

ESCUELA MILITAR DE INGENIERÍA
"MCAL. ANTONIO JOSÉ DE SUCRE"
BOLIVIA

PROYECTO FINAL



SISTEMA DE CONTROL DE INVENTARIO Y ABASTECIMIENTO DE PRODUCTOS ÓPTICOS

Thiago Leonardo Sossa Chugar

Gabriel Camacho Alvarez

Elvin Andrés Gutiérrez

Richard Vargas Cachi

Juan Pablo Jiménez Siles

COCHABAMBA, 2024

ÍNDICE



INDICE

CONTENIDO

1.	INTRODUCCIÓN.....	1
2.	ANTECEDENTES	2
1.	PLANTEAMIENTO DEL PROBLEMA	3
2.1.	Identificación de los escenarios operacionales.....	3
2.2.	Identificación del problema	4
2.3.	Formulación del problema	5
3.	OBJETIVOS	5
3.1.	Objetivo general	5
3.2.	Objetivos específicos.....	6
4.	JUSTIFICACIÓN	7
5.	ALCANCE.....	7
6.	INGENIERÍA DEL PROYECTO.....	8
6.1.	Identificación de necesidades.....	8
6.2.	Identificación de los requisitos.....	9
6.3.	Caracterización de la solución.....	11
6.4.	Evaluación de las soluciones.....	12
6.4.1.	Valoración de soluciones candidatas	12
6.4.2.	Especificación de la solución identificada.....	14
6.5.	Matriz de Requisitos del sistema.	17
6.6.	Diagramas.	25
6.6.1.	Diagramas FFBD.....	25
6.6.2.	Casos de Uso	28

6.6.3.	EDT del Sistema.....	32
6.7.	Diseño conceptual de la Base de Datos.	33
6.8.	Diseño Lógico de la Base de Datos.....	34
6.9.	Diseño Físico de la Base de Datos.....	34
6.10.	DML de Inserción de datos a las diferentes tablas de la Base de Datos.	39
6.11.	DML de Selección a las diferentes tablas de la Base de Datos.....	41
6.12.	DML de Actualización a las diferentes tablas de la Base de Datos.	41
6.13.	Selección de la(s) estructuras de datos del sistema.....	42
6.13.1.	Descripción de clases del sistema.....	42
6.13.2.	Descripción de método de la clase.....	45
6.13.3.	Relación de clase.	46
6.14.	Implementación.	47
6.14.1.	Interfaces de entrada.....	47
6.14.2.	Interfaces de salida.	50
6.14.3.	Código del sistema	53
6.15.	Aplicar los elementos fundamentales de la oferta y la demanda de bienes y servicios a los hechos y fenómenos económicos vigentes en el país que permita formular sus leyes y comprender sus resultados.	56
6.15.1.	Análisis de mercado	56
6.15.2.	Costos	57
6.15.3.	Rotación	58
6.16.	Determinar el equilibrio microeconómico entre los componentes de ingresos y gastos de una entidad económica.	58
6.16.1.	Cantidad Económica de Pedido	58
6.16.2.	Proyección económica	59

6.16.3.	Indicadores económicos	59
6.17.	Establecer la sensibilidad de la permanencia de las cantidades por adquirir o producir ante una elevación o disminución del precio de los bienes o servicios básicos. 59	
6.17.1.	Análisis	59
6.17.2.	Adaptabilidad	60
6.17.3.	Elasticidad	60
7.	COCLUSIONES Y RECOMENDACIONES	60
8.	BIBLIOGRAFÍA.....	60
	ANEXOS	1

ÍNDICE DE TABLAS

Tabla 1:Identificación de los escenarios operacionales	3
Tabla 2:Identificación de las necesidades	8
Tabla 3:Requisitos funcionales.....	9
Tabla 4:Criterios de evaluación	13
Tabla 5: Requerimientos técnicos	16
Tabla 6:Tipo para la matriz de requisitos	17
Tabla 7: RF-001	17
Tabla 8:RF-002	19
Tabla 9:RF-003	20
Tabla 10:RF-004	22
Tabla 11:RF-005	23
Tabla 12:Productos	42
Tabla 13:Clasificacion	42
Tabla 14:Categorías	43
Tabla 15:Producto Categoría	43
Tabla 16:Locales	43
Tabla 17:Inventario local	43

Tabla 18:Proveedores	43
Tabla 19:Movimiento inventario.....	44
Tabla 20:Usuarios	44
Tabla 21:Roles	44
Tabla 22:Controladores	44
Tabla 23:Permisos Rol	45
Tabla 24:Árbol Rojo negro	45
Tabla 25:Pila	45
Tabla 26:Cola	46
Tabla 27:Lista	46

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Árbol de problemas	5
Ilustración 2:Diagrama FFBD General	25
Ilustración 3:Diagrama FFBD de RF-001	26
Ilustración 4:Diagrama FFBD de RF-002	26
Ilustración 5:Diagrama FFBD de RF-003	27
Ilustración 6:Diagrama FFBD de RF-003	27
Ilustración 7:Diagrama FFBD de RF-003	28
Ilustración 8:Casos de uso general	28
Ilustración 9:Casos de uso de RF-001	29
Ilustración 10: Casos de uso de RF-002	29
Ilustración 11: Casos de uso de RF-003	30
Ilustración 12: Casos de uso de RF-004	30
Ilustración 13: Casos de uso de RF-005	31
Ilustración 14:EDT del proceso de producción de la óptica Tokio	32
Ilustración 15:Diseño conceptual de base de datos	33
Ilustración 16:Diseño lógico de la base de datos	34
Ilustración 17:Diagrama de UML	47

Ilustración 18:Interfaz de proveedores	47
Ilustración 19:Interfaz de Productos	48
Ilustración 20:Interfaz de entradas	49
Ilustración 21:Interfaz para añadir usuarios	49
Ilustración 22:Interfaz de login.....	50
Ilustración 23:Interfaz de Home.....	50
Ilustración 24:Interfaz de Categorías.....	51
Ilustración 25:Interfaz de salidas	51
Ilustración 26:Interfaz de Roles	52
Ilustración 27:Interfaz de Usuarios	52
Ilustración 28:Interfaz de About.....	53

ÍNDICE DE ECUACIONES

Ecuación 1: Modelo EOQ	57
Ecuación 2: Rotación del inventario	58
Ecuación 3: Modelo EOQ	58
Ecuación 4: Modelo EOQ	59

ÍNDICE DE ANEXOS

Anexo “A”: Repositorio de GitHub donde se esta trabajando	1
---	---

1. INTRODUCCIÓN

En el dinámico entorno empresarial actual, las organizaciones enfrentan desafíos constantes para mantenerse competitivas y eficientes. En este contexto, la tecnología se convierte en una herramienta fundamental para optimizar procesos y garantizar la satisfacción del cliente. La Óptica Tokio, fundada en 2001 en Cochabamba y con sucursales en La Paz, ha consolidado su posición como una de las empresas líderes en el sector óptico en Bolivia. Sin embargo, el crecimiento y la diversificación de sus operaciones han evidenciado limitaciones en su sistema actual de gestión de inventarios, el cual opera de manera manual, generando ineficiencias, errores y costos innecesarios.

El presente proyecto tiene como propósito principal desarrollar un sistema automatizado de gestión de inventarios que permita a la Óptica Tokio modernizar sus procesos y responder eficazmente a los desafíos operativos. Este sistema estará diseñado para abordar necesidades críticas, como el registro y control de productos, la gestión de proveedores y la planificación de inventarios. Para ello, el proyecto se estructura sobre una infraestructura tecnológica centralizada, donde se instalará el software que gestionará la información de toda la empresa. Desde esta infraestructura central, el sistema transmitirá datos a los diferentes locales de la Óptica Tokio, garantizando una administración coherente y bien organizada en cada una de sus sucursales.

Esta infraestructura central permitirá no solo una mejor coordinación entre las áreas de la empresa, sino también un seguimiento detallado de las operaciones, eliminando errores manuales y optimizando los recursos disponibles. El sistema incluirá funcionalidades clave, como la automatización del registro de productos, la creación de módulos de gestión de proveedores y auditorías periódicas de inventarios, lo que garantizará un flujo de trabajo más eficiente y preciso. Asimismo, su diseño incluirá herramientas que permitan el análisis y la generación de reportes, facilitando la toma de decisiones estratégicas basadas en datos confiables.

El impacto esperado de este proyecto trasciende la simple mejora de los procesos internos. Su implementación contribuirá a reducir costos operativos, mejorar los tiempos de atención al cliente y fortalecer la posición competitiva de la empresa en el mercado. Además, el enfoque en la centralización de la gestión de inventarios asegura que los beneficios del sistema se distribuyan uniformemente entre los diferentes locales, promoviendo la uniformidad y estandarización en las operaciones de la Óptica Tokio.

En suma, el desarrollo de este sistema representa un paso estratégico hacia la modernización de la Óptica Tokio, posicionándola como una empresa eficiente, confiable y preparada para los retos del futuro en un mercado altamente exigente. Este proyecto no solo optimizará las operaciones actuales, sino que también sentará las bases para un crecimiento sostenible, mejorando la experiencia tanto de los clientes como del personal administrativo y operativo.

2. ANTECEDENTES

La Óptica “Tokio” inicia sus actividades en 2001 en la ciudad de Cochabamba brindados servicios de venta de lentes con medición, posteriormente amplía sus actividades a ofreciendo servicios de mediciones en sus instalaciones (2011). Actualmente cuenta con una gran cartera de clientes y dos sucursales en la ciudad de Cochabamba, una en la ciudad de La Paz, siendo una de las empresas más importantes en este rubro.

Los servicios que ofrece son los siguientes:

- Venta de lentes de sol.
- Venta de lentes con medida.
- Mediciones personalizadas.

La empresa está estructurada de la siguiente manera:

- Área de producción: responsable de la fabricación de los lentes y su ensamblado con la montura respectiva.
- Área de Almacenes: responsable del control, pedido y despacho de materiales e insumos.
- Área de Optometría: responsable de la medición y formulación medica de los lentes.
- Área administrativa: responsable de la gestión de recursos necesarios para la operación de la empresa.

Actualmente, la gestión de procesos en Óptica Tokio es manual, incluyendo la administración de la información de clientes, empleados, citas, control de inventarios (almacenes), asistencia con el optómetra y facturación. Esta modalidad limita significativamente la eficiencia en el control de activos y la administración de inventarios, dificultando la realización oportuna de pedidos de materiales e insumos. Como resultado, aumenta el tiempo de atención de a los clientes y se genera una experiencia de servicio deficiente, lo que puede llevar a que los clientes opten por la competencia. Además, la empresa carece de un registro actualizado de proveedores, lo cual complica la reposición de insumos y materiales en el momento adecuado.

1. PLANTEAMIENTO DEL PROBLEMA

2.1. Identificación de los escenarios operacionales

Tabla 1:Identificación de los escenarios operacionales

CODIGO	NOMBRE	DESCRIPCIÓN
--------	--------	-------------

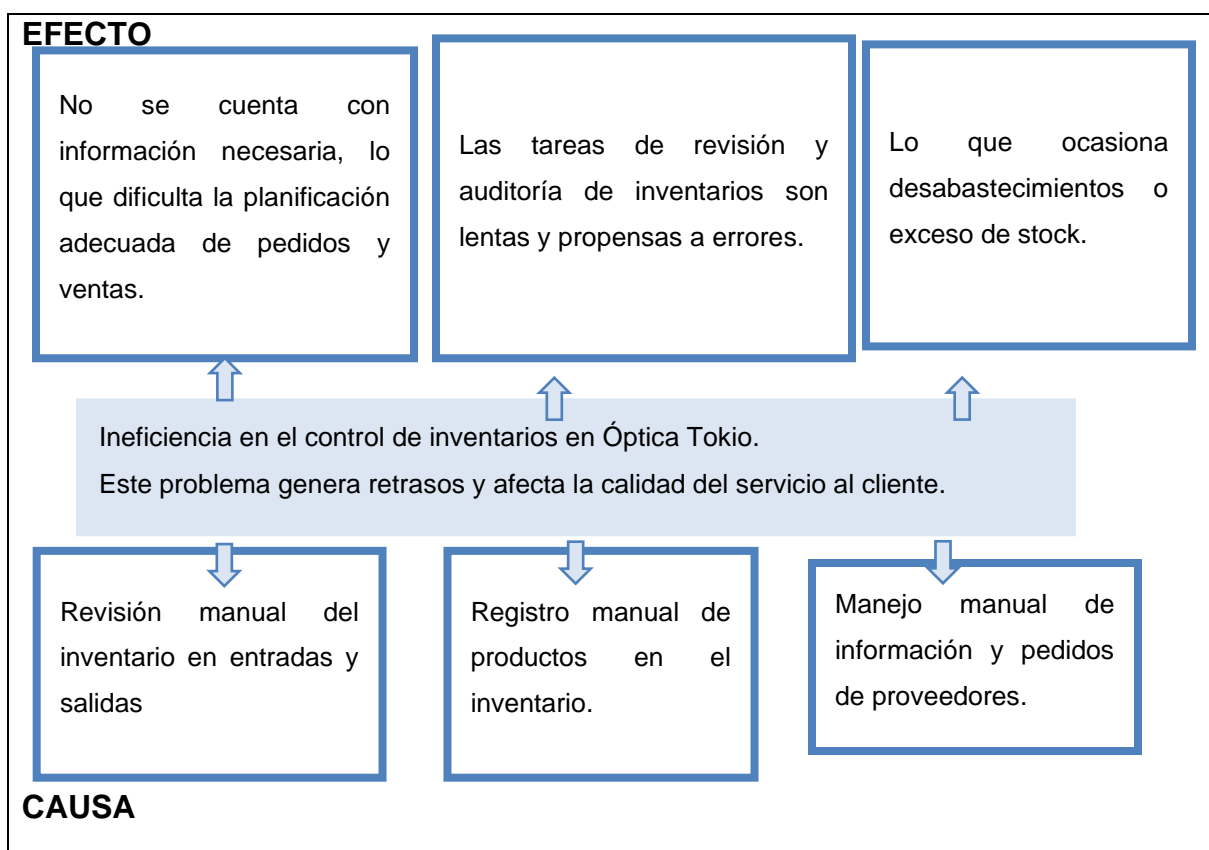
EO-001	Recepción de nuevos productos	Recibe regularmente nuevos productos. Actualmente, el registro de estos productos en el inventario se realiza de forma manual, lo que puede llevar a errores y demoras.
EO-002	Gestión de proveedores	Trabaja con varios proveedores para el abastecimiento de productos. Actualmente, la gestión de información de proveedores, como detalles de contacto, condiciones de entrega y pedidos, se realiza manualmente, lo cual puede resultar en errores en la planificación de pedidos, demoras en las entregas, y dificultades para mantener un control eficiente de los productos suministrados.
EO-003	Revisión y auditoría de inventarios	Realiza revisiones periódicas de su inventario para asegurar que las existencias coincidan con los registros. Este proceso es manual y consume tiempo, con riesgo de discrepancias debido a errores humanos. Además, la gestión de entradas y salidas de productos no se registra de manera automatizada, lo que dificulta el control en tiempo real de los movimientos de inventario, generando posibles faltantes o excesos.

