



INGENIERÍA DE SISTEMAS

SISTEMA DE ADMINISTRACIÓN DE INVENTARIOS DE

INSUMOS PARA LABORATORIO DE ÓPTICA

CASO: OPTALVISION

Proyecto de Grado para optar al grado de licenciatura en

Ingeniería de sistemas

Autor: Daniel Santiago Soto Villamil

Tutor: Maritza Netzy Paiva Zapana

La Paz - Bolivia

2024

RESUMEN

TÍTULO: SISTEMA DE ADMINISTRACIÓN DE INVENTARIOS DE INSUMOS PARA
LABORATORIO DE ÓPTICA CASO: OPTALVISION

AUTOR: DANIEL SANTIAGO SOTO VILLAMIL

PROBLEMÁTICA

Dificultades en el control y la administración de inventarios de insumos de laboratorios de ópticas.

OBJETIVO GENERAL

Desarrollar un sistema de administración de inventarios de insumos para laboratorio de óptica para el manejo administrativo.

CONTENIDO

Desarrollo de un sistema de administración de inventarios para laboratorios de óptica, tomando controles tecnológicos de la norma UNE-ISO/IEC 27001:2023, implementado con tecnologías de desarrollo web y la centralización de datos en un gestor de base de datos relacional, adaptado a las necesidades específicas del laboratorio.

CARRERA	: Ingeniería de Sistemas
PROFESOR GUÍA	: Lic. Maritza Netzy Paiva Zapana
DESCRIPTORES O TEMAS	: Sistema web, Vue.js Nest.js, Node.js, PostgreSQL, TypeScript, JavaScript.
PERÍODO DE INVESTIGACIÓN	: 2024
EMAIL DE LOS AUTORES	: Santiago_SV@outlook.es

ÍNDICE

Página

CAPITULO I

INTRODUCCIÓN

1.1.	ANTECEDENTES	1
1.2.	PLANTEAMIENTO DEL PROBLEMA.....	4
1.2.1.	Identificación Del Problema	5
1.2.2.	Problema Central	6
1.2.3.	Formulación Del Problema	6
1.3.	OBJETIVOS DE LA INVESTIGACIÓN	7
1.3.1.	Objetivo General	7
1.3.2.	Objetivos Específicos.....	7
1.4.	DELIMITACIÓN	8
1.4.1.	Límite Temporal.....	8
1.4.2.	Límite Espacial.....	8
1.5.	JUSTIFICACIÓN	8
1.5.1.	Justificación Social.....	8

1.5.2.	Justificación Económica.....	9
1.5.3.	Justificación Tecnológica	10
1.6.	TIPOLOGÍA DEL PROYECTO.....	10
1.7.	MÉTODOS DE INVESTIGACIÓN.....	11
1.7.1.	Enfoque De La Investigación	11
1.7.2.	Métodos De Investigación.....	11
1.7.3.	Diseño De La Investigación	12
1.7.4.	Tipo De Investigación.....	13
1.8.	TÉCNICAS DE INVESTIGACIÓN Y SUS INSTRUMENTOS.....	14
1.8.1.	Las Entrevistas en Profundidad	14
1.9.	POBLACIÓN Y MUESTRA	14
1.9.1.	Población	14
1.9.2.	Muestra	15

CAPITULO II

MARCO TEÓRICO

2.1.	SISTEMA	16
2.2.	SISTEMA WEB	16

2.3.	HERRAMIENTAS DE DESARROLLO	17
2.3.1.	JavaScript	18
2.3.2.	TypeScript	20
2.3.3.	Node.JS	22
2.3.4.	Nest.JS.....	23
2.3.5.	Vue.JS.....	24
2.3.6.	TyperORM.....	25
2.3.7.	TailWind CSS	26
2.4.	BASE DE DATOS	27
2.4.1.	Sistema de Gestión de Bases de Datos	28
2.4.2.	PostgreSQL.....	30
2.5.	API Y API REST	31
2.5.1.	API.....	31
2.5.2.	REST.....	31
2.5.3.	API REST	32
2.6.	INVENTARIO	33
2.6.1.	Métodos de Control de Inventarios.....	34

2.7.	UNE-ISO/IEC 27001:2023	36
2.7.1.	Principios Fundamentales de la UNE-ISO/IEC 27001:2023	37
2.7.1.1.	Contexto de la organización	37
2.7.1.2.	Necesidades y expectativas de las partes interesadas	38
2.7.2.	Soporte en el Sistema de Gestión de la Seguridad de la Información (SGSI)...	39
2.7.2.1.	Recursos necesarios	39
2.7.2.2.	Competencia y capacitación.....	39
2.7.2.3.	Concienciación del personal.....	40
2.7.2.4.	Comunicación efectiva	40
2.7.2.5.	Gestión de la información documentada	41
2.7.3.	Contexto de la Organización.....	42
2.7.3.1.	Comprensión de la organización y de su contexto	42
2.7.3.2.	Comprensión de las necesidades de las partes interesadas.....	42
2.7.3.3.	Sistema de gestión de la seguridad de la información.....	43
2.7.4.	Compatibilidad con Otras Normas de Sistemas de Gestión	43
2.7.5.	Controles Tecnológicos	44
2.7.6.	Comparación de la versión 2023 contra la versión 2013	45

2.8.	MARCO DE TRABAJO SCRUM.....	46
2.8.1.	Roles	48
2.8.1.1.	Product owner.....	48
2.8.1.2.	Scrum master	49
2.8.1.3.	Desarrollo	49
2.8.2.	Sprint Backlog.....	50
2.8.3.	Objetivo del Sprint.....	50

CAPITULO III

MARCO PRACTICO

3.1.	IMPLEMENTACIÓN DE SCRUM PARA EL DESARROLLO DEL PROYECTO.	52
3.1.1.	Roles en el Marco Scrum.....	52
3.1.2.	Historias de Usuario.....	53
3.1.2.1.	Historia de administrador.....	53
3.1.2.2.	Historia de encargado proveedores-productos	54
3.1.2.3.	Historia de encargado trabajos.....	54
3.1.2.4.	Historia de encargado ventas.....	55
3.1.3.	Product Backlog	55

3.1.4.	Tabla de Requerimientos	58
3.2.	SPRINT 1: GESTIÓN DE AUTENTICACIÓN	60
3.2.1.	Objetivos del Sprint.....	60
3.2.2.	Definición de la Gestión de Autenticación	61
3.2.3.	Diagramas UML del Sprint.....	62
3.2.3.1.	Diagrama de caso de uso.....	62
3.2.3.2.	Diagrama de secuencia	62
3.2.3.3.	Diagrama actividades	63
3.2.4.	Codificación.....	63
3.2.5.	Tareas del Sprint Backlog	64
3.2.6.	Desarrollo Iterativo y Validación.....	65
3.2.7.	Evaluación de Resultados	66
3.2.7.1.	Resultados de roles	66
3.2.7.2.	Resultados de usuarios	67
3.3.	SPRINT 2: GESTIÓN PERSONAL	69
3.3.1.	Objetivos del Sprint.....	69
3.3.2.	Definición de La Gestión De Personal	70

3.3.3.	Diagramas UML del Sprint.....	72
3.3.3.1.	Diagrama de caso de uso.....	72
3.3.3.2.	Diagrama de secuencia.....	72
3.3.3.3.	Diagrama actividades	73
3.3.4.	Codificación.....	73
3.3.5.	Tareas del Sprint Backlog	74
3.3.6.	Desarrollo Iterativo y Validación.....	75
3.3.7.	Evaluación de Resultados	76
3.3.7.1.	Resultados de personal.....	77
3.4.	SPRINT 3: GESTIÓN DE PROVEEDORES Y PRODUCTOS	78
3.4.1.	Objetivos del Sprint.....	78
3.4.2.	Definición de la Gestión de Proveedores y Productos	79
3.4.3.	Diagramas UML del Sprint.....	81
3.4.3.1.	Diagrama de caso de uso.....	81
3.4.3.2.	Diagrama de secuencia.....	81
3.4.3.3.	Diagrama actividades	82
3.4.4.	Codificación.....	83

3.4.5.	Tareas del Sprint Backlog	85
3.4.6.	Desarrollo Iterativo y Validación.....	86
3.4.7.	Evaluación de Resultados	86
3.4.7.1.	Resultados de proveedores.....	87
3.4.7.2.	Resultados de productos.....	89
3.4.7.3.	Resultados de promedio ponderado	91
3.5.	SPRINT 4: SPRINT DE GESTIÓN DE TRABAJOS	92
3.5.1.	Objetivos del Sprint.....	92
3.5.2.	Definición de la Gestión de Trabajos	93
3.5.3.	Diagrama UML del Sprint.....	94
3.5.3.1.	Diagrama de caso de uso.....	94
3.5.3.2.	Diagrama de secuencia	95
3.5.3.3.	Diagrama actividades	95
3.5.4.	Codificación.....	96
3.5.5.	Tareas del Sprint Backlog	97
3.5.6.	Desarrollo Iterativo y Validación.....	98
3.5.7.	Evaluación de Resultados	99

3.5.7.1.	Resultados de trabajos	100
3.6.	SPRINT 5: GESTIÓN DE VENTAS	102
3.6.1.	Objetivos del Sprint.....	102
3.6.2.	Definición de la Gestión de Ventas.....	103
3.6.3.	Diagramas UML del Sprint.....	104
3.6.3.1.	Diagrama de caso de uso.....	104
3.6.3.2.	Diagrama de secuencia.....	104
3.6.3.3.	Diagrama actividades	105
3.6.4.	Codificación.....	105
3.6.5.	Tareas del Sprint Backlog	106
3.6.6.	Desarrollo Iterativo y Validación.....	107
3.6.7.	Evaluación de Resultados	107
3.6.7.1.	Resultados de ventas	108
3.7.	SPRINT 6: GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN.....	110
3.7.1.	Introducción.....	110
3.7.2.	Alcance del Sistema	110
3.7.3.	Contexto de la Organización.....	111

3.7.3.1.	Análisis de factores internos.....	111
3.7.3.2.	Identificación de partes interesadas y sus necesidades	112
3.7.4.	Liderazgo	112
3.7.4.1.	Roles, responsabilidades y autoridades.....	112
3.7.5.	Planificación	113
3.7.5.1.	Declaración de aplicabilidad (SoA)	113
3.7.6.	Evaluación del Controles	116

CAPITULO IV

ANÁLISIS DE FACTIBILIDAD

4.1.	FACTIBILIDAD TÉCNICA	125
4.1.1.	Hardware.....	125
4.1.2.	Software	126
4.2.	ANÁLISIS DE FACTIBILIDAD ECONÓMICA.....	127
4.2.1.	COCOMO II.....	127
4.2.1.1.	Componentes funcionales	127
4.2.1.2.	Ponderación de funciones	128
4.2.1.3.	Estimación inicial de puntos de función	129

4.2.2.	Resultados generales de la estimación	129
4.2.3.	Distribución por Fases (Adquisición)	130
4.2.4.	Distribución del Esfuerzo por Fases y Actividades (RUP/MBASE)	131
4.3.	ANÁLISIS COSTO-BENEFICIO	132
4.3.1.	Cálculo de ROI y Punto Equilibrio.	133

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1.	CONCLUSIONES	134
5.2.	RECOMENDACIONES	136

REFERENCIAS BIBLIOGRÁFICAS

APÉNDICE

ANEXOS

ÍNDICE DE TABLAS

	Página
Tabla 1 Tipos de Gestión de Inventarios.....	35
Tabla 2 Comparación entre versiones de la ISO 27001.	45
Tabla 3 Roles de Scrum	52
Tabla 4 Historia de Administrador.....	53
Tabla 5 Historia de Encargado Proveedores-Productos.....	54
Tabla 6 Historia de Encargado Trabajos.	54
Tabla 7 Historia de Encargado Ventas.	55
Tabla 8 Product Backlog.....	55
Tabla 9 Tabla de Requerimientos.....	58
Tabla 10 Requisitos De La Gestión De Autenticación.	61
Tabla 11 Tareas Del Sprint Backlog De Gestión De Autenticación.	64
Tabla 12 Desarrollo Iterativo y Validación de Gestión de Autenticación.....	65
Tabla 13 Resultados del Módulo de Autenticación.	66
Tabla 14 Resultados del Módulo de Usuarios.....	68
Tabla 15 Requisitos De La Gestión Del Personal	71

Tabla 16 Tareas Del Sprint Backlog De Gestión Del Personal.	75
Tabla 17 Desarrollo Iterativo y Validación de Gestión del Personal.	76
Tabla 18 Resultados del Módulo de Personal.....	77
Tabla 19 Requisitos De La Gestión De Proveedores y Productos.	80
Tabla 20 Tareas Del Sprint Backlog De Gestión de Proveedores y Productos.	85
Tabla 21 Desarrollo Iterativo y Validación De Gestión de Proveedores Y Productos.....	86
Tabla 22 Resultados del Módulo de Proveedores.	87
Tabla 23 Resultados del Módulo de Productos.....	89
Tabla 24 Resultados de Promedio Ponderado.	91
Tabla 25 Requisitos De La Gestión De Trabajos.	93
Tabla 26 Tareas Del Sprint Backlog De Gestión de Trabajos.	97
Tabla 27 Desarrollo Iterativo y Validación De Gestión de Trabajos.	99
Tabla 28 Resultados del Módulo de Trabajos.	100
Tabla 29 Requisitos De La Gestión De Ventas.	103
Tabla 30 Tareas Del Sprint Backlog De Gestión de Ventas.....	106
Tabla 31 Desarrollo Iterativo y Validación De Ventas.	107
Tabla 32 Resultados del Módulo de Ventas.....	108

Tabla 33 Declaración de Aplicabilidad (SoA)	114
Tabla 34 Seguridad en el ciclo de vida del software.	122
Tabla 35 Hardware Utilizado.	125
Tabla 36 Software Utilizado.....	126
Tabla 37 Componentes Funcionales.....	127
Tabla 38 Ponderación de Funciones.....	128
Tabla 39 Tabla Estimación Inicial de Puntos de Función.....	129
Tabla 40 Resultados de Estimación.....	130
Tabla 41 Distribución del Esfuerzo por Fases y Actividades.	131
Tabla 42 Costos y Beneficios.	132
Tabla 43 Cronograma de Gant.....	141

ÍNDICE DE FIGURAS

	Página
Figura 1 Diagrama Ishikawa	6
Figura 2 Diagrama de caso de uso Gestión de Autenticación.....	62
Figura 3 Diagrama de secuencia de Gestión de Autenticación.....	62
Figura 4 Diagrama actividades de Gestión de Autenticación	63
Figura 5 Código de Autenticacion.	63
Figura 6 Resultado de Gestión de Roles	67
Figura 7 Resultados de Gestión de Usuarios	69
Figura 8 Diagrama de caso de uso de Gestión de Personal.....	72
Figura 9 Diagrama de secuencia de Gestión de Personal.	72
Figura 10 Diagrama actividades Gestión de Personal.	73
Figura 11 Código de Personal.....	73
Figura 12 Resultados de Gestión de Personal Frontend.....	78
Figura 13 Resultados de Gestión Personal Backend.	78
Figura 14 Diagrama de caso de uso de Gestión Proveedores y Producto.	81
Figura 15 Diagrama de secuencia de Gestión de Proveedores y Productos.....	81

Figura 16 Diagrama actividades de Gestión Productos y Proveedores.....	82
Figura 17 Código de Proveedores y Productos.....	83
Figura 18 Resultados Gestión de Proveedores Backend.....	88
Figura 19 Resultados de Gestión de Proveedores FrontEnd.....	88
Figura 20 Resultados de Gestión de Productos Backend.....	90
Figura 21 Resultados de Gestión de Productos FrontEnd.....	90
Figura 22 Resultados de Gestión de Promedio Ponderado.....	92
Figura 23 Diagrama de caso de uso de Gestión de Trabajos.....	94
Figura 24 Diagrama de secuencia de Gestión de Trabajos.....	95
Figura 25 Diagrama actividades de Gestión de Trabajos.....	95
Figura 26 Código de Trabajos.....	96
Figura 27 Resultados de Gestión de Trabajos Frontend.....	101
Figura 28 Resultados de Gestión de Trabajos Backend.....	101
Figura 29 Diagrama de caso de uso de Gestión de Ventas.....	104
Figura 30 Diagrama Secuencia de Gestión de Ventas.....	104
Figura 31 Diagrama actividades de Gestión de Ventas.....	105
Figura 32 Código de Ventas.....	105

Figura 33 Resultado de Gestión de Ventas Trabajos Frontend.....	109
Figura 34 Resultado Gestión de Ventas Backend.	109
Figura 35 Dispositivos finales del usuario.....	116
Figura 36 Gestión de Privilegios de Acceso.	117
Figura 37 Restricción de Acceso a la Información.....	117
Figura 38 Autenticación Segura.	118
Figura 39 Gestión de Capacidades.....	119
Figura 40 Prevención de Fugas de Datos.	119
Figura 41 Copias de seguridad de la Información.	120
Figura 42 Seguridad de Redes.	121
Figura 43 Seguridad de los Servicios de Red.....	121
Figura 44 Uso de la Criptografía.	122
Figura 45 Codificación Segura.....	123
Figura 46 Datos de Prueba.	124

ÍNDICE DE ANEXOS

Anexo 1 Alcance ISO 27001

Anexo 2 Contexto de la organización ISO 27001

Anexo 3 Liderazgo ISO 27001

Anexo 4 Planificación ISO 27001

Anexo 5 Evaluación de desempeño ISO 27001

Anexo 6 Entrevista al dueño de OptalVision

Anexo 7 Boleta de trabajo

Anexo 8 Carta de conformidad

CAPITULO I

INTRODUCCIÓN

1.1. ANTECEDENTES

El desarrollo de sistemas de gestión de inventarios ha sido un tema recurrente en el ámbito de la ingeniería de sistemas, dado su impacto directo en la eficiencia técnica y en la optimización de recursos dentro de las organizaciones. Diversos proyectos han abordado esta problemática desde distintas perspectivas, adaptándose a las necesidades específicas de cada sector. La implementación de un sistema centralizado de administración de inventarios, en particular para el sector óptico, representa un desafío técnico y organizacional, que no sólo busca mejorar la eficiencia en la administración de stock.

Un primer antecedente relevante es el "Sistema Integrado de Control de Inventario 'ATIPAJ' Compañía Cervecera Boliviana S.A.", desarrollado por Verónica Coarite Tumiri. Este proyecto se centra en la implementación de un sistema de control de inventarios que busca optimizar la gestión de insumos y productos terminados en la empresa cervecera. Se destaca por su enfoque en la integración de diferentes procesos dentro de la empresa, permitiendo una gestión más eficiente y precisa del inventario. La metodología utilizada, basada en la optimización de flujos de trabajo y en la automatización de procesos, proporciona una base sólida para el desarrollo de sistemas similares en otros contextos, como el de las ópticas, donde la precisión en la gestión de inventarios es crucial.

Otro proyecto relevante es el "Sistema para la Gestión de Ventas e Inventario Caso: Importadora Soluciones Médicas Lifemed S.R.L." de Johovana La Fuente Choque. Este sistema fue diseñado para mejorar la gestión de inventarios y ventas en una importadora de soluciones médicas, enfocándose en la trazabilidad y control de productos sensibles. La experiencia obtenida en la gestión de productos de alta rotación y la necesidad de mantener un control estricto de los inventarios puede ser directamente aplicable a la gestión de inventarios en ópticas, donde los productos manejados, como lentes y equipos oftálmicos, también requieren un manejo cuidadoso para evitar pérdidas y optimizar la disponibilidad.

El "Sistema de Control de Inventarios para Laboratorios Crespal S.A. Regional Sucre" desarrollado por Juan Lucio Ramos Paye es otro antecedente que aporta valor a este análisis. Este proyecto aborda la necesidad de un control riguroso de inventarios en un entorno de laboratorio, donde la precisión y la confiabilidad de los datos son fundamentales. La implementación de un sistema que permite un seguimiento detallado de las entradas y salidas de materiales proporciona un marco útil para la gestión de inventarios en ópticas, donde se manejan productos delicados y costosos que deben estar disponibles en el momento justo para satisfacer las necesidades de los clientes.

Por su parte, el proyecto "Sistema de Información de Compras e Inventarios SAMA" de Raúl Francisco Choque Chambilla se centra en la gestión de compras e inventarios en una empresa manufacturera. La implementación de un sistema que no sólo gestiona el inventario, sino que también se integra con los procesos de compras permite una gestión más eficiente y coordinada de los recursos. En el contexto de una óptica, donde la

coordinación entre la adquisición de productos y su disponibilidad en inventario es crucial, las lecciones aprendidas de este proyecto son particularmente relevantes.

El "Sistema de Control y Seguimiento de Almacenes para la Corte Departamental Electoral La Paz, Sala Provincias" desarrollado por Virginia Suarez Marin, aborda un contexto completamente diferente, pero con desafíos similares en términos de gestión y seguridad de la información. En este caso, el sistema implementado debía favorecer la integridad y disponibilidad de los materiales almacenados. La implementación de controles sistema puede ser adaptada para asegurar que los inventarios en una óptica estén no sólo bien gestionados, sino también protegidos contra accesos no autorizados y manipulaciones indebidas, inspirándose en los estándares ISO 27001.

El proyecto "Sistema de Entradas y Salidas e Inventario Caso: BOLITAL S.R.L." de Claudia Chiri Honorio, aporta otro ejemplo de cómo la gestión de inventarios puede ser optimizada a través de un sistema automatizado que permita un seguimiento preciso de todos los movimientos de stock. La automatización de estos procesos no sólo mejora la eficiencia técnica, sino que también reduce el riesgo de errores humanos, un aspecto crítico cuando se manejan productos tan específicos como los que se encuentran en una óptica.

Finalmente, el "Software de Gestión y Control de Inventarios Caso: AGADON S.R.L." de Wilmer David Callisaya Apaza, destaca por su enfoque en la implementación de un sistema de gestión de inventarios con una alta dependencia en la tecnología y metodologías ágiles. Este proyecto es especialmente relevante porque integra prácticas

de seguridad en la administración de inventarios, utilizando metodologías como Scrum y estándares de calidad como ISO 9126 para asegurar un producto final robusto y seguro.

La revisión de estos proyectos muestra la importancia de un enfoque integral en la administración de inventarios, que combine la eficiencia técnica con la seguridad de la información. La implementación de un sistema centralizado de administración de inventarios para ópticas, basado en estándares de seguridad internacionales, no sólo mejorará la administración y el control de los productos, sino que también favorecerá la protección de la información, un aspecto cada vez más crítico en el entorno empresarial actual.

1.2. PLANTEAMIENTO DEL PROBLEMA

En el contexto actual de las ópticas, la administración de inventarios es un proceso fundamental que, si no se gestiona de manera adecuada, puede generar pérdidas económicas, desabastecimiento de productos y una limitada capacidad para satisfacer la demanda de los clientes. Muchas ópticas aún operan con sistemas manuales para la gestión de sus inventarios, lo que dificulta el control eficiente del stock, la planificación de reposiciones y el seguimiento de productos en almacén. Esto puede derivar en costos operativos elevados y una disminución en la competitividad de estas empresas.

La implementación de un Sistema de Administración de Inventarios de Insumos para Laboratorio de Óptica tiene como objetivo optimizar la gestión del stock, favoreciendo la disponibilidad constante de insumos y mejorando los procesos operativos relacionados con el inventario. El problema principal radica en el uso de métodos manuales, que

resultan ineficientes para administrar el inventario de manera efectiva, afectando la capacidad de las ópticas para responder oportunamente a las necesidades del mercado.

Evidencia de esta problemática se encuentra en estudios previos que muestran cómo las empresas que no cuentan con herramientas de administración automatizadas enfrentan pérdidas significativas en su stock, dificultades en el seguimiento de productos y una disminución en su capacidad para prever necesidades futuras (López, 2021). Esto repercute directamente en su capacidad operativa, incrementando costos innecesarios y afectando su desempeño en un mercado altamente competitivo.

Se espera que el desarrollo de este sistema no solo optimice los procesos operativos relacionados con el manejo de inventarios, sino que también permita a las ópticas mantener un control preciso del stock, mejorar su capacidad de respuesta y reducir costos derivados de ineficiencias en la gestión de insumos.

1.2.1. Identificación Del Problema

El diagrama de Ishikawa se justifica como una herramienta esencial para desglosar y analizar de manera estructurada las causas que contribuyen a la administración ineficaz de inventarios en ópticas. Al identificar las principales áreas problemáticas, como tecnología, procesos, personal, seguridad, recursos y comunicación, el diagrama facilita una comprensión clara de los factores subyacentes que afectan la eficiencia y seguridad del sistema de inventarios. Esto permite orientar mejor las acciones correctivas y diseñar soluciones que aborden las causas raíz, asegurando una implementación más efectiva de un sistema centralizado y seguro (Miro, 2023, párr. 2).

Figura 1*Diagrama Ishikawa*

Nota. Elaboración propia.

1.2.2. Problema Central

El laboratorio de OptalVision enfrenta ineficiencias en la administración de inventarios y el manejo de datos, causadas por el uso de procesos manuales propensos a errores y la falta de automatización en el registro y actualización de información. Estas problemáticas generan discrepancias entre los inventarios físicos, inconsistencias en los registros lo que dificulta la continuidad operativa y la toma de decisiones basada en datos precisos. Además, la falta de concordancia en la información afecta la coordinación interna, limitando la capacidad de responder eficientemente a las demandas operativas.

1.2.3. Formulación Del Problema

¿Cómo puede un sistema de administración de inventarios ayudar en el manejo de insumos y reducir las pérdidas en el laboratorio de óptica?

1.3. OBJETIVOS DE LA INVESTIGACIÓN

1.3.1. Objetivo General

Desarrollar un sistema de administración de inventarios de insumos para el laboratorio “OptalVision” para reducir perdidas y hacer el seguimiento de sus productos.

1.3.2. Objetivos Específicos

- Analizar la trazabilidad en la administración de insumos para laboratorios de óptica, considerando los procesos y flujos de información involucrados.
- Diseñar la base de datos relacional alineadas a las características operativas de los laboratorios de óptica, respetando los principios de organización y seguridad de los datos.
- Desarrollar los módulos de usuario, personal, productos, proveedores y ventas en el sistema de administración de inventarios para gestionar los movimientos de los insumos.
- Seleccionar controles tecnológicos de la norma UNE-ISO/IEC 27001:2023 que más se adecuen al desarrollo del sistema de administración de inventarios.
- Generar reportes automatizados a partir de los datos de ventas, trabajos e insumos del laboratorio, para facilitar el análisis de la información y la toma de decisiones.

1.4. DELIMITACIÓN

1.4.1. Límite Temporal

La investigación y desarrollo del Sistema de Administración de Inventarios para Ópticas se llevará a cabo durante el período comprendido entre octubre de 2024 y enero de 2025. Este intervalo de tiempo está diseñado para permitir la implementación del sistema en un entorno controlado, garantizando la recopilación de datos relacionados con la optimización de la gestión de inventarios y la mejora en la seguridad de la información. Este marco temporal asegura que los resultados obtenidos sean consistentes con los objetivos del proyecto y que se logren dentro del plazo establecido.

1.4.2. Límite Espacial

La investigación se llevará a cabo en el laboratorio óptico "Visión", ubicado en la ciudad de La Paz, Bolivia. Este entorno ha sido seleccionado como el espacio focal de estudio para implementar y evaluar el Sistema de Administración de Inventarios de insumos ópticos. La delimitación espacial permite analizar la aplicación del sistema en un contexto real y específico, asegurando que los resultados obtenidos sean relevantes y aplicables al sector óptico de la región.

1.5. JUSTIFICACIÓN

1.5.1. Justificación Social

La implementación de un Sistema de Administración de Inventarios de insumos para laboratorios de Ópticas tiene un impacto significativo en el ámbito social. Este sistema

mejora la estabilidad laboral del personal al optimizar la planificación y el control del stock, reduciendo riesgos de desabastecimiento o sobreabundancia. Como resultado, se genera un ambiente de trabajo más organizado y eficiente, lo que disminuye la presión sobre los empleados y fomenta un clima laboral favorable. Este beneficio se traduce en un mejor servicio al cliente, fortaleciendo la relación entre las ópticas y la comunidad que atienden.

Asimismo, este sistema puede influir positivamente en otros laboratorios de ópticas, promoviendo prácticas más eficientes en la administración de inventarios. La optimización de estos procesos fortalece la competitividad del sector óptico, beneficiando tanto a los trabajadores como a los clientes a través de un servicio de mayor calidad y una operación más efectiva.

