



**DECANATO DE INGENIERÍA E INFORMÁTICA ESCUELA  
DE INFORMÁTICA**

**PROYECTO DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE:  
INGENIERO(A) DE SOFTWARE**

**Tema:**

**“PROPUESTA DE DESARROLLO E IMPLEMENTACIÓN DE UN  
SISTEMA DE INVENTARIO POR SUSCRIPCIÓN PARA PYMES  
BAJO LA ARQUITECTURA SAAS EN LA NUBE, SANTO  
DOMINGO, 2021”**

**SUSTENTANTE:**

**EMERSON PICHARDO GARCIA**

**20150437**

**ASESOR:**

**Lic., MCE, MGP. Juan Pablo Valdez**

**PROYECTO DE TESIS 2021-3**

**Santo Domingo, D.N.**

**Septiembre 2021**



Tema: “PROPUESTA DE DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO POR SUSCRIPCIÓN PARA PYMES BAJO LA ARQUITECTURA SAAS EN LA NUBE, SANTO DOMINGO, 2021”



## **I. AGRADECIMIENTOS**

## **Agradecimientos**

Agradezco a Dios en primer lugar por darme la oportunidad de llegar hasta este punto de mi vida, y por las consecuentes oportunidades que dentro de su voluntad me abrirá. Me siento soberanamente agradecido del apoyo total e incondicional que siempre recibí de mi familia, tengo mucha suerte de tener a personas tan buenas y tan llenos de amor; sin tomar en cuenta ningún orden en particular, agradezco a mis padres Sonia García y Pablo Pichardo por brindar todo de sí para educarme de la mejor forma posible, a mi amada esposa Dahiry Castillo por acompañarme tan amorosamente en este trayecto y por el resto de mi vida, a mi queridísima hermana Emely Pichardo por siempre brindarme palabras de consejo y amor cuando lo necesité; estas personas generan que de mi corazón nazca el más grande de los agradecimiento.

Agradezco grandemente a mi asesor de tesis Juan Pablo Valdez por su empeño, dedicación y compromiso hacia los estudiantes de UNAPEC, siempre brindando la milla extra para entregarnos información y educación de calidad, un verdadero ejemplo personal a seguir. Durante la creación de esta tesis, Juan Pablo fue un pilar fundamental, de no ser por él, este trabajo no contaría con el nivel de calidad que tiene, y con toda honestidad no sabría cómo lo habría hecho sin él. Finalmente estoy agradecido con mi casa de estudio UNAPEC por su gran aporte a mi desarrollo profesional.

**Emerson Pichardo Garcia**



## **II. DEDICATORIAS**

## **Dedicatorias**

Dedico este trabajo a mis padres Pablo Pichardo y Sonia Garcia, quienes por su esfuerzo y dedicación estoy aquí.

**Emerson Pichardo García**



### **III. RESUMEN EJECUTIVO**

## Resumen Ejecutivo

El desarrollo del proyecto tiene como finalidad presentar una propuesta de desarrollo e implementación de un sistema de inventario por suscripción para PymES de Santo Domingo bajo la arquitectura SaaS en la nube, con el fin de ofrecer una plataforma que mejore el proceso de gestión de inventario y abastecimiento de productos, y proveer la información necesaria para tomar mejores decisiones de negocios.

Esta solución ofrecerá herramientas de registro de transacciones de inventario, dicha información luego se utilizará para automatizar las operaciones más comunes dentro de la gestión de inventario. Debido a la variabilidad de las entradas y salidas de productos específicos, se integrará en esta solución, un sistema de predicción de demanda adaptativa, contribuyendo a los usuarios una línea base para mejorar la toma de decisiones.

En este trabajo se entregará el diseño de alto y bajo nivel de la solución y de las aplicaciones que la integran; se dará los requerimientos funcionales y no funcionales de la solución, junto las herramientas y tecnologías a utilizar para desarrollar e implementar exitosamente esta aplicación.

# ÍNDICE

<b>AGRADECIMIENTOS</b>	<b>2</b>
<b>DEDICATORIAS</b>	<b>4</b>
<b>RESUMEN EJECUTIVO</b>	<b>6</b>
<b>INTRODUCCIÓN</b>	<b>25</b>
<b>Capítulo I: Marco Metodológico</b>	<b>29</b>
Planteamiento del Problema	31
Situación Problemática	31
Problema	31
Objeto	32
Campo	32
Objetivos de Investigación	32
Objetivo General	32
Objetivos Específicos	32
Tipo de Investigación	33
Investigación Explicativa	33
Investigación Descriptiva	33
Investigación Analítica	33
Método de Investigación	34
Método Deductivo	34
Método de Observación	34
Método de Estadística	34
Método de Análisis	34
Fuentes y Técnicas	35
Observación	35
Entrevistas	35
Encuestas	35
Población de la encuesta	35
Hipótesis	37
Hipótesis	37
Definición conceptual y operacionalización de las variables	37
<b>Capítulo II: Marco Referencial</b>	<b>40</b>
Marco Conceptual	42

Inventario	42
Conceptualización	42
Historia	42
PyMES	43
Conceptualización	43
Clasificación	43
Ciclo de desarrollo de software	43
Conceptualización	43
Metodologías de Desarrollo de Software	44
Conceptualización	44
Metodología Tradicional	44
Metodología Ágil	45
Artefacto	45
Conceptualización	45
Nube	46
Conceptualización	46
Infraestructura como servicio (IaaS)	46
Conceptualización	46
Plataforma como servicio (PaaS)	46
Conceptualización	46
Software como servicio (SaaS)	47
Conceptualización	47
Marco Teórico	48
Características del Proceso de Inventario a nivel mundial	48
Buenas Prácticas	49
Nivel de Automatización del Proceso de Gestión de Inventario a Nivel Mundial	51
Hojas de Cálculo	51
Principales Ventajas	51
Principales Desventajas	51
Aplicaciones Especializadas	52
Principales Ventajas	52
Principales Desventajas	52
Códigos de Barra	53
Principales Ventajas	53
Principales Desventajas	53
RFID	54

Principales Ventajas	54
Principales Desventajas	54
Tipos de Inventario	55
Regulación en la Gestión de Inventario a Nivel Mundial	56
Métodos de Costeo de Inventario	56
Método del Costo Promedio o Ponderado	56
Ventajas	57
Desventajas	57
Primeras en Entrar Primeras en Salir	57
Ventajas	57
Desventajas	58
Últimas en Entrar Primeras en Salir	58
Ventajas	58
Desventajas	58
Softwares de Gestión de Inventario para PyMES en República Dominicana	59
Holded	59
FACTUSOL 360	59
Cashflow	59
<b>Capítulo III: Estudio de Factibilidad de la Aplicación</b>	<b>61</b>
Situación Actual	63
Estudio de Factibilidad	64
Innovación del Producto Propuesto	64
Reducción de Costos	65
Análisis de Factibilidad Técnica	65
Análisis de Factibilidad Económica	67
Análisis costos-beneficios	68
Modelo del Negocio	70
Análisis de Factibilidad Operativa	71
Estudio de Mercado	71
Encuesta	73
Presentación y Análisis de los Resultados	73
Resultados Encuesta Realizada a Propietarios de PyMES	73
Resultados de Encuesta Realizada a Empleados Operativos de las PyMES	81
Beneficios	86

<b>Capítulo IV: Análisis y Diseño del un Sistema</b>	<b>88</b>
Acta de constitución del Proyecto	90
Nombre del Proyecto / Información del Proyecto	90
Propósito y Justificación de proyecto	90
Alcance	90
Descripción del Proyecto y Entregables	91
Premisas y Restricción	91
Riesgos Iniciales de Alto Nivel	92
Cronograma de trabajo	92
Sesión de Requerimientos	95
Requisitos Funcionales	95
Requisitos No Funcionales	97
Documento Visión y Alcance	98
Introducción	98
Propósito	98
Alcance	98
Definiciones, Acrónimos y Abreviaturas	99
Referencias	99
Posicionamiento	100
Oportunidad de Negocio	100
Declaración de Problema	101
Declaración de Posición de Producto	101
Descripción de los stakeholders	102
Resumen de los Stakeholders	102
Resumen de los Usuarios	102
Alternativas y Competencia	103
Estándares aplicables	103
Rangos de Calidad	103
Requisitos de Documentación	104
Conclusión del Alcance	104
Casos de Uso	105
Listado de los Casos de Uso	105
Diagrama General Casos de Uso	106
Descripción de Actores	107
Usuario Administrador	107
Usuario Transaccional	107

Especificación de Casos de Uso	108
CUS-1 Registro de Usuario Administrador	108
CUS-2 Gestión de Sucursales	110
CUS-3 Gestión de Proveedores	112
CUS-4 Gestión de catálogo de productos	114
CUS-5 Gestión de Usuarios Transaccionales	116
CUS-6 Movimiento de Productos	118
CUS-7 Entrada de Productos	119
CUS-8 Salida de Productos	121
CUS-9 Visualización de Inventario	122
CUS-10 Visualización de Transacciones de Productos	123
Diseño de Alto Nivel	126
Diagrama de Arquitectura	126
Diagrama General de Clases	127
Diagrama de la Base de Datos	128
Diseño de Bajo Nivel	129
Diagramas de secuencia	129
Registro de Usuario Administrador	129
Gestión de Sucursales	129
Gestión de Catálogo de Productos	131
Gestión de Catálogo de Productos	132
Gestión de usuarios transaccionales	133
Movimiento de Productos	133
Entrada de Productos	134
Salida de Productos	135
Visualización de Transacciones de Productos	137
Diseño de Vistas	137
Inicio de Sesión	138
Registro de Usuario Administrador	139
Gestión de Sucursales	139
Lista	139
Formulario	140
Gestión de Proveedores	140
Lista	140
Formulario	141
Gestión de Catálogo de Productos	141

Lista	142
Formulario	142
Gestión de Usuarios Transaccionales	143
Lista	143
Formulario	143
Movimiento de Productos	143
Lista	143
Formulario	144
Entradas de productos	144
Lista	144
Formulario	145
Salidas de productos	145
Lista	145
Formulario	146
Visualización de Inventario	146
Visualización de Transacción de Productos	148
<b>Capítulo V: Desarrollo de la Aplicación</b>	<b>150</b>
Selección de los Macro-patrones de Arquitectura	152
Single Page Application (SPA)	152
Características Principales	152
Selección de los Patrones de Diseño	153
Inyección de Dependencias	153
Buenas Prácticas de Codificación a Implementar	154
Principios Solid Gold	154
Código Limpio (Clean Code)	154
Selección de las Plataformas de Desarrollo y DevOps	155
Selección de la Plataforma de Desarrollo	155
Base de Datos	155
Web API	155
Aplicación SPA	155
Selección de la Plataforma de DevOps	156
Selección del Lenguaje de Programación	158
Selección del Lenguaje de Programación Frontend	158
Selección del Lenguaje de Programación Backend	158
Desarrollo de la Arquitectura Candidata	159
Ejemplo del CRUD de Gestión de Sucursales	159

Modelo	159
Vistas	160
Listado	160
Formulario	160
Controlador	161
Desarrollo del Menú	162
Usuario Administrador	163
Usuario Transaccional	163
<b>Capítulo VI: Implementación de la Aplicación</b>	<b>166</b>
Selección del Proveedor de Servicios de Nube	168
Selección del Dimensionamiento para el Servidor Web	171
Selección del Dimensionamiento para la Base de Datos	172
Selección y Configuración del WAF Como Servicio	173
Despliegue de la Aplicación en la Nube	175
Ventajas de Docker	175
Ventajas de la Integración de Azure	176
Plan de Implementación y de Reverso	177
Prueba de Estrés de la Aplicación Desplegada	178
Revisión Post-Implementación	179
<b>CONCLUSIONES</b>	<b>181</b>
<b>RECOMENDACIONES</b>	<b>183</b>
<b>BIBLIOGRAFÍAS</b>	<b>186</b>
<b>WEBGRAFÍAS</b>	<b>190</b>
<b>ANEXOS</b>	<b>192</b>

## ÍNDICE DE FIGURAS

- Figura 1. Diagrama general de casos de uso (Elaboración propia)
- Figura 2. Diagrama de arquitectura (Elaboración propia)
- Figura 3. Diagrama general de clases (Elaboración propia)
- Figura 4. Diagrama de la base de datos (Elaboración propia)
- Figura 5. Diagrama de secuencia 01 (Elaboración propia)
- Figura 6. Diagrama de secuencia 02 (Elaboración propia)
- Figura 7. Diagrama de secuencia 03 (Elaboración propia)
- Figura 8. Diagrama de secuencia 04 (Elaboración propia)
- Figura 9. Diagrama de secuencia 05 (Elaboración propia)
- Figura 10. Diagrama de secuencia 06 (Elaboración propia)
- Figura 11. Diagrama de secuencia 07 (Elaboración propia)
- Figura 12. Diagrama de secuencia 08 (Elaboración propia)
- Figura 13. Diagrama de secuencia 09 (Elaboración propia)
- Figura 14. Diagrama de secuencia 07 (Elaboración propia)
- Figura 15. Vista de registro de usuario administrador (Elaboración propia)
- Figura 16. Vista del listado de sucursales (Elaboración propia)
- Figura 17. Vista del formulario para la gestión de sucursales (Elaboración propia)
- Figura 18. Vista del listado de proveedores (Elaboración propia)
- Figura 19. Vista del formulario de gestión de proveedores (Elaboración propia)
- Figura 20. Vista del listado de productos (Elaboración propia)

Figura 21. Vista del formulario de gestión de catálogos de productos (Elaboración propia)

Figura 22. Vista del listado de usuarios transaccionales (Elaboración propia)

Figura 23. Vista del formulario de gestión de usuarios transaccionales (Elaboración propia)

Figura 24. Vista del listado de movimientos (Elaboración propia)

Figura 25. Vista del formulario de movimiento de productos (Elaboración propia)

Figura 26. Vista del listado de entradas (Elaboración propia)

Figura 27. Vista del formulario de entradas de producto (Elaboración propia)

Figura 28. Vista del listado de salidas (Elaboración propia)

Figura 29. Vista del formulario de salidas de productos (Elaboración propia)

Figura 30. Vista de visualización de inventario (Elaboración propia)

Figura 31. Vista de visualización de transacciones (Elaboración propia)

Figura 32. Modelo de la entidad producto (Elaboración propia)

Figura 33. Vista programada del listado de productos (Elaboración propia)

Figura 34. Vista programada del formulario de registro de productos (Elaboración propia)

Figura 35. Controlador de la entidad productos (Elaboración propia)

Figura 36. Menú del usuario administrador (Elaboración propia)

Figura 37. Menú del usuario transaccional (Elaboración propia)

Figura 38. Principales proveedores de servicios de nube, de gartner.com, 2021

Figura 39. Arquitectura WAF de Azure, recuperado de azure.microsoft.com, 2021

Figura 40. Docker y Kubernetes, recuperada de <https://www.xataka.com/>, 2021

Figura 41. Pruebas de estrés, recuperado de <https://testeandosoftware.com/>, 2021



## ÍNDICE DE TABLAS

Tabla 1. Tabla de nivel de confianza (Elaboración propia)

Tabla 2. Tabla de operacionalización de variables de la primera hipótesis (Elaboración propia)

Tabla 3. Tabla de operacionalización de variables de la segunda hipótesis (Elaboración propia)

Tabla 4. Tabla de tipos de inventario tomada de <https://www.redalyc.org/pdf/4655/465545892008.pdf> (Durán, 2012, 63)

Tabla 5. Tabla de características mínimas de computadores (Elaboración propia)

Tabla 6. Tabla de costo de desarrollo presupuestado para la aplicación (Elaboración propia)

Tabla 7. Tabla de costo de nómina (Elaboración propia)

Tabla 8. Tabla de costo total del proyecto (Elaboración propia)

Tabla 9. Tabla inicial del proyecto (Elaboración propia)

Tabla 10. Tabla de premisas y restricciones (Elaboración propia)

Tabla 11. Tabla de cronograma de trabajo (Elaboración propia)

Tabla 12. Tabla de requisitos funcionales (Elaboración propia)

Tabla 13. Tabla de requisitos no funcionales (Elaboración propia)

Tabla 14. Tabla historial de revisiones (Elaboración propia)

Tabla 15. Tabla declaración de problema (Elaboración propia)

Tabla 16. Tabla declaración de posición de producto (Elaboración propia)

Tabla 17. Tabla resumen de los stakeholders (Elaboración propia)

Tabla 18. Tabla resumen de los usuarios (Elaboración propia)

- Tabla 19. Tabla de listado de casos de uso (Elaboración propia)
- Tabla 20. Descripción actor usuario administrador (Elaboración propia)
- Tabla 21. Descripción actor usuario transaccional (Elaboración propia)
- Tabla 22. Tabla descripción y flujo de caso de uso 01 (Elaboración propia)
- Tabla 23. Tabla descripción y flujo de caso de uso 02 (Elaboración propia)
- Tabla 24. Tabla descripción y flujo de caso de uso 03 (Elaboración propia)
- Tabla 25. Tabla descripción y flujo de caso de uso 04 (Elaboración propia)
- Tabla 26. Tabla descripción y flujo de caso de uso 05 (Elaboración propia)
- Tabla 27. Tabla descripción y flujo de caso de uso 06 (Elaboración propia)
- Tabla 28. Tabla descripción y flujo de caso de uso 07 (Elaboración propia)
- Tabla 29. Tabla descripción y flujo de caso de uso 08 (Elaboración propia)
- Tabla 30. Tabla descripción y flujo de caso de uso 09 (Elaboración propia)
- Tabla 31. Tabla descripción y flujo de caso de uso 10 (Elaboración propia)

## ÍNDICE DE GRÁFICOS

Gráfica 1. Representación gráfica de los resultados de la pregunta “Tipo de productos / servicios suplidos por su empresa” (Construcción propia)

Gráfica 2. Representación gráfica de los resultados de la pregunta “Indicar los días picos (los días de mayor concurrencia) dentro de su proceso de inventario” (Construcción propia)

Gráfica 3. Gráfica representativa de los resultados de la pregunta “¿Con qué frecuencia realizan pedidos para reabastecerse? (Marque del 1 al 5, donde 1 = Diario, 2= 2 veces x semana, 3= Semanal, 4 = quincenal y 5 = mensual)” (Construcción propia)

Gráfica 4. Representación gráfica de los resultados de la pregunta “¿Dentro de cuál rango de edad se encuentra usted?” (Construcción propia)

Gráfica 5. Representación gráfica de los resultados de la pregunta “¿Son sus productos perecederos?” (Construcción propia)

Gráfica 6. Representación gráfica de los resultados de la pregunta “¿Qué tan frecuente es que una persona se vaya del establecimiento por no encontrar un producto deseado? (Marque del 1 al 5, donde 1 = Muy Inusual y 5 = Muy Frecuente)” (Construcción propia)

Gráfica 7. Representación gráfica de los resultados de la pregunta “¿En qué tanda suelen realizar con más frecuencia el conteo físico de productos (toma física)?”. (Construcción propia)

Gráfica 8. Representación gráfica de los resultados de la pregunta “¿Cuál es el tiempo promedio que tarda una persona desde que inicia la búsqueda de un producto hasta que lo localiza en el establecimiento?”. (Construcción propia)

Gráfica 9. Representación gráfica de los resultados de la pregunta ¿Cuál es el % de clientes que entiendes se retiran disgustados del establecimiento (por tardanza, por expectativas distintas en cuanto al embalaje, porque no encontró su producto, porque el producto que había no era exactamente el que buscaba, etc.)? (Construcción propia)

Gráfica 10. Representación gráfica de los resultados de la pregunta “¿Ha sido objeto de robo tu empresa por empleados que sustraen artículos?” (Construcción propia)

Gráfica 11. Representación gráfica de los resultados de la pregunta “¿Conoce del abastecimiento “Justo a tiempo?”” (Construcción propia)

Gráfica 12. Representación gráfica de los resultados de la pregunta “¿Sabe de cual proveedor es un artículo que haya sido devuelto por estar defectuoso o por que se haya deteriorado antes de tiempo?” (Construcción propia)

Gráfica 13. Representación gráfica de los resultados de la pregunta “¿Sabes en que % han aumentado o disminuido el costo de tus artículos comprados?” (Construcción propia)

Gráfica 14. Representación gráfica de los resultados de la pregunta “¿Usas algún método para costear tu inventario (UEPS, PEPS, promedio)?” (Construcción propia)

Gráfica 15. Representación gráfica de los resultados de la pregunta ¿Cuál es su género?. (Construcción propia)

Gráfica 17. Representación gráfica de los resultados de la pregunta “¿En qué municipio te ubicas? . (Construcción propia)

Gráfica 18. Representación gráfica de la pregunta “¿Con qué frecuencia asistes a la empresa a la que labora en este periodo de pandemia?”. (Construcción propia)

Gráfica 19. Representación gráfica de los resultados de la pregunta “¿Dónde está ubicada la empresa para la cual laboras?”. (Construcción propia)

Gráfica 20. Representación gráfica de los resultados de la pregunta “¿Cuánto dinero consume en transporte, llamadas al mantener contacto con otros empleados u empresas mientras gestiona el inventario? (Los montos están representados en pesos dominicanos)”. (Construcción propia)

Gráfica 21. Representación gráfica de los resultados de la pregunta “¿Cuál de las siguientes opciones es la principal razón por la cual usted entiende prudente que la empresa adquiera una aplicación para gestionar el inventario?”. (Construcción propia)



## **IV. INTRODUCCIÓN**

## Introducción

Para el primer semestre del 2020 en la República Dominicana se encontraban registradas 1.5 millones de PyMES de las cuales el 46.7% pertenecen al sector comercio y el 14.9% al sector industrias (Ministerio de Industria, Comercio y Mipymes, 2020) por lo que 924,000 empresas, equivalentes al 61.6% de las PyMES, se encuentran involucradas en actividades comerciales que requieren de la administración de inventario de productos, y una gran porción de estas empresas no disponen de un sistema de software apropiado para manejar adecuadamente este proceso o simplemente no cuentan con la infraestructura necesaria; por lo que administran su inventario de forma manual.

Los principales inconvenientes de manejar de forma manual el proceso de inventario son la extensión del tiempo necesario para su administración, la imprecisión de los datos obtenidos y registrados, pérdida potencial de la información de inventario, complejidad en la cadena de suministro y pérdida potencial de productos y dinero.

El presente proyecto brinda el análisis, diseño, desarrollo e implementación para una solución de software que brinde a las PyMES una aplicación en la nube para la administración de su inventario a través de una suscripción adaptada a sus necesidades y a sus posibilidades económicas.

Para dar una solución adecuada y de éxito, se desarrollará una estructura que servirá de guía para detallar los resultados de una serie de estudios que determine la eficacia y efectividad de la solución.

En el capítulo uno, se presentarán los métodos de investigación, la estructura metodológica, las técnicas y fuentes de recolección de datos, los objetivos de la investigación y las hipótesis.

En el capítulo dos, se definen los conceptos necesarios para desarrollar el contenido adecuadamente junto a su marco referencial. Se muestran y describen el proceso de administración de inventario que utilizan las PyMES de Santo Domingo.

En el capítulo tres, se expondrá el estudio de viabilidad de la propuesta de software, todos los aspectos relacionados al mismo como los técnicos y sociales, de igual forma se mostraran los resultados obtenidos en las encuestas realizadas a los potenciales usuarios que muestran relación con la implementación de la propuesta.

En el capítulo cuatro, se indican de manera minuciosa los pormenores del análisis y diseño de la propuesta de software planteada en el acta de proyecto, los requisitos funcionales y no funcionales, así como también se mostrará el documento de visión y alcance, las distintas formas de implementación detallados con sus diagramas y

diseños de alto y bajo nivel, por último se describirá el diseño de las diferentes vistas del software.

En el capítulo cinco, se detalla la metodología de desarrollo a utilizar; los patrones, tecnologías, y marcos de trabajos necesarios para la adecuada construcción y mantenimiento de la solución.

En el capítulo seis, se da a conocer el plan de puesta a producción de la solución propuesta, la arquitectura en la nube a utilizar y las mejores prácticas de implementación para asegurar la calidad de la puesta en línea.

# Capítulo I: Marco Metodológico

## Introducción

Este capítulo presenta tanto los tipos de investigación como la metodología usada en el presente documento. Los objetivos específicos y general, que serán alcanzados en este trabajo, también se establecerán en este primer capítulo. Se determinarán las fuentes de información de datos para los siguientes capítulos, y las técnicas de recopilación de los mismos. Finalmente se desarrollará la hipótesis fundamental.

A continuación, se examina la situación problemática en la que se basa este trabajo, partiendo de la situación general frente a la cual se va a evaluar el proceso específico.

## **1.1. Planteamiento del Problema**

### **1.1.1. Situación Problemática**

Los inventarios son una parte fundamental en el proceso de comercialización de productos, por lo que su buena administración es crucial para el desarrollo exitoso de cualquier empresa; no obstante, no todas las PyMES tienen acceso a una gestión adecuada de su inventario.

La falta de opciones de software para administración de inventario en el mercado actual de República Dominicana ha llevado a muchas PyMES a: gestionar su inventario con aplicaciones que no se encuentran diseñadas explícitamente para esta tarea, utilizar métodos manuales, o directamente manejar el proceso de forma empírica. La utilización de estos métodos no convencionales generan un descontrol en los registros de inventario, que conlleva a pérdidas como: tiempo, productos, dinero, clientes potenciales, y finalmente la falta de interés en organizar este proceso.

### **1.1.2. Problema**

Ineficiencia del proceso de gestión inventario de las PyMES en Santo Domingo, incluyendo el reabastecimiento del mismo.

### **1.1.3. Objeto**

El proceso de inventario en las PyMES de Santo Domingo.

### **1.1.4. Campo**

La implementación de una aplicación web para el control de inventario y de la cadena de suministro de las PyMES de Santo Domingo.

## **1.2. Objetivos de Investigación**

### **1.2.1. Objetivo General**

Desarrollar e implementar un sistema de inventario por suscripción para PyMES de Santo Domingo bajo la arquitectura SaaS en la nube.

### **1.2.2. Objetivos Específicos**

Los objetivos específicos para alcanzar el objetivo general se detallarán a continuación:

- Evaluar la factibilidad de la implementación de la propuesta
- Modelar un sistema de inventario acorde a las necesidades de las PyMES
- Desarrollar la propuesta de software y elaborar una guía de implementación de la misma
- Identificar las mejores prácticas de implementación de software en la nube

## **1.3. Tipo de Investigación**

### **1.3.1. Investigación Explicativa**

Estudiaremos a fondo las problemáticas ya expuestas, con la finalidad de obtener un mejor entendimiento de estos y así poder tener resultados concluyentes a la hora de diseñar una solución.

### **1.3.2. Investigación Descriptiva**

Durante el desarrollo de este proyecto se utilizará la metodología de investigación descriptiva, mediante la cual definiremos, analizaremos y planteamos las implicaciones más relevantes sobre las problemáticas que enfrentan las PyMES al intentar administrar sus inventarios con métodos poco convencionales.

### **1.3.3. Investigación Analítica**

Se agotará un exhaustivo proceso de análisis para determinar, no solo la raíz de los problemas que enfrentan las PyMES sino también para encontrar la forma más simple y eficaz de resolverlos.

## **1.4. Método de Investigación**

### **1.4.1. Método Deductivo**

Nos permitirá inferir sobre datos generales, que sean validados con anticipación, con el propósito de obtener conclusiones válidas.

### **1.4.2. Método de Observación**

A través de este método se obtendrá información de la cantidad de productos que se mueven por inventario según transcurre el tiempo, así como los distintos métodos y dificultades que las PyMES enfrentan.

### **1.4.3. Método de Estadística**

Los resultados de medición y comparación de los procesos de eficacia que se usarán para la implementación de la solución, serán obtenidos a través de la estadística.

### **1.4.4. Método de Análisis**

Una vez obtenidos los resultados se procederá a un minucioso análisis que permitirá determinar el origen de los problemas a los que se exponen las PyMES, para ofrecer una solución eficiente, efectiva y vanguardista.

## **1.5. Fuentes y Técnicas**

### **1.5.1. Observación**

Con la técnica de observación captamos el flujo de personas que interactúan en los procesos de inventariado, así como los diversos métodos que las PyMES tienen para gestionarlo.

### **1.5.2. Entrevistas**

Realizaremos entrevistas a los dueños y empleados de diversas PyMES que estén involucrados en el proceso de inventariado para conocer este proceso con exactitud.

### **1.5.3. Encuestas**

Realizaremos encuestas a los dueños y empleados de diversas PyMES que sean parte del proceso de inventario para entender a fondo las dificultades que se presentan en el mismo.

### **1.5.4. Población de la encuesta**

La muestra significativa se tomará de los empleados y/o dueños de PyMES de Santo Domingo, República Dominicana, que deban interactuar directamente o indirectamente con la administración de inventario de su empresa. La encuesta se distribuirá de forma digital y física a persona de entre 18 a 70 años de edad, sin

distinción de sexo o genero.

Con el fin de determinar el tamaño de la muestra, se utilizó la siguiente fórmula:

n = Tamaño de la muestra.

z = Nivel de confianza.

p = probabilidad de éxito.

q = probabilidad de fracaso.

d = error máximo admitido en términos de proporción.