Fuente: Elaboración propia 2024

2.2. Identificación del problema

La identificación del problema:

Ilustración 1: Árbol de problemas



Fuente: Elaboración propia 2024

2.3. Formulación del problema

¿Cómo puede la Óptica Tokio implementar mejoras en los, para optimizar el control de inventarios, agilizar los procesos operativos y garantizar la precisión en sus registros, manteniendo así la calidad del servicio y la satisfacción del cliente en su sucursal de Cochabamba?

3. OBJETIVOS

3.1. Objetivo general

Desarrollar un sistema de gestión de inventarios que optimice el control de productos en la Óptica Tokio, mediante la actualización del stock y una gestión eficiente de

inventarios, que integre los procesos de recepción de productos, gestión de proveedores y revisión de existencias, para agilizar los procesos operativos, mantener la alta calidad de los servicios y garantizar la satisfacción del cliente.

3.2. Objetivos específicos

1. Analizar los procesos operativos actuales de Óptica Tokio para identificar áreas de mejora en la recepción de productos, gestión de proveedores y control de inventarios, con el fin de definir los requerimientos específicos del sistema.
2. Diseñar la arquitectura del sistema de gestión de inventarios, asegurando la integración de todos los componentes necesarios para automatizar los procesos operativos de la empresa.
3. Crear una interfaz de usuario intuitiva y funcional que facilite la interacción del personal administrativo y operativo con el sistema, mejorando la eficiencia en las tareas diarias.
4. Desarrollar un modelo de base de datos relacional que almacene la información de productos, proveedores, movimientos de inventario y usuarios, garantizando la integridad y accesibilidad de los datos.
5. Implementar consultas SQL optimizadas para gestionar las operaciones de inserción, actualización y recuperación de datos en el sistema de inventario.
6. Crear índices y vistas en la base de datos para mejorar el rendimiento de las consultas y permitir un acceso eficiente a la información clave del inventario.
7. Implementar estructuras de datos como listas, pilas, colas y árboles rojo-negro para optimizar el almacenamiento y acceso a la información de productos y movimientos de inventario en memoria.
8. Desarrollar métodos de búsqueda y ordenamiento utilizando las estructuras de datos seleccionadas para mejorar la eficiencia en la gestión de inventario.
9. Desarrollar el software de gestión de inventarios utilizando las estructuras de datos seleccionadas, garantizando que el sistema tenga un rendimiento óptimo y sea capaz de manejar de manera eficiente las operaciones de inventario y consultas en tiempo real.

10. Realizar un análisis de costo-beneficio de la implementación del sistema de inventario, considerando los costos asociados y los beneficios esperados en términos de reducción de errores y ahorro de tiempo.
11. Establecer el inventario ideal que permita cubrir necesidades sin realizar muchos gastos.
12. Evaluar el impacto económico del sistema en la operación de Óptica Tokio, analizando cómo la automatización de inventarios puede reducir costos y mejorar la rentabilidad de la empresa.
13. Establecer indicadores de rendimiento económico para medir la eficiencia del sistema, como la reducción de costos operativos, el aumento en la disponibilidad de productos y la mejora en la satisfacción del cliente.

4. JUSTIFICACIÓN

La implementación de un sistema de gestión de inventarios en la Óptica Tokio es esencial para optimizar sus operaciones y mejorar la precisión en el manejo de productos. Actualmente, la dependencia de procesos manuales en la recepción de productos, gestión de proveedores y control de inventarios genera errores, demoras y una falta de control en tiempo real, lo cual afecta tanto la eficiencia interna como la satisfacción del cliente. Este proyecto permitirá automatizar y estructurar estos procesos críticos, facilitando el acceso a datos actualizados, mejorando la disponibilidad de productos y reduciendo los costos operativos. Al integrar herramientas de bases de datos y estructuras de datos avanzadas, el sistema será capaz de gestionar grandes volúmenes de información de manera rápida y eficaz, proporcionando una plataforma robusta y confiable que optimizará el flujo de trabajo y contribuirá al crecimiento sostenible de la Óptica Tokio en el mercado.

5. ALCANCE

El alcance del proyecto “Sistema de Gestión de Inventarios para Óptica Tokio” se centra en el desarrollo e implementación de un sistema automatizado que optimice la

gestión de inventarios en la empresa. Este sistema abarcará desde la recepción de productos, el control de movimientos y auditorías de inventario, hasta la gestión de proveedores, permitiendo registrar y consultar información en tiempo real con mayor precisión. Además, incluirá el diseño de una base de datos relacional que garantice la integridad y accesibilidad de los datos, así como una interfaz de usuario intuitiva que facilite la interacción del personal administrativo y operativo en sus tareas diarias. El proyecto también incorporará estructuras de datos avanzadas para optimizar las búsquedas y el ordenamiento, asegurando un rendimiento eficiente. Este sistema busca reducir errores, agilizar los procesos operativos y mejorar la experiencia del cliente, posicionando a la Óptica Tokio como una empresa más competitiva y sostenible en su sector.

6. INGENIERÍA DEL PROYECTO

6.1. Identificación de necesidades

Tabla 2: Identificación de las necesidades

CODIGO	IDENTIFICADOR DE ESCENARIO	NOMBRE	DESCRIPCION	JUSTIFICACIÓN
NB-001	EO-001	Registro y trazabilidad de productos nuevos	Recibe regularmente nuevos productos que actualmente se registran manualmente. Este proceso genera riesgos de errores, retrasos en la actualización del inventario y dificultades en la trazabilidad de los productos dentro del almacén.	Es fundamental automatizar el registro y seguimiento de los productos nuevos para mejorar la eficiencia, reducir errores y asegurar la disponibilidad de información actualizada para la toma de decisiones en el inventario.

NB-002	EO-002	Optimización de la relación con proveedores	La gestión de proveedores se realiza manualmente, incluyendo el registro de detalles de contacto, condiciones de entrega y pedidos. Esto genera errores, retrasos en las entregas y dificultades para planificar y controlar el suministro de productos.	Centralizar y digitalizar la información de proveedores permitirá gestionar pedidos con mayor precisión, mejorar el cumplimiento de entregas y mantener un control eficiente del suministro de productos.
NB-003	EO-003	Automatización del control de inventarios	Las auditorías y revisiones de inventarios se realizan manualmente, lo que consume tiempo y genera discrepancias entre los registros y el inventario físico. Además, la falta de un registro automatizado de entradas y salidas dificulta el control en tiempo real.	Automatizar el control de inventarios garantizará la precisión en los registros, facilitará auditorías rápidas y reducirá el tiempo requerido para mantener el inventario actualizado, optimizando los procesos de la empresa.

Fuente: Elaboración propia 2024

6.2. Identificación de los requisitos.

Tabla 3: Requisitos funcionales

CODIGO	NOMBRE	DESCRIPCIÓN	ESCENARIO	PRIORIDAD
--------	--------	-------------	-----------	-----------

			ASOCIADO	
RF-001	Registro de nuevos productos	El sistema debe permitir el registro automatizado de productos nuevos, asociándolos con información clave como proveedor, cantidad, precio de costo y fecha de recepción.	EO-001	ALTA
RF-002	Gestión de proveedores	El sistema debe incluir un módulo para la gestión de proveedores, permitiendo registrar y actualizar datos como contacto, condiciones de entrega y pedidos realizados.	EO-002	ALTA
RF-003	Control de entradas y salidas	El sistema debe registrar las entradas y salidas de productos en tiempo real, reflejando los movimientos en el inventario.	EO-003	ALTA
RF-004	Auditorías automatizadas	El sistema debe generar informes automáticos que permitan comparar las existencias físicas con los registros, identificando discrepancias.	EO-003	MEDIA
RF-005	Reportes de inventario	El sistema debe permitir generar reportes detallados del inventario, filtrados por	Todos.	ALTA

		local, proveedor o categoría de producto.		
--	--	---	--	--

Fuente: Elaboración propia 2024

6.3. Caracterización de la solución.

Soluciones:

NECESIDAD	REQUERIMIENTO	SOLUCIÓN	
NB-001: Registro y trazabilidad de productos nuevos.	RF-001: El sistema debe permitir el registro automatizado de productos nuevos, asociándolos con información clave como proveedor, cantidad, precio de costo y fecha de recepción.	SC-01	SC-02 Continuar con el sistema actual
NB-002: Optimización de la relación con proveedores.	RF-002: El sistema debe incluir un módulo para la gestión de proveedores, permitiendo registrar y actualizar datos como contacto, condiciones de entrega y pedidos realizados. movimientos en el inventario.	Desarrollar un sistema de gestión de inventarios que optimice el control de productos en la Óptica Tokio, mediante la actualización del stock y una gestión eficiente de inventarios, que integre los procesos de recepción de productos, gestión de proveedores y revisión de existencias, para agilizar los procesos operativos, mantener la alta calidad de los servicios y garantizar la satisfacción del cliente.	
NB-003: Automatización del control de inventarios.	RF-003: El sistema debe registrar las entradas y salidas de productos en tiempo real, reflejando los		

	RF-004: El sistema debe generar informes automáticos que permitan comparar las existencias físicas con los registros, identificando disc RF-005: El sistema debe permitir generar reportes detallados del inventario, filtrados por local, proveedor o categoría de producto.		
--	--	--	--

Fuente: Elaboración propia 2024

6.4. Evaluación de las soluciones.

6.4.1. Valoración de soluciones candidatas

La evaluación de la factibilidad de cada solución se realiza de acuerdo con los siguientes aspectos:

- Costo: Evaluar si la solución propuesta se ajusta al presupuesto disponible, que no debe exceder de 1000 \$u\$.
- Tiempo: Considerar el tiempo necesario para desarrollar, implementar y capacitar al personal sobre la nueva solución.
- Tecnología: Analizar la compatibilidad tecnológica de la solución con el hardware y software existentes en la empresa.

- **Requerimientos del negocio:** Asegurarse de que la solución cumpla con los requerimientos identificados, incluyendo automatización, acceso en tiempo real y alertas automáticas.

A partir de ese análisis, se valora cada solución utilizando un sistema de puntaje. Este sistema considera los aspectos mencionados y asigna un puntaje del 1 al 5 para cada criterio, donde 1 representa la menor factibilidad y 5 la mayor factibilidad.

Tabla 4: Criterios de evaluación

SOLUCIÓN	JUSTIFICACIÓN: CRITERIOS DE EVALUACIÓN				
	COSTO	TIEMPO	TECNOLOGÍA	REQUERIMIENTOS DEL NEGOCIO	TOTAL
SC-01: Desarrollar un sistema de gestión de inventarios que optimice el control de productos en la Óptica Tokio, mediante la actualización del stock y una gestión eficiente de inventarios, que integre los procesos de recepción de productos, gestión de proveedores y revisión de existencias, para agilizar los procesos operativos, mantener la alta calidad de los servicios y garantizar la satisfacción del cliente.	3	5	3	5	16

SC-001	El costo de implementar un nuevo sistema automatizado de gestión de inventarios es considerable. La solución implica el desarrollo o adquisición de un software especializado que registre productos, actualice el stock en tiempo real y genere alertas automáticas de reabastecimiento.	El desarrollo y la implementación de un sistema automatizado toman tiempo, ya que involucran varias fases como la adquisición o desarrollo del software, la integración con los sistemas existentes, y la capacitación del personal. Sin embargo, aunque la fase de implementación puede ser prolongada, una vez que el sistema está operativo, los beneficios en cuanto a la reducción del tiempo de operación son notables.	Aunque la solución propuesta ofrece una modernización tecnológica importante, no cuenta con todas las tecnologías actualmente disponibles en el mercado.	Cumple completamente con los requerimientos del negocio establecidos al inicio del proyecto.	
SC-002: Continuar Con el sistema actual.	5	2	4	3	14
SC-002	Tiene un costo menor, no se necesita una inversión grande.	Al tener todo manual, el tiempo en los procesos aumenta.	Tiene la mayor de las tecnologías.	Cumple la mayoría de los requerimientos.	

Fuente: Elaboración propia 2024

6.4.2. Especificación de la solución identificada

La solución que obtuvo la mejor valoración y cumple con los requerimientos de Óptica Tokio es SC-001: Desarrollar un sistema de gestión de inventarios que optimice el control de productos en la Óptica Tokio, mediante la actualización del stock y una

gestión eficiente de inventarios, que integre los procesos de recepción de productos, gestión de proveedores y revisión de existencias, para agilizar los procesos operativos, mantener la alta calidad de los servicios y garantizar la satisfacción del cliente.

6.4.2.1. Consideraciones principales

Para garantizar el éxito del sistema de gestión de inventarios, se han establecido varias consideraciones fundamentales. En primer lugar, es esencial la automatización de los procesos actualmente manuales, como el registro de productos y la gestión de inventarios, para eliminar errores humanos y mejorar la eficiencia operativa. El acceso en tiempo real a los datos del inventario será otro aspecto crucial, permitiendo a los usuarios autorizados tomar decisiones informadas de manera rápida y efectiva.