1.5.2. Justificación Económica

Desde una perspectiva económica, la implementación de un Sistema de Administración de Inventarios con controles tecnológicos de la ISO 27001 en las ópticas de La Paz tiene el potencial de generar importantes beneficios financieros. Al centralizar y optimizar la administración de inventarios, se pueden reducir los costos asociados con el almacenamiento ineficiente, las pérdidas por productos faltantes o deteriorados, y los errores en el control del stock. Esto permitirá a las ópticas minimizar el capital inmovilizado en productos que no rotan rápidamente, liberando recursos que podrán destinarse a inversiones más estratégicas, como la adquisición de nueva tecnología o mejoras en el servicio al cliente.

1.5.3. Justificación Tecnológica

El proyecto está respaldado por una arquitectura tecnológica moderna diseñada para garantizar escalabilidad, seguridad y eficiencia, elementos esenciales para cumplir con los objetivos propuestos. Se emplea NestJS con Node.js en el backend, seleccionados por su enfoque modular y su capacidad para construir aplicaciones robustas y escalables. Para el frontend, se utiliza Vue.js, una herramienta versátil que permite crear interfaces dinámicas y amigables para el usuario. La base de datos está gestionada por PostgreSQL, reconocida por su confiabilidad y rendimiento en entornos de alta complejidad. En términos de seguridad, se implementaron autenticación con JWT, encriptación de contraseñas con bcrypt, y configuración de CORS para un control adecuado de accesos. Además, herramientas como VSCode, Postman y PlantUML fueron clave para optimizar el desarrollo, pruebas y documentación. Esta selección tecnológica asegura que el sistema no solo cumple con los requerimientos actuales, sino que está preparado para escalar y adaptarse a futuras necesidades.

1.6. TIPOLOGÍA DEL PROYECTO

Este proyecto se enmarca en la categoría de proyecto tecnológico, ya que el resultado de la investigación es un Sistema de Administración de Inventarios diseñado específicamente para optimizar la administración de productos en las ópticas, integrando medidas de seguridad basadas en los estándares internacionales ISO 27001. El principal objetivo de este sistema es mejorar y automatizar los procesos técnicos de inventario, proporcionando un producto tecnológico que facilitará la vida de los empleados y administradores en las ópticas.

1.7. MÉTODOS DE INVESTIGACIÓN

1.7.1. Enfoque De La Investigación

El presente proyecto de investigación utilizará un enfoque cualitativo.

Las investigaciones cualitativas suelen producir preguntas antes, durante o después de la recolección y análisis de los datos. La acción indagatoria se mueve de manera dinámica entre los hechos y su interpretación, y resulta un proceso más bien “circular” en el que la secuencia no siempre es la misma, puede variar en cada estudio. (Sampieri, 2018, p. 9)

1.7.2. Métodos De Investigación

El método de investigación adoptado será inductivo, ya que:

Van de lo particular a lo general. Por ejemplo, en un estudio cualitativo típico, el investigador entrevista a una persona, analiza los datos que obtuvo y saca conclusiones; posteriormente, entrevista a otra persona, analiza esta nueva información y revisa sus resultados y conclusiones; del mismo modo, efectúa y analiza más entrevistas para comprender el fenómeno que estudia. Es decir, procede caso por caso, dato por dato, hasta llegar a una perspectiva más general. (Sampieri, 2018 p. 8)

La investigación cualitativa se basará en la observación y el análisis detallado de las experiencias de los empleados y administradores en las ópticas que implementen el Sistema de Administración de Inventarios con medidas de seguridad inspiradas en la

norma ISO 27001. A través de entrevistas estructuradas y observaciones cuidadosas, se recopilarán datos empíricos que contribuirán a construir una comprensión teórica general sobre cómo este sistema impacta en la eficiencia operativa y en la seguridad de la información. Este enfoque permitirá generar conclusiones fundamentadas en las vivencias concretas dentro del entorno específico de las ópticas, proporcionando una visión más completa y contextualizada del impacto del sistema.

1.7.3. Diseño De La Investigación

El diseño de esta investigación será de tipo experimental, dado que cumple con los requisitos de control y validez interna mediante el uso de grupos de comparación y la manipulación de la variable independiente, en este caso, la implementación del sistema de administración de inventarios con medidas de seguridad inspiradas basadas en la norma ISO 27001. Este diseño permite analizar los efectos del sistema en variables dependientes, como la eficiencia operativa y la seguridad de la información.

Sampieri (2018) explica lo siguiente:

Los experimentos son aquellos que reúnen los dos requisitos para lograr el control y la validez interna: 1) grupos de comparación (manipulación de la variable independiente o de varias independientes) y 2) equivalencia de los grupos. Estos diseños llegan a incluir una o más variables independientes y una o más dependientes. Asimismo, pueden utilizar prepruebas y pospruebas para analizar la evolución de los grupos antes y después del tratamiento experimental. Desde luego, no todos los diseños experimentales utilizan preprueba; aunque la

posprueba sí es necesaria para determinar los efectos de las condiciones experimentales (p. 188).

Este diseño permitirá establecer grupos equivalentes para comparar las condiciones operativas y de seguridad antes y después de la implementación del sistema.

1.7.4. Tipo De Investigación

Este estudio se clasifica como aplicado y teórico, ya que aborda tanto la solución de un problema práctico relacionado con la administración de inventarios en las ópticas como la generación de aportes al desarrollo teórico en el ámbito de la seguridad de la información.

Desde el enfoque aplicado, la investigación tiene como objetivo principal implementar un Sistema de Administración de Inventarios con medidas de seguridad inspiradas en la norma ISO 27001. Esta implementación busca resolver problemas específicos relacionados con la eficiencia técnica en la gestión de insumos y la protección de datos en el contexto particular de las ópticas.

Simultáneamente, desde el enfoque teórico, se pretende contribuir al conocimiento sobre cómo los estándares de seguridad de la información pueden ser adaptados y aplicados en pequeñas y medianas empresas del sector óptico. Este análisis busca derivar principios generales que puedan servir como referencia para otros contextos similares, ampliando así el entendimiento sobre la aplicación práctica de las normas de seguridad en entornos empresariales específicos (Hernández, 2010, p. 57).

1.8. TÉCNICAS DE INVESTIGACIÓN Y SUS INSTRUMENTOS

Las técnicas de investigación seleccionadas para este estudio serán:

1.8.1. Las Entrevistas en Profundidad

Debido a que “La entrevista cualitativa es más íntima, flexible y abierta. Esta se define como una reunión para intercambiar información entre una persona (el entrevistador) y otra (el entrevistado) u otras (entrevistados)” (Sampieri, 2018, p. 597).

Se llevarán a cabo con empleados clave de las ópticas, permitiendo explorar de manera detallada sus experiencias y percepciones sobre el sistema de administración de inventarios y su impacto en la operatividad y seguridad. Simultáneamente, el investigador realizará observación participante dentro de las ópticas, involucrándose directamente en el entorno para observar de primera mano cómo se manejan los inventarios y cómo interactúan los empleados con el sistema implementado. Los datos recolectados a través de ambas técnicas se analizarán para identificar patrones de comportamiento y temas comunes, proporcionando una visión más profunda del impacto del sistema en el entorno laboral. A partir de estos hallazgos, se evaluarán las mejoras operativas y los desafíos que enfrenta el sistema, con el fin de generar conclusiones que ayuden a optimizar su implementación en el futuro.

1.9. POBLACIÓN Y MUESTRA

1.9.1. Población

La población de esta investigación estará compuesta por el laboratorio óptico de la "OptalVision " de la ciudad de La Paz, Bolivia. Este laboratorio interactúa con sistemas

de administración de inventarios y se centra en la gestión y control de insumos. La población incluye al dueño del laboratorio óptico, quien tiene conocimiento directo sobre la administración de inventarios en laboratorios de ópticas.

1.9.2. Muestra

La muestra de esta investigación será el laboratorio óptico "OptalVision". En este caso, se entrevistará exclusivamente al dueño del laboratorio óptico, quien proporciona una perspectiva clave sobre la implementación y operación del sistema.

Por lo tanto, será una muestra no probabilística como afirma Sampieri (2018):

Para esta investigación se utiliza las muestras por conveniencia Sampieri (2018) afirman que “estas muestras están formadas por los casos disponibles a los cuales tenemos acceso. Tal fue la situación de Rizzo (2004), quien no pudo ingresar a varias empresas para efectuar entrevistas a profundidad en niveles gerenciales acerca de los factores que conforman el clima organizacional, y entonces decidió entrevistar a compañeros que junto con ella cursaban un posgrado en desarrollo humano y eran directivos de diferentes organizaciones (p. 433)

Por lo tanto, la selección de la muestra será de muestreo por criterio, permitiendo incluir al dueño del laboratorio óptico, quien interactúa directamente con el Sistema de Administración de Inventarios y cuenta con experiencia en la gestión de inventarios.

CAPITULO II

MARCO TEÓRICO

2.1. SISTEMA

Según Sommerville (2011), "un sistema puede entenderse como un conjunto de componentes interrelacionados que trabajan juntos para realizar una función específica o alcanzar un objetivo común" (p. 15).

Este concepto implica que cada elemento dentro del sistema cumple un rol particular, contribuyendo al funcionamiento integral y coherente del conjunto. Además, Sommerville enfatiza que la efectividad de un sistema no radica solo en la capacidad de sus partes individuales, sino en cómo estas están organizadas y en la forma en que interactúan entre sí para lograr el propósito general para el cual fueron diseñadas (Sommerville, 2011, p. 57). Así, un sistema bien estructurado no solo optimiza los recursos disponibles, sino que también mejora la eficiencia y la capacidad de respuesta de una organización frente a las demandas y cambios en su entorno operativo.

2.2. SISTEMA WEB

Según TechTarget (2023) un sistema o aplicación web (o web app) es un programa de aplicación que se almacena en un servidor remoto y se entrega a través de Internet mediante una interfaz de navegador. Por definición, los servicios web también son aplicaciones web, y muchos sitios web, aunque no todos, contienen aplicaciones web.

Los desarrolladores diseñan aplicaciones web para una amplia variedad de usos y usuarios, desde organizaciones hasta individuos, por diversas razones. Las aplicaciones web comúnmente utilizadas pueden incluir correo web, calculadoras en línea o tiendas de comercio electrónico. Aunque algunos usuarios solo pueden acceder a ciertas aplicaciones web mediante un navegador específico, la mayoría están disponibles sin importar el navegador.

Para que una aplicación web funcione, necesita un servidor web, un servidor de aplicaciones y una base de datos. Los servidores web gestionan las solicitudes que provienen de un cliente, mientras que el servidor de aplicaciones completa la tarea solicitada. Una base de datos almacena cualquier información necesaria (TechTarget, 2023). Las aplicaciones web suelen tener ciclos de desarrollo cortos y equipos de desarrollo reducidos. La mayoría de los desarrolladores escriben aplicaciones web en JavaScript, HTML5 o CSS. La programación del lado del cliente generalmente utiliza estos lenguajes, que ayudan a construir la parte frontal de la aplicación. La programación del lado del servidor crea los scripts que utilizará la aplicación web. Lenguajes como Python, Java y Ruby se utilizan comúnmente en la programación del lado del servidor.

2.3. HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo por definición:

Las herramientas de desarrollo del software (llamadas en ocasiones herramientas de Ingeniería de Software Asistido por Computadora o CASE, por las siglas de Computer-Aided Software Engineering) son programas usados para apoyar las

actividades del proceso de la ingeniería de software. En consecuencia, estas herramientas incluyen editores de diseño, diccionarios de datos, compiladores, depuradores (debuggers), herramientas de construcción de sistema, etcétera. (Sommerville, 2011, p. 37)

2.3.1. JavaScript

JavaScript fue introducido en 1995 como un lenguaje diseñado para agregar interactividad a las páginas web en el navegador Netscape Navigator. Desde entonces, ha sido adoptado por todos los navegadores principales, permitiendo el desarrollo de aplicaciones web modernas que facilitan la interacción directa del usuario sin necesidad de recargar la página constantemente. Como señala Haverbeke (2018), "JavaScript hizo posible una nueva era de aplicaciones web dinámicas, transformando la manera en que los usuarios interactúan con las páginas" (p. 6).

Es importante destacar que, a pesar de compartir parte del nombre, JavaScript y Java tienen muy poco en común. Según Haverbeke (2018), "el nombre fue una estrategia de marketing para aprovechar la popularidad que Java tenía en ese momento, lo que dejó a JavaScript con un nombre que no refleja su verdadera naturaleza" (p. 6). A medida que fue adoptado fuera de Netscape, surgió la necesidad de estandarizarlo, dando lugar al Estándar ECMAScript, desarrollado por Ecma International, que unificó las implementaciones del lenguaje. Aunque JavaScript y ECMAScript suelen utilizarse indistintamente, ambos términos representan el mismo lenguaje bajo diferentes contextos.

El diseño inicial de JavaScript era extremadamente permisivo, lo que lo hacía accesible para principiantes, pero también dificultaba la detección de errores.

Haverbeke (2018) comentó lo siguiente:

El lenguaje aceptaba casi cualquier cosa que escribiera, pero la interpretaba de una manera que era completamente diferente de lo que quería decir. Por supuesto, esto tenía mucho que ver con el hecho de que no tenía idea de lo que estaba haciendo, pero hay un problema real aquí: JavaScript es ridículamente liberal en lo que permite. (p. 7).

Sin embargo, esta flexibilidad también permitió la implementación de técnicas avanzadas que serían imposibles en lenguajes más rígidos. Con el tiempo, los desarrolladores aprendieron a apreciar estas características.

JavaScript ha evolucionado significativamente desde su creación. Entre 2000 y 2010, ECMAScript versión 3 fue ampliamente compatible, estableciendo las bases del dominio de JavaScript en la web. Según Haverbeke (2018), "la ambiciosa versión 4, que planeaba mejoras radicales, fue abandonada en 2008 debido a su complejidad, lo que dio lugar a una versión 5 más práctica y accesible en 2009" (p. 7). La versión ECMAScript 6, lanzada en 2015, incluyó varias de las innovaciones planificadas para la versión 4 y marcó el inicio de actualizaciones anuales para el lenguaje.

La evolución constante del lenguaje requiere que los navegadores y otros entornos se actualicen regularmente para soportar las nuevas características. Como menciona

Haverbeke (2018), "los diseñadores de lenguajes tienen cuidado de no realizar cambios que puedan romper programas existentes, asegurando la compatibilidad hacia atrás en nuevos navegadores" (p. 8). Esta compatibilidad garantiza que las aplicaciones más antiguas puedan seguir funcionando, incluso con las versiones más recientes del lenguaje.

JavaScript también se ha expandido más allá de los navegadores web. Algunas bases de datos, como MongoDB y CouchDB, utilizan JavaScript como lenguaje de scripting y consulta, mientras que plataformas como Node.js proporcionan un entorno para programar en JavaScript fuera del navegador. Este uso en diversas plataformas lo ha convertido en un lenguaje versátil y esencial para el desarrollo moderno, tanto en el frontend como en el backend (Haverbeke, 2018, p. 7).

2.3.2. TypeScript

TypeScript se ha convertido en uno de los lenguajes de programación con mayor auge en los últimos años. En el informe anual de GitHub, TypeScript ocupa la cuarta posición entre los lenguajes más utilizados, después de JavaScript, Python y Java, lo que resalta su relevancia y adopción en el desarrollo moderno. Según Stack Overflow, TypeScript es también el segundo lenguaje más apreciado por los desarrolladores, después de Rust, una posición que evidencia su gran aceptación en la comunidad de programación (Talaminos, 2022, pp. 9-10)

Este lenguaje ha logrado posicionarse de manera destacada dentro del ecosistema de JavaScript. Como señala el informe "State of JavaScript" (2021), en una encuesta que

involucró a más de 23,000 programadores de 137 países, TypeScript fue galardonado como la tecnología más adoptada dentro de la comunidad de JavaScript, lo que refleja su influencia y uso cada vez mayor. Según State of JavaScript (2021), "TypeScript recibió el premio a la tecnología más adoptada" (pp. 9-10), lo que confirma la popularidad del lenguaje a nivel global.

TypeScript no solo ha sido adoptado por proyectos de software importantes, sino también por empresas de renombre mundial. Herramientas y plataformas como Angular, Vue, Jest, Ionic y Visual Studio Code han incluido TypeScript en su desarrollo, y compañías como Google, Airbnb, PayPal y Slack lo han implementado en sus sistemas para aprovechar sus capacidades en aplicaciones empresariales de gran escala. Este creciente uso es prueba de la versatilidad y robustez que TypeScript ofrece en la programación, especialmente cuando se buscan aplicaciones escalables y seguras (Talaminos, 2022, p. 8)

A su vez, la comunidad de TypeScript experimenta un crecimiento constante, proporcionando una base sólida para los desarrolladores. Cada vez hay una mayor cantidad de documentación en línea y repositorios de GitHub con recursos y bibliotecas específicas, que facilitan la integración del lenguaje en diversos proyectos. Esta disponibilidad de recursos es clave para el aprendizaje y adopción de TypeScript en entornos profesionales y educativos. Según Talaminos, (2022), "la comunidad de TypeScript crece constantemente y cada vez existe mayor cantidad de documentación en la red e incluso repositorios de GitHub muy completos con multitud de recursos relacionados con el lenguaje" (p. 10).

Frente al tema, "TypeScript extiende las funcionalidades de JavaScript, proporcionando características avanzadas como genéricos y decoradores" (Talaminos, 2022, p. 11). Este diseño le permite a TypeScript aumentar la seguridad y el control en el desarrollo de aplicaciones. Además, al ser un superconjunto de JavaScript, TypeScript permite a los desarrolladores beneficiarse de todas las características de JavaScript, pero con ventajas adicionales en la depuración y mantenimiento del código. Como sugieren estos aspectos, el dominio de TypeScript resulta valioso para aquellos que buscan mejorar la calidad y eficiencia de sus proyectos web (Talaminos, 2022, p. 11).

2.3.3. Node.JS

Node.js representa una evolución significativa en el uso de JavaScript al permitir que funcione en el lado del servidor. Su arquitectura asincrónica ofrece una ventaja importante al ejecutar múltiples solicitudes sin bloquearse, mejorando la eficiencia y velocidad en comparación con tecnologías de servidor tradicionales. Según López (2021) "lo que se pretende con llevar JavaScript y su asincronía al lado del servidor es tener una asincronía real en el lado del servidor" (p. 15). Esta asincronía permite que una máquina servidora gestione múltiples solicitudes de forma simultánea, esperando las respuestas de otros servidores sin que esto afecte el rendimiento general del sistema.

Node.js, además, emplea el motor de JavaScript para ejecutar estas tareas, lo que lo convierte en una tecnología rápida y eficiente. A diferencia de lenguajes como PHP o Java, que requieren de un servidor web (como Apache o Tomcat) para gestionar peticiones, Node.js opera de manera autónoma, evitando así la necesidad de

configuraciones adicionales. Como explican López (2021), "lo mejor de todo, [Node.js] no necesita de servidor Web, ni Apache, ni Tomcat, ni IIS, ni NGinx ni ningún otro" (p. 15). Esta característica es especialmente beneficiosa en aplicaciones de tiempo real, como chats y juegos en línea, donde el tiempo de respuesta es crucial para la experiencia del usuario. Por último, Node.js destaca en la gestión de operaciones concurrentes gracias a su diseño no bloqueante, lo que permite una mejor respuesta en aplicaciones con altos volúmenes de tráfico, adaptándose así a las necesidades de las aplicaciones web modernas (López, 2021, p. 18).

2.3.4. Nest.JS

Nest.js es uno de los frameworks de Node.js de más rápido crecimiento para construir aplicaciones backend eficientes, escalables y de nivel empresarial. Este framework, que se basa en el moderno JavaScript y TypeScript, permite desarrollar aplicaciones altamente comprobables y mantenibles, lo que lo convierte en una opción popular entre los desarrolladores que buscan estructura y organización en sus proyectos. Con más de 46,6k estrellas y 5,4k bifurcaciones en GitHub, y un promedio de 700,000 descargas semanales, Nest.js se destaca como un recurso confiable para la construcción de backend con Node.js. Según su documentación: "El framework se ha diseñado para aprovechar las capacidades de TypeScript, el cual agrega tipado estático y mejora la calidad del código, además de ser compatible con otros patrones de diseño arquitectónicos como MVC (Modelo-Vista-Controlador)" (kinsta, 2022, párr. 1), lo cual facilita la estructuración de aplicaciones complejas.

Nest.js es especialmente útil en proyectos que requieren alta escalabilidad y sostenibilidad a largo plazo, y es frecuentemente adoptado por empresas que buscan una plataforma robusta para sus servicios en producción. Entre sus características distintivas, Nest.js destaca por su enfoque modular, lo que permite dividir el proyecto en módulos individuales, facilitando su mantenimiento y prueba en equipos de trabajo grandes. Esta estructura modular es una de las razones por las que se ha convertido en la elección preferida para el desarrollo de APIs, microservicios y aplicaciones de gran escala en la industria. Empresas de renombre como Adidas, Decathlon y Capgemini utilizan Nest.js para sus aplicaciones backend, lo que resalta su capacidad para manejar exigencias empresariales (kinsta, 2022, párr. 3).

2.3.5. Vue.JS

Vue.js, conocido comúnmente como Vue (pronunciado /vju:/, como "view"), es un framework progresivo diseñado para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue ha sido creado para ser utilizado de manera incremental, lo que lo convierte en una herramienta altamente adaptable para diversos proyectos. Su librería central se enfoca únicamente en la capa de visualización, lo que facilita su integración con otras librerías o sistemas ya existentes, proporcionando flexibilidad tanto para proyectos pequeños como para aplicaciones más complejas.

Entre sus características más destacadas, Vue permite el desarrollo de aplicaciones web de una sola página (SPA, por sus siglas en inglés) cuando se combina con herramientas modernas y librerías de apoyo. Esta capacidad lo posiciona como una opción ideal para

proyectos que buscan sofisticación sin perder la simplicidad en su implementación inicial. Según el equipo de desarrollo de Vue, "la simplicidad progresiva es una de las claves que hacen de Vue una herramienta única en el ecosistema del desarrollo frontend" (Vue Team, 2024, párr. 1).

Además, Vue ha ganado popularidad gracias a su enfoque intuitivo y a su curva de aprendizaje asequible, incluso para desarrolladores que recién comienzan en el área del desarrollo frontend. Este framework también se distingue por su comunidad activa y su compatibilidad con herramientas modernas como Webpack, Babel y TypeScript, lo que lo convierte en una solución versátil para proyectos de diversa escala (Vue Team, 2024, párr. 3).

En resumen, Vue.js es más que una herramienta para crear interfaces: es un framework progresivo que combina simplicidad, flexibilidad y sofisticación, permitiendo a los desarrolladores adaptar su uso a las necesidades específicas de sus proyectos.

2.3.6. TyperORM

TypeORM es una biblioteca de mapeo objeto-relacional (ORM) diseñada para operar en múltiples entornos, incluyendo Node.js, React Native, Ionic, Electron y otros, ofreciendo soporte para TypeScript y JavaScript (ES2021). Esta herramienta tiene como objetivo facilitar el desarrollo de aplicaciones que utilicen bases de datos, desde proyectos pequeños con pocas tablas hasta aplicaciones empresariales de gran escala que requieren múltiples bases de datos. Según su documentación, "TypeORM es un ORM que puede ejecutarse en NodeJS, Browser, Cordova, PhoneGap, Ionic, React Native,

NativeScript, Expo y Electron" (typeorm, 2023, párr. 1). Su versatilidad lo convierte en una excelente opción para desarrolladores que buscan una solución adaptable y eficiente en la gestión de datos.

TypeORM destaca por ser el único ORM en JavaScript que admite tanto los patrones Active Record como Data Mapper, lo que permite escribir aplicaciones escalables y mantenibles de forma productiva y con un bajo acoplamiento entre componentes. Además, su diseño ha sido influenciado por otros ORMs populares, como Hibernate, Doctrine y Entity Framework, lo que le permite aprovechar las mejores prácticas de estas herramientas para mejorar la experiencia de desarrollo en aplicaciones que dependen de bases de datos (typeorm, 2023, párr. 1).

2.3.7. TailWind CSS

En el desarrollo de interfaces web, los desarrolladores enfrentan constantemente retos asociados al diseño y la personalización de estilos. Entre estos desafíos se encuentran la necesidad de anular estilos predefinidos, mantener una especificidad adecuada y optimizar la velocidad de desarrollo y mantenimiento, especialmente en aplicaciones de gran escala. Según Carreón (2020), "el uso de herramientas avanzadas para gestionar estilos puede simplificar procesos complejos y mejorar la eficiencia en el diseño frontend" (párr. 1). Además, crear diseños personalizados que reflejen la identidad del producto o idea es esencial para destacar en un entorno altamente competitivo.

En este contexto, Tailwind CSS se presenta como una solución innovadora. Este framework, basado en PostCSS, permite construir sitios web con estilos altamente

personalizados, ofreciendo la posibilidad de ajustar colores, tamaños de borde, pesos de fuente, espaciado, puntos de interrupción, sombras y mucho más. Según Carreón (2020), "los frameworks basados en PostCSS han transformado la manera en que los desarrolladores manejan estilos, al permitir un control granular y altamente personalizable en el diseño de interfaces" (párr. 2). Una de las principales ventajas de Tailwind CSS es su capacidad para evitar los problemas comunes de personalización que suelen presentarse en otros entornos de trabajo que utilizan componentes prediseñados, los cuales, aunque prácticos, pueden llevar a estilos genéricos y a la pérdida de originalidad en los diseños.

Tailwind CSS no solo optimiza el flujo de trabajo, sino que también asegura que los desarrolladores puedan crear interfaces visualmente atractivas y funcionales sin comprometer la identidad y valores del producto (Carreón, 2020, párr. 3). Esto se logra gracias a su enfoque en clases utilitarias que facilitan la creación de estilos precisos y consistentes. Como resultado, Tailwind CSS se ha consolidado como una herramienta valiosa para diseñar interfaces modernas, potentes y alineadas con las necesidades específicas de cada proyecto.

2.4. BASE DE DATOS

Una base de datos es un sistema que organiza y almacena datos de manera centralizada, permitiendo que distintos usuarios accedan a la información de manera consistente y controlada. En contraste con los sistemas de ficheros descentralizados, una base de datos centraliza y reduce la duplicidad de información dentro de la organización. Esto

permite que los datos sean compartidos y utilizados por diferentes departamentos, eliminando inconsistencias y facilitando la independencia lógica-física de los datos. Como se menciona en el texto:

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento, sino que se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. (Marqués, 2011, p. 2)

2.4.1. Sistema de Gestión de Bases de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) es una aplicación que permite a los usuarios definir, crear, mantener y controlar el acceso a una base de datos, además de gestionar la estructura física y lógica de los datos almacenados. Este sistema separa la estructura física de la lógica, almacenando la definición de datos en un diccionario o catálogo.

Que según Marqués (2011):

El SGBD permite la inserción, actualización, eliminación y consulta de datos mediante un lenguaje de manejo de datos. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de ficheros, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de

un gran número de programas de aplicación costosos de gestionar. Hay dos tipos de lenguajes de manejo de datos: los procedurales y los no procedurales. Estos dos tipos se distinguen por el modo en que acceden a los datos. Los lenguajes procedurales manipulan la base de datos registro a registro, mientras que los no procedurales operan sobre conjuntos de registros. En los lenguajes procedurales se especifica qué operaciones se debe realizar para obtener los datos resultados, mientras que en los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL (Structured Query Language) que, de hecho, es un estándar y es el lenguaje de los SGBD relacionales. (p. 3)

Lo cual nos da esta capacidad de manejar datos de manera relacional mediante SQL que resuelve los problemas de los sistemas de ficheros, donde los datos se duplicaban y no existía un control de consistencia. Además, el SGBD incluye sistemas de seguridad, integridad, concurrencia y recuperación para garantizar el acceso controlado y confiable a la información, adaptándose a las necesidades de los usuarios mediante vistas personalizadas de la base de datos. Esto permite que el sistema sea accesible y comprensible para los usuarios sin necesidad de interactuar directamente con la estructura física de los datos, asegurando la independencia lógica-física (Marqués, 2011, pp. 3-4).

2.4.2. PostgreSQL

PostgreSQL es una herramienta ampliamente reconocida en el ámbito del software libre por su cumplimiento de estándares internacionales y funcionalidades avanzadas. Su flexibilidad, escalabilidad y adaptabilidad lo convierten en una de las opciones más utilizadas para la gestión de bases de datos. Según (Gibert, 2007), "la escalabilidad y la interoperabilidad son claves en cualquier sistema de gestión de bases de datos moderno" (p. 5). Estas características han permitido que PostgreSQL se mantenga competitivo frente a herramientas comerciales.