N = Tamaño de la población.

$$n = \frac{z^2(p \cdot q)}{e^2 + \frac{z^2(p \cdot q)}{N}}$$

Para obtener el nivel de confianza deseado dentro de la puntuación z, se utiliza la siguiente tabla:

<b>Nivel de confianza deseado</b>	<b>Cantidad de encuestas</b>
90%	68
95%	96
99%	165

Tabla 1. Tabla de nivel de confianza. (Elaboración propia)

## 1.6. Hipótesis

### 1.6.1. Hipótesis

1. Si las PyMES tuvieran acceso a una aplicación diseñada para la gestión de inventario que además sea asequible, entonces se mejoraría la comercialización de productos en Santo Domingo y la eficiencia operativa de las mismas
2. La utilización de métodos no convencionales para la administración de inventario generan un descontrol en este proceso, que conlleva a pérdidas como: tiempo, productos, dinero, clientes potenciales, y finalmente la falta de interés en organizar este proceso.

### 1.6.2. Definición conceptual y operacionalización de las variables

Hipótesis	Variables	Definición conceptual	Indicadores	Instrumentos de medición
<i>“Si las PyMES tuvieran acceso a una aplicación diseñada para la gestión de inventario que además sea asequible, entonces se mejoraría la comercialización de productos en Santo Domingo y la eficiencia operativa de las mismas”</i>	Digitalización del proceso de gestión de inventario de las PyMES	Es la implementación del conjunto de pasos en el proceso de inventario a través de la tecnología.	Eficacia en el tiempo de los procesos. Estandarización de los procesos. Reducción de los costos.	Pruebas de aseguramiento de la calidad
	Eficiencia operativa de las PyMES	Es la forma en la que la institución maneja la disponibilidad, abastecimiento y distribución de productos para lograr una alta competitividad.	Mejora en la productividad de la institución. Calidad en los procesos relacionados Mejor predicción para el abastecimiento de los productos	Encuestas

Tabla 2. Tabla de operacionalización de variables de la primera hipótesis. (Elaboración propia)

<b>Hipótesis</b>	<b>VARIABLES</b>	<b>Definición conceptual</b>	<b>Indicadores</b>	<b>Instrumentos de medición</b>
<b>“La utilización de métodos no convencionales para la administración de inventario generan un descontrol en este proceso, que conlleva a pérdidas como: tiempo, productos, dinero, clientes potenciales, y finalmente la falta de interés en organizar este proceso.”</b>	Digitalización del proceso de gestión de inventario de las PyMES	Es la implementación del conjunto de pasos en el proceso de inventario a través de la tecnología.	Eficacia en el tiempo de los procesos. Estandarización de los procesos. Reducción de los costos.	Pruebas de aseguramiento de la calidad
	Mayor eficiencia en la administración e inventario de las PyMES	Es la cantidad de recursos por productos necesitados para gestionar correctamente el inventario	Tiempo necesario en conocer el estado actual del inventario Margen de error promedio en el conteo de productos Costo mensual de mantenimiento del sistema de inventario	Observación Encuestas

*Tabla 3. Tabla de operacionalización de variables de la segunda hipótesis. (Elaboración propia)*

## **Conclusión**

En este capítulo, se establecieron las bases que originan esta investigación y que se detallará agotando el proceso investigativo que se llevará a cabo. Todo lo mostrado, descrito y desarrollado en este capítulo serán los fundamentos de las etapas subsecuentes debido a la importancia de las información presentadas.

Se continuará presentando los objetivos específicos y generales de la investigación, incluyendo la propuesta de solución al problema planteado. Los objetivos alcanzados, los métodos de investigación, el tipo de investigación utilizada para lograr los objetivos y, finalmente, las técnicas fuentes y de recogida de la información serán verificadas como resultado del desarrollo de este capítulo.

## **Capítulo II: Marco Referencial**

## Introducción

En el presente capítulo, se muestran las dos secciones en la se divide el marco referencial: el marco conceptual y el marco teórico. El marco conceptual define los conceptos principales para poder comprender fundamentalmente los temas a abarcar a lo largo de este documento. El marco teórico no dará un visión general del proceso de inventario al rededor de todo el mundo, para aprender y aplicar las mejores prácticas obtenidas; debido a la naturaleza cambiante de este proceso, analizaremos como algunas de las empresas más importante de todo el mundo administran este proceso, y así tener un referente sólido al momento de implementar nuestra propuesta.

## **2.1. Marco Conceptual**

### **2.1.1. Inventario**

#### **2.1.1.1. Conceptualización**

Es el registro documental de los bienes tangibles que se tienen para la venta en el curso ordinario del negocio o para ser consumidos en la producción de bienes o servicios para su posterior comercialización. (Laveriano, 2010, II-1)

#### **2.1.1.2. Historia**

Los inventarios tienen su origen en los egipcios y demás pueblos de la antigüedad, donde acostumbraban a almacenar grandes cantidades de alimentos para ser utilizados en los tiempos de sequía o de calamidades. Es así como surge el problema de los inventarios, como una forma de hacer frente a los periodos de escasez. Estos permiten asegurar la subsistencia del negocio y el desarrollo de sus actividades operativas. Esta forma de almacenamiento de todos los bienes y alimentos necesarios para sobrevivir fue lo que motivó la existencia de los inventarios. (Durán, 2012, 56)

## 2.1.2. PyMES

### 2.1.2.1. Conceptualización

Son las siglas de pequeñas y medianas empresas.

### 2.1.2.2. Clasificación

- **Microempresa:** hasta 10 trabajadores y ventas brutas anuales de hasta ocho millones seiscientos treinta y nueve mil doscientos pesos dominicanos (RD\$8,639,200.00).
- **Pequeña empresa:** de 11 a 50 trabajadores y ventas brutas anuales de hasta cincuenta y ocho millones trescientos catorce mil seiscientos pesos dominicanos (RD\$58,314,600.00).
- **Mediana empresa:** de 51 a 150 trabajadores y ventas brutas anuales de hasta doscientos dieciocho millones ciento treinta y nueve mil ochocientos pesos dominicanos (RD\$218,139,800.00). (Impuestos Internos, 2020, 7)

## 2.1.3. Ciclo de desarrollo de software

### 2.1.3.1. Conceptualización

Son un conjunto de etapas que se agotan para desarrollar adecuadamente un software, las cuales son: etapa conceptual, análisis y diseño inicial, plan de iteraciones, diseño computacional, desarrollo y fase de despliegue. (Figueroa, 2009, 3-5)

## **2.1.4. Metodologías de Desarrollo de Software**

### **2.1.4.1. Conceptualización**

Una metodología es un conjunto de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo.

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado. (Maida & Pacienza, 2015, 12)

### **2.1.4.2. Metodología Tradicional**

Las metodologías tradicionales son denominadas, como metodologías pesadas, y centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. (Maida & Pacienza, 2015, 17)

### **2.1.4.3. Metodología Ágil**

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Un modelo de desarrollo ágil, generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento). (Maida & Pacienza, 2015, 18)

### **2.1.5. Artefacto**

#### **2.1.5.1. Conceptualización**

Parte diferenciada de un producto software que es creada o modificada por las actividades. Puede ser un documento (textual o gráfico), un componente COTS, o un módulo de código. Ejemplos: documento de especificación de requisitos, plan de calidad, módulo de clase, rutina, informe de pruebas, manual de usuario. Sinónimos: elemento software, producto de trabajo, ítem de producto. (Vizcaíno et al., 2006, 95)

## **2.1.6. Nube**

### **2.1.6.1. Conceptualización**

La nube es una combinación de recursos tecnológicos como redes, servidores y aplicaciones en un espacio común en el que las empresas y las personas pueden arrendar una serie de servicios y almacenamiento de acuerdo a sus necesidades específicas. (Benítez Flores et al., 2019, 52)

## **2.1.7. Infraestructura como servicio (IaaS)**

### **2.1.7.1. Conceptualización**

Es el servicio más básico del modelo de servicio de la computación en la nube. Ofrece recursos de cómputo como CPU, memoria, almacenamiento, entre muchos otros. (Nieto, 2013, 10)

## **2.1.8. Plataforma como servicio (PaaS)**

### **2.1.8.1. Conceptualización**

Este servicio ofrece herramientas de software orientadas a desarrolladores de aplicaciones. Permite abstraer al desarrollador de las particularidades de la infraestructura proveyendo una capa de más alto nivel para administrarla. Desde el punto de vista del desarrollador de aplicaciones, la plataforma funciona como un intermediario entre su aplicación y la infraestructura en la nube del proveedor. (Nieto, 2013, 10)

## **2.1.9. Software como servicio (SaaS)**

### **2.1.9.1. Conceptualización**

Es el servicio que permite al usuario utilizar aplicaciones que están corriendo en una infraestructura cloud. Estas aplicaciones son accesibles desde diferentes dispositivos a través de un browser. El consumidor de este servicio no administra ni tiene control sobre la infraestructura cloud subyacente, como ser la red, los servidores, sistemas operativos, almacenamiento, ni siquiera sobre las características técnicas de la aplicación. (Nieto, 2013, 11)

## **2.2. Marco Teórico**

### **2.2.1. Características del Proceso de Inventario a nivel mundial**

No existe realmente una segmentación regional al momento de evaluar los procesos de inventarios de las compañías alrededor del mundo, más bien este es un proceso que cada empresa adapta a sus necesidades, por lo que podemos notar similitudes entre empresas de diferentes países y grandes diferencias entre empresas del mismo país.

Muchas empresas optan por mantener sus procesos internos como secretos comerciales<sup>1</sup>, siendo con frecuencia el manejo de inventarios uno de estos procesos. Esta impermeabilidad en las empresas es otro de los factores que influyen en el hecho de que no existan instituciones internacionales que regulen este proceso, dejando que cada empresa decida qué métodos utilizar para administrar sus inventarios.

Dependiendo del tipo de empresa, el proceso de inventario puede tener un gran impacto en las ganancias directas e indirectas de la misma, implicando que es de suma importancia elegir correctamente la forma, métodos y prácticas a utilizar al momento de administrar el proceso de inventario.

---

<sup>1</sup> También llamada "información confidencial", es cualquier información que sea de valor para su dueño por no ser de conocimiento general en la industria a la que pertenece.

### 2.2.2. Buenas Prácticas

Uno de los principales problemas que se deben enfrentar los administradores, gerentes y responsables de la gestión logística es la administración de los inventarios, siendo algunos de los problemas típicos la existencia de excesos y/o de faltantes debido a la variabilidad de la demanda.

Según Carlos Osorio en su artículo: Modelos para el control de inventarios en las pymes<sup>2</sup>, menciona que *“Las causas principales para acudir a la necesidad del mantener inventarios en cualquier organización son las fluctuaciones aleatorias de la demanda y de los tiempos de entrega de pedidos”* sin embargo también ofrece una lista de prácticas estratégicas para atenuar estas causas, las cuales son:

- La **obtención de información** precisa y en tiempo real sobre la demanda en el punto de consumo. A mayor información disponible oportunamente, la planeación será mucho más fácil y eficaz.
- La **consolidación** de centros de distribución y bodegas para aumentar los volúmenes de demanda por instalación, ya que más altos volúmenes de demanda conducen generalmente a menores niveles de variabilidad de la misma.

---

<sup>2</sup> Osorio, C. A. (2013, 06 17). Modelos para el control de inventarios en las pymes. PANORAMA - Revista semestral especializada en Educación y E-Learning, 2(6), 4-10.  
<https://doi.org/10.15765/pnrm.v2i6.241>

- La **estandarización de productos** para evitar el mantenimiento de inventarios de una gran diversidad de ítems que sólo difieren en aspectos menores.
- El **mejoramiento de los sistemas de pronósticos** de demanda a través de técnicas estadísticas ampliamente reconocidas.
- El **mejoramiento de alianzas** y de sistemas de comunicación con proveedores y clientes para la reducción de tiempos de entrega.
- La **emisión de órdenes conjuntas** para diversos grupos de ítems con el objeto de balancear su inventario y la consolidación de despachos desde o hacia las localidades, utilizando instalaciones como el cross-docking.
- La **reducción de demoras** a lo largo de toda la cadena de abastecimiento, incluyendo los tiempos de tránsito en los sistemas de transporte.

Otro concepto a tomar en cuenta para gestionar de forma efectiva el inventario, principalmente dentro de las PyMES, es el método a utilizar para la definición de inventarios de seguridad<sup>3</sup> y puntos de reorden<sup>4</sup>; estos usualmente se suelen definir proporcionalmente a la demanda promedio del producto exclusivamente, esta práctica lleva a desbalances inesperados del inventario, la clave está en tomar en cuenta la variabilidad en la demanda de cada producto; liberando el capital invertido en inventario de seguridad de productos con baja variabilidad y distribuirlo en inventarios de seguridad de artículos con alta variabilidad.

---

<sup>3</sup> Cantidad de productos extras que se mantienen en inventario para hacer frente a cambios inesperados en la demanda de dicho producto

<sup>4</sup> Es el nivel de existencias señalado para abastecer un artículo específico

## **2.2.3. Nivel de Automatización del Proceso de Gestión de Inventario a Nivel Mundial**

El nivel de automatización aplicado varía bastante de empresa a empresa, sin realmente tomar en consideración su ubicación regional. Al hacer un análisis de las herramientas que tienen las PyMES a su disposición para mantener un control del inventario que manejan, se pueden clasificar estas herramientas de menor a mayor nivel de automatización como podemos ver a continuación:

### **2.2.3.1. Hojas de Cálculo**

Las hojas de cálculo son aplicaciones que permiten a los usuarios organizar datos en forma de tabla (filas y columnas), estos datos, pueden ser manipulados a voluntad de los usuarios que tengan acceso a esta información.

#### **2.2.3.1.1. Principales Ventajas**

- Fácil de usar al manejar pocos datos o datos muy básicos
- Accesibles, se pueden encontrar versiones gratis y web de estas aplicaciones

#### **2.2.3.1.2. Principales Desventajas**

- La dificultad de uso aumenta exponencialmente con la cantidad de datos
- Pobre rendimiento al usar muchos datos
- Pocas herramientas predeterminadas para repostería

- Tienen un límite de datos que se puede manejar por archivo
- Difícil de compartir y trabajar en equipo

### **2.2.3.2. Aplicaciones Especializadas**

Existen aplicaciones orientadas al manejo de la contabilidad e inventario de una empresa, las mismas ofrecen un nivel de automatización y personalización superior a las hojas de cálculo por lo que casi siempre son una mejor opción, principalmente a largo plazo.

#### **2.2.3.2.1. Principales Ventajas**

- Alto nivel de automatización en la recolección y presentación de datos
- Buenas opciones de reportería
- Ayudan a realizar modelos de predicciones
- Fácil de trabajar en equipo

#### **2.2.3.2.2. Principales Desventajas**

- Accesibilidad media, suelen ser parte de un sistema más grande y costoso
- Suelen requerir de un mantenimiento o de un equipo técnico que realice esta tarea

### **2.2.3.3. Códigos de Barra**

Los códigos de barra son un identificador único por producto (no por unidad) compuesto por una parte de números y otra parte de barras verticales. Estos códigos se utilizan como una extensión de las aplicaciones especializadas de gestión de inventario para mejorar aún más el nivel de automatización de este proceso.

#### **2.2.3.3.1. Principales Ventajas**

- Agiliza la recolección de datos
- Mejora la precisión de los datos de entrada y salida
- Fácil implementación

#### **2.2.3.3.2. Principales Desventajas**

- Se necesita de herramientas especiales para su generación, impresión, colocación y lectura
- Requiere de visión directa para realizar una lectura efectiva

#### **2.2.3.4. RFID<sup>5</sup>**

Es un sistema para almacenar y leer datos remotos alojados en pequeños dispositivos denominados etiquetas, tarjetas o transpondedores RFID, a diferencia de los códigos de barra, que utiliza imágenes o números para identificar un producto, la RFID utiliza ondas de radio para comunicarse con un microchip, esto permite a la RFID identificar a una unidad específica de un producto determinado.

##### **2.2.3.4.1. Principales Ventajas**

- La recolección de datos es totalmente automatizada, no requiere de intervención humana
- Permite la trazabilidad exacta de una unidad específica a través del almacén
- No requiere de visión directa para realizar una lectura efectiva

##### **2.2.3.4.2. Principales Desventajas**

- Se necesita de herramientas e infraestructura especiales para su configuración y lectura

---

<sup>5</sup> Identificación por radiofrecuencia (del inglés Radio Frequency Identification)

## 2.2.4. Tipos de Inventario

Los inventarios se pueden clasificar de acuerdo al criterio que se considere, como se presenta en el siguiente cuadro:

Criterio	Concepto	Tipo de inventarios
<b>Funcional</b>	Se considera tomando la función o naturaleza de la empresa. El inventario dependerá si es una empresa manufacturera, comercial o de servicio.	<ul style="list-style-type: none"> <li>• Materia Prima (material utilizado como punto de partida para el proceso de producción)</li> <li>• Productos en proceso (productos que están sin terminar)</li> <li>• Productos terminados (productos que están listos para la venta, envío o consumidor final)</li> </ul>
<b>Razones para mantenerlo</b>	Depende del motivo por el cual se mantiene el inventario en una empresa.	<ul style="list-style-type: none"> <li>• Precautelativo (material o productos terminados como medida de prevención por una demanda mayor)</li> <li>• Transaccional u operativo (mercancía operable que dispone la empresa para funcionar y generar recursos y ganancias)</li> <li>• Especulativo (material o productos terminados retenido para obtener mayores ganancias debido a la variación de los precios que experimentan los productos destinados a la venta)</li> </ul>
<b>Duración</b>	Su clasificación depende de la durabilidad del mismo.	<ul style="list-style-type: none"> <li>• Perecedero (mercancía que tiene fecha de vencimiento)</li> <li>• No perecedero (mercancía que no se vencen)</li> </ul>
<b>Origen</b>	Se considera el inventario de acuerdo a la procedencia del inventario.	<ul style="list-style-type: none"> <li>• Importados (mercancía fabricada y proveniente del exterior del país)</li> <li>• Nacionales (mercancía elaborada y adquirida dentro del país)</li> </ul>
<b>Valor (Pareto)</b>	Se clasifica el inventario por la forma como se establece el precio de un inventario.	<ul style="list-style-type: none"> <li>• Grupo A (Mayor valor –se mantiene pocas cantidades-)</li> <li>• Grupo B (Valor medio –cantidades medias-)</li> <li>• Grupo C (Bajo valor –se mantiene grandes cantidades-)</li> </ul>
<b>Tipo de producto</b>	Se clasifica de acuerdo a la naturaleza y rotación del inventario. Es decir, de la forma como está compuesto el inventario físicamente.	Empresa licorera <ul style="list-style-type: none"> <li>• Whisky</li> <li>• Ron</li> <li>• Vino</li> <li>• Cerveza</li> </ul>

Tabla 4. Tabla de tipos de inventario tomada de <https://www.redalyc.org/pdf/4655/465545892008.pdf> (Durán, 2012, 63)

### **2.2.5. Regulación en la Gestión de Inventario a Nivel Mundial**

Debido a la alta variabilidad del proceso de inventario, no existe oficialmente una institución formal que regule este proceso, sin embargo, sí existen instituciones que ofrecen certificación de estandarización del proceso de inventario, siendo la más conocida el estándar **IAS2**<sup>6</sup> definido por la organización sin fines de lucro **IFRS**<sup>7</sup>.

El objetivo del estándar AIS2 es prescribir el tratamiento contable de los inventarios, por lo que no estandariza la organización o el mantenimiento del inventario, sino la percepción contable del mismo.

### **2.2.6. Métodos de Costeo de Inventario**

Los métodos de costeo se utilizan como herramienta para la contabilización formal de un inventario, los principales métodos de valuación de inventarios son: método del costo promedio o ponderado, primeras en entrar primeras en salir (PEPS o FIFO), últimas en entrar primeras en salir (UEPS o LIFO)

#### **2.2.6.1. Método del Costo Promedio o Ponderado**

Este método pondera el costo por unidad como el costo unitario promedio durante un periodo contable.

---

<sup>6</sup> Más detalles en: <https://www.ifrs.org/issued-standards/list-of-standards/ias-2-inventories/#about>

<sup>7</sup> Más detalles en: <https://www.ifrs.org/about-us/who-we-are/>

#### **2.2.6.1.1. Ventajas**

- De fácil aplicación
- En una economía inflacionaria presenta una utilidad razonable ya que promedia costos antiguos y actuales

#### **2.2.6.1.2. Desventajas**

- No permite llevar un control detallado del costo de la mercadería entrante y saliente

#### **2.2.6.2. Primeras en Entrar Primeras en Salir**

Bajo este método se debe llevar un registro del costo de cada unidad comprada del inventario y los primeros costos que entran al inventario son los primeros costos que salen al costo de las mercancías vendidas.

#### **2.2.6.2.1. Ventajas**

- El inventario final queda valuado al precio de la últimas compras
- El costo de los inventarios vendidos se valúa al costo de las primeras compras por lo que el costo reconocido en el estado de resultado es menor que el reconocido por los otros métodos de valuación
- El costo menor en el estado de resultado resulta en una utilidad bruta mayor que la obtenida con los otros métodos

#### **2.2.6.2.2. Desventajas**

- La utilidad mayor repercute en un mayor pago de impuestos
- En una economía inflacionaria puede presentar una utilidad exagerada ya que confronta costos de compra antiguos con precios de venta actuales

#### **2.2.6.3. Últimas en Entrar Primeras en Salir**

Los últimos costos que entran al inventario son los primeros costos que salen al costo de mercancías vendidas.

##### **2.2.6.3.1. Ventajas**

- Confronta costos actuales con precios de venta actuales por lo que en el Estado de Resultado refleja una utilidad menor que los demás métodos
- La utilidad menor permite un menor pago de impuestos

##### **2.2.6.3.2. Desventajas**

- El inventario final queda valuado según costos antiguos

## **2.2.7. Softwares de Gestión de Inventario para PyMES en República Dominicana**

Existen diferentes software de inventario disponibles en mercado, según las necesidades de la empresa, a continuación se presentarán algunos de estos.

### **2.2.7.1. Holded**

Es un sistema contable con un módulo que permite la gestión de inventario, por lo que ofrece varias opciones importantes; aún así no posee algún tipo de reportería o gráficos que ayuden a las empresas en la toma de decisiones durante la gestión de inventario.

### **2.2.7.2. FACTUSOL 360**

Es un programa de facturación con opciones para gestión de inventario, tiene la ventaja de estar alojada en la nube, sin embargo, no tiene herramienta de reportería, ni posee un API que permita la conexión con aplicaciones externas.

### **2.2.7.3. Cashflow**

Es un sistema contable sofisticado que cuenta con soporte para la gestión de inventario permitiendo administrar los artículos de ventas de la empresa y a pesar de tener integración con otros sistemas como Ebay o Amazon, no posee su propia API.

## Conclusión

Conforme a lo desarrollado en este capítulo, se dieron a conocer los conceptos fundamentales relacionados al inventario y la nube, con el enfoque de comprender estos conceptos y algo de historia de los mismos, junto a cómo se han desarrollado en el transcurso del tiempo.

Se analizó la forma en la que las empresas alrededor del mundo manejan sus inventarios, y cómo varía de empresa a empresa. También se dio a conocer las mejores prácticas y estándares conocidos para gestionar el inventario de forma eficaz.

Por último se listaron algunas de las aplicaciones más conocidas para la gestión de inventario dentro de las PyMES, junto a las ventajas y desventajas que tienen cada una de estas.

## **Capítulo III: Estudio de Factibilidad de la Aplicación**

## Introducción

En el siguiente capítulo se estará presentando el estudio de factibilidad sobre la implementación de un sistema de inventario por suscripción para las PyMES de Santo Domingo.

En este capítulo también se expone la situación actual de las PyMES y cómo realizan este proceso, la reducción de los costos, la innovación del producto propuesto, las ventajas y el planteo del análisis de factibilidad económica, técnica y social.

Finalmente, el capítulo abarca los aspectos técnicos que traerá la implementación de esta solución y los resultados de las encuestas realizadas a las PyMES de Santo Domingo, con la interpretación de los resultados obtenidos de estas.

### **3.1. Situación Actual**

En República Dominicana, 61.6% de las PyMEs se encuentran involucradas en actividades comerciales que requieren la gestión de inventarios de productos y una gran proporción de estas empresas no cuentan con un sistema de software adecuado para gestionar adecuadamente este proceso o simplemente no cuentan con la infraestructura necesaria; Por lo tanto, administran su inventario manualmente.

El inventario es una parte fundamental del proceso de comercialización del producto, por lo que una buena gestión es fundamental para el desarrollo exitoso de cualquier negocio. Sin embargo, no todas las PyMEs tienen acceso a una adecuada gestión de inventarios.

La falta de opciones de software de administración de inventario en el mercado actual de República Dominicana ha llevado a muchas PyMEs a: administrar su inventario con aplicaciones no diseñadas específicamente para la tarea, utilizar métodos manuales o no administrarlos directamente en los registros de inventario lo que genera pérdidas tales como: tiempo, productos, dinero, clientes potenciales y, en última instancia, falta de interés en organizar este proceso.

## **3.2. Estudio de Factibilidad**

El estudio de factibilidad tiene como objetivo dar a conocer las ventajas que suponen para las PyMES el desarrollo de una aplicación en la nube para la gestión del proceso de inventario.

### **3.2.1. Innovación del Producto Propuesto**

Un sistema que administre el proceso de inventario representa un avance colosal para las PyMES, y abre la puerta a la virtualización de los procesos relacionados y consecuentes de la institución, junto a los beneficios de confiabilidad, eficacia y rapidez en la gestión del proceso mencionado.

Esta aplicación contará con herramientas y funcionalidades aparte de agregar un valor añadido al proceso de inventario, también exhibe características que lo separan de las aplicaciones que existen actualmente; las cuales son:

- La implementación de un sistema gráfico de predicción de demanda de los productos, dando una visibilidad al futuro próximo de un producto específico, permitiendo tomar decisiones más adecuadas a la hora del reabastecimiento.
- La implementación de un API web que permita la interacción de otros procesos y/o aplicaciones, aportando un nivel de flexibilidad no visto en el mercado actual.

### **3.2.2. Reducción de Costos**

A continuación, se listan los principales costos que se disminuirán o eliminarán a la hora de implementar una aplicación web para las PyMES:

- Pérdida de productos perecederos ocasionada por la pobre administración de inventario.
- Pérdida de ventas y deficiencia en la producción por falta disponibilidad no prevista.
- Materiales gastables como papel tinta de impresoras, lapiceros, folders, entre otros artículos utilizados para administrar el inventario.
- Reducción del personal requerido para administrar el inventario.

### **3.2.3. Análisis de Factibilidad Técnica**

Esta aplicación se encontrará disponible en internet disponible para todas las PyMES, siempre y cuando adquieran una suscripción a la misma, por lo que no necesitarán de un personal técnico que le desarrolle y/o le implemente el sistema.

Esta suscripción será necesaria para poder tener acceso al sistema y contará con varios de precios lo que dará mayor flexibilidad a que las PyMES que deseen tener acceso al servicio.

Esta solución se desarrollará utilizando las técnicas y tecnologías recomendadas para obtener los mejores resultados de velocidad, seguridad y mantenimiento; por lo que se necesitan computadores a fines que puedan soportar desarrollar aplicaciones de esta gama. Para desarrollar esta aplicación se necesita contar con una computadora con las siguientes características técnicas:

<b>Programación</b>	
<b>Características</b>	<b>Requisitos</b>
Procesador	Apple M1
Sistema operativo	macOS 11.6
RAM	8 GB DDR4
Almacenamiento	SSD 256 GB
Software Adicional	<ul style="list-style-type: none"> <li>● Google Chrome</li> <li>● MySQL Workbench 8.0 CE</li> <li>● Visual Studio Code</li> <li>● Net Framework 5.0</li> <li>● Adobe XD</li> <li>● StarUML</li> </ul>
<b>Pruebas y administración</b>	
<b>Características</b>	<b>Requisitos</b>
Procesador	Intel Celeron N4000
Sistema operativo	Chrome OS 94.0
RAM	4 GB LPDDR4
Almacenamiento	eMMC 32 GB locales (100 GB en la nube)
Software Adicional	<ul style="list-style-type: none"> <li>● Selenium IDE extension for Chrome</li> </ul>

Tabla 5. Tabla de características mínimas de computadores. (Elaboración propia)

### **3.2.4. Análisis de Factibilidad Económica**

La implementación de un sistema de inventario por suscripción para las PyMES de Santo Domingo será factible por las siguientes razones:

- Tomando en cuenta que el proceso de inventario se realiza en gran parte de forma manual o empírica, el tiempo dedicado por los empleados disminuirá de manera exponencial, ya que no tendrán que hacer inventario, porque el sistema proveerá esta información.
- Las PyMES no tendrían los gastos en materiales gastables como papel, lapiceros, tintas de impresora, folder, entre otros.
- Se reducirá casi absolutamente la pérdida de productos por vencimiento, producto extraviado de su ubicación, robo por empleados, entre otros
- Aumentarán potencialmente las ventas y la velocidad general de crecimiento de las PyMES, debido a que podrán tomar mejores decisiones de ventas para su negocio apoyándose en las herramientas de reportería del sistema.
- Los costos del personal solicitado para apoyar durante el proceso de inventario disminuirían, ya que esta información estará disponible en todo momento en la aplicación.