El diseño del sistema deberá considerar la escalabilidad, asegurando que pueda adaptarse al crecimiento futuro de la empresa, como la apertura de nuevas sucursales o el aumento en la cartera de clientes. Asimismo, la interfaz de usuario debe ser altamente intuitiva, adaptada al nivel técnico del personal, para garantizar una adopción fluida del sistema. La seguridad de los datos también es una prioridad, por lo que se implementarán autenticación por roles y mecanismos de cifrado para proteger información crítica.

Finalmente, la optimización del rendimiento es esencial. El uso de estructuras de datos avanzadas y consultas SQL optimizadas permitirá que el sistema gestione grandes volúmenes de datos sin comprometer el tiempo de respuesta. Esto será especialmente importante para garantizar que el sistema pueda manejar operaciones en tiempo real y satisfacer las demandas operativas de Óptica Tokio.

6.4.2.2. Detalles de la solución

La solución propuesta consiste en desarrollar un sistema de gestión de inventarios robusto y eficiente para Óptica Tokio, centrado en la automatización de procesos críticos. Este sistema incluirá un módulo de recepción de productos que permitirá registrar automáticamente los nuevos productos con información detallada, como

cantidad, proveedor, precio de costo y fecha de recepción. Además, se integrará un módulo de gestión de proveedores, el cual centralizará todos los datos relevantes de los proveedores, incluyendo contactos, condiciones de entrega y pedidos realizados. Esto permitirá una planificación más precisa y una relación más efectiva con los proveedores.

El módulo de control de inventarios será clave para el registro en tiempo real de entradas y salidas de productos, lo que facilitará la supervisión de los movimientos de inventario en cada sucursal. Asimismo, el sistema generará auditorías automáticas, comparando las existencias físicas con los registros digitales para identificar y corregir discrepancias rápidamente. Para facilitar el análisis y la toma de decisiones, se incluirá un módulo de reportes y estadísticas, capaz de generar informes detallados y personalizables que proporcionen una visión clara del desempeño del inventario, tanto por categorías de producto como por sucursales o proveedores.

El sistema se apoyará en estructuras de datos avanzadas, como árboles rojo-negro, listas, pilas y colas, para optimizar el almacenamiento y la recuperación de información en memoria. Esto asegurará un rendimiento óptimo en operaciones críticas como búsquedas y ordenamientos. La arquitectura del sistema será diseñada utilizando C# con Windows Forms para una interfaz amigable y PostgreSQL como base de datos relacional para garantizar la integridad y accesibilidad de los datos.

6.4.2.3. Requerimientos técnicos

Tabla 5: Requerimientos técnicos

Hardware:	Software:	Herramientas recomendadas:
Procesador: Intel i5 o superior. Memoria RAM: 8 GB mínimo. Almacenamiento: 250 GB (SSD recomendado para mejor rendimiento).	Lenguaje de programación: C#. Entorno de desarrollo: Visual Studio. Sistema operativo: Windows 10 o superior. Sistema de base de datos: PostgreSQL.	IDE: Visual Studio 2022. Sistema de control de versiones: Git. Librerías de C#

Monitor: Resolución mínima de 1920x1080.	Librerías necesarias: .NET Framework 4.7 o superior.	
--	--	--

Fuente: Elaboración propia 2024

6.5. Matriz de Requisitos del sistema.

La matriz de requisitos:

Tabla 6:Tipo para la matriz de requisitos

TIPO	
RI	Requerimiento Inmediato
RNI	Requerimiento No Inmediato
RD	Requerimiento Deseable

Fuente: Elaboración propia 2024

Tabla 7: RF-001

Gestión de Productos y Almacenes	RF-001	DEBE PERMITIR el registro de nuevos productos.	El sistema debe permitir el registro automatizado de productos nuevos, asociándolos con información clave como proveedor, cantidad recibida, precio de costo y fecha de recepción. Esto garantizará una trazabilidad precisa y reducirá errores en el inventario.	RI
	1.1	DEBE PERMITIR la captura de datos básicos del producto.	El sistema debe permitir la entrada de información básica como nombre del producto, categoría, proveedor, y cantidad inicial. Esto garantizará que cada producto esté correctamente identificado.	RI

	1.2	DEBE PERMITIR registrar información avanzada del producto.	El sistema debe incluir campos para detalles adicionales como fecha de fabricación, fecha de caducidad, especificaciones técnicas o código de barras, facilitando la identificación y organización del inventario.	RNI
	1.3	DEBE PERMITIR vincular productos con proveedores.	El sistema debe permitir asociar cada producto registrado con un proveedor específico para garantizar la trazabilidad del suministro.	RI
	1.4	DEBE GENERAR un número de identificación único para cada producto.	El sistema debe asignar automáticamente un ID único para cada producto registrado, facilitando búsquedas y consultas rápidas en el inventario.	RI
	1.5	DEBE ALERTAR sobre productos duplicados.	El sistema debe verificar que no existan productos duplicados antes de completar el registro, reduciendo errores en el inventario.	RI
	1.6	DEBE PERMITIR registrar la ubicación de almacenamiento inicial.	El sistema debe incluir un campo para definir el lugar de almacenamiento inicial de los productos, como un almacén específico o una sección determinada.	RNI
	1.7	DEBE OFRECER opciones de importación de datos.	El sistema debe permitir la importación masiva de datos de productos desde archivos Excel o CSV para agilizar el registro inicial en caso de grandes volúmenes de inventario.	RI
	1.8	DEBE INTEGRARSE con el módulo de auditorías.	El sistema debe registrar automáticamente los productos nuevos en el módulo de auditorías para que estas	RI

			transacciones puedan ser verificadas posteriormente.	
--	--	--	--	--

Fuente: Elaboración propia 2024

Tabla 8:RF-002

Gestión de Proveedores	RF-002	Diseñar un Sistema de Información para administrar todos los proveedores y así no tener alteraciones en los pedidos	Un sistema que permita al usuario cuantos productos debe de pedir para no tener un exceso o escasez.	RI
	2.1	DEBE PERMITIR el registro de nuevos proveedores.	El sistema debe incluir un módulo para registrar proveedores con información clave como nombre, dirección, contacto, condiciones de entrega y productos suministrados.	RI
	2.2	DEBE PERMITIR la actualización de la información de proveedores.	El sistema debe proporcionar una opción para modificar datos existentes de los proveedores, asegurando que toda la información esté actualizada para una gestión eficiente.	RI
	2.3	DEBE GENERAR reportes sobre el historial de órdenes con cada proveedor.	El sistema debe incluir un registro histórico de todas las órdenes realizadas a cada proveedor, incluyendo fechas, costos y productos solicitados, para facilitar decisiones de compra.	RNI
	2.4	DEBE PERMITIR evaluar el rendimiento de los proveedores.	El sistema debe ofrecer métricas que analicen la confiabilidad y puntualidad de los proveedores basándose en entregas anteriores, para optimizar	RI

			futuras relaciones comerciales.	
	2.5	DEBE PERMITIR gestionar pedidos automáticamente según el inventario.	El sistema debe integrar algoritmos que calculen las necesidades de reabastecimiento y generen automáticamente pedidos a los proveedores, evitando faltantes o exceso de productos.	RI
	2.6	DEBE PERMITIR enviar notificaciones automáticas a proveedores.	El sistema debe contar con una función que envíe notificaciones a los proveedores confirmando pedidos, cambios o cancelaciones, para garantizar una comunicación eficiente.	RI
	2.7	DEBE INTEGRARSE con el módulo de inventarios para sincronización en tiempo real.	El sistema debe estar vinculado con el módulo de inventarios, de manera que las actualizaciones en los pedidos o en el estado de inventario se reflejen inmediatamente en el sistema de proveedores.	RI

Fuente: Elaboración propia 2024

Tabla 9:RF-003

Entradas y salidas de productos .	RF-003	DEBE REGISTRAR las entradas y salidas de productos.	El sistema debe registrar en tiempo real los movimientos de productos, incluyendo entradas por recepción de proveedores y salidas hacia las sucursales. Cada transacción debe incluir datos como cantidad, fecha y destino. Esto permitirá un control preciso del inventario.	RI
	3.1	DEBE REGISTRAR las entradas de productos	El sistema debe capturar automáticamente la información de cada entrada de productos, incluyendo el proveedor, cantidad recibida,	RI

		recibidos de proveedores.	fecha de recepción y detalles adicionales como lote y ubicación en el almacén.	
	3.2	DEBE REGISTRAR las salidas de productos hacia sucursales.	El sistema debe registrar todas las transacciones de salida de productos desde el almacén central hacia las sucursales, especificando cantidades enviadas, fecha y destino.	RI
	3.3	DEBE PERMITIR rastrear el historial de movimientos de productos.	El sistema debe incluir una funcionalidad para consultar el historial completo de entradas y salidas de un producto específico, permitiendo identificar fechas, cantidades y ubicaciones relacionadas.	RI
	3.4	DEBE PERMITIR registrar devoluciones de productos.	El sistema debe ser capaz de registrar devoluciones de productos tanto desde proveedores como desde sucursales, especificando las razones de la devolución y ajustando automáticamente el inventario.	RI
	3.5	DEBE GENERAR notificaciones automáticas para movimientos excepcionales	El sistema debe enviar alertas automáticas cuando se detecten movimientos no programados o que superen ciertos umbrales predefinidos, garantizando un monitoreo constante.	RNI
	3.6	DEBE GENERAR reportes de movimiento de inventario.	El sistema debe generar reportes detallados y gráficos que muestren las tendencias de entradas y salidas de productos, permitiendo una mejor planificación del inventario.	RI
	3.7	DEBE INTEGRARSE con el módulo de auditoría de inventario.	El sistema debe estar vinculado al módulo de auditoría para garantizar que todos los movimientos de productos registrados puedan ser comparados con las existencias físicas durante auditorías programadas.	RI

Fuente: Elaboración propia 2024

Tabla 10:RF-004

Logística interna	RF-004	DEBE GENERAR auditorías automatizadas.	El sistema debe generar informes automáticos que comparen las existencias físicas con los registros digitales del inventario. Esto identificará discrepancias y garantizará la exactitud de los datos, facilitando auditorías periódicas.	RNI
	4.1	DEBE GENERAR informes de discrepancias entre inventario físico y digital.	El sistema debe proporcionar un reporte detallado que resuma todas las diferencias detectadas entre las existencias físicas y los registros digitales, indicando posibles causas y ubicaciones específicas.	RNI
	4.2	DEBE PERMITIR la programación de auditorías automáticas.	El sistema debe incluir una funcionalidad para programar auditorías en intervalos específicos, ejecutándose automáticamente y notificando a los administradores con los resultados.	RNI
	4.3	DEBE PERMITIR visualizar los resultados de auditorías en tiempo real.	Los administradores deben poder consultar en tiempo real los resultados de auditorías en curso, incluyendo indicadores visuales de discrepancias críticas o recurrentes.	RI
	4.4	DEBE GENERAR recomendaciones basadas en las auditorías.	El sistema debe incluir un módulo que sugiera acciones correctivas basadas en las discrepancias detectadas, como ajustes de inventario, cambios en procesos o	RD

			capacitaciones para el personal.	
	4.5	DEBE GENERAR un historial de auditorías.	El sistema debe almacenar los resultados de todas las auditorías realizadas, permitiendo acceder a informes históricos para identificar patrones o tendencias a lo largo del tiempo.	RI

Fuente: Elaboración propia 2024

Tabla 11:RF-005

Reportes de inventario	RF-005	DEBE PERMITIR la generación de reportes de inventario.	El sistema debe permitir generar reportes detallados del inventario, filtrados por local, proveedor, categoría de producto o estado del inventario. Estos reportes deben ser exportables en formatos como PDF y Excel para facilitar su análisis.	RI
	5.1	DEBE PERMITIR filtrar los reportes por local.	El sistema debe incluir una funcionalidad que permita generar reportes específicos para cada sucursal o local, detallando la disponibilidad y movimiento de productos.	RI

	5.2	DEBE PERMITIR filtrar los reportes por proveedor.	El sistema debe permitir generar reportes que agrupen y analicen los productos suministrados por cada proveedor, incluyendo datos como frecuencia de entrega y volumen suministrado.	RI
	5.3	DEBE PERMITIR filtrar los reportes por categoría de producto.	El sistema debe ofrecer la opción de generar reportes específicos para diferentes categorías de productos, como lentes de contacto, gafas de sol, accesorios, etc., permitiendo un análisis detallado.	RI
	5.4	DEBE EXPORTAR reportes en formatos PDF.	El sistema debe permitir a los usuarios exportar los reportes generados en formatos PDF y Excel para facilitar su análisis, distribución y almacenamiento.	RI

	5.5	DEBE INCLUIR indicadores gráficos en los reportes.	El sistema debe integrar gráficos e indicadores visuales en los reportes, como barras, líneas o tortas, para representar tendencias de inventario y facilitar la interpretación de datos.	RN I
--	-----	--	---	---------

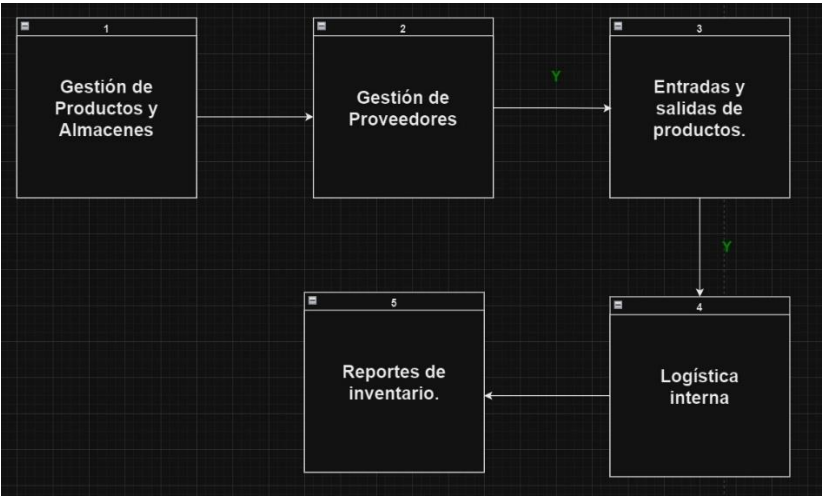
Fuente: Elaboración propia 2024

6.6. Diagramas.

Los diagramas:

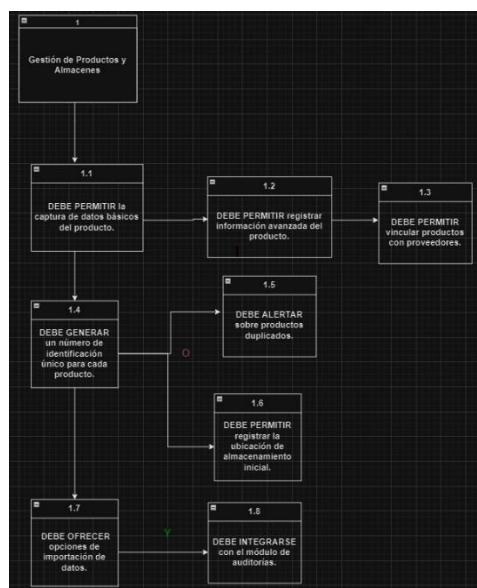
6.6.1. Diagramas FFBD

Ilustración 2:Diagrama FFBD General



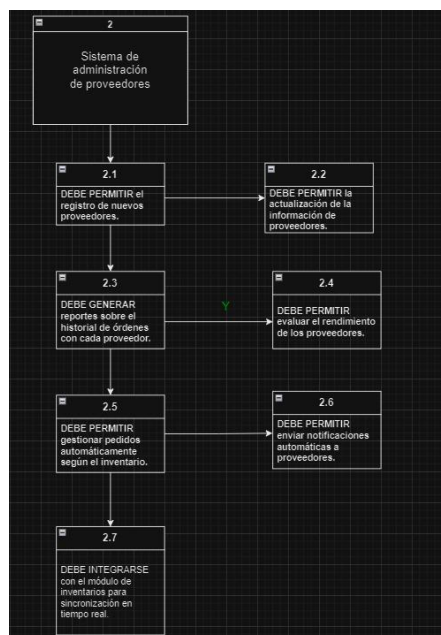
Fuente: Elaboración propia 2024

Ilustración 3:Diagrama FFBD de RF-001



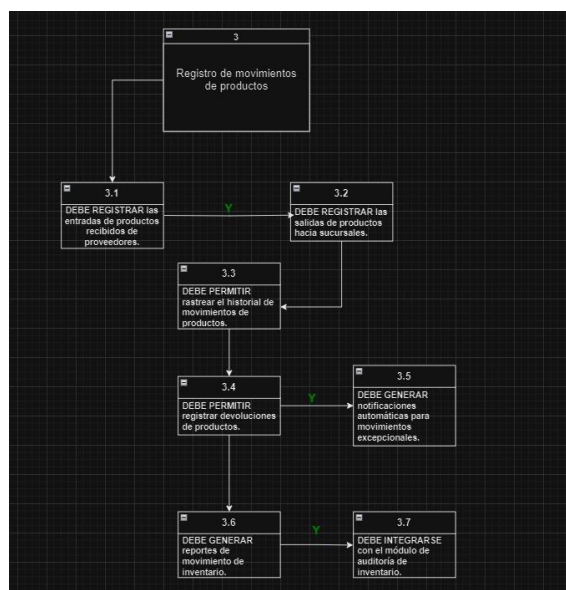
Fuente: Elaboración propia 2024

Ilustración 4:Diagrama FFBD de RF-002



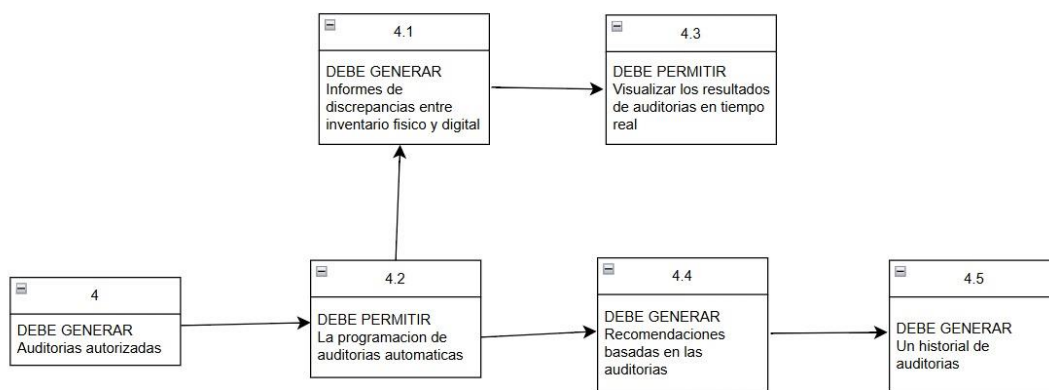
Fuente: Elaboración propia 2024

Ilustración 5:Diagrama FFBD de RF-003



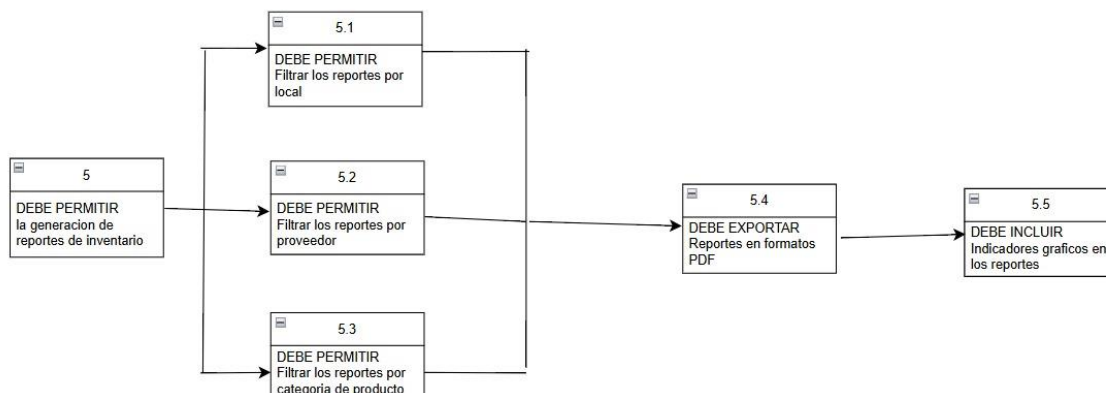
Fuente: Elaboración propia 2024

Ilustración 6:Diagrama FFBD de RF-003



Fuente: Elaboración propia 2024

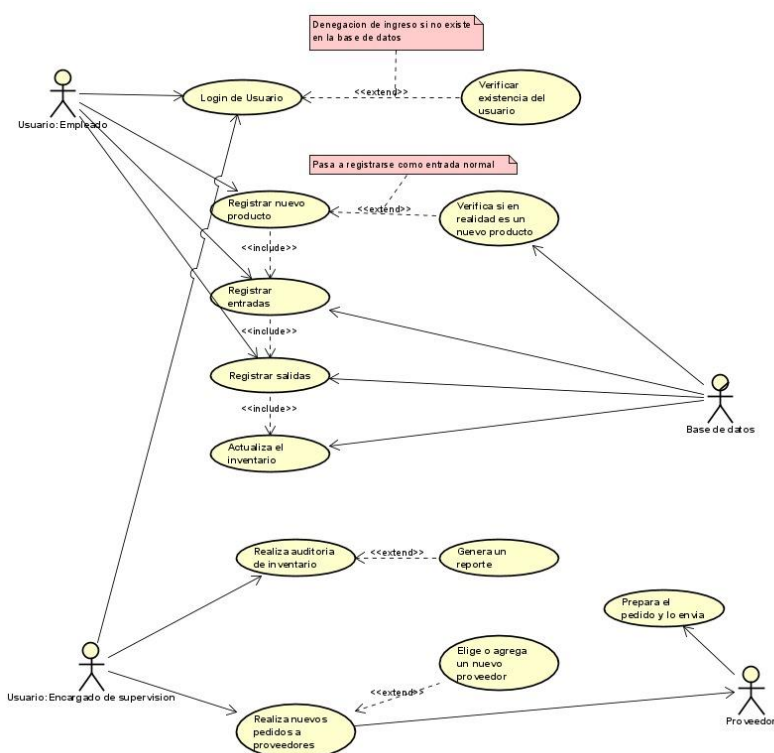
Ilustración 7:Diagrama FFBD de RF-003



Fuente: Elaboración propia 2024

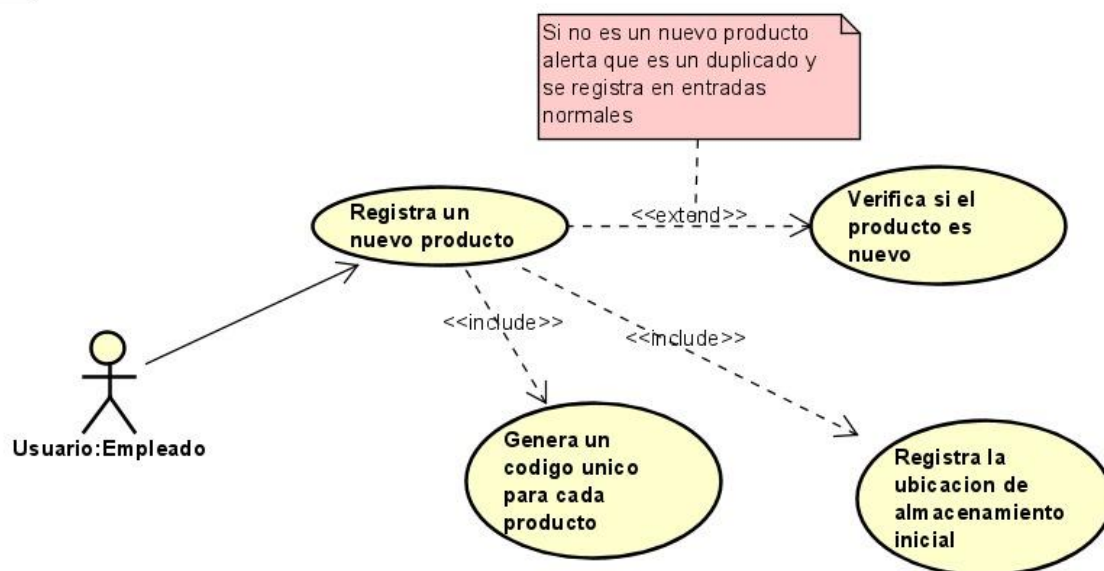
6.6.2. Casos de Uso

Ilustración 8:Casos de uso general



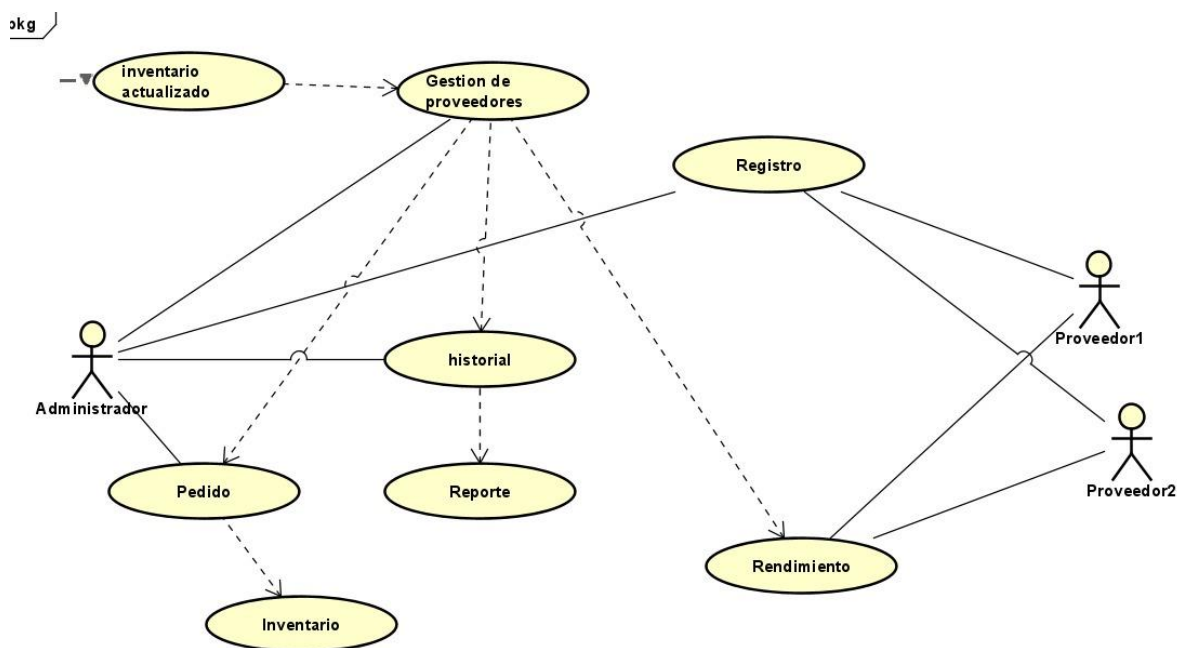
Fuente: Elaboración propia 2024

Ilustración 9: Casos de uso de RF-001



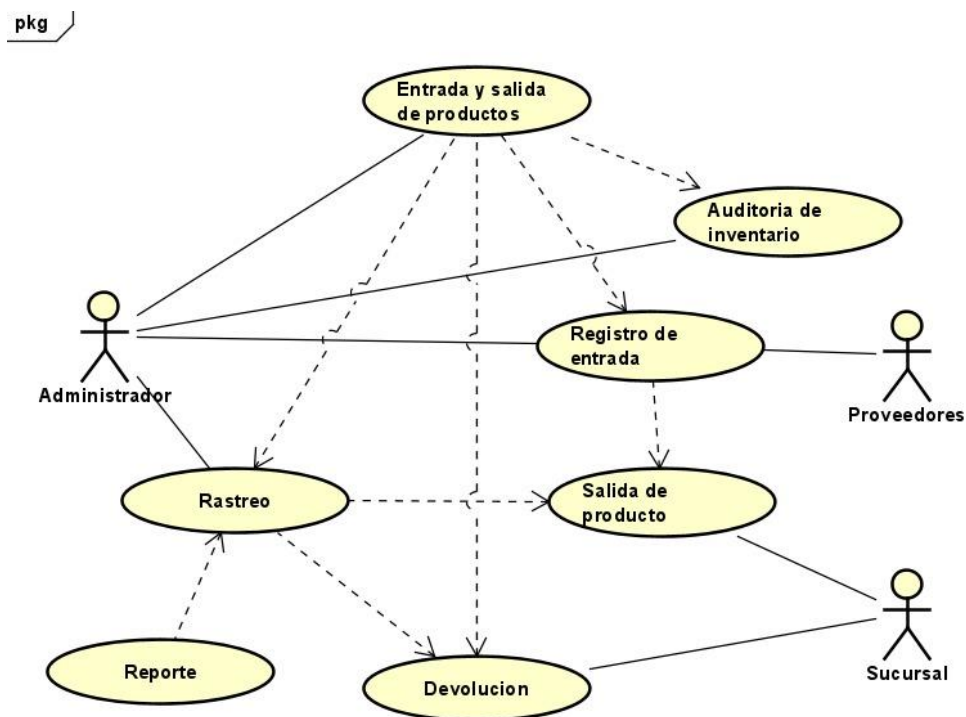
Fuente: Elaboración propia 2024

Ilustración 10: Casos de uso de RF-002



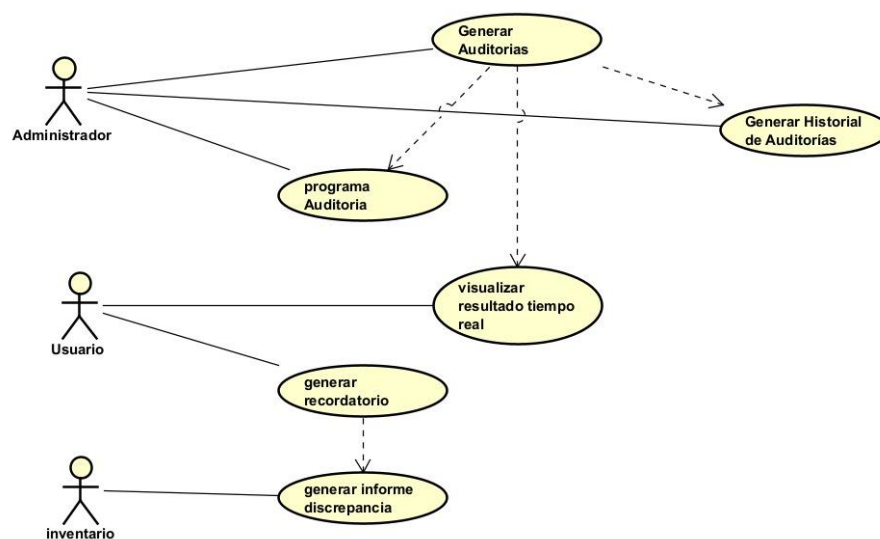