El origen de PostgreSQL se remonta a POSTGRES, desarrollado en la Universidad de Berkeley, y desde 1994 ha evolucionado como un sistema líder en su categoría. Como se menciona, "PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta" (Gibert, 2007, p. 5).

Gibert (2021) destaca que "PostgreSQL ha sido adoptado ampliamente debido a su adaptabilidad, lo que lo posiciona como una de las principales herramientas en el mercado de bases de datos" (p. 5). Además, su capacidad para gestionar datos estructurados y no estructurados es esencial en entornos modernos, ya que "ofrece capacidades avanzadas para ambos tipos de datos" (Gibert, 2007, p. 5). La comunidad de desarrollo también juega un papel fundamental, promoviendo mejoras constantes que lo mantienen a la vanguardia.

2.5. API Y API REST

2.5.1. API

Según Red Hat (2023), "Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar sistemas de software de las aplicaciones" (párr. 2). Además, permite que las empresas compartan recursos e información mientras garantizan la seguridad y autenticación.

Frente al tema, "las API actúan como mediadores entre los usuarios o clientes y los recursos o servicios web, definiendo los accesos y permitiendo la interacción sin necesidad de conocer detalles internos" (Red Hat, 2023, párr. 3). Esto las convierte en herramientas clave para la interoperabilidad de sistemas.

Red Hat (s.f.) afirmó lo siguiente:

Una API puede considerarse como el contrato entre el usuario y el proveedor de información, estableciendo el contenido necesario en la llamada y la respuesta requerida. Por ejemplo, una API de servicio meteorológico podría requerir un código postal como entrada y devolver una temperatura máxima y mínima como respuesta (párr. 4).

2.5.2. REST

Según Red Hat (2023), "REST no es un protocolo ni un estándar, sino un conjunto de límites relacionados con la arquitectura" (párr. 5). Este estilo arquitectónico define cómo estructurar APIs para garantizar interoperabilidad y eficiencia en la web.

Frente al tema, Red Hat (2023) afirma que "los principios fundamentales de REST incluyen una arquitectura cliente-servidor, comunicación sin estado y almacenamiento en caché, lo cual optimiza las interacciones entre cliente y servidor" (párr. 6).

Se describe que:

Los desarrolladores de las API pueden implementar REST de diversas maneras. Cuando el cliente envía una solicitud a través de una API RESTful, esta transfiere una representación del estado del recurso requerido al cliente o extremo. La información se entrega por medio de HTTP en formatos como JSON, XML, o texto plano (Red Hat, 2023, párr. 7).

2.5.3. API REST

Frente al tema, "una API REST es una interfaz que sigue los principios arquitectónicos de REST para facilitar la comunicación entre sistemas, optimizando el uso de recursos con respuestas en formatos estándar como JSON o XML" (Red Hat, 2023, párr. 8).

Red Hat (2023) afirmó lo siguiente:

Para que una API sea considerada RESTful, debe cumplir varios principios, como arquitectura cliente-servidor, comunicación sin estado, datos almacenables en caché, una interfaz uniforme, mensajes autodescriptivos y un sistema jerárquico en capas. Opcionalmente, puede incluir código bajo demanda para extender las capacidades del cliente (párr. 9).

Por lo tanto, "las API REST son ideales para aplicaciones modernas debido a su flexibilidad, velocidad y adaptabilidad, especialmente en entornos como el desarrollo de aplicaciones móviles y el Internet de las cosas (IoT)" (Red Hat, 2023., párr. 10).

2.6. INVENTARIO

El inventario representa un activo esencial para las empresas, actuando como respaldo en la cadena de suministro y asegurando la continuidad de las operaciones. Los inventarios son materiales o productos almacenados que las organizaciones mantienen para cubrir la demanda en momentos específicos, protegiéndose contra posibles interrupciones en el flujo de suministro.

Según Muller (2003), se destaca lo siguiente:

El inventario es un activo, pero un tipo de activo del cual las empresas no quieren en exceso. Sin embargo, no tener 'en exceso' pondría a la organización en riesgo de posibles interrupciones en la cadena de suministro y de costos extremos imprevistos. Entonces, la clave para una administración efectiva de los inventarios es el equilibrio: mantener los inventarios adecuados para garantizar la producción continua y los flujos comerciales, al mismo tiempo que se minimiza la inversión de inventario para asegurar un desempeño financiero sólido (p. 4).

Así, los inventarios cumplen un rol de equilibrio: permiten a las empresas responder a la demanda y asegurar la estabilidad en el suministro de productos y materiales, al mismo tiempo que facilitan la planificación y administración de recursos.

Este activo es una parte integral del sistema productivo, ya que su existencia garantiza que los materiales y productos necesarios estarán disponibles en el momento adecuado, sin depender completamente de las fluctuaciones en la producción o el transporte. A medida que las cadenas de suministro se vuelven más complejas y las empresas enfrentan demandas variables, el inventario se convierte en un recurso estratégico que, si se gestiona adecuadamente, puede minimizar los riesgos y contribuir al éxito operativo (Muller, 2003, p. 6).

2.6.1. Métodos de Control de Inventarios

La gestión de inventarios es un componente esencial en la administración eficiente de recursos dentro de una organización. Los métodos de control y valorización de inventarios permiten optimizar los procesos de almacenamiento, distribución y reposición de productos, garantizando un balance entre disponibilidad y costos. Desde técnicas como el método ABC, que clasifica los productos según su impacto financiero, hasta el Just in Time (JIT), que minimiza el almacenamiento innecesario, cada metodología responde a necesidades específicas del mercado y del modelo de negocio. Su correcta implementación no solo asegura el flujo constante de bienes, sino que también reduce riesgos, mejora la toma de decisiones y potencia la competitividad empresarial. (Gonzales, 2023, párr. 2)

Tabla 1

Tipos de gestión de inventarios

Método	Descripción
Método ABC	Clasifica las existencias en tres categorías (A, B, C) basadas en su importancia, volumen y precio. Los artículos clase A son de alta gama, los B de prioridad media, y los C de bajo valor pero alto volumen de ventas.
Método PEPS/FIFO	"Primeras entradas, primeras salidas". Prioriza las existencias más antiguas para garantizar productos frescos o evitar deterioros.
Método UEPS/LIFO	"Últimas entradas, primeras salidas". Despacha primero los lotes más recientes, para inventarios de mediano a pequeño tamaño.
Método EOQ	Busca calcular el monto de pedido que minimice los costos de inventario. Funciona con una demanda constante y reabastecimiento inmediato tras agotar el stock.
Conteo cíclico	Conteo periódico del inventario. Puede integrarse con el método ABC asignando distintas frecuencias de conteo para cada clase (A, B, C).
Stock de Seguridad	Mantiene un nivel adicional de productos como reserva para anticipar fluctuaciones de la demanda o retrasos en el suministro.
Seguimiento de Lotes	Organiza productos según la fecha de producción y materias primas utilizadas, permitiendo rastrear procedencia, destino y caducidad.
Método JIT (Just in Time)	Gestiona inventarios para que los productos estén disponibles solo en el momento necesario, reduciendo desperdicios y costos de almacenamiento.
Promedio Ponderado	Calcula el costo promedio de inventarios basado en precios y cantidades de cada lote.

Nota. Elaboración propia

Para el siguiente proyecto se hará uso del promedio ponderado.

2.7. UNE-ISO/IEC 27001:2023

La Norma UNE-ISO/IEC 27001:2023 establece los requisitos para la creación, implementación, mantenimiento y mejora continua de un Sistema de Gestión de la Seguridad de la Información (SGSI). Según la norma, la adopción de un SGSI es una decisión estratégica que permite a las organizaciones proteger sus activos más valiosos: la información. Esta protección se logra preservando la confidencialidad, integridad y disponibilidad de los datos a través de un proceso continuo de gestión de riesgos.

Como señala el documento,

El sistema de gestión de la seguridad de la información preserva la confidencialidad, integridad y disponibilidad de la información mediante la aplicación de un proceso de gestión de riesgos y otorga a las partes interesadas confianza sobre la adecuada gestión de los riesgos (ISO/IEC, 2023, p. 6).

El establecimiento de un SGSI se ajusta a las necesidades específicas de la organización, considerando sus objetivos, requisitos de seguridad, procesos internos, tamaño y estructura. Estos factores cambian con el tiempo, lo que obliga a las organizaciones a mantener un enfoque dinámico y adaptativo (ISO/IEC, 2023, p. 7). Además, el SGSI no es un sistema aislado; su implementación debe estar completamente integrada con los procesos organizativos y la estructura de gestión general. Este enfoque asegura que la seguridad de la información esté presente desde la fase de diseño de procesos y sistemas, fortaleciendo la confianza y eficiencia en la protección de datos.

La norma no establece un orden estricto para la implementación de los requisitos, lo que permite a las organizaciones priorizar en función de sus contextos y necesidades específicas. Además, se alienta a las partes interesadas, tanto internas como externas, a utilizar el SGSI para evaluar la capacidad de la organización de cumplir con sus requisitos de seguridad.

Lo cual según la norma:

La ISO/IEC 27000, como parte de la misma familia de normas, proporciona una visión general y un vocabulario común, incluyendo referencias a normas complementarias como la ISO/IEC 27003 (guía de implementación), ISO/IEC 27004 (métricas de seguridad) y ISO/IEC 27005 (gestión de riesgos) (ISO/IEC, 2023, p. 8).

En esencia, la implementación de un SGSI según la UNE-ISO/IEC 27001:2023 fortalece la capacidad de las organizaciones para gestionar los riesgos relacionados con la información, ofreciendo un marco sólido y reconocido internacionalmente para la protección de datos en un entorno dinámico y competitivo.

2.7.1. Principios Fundamentales de la UNE-ISO/IEC 27001:2023

2.7.1.1. Contexto de la organización

El éxito de un Sistema de Gestión de la Seguridad de la Información (SGSI) depende de su capacidad para adaptarse al contexto organizacional, tanto interno como externo. La norma ISO/IEC 27001 establece que las organizaciones deben identificar y comprender

las condiciones y factores que impactan su propósito y capacidad para alcanzar los resultados deseados del SGSI. Este análisis incluye aspectos económicos, legales, tecnológicos y culturales que afectan la seguridad de la información. Según la norma, "la organización debe determinar las cuestiones externas e internas que son pertinentes para su propósito y que afectan su capacidad para lograr los resultados previstos de su sistema de gestión de la seguridad de la información" (UNE-ISO/IEC, 2023, p. 8).

2.7.1.2. Necesidades y expectativas de las partes interesadas

La norma requiere que las organizaciones identifiquen las partes interesadas que son relevantes para el SGSI, junto con sus necesidades y expectativas específicas (ISO/IEC, 2023, p. 8). Estas partes interesadas pueden incluir:

- Clientes, que demandan confidencialidad y protección de datos.
- Reguladores, que exigen cumplimiento normativo.
- Proveedores, que necesitan garantías de integridad en las transacciones.
- Empleados, que buscan herramientas seguras para manejar información.

Según la norma, "la organización debe determinar las partes interesadas relevantes para el SGSI y cuáles de sus requisitos se abordarán mediante este sistema" (UNE-ISO/IEC, 2023, p. 8). Esto permite al SGSI alinearse con los objetivos estratégicos y operativos de la organización, garantizando una gestión efectiva de los riesgos relacionados con la seguridad de la información.

Al incorporar estas expectativas, las organizaciones logran construir confianza con las partes interesadas, mejorar su resiliencia ante riesgos y mantener la integridad de sus procesos operativos.

2.7.2. Soporte en el Sistema de Gestión de la Seguridad de la Información (SGSI)

2.7.2.1. Recursos necesarios

Un Sistema de Gestión de la Seguridad de la Información (SGSI) requiere recursos adecuados para su establecimiento, implementación, mantenimiento y mejora continua. Según la norma ISO/IEC 27001, "la organización debe determinar y proporcionar los recursos necesarios para el establecimiento, implementación, mantenimiento y mejora continua del sistema de gestión de la seguridad de la información" (UNE-ISO/IEC, 2023, p. 13). Estos recursos incluyen infraestructura, tecnología, personal capacitado y soporte técnico. La disponibilidad de estos recursos es esencial para garantizar la eficacia y continuidad del SGSI, promoviendo una gestión adecuada de la seguridad de la información (ISO/IEC, 2023, p. 13).

2.7.2.2. Competencia y capacitación

La competencia del personal que opera bajo el SGSI es un factor clave para su éxito. La norma establece que "la organización debe determinar la competencia necesaria de las personas que realizan, bajo su control, un trabajo que afecta a su desempeño en seguridad de la información" (UNE-ISO/IEC, 2023, p. 13).

Además, según la norma,

Se debe garantizar que el personal posea la formación, educación o experiencia adecuada, y cuando sea necesario, implementar acciones como programas de formación o tutorías para cubrir cualquier brecha en las competencias. También se debe conservar evidencia documentada de la competencia del personal, asegurando el desempeño efectivo del SGSI (UNE-ISO/IEC, 2023, p. 14).

2.7.2.3. Concienciación del personal

La concienciación es un componente fundamental para el éxito del SGSI. Según la norma, "las personas que trabajan bajo el control de la organización deben ser conscientes de la política de la seguridad de la información, su contribución a la eficacia del SGSI, y las implicaciones de no cumplir con los requisitos del sistema" (UNE-ISO/IEC, 2023, p. 13). Haciendo así que al promover una cultura organizacional que valore la seguridad de la información asegura un mayor compromiso por parte de los empleados, lo que fortalece el sistema y reduce riesgos relacionados con fallos humanos.

2.7.2.4. Comunicación efectiva

La comunicación interna y externa desempeña un papel crucial en la gestión del SGSI. Las organizaciones deben determinar el contenido, los destinatarios, los tiempos y los medios adecuados para transmitir la información.

Como señala la norma,

La organización debe determinar la necesidad de comunicaciones internas y externas pertinentes al sistema de gestión de la seguridad de la información,

incluyendo su contenido, cuándo comunicar, con quién comunicar y cómo comunicar (UNE-ISO/IEC, 2023, p. 14).

Siendo que un flujo de comunicación claro asegura que todos los involucrados estén alineados con los objetivos de seguridad de la organización, facilitando la implementación de medidas de seguridad eficaces.

2.7.2.5. Gestión de la información documentada

La información documentada es un componente esencial del SGSI. La norma detalla varios aspectos clave para su gestión:

- a) **Requisitos Generales:** "El sistema de gestión de la seguridad de la información de la organización debe incluir la información documentada requerida por este documento y aquella que la organización ha determinado como necesaria para la eficacia del sistema" (UNE-ISO/IEC, 2023, p. 14).
- b) **Creación y Actualización:** "Cuando se crea y actualiza la información documentada, la organización debe asegurarse de la identificación, descripción, formato y revisión adecuada para su idoneidad y adecuación" (UNE-ISO/IEC, 2023, p. 14).
- c) **Control de Documentación:** La norma indica que "la información documentada debe estar disponible y protegida contra pérdida de confidencialidad, uso inadecuado o pérdida de integridad, y controlada mediante actividades como distribución, almacenamiento y control de cambios" (UNE-ISO/IEC, 2023, p. 14).

La correcta gestión de la documentación asegura que los procesos del SGSI sean consistentes y rastreables, permitiendo un monitoreo efectivo de su implementación.

2.7.3. Contexto de la Organización

2.7.3.1. Comprensión de la organización y de su contexto

El éxito de un Sistema de Gestión de la Seguridad de la Información (SGSI) depende de su capacidad para adaptarse al contexto organizacional. Según la UNE-ISO/IEC (2023), "la organización debe determinar las cuestiones externas e internas que son pertinentes para su propósito y que afectan a su capacidad para lograr los resultados previstos de su sistema de gestión de la seguridad de la información" (p. 7). Estas cuestiones pueden incluir factores económicos, tecnológicos, legales, culturales y sociales, así como aspectos internos como la estructura organizativa, los procesos y las capacidades existentes. Además, la norma recomienda referirse al apartado 5.4.1 de la ISO 31000:2018 para un análisis sistemático y estructurado del contexto organizacional, permitiendo anticipar riesgos y oportunidades (UNE-ISO/IEC, 2023, p. 9).

2.7.3.2. Comprensión de las necesidades de las partes interesadas

El SGSI debe alinearse con las expectativas y requisitos de las partes interesadas clave. La norma establece que "la organización debe determinar las partes interesadas que son relevantes para el sistema de gestión de la seguridad de la información, así como sus requisitos relevantes" (UNE-ISO/IEC, 2023, p. 7).

Según la norma "Estos requisitos pueden incluir obligaciones legales, regulatorias, contractuales y de cumplimiento, que deben ser considerados para garantizar la efectividad del SGSI" (UNE-ISO/IEC, 2023, p. 7). Lo cual al identificar y priorizar estas

necesidades permite a las organizaciones diseñar un sistema que no solo cumpla con los estándares internacionales, sino que también se alinee con sus objetivos estratégicos.

2.7.3.3. Sistema de gestión de la seguridad de la información

El núcleo de la norma ISO/IEC 27001 reside en el establecimiento y mantenimiento de un SGSI. Como se indica, "la organización debe establecer, implementar, mantener y mejorar de manera continua un sistema de gestión de la seguridad de la información, incluyendo los procesos requeridos y sus interacciones de acuerdo con los requisitos de este documento" (UNE-ISO/IEC, 2023, p. 8). Al tener este enfoque integrado permite a las organizaciones gestionar la seguridad de la información de manera coherente y efectiva, incorporando procesos como la gestión de riesgos, la evaluación de controles y la mejora continua, asegurando así la protección de la información crítica frente a amenazas internas y externas.

2.7.4. Compatibilidad con Otras Normas de Sistemas de Gestión

La norma UNE-ISO/IEC 27001:2023 está diseñada para garantizar la compatibilidad con otras normas de sistemas de gestión mediante el uso de la estructura de alto nivel, términos y definiciones comunes establecidos en el Anexo SL de las Directivas ISO/IEC. Según la norma, "este enfoque común definido en el Anexo SL será útil para aquellas organizaciones que deciden implantar un sistema de gestión que cumpla con los requisitos de dos o más normas de sistemas de gestión" (UNE-ISO/IEC, 2023, p. 6).

Esto permite a las organizaciones integrar múltiples sistemas de gestión, como ISO 9001 (gestión de calidad) o ISO 14001 (gestión ambiental), de manera más eficiente,

reduciendo duplicidades y mejorando la coherencia entre procesos (UNE-ISO/IEC, 2023). Siendo así que un marco compartido proporciona una base estandarizada para gestionar distintos aspectos organizacionales bajo un enfoque integrado, facilitando la implementación y el mantenimiento de varios estándares en una única estructura operativa.

2.7.5. Controles Tecnológicos

Un control tecnológico es una medida implementada para garantizar la seguridad de la información mediante el uso de tecnologías específicas que según los principios establecidos en la norma estos controles tienen como objetivo mitigar los riesgos asociados a la confidencialidad, integridad y disponibilidad de los datos dentro de una organización (UNE-ISO/IEC, 2023, p. 15). Los controles tecnológicos abarcan una amplia gama de herramientas y procedimientos, entre los cuales se destacan:

- Cifrado de datos, que asegura que la información almacenada o transmitida sea accesible únicamente por usuarios autorizados, reduciendo el riesgo de accesos no autorizados.
- Autenticación multifactor (MFA), que añade capas de verificación para garantizar que solo personas legítimas puedan acceder al sistema.
- Sistemas de prevención de intrusos (IPS), diseñados para identificar y bloquear actividades maliciosas que puedan comprometer los datos o sistemas.
- Control de acceso basado en roles (RBAC), que restringe las acciones de los usuarios según sus responsabilidades y niveles de autorización dentro del sistema.

Estos controles, alineados con el Anexo A de la norma ISO/IEC 27001, constituyen una base fundamental para proteger los activos de información, permitiendo a las organizaciones gestionar los riesgos tecnológicos de manera efectiva y asegurar el cumplimiento de los estándares internacionales en seguridad de la información (UNE-ISO/IEC, 2023, p. 15).

2.7.6. Comparación de la versión 2023 contra la versión 2013

La evolución de los estándares de seguridad de la información en la norma ISO/IEC 27001 refleja la necesidad de adaptarse a los avances tecnológicos y a los riesgos emergentes. Este apartado presenta una comparación estas normas.

Tabla 2
Comparación entre versiones de la ISO 27001

Aspecto	ISO/IEC 27001:2013	ISO/IEC 27001:2023
Estructura de controles	Contaba con 114 controles agrupados en 14 cláusulas (Anexo A).	Ahora incluye 93 controles reorganizados en 4 categorías: organizativos, personales, físicos y tecnológicos.
Enfoque en controles tecnológicos	Controles como cifrado, control de accesos, registro de eventos, entre otros.	Introduce conceptos actualizados como inteligencia sobre amenazas, seguridad en la nube y filtrado de tráfico web.
Cifrado de datos	Recomendaba el cifrado de datos sensibles durante el almacenamiento y la transmisión.	Sigue recomendando el cifrado, pero ahora enfatiza estándares modernos y estrategias para entornos complejos (p. ej., nube).

Gestión de accesos	Control de acceso basado en privilegios mínimos y gestión de identidades.	Amplía la gestión de accesos con enfoque en gestión de identidades digitales y accesos en múltiples entornos.
Seguridad en la nube	Mencionada indirectamente en el control de terceros y en las comunicaciones seguras.	Introduce un control específico para seguridad en la nube, alineado con las necesidades actuales.
Monitoreo y registro	Recomendaba el monitoreo de eventos de seguridad y registros de auditoría.	Refuerza este control con análisis avanzados de registros y capacidades de detección proactiva de amenazas.
Gestión de incidentes	Enfocada en la respuesta a incidentes y recuperación.	Amplía la gestión de incidentes, incluyendo automatización en respuestas y detección basada en inteligencia artificial.
Innovaciones incluidas	No incluía referencias específicas a tecnologías emergentes.	Añade nuevos controles como prevención de pérdida de datos (DLP), filtrado de tráfico web y gestión de amenazas.

Nota. Elaboración propia.

2.8. MARCO DE TRABAJO SCRUM

El surgimiento de la agilidad como enfoque en la gestión de proyectos puede entenderse como una respuesta a las limitaciones del modelo en cascada en entornos cada vez más complejos. Durante los años 90, el Reporte CHAOS de 1994 puso de manifiesto las bajas tasas de éxito de los proyectos gestionados bajo metodologías tradicionales, evidenciando la necesidad de enfoques más flexibles y adaptativos. Esto dio lugar a las llamadas Metodologías Livianas, entre las que destacan Extreme Programming (XP),

Scrum, Software Craftmanship y Lean Software Development, las cuales ofrecían alternativas más alineadas con la naturaleza cambiante de los proyectos de software.

En febrero de 2001, un grupo de 17 expertos en desarrollo de software y representantes de estas metodologías livianas se reunió en Utah, Estados Unidos. De esta reunión surgió el Manifiesto por el Desarrollo Ágil de Software, una declaración de valores y principios que transformó la forma de gestionar proyectos de software. Según Alaimo (2021), "el manifiesto no solo dio identidad y unidad a diversas metodologías, sino que también inició un movimiento que continúa evolucionando con el tiempo". (p. 55)

Sobre este momento clave, se afirma lo siguiente:

El Manifiesto para el Desarrollo Ágil de Software, más allá de dar unidad e identidad a una serie de métodos y patrones de práctica, ha determinado el surgimiento de un movimiento. Un movimiento en el que cada uno de quienes nos acercamos a él y nos sentimos cautivados, desde ese preciso momento pasamos a formar parte responsable de cómo los valores, principios y prácticas ágiles se llevan a la realidad de las organizaciones, a la vez que construimos nuevos enfoques, los probamos y mejoramos con vistas al futuro. (Alaimo, 2020, p. 14).

El Manifiesto Ágil marcó un hito al proponer una alternativa a los procesos rígidos y dominados por la documentación que caracterizaban al desarrollo de software en esa época. Frente a una complejidad que, según DeGrace y Hulet Stahl (1990), consistía en gestionar cambios frecuentes para un número limitado de usuarios, hoy en día las aplicaciones deben adaptarse a millones de usuarios en constante demanda de nuevas

funcionalidades. Esto refleja cómo la complejidad de los años 90 ha escalado a órdenes de magnitud mayores en la actualidad.

Además de establecer valores y principios, el movimiento ágil se caracteriza por su dinamismo y evolución. Alaimo (2021) señala que "el manifiesto sigue tan vigente como en sus inicios, inspirando nuevas prácticas y enfoques que son probados y mejorados constantemente" (p. 17). Esta capacidad de adaptarse a los desafíos del presente asegura que los métodos ágiles sigan siendo relevantes para responder a las demandas del mercado, la tecnología y las organizaciones modernas.

2.8.1. Roles

En Scrum, los roles fundamentales son el Product Owner, el Scrum Master y el Desarrollador, cada uno con responsabilidades específicas dentro del equipo. Estos roles están diseñados para fomentar la colaboración y garantizar la entrega de Incrementos al final de cada Sprint. Según Alaimo (2021), "Scrum no reconoce ningún tipo de jerarquía dentro de tu equipo, subequipos ni roles que no sean los mencionados previamente" (p. 99). Esto subraya la importancia de que los miembros del equipo trabajen de manera autogestionada y multifuncional.

2.8.1.1. Product owner

El Product Owner es responsable de maximizar el valor del producto y gestionar el Product Backlog. Este rol implica una constante comunicación con los stakeholders para priorizar las tareas y garantizar que el equipo Scrum trabaje en las funcionalidades de mayor valor. Según Alaimo (2021), el Product Owner "colabora con los stakeholders,

investiga, experimenta, negocia y aprende para asegurar que el Incremento final sea de valor" (p. 99).

Es importante destacar que este rol no puede combinarse con el de Scrum Master, ya que sus funciones pueden entrar en conflicto. Al respecto, Alaimo (2021) afirma que "no debes ocupar el rol de Product Owner y el de Scrum Master juntos dado que ambos roles funcionan por oposición de intereses" (p. 99). Esta separación garantiza la objetividad en la planificación y ejecución del Sprint.

2.8.1.2. Scrum master

El Scrum Master actúa como facilitador del equipo, asegurándose de que los principios y prácticas de Scrum se respeten. Este rol es fundamental para eliminar obstáculos que puedan surgir durante el desarrollo. Según Alaimo (2021), "el Scrum Master colabora con el equipo en la autogestión, asegurando que las decisiones sobre qué, cuándo y cómo realizar el trabajo sean potestad exclusiva del equipo" (p. 100).

Además, el Scrum Master tiene la responsabilidad de promover la autogestión del equipo y fomentar la multifuncionalidad. Esto implica no solo resolver impedimentos, sino también facilitar la capacitación y el aprendizaje continuo del equipo.

2.8.1.3. Desarrollo

Los Desarrolladores son responsables de construir el Incremento que se entregará al final de cada Sprint. Este rol no solo se limita a la construcción técnica, sino que también abarca actividades como la investigación, la experimentación y la colaboración con los

stakeholders. Según Alaimo (2021), "el Equipo Scrum es responsable de hacer todo lo necesario para entregar un Incremento al final de cada Sprint" (pp. 99-100).

Los desarrolladores trabajan en un entorno multifuncional, lo que significa que poseen todas las habilidades necesarias para completar las tareas asignadas sin depender de personas externas al equipo. Alaimo (2021) explica que "se espera que dentro de tu Equipo Scrum cuentes con todas las habilidades necesarias para poder entregar Incrementos de valor al final de cada Sprint" (p. 100).

2.8.2. Sprint Backlog

El Sprint Backlog constituye una herramienta esencial en Scrum, ya que describe el trabajo necesario para alcanzar el objetivo establecido en cada Sprint. Según Alaimo (2021), el Sprint Backlog está compuesto por el objetivo del Sprint, un conjunto de PBIs (Product Backlog Items) seleccionados, y un plan de acción detallado para entregar el Incremento al finalizar el Sprint (p. 52). Este enfoque garantiza claridad y dirección al equipo durante el desarrollo.

2.8.3. Objetivo del Sprint

El objetivo del Sprint define la razón principal para ejecutar el trabajo planificado en el Sprint. Es una guía que proporciona un propósito claro para el Equipo Scrum.

Alaimo (2021) afirma que:

Para el Objetivo del Sprint describe la razón por la cual vale la pena realizar el trabajo del Sprint y, si eres desarrollador, te proporciona un norte para determinar si el camino recorrido te está conduciendo hacia el lugar esperado (p. 52).

