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

Contenido **2**
1. Introducción **3**
2. Perfil Empresarial 4
2.1 Nombre y Razón Social 4
3. Modelo de Servicio 4
3.1 Descripción General del Servicio 4
3.2 Alcance del Servicio 4
3.3 Flujo Operativo 4
3.4 Fases 5
Fase Pre-Operativa 5
Fase Operativa 5
4. Modelo de Pruebas 5
4.1 Objetivo del modelo 5
4.2 Tipos de Pruebas Incluidas 6
4.3 Estrategia de Ejecución 6
4.4 Automatización de Pruebas 6
4.5 Análisis Estático de Código 7
4.6 Gestión de Defectos 7
4.7 Validación y Aceptación de Pruebas 7
5. Casos de prueba 7
5.1 Casos funcionales 7
6. Herramientas y entornos tecnológicos 8
6.1 Tecnologías objetivo 8
6.2 Integración continua y automatización 8
Proceso propuesto 8
Beneficios 8
7. Planificación de la calidad **9**
7.1 Marco Metodológico de gestión de la calidad 9
7.1.1 MoProSoft como marco de procesos 9
7.1.2 Mapa de procesos MoProSoft aplicado al servicio 9
7.1.3 Paquete normativo de calidad 11
7.1.4 Matriz de correspondencia (MoProSoft ↔ artefactos del paquete normativo) 12
7.1.5 Cómo se integra el paquete en el ciclo de vida 13
7.1.6 Por qué este conjunto y no alternativas internacionales (síntesis comparativa) 13
7.2 Normas aplicables 15
7.2.1 Normas ISO/IEEE Aplicables 15
7.3 Cuadro de Pruebas de Calidad 16
8. Modelado de Amenazas del sistema Bitwarden **17**
9. Referencias Bibliograficas **18**

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

Los defectos serán registrados en el sistema de seguimiento de incidencias que utilice la organización (p. ej. Jira o GitHub Issues). Cada defecto incluirá:

- Pasos para reproducir
- Evidencias (logs, capturas de pantalla o vídeo)
- Severidad y prioridad
- Asignación a un responsable y estimación de corrección

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.

4.7 Validación y Aceptación de Pruebas

Los criterios de aceptación en formato Gherkin, incluidos en la sección 6, se usarán como referencia por los usuarios de negocio para la aceptación formal. Para cada criterio se definirá su trazabilidad hacia uno o más casos de prueba (manuales o automatizados) y se registrará el resultado de la verificación en el sistema de evidencias.

Los criterios mínimos de aceptación para completar la fase piloto incluirán: cumplimiento de autenticación 2FA en >99% de intentos válidos, tiempos de respuesta dentro de los umbrales definidos y ausencia de defectos de severidad crítica en la producción piloto.

5. Casos de prueba

5.1 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.

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 Marco Metodológico de gestión de la calidad

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.