Fuente: Elaboración propia 2024

Ilustración 11: Casos de uso de RF-003



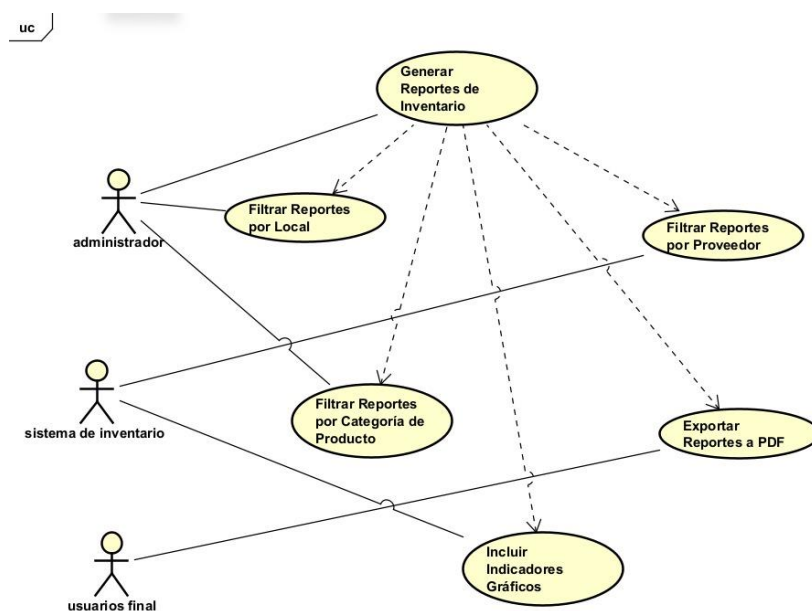
Fuente: Elaboración propia 2024

Ilustración 12: Casos de uso de RF-004



Fuente: Elaboración propia 2024

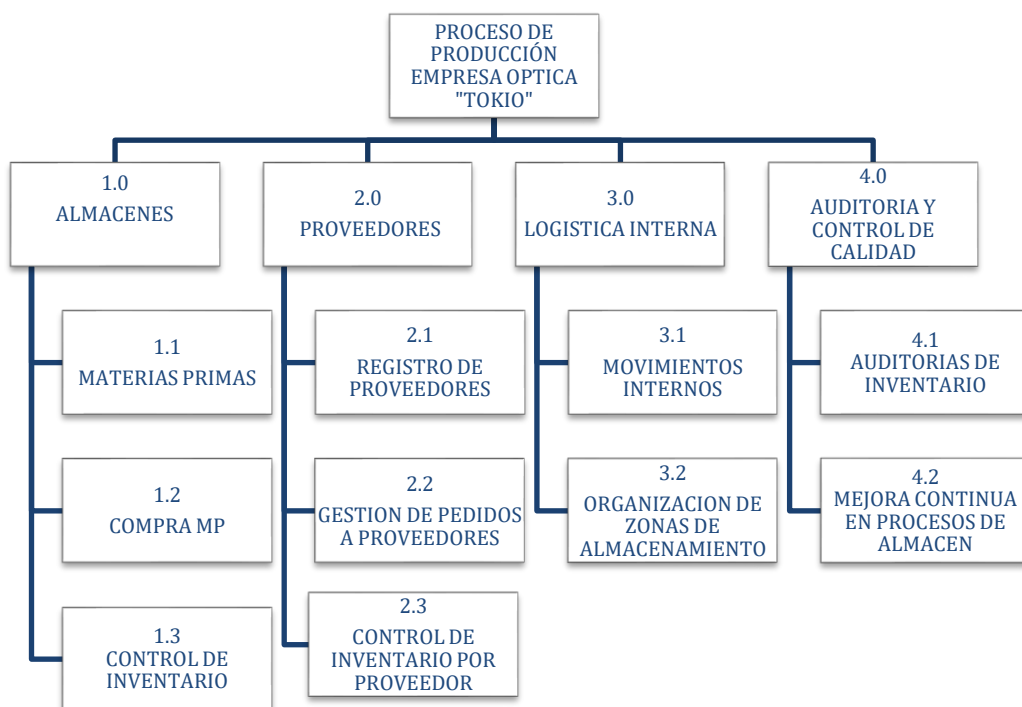
Ilustración 13: Casos de uso de RF-005



Fuente: Elaboración propia 2024

6.6.3. EDT del Sistema

Ilustración 14: EDT del proceso de producción de la óptica Tokio

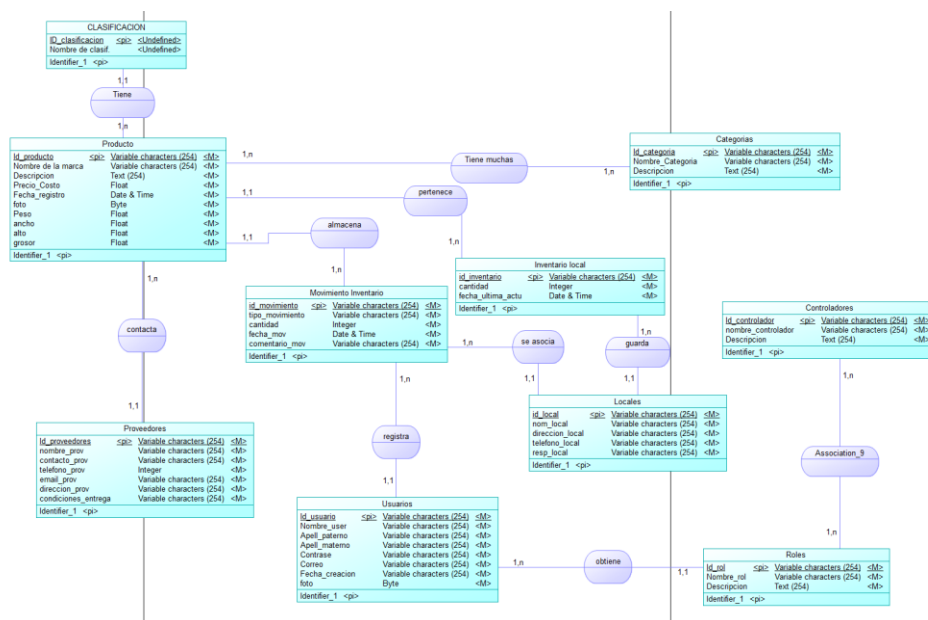


Fuente: Elaboración propia 2024

1. **Almacenes:** Incluye la preparación de las materias primas, control de inventario ,etc.
2. **Proveedores:** Control y registros de los proveedores con respecto a los pedidos.
3. **Logística interna:** Administración de espacios, preparación de pedidos y movimientos internos.
4. **Auditoría y control de calidad:** Control de inventarios con auditorias, control de calidad de productos y siempre con mejoras continuas.

6.7. Diseño conceptual de la Base de Datos.

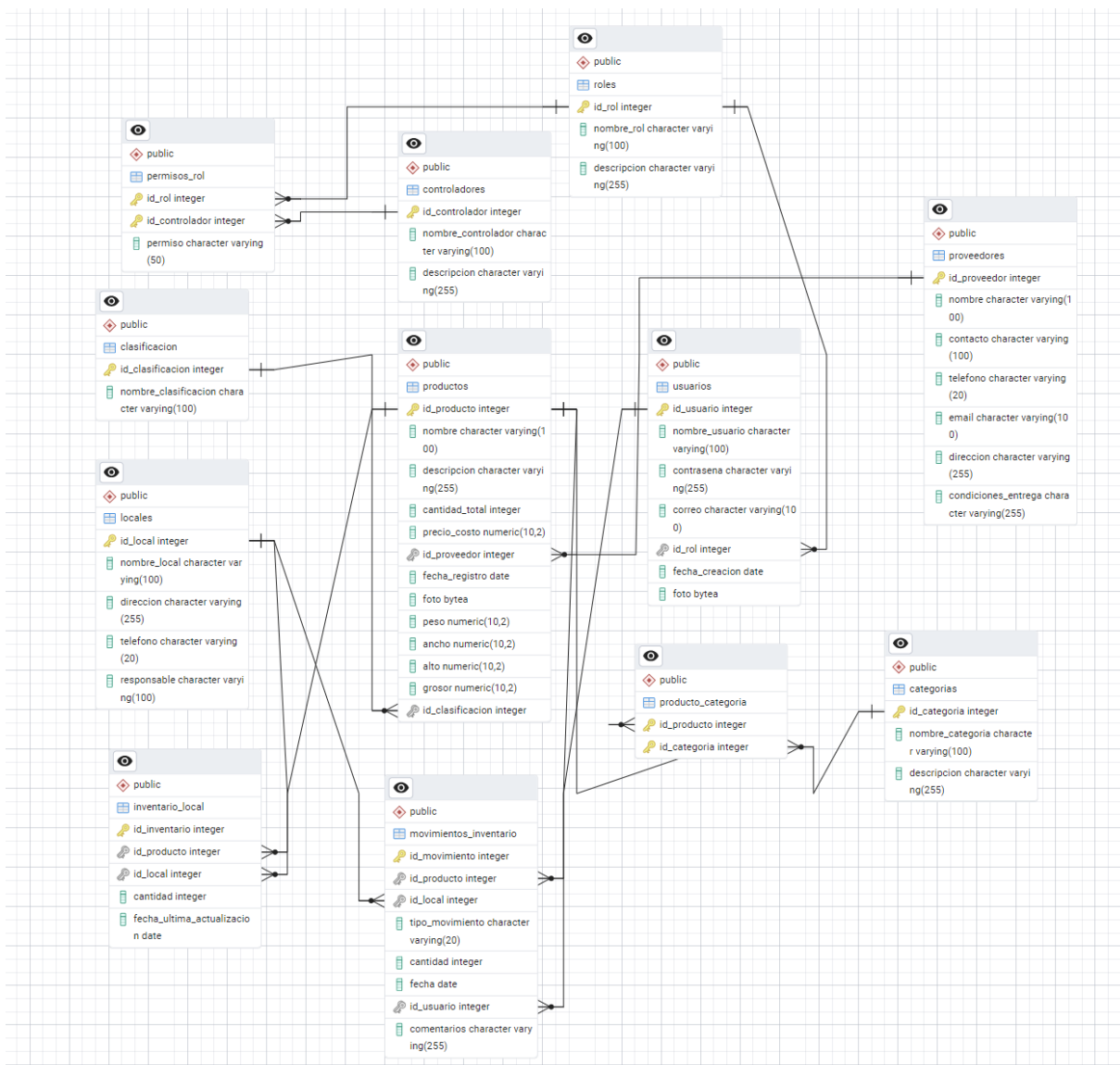
Ilustración 15: Diseño conceptual de base de datos