En entornos complejos, donde no es posible planificar con precisión, el Sprint Backlog actúa como un plan dinámico que evoluciona durante el Sprint. Esto permite la inspección constante del progreso y la adaptación en función de los desafíos encontrados. Según Alaimo (2021), "contar con un objetivo les permitirá inspeccionar su progreso y tomar decisiones de adaptación del plan del Sprint de manera frecuente" (p. 8).

CAPITULO III

MARCO PRACTICO

3.1. IMPLEMENTACIÓN DE SCRUM PARA EL DESARROLLO DEL PROYECTO

3.1.1. Roles en el Marco Scrum

La siguiente tabla detalla los roles establecidos para el equipo Scrum para el desarrollo del proyecto.

Tabla 3*Roles de scrum*

Rol	Nombre
Product Owner	Maritza Netzy Paiva Zapana
Scrum Master	Rodmy Orellana Illanes
Desarrollador	Daniel Santiago Soto Villamil
Tester	Marco Antonio Luna Gonzales
Diseñador	Marlene Monica Suntura Chura

Nota. Elaboración propia.

3.1.2. Historias de Usuario

Las historias se puntuarán con en base a la estimación de complejidad siendo 1 lo más bajo y 10 lo más alto.

3.1.2.1. Historia de administrador

Tabla 4

Historia de administrador

Historia de Administrador		Estimación: 8	ID: HU-1
Historia de Usuario	Como administrador, quiero gestionar el personal del sistema (registrar, actualizar, desactivar), para mantener un control adecuado de los usuarios y sus datos.		
Criterios de Aceptación:	Dado que el administrador accede al módulo de personal, Cuando realiza una acción (registro, actualización o desactivación) y confirma, Entonces el sistema debe reflejar los cambios de manera inmediata y correcta.		

Nota. Elaboración propia.

3.1.2.2. Historia de encargado proveedores-productos

Tabla 5

Historia de encargado proveedores-productos

Historia de Encargado Proveedores-Productos		Estimación: 6	ID: HU-2
Historia de Usuario	Como encargado de proveedores-productos, quiero gestionar insumos y productos (registrar, actualizar, asociar), para garantizar un control eficiente del inventario.		
Criterios de Aceptación:	Dado que el encargado accede al módulo de proveedores-productos, Cuando realiza una acción (registro, asociación o actualización) y confirma, Entonces el sistema debe guardar y reflejar los cambios en el inventario.		

Nota. Elaboración propia.

3.1.2.3. Historia de encargado trabajos

Tabla 6

Historia de encargado trabajos

Historia de Encargado Trabajos		Estimación: 6	ID: HU-3
Historia de Usuario	Como encargado de trabajos, quiero gestionar los trabajos ópticos (registrar, asignar, actualizar), para garantizar un seguimiento claro y eficiente de los servicios.		
Criterios de Aceptación:	Dado que el encargado accede al módulo de trabajos, Cuando realiza una acción (registro, asignación o actualización) y confirma, Entonces el sistema debe guardar los cambios y mostrar el estado actualizado de los trabajos.		

Nota. Elaboración propia.

3.1.2.4. Historia de encargado ventas

Tabla 7

Historia de encargado ventas

Historia de Encargado Ventas		Estimación: 6	ID: HU-4
Historia de Usuario	Como encargado de ventas, quiero gestionar las ventas (registrar, actualizar, consultar), para llevar un control detallado de las transacciones realizadas.		
Criterios de Aceptación:	Dado que el encargado accede al módulo de ventas, Cuando realiza una acción (registro, consulta o actualización) y confirma, Entonces el sistema debe reflejar las operaciones en el historial de ventas.		

Nota. Elaboración propia.

3.1.3. Product Backlog

Tabla 8

Product backlog

ID	Historia de Usuario	Estimación	Duración (hr)
PB-1	El administrador requiere registrar nuevos miembros del personal con datos básicos, como nombre y contacto, para mantener un registro organizado.	4	39.3
PB-2	El administrador necesita editar la información del personal registrado, asegurando que los datos estén siempre actualizados y consistentes.	3	26.2

PB-3	El administrador requiere desactivar lógicamente a miembros del personal que ya no formen parte de la organización, para evitar errores en la gestión.	2	26.2
PB-4	El administrador solicita registrar nuevos usuarios asociados al personal existente, asignándoles un rol predefinido, para gestionar su acceso al sistema de manera controlada.	5	39.3
PB-5	El administrador necesita asignar roles predefinidos a los usuarios registrados, asegurando que cada uno tenga acceso exclusivamente a las funcionalidades correspondientes a su rol.	6	39.3
PB-6	Los usuarios deben acceder únicamente a las funcionalidades permitidas según el rol asignado, para evitar acciones no autorizadas y garantizar la seguridad del sistema.	4	26.2
PB-7	El supervisor requiere consultar los roles asignados a los usuarios para verificar que los accesos estén configurados correctamente.	2	13.1
PB-8	El administrador requiere registrar nuevos proveedores con datos básicos, como nombre y contacto, para gestionar adecuadamente su relación con los productos del inventario.	3	52.4
PB-9	El administrador solicita registrar productos con información detallada, como nombre, stock y precios, para garantizar un control preciso y actualizado del inventario.	5	39.3
PB-10	El administrador necesita asignar proveedores específicos a los productos registrados, facilitando la trazabilidad y la gestión eficiente del suministro.	7	52.4

PB-11	El administrador desea desactivar proveedores que ya no trabajen con la empresa para mantener el sistema organizado y depurado.	3	26.2
PB-12	El administrador requiere registrar trabajos ópticos con detalles técnicos, como colores y tratamientos, para organizar y documentar los servicios ofrecidos de manera efectiva.	4	39.3
PB-13	El usuario encargado necesita actualizar el estado de los trabajos (pendiente, en proceso, finalizado, entregado) con el fin de llevar un control claro del progreso.	3	26.2
PB-14	El administrador solicita asociar productos, tratamientos y colores a los trabajos registrados para garantizar un seguimiento adecuado de los insumos utilizados.	6	39.3
PB-15	El administrador requiere asignar trabajos al personal encargado para identificar responsabilidades y distribuir de manera eficiente la carga de trabajo.	5	39.3
PB-16	El administrador solicita registrar ventas con detalles completos, como productos vendidos y precios, para mantener un control adecuado de las transacciones realizadas.	7	52.4
PB-17	El usuario encargado necesita ingresar los productos y servicios vendidos con el fin de calcular correctamente el monto total de cada venta.	6	39.3
PB-18	El administrador solicita visualizar el historial de ventas para realizar un seguimiento eficiente de las operaciones y detectar patrones o tendencias.	3	26.2

PB-19	El usuario encargado desea modificar una venta registrada para corregir errores en los detalles de la transacción y para la exactitud de la información.	4	39.3
PB-20	El administrador requiere filtrar las ventas por fecha y usuario con el objetivo de analizar la información de manera eficiente.	5	39.3

Nota. Elaboración propia.

3.1.4. Tabla de Requerimientos

La prioridad se establecerá en 3 niveles de prioridad bajo, media y alta, para establecer su orden de desarrollo.

Tabla 9

Tabla de requerimientos

ID	Tarea	Descripción	Prioridad	Historia
R-1	Registro de personal	Permitir registrar datos del personal (nombres, apellidos, email, etc.).	Alta	HU-1
R-2	Actualización de personal	Editar información del personal.	Alta	HU-1
R-3	Eliminación lógica de personal	Marcar como inactivo a personal no asociado a procesos del sistema.	Media	HU-1
R-4	Registro de usuarios	Asociar usuarios al personal existente con credenciales seguras.	Alta	HU-1
R-5	Configuración de autenticación	Permitir el inicio de sesión con validación segura.	Alta	HU-1
R-6	Gestión de roles	Crear, editar y eliminar roles según requerimientos.	Alta	HU-1

R-7	Registro de proveedores	Registrar nuevos proveedores con datos básicos.	Alta	HU-2
R-8	Actualización de proveedores	Editar información de los proveedores registrados.	Media	HU-2
R-9	Eliminación lógica de proveedores	Marcar proveedores como inactivos para mantener orden en el sistema.	Media	HU-2
R-10	Registro de productos	Permitir registrar productos con detalles completos.	Alta	HU-2
R-11	Actualización de productos	Editar información de productos como stock y precios.	Alta	HU-2
R-12	Eliminación lógica de productos	Marcar productos como inactivos.	Media	HU-3
R-13	Relación de productos con proveedores	Asignar uno o varios proveedores a productos registrados.	Media	HU-3
R-14	Registro de trabajos	Registrar trabajos ópticos con detalles técnicos.	Alta	HU-3
R-15	Gestión de parámetros técnicos	Registrar y editar colores, tratamientos y características ópticas.	Media	HU-3
R-16	Actualización de trabajos	Editar información de trabajos ya registrados.	Media	HU-3
R-17	Seguimiento de estados de trabajos	Administrar el estado de los trabajos.	Media	HU-4
R-18	Registro de ventas	Capturar detalles de ventas realizadas.	Alta	HU-4
R-19	Generación de detalles de ventas	Registrar los servicios ópticos asociados a cada venta.	Media	HU-4
R-20	Visualización de historial de ventas	Mostrar un historial detallado de las ventas realizadas.	Media	HU-4

Nota. Elaboración propia.

3.2. SPRINT 1: GESTIÓN DE AUTENTICACIÓN

3.2.1. Objetivos del Sprint

El objetivo es desarrollar y validar las funcionalidades esenciales del módulo de gestión de Autenticación, que será la base para controlar los accesos y funcionalidades del sistema. Esto incluye las siguientes metas:

a) Gestión de Roles Predefinidos:

- Implementar la creación inicial de roles predefinidos en el sistema mediante seeding (Inserción de Datos por código).
- Validar que estos roles no sean editables ni eliminables por los usuarios del sistema.

b) Gestión de Usuarios:

- Implementar la lógica para la creación, edición y eliminación de usuarios desde el sistema administrativo.
- Validar que los usuarios puedan ser asignados únicamente a roles predefinidos.
- Garantizar que los usuarios desactivados no tengan acceso al sistema.

c) Validación de Accesos:

- Implementar validaciones que aseguren que cada usuario accede únicamente a las funcionalidades permitidas según su rol asignado.

d) Seguridad en el Proceso de Autenticación:

- Implementar autenticación segura basada en tokens (JWT) y bcrypt para el manejo de contraseñas.

- Configurar restricciones para evitar accesos no autorizados mediante mecanismos de verificación como tokens de acceso con expiración y validaciones estrictas en el inicio de sesión.
- Proteger los puntos de accesos relacionados con roles y usuarios mediante middlewares de autenticación y validación de roles.

3.2.2. Definición de la Gestión de Autenticación

Identificación de las funcionalidades necesarias para el correcto funcionamiento del módulo de Autenticación.

Tabla 10

Requisitos de la gestión de autenticación

REQUISITO	DESCRIPCIÓN
Gestión de roles predefinidos	Implementar roles preexistentes (Administrador, Supervisor, Encargado de Ventas, Encargado de Trabajos) que sean inmutables.
Relación roles- usuarios	Establecer la asociación entre los usuarios y los roles predefinidos para determinar accesos y funcionalidades disponibles.
Validación de accesos	Restringir accesos y funcionalidades según el rol asignado a cada usuario.
Integridad de roles	Asegurar que los roles predefinidos no puedan ser creados, editados ni eliminados desde la interfaz del sistema.

Consistencia en la asignación	Garantizar que los usuarios puedan ser asignados únicamente a roles válidos definidos en el sistema.
-------------------------------	--

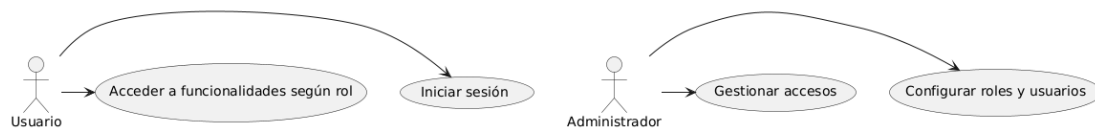
Nota. Elaboración propia.

3.2.3. Diagramas UML del Sprint

3.2.3.1. Diagrama de caso de uso

Figura 2

Diagrama de caso de uso gestión de autenticación

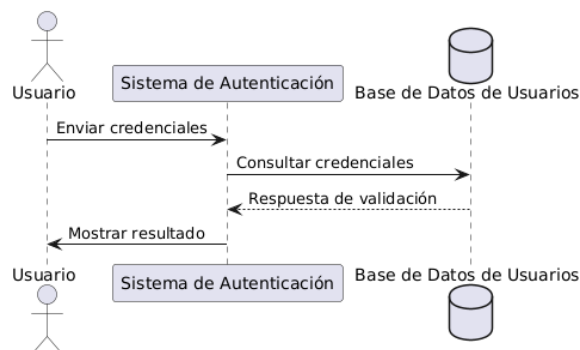


Nota. Elaboración propia.

3.2.3.2. Diagrama de secuencia

Figura 3

Diagrama de secuencia de gestión de autenticación



Nota. Elaboración propia.

3.2.3.3. Diagrama actividades

Figura 4

Diagrama actividades de gestión de autenticación



Nota. Elaboración propia.

3.2.4. Codificación

Figura 5

Código de autenticación

```

import ( Controller, Get, Post, Body, Patch, Param, Delete, UseGuards, SetMetadata ) from '@nestjs/common';
import ( UsuariosService ) from './usuarios.service';
import ( CreateUserDto, UpdateUsuarioDto, LoginUsuarioDto ) from './dto/index';
import ( Auth ) from './decorators/get-usuario.decorator';
import ( ValidRoles ) from './interfaces/valid-roles.interface';

@Controller( 'usuarios' )
export class UsuariosController {
  constructor( private readonly usuariosService: UsuariosService ) {}

  @Post()
  @Auth( ValidRoles.admin )
  create( @Body() createUsuarioDto: CreateUserDto ) {
    return this.usuariosService.create( createUsuarioDto );
  }

  @Post( 'login' )
  login( @Body() loginUsuarioDto: LoginUsuarioDto ) {
    return this.usuariosService.login( loginUsuarioDto );
  }

  @Get()
  @Auth( ValidRoles.admin )
  findAll() {
    return this.usuariosService.findAll();
  }

  @Get( ':id' )
  @Auth( ValidRoles.admin )
  findOne( @Param( 'id' ) id: string ) {
    return this.usuariosService.findOne( id );
  }
}

```

Nota. Elaboración propia.

Este código corresponde al módulo de autenticación del proyecto. Implementa un controlador (UsuariosController) que maneja rutas HTTP para la creación, autenticación y gestión de usuarios, delegando la lógica al servicio UsuariosService. Utiliza decoradores personalizados como Auth y la interfaz ValidRoles para proteger rutas y limitar el acceso según roles, como el de administrador. Los métodos incluyen funcionalidades como registrar usuarios (create), iniciar sesión (login), listar usuarios (findAll), obtener detalles de un usuario específico (findOne), actualizar usuarios (update) y eliminarlos (remove). Además, emplea Data Transfer Objects (DTOs) como CreateUsuarioDto, UpdateUsuarioDto y LoginUsuarioDto para validar los datos de entrada en cada operación.

3.2.5. Tareas del Sprint Backlog

Listado de las tareas seleccionadas del Product Backlog para ser implementadas durante este Sprint.

Tabla 11

Tareas del sprint backlog de gestión de autenticación

ID	TAREA	DESCRIPCIÓN	RESPONSABLE	ESTADO	TIEMPO ESTIMADO
1	Crear roles predefinidos	Implementar la creación de roles fijos mediante seed o código estático.	Daniel Santiago Soto Villamil	Terminado	3 días

2	Asignar roles a usuarios	Desarrollar la lógica para asignar roles a usuarios del sistema.	Daniel Santiago Soto Villamil	Terminado	4 días
3	Validar accesos por roles	Implementar validaciones que restrinjan accesos a funcionalidades según el rol asignado.	Daniel Santiago Soto Villamil	Terminado	4 días
4	Proteger la edición de roles	Asegurar que los roles no puedan ser creados, editados o eliminados desde la interfaz del sistema.	Daniel Santiago Soto Villamil	Terminado	3 días

Nota. Elaboración propia.

3.2.6. Desarrollo Iterativo y Validación

Descripción del progreso de las tareas desarrolladas y validación de las funcionalidades implementadas.

Tabla 12

Desarrollo iterativo y validación de gestión de autenticación

TAREA	PROGRESO	VALIDACIÓN REALIZADA
Crear roles predefinidos	Terminado	Validación de la creación correcta de roles fijos (Administrador, Supervisor, Encargado de Ventas, etc.).

Asignar roles a usuarios	Terminado	Comprobación de que los usuarios tienen asignado el rol adecuado en la base de datos.
Validar accesos por roles	Terminado	Verificación de que los usuarios solo acceden a las funcionalidades permitidas según su rol asignado.
Proteger la edición de roles	Terminado	Validación de que los roles no puedan ser creados, editados o eliminados desde la interfaz del sistema.

Nota. Elaboración propia.

3.2.7. Evaluación de Resultados

Análisis de los resultados obtenidos en el Sprint y su alineación con los objetivos del módulo.

3.2.7.1. Resultados de roles

Tabla 13

Resultados del módulo de autenticación

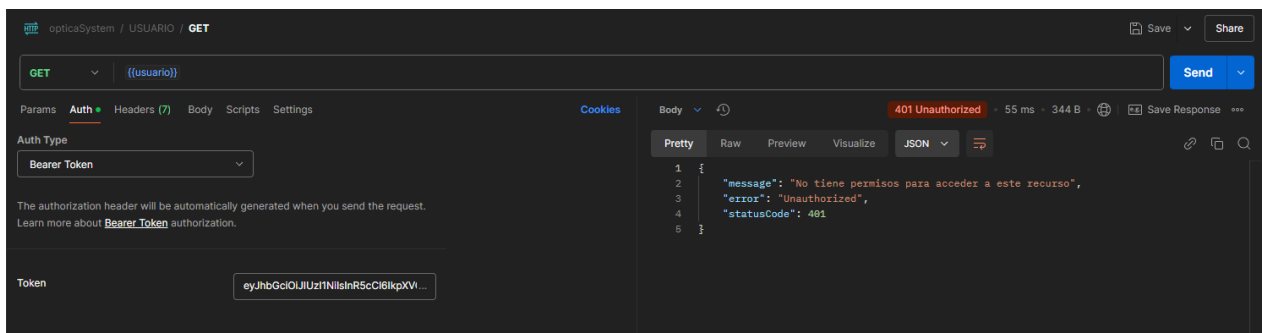
PRUEBA FUNCIONAL		REGISTRO DE ROLES
Descripción	<ul style="list-style-type: none"> Permite verificar la existencia de roles predefinidos. 	
Objetivos	<ul style="list-style-type: none"> Validar que los roles predefinidos existan en la base de datos. Verificar que los usuarios tienen roles asignados correctamente. Confirmar que los usuarios acceden únicamente a las funcionalidades permitidas según su rol. 	

Condiciones	<ul style="list-style-type: none"> Los roles deben ser creados inicialmente mediante seeds. Los roles deben estar correctamente asignados a los usuarios.
Resultado esperado	<ul style="list-style-type: none"> Los roles se validan y no presentan conflictos. Los usuarios acceden a las funcionalidades correspondientes a su rol asignado.
Resultado obtenido	<ul style="list-style-type: none"> El sistema valida correctamente los roles predefinidos y su asignación. El acceso se restringe adecuadamente, cumpliendo con las reglas de roles.

Nota. Elaboración propia.

Figura 6

Resultado de gestión de roles



Nota. Elaboración propia.

3.2.7.2. Resultados de usuarios

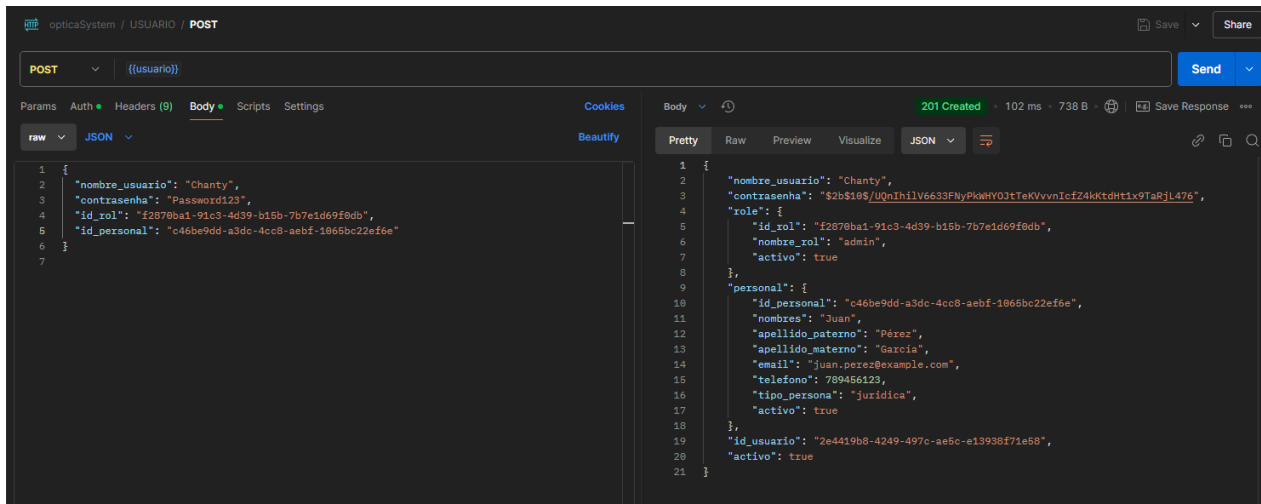
Se obtuvieron los siguientes resultados de las pruebas funcionales al módulo de usuarios.

Tabla 14

Resultados del módulo de usuarios

PRUEBA FUNCIONAL		REGISTRO DE USUARIOS
Descripción	<ul style="list-style-type: none"> Permite el registro de nuevos usuarios mediante un punto de acceso que asocia roles asignados. 	
Objetivos	<ul style="list-style-type: none"> Registrar un nuevo usuario enviando datos válidos a través de la API. Verificar que los datos del usuario se almacenen correctamente en la base de datos y estén asociados a un rol predefinido. 	
Condiciones	<ul style="list-style-type: none"> Interactuar con el punto de acceso de registro de usuarios con rol de administrador. La base de datos debe estar activa y funcional. 	
Resultado esperado	<ul style="list-style-type: none"> Al enviar una solicitud válida, el usuario se registra exitosamente en la base de datos, asignándosele un rol predefinido según los parámetros configurados. 	
Resultado obtenido	<ul style="list-style-type: none"> El sistema responde con un código de estado 201 Created y el usuario se registra correctamente en la base de datos, validando los datos únicos y asignando el rol correspondiente. 	

Nota. Elaboración propia.

Figura 7*Resultados de gestión de usuarios*

Nota. Elaboración propia.

3.3. SPRINT 2: GESTIÓN PERSONAL

3.3.1. Objetivos del Sprint

El objetivo es desarrollar y validar las funcionalidades esenciales del módulo de gestión de personal, consolidando las operaciones necesarias para administrar eficientemente los datos de los empleados y garantizar la seguridad en el sistema. Los objetivos específicos incluyen:

a) Registro del personal:

- Implementar la funcionalidad para que el administrador registre datos esenciales, como nombre, correo electrónico, tipo de empleado, y estado (activo/inactivo).

- Vincular automáticamente roles predefinidos al personal al momento de su registro, según el tipo de empleado (por ejemplo, administrador, encargado de ventas, etc.).

b) Actualización y consulta del personal:

- Proporcionar herramientas para que el administrador actualice la información del personal de manera segura, evitando duplicidad o errores en los datos.
- Permitir al administrador y supervisor consultar la información del personal activo e inactivo, con filtros para mejorar la búsqueda y el seguimiento.

c) Control de accesos mediante desactivación lógica:

- Incorporar la funcionalidad para activar o desactivar empleados sin eliminar sus registros de la base de datos, garantizando un control seguro y reversible.
- Asegurar que el personal desactivado no pueda acceder al sistema bajo ninguna circunstancia.

d) Validación de datos:

- La unicidad de los nombres de usuario asociados al personal.
- Verificar que la información del personal esté correctamente asociada en la base de datos, manteniendo una relación clara y sin conflictos

3.3.2. Definición de La Gestión De Personal

Identificación de las funcionalidades necesarias para el correcto funcionamiento del módulo de gestión de personal.

Tabla 15

Requisitos de la gestión del personal

REQUISITO	DESCRIPCIÓN
Registro de personal	Permitir al administrador registrar al personal con datos básicos como nombre, correo, teléfono, tipo de persona y estado activo/inactivo.
Edición de personal	Implementar la funcionalidad para que el administrador actualice los datos del personal registrado, asegurando la integridad de los datos.
Eliminación lógica	Garantizar que el personal pueda ser desactivado sin eliminar los registros del sistema, manteniendo un historial íntegro.
Validación de datos	Asegurar la unicidad de los correos electrónicos y otros datos críticos, evitando duplicados y manteniendo la consistencia en el sistema.
Consulta de personal	Habilitar una funcionalidad para listar y buscar personal registrado, con filtros por estado, tipo de persona y otros criterios relevantes.

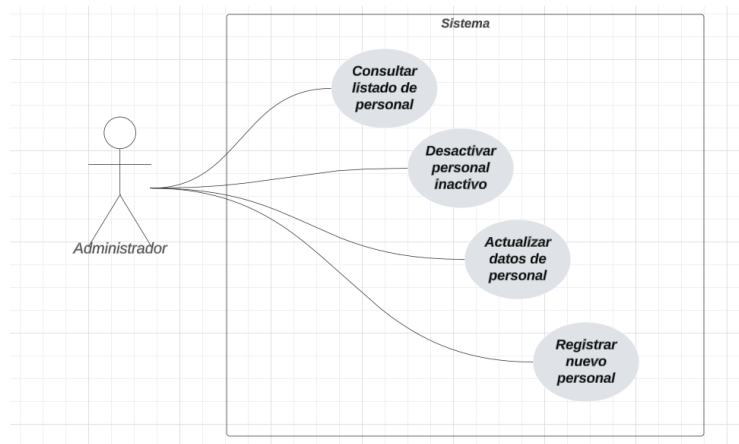
Nota. Elaboración propia.

3.3.3. Diagramas UML del Sprint

3.3.3.1. Diagrama de caso de uso

Figura 8

Diagrama de caso de uso de gestión de personal

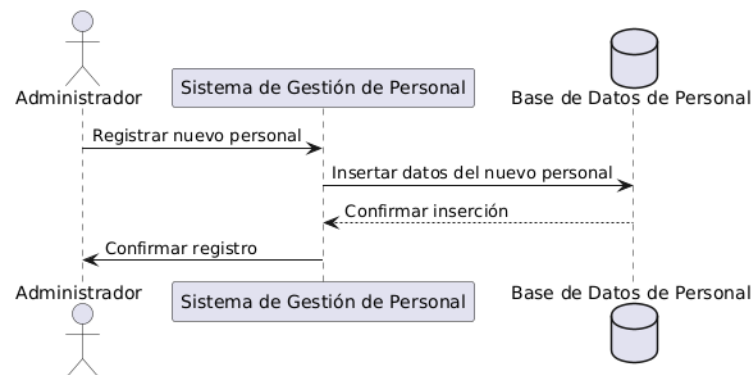


Nota. Elaboración propia.

3.3.3.2. Diagrama de secuencia

Figura 9

Diagrama de secuencia de gestión de personal

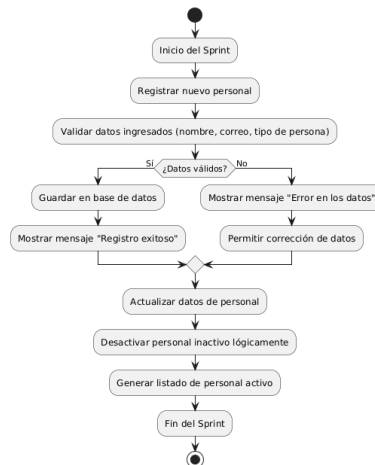


Nota. Elaboración propia.

3.3.3.3. Diagrama actividades

Figura 10

Diagrama actividades gestión de personal



Nota. Elaboración propia.

