Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA MENCIÓN EN INGENIERÍA DE SOFTWARE

AVANCE PROYECTO FINAL

Nombre de la asignatura: GESTIÓN DE LA CALIDAD DEL SOFTWARE

Docente responsable: REMBRANDT UBALDE

Integrantes:

- Heber Hualpa Canales.
- Melissa Rodriguez Sandoval.
- Ronald Ticona Humpiri.
- Sihomara Ochoa Cisneros.
- Jhonathan Pauca Joya.

Lima, Octubre de 2025

Contenido

1. Introducción

Este documento describe el plan estratégico y operativo para asegurar la calidad en la implementación del cliente web de Bitwarden, adaptado a las necesidades de Backus. Nuestro objetivo es doble: por un lado, garantizar que la solución cumpla con los requisitos funcionales, de disponibilidad y de negocio; por otro, alcanzar estándares elevados de seguridad, fiabilidad y usabilidad para proteger los activos de información críticos y asegurar la confianza de los usuarios.

La gestión de la calidad se entiende como un proceso integral que cubre especificación de requisitos, verificación, validación y mejora continua. Incluye actividades preventivas (análisis de riesgos, revisión de dependencias), actividades de verificación y validación (pruebas manuales y automatizadas, revisiones de código y análisis estático) y actividades correctivas y de mejora (gestión de defectos, auditorías y métricas).

2. Perfil Empresarial

2.1 Nombre y Razón Social

- Backus S.A. (organización para la cual se diseña este plan).

3. Modelo de Servicio

3.1 Descripción General del Servicio

El servicio comprende la implementación, configuración y gobernanza del cliente web de Bitwarden para uso corporativo. Esto incluye el despliegue del cliente web y sus extensiones, la configuración de políticas de seguridad (por ejemplo, complejidad de contraseñas y obligatoriedad de 2FA), la definición de roles y permisos para bóvedas organizacionales, y los procedimientos de soporte y formación para usuarios finales.

El objetivo del servicio es centralizar y proteger las credenciales y secretos de la organización, facilitando el cumplimiento de políticas de seguridad y reduciendo la exposición a prácticas inseguras (almacenamiento en hojas de cálculo, notas, o reutilización de contraseñas).

3.2 Alcance del Servicio

El alcance inicial contempla:

- Preparación del entorno y despliegue piloto (nube o auto-alojado según decisión de la organización).
- Configuración de políticas de seguridad y parámetros maestros.
- Implementación de la extensión de navegador para autocompletado en navegadores soportados.
- Implantación de la funcionalidad de "Organizaciones" para compartir credenciales por equipos.
- Formación inicial y material de soporte para usuarios.

La fase piloto se ejecutará con un equipo representativo para recoger feedback y ajustar documentación y procesos antes del despliegue general.

3.3 Flujo Operativo

El flujo operativo del servicio está diseñado para ser intuitivo y seguro:

- Acceso Seguro: los usuarios inician sesión con su correo y contraseña maestra. Cuando 2FA está habilitado, se requiere el segundo factor (app de autenticación o llave hardware) para completar el acceso.

- Gestión de Elementos: los usuarios pueden ver, añadir, editar y organizar credenciales, notas y tarjetas. Se contemplan opciones de agrupación (carpetas/etiquetas) y búsqueda avanzada.

- Autocompletado: la extensión del navegador, al detectar una coincidencia con el sitio web visitado, propone el llenado automático de credenciales, minimizando la necesidad de introducir contraseñas manualmente.

Estos flujos serán validados durante el piloto y ajustados con base en la retroalimentación de los usuarios.

3.4 Fases

Fase Pre-Operativa

Actividades incluidas en el avance: despliegue de infraestructura, configuración de políticas maestras, pruebas internas por TI.

Fase Operativa

Actividades incluidas en el avance: piloto controlado con un departamento representativo, recopilación de feedback, ajustes a la documentación, despliegue general y formación, y mantenimiento y mejora continua.

4. Modelo de Pruebas

4.1 Objetivo del modelo

El objetivo del modelo de pruebas es verificar y validar que las funcionalidades críticas del cliente web (autenticación segura, gestión de la bóveda, autocompletado y mecanismos de compartición) cumplen con los requisitos de seguridad, funcionalidad y usabilidad definidos por la organización.

El modelo apunta a reducir riesgos mediante:

- Pruebas automatizadas repetibles para detectar regresiones tempranas.
- Pruebas manuales exploratorias para descubrir problemas de usabilidad o vulnerabilidades no cubiertas por scripts.
- Validaciones de políticas de seguridad (por ejemplo, reglas de complejidad de contraseñas y enforcement de 2FA).

4.2 Tipos de Pruebas Incluidas

- Pruebas funcionales: validación de flujos básicos (login, CRUD de elementos, búsquedas y autocompletado).
- Pruebas de regresión: conjunto reducido de pruebas críticas que se ejecutarán en cada build para asegurar estabilidad.
- Pruebas de integración: verificar la interacción entre la interfaz, extensión de navegador y servicios de backend (si aplican).
- Pruebas de seguridad: pruebas específicas para vectores comunes (XSS, CSRF, gestión de tokens, almacenamiento seguro en el cliente).
- Pruebas de usabilidad: sesiones con usuarios representativos durante el piloto para ajustar la interfaz.
- Pruebas de rendimiento básicas: medir tiempos de respuesta en operaciones clave.

4.3 Estrategia de Ejecución

- Integración de pruebas automatizadas en un pipeline de CI para ejecución con cada actualización crítica.
- Ejecución de ciclos de pruebas en entornos de prueba controlados antes del despliegue a producción.

4.4 Automatización de Pruebas

Se propone utilizar Selenium WebDriver para automatizar pruebas de la interfaz web, complementado por un framework de pruebas (ej. pytest + Selenium en Python, o Jest + WebDriver en JavaScript).

Alcance inicial de automatización:

- Login y autenticación (2FA): casos de éxito y fallo.
- Creación/edición/eliminación de elementos en la bóveda.
- Verificación de políticas (ej. rechazo de contraseñas que no cumplen la complejidad mínima).

Estrategia de ejecución:

- Ejecutar el conjunto crítico en el pipeline CI en cada build relevante.
- Ejecutar suites extendidas en entornos de integración o staging antes del despliegue general.

4.5 Análisis Estático de Código

- Detectar de forma automatizada defectos, vulnerabilidades, malos olores de código y violaciones de estilo antes de la integración en ramas principales, reduciendo riesgos de seguridad y mantenimiento.
- Herramientas:
- Análisis de seguridad SAST: SonarQube / SonarCloud, CodeQL (GitHub), Semgrep.
- Escaneo de dependencias y vulnerabilidades: Snyk, Dependabot, npm audit, Retire.js.
- Detección de secretos: GitLeaks, truffleHog.

4.6 Gestión de Defectos

Definición y Propósito:

La gestión de defectos consiste en la identificación, registro, análisis, corrección y prevención de errores que surgen durante el ciclo de vida del software. Su objetivo principal es mantener y mejorar la calidad del producto, garantizando la estabilidad, seguridad y usabilidad del sistema mediante un control continuo y documentado de incidencias.

La gestión de defectos en este proyecto se realizará mediante Jira Software Cloud, integrado con el repositorio GitHub del cliente web de Bitwarden. Cada incidencia generada desde el pipeline CI/CD (GitHub Actions) se registrará automáticamente en Jira con los siguientes campos mínimos:

- **ID del defecto:** generado por Jira.
- **Resumen:** descripción breve del fallo detectado.
- **Componentes:** módulo afectado (App cliente, SSO, API, DB, etc.).
- **Prioridad:** Crítica / Alta / Media / Baja.
- **Tipo de error:** funcional, de seguridad, rendimiento o usabilidad.
- **Pasos para reproducir:** instrucciones detalladas o enlace al log del pipeline.
- **Evidencias:** capturas, vídeo o enlace al reporte automatizado.
- **Estado:** “Open”, “In Progress”, “Resolved”, “Closed”.
- **Responsable:** miembro asignado del equipo QA o Dev.
- **Versión afectada / corregida.**

El pipeline CI generará reportes automáticos que se enlazarán con los tickets correspondientes para facilitar la trazabilidad y priorización por parte del equipo de desarrollo y QA.

Base Normativa:

Este proceso se apoya en tres estándares internacionales:

- **IEEE 1044:** establece la clasificación y gestión estandarizada de incidentes y defectos.
- **ISO/IEC 12207:** define los procesos del ciclo de vida del software, incluyendo mantenimiento, verificación y validación.
- **ISO 9001:** introduce principios de mejora continua y gestión documentada de no conformidades.

El plan adopta buenas prácticas de estas normas para asegurar trazabilidad, registro, cierre y mejora continua de los defectos detectados durante el ciclo de desarrollo.

Flujo de Gestión de Defectos

El flujo de gestión de defectos se desarrolla en cinco etapas, integradas al entorno Jira–GitHub:

1. **Detección y Registro:** Incidencias detectadas en pruebas o CI/CD.
2. **Clasificación:** Prioridad (crítico, mayor, menor) y tipo (funcional, seguridad, rendimiento).
3. **Asignación y Resolución:** Se asigna responsable en Jira; se enlaza con el commit o PR en GitHub.
4. **Verificación y Cierre:** QA valida la corrección; se actualiza el estado del ticket.
5. **Análisis de Causa Raíz:** Identificación de causas recurrentes y acciones preventivas.

Integración Herramientas Jira–GitHub:

Cada defecto en Jira se vincula automáticamente con commits, ramas o pull requests en GitHub. Esto permite una trazabilidad total entre el código corregido, las pruebas ejecutadas y el ticket de calidad.

Ejemplo de commit:
`GCS-15: Corrección de validación en login 2FA`
se asocia automáticamente al ticket GCS-15 en Jira.

La sincronización bidireccional GitHub Issue ↔ Jira Ticket permite visualizar defectos pendientes, métricas de densidad y tiempo medio de corrección (MTTR).