### 3.2.4.1. Análisis costos-beneficios

A continuación, se detalla el costo de desarrollo presupuestado para la aplicación propuesta:

Herramientas, equipos y suscripciones				
No.	Descripción	Cantidad	Precio unitario (RD\$)	Precio total (RD\$)
1	Apple MacBook Air 2020: chip Apple M1, pantalla Retina de 13 pulgadas, 8 GB de RAM, 256 GB de almacenamiento SSD	3	55,100.00	165,300
2	Acer Chromebook 315, Intel Celeron N4000, pantalla táctil IPS Full HD de 15,6", LPDDR4 de 4GB	2	17,400.00	34,800.00
No.	Descripción	Cantidad	Precio unitario (RD\$)	Precio total (RD\$)
3	Google Chrome	3	0.00	0.00
4	Suscripción por un año de Microsoft Project	1	20,880.00	20,880.00
5	MySQL Workbench 8.0 CE	2	0.00	0.00
6	Selenium IDE	1	0.00	0.00
7	Adobe xD	1	579.42 / mes	579.42
8	StarUML	1	7,482.00	7,482.00
<b>Totales</b>		<b>14</b>	<b>101,441.42</b>	<b>229,041.42</b>

Tabla 6. Tabla de costo de desarrollo presupuestado para la aplicación. (Elaboración propia)

El costo total asociados a los equipos y licencias de software estimado es de RD\$220,980.00 pesos. Este costo puede ser reducido, si el personal que desarrollará la aplicación cuenta con los equipos necesarios.

Los desarrolladores de esta aplicación ya cuentan con las herramientas mencionadas anteriormente, lo que sugiere que reducirá significativamente estos costos por licencias, equipos y suscripciones; estos costos sólo aplicarán si el equipo de trabajo no cuenta con las herramientas mencionadas.

Recursos Humanos			
Cantidad	Posición	Costo mensual	Costo total por proyecto
1	Gerente de proyecto	105,000.00	840,000.00
1	Especialista de computación en la nube	115,000.00	920,000.00
1	Ingeniero de software	80,000.00	640,000.00
1	Desarrollador web	45,000.00	360,000.00
1	Analista de calidad	30,000.00	240,000.00
1	Diseñador gráfico	35,000.00	280,000.00
<b>Total</b>		<b>410,000.00</b>	<b>3,280,000.00</b>

Tabla 7. Tabla de costo de nómina. (Elaboración propia)

El tiempo estimado para el desarrollo completo de esta aplicación son 8 meses. En las tablas presentadas anteriormente se listaron los costos de recursos humanos y de las herramientas necesarias para el desarrollo de la aplicación, al realizar la suma de estos costos obtenemos la inversión total del proyecto, siendo este de RD\$3,509,041.42

Se obtendrán las ganancias a través de planes de suscripción mensual que tendrá precios desde RD\$1,500.00 hasta los \$3,000.00 que serían pagados por las

PyMES que deseen crear una cuenta administrativa para poder tener acceso a la aplicación. Si al menos 65 PyMES tuvieran una inscripción activa con el plan de \$1,500.00 se recaudaría anualmente la cantidad de RD\$1,170,000.00, este valor pudiendo aumentar mientras más suscripciones activas existan y mientras mejores sean los planes elegidos. Bajo las condiciones definidas, la inversión inicial sería recuperada y superada en una máximo de apenas 4 años, siendo este tiempo reducible dependiendo de la cantidad y la calidad de las suscripciones activas en el sistema. Como el proyecto cuenta actualmente con parte de las herramientas y con varios de los roles necesarios el costo total del proyecto se reduciría bastante.

**Formula ROI=**  $(4,680,000.00 - 3,509,041.42) / 3,509,041.42 \Rightarrow 33.4\%$

<b>Costo total para el desarrollo del proyecto.</b>	
<b>Concepto</b>	<b>Costo</b>
Gastos de equipos y herramientas	229,041.42
Gastos del personal	3,280,000.00
<b>Total</b>	<b>3,509,041.42</b>

*Tabla 8. Tabla de costo total del proyecto. (Elaboración propia)*

### **3.2.5. Modelo del Negocio**

Se utilizará un modelo de negocio basado en la nube a través de una suscripción mensual que se cobrará individualmente a cada PyME para poder tener acceso a la aplicación, esta suscripción tendrá diferentes precios dependiendo de las funcionalidades que necesite la empresa, para así poder alcanzar la mayor

cantidad de segmentos de mercado posibles. Dicha suscripción cubrirá los gastos de la infraestructura y mantenimiento de la aplicación.

### **3.2.6. Análisis de Factibilidad Operativa**

El análisis de la factibilidad operativa dará a conocer el punto de vistas de las PyMES sobre la utilización de esta aplicación en su empresa y si la misma cuentan con el personal necesario para poder administrar el sistema. Para obtener información útil sobre la viabilidad, se realizará un análisis para ver si el personal considera la implementación beneficiosa y/o necesaria.

### **3.2.7. Estudio de Mercado**

La mayoría de las PyMES en Santo Domingo no cuentan con herramientas útiles para administrar eficientemente el inventario de su empresa y que le permita identificar patrones que les ayude a tomar buenas decisiones de abastecimiento.

Se conoce de algunas PyMES que utilizan hojas de cálculo para llevar un seguimiento del inventario de su empresa, sin embargo, estos tipos de aplicaciones no están diseñadas explícitamente para este propósito, por lo que arrastrar consigo varias desventajas, como son: lentitud cuando se trabaja con muchas información, es extremadamente complicado y tedioso compartir el archivo con otras personas, requiere un alto nivel de conocimientos para realizar

cálculos complejos y suelen tener un límite de registros por archivo.

Existen también PyMES que no utilizan ningún tipo de herramientas para administrar su inventario, volviendo su gestión increíblemente lenta, ineficiente e inexacta, llevando a estas empresas a perder control casi total de este proceso.

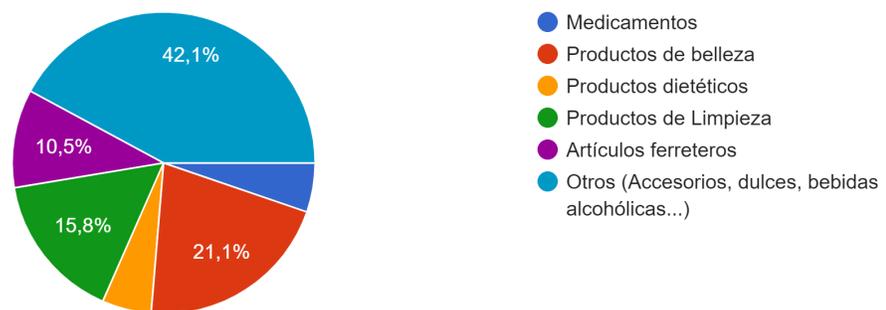
### 3.3. Encuesta

Para poder obtener la información necesaria sobre la realidad y situación actual sobre cómo el proceso de inventario afecta a las PyMES de Santo Domingo y para comprobar la viabilidad de la aplicación, se realizó una encuesta dirigidas a estas PyMES.

#### 3.4.1. Presentación y Análisis de los Resultados

##### 3.4.1.1. Resultados Encuesta Realizada a Propietarios de PyMES

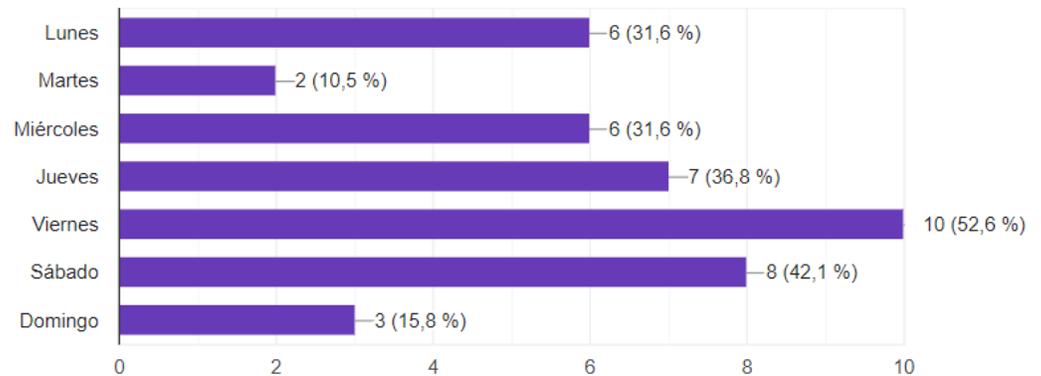
Se presentan los resultados obtenidos en la encuesta realizada a los propietarios de las PyMES, que recopila la experiencia de los propietarios en la realización del proceso de inventario. Se tomaron 19 respuestas para su preparación.



*Gráfica 1. Representación gráfica de los resultados de la pregunta “Tipos de productos / servicios suplidos por su empresa” (Construcción propia)*

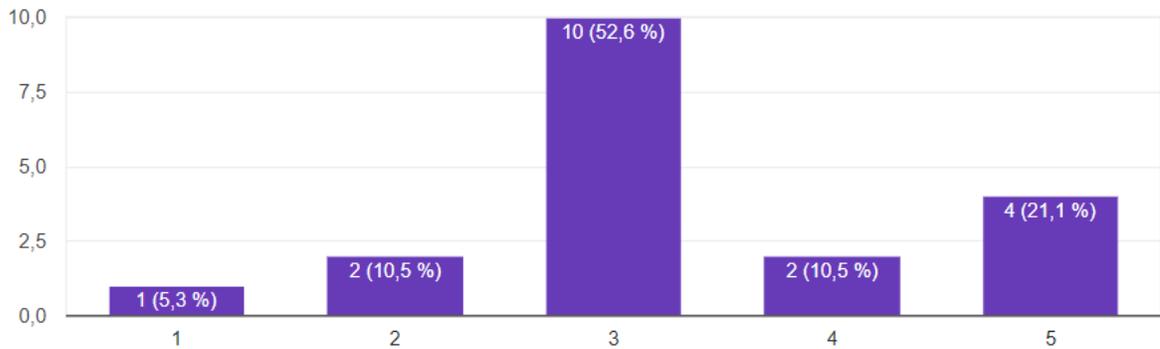
En la gráfica se puede visualizar que la mayor parte de los encuestados se dedican a la venta de medicamentos con una participación del 42,1%, en la venta

de productos de belleza del 21,1%, en la venta de productos de limpieza de un 15,8%, en la venta de artículos ferreteros de un 10,5%, mientras que la minoría se dedica a la venta de a la venta de productos dietéticos y otros con un 5,3%



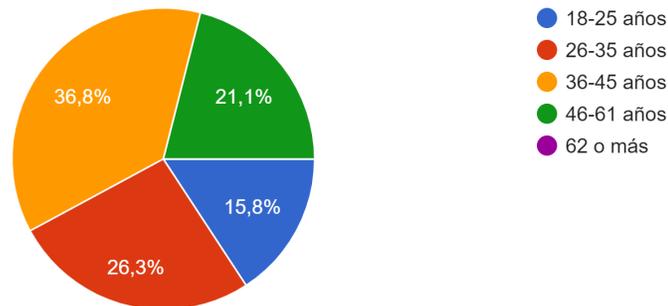
*Gráfica 2. Representación gráfica de los resultados de la pregunta "Indicar los días picos (los días de mayor concurrencia) dentro de su proceso de inventario" (Construcción propia)*

En el gráfico se puede observar los días con mayor concurrencia para llevar a cabo el proceso de inventarios, viernes con un porcentaje de 52,6% siendo este el día con mayor concurrencia a diferencia de los días, sábado con un porcentaje de 42,1%, jueves con un porcentaje de 36,8%, lunes y miércoles con un porcentaje de 31,6%, domingo con un porcentaje de 15,8% y ,martes con un porcentaje de un 10,5% siendo el día con menos concurrencia para la realización del proceso de inventario.



Gráfica 3. Gráfica representativa de los resultados de la pregunta “¿Con qué frecuencia realizan pedidos para reabastecerse? (Marque del 1 al 5, donde 1 = Diario, 2= 2 veces x semana, 3= Semanal, 4 = quincenal y 5 = mensual)” (Construcción propia)

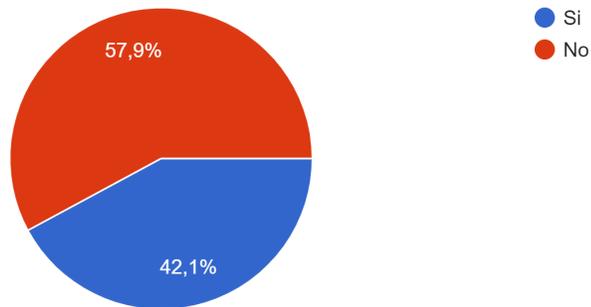
Como se puede visualizar en el gráfico, la mayoría de los encuestados, representando un 52,6% del total de los participantes, realizan pedidos para reabastecerse semana, el 21,1% se reabastece mensual, el 10,5% se 2 veces por semana y quincenal, siendo el de menor frecuencia diario con un 5,3%



Gráfica 4. Representación gráfica de los resultados de la pregunta “¿Dentro de cuál rango de edad se encuentra usted?” (Construcción propia)

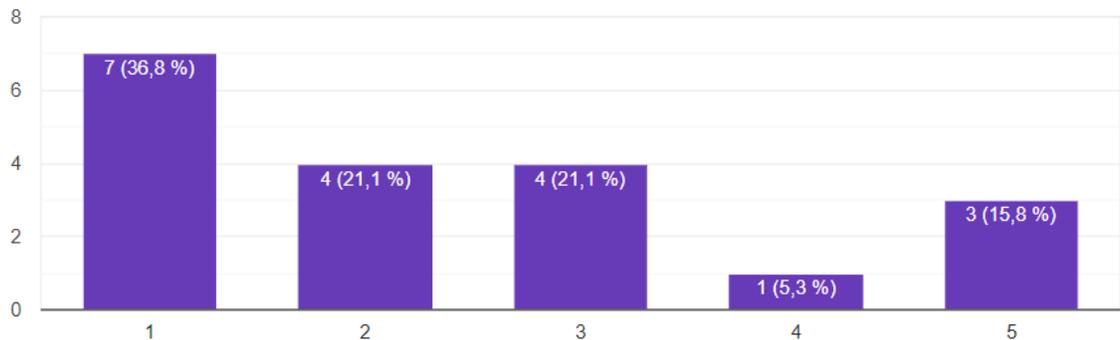
Como se puede visualizar en el gráfico, la gran mayoría de los encuestados, representando por un 36% del total de los participantes, se encuentran en la edad

de 36-45 años, mientras que la minoría se encuentra en la edad de 18-25 años.



Gráfica 5. Representación gráfica de los resultados de la pregunta “¿Son sus productos perecederos?” (Construcción propia)

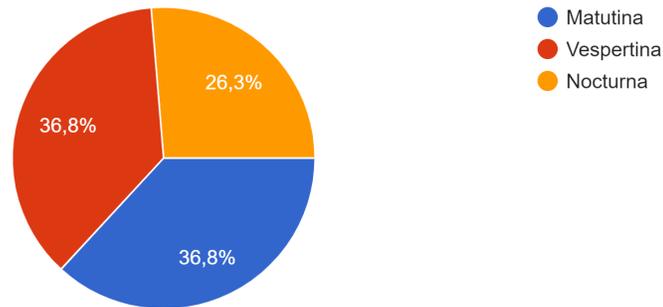
Como se puede visualizar en el gráfico, la mayoría de los encuestados, representando por un 57,9% del total de los participantes, no tienen productos perecederos, mientras que el 42,1% si tiene productos perecederos.



Gráfica 6. Representación gráfica de los resultados de la pregunta “¿Qué tan frecuente es que una persona se vaya del establecimiento por no encontrar un producto deseado? (Marque del 1 al 5, donde 1 = Muy Inusual y 5 = Muy Frecuente)” (Construcción propia)

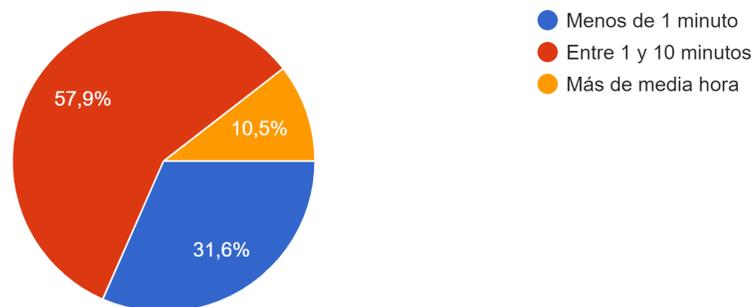
Según se visualiza en la gráfica, el 36,8% de los encuestados indicó que es muy

inusual que una persona se vaya de su establecimiento por no encontrar algún producto.



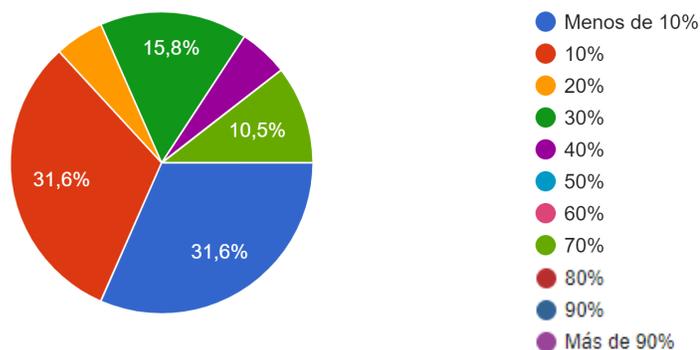
Gráfica 7. Representación gráfica de los resultados de la pregunta “¿En qué tanda suelen realizar con más frecuencia el conteo físico de productos (toma física)?”. (Construcción propia)

Con el fin de conocer en qué tanda suelen realizar con más frecuencia el conteo físico de los productos, el 36,8% de los encuestados indicó que se realiza con más frecuencia en la tanda matutina y vespertina siendo la tanda nocturna la menos frecuente con un 26,3%.



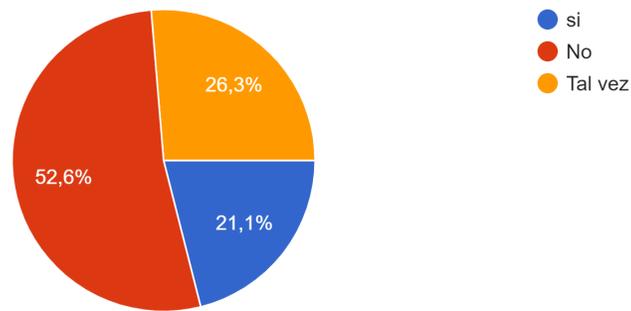
Gráfica 8. Representación gráfica de los resultados de la pregunta “¿Cuál es el tiempo promedio que tarda una persona desde que inicia la búsqueda de un producto hasta que lo localiza en el establecimiento?”. (Construcción propia)

Según se visualiza en el gráfico el 57,9% de los encuestado indicó que se toma entre 1 y 10 minutos en buscar un producto en su establecimiento, el 31,6% indicó que se toma menos de 1 minuto en buscar un producto en su establecimiento y el 10,5% se toma más de media hora.



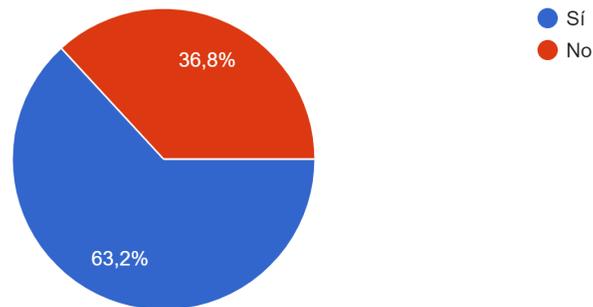
*Gráfica 9. Representación gráfica de los resultados de la pregunta ¿Cuál es el % de clientes que entiendes se retiran disgustados del establecimiento (por tardanza, por expectativas distintas en cuanto al embalaje, porque no encontró su producto, porque el producto que había no era exactamente el que buscaba, etc.)? (Construcción propia)*

Como se visualiza en el gráfico, el 31,6% de los encuestado indicó que el 10% y menos del 10% de los clientes se retiran disgustados del establecimiento por el tiempo que se toman para ser atendidos, el 15,8% indicó que el 30% se retiran disgustados del establecimiento por el tiempo que se toman en ser atendidos y el 10,5% indicó que el 70% se retiran disgustados del establecimiento por el tiempo que se toman en ser atendidos.



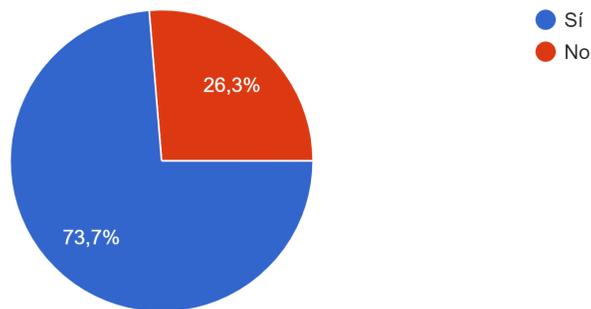
Gráfica 10. Representación gráfica de los resultados de la pregunta "¿Ha sido objeto de robo tu empresa por empleados que sustraen artículos?" (Construcción propia)

Como se puede visualizar en el gráfico, el 52,6% de los encuestados indicaron que no han sido objeto de robo en su establecimiento, el 26,3% indicaron que tal vez han sido objeto de robo en su establecimiento y el 21,1% indicó que si ha sido objeto de robo en su establecimiento.



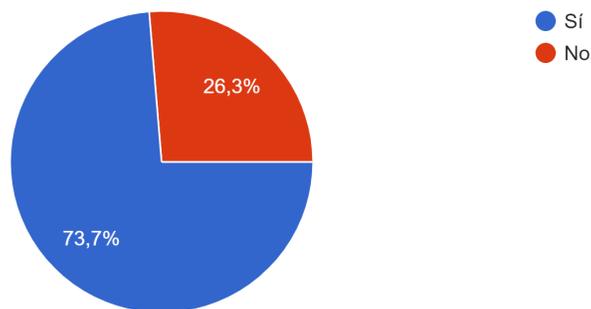
Gráfica 11. Representación gráfica de los resultados de la pregunta "¿Conoce del abastecimiento 'Justo a tiempo'?" (Construcción propia)

En la gráfica se muestra que el 63,2% si conoce el abastecimiento justo a tiempo, mientras que el 36,8% no lo conoce.



Gráfica 12. Representación gráfica de los resultados de la pregunta "¿Sabe de cual proveedor es un artículo que haya sido devuelto por estar defectuoso o por que se haya deteriorado antes de tiempo?" (Construcción propia)

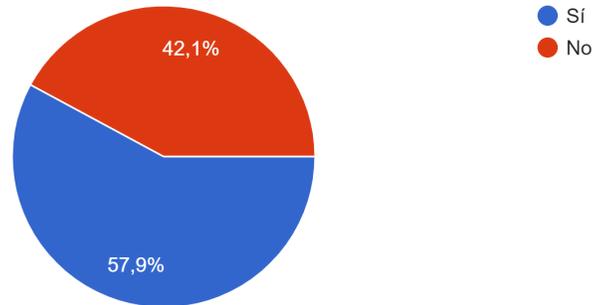
En la gráfica 73,7% indicó que si conoce cual es el proveedor de un artículo que haya sido devuelto por estar defectuoso o por que se haya deteriorado antes de tiempo , mientras que el 26,3% no conoce el proveedor.



Gráfica 13. Representación gráfica de los resultados de la pregunta "¿Sabes en que % han aumentado o disminuido el costo de tus artículos comprados?" (Construcción propia)

Según el gráfico el 73,7% indicó que si sabe en qué porcentaje han aumentado o disminuido un producto, mientras que el 26,3% no conoce el porcentaje que ha

aumentado o disminuido su producto.

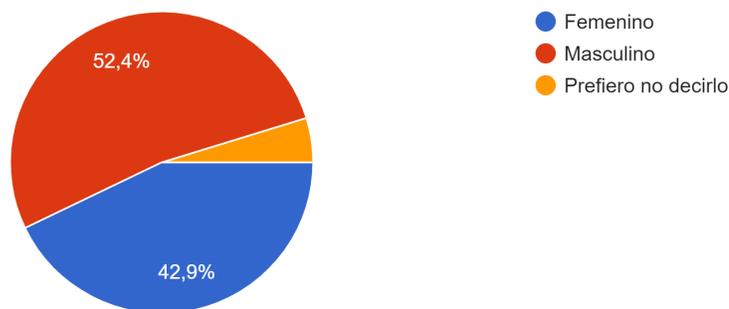


Gráfica 14. Representación gráfica de los resultados de la pregunta “¿Usas algún método para costear tu inventario (UEPS, PEPS, promedio)?” (Construcción propia)

En la presente gráfica se muestra que el 57,9% de los encuestados utiliza algún método para costear el inventario, mientras que el 42,1% no utiliza algún método de costeo para el inventario.

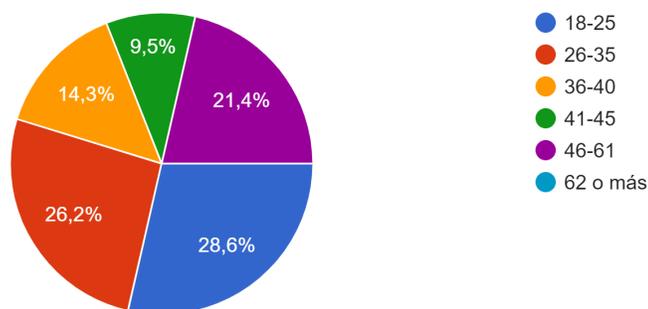
#### 3.4.1.2. Resultados de Encuesta Realizada a Empleados Operativos de las PyMES

Se presentan los resultados obtenidos de la encuesta para los empleados operativos de las empresas categorizadas como PyMES que recopila la experiencia de las personas que han trabajado directamente en la realización de inventario. Se tomaron 42 respuestas para su preparación.



Gráfica 15. Representación gráfica de los resultados de la pregunta ¿Cuál es su género?. (Construcción propia)

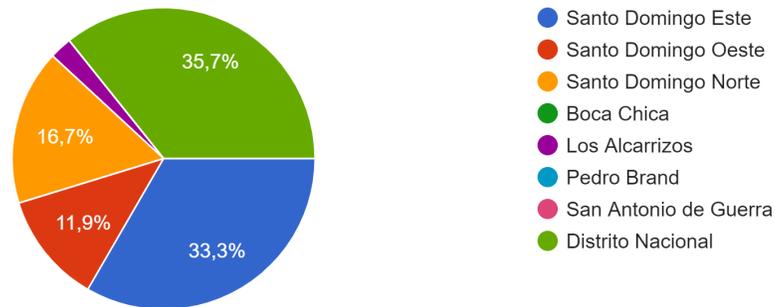
En el presente gráfico el 52,4% de los encuestados es masculino, el 42,9% de los encuestados es de género femenino, mientras que el 4,8% prefiere no decir su género.



Gráfica 16. Representación gráfica de los resultados del rango de edades de los encuestados. (Construcción propia)

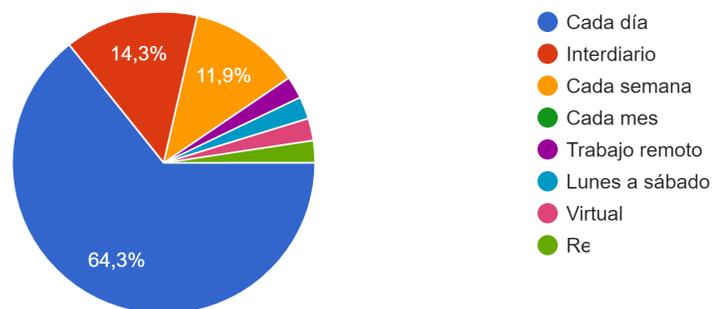
En el gráfico se puede observar el rango de edades de los encuestados. Según se puede visualizar, la mayor parte de los participantes de la encuesta se encuentran entre los 18 a los 25 años, representando un 28,6% del total de los encuestados; y la segunda mayoría se encuentra entre los mayores de 26 a los 35 años,

representando el 26,2% del total de los participantes.