Fuente: Elaboración propia 2024

6.8. Diseño Lógico de la Base de Datos.

Ilustración 16:Diseño lógico de la base de datos



Fuente: Elaboración propia 2024

6.9. Diseño Físico de la Base de Datos.

BEGIN;

```
CREATE TABLE movimientos_inventario(  
  
    id_movimiento VARCHAR(30) PRIMARY KEY,  
  
    id_producto VARCHAR(30),  
  
    id_local VARCHAR(30),  
  
    tipo_movimiento VARCHAR(50) CHECK (tipo_movimiento IN ('entrada', 'salida')),  
  
    cantidad INT NOT NULL,  
  
    fecha_movimiento TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
    id_usuario VARCHAR(30),  
  
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto),  
  
    FOREIGN KEY (id_local) REFERENCES locales(id_local),  
  
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
  
);
```

```
CREATE TABLE inventario_local(  
  
    id_inventario VARCHAR(30) PRIMARY KEY,  
  
    id_producto VARCHAR(30),  
  
    id_local VARCHAR(30),  
  
    cantidad INT NOT NULL,  
  
    fecha_ultima_actualizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
FOREIGN KEY (id_producto) REFERENCES productos(id_producto),

FOREIGN KEY (id_local) REFERENCES locales(id_local)

);

CREATE TABLE producto_categoria(

    id_producto VARCHAR(30) REFERENCES productos(id_producto),

    id_categoria VARCHAR(30) REFERENCES categorias(id_categoria),

    PRIMARY KEY (id_categoria, id_producto)

);

CREATE TABLE categorias(

    id_categoria VARCHAR(30) PRIMARY KEY,

    nombre_categoria VARCHAR(255) UNIQUE NOT NULL,

    descripcion_categoria TEXT

);

CREATE TABLE locales(

    id_local VARCHAR(30) PRIMARY KEY,

    nombre_local VARCHAR(255) NOT NULL,

    direccion_local VARCHAR(255),

    telefono_local VARCHAR(20),
```

```
responsable_local VARCHAR(255)

);

CREATE TABLE productos(

    id_producto VARCHAR(30) PRIMARY KEY,

    nombre_producto VARCHAR(100) NOT NULL,

    descripcion_producto TEXT,

    cantidad_total INT NOT NULL,

    precio_costo DECIMAL NOT NULL,

    id_proveedor VARCHAR(30),

    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (id_proveedor) REFERENCES proveedores(id_proveedor)

);
```

```
CREATE TABLE proveedores(

    id_proveedor VARCHAR(30) PRIMARY KEY,

    nombre_proveedor VARCHAR(255) NOT NULL,

    contacto_proveedor VARCHAR(255),

    telefono_proveedor VARCHAR(20),

    email_proveedor VARCHAR(255),
```



```

    direccion_proveedor VARCHAR(255),

    condiciones_entrega TEXT

);

CREATE TABLE roles(

    id_rol VARCHAR(30) PRIMARY KEY,

    nombre_rol VARCHAR(255) UNIQUE NOT NULL,

    descripcion_rol TEXT

);

CREATE TABLE controladores(

    id_controlador VARCHAR(30) PRIMARY KEY,

    nombre_controlador VARCHAR(255) UNIQUE NOT NULL,

    descripcion_controlador TEXT

);

CREATE TABLE permisos_rol(

    id_rol VARCHAR(30) REFERENCES roles(id_rol),

    id_controlador VARCHAR(30) REFERENCES controladores(id_controlador),

    permiso VARCHAR(50) CHECK (permiso IN ('Lectura', 'Escritura', 'Completo')),

    PRIMARY KEY (id_rol, id_controlador)

```

);

CREATE TABLE usuarios(

id_usuario VARCHAR(30) PRIMARY KEY,

nombre_usuario VARCHAR(255) NOT NULL,

apellido_paterno VARCHAR(255) NOT NULL,

apellido_materno VARCHAR(255) NOT NULL,

contrasena VARCHAR(255) NOT NULL,

correo_usuario VARCHAR(255) UNIQUE,

id_rol VARCHAR(30) REFERENCES roles(id_rol),

fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

foto BYTEA

);

END;

6.10. DML de Inserción de datos a las diferentes tablas de la Base de Datos.

BEGIN;

INSERT INTO public.clasificacion (id_clasificacion, nombre_clasificacion)

VALUES

(1, 'Marcos'),

(2, 'vidrio'),

(3, 'estuche'),

(4, 'limpiador');

COMMIT;

BEGIN;

INSERT INTO public.proveedores (id_proveedor, nombre, contacto, telefono, email, direccion, condiciones_entrega)

VALUES

(1, 'Proveedor Uno', 'Contacto Uno', '123456789', 'proveedor1@email.com', 'Calle Principal 123', 'Entrega en 3 días hábiles'),

(2, 'Proveedor Dos', 'Contacto Dos', '987654321', 'proveedor2@email.com', 'Avenida Secundaria 456', 'Entrega inmediata'),

(3, 'Proveedor Tres', 'Contacto Tres', '456123789', 'proveedor3@email.com', 'Boulevard Central 789', 'Entrega en 7 días hábiles'),

(4, 'Proveedor Cuatro', 'Contacto Cuatro', '321654987', 'proveedor4@email.com', 'Carretera Norte 321', 'Entrega semanal'),

(5, 'Proveedor Cinco', 'Contacto Cinco', '789123456', 'proveedor5@email.com', 'Zona Industrial 654', 'Entrega mensual');

COMMIT

6.11. DML de Selección a las diferentes tablas de la Base de Datos.

```
SELECT * FROM movimientos_inventario;
```

```
SELECT * FROM inventario_local;
```

```
SELECT * FROM proveedores;
```

```
SELECT * FROM roles;
```

```
SELECT * FROM controladores;
```

6.12. DML de Actualización a las diferentes tablas de la Base de Datos.

```
BEGIN;
```

```
UPDATE usuarios
```

```
SET id_rol = 1 -- Administrador
```

```
WHERE id_usuario = 1;
```

```
UPDATE usuarios
```

```
SET id_rol = 2 -- Supervisor
```

```
WHERE id_usuario = 2;
```

```
UPDATE usuarios
```

```
SET id_rol = 3 -- Almacenero
```

```
WHERE id_usuario = 3;
```

```
UPDATE usuarios
```

SET id_rol = 4 -- Vendedor

WHERE id_usuario = 4;

UPDATE usuarios

SET id_rol = 5 -- Reportes

WHERE id_usuario = 5;

COMMIT;

6.13. Selección de la(s) estructuras de datos del sistema.

6.13.1. Descripción de clases del sistema.

Las tablas del sistema:

Tabla 12:Productos

Productos	TIPO	Descripcion
ID_Producto	String	Identificador único para cada producto.
Nombre:	String	Nombre del producto.
Descripcion	String	Descripción detallada del producto.
Cantidad_Total	int	Cantidad total en el inventario.
Precio_Costo	Float	Precio de costo del producto.
ID_Proveedor	int	Relación con el proveedor del producto.
Foto	byte	Fotografía del producto
Peso(g)	float	peso del producto
ancho	float	ancho del producto
alto	float	alto del producto
grosor	float	grosor del producto
id_clasificacion	int	Relación con la clasificacion
id_categorias	int	Relacion con las categorias

Fuente: Elaboración propia 2024

Tabla 13:Clasificacion

Clasificacion	Tipo	Descripcion
ID_Clasificacion	int	Identificador único para cada clasificacion.
Nombre_clasificacion	string	Nombre de la clasificacion especifica

Fuente: Elaboración propia 2024

Tabla 14: Categorías

Categorías	Tipo	Descripcion
ID_Categoria	string	Identificador único para cada categoría.
Nombre_Categoria	string	Nombre de la categoría (por ejm, "Lentes de Sol", "monturas" ...)
Descripcion	String	Descripción de la categoría

Fuente: Elaboración propia 2024

Tabla 15: Producto Categoría

Producto_Categoria	Tipo	Descripcion
ID_Producto	int	Relación con la tabla Productos
ID_Categoria	int	Relación con la tabla Categorías.

Fuente: Elaboración propia 2024

Tabla 16: Locales

Locales	TIPO	Descripcion
ID_Local	String	Identificador único para cada local.
Nombre_Local	String	Nombre del local
Direccion	String	Ubicación física del local.
Telefono	String	Número de contacto del local.
Responsable	String	Número de contacto del local.

Fuente: Elaboración propia 2024

Tabla 17: Inventario local

Inventario_Local	TIPO	Descripcion
ID_Inventario	String	Identificador único para el registro en el inventario.
ID_Producto	int	Relación con el producto.
ID_Local	int	Relación con el local.
Cantidad	int	Cantidad de este producto en el local específico.
Fecha_Ultima_Actualizacion	String	Fecha de la última actualización, para tener un seguimiento del inventario por local

Fuente: Elaboración propia 2024

Tabla 18: Proveedores

Proveedores	TIPO	Descripcion
ID_Proveedor	String	Identificador único del proveedor.
Nombre	String	Nombre de la empresa proveedora.
Contacto	String	Nombre de la persona de contacto en el proveedor.
Telefono	String	Teléfono del proveedor.
Email	String	Correo electrónico del proveedor.
Direccion	String	Dirección física del proveedor.
Condiciones_Entrega	String	Detalles sobre las condiciones de entrega o tiempos de despacho.

Fuente: Elaboración propia 2024

Tabla 19: Movimiento inventario

Movimientos_Inventario	TIPO	Descripcion
ID_Movimiento	String	Identificador único para cada movimiento.
ID_Producto	int	Relación con el producto involucrado
ID_Local	int	Relación con el local donde ocurre el movimiento.
Tipo_Movimiento	String	Tipo de movimiento ("entrada" o "salida").
Cantidad	int	Cantidad involucrada en el movimiento.
Fecha	string	Fecha del movimiento
ID_Usuario	int	Identificador del usuario responsable del movimiento.
Comentarios	String	Observaciones adicionales sobre el movimiento

Fuente: Elaboración propia 2024

Tabla 20: Usuarios

Usuarios	TIPO	Descripcion
ID_Usuario	String	Identificador único para cada usuario.
Nombre_Usuario	String	Nombre de usuario para iniciar sesión.
Contraseña	String	Contraseña encriptada para el usuario. Correo
Correo	String	Correo del usuario
ID_Rol	int	Relación con la tabla Roles para asignar el rol al usuario.
Fecha_Creacion	String	Fecha en que se creó el usuario.
Foto	BYT	Imagen o Foto del usuario

Fuente: Elaboración propia 2024

Tabla 21: Roles

Roles	TIPO	Descripcion
ID_Rol	String	Identificador único para cada rol.
Nombre_Rol	String	rol (por ejemplo, "Administrador", "Empleado", "Supervisor")
Descripcion	String	Descripción del rol.

Fuente: Elaboración propia 2024

Tabla 22: Controladores

Controladores	TIPO	Descripcion
ID_Controlador	String	Identificador único para cada controlador o sección de la aplicación.
Nombre_Controlador	String	Nombre del controlador (por ejemplo, "Gestión de Productos", "Reportes", "Usuarios").
Descripcion	String	Descripción de la funcionalidad del controlador.

Fuente: Elaboración propia 2024

Tabla 23:Permisos Rol

Permisos_Rol	TIPO	Descripcion
ID_Rol	int	Relación con la tabla Roles.
ID_Controlador	int	Relación con la tabla Controladores.
Permiso	string	Valor booleano o de texto (por ejemplo, "Lectura", "Escritura", "Completo") que especifica el nivel de acceso.

Fuente: Elaboración propia 2024

6.13.2. Descripción de método de la clase.

Las estructuras y sus métodos:

Tabla 24:Árbol Rojo negro

Arbol rojo-negro	
Metodos	Descripcion
Insertar(T elemento)	Inserta un nuevo elemento en el árbol, manteniendo la estructura balanceada.
Eliminar(T elemento)	Elimina un elemento específico del árbol y reorganiza los nodos si es necesario para mantener la propiedad de balance del árbol.
Buscar	Busca y devuelve un elemento en el árbol que cumpla con un criterio específico
BusquedaPorAmplitud()	Utiliza una cola para recorrer el árbol nivel por nivel (BFS), devolviendo los elementos en el orden en que se visitan.
BusquedaPorProfundidad()	Utiliza una pila para recorrer el árbol rama por rama (DFS), devolviendo los elementos en el orden en que se visitan
RecorrerEnOrdenAscendente()	Devuelve una lista de los elementos del árbol en orden ascendente.
RecorrerEnOrdenDescendente()	Devuelve una lista de los elementos del árbol en orden descendente
Altura()	Devuelve la altura del árbol B.
rotarIzquierda(nodo)	Realiza una rotación a la izquierda en el árbol para ayudar a mantener el equilibrio tras una inserción o eliminación.
rotarDerecha(nodo)	Realiza una rotación a la derecha en el árbol para ayudar a mantener el equilibrio tras una inserción o eliminación.
cambiarColor(nodo)	Cambia el color del nodo y sus hijos.
mover_rojo_izq(nodo)	Mueve un enlace rojo hacia la izquierda para reestructurar el árbol.
mover_rojo_der(nodo)	Mueve un enlace rojo hacia la derecha para reestructurar el árbol.
balancear(nodo h)	Balancea el árbol después de una operación de inserción o eliminación.

Fuente: Elaboración propia 2024

Tabla 25:Pila

Pila (Stack)	
Metodos	Descripcion
Push(T elemento)	Agrega un elemento a la parte superior de la pila.
Pop()	Elimina y devuelve el elemento en la parte superior de la pila.
Peek()	Devuelve el elemento en la parte superior de la pila sin eliminarlo.
EstaVacía()	Devuelve true si la pila está vacía, y false en caso contrario.
Tamaño()	Devuelve el número de elementos en la pila.
RevertirPila()	Reorganiza los elementos de la pila en orden inverso.
BuscarElemento(T elemento)	Verifica si un elemento específico está en la pila.
Usada para DFS	Integrada con el B-tree para realizar búsquedas en profundidad.