Métricas de Calidad:

Se propone trabajar con las siguientes métricas:

- Densidad de defectos: Defectos / KLOC.
- Tasa de reapertura: % de defectos reabiertos.

- MTTR (Mean Time To Repair): tiempo promedio desde detección hasta cierre.

Métrica	Descripción	Fórmula	Objetivo
Densidad de defectos	Número de defectos por KLOC	# Defectos / 1000 líneas de código	< 0.5
MTTR	Tiempo promedio de corrección	Σ tiempos de resolución / # defectos	< 3 días
Tasa de reapertura	Porcentaje de defectos reabiertos	(# reabiertos / # cerrados) × 100	< 5%

Mejora Continua y Prevención:

El proceso de mejora continua se rige por el ciclo PDCA (Plan–Do–Check–Act), conforme a ISO 9001. Al cierre de cada sprint, los reportes de defectos se analizan para proponer acciones preventivas y ajustes en los casos de prueba, fomentando el aprendizaje organizacional y la calidad sostenible.

Evidencias y Trazabilidad:

Todos los defectos y evidencias estarán registrados en Jira y enlazados con commits en GitHub y reportes CI/CD. Los resultados de pruebas automatizadas (Allure, JUnit XML, SonarQube) se almacenarán como artefactos de calidad, garantizando trazabilidad entre requerimientos, defectos, correcciones y validaciones.

4.7 Validación y Aceptación de Pruebas

Objetivo:

El objetivo de esta etapa es confirmar que el sistema implementado cumple con las expectativas de los usuarios finales (Backus S.A.) y que satisface los requisitos de negocio definidos. Se busca asegurar la conformidad del producto antes de su despliegue a producción, siguiendo las recomendaciones de las normas IEEE 1012 y ISO/IEC 25010 sobre verificación y validación de software.

Enfoque Metodológico:

La validación se realizará mediante un enfoque **Behavior Driven Development (BDD)**, utilizando **criterios Gherkin** como lenguaje común entre usuarios de negocio, QA y desarrolladores.
Cada historia de usuario o requisito funcional estará asociada a escenarios Gherkin que describen el comportamiento esperado del sistema.

Ejemplo de Criterios Gherkin:

A modo de ejemplo, la autenticación 2FA se valida mediante escenarios definidos en formato Gherkin (ver sección 5.2 para el detalle completo)

Criterios de Aceptación del Piloto:

Los criterios mínimos para la aceptación de la fase piloto incluirán:

- **Cumplimiento de autenticación 2FA**: ≥ 99% de intentos válidos exitosos.
- **Tiempo de respuesta**: dentro de los umbrales definidos (< 2 segundos).
- **Defectos críticos**: ausencia total en la versión piloto.
- **Cobertura de pruebas automatizadas**: al menos 80% del código.
- **Satisfacción del usuario**: confirmación por parte del área de seguridad y TI de Backus.

Validación y Registro de Resultados:

El resultado de cada caso de prueba será registrado en el sistema Jira o TestRail, con los siguientes campos:

- ID del caso / historia de usuario.
- Escenario Gherkin asociado.
- Estado de ejecución (Passed / Failed).
- Evidencia (captura, log, o video).
- Fecha de ejecución y responsable.

Las pruebas exitosas se marcarán como “Validadas” y las fallidas generarán automáticamente un ticket de defecto, enlazado al commit o build afectado en GitHub.

Aceptación Formal:

Una vez superadas todas las pruebas y validados los criterios Gherkin, los usuarios de negocio y el área de QA firmarán el Acta de Aceptación, registrando la conformidad del producto en Jira.
Esta validación formal marca el cierre de la fase piloto y autoriza el paso a producción del cliente web Bitwarden corporativo.

Mejora Continua:

Los resultados de validación se analizarán durante la reunión de retrospectiva del sprint. A partir de ellos se identificarán oportunidades de mejora en los scripts de pruebas, los criterios Gherkin y los pipelines de CI/CD, siguiendo el modelo PDCA (Plan–Do–Check–Act) y las directrices de ISO 9001.

5. Casos de prueba

5.1 Casos funcionales

Los casos funcionales constituyen la base del proceso de validación y verificación del sistema, asegurando que las funcionalidades críticas del cliente web Bitwarden operen conforme a los requisitos definidos por la organización.

Cada caso de prueba se expresa mediante el lenguaje Gherkin, siguiendo el enfoque BDD (Behavior Driven Development), lo que permite una comunicación clara entre los equipos de desarrollo, QA y los usuarios de negocio de Backus S.A.

Los escenarios seleccionados en esta fase piloto se enfocan en la autenticación segura de usuarios mediante 2FA, una funcionalidad esencial para garantizar la integridad y confidencialidad de las credenciales corporativas.

5.2 Casos funcionales

A continuación se incluyen los escenarios Gherkin disponibles en el borrador inicial para la autenticación con 2FA:

Feature: Autenticación segura del usuario en la bóveda

Scenario: Inicio de sesión exitoso con 2FA habilitado
Given un usuario con 2FA activado accede a la página de login
When introduce su email y contraseña maestra correctos
And introduce el código 2FA válido de su aplicación de autenticación
Then el sistema le concede acceso a su bóveda personal.

Scenario: Fallo de inicio de sesión con código 2FA incorrecto
Given un usuario con 2FA activado accede a la página de login
When introduce su email y contraseña maestra correctos
And introduce un código 2FA inválido
Then el sistema muestra un mensaje de error y deniega el acceso.

5.3 Trazabilidad entre Requisitos y Casos de Prueba

ID Requisito	Descripción del Requisito	ID Escenario Gherkin	Tipo de Prueba	Resultado Esperado
RF-001	El sistema debe autenticar usuarios mediante doble factor (2FA)	SCN-01	Funcional / Seguridad	Acceso autorizado solo con código válido
RF-002	El sistema debe registrar los intentos de acceso en logs	SCN-01 / SCN-02	Seguridad / Auditoría	Evento registrado en log
RF-003	El sistema debe denegar acceso con código inválido	SCN-02	Funcional / Seguridad	Acceso denegado y mensaje de error visible

5.4 Evidencias y Resultados Esperados

Resultado Esperado (SCN-01): acceso exitoso, log de auditoría con sello de tiempo y usuario válido.
Resultado Esperado (SCN-02): mensaje de error “Código inválido”, sin acceso a la bóveda y log de intento fallido.
Evidencias: capturas de pantalla, registros de ejecución automatizada (Allure / JUnit XML), y resultados almacenados en Jira–Xray.

5.5 Conclusión

Estos escenarios representan el conjunto inicial de casos funcionales críticos para la validación de seguridad en la etapa piloto.
Su correcta ejecución y trazabilidad dentro del pipeline CI/CD permitirán demostrar la conformidad del sistema con los requisitos de calidad definidos en el plan.

6. Herramientas y entornos tecnológicos

6.1 Tecnologías objetivo

- Bitwarden (cliente web).
- Selenium WebDriver (automatización de UI) y Selenium Grid o servicios equivalentes para ejecución paralela.
- Framework de pruebas (pytest, Jest/Mocha, o similar) para organización y reportes.
- CI/CD: GitHub Actions, Jenkins o equivalente para orquestar builds y ejecución de tests.
- Herramientas de análisis de dependencias y alertas de seguridad

(Dependabot, Snyk) para monitoreo de librerías de terceros.

6.2 Integración continua y automatización

Proceso propuesto

Integración de pruebas automatizadas en pipeline CI; alertas y reportes cuando se detecten fallos.

Beneficios

Detección temprana de regresiones, trazabilidad de resultados y visibilidad ejecutiva mediante reportes automatizados.

7. Planificación de la calidad

7.1 Introducción a la Planificación de la Calidad del Servicio

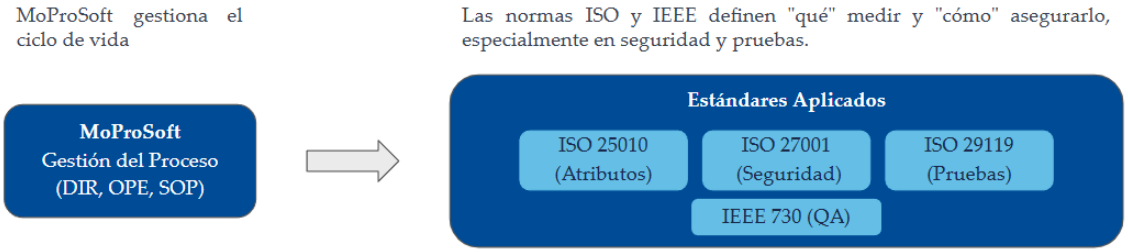
Esta sección constituye el plan integral de calidad para el servicio de despliegue, gobierno y operación del cliente web Bitwarden en Backus S.A. El propósito de este plan es establecer un marco de trabajo sistemático y auditable que asegure que el servicio no solo cumpla con los requisitos funcionales y técnicos, sino que también se alinee con los objetivos estratégicos de la organización. Dichos objetivos incluyen la mitigación de riesgos asociados a la gestión de credenciales, el fortalecimiento de la postura de seguridad de la información y la mejora de la productividad del personal mediante un acceso seguro y eficiente a los recursos corporativos.

El marco metodológico para la gestión de la calidad, se fundamenta en el modelo de procesos MoProSoft. Este modelo actúa como el esqueleto procesal sobre el cual se articulan y gestionan los estándares internacionales específicos que gobiernan cada faceta de la calidad. La estructura de este plan se ha diseñado para abordar la calidad de manera holística, integrando las siguientes perspectivas normativas:

- Aseguramiento de la Calidad (IEEE 730): Define el gobierno general, los roles, las responsabilidades y las actividades de control.
- Calidad del Servicio (ISO/IEC 25010): Establece un vocabulario común y un modelo para definir qué significa "calidad" en el contexto del servicio Bitwarden.
- Seguridad de la Información (ISO/IEC 27001): Proporciona los controles y políticas para proteger el activo más crítico gestionado por el servicio: las credenciales.
- Verificación y Validación (ISO/IEC/IEEE 29119): Estandariza el proceso mediante el cual se verifica que el servicio opera conforme a lo

especificado.