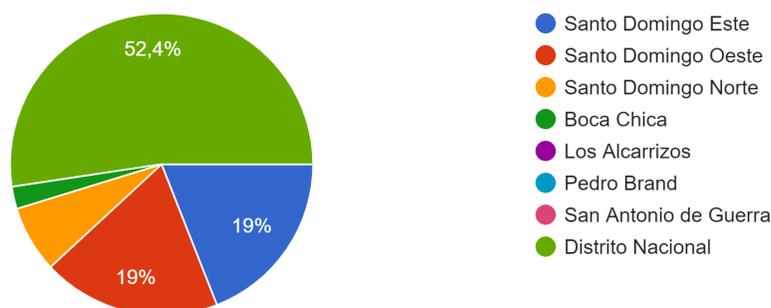
Gráfica 17. Representación gráfica de los resultados de la pregunta “¿En qué municipio te ubicas? . (Construcción propia)

En la gráfica se puede visualizar que la mayor parte de los encuestados se ubican en el Distrito Nacional, representado en 35,7% de los encuestados, el 33,3% se ubica en Santo Domingo Este, el 16,7% de los encuestado se ubica en Santo Domingo Norte, el 11,9% de los encuestados se ubica en Santo Domingo Oeste y el 2,4% de los encuestados se ubica en Los Alcarrizos.



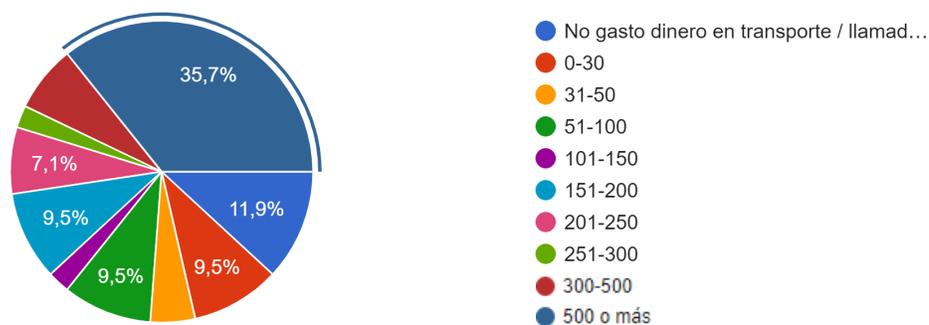
Gráfica 18. Representación gráfica de la pregunta “¿Con qué frecuencia asistes a la empresa a la que labora en este periodo de pandemia?”. (Construcción propia)

En esta gráfica se muestra como el 64.3% de los encuestados asiste de forma diaria a la empresa en la que labora, mientras que el 14.3% asiste de forma interdiaria y el 11.9% semanalmente.



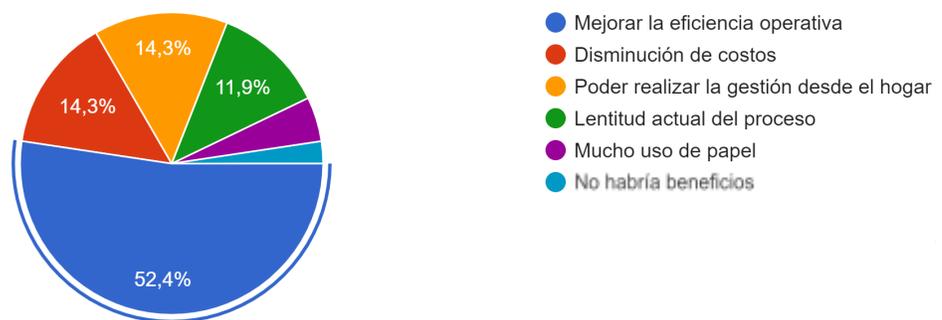
Gráfica 19. Representación gráfica de los resultados de la pregunta “¿Dónde está ubicada la empresa para la cual laboras?”. (Construcción propia)

Como puede verse en la gráfica, la mayoría de las personas laboran en Distrito Nacional siendo un 52,4% de los encuestados, mientras que los que trabajan en Santo Domingo Este y Santo Domingo Oeste son un 19% de los encuestados.



Gráfica 20. Representación gráfica de los resultados de la pregunta “¿Cuánto dinero consume en transporte, llamadas al mantener contacto con otros empleados u empresas mientras gestiona el inventario? (Los montos están representados en pesos dominicanos)”. (Construcción propia)

En la presente gráfica puede verse que el 35,7% de los encuestados gastan 500 pesos o más en transporte mientras gestiona el inventario siendo este el mayor porcentaje.



Gráfica 21. Representación gráfica de los resultados de la pregunta “¿Cuál de las siguientes opciones es la principal razón por la cual usted entiende prudente que la empresa adquiera una aplicación para gestionar el inventario?”. (Construcción propia)

En esta gráfica puede que el 52,4% de los encuestados entienda que la principal razón para adquirir una aplicación de gestión del inventario mejoraría la eficiencia operativa.

### **3.4. Beneficios**

Los beneficios asociados a la utilización de esta aplicación web, son los siguientes:

- Reducción del tiempo invertido en mantener la información del inventario actualizada
- Simplificación del proceso de inventario
- Reducción de pérdida de dinero, productos y posibles clientes por factores como caducidad o falta de existencia de productos primarios
- Mejor toma de decisiones de las PyMES al momento del reabastecimiento
- Aumento en la motivación de las PyMES en mejorar los procesos relacionados al inventario

## Conclusión

En este capítulo se analizó la situación actual de las PyMES de Santo Domingo, desarrollando un estudio de factibilidad realizando análisis como: estudio de mercado, factibilidad operativa, aspectos técnicos y sociales, entre otros.

También se presentó el resultado de las encuestas realizadas a las PyMES para la obtención de información relevante para la toma de decisiones y diseño de esta propuesta, tomando en cuenta estos resultados podemos determinar que cerca del 92% de los encuestados no están contentos con su situación actual y entienden prudente la creación de una aplicación que permita automatizar dicho proceso.

## **Capítulo IV: Análisis y Diseño del Sistema**

## Introducción

En este capítulo se tratan los distintos apartados y documentos necesarios para el análisis y diseño de la solución de software propuesta desde el inicio de este trabajo, incluyendo el estatuto del proyecto, los requisitos funcionales y no funcionales, la visión y alcance del documento, los casos de uso, diseños de alto y bajo nivel y diseños de vista web.

## 4.1. Acta de constitución del Proyecto

### 4.1.1. Nombre del Proyecto / Información del Proyecto

Aplicación web por suscripción para la gestión de inventario, “**Tarium**”.

Proyecto	Aplicación web por suscripción para la gestión de inventario, “Tarium”
Fecha de preparación	03/01/2022
Cliente	PyMES de Santo Domingo
Patrocinador principal	N/A

*Tabla 9. Tabla inicial del proyecto. (Elaboración propia)*

### 4.1.2. Propósito y Justificación de proyecto

El propósito de este proyecto es ofrecer a las PyMES de Santo Domingo una plataforma en la cual puedan gestionar la disponibilidad de sus productos, obtener información relevante del movimiento de su inventario y manejar de forma efectiva el reabastecimiento, con el fin de mejorar el tiempo dedicado a la gestión y hacer este proceso más cómodo y eficiente

### 4.1.3. Alcance

Este proyecto contempla el análisis, diseño, desarrollo e implementación de un Sistema de Inventario por Suscripción para PyMES Bajo la Arquitectura SaaS en la Nube en Santo Domingo, República Dominicana.

#### 4.1.4. Descripción del Proyecto y Entregables

Este proyecto tiene como objetivo ofrecer una aplicación web que le permita a las PyMES de Santo Domingo poder administrar sus inventarios de forma rápida, sencilla y efectiva. Esta aplicación contará con herramientas de reportería y predicciones para optimizar la toma de decisiones para el reabastecimiento.

Los entregables de este proyecto son los siguientes:

- Acta constitutiva del proyecto
- Captura de los requerimientos
- Diagrama de arquitectura de la aplicación
- Diagrama de la Base de Datos
- Cronograma de actividades del proyecto
- Diseño de UI/ Interfaces Gráficas
- Prototipo de la aplicación

#### 4.1.5. Premisas y Restricción

<b>Premisas</b>	<b>Restricciones</b>
1. Se contará con las licencias de las tecnologías a ser utilizadas para el desarrollo del proyecto	1. El desarrollo del proyecto no debe exceder el presupuesto de RD\$3,500,980.00
2. Se cuenta con dispositivos necesarios para realizar el desarrollo y las pruebas necesarias	2. Se debe cumplir con el cronograma del proyecto
3. Se documentarán apropiadamente todas las fases del proyecto	3. Se deben priorizar los requerimientos funcionales

--	--

Tabla 10 . Premisas y restricciones. (Elaboración propia)

#### 4.1.6. Riesgos Iniciales de Alto Nivel

Se identificaron los siguientes riesgos iniciales de alto nivel:

- Errores de estimación en el tiempo de desarrollo y/o presupuesto
- Cambios imprevistos en los requerimientos del proyecto
- Definición de requerimientos ambiguos
- Falta de herramientas necesarias
- Personal no capacitado
- Salida temporal o permanente de personal clave para el proyecto

#### 4.1.7. Cronograma de trabajo

Tarea	Fecha Inicio	Fecha Fin	Duración
<b>Proyecto Tarium</b>	<b>03/01/2022</b>	<b>02/09/2022</b>	<b>242 días</b>
<b>Definición y alcance</b>	<b>03/01/2022</b>	<b>05/01/2022</b>	<b>3 días</b>
Reunión Kick-off parte 1	03/01/2022	03/01/2022	2 horas
Dimensionamientos de recursos	03/01/2022	04/01/2022	1 día
Reunión Kick-off parte 2	04/01/2022	04/01/2022	2 hora
Enviar plan de trabajo	04/01/2022	04/01/2022	4 horas
Reunión Kick-off parte 3	05/01/2022	05/01/2022	1 hora
Aprobar plan de proyecto	05/01/2022	05/01/2022	1 hora

Tarea	Fecha Inicio	Fecha Fin	Duración
Definición de equipo de proyecto	05/01/2022	05/01/2022	3 horas
Inicio de proyecto completo	05/01/2022	05/01/2022	0 horas
<b>Análisis y diseño</b>	<b>06/01/2022</b>	<b>19/01/2022</b>	<b>13 días</b>
Levantamiento de los requerimientos	06/01/2022	10/01/2022	3 días
Diseño de Base de Datos	11/01/2022	11/01/2022	1 día
Diseño Arquitectura Aplicativo	12/01/2022	13/01/2022	2 días
Diseño Interfaces de Usuario	14/01/2022	19/01/2022	4 días
<b>Desarrollo marco de trabajo</b>	<b>20/01/2022</b>	<b>27/01/2022</b>	<b>7 días</b>
Creación arquitectura de aplicación web	20/01/2022	25/01/2022	3 días
Creación de la Base de Datos	26/01/2022	27/01/2022	2 días
<b>Desarrollo módulo seguridad</b>	<b>28/01/2022</b>	<b>10/02/2022</b>	<b>13 días</b>
Creación sistema de usuarios	28/01/2022	31/01/2022	2 días
Creación sistema de pagos	01/02/2022	04/02/2022	4 días
Creación sistema de permisos	07/02/2022	10/02/2022	4 días
<b>Desarrollo módulo administrativo</b>	<b>11/02/2022</b>	<b>07/04/2022</b>	<b>55 días</b>
Mantenimiento de sucursales	11/02/2022	24/02/2022	10 días
Mantenimiento de proveedores	25/02/2022	10/03/2022	10 días
Mantenimiento de productos	11/03/2022	07/04/2022	20 días
<b>Módulo de gráficas</b>	<b>08/04/2022</b>	<b>27/05/2022</b>	<b>49 días</b>
Creación de gráfico principal	08/04/2022	28/04/2022	15 días

Tarea	Fecha Inicio	Fecha Fin	Duración
Creación de sistemas de predicciones	29/04/2022	27/05/2022	21 días
<b>Pruebas e integraciones</b>	<b>30/05/2022</b>	<b>13/07/2022</b>	<b>44 días</b>
Documentar casos de prueba	30/05/2022	03/06/2022	5 días
Realizar las pruebas	06/06/2022	17/06/2022	10 días
Corregir defectos encontrados	20/06/2022	08/07/2022	15 días
Realizar reportes de pruebas	11/07/2022	13/07/2022	3 día
<b>Puesta en producción</b>	<b>14/07/2022</b>	<b>20/07/2022</b>	<b>6 días</b>
Determinar estrategia del despliegue	14/07/2022	14/07/2022	1 día
Preparar recursos para el despliegue	15/07/2022	15/07/2022	1 día
Realizar despliegue	18/07/2022	19/07/2022	2 días
Finalización del despliegue	20/07/2022	20/07/2022	1 día
<b>Documentación</b>	<b>21/07/2022</b>	<b>10/08/2022</b>	<b>20 días</b>
Redactar documentación técnica	21/07/2022	27/07/2022	5 días
Redactar documentación operativa	28/07/2022	03/08/2022	5 días
Redactar guía de usuario	04/08/2022	10/08/2022	5 días
<b>Cierre del proyecto</b>	<b>11/08/2022</b>	<b>02/09/2022</b>	<b>22 días</b>
Últimas gestiones	11/08/2022	30/08/2022	14 días
Envío de carta de cierre	31/08/2022	02/09/2022	3 días

Tabla 11. Tabla de cronograma de trabajo. (Elaboración propia)

## 4.2. Sesión de Requerimientos

### 4.2.1. Requisitos Funcionales

ID	Requisito Funcional	Actores	Prioridad
RF-01	La aplicación debe tener una vista de registro de usuario	Usuario	Alta
RF-02	La aplicación debe tener una vista para que los usuarios registren el método de pago a utilizar para el cobro de la suscripción	Usuario administrador	Alta
RF-03	La aplicación debe tener una vista que permita administrar las sucursales del usuario. Estas sucursales se podrán crear, editar y eliminar. Habrán dos tipos de sucursales a elegir: almacén o tienda	Usuario administrador	Alta
RF-04	La aplicación debe tener una vista que permita administrar los proveedores del usuario	Usuario administrador	Alta
RF-05	La aplicación debe tener una vista que permita administrar el catálogo de productos del usuario	Usuario administrador	Alta
RF-06	La aplicación debe tener una vista que permita administrar los usuarios transaccionales de la aplicación	Usuario administrador	Alta
RF-07	La aplicación debe tener una vista que permita mover productos entre sucursales. En esta vista se debe especificar el tipo de movimiento: instantáneo o transporte. Esta vista debe permitir imputación individual y masiva a través de un lector de código de barras o un archivo de formato especial	Usuario administrador Usuario transaccional	Alta

ID	Requisito Funcional	Actores	Prioridad
RF-08	La aplicación debe tener una vista que permita registrar la entrada de productos. En esta vista se debe especificar el tipo de entrada: proveedor o movimiento. Esta vista debe permitir imputación individual y masiva a través de un lector de código de barras o un archivo de formato especial	Usuario administrador Usuario transaccional	Alta
RF-09	La aplicación debe tener una vista que permita registrar la salida de productos. Esta vista debe permitir imputación individual y masiva a través de un lector de código de barras o un archivo de formato especial	Usuario administrador Usuario transaccional	Alta
RF-10	La aplicación debe tener una vista que permita visualizar el estado actual del inventario con filtros de sucursales, productos y/o rango de fecha, esta información se presentará en forma tabular y deberá poder ser exportada	Usuario administrador	Media alta
RF-11	La aplicación debe poder mantener un registro de todas las transacciones de productos realizadas en la aplicación	Sistema	Media
RF-12	La aplicación debe tener una vista que permita visualizar las transacciones de productos realizadas. Esta información debe presentarse de forma tabular y gráfica, con filtros de sucursales, productos y/o rango de fecha. La forma gráfica deberá graficar una predicción de las demandas futuras basado en el historial de transacciones	Usuario administrador	Media

ID	Requisito Funcional	Actores	Prioridad
RF-13	La aplicación debe notificar cuando algún producto está cerca de caducar	Sistema Usuario administrador	Bajo

Tabla 12. Tabla de requisitos funcionales. (Elaboración propia)

#### 4.2.2. Requisitos No Funcionales

ID	Requisito No Funcional
<b>Rendimiento</b>	
RNF-01	La aplicación debe ser óptimamente funcional al tener hasta 2,000 usuarios simultáneos
RNF-02	La aplicación debe de tener un tiempo máximo de respuesta a las peticiones de 3 segundos en secciones no gráficas
<b>Disponibilidad</b>	
RNF-03	La aplicación debe poder ejecutarse en los navegadores Chrome, Edge, Opera y Firefox, en sus versiones móviles y de escritorio.
RNF-04	La aplicación debe de estar disponible las 24 horas al día, 7 días a la semana, con excepción de los tiempos de baja previamente notificados.
<b>Mantenimiento</b>	
RNF-05	La aplicación debe de ser desarrollada utilizando una arquitectura que garantice despliegues efectivos y mantenimientos sencillos.
RNF-06	La aplicación debe ser desarrollada utilizando una herramienta de control de versiones.
<b>Seguridad</b>	
RNF-07	La aplicación debe almacenar toda información sensible utilizando métodos de encriptación seguros y confiables.
RNF-08	La aplicación debe realizar el respaldo de seguridad de los datos diariamente, con una rotación mensual.
RNF-09	La aplicación debe de ser intuitiva, fácil de utilizar y navegar, con elementos visuales conocidos popularmente.
RNF-10	La aplicación debe de cumplir con los patrones de diseño establecidos para dispositivos móviles
RNF-11	La aplicación debe de contar con una sección de ayuda al usuario.
RNF-12	Se debe de notificar de forma clara los mensajes, las informaciones, los errores y orientaciones de la aplicación.

Tabla 13. Tabla de requisitos no funcionales. (Elaboración propia)

## 4.3. Documento Visión y Alcance

PROPUESTA DE DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO POR SUSCRIPCIÓN PARA PYMES BAJO LA ARQUITECTURA SAAS EN LA NUBE, SANTO DOMINGO, 2021.

### Historial de Revisiones

Fecha	Versión	Descripción	Autores
16 nov. 2021	0.1	Primera versión del documento Visión	Emerson Pichardo

Tabla 14. Tabla historial de revisiones. (Elaboración propia)

### 4.3.1. Introducción

#### 4.3.1.1. Propósito

El propósito de este documento es definir y analizar las necesidades de alto nivel de la aplicación a desarrollar cuya función es destinada para agilizar los procesos de gestión de inventario en las PyMES de Santo Domingo.

#### 4.3.1.2. Alcance

Este documento abarca las principales funcionalidades y características para los usuarios finales que utilizarán la aplicación web.

También este documento define las oportunidades de mejoras en el proceso de gestión de inventario que las PyMES utilizan en la actualidad.

#### 4.3.1.3. Definiciones, Acrónimos y Abreviaturas

- **PyMES:** Son las siglas de pequeñas y medianas empresas.
- **APP:** Es la abreviatura de “Aplicación” y puede hacer referencia a un tipo de programa informático que se utiliza como herramienta.
- **Stakeholders:** Son todas las personas, instituciones u organizaciones de interés para una empresa, ya que permite su funcionamiento relacionándose con las actividades y decisiones de una empresa.
- **Nube:** Es una combinación de recursos tecnológicos como redes, servidores y aplicaciones en un espacio común en el que las empresas y las personas pueden arrendar una serie de servicios y almacenamiento de acuerdo a sus necesidades específicas. (Benítez Flores et al., 2019, 52)
- **UML:** Es una especificación que define un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de los sistemas de objetos distribuidos.

#### 4.3.1.4. Referencias

- Acta de constitución
- Documento de especificación de casos de usos
- Glosario

## **4.3.2. Posicionamiento**

### **4.3.2.1. Oportunidad de Negocio**

Actualmente, una gran porción de estas empresas no disponen de un sistema de software apropiado para manejar adecuadamente este proceso o simplemente no cuentan con la infraestructura necesaria; por lo que administran su inventario de forma manual.

La falta de opciones de software para administración de inventario en el mercado actual de República Dominicana ha llevado a muchas PyMES a: gestionar su inventario con aplicaciones que no se encuentran diseñadas explícitamente para esta tarea, utilizar métodos manuales, o directamente manejar el proceso de forma empírica.

Los principales inconvenientes de manejar de forma manual el proceso de inventario son la extensión del tiempo necesario para su administración, la imprecisión de los datos obtenidos y registrados, pérdida potencial de la información de inventario, complejidad en la cadena de suministro y pérdida potencial de productos y dinero.

#### 4.3.2.2. Declaración de Problema

El problema	Ineficiencia del proceso de gestión inventario de las PyMES en Santo Domingo, incluyendo el reabastecimiento del mismo
Afecta a	Las PyMES de Santo Domingo y los clientes de las mismas
El impacto asociado del problema es	<ul style="list-style-type: none"> <li>● Pérdida de productos</li> <li>● Lentitud en la administración de inventario</li> <li>● Pérdida de ventas</li> <li>● Deficiencia en la producción</li> <li>● Gastos innecesarios de materiales gastables</li> </ul>
Una solución eficaz sería	La implementación de una aplicación en la nube que permita a las PyMES gestionar su inventario de forma efectiva

Tabla 15. Tabla declaración de problema. (Elaboración propia)

#### 4.3.2.3. Declaración de Posición de Producto

Para	Las PyMES de Santo Domingo
Quienes	Carecen de una aplicación efectiva y diseñada para la administración adecuada de su inventario
Aplicación web por suscripción para la gestión de inventario, (Tarium)	Es una aplicación web que le permite a las PyMES de Santo Domingo poder administrar sus inventarios de forma rápida, sencilla y efectiva
Que	Permitirá mejorar el tiempo necesario para la administración de inventario, reducir los gastos en pérdidas de clientes y productos y mejorar la eficiencia general de su inventario
A diferencia de	<ul style="list-style-type: none"> <li>● Administrar su inventario de forma manual o empírica</li> <li>● Administrar su inventario con herramientas eficientes, no diseñadas para administrar inventario</li> </ul>

Tabla 16. Tabla declaración de posición de producto. (Elaboración propia)

### 4.3.3. Descripción de los stakeholders

En esta sección se ofrece un perfil de información de los stakeholders y usuarios involucrados en el proyecto. También se muestran los problemas principales que los usuarios y stakeholders consideran que la aplicación debe solucionar.

#### 4.3.3.1. Resumen de los Stakeholders

Nombre	Representaciones	Responsabilidades
PyMES de Santo Domingo	Son la entidad principal del proceso. En ellas está enfocado el análisis y diseño de la aplicación propuesta. Estas se dedican a la creación, distribución o comercialización de productos	Gestionar todo el proceso de inventario de su empresa

Tabla 17. Tabla resumen de los stakeholders. (Elaboración propia)

#### 4.3.3.2. Resumen de los Usuarios

Nombre	Descripción	Competencias (de cara al uso del sistema)
Usuario administrador	Este actor es el encargado del registro del catálogo de productos y de los usuarios que interactúan con este	<ul style="list-style-type: none"><li>• Experiencia en administración de aplicaciones web</li></ul>
Usuario transaccional	Este actor es el encargado de realizar las transacciones de productos	<ul style="list-style-type: none"><li>• Conocimiento del registro de productos en la plataforma</li></ul>

Tabla 18. Tabla resumen de los usuarios. (Elaboración propia)

#### 4.3.4. Alternativas y Competencia

Actualmente existen sitios web donde los usuarios pueden administrar su sistema de inventario. Aun así, estas plataformas están orientadas mayormente al aspecto contable de la empresa y no directamente en la gestión de inventario, llevando a las empresas a tener menos flexibilidad de opciones si desea un sistema de contabilidad distinto; tampoco ofrecen un sistema de predicciones de demanda futuras basadas en el historial de salidas.

Algunas de las posibles competencias son:

- Alegra
- Claro cloud

#### 4.3.5. Estándares aplicables

- ISO/IEC 27001 – Gestión de la Seguridad de la Información.
- ISO 12207 – Modelos de Ciclos de Vida del Software.
- ISO 9001:2015 – Gestión de la Calidad del Software.

#### 4.3.6. Rangos de Calidad

- **Disponibilidad:** El sistema debe estar disponible todo el día todos los días, exceptuando los momentos anticipados de actualización y mantenimiento.
- **Mantenimiento:** Debe ser un sistema que cuente con fácil actualización y mantenimiento.

- **Rendimiento:** La plataforma debe garantizar el uso de unos 45,000 usuarios a la vez.
- **Robustez:** Se debe asegurar que la plataforma cumpla con los parámetros de seguridad requeridos en la transferencia y almacenaje de la información.
- **Usabilidad:** Garantizar la facilidad de uso del usuario a través de un conjunto de características visuales y de navegación correctamente diseñadas.

#### 4.3.7. Requisitos de Documentación

- Ayuda en línea: El sistema contará con manuales de uso para los usuarios administradores y transaccionales para que conozcan cómo manejar cada sección de la plataforma

#### 4.3.8. Conclusión del Alcance

En la sección presentada se planteó toda la información del alcance, visión y objetivos de alto nivel de la aplicación web para las PyMES de Santo Domingo. Este módulo sirve como referencia o guía para tener conocimiento de los aspectos fundamentales que involucran el desarrollo de la aplicación web propuesta.

## 4.4. Casos de Uso

### 4.4.1. Listado de los Casos de Uso

ID	Caso de uso	Descripción
CUS-1	Registro de usuario administrador	Permite a los usuarios crear una cuenta de administrador al unirse a una suscripción de pago
CUS-2	Gestión de sucursales	Permite a los usuarios administradores agregar, editar y eliminar las sucursales de su empresa
CUS-3	Gestión de proveedores	Permite a los usuarios administradores agregar, editar y eliminar los proveedores de su empresa
CUS-4	Gestión de catálogo de productos	Permite a los usuarios administradores agregar, editar y eliminar productos de su catálogo
CUS-5	Gestión de usuarios transaccionales	Permite a los usuarios administradores otorgar permisos de realizar movimientos de productos a otros usuarios
CUS-6	Movimiento de productos	Permite a los usuarios transaccionales mover productos entre sucursales
CUS-7	Entrada de productos	Permite a los usuarios transaccionales registrar la entrada de productos
CUS-8	Salida de productos	Permite a los usuarios transaccionales registrar la salida de productos
CUS-9	Visualización de inventario	Permite a los usuarios administradores visualizar el estado actual del inventario y consultar las existencias de productos a través de una serie de filtros

ID	Caso de uso	Descripción
CUS-10	Visualización de transacciones de productos	Permite a los usuarios administradores visualizar las transacciones de productos a través del tiempo para conocer los costos, puntos de reabastecimiento, vencimientos y predicción de demanda de cada producto utilizando una serie de filtros

Tabla 19. Tabla listado de casos de uso (Elaboración propia)

#### 4.4.2. Diagrama General Casos de Uso

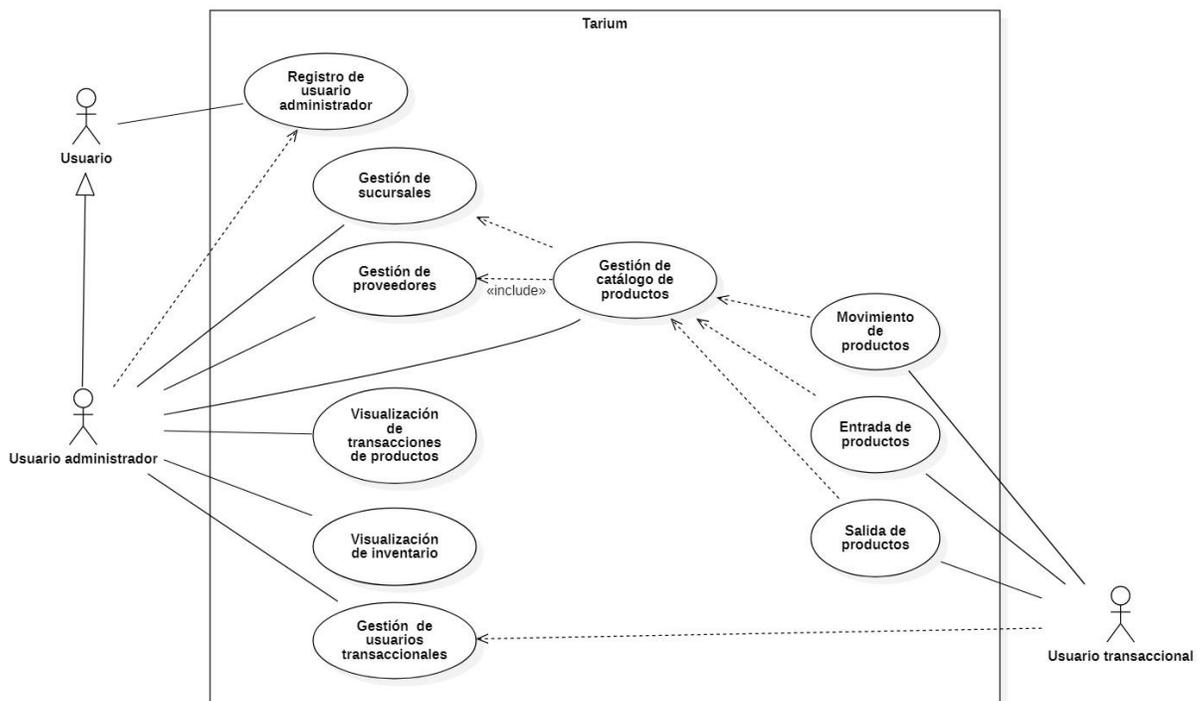


Figura 1. Diagrama general de casos de uso (Elaboración propia)

