

UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO
Facultad de Ingeniería en Ciencias de la Computación y
Telecomunicaciones



PLAN DE ASEGURAMIENTO DE CALIDAD DEL
SOFTWARE (SQAP)

Estudiantes:

Univ. Montaña Claros Wilson Brandon 220001911

Univ. Zamo Adunay Clemente Junior 217057251

Materia: Ingeniería en Software II

Docente: Ing. Martínez Canedo Rolando Antonio

Grupo: #24

Fecha: 05/06/2025

Semestre I - 2025
Santa Cruz de la Sierra – Bolivia
UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO
Facultad de Ingeniería en Ciencias de la Computación y
Telecomunicaciones



**PLAN DE ASEGURAMIENTO DE CALIDAD
DEL SOFTWARE (SQAP)**

Estudiantes:

Univ. Montaña Claros Wilson Brandon 220001911

Univ. Zamo Adunay Clemente Junior 217057251

Materia: Ingeniería en Software II

Docente: Ing. Martínez Canedo Rolando Antonio

Grupo: #24

Fecha: 05/06/2025

Semestre I - 2025
Santa Cruz de la Sierra – Bolivia

Capítulo 1: Manual De Calidad SQAP	7
1.1 Introducción.....	7
1.1.1 Antecedentes de la empresa	7
1.2 Objetivos	9
1.2.1 Objetivo General	9
1.2.2 Objetivos Específicos.....	9
1.3 Misión	10
1.4 Visión.....	10
1.5 Valores	10
1.6 Estándares.....	11
1.7 Organigrama	13
1.8 Políticas de Calidad	13
1.9 Gestión	15
1.9.1 Organización	15
1.9.2 Tareas y Actividades.....	17
1.10 Roles y Responsabilidades	22
1.10.1 Responsabilidades del grupo de desarrollo	23
1.10.2 Responsabilidades del Cliente	23
1.10.3 Responsabilidades de la SQA	24
1.11 Mediciones.....	28
1.11.1 Tipos de métricas empleadas.....	28
1.11.2 Recolección y análisis de métricas	30
1.11.3 Trazabilidad y acción correctiva.....	30
1.12 Evolución del SQAP	30
1.12.1 Control de cambio.....	31
1.12.2 Cuando hacer modificaciones.....	31
1.12.3 Implementación de modificaciones.....	32
1.12.4 Publicación del SQAP	33
1.13 Documentación.....	33
1.13.1 Propósito.....	34
1.13.2 Documentación mínima requerida	35
1.13.3 Plan de Verificación y Validación	36

1.13.4	Documentación de Usuario	38
1.14	Estándares, Prácticas y Convenciones	39
1.14.1	Estándar de documentación	39
1.14.2	Estándar de verificación y prácticas.....	40
1.14.3	Estándar de codificación	41
1.14.4	Responsable de verificar el rendimiento	42
1.15	Revisiones y Auditoría.....	43
1.15.1	Objetivo.....	43
1.15.2	Requerimientos mínimos.....	43
1.15.3	Tipos de Auditoria	44
1.15.4	Auditoría Administrativa del SQAP	45
1.15.5	Responsabilidades y Seguimiento.....	46
1.16	Verificación	46
1.16.1	Propósito.....	46
1.16.2	Alcance de la Verificación Complementaria	47
1.16.3	Métodos de Verificación según IEEE 730.....	48
1.16.4	Registro y Control de Verificación Adicional.....	48
1.16.5	. Responsabilidades.....	49
1.17	Reporte De Problemas y Acciones Correctivas	49
1.17.1	Enfoque según IEEE 730.....	49
1.17.2	Tipos de Informes Utilizados.....	50
1.17.3	Proceso de Gestión de Problemas	52
1.17.4	Registro y Trazabilidad	52
1.17.5	Indicadores de Control.....	53
1.17.6	Informe de verificación y validación	53
1.17.7	Reporte sumario de fase V&V.....	54
1.18	Herramientas, Técnicas y Metodología	55
1.18.1	Herramientas	55
1.18.2	Técnicas.....	56
1.18.3	Metodología.....	57
1.19	Control de Código	59
1.19.1	Software sujeto a control.....	59

1.19.2	Identificación, etiquetado y versionado	59
1.19.3	Repositorio y ubicación física.....	60
1.19.4	Procedimientos de construcción de nuevas versiones.....	61
1.19.5	Control de documentación relacionada	61
1.19.6	Alineación con IEEE 730	61
1.20	Control de Medios.	62
1.20.1	Medios de Comunicación Interna	62
1.20.2	Medios de Comunicación Externa	63
1.20.3	Control de Información Técnica y Documental	64
1.20.4	Seguridad y Trazabilidad en los Medios Utilizados.....	65
1.21	Control de Proveedores	65
1.21.1	Tipos de proveedores identificados.....	66
1.21.2	Procedimiento para la selección de proveedores.....	66
1.21.3	Control de calidad y seguimiento	67
1.21.4	Documentación relacionada a proveedores	68
1.22	Gestión de Registros	68
1.22.1	Tipos de registros gestionados.....	69
1.22.2	Requisitos para la administración de registros.....	70
1.22.3	Procedimientos asociados a la gestión de registros	70
1.23	Capacitación	71
1.23.1	Objetivos de la capacitación	71
1.23.2	Modalidades de capacitación	72
1.23.3	Temáticas prioritarias	73
1.23.4	Registro y seguimiento de la capacitación.....	73
1.23.5	Evaluación de efectividad	74
1.24	Gestión de Riesgos	74
1.24.1	Proceso de Gestión de Riesgos	74
1.24.2	Tipos de riesgos más comunes en ShopMind	76
1.24.3	Documentación de riesgos.....	77



UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO
 Facultad de Ingeniería en Ciencias de la Computación y
 Telecomunicaciones



ShopMind

“El copiloto inteligente para tu negocio.”

PLAN DE ASEGURAMIENTO DE CALIDAD DEL SOFTWARE (SQAP)

Versión 1.1

Historia de revisiones

Fecha	Versión	Descripción	Autor/es
10/05/2025	1.0	Creación del documento.	Montaño Claros Wilson Brandon Zamo Adunay Clemente Junior
04/06/2025	1.1	Pequeñas correcciones.	Montaño Claros Wilson Brandon Zamo Adunay Clemente Junior

Capítulo 1: Manual De Calidad SQAP

1.1 Introducción

Este documento tiene como finalidad establecer las bases del aseguramiento de calidad del software desarrollado por ShopMind, definiendo los lineamientos técnicos y organizacionales que se aplicarán durante todo el ciclo de vida del proyecto. Dado que ShopMind ofrece soluciones SaaS avanzadas para comercio electrónico, es fundamental garantizar que cada componente del sistema cumpla con criterios estrictos de calidad, fiabilidad y rendimiento.

La calidad no es asumida como una etapa aislada, sino como un principio transversal que guía cada fase del desarrollo. Por ello, se adopta un enfoque preventivo, basado en procesos documentados, estándares reconocidos y mecanismos de evaluación continua.

Este manual describe el conjunto de actividades, prácticas y responsabilidades asignadas al equipo encargado de calidad, denominado SQA (Software Quality Assurance). Entre sus funciones están la supervisión del cumplimiento de los procedimientos establecidos, la ejecución de revisiones sistemáticas, y la validación de entregables en función de métricas y criterios definidos.

Además, se contemplan herramientas y técnicas que permitirán mantener trazabilidad, minimizar defectos y asegurar que el software de ShopMind se entregue conforme a las expectativas del cliente, dentro del marco de calidad que caracteriza a la organización.

1.1.1 Antecedentes de la empresa

Esta Startup Tecnológica de reciente creación, fundada en 2024 por un equipo multidisciplinario de emprendedores con una visión clara: transformar el comercio electrónico a

través de soluciones inteligentes, accesibles y escalables. La empresa nace con el propósito de integrar herramientas de Inteligencia Artificial (IA) y Business Intelligence (BI) dentro de plataformas SaaS de e-commerce, ofreciendo a las tiendas digitales una ventaja competitiva basada en datos y automatización.

Desde sus inicios, ShopMind ha mantenido el compromiso de entregar software de alta calidad que responda a las necesidades reales de los negocios en crecimiento. La innovación, la eficiencia operativa y la orientación al cliente son pilares fundamentales en cada proyecto desarrollado.

Actualmente, está conformado por profesionales especializados en las siguientes áreas clave:

- Análisis de requisitos y procesos de negocio
- Diseño de interfaces y experiencia de usuario (UX/UI)
- Desarrollo e implementación de soluciones software
- Integración de módulos de IA y BI para toma de decisiones

Este equipo altamente capacitado permite abordar proyectos complejos con agilidad, garantizando resultados sólidos y alineados con las mejores prácticas del desarrollo moderno.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar soluciones tecnológicas inteligentes para el sector e-commerce, garantizando altos estándares de calidad en cada etapa del desarrollo, con el fin de generar valor real a los clientes y contribuir al crecimiento y competitividad de sus negocios digitales.

1.2.2 Objetivos Específicos

- Comprender a profundidad los requerimientos funcionales y estratégicos de los clientes mediante una comunicación constante y efectiva.
- Diseñar e implementar soluciones de software que integren Inteligencia de Negocio (BI) e Inteligencia Artificial (IA), bajo criterios rigurosos de calidad técnica y usabilidad.
- Establecer y aplicar métricas y criterios de calidad verificables en todas las fases del ciclo de vida del desarrollo de software.
- Documentar de forma detallada las actividades, roles, procesos y procedimientos relacionados con el aseguramiento de la calidad, asegurando trazabilidad y mejora continua.
- Promover una cultura organizacional centrada en la calidad, la innovación y la satisfacción del cliente como principios fundamentales de ShopMind.

1.3 Misión

La misión de ShopMind es ofrecer soluciones tecnológicas inteligentes y accesibles que impulsen el crecimiento del comercio electrónico, mediante el desarrollo de plataformas SaaS que integren Inteligencia Artificial y Business Intelligence, asegurando altos estándares de calidad, eficiencia operativa y una experiencia centrada en el cliente.

1.4 Visión

La visión de esta Startup es consolidarse como un aliado estratégico y confiable para negocios digitales, liderando el mercado de soluciones inteligentes para comercio electrónico. Se aspira a transformar la forma en que las tiendas online operan, facilitando su crecimiento mediante herramientas basadas en Inteligencia Artificial, Business Intelligence y plataformas SaaS de alto rendimiento.

1.5 Valores

- **Compromiso:** ShopMind se compromete a entender a fondo los desafíos del e-commerce actual para ofrecer soluciones que realmente generen valor y mejoren la rentabilidad de sus clientes.
- **Calidad:** Cada línea de código, cada interfaz y cada entrega están orientadas a cumplir con altos estándares de calidad, confiabilidad y escalabilidad.
- **Colaboración:** Se cultiva una relación cercana y proactiva con cada cliente, trabajando como un verdadero socio tecnológico durante todas las fases del proyecto.

- **Trabajo en equipo:** El talento colectivo es el motor de ShopMind. Se promueve un entorno colaborativo donde cada integrante puede aportar, crecer y destacar.
- **Integridad:** La transparencia, la ética profesional y la honestidad son principios fundamentales que rigen las decisiones internas y las relaciones externas.
- **Respeto:** Se valora la diversidad, se escucha activamente y se promueve un ambiente inclusivo, seguro y respetuoso para todos.
- **Aprendizaje continuo:** Se apuesta por la mejora constante, impulsando la capacitación del equipo y la adopción de nuevas tecnologías como parte de su ADN innovador.

1.6 Estándares

Con el objetivo de garantizar la calidad total del software desarrollado y promover una cultura de mejora continua, ShopMind adopta un conjunto de estándares internacionales reconocidos, principalmente emitidos por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). Estos estándares sirven como base para definir procesos estructurados, asegurar la trazabilidad, y establecer criterios de validación y documentación rigurosos.