Las subsecciones subsiguientes detallan cada uno de estos componentes, culminando en un mapa de integración que demuestra cómo las actividades de calidad, los Acuerdos de Nivel de Servicio (SLA) y los Indicadores Clave de Desempeño (KPIs) se insertan de manera coherente en el modelo de procesos MoProSoft adoptado por la organización.



Este marco combina MoProSoft como armazón de procesos (para gestionar el servicio extremo a extremo) y un paquete de normas de calidad que aterriza los criterios técnicos y de aseguramiento requeridos por Backus S.A. para la implantación y gobernanza del cliente web de Bitwarden en contexto corporativo. Las cuatro piezas normativas del paquete ISO/IEC 25010, ISO/IEC 27001, ISO/IEC/IEEE 29119 e IEEE 730 cubren, respectivamente: calidad del producto/servicio, seguridad de la información (SGSI), pruebas de software, y plan de aseguramiento de la calidad. Esto permite que el servicio sea trazable, auditable y medible.

7.1.1 MoProSoft como marco de procesos

Enfoque y tamaño. MoProSoft fue diseñado para organizaciones latinoamericanas y equipos pequeños/medianos; agrupa un conjunto reducido y práctico de procesos en tres macro-capas (Dirección, Gerencia, Operación) que facilitan adopción rápida y control efectivo sin sobrerregulación. Su núcleo cubre gestión del negocio, de procesos, de proyectos, de recursos y del desarrollo/servicios de TI, con artefactos concretos (políticas, planes, bitácoras, métricas) y un ciclo explícito de mejora.

Alineación con estándares. La estructura de procesos de MoProSoft encaja con el lenguaje de ISO/IEC 12207 (procesos del ciclo de vida) y con los modelos de madurez (p.ej., CMMI), lo que facilita evolución posterior si Backus decide escalar el sistema de gestión de TI o someterlo a appraisal externo.

Ajuste al servicio. Para un servicio de implantación, configuración y gobernanza del cliente web de Bitwarden, MoProSoft evita burocracia y permite gestionar con claridad: alcance, riesgos, proveedores, activos, despliegues, soporte y mejora continua, manteniendo una cadena de trazabilidad desde la política corporativa hasta los cambios en el entorno del usuario final. Esto se integra naturalmente con las políticas de organización que Bitwarden ofrece (p. ej.,

obligatoriedad de 2FA, reglas de contraseñas, dominios verificados y SSO).

7.1.2 Mapa de procesos MoProSoft aplicado al servicio

Tabla 1. Mapa de procesos y aterrizaje en Backus S.A. (cliente web Bitwarden)

Capa	**Proceso MoProSoft (referencia)**	**Aterrizaje en Backus (servicio Bitwarden web)**	**Entregables clave**
Dirección	Gestión del negocio (DNG)	Patrocinio, objetivos del programa (reducción de riesgo por contraseñas débiles, cumplimiento), definición de indicadores.	Política corporativa de gestión de credenciales, OKR/KPI del servicio.
Gerencia	Gestión de procesos (GPR)	Definición/actualización del proceso de alta/baja de usuarios, flujo de solicitudes, cambios y mejoras.	Mapa de proceso, SIPOC, métricas de capacidad.
Gerencia	Gestión de proyectos (GPROJ)	Plan de despliegue por áreas; cronograma de capacitación y adopción; gestión de riesgos.	Acta, EDT, plan, matriz de riesgos.
Gerencia	Gestión de recursos (GRH/Infra/Activos)	Gobernanza de identidades y activos (grupos, políticas, dominios); coordinación con TI/Seguridad.	Inventario de activos y roles; matriz RACI.
Operación	Administración de servicios/implementación	Configuración del cliente web y políticas de organización en Bitwarden (2FA obligatoria, reglas de contraseñas, SSO, dominios verificados), soporte de primer nivel y catálogo de servicios.	Guía de configuración, runbooks, catálogo y SLAs.

Operación	Medición y mejora	Cuadro de mando (adopción, incidentes, tiempo de resolución, auditorías de acceso), revisión periódica y acciones de mejora.	Plan de medición, informes y backlog de mejora.

Nota: La nomenclatura de procesos puede cambiar según la versión, pero se mantienen los tres niveles (Dirección, Gerencia, Operación) y el enfoque de procesos compactos y ensamblables.

7.1.3 Paquete normativo de calidad

Tabla 2. Paquete de normas y uso operativo en el servicio

Norma	**Qué regula**	**Uso concreto en el servicio**
ISO/IEC 25010	Modelo de calidad del producto/servicio: 8 características (adecuación funcional, desempeño/eficiencia, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad) y subcaracterísticas.	Derivar requisitos no funcionales (NFRs) y criterios de aceptación para la experiencia web de Bitwarden (p. ej., desempeño de carga, accesibilidad, facilidad de aprendizaje, robustez ante fallos, protección contra uso no autorizado).
ISO/IEC 27001:2022	Sistema de Gestión de Seguridad de la Información (SGSI); Anexo A con 93 controles en 4 temas: organizacionales, personales, físicos y tecnológicos.	Políticas y controles para el uso corporativo de Bitwarden: 2FA obligatoria, gestión de acceso y roles, protección de credenciales, registro y monitoreo, respuesta a incidentes, continuidad.
ISO/IEC/IEEE 29119	Estándar de pruebas: procesos, técnicas y documentación (plan, diseño, casos, ejecución, reporte).	Definir el proceso de pruebas del servicio (funcionales y no funcionales), diseñar casos para políticas (p.ej., regla de contraseña), regresiones y reporting de resultados.

IEEE 730	Plan de Aseguramiento de Calidad de Software (SQAP): estructura mínima del plan, revisiones y auditorías, criterios de aceptación, independencia de QA.	Elaborar el SQAP del servicio: qué se revisa, cómo se audita, qué métricas se controlan y qué evidencia se conserva.

Nota: ISO/IEC 27001 versión 2022 redujo los dominios y reorganizó controles a 93 en 4 temas; 25010 define el vocabulario de calidad; 29119 estandariza procesos/plantillas de pruebas; IEEE 730 estructura el SQAP.

Ajuste al contexto Backus. Dado que el objetivo es centralizar y proteger credenciales con gobierno corporativo, el paquete cubre:

- Calidad observable por el usuario (25010)
- Cumplimiento y gestión del riesgo (27001)
- Verificación rigurosa del funcionamiento (29119)
- Aseguramiento sistemático y auditable (IEEE 730)

En conjunto, este set delimita qué se exige (calidad), cómo se protege (seguridad), cómo se verifica (pruebas) y cómo se asegura (QA), todo sobre el proceso articulado por MoProSoft.

7.1.4 Matriz de correspondencia (MoProSoft ↔ artefactos del paquete normativo)

La matriz conecta procesos con evidencias. Así se evita duplicidad y se asegura trazabilidad en auditorías internas/externas.

Tabla 3. Correlación de procesos y evidencias

Proceso MoProSoft	**Artefacto / actividad**	**Norma(s) de referencia**
DNG – Gestión del negocio	Política corporativa de credenciales (uso obligatorio de gestor, 2FA, SSO, dominios verificados)	ISO/IEC 27001 (A.5 política, A.8 control de acceso, A.8.2 gestión de identidades), Bitwarden Organization Policies.
GPR – Gestión de procesos	Procedimiento de alta/baja y flujo de cambios; ciclo de mejora	IEEE 730 (plan QA y revisiones); ISO/IEC 27001 (operación del SGSI)

GPROJ – Gestión de proyectos	Plan de implementación por áreas; matriz de riesgos; plan de capacitación	IEEE 730 (plan de aseguramiento); 29119 (plan de pruebas)
Operación – Administración del servicio	Configuración del cliente web y políticas en Bitwarden (enforcement 2FA, reglas de contraseñas, SSO)	ISO/IEC 27001 (controles de acceso y autenticación); Bitwarden Policies
Operación – Medición y mejora	KPIs: adopción, incidentes, tiempos de resolución, hallazgos de auditoría	IEEE 730 (métricas y auditorías), 27001 (monitorización/medición)
QA – Verificación y validación	Plan/diseño/ejecución de pruebas (funcionales, seguridad, rendimiento), regresión ante cambios de políticas	ISO/IEC/IEEE 29119 (procesos y documentación de pruebas)
Producto/Servicio – NFRs	Catálogo de NFRs (usabilidad, desempeño, seguridad, fiabilidad, mantenibilidad) y criterios de aceptación	ISO/IEC 25010 (características y subcaracterísticas)

Nota: La correlación es operativa se recomienda mantener una Matriz de Trazabilidad que enlace: requisito/NFR → control 27001 → caso de prueba 29119 → ítem del SQAP (IEEE 730)

7.1.5 Cómo se integra el paquete en el ciclo de vida

1. Planificar (DNG/GPROJ). Política corporativa, objetivos y KPIs; plan del proyecto y SQAP (IEEE 730).
2. Definir requisitos. Catálogo NFRs con 25010; criterios de aceptación.
3. Configurar y controlar. Parametrizar Bitwarden (2FA, contraseñas, SSO, dominios), gestión de cambios y de identidades (27001).
4. Verificar. Diseñar y ejecutar pruebas (29119): funcionales (políticas activas), no funcionales (rendimiento, usabilidad básica), regresión.
5. Asegurar y auditar. Revisiones/auditorías de QA (IEEE 730), evidencias de cumplimiento (27001) y reportes de métricas.
6. Mejorar. Análisis de KPIs, lecciones aprendidas y plan de mejora (MoProSoft – medición/mejora).