3.3.4. Codificación

Figura 11

Código de personal

```

@Controller("personal")
export class PersonalController {
  constructor(private readonly personalService: PersonalService) {}

  @Post()
  @Auth(ValidRoles.admin)
  create(@Body() createPersonalDto: CreatePersonalDto) {
    return this.personalService.create(createPersonalDto);
  }

  @Get("juridicos")
  // @Auth(ValidRoles.admin)
  findAllJuridicos(@Query() paginationDto: PaginationDto) {
    return this.personalService.findAllJuridicos(paginationDto);
  }

  @Get("naturales")
  // @Auth(ValidRoles.admin)
  findAllNaturales(@Query() paginationDto: PaginationDto) {
    return this.personalService.findAllNaturales(paginationDto);
  }

  @Get()
  // @Auth(ValidRoles.admin)
  findAll(@Query() paginationDto: PaginationDto) {
    return this.personalService.findAll(paginationDto);
  }
}

```

Nota. Elaboración propia.

Este código pertenece al módulo de gestión de personal. El controlador `PersonalController` está diseñado para manejar las rutas relacionadas con la creación, actualización, eliminación, y consulta de datos del personal. Los métodos delegan la lógica de negocio al servicio `PersonalService` y utilizan decoradores personalizados como `Auth` para proteger las rutas mediante roles específicos, como el rol de administrador (`ValidRoles.admin`). El método `create` permite registrar nuevos datos de personal utilizando un DTO llamado `CreatePersonalDto`. Los métodos `findAllJuridicos`, `findAllNaturales` y `findAll` permiten consultar diferentes tipos de personal (jurídico o natural) con soporte para paginación a través de un DTO de paginación. Por otro lado, `findOne` recupera un registro específico mediante su identificador, mientras que `update` y `remove` permiten actualizar y eliminar registros respectivamente. Este controlador garantiza una gestión eficiente y segura de los datos de personal, integrando validaciones mediante roles y el uso de DTOs para estructurar las solicitudes.

3.3.5. Tareas del Sprint Backlog

Listado de las tareas seleccionadas del Product Backlog para ser implementadas durante este Sprint.

Tabla 16*Tareas del sprint backlog de gestión del personal*

ID	TAREA	DESCRIPCIÓN	RESPONSABLE	ESTADO	TIEMPO ESTIMADO
1	Registrar personal	Implementar funcionalidad para registrar nuevos miembros del personal.	Daniel Santiago Soto Villamil	Terminado	5 días
2	Editar información del personal	Desarrollar lógica para actualizar los datos personal, asegurando validaciones y consistencia.	Daniel Santiago Soto Villamil	Terminado	4 días
3	Consultar lista del personal	Crear un módulo para consultar el listado de personal registrado, incluyendo filtros.	Daniel Santiago Soto Villamil	Terminado	3 días
4	Desactivar personal	Implementar funcionalidad para cambiar el estado del personal a inactivo sin eliminar su registro del sistema.	Daniel Santiago Soto Villamil	Terminado	

Nota. Elaboración propia.**3.3.6. Desarrollo Iterativo y Validación**

Descripción del progreso de las tareas desarrolladas y validación de las funcionalidades implementadas.

Tabla 17*Desarrollo iterativo y validación de gestión del personal*

TAREA	PROGRESO	VALIDACIÓN REALIZADA
Registrar personal	Completada	Validación de datos únicos (nombre, correo). Verificación de almacenamiento correcto en la base de datos.
Editar información del personal	Completada	Confirmación de actualizaciones en la base de datos con validaciones de integridad.
Consultar lista del personal	Completada	Validación de la correcta visualización del listado con filtros funcionales (estado activo/inactivo).
Desactivar personal	Completada	Comprobación de cambio de estado del personal a inactivo, garantizando la preservación de su registro en el sistema.

Nota. Elaboración propia.

3.3.7. Evaluación de Resultados

Análisis de los resultados obtenidos en el Sprint y su alineación con los objetivos del módulo.

3.3.7.1. Resultados de personal

Se obtuvieron los siguientes resultados de las pruebas funcionales realizadas al módulo de Personal.

Tabla 18

Resultados del módulo de personal

PRUEBA	REGISTRO DE PERSONAL
Descripción	<ul style="list-style-type: none"> Permite el registro, actualización y desactivación lógica del personal mediante un punto de acceso.
Objetivos	<ul style="list-style-type: none"> Registrar personal con datos válidos (nombres, apellidos, email, teléfono, tipo de persona, etc.). Actualizar información existente de personal, como datos de contacto o tipo de persona. Desactivar lógicamente personal que ya no esté en activo.
Condiciones	<ul style="list-style-type: none"> La base de datos debe estar activa y funcional. Los usuarios deben tener rol de administrador para gestionar el personal.
Resultado esperado	<ul style="list-style-type: none"> Los datos del personal se registran correctamente en la base de datos, asociados a su tipo de persona. Desactivación lógica sin eliminar registros del sistema.
Resultado obtenido	<ul style="list-style-type: none"> El sistema responde con un código de estado 201 Created para nuevos registros y 200 OK para actualizaciones o desactivaciones. Las desactivaciones lógicas restringen la interacción del personal desactivado en otros módulos del sistema.

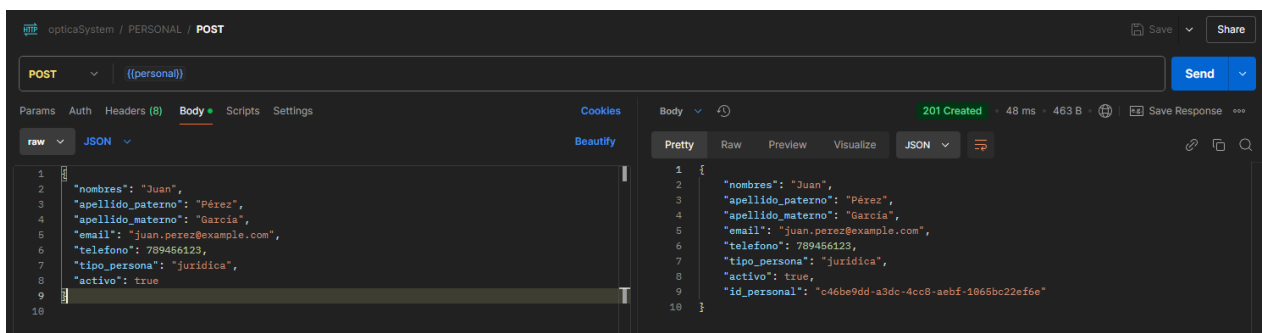
Nota. Elaboración propia.

Figura 12*Resultados de gestión de personal frontend*

The screenshot shows a web application interface with an orange header bar. On the right of the header, it says "Hola, admin" and "Cerrar Sesión". The main content area has a title "Datos del Personal" above a table. The table has columns: NOMBRES, APELLIDO PATERNO, APELLIDO MATERNO, EMAIL, TELÉFONO, TIPO DE PERSONA, ACTIVO, and ID PERSONAL. There are two rows of data: one for Juan Pérez García and one for María Gómez López.

NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	EMAIL	TELÉFONO	TIPO DE PERSONA	ACTIVO	ID PERSONAL
Juan	Pérez	García	juan.perez@example.com	709465123	Jurídica	Si	c64be9d9-3adc-4c08-aebf-1665be2ef6e
María	Gómez	López	maria.gomez@example.com	709465124	Natural	No	d73be9d9-3adc-4c08-aebf-1665be2ef7f

Nota. Elaboración propia.

Figura 13*Resultados de gestión personal backend*

The screenshot shows a REST client interface. The top bar indicates the endpoint is "opticaSystem / PERSONAL / POST" with a "Send" button. The "Body" tab is selected, showing a JSON payload. The response is a 201 Created status with a JSON body containing the same data as the request, plus an "id_personal" field.

```
POST {{personal}}

{
  "nombres": "Juan",
  "apellido_paterno": "Pérez",
  "apellido_materno": "García",
  "email": "juan.perez@example.com",
  "telefono": "789465123",
  "tipo_persona": "jurídica",
  "activo": true
}
```

```
201 Created 48 ms - 463 B
{
  "nombres": "Juan",
  "apellido_paterno": "Pérez",
  "apellido_materno": "García",
  "email": "juan.perez@example.com",
  "telefono": "789465123",
  "tipo_persona": "jurídica",
  "activo": true,
  "id_personal": "c64be9dd-a3dc-4c08-aebf-1665bc22ef6e"
}
```

Nota. Elaboración propia.

3.4. SPRINT 3: GESTIÓN DE PROVEEDORES Y PRODUCTOS

3.4.1. Objetivos del Sprint

El objetivo es desarrollar y validar las funcionalidades del módulo de gestión de proveedores y productos. Las metas específicas incluyen:

a) Registro de Proveedores:

- Implementar un sistema para registrar proveedores con datos básicos (nombre, número de contacto, empresa, etc.).

b) Actualización y Desactivación de Proveedores:

- Habilitar la edición de datos existentes y la desactivación lógica de proveedores no activos.

c) Registro de Productos:

- Permitir registrar productos con información relevante (nombre, descripción, stock, precios de compra y venta).

d) Relación entre Productos y Proveedores:

- Configurar la asignación de proveedores específicos a productos registrados, garantizando la trazabilidad.

e) Gestión del Promedio Ponderado:

- Implementar el cálculo del Precio Promedio Ponderado (PMP) para el manejo de inventarios, reflejando entradas y salidas de productos con sus respectivos costos unitarios y totales.

f) Validación de Datos:

- Asegurar que los datos ingresados sean únicos y consistentes con los requerimientos funcionales establecidos.

3.4.2. Definición de la Gestión de Proveedores y Productos

Identificación de las funcionalidades necesarias para el correcto funcionamiento del módulo.

Tabla 19

Requisitos de la gestión de proveedores y productos

REQUISITO	DESCRIPCIÓN
Registro de proveedores	Permitir registrar datos básicos de proveedores como nombre, contacto y empresa.
Actualización de proveedores	Modificar información existente de los proveedores registrados.
Desactivación lógica de proveedores	Marcar como inactivos a proveedores que no estén en uso.
Registro de productos	Registrar productos con detalles como nombre, descripción, precios y stock.
Actualización de productos	Editar información de productos ya registrados.
Relación productos-proveedores	Permitir asociar productos a uno o varios proveedores.
Gestión del Promedio Ponderado	Calcular el Precio Promedio Ponderado (PMP) para el manejo de inventarios, reflejando entradas y salidas con costos asociados.

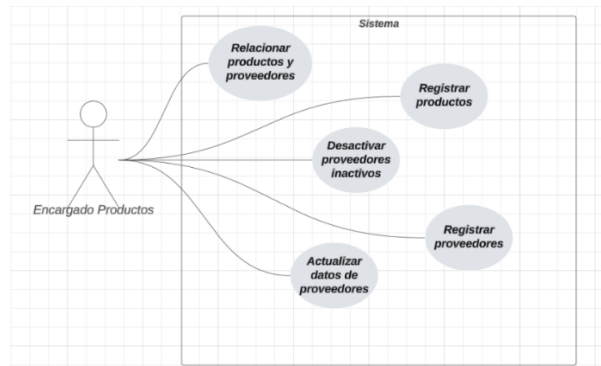
Nota. Elaboración propia.

3.4.3. Diagramas UML del Sprint

3.4.3.1. Diagrama de caso de uso

Figura 14

Diagrama de caso de uso de gestión proveedores y producto

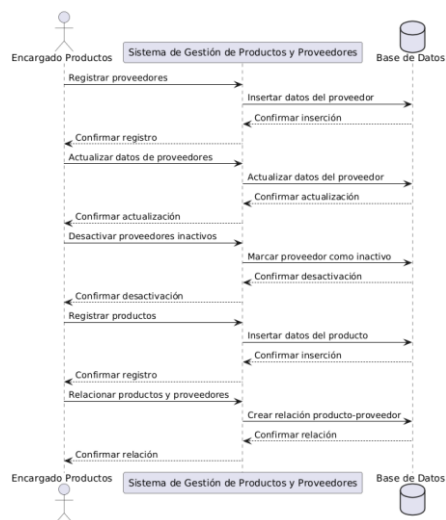


Nota. Elaboración propia.

3.4.3.2. Diagrama de secuencia

Figura 15

Diagrama de secuencia de gestión de proveedores y productos

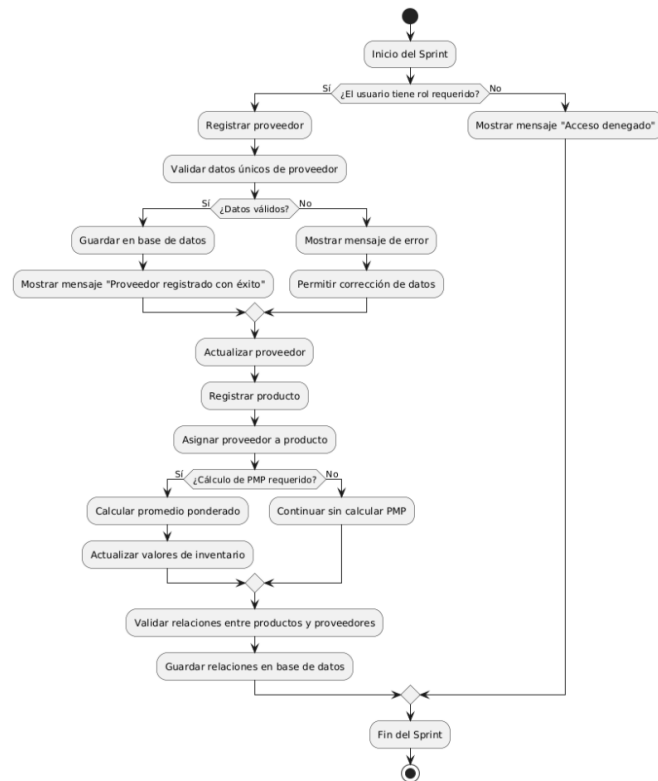


Nota. Elaboración propia.

3.4.3.3. Diagrama actividades

Figura 16

Diagrama actividades de gestión productos y proveedores



Nota. Elaboración propia.

3.4.4. Codificación

Figura 17

Código de proveedores y productos

```
@Controller("productos")
export class ProductosController {
  constructor(private readonly productosService: ProductosService) {}

  @Post()
  @Auth(ValidRoles.encargadoProductos)
  create(@Body() createProductoDto: CreateProductoDto) {
    return this.productosService.create(createProductoDto);
  }

  @Get('admin')
  @Auth(ValidRoles.encargadoProductos)
  findAllAdmin(
    @Query() paginationDto: PaginationDto,
    @Query() queryGetDto: QueryGetDto,
    @Query() queryProducto: QueryProductoDto
  ) {
    return this.productosService.findAllAdmin(paginationDto, queryGetDto, queryProducto);
  }

  @Get()
  @Auth(ValidRoles.encargadoProductos)
  findAll(
    @Query() paginationDto: PaginationDto,
    @Query() queryGetDto: QueryGetDto,
    @Query() queryProducto: QueryProductoDto
  ) {
    return this.productosService.findAll(paginationDto, queryGetDto, queryProducto);
  }

  @Get('search')
  @Auth(ValidRoles.encargadoProductos)
  searchProducto(
    @Query('search') search: string,
    @Query() paginationDto: PaginationDto
  ) {
    return this.productosService.searchProducto(search, paginationDto);
  }

  @Get(':id')
  @Auth(ValidRoles.encargadoProductos)
  findOne(@Param('id', ParseUUIDPipe) id: string) {
    return this.productosService.findOne(id);
  }

  @Patch(':id')
  @Auth(ValidRoles.encargadoProductos)
  update(@Param('id', ParseUUIDPipe) id: string, @Body() updateProductoDto: UpdateProductoDto) {
    return this.productosService.update(id, updateProductoDto);
  }

  @Delete(':id')
  @Auth(ValidRoles.encargadoProductos)
  remove(@Param('id', ParseUUIDPipe) id: string) {
    return this.productosService.remove(id);
  }
}
```

Nota. Elaboración propia.

Este código corresponde al módulo de gestión de productos y proveedores. El controlador ProductosController gestiona las operaciones relacionadas con los productos del sistema, delegando la lógica de negocio al servicio ProductosService. Utiliza decoradores personalizados como Auth para proteger las rutas según roles, como encargadoProductos. Las funcionalidades incluyen el registro de nuevos

productos mediante el método `create`, que utiliza el DTO `CreateProductoDto` para validar los datos de entrada.

El controlador soporta consultas avanzadas con métodos como `findAllAdmin` y `findAll`, que permiten listar productos con opciones de paginación y filtros avanzados mediante DTOs como `PaginationDto`, `QueryGetDto` y `QueryProductoDto`. Además, el método `searchProducto` implementa una búsqueda específica basada en cadenas de texto, también compatible con paginación.

Para gestionar productos individuales, el método `findOne` recupera un producto específico mediante su identificador (`id`), mientras que `update` permite actualizar sus detalles utilizando el DTO `UpdateProductoDto`. Finalmente, el método `remove` realiza la eliminación lógica de un producto, garantizando su trazabilidad en el sistema. Este controlador está diseñado para proporcionar una gestión segura y eficiente de los productos, asegurando que todas las operaciones estén protegidas mediante roles y validaciones robustas.

3.4.5. Tareas del Sprint Backlog

Tabla 20

Tareas del sprint backlog de gestión de proveedores y productos

ID	TAREA	DESCRIPCIÓN	RESPONSABLE	ESTADO	TIEMPO ESTIMADO
1	Registrar nuevos proveedores	Crear formulario y lógica para registrar proveedores con datos básicos.	Daniel Santiago Soto Villamil	Terminado	5 días
2	Actualizar datos de proveedores	Implementar lógica para editar información de proveedores ya registrados.	Daniel Santiago Soto Villamil	Terminado	4 días
3	Registrar nuevos productos	Crear formulario y lógica para registrar productos con detalles completos.	Daniel Santiago Soto Villamil	Terminado	6 días
4	Relacionar productos-proveedores	Configurar la lógica para asignar proveedores específicos a productos registrados.	Daniel Santiago Soto Villamil	Terminado	5 días
5	Implementar cálculo PMP	Desarrollar la funcionalidad para calcular el Precio Promedio Ponderado en la gestión de stock.	Daniel Santiago Soto Villamil	Terminado	6 días

Nota. Elaboración propia.

3.4.6. Desarrollo Iterativo y Validación

Descripción del progreso y validación de las tareas desarrolladas:

Tabla 21

Desarrollo iterativo y validación de gestión de proveedores y productos

TAREA	PROGRESO	VALIDACIÓN REALIZADA
Registrar nuevos proveedores	Terminado	Validación de datos únicos y almacenamiento correcto en la base de datos.
Actualizar datos de proveedores	Terminado	Confirmar que las actualizaciones no generen duplicados ni errores en los registros.
Registrar nuevos productos	Terminado	Validación de datos ingresados y registro correcto en la base de datos.
Relacionar productos-proveedores	Terminado	Comprobar que la asignación de productos a proveedores se realiza sin conflictos.
Implementar cálculo PMP	Terminado	Validación del cálculo correcto del Promedio Ponderado basado en entradas y salidas de stock.

Nota. Elaboración propia.

3.4.7. Evaluación de Resultados

Análisis de los resultados obtenidos en el Sprint y su alineación con los objetivos del módulo.

3.4.7.1. Resultados de proveedores

Tabla 22

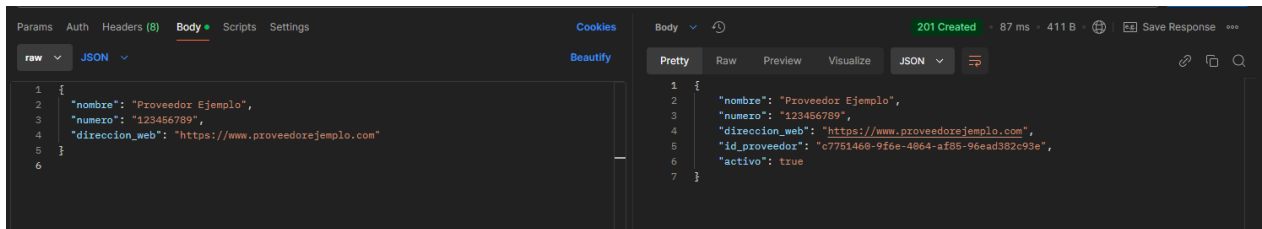
Resultados del módulo de proveedores

PRUEBA FUNCIONAL	REGISTRO DE PROVEEDORES
Descripción	<ul style="list-style-type: none"> • Permite el registro, actualización y desactivación lógica de proveedores mediante un punto de acceso.
Objetivos	<ul style="list-style-type: none"> • Registrar proveedores con datos válidos (nombre, contacto, empresa). • Actualizar información existente de proveedores, como datos de contacto o empresa. • Desactivar lógicamente proveedores que ya no suministran productos.
Condiciones	<ul style="list-style-type: none"> • La base de datos debe estar activa y funcional. • Los usuarios deben tener rol de administrador para gestionar el personal.
Resultado esperado	<ul style="list-style-type: none"> • Los datos de los proveedores se registran correctamente en la base de datos, asociados a los productos correspondientes. • Actualización de datos válida sin duplicados. • Desactivación lógica sin eliminar registros del sistema.
Resultado obtenido	<ul style="list-style-type: none"> • El sistema responde con un código de estado 201 Created para nuevos registros y 200 OK para actualizaciones o desactivaciones. • Los datos únicos del proveedor se validan correctamente. • Las desactivaciones lógicas restringen la interacción de los proveedores desactivados en otros módulos del sistema.

Nota. Elaboración propia.

Figura 18

Resultados gestión de proveedores backend



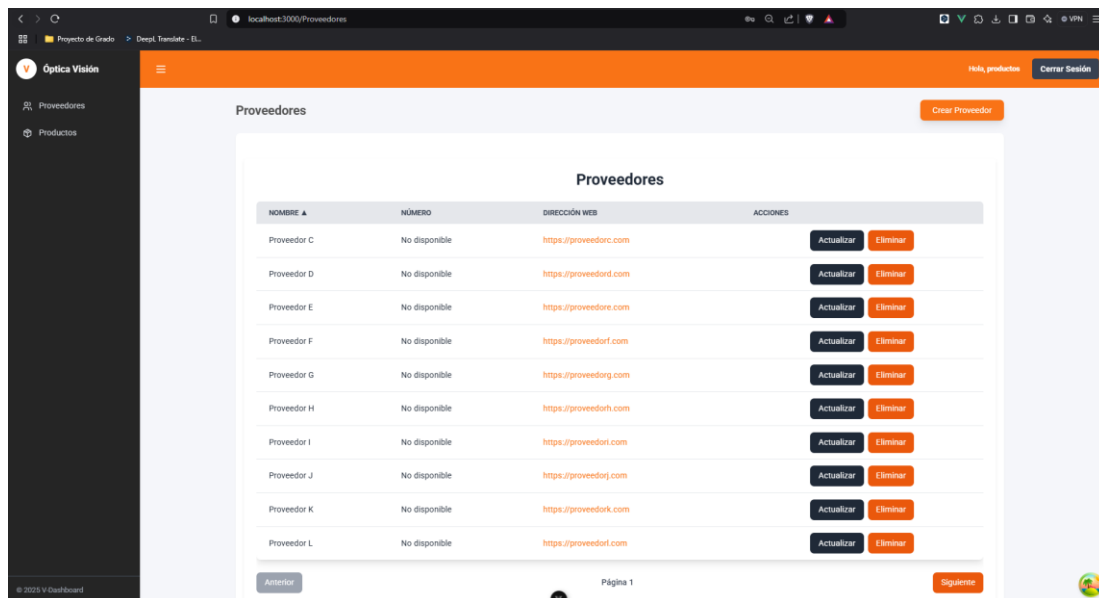
```
1 {
2   "nombre": "Proveedor Ejemplo",
3   "numero": "123456789",
4   "direccion_web": "https://www.proveedorejemplo.com"
5 }
6
```

```
1 {
2   "nombre": "Proveedor Ejemplo",
3   "numero": "123456789",
4   "direccion_web": "https://www.proveedorejemplo.com",
5   "id_proveedor": "c7751468-9f6e-4864-af85-96ead382c93e",
6   "activo": true
7 }
```

Nota. Elaboración propia.

Figura 19

Resultados de gestión de proveedores frontend



NOMBRE	NÚMERO	DIRECCIÓN WEB	ACCIONES
Proveedor C	No disponible	https://proveedorc.com	Actualizar Eliminar
Proveedor D	No disponible	https://proveedord.com	Actualizar Eliminar
Proveedor E	No disponible	https://proveedore.com	Actualizar Eliminar
Proveedor F	No disponible	https://proveedorf.com	Actualizar Eliminar
Proveedor G	No disponible	https://proveedorg.com	Actualizar Eliminar
Proveedor H	No disponible	https://proveedorh.com	Actualizar Eliminar
Proveedor I	No disponible	https://proveedorl.com	Actualizar Eliminar
Proveedor J	No disponible	https://proveedorj.com	Actualizar Eliminar
Proveedor K	No disponible	https://proveedork.com	Actualizar Eliminar
Proveedor L	No disponible	https://proveedorl.com	Actualizar Eliminar

Nota. Elaboración propia.

3.4.7.2. Resultados de productos

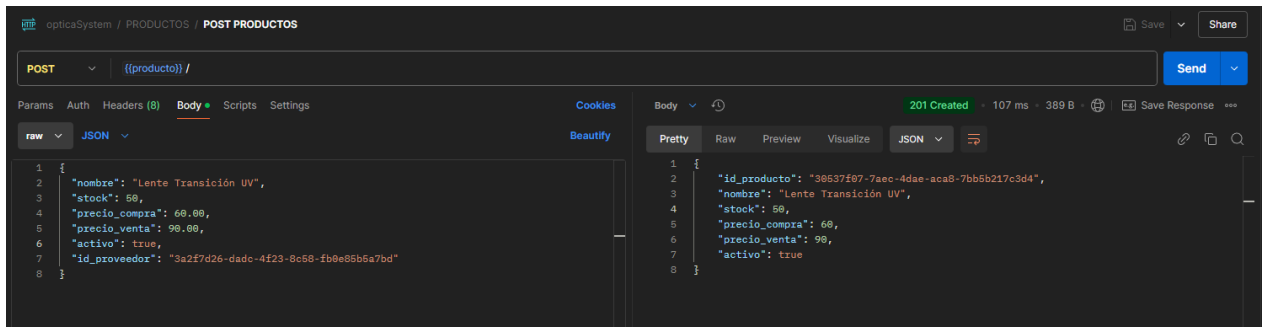
Tabla 23

Resultados del módulo de productos

PRUEBA FUNCIONAL	REGISTRO DE PRODUCTOS
Descripción	<ul style="list-style-type: none"> • Permite el registro, actualización y desactivación lógica de productos mediante un punto de acceso.
Objetivos	<ul style="list-style-type: none"> • Registrar productos con datos válidos (nombre, stock, precio de compra y venta). • Actualizar información existente de productos, como precios y stock. • Desactivar lógicamente productos que ya no estén disponibles.
Condiciones	<ul style="list-style-type: none"> • La base de datos debe estar activa y funcional. • Los usuarios deben tener rol de administrador para gestionar los productos.
Resultado esperado	<ul style="list-style-type: none"> • Los datos de los productos se registran correctamente en la base de datos, asociados a proveedores cuando corresponda. • Actualización de datos válida sin duplicados. • Desactivación lógica sin eliminar registros del sistema.
Resultado obtenido	<ul style="list-style-type: none"> • El sistema responde con un código de estado 201 Created para nuevos registros y 200 OK para actualizaciones o desactivaciones. • Los datos únicos de los productos se validan correctamente. • Las desactivaciones lógicas restringen la interacción de los productos desactivados en otros módulos del sistema.

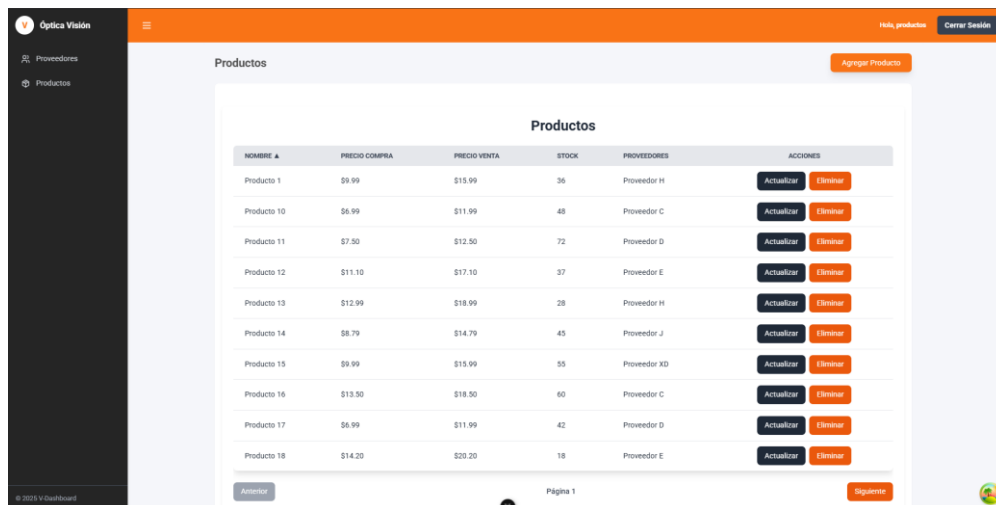
Nota. Elaboración propia.