A continuación, se detallan los estándares aplicados en el presente plan:

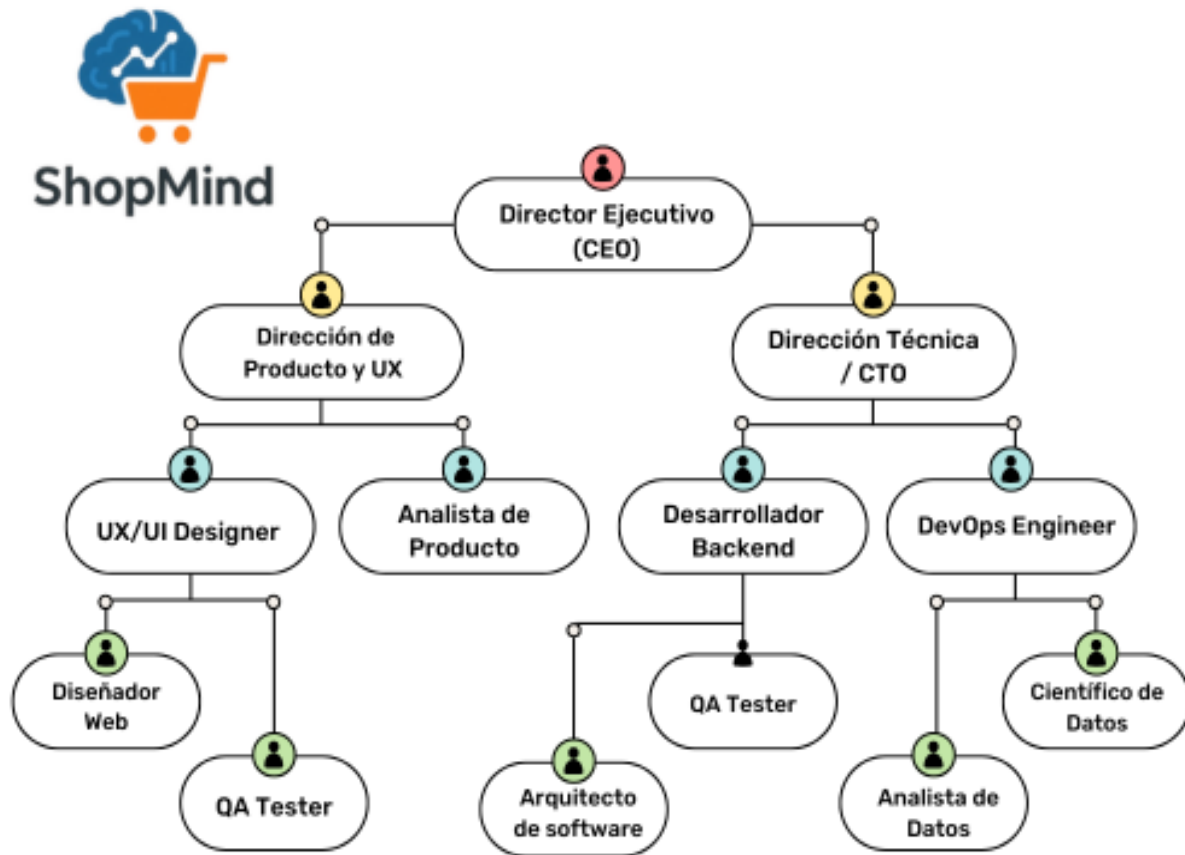
- **IEEE STD 730 – Software Quality Assurance Plans:** Define un marco sistemático y planificado de actividades destinadas a asegurar que el software cumpla con los requisitos técnicos, funcionales y de calidad establecidos.

- IEEE STD 729 – Standard Glossary of Software Engineering Terminology:

Proporciona un lenguaje común para la interpretación de términos clave relacionados con el aseguramiento de calidad y la ingeniería de software.

- IEEE STD 828 – Software Configuration Management Plans: Establece los lineamientos para la gestión de la configuración del software, asegurando el control de versiones, integridad y trazabilidad de los componentes desarrollados.
- IEEE STD 829 – Software Test Documentation: Especifica el formato y contenido requerido para la documentación de pruebas, incluyendo planes, casos, procedimientos y reportes de resultados.
- IEEE STD 830 – Software Requirements Specifications: Define cómo deben estructurarse y documentarse los requerimientos funcionales y no funcionales del software.
- IEEE STD 1012 – Software Verification and Validation Plans: Establece las prácticas necesarias para planificar y ejecutar las actividades de verificación y validación a lo largo del ciclo de vida del software.

1.7 Organigrama



1.8 Políticas de Calidad

En esta Startup especializado en el desarrollo de soluciones SaaS para comercio electrónico, la calidad no es solo un objetivo técnico, sino un principio estratégico que guía todas las decisiones de negocio. La empresa adopta un enfoque integral de calidad basado en estándares internacionales, buenas prácticas de ingeniería y un fuerte compromiso con la satisfacción del cliente. La Dirección General establece esta política como un compromiso institucional y se asegura de que sea conocida, comprendida y aplicada por todo el equipo. Cada

área y cada colaborador contribuye a sostener un ecosistema de mejora continua y de excelencia operativa.

Con este propósito, se promueven los siguientes lineamientos fundamentales:



POLÍTICAS DE CALIDAD



- Esta política será Organización de Grupo de Desarrollo.

- Organización del Consultor o Especialistas en SQA.
- Organización del Cliente.
- Organización del grupo SQA.
- Entidad fiscalizadora (opcional, según contrato o normativa aplicable)

Evaluada regularmente para asegurar su relevancia frente a los cambios del mercado, las exigencias de los usuarios y la evolución tecnológica del entorno digital.

1.9 Gestión

1.9.1 Organización

En el contexto del desarrollo se identifican varias organizaciones y actores clave que intervienen en la ejecución, supervisión y validación del proyecto. Cada una de estas partes cumple un rol fundamental para garantizar que el proceso se lleve a cabo de forma transparente, eficiente y con altos estándares de calidad.

1.9.1.1 Organización en el grupo de Desarrollo.

Esta unidad representa el núcleo operativo responsable de diseñar, construir y mantener el software desarrollado. Opera bajo los lineamientos definidos en el contrato del proyecto y se enfoca en cumplir los requerimientos técnicos y funcionales acordados. Mantiene una comunicación constante con el equipo de calidad y con los consultores, reportando avances, riesgos y decisiones técnicas clave para garantizar la trazabilidad del proceso.

1.9.1.2 Organización del consultorio o especialistas en SQA.

Cuando el proyecto lo requiere, ShopMind puede contar con el apoyo de profesionales externos en calidad de software que actúan como auditores, evaluadores o asesores metodológicos. Estos especialistas ayudan a implementar prácticas alineadas con normas internacionales, revisar la conformidad del producto y facilitar la interacción entre los intereses del cliente y el equipo de desarrollo. Su participación es especialmente valiosa en proyectos estratégicos o con requisitos de alta sensibilidad.

1.9.1.3 Organización del cliente.

La participación del cliente puede variar dependiendo del tamaño del proyecto, su nivel de madurez digital y el alcance funcional esperado. En todos los casos, el cliente proporciona los requerimientos, valida entregables y participa en decisiones clave. Su rol también puede incluir la designación de representantes técnicos o funcionales para asegurar la coherencia entre las expectativas del negocio y la solución entregada.


1.9.1.4 Organización del grupo SQA

El equipo de SQA tiene la responsabilidad de establecer, aplicar y mejorar los procesos de control de calidad del software. Esta unidad actúa como un punto de convergencia entre desarrollo, cliente y consultores, garantizando que se cumplan los estándares definidos y que se documenten adecuadamente las decisiones y resultados. A partir de las recomendaciones recibidas, el equipo SQA elabora, actualiza y valida versiones sucesivas del Plan de Aseguramiento de Calidad del Software (SQAP), velando por su correcta implementación durante todo el ciclo de vida del proyecto.

1.9.2 Tareas y Actividades


El equipo de Aseguramiento de la Calidad (SQA) se encarga de planificar, monitorear y validar la calidad de los productos desarrollados.

Tareas del grupo SQA:

	
Ciclo de vida del software	Tareas y Actividades Asociadas al Ciclo de Vida del Software
Requerimientos	<ul style="list-style-type: none"> ● Captura de requisitos del software ● Elaboración de la especificación de requerimientos de software (SRS) ● Validación y revisión formal de los requisitos junto al cliente ● Revisión de las especificaciones del software
Análisis	<ul style="list-style-type: none"> ● Análisis técnico y de viabilidad de los requisitos documentados ● Evaluación del impacto de requisitos sobre la arquitectura ● Revisión cruzada entre analistas, QA y cliente ● Aprobación del análisis para avanzar a la etapa de diseño
Diseño	<ul style="list-style-type: none"> ● Diseño preliminar de arquitectura de software y componentes ● Documentación de especificaciones de diseño ● Revisión estructurada del diseño preliminar ● Diseño detallado de clases, interfaces y flujos de datos ● Revisión de consistencia entre diseño y requisitos aprobados
Implementación	<ul style="list-style-type: none"> ● Codificación de módulos conforme a estándares internos ● Revisión de consistencia entre código y documentación técnica ● Uso de herramientas de análisis estático y control de calidad de código fuente ● Revisión de código versus la documentación generada

Pruebas	<ul style="list-style-type: none"> ● Diseño de pruebas unitarias, de integración y funcionales ● Ejecución controlada de pruebas automatizadas y manuales ● Evaluación de resultados de prueba frente a criterios de aceptación ● Registro de incidencias y gestión del ciclo de defectos ● Validación de cumplimiento de requisitos a través de pruebas
Instalación y mantenimientos	<ul style="list-style-type: none"> ● Preparación del entorno de producción ● Ejecución de pruebas finales en condiciones reales de operación ● Evaluación de desempeño y validación de instalación ● Documentación de resultados finales y plan de mantenimiento correctivo ● Retroalimentación del cliente post-entrega y monitoreo de incidencias

Actividad del grupo SQA:

	
Actividad	Entregable Asociado.
Redacción y aprobación del Plan de Aseguramiento de Calidad	Documento oficial del Plan de SQA
Identificación de métricas y atributos clave de calidad	Sección de atributos de calidad dentro del Plan de SQA
Evaluación periódica de la calidad de los artefactos desarrollados	Informes de revisión y evaluación del SQA
Verificación del cumplimiento del proceso definido	Reporte de conformidad con procesos establecidos

Ejecución de revisiones técnicas formales en hitos del proyecto	Acta de Revisión Técnica Formal
Análisis y mejora del Plan de SQA en función de retroalimentación	Documento de ajustes y versión actualizada del Plan de SQA
Evaluación global al cierre del proyecto	Informe Final de Evaluación de Calidad SQA
Seguimiento continuo a través de entregas parciales o semanales	Reportes semanales de control de calidad SQA

1.9.2.1 Actividades de calidad a realizar

Para garantizar la calidad integral del software desarrollado, se ejecutan actividades sistemáticas de control y evaluación en diferentes etapas del ciclo de vida del proyecto. Estas acciones permiten verificar el cumplimiento de los estándares técnicos y asegurar que cualquier desviación sea identificada, corregida y documentada apropiadamente.

Las actividades clave de aseguramiento de la calidad planificadas son las siguientes:

1. Supervisión de Productos Críticos

- Se revisan todos los entregables definidos como esenciales en el Plan de Calidad del Proyecto, conforme a los criterios establecidos previamente.

- Los productos son evaluados mediante listas de verificación específicas, que ayudan a asegurar que cumplan con los estándares funcionales, técnicos y documentales.
- Se revisan versiones anteriores para verificar que las observaciones y correcciones pendientes hayan sido resueltas.
- En caso de hallarse desviaciones o inconformidades, se documentan detalladamente y se asignan responsables para su corrección.
- Se genera un Informe de Revisión de SQA, el cual se distribuye al equipo responsable del producto para su seguimiento.