7.1.6 Por qué este conjunto y no alternativas internacionales (síntesis comparativa)

Tabla 4. Comparación operativa (conjunto propuesto vs. marcos alternativos)

Criterio	**Propósito central**	**Cobertura técnica inmediata**	**Esfuerzo de adopción (proyecto)**	**Ajuste al alcance Bitwarden–Backus**
Conjunto propuesto: MoProSoft + 25010 + 27001 + 29119 + 730	Gobernar proceso del servicio; calidad, seguridad, pruebas y SQA con artefactos específicos.	Alta: NFRs (25010), SGSI (27001), pruebas (29119), SQAP (730).	Bajo–medio (artefactos concretos, resultados rápidos).	Muy alto: políticas, roles, evidencias y métricas operativas.
CMMI v2.x	Mejora de capacidad y desempeño organizacional; *appraisals* formales.	Indirecta: requiere anexar seguridad, pruebas y calidad de producto.	Medio–alto (formación, *appraisal*, despliegue organizacional).	Medio: disciplina organizacional, pero no sustituye los requisitos técnicos.
ISO 9001:2015	QMS genérico para mejorar desempeño y satisfacción del cliente.	Indirecta: requiere anexar 25010/27001/29119/730.	Medio (definir QMS, auditorías).	Medio: mejora por procesos; sin detalle técnico de producto/seguridad/pruebas.
ISO/IEC 12207	Marco de procesos de ciclo de vida de software/sistemas.	Abstracción alta; requiere especificar calidad/seguridad/pruebas.	Medio (ingeniería de procesos para concreción).	Medio: útil como referencia; necesita complementos.
ITIL/COBIT	Buen gobierno/operación de TI y control.	Foco en servicio TI; requiere “aterrizar” a producto y evidencias.	Medio (gobierno y operación transversales).	Medio: útil para operación TI; requiere complementos de seguridad/prueba.

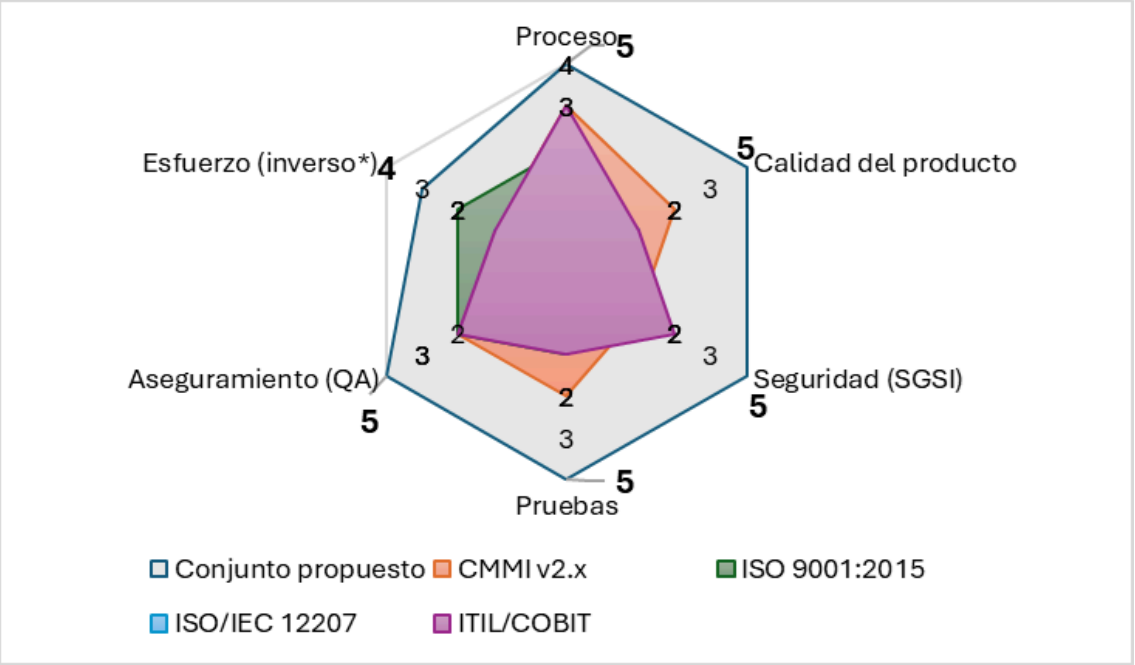
Nota: CMMI v2.x es un modelo de mejores prácticas para mejorar desempeño/capacidades y exige appraisals; ISO 9001 define requisitos de QMS transversales; ambos se pueden complementar con el paquete propuesto o adoptarse como trayectoria evolutiva posterior

Tabla 5. Estimación cualitativa basada en propósito y alcance normativo

Marco	Proceso	Calidad del producto	Seguridad (SGSI)	Pruebas	Aseguramiento (QA)	Esfuerzo (inverso*)
Conjunto propuesto	5	5	5	5	5	4
CMMI v2.x	4	3	2	3	3	2
ISO 9001:2015	3	2	2	2	3	3
ISO/IEC 12207	3	2	2	2	2	2
ITIL/COBIT	4	2	3	2	3	2

*Nota: Esfuerzo (inverso): 5 = adopción relativamente baja a nivel proyecto (artefactos concretos, resultados rápidos); 1 = adopción alta/lenta (entrenamiento, *appraisals*, despliegue transversal) y **Conjunto propuesto** (MoProSoft + ISO/IEC 25010 + ISO/IEC 27001 + ISO/IEC/IEEE 29119 + IEEE 730)

Figura 1. Cobertura vs. esfuerzo relativo por marco (1–5). Estimación cualitativa basada en propósito y alcance normativo



Nota: Elaboración propia

Conclusión operativa: Elegimos MoProSoft para nuestro proyecto y el despliegue/gobierno del cliente web de Bitwarden en Backus S.A., el conjunto propuesto ofrece la densidad metodológica justa: proceso ligero y evaluable (MoProSoft), calidad observable (25010), seguridad con controles actuales (27001:2022), pruebas repetibles (29119) y aseguramiento formal (IEEE 730). Con esto, Backus obtiene control operativo inmediato y base escalable hacia marcos corporativos de mayor alcance.

7.2 Plan de Aseguramiento de la Calidad del Servicio (SQAP)

Este Plan de Aseguramiento de la Calidad del Servicio (SQAP, por sus siglas en inglés) se desarrolla conforme a los lineamientos de la norma IEEE 730 - Standard for Software Quality Assurance Processes. El SQAP es el documento maestro que articula el conjunto de actividades planificadas y sistemáticas destinadas a garantizar que el ciclo de vida completo del servicio Bitwarden desde su despliegue hasta su operación y mejora continua cumpla con los estándares de calidad establecidos.

7.2.1 Objetivos del SQAP

Los objetivos específicos de este plan son:

- **Establecer Conformidad:** Garantizar que el servicio Bitwarden y sus procesos de soporte se adhieren a las políticas corporativas, los requisitos de negocio y los estándares normativos definidos (ISO/IEC 27001, ISO/IEC 25010).
- **Definir Procesos de Control:** Formalizar los procesos de revisión, auditoría, gestión de no conformidades y control de cambios para asegurar la integridad y calidad del servicio.
- **Proveer Visibilidad:** Generar evidencia documentada y métricas que proporcionen a la dirección una visibilidad clara sobre el estado de la calidad del servicio, facilitando la toma de decisiones informada.
- **Asegurar Objetividad:** Establecer una función de Aseguramiento de la Calidad (QA) con la independencia y autoridad necesarias para realizar evaluaciones objetivas, tal como lo recomienda la norma IEEE 730.

7.2.2 Roles y Responsabilidades (RACI)

Para garantizar una ejecución clara y efectiva de las actividades de calidad, se define la siguiente matriz de roles y responsabilidades (RACI). La estructura de roles se alinea directamente con las capas del modelo MoProSoft ya adoptado en (Dirección, Gerencia, Operación), asegurando una coherencia vertical entre el modelo de procesos de la organización y el plan de calidad del servicio. Esta alineación transforma la matriz de un simple artefacto a una herramienta de integración metodológica, donde cada actividad de calidad tiene un responsable claro dentro de la estructura de gobierno establecida.

Actividad de Calidad	Dirección (MoProSoft)	Gerencia (MoProSoft)	Operación (MoProSoft)	Seguridad de la Información
Definición y Aprobación del Plan de Calidad (SQAP)	A	R	C	C
Diseño y Aprobación de SLA/SLOs	A	R	C	I
Ejecución de Pruebas de Verificación y Validación (V&V)	I	I	R	C
Monitoreo y Reporte de SLIs/KPIs	I	A	R	I
Auditoría de Procesos de Calidad	A	R	C	R
Gestión de No Conformidades y Acciones Correctivas	I	A	R	C
Aprobación del Pase a Producción de Cambios	A	R	I	A

Críticos				

*Leyenda: **R** - Responsable (Responsible), **A** - Aprobador (Accountable), **C** - Consultado (Consulted), **I** - Informado (Informed).*

7.2.3 Normas ISO/IEEE Aplicables

El plan de calidad se sustenta en estándares internacionales que permiten establecer métricas, procesos y criterios de validación claros. Para el caso del cliente web de Bitwarden en Backus se consideran:

Norma	**Descripción**	**Aplicación**
ISO/IEC 25010	Modelo de calidad del software que define atributos como seguridad, confiabilidad, usabilidad, mantenibilidad y portabilidad.	Se usa para establecer métricas de calidad y asegurar cumplimiento de requisitos no funcionales.
ISO/IEC 27001	Sistema de Gestión de Seguridad de la Información.	Orienta la gestión de riesgos relacionados con credenciales, llaves RSA y almacenamiento seguro de datos.
ISO/IEC/IEEE 29119	Estándar internacional para pruebas de software.	Estructura la planificación, diseño, ejecución y documentación de pruebas.
IEEE 730	Estándar para planes de aseguramiento de la calidad.	Base documental para elaborar el plan QA formal de la organización.