### 4.4.3. Descripción de Actores

#### 4.4.3.1. Usuario Administrador

<b>Actor</b>	Usuario administrador	<b>Identificador:</b> AC-1.0
<b>Descripción</b>	Este es el usuario administrador general del sistema es quien proporciona mantenimiento a la información de su empresa	
<b>Características</b>	Este es un actor principal que interactúa directamente con la información principal de su empresa en el sistema	
<b>Relación</b>	Este actor es el encargado del registro del catálogo de productos y de los usuarios que interactúan con este	
<b>Referencias</b>	Diagrama general de caso de uso: CUS-1, CUS-2, CUS-3, CUS-4, CUS-5, CUS-9, CUS-10, CUS-11	

Tabla 20. Descripción actor usuario administrador (Elaboración propia)

#### 4.4.3.2. Usuario Transaccional

<b>Actor</b>	Usuario transaccional	<b>Identificador:</b> AC-2.0
<b>Descripción</b>	Este es un actor secundario, está encargado del registro de los movimientos de inventario	
<b>Características</b>	Este actor registra los movimientos de productos entre sucursales y registra las entradas y salidas de productos	

<b>Relación</b>	Este actor es el encargado de realizar las transacciones de productos
<b>Referencias</b>	Diagrama general de caso de uso: CUS-6, CUS-7, CUS-8

Tabla 21. Descripción actor usuario transaccional (Elaboración propia)

#### 4.4.4. Especificación de Casos de Uso

##### 4.4.4.1. CUS-1 Registro de Usuario Administrador

<b>Caso de Uso</b>	Registro de usuario administrador	<b>Identificador:</b> CUS-1
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Usuario ingresa a la plataforma	
<b>Postcondición</b>	Nuevo usuario administrador agregado al sistema junto a una suscripción	
<b>Descripción</b>	Permite a los usuarios crear una cuenta de administrador al unirse a una suscripción de pago	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Registro de usuario administrador</b>		
<b>FB1</b>	El usuario ingresa a la aplicación y da click en la opción <b>“Registrarse”</b>	
<b>FB2</b>		El sistema muestra el formulario de registro de usuario con todos los campos necesarios
<b>FB3</b>	El usuario introduce los datos correspondientes, y da click	

	en la opción <b>“Siguiete”</b>	
<b>FB4</b>		El sistema muestra el formulario para el registro del método de pago
<b>FB5</b>	El usuario introduce sus datos de pago en el formulario, y selecciona la opción <b>“Suscribirse”</b>	
<b>FB6</b>		El sistema muestra un mensaje de registro exitoso
<b>FB7</b>	El usuario da click en <b>“Aceptar”</b>	
<b>FB8</b>		El sistema redirecciona al usuario a la página principal
<b>Flujos Alternos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB3 y FB5: El usuario decide cancelar el registro</b>		
<b>FA1.1</b>	El usuario selecciona la opción <b>“Cancelar”</b> .	
<b>FA1.2</b>		El sistema muestra un mensaje para confirmar la cancelación del registro
<b>FA1.3</b>	El usuario administrador da click en <b>“Aceptar”</b>	
<b>FA1.4</b>		El sistema ignora los datos ingresados y redirecciona a la pantalla principal
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB3: Los datos introducidos no son válidos o están incompletos</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de

		validaciones en los campos correspondientes
<b>FE1.2</b>	Se ejecuta el <b>FB3</b>	
<b>FE2 en FB5: El pago no pudo ser procesado</b>		
<b>FE2.1</b>		El sistema muestra un mensaje indicando que el pago no pudo ser procesado
<b>FE2.2</b>	Se ejecuta el <b>FB5</b>	

Tabla 22. Tabla descripción y flujo de caso de uso 01 (Elaboración propia)

#### 4.4.4.2. CUS-2 Gestión de Sucursales

<b>Caso de Uso</b>	Gestión de sucursales	<b>Identificador:</b> CUS-2
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Tener una suscripción activa e ingresar a la aplicación	
<b>Postcondición</b>	Agregar, editar o eliminar alguna sucursal	
<b>Descripción</b>	Permite a los usuarios administradores agregar, editar y eliminar las sucursales de su empresa	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Agregar una sucursal</b>		
<b>FB1</b>	El usuario administrador ingresa a la opción <b>“Sucursales”</b>	
<b>FB2</b>		El sistema redirecciona al usuario a la vista de gestión de sucursales
<b>FB3</b>	El usuario administrador da click en la opción <b>“Agregar”</b> .	
<b>FB4</b>		El sistema muestra el formulario de

		creación con los campos necesarios
<b>FB5</b>	El usuario administrador introduce los datos correspondientes, y da click en la opción <b>“Guardar”</b>	
<b>FB6</b>		El sistema guardar los datos suministrados y muestra un mensaje de registro exitoso
<b>FB7</b>	El usuario administrador da click en <b>“Aceptar”</b>	
<b>FB8</b>		Se ejecuta el <b>FB2</b>
<b>Flujos Alternos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB3: Editar sucursal</b>		
<b>FA1.1</b>	El usuario administrador selecciona una sucursal de la lista y da click en la opción <b>“Editar”</b>	
<b>FA1.2</b>		El sistema muestra el formulario de edición con los datos de la sucursal a editar
<b>FA1.3</b>	Se ejecuta <b>FB5</b>	
<b>FA2 en FB3: Eliminar sucursal</b>		
<b>FA2.1</b>	El usuario administrador selecciona una sucursal de la lista y da click en la opción <b>“Eliminar”</b>	
<b>FA2.2</b>		El sistema muestra un mensaje de confirmación
<b>FA2.3</b>	El usuario administrador da click en <b>“Confirmar”</b>	
<b>FA2.4</b>		El sistema elimina la sucursal seleccionada

Flujos de Error (Excepción)		
Paso	Actor(es)	Sistema
<b>FE1 en FB5: Los datos introducidos no son válidos o están incompletos.</b>		
FE1.1		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes
FE1.2	Se ejecuta el <b>FB5</b>	

Tabla 23. Tabla descripción y flujo de caso de uso 02 (Elaboración propia)

#### 4.4.4.3. CUS-3 Gestión de Proveedores

<b>Caso de Uso</b>	Gestión de proveedores	<b>Identificador:</b> CUS-3
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Tener una suscripción activa e ingresar a la aplicación	
<b>Postcondición</b>	Agregar, editar o eliminar algún proveedor	
<b>Descripción</b>	Permite a los usuarios administradores agregar, editar y eliminar los proveedores de su empresa	
Flujo Básico		
Paso	Actor(es)	Sistema
<b>Agregar un proveedor</b>		
FB1	El usuario administrador ingresa a la opción <b>“Proveedores”</b>	
FB2		El sistema redirecciona al usuario a la vista de gestión de proveedores
FB3	El usuario administrador da click en la opción <b>“Agregar”</b> .	

<b>FB4</b>		El sistema muestra el formulario de creación con los campos necesarios
<b>FB5</b>	El usuario administrador introduce los datos correspondientes, y da click en la opción <b>“Guardar”</b>	
<b>FB6</b>		El sistema guardar los datos suministrados y muestra un mensaje de registro exitoso
<b>FB7</b>	El usuario administrador da click en <b>“Aceptar”</b>	
<b>FB8</b>		Se ejecuta el <b>FB2</b>
<b>Flujos Alternos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB3: Editar proveedor</b>		
<b>FA1.1</b>	El usuario administrador selecciona un proveedor de la lista y da click en la opción <b>“Editar”</b>	
<b>FA1.2</b>		El sistema muestra el formulario de edición con los datos del proveedor a editar
<b>FA1.3</b>	Se ejecuta <b>FB5</b>	
<b>FA2 en FB3: Eliminar proveedor</b>		
<b>FA2.1</b>	El usuario administrador selecciona un proveedor de la lista y da click en la opción <b>“Eliminar”</b>	
<b>FA2.2</b>		El sistema muestra un mensaje de confirmación
<b>FA2.3</b>	El usuario administrador da click en <b>“Confirmar”</b>	

<b>FA2.4</b>		El sistema elimina el proveedor seleccionado
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB5: Los datos introducidos no son válidos o están incompletos.</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes
<b>FE1.2</b>	Se ejecuta el <b>FB5</b>	

Tabla 24. Tabla descripción y flujo de Caso de uso 03 (Elaboración propia)

#### 4.4.4.4. CUS-4 Gestión de catálogo de productos

<b>Caso de Uso</b>	Gestión de catálogo de productos	<b>Identificador: CUS-4</b>
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Tener una suscripción activa e ingresar a la aplicación	
<b>Postcondición</b>	Agregar, editar o eliminar algún producto del catálogo	
<b>Descripción</b>	Permite a los usuarios administradores agregar, editar y eliminar productos de su catálogo	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Agregar un producto al catálogo</b>		
<b>FB1</b>	El usuario administrador ingresa a la opción " <b>Catálogo de productos</b> "	
<b>FB2</b>		El sistema redirecciona al usuario a la vista de gestión del catálogo de productos

<b>FB3</b>	El usuario administrador da click en la opción <b>“Agregar producto”</b> .	
<b>FB4</b>		El sistema muestra el formulario de creación con los campos necesarios
<b>FB5</b>	El usuario administrador introduce los datos correspondientes, y da click en la opción <b>“Guardar”</b>	
<b>FB6</b>		El sistema guardar los datos suministrados y muestra un mensaje de registro exitoso
<b>FB7</b>	El usuario administrador da click en <b>“Aceptar”</b>	
<b>FB8</b>		Se ejecuta el <b>FB2</b>
<b>Flujos Alternos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB3: Editar un producto del catálogo</b>		
<b>FA1.1</b>	El usuario administrador selecciona un producto del catálogo y da click en la opción <b>“Editar”</b>	
<b>FA1.2</b>		El sistema muestra el formulario de edición con los datos del producto a editar
<b>FA1.3</b>	Se ejecuta <b>FB5</b>	
<b>FA2 en FB3: Eliminar producto del catálogo</b>		
<b>FA2.1</b>	El usuario administrador selecciona un producto del catálogo y da click en la opción <b>“Eliminar”</b>	
<b>FA2.2</b>		El sistema muestra un mensaje de confirmación

<b>FA2.3</b>	El usuario administrador da click en “ <b>Confirmar</b> ”	
<b>FA2.4</b>		El sistema elimina el producto seleccionado
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB5: Los datos introducidos no son válidos o están incompletos.</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes
<b>FE1.2</b>	Se ejecuta el <b>FB5</b>	

Tabla 25. Tabla descripción y flujo de caso de uso 04 (Elaboración propia)

#### 4.4.4.5. CUS-5 Gestión de Usuarios Transaccionales

<b>Caso de Uso</b>	Gestión de usuarios transaccionales	<b>Identificador:</b> CUS-5
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Tener una suscripción activa e ingresar a la aplicación	
<b>Postcondición</b>	Agregar, editar o eliminar algún usuario transaccional	
<b>Descripción</b>	Permite a los usuarios administradores otorgar permisos de realizar movimientos de productos a otros usuarios	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Agregar un usuario transaccional</b>		
<b>FB1</b>	El usuario administrador ingresa a la opción “ <b>Usuarios</b> ”	

<b>FB2</b>		El sistema redirecciona al usuario a la vista de gestión de usuarios
<b>FB3</b>	El usuario administrador da click en la opción <b>“Agregar usuario”</b> .	
<b>FB4</b>		El sistema muestra el formulario de creación con los campos necesarios
<b>FB5</b>	El usuario administrador introduce los datos correspondientes, y da click en la opción <b>“Guardar”</b>	
<b>FB6</b>		El sistema guardar los datos suministrados y muestra un mensaje de registro exitoso
<b>FB7</b>	El usuario administrador da click en <b>“Aceptar”</b>	
<b>FB8</b>		Se ejecuta el <b>FB2</b>
<b>Flujos Alternos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB3: Editar un usuario</b>		
<b>FA1.1</b>	El usuario administrador selecciona un usuario de la lista y da click en la opción <b>“Editar”</b>	
<b>FA1.2</b>		El sistema muestra el formulario de edición con los datos del usuario a editar
<b>FA1.3</b>	Se ejecuta <b>FB5</b>	
<b>FA2 en FB3: Eliminar producto del catálogo</b>		
<b>FA2.1</b>	El usuario administrador selecciona un usuario de la lista y da click en la opción <b>“Eliminar”</b>	
<b>FA2.2</b>		El sistema muestra un mensaje

		de confirmación
<b>FA2.3</b>	El usuario administrador da click en <b>“Confirmar”</b>	
<b>FA2.4</b>		El sistema elimina el usuario seleccionado
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB5: Los datos introducidos no son válidos o están incompletos.</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes
<b>FE1.2</b>	Se ejecuta el <b>FB5</b>	

Tabla 26. Tabla descripción y flujo de caso de uso 05 (Elaboración propia)

#### 4.4.4.6. CUS-6 Movimiento de Productos

<b>Caso de Uso</b>	Movimiento de productos	<b>Identificador:</b> CUS-6
<b>Actores</b>	Usuario transaccional	
<b>Precondición</b>	Tener e ingresar a la aplicación con un usuario transaccional activo	
<b>Postcondición</b>	Mover algún producto de alguna sucursal a otra	
<b>Descripción</b>	Permite a los usuarios transaccionales mover productos entre sucursales	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Mover un producto</b>		
<b>FB1</b>	El usuario transaccional ingresa a la opción <b>“Mover producto”</b>	

<b>FB2</b>		El sistema muestra el formulario de movimiento de productos con los campos necesarios
<b>FB3</b>	El usuario transaccional introduce los datos correspondientes, y da click en la opción “ <b>Guardar</b> ”	
<b>FB4</b>		El sistema utiliza los datos suministrados para realizar el movimiento y muestra un mensaje de éxito
<b>FB5</b>	El usuario transaccional da click en “ <b>Aceptar</b> ”	
<b>FB6</b>		El sistema redirecciona a la pantalla principal
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB3: Los datos introducidos no son válidos o están incompletos.</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes
<b>FE1.2</b>	Se ejecuta el <b>FB3</b>	

Tabla 27. Tabla descripción y flujo de caso de uso 06 (Elaboración propia)

#### 4.4.4.7. CUS-7 Entrada de Productos

<b>Caso de Uso</b>	Entrada de productos	<b>Identificador:</b> CUS-7
<b>Actores</b>	Usuario transaccional	
<b>Precondición</b>	Tener e ingresar a la aplicación con un usuario transaccional activo	

<b>Postcondición</b>	Registrar la entrada de algún producto a alguna sucursal	
<b>Descripción</b>	Permite a los usuarios transaccionales registrar la entrada de productos	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Registrar entrada de producto</b>		
<b>FB1</b>	El usuario transaccional ingresa a la opción <b>“Entrada de producto”</b>	
<b>FB2</b>		El sistema muestra el formulario de registro de entrada de productos con los campos necesarios
<b>FB3</b>	El usuario transaccional introduce los datos correspondientes, y da click en la opción <b>“Guardar”</b>	
<b>FB4</b>		El sistema utiliza los datos suministrados para registrar la entrada y muestra un mensaje de éxito
<b>FB5</b>	El usuario transaccional da click en <b>“Aceptar”</b>	
<b>FB6</b>		El sistema redirecciona a la pantalla principal
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB3: Los datos introducidos no son válidos o están incompletos.</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes

<b>FE1.2</b>	Se ejecuta el <b>FB3</b>	
--------------	--------------------------	--

Tabla 28. Tabla descripción y flujo de caso de uso 07 (Elaboración propia)

#### 4.4.4.8. CUS-8 Salida de Productos

<b>Caso de Uso</b>	Salida de productos	<b>Identificador:</b> CUS-8
<b>Actores</b>	Usuario transaccional	
<b>Precondición</b>	Tener e ingresar a la aplicación con un usuario transaccional activo	
<b>Postcondición</b>	Registrar la salida de algún producto a alguna sucursal	
<b>Descripción</b>	Permite a los usuarios transaccionales registrar la salida de productos	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Registrar salida de producto</b>		
<b>FB1</b>	El usuario transaccional ingresa a la opción " <b>Salida de producto</b> "	
<b>FB2</b>		El sistema muestra el formulario de registro de salida de productos con los campos necesarios
<b>FB3</b>	El usuario transaccional introduce los datos correspondientes, y da click en la opción " <b>Guardar</b> "	
<b>FB4</b>		El sistema utiliza los datos suministrados para registrar la salida y muestra un mensaje de éxito

<b>FB5</b>	El usuario transaccional da click en “ <b>Aceptar</b> ”	
<b>FB6</b>		El sistema redirecciona a la pantalla principal
<b>Flujos de Error (Excepción)</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FE1 en FB3: Los datos introducidos no son válidos o están incompletos.</b>		
<b>FE1.1</b>		El sistema muestra mensajes indicando los errores de validaciones en los campos correspondientes
<b>FE1.2</b>	Se ejecuta el <b>FB3</b>	

Tabla 29. Tabla descripción y flujo de caso de uso 08 (Elaboración propia)

#### 4.4.4.9. CUS-9 Visualización de Inventario

<b>Caso de Uso</b>	Visualización de inventario	<b>Identificador:</b> CUS-9
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Tener e ingresar a la aplicación con un usuario administrador activo	
<b>Postcondición</b>	Visualizar el estado actual del inventario y consultar las existencias de los productos	
<b>Descripción</b>	Permite a los usuarios administradores visualizar el estado actual del inventario y consultar las existencias de productos a través de una serie de filtros	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Visualizar el estado del inventario</b>		

<b>FB1</b>	El usuario administrador ingresa a la opción <b>“Inventario”</b>	
<b>FB2</b>		El sistema muestra la vista de consulta del inventario con los filtros correspondientes
<b>FB3</b>	El usuario administrador filtra los datos que desea visualizar y da click en <b>“Consultar”</b>	
<b>FB4</b>		El sistema muestra una tabla con la información necesaria y filtrada de inventario y existencias
<b>Flujos alternativos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB4: Exportar datos</b>		
<b>FA1.1</b>	El usuario administrador da click en la opción <b>“Exportar”</b>	
<b>FA1.2</b>		El sistema descarga un archivo de excel con la información de inventario

Tabla 30. Tabla descripción y flujo de caso de uso 09 (Elaboración propia)

#### 4.4.4.10. CUS-10 Visualización de Transacciones de Productos

<b>Caso de Uso</b>	Visualización de transacciones de productos	<b>Identificador:</b> CUS-10
<b>Actores</b>	Usuario administrador	
<b>Precondición</b>	Tener e ingresar a la aplicación con un usuario administrador activo	
<b>Postcondición</b>	Visualizar las transacciones de productos, conocer los costos, puntos de reabastecimiento, vencimientos y	

	predicción de demanda de los productos	
<b>Descripción</b>	Permite a los usuarios administradores visualizar las transacciones de productos a través del tiempo para conocer los costos, puntos de reabastecimiento, vencimientos y predicción de demanda de cada producto utilizando una serie de filtros	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>Consultar transacciones de productos</b>		
<b>FB1</b>	El usuario administrador ingresa a la opción <b>“Visualización avanzada”</b>	
<b>FB2</b>		El sistema muestra la vista de transacciones de productos con los filtros necesarios
<b>FB3</b>	El usuario administrador filtra la información que desea consultar y da click en <b>“Consultar”</b>	
<b>FB4</b>		El sistema muestra una tabla y una gráfica con por cada producto dentro de los filtros suministrados, mostrando los costos, los puntos de reabastecimiento recomendados, una predicción de la demanda futura y los productos próximos a vencer para poder ser devueltos al proveedor o ponerlos en oferta
<b>Flujos Alternos</b>		
<b>Paso</b>	<b>Actor(es)</b>	<b>Sistema</b>
<b>FA1 en FB3: Exportar datos</b>		

<b>FA1.1</b>	El usuario administrador da click en la opción " <b>Exportar</b> "	
<b>FA1.2</b>		El sistema descarga un archivo de excel con la información de las transacciones

*Tabla 31. Tabla descripción y flujo de caso de uso 10 (Elaboración propia)*

## 4.5. Diseño de Alto Nivel

### 4.5.1. Diagrama de Arquitectura

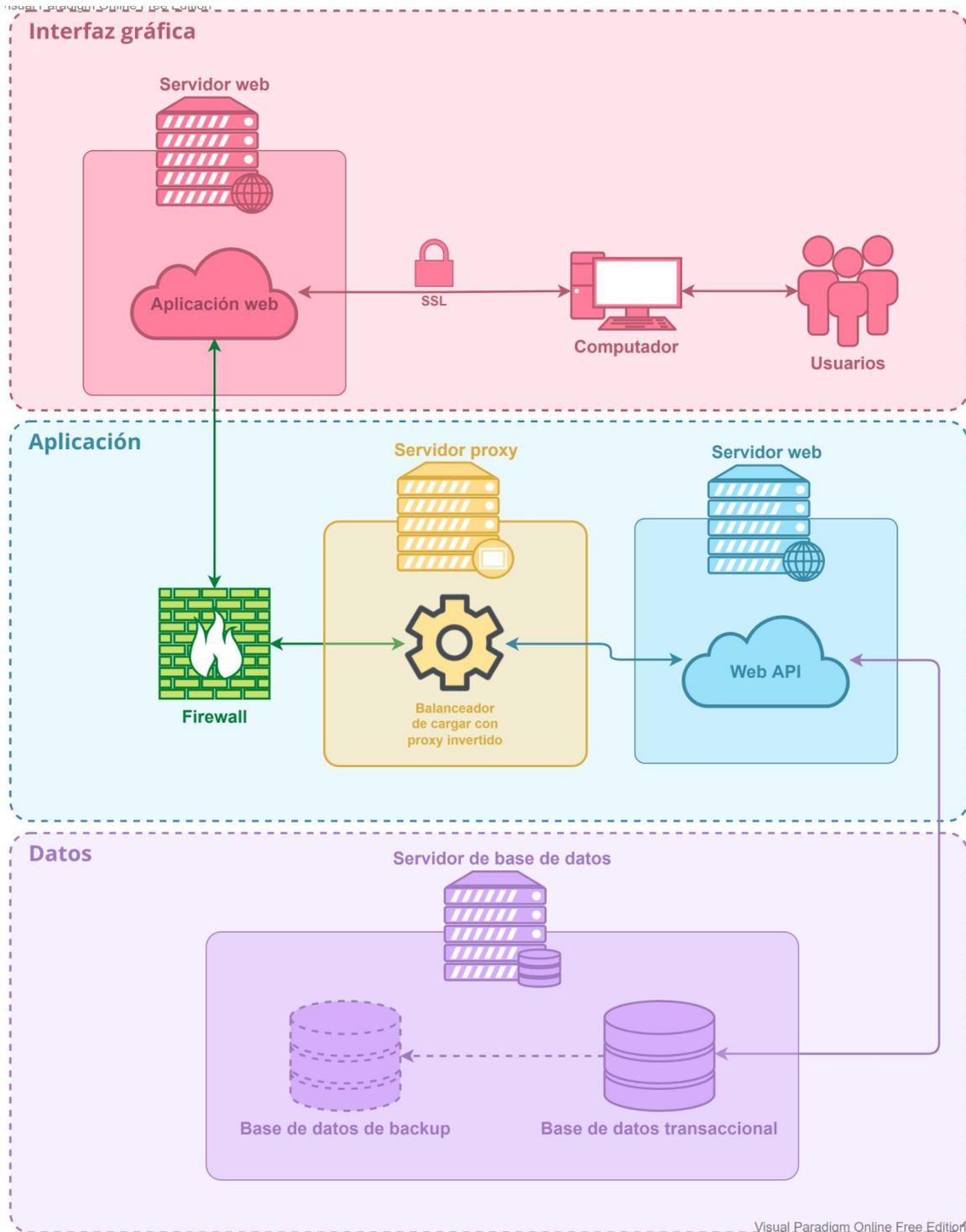


Figura 2. Diagrama de arquitectura (Elaboración propia)

## 4.5.1. Diagrama General de Clases

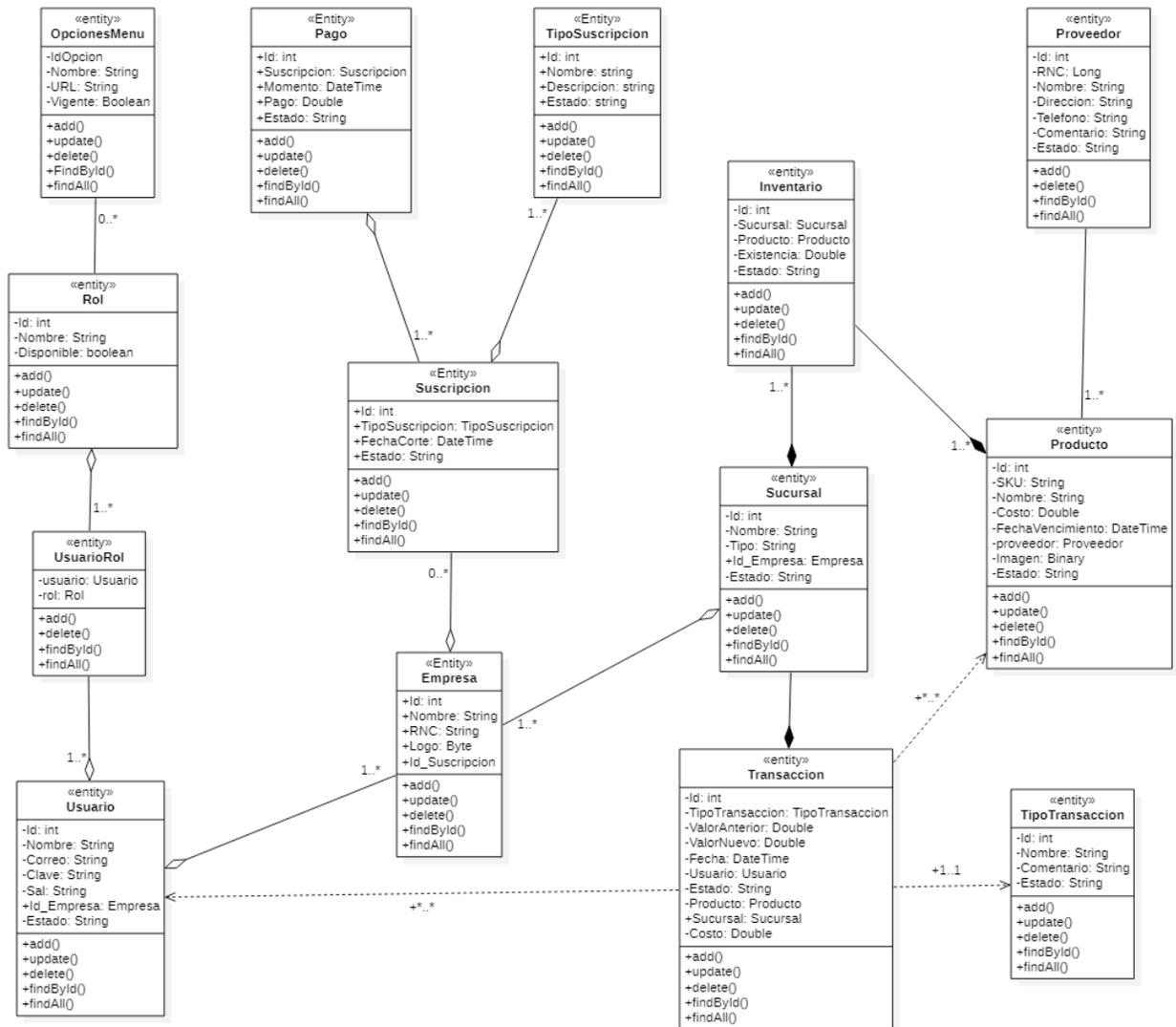


Figura 3. Diagrama general de clases (Elaboración propia)

## 4.5.2. Diagrama de la Base de Datos

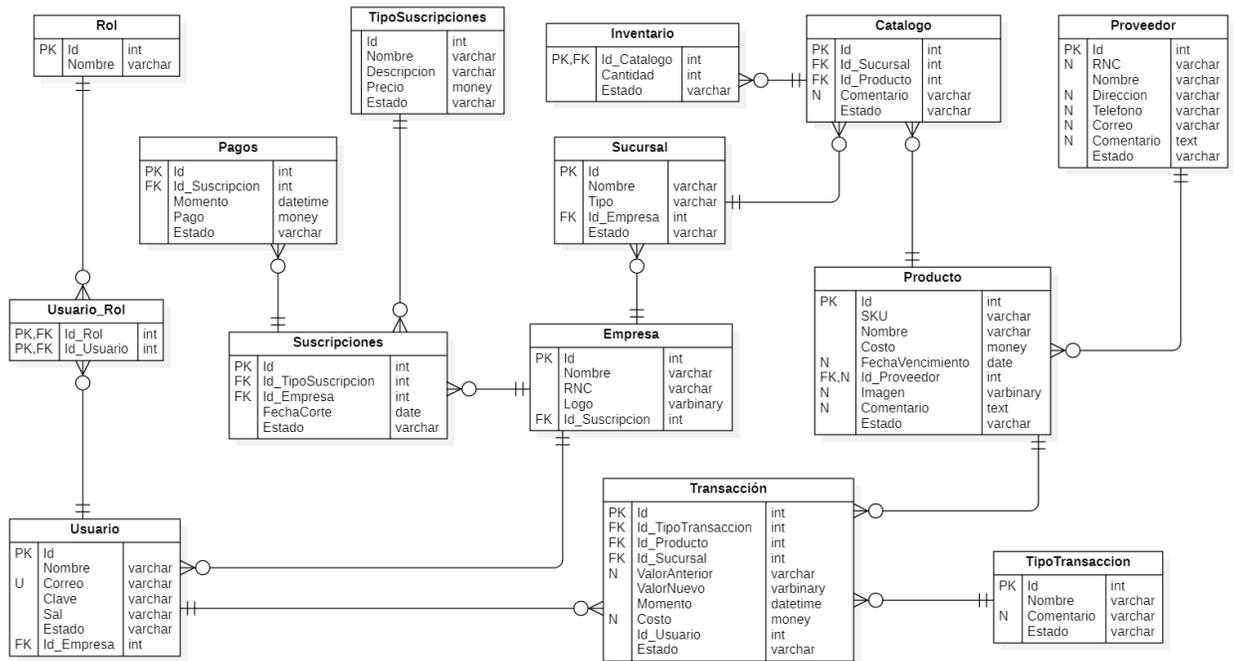


Figura 4. Diagrama de la base de datos (Elaboración propia)