2. Evaluación de Conformidad con el Proceso


- Esta actividad verifica que el desarrollo de cada producto siga las etapas, flujos y prácticas definidas en el proceso de ingeniería de software establecido por ShopMind.
- Se recopila la documentación de cada entregable y se revisan sus versiones previas, contrastándolas con las actividades definidas en el flujo del proyecto.
- Se valida que las desviaciones detectadas en evaluaciones anteriores hayan sido tratadas y resueltas correctamente.
- Se elabora un informe específico sobre el cumplimiento del proceso, el cual es compartido con los responsables de cada componente o módulo del sistema.

3. Ejecución de Revisión Técnica Formal (RTF)

- La Revisión Técnica Formal (RTF) es una sesión estructurada cuyo propósito es identificar errores funcionales, lógicos o de implementación en cualquier producto del software (código, diseño, documentación, etc.).
- Participan en la sesión el responsable de calidad (SQA) y miembros clave del equipo de desarrollo involucrados en el producto revisado.
- Los participantes son convocados previamente y se les entrega el material de revisión con una lista de observaciones, criterios de evaluación y puntos críticos a validar.
- La reunión tiene una duración máxima de dos horas y sigue un formato previamente definido.
- Como resultado, se genera un Informe de RTF con hallazgos, observaciones y acciones correctivas sugeridas.

Para asegurar un control de calidad efectivo durante el desarrollo del software, las actividades del equipo de Aseguramiento de la Calidad (SQA) se distribuyen estratégicamente a lo largo del cronograma del proyecto. Esta planificación permite realizar validaciones progresivas, aplicar correcciones oportunas y mantener un control continuo sobre los entregables.

A continuación, se presenta la programación de las principales actividades del SQA:

	
Actividad del SQA	Semana de Ejecución
Planificación inicial del sistema de calidad	Semana 2
Evaluación y ajuste del Plan de SQA	Semanas 3 y 4
Revisión Técnica Formal (RTF)	Semanas 5, 6, 7, 8, 10, 11 y 12
Supervisión de entregas semanales	Todas las semanas del proyecto
Verificación del cumplimiento del proceso	Semanas 3 a 12 (de forma continua)
Evaluación de calidad de los productos clave	Semanas 3 a 12 (en paralelo con desarrollo)
Redacción del Informe Final de Calidad	Semana 14
Descripción de versiones parciales del producto	Semanas 5, 7, 9, 11 y 12
Redacción de notas de versión	Semanas 6, 8, 10 y 12

1.10 Roles y Responsabilidades

En el marco del desarrollo de soluciones tecnológicas, se establecen responsabilidades claras para cada uno de los actores clave involucrados en el proyecto. Estas funciones están alineadas con los principios de calidad definidos en el Plan de Aseguramiento de la Calidad del Software (SQAP) y permiten mantener una gestión ordenada, transparente y orientada a resultados.

1.10.1 Responsabilidades del grupo de desarrollo

El equipo de desarrollo es el encargado de materializar la solución tecnológica, asegurando su funcionamiento, rendimiento y calidad. Sus responsabilidades incluyen:



1.10.2 Responsabilidades del Cliente

El cliente es un actor activo en el desarrollo del software, ya que su participación garantiza que la solución entregada responda a sus necesidades reales. Entre sus responsabilidades se encuentran:



1.10.3 Responsabilidades de la SQA

En ShopMind, el equipo de Aseguramiento de la Calidad del Software (SQA) juega un papel clave en el control, evaluación y mejora continua de la calidad del producto y del proceso de desarrollo. Sus funciones están orientadas a garantizar que cada etapa del ciclo de vida cumpla con los estándares definidos, minimizando riesgos y maximizando la fiabilidad del sistema final. Diseño y formalización del Plan SQA: Desarrollar y mantener actualizado el Plan de Aseguramiento de Calidad, que actúa como guía estructurada para todas las actividades de control de calidad del proyecto.

Actividades generales del equipo SQA:

- Participación en la definición de procesos: Colaborar activamente en la construcción y documentación del proceso de desarrollo, asegurando que sea claro, medible y alineado con estándares reconocidos.
- Supervisión de actividades de ingeniería: Verificar que cada fase del ciclo de vida del software (análisis, diseño, implementación, pruebas, entrega) se lleve a cabo de acuerdo con los procesos establecidos.
- Auditoría de productos y entregables: Evaluar productos intermedios y finales para garantizar su alineación con requisitos, especificaciones y buenas prácticas.
- Identificación y documentación de desviaciones: Detectar cualquier no conformidad respecto al proceso, documentarla y comunicarla oportunamente a los responsables para su tratamiento.
- Gestión del control de cambios: Supervisar y coordinar el registro, revisión y aprobación de modificaciones tanto en el software como en la documentación técnica.
- Evaluación de métricas de calidad: Analizar indicadores clave (defectos, cobertura de pruebas, cumplimiento de cronogramas, entre otros) para identificar oportunidades de mejora.

Responsabilidades del Consultor o Responsable de Calidad:


- Verificar la calidad final del producto software: Validar que el sistema desarrollado cumpla con las expectativas del cliente y con los estándares definidos.

- Establecer normas y políticas de calidad: Definir prácticas, metodologías y lineamientos que rijan el desarrollo con un enfoque orientado a la calidad, eficiencia y trazabilidad.
- Facilitar reuniones de resolución de conflictos: Coordinar sesiones estructuradas para alinear criterios, resolver diferencias técnicas y garantizar fluidez en la colaboración entre equipos.
- Aprobar y difundir el SQAP: Autorizar la versión definitiva del plan de calidad y asegurar que sea accesible y comprendido por todos los actores del proyecto.
- Monitorear y mejorar el plan SQA: Identificar debilidades del plan vigente, proponer mejoras y validar su eficacia tras cada ciclo del proyecto.
- Autorizar la puesta en producción del software: Emitir la aprobación final para la implantación del sistema una vez validados todos los criterios de calidad.

Elementos clave bajo supervisión del SQA

- Gestión de calidad aplicada al software: Incluye la definición de políticas, estándares internos, procedimientos de control y evaluación durante todo el ciclo de vida.
- Tecnologías y herramientas utilizadas: Supervisión de frameworks, plataformas, metodologías ágiles, sistemas de versionado, y herramientas de seguimiento utilizadas en el proyecto.
- Revisiones técnicas formales (RTF): Planificación y ejecución de sesiones de revisión multidisciplinaria para validar entregables clave como el diseño, código, pruebas o manuales.

- Estrategia de pruebas: Definición de los tipos de pruebas a ejecutar, criterios de aceptación, ambientes de pruebas y herramientas de gestión de testing.
- Control documental y gestión de versiones: Establecer mecanismos para mantener actualizada toda la documentación técnica y controlar adecuadamente los cambios aplicados al sistema.
- Conformidad con estándares: Asegurar que todo el proceso de desarrollo cumpla con estándares reconocidos (por ejemplo, IEEE 730)
- Medición de calidad y generación de reportes: Implementar métricas objetivas y emitir informes periódicos sobre el avance, calidad y riesgos del proyecto, facilitando la toma de decisiones basada en datos.

	
Producto de Trabajo	Responsable
Documento de Requisitos Funcionales	Analista de Producto
Modelo de Casos de Uso	Arquitecto de Software
Alcance Funcional del Sistema	Analista de Producto
Descripción de la Arquitectura del Sistema	Arquitecto de Software
Modelo de Diseño del Sistema	Arquitecto de Software
Modelo de Datos / Esquema de Base de Datos	Científico de Datos
Estándares de Implementación (código y estructuras)	DevOps Engineer
Guía de Documentación Técnica	QA Tester


Documento de Estimaciones de Tiempo y Recursos	Analista de Producto
Análisis y Gestión de Riesgos	Dirección Técnica / CTO
Planificación General del Proyecto	Dirección Técnica / CTO
Plan de Verificación y Validación	QA Tester
Reportes de Pruebas Unitarias, Integración y del Sistema	QA Tester
Plan de Despliegue / Implantación	DevOps Engineer
Manual de Usuario / Documentación para Clientes	UX/UI Designer + QA Tester
Plan de Gestión de la Configuración del Software	DevOps Engineer

1.11 Mediciones


Las mediciones dentro del plan de aseguramiento de la calidad del software de ShopMind constituyen un mecanismo fundamental para evaluar la conformidad, rendimiento y evolución del producto y del proceso. Estas métricas no solo permiten detectar desviaciones respecto a los objetivos de calidad definidos, sino que además son insumos clave para la mejora continua, la toma de decisiones estratégicas y la sustentabilidad del negocio.

1.11.1 Tipos de métricas empleadas

Métricas de producto: Estas permiten evaluar la calidad interna y externa del software entregado

		
Métrica	Descripción	Unidad / Método
Densidad de defectos	Número de errores por cada 1.000 líneas de código (KLOC)	# errores / KLOC
Cobertura de pruebas	Porcentaje del código cubierto por pruebas automatizadas	% líneas cubiertas
Complejidad ciclomática	Medida de la lógica condicional del código fuente	Valor numérico por módulo
Tasa de éxito en pruebas	Porcentaje de pruebas que pasan en cada ejecución del pipeline	% casos exitosos
Tiempo medio entre fallos (MTBF)	Intervalo promedio entre errores funcionales reportados	Horas / días

Métricas de proceso: Evalúan la eficiencia del desarrollo, testing y entrega continua

		
Métrica	Descripción	Unidad / Método
Velocidad del equipo	Puntos de historia completados por sprint	Puntos / sprint
Lead time	Tiempo promedio entre la creación de una tarea y su despliegue	Días / tarea
Frecuencia de despliegues	Número de despliegues a producción por semana	Despliegues / semana
Tasa de retrabajo	Porcentaje de código que fue reescrito debido a errores	% líneas reescritas
Tasa de cumplimiento de entregas	Entregables completados a tiempo respecto al cronograma	% entregas a tiempo

1.11.2 Recolección y análisis de métricas

- Las métricas de código y pruebas se recolectan automáticamente mediante herramientas como GitHub Actions, SonarQube, Jest, Postman Tests, entre otras.
- Las métricas de proceso son recopiladas desde Trello, Notion, y reportes internos de sprint.
- Los datos se consolidan y analizan por el área de SQA cada dos sprints, y se presentan en un informe de calidad mensual.
- Estas métricas se utilizan como base para los informes de revisión técnica y de mejora de procesos.

1.11.3 Trazabilidad y acción correctiva

Toda métrica registrada está asociada a un módulo, responsable técnico y sprint correspondiente, asegurando trazabilidad. En caso de valores críticos o tendencias negativas:

- Se activa una revisión técnica formal.
- Se implementan acciones correctivas, incluyendo refactorización, aumento de pruebas o capacitación.
- El seguimiento se documenta en los informes de SQA y el backlog de calidad.

1.12 Evolución del SQAP

Con el objetivo de mantener la coherencia, trazabilidad y relevancia del documento a lo largo del proyecto, ShopMind adopta un enfoque controlado y estructurado para la gestión de versiones del Plan de Aseguramiento de la Calidad del Software (SQAP). Este enfoque permite

reflejar adecuadamente los cambios técnicos, organizativos o metodológicos que se presenten durante el desarrollo del software.