7.3 Modelo de Calidad y Atributos del Servicio (ISO/IEC 25010)

Para definir de manera objetiva, medible y estandarizada qué constituye un "servicio de calidad" en el contexto de la implementación de Bitwarden, se adopta el modelo de calidad de producto definido en la norma **ISO/IEC 25010**.[7] Este estándar proporciona un marco de ocho características que descomponen

el concepto abstracto de "calidad" en atributos evaluables.

Si bien todas las características son relevantes, para un servicio centrado en la gestión de credenciales corporativas, se deben priorizar aquellas que impactan directamente en el riesgo, la confianza y la adopción por parte del usuario. Las características prioritarias para el servicio Bitwarden en Backus S.A. son:

1. **Seguridad:** Es la característica más crítica. El propósito fundamental del servicio es proteger los activos de información. Las sub-características de *Confidencialidad*, *Integridad*, *No Repudio*, *Rendición de Cuentas* y *Autenticidad* son el núcleo de la propuesta de valor.

2. **Fiabilidad:** El servicio debe ser resiliente y estar disponible cuando los usuarios lo necesiten para realizar sus tareas. La interrupción del servicio no solo afecta la productividad, sino que puede incentivar a los usuarios a recurrir a prácticas inseguras. Las sub-características clave son *Disponibilidad* y *Tolerancia a Fallos*.

3. **Usabilidad:** Una alta adopción es crucial para el éxito del programa de gestión de credenciales. Si el servicio es complejo o poco intuitivo, los usuarios buscarán alternativas, anulando los beneficios de seguridad. Las sub-características de *Reconocimiento de la Adecuación*, *Aprendizaje* y *Operabilidad* son fundamentales.

4. **Eficiencia de Desempeño:** Para que la herramienta se integre de forma transparente en el flujo de trabajo del usuario, debe ofrecer tiempos de respuesta rápidos, especialmente en operaciones frecuentes como la autenticación y el autocompletado de credenciales. La sub-característica de *Comportamiento Temporal* es la más relevante.

La siguiente tabla establece un puente lógico entre estos atributos de calidad abstractos y los indicadores concretos que se utilizarán para medir el desempeño del servicio. Este mapeo asegura que los compromisos operativos definidos en el SLA no son arbitrarios, sino que se derivan directamente de un modelo de calidad estándar y de las prioridades del negocio.

Característica (ISO/IEC 25010)	**Sub-característica**	**Relevancia para el Servicio Bitwarden**	**Métrica / Indicador de Nivel de**

			Servicio (SLI) Asociado
Seguridad	Autenticidad	Garantizar que solo los usuarios legítimos accedan a sus bóvedas, especialmente mediante la aplicación de políticas como 2FA.	Tasa de éxito en autenticación con 2FA.
Fiabilidad	Disponibilidad	Asegurar que el servicio esté operativo y accesible para los usuarios durante el horario laboral establecido.	Disponibilidad del Servicio (%).
Fiabilidad	Recuperabilidad	Garantizar que, ante un incidente mayor, el servicio pueda ser restaurado dentro de un tiempo aceptable para el negocio.	Tiempo de Restauración del Servicio (MTTR).
Eficiencia de Desempeño	Comportamiento Temporal	Asegurar una experiencia de usuario fluida y sin demoras en las interacciones clave con la aplicación web y la extensión.	Tiempo de respuesta de la API de login (percentil 95).
Usabilidad	Aprendizaje	Facilitar la adopción rápida por parte de los empleados, minimizando la curva de aprendizaje y la necesidad de soporte extensivo.	Nivel de Satisfacción del Usuario (CSAT) post-capacitación.
Mantenibilidad	Modificabilidad	Asegurar que los cambios en las políticas de seguridad (p. ej., complejidad de contraseñas) se puedan aplicar sin introducir defectos.	Tasa de éxito en pruebas de regresión post-cambio.

La matriz de pruebas se construyó considerando el modelo STRIDE del esquema de arquitectura (App Cliente, SSO Service, Identity Service, Key Connector y bases de datos).

ID	Componente	Amenaza (STRIDE)	Norma asociada	Tipo de Prueba	Caso de Prueba	Resultado Esperado	Prioridad	Estado
T1	App Cliente	S (Spoofing)	ISO 27001 / ISO 25010	Prueba de autenticación	Intento de login con identidad falsa	Sistema rechaza el acceso y genera log	Alta	Pendiente
T5	App Cliente	T (Tampering)	ISO 27001 / ISO 29119	Prueba de integridad	Modificación de datos locales de la bóveda	Datos cifrados no son alterados, sistema detecta inconsistencia	Alta	Pendiente
T11	SSO Service	R (Repudiation)	ISO 25010 / IEEE 730	Auditoría y logging	Usuario intenta negar transacción	Registro en logs evidencia la operación	Media	Pendiente
T22	Identity Service	I (Information Disclosure)	ISO 27001	Prueba de confidencialidad	Intento de acceso a datos sin permisos	Datos no son expuestos, acceso denegado	Alta	Pendiente
T36	Key Connector	D (Denial of Service)	ISO 25010 / ISO 29119	Stress test	Inyección de múltiples solicitudes simultáneas	El sistema mantiene disponibilidad ≥ 99.9%	Alta	Pendiente
T42	SQL/Postgres	E (Elevation of Privilege)	ISO 27001 / ISO 25010	Prueba de autorización	Usuario normal intenta ejecutar acción de administrador	Acceso denegado, evento registrado	Crítica	Pendiente

7.4 Acuerdos de Nivel de Servicio (SLA)

El Acuerdo de Nivel de Servicio (SLA) es el documento contractual que formaliza los compromisos de calidad entre el proveedor del servicio (el equipo de TI de Backus) y sus consumidores (los usuarios y las unidades de negocio). Su propósito es establecer expectativas claras y definir umbrales medibles para el desempeño operativo del servicio. La estructura se inspira en el documento guía, pero su contenido se adapta a un contexto de servicio interno.

7.4.1 Especificación General del SLA

Componente	Descripción

Nombre del Servicio	Servicio de Gestión de Credenciales Corporativas (Bitwarden)
Partes Involucradas	Proveedor: Gerencia de TI de Backus S.A. Consumidores: Todos los empleados de Backus S.A.
Período de Vigencia	Anual, con revisiones trimestrales.
Alcance del Servicio	Provisión, configuración, soporte y gobierno del cliente web y extensiones de navegador de Bitwarden. Incluye la gestión de políticas de seguridad, bóvedas organizacionales y soporte a usuarios.
Horario de Soporte	Lunes a Viernes, de 8:00 a 18:00 (hora local), excepto feriados.
Canales de Comunicación	Portal de Autoservicio de TI (para solicitudes y tickets), Correo electrónico (para comunicaciones generales).
Proceso de Escalado	Definido en el sistema de gestión de tickets, escalando desde Soporte N1 hasta el Gerente de Operaciones de TI según la prioridad y el tiempo de resolución.

7.4.2 Indicadores (SLIs) y Objetivos (SLOs)

Los Indicadores de Nivel de Servicio (SLI) son las métricas específicas que se miden, mientras que los Objetivos de Nivel de Servicio (SLO) son los umbrales que el servicio se compromete a cumplir. A diferencia de los SLAs con proveedores externos, que a menudo implican penalizaciones financieras, en este contexto interno las "consecuencias por incumplimiento" se reinterpretan como acciones de gestión y mejora. Este enfoque es más realista y constructivo, ya que activa procesos de análisis y rendición de cuentas en lugar de imponer multas. Los siguientes SLOs se derivan de los atributos de calidad priorizados y se basan en umbrales plausibles para un servicio crítico.

ID	Atributo de Calidad (ISO 25010)	Indicador (SLI)	Objetivo (SLO)	Frecuencia de Medición	Fuente de Datos	Consecuencia por Incumplimiento
SLO-01	Fiabilidad	Disponibilidad del Servicio	$ \geq 99.5% $ (en horario laboral)	Mensual	Herramienta de Monitoreo Sintético (p.ej., UptimeRobot, Grafana)	1. Notificación automática al Líder de Operaciones. 2. Análisis Causa Raíz (RCA) obligatorio si el incumplimiento se repite en 2 meses consecutivos.
SLO-02	Eficiencia de Desempeño	Tiempo de Respuesta de Login (p95)	$ \leq 300 \text{ ms} $	Semanal	Herramienta de Monitoreo de Rendimiento de Aplicaciones (APM)	1. Alerta al equipo de Operaciones. 2. Investigación requerida si el umbral se supera por más de 1 hora continua.
SLO-03	Fiabilidad	Tiempo de Restauración del Servicio (Incidente P1)	$ \leq 4 \text{ horas} $	Por incidente	Sistema de Gestión de Tickets (p.ej., Jira Service Management)	1. Escalado inmediato al Gerente de TI. 2. Reunión de Post-mortem obligatoria dentro de las 72 horas

						posteriores al incidente.
SLO-04	Usabilidad	Tiempo de Primera Respuesta (Soporte N1)	$ \leq 1 \text{ hora} $ (en horario laboral)	Por ticket	Sistema de Gestión de Tickets	1. Notificación al Líder de Soporte al Usuario. 2. Inclusión en el reporte semanal de desempeño del equipo de soporte.