## 4.6. Diseño de Bajo Nivel

### 4.6.1. Diagramas de secuencia

#### 4.6.1.1. Registro de Usuario Administrador

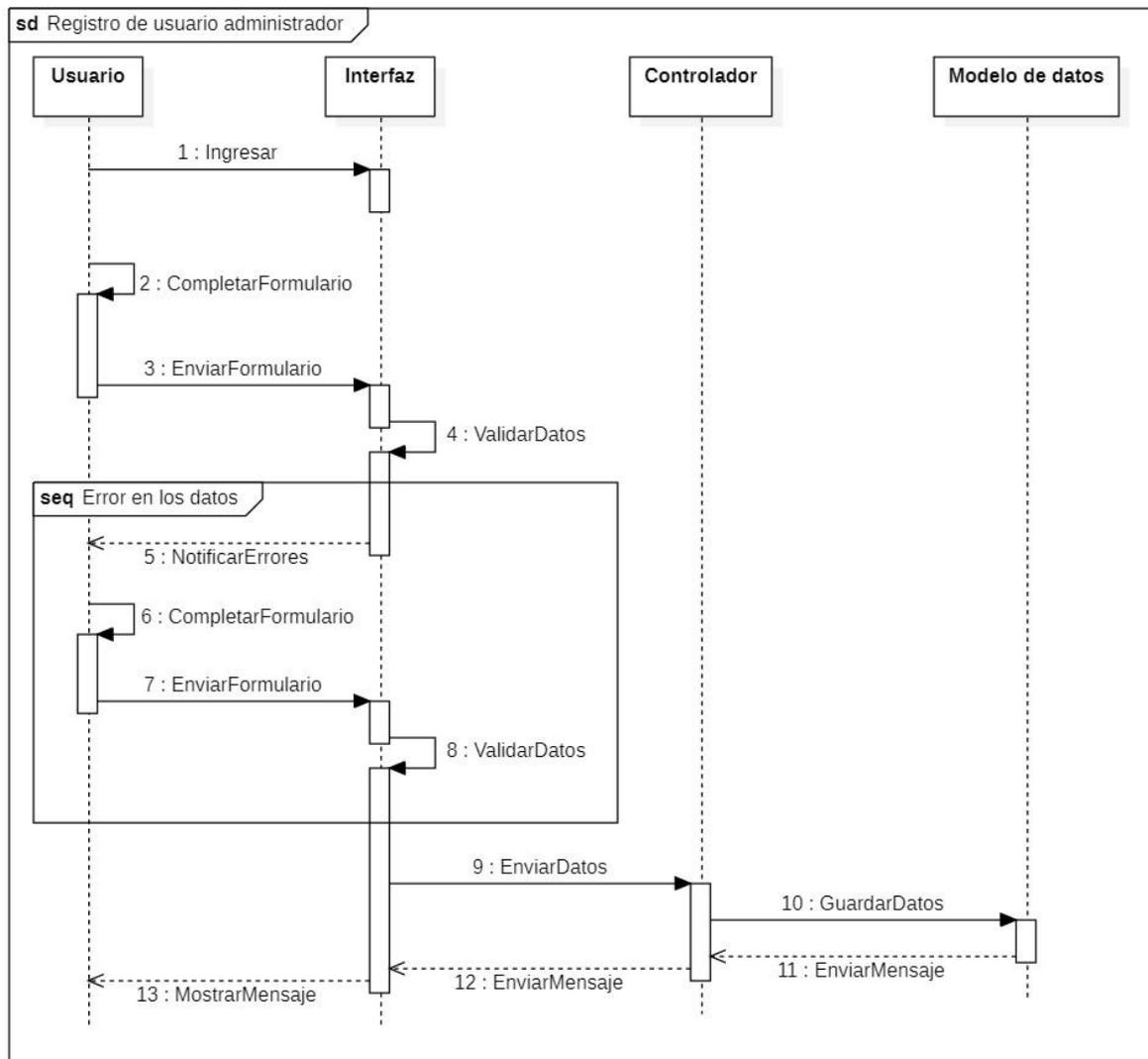


Figura 5. Diagrama de Secuencia 01 (Elaboración propia)

### 4.6.1.2. Gestión de Sucursales

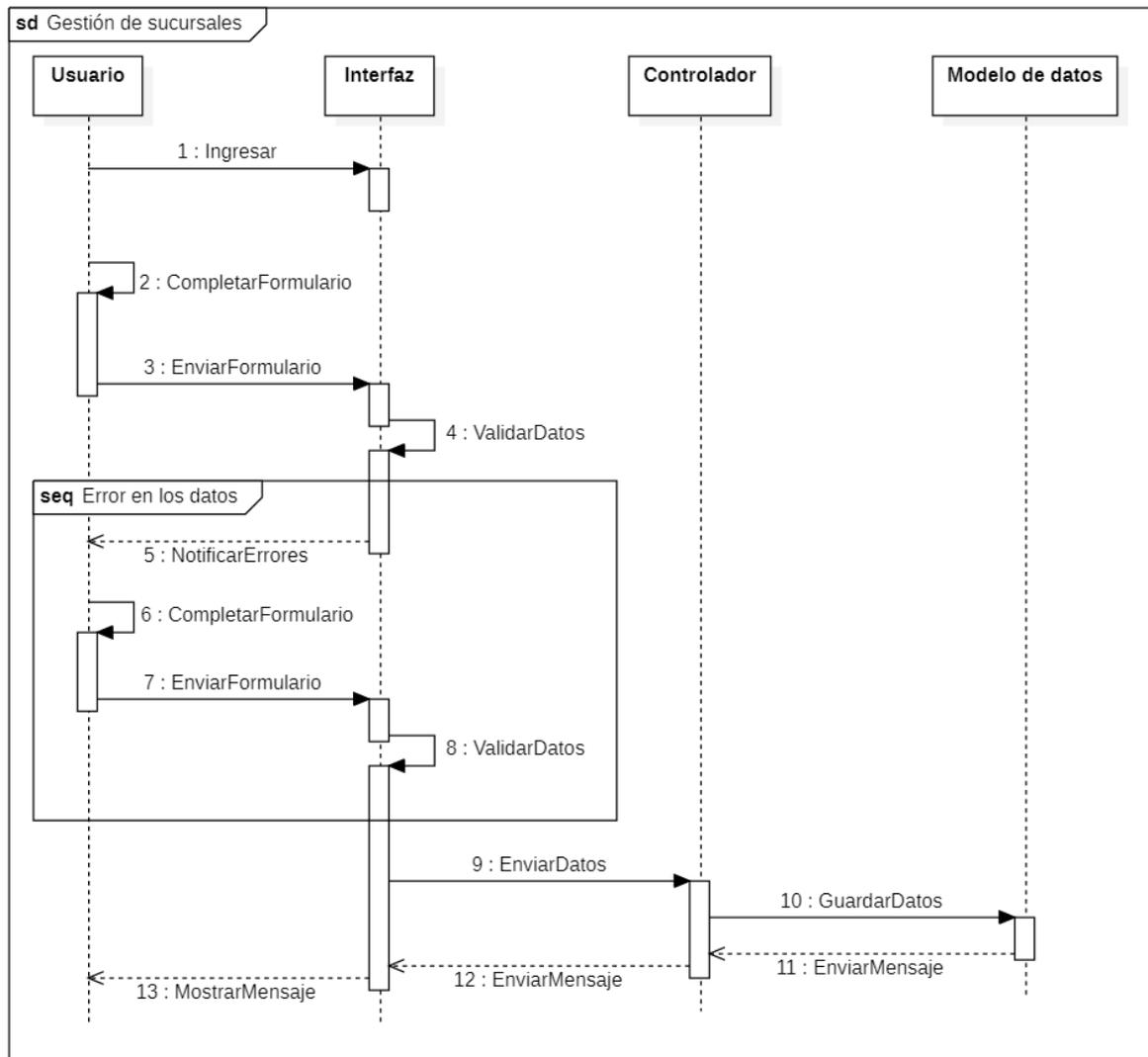


Figura 6. Diagrama de Secuencia 02 (Elaboración propia)

### 4.6.1.3. Gestión de Catálogo de Productos

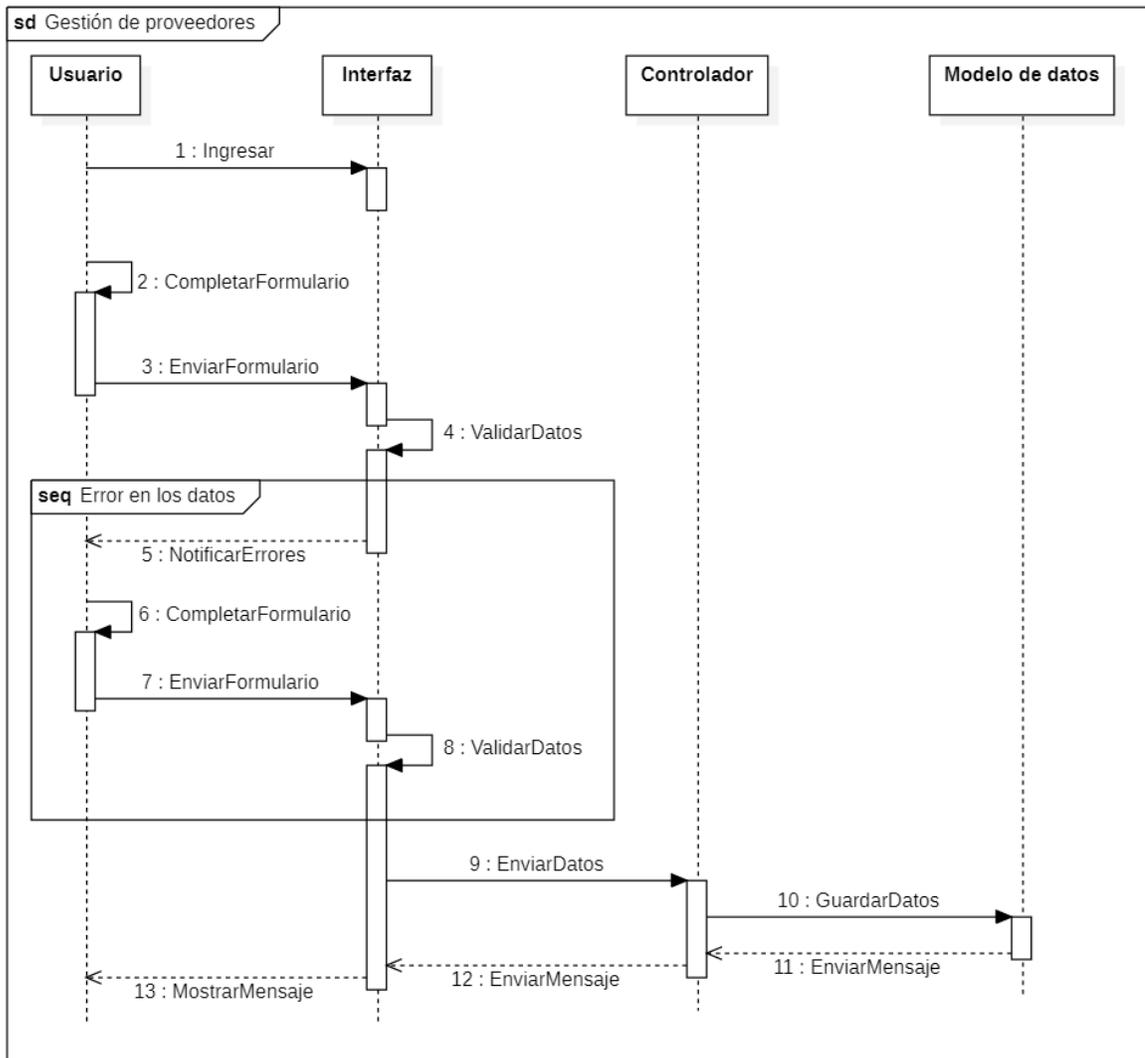


Figura 7. Diagrama de Secuencia 03 (Elaboración propia)

#### 4.6.1.4. Gestión de Catálogo de Productos

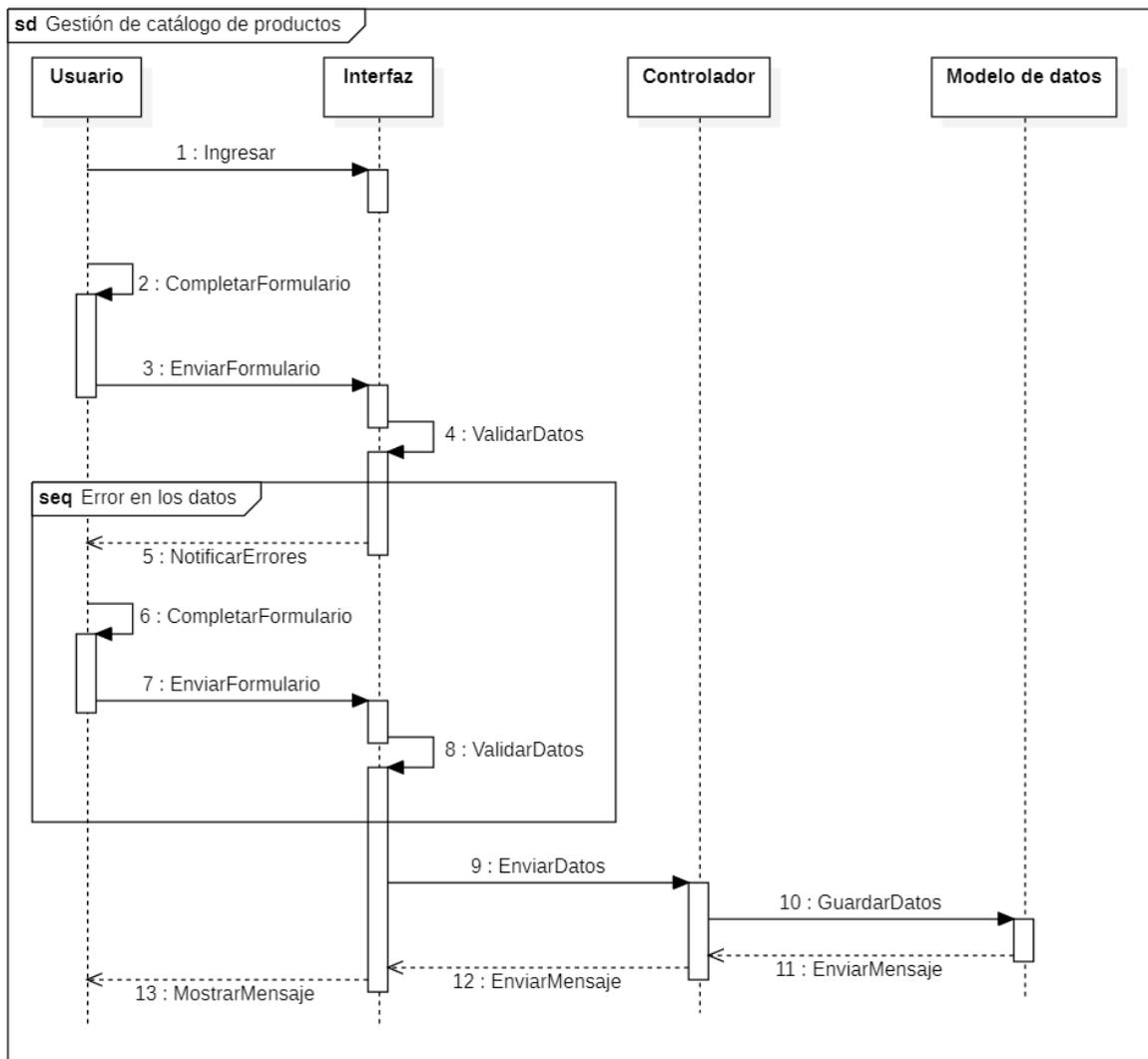


Figura 8. Diagrama de Secuencia 04 (Elaboración propia)

#### 4.6.1.5. Gestión de usuarios transaccionales

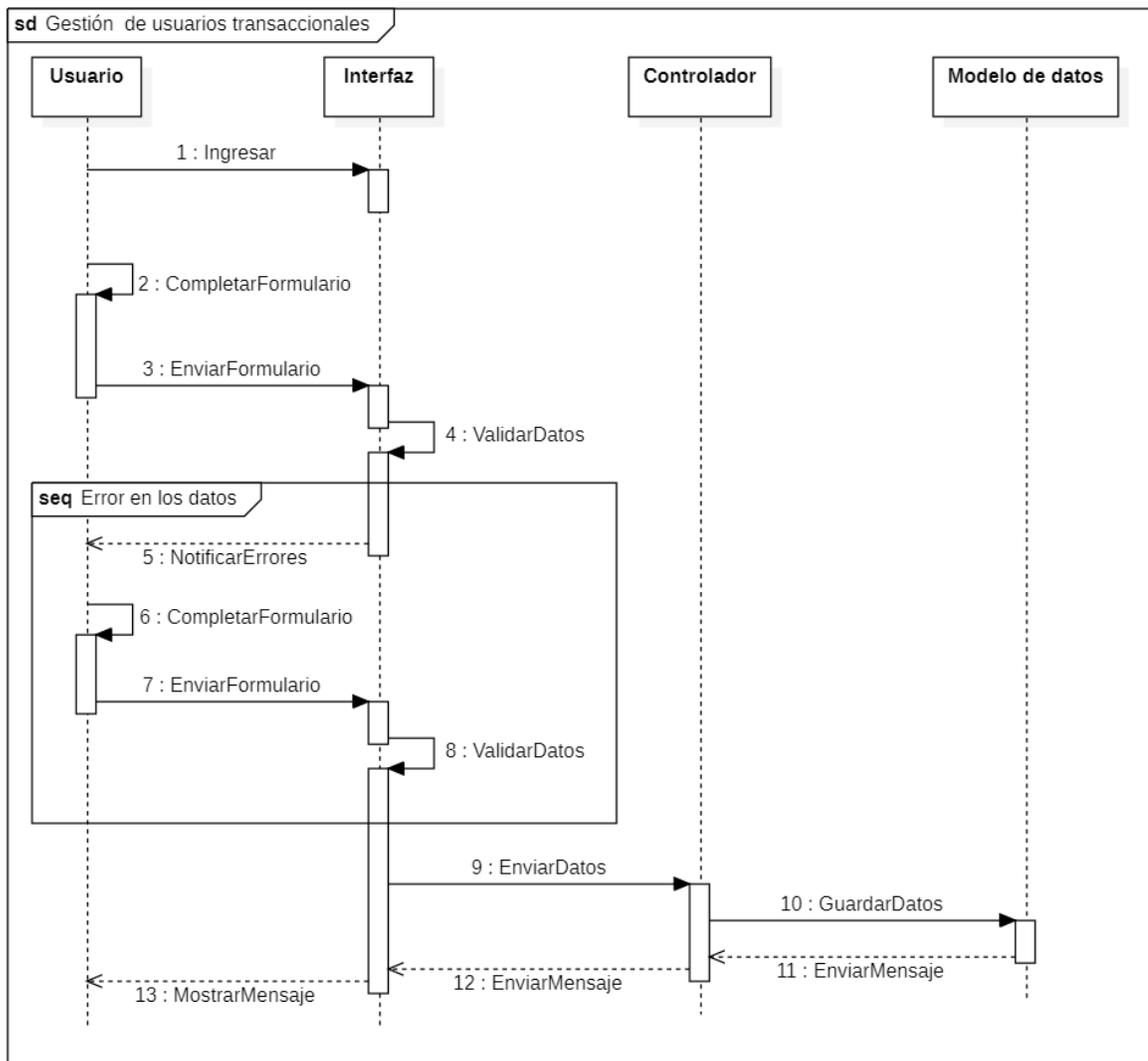


Figura 9. Diagrama de Secuencia 05 (Elaboración propia)

#### 4.6.1.6. Movimiento de Productos

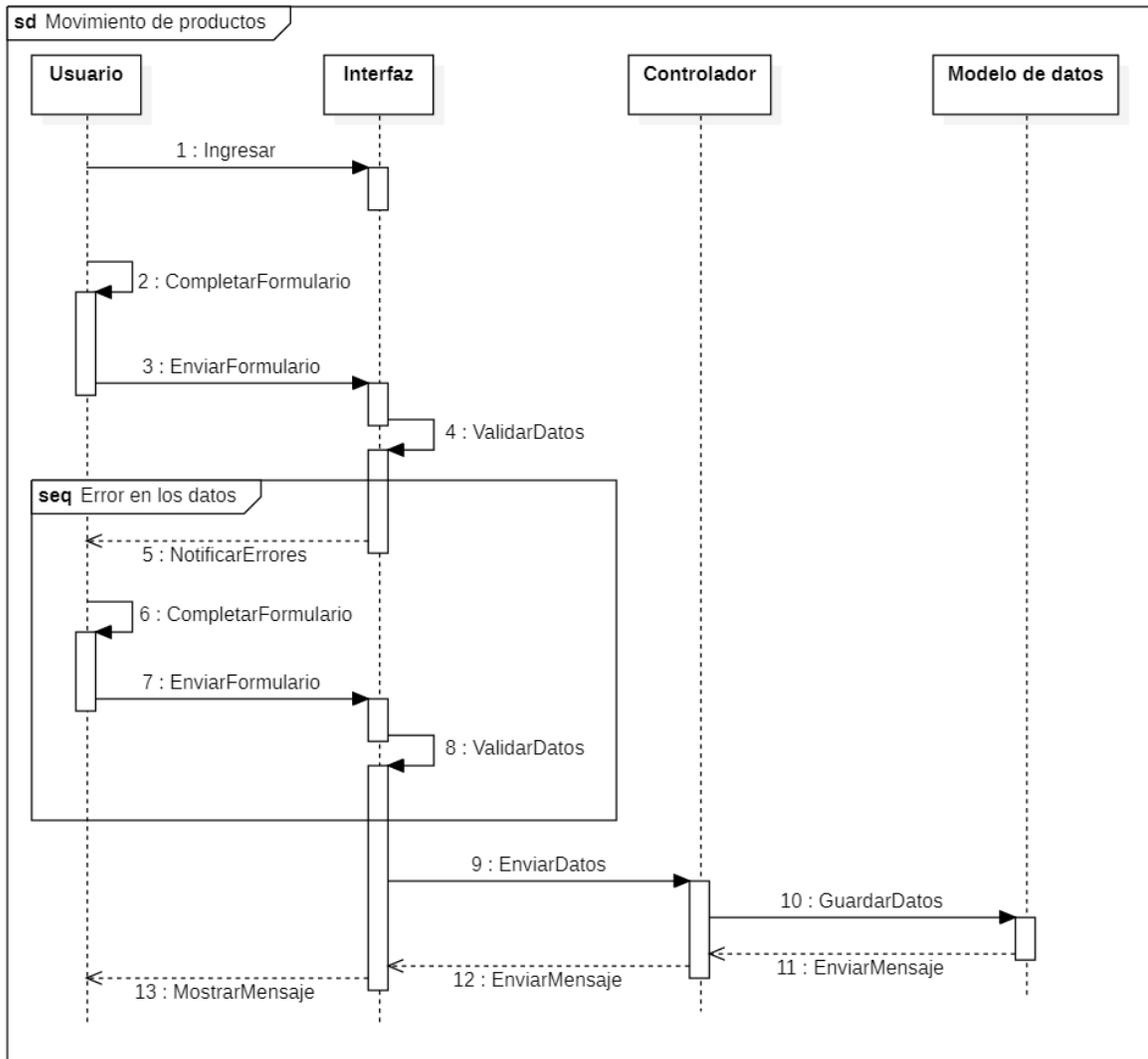


Figura 10. Diagrama de Secuencia 06 (Elaboración propia)

#### 4.6.1.7. Entrada de Productos

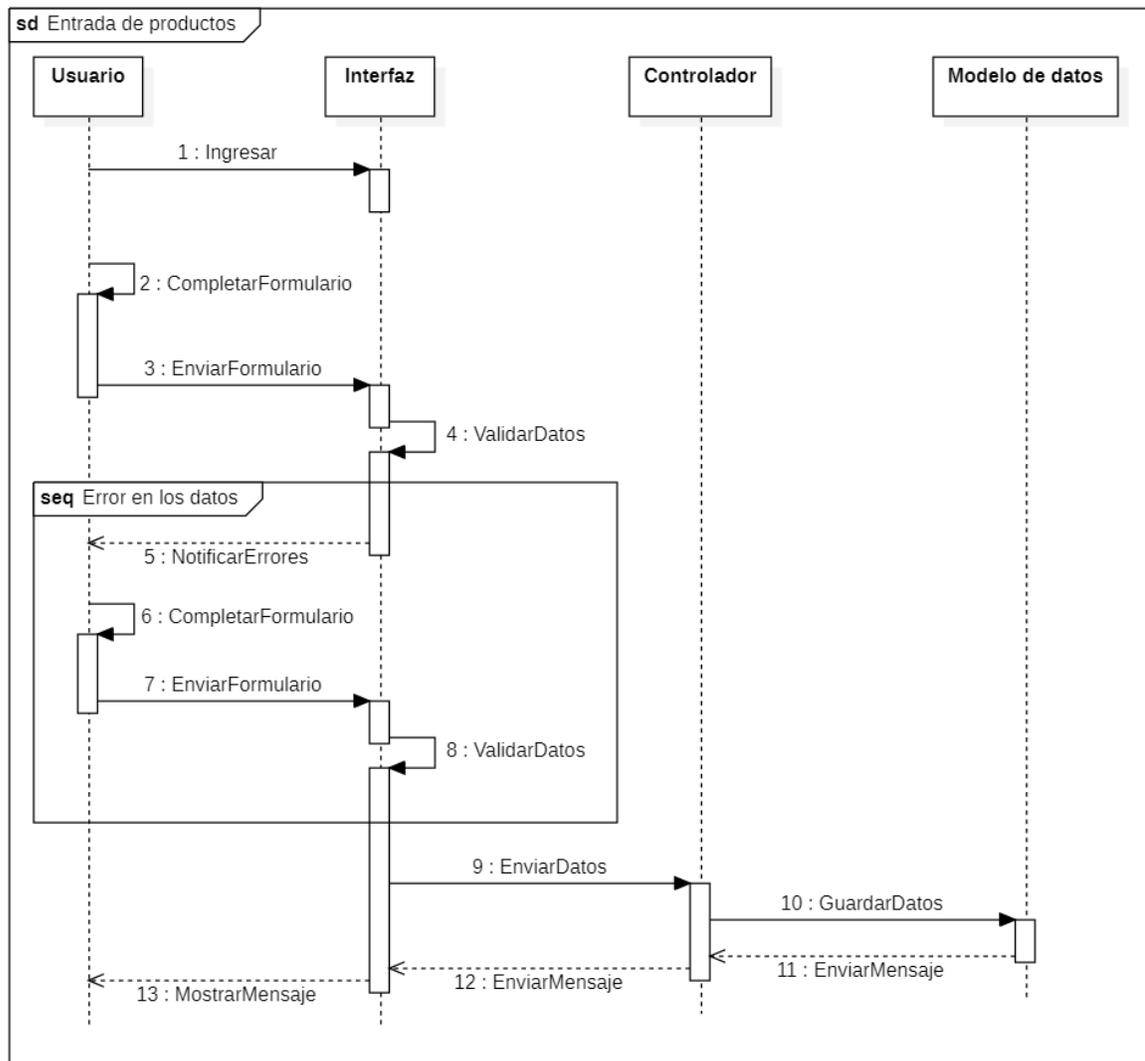


Figura 11. Diagrama de Secuencia 07 (Elaboración propia)