1.12.1 Control de cambio

El control de versiones del SQAP se basa en una política de numeración incremental que facilita la identificación del alcance de cada modificación. Las reglas para asignar nuevas versiones son las siguientes:

- Cambios menores (parciales): cuando las modificaciones afectan secciones específicas del SQAP sin alterar la línea base general, la versión se incrementa en 0.1. Estas actualizaciones se integran directamente en la versión activa del documento, conservando su estructura principal.
- Cambios mayores (revisión completa): Si se introducen cambios significativos que alteran procesos clave, estructuras organizacionales o criterios técnicos, la versión se actualiza al siguiente número entero (por ejemplo: de 1.0 a 2.0) y se publica una nueva edición completa del documento.
- Revisión acumulada no publicada: Si se realizan múltiples modificaciones menores sin haber generado una nueva edición completa, se consolidan en una versión formal completa y se considera como un salto de versión principal. A partir de ahí, se reinicia la numeración incremental.

1.12.2 Cuando hacer modificaciones

Las actualizaciones al SQAP se aplicarán en los siguientes casos:

- Cambios en el plan del proyecto: Si se modifican los objetivos, plazos, entregables o alcance general del desarrollo, el SQAP debe reflejar estas variaciones para conservar la coherencia metodológica.
- Ajustes al modelo de desarrollo: Cualquier transformación en el modelo de ingeniería del software (requisitos, arquitectura, flujos, herramientas, metodología) obliga a revisar el SQAP para garantizar su adecuación.
- Incumplimiento de lineamientos del SQAP: Cuando se detectan desviaciones significativas respecto a lo establecido en el plan (por ejemplo, procesos no ejecutados, métricas omitidas o roles mal asignados), se requiere la modificación del documento para corregirlo y restablecer el alineamiento con los estándares de calidad.

1.12.3 Implementación de modificaciones

El procedimiento para actualizar formalmente el SQAP ante cambios o problemas es el siguiente:

- Diagnóstico del problema: El equipo SQA realiza un análisis detallado del incidente, inconsistencia o necesidad de mejora.
- Revisión colaborativa: Se convocan reuniones técnicas con los responsables involucrados (desarrollo, producto, QA, CTO) para discutir el cambio requerido y definir acciones.
- Redacción del ajuste: Se edita la sección correspondiente del SQAP según los acuerdos y recomendaciones generadas.

- Aplicación formal del cambio: La modificación se implementa en la versión vigente o, si aplica, se genera una nueva versión del documento.
- Evaluación posterior: Se verifica si el ajuste solucionó el problema original y mejoró la calidad del proceso o producto.
- Divulgación de la nueva versión: Una vez validada la implementación, se emite y distribuye la nueva versión del SQAP para su uso oficial.

1.12.4 Publicación del SQAP

Cada vez que se apruebe una nueva versión del SQAP, esta será:

- Firmada digital o físicamente por el responsable SQA.
- Registrada en el historial de versiones del proyecto.
- Distribuida a todos los equipos involucrados (producto, desarrollo, QA, cliente).
- Incorporada en el repositorio oficial de documentación del proyecto.

La publicación garantiza que todos los actores del proyecto trabajen bajo una misma directriz actualizada, mejorando la coordinación, la trazabilidad y el control de calidad.

1.13 Documentación

La documentación técnica y funcional , constituye un elemento esencial del aseguramiento de la calidad. Su propósito es brindar soporte estructurado al proceso de desarrollo, facilitar la colaboración entre equipos, asegurar la trazabilidad de decisiones y garantizar que el producto final sea comprensible, usable y mantenible tanto para usuarios como para desarrolladores.

1.13.1 Propósito

El propósito de la documentación en ShopMind es asegurar que todas las fases del ciclo de vida del software desde la definición de requisitos hasta la entrega al cliente estén claramente descritas, alineadas con los estándares de calidad, y disponibles para su revisión y validación en cualquier momento del proyecto. Entre sus objetivos específicos destacan:



1.13.2 Documentación mínima requerida

1.13.2.1 Especificación de requisitos de software

Documento que describe detalladamente las funcionalidades esperadas del sistema, incluyendo requisitos funcionales, no funcionales y restricciones operativas. Sirve como base para el diseño y la validación.

Los atributos de calidad considerados incluyen:

- Funcionalidad: Exactitud, integridad, seguridad y adecuación al propósito.
- Confiabilidad: Capacidad de recuperación, estabilidad y tolerancia a errores.
- Usabilidad: Claridad, facilidad de aprendizaje, estética y operabilidad.
- Eficiencia: Rendimiento y uso óptimo de recursos.
- Mantenibilidad: Capacidad de adaptación, actualización y verificación.
- Portabilidad: Facilidad de instalación, adaptación a diferentes entornos y reemplazo

1.13.2.2 Descripción del diseño del software (SDD)

La Descripción del Diseño del Software (SDD) es un documento que detalla la arquitectura y estructura interna del software, así como los componentes y módulos que lo componen. Esta documentación proporciona una visión general del diseño del sistema, incluyendo diagramas, diagramas de flujo, modelos de datos y cualquier otra información relevante para comprender cómo está organizado y cómo funcionan los diferentes elementos del software.

El diseño del software debe cumplir con los siguientes criterios:

- Correspondencia con los requerimientos: Cada elemento del diseño debe contribuir a satisfacer algún requerimiento del software.
- Consistencia con la calidad del producto: El diseño del software debe estar alineado con los requisitos de calidad establecidos para el producto.

1.13.3 Plan de Verificación y Validación


El Plan de Verificación y Validación (V&V) es un componente clave del aseguramiento de la calidad del software. Su finalidad es definir las actividades necesarias para garantizar que el producto entregado cumple con los requisitos acordados con el cliente, funciona conforme a lo esperado y se ajusta a los estándares de calidad definidos por la organización.

El plan establece los procedimientos, herramientas, responsables y entregables necesarios para comprobar que:

- El producto ha sido correctamente construido.
- El producto responde fielmente a lo solicitado por el cliente.

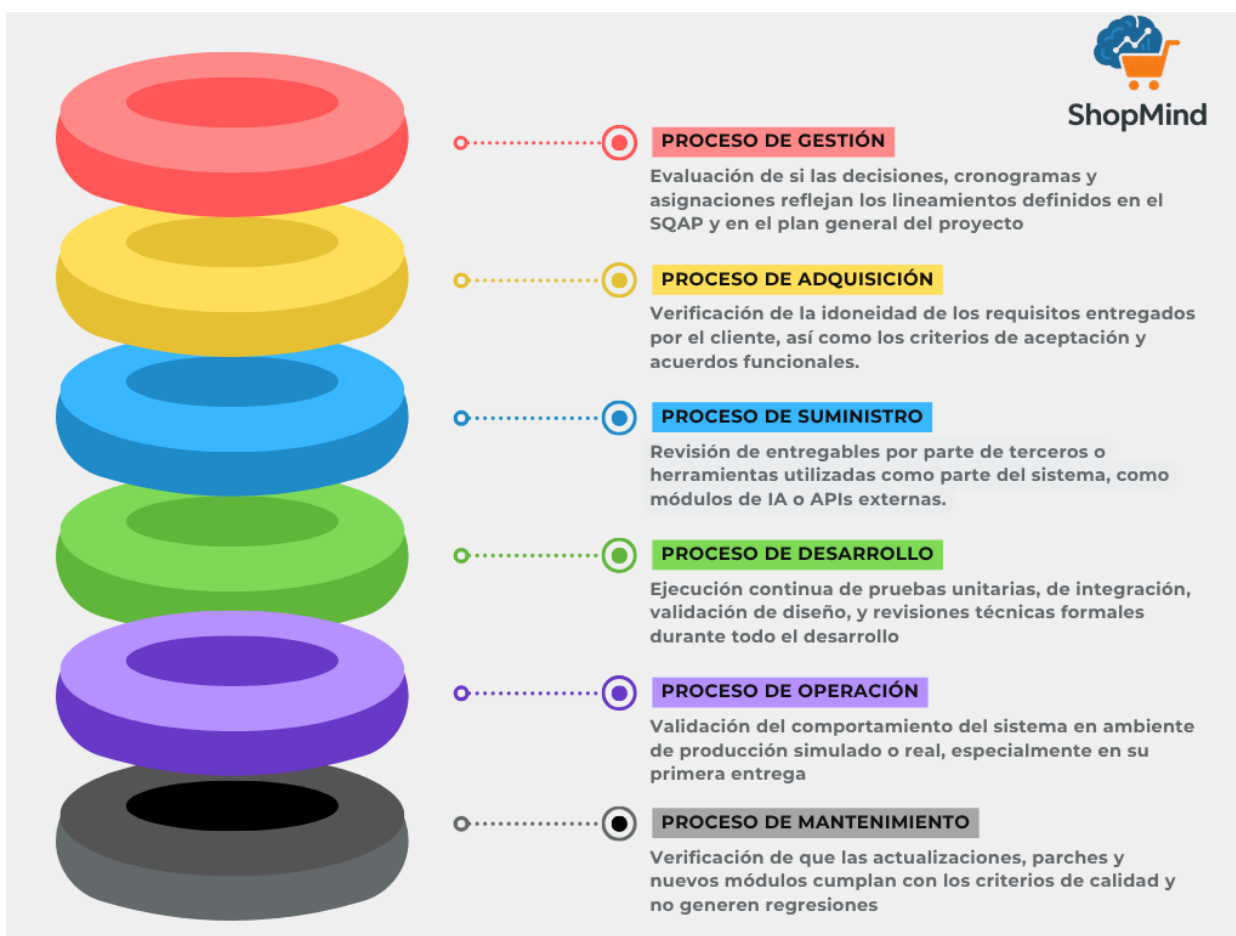
1.13.4 Áreas de aplicación del Plan de Verificación y Validación

Las actividades de V&V se dividen en dos grandes áreas:

	
Verification	Validación
Verificación de requisitos: Confirmar que los requisitos documentados fueron aprobados por las partes pertinentes y cumplen con el contrato funcional y técnico del proyecto.	Ejecutar el producto bajo condiciones reales o simuladas para comprobar que se comporta como se definió en el documento de requisitos
Verificación del diseño: Verificar que los requisitos definidos se encuentran reflejados correctamente en el documento de diseño del sistema.	Evaluar la experiencia del usuario final, integrando criterios de rendimiento, seguridad, usabilidad y funcionalidad clave
Verificación del código: Validar que el diseño aprobado se implementó adecuadamente en el código fuente mediante revisión estructurada, herramientas de análisis estático y pruebas técnicas	Confirmar que cada fase del desarrollo se ejecutó correctamente en relación a sus entradas y salidas

1.13.4.1 Plan para el ciclo de vida de verificación y validación

El enfoque de V&V se aplica de forma transversal a los siguientes procesos:



Todas estas actividades se planifican y ejecutan como parte del cronograma de calidad del proyecto, y su trazabilidad es documentada por el equipo de SQA en cada sprint o entrega.

1.13.5 Documentación de Usuario

La documentación de usuario elaborada en ShopMind está orientada a garantizar que los usuarios finales puedan utilizar el sistema de forma eficiente, segura y autónoma, sin requerir asistencia técnica constante.

Esta documentación es elaborada por el equipo de QA en colaboración con UX/UI, y validada a través de pruebas de usuario o sesiones de onboarding.

Los elementos mínimos que debe contener son:

- **Datos y Entradas de Control:** Explicación clara de los tipos de datos requeridos por el sistema (texto, fechas, números, archivos) y cómo deben ingresarse correctamente.
- **Secuencia de Operaciones:** Guía paso a paso para realizar tareas comunes dentro del sistema (por ejemplo, registrar productos, generar reportes, editar perfiles, etc.).
- **Opciones y Restricciones del Sistema:** Detalle sobre qué funciones están disponibles para el usuario y cuáles no. También se aclaran las configuraciones que pueden personalizarse y los límites del sistema según su rol.
- **Manejo de Errores Comunes:** Identificación de errores frecuentes (como credenciales incorrectas, formatos inválidos, etc.) y cómo solucionarlos. Esto reduce la necesidad de soporte técnico.