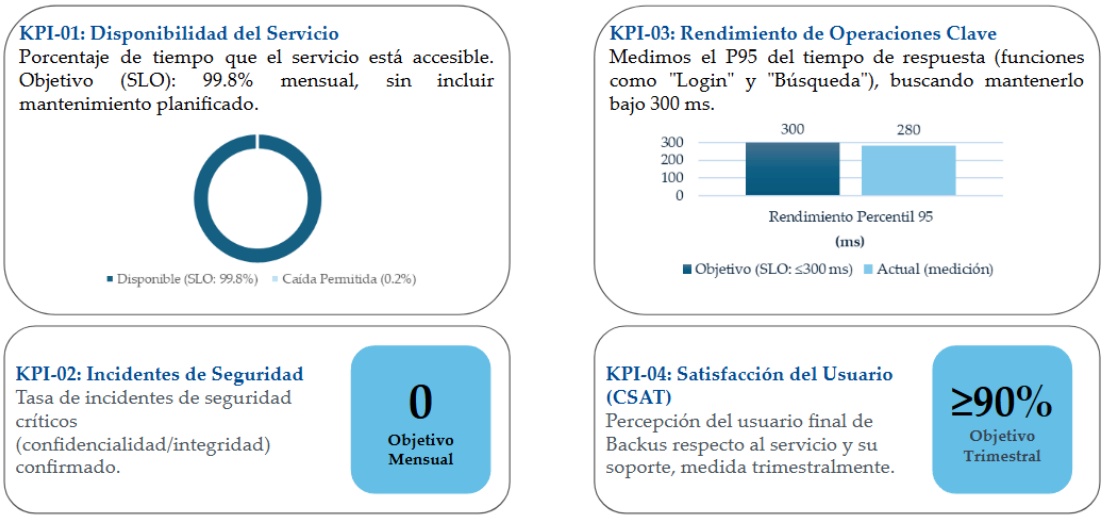
7.5 Indicadores Clave de Desempeño (KPIs) del Servicio

Mientras que los SLOs miden la salud y el rendimiento operativo del servicio (¿funciona correctamente?), los Indicadores Clave de Desempeño (KPIs) miden su impacto y valor para el negocio (¿está logrando sus objetivos estratégicos?). Esta distinción es crucial para comunicar el valor de TI a la dirección. Siguiendo las buenas prácticas de medición, que recomiendan priorizar un conjunto reducido de métricas relevantes en lugar de intentar medirlo todo, se proponen los siguientes KPIs. Estos indicadores están diseñados para responder directamente a las justificaciones de negocio para la implementación de Bitwarden en Backus S.A.

La estructura del siguiente cuadro de mando se inspira en el capítulo de indicadores del documento guía 3, asegurando la alineación con las expectativas académicas del proyecto.

ID	KPI	Descripción	Objetivo de Negocio Asociado	Fórmula de Cálculo	Meta	Frecuencia de Reporte	Responsable (RACI)
KPI-01	**Tasa de Adopción de Usuarios**	Porcentaje de empleados con cuentas activas que han configurado y utilizan regularmente su bóveda Bitwarden.	Maximizar el uso de la herramienta para reducir la superficie de riesgo global de la organización.	$ (\frac{\text{Nº usuarios activos}}{\text{Nº total de empleados}}) \times 100 $	> 90% al final del primer año.	Trimestral	Gerencia (A), Operación (R)
KPI-02	**Reducción de Incidentes por Credenciales**	Disminución porcentual en el número de incidentes de seguridad cuya causa raíz fue una contraseña débil, reutilizada o comprometida, en comparación con el período base pre-implementación.	Justificar la inversión en el servicio a través de la mitigación tangible de riesgos de seguridad.	$ 1 - (\frac{\text{Incidentes}_{\text{periodo actual}}}{\text{Incidentes}_{\text{periodo base}}}) $	Reducción del 50% en 18 meses.	Semestral	Dirección (A), Seg. Info. (R)

ID	KPI	Descripción	Objetivo de Negocio Asociado	Fórmula de Cálculo	Meta	Frecuencia de Reporte	Responsable (RACI)
KPI-03	**Nivel de Satisfacción del Usuario (CSAT)**	Puntuación promedio de satisfacción de los usuarios con el servicio (facilidad de uso, soporte, rendimiento), medida a través de encuestas post-soporte y una encuesta anual.	Asegurar que el servicio es percibido como un habilitador de productividad y no como un obstáculo burocrático.	$ \frac{\text{Suma de puntuaciones}}{\text{Nº de respuestas}} $	$ \geq 4.0 / 5.0 $	Continuo / Anual	Gerencia (A), Operación (R)



7.6 Plan de Verificación y Validación (V&V)

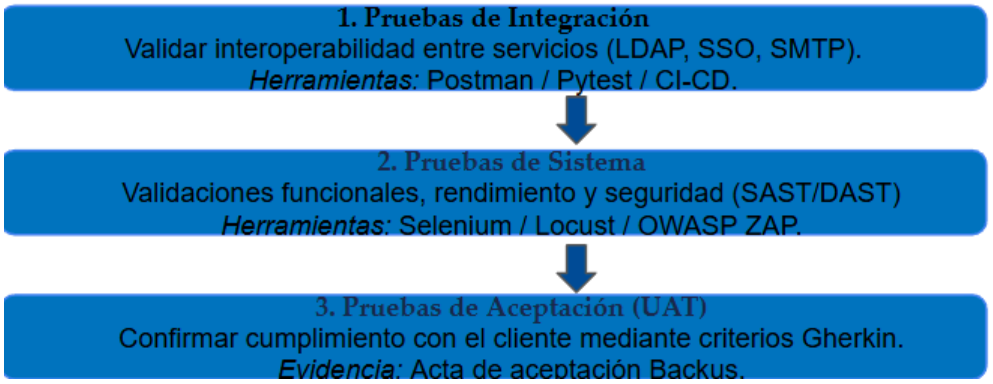
Las actividades de Verificación y Validación (V&V) son esenciales para confirmar que el servicio Bitwarden se implementa y opera de acuerdo con los requisitos especificados y satisface las necesidades de la organización. Este plan se enmarca en los procesos y la terminología del estándar ISO/IEC/IEEE 29119 - Software Testing, que proporciona un enfoque estructurado para la planificación, diseño, ejecución y documentación de las pruebas.

La estrategia de V&V para este servicio es un proceso continuo, no un evento único. Las actividades de prueba se aplican no solo durante el despliegue inicial, sino también durante la fase operativa para validar cambios de configuración, nuevas políticas de seguridad o actualizaciones de la plataforma. Cada tipo de prueba está diseñado para verificar el cumplimiento de los atributos de calidad definidos en la sección 7.3, creando una trazabilidad directa entre lo que se define como calidad y cómo se comprueba.

5. Verificación y Validación (V&V)

Calidad garantizada por ISO 29119. Pruebas UAT verifican requisitos ISO 27001.

Estrategia de Pruebas (Flujo ISO 29119)



La siguiente tabla resume los niveles y tipos de prueba clave para el servicio.

Nivel de Prueba	Objetivo Principal	Tipos de Prueba Aplicables	Técnicas Clave (ISO 29119-4)	Criterios de Aceptación/Salida	Artefacto Principal (ISO 29119-3)
Pruebas de Sistema	Validar que la configuración de Bitwarden en el entorno de Backus cumple con todos los requisitos funcionales, de seguridad y de rendimiento definidos.	Funcionales (políticas de contraseñas, 2FA), de Seguridad (controles ISO 27001), de Rendimiento (SLOs).	Pruebas basadas en especificación (casos de uso), pruebas de transición de estado (para flujos de autenticación), pruebas de carga.	100% de casos de prueba críticos ejecutados y aprobados. 0 vulnerabilidades de severidad Alta/Crítica. Cumplimiento de todos los SLOs de rendimiento.	Plan de Pruebas (PP-001), Casos de Prueba, Informe de Resultados de Pruebas.
Pruebas de Aceptación de Usuario (UAT)	Confirmar formalmente que el servicio es aceptable para los usuarios finales y cumple con las necesidades del negocio desde su perspectiva.	Pruebas basadas en escenarios de negocio, pruebas exploratorias.	Pruebas basadas en experiencia.	Tasa de éxito $ \geq 98% $ en la ejecución de escenarios de negocio clave. Aprobación formal firmada por el Business Owner.	Checklist de Aceptación, Acta de Conformidad de UAT.
Pruebas de Regresión	Asegurar que los cambios de configuración o las actualizaciones de la plataforma no han introducido defectos ni afectado negativamente la funcionalidad existente.	Pruebas funcionales automatizadas, pruebas manuales de funcionalidades críticas.	Re-ejecución de pruebas.	100% de la suite de regresión automatizada ejecutada con éxito. Verificación manual de los 5 flujos de usuario más críticos sin errores.	Suite de Pruebas de Regresión, Informe de Ejecución de Regresión.

7.7 Integración con el Marco de Procesos MoProSoft

Un plan de calidad solo es efectivo si está plenamente integrado en los

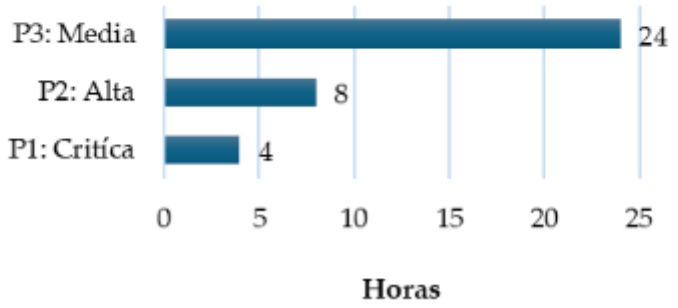
procesos de gestión de la organización. Esta sección garantiza que las actividades y artefactos definidos previamente no operen en un vacío, sino que sean gobernados y ejecutados dentro del marco de procesos MoProSoft ya adoptado.

El siguiente mapeo es la culminación de la estrategia de utilizar MoProSoft como el esqueleto del sistema de gestión. Demuestra explícitamente cómo cada componente del plan de calidad se ancla en un proceso específico de MoProSoft, creando un sistema de gestión de calidad coherente, trazable y auditable. Por ejemplo, la generación del informe mensual de SLA no es una tarea aislada, sino una salida formal del proceso de "Medición y mejora" de la capa de Operación.

4. Acuerdos de Nivel de Servicio (SLA)

Los SLA de Backus transforman los KPI en compromisos operativos.

Tiempos de Resolución de Incidentes (SLOs): Acuerdos de Nivel de Servicio (SLOs) para P1/P2.

