos de Uso, Matriz de Trazabilidad
- Diseño: Arquitectura, Diagramas UML, Diseño de BD, APIs
- Plan V&V: Plan de Pruebas (IEEE 829), Casos de Prueba, Reportes
- Usuario: Manuales de Usuario, Guías de Instalación, Tutoriales
- Configuración: Plan SCM, Registro de CI, Control de Cambios

## 7. GESTIÓN DE CONFIGURACIÓN

### 7.1. Herramientas

Sistema: Git/GitLab Enterprise. Estrategia: GitFlow. Ramas: main (producción), develop (desarrollo), feature/\*, hotfix/\*, release/\*.

### 7.2. Control de Cambios

Flujo: Solicitud (CR en Jira) → Evaluación (CCB) → Aprobación → Implementación (rama feature) → Revisión → Verificación → Merge.

### 7.3. Auditorías

- Funcional (FCA): Al completar cada sprint
- Física (PCA): Previo a cada release

## 8. ESTÁNDARES, PRÁCTICAS Y MÉTRICAS

### 8.1. Métricas Principales

MÉTRICA	FÓRMULA	META
Densidad de Defectos	Defectos / KLOC	$\leq 2$
Cobertura de Código	(Líneas cubiertas / Total) × 100	$\geq 80\%$
Adherencia al Cronograma	(Tareas a tiempo / Total) × 100	$\geq 85\%$
Tasa de Éxito de Pruebas	(Pruebas pasadas / Total) × 100	$\geq 95\%$

### 8.2. Estándares Aplicados

IEEE 730-2002, IEEE 829-1998, IEEE 830-1998, IEEE 1012-2004, ISO 9001, ISO/IEC 12207, ISO/IEC 25010.

## 9. REVISIONES Y AUDITORÍAS

TIPO	MOMENTO	ALCANCE	RESPONSABLE
PDR	Semana 8	Arquitectura	Líder SQA
CDR	Semana 12	Diseño Detallado	Admin. Desarrollo
FCA	Cada Sprint	Funcionalidades	Admin. Calidad
PCA	Pre-Release	Sistema Completo	Líder SQA

## 10. VERIFICACIÓN

- Inspecciones de Código: Fagan Inspection, cada commit significativo
- Análisis Estático: SonarQube Enterprise (automático en CI/CD)
- Walkthroughs de Diseño: Post-diseño de componentes
- Pruebas de Penetración: OWASP ZAP, antes de cada release

## 11. REPORTE DE PROBLEMAS Y ACCIONES CORRECTIVAS

### 11.1. Flujo de Defectos

Detección → Registro (Jira) → Clasificación (Severidad) → Asignación → Análisis de Causa Raíz → Corrección (rama bugfix) → Verificación (QA) → Cierre.

SEVERIDAD	DESCRIPCIÓN	SLA
<b>Crítica</b>	Sistema inoperativo	4 horas
<b>Alta</b>	Funcionalidad principal afectada	24 horas
<b>Media</b>	Funcionalidad con workaround	1 semana
<b>Baja</b>	Mejora cosmética	Siguiente sprint

## 12. GESTIÓN DE RIESGOS

**PLAN DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE**

ID	RIESGO	PROB.	MITIGACIÓN
R-01	Cambios frecuentes en requisitos	Media	Prototipos, CCB, buffer 20%
R-02	Defectos críticos en producción	Media	Cobertura 80%, UAT, rollback
R-03	Vulnerabilidades de seguridad	Media	SAST/DAST, pentesting
R-04	Rendimiento inadecuado	Media	SLAs, JMeter, APM