- Ajustes según requerimientos del cliente: La documentación se personaliza según el proyecto, incluyendo módulos específicos, reglas de negocio particulares o flujos adaptados a la operación del cliente

1.14 Estándares, Prácticas y Convenciones

El cumplimiento de estándares técnicos y de calidad es esencial para garantizar que el software desarrollado en ShopMind sea confiable, mantenible, escalable y fácil de usar. Esta sección describe los estándares, prácticas y convenciones adoptadas por el equipo para asegurar consistencia en la documentación, el desarrollo, la verificación y el rendimiento del sistema, alineándose con normas internacionales y buenas prácticas de la industria del software.

1.14.1 Estándar de documentación

La documentación cumple un rol fundamental en la trazabilidad del proyecto y la transferencia de conocimiento. Por ello, se definen dos estándares específicos:

Estándar de Documentación Técnica: Este estándar asegura que toda la información necesaria para comprender, mantener y escalar el sistema esté claramente registrada. La documentación técnica debe permitir que cualquier equipo ajeno al de desarrollo pueda continuar con el mantenimiento del software de forma autónoma.

Incluye, como mínimo:

- Modelos de casos de uso
- Diagramas de arquitectura y flujo
- Esquemas de base de datos y relaciones entre componentes

- Código fuente estructurado y comentado
- Información sobre versiones, autores, fecha de emisión y control de cambios

Estándar de Documentación de Usuario: Este estándar garantiza que los usuarios finales comprendan y utilicen el sistema de forma intuitiva. La documentación debe incluir:

- Guías paso a paso
- Capturas de pantalla representativas del flujo del sistema
- Preguntas frecuentes (FAQ)
- Glosario de términos clave
- Canales de soporte y contacto

1.14.2 Estándar de verificación y prácticas

Las actividades de verificación y validación en ShopMind se rigen por las prácticas descritas en el Plan de Verificación y Validación (V&V) y se apoyan en el estándar internacional IEEE Std 1012-2016 (Standard for System and Software Verification and Validation).

Entre las prácticas adoptadas se incluyen:

- Definición de criterios de aceptación por fase
- Revisiones técnicas formales (RTF)
- Verificación de trazabilidad entre requerimientos, diseño, código y pruebas
- Validaciones funcionales y de usabilidad con usuarios reales
- Auditorías internas de calidad por el equipo SQA

El objetivo es garantizar que cada producto intermedio cumpla con su propósito antes de pasar a la siguiente etapa del desarrollo.

1.14.3 Estándar de codificación

Para asegurar legibilidad, mantenibilidad y coherencia en el código fuente, el equipo de desarrollo de ShopMind adopta convenciones comunes y buenas prácticas de programación. Estas reglas están documentadas en la Guía de Estilo de Codificación y se aplican en todos los lenguajes y frameworks utilizados por la organización.

Algunos lineamientos principales incluyen:

Convenciones de nombres

- Clases, interfaces y estructuras: CamelCase (ej. UserProfileController)
- Métodos y funciones: camelCase
- Variables constantes: MAYUS_CON_GUIONES

Estilo del código

- Código indentado con 2 o 4 espacios (según el lenguaje)
- Comentarios descriptivos en inglés técnico
- Prohibición de acrónimos ambiguos o nombres abreviados sin contexto
- Control de versiones en Git con convenciones de commits (feat:, fix:, refactor:)

Herramientas de soporte

- Uso de linters, formateadores automáticos y análisis estático (ex. ESLint, Pylint, SonarQube)

Este estándar es revisado regularmente por el CTO y el equipo de QA para adaptarse a nuevas tecnologías o prácticas emergentes.

1.14.4 Responsable de verificar el rendimiento

La responsabilidad de verificar el rendimiento del sistema recae de forma conjunta en:

- El Desarrollador Backend, como responsable técnico de la lógica de negocio, consultas y procesamiento del sistema.
- El DevOps Engineer, encargado de monitorear el comportamiento del software en los distintos ambientes (staging, producción).
- El equipo SQA, como auditor independiente que valida el cumplimiento de los umbrales de rendimiento definidos.

Las pruebas de rendimiento incluyen:

- Carga y estrés: validación de la respuesta del sistema ante grandes volúmenes de usuarios o peticiones concurrentes.
- Escalabilidad: análisis del comportamiento del sistema ante crecimiento progresivo de datos.
- Tiempo de respuesta: medición del tiempo promedio de procesamiento por transacción crítica.

- Consumo de recursos: monitoreo de uso de CPU, memoria y disco durante la ejecución.

1.15 Revisiones y Auditoría

1.15.1 Objetivo

El propósito de esta sección es establecer las actividades de revisión y auditoría que permitirán evaluar de forma sistemática y objetiva tanto los productos de trabajo como los procesos seguidos en el desarrollo de ShopMind. Se busca asegurar la conformidad con los requisitos establecidos, estándares de calidad definidos y lineamientos del proyecto, así como identificar desviaciones o no conformidades que puedan comprometer el éxito del producto.

1.15.2 Requerimientos mínimos

Durante el ciclo de vida del software se llevarán a cabo las siguientes revisiones formales:

Revisión de Requerimientos: Se realiza para confirmar que los requerimientos del cliente están correctamente documentados, completos, no ambiguos, y son verificables. También se comprueba su trazabilidad con otros documentos de gestión del proyecto. Esta revisión es realizada por el equipo SQA junto con el responsable de requisitos y el arquitecto de software.

Revisión de Diseño Preliminar (PDR): Evalúa la adecuación técnica del diseño general propuesto frente a los requisitos. Se analiza la viabilidad de los componentes, estructuras y estrategias propuestas. Participan el arquitecto de software, desarrolladores principales y representantes de SQA.

Revisión de Diseño Crítico (CDR): Se revisa en detalle la arquitectura y el diseño modular, validando que cada componente esté correctamente definido, integre adecuadamente sus interfaces y dé cumplimiento a los requerimientos funcionales y de calidad.

Revisión del Plan de Verificación y Validación: Se verifica que el plan contenga una descripción completa de las técnicas, herramientas, cronograma y criterios de aceptación para las actividades de verificación y validación. También se evalúa su alineación con los objetivos del proyecto y estándares de prueba.

Revisión del Plan de Gestión de Configuración: Se asegura que los métodos y herramientas especificados sean suficientes para controlar el versionado de artefactos, el seguimiento de cambios y la integridad de la línea base del sistema.

1.15.3 Tipos de Auditoria

Las auditorías permiten identificar si los productos o procesos se desvían de lo establecido en el SQAP, los requerimientos funcionales, o los estándares organizacionales. Se definen las siguientes:

Auditoría Funcional: Validación final de que el software cumple completamente con los requerimientos funcionales aprobados. Se realiza antes de la liberación del producto y es obligatoria para pasar a producción. Requiere la validación cruzada de pruebas funcionales, reportes de V&V, y conformidad del cliente.

Auditoría Física: Verifica que todos los componentes físicos del sistema (ejecutables, configuraciones, documentación técnica y de usuario) estén presentes, versionados correctamente, y coincidan con lo descrito en el diseño aprobado. Es responsabilidad del equipo de QA, con participación de desarrollo y documentación.

Auditorías Internas al Proceso: Son auditorías programadas en las que se inspecciona si las actividades realizadas (codificación, pruebas, integración, documentación) siguen los procesos definidos. Se evalúa la coherencia entre código, diseño y requisitos.

Revisiones de Gestión: Evaluaciones periódicas para comprobar el estado general del proyecto, la ejecución del cronograma, la cobertura de pruebas, la gestión de riesgos y los indicadores de calidad. Se basan en entregables, informes de progreso y KPIs del equipo PM.

1.15.4 Auditoría Administrativa del SQAP

Como parte del ciclo de mejora continua, se realizarán auditorías administrativas del propio SQAP para garantizar su ejecución y validez continua en el proyecto. Estas auditorías serán responsabilidad del equipo SQA e incluirán:

- Revisión del cumplimiento de cada sección del SQAP (requisitos, documentación, roles, métricas, etc.)
- Verificación del cumplimiento de las actividades de aseguramiento de calidad planificadas
- Identificación de desviaciones o secciones obsoletas
- Recomendación de ajustes o mejoras, si corresponde

Estas revisiones se documentarán en un informe de cumplimiento del SQAP, el cual será compartido con los responsables del proyecto y el cliente cuando sea necesario.

1.15.5 Responsabilidades y Seguimiento

- El equipo SQA es responsable de planificar, coordinar y ejecutar las revisiones y auditorías.
- El jefe de desarrollo y los arquitectos técnicos participan como sujetos auditados y colaboradores en revisiones técnicas.
- Los resultados de cada revisión o auditoría se documentarán formalmente y se asignarán acciones correctivas, con responsables y plazos definidos.
- Se empleará un sistema de seguimiento de hallazgos (Issue Tracker o sistema interno) para garantizar que las observaciones se resuelvan antes de liberar el producto.

1.16 Verificación

1.16.1 Propósito

En conformidad con lo establecido en el IEEE 730, esta sección tiene como objetivo ampliar la cobertura de aseguramiento de calidad mediante la identificación y ejecución de actividades de verificación adicionales que no fueron explícitamente incluidas en el Plan de Verificación y Validación (V&V) inicial. La verificación, definida como el proceso que asegura que los productos de trabajo cumplen con sus especificaciones designadas, debe ser exhaustiva, sistemática y documentada, y debe realizarse en cada etapa del ciclo de vida del software.

1.16.2 Alcance de la Verificación Complementaria

Las siguientes áreas de verificación complementaria serán abordadas para reforzar el control de calidad:

Verificación de Plantillas y Estandarización: Se verificará que toda la documentación (técnica y de usuario) respete las plantillas definidas por el SQAP, incluyendo encabezados obligatorios, numeración, versión y metadatos clave. (*Método: Revisión por lista de control de conformidad documental*).


Verificación de la Correcta Aplicación de Prácticas de Ingeniería: Incluye asegurar que las prácticas de diseño, codificación, pruebas y documentación cumplan con los estándares definidos. (*Método: Auditorías internas SQA y revisiones técnicas formales (FTR)*)

Verificación de Productos Intermedios No Cubiertos por el Plan de V&V: Incluye artefactos como prototipos, diagramas de estado, arquitectura preliminar o simulaciones que no estaban listados originalmente. (*Método: Validación técnica con checklist IEEE 730 Apéndice D*)

Verificación de la Aplicación del Estándar de Codificación: Para asegurar uniformidad y legibilidad en el código fuente. (*Método: Inspecciones de código (peer review), análisis estático automatizado.*)

Verificación de la Gestión de Cambios y Trazabilidad: Asegura que los cambios aprobados están adecuadamente documentados, relacionados con su requerimiento y reflejados en el diseño y código. (*Método: Revisión de registros en sistema de control de versiones (Git) y herramientas de trazabilidad (JIRA, Azure Boards, etc.).*)

1.16.3 Métodos de Verificación según IEEE 730

	
Método de Verificación	Descripción según IEEE 730
Revisión técnica formal	Evaluación planificada de productos intermedios y finales.
Análisis de productos	Evaluación estática y dinámica del código/documentos.
Inspecciones	Evaluación manual detallada con checklist.
Auditoría interna	Revisión sistemática de procesos y cumplimiento del SQAP.
Trazabilidad bidireccional	Validación de correspondencia entre requisitos y productos.

1.16.4 Registro y Control de Verificación Adicional

Cada verificación adicional se registrará en el Bitácora de Verificación Complementaria (BVC), la cual incluirá:

- Identificador del producto de trabajo verificado
- Fecha de revisión
- Método aplicado
- Resultado (Conforme / No conforme)
- Acción correctiva recomendada (si aplica)
- Responsable del seguimiento

Este registro será conservado por el líder de SQA y será revisado en cada auditoría interna

1.16.5 . Responsabilidades

	
Actividad	Responsable
Planificación de verificaciones extra	Coordinador SQA
Ejecución de revisiones documentales	Arquitecto de Software / QA
Inspección de código fuente	Desarrollador senior / QA
Validación de trazabilidad	Líder de Requisitos / QA
Seguimiento de acciones correctivas	Jefe Técnico / Líder S QA

1.17 Reporte De Problemas y Acciones Correctivas

Esta sección tiene como propósito establecer los procedimientos y formatos necesarios para reportar, analizar, registrar y resolver cualquier problema identificado durante la ejecución del proyecto, ya sea durante la verificación, validación, auditorías, pruebas o revisiones. Este proceso es clave para mantener la conformidad del producto con los requisitos y estándares definidos, y forma parte del sistema de gestión de calidad definido en este SQAP.