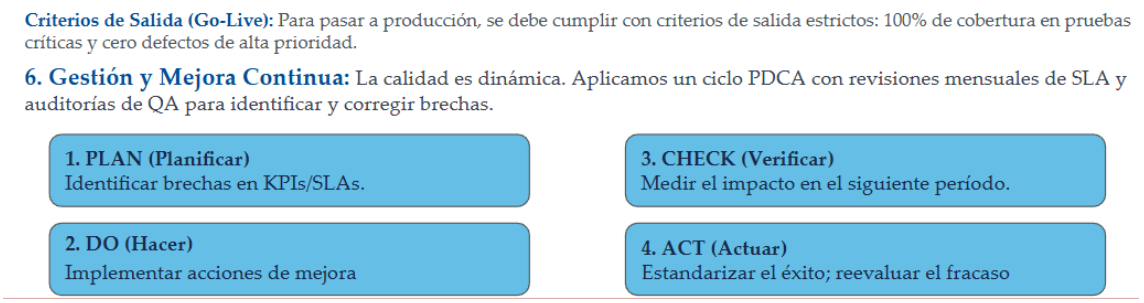


Capa MoProSoft	Proceso MoProSoft	Actividad de Calidad Asociada	Artefacto(s) de Calidad Generado(s)
Dirección	Gestión del negocio (DNG)	Establecimiento de los objetivos de negocio del servicio y aprobación de los KPIs estratégicos.	Cuadro de Mando de KPIs (aprobado).

Gerencia	Gestión de proyectos (GPROJ)	Planificación de las actividades de V&V para el despliegue inicial y para cambios mayores en el servicio.	Plan de Aseguramiento de Calidad (SQAP), Plan de Pruebas (PP-001).
Gerencia	Gestión de procesos (GPR)	Definición, aprobación y revisión periódica del SLA para asegurar su alineación con las necesidades del negocio.	Documento SLA.
Operación	Administración de servicios	Ejecución de pruebas de regresión como parte del proceso de gestión de cambios en la configuración de Bitwarden.	Informe de Resultados de Pruebas de Regresión, Registro de Defectos.
Operación	Medición y mejora	Generación, análisis y distribución de los reportes de cumplimiento de SLOs y desempeño de KPIs.	Informe Mensual de Desempeño del Servicio, Informe Trimestral de KPIs.

7.8 Artefactos y Cronograma de Hitos de Calidad

Esta sección final consolida los entregables documentales que se deben producir y mantener como parte de este plan de calidad, y establece una hoja de ruta temporal con los hitos más importantes.



7.8.1 Lista de Artefactos de Calidad

La siguiente es una lista de los artefactos clave que formalizan y evidencian la ejecución del plan de calidad. La nomenclatura se alinea con los formatos proporcionados para mantener la consistencia del proyecto.

- **SQAP-BITWARDEN-V1:** Plan de Aseguramiento de la Calidad del Servicio (este documento).
- **SLA-BITWARDEN-V1:** Acuerdo de Nivel de Servicio.
- **KPI-DASHBOARD-V1:** Cuadro de Mando de Indicadores Clave de Desempeño.
- **PP-001:** Plan de Pruebas (documento vivo, actualizado para la operación continua).
- **MATRIZ-PRUEBAS-BITWARDEN:** Matriz de Trazabilidad de Requisitos a Casos de Prueba.
- **CASOS-PRUEBA-BITWARDEN:** Repositorio de Casos de Prueba.
- **REG-DEFECTOS:** Registro y Bitácora de Defectos y No Conformidades.
- **INF-SLA-YYYY-MM:** Informes Mensuales de Cumplimiento de SLA.
- **INF-KPI-YYYY-QX:** Informes Trimestrales de Desempeño de KPIs.

7.8.2 Cronograma de Hitos de Calidad

Hito	Descripción	Entregable Asociado	Plazo / Frecuencia
H1	**Aprobación del Plan de Calidad y SLA**	Formalización de los compromisos y el marco de gobernanza de la calidad antes del despliegue masivo.	SQAP-BITWARDEN-V1, SLA-BITWARDEN-V1
H2	**Ejecución de Pruebas de Aceptación (UAT)**	Validación final por parte de representantes de negocio y usuarios clave antes del despliegue a toda la organización.	Acta de Conformidad de UAT
H3	**Primera Revisión de KPIs de Negocio**	Evaluación inicial del impacto del servicio en los objetivos estratégicos (adopción, seguridad).	INF-KPI-202X-QX
H4	**Auditoría Anual de Calidad**	Revisión formal del cumplimiento de los procesos definidos en el SQAP y la efectividad de los controles.	Informe de Auditoría de Calidad
H5	**Revisión y Ajuste Anual del SLA**	Ajuste de los SLOs y procesos basado en la experiencia operativa del año anterior y el feedback de los usuarios.	SLA-BITWARDEN-V(n+1)

8. Modelado de Amenazas del sistema Bitwarden

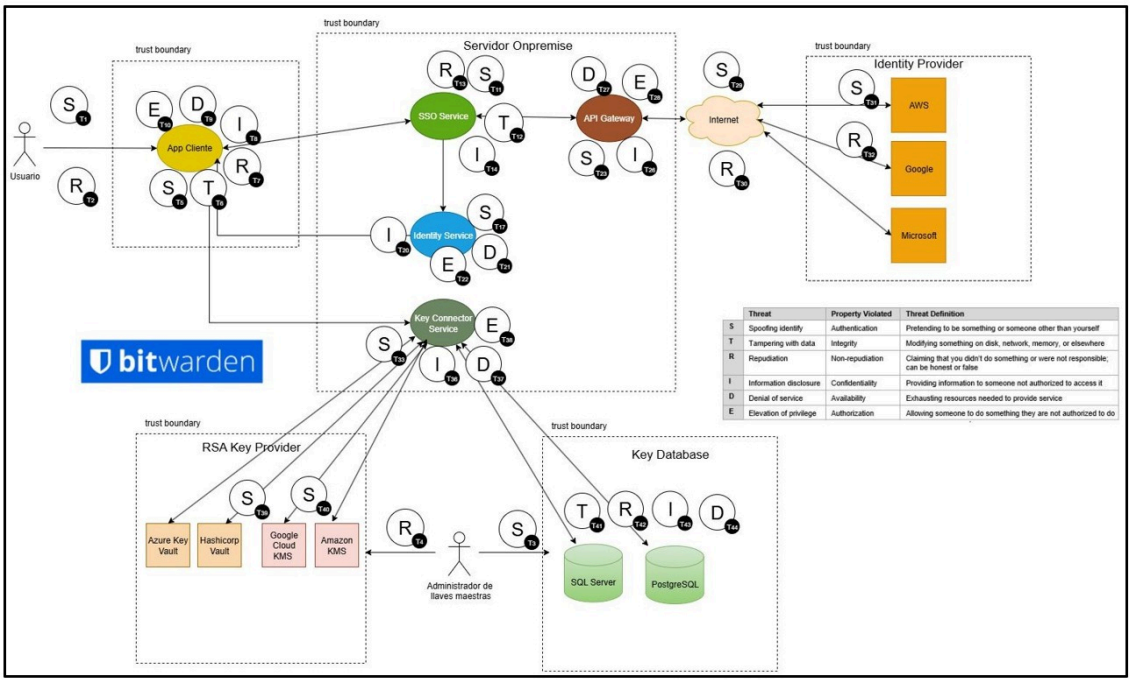
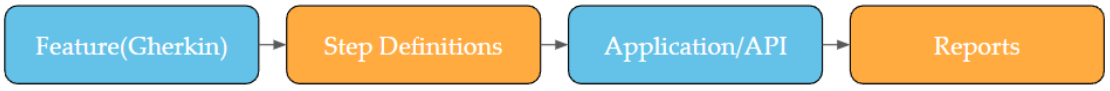


Figura 2: Modelado de amenazas STRIDE de software Bitwarden. Fuente: Elaboración Propia

8.1 Arquitectura de Automatización de Pruebas

Objetivo:"Reducir el tiempo de ejecución de pruebas y mejorar la trazabilidad con escenarios legibles por negocio"
Alcance:Pruebas Funcionales / Pruebas de Integración / Pruebas de Regresión

Tecnologías utilizadas
Lenguaje: Python
Framework: Cucumber + Behave (implementación de Gherkin en Python)
Librerías de soporte: pytest, selenium, requests, et