Fuente: Elaboración propia 2024

Tabla 26:Cola

Cola (Queue)	
Metodos	Descripcion
Enqueue(T elemento)	Agrega un elemento al final de la cola
Dequeue()	Elimina y devuelve el elemento al frente de la cola.
Peek()	Devuelve el elemento al frente de la cola sin eliminarlo
EstaVacia()	Devuelve true si la cola está vacía, y false en caso contrario.
Tamaño()	Devuelve el número de elementos en la cola.
FusionarColas(Cola<T> otraCola)	Combina dos colas en una sola manteniendo el orden FIFO.
BuscarElemento(T elemento)	Verifica si un elemento específico está en la cola.
Usada para BFS	Integrada con el B-tree para realizar búsquedas en amplitud.

Fuente: Elaboración propia 2024

Tabla 27:Lista

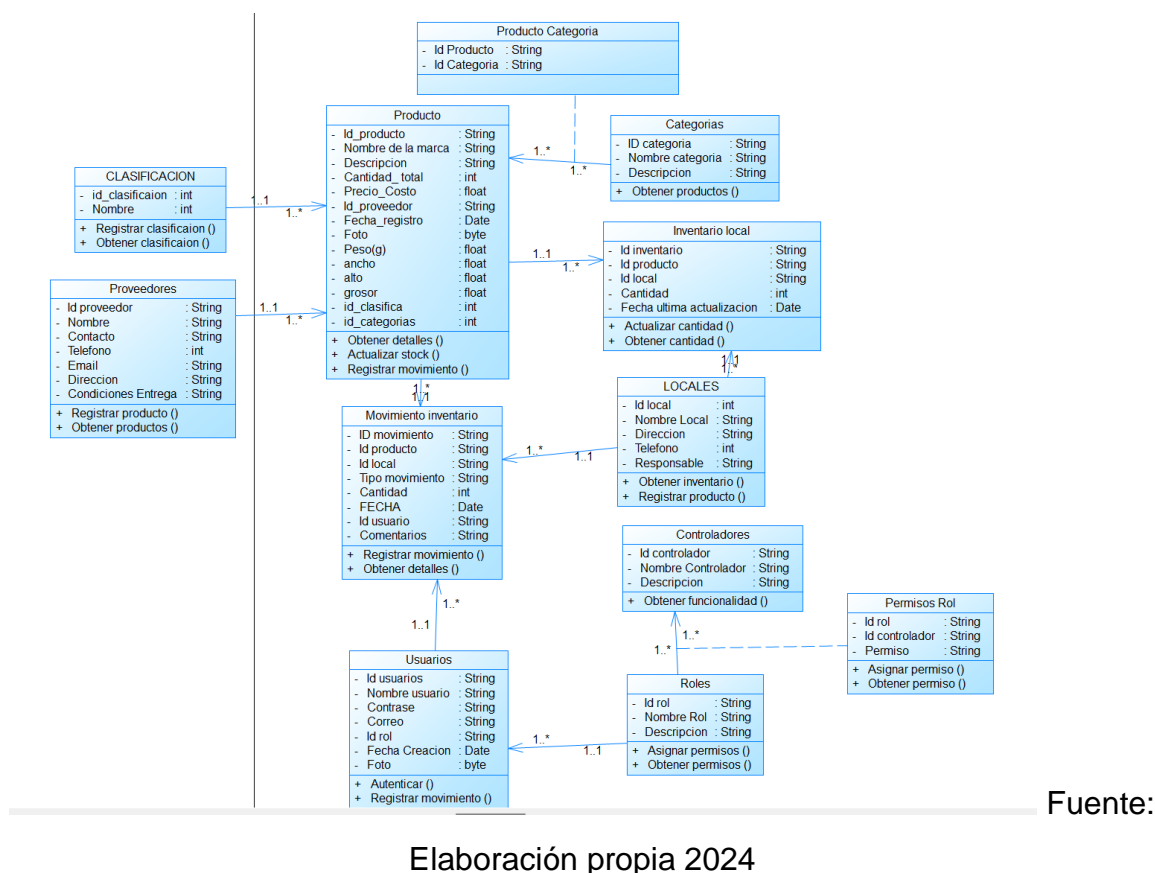
Lista(List)	
Metodos	Descripcion
Agregar(T elemento)	Agrega un elemento al final de la lista.
Eliminar(T elemento)	Elimina un elemento de la lista si se encuentra.
Buscar(Func<T, bool> criterio)	Busca un elemento en la lista basado en un criterio específico y lo devuelve.
Ordenar()	Ordena los elementos de la lista en orden ascendente o descendente.
Contiene(T elemento)	Devuelve true si la lista contiene el elemento, y false en caso contrario.
InsertarEnPosicion(int indice, T elemento)	Inserta un elemento en una posición específica de la lista.
ObtenerEnPosicion(int indice)	Devuelve el elemento en una posición específica de la lista.
RecorrerLista()	Devuelve todos los elementos de la lista en el orden en que están almacenados.
Tamaño()	Devuelve el número de elementos en la lista.
RevertirLista()	Invierte el orden de los elementos en la lista.
EliminarDuplicados()	Elimina elementos duplicados de la lista para mantener solo elementos únicos.

Fuente: Elaboración propia 2024

6.13.3. Relación de clase.

Diagramas de clases:

Ilustración 17:Diagrama de UML



6.14. Implementación.

6.14.1. Interfaces de entrada.

Interfaces de entrada:

Ilustración 18: Interfaz de proveedores

La imagen muestra una interfaz gráfica para la gestión de proveedores, diseñada con un menú lateral de opciones que incluye botones para agregar, editar, eliminar, buscar y exportar datos. En el centro, destaca un formulario para registrar información clave del proveedor, como nombre, contacto, dirección, email y teléfono. Debajo, se visualiza una tabla con los datos registrados. El diseño utiliza un esquema de colores

rojo, verde y amarillo, con un enfoque en la organización y facilidad de uso para optimizar la administración de proveedores.

Fuente: Elaboración propia 2024

Ilustración 19: Interfaz de Productos

La imagen muestra una interfaz para gestionar productos, con opciones para añadir nuevos, buscarlos y una tabla que lista categoría, nombre, y botones para editar o eliminar registros existentes.

Fuente: Elaboración propia 2024

Ilustración 20: Interfaz de entradas

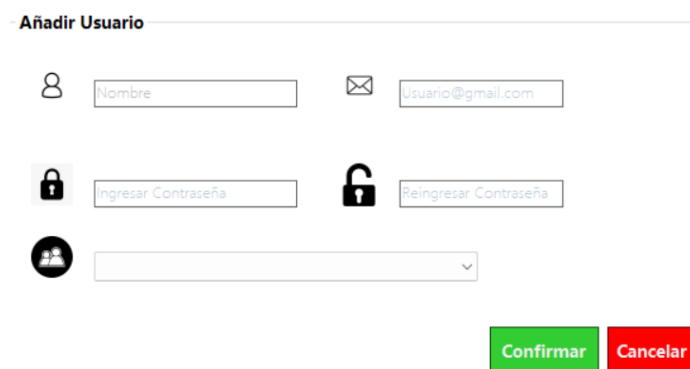
La imagen muestra una interfaz para la gestión de entradas, con un formulario para registrar producto, cantidad, proveedor, fecha y descripción. Incluye botones para agregar, editar, buscar, eliminar entradas y cancelar acciones.



Fuente: Elaboración propia 2024

Ilustración 21: Interfaz para añadir usuarios

La imagen muestra un formulario para añadir usuarios, con campos para ingresar nombre, correo, contraseña y su confirmación. Incluye un menú desplegable para roles y botones de acción: "Confirmar" (verde) y "Cancelar" (rojo).



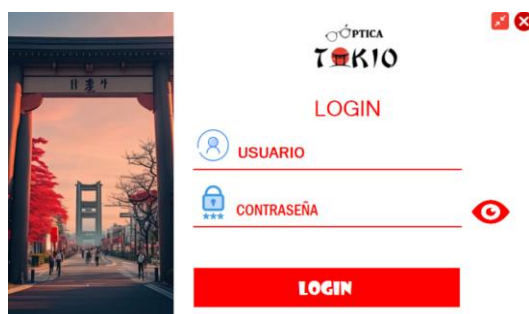
Fuente: Elaboración propia 2024

6.14.2. Interfaces de salida.

Interfaces de salida:

Ilustración 22: Interfaz de login

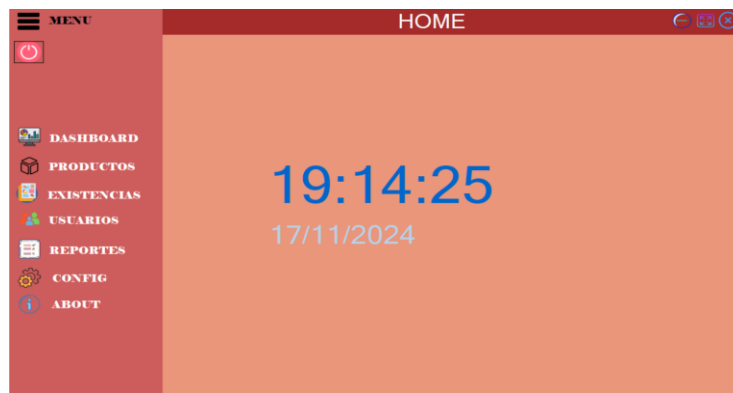
La imagen muestra una pantalla de inicio de sesión de Óptica Tokio, con campos para ingresar usuario y contraseña. Incluye un botón rojo de "Login" y un ícono para visualizar la contraseña ingresada.



Fuente: Elaboración propia 2024

Ilustración 23: Interfaz de Home

La imagen muestra la pantalla principal de inicio Home, con un reloj digital y fecha al centro. En el menú lateral izquierdo se destacan opciones como dashboard, productos, usuarios, reportes y configuración.



Fuente: Elaboración propia 2024

Ilustración 24: Interfaz de Categorías

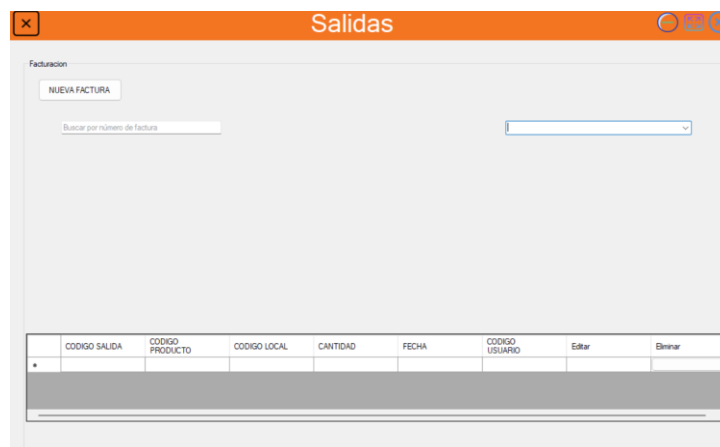
La imagen muestra una interfaz de gestión de categorías, con un botón para agregar nuevas categorías, un buscador y una tabla que lista nombres de categorías con opciones para editar o eliminar registros.



Fuente: Elaboración propia 2024

Ilustración 25: Interfaz de salidas

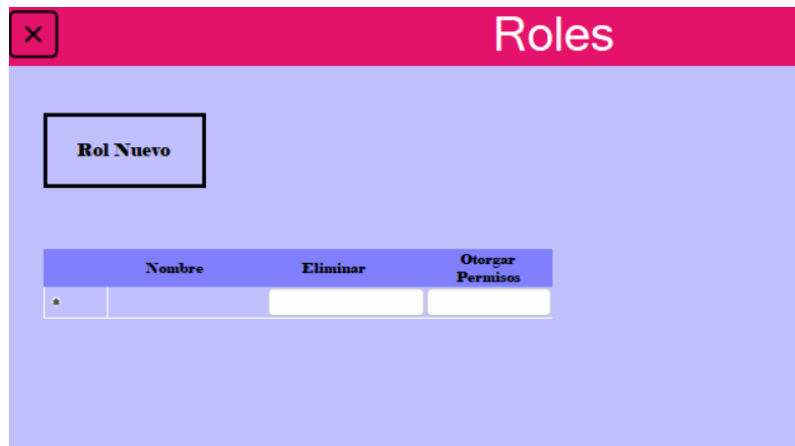
La imagen muestra una interfaz para la gestión de salidas, con opciones para generar una nueva factura y buscar por número. Incluye una tabla para registrar detalles como producto, local, cantidad y fecha.



Fuente: Elaboración propia 2024

Ilustración 26: Interfaz de Roles

La imagen muestra una interfaz para la gestión de roles, con un botón para crear nuevos roles. Incluye una tabla donde se listan roles con opciones para eliminar y otorgar permisos correspondientes.



Fuente: Elaboración propia 2024

Ilustración 27: Interfaz de Usuarios

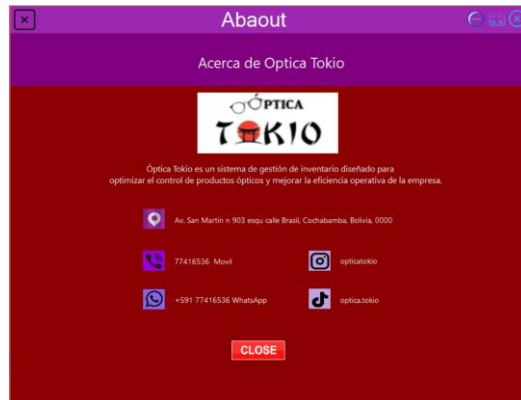
La imagen muestra una interfaz para gestionar usuarios, con opciones para agregar nuevos, buscar por nombre o correo electrónico. Incluye una tabla que lista usuarios con columnas para roles, editar y eliminar registros.