Figura 20

Resultados de gestión de productos backend

Nota. Elaboración propia.

Figura 21

Resultados de gestión de productos frontend

Nota. Elaboración propia.

3.4.7.3. Resultados de promedio ponderado

Tabla 24

Resultados de promedio ponderado

PRUEBA	REGISTRO DE PROMEDIO PONDERADO
Descripción	<ul style="list-style-type: none"> • Permite calcular automáticamente el valor unitario promedio de los productos registrados en inventario.
Objetivos	<ul style="list-style-type: none"> • Garantizar un cálculo preciso del promedio ponderado considerando entradas y salidas de stock. • Actualizar el costo unitario promedio tras cada movimiento (compra o venta).
Condiciones	<ul style="list-style-type: none"> • La base de datos debe estar activa y funcional. • Los usuarios deben tener rol de administrador para gestionar los productos. • La base de datos debe contener registros de productos con movimientos de stock asociados.
Resultado esperado	<ul style="list-style-type: none"> • El cálculo del promedio ponderado refleja con precisión los costos actualizados del inventario. • Los datos se actualizan automáticamente tras cada operación de entrada o salida.
Resultado obtenido	<ul style="list-style-type: none"> • El cálculo del promedio ponderado se valida correctamente mediante pruebas funcionales. • Los costos actualizados se reflejan en tiempo real en la base de datos.

Nota. Elaboración propia.

Figura 22*Resultados de gestión de promedio ponderado*

Método de Manejo de Inventario Precio Promedio Ponderado										
FECHA	CONCEPTO	CANTIDAD	VALOR UNIDAD	VALOR TOTAL	CANTIDAD	VALOR UNIDAD	VALOR TOTAL	CANTIDAD	VALOR UNIDAD	VALOR TOTAL
Marzo	Compra	150	\$1000.00	\$150000.00	150	\$1000.00	\$150000.00	150	\$1000.00	\$150000.00
Junio	Compra	50	\$1200.00	\$60000.00	50	\$1200.00	\$60000.00	50	\$1200.00	\$60000.00
Agosto	Venta	80	\$1050.00	\$84000.00	80	\$1050.00	\$84000.00	80	\$1050.00	\$84000.00
Diciembre	Venta	45	\$1050.00	\$47250.00	45	\$1050.00	\$47250.00	45	\$1050.00	\$47250.00

Nota. Elaboración propia.

3.5. SPRINT 4: SPRINT DE GESTIÓN DE TRABAJOS

3.5.1. Objetivos del Sprint

El objetivo es desarrollar y validar las funcionalidades esenciales del módulo de trabajos, garantizando un proceso eficiente de gestión y seguimiento de trabajos. Las metas específicas incluyen:

a) Registro de trabajos ópticos:

- Implementar funcionalidades para registrar trabajos con detalles técnicos específicos.

b) Gestión de colores:

- Permitir la administración y asociación de colores a los trabajos registrados.

c) Gestión de tratamientos:

- Desarrollar un sistema que permita gestionar los tratamientos ópticos disponibles y asociarlos a trabajos.

d) Gestión de parámetros técnicos:

- Implementar lógica para capturar detalles técnicos (esfera, cilindro, prisma, etc.) en cada trabajo.

e) Actualización de trabajos:

Permitir modificar información existente de trabajos registrados.

f) Seguimiento de estados de trabajos:

- Implementar funcionalidades para cambiar y visualizar el estado de los trabajos (Pendiente y Entregado).

3.5.2. Definición de la Gestión de Trabajos**Tabla 25**

Requisitos de la gestión de trabajos

REQUISITO	DESCRIPCIÓN
Registro de trabajos	Permitir registrar trabajos ópticos con detalles técnicos y asignación al personal encargado del trabajo.
Gestión de detalles técnicos	Capturar parámetros técnicos como esfera, cilindro, eje, prisma, base y altura de los trabajos ópticos.
Gestión de colores y tratamientos	Permitir registrar y asociar colores y tratamientos a trabajos específicos, según requerimiento del cliente.
Actualización de trabajos	Editar información existente sobre un trabajo, como parámetros técnicos, costos y fechas.

Seguimiento de estado de trabajos	Administrar y actualizar el estado del trabajo (pendiente, en proceso, finalizado o entregado).
Relación con personal	Asociar trabajos registrados al personal encargado para identificar responsables del proceso.
Relación con productos	Relacionar productos registrados en el sistema a cada trabajo óptico, como insumos o materiales utilizados.

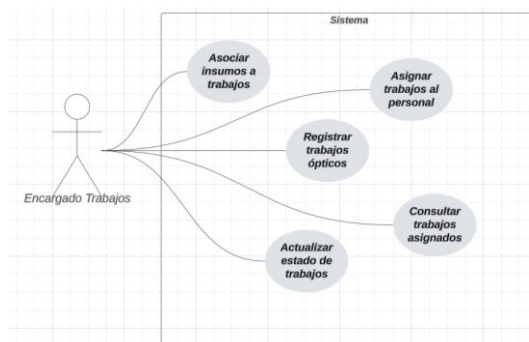
Nota. Elaboración propia.

3.5.3. Diagrama UML del Sprint

3.5.3.1. Diagrama de caso de uso

Figura 23

Diagrama de caso de uso de gestión de trabajos

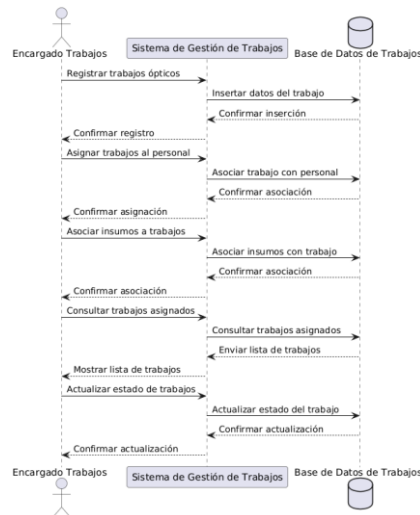


Nota. Elaboración propia.

3.5.3.2. Diagrama de secuencia

Figura 24

Diagrama de secuencia de gestión de trabajos

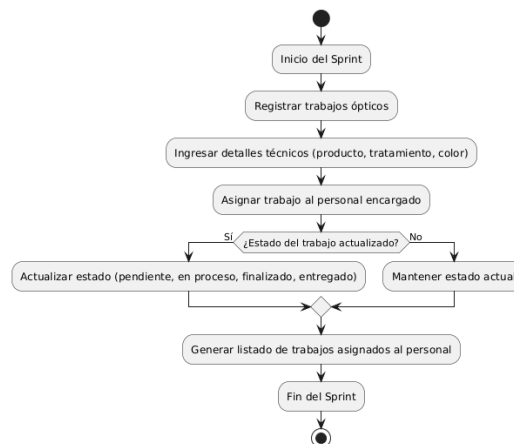


Nota. Elaboración propia.

3.5.3.3. Diagrama actividades

Figura 25

Diagrama actividades de gestión de trabajos



Nota. Elaboración propia.

3.5.4. Codificación

Figura 26

Código de trabajos

```
@Controller('trabajos')
export class TrabajosController {
  constructor(private readonly trabajosService: TrabajosService) {}

  @Post()
  @Auth(ValidRoles.encargadoTrabajos)
  create(@Body() createTrabajoDto: CreateTrabajoDto) {
    return this.trabajosService.create(createTrabajoDto);
  }

  @Get()
  @Auth(ValidRoles.encargadoTrabajos)
  findAll(
    @Query() paginationDto: PaginationDto,
  ) {
    return this.trabajosService.findAll(paginationDto);
  }

  @Get(':id')
  @Auth(ValidRoles.encargadoTrabajos)
  findOne(@Param('id') id: string) {
    return this.trabajosService.findOne(id);
  }

  @Patch(':id')
  @Auth(ValidRoles.encargadoTrabajos)
  update(@Param('id') id: string, @Body() updateTrabajoDto: UpdateTrabajoDto) {
    return this.trabajosService.update(id, updateTrabajoDto);
  }

  @Delete(':id')
  @Auth(ValidRoles.encargadoTrabajos)
  remove(@Param('id') id: string) {
    return this.trabajosService.remove(id);
  }
}
```

Nota. Elaboración propia.

Este código pertenece al módulo de gestión de trabajos. El controlador `TrabajosController` maneja las rutas relacionadas con los trabajos ópticos, delegando la lógica de negocio al servicio `TrabajosService`. Este módulo está protegido por el decorador `Auth`, que asegura que solo usuarios con el rol `encargadoTrabajos` puedan acceder a las funcionalidades.

El método `create` permite registrar nuevos trabajos ópticos, utilizando el DTO `CreateTrabajoDto` para validar los datos de entrada. El método `findAll` lista todos los trabajos registrados con soporte para paginación mediante el DTO `PaginationDto`. Para recuperar información específica de un trabajo, el método `findOne` busca el registro

basado en el identificador (id) proporcionado. Asimismo, el método update permite modificar los detalles de un trabajo existente utilizando el DTO UpdateTrabajoDto, mientras que remove realiza la eliminación lógica del trabajo, manteniendo su trazabilidad en el sistema.

Este controlador garantiza una gestión segura y eficiente de los trabajos ópticos al integrar validaciones, roles y paginación, ofreciendo una estructura clara y escalable para el manejo de datos.

3.5.5. Tareas del Sprint Backlog

Listado de las tareas seleccionadas del Product Backlog para ser implementadas durante este Sprint.

Tabla 26

Tareas del sprint backlog de gestión de trabajos

ID	TAREA	DESCRIPCIÓN	RESPONSABLE	ESTADO	TIEMPO ESTIMADO
1	Registrar trabajos ópticos	Crear formulario y lógica para registrar trabajos ópticos con detalles técnicos completos.	Daniel Santiago Soto Villamil	Terminado	6 días
2	Asignar trabajos al personal	Implementar funcionalidad para asignar trabajos al	Daniel Santiago Soto Villamil	Terminado	4 días

		personal responsable, permitiendo trazabilidad.			
3	Capturar detalles técnicos	Capturar y validar parámetros técnicos como esfera, cilindro, prisma, eje y altura.	Daniel Santiago Soto Villamil	Terminado	5 días
4	Actualización de trabajos	Permitir la edición de detalles técnicos y actualización del estado del trabajo.	Daniel Santiago Soto Villamil	Terminado	5 días
5	Seguimiento del estado de trabajos	Crear funcionalidad para visualizar, filtrar y actualizar el estado de trabajos asignados.	Daniel Santiago Soto Villamil	Terminado	4 días
6	Asociar productos y tratamientos	Implementar relación de productos, colores y tratamientos registrados a cada trabajo óptico.	Daniel Santiago Soto Villamil	Terminado	6 días

Nota. Elaboración propia.

3.5.6. Desarrollo Iterativo y Validación

Descripción del progreso de las tareas desarrolladas y validación de las funcionalidades implementadas.

Tabla 27

Desarrollo iterativo y validación de gestión de trabajos

TAREA	PROGRESO	VALIDACIÓN REALIZADA
Registrar trabajos ópticos	Terminado	Validar almacenamiento de parámetros técnicos y asignación al personal.
Asociar trabajos al personal	Terminado	Verificar la asociación del trabajo con un responsable registrado.
Gestión de detalles técnicos	Terminado	Validar la captura de datos técnicos (cilindro, esfera, etc.).
Actualización de trabajos	Terminado	Confirmar la edición de detalles y actualización del estado del trabajo.
Seguimiento del estado de trabajos	Terminado	Verificar cambios en el estado del trabajo y su visualización.
Asociar productos y tratamientos	Terminado	Validar la relación entre productos registrados y los trabajos ópticos.

Nota. Elaboración propia.

3.5.7. Evaluación de Resultados

Análisis de los resultados obtenidos en el Sprint y su alineación con los objetivos del módulo.

3.5.7.1. Resultados de trabajos.

Tabla 28

Resultados del módulo de trabajos

PRUEBA FUNCIONAL	REGISTRO DE TRABAJOS
Descripción	<ul style="list-style-type: none"> • Permite el registro, actualización y desactivación lógica de trabajos ópticos mediante un punto de acceso.
Objetivos	<ul style="list-style-type: none"> • Registrar trabajos con detalles técnicos válidos (esfera, cilindro, eje, altura, prisma). • Actualizar información existente de trabajos, como parámetros técnicos, costos o estados. • Desactivar lógicamente trabajos finalizados o no activos, evitando eliminaciones permanentes.
Condiciones	<ul style="list-style-type: none"> • La base de datos debe estar activa y funcional. • Los usuarios deben tener rol de administrador para gestionar los trabajos. • Productos, colores y tratamientos deben estar registrados previamente en el sistema.
Resultado esperado	<ul style="list-style-type: none"> • Los trabajos se registran correctamente con detalles técnicos y asignación de personal responsable. • Actualización de datos válida, reflejando los nuevos parámetros y costos sin duplicados. • Desactivación lógica que restringe la edición y visualización de trabajos inactivos en otros módulos.

Resultado obtenido

- El sistema responde con un código de estado 201 Created para nuevos registros y 200 OK para actualizaciones o desactivaciones.
- Los detalles técnicos del trabajo (esfera, cilindro, prisma, etc.) se almacenan correctamente.
- La desactivación lógica funciona correctamente, restringiendo trabajos inactivos en el sistema.

Nota. Elaboración propia.

Figura 27

Resultados de gestión de trabajos frontend

Detalles de Trabajo						
FECHA DE SALIDA	COSTO	DETALLES DE TRABAJO	PERSONAL	TRATAMIENTO	PRODUCTO	COLOR
2025-01-05	\$150.56	Ver Detalles	Ramiro Cuellar	Ninguno	Lentes UV	Color

Nota. Elaboración propia.

Figura 28

Resultados de gestión de trabajos backend

```

POST /api/trabajo/
Content-Type: application/json
{
  "fecha_salida": "2025-01-05",
  "costo": 150.56,
  "id_personal": "2025-01-05-01",
  "detalle_trabajo": {
    "esfera": 0,
    "cilindro": 1,
    "prisma": 0,
    "lentes_uv": 1,
    "color": "Color"
  },
  "personal": {
    "id_personal": "2025-01-05-01",
    "nombre": "Ramiro Cuellar",
    "email": "ramiro.cuellar@empresa.com",
    "telefono": "011-123456789",
    "activo": true
  }
}
201 Created
{
  "id_trabajo": "2025-01-05-01",
  "fecha_salida": "2025-01-05",
  "costo": 150.56,
  "id_personal": "2025-01-05-01",
  "detalle_trabajo": {
    "esfera": 0,
    "cilindro": 1,
    "prisma": 0,
    "lentes_uv": 1,
    "color": "Color"
  },
  "personal": {
    "id_personal": "2025-01-05-01",
    "nombre": "Ramiro Cuellar",
    "email": "ramiro.cuellar@empresa.com",
    "telefono": "011-123456789",
    "activo": true
  }
}

```

Nota. Elaboración propia.

3.6. SPRINT 5: GESTIÓN DE VENTAS

3.6.1. Objetivos del Sprint

El objetivo es desarrollar y validar las funcionalidades esenciales del módulo de ventas, garantizando un proceso eficiente de registro, gestión y seguimiento de ventas. Las metas específicas incluyen:

a) Registro de ventas:

- Implementar la funcionalidad para registrar ventas con datos esenciales como fecha, monto total y usuario responsable.

b) Detalle de ventas:

- Permitir capturar los detalles de productos y servicios vendidos, incluyendo cantidad, precio unitario y total parcial.

c) Visualización de historial:

- Desarrollar una funcionalidad para visualizar y filtrar el historial de ventas, permitiendo su seguimiento eficiente.

d) Cálculo automático del total:

- Asegurar que el monto total de las ventas se calcule automáticamente según los detalles ingresados.

e) Validación de datos:

- Garantizar la consistencia de la información registrada, evitando duplicados o registros incompletos.

3.6.2. Definición de la Gestión de Ventas

Tabla 29

Requisitos de la gestión de ventas

REQUISITO	DESCRIPCIÓN
Registro de ventas	Permitir registrar transacciones de ventas con información básica (fecha, monto total, usuario responsable).
Registro de detalle de ventas	Capturar detalles de los productos y servicios vendidos, como cantidad, precio unitario y total parcial.
Actualización de ventas	Permitir modificar información de ventas, como ajustes en detalles o montos registrados.
Visualización de ventas	Mostrar un historial detallado de ventas registradas, con filtros por fecha y usuario.
Cálculo del monto total	Calcular automáticamente el monto total de la venta con base en los detalles proporcionados.
Asociación de ventas	Relacionar cada venta registrada con el usuario responsable y productos involucrados.

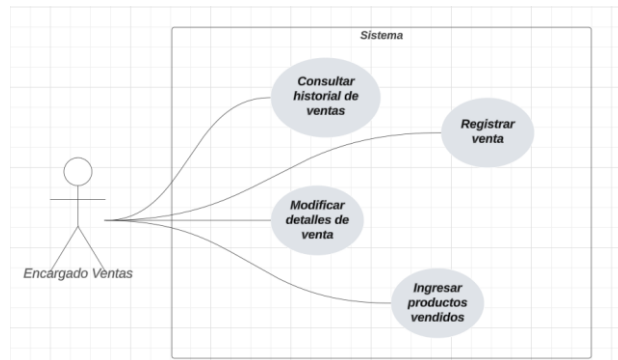
Nota. Elaboración propia.

3.6.3. Diagramas UML del Sprint

3.6.3.1. Diagrama de caso de uso

Figura 29

Diagrama de caso de uso de gestión de ventas

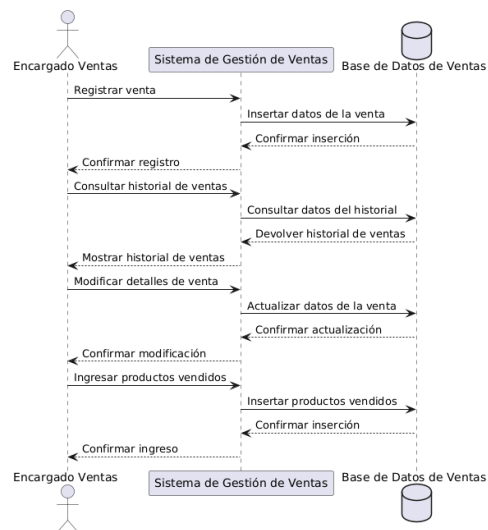


Nota. Elaboración propia.

3.6.3.2. Diagrama de secuencia

Figura 30

Diagrama secuencia de gestión de ventas

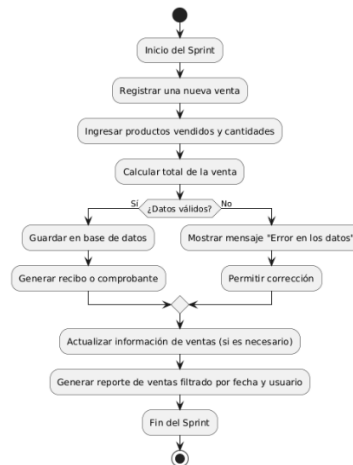


Nota. Elaboración propia.

3.6.3.3. Diagrama actividades

Figura 31

Diagrama actividades de gestión de ventas



Nota. Elaboración propia.

3.6.4. Codificación

Figura 32

Código de ventas

```

@Controller(Ventas)
export class VentasController {
  constructor(private readonly ventasService: VentasService) {}

  @Post()
  @Auth(ValidRoles.encargadoVentas)
  create(
    @Body() createVentaDto: CreateVentaDto,
    @Req() req: any,
  ) {
    return this.ventasService.create(createVentaDto, req.user);
  }

  @Get()
  @Auth(ValidRoles.encargadoVentas)
  findAll() {
    return this.ventasService.findAll();
  }

  @Get("{id}")
  @Auth(ValidRoles.encargadoVentas)
  findOne(@Param("id") id: string) {
    return this.ventasService.findOne(id);
  }

  @Patch("{id}")
  @Auth(ValidRoles.encargadoVentas)
  update(@Param("id") id: string, @Body() updateVentaDto: UpdateVentaDto) {
    return this.ventasService.update(id, updateVentaDto);
  }
}

```

Nota. Elaboración propia.

3.6.5. Tareas del Sprint Backlog

Listado de las tareas seleccionadas del Product Backlog para ser implementadas durante este Sprint.

Tabla 30

Tareas del sprint backlog de gestión de ventas

ID	TAREA	DESCRIPCIÓN	RESPONSABLE	ESTADO	TIEMPO ESTIMADO
1	Registro de ventas	Crear punto de acceso y lógica para registrar ventas con fecha, usuario y monto total.	Daniel Santiago Soto Villamil	Terminado	5 días
2	Registro de detalle de ventas	Implementar la captura de productos y servicios vendidos, con cantidad y precios.	Daniel Santiago Soto Villamil	Terminado	4 días
3	Cálculo automático del total	Automatizar el cálculo del monto total con base en los productos ingresados.	Daniel Santiago Soto Villamil	Terminado	3 días
4	Actualización de ventas	Permitir la edición de los datos de ventas registradas, como detalles y montos.	Daniel Santiago Soto Villamil	Terminado	4 días
5	Visualización de ventas	Crear funcionalidad para mostrar y filtrar el historial de ventas registradas.	Daniel Santiago Soto Villamil	Terminado	5 días

Nota. Elaboración propia.

3.6.6. Desarrollo Iterativo y Validación.

Descripción del progreso de las tareas desarrolladas y validación de las funcionalidades implementadas.

Tabla 31

Desarrollo iterativo y validación de ventas

TAREA	PROGRESO	VALIDACIÓN REALIZADA
Registro de ventas	Terminado	Se verificará el almacenamiento correcto en la base de datos con datos válidos.
Registro de detalle de ventas	Terminado	Se validará la captura correcta de productos y su asociación con la venta registrada.
Cálculo automático del total	Terminado	Se asegurará que el cálculo del total se realice automáticamente sin errores.
Actualización de ventas	Terminado	Se confirmará que las modificaciones se reflejen correctamente en los registros.
Visualización de ventas	Terminado	Se validará la visualización ordenada y el filtrado eficiente de ventas registradas.

Nota. Elaboración propia.

3.6.7. Evaluación de Resultados

Análisis de los resultados obtenidos durante el sprint y alineación con los objetivos del módulo.

3.6.7.1. Resultados de ventas

Tabla 32

Resultados del módulo de ventas

PRUEBA FUNCIONAL		REGISTRO DE PROVEEDORES
Descripción	<ul style="list-style-type: none"> Permite el registro, actualización y visualización de ventas, incluyendo detalles como productos, cantidad y total. 	
Objetivos	<ul style="list-style-type: none"> Registrar ventas con información válida (fecha, monto total y usuario responsable). Capturar detalles de productos vendidos, incluyendo cantidad y precio unitario. Actualizar registros de ventas existentes. Visualizar un historial detallado y filtrable de ventas. 	
Condiciones	<ul style="list-style-type: none"> La base de datos debe estar activa y funcional. El usuario debe tener rol de administrador o encargado de ventas. 	
Resultado esperado	<ul style="list-style-type: none"> Los datos de la venta se registran correctamente en la base de datos, asociados al usuario responsable. El detalle de productos y servicios vendidos se almacena sin errores. El monto total de la venta se calcula automáticamente con base en los detalles ingresados. Las actualizaciones modifican los registros correctamente. La visualización muestra el historial ordenado y permite filtros por fecha o usuario. 	
Resultado obtenido	<ul style="list-style-type: none"> El sistema responde con un código de estado 201 Created para nuevos registros y 200 OK para actualizaciones. 	

- Los productos y servicios vendidos se registran correctamente en la base de datos, validando los datos únicos.
- El monto total se calcula sin errores, sumando correctamente los valores parciales.
- Las actualizaciones se reflejan en tiempo real sin inconsistencias.
- La visualización del historial de ventas es ordenada y funcional, permitiendo filtros eficientes.

Nota. Elaboración propia.

Figura 33

Resultado de gestión de ventas trabajos frontend

MONTO TOTAL	ID PERSONAL	DETALLES DE VENTA	ID TRABAJO	CANTIDAD	PRECIO UNITARIO
\$150.50	d2c2af7b-8a8c-482e-ad77-3b6f5435c97e	Ver Detalles	211b5a-0625-4a83-ab07-cc99b0b3f660	2	\$50.25
\$150.50	d2c2af7b-8a8c-482e-ad77-3b6f5435c97e	Ver Detalles	3c71343f-3f65-4825-9f27-6eade6c66330	1	\$50.00

Nota. Elaboración propia.

Figura 34

Resultado gestión de ventas backend

```

1 {
2   "monto_total": 150.50,
3   "id_persona": "d2c2af7b-8a8c-482e-ad77-3b6f5435c97e",
4   "detalleVentas": [
5     {
6       "id_trabajo": "c211b5a-0625-4a83-ab07-cc99b0b3f660",
7       "cantidad": 2,
8       "precio_unitario": 50.25
9     },
10    {
11      "id_trabajo": "3c71343f-3f65-4825-9f27-6eade6c66330",
12      "cantidad": 1,
13      "precio_unitario": 50.00
14    }
15  ]
16 }

```

Nota. Elaboración propia.

3.7. SPRINT 6: GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN

3.7.1. Introducción

El presente sprint se enfoca en la implementación de controles tecnológicos tomados de la norma ISO/IEC 27001, complementadas con lineamientos relevantes de la Guía de Implementación de Sistemas de Gestión de Seguridad de la Información (ver anexo 1), de NQA (National Quality Assurance). Estas referencias proporcionan un marco conceptual para la confidencialidad, integridad y disponibilidad de la información, mediante un enfoque estructurado que incluye el análisis del contexto organizacional, la planificación estratégica y la aplicación de controles tecnológicos, adaptados a las necesidades específicas del laboratorio de óptica.

3.7.2. Alcance del Sistema

El alcance de este sistema se centra en la gestión de la seguridad de la información específicamente relacionada con el laboratorio óptico en la ciudad de La Paz. Este proyecto toma controles tecnológicos de la norma UNE-ISO/IEC 27001:2023 y la Guía de Implementación de NQA, adaptándolos al contexto de un sistema de administración de inventarios de insumos ópticos.

La implementación está limitada a las necesidades del sistema de software y no abarca la totalidad de las exigencias normativas aplicables a una organización. Se priorizan los objetivos de confidencialidad, integridad y disponibilidad de la información, enfocados únicamente en los procesos del laboratorio óptico.

El alcance incluirá las siguientes áreas específicas:

- Gestión de inventarios y trazabilidad de insumos ópticos.
- Control seguro del acceso a la información del sistema.
- Protección de datos sensibles relacionados con productos y usuarios.

Con este enfoque, se asegura una aplicación práctica y adaptada de los principios de la UNE-ISO/IEC 27001:2023, permitiendo fortalecer la seguridad de la información en el sistema.

3.7.3. Contexto de la Organización

La organización, el Laboratorio de Óptica, opera en un entorno dinámico que requiere atención a factores internos y externos que pueden influir en su capacidad para gestionar de forma efectiva la seguridad de la información.

3.7.3.1. Análisis de factores internos

- a) Madurez organizacional: El laboratorio cuenta con procesos manuales de inventario establecidos, pero está en transición hacia un sistema digital.
- b) Cultura organizacional: Existe un enfoque regulado en la gestión del inventario, priorizando los datos de productos.
- c) Recursos disponibles: El sistema es gestionado por diferentes personas sin un enfoque en específico.
- d) Formatos de activos de información: La información se almacena principalmente en formatos físicos como cuadernos de papel.

- e) Complejidad del sistema: El laboratorio no utiliza un sistema que centralice la información del inventario, ventas y trabajos.
- f) Espacio físico: El laboratorio dispone de instalaciones propias, pero no con un área exclusiva donde se encuentre al almacenamiento de equipos y sistemas.

3.7.3.2. Identificación de partes interesadas y sus necesidades

- a) Propietarios: Esperan un sistema eficiente y confiable que garantice la seguridad de los datos y apoye la gestión operativa del laboratorio.