1.17.1 Enfoque según IEEE 730

El enfoque adoptado por ShopMind para la gestión de problemas y acciones correctivas sigue lo establecido en el IEEE Std 730-2014, sección 6.3, el cual requiere que:

- Se definan criterios claros para identificar y documentar no conformidades (anomalías, errores, defectos, omisiones, desviaciones).

- Se establezca un proceso estructurado para analizar la causa raíz de cada problema.
- Se definan acciones correctivas y preventivas específicas.
- Se asegure la asignación de responsables y plazos para la resolución.
- Se mantenga registro trazable y auditable de cada evento, con su estado actualizado.

1.17.2 Tipos de Informes Utilizados

Reporte de Anomalía

Documento estandarizado que registra de manera detallada cualquier desviación observada durante las tareas de verificación y validación.

Contenido mínimo:

- Número de reporte y fecha
- Fase del ciclo de vida en que se detectó
- Flujo de trabajo o módulo afectado
- Descripción del problema y su localización
- Impacto estimado sobre el sistema
- Causa probable (si se conoce)
- Nivel de criticidad (alta, media, baja) según riesgo o impacto
- Resultado de la revisión (aceptado, aceptado con cambios, no aceptado)
- Recomendaciones propuestas

- Firmas de los responsables técnicos y QA

Este reporte puede ser generado tanto por miembros del equipo de QA como por desarrolladores, testers o analistas, y es obligatorio que sea gestionado por el equipo de aseguramiento de calidad.

Reporte Final de Verificación y Validación (V&V)

Este reporte sintetiza todos los resultados de verificación y validación de una fase o entrega del proyecto, y documenta el estado de las anomalías encontradas, las acciones tomadas y la evaluación final del producto.

Contenido mínimo:

- Resumen de tareas V&V ejecutadas
- Resultados cuantitativos (n° de pruebas, % éxito, cobertura)
- Listado de problemas encontrados y su estado
- Evaluación global de calidad del producto
- Recomendaciones para próxima fase o liberación
- Firmas del equipo SQA

Este documento se genera al cierre de cada fase importante del proyecto (fin de diseño, pruebas unitarias, integración, entrega final).

1.17.3 Proceso de Gestión de Problemas

El ciclo de vida de cada problema se gestiona bajo el siguiente flujo:

1. Detección: Cualquier miembro del equipo puede identificar un problema durante el desarrollo, revisión o pruebas.
2. Documentación: Se registra el problema en el formato estandarizado de reporte de anomalías.
3. Evaluación: El equipo SQA determina la criticidad y validez del reporte.
4. Asignación: Se designa un responsable técnico y un plazo para resolver el problema.
5. Acción correctiva: Se implementa la solución, la cual puede incluir reconfiguración, rediseño, recodificación o replanificación.
6. Verificación: QA valida que el problema ha sido resuelto adecuadamente.
7. Cierre: Se actualiza el estado como resuelto/verificado y se archiva.

1.17.4 Registro y Trazabilidad

Todos los reportes de problemas y acciones correctivas deberán:


- Tener un identificador único (ID)
- Estar almacenados en un repositorio digital seguro (por ejemplo: Git, JIRA, Azure DevOps)

- Tener estados de seguimiento: abierto, en proceso, resuelto, verificado, cerrado
- Incluir fecha de detección, fecha de resolución, y responsable

El equipo SQA será el encargado de monitorear y auditar semanalmente el estado de los reportes activos y generar informes de avance en reuniones de gestión.


1.17.5 Indicadores de Control

Para evaluar la eficacia del proceso de gestión de problemas, se utilizarán los siguientes indicadores:

	
Indicador	Fórmula
Tasa de resolución de anomalías	$(\text{N}^\circ \text{ de problemas cerrados} / \text{total detectados}) \times 100\%$
Tiempo medio de resolución	Promedio de días entre detección y cierre
Tasa de reincidencia	$(\text{Problemas reabiertos} / \text{problemas resueltos}) \times 100\%$
Porcentaje de problemas críticos	$\text{Problemas críticos} / \text{total problemas}) \times 100\%$

1.17.6 Informe de verificación y validación

El formato para la documentación de los resultados de la implementación del plan de verificación y validación del software (V & V) es la siguiente:



REPORTE SUMARIO DE FASE V&V		Pág.								
FASE:										
2	# De Reporte: Lugar:	Fecha: ____/____/____ Hora:								
a) Descripción de las tareas de V&V realizadas: b) Sumario de resultados de tareas: c) Sumario de anomalías y resolución: d) Evaluación de calidad del software: e) Recomendaciones:										
<div style="border: 1px solid black; padding: 5px;"> Equipo de Trabajo: <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left; padding: 2px;">Nombre:</th> <th style="width: 50%; text-align: left; padding: 2px;">Firma</th> </tr> </thead> <tbody> <tr> <td style="border-top: 1px solid black; height: 20px;"></td> <td style="border-top: 1px solid black; height: 20px;"></td> </tr> <tr> <td style="border-top: 1px solid black; height: 20px;"></td> <td style="border-top: 1px solid black; height: 20px;"></td> </tr> <tr> <td style="border-top: 1px solid black; height: 20px;"></td> <td style="border-top: 1px solid black; height: 20px;"></td> </tr> </tbody> </table> </div>			Nombre:	Firma						
Nombre:	Firma									

1.17.7 Reporte sumario de fase V&V


Reporte sumario de fase V&V

Reporte de anomalía:



REPORTE DE ANOMALIAS		Pág. ____
Nro. Reporte: ____		
Flujo de trabajo: ____	Fase: ____	
Fecha: ____/____/____	Hora: ____:____	Lugar: ____
a) Descripción y ubicación		
b) Impacto		
c) Causa		
d) <u>Critibilidad</u> Aceptado como esta <input type="checkbox"/> Con modificaciones menores <input type="checkbox"/> No aceptado <input type="checkbox"/> Revisión no terminada (explicar) <input type="checkbox"/>		
e) Recomendaciones		
Equipo de trabajo: ____		
Integrante: ____	Firma: ____	

Reporte final V&V:

 ShopMind	REPORTE FINAL DE V&V		Pág.								
	FASE:										
	# De Reporte:	Fecha:/...../.....									
	Lugar:	Hora:									
a) Resumen de todas las tareas V&V, durante el ciclo de vida del software:											
b) Resumen de resultados de tareas:											
c) Resumen de anomalías, resolución:											
d) Evaluación total de la calidad del software:											
e) Recomendaciones:											
<table border="1"> <tr> <td colspan="2">Equipo de Trabajo:</td> </tr> <tr> <td>Nombre:</td> <td>Firma</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> </table>				Equipo de Trabajo:		Nombre:	Firma
Equipo de Trabajo:											
Nombre:	Firma										
.....										
.....										

1.18 Herramientas, Técnicas y Metodología

1.18.1 Herramientas

Conforme al enfoque de calidad promovido por el estándar IEEE 730, el aseguramiento de calidad requiere la utilización de herramientas que apoyen los procesos definidos y faciliten el cumplimiento de las actividades establecidas en el plan. En nuestro proyecto, hemos seleccionado un conjunto de herramientas orientadas a la trazabilidad, colaboración, control de versiones y automatización de pruebas:

Trello: Herramienta de gestión ágil visual basada en tableros Kanban. Permite una clara asignación de tareas, seguimiento del progreso, vinculación con objetivos de calidad, y soporte a los flujos definidos en el ciclo de vida del software. Su uso se alinea con el control del proceso de desarrollo recomendado por IEEE 730.


GitHub: Plataforma de control de versiones distribuido basada en Git. Facilita la gestión del ciclo de vida del código fuente, incluyendo integraciones CI/CD, revisión de código (peer reviews), trazabilidad de cambios y documentación en línea. Además, permite mantener la integridad del software a través de ramas protegidas y verificación automática, conforme a las prácticas de verificación (IEEE 1012, IEEE 828 y 730).

Jupyter Notebooks y/o Google Colab (cuando aplique IA): Para actividades de validación de modelos y pruebas exploratorias de algoritmos, facilitando la repetibilidad de experimentos y el versionado del código de machine learning (IEEE 829 y 1012).

SonarQube (opcional): Para análisis estático de código y detección de code smells, bugs y métricas de mantenibilidad en el desarrollo, recomendado como práctica de calidad técnica (IEEE 730 sección 7.3)

1.18.2 Técnicas

En concordancia con la guía IEEE 730 y sus referencias complementarias, las siguientes técnicas normalizadas se utilizan para asegurar la calidad del producto de software en cada etapa del ciclo de vida:

		
Norma IEEE	Técnica Aplicada	Propósito
IEEE 830	Software Requirements Specification	Especificación estructurada, verificable y comprensible de los requerimientos funcionales y no funcionales.
IEEE 1016	Software Design Description (SDD)	Documentación formal del diseño del sistema, incluyendo arquitectura, componentes y interfaces.
IEEE 1008	Unit Testing Procedures	Definición de pruebas unitarias, incluyendo entradas, salidas esperadas, y criterios de aceptación.
IEEE 1063	User Documentation Standards	Reglas y estructura para documentación del usuario, centradas en claridad, navegabilidad y mantenimiento.
IEEE 1028	Software Reviews and Audits	Procedimientos para revisiones técnicas formales, inspecciones, walkthroughs y auditorías internas.
IEEE 730	SQA Management Plan	Estructura de planificación del aseguramiento de la calidad

1.18.3 Metodología

Marco de Trabajo Adoptado: Scrum Ágil

El modelo metodológico empleado es Scrum, de tipo incremental e iterativo, altamente alineado con la gestión del cambio promovida por el estándar IEEE 730. Esta metodología permite implementar controles de calidad en tiempo real, fomentando entregas tempranas, validación continua y adaptación dinámica al contexto del cliente.

Componentes de Scrum utilizados:

- **Sprints:** Iteraciones de 4 semanas en las cuales se planifican y entregan incrementos funcionales del sistema.
- **Ceremonias clave:**
 - Sprint Planning: Definición del backlog y objetivos del sprint.
 - Daily Stand-up: Evaluación diaria del avance y obstáculos.
 - Sprint Review: Validación del producto con stakeholders.
 - Retrospective: Revisión de procesos internos y mejora continua.
- **Artefactos:**
 - Product Backlog: Repositorio dinámico de requisitos y mejoras.
 - Sprint Backlog: Selección priorizada de tareas para el sprint actual.
 - Incremento: Versión ejecutable y potencialmente entregable del producto.

Gestión del Cambio y la Calidad dentro de Scrum:

- Priorización continua: Reevaluación semanal del Product Backlog según el feedback del cliente o la aparición de nuevos requerimientos.
- Validación por criterios de aceptación: Toda historia de usuario se valida contra criterios acordados y testeables.
- Auditoría empírica y retrospectivas: La metodología favorece revisiones estructuradas que se alinean con las revisiones técnicas formales descritas en IEEE 1028.

1.19 Control de Código

En ShopMind, el control del código fuente se implementa como un componente central del sistema de aseguramiento de calidad del software (SQA), conforme a las directrices del estándar IEEE 730-2014, así como alineado con prácticas complementarias de los estándares.

El propósito principal de esta sección es asegurar la integridad, trazabilidad, seguridad y control formal sobre todos los artefactos de software desarrollados en el proyecto.