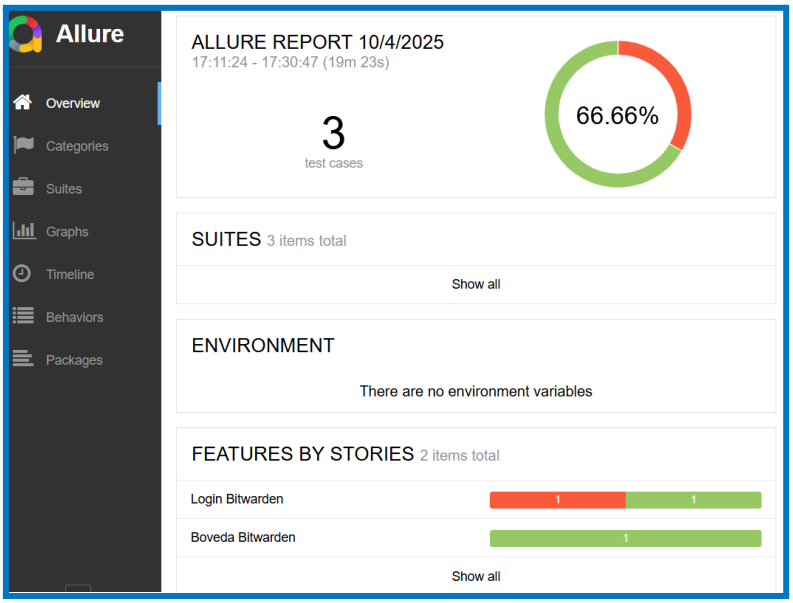
Diseño Escenarios / Estructura Gherkin

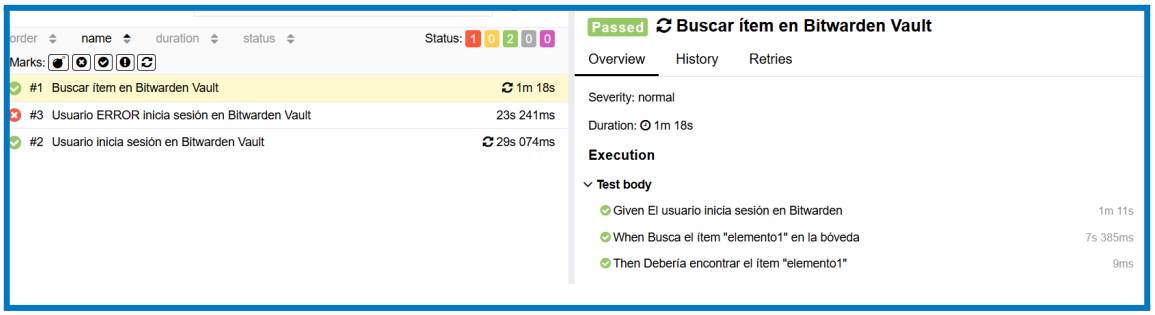
Ejecución de Escenarios / Implementación Steps

```gherkin
Feature: Login Bitwarden

  Scenario: Usuario inicia sesión en Bitwarden Vault
    Given El usuario abre la página de login de Bitwarden
    When Ingresa el correo "melissadrrs@gmail.com"
    And Hace clic en el botón de continuar
    And Ingresa la clave "XXXXXXXXX"
    And Hace clic en el botón de login
    Then Debería ver la página principal de Vault
```

```python
@given("El usuario abre la página de login de Bitwarden")
def step_open_login(context):
    context.driver.get("https://vault.bitwarden.com/")
    time.sleep(4)  # esperar a que cargue la página

@when('Ingresa el correo "{correo}"')
def step_enter_credentials(context, correo):
    context.driver.find_element(By.CSS_SELECTOR, "input[type='email']").send_keys(correo)

@when('Hace clic en el botón de continuar')
def step_enter_credentials(context):
    context.driver.find_element(By.CSS_SELECTOR, "button[buttontype='primary']").click()
    time.sleep(3)  # esperar a que cargue la página
```

Reportes





AssertionError

Categories: Product defects

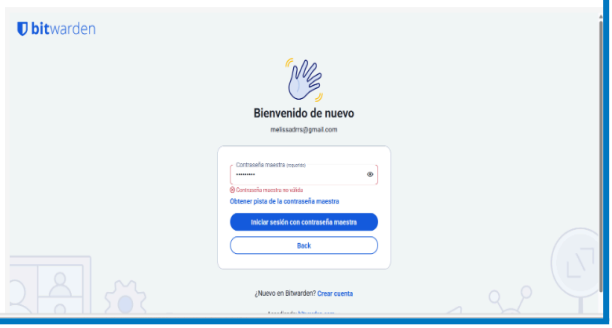
Severity: normal

Duration: 23s 241ms

Execution

Test body

- Given El usuario abre la página de login de Bitwarden — 4s 753ms
- When Ingresa el correo "melissadrrs@gmail.com" — 197ms
- And Hace clic en el botón de continuar — 3s 160ms
- And Ingresa la clave "claverror" — 57ms
- And Hace clic en el botón de login — 3s 072ms
- Then Debería ver la página principal de Vault 1 attachment — 12s
  - Debería ver la página principal de Vault — 88.6 KiB



9. Referencias Bibliograficas

Abu Bakar, N. S. A. (2025). Machine Learning Implementation in Automated Software Testing: A Review. Journal of Data Analytics and Artificial Intelligence Applications, 1(1), 110-122.

A. Fiegler, A. Zwanziger, S. Herden and R. R. Dumke, "Quality Measurement of ITIL Processes in Cloud Systems," 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), Berlin, Germany, 2016, pp. 87-94, doi: https://doi.org/10.1109/IWSM-Mensura.2016.022

A. Dávila and M. Pessoa, "Factors driving the adoption of ISO/IEC 29110: A case study of a small software enterprise," 2015 Latin American Computing Conference (CLEI), Arequipa, Peru, 2015, pp. 1-8, doi: 10.1109/CLEI.2015.7360042.

Awojana, T. B. (2018b). THREAT MODELLING AND ANALYSIS OF WEB APPLICATION ATTACKS.

Campos, E. J. M., Sanchez-Gordón, M.-L., & Colomo-Palacios, R. (n.d.). Article · January 2013 CITATIONS 4 READS 2,407 ISO/IEC 29110: Current overview of the standard. https://www.researchgate.net/publication/314346172

Casola, V., De Benedictis, A., Mazzocca, C., & Orbinato, V. (2024b). Secure software development and testing: A model-based methodology. Computers and Security, 137. https://doi.org/10.1016/j.cose.2023.103639

C. Y. Laporte, M. Muñoz and B. Gerançon, "The education of students about ISO/IEC 29110 software engineering standards and their implementations in very small entities," 2017 IEEE Canada International Humanitarian Technology Conference (IHTC), Toronto, ON, Canada, 2017, pp. 94-98, doi: https://doi.org/10.1109/IHTC.2017.8058208

Castillo-Salinas, L., Sanchez-Gordon, S., Villarroel-Ramos, J., & Sánchez-Gordón, M. (2020). Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment. Computer Standards and Interfaces, 70. https://doi.org/10.1016/j.csi.2020.103430

Charuenporn, P., & Intakosum, S. (n.d.). Qos-Security Metrics Based on ITIL and COBIT Standard for Measurement Web Services.

Gil-Gómez, H., Oltra-Badenes, R., & Adarme-Jaimes, W. (2014). Service quality management based on the application of the ITIL standard. *DYNA*, *81*(186), 51. https://doi.org/10.15446/dyna.v81n186.37953

García, L., Laporte, C. Y., Arteaga, J., & Bruggmann, M. (2015). Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru (Vol. 17, Issue ©). www.asq.org

Jyoti, S. N., Islam, M. R., & Kudapa, S. P. (2024). THE ROLE OF TEST AUTOMATION FRAMEWORKS IN ENHANCING SOFTWARE RELIABILITY: A REVIEW OF SELENIUM, PYTHON, AND API TESTING TOOLS. International Journal of Business and Economics Insights, 04(04), 01–34. https://doi.org/10.63125/bvv8r252

Muñoz, M., Mejia, J., Peña, A., Lara, G., & Laporte, C. Y. (2019). Transitioning international software engineering standards to academia: Analyzing the results of the adoption of ISO/IEC 29110 in four Mexican universities. Computer Standards and Interfaces, 66. https://doi.org/10.1016/j.csi.2019.03.008

Portela-Peñúñuri, L. T. (n.d.). Integrating ISO/IEC 29110 into Agile Workflows: A Practical Intervention Strategy for Very Small Entities.

Van, N., Tra, A., Saeed, A.-F., & Mohammed, I. (n.d.). International Journal of Computer Technology and Electronics Communication (IJCTEC) Software Testing Automation: Tools, Techniques, and Best Practices. A Peer-Reviewed, Refereed, and Biannual Scholarly Journal| ||, 8(1), 23. www.ijctece.com

Vives, L., Melendez, K., & Dávila, A. (2023). A Systematic Mapping Study of ISO/IEC 29110 and Software Engineering Education. Proceedings of the Institute for System Programming of the RAS, 35(1), 189–204. https://doi.org/10.15514/ispras-2023-35(1)-12

Yogesh Joshi Sr Manager, N. (n.d.). Implementing Automated Testing Frameworks in CI/CD Pipelines: Improving Code Quality and Reducing Time to Market. In International Journal on Recent and Innovation Trends in Computing and Communication. http://www.ijritcc.org

Z. Ateş and Y. E. Şencan, "A Case Study to Increase Quality of Industrial Edge Software Product's Dynamic Data Testing," 2023 International Conference on Software and System Engineering (ICoSSE), Marseille, France, 2023, pp. 25-29, doi: https://doi.org/10.1109/ICoSSE58936.2023.00013

MoProSoft®: A Software Process Model for Small Enterprises - ResearchGate, fecha de acceso: octubre 18, 2025, https://www.researchgate.net/publication/284024842_MoProSoftR_A_Software_Process_Model_for_Small_Enterprises

Moprosoft | PDF | Ingeniería de software - Scribd, fecha de acceso: octubre 18, 2025, https://www.scribd.com/document/389361445/moprosoft-docx

Formato excel de artefactos.xlsx

IEEE 730-2014 - IEEE SA, fecha de acceso: octubre 18, 2025, https://standards.ieee.org/ieee/730/5284/

IEEE 730-2014 - IEEE Standard for Software Quality Assurance Processes - ANSI Webstore, fecha de acceso: octubre 18, 2025, https://webstore.ansi.org/standards/ieee/ieee7302014

IEEE Standard 730-2014 Software Quality Assurance Processes - sqgne.org, fecha de acceso: octubre 18, 2025, http://sqgne.org/presentations/2014-15/Heimann-Apr-2015.pdf

What Is ISO 25010? | Perforce Software, fecha de acceso: octubre 18, 2025, https://www.perforce.com/blog/qac/what-is-iso-25010

ISO/IEC 25010, fecha de acceso: octubre 18, 2025, https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

ISO 25010 - Software Quality Requirements in the USA - Pacific Certifications, fecha de acceso: octubre 18, 2025, https://blog.pacificcert.com/iso-25010-software-quality-requirements-in-the-usa/

ISO 25010: Enhancing Our Software Quality Management Process - Helpware, fecha de acceso: octubre 18, 2025, https://helpware.com/blog/tech/iso-25010-enhancing-our-software-quality-management-process

Software Testing Standards | Setting the benchmark for Testing, fecha de acceso: octubre 18, 2025, https://softwaretestingstandard.org/