3.7.4. Liderazgo

3.7.4.1. Roles, responsabilidades y autoridades

Según el Anexo A 5.2, se han definido roles y responsabilidades claras para garantizar la seguridad de la información:

- a) Administrador del sistema: Gestiona los accesos al sistema y reporta incidentes o hallazgos relevantes a la alta dirección.
- b) Encargado de ventas: Se responsabiliza de proteger la confidencialidad de los datos de los clientes y la información relacionada con las transacciones de ventas.
- c) Encargado de trabajos: Asegura la protección y correcta gestión de los datos técnicos generados en el laboratorio, garantizando que se sigan los procedimientos establecidos.
- d) Encargado de proveedores y productos: Gestiona la información de los proveedores, mantiene actualizados los registros de los productos y asegura la trazabilidad de estos en el sistema.

3.7.5. Planificación

La planificación en el contexto del Sistema de Gestión de Seguridad de la Información (SGSI) adaptado para el laboratorio de óptica se centra en identificar riesgos específicos asociados con la gestión de inventarios, establecer objetivos claros de seguridad de la información y definir controles que sean prácticos y eficaces para mitigar estos riesgos. Este proceso asegura que las medidas implementadas no solo sean pertinentes al entorno del laboratorio, sino que también estén alineadas con sus necesidades operativas y estratégicas, priorizando la protección de los datos sensibles y la continuidad de los procesos clave.

3.7.5.1. Declaración de aplicabilidad (SoA)

La Declaración de Aplicabilidad (SoA) es un documento clave del para las buenas prácticas de gestión de la seguridad de la información que identifica los controles aplicables del Anexo A y su relación con los riesgos identificados. Este documento incluye:

- a)** Controles seleccionados: Identificados en el Anexo A para tratar los riesgos prioritarios.
- b)** Justificación de inclusión: Razones por las cuales un control es aplicable.
- c)** Estado de implementación: Si el control está implementado, en proceso o pendiente.

Tabla 33

Declaración de Aplicabilidad (SoA)

Control	Descripción del Control	¿Aplicable?	Justificación
8.1	La información accesible en dispositivos finales debe estar protegida, evitando accesos no autorizados o pérdidas de datos.	Sí	Al implementar controles en dispositivos finales, se previenen fugas de información, ataques a la integridad del sistema y accesos no autorizados.
8.2	La asignación y uso de acceso con privilegios deben ser restringidos y controlados.	Sí	Un mal uso o exceso de privilegios puede permitir accesos no autorizados, aumentando el riesgo de brechas de seguridad
8.3	Se debe limitar el acceso a la información y activos según las políticas definidas.	Sí	Limitar el acceso a la información estrictamente necesaria según el rol del usuario.
8.5	Implementar tecnologías y procedimientos de autenticación segura.	Sí	La autenticación débil es una de las principales causas de brechas de seguridad; una protección robusta reduce el riesgo de intrusiones.
8.6	Se debe supervisar y ajustar el uso de los recursos en función de los requisitos actuales.	Sí	Una infraestructura sin control de capacidad puede degradar el rendimiento y afectar la disponibilidad del servicio.
8.12	Se deben aplicar medidas para prevenir fugas de datos en sistemas, redes y dispositivos que manejen información confidencial.	Sí	La prevención efectiva evita filtraciones accidentales o intencionales

8.13	Se deben realizar copias de seguridad periódicas.	Sí	La pérdida de datos sin una copia de seguridad genera impactos irreversibles en la operación del sistema.
8.20	Se deben proteger y dispositivos de red para evitar accesos no autorizados o ataques.	Sí	La seguridad de la red es clave para evitar accesos indebidos y robo de información.
8.21	Se deben implementar mecanismos de seguridad para proteger los servicios de red.	Sí	La falta de seguridad en los servicios de red puede exponer datos sensibles y permitir ataques como MITM (Man-in-the-Middle).
8.24	Se deben definir e implementar reglas para el uso eficaz de criptografía, asegurando la confidencialidad e integridad de la información.	Sí	La criptografía es esencial para proteger datos contra accesos indebidos, interceptaciones y manipulaciones
8.25	Se deben establecer y aplicar medidas de seguridad en todas las etapas del desarrollo del software.	Sí	Integrar la seguridad en el ciclo de desarrollo evita problemas futuros y reduce costos de corrección.
8.28	Se deben aplicar principios de codificación segura para reducir riesgos de vulnerabilidades.	Sí	Una codificación insegura puede permitir ataques que comprometan la confidencialidad e integridad del sistema.
8.33	Los datos de prueba deben seleccionarse cuidadosamente y protegerse adecuadamente.	Sí	El uso indebido de datos reales en pruebas puede generar filtraciones de información.

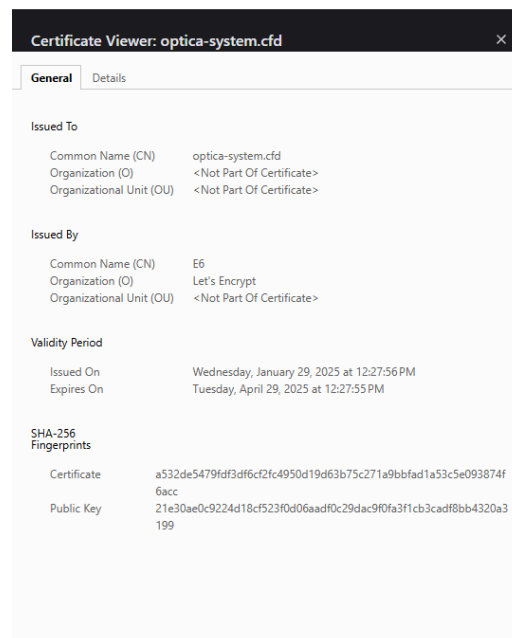
Nota. Elaboración propia.

3.7.6. Evaluación del Controles

8.1 - Dispositivos finales de usuario: La imagen muestra un certificado SSL/TLS para optica-system.cfd, asegurando conexiones cifradas. Esto cumple con el control 8.1 al proteger la información en tránsito entre dispositivos finales y el servidor.

Figura 35

Dispositivos finales del usuario



Nota. Elaboración propia.

8.2 - Gestión de privilegios de acceso: Se muestra código en NestJS utilizando metadatos (SetMetadata) para definir un sistema de control de acceso basado en roles. Esto cumple con el control 8.2, ya que restringe y controla los derechos de acceso de los usuarios según sus roles.

Figura 36

Gestión de privilegios de acceso

```
import { SetMetadata } from '@nestjs/common';
import { ValidRoles } from '../interfaces/valid-roles.interface';

export const META_ROLES = 'role'

export const RoleProtected = (...args: ValidRoles[]) => SetMetadata(META_ROLES, args);
```

Nota. Elaboración propia.

8.3 - Restricción del acceso a la información: La imagen refleja la implementación de un control de acceso basado en roles, donde el encargado de ventas solo puede visualizar los trabajos pendientes sin acceso a sus propiedades detalladas.

Figura 37

Restricción de acceso a la información

Crear Venta

Buscar Personal

Escribe el nombre del personal...

Comprador: Rels

Trabajos Pendientes

Numero Trabajo: 23 - Squirtle Trabajos Agregar

Trabajos Seleccionados

23 - Precio sugerido: 37.98 Quitar

Precio Sugerido Total: 37.98 Monto Total: 0

Cancelar Crear Venta

Nota. Elaboración propia.

8.5 - Autenticación segura: Se muestra la implementación de JWT (JSON Web Token) Strategy en NestJS, utilizando Passport para la autenticación. Esto se cumple con el

control 8.5, ya que la aplicación valida tokens de autenticación antes de permitir el acceso a los usuarios. Además, se verifica la existencia del usuario y su estado activo, evitando accesos no autorizados.

Figura 38

Autenticación segura

```
export class JwtStrategy extends PassportStrategy(Strategy) {
  constructor(
    @InjectRepository(Usuario)
    private readonly usuarioRepository: Repository<Usuario>,
    private readonly configService: ConfigService,
  ) {
    super({
      secretOrKey: configService.get<string>(JWT_SECRET),
      jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
    });

    if (!configService.get(JWT_SECRET)) {
      throw new Error(JWT_SECRET is not defined in the environment variables);
    }
  }

  async validate(payload: JwtPayload): Promise<Usuario> {
    try {
      const { id_usuario } = payload;

      const user = await this.usuarioRepository.findOne({
        where: { id_usuario, activo: true },
        relations: ['role'],
      });

      if (!user) {
        throw new UnauthorizedException('Token not valid: User not found');
      }

      if (!user.activo) {
        throw new UnauthorizedException('User is not active');
      }

      return user;
    } catch (error) {
      console.error('Error during JWT validation:', error.message);
      throw new UnauthorizedException('Invalid authentication token');
    }
  }
}
```

Nota. Elaboración propia.

8.6 - Gestión de capacidades: Se muestra una prueba de carga con K6, donde el sistema se evalúa en diferentes niveles de concurrencia, aumentando progresivamente hasta 200 usuarios simultáneos. Sin embargo, la caída del sistema ocurre entre 150 y 200 usuarios, lo que indica que esta es su capacidad máxima antes de experimentar degradación en el rendimiento. Esto está directamente relacionado con el control 8.6, ya que se está monitoreando el comportamiento del sistema bajo carga y ajustando su capacidad en función de los requisitos operativos.

Figura 39*Gestión de capacidades*

```

import http from 'k6/http';
import { sleep, check } from 'k6';

export const options = {
  stages: [
    { duration: '10s', target: 300 },
    { duration: '1m', target: 500 },
    { duration: '2m', target: 800 },
    { duration: '1m', target: 1000 },
  ],
  thresholds: {
    http_req_failed: ['rate<0.1'],
    http_req_duration: ['p(95)<500'],
  },
};

export default function () {
  const url = 'http://127.0.0.1:4000/api/usuarios/sign-up';
  const payload = JSON.stringify({
    nombre_usuario: 'testuser',
    contrasena: 'Abc123',
  });

  const params = {
    headers: { 'Content-Type': 'application/json' },
  };

  const response = http.post(url, payload, params);

```

Nota. Elaboración propia.

8.12 - Prevención de Fugas de Datos: Se muestra la configuración de UFW (Uncomplicated Firewall) para permitir tráfico en el puerto 5432 (PostgreSQL) dentro de la interfaz docker0, restringiendo el acceso a la base de datos. Esto se relaciona con el control 8.12, ya que la gestión de reglas de firewall ayuda a evitar accesos no autorizados y minimizar el riesgo de exposición de información confidencial. (Este control se refuerza con los controles 8.1, 8.5 y 8.20)

Figura 40*Prevención de Fugas de Datos.*

```

root@srv707482:~# sudo ufw allow in on docker0 to any port 5432 proto tcp
Rules updated
Rules updated (v6)

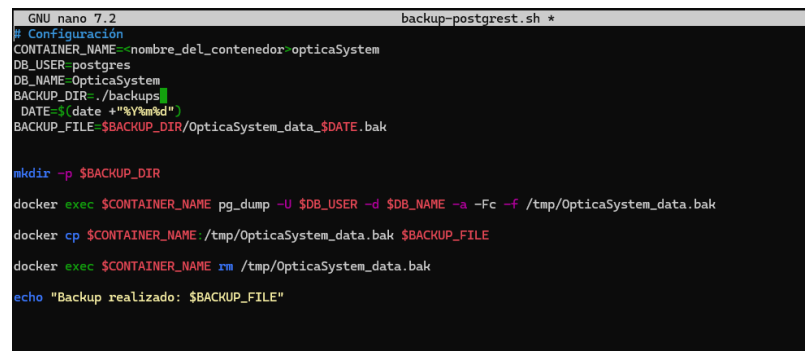
```

Nota. Elaboración propia.

8.13 - Copias de Seguridad de la Información: La imagen muestra un script de backup para PostgreSQL en un contenedor Docker, que ejecuta un pg_dump, copia el archivo de respaldo fuera del contenedor y luego elimina el archivo temporal. Esto está directamente relacionado con el control 8.13, ya que garantiza que la base de datos se respalde periódicamente.

Figura 41

Copias de seguridad de la información



```
GNU nano 7.2                                backup-postgres.sh *
```

```
# Configuración
CONTAINER_NAME=nombre_del_contenedor-opticaSystem
DB_USER=postgres
DB_NAME=OpticaSystem
BACKUP_DIR=./backups
DATE=$(date +%Y%m%d)
BACKUP_FILE=$BACKUP_DIR/OpticaSystem_data_$DATE.bak

mkdir -p $BACKUP_DIR

docker exec $CONTAINER_NAME pg_dump -U $DB_USER -d $DB_NAME -a -Fc -f /tmp/OpticaSystem_data.bak
docker cp $CONTAINER_NAME:/tmp/OpticaSystem_data.bak $BACKUP_FILE
docker exec $CONTAINER_NAME rm /tmp/OpticaSystem_data.bak
echo "Backup realizado: $BACKUP_FILE"
```

Nota. Elaboración propia.

8.20- Seguridad de Redes: La siguiente imagen muestra las reglas de firewall (UFW) configuradas en un servidor, permitiendo el acceso a puertos esenciales como 22 (SSH), 80, 443 y 5432, además de limitar conexiones SSH para prevenir ataques de fuerza bruta. Esto se relaciona con el control 8.20, ya que busca protección de los servicios de red, restringiendo accesos no autorizados y reduciendo la exposición a amenazas externas.

Figura 42

Seguridad de redes

```
root@srv707482:~# sudo ufw show added
Added user rules (see 'ufw status' for running firewall):
ufw allow 22
ufw allow 80
ufw allow 443
ufw allow 7080
ufw allow 80/tcp
ufw allow 443/tcp
ufw limit 22/tcp
ufw allow in on docker0 to any port 5432 proto tcp
```

Nota. Elaboración propia.

8.21 - Seguridad de los Servicios de Red: La imagen muestra la configuración de CORS (Cross-Origin Resource Sharing) en una aplicación NestJS, permitiendo solicitudes solo desde orígenes definidos en CORS_ORIGIN, con métodos específicos (GET, HEAD, PATCH, POST) y credenciales controladas. Esto está relacionado con el control 8.21, ya que limita el acceso a la API solo desde orígenes autorizados, reduciendo el riesgo de ataques como Cross-Site Request Forgery (CSRF) y Cross-Origin Attacks.

Figura 43

Seguridad de los servicios de red.

```
app.enableCors({
  origin: configService.get<string>('CORS_ORIGIN').split(','),
  methods: 'GET,HEAD,PATCH,POST',
  credentials: configService.get<boolean>('CORS_CREDENTIALS'),
});
```

Nota. Elaboración propia.

8.24 - Uso de la Criptografía: La imagen muestra la implementación de bcrypt para el hashing de contraseñas, donde las credenciales de los usuarios se almacenan de forma

segura utilizando un factor de costo de 10 rondas. Esto se relaciona con el control 8.24, ya que garantiza que las contraseñas no sean almacenadas en texto plano, protegiéndolas contra ataques de fuerza bruta y accesos no autorizados.

Figura 44

Uso de la criptografía

```
const hashedPassword = await bcrypt.hash(contrasenha, 10);

const usuario = this.usuarioRepository.create({
  ...usuarioData,
  role,
  personal,
  contrasenha: hashedPassword,
```

Nota. Elaboración propia.

8.25 - Seguridad en el Ciclo de Vida del Desarrollo: La tabla presentada define principios de seguridad aplicables al desarrollo del sistema, alineados con el control 8.25. Este control establece que la seguridad debe integrarse en todas las etapas del desarrollo del software, garantizando que las vulnerabilidades sean minimizadas desde el diseño hasta la implementación y mantenimiento.

Tabla 34

Seguridad en el ciclo de vida del software

#	Principio de Seguridad	Regla
1	Control de Acceso Basado en Roles	Ningún usuario tendrá más acciones de los estrictamente necesarios según su rol.
2	Almacenamiento Seguro de Datos Sensibles	Toda la información confidencial debe estar cifrada y nunca en texto plano.

3	Uso Exclusivo de HTTPS	Todo tráfico entre cliente y servidor debe estar cifrado con HTTPS.
4	Eliminación Segura de Datos	Los datos que ya no sean necesarios deben ser eliminados de forma lógica no física.
5	Aplicación de Principios de Codificación Segura	El código debe cumplir buenas prácticas de codificación seguridad.

Nota. Elaboración Propia

8.28 - Codificación Segura: La imagen muestra cómo se gestiona de manera general la definición de las entidades en TypeORM, siguiendo principios de diseño seguro, como el uso de UUID como identificador, restricciones en contraseñas (select: false) y relaciones adecuadamente estructuradas con otras entidades. Esto está relacionado con el control 8.28, ya que garantiza que la estructura de datos cumpla con buenas prácticas de seguridad, reduciendo vulnerabilidades como inyecciones SQL.

Figura 45

Codificación segura

```
import {
  Entity,
  PrimaryGeneratedColumn,
  Column,
  ManyToOne,
  JoinColumn,
  OneToOne,
  OneToMany,
} from 'typeorm';
import { Role } from '../roles/entities/role.entity';
import { Personal } from 'src/personal/entities/personal.entity';
import { Venta } from 'src/ventas/entities/venta.entity';

@Entity('usuarios')
export class Usuario {
  @PrimaryGeneratedColumn('uuid')
  id_usuario: string;

  @Column('varchar', { unique: true })
  nombre_usuario: string;

  @Column('varchar', { select: false })
  contraseña: string;

  @Column('boolean', { default: true })
  activo: boolean;

  @ManyToOne(() => Role, (role) => role.usuarios, { nullable: true })
  @JoinColumn({ name: 'id_rol' })
  role: Role;

  @OneToOne(() => Personal, (personal) => personal.usuario, { nullable: true })
  @JoinColumn({ name: 'id_personal' })
  personal: Personal;

  @OneToMany(() => Venta, (venta) => venta.usuario)
  venta: Venta[];
}
```

Nota. Elaboración propia.

8.33 - Datos de Prueba: La imagen muestra un conjunto de datos de prueba (seeds) utilizados para poblar la base de datos con información ficticia, incluyendo nombres, correos electrónicos y teléfonos, asegurando que no se utilicen datos reales. Esto se relaciona con el control 8.33, ya que establece que los datos utilizados en entornos de prueba deben ser generados específicamente para ese propósito y no deben incluir información sensible o real.

Figura 46

Datos de prueba

```
export const seedPersonal: SeedPersonal[] = [
  {
    id_personal: 'd2c0a7b5-8a8c-482e-a6d7-3b6f5435c97e',
    nombres: 'John',
    apellido_paterno: 'Admin',
    email: 'john.doe@example.com',
    telefono: '123456789',
    tipo_persona: 'juridica',
    activo: true,
  },
  {
    id_personal: 'f3b5c6d7-4a7c-4e8f-b9e2-1c5d7f9b3e6a',
    nombres: 'Jane',
    apellido_paterno: 'Productos',
    email: 'jane.smith@example.com',
    telefono: '987654321',
    tipo_persona: 'juridica',
    activo: true,
  },
  {
    id_personal: 'f3b5c6d7-4a7c-4e8f-b9e2-1c5d7f9f4e6a',
    nombres: 'Bulbasaur',
    apellido_paterno: 'Ventas',
    email: 'Verde@example.com',
    telefono: '123459876',
    tipo_persona: 'juridica',
    activo: true,
  },
  {
    id_personal: '7d405d21-d519-44f7-801c-0f88d9d158bd',
    nombres: 'Squirtle',
    apellido_paterno: 'Trabajos',
    email: 'Squirtle@example.com',
    telefono: '123456987',
    tipo_persona: 'juridica',
    activo: true,
  },
]
```

Nota. Elaboración propia.

CAPITULO IV

ANÁLISIS DE FACTIBILIDAD

4.1.FACTIBILIDAD TÉCNICA

La factibilidad técnica identifica si contamos con los recursos y conocimientos necesarios para el desarrollo e implementación del proyecto.

Aplicación al proyecto

Software libre: Uso de JS, frameworks de Nest.JS/ Vue.JS y PostgreSQL sin licencias de pago.

Entorno de desarrollo local: Se empleo una computadora personal con al menos 16 GB de RAM y procesador i5.

4.1.1. Hardware

Se presenta una descripción del hardware empleado durante el desarrollo del sistema.

Tabla 35

Hardware utilizado

Recurso	Estado Actual
Procesador	Intel i5 13-400F
Memoria RAM	16 GB RAM Corsair
Tarjeta de Video	NVIDIA RTX 4060TI

Almacenamiento	1TB
Sistema Operativo Base	Windows 11 Home

Nota. Elaboración propia.

4.1.2. Software

Tabla 36

Software utilizado

Herramienta	Categoría	Funcionalidad
Windows 11	Sistema Operativo	Plataforma base de desarrollo
Visual Studio Code	Editor / IDE	Edición de código fuente y debugging
JS/TS	Lenguaje de Programación	Desarrollo de código para funciones web
Nest.JS	Framework	Desarrollo del Backend.
Vue.JS	Framework	Desarrollar una interfaz web ágil para interactuar con el sistema
PostgreSQL	Base de Datos	Almacenar Datos

Postman	Herramienta de Testing	Probar funcionalidades del backend
----------------	------------------------	------------------------------------

Nota. Elaboración propia.

4.2. ANÁLISIS DE FACTIBILIDAD ECONÓMICA

Determina la viabilidad económica del proyecto considerando costos iniciales, costos operativos y beneficios esperados.

4.2.1. COCOMO II

4.2.1.1. Componentes funcionales

Tabla 37

Componentes funcionales

Componente	Descripción
Entradas Externas (EI)	Datos que ingresan al sistema desde el usuario u otros sistemas.
Salidas Externas (EO)	Información que el sistema genera para los usuarios u otros sistemas.
Consultas Externas (EQ)	Peticiones específicas del usuario que generan una respuesta inmediata (sin modificaciones de datos).

Archivos Lógicos Internos (ILF)	Bases de datos o estructuras de datos gestionadas por el sistema.
Interfaces de Archivos Externos (EIF)	Archivos o bases de datos externos que el sistema utiliza, pero no controla.

Nota. Elaboración propia.

4.2.1.2. Ponderación de funciones

Cada función se clasifica como simple, media o compleja según su cantidad de datos involucrados o el esfuerzo necesario para implementarla. Esto otorga un valor base:

Tabla 38

Ponderación de funciones

Tipo de Componente	Simple	Medio	Complejo
Entrada Externa (EI)	3	4	6
Salida Externa (EO)	4	5	7
Consulta Externa (EQ)	3	4	6
Archivo Lógico Interno (ILF)	7	10	15
Interfaz Archivo Externo (EIF)	5	7	10

Nota. Elaboración Propia.

4.2.1.3. Estimación inicial de puntos de función

Tabla 39

Tabla estimación inicial de puntos de función

Componente	Cantidad	Complejidad	Puntos
Entradas Externas (EI)	4	Medio (4)	16
Salidas Externas (EO)	3	Complejo (7)	21
Consultas Externas (EQ)	2	Simple (3)	6
Archivos Lógicos Internos (ILF)	3	Medio (10)	30
Interfaces Externas (EIF)	2	Medio (7)	14
Total	14		87 PF

Nota. Elaboración propia.

4.2.2. Resultados generales de la estimación

Tamaño total (equivalente): 6,960 SLOC

Factor de Ajuste de Esfuerzo (EAF): 1.00

Esfuerzo total estimado: 24.8 persona-meses

Duración total (cronograma): 10.2 meses

Costo total: \$1,241

Puntos de Función (sin ajustar): 87

Lenguaje de 3ª Generación: (Ejemplo: Python, Java, C#, etc.)

4.2.3. Distribución por Fases (Adquisición)

La herramienta COCOMO II desglosa la Fase de Adquisición (Acquisition Phase) en cuatro etapas principales: Inception, Elaboration, Construction y Transition. A continuación, se muestra el esfuerzo (en persona-meses), el cronograma (en meses).

Tabla 40
Resultados de estimación

Fase	Esfuerzo(persona-meses)	Cronograma (meses)	Personal Promedio	Costo
Comienzo	1.5	1.3	1.2	\$74
Elaboración	6.0	3.2	1.6	\$298
Construcción	18.9	6.4	3.0	\$944
Transición	3.0	1.3	2.3	\$149

Nota. Elaboración propia.

4.2.4. Distribución del Esfuerzo por Fases y Actividades (RUP/MBASE)

Además de la división por fases, COCOMO II proporciona una distribución del esfuerzo (persona-meses) de acuerdo con actividades específicas típicas de metodologías como RUP (Rational Unified Process) o MBASE:

Tabla 41

Distribución del esfuerzo por fases y actividades

Actividad / Fase	Comienzo	Elaboración	Construcción	Transición
Gestión	0.2	0.7	1.9	0.4
Medio ambiente/CM	0.1	0.5	0.9	0.1
Requisitos	0.7	1.4	0.9	0.0
Diseño	0.3	0.9	2.3	0.0
Implementación	0.1	3.0	11.4	1.4
Evaluación	0.0	0.2	0.7	0.0
Despliegue	0.0	0.1	0.2	0.9

Nota. Elaboración propia.

4.3. ANÁLISIS COSTO-BENEFICIO

Se comparan los costos totales (iniciales y operativos) con los beneficios anuales. Esto permite estimar la rentabilidad y el plazo de recuperación de la inversión.

Tabla 42
Costos y beneficios

Concepto	Costo/Beneficio (USD)	Periodo	Comentarios
INVERSIÓN INICIAL			
Licencias de Software	0	Único	Uso de software libre
Infraestructura de Servidor/Nube	0	Único	Ejecución local
Salario del Desarrollador	0	-	No se contemplan honorarios
Capacitación	100	Único	Único costo real incurrido
Total Inversión Inicial	100		
COSTOS DE OPERACIÓN ANUAL			
Mantenimiento, Actualizaciones, Reentrenamiento	50	Anual	Costo simbólico por tiempo, electricidad, imprevistos, etc.

Total Costos Operativos	50		
BENEFICIOS ANUALES (Estimados)			
1. Ahorro por reducción de tiempo en imprevistos	1.600	Anual	160 h/año ahorradas a 10 USD/h
2. Beneficios intangibles	N/A	Anual	Mayor calidad, reducción de correcciones en trabajos, menor tiempo empleado en trazabilidad.
Total Beneficios Anuales	1.600		

Nota. Elaboración propia.

4.3.1. Cálculo de ROI y Punto Equilibrio.

- Inversión Inicial: 100 USD
- Costos Operativos Anuales: 50 USD
- Beneficios Anuales: 1.600 USD

Conclusión del Análisis Costo-Beneficio: El proyecto recupera la inversión (100 USD) en alrededor de 23.5 días.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Optimización de la administración de insumos

El desarrollo del sistema de administración de inventarios permitió reducir las pérdidas de insumos en el laboratorio óptico “OptalVision” mediante la implementación de procesos eficientes y un seguimiento detallado de los productos. La trazabilidad de los movimientos de inventario proporcionó una visibilidad completa de los flujos de información, facilitando una gestión operativa más organizada y efectiva.

- Diseño robusto de la base de datos

La base de datos relacional fue diseñada respetando los principios de organización y seguridad de los datos, lo que aseguró la integridad de la información almacenada. Este diseño, alineado con las características operativas de los laboratorios de óptica, permitió gestionar de manera confiable los datos de usuarios, productos, proveedores y ventas.

- Implementación efectiva de módulos clave

Los módulos desarrollados para usuarios, personal, productos, proveedores y ventas contribuyeron a una gestión integral del sistema, permitiendo registrar y monitorear los movimientos de insumos de manera eficiente. La integración de estas

funcionalidades aseguró que el sistema respondiera a las necesidades específicas de “OptalVision”, mejorando los procesos internos de la óptica.

- Controles tecnológicos alineados con la norma ISO/IEC 27001:2023

La incorporación de controles tecnológicos de la norma ISO/IEC 27001:2023 ayudo a la protección de la información sensible del laboratorio. La selección de controles adecuados permitió reforzar la confidencialidad, integridad y disponibilidad de los datos, asegurando la conformidad con estándares internacionales y generando confianza en los procesos del sistema.

- Automatización de reportes y soporte en la toma de decisiones

La generación automatizada de reportes basada en los datos de ventas, trabajos e insumos proporcionó a los administradores de “OptalVision” información clave para el análisis y la toma de decisiones estratégicas. Estos reportes facilitaron la identificación de patrones, optimización de recursos y planificación de acciones futuras para mejorar la gestión del laboratorio.

- Satisfacción del cliente y conformidad con los objetivos planteados

La implementación del sistema cumplió con las expectativas del laboratorio óptico “OptalVision”, logrando los objetivos específicos definidos. El sistema no solo mejoró la operatividad del laboratorio, sino que también promovió una gestión más segura y eficiente, lo que fue reconocido como un aporte valioso por parte del personal administrativo y técnico de la óptica.