1.19.1 *Software sujeto a control*

De acuerdo con IEEE 730, todos los elementos configurables del software deben estar identificados, versionados y controlados. En ShopMind, se consideran bajo control de código los siguientes componentes:

- Módulos de código fuente de backend y frontend.
- Scripts de infraestructura (Docker, Kubernetes, Terraform).
- Modelos entrenados de machine learning (archivos .pkl, .h5, .tflite).
- Esquemas de base de datos y migraciones.
- APIs y definiciones GraphQL/REST.
- Archivos de configuración (.env, .json, YAML).
- Scripts de pruebas automatizadas.

1.19.2 *Identificación, etiquetado y versionado*

Todo artefacto bajo control se rige por un sistema de versionado semántico (SemVer):

Cada commit está vinculado a una tarea registrada (Trello, GitHub Issues) y firmado digitalmente para mantener integridad.

Además, los módulos están etiquetados por funcionalidad, por ejemplo:

- stomer-v1.0
- ML-Recommender-v2.2
- API-Orders-v1.1

Estas etiquetas permiten una trazabilidad completa desde la versión hasta su componente funcional.

1.19.3 Repositorio y ubicación física

La gestión de versiones y colaboración se realiza mediante GitHub, asegurando control de accesos, visibilidad de cambios y auditoría. Se establecen los siguientes entornos:

- main: versiones estables y liberadas.
- develop: integración continua y QA.
- feature/*: desarrollo de nuevas funciones.
- hotfix/*: correcciones urgentes de errores.

Además, se realiza un backup automatizado diario del repositorio en un bucket de Amazon S3 cifrado, cumpliendo con las buenas prácticas de continuidad de negocio.

1.19.4

Procedimientos de construcción de nuevas versiones

Todo lanzamiento sigue un pipeline CI/CD con pasos claramente definidos:

- Integración: revisión por pares del código mediante pull request.
- Testing automático: ejecución de pruebas unitarias y de integración.
- Build: generación del artefacto deployable (Docker image, APK, ZIP).
- UAT: pruebas manuales en entorno staging.
- Deploy: despliegue automatizado con notificación en canal Slack.

Este proceso cumple con IEEE 730 en cuanto a la verificación previa a liberación y control de calidad del producto final.

1.19.5 Control de documentación relacionada

Por cada versión nueva se debe actualizar y publicar:

- Manual técnico del componente afectado.
- Bitácora de cambios (CHANGELOG.md).
- Notas de versión (RELEASE_NOTES.pdf).

Actualización del repositorio de documentación en Notion o Confluence.

1.19.6 Alineación con IEEE 730

El sistema de control de código de cumple con las siguientes directrices del estándar IEEE 730:

- 1 control del producto de software.
- 2 gestión: planificación del control del software como parte del SQAP.
- 3 métricas de trazabilidad del cambio (por ejemplo, cambios por sprint).
- 4 gestión de proveedores externos en caso de módulos de terceros (API, librerías).
- 5 registros: historial completo de versiones y cambios implementados.

1.20 Control de Medios.

El control de los medios de comunicación es un elemento esencial dentro del plan de aseguramiento de la calidad del software. Su propósito es garantizar que toda la información relevante fluya de manera clara, segura, registrada y rastreable durante todas las fases del ciclo de vida del software. La gestión adecuada de estos medios no solo favorece la colaboración y evita ambigüedades, sino que también asegura la conformidad con los estándares y requisitos técnicos del cliente.

1.20.1 Medios de Comunicación Interna

La comunicación dentro del equipo técnico, de QA y gestión se realiza exclusivamente por canales digitales establecidos y autorizados por la organización. Para la coordinación diaria del trabajo se utiliza Slack, donde se centralizan conversaciones por canal (por ejemplo: #desarrollo, #qa, #bugs, #lanzamientos). Todas las reuniones internas se gestionan mediante Google Calendar y se realizan por Google Meet, dejando registro de grabaciones o minutas en caso de decisiones críticas.

Para la gestión de tareas, se emplea Trello como sistema visual de seguimiento, donde se definen los responsables, prioridades y fechas límite. Los cambios relevantes en diseño,

codificación o pruebas deben referenciarse en los tableros activos con vínculos a GitHub o a la documentación relacionada. Estas prácticas aseguran trazabilidad total desde la toma de decisiones hasta su ejecución, tal como recomienda IEEE 730 en lo relativo al monitoreo de procesos y control documental.

Todas las decisiones que afecten directamente la arquitectura del sistema, las historias de usuario o los objetivos del sprint deben estar respaldadas por documentos de acta (cuando son resultado de reuniones) o por comentarios verificados en herramientas formales. No se considera válida ninguna decisión acordada por medios no oficiales o no registrada en el historial del proyecto

1.20.2 Medios de Comunicación Externa

Las interacciones con actores externos —ya sea clientes, proveedores, mentores o auditores— se canalizan exclusivamente por medios institucionales, principalmente el correo corporativo de ShopMind (ej. info@shopmind.ai, soporte@shopmind.ai). Cada mensaje enviado debe contener una firma digital del remitente, la fecha y un encabezado claro que defina el propósito del mensaje, asegurando la formalidad y claridad de la comunicación.

Las reuniones externas se documentan mediante minutas en formato estándar, que incluyen lista de participantes, puntos tratados, acuerdos alcanzados, decisiones técnicas y tareas asignadas. Estas minutas son archivadas en la unidad de Google Drive oficial, y referenciadas en el sistema Trello o en el módulo de QA cuando sea pertinente.

Los documentos legales, como contratos, acuerdos de confidencialidad o términos técnicos de referencia, se almacenan bajo permisos restringidos, firmados digitalmente, y se

encuentran resguardados en una carpeta cifrada. Estas prácticas aseguran la conformidad con las exigencias de IEEE 730 para mantener registros formales, auditables y protegidos en contextos colaborativos de desarrollo de software.

1.20.3 Control de Información Técnica y Documental

Todos los productos de trabajo generados durante el ciclo de desarrollo, incluyendo código fuente, documentación técnica, reportes de QA, actas, modelos de datos y manuales, son tratados como activos críticos. Por ello, ShopMind establece un procedimiento sistemático para su generación, almacenamiento, control de versiones y acceso.

El código fuente se almacena exclusivamente en repositorios GitHub privados, donde cada commit queda registrado con autor, fecha y mensaje. La documentación asociada se encuentra sincronizada con repositorios de Google Drive estructurados jerárquicamente por proyecto, versión y área. El acceso está limitado por permisos según el rol (QA, desarrollo, gestión).

Cualquier archivo sensible (como tokens, claves API, credenciales o variables de entorno) es gestionado mediante un gestor de secretos cifrado (por ejemplo, 1Password), y bajo supervisión del responsable de seguridad y DevOps. La documentación técnica y de usuario se revisa periódicamente y toda modificación debe ser registrada, de modo que sea posible reconstruir el historial completo de cambios.

IEEE 730 establece que el proceso de documentación debe garantizar consistencia, control de acceso y trazabilidad; ShopMind cumple con estos principios aplicando políticas de control de medios combinadas con buenas prácticas de ciberseguridad.

1.20.4 Seguridad y Trazabilidad en los Medios Utilizados

En lo referente a la seguridad y trazabilidad de la información transmitida y almacenada, ShopMind adopta una política de medios digitales seguros. No se permite el uso de memorias USB, discos portátiles o correos personales para transportar archivos relacionados al desarrollo. Las excepciones deben ser autorizadas por escrito por el CTO, registradas en el log de operaciones, y sometidas a revisión posterior.

Las copias de seguridad de los repositorios de código y documentación se realizan automáticamente cada 24 horas en un bucket privado de Amazon S3 con cifrado en tránsito y en reposo. Se conserva un historial de copias por 90 días, con validación de integridad por hash.

Adicionalmente, toda interacción relevante con el cliente (como entregas, releases o validaciones) se acompaña de un acuse de recibo digital y se conserva una bitácora de eventos críticos que pueden ser requeridos en auditorías internas o externas.

1.21 Control de Proveedores

En el contexto de ShopMind, una startup orientada al desarrollo de soluciones SaaS inteligentes para comercio electrónico, el control de proveedores representa una actividad crítica para asegurar que todos los recursos, servicios, licencias, librerías, APIs y herramientas externas utilizadas en el ciclo de vida del software cumplan con los estándares de calidad, seguridad y confiabilidad establecidos por el proyecto.

Dado que el desarrollo ágil y escalable de nuestras soluciones depende en gran medida de componentes de terceros (tales como frameworks, microservicios, modelos de IA preentrenados, plataformas cloud y servicios de autenticación), es fundamental establecer mecanismos claros

para evaluar, seleccionar, monitorear y documentar la relación con cada proveedor, conforme a las prácticas del estándar IEEE 730.

1.21.1 Tipos de proveedores identificados

ShopMind clasifica a los proveedores involucrados en su ciclo de desarrollo en las siguientes categorías:

- Proveedores de infraestructura: empresas que brindan servicios de nube, hosting y almacenamiento (por ejemplo, Railway, AWS, Vercel).
- Proveedores de herramientas de desarrollo y colaboración: plataformas como GitHub, Trello, Slack, Google Workspace.
- Proveedores de librerías de software y APIs externas: como Stripe para pagos, Firebase Auth para autenticación, OpenAI o HuggingFace para modelos de IA.
- Proveedores de datasets y contenidos: fuentes de datos abiertos o de pago utilizados para entrenar modelos o alimentar dashboards BI.
- Consultores o freelancers técnicos: profesionales externos que colaboran en áreas específicas bajo contrato temporal (por ejemplo, DevOps, diseño UX/UI, expertos en regulación de datos).

1.21.2 Procedimiento para la selección de proveedores

Todo proveedor que influya directa o indirectamente en la calidad del producto final debe ser evaluado antes de ser integrado al flujo de trabajo. Los criterios para su selección incluyen:

- Cumplimiento con estándares de seguridad (por ejemplo, GDPR, ISO/IEC 27001).

- Reputación técnica y nivel de soporte disponible.
- Compatibilidad tecnológica con los sistemas y lenguajes utilizados por ShopMind.
- Nivel de integración con nuestras herramientas internas.
- Costos y escalabilidad.

1.21.3 Control de calidad y seguimiento

Una vez seleccionado un proveedor, se establecen controles regulares para monitorear el cumplimiento de sus compromisos técnicos y contractuales. Las actividades de seguimiento incluyen:

- Auditorías internas de uso y dependencia del proveedor cada 2 sprints.
- Verificación de disponibilidad y SLA (Service Level Agreement) en entornos de staging y producción.
- Monitoreo de rendimiento, seguridad y actualizaciones de las herramientas/librerías integradas.
- Generación de informes de revisión cuando un proveedor genera fallas o inconsistencias.

En caso de que un proveedor crítico incumpla requisitos esenciales, el área de SQA puede emitir una recomendación de reemplazo o mitigación respaldada por métricas y reportes.

1.21.4 Documentación relacionada a proveedores

Todos los proveedores aprobados deben estar documentados en el registro oficial de terceros, el cual incluye:

- Nombre de la empresa/proveedor
- Tipo de servicio que provee
- Fecha de integración al sistema
- Evaluación técnica y funcional inicial
- Enlace a términos y condiciones vigentes
- Responsable técnico interno que gestiona la relación
- Historial de incidencias o auditorías si las hubiera

Este registro debe mantenerse actualizado y a disposición del equipo de calidad, gerencia técnica y auditores externos. Su formato está alineado con las prácticas sugeridas por IEEE 730 para trazabilidad organizacional y cumplimiento del aseguramiento de calidad en entornos colaborativos.