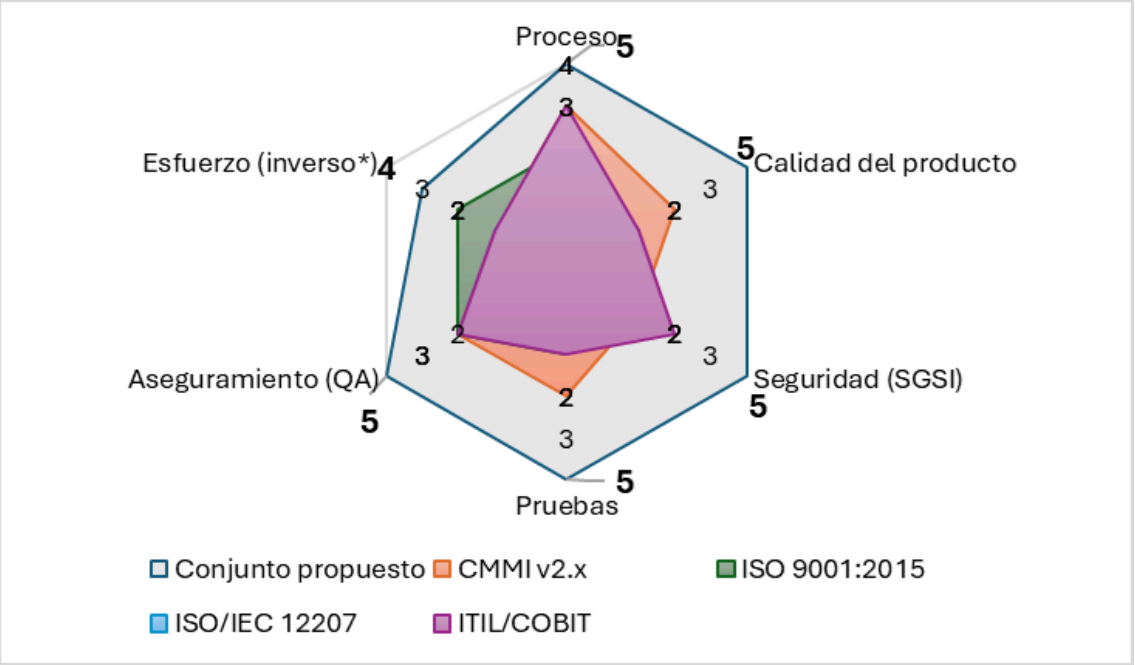
Nota: CMMI v2.x es un modelo de mejores prácticas para mejorar desempeño/capacidades y exige appraisals; ISO 9001 define requisitos de QMS transversales; ambos se pueden complementar con el paquete propuesto o adoptarse como trayectoria evolutiva posterior

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 Normas aplicables

7.2.1 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 Cuadro de Pruebas de Calidad

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

8. Modelado de Amenazas del sistema Bitwarden

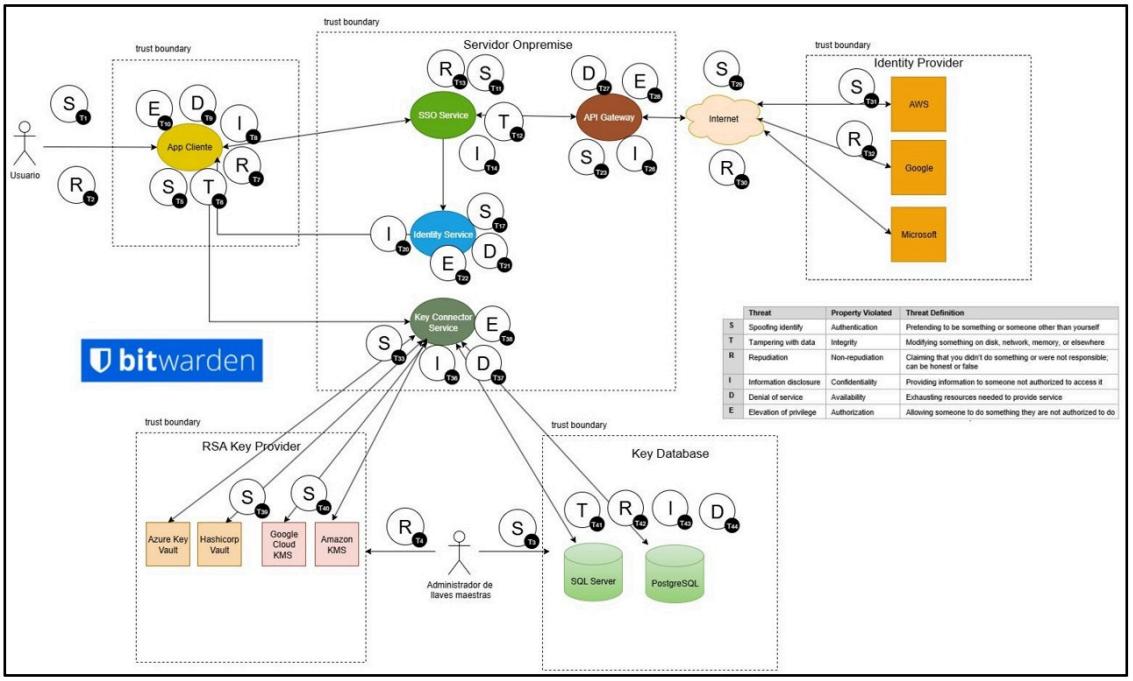


Figura 2: Modelado de amenazas STRIDE de software Bitwarden. Fuente: Elaboración Propia

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