#### 4.6.1.8. Salida de Productos

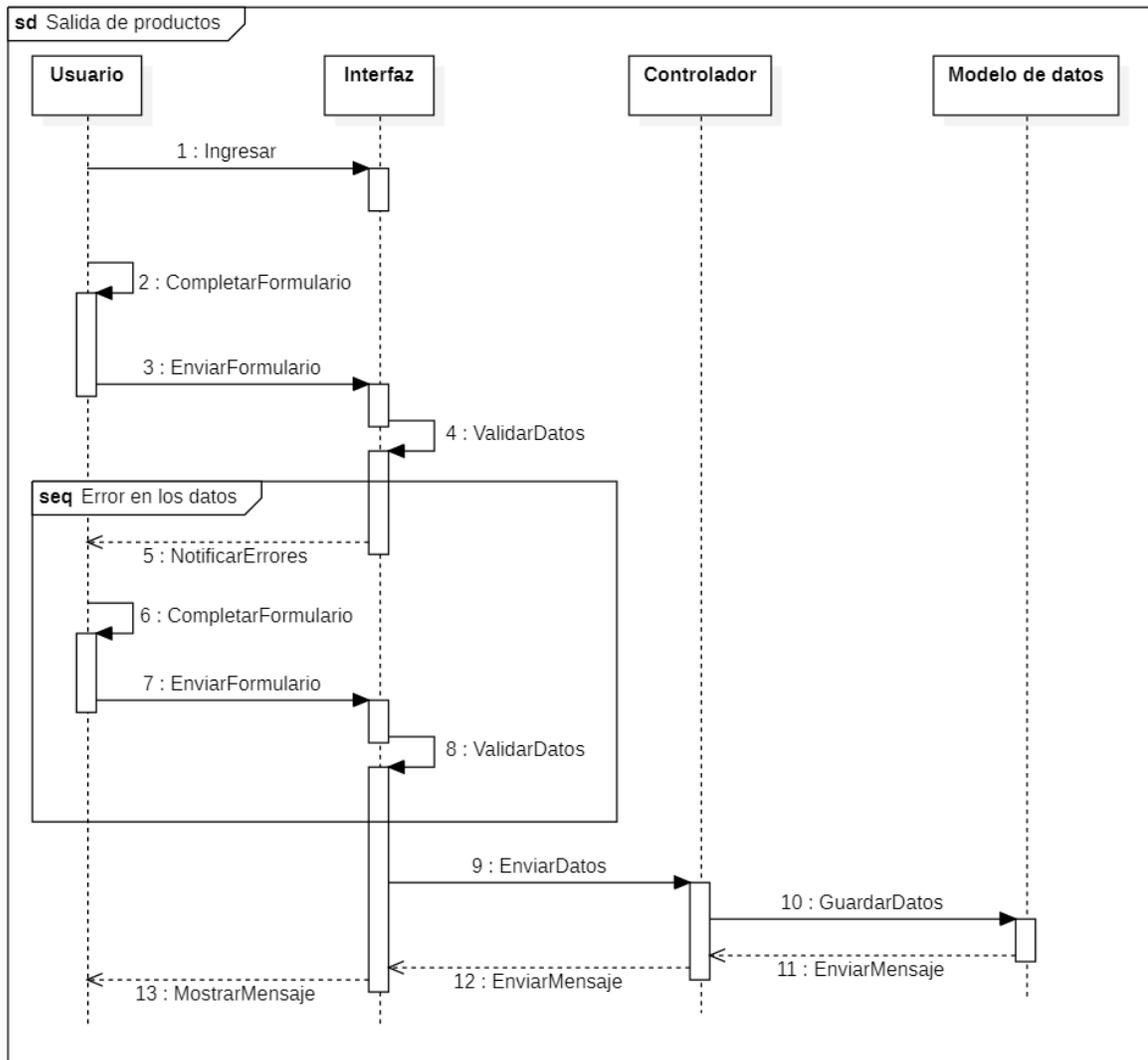


Figura 12. Diagrama de Secuencia 08 (Elaboración propia)

#### 4.6.1.9. Visualización de Transacciones de Productos

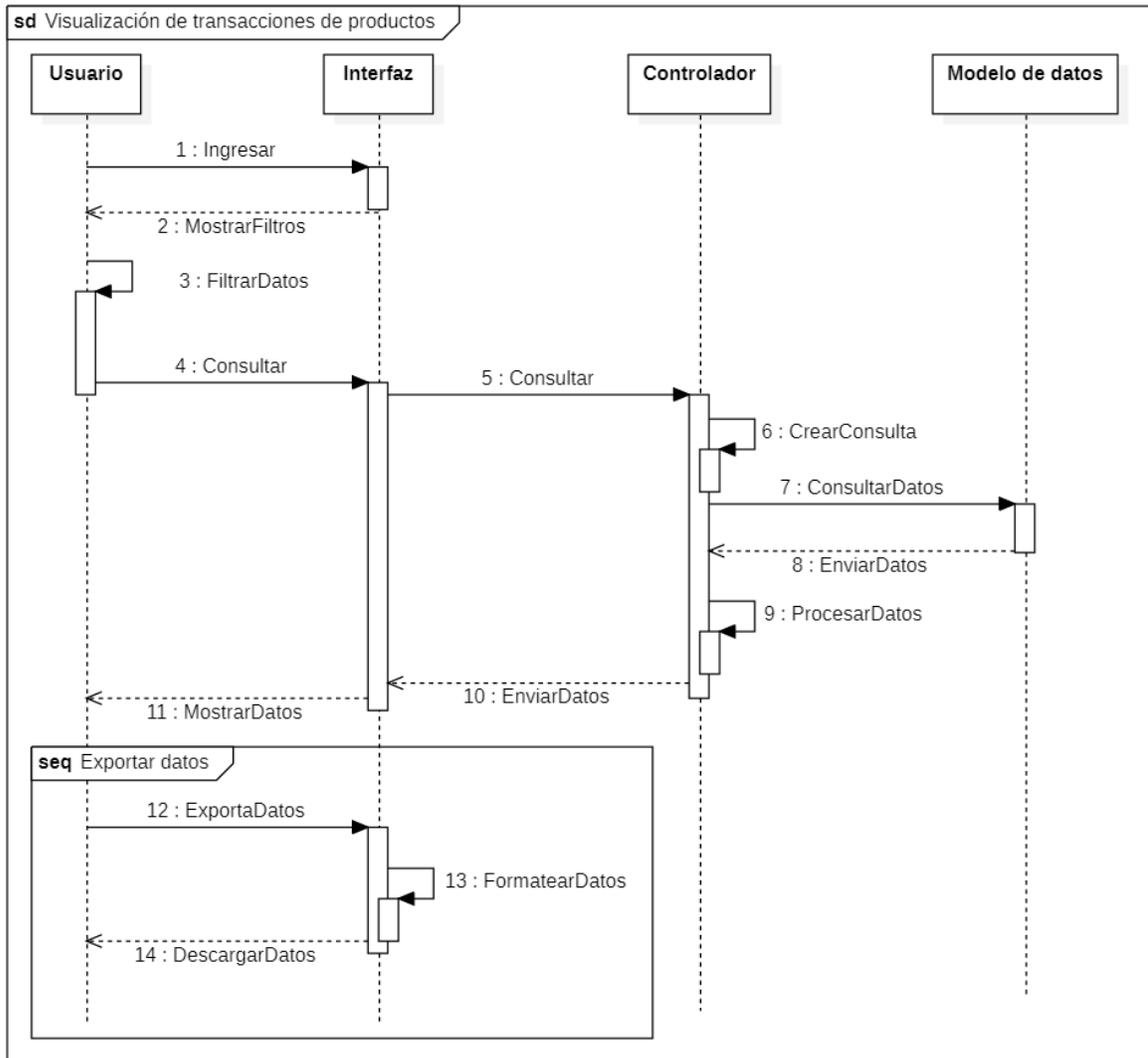


Figura 13. Diagrama de Secuencia 09 (Elaboración propia)

## 4.7. Diseño de Vistas

### 4.7.1. Inicio de Sesión



Figura 14. Vista de inicio de sesión (Elaboración propia)

## 4.7.2. Registro de Usuario Administrador

**Tarium**  
Sistema de inventario

**Información de usuario**

- \* Nombre
- \* Correo
- \* Contraseña
- \* Repetir contraseña

**Método de pago**

- \* Plan de suscripción  Básico  Avanzado  Premium
- \* Número de tarjeta
- Fecha
- \* CVV

Total (RD\$)  
RD\$1,500.00

Figura 15. Vista de registro de usuario administrador (Elaboración propia)

## 4.7.3. Gestión de Sucursales

### 4.7.3.1. Lista

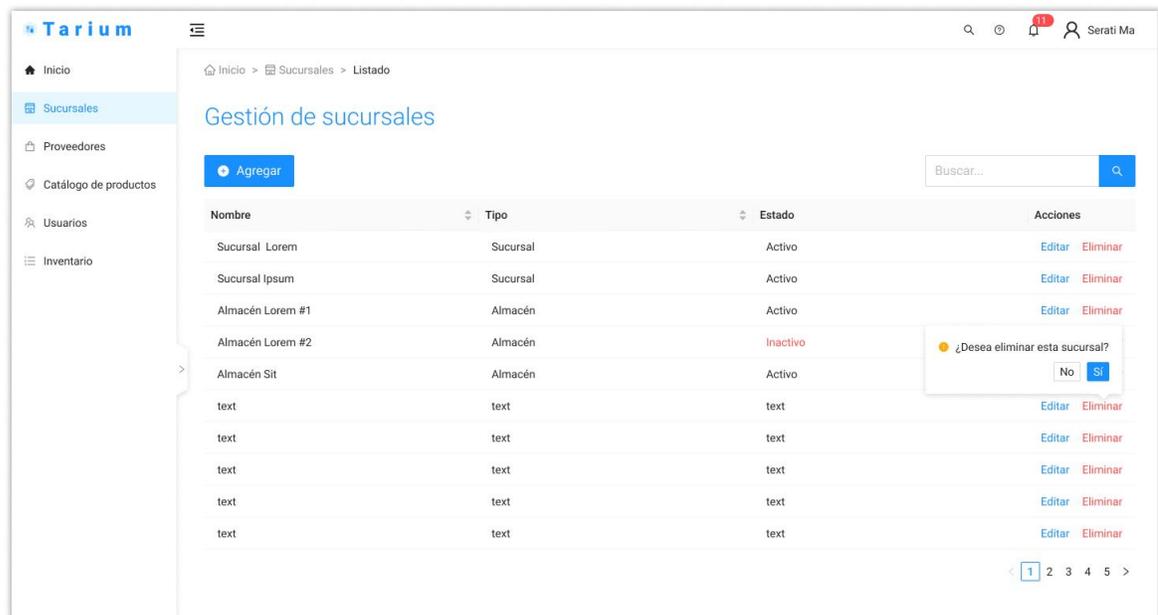


Figura 16. Vista del listado de sucursales (Elaboración propia)

### 4.7.3.2. Formulario

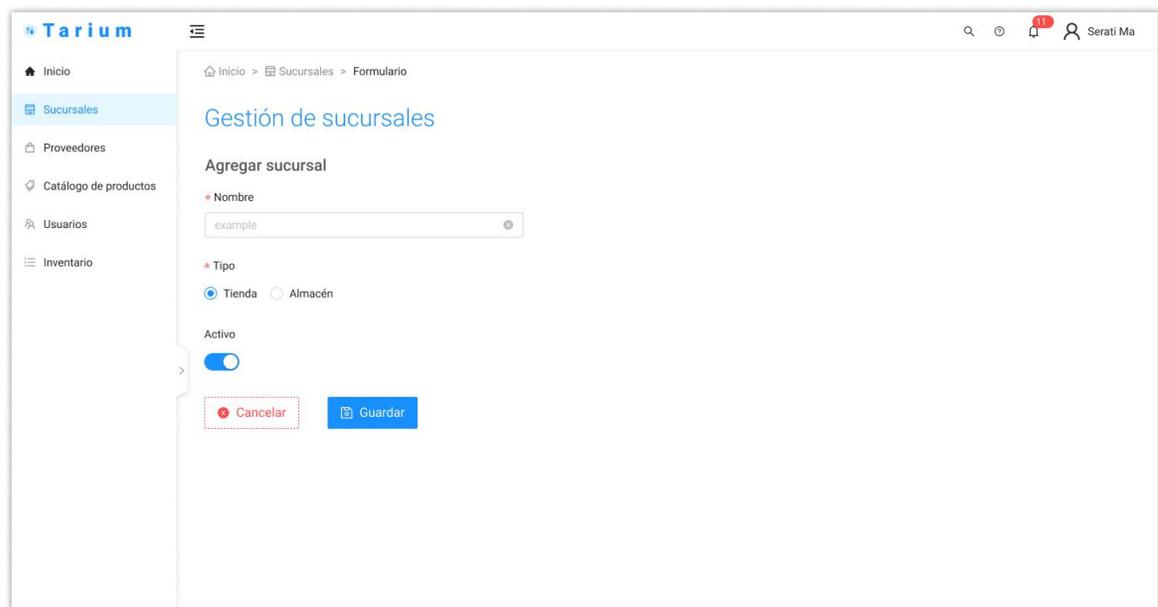


Figura 17. Vista del formulario para la gestión de sucursales (Elaboración propia)

## 4.7.4. Gestión de Proveedores

### 4.7.4.1. Lista

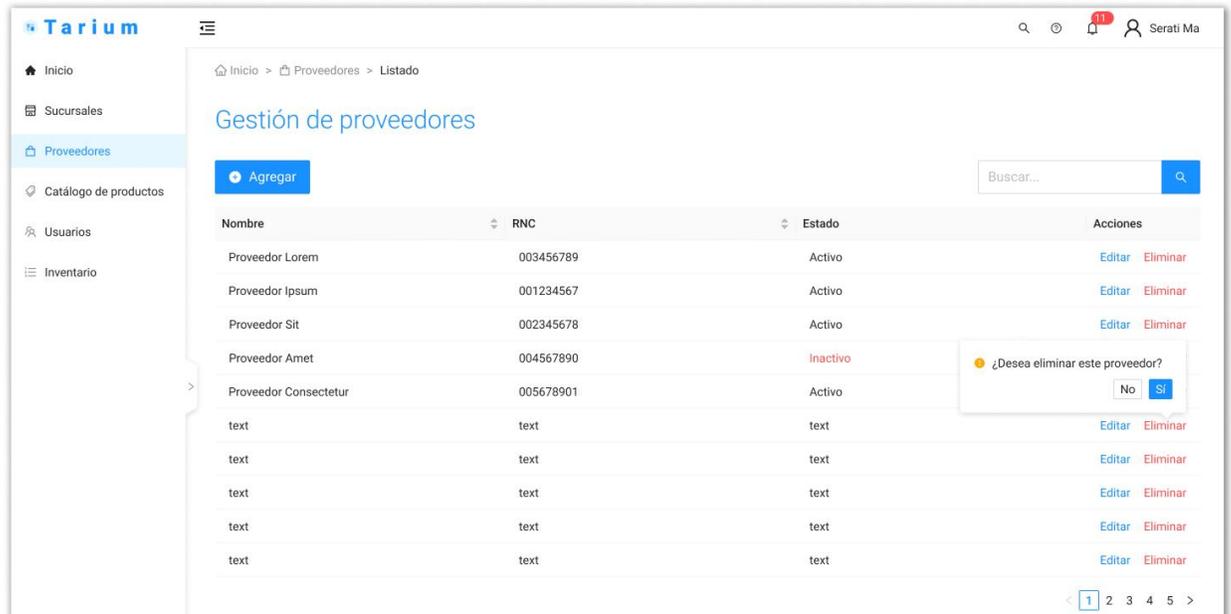


Figura 18. Vista del listado de proveedores (Elaboración propia)

### 4.7.4.2. Formulario

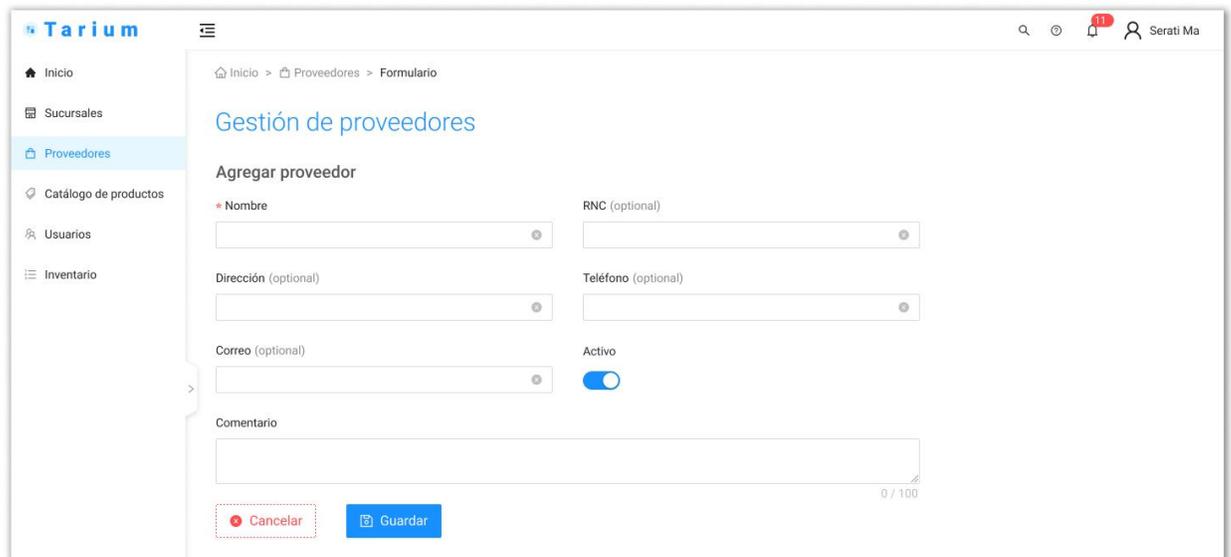


Figura 19. Vista del formulario de gestión de proveedores (Elaboración propia)

## 4.7.5. Gestión de Catálogo de Productos

### 4.7.5.1. Lista

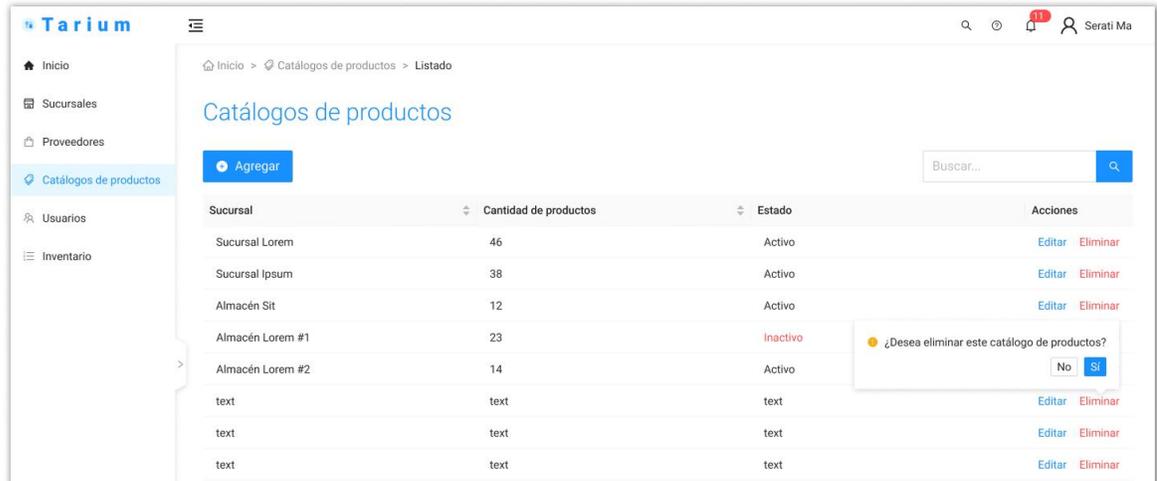


Figura 20. Vista del listado de catálogos de productos (Elaboración propia)

### 4.7.5.2. Formulario

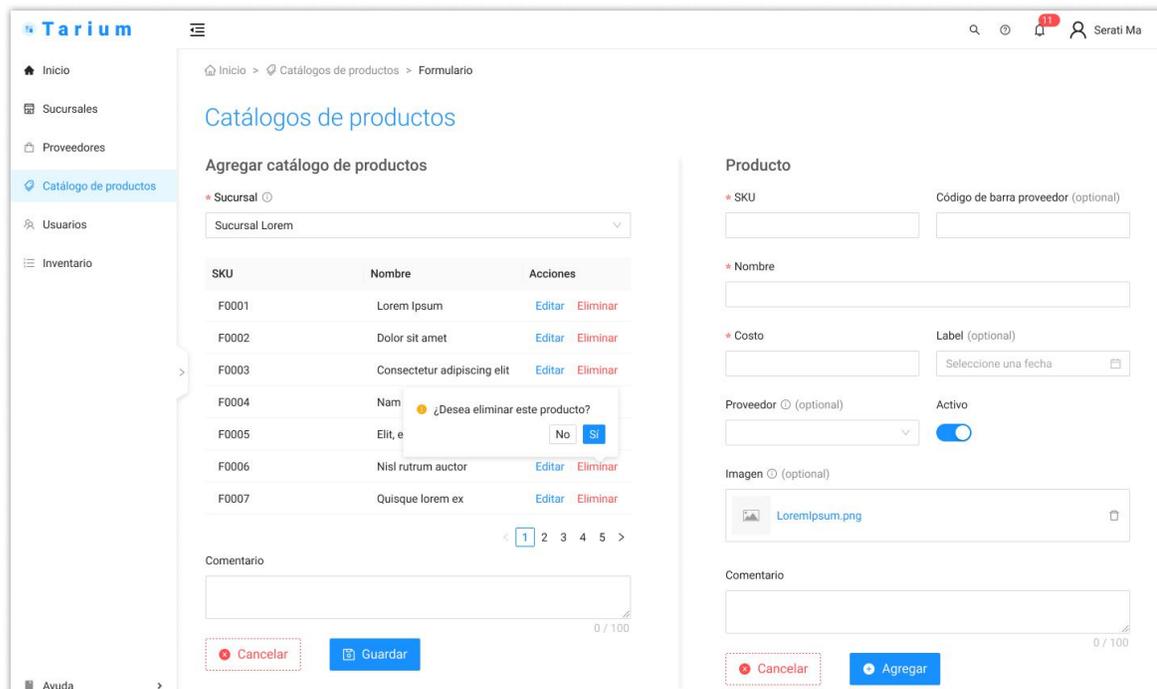


Figura 21. Vista del formulario de gestión de catálogos de productos (Elaboración propia)

## 4.7.6. Gestión de Usuarios Transaccionales

### 4.7.6.1. Lista

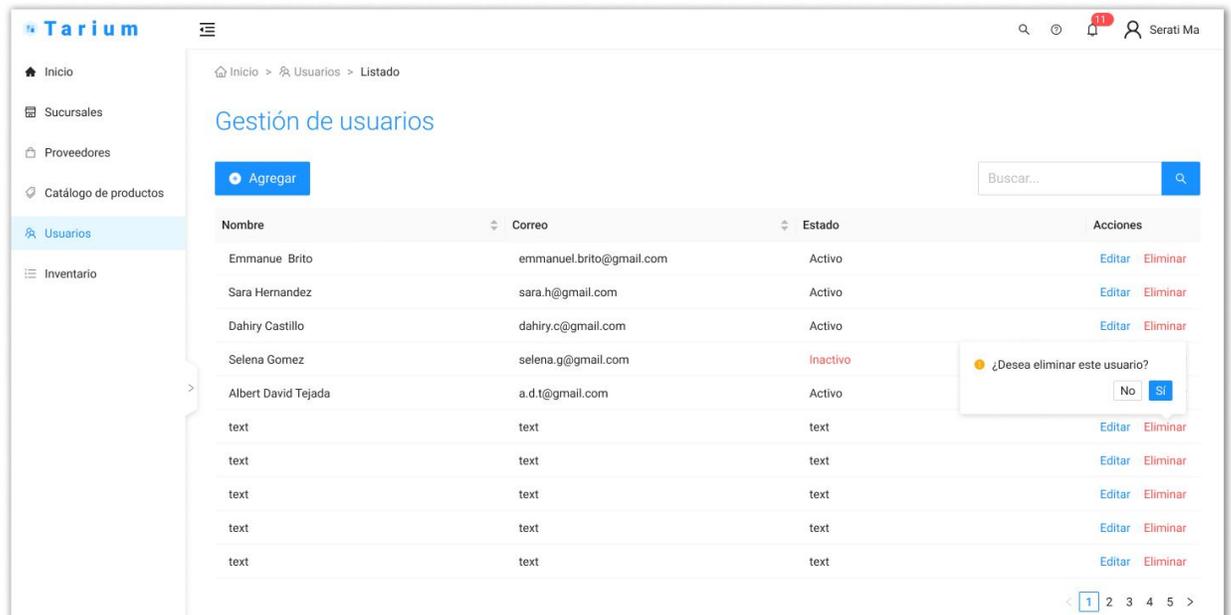


Figura 22. Vista del listado de usuarios transaccionales (Elaboración propia)

### 4.7.6.2. Formulario

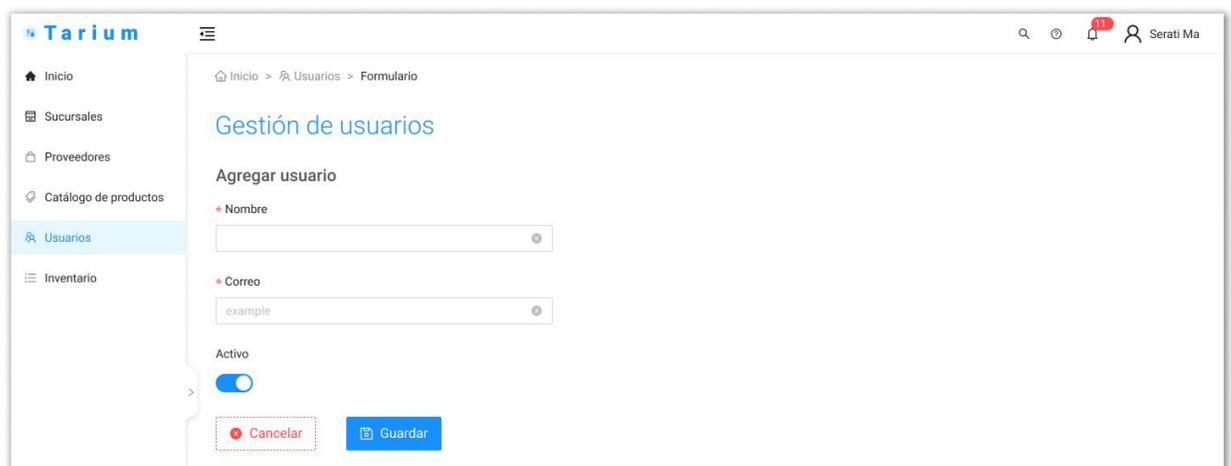


Figura 23. Vista del formulario de gestión de usuarios transaccionales (Elaboración propia)

## 4.7.7. Movimiento de Productos

### 4.7.7.1. Lista

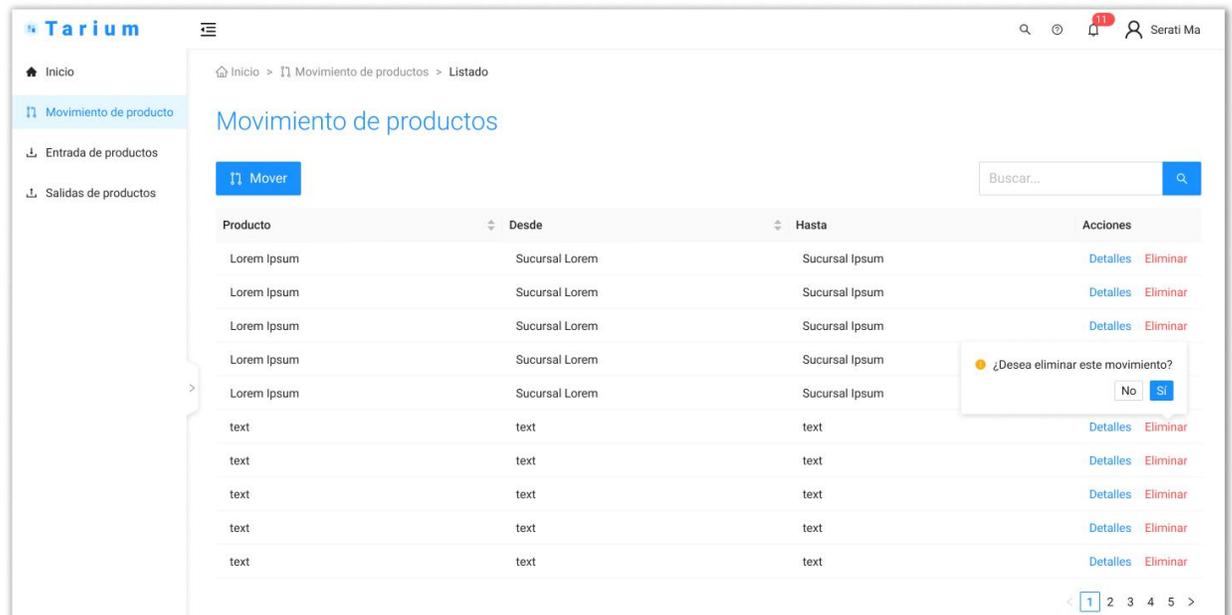


Figura 24. Vista del listado de movimientos (Elaboración propia)