1.22 Gestión de Registros

La gestión de registros en ShopMind tiene como propósito garantizar que toda la información generada y recolectada a lo largo del ciclo de vida del software sea adecuadamente almacenada, protegida, accesible, trazable y verificable. Esta gestión se considera una responsabilidad crítica dentro del aseguramiento de calidad del software, ya que permite respaldar decisiones técnicas, demostrar cumplimiento de procesos, y facilitar auditorías internas y externas.

En concordancia con el estándar IEEE 730, los registros deben ser tratados como evidencia objetiva del cumplimiento con los requisitos definidos en el plan de calidad, y deben mantenerse actualizados, completos y disponibles según su relevancia y confidencialidad.

1.22.1 Tipos de registros gestionados

En ShopMind se generan y gestionan distintos tipos de registros, los cuales se clasifican de la siguiente manera:

- Registros técnicos del desarrollo:
- Historial de versiones del código fuente (commits y pull requests).
- Resultados de pruebas unitarias, de integración y sistema.
- Bitácoras de cambios, logs de ejecución y reportes de errores.
- Registros de calidad y procesos:
- Actas de revisión técnica formal (RTF).
- Informes de cumplimiento de sprint y entregables.
- Reportes de métricas de calidad del software.
- Registros de gestión de configuración:
- Historial de versiones de artefactos documentales.
- Tablas de trazabilidad entre requisitos, diseño, código y pruebas.
- Registros administrativos y contractuales:
- Contratos con proveedores y colaboradores.
- Confirmaciones de entrega al cliente (firmas digitales).
- Fichas de evaluación de riesgos y plan de mitigación.

1.22.2 Requisitos para la administración de registros

Todo registro generado por el equipo de desarrollo, QA, DevOps o gestión debe cumplir con los siguientes requisitos:

- Identificación única: cada registro debe tener un nombre claro, versión, fecha y responsable.
- Accesibilidad controlada: los registros deben estar disponibles únicamente para los roles autorizados.
- Formato digital seguro: el almacenamiento se realiza exclusivamente en sistemas digitales con respaldo automático (ej. Google Drive corporativo, GitHub, Notion, Amazon S3).
- Trazabilidad: todo cambio en un registro debe dejar un historial auditable, especialmente si está relacionado a decisiones técnicas, funcionales o de calidad.
- Permanencia y retención: se define un período mínimo de retención de 12 meses para los registros relacionados a desarrollo, y de 24 meses para aquellos que involucran clientes, entregas o validaciones formales.

1.22.3 Procedimientos asociados a la gestión de registros

- Todos los documentos generados durante un sprint deben ser archivados por el responsable del área correspondiente (desarrollo, QA, gestión).
- Los documentos oficiales deben almacenarse en la carpeta compartida “/SQA/Documentos_Oficiales” en Google Drive, estructurados por año, proyecto y fase.

- El líder de calidad revisará mensualmente el estado de los registros y emitirá un informe de consistencia documental.
- En cada release o hito de entrega, se debe generar un “paquete de entrega” que contenga todos los registros vinculados (requisitos, pruebas, versiones, validaciones, notas técnicas).
- En caso de auditorías o revisiones por parte de terceros (clientes, docentes o auditores), se debe preparar un índice de registros solicitados para consulta inmediata.

1.23Capacitación

La capacitación es un componente esencial en el aseguramiento de la calidad del software en ShopMind. Un equipo técnicamente competente y actualizado es clave para garantizar que cada fase del desarrollo del software se ejecute conforme a los estándares establecidos, incluyendo la correcta implementación de procesos, metodologías, herramientas y buenas prácticas. El presente apartado establece las políticas, modalidades y responsabilidades para asegurar la formación continua del personal involucrado en el desarrollo, prueba, gestión y mantenimiento del producto.

1.23.1 Objetivos de la capacitación

El programa de capacitación de ShopMind tiene como objetivos principales:

- Asegurar el dominio técnico de las herramientas, lenguajes, frameworks y plataformas utilizadas en cada etapa del desarrollo.

- Garantizar la comprensión e implementación adecuada de los procedimientos definidos en el SQAP.
- Fortalecer las habilidades blandas relacionadas a colaboración ágil, liderazgo técnico, gestión de cambios y documentación efectiva.
- Actualizar al equipo en normativas y estándares internacionales, como IEEE 730, ISO/IEC 25010, ISO/IEC 12207, entre otros aplicables al ciclo de vida del software.

1.23.2 Modalidades de capacitación

El programa de formación en ShopMind se implementa a través de diversas modalidades adaptadas al contexto startup y a la disponibilidad operativa del equipo:

- Capacitaciones internas: sesiones lideradas por miembros senior del equipo o el responsable de calidad, donde se comparten buenas prácticas, herramientas nuevas o revisiones de procesos. Estas sesiones se realizan al menos una vez por sprint.
- Cursos online certificados: se promueve la inscripción a cursos en plataformas como Coursera, Udemy, edX o LinkedIn Learning, especialmente en temas como testing automatizado, DevOps, arquitectura en la nube y gestión de calidad.
- Workshops prácticos: talleres intensivos para aplicar conocimientos sobre CI/CD, control de versiones, revisión de código o auditorías internas.
- Simulacros de auditoría o revisión técnica formal (RTF): ejercicios internos para entrenar la preparación documental y la argumentación técnica frente a procesos formales de revisión.

1.23.3 Temáticas prioritarias

Las principales áreas temáticas que el programa de capacitación cubre incluyen:

- Metodologías ágiles (Scrum, Lean, Kanban).
- Control de calidad del software y pruebas (QA, TDD, BDD).
- Seguridad de la información y buenas prácticas de codificación segura.
- Verificación y validación de software.
- Gestión de configuración y control de versiones.
- Redacción técnica y documentación orientada a auditoría.
- Normas IEEE y procesos de aseguramiento de calidad.

1.23.4 Registro y seguimiento de la capacitación

Todos los cursos, talleres o sesiones internas deben ser registrados por el área de calidad, manteniendo un registro digital individual por colaborador que incluya:

- Fecha de la capacitación.
- Tema y modalidad.
- Responsable o proveedor.
- Evaluación o retroalimentación del participante.
- Evidencia de finalización (certificados, capturas, informes).

Este registro forma parte del sistema de gestión de registros documentales descrito en la sección 1.21 y se utilizará como insumo durante auditorías internas o revisiones formales.

1.23.5 Evaluación de efectividad

Al finalizar cada ciclo trimestral, el equipo de calidad evaluará el impacto de las capacitaciones en la práctica diaria mediante:

- Encuestas de autoevaluación del personal.
- Revisión de errores recurrentes o mejoras en métricas de calidad.
- Observaciones en retrospectivas de sprint.
- Resultados en auditorías internas y revisiones técnicas.

En base a estos resultados, se planificará la siguiente ronda de temas prioritarios y se ajustará el plan de capacitación continua.

1.24 Gestión de Riesgos

La gestión de riesgos es una función crítica dentro del aseguramiento de la calidad del software en ShopMind. Su objetivo es identificar, analizar, priorizar, mitigar y monitorear posibles eventos o condiciones que puedan afectar negativamente la calidad del producto, los plazos del proyecto, los costos o la satisfacción del cliente.

En consonancia con el enfoque sistemático recomendado por IEEE 730, ShopMind incorpora actividades de gestión de riesgos desde las primeras etapas del ciclo de vida del software, integrándolas como parte fundamental del proceso de planificación y mejora continua.

1.24.1 Proceso de Gestión de Riesgos

El proceso de gestión de riesgos en ShopMind sigue las siguientes fases:

Identificación de riesgos:

- Se realiza durante la planificación inicial del proyecto y se actualiza de forma continua.
- Involucra a todos los actores: desarrollo, QA, gestión, diseño y DevOps.
- Se utilizan checklists, retrospectivas previas y experiencias similares como fuentes de detección.

Análisis de riesgos:

- Cada riesgo identificado se evalúa en términos de probabilidad de ocurrencia y nivel de impacto (bajo, medio, alto).
- Se clasifica si el riesgo afecta a: calidad del producto, seguridad, cumplimiento de entregas, costos, rendimiento, experiencia de usuario o integridad de datos.

Priorización:

- Los riesgos se ordenan usando una matriz de impacto-probabilidad y se clasifican como críticos, moderados o tolerables.
- Los críticos se abordan inmediatamente; los moderados se planifican; los tolerables se documentan para seguimiento.

Plan de mitigación y contingencia:

- Se define una estrategia para prevenir o reducir el riesgo (mitigación), y una para actuar si ocurre (contingencia).

- Cada plan tiene asignado un responsable, una acción concreta y un umbral de activación.

Monitoreo y revisión continua:

- Se revisan los riesgos en cada reunión de planificación de sprint y retrospectiva.
- Los indicadores de riesgo se incorporan a los tableros de gestión para su visibilidad y seguimiento.

1.24.2 Tipos de riesgos más comunes en ShopMind

En el contexto tecnológico y de negocio de ShopMind, los riesgos más frecuentes son:

- Técnicos: errores en la integración de APIs externas, dependencia de paquetes obsoletos, incompatibilidades entre microservicios.
- Operativos: falta de capacitación en nuevas herramientas, rotación de personal, ausencia de documentación crítica.
- De calidad: cobertura insuficiente de pruebas automatizadas, fallos no detectados en staging, deuda técnica acumulada.
- De gestión: cambios bruscos en los requerimientos, retrasos en entregas clave, sobrecarga de tareas en el equipo.
- De seguridad: brechas en la autenticación de usuarios, exposición accidental de claves API, vulnerabilidades no parcheadas.
- Externos: caída de servicios de terceros, cambios regulatorios, inestabilidad del mercado o interrupción de financiamiento.

1.24.3 Documentación de riesgos

Todos los riesgos identificados, junto con su evaluación, acciones de mitigación y responsables, se registran en un Documento de Gestión de Riesgos del Proyecto, que contiene:

- Código de riesgo (ej. RQ-01, RS-03)
- Descripción detallada
- Fecha de detección
- Clasificación (técnico, operativo, etc.)
- Nivel de impacto y probabilidad
- Plan de mitigación
- Plan de contingencia
- Responsable asignado
- Estado actual (abierto, mitigado, cerrado)

Este documento se actualiza de forma dinámica y se almacena junto al resto de la documentación de calidad del proyecto.

1.25 Bibliografía

Institute of Electrical and Electronics Engineers. (1987). IEEE Standard for Software Unit Testing (IEEE Std 1008-1987). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1988a). IEEE Standard Dictionary of Measures to Produce Reliable Software (IEEE Std 982.1-1988). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1988b). IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software (IEEE Std 982.2-1988). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1994). IEEE Standard for Software Safety Plans (IEEE Std 1228-1994). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1996). Industry Implementation of International Standard ISO/IEC 12207:1995—Software Life-Cycle Processes (IEEE/EIA 12207.0-1996). IEEE Standards Association & Electronic Industries Alliance.

Institute of Electrical and Electronics Engineers. (1997). IEEE Standard for Software Reviews and Audits (IEEE Std 1028-1997). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1998a). IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1998b). IEEE Standard for Software Test Documentation (IEEE Std 829-1998). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1998c). IEEE Standard for Software Verification and Validation Plans (IEEE Std 1012-1998). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1998d). IEEE Standard for Software Project Management Plans (IEEE Std 1058-1998). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1998e). IEEE Standard for Software Configuration Management Plans (IEEE Std 828-1998). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (1998f). IEEE Standard for Software Maintenance (IEEE Std 1219-1998). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (2001). IEEE Standard for Software Life-Cycle Processes—Risk Management (IEEE Std 1540-2001). IEEE Standards Association.

Institute of Electrical and Electronics Engineers. (2002). IEEE Standard for Software Quality Assurance Plans (IEEE Std 730-2002). IEEE Standards Association.

International Organization for Standardization & International Electrotechnical Commission. (1998). Information Technology — System and Software Integrity Levels (ISO/IEC 15026:1998). ISO / IEC.