5.2. RECOMENDACIONES

Como resultado del presente proyecto, se sugieren algunas recomendaciones que pueden ser llevadas a cabo en futuros trabajos, los cuales son:

- Mejoras en Reportes de Inventarios

Se sugiere implementar reportes avanzados que permitan filtrar por periodos personalizados y categorizar productos según movimientos recientes o históricos. Actualmente, el sistema ofrece reportes básicos que cumplen con los requisitos funcionales iniciales, pero estas mejoras no fueron incluidas debido a la prioridad de garantizar la funcionalidad esencial antes de abordar características adicionales.

- Gestión Detallada de Usuarios

Una funcionalidad futura podría incluir la posibilidad de establecer permisos a nivel de acciones específicas dentro del sistema (por ejemplo, permitir que ciertos usuarios visualicen, pero no editen datos). Esto no se incorporó en esta fase debido a que los roles y permisos actuales fueron diseñados de acuerdo con los requerimientos específicos del cliente, priorizando simplicidad y rapidez en la implementación.

- Ampliación de Controles de Seguridad

Aunque el sistema ya registra las acciones principales realizadas por los usuarios, se podría incluir una gestión de seguridad más detallada que registre eventos específicos como intentos fallidos de inicio de sesión o modificaciones en configuraciones críticas.

- Exportación Personalizada de Datos

Actualmente, el sistema permite exportar datos en formatos predefinidos. Se recomienda en una fase futura incorporar opciones de personalización para que los usuarios puedan seleccionar columnas y formatos específicos. Esto no se incluyó porque las opciones existentes son suficientes para cumplir con los requerimientos actuales de la empresa.

- Mejoras en la Visualización de Inventarios

Se puede considerar agregar gráficos dinámicos que resuman el estado del inventario en tiempo real. Aunque esto aportaría valor, no se priorizó en esta fase para garantizar que los recursos se destinaran a funcionalidades operativas críticas.

- Respaldo de Datos

Una mejora futura podría incluir la variabilidad de los respaldos de las copias de seguridad para que se realicen en base a eventos específicos. Actualmente, el sistema permite realizar respaldos automáticos, lo cual fue considerado suficiente para cumplir con los requisitos mínimos de seguridad.

REFERENCIAS BIBLIOGRÁFICAS

Coarite Tumiri, V. (2007). Sistema Integrado de Control de Inventario 'ATIPAJ' Compañía Cervecera Boliviana S.A. Universidad Mayor de San Andrés, Carrera de Informática.

La Fuente Choque, J. (2008). Sistema para la Gestión de Ventas e Inventario Caso: Importadora Soluciones Médicas Lifemed S.R.L. Universidad Mayor de San Andrés, Carrera de Informática.

Ramos Paye, J. L. (2005). Sistema de Control de Inventarios para Laboratorios Crespal S.A. Regional Sucre. Universidad Mayor de San Andrés, Carrera de Informática.

Choque Chambilla, R. F. (2007). Sistema de Información de Compras e Inventarios SAMA. Universidad Mayor de San Andrés, Carrera de Informática.

Suarez Marin, V. (2008). Sistema de Control y Seguimiento de Almacenes para la Corte Departamental Electoral La Paz, Sala Provincias. Universidad Mayor de San Andrés, Carrera de Informática.

Chiri Honorio, C. (2009). Sistema de Entradas y Salidas e Inventario Caso: BOLITAL S.R.L. Universidad Mayor de San Andrés, Carrera de Informática.

Callisaya Apaza, W. D. (2017). Software de Gestión y Control de Inventarios Caso: AGADON S.R.L. Universidad Mayor de San Andrés, Carrera de Informática.

Alaimo, M. (2021). *Scrum y algo más*.

Carreón, A. (2020). dev: <https://dev.to/alfredocu/tailwind-css-spanish-4lk4>

Gibert, M. (2007). *Bases de datos en PostgreSQL*.
https://doi.org/https://dgvs.mspbs.gov.py/files/documentos/06_06_2016_20_09_46_P06_M2109_02152.pdf

Gonzales, E. (2023). *cimatic*. <https://cimatic.com.mx/blog/todo-lo-que-debes-conocer-de-sistema-de-inventarios/>

Haverbeke, M. (2018). *Eloquent JavaScript*. No Starch Press.
https://doi.org/https://eloquentjs-es.thedojo.mx/Eloquent_JavaScript.pdf

Jiménez, C. D. (2020). *Ciberseguridad*. Marcombo. Ciberseguridad.

kinsta. (Julio de 2022). *Kinsta Inc*. <https://kinsta.com/es/base-de-conocimiento/nestjs/>

López, I. (2021). *Node.js Javascript del lado del servidor*. <https://doi.org/https://annas-archive.li/md5/f8a2c5d2aeca418927b369aff0133096>

miro. (2023). <https://miro.com/es/diagrama/que-es-diagrama-ishikawa/>

Muller, M. (2003). *Fundamentos de administración de inventarios*. FreeLibros.

NestJS, C. (2023). NestJS - A progressive Node.js framework: <https://docs.nestjs.com>

PostgreSQL, G. D. (2023). PostgreSQL Documentation: <https://www.postgresql.org/docs/>

Red Hat. (2023). *redhat*. <https://www.redhat.com/es/topics/api/what-is-a-rest-api#%C2%BFqu%C3%A9-es-una-api-de-rest>

Roberto Hernández Sampieri, C. F. (2010). *Metodología de la investigación*. México: McGraw-Hill.

Salas, H. G. (2009). *Inventarios: manejo y control*. Ecoe Ediciones.

Sampieri, H. (2018). *METODOLOGÍA DE LA INVESTIGACIÓN: LAS RUTAS CUANTITATIVA, CUALITATIVA Y MIXTA* (Sexta ed.). McGraw-Hill.

Sommerville, I. (2011). *Ingeniería de software*. Pearson Educación.
<https://doi.org/https://annas-archive.li/md5/e1cb1c2ff784861f5dfc329bfae04be8>

Standardization, I. O. (2022). *Controles de seguridad para la información*. ISO.

Talaminos, A. (2022). *TypeScript para todo*. <https://doi.org/https://annas-archive.li/md5/a0a3b6094645a448e33e8bfb986fb67e>

TechTarget. (Enero de 2023). *web application (web app)*. TechTarget:
<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>

typeorm. (2023). typeorm: <https://typeorm.io/>

UNE-ISO/IEC. (2023). *Sistema de Gestión de la Seguridad de la Información*.

UNE-ISO/IEC. (2023). *Tecnología de la información - Técnicas de seguridad - Código de prácticas para los controles de seguridad de la información*.

Vue.js Team. (s.f.). *vuejs*. <https://es.vuejs.org/v2/guide/>

Waller, M. A. (2015). *Administración de inventarios*. Pearson.

APÉNDICE

Tabla 43

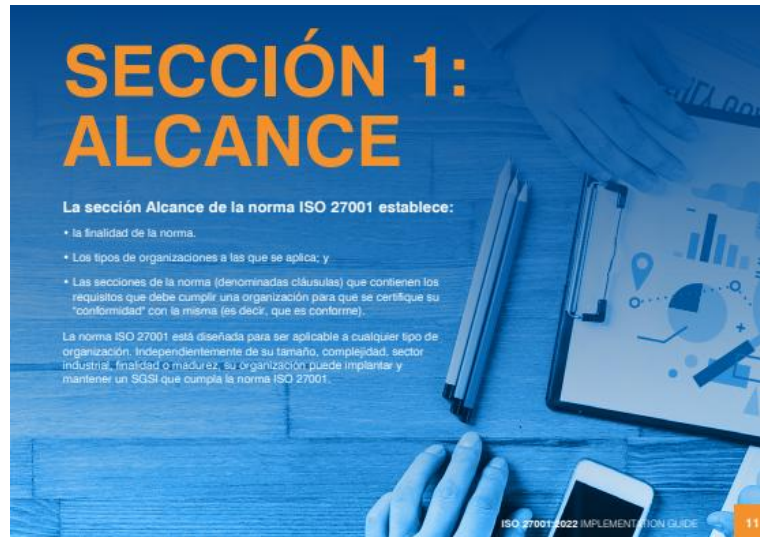
Cronograma de Gant

Nota. Elaboración propia.

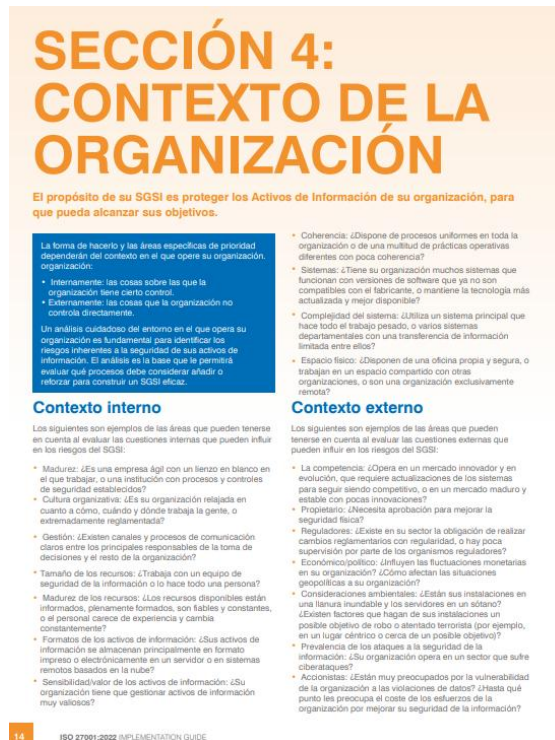
	Tiempo en Semanas (Octubre 2024 - Enero 2025)																
Actividades	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Elaboración de Marco Teórico																	
Elaboración de Diseño Metodológico																	
Revisión y Corrección del documento																	
Elaboración de la propuesta																	
Conclusiones, recomendaciones, bibliografía y anexos																	
Revisión y Corrección del documento																	
Defensa interna																	
Corrección de observaciones y recomendaciones																	
Orden de Empaste																	

ANEXOS

Anexo 1: Alcance ISO 27001



Anexo 2: Contexto de la organización ISO 27001



Partes interesadas

Una parte interesada es cualquiera que está, pueda estar o se pueda afectar por una acción u omisión de su organización. Las partes interesadas se irán aclarando a lo largo del proceso de análisis exhaustivo de los problemas internos y externos.

Probablemente incluirá a accionistas, propietarios, reguladores, clientes, empleados y competidores. Dependiendo de su empresa, pueden incluir al público en general y al medio ambiente. No tiene que intentar comprender o satisfacer todos sus caprichos, pero sí determinar cuáles de sus necesidades y expectativas son relevantes para su SGSI.

Alcance del SGSI

Para cumplir con la norma ISO 27001, debe documentar el alcance de su SGSI. Los alcances documentados suelen describir:

- Los límites del lugar o lugares físicos incluidos (o no incluidos).
- Los límites de las redes físicas y lógicas incluidas (o no incluidas).
- Los grupos de empleados internos y externos incluidos (o no incluidos).
- Los procesos, actividades o servicios internos y externos incluidos (o no incluidos).
- Interfaces clave en los límites del ámbito de aplicación.

Si quiere priorizar recursos creando un SGSI que no cubra toda su organización, seleccionar un ámbito limitado a la gestión de los riesgos de las partes interesadas tiene un enfoque pragmático. Esto se puede hacer incluyendo sólo sitios, activos, procesos o unidades de negocio o departamentos específicos. Algunos ejemplos son:

- "Todas las operaciones realizadas por el Depto. IT."
- "Soporte y gestión del correo electrónico".
- "Todos los equipos, sistemas, datos e infraestructuras del centro de datos de la organización, situado en la sede de Bangalore".

CONSEJO: Documente o mantenga un archivo de toda la información recopilada en el análisis del contexto de su organización y de las partes interesadas, como por ejemplo:

- Conversaciones con un alto representante de la organización, por ejemplo, un director general, CFO, CTO...
- Actas de reuniones o planes de negocio.
- Un documento específico que identifique los problemas internos/externos y las partes interesadas, así como sus necesidades y expectativas, por ejemplo, un análisis DAFO, un estudio PESTLE o una evaluación de riesgos empresariales de alto nivel.



Nueva consideración para el cambio climático

ISO ha introducido cambios en la norma ISO 27001 para subrayar la importancia de abordar los efectos del cambio climático en el marco de los sistemas de gestión de las organizaciones.

Para mejorar la concienciación y la respuesta de las organizaciones al cambio climático, ISO ha introducido dos cambios fundamentales en la cláusula 4:

Cláusula original 4.1: "Comprensión de la organización y su contexto. La organización debe determinar las cuestiones externas e internas que son relevantes para su propósito y que afectan a su capacidad para lograr el resultado o resultados previstos de su sistema de gestión de XXX."

Esta cláusula incluye ahora explícitamente la afirmación "La organización determinará si el cambio climático es una cuestión relevante".

Cláusula original 4.2: "Comprensión de las necesidades y expectativas de las partes interesadas. La organización debe determinar:

- Las partes interesadas que son relevantes para el sistema de gestión XXX.
- Los requisitos pertinentes de estas partes interesadas.
- Cómo de estos requisitos se abordarán a través del sistema de gestión XXX."

La cláusula ahora también dice: "Nota: Las partes interesadas pertinentes pueden tener requisitos relacionados con el cambio climático".

Anexo 3: Liderazgo ISO 27001

SECCIÓN 5: LIDERAZGO

La importancia del liderazgo

El liderazgo en este contexto significa la participación en el establecimiento de la dirección del SGSI, su aplicación y provisión de recursos. Esto incluye:

- Garantizar que los objetivos del SGSI sean claros y estén alineados con la estrategia general.

- Claridad en las responsabilidades y la rendición de cuentas.
- El pensamiento basado en el riesgo está en el centro de toda toma de decisiones.

- Comunicación clara de esta información a todas las personas dentro del ámbito de su SGSI.

La norma ISO 27001 concede gran importancia al compromiso activo de la Dirección en el SGSI, partiendo de la base de que el compromiso de la Dirección es crucial para garantizar la implantación efectiva y el mantenimiento de un SGSI eficaz por parte de los empleados.

Política de seguridad info.

Una responsabilidad vital de la dirección es establecer y documentar una Política de Seguridad de la Información (PSI) que esté alineada con los objetivos clave de la organización. Debe incluir objetivos o un marco para establecerlos. Para demostrar que está alineada con el contexto de la organización y los requisitos de las principales partes interesadas, se recomienda que haga referencia o contenga un resumen de los principales problemas y requisitos que debe gestionar. También debe incluir el compromiso de:

- Cumplir los requisitos aplicables en materia de seguridad de la información, como los requisitos legales, las expectativas de los clientes y los compromisos contractuales.
- La mejora continua de su SGSI.

El PSI puede hacer referencia a, o incluir sub-políticas que cubran, los controles clave del SGSI de la organización. Algunos ejemplos son: la selección de proveedores críticos para la seguridad de la información, la contratación y formación de los empleados, clear desk y clear screen, controles criptográficos, controles de acceso, etc.

Para demostrar la importancia del PSI, es aconsejable que lo autorice el miembro de mayor rango de su Alta Dirección o cada uno de los miembros del equipo de Alta Dirección.

CONSEJO: Para asegurarse de que su PSI está bien comunicado y a disposición de las partes interesadas, recomendamos:

- Incluirlo en los paquetes de iniciación y en las presentaciones para nuevos empleados y contratistas.
- Publique la declaración clave en los tablones de anuncios internos, las intranets y el sitio web de su organización.
- Haga que su cumplimiento y/o apoyo sea un requisito contractual para empleados, contratistas y proveedores críticos para la seguridad de la información.

Funciones y responsabilidades

Para que las actividades de seguridad de la información formen parte de las actividades de la mayoría de las personas de la organización, las responsabilidades y las obligaciones de rendir cuentas deben definirse y comunicarse claramente.

Aunque la norma no exige la designación de un representante de seguridad de la información, puede ser útil para algunas organizaciones nombrar a uno que dirija un equipo de seguridad de la información para coordinar la formación, supervisar los controles e informar sobre el funcionamiento del SGSI a la alta dirección. Es posible que esta persona ya sea responsable de la protección de datos.

Sin embargo, para desempeñar su función con eficacia, lo ideal es que forme parte del equipo de alta dirección y que tenga sólidos conocimientos técnicos sobre gestión de la seguridad de la información.

Evidenciar liderazgo al auditor

La Dirección serán aquellos que establecen la dirección estratégica y aprueban la asignación de recursos para la organización o área de negocio con el alcance de su SGSI. Dependiendo de cómo esté estructurada su organización, estas personas pueden ser el equipo directivo diario. Un auditor normalmente pondrá a prueba el liderazgo mediante una entrevista, y evaluará su nivel de implicación en el:

- Evaluación de riesgos y oportunidades.
- Establecimiento y comunicación de políticas.
- Fijación y comunicación de objetivos.
- Revisión y comunicación del rendimiento del sistema.
- Asignación de recursos, responsabilidades y obligaciones adecuadas.

CONSEJO: Antes de su auditoría externa, identifique quién de la alta dirección se reunirá con el auditor externo. Prepárenlos con un simulacro de entrevista que incluya las preguntas que espera que les hagan.

Anexo 4: Planificación ISO 27001

SECCIÓN 6: PLANIFICACIÓN

La norma ISO 27001 es una herramienta de gestión de riesgos que orienta a una organización para que identifique las causas de sus riesgos para la seguridad de la información. Como tal, el propósito de un SGSI es:

- Identificar los riesgos importantes, los obvios y los ocultos pero peligrosos.
- Garantizar que las actividades y los procesos operativos de una organización estén diseñados, dirigidos y dotados de recursos para gestionar intrínsecamente esos riesgos.
- Responder y adaptarse automáticamente a los cambios para hacer frente a los nuevos riesgos y reducir continuamente la exposición al riesgo de la organización.

Disponer de un plan de acción detallado que se supervise de forma alineada y se apoye en revisiones periódicas es crucial, y proporciona al auditor la mejor prueba de que la planificación del sistema está claramente definida.

Evaluación de riesgos

La evaluación de riesgos es el núcleo de cualquier SGSI eficaz. Ni siquiera la organización mejor dotada de recursos puede eliminar la posibilidad de que se produzca un incidente de seguridad de la información. Para todas las organizaciones, la evaluación de riesgos es esencial:

- Aumentar la probabilidad de identificar todos los riesgos potenciales mediante la participación de personas que utilicen técnicas de evaluación.
- Asignar recursos para abordar las áreas más prioritarias.
- Tomar decisiones estratégicas sobre cómo gestionar los riesgos que permitan alcanzar con mayor probabilidad sus objetivos.

La mayoría de los marcos de evaluación de riesgos consisten en una tabla que contiene los resultados de los elementos 1-4 con una tabla o matriz suplementaria que cubre el punto 5.

Un auditor externo esperará ver un registro de su evaluación de riesgos, un propietario asignado para cada riesgo identificado y los criterios que ha utilizado.

CONSEJO: El anexo A (5.9) contiene el requisito de mantener una lista de los activos de información, los activos asociados a la información (por ejemplo, edificios, archivadores, ordenadores portátiles, licencias) y las instalaciones de procesamiento de la información. Si completa su evaluación de riesgos evaluando sistemáticamente los riesgos que plantea cada elemento de esta lista, habrá cumplido dos requisitos en el mismo ejercicio.

La norma ISO 27005 - Gestión de riesgos para la seguridad de la información ofrece orientación sobre el desarrollo de una técnica de evaluación de riesgos para su organización. Sea cual sea la técnica que elija, debe incluir los siguientes elementos:

- 1 Proporcionar un impulso para la identificación sistemática de los riesgos (por ejemplo, revisando los activos, grupos de activos, procesos, tipos de información uno a uno, comprobando en cada uno la presencia de amenazas y vulnerabilidades comunes, y registrando los controles que tiene actualmente para gestionarlos).
- 2 Proporcionar un marco para evaluar la probabilidad de que se produzca cada riesgo de forma sistemática (por ejemplo, una vez al mes, o a vez al año).
- 3 Proporcionar un marco para evaluar las consecuencias de que se produzca cada riesgo sobre una base coherente (por ejemplo, pérdida de 1.000 €, pérdida de 100.000 €).
- 4 Proporcionar un marco para clasificar cada riesgo identificado sobre una base coherente teniendo en cuenta su evaluación de la probabilidad y las consecuencias.
- 5 Establecer criterios documentados que especifiquen, para cada puntuación o categoría de riesgo, el tipo de acción que debe emprenderse y el nivel o prioridad que se le asigna.

Anexo 5: Evaluación de desempeño ISO 27001

SECCIÓN 9: EVALUACIÓN DEL DESEMPEÑO

Existen tres formas principales de evaluar el rendimiento de un SGSI. Éstas son:

- Supervisar la eficacia de los controles del SGSI.
- Mediante auditorías internas.
- Reuniones de revisión por la dirección.

Seguimiento, medición, análisis y evaluación

Su organización tendrá que decidir qué necesita supervisar para asegurarse de que el proceso del SGSI y los controles de seguridad de la información funcionan según lo previsto. No es práctico para una organización supervisar manualmente todo en todo momento. Si intentara hacerlo, es probable que el volumen de datos fuera tan grande que resultara prácticamente imposible utilizarlo con eficacia. Por lo tanto, tendrá que tomar una decisión informada sobre qué supervisar. Las siguientes consideraciones serán importantes:

- ¿Qué procesos y actividades están sujetos a las amenazas más frecuentes y significativas?
- ¿Qué procesos y actividades presentan las vulnerabilidades inherentes más importantes?
- ¿Qué resulta práctico controlar y generar información significativa y oportuna?
- ¿Está automatizando su supervisión?
- Con cada proceso de supervisión que ponga en marcha, para que sea eficaz debe definirse claramente:
 - Cómo se lleva a cabo el seguimiento
 - Cuando se emprende.
 - Quién es responsable de llevarla a cabo.
 - Cómo se comunican los resultados, cuándo, a quién y qué hacen con ellos.
 - Si los resultados de la supervisión identifican un rendimiento inaceptable, ¿cuál es el proceso o procedimiento de escalada para hacer frente a esta situación?

Para demostrar a un auditor que dispone de un proceso de supervisión adecuado, deberá conservar registros de los resultados de la supervisión, los análisis, las revisiones de evaluación y cualquier actividad de escalado.

Auditorías internas

El objetivo de las auditorías internas es comprobar los puntos débiles de los procesos del SGSI e identificar oportunidades de mejora. También son una oportunidad para que la alta dirección compruebe la eficacia del SGSI. Si se hacen bien, las auditorías internas pueden garantizar que no haya sorpresas en las auditorías externas.

Las auditorías internas que realice deben comprobar

- La coherencia con que se siguen y aplican los procesos, procedimientos y controles.
- Hasta qué punto sus procesos, procedimientos y controles generan los resultados previstos.
- Si su SGSI sigue cumpliendo la norma ISO 27001 y los requisitos de las partes interesadas.

Para garantizar que las auditorías se llevan a cabo con un calidad y de forma que aporten valor añadido, es necesario que las realicen personas que sean:

- Respetadas.
- Competentes.
- Familiarizadas con los requisitos de la norma ISO 27001.
- Capaces de interpretar su documentación y concedores de técnicas y comportamientos de auditoría sólidos.

Lo más importante es que se le asigne tiempo suficiente para completar la auditoría y se le garantice la cooperación de los empleados pertinentes. Debe mantener un plan para llevar a cabo sus auditorías internas. Un auditor externo esperará que este plan garantice que todos los procesos de su SGSI se auditan en un ciclo de tres años y que cuenten con procesos que:

- Muestran pruebas de un rendimiento deficiente (por ejemplo, a través de auditorías previas, o resultados de supervisión o incidentes de seguridad de la información).
- Gestionan los riesgos más importantes para la seguridad de la información.
- Se auditan con mayor frecuencia.

El auditor externo también esperará que las acciones identificadas en las auditorías se registren, sean revisadas por los empleados adecuados y se apliquen a tiempo para rectificar cualquier problema significativo. En el plazo de cierre de las auditorías, los auditores deben tener en cuenta las oportunidades de mejora identificadas que requieran una inversión significativa de recursos.

Anexo 6: Entrevista al dueño de OptalVision

ENTREVISTA AL DUEÑO DE OPTALVISION

1 Sobre la Operatividad General

¿Cuáles son los procesos más importantes que se realizan actualmente en el laboratorio óptico?

R -

¿Qué problemas o dificultades enfrenta en la gestión diaria del inventario y de los insumos ópticos?

R -

¿Qué aspectos del negocio considera que podrían optimizarse con el nuevo sistema?

R -

2 Sobre la Gestión de Inventarios

¿Qué información necesita registrar sobre los productos e insumos ópticos?

R -

¿Cómo realiza actualmente el seguimiento de los productos en el inventario?

R -

¿Es importante para usted saber la trazabilidad de los productos, como quién los suministra y cómo se utilizan?

R -

¿Con qué frecuencia necesita actualizar los datos de inventario?

R -

3 Sobre la Gestión de Ventas

¿Qué detalles considera necesarios registrar en cada venta?

R -

¿Cómo realiza actualmente el seguimiento de las ventas?

R -

¿Necesita reportes de ventas? Si es así, ¿qué tipo de información le gustaría incluir en ellos?

R -

4 Sobre los Proveedores

¿Qué datos considera importantes registrar sobre los proveedores?

R -

¿Con qué frecuencia actualiza la relación entre productos y proveedores?

R -

¿Es importante para usted tener un registro histórico de los proveedores y su relación con los productos?

R -

5 Sobre los Trabajos Ópticos

¿Qué detalles técnicos de los trabajos ópticos necesitan registrarse en el sistema?

R -

¿Cómo se realiza el seguimiento de los estados de los trabajos (pendiente, en proceso, finalizado)?

R -

¿Es importante para usted asociar los trabajos con los insumos utilizados?

Anexo 7: Boleta de trabajo

Telf.: 2409468

L N° 000196

Optica: 2000 OROPERA

Fecha de Recepción: / / Hrs.

Fecha de Entrega: 18 / 12 / 24 Hrs. 10:00

INSTRUCCIONES ESP: JENNY PACHECO

	ESF.	CIL.	EJE	BASE
O.D.	<u>-125</u>	<u>-275</u>	<u>180</u>	<u>4</u>
	<u>-100</u>	<u> </u>	<u> </u>	
O.I.	<u>+250</u>	<u> </u>	<u> </u>	<u> </u>
	<u> </u>	<u> </u>	<u> </u>	

TIPO DE MATERIAL

TRANSITIONS

PROGRESIVOS

BIFOCAL FLAT TOP

BIFOCAL INVISIBLE

FOTOCROMATICO

POLYCARBONATO

ORGANICO

ANTIREFLEX

ESPEJADO

BLUE-RAY

HIGH INDEX

POLARIZADO

COLOR

TOTAL 125

NOTA: Para cualquier reclamo presentar la copia verde, por pedidos doble la Optica no se hace responsable

Anexo 8: Carta de conformidad

OPTALVISION

Teléfono: 77718051

Dirección: Calacoto Calle #15 Torre Ketal

A quien corresponda,

Por la presente, **Optica Visión**, representada legalmente por Daniel Torrico Voet, manifiesta su conformidad con el sistema de administración de inventarios desarrollado, implementado y entregado por [Tu nombre o tu empresa], bajo los términos y condiciones previamente establecidos.

El sistema desarrollado cumple con los siguientes aspectos:

1. **Gestión de Inventarios:** Permite el registro, actualización y seguimiento de insumos y productos ópticos.

2. **Gestión de Proveedores:** Facilita la asociación entre productos y proveedores para garantizar trazabilidad.

3. **Gestión de Ventas:** Proporciona herramientas para el registro y análisis de ventas, asegurando un control efectivo de las transacciones.

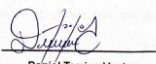
4. **Gestión de Trabajos Ópticos:** Implementa funcionalidades para registrar y monitorear el progreso de los trabajos ópticos asignados.

5. **Reportes Automatizados:** Genera reportes detallados para facilitar la toma de decisiones y el análisis de datos.

Tras realizar las pruebas pertinentes y verificar la funcionalidad del sistema, confirmamos que este cumple con nuestras expectativas y requisitos. Nos comprometemos a mantener el sistema operando bajo las condiciones establecidas y a notificar cualquier inconveniente o requerimiento adicional al desarrollador.

Sin más que agregar, agradecemos su atención y profesionalismo en la entrega del sistema.

Atentamente,



Daniel Torrico Voets

OPTALVISION