Fuente: Elaboración propia 2024

Ilustración 28: Interfaz de About

La imagen muestra una sección "Acerca de Óptica Tokio", con una breve descripción del sistema, información de contacto, como dirección, teléfono, WhatsApp, e íconos de redes sociales (Instagram y TikTok). Incluye un botón "Close".



Fuente: Elaboración propia 2024

6.14.3. Código del sistema

Ilustración 29: Código del formulario Main 1

La imagen muestra un fragmento de código en C# correspondiente a un formulario de Windows Forms. Incluye referencias, declaraciones de variables, métodos para manejar eventos, y ajustes visuales personalizados.


```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Runtime.InteropServices;
8  using System.Security.Cryptography;
9  using System.Text;
10 using System.Threading;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13 using Optica_Tokio.UI.Controles;
14 using Optica_Tokio.UI.Formularios;
15
16 namespace Optica_Tokio.UI.Formularios
17 {
18     4 referencias
19     public partial class FrmMain_Menu : Form
20     {
21         private Button currentBoton;
22         private Random random;
23         private int tempIndex;
24         private Form activateForm;
25         private int bordersize = 2;
26
27         1 referencia
28         public FrmMain_Menu()
29         {
30             InitializeComponent();
31             random = new Random();
32             this.Padding = new Padding(bordersize);
33             this.BackColor = Color.FromArgb(233,150,122);
34         }
35         [DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
36         8 referencias
37         private extern static void ReleaseCapture();
38         [DllImport("user32.DLL", EntryPoint = "SendMessage")]
39         8 referencias
40         private extern static void SendMessage(System.IntPtr hwnd, int wmsg, int wparam, int lparam);
41
42         1 referencia
43         private void pictureBox1_Click(object sender, EventArgs e)
44         {
45         }
46
47         0 referencias
48         protected override void WndProc(ref Message message)
49         {
50             const int WM_NCCALSIZE = 0x0083;
51             const int WM_NCHITTEST = 0x0084;
52             const int resizeAreaSize = 10;
53
54             const int HTCLIENT = 1;
55             const int HTLEFT = 10;
56             const int HTRIGHT = 11;

```

Fuente: Elaboración propia 2024

Ilustración 30: Código del formulario Main 2

La imagen muestra código en C# implementando la redimensión de ventanas en un formulario de Windows Forms, utilizando el método WndProc. Define bordes interactivos y lógica para detectar áreas clickeables.

```

50  const int HTRIGHT = 11;
51  const int HTTP = 12;
52  const int HTTPLEFT = 13;
53  const int HTTPRIGHT = 14;
54  const int HTBOTTOM = 15;
55  const int HTBOTTOMLEFT = 16;
56  const int HTBOTTOMRIGHT = 17;
57  if (message.Msg == WM_NCHITTEST)
58  {
59      base.WndProc(ref message);
60      if (this.WindowState == FormWindowState.Normal)
61      {
62          if ((int)message.Result == HTCLIENT)
63          {
64              Point screenPoint = new Point(message.LParam.ToInt32());
65              Point clientPoint = this.PointToClient(screenPoint);
66
67              if (clientPoint.Y <= resizeAreaSize)
68              {
69                  if (clientPoint.X <= resizeAreaSize)
70                      message.Result = (IntPtr)HTTOPLEFT;
71                  else if (clientPoint.X < (this.Size.Width - resizeAreaSize))
72                      message.Result = (IntPtr)HTTOP;
73                  else
74                      message.Result = (IntPtr)HTTOPRIGHT;
75              }
76              else if (clientPoint.Y >= (this.Size.Height - resizeAreaSize))
77              {
78                  if (clientPoint.X <= resizeAreaSize)
79                      message.Result = (IntPtr)HTBOTTOMLEFT;
80                  else if (clientPoint.X < (this.Size.Width - resizeAreaSize))
81                      message.Result = (IntPtr)HTBOTTOM;
82                  else
83                      message.Result = (IntPtr)HTBOTTOMRIGHT;
84              }
85              else
86              {
87                  if (clientPoint.X <= resizeAreaSize)
88                      message.Result = (IntPtr)HTLEFT;
89                  else if (clientPoint.X >= (this.Size.Width - resizeAreaSize))
90                      message.Result = (IntPtr)HTRIGHT;
91              }
92          }
93      }
94      return;
95  }
96
97  if (message.Msg == WM_NCCALSIZE && message.WParam.ToInt32() == 1)
98  {
99      return;
100  }
101  base.WndProc(ref message);
102  }
103  10 referencias
104  private void OpenChildFrom(Form ChildFrom, object btnsender)

```

Fuente: Elaboración propia 2024

Ilustración 31: Código del formulario Main 3

El código muestra métodos en C# para gestionar formularios secundarios, seleccionar colores temáticos y activar botones, personalizando estilos, colores y funcionalidades en un entorno de Windows Forms.

```

181         base.WindowTitle = "message/";
182     }
183     10 referencias
184     private void OpenChildFrom(Form ChildFrom, object btnsender)
185     {
186         if (activateForm != null)
187         {
188             activateForm.Close();
189         }
190         ActivateColor(btnsender);
191         activateForm = ChildFrom;
192         ChildFrom.TopLevel = false;
193         ChildFrom.FormBorderStyle = FormBorderStyle.None;
194         ChildFrom.Dock = DockStyle.Fill;
195         this.panelDesktop.Controls.Add(ChildFrom);
196         this.panelDesktop.Tag = ChildFrom;
197         ChildFrom.BringToFront();
198         ChildFrom.Show();
199         lbltitle.Text = ChildFrom.Text;
200     }
201     1 referencia
202     private Color SelectThemeColor()
203     {
204         int index = random.Next(ThemeColor.colores.Count);
205         while (tempIndex == index)
206         {
207             index = random.Next(ThemeColor.colores.Count);
208         }
209         tempIndex = index;
210         string color = ThemeColor.colores[index];
211         return colorTranslator.FromHtml(color);
212     }
213     5 referencias
214     private void ActivateColor(object btnsender)
215     {
216         if (btnsender != null)
217         {
218             if (currentBoton != (Button)btnsender)
219             {
220                 DisableControls();
221                 Color color = SelectThemeColor();
222                 currentBoton = (Button)btnsender;
223                 currentBoton.BackColor = color;
224                 currentBoton.ForeColor = Color.White;
225                 currentBoton.Font = new System.Drawing.Font("Elephant", 15F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
226                 panTop.BackColor = color;
227                 ThemeColor.primaryColor = color;
228                 ThemeColor.secondaryColor = ThemeColor.ChangeColorBrighthness(color, ~0.3);
229                 btnCloseForm.Visible = true;
230                 ApplyColorToAssociatedControls(currentBoton.Parent, color);
231             }
232         }
233     }
234 }

```

Fuente: Elaboración propia 2024

6.15. Aplicar los elementos fundamentales de la oferta y la demanda de bienes y servicios a los hechos y fenómenos económicos vigentes en el país que permita formular sus leyes y comprender sus resultados.

6.15.1. Análisis de mercado

El análisis de las ventas históricas en la Óptica Tokio permitirá identificar los productos con mayor rotación y los patrones de consumo recurrentes. Este proceso incluirá la evaluación de datos pasados para determinar la demanda promedio y los ciclos de consumo, como temporadas altas o promociones específicas. Con esta información, se podrán ajustar las estrategias de inventario para asegurar la disponibilidad de los productos más solicitados, evitando desabastecimientos y mejorando la experiencia del cliente. Además, este enfoque permitirá optimizar la gestión de recursos al alinear las necesidades del mercado con el stock disponible.

La implementación de la metodología ABC será fundamental para clasificar y priorizar los productos en función de su impacto en las ventas y frecuencia de rotación. Los productos "A", que representan una pequeña parte del inventario, pero generan la mayor proporción de ingresos, tendrán prioridad máxima y un mayor nivel de stock. Los productos "B", de importancia moderada, serán gestionados de forma balanceada, mientras que los "C", de menor rotación, tendrán un enfoque de control para minimizar costos. Esta clasificación permitirá una gestión eficiente del inventario, enfocando los recursos en los productos más rentables y asegurando un portafolio diversificado que responda a las necesidades del mercado.

Ecuación 1: Modelo EOQ

$$EOQ = \sqrt{\frac{2DS}{H}}$$

D: Demanda anual del producto.

S: Costo de hacer un pedido.

H: Costo anual de mantener una unidad en inventario.

Un inventario balanceado que responda a la demanda sin generar costos excesivos de almacenamiento o rotura de stock.

6.15.2. Costos

La automatización del sistema permitirá disminuir costos asociados a errores manuales, como datos incorrectos o pedidos duplicados, y minimizar problemas de rotura de stock y excedentes. Esto optimizará la gestión del inventario, reduciendo pérdidas económicas y mejorando la eficiencia operativa.

El sistema automatizado será capaz de programar pedidos basándose en la demanda histórica, ajustando las compras a las necesidades reales. Esto reducirá las

adquisiciones innecesarias y garantizará que los niveles de inventario sean adecuados, mejorando el flujo de trabajo y evitando costos adicionales.

6.15.3. Rotación

Porcentaje de pedidos completados sin rotura de stock.

Ecuación 2: Rotación del inventario

$$Rotación = \frac{\text{Costo de bienes vendidos}}{\text{Promedio de inventario}}$$

Establecer paneles de control para analizar cómo el sistema cumple con la demanda del mercado.

6.16. Determinar el equilibrio microeconómico entre los componentes de ingresos y gastos de una entidad económica.

6.16.1. Cantidad Económica de Pedido

Ecuación 3: Modelo EOQ

$$Q = \frac{FC}{P - VC}$$

FC: Costos fijos (almacenamiento, personal).

P: Precio de venta por unidad.

VC: Costo variable por unidad.

Ajustar los niveles de inventario para minimizar gastos de almacenamiento sin afectar la atención al cliente.

6.16.2. Proyección económica

Comparar los costos operativos antes y después de implementar el sistema, considerando:

- Costos de adquisición y mantenimiento del sistema.
- Ahorros generados por la reducción de errores en inventario.

- **Ecuación 4: Modelo EOQ**

$$ROI = \frac{\text{Beneficio neto obtenido}}{\text{Inversión total}} * 100$$

Esto permitirá evaluar si el sistema es rentable y en cuánto tiempo se recuperará la inversión.

6.16.3. Indicadores económicos

Evaluar la eficiencia del sistema en la reducción de costos fijos y variables.

Medir el impacto directo de la automatización en la reducción de costos.

Analizar cómo estos indicadores evolucionan antes y después de implementar el sistema.

6.17. Establecer la sensibilidad de la permanencia de las cantidades por adquirir o producir ante una elevación o disminución del precio de los bienes o servicios básicos.

6.17.1. Análisis

El análisis de elasticidad permitirá identificar cómo las variaciones en los precios afectan la cantidad demandada de los productos. Esto ayudará a ajustar los niveles de inventario según la sensibilidad del cliente a los cambios de precio, optimizando la disponibilidad de productos para maximizar ingresos y satisfacer la demanda.

Se realizarán simulaciones para evaluar el impacto de las fluctuaciones en los precios de los proveedores sobre la cantidad económica de pedido (EOQ). Estas simulaciones permitirán tomar decisiones de compra más informadas, garantizando un equilibrio entre costos de adquisición, almacenamiento y las necesidades del inventario.

6.17.2. Adaptabilidad

Analizar cómo el sistema responde a fluctuaciones de precios en insumos, ajustando los costos operativos para mantener la rentabilidad.

Evaluar cómo la eficiencia del sistema mitiga el impacto de aumentos en precios de proveedores, protegiendo los márgenes operativos.

6.17.3. Elasticidad

La evaluación de indicadores de elasticidad permitirá analizar cómo los cambios en los precios de los insumos afectan los costos operativos y el manejo del inventario, identificando relaciones directas entre variaciones de precio y ajustes necesarios en la gestión. Asimismo, se medirá el margen de ganancia neto para determinar la capacidad del sistema de proteger la rentabilidad frente a fluctuaciones en los precios del mercado. Este enfoque proporcionará una visión integral sobre la resiliencia operativa y financiera del sistema ante dinámicas del entorno comercial.

7. COCLUSIONES Y RECOMENDACIONES

8. BIBLIOGRAFÍA

Edward, R. (2015). *Strategic Management: A Stakeholder Approach*. Cambridge: Cambridge University Press.

Enfoques y métodos de investigación. (2018). Bogotá: Ediciones de la U.

Morales, O. A. (2019). *FUNDAMENTOS DE LA INVESTIGACIÓN DOCUMENTAL Y LA MONOGRAFÍA*. La Paz: Universidad de los Andes.

Sánchez, C. H. (2018). *Metodología de la investigación*. Lima: San Marcos.

Universidad de la República. (2017). *Etapas de la investigación bibliográfica*. Montevideo: Universidad de la República.

ANEXOS



ANEXOS

Anexo “A”: Repositorio de GitHub donde se esta trabajando

Estructura-de-datos-2 Private

Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

Skaydox Funciona 69c6206 · 10 hours ago 115 Commits

Base de datos	cambiado de grb a form para registrar salidas	2 days ago
Documentacion	documeno	2 days ago
Estructura-de-datos-2	mejora de formularios adaptandolo mas al concepto logico	2 weeks ago
Excel	errores	3 days ago
Optica Tokio	Funciona	10 hours ago
Proyecto_estructura	ho	2 weeks ago
Tareas	tareas	2 weeks ago
Estructura del programa.txt	Modifique estructuras	last month
Estructuras.txt	Modifique estructuras	last month
Formularios basicos.txt	Modifique estructuras	last month
Metodos basicos.txt	Metodos basicos	last month
Pasoso para el git.txt	Concluyendo la fusión pendiente	last month
Servidor.txt	servidor que deben realizar	last month
Tareas a realizar.txt	Tareas Realizadas	last month
Videos clave.txt	Videos clave	3 weeks ago

About

En este repositorio las actualizaciones del Proyecto de estructura de datos dos

Activity 0 stars 1 watching 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 5

Languages

C# 100.0%

Suggested workflows

Based on your tech stack