### 4.7.7.2. Formulario

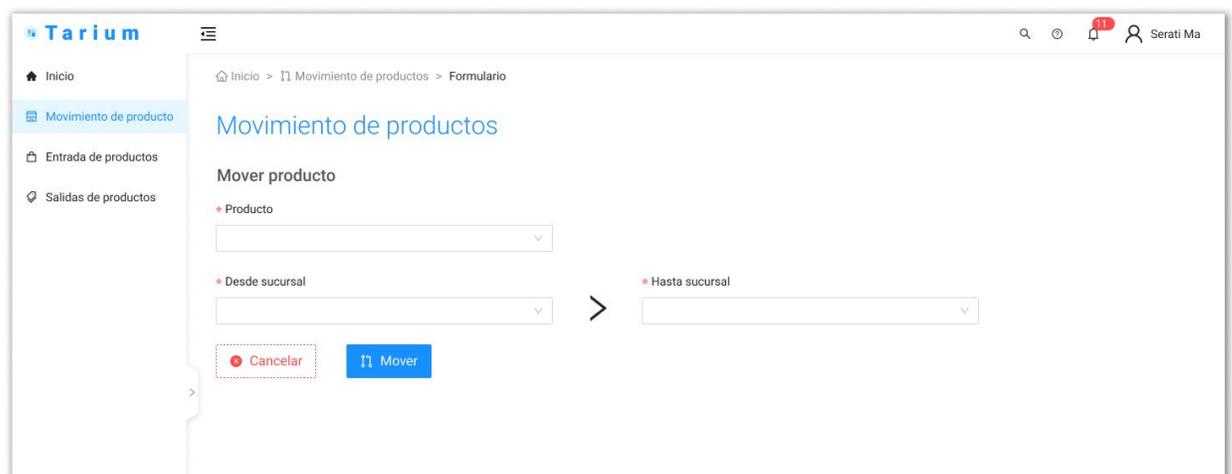


Figura 25. Vista del formulario de movimiento de productos (Elaboración propia)

## 4.7.8. Entradas de productos

### 4.7.8.1. Lista

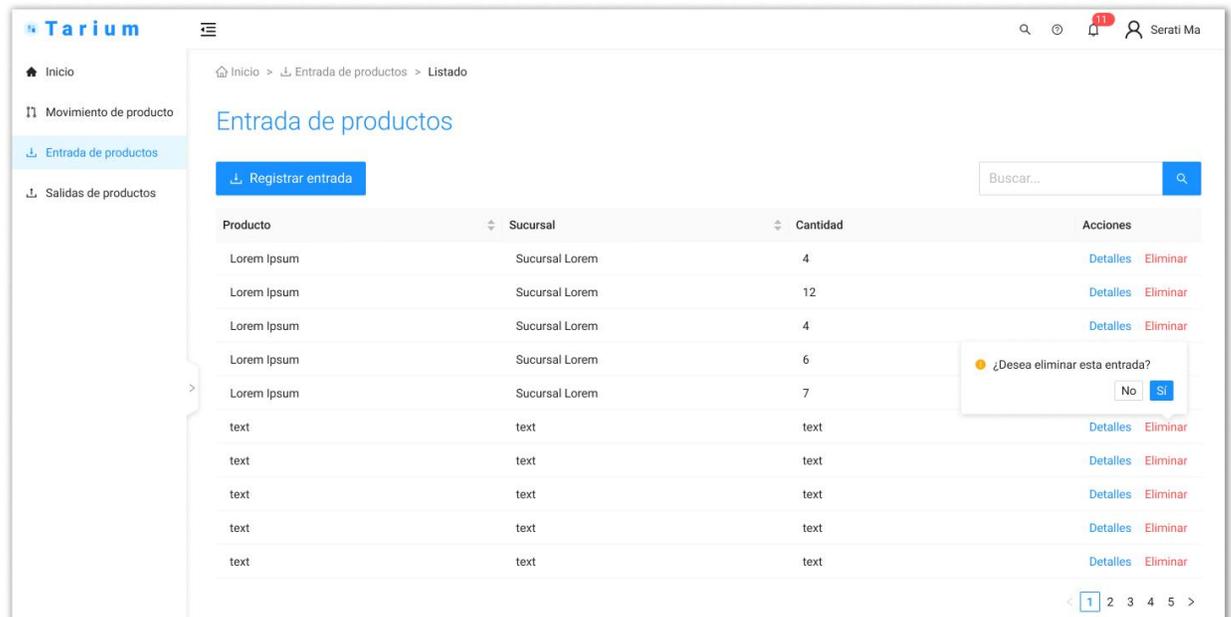


Figura 26. Vista del listado de entradas (Elaboración propia)

### 4.7.8.2. Formulario

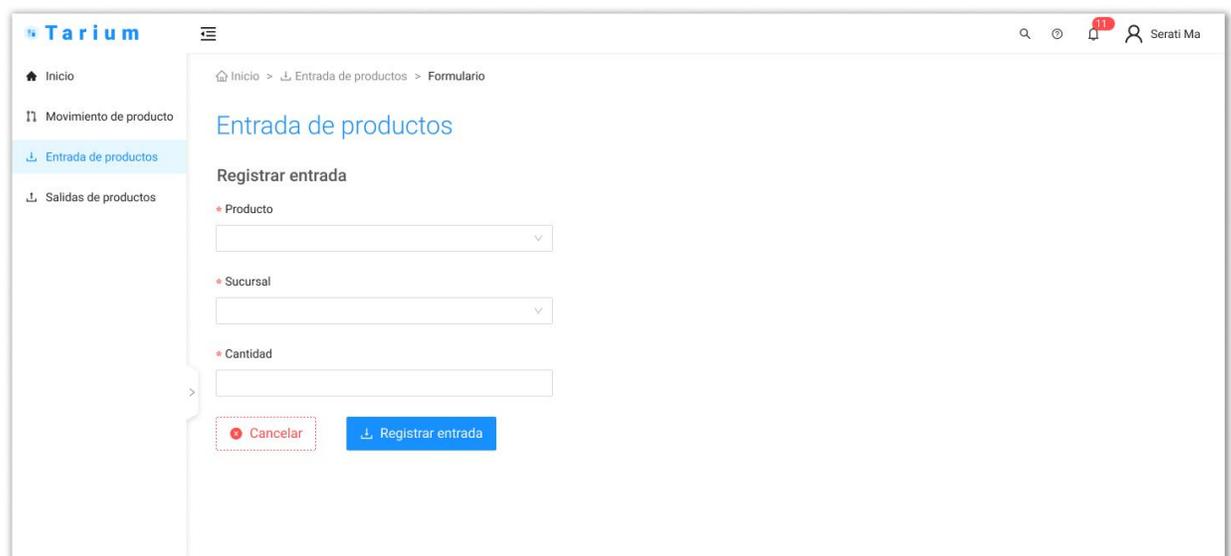


Figura 27. Vista del formulario de entradas de productos (Elaboración propia)

## 4.7.9. Salidas de productos

### 4.7.9.1. Lista

The screenshot displays the 'Salida de productos' (Product Outputs) list view. The interface includes a sidebar with navigation options: Inicio, Movimiento de producto, Entrada de productos, and Salidas de productos. The main content area shows a table with columns for Producto, Sucursal, Cantidad, and Acciones. A search bar is located at the top right. A confirmation dialog is overlaid on the table, asking '¿Desea eliminar esta salida?' with 'No' and 'Si' buttons.

Producto	Sucursal	Cantidad	Acciones
Lorem Ipsum	Sucursal Lorem	4	<a href="#">Detalles</a> <a href="#">Eliminar</a>
Lorem Ipsum	Sucursal Lorem	12	<a href="#">Detalles</a> <a href="#">Eliminar</a>
Lorem Ipsum	Sucursal Lorem	4	<a href="#">Detalles</a> <a href="#">Eliminar</a>
Lorem Ipsum	Sucursal Lorem	6	<a href="#">Detalles</a> <a href="#">Eliminar</a>
Lorem Ipsum	Sucursal Lorem	7	<a href="#">Detalles</a> <a href="#">Eliminar</a>
text	text	text	<a href="#">Detalles</a> <a href="#">Eliminar</a>
text	text	text	<a href="#">Detalles</a> <a href="#">Eliminar</a>
text	text	text	<a href="#">Detalles</a> <a href="#">Eliminar</a>
text	text	text	<a href="#">Detalles</a> <a href="#">Eliminar</a>
text	text	text	<a href="#">Detalles</a> <a href="#">Eliminar</a>

Figura 28. Vista del listado de salidas (Elaboración propia)

### 4.7.9.2. Formulario

The screenshot displays the 'Registrar salida de producto' (Register Product Output) form. The form includes three required fields: Sucursal (dropdown), Producto (dropdown), and Cantidad (text input). Below the fields are two buttons: Cancelar and Registrar salida.

Figura 29. Vista del formulario de salidas de productos (Elaboración propia)

## 4.7.10. Visualización de Inventario

The screenshot shows the Tarium web application interface. The top navigation bar includes the Tarium logo, search, and user profile (Serati Ma). The left sidebar contains navigation links: Inicio, Sucursales, Proveedores, Catálogo de productos, Usuarios, Inventario, and Ayuda. The main content area is titled 'Inventario > Existencias'. It features search filters for 'Sucursal' (with 'item 01' and 'item 02' selected), 'Producto' (with 'item 01' and 'item 02' selected), and 'Fecha' (ranging from 2021-11-08 to 2021-12-23). An 'Exportar' button is visible in the top right. A table displays the inventory data with the following columns: Sucursal, Producto, Cantidad, and Acciones. A dropdown menu is open over the 'Existencias' link in the sidebar, showing 'Visualización avanzada' as an option. The table contains 10 rows of data with various product names and quantities.

Sucursal	Producto	Cantidad	Acciones
Sucursal Lorem	Lorem ipsum	15	<a href="#">Visualización avanzada</a>
Sucursal Lorem	Dolor sit amet	3	<a href="#">Visualización avanzada</a>
Sucursal Lorem	Consectetur adipiscing elit	7	<a href="#">Visualización avanzada</a>
Sucursal Lorem	Nam laoreet luctus	12	<a href="#">Visualización avanzada</a>
Sucursal Ipsum	Lorem ipsum	21	<a href="#">Visualización avanzada</a>
Sucursal Ipsum	Dolor sit amet	6	<a href="#">Visualización avanzada</a>
Sucursal Ipsum	Consectetur adipiscing elit	5	<a href="#">Visualización avanzada</a>
Almacen Sit	Lorem ipsum	8	<a href="#">Visualización avanzada</a>
Almacen Sit	Dolor sit amet	2	<a href="#">Visualización avanzada</a>
Almacen Lorem #2	Lorem ipsum	4	<a href="#">Visualización avanzada</a>

Figura 30. Vista de visualización de inventario (Elaboración propia)

## 4.7.11. Visualización de Transacción de Productos

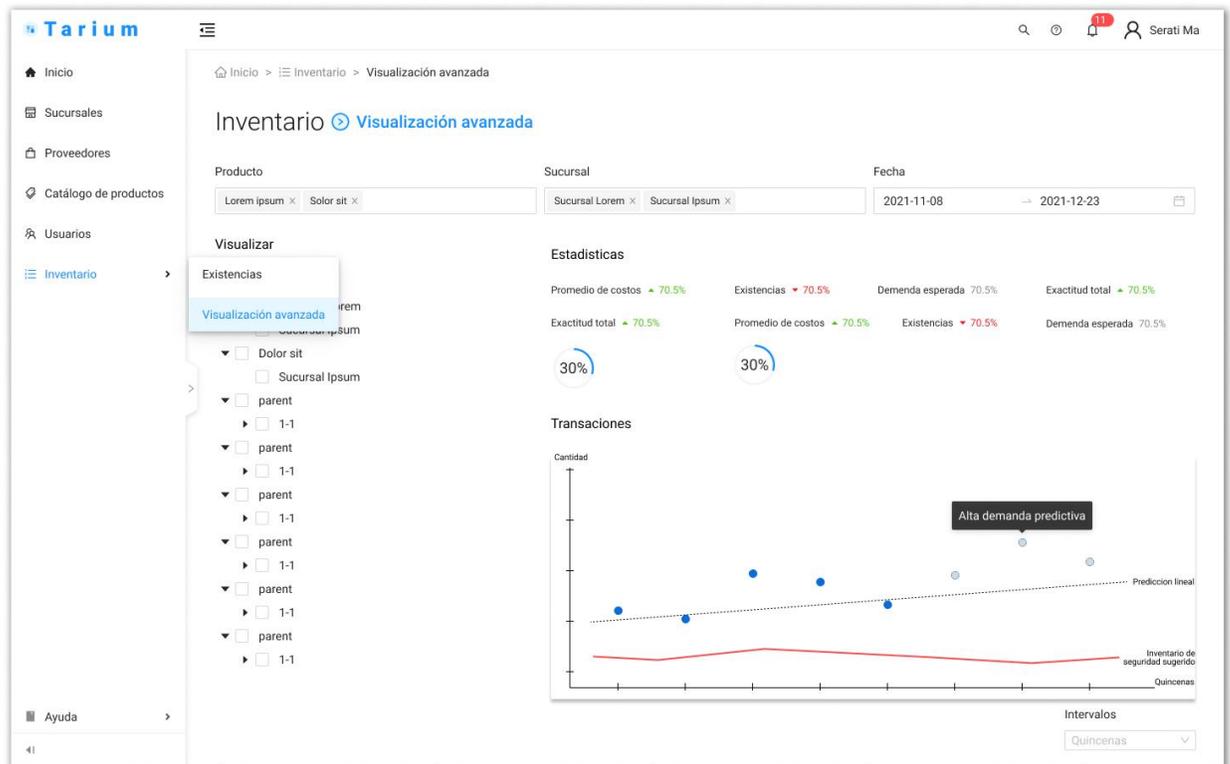


Figura 31. Vista de visualización de transacciones (Elaboración propia)

## Conclusión

En este capítulo se ha trabajado el modelado (incluyendo el visual) de la aplicación web para la automatización del proceso de gestión de inventario. Esto se queda demostrado en el acta de constitución del proyecto, su sesión de requerimientos, el documento visión y alcance, y especialmente a través de los casos de uso, diseños de alto y bajo nivel y las vistas de la aplicación.

También en este capítulo se identificaron y definieron los requerimientos funcionales y no funcionales que se tomarán en cuenta en el proceso de programación de la solución; estos requerimientos también se utilizaron como base para la definición de los casos de uso de la aplicación, que a su vez serán el fundamento para medir el avance del proyecto con el paso del tiempo.

## **Capítulo V: Desarrollo de la Aplicación**

## Introducción

En el presente capítulo, se determinarán los patrones y macro-patrones de arquitectura y diseño a utilizar para el desarrollo de esta solución web, implementando las mejores prácticas de programación para el tipo de solución a desarrollar.

También se estará seleccionando la plataforma de desarrollo y el lenguaje de programación a utilizar, mostrando finalmente una versión de demostración de la aplicación implementando las tecnologías a utilizar, funcionando este como marco de trabajo para el desarrollo funcional de la solución .

## 5.1. Selección de los Macro-patrones de Arquitectura

### 5.1.1. Single Page Application (SPA)

Este tipo de aplicaciones está conformado de componentes individuales que se pueden reemplazar o actualizar de forma independiente, de modo que no es necesario recargar toda la página en cada acción del usuario, lo que ahorra ancho de banda y no carga archivos externos cada vez que se carga una página, como imágenes o archivos CSS, etc. El propósito detrás de esto es hacer que la página subsiguiente se cargue muy rápido en comparación con el ciclo tradicional de solicitud-respuesta.

SPA permite una forma más flexible y elegante de tratar los datos; actualizar una parte particular o una sección de una página sin actualizar una página completa es el objetivo principal que ofrece SPA, pero toda esta flexibilidad requiere una interfaz más interactiva y esto conduce a una mejor experiencia de usuario.

#### 5.1.1.1. Características Principales

Las características principales de este tipo de aplicación son:

- **Componentes individuales:** toda la página se divide en componentes más pequeños que interactúan entre sí.
- **Reemplazado y actualizado:** un componente, región o parte de una página se puede reemplazar o actualizar con cualquier cambio en las solicitudes del

usuario.

- **Actualización y recarga:** la página completa nunca se recarga ni se actualiza, sino que se carga contenido nuevo en alguna parte o sección según los nuevos datos.
- **Acciones del usuario:** la aplicación es responsable de manejar todas las interacciones del usuario, como el clic de un botón, la entrada desde el teclado, etc., de manera muy rápida y, por lo tanto, conduce a una interfaz de usuario muy fluida.

## 5.2. Selección de los Patrones de Diseño

Los patrones de diseño son técnicas que se utilizan para resolver problemas comunes en el desarrollo de software y otras áreas relacionadas con la interacción o el diseño de interfaces.

Para que una solución se considere un patrón, debe tener ciertas propiedades: reutilizable y eficaz resolviendo problemas similares en el pasado.

### 5.2.1. Inyección de Dependencias

Se utilizará inyección de dependencias como patrón de diseño principal, ya que nos permite solucionar la necesidad de creación de los objetos, principalmente de objetos que manejan las instancias de base de datos, de una manera práctica.

Este patrón de diseño, tiene una integración nativa con las herramientas y los lenguajes de programación que se utilizarán para desarrollar esa solución, principalmente en la construcción del Web API del lado del servidor; esto nos ayudará a tener una aplicación más simple de mantener y adaptada a las necesidades de programación actuales.

## **5.3. Buenas Prácticas de Codificación a Implementar**

### **5.3.1. Principios Solid Gold**

El término Solid Gold se refiere al conjunto de principios de diseño implementados en el desarrollo de software orientado a objetos. SOLID es un acrónimo que deriva de cinco principios clave de diseño: el principio de responsabilidad única, el principio de abierto-cerrado, el principio de sustitución de Liskov, el principio de segregación de interfaz y el principio de inversión de dependencia.

### **5.3.2. Código Limpio (Clean Code)**

En palabras simples este código se refiere a aquellos códigos fáciles de entender y modificar. En otras palabras son esos códigos que resultan manejables de ampliar y refactorizar y que de igual forma resulta sencillo corregir errores en la base del código.

## **5.4. Selección de las Plataformas de Desarrollo y DevOps**

### **5.4.1. Selección de la Plataforma de Desarrollo**

Debido a que esta solución se divide en tres partes fundamentales: base de datos, Web API, aplicación SPA; la misma tendrá plataformas de desarrollo distintas por cada parte de la solución.

#### **5.4.1.1. Base de Datos**

Para el desarrollo de la base de datos se utilizará el motor MySQL, desarrollado y mantenido por Oracle. Este motor de base de datos está bajo la licencia de open-source y multiplataforma, lo que ayuda a que la solución mantenga el nivel de flexibilidad necesario como para poderla desplazar en cualquier sistema operativo del mercado.

El sistema de gestión de base de datos a utilizar será MySQL Workbench, también desarrollada por Oracle, nos da acceso a todas las herramientas básicas y avanzadas para administrar nuestra base de datos mientras se encuentre en desarrollo.

#### **5.4.1.2. Web API**

Se utilizará Visual Studio 2019 para el desarrollo del Web API que se necesita del lado del servidor, esta herramienta es un entorno de desarrollo integrado (IDE por

sus siglas en inglés), desarrollado por Microsoft, que tiene soporte nativo con el marco de trabajo .NET 5.0 y C#, lo que permitirá un entorno de desarrollo confiable y con los estándares de eficiencia necesarios para desarrollar esta aplicación con éxito.

#### **5.4.1.3. Aplicación SPA**

Para el desarrollo de la aplicación web se utilizará Visual Studio Code, el cual es un editor de código fuente popular para el desarrollo de aplicación utilizando ReactJs, gracias al gran soporte que ofrece su comunidad para ofrecer extensiones que ayudan a desarrollar este tipo de aplicaciones, y además este editor cuenta con soporte nativo para NodeJs.

#### **5.4.2. Selección de la Plataforma de DevOps**

El software de control de versiones a utilizar es GIT, es cual deberá estar instalado en las computadoras de los desarrolladores, este software ofrece funcionalidades de control de versiones descentralizadas, lo que permite que los desarrolladores tengan muy buena flexibilidad al momento de gestionar las versiones locales de su trabajo.

El sistema descentralizado de GIT también permite tener varias opciones de repositorio para compartir versiones entre desarrolladores, en este caso, se optará

por utilizar GitHub, actualmente adquirida por Microsoft, lo que ofrece un ambiente seguro y eficiente para compartir versiones de código fuentes de las aplicaciones a desarrollar.

Otra de las ventajas de utilizar GIT, es que tenemos un rango de aplicaciones que ofrecen una interfaz gráfica para interactuar con nuestro repositorio, estas aplicaciones se eligen a criterio de cada desarrollador, sin embargo, recomendamos utilizar, GitKraken por la gran cantidad de opciones que ofrece para la gestión de las versiones locales del código fuente.

## **5.5. Selección del Lenguaje de Programación**

### **5.5.1. Selección del Lenguaje de Programación Frontend**

Se utilizará Javascript para la programación de la interfaz de la aplicación web, que en conjunto a la librería ReactJS se pueden desarrollar aplicaciones SPA de la mayor calidad en el mercado en términos de velocidad y mantenibilidad.

### **5.5.2. Selección del Lenguaje de Programación Backend**

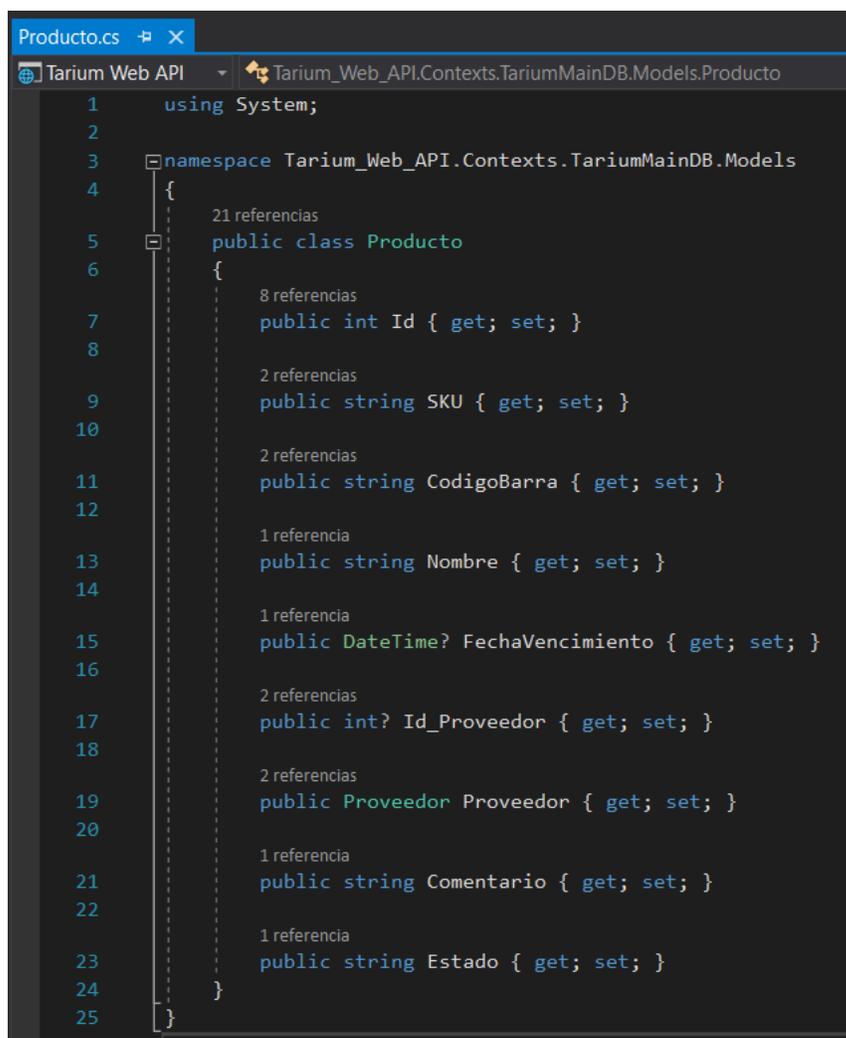
Se utilizará C# para la programación del lado del servidor, debido a que este lenguaje cuenta con plataformas desarrolladas por Microsoft, que permiten el desarrollo rápido, seguro y eficaz de aplicaciones Web API, como necesitamos en este caso, y en conjunto con el marco de trabajo .NET 5.0 de C#, este desarrollo puede ser desplegado en cualquier sistema operativo del mercado, lo que nos brinda muchas posibilidades al momento del despliegue de la aplicación.

## 5.6. Desarrollo de la Arquitectura Candidata

### 5.6.1. Ejemplo del CRUD de Gestión de Sucursales

Como bien describen las siglas de un CRUD (en inglés), este nos permite crear, leer, actualizar y eliminar los elementos que este administra, para el caso de este ejemplo, la gestión de productos.

#### 5.6.1.1. Modelo



```
1 using System;
2
3 namespace Tarium_Web_API.Contexts.TariumMainDB.Models
4 {
5     21 referencias
6     public class Producto
7     {
8         8 referencias
9         public int Id { get; set; }
10
11         2 referencias
12         public string SKU { get; set; }
13
14         2 referencias
15         public string CodigoBarra { get; set; }
16
17         1 referencia
18         public string Nombre { get; set; }
19
20         1 referencia
21         public DateTime? FechaVencimiento { get; set; }
22
23         2 referencias
24         public int? Id_Proveedor { get; set; }
25
26         2 referencias
27         public Proveedor Proveedor { get; set; }
28
29         1 referencia
30         public string Comentario { get; set; }
31
32         1 referencia
33         public string Estado { get; set; }
34     }
35 }
36
```

Figura 32. Modelo de la entidad producto (Elaboración propia)

Este es el modelo de datos utilizado para la transferencia de datos entre la Web API y la base de datos, utilizando el marco de trabajo Entity Framework Core 5 que nos provee .NET 5.0.

## 5.6.1.2. Vistas

### 5.6.1.2.1. Listado

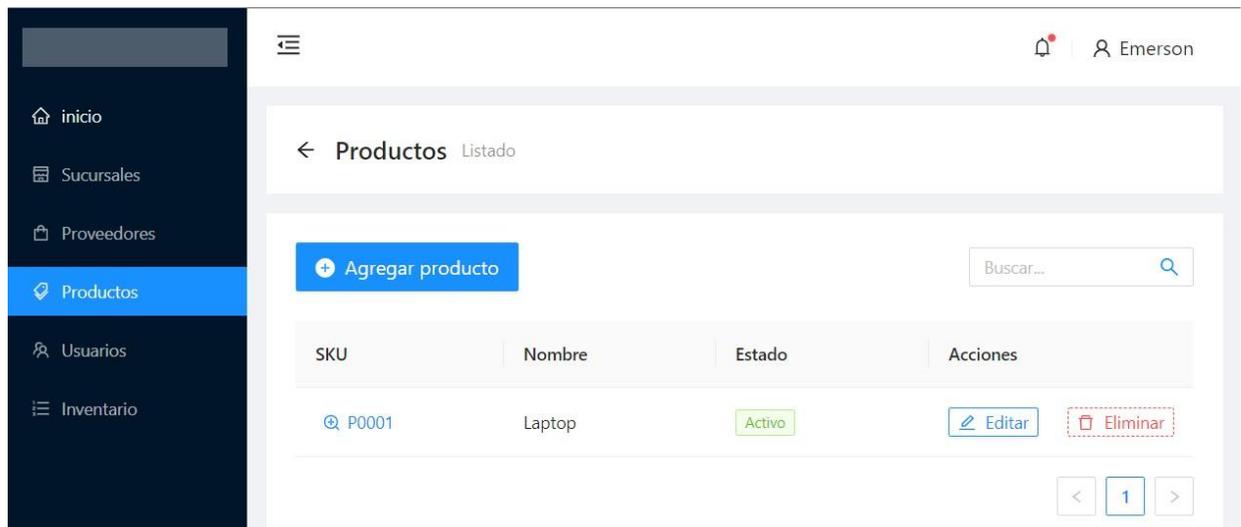


Figura 33. Vista programada del listado de productos (Elaboración propia)

En esta vista se muestra cómo el usuario puede visualizar el listado de productos que tiene registrados, con la opción de realizar búsqueda básica; en esta pantalla el usuario también posee acceso para ver el detalle de cada producto, la posibilidad de editarlos, eliminarlos o agregar algún producto a la lista.

## 5.6.1.2.2. Formulario

The screenshot displays a web interface for adding a product. On the left is a dark sidebar with navigation links: Inicio, Sucursales, Proveedores, Productos, Usuarios, and Inventario. The main content area is titled 'Productos Formulario' and contains the following fields:

- Agregar producto**
- SKU**: A text input field with a red asterisk indicating it is required.
- Código de barra del proveedor**: A text input field.
- Nombre**: A text input field with a red asterisk indicating it is required.
- Proveedor**: A dropdown menu.
- Fecha de vencimiento**: A date picker with the text 'Select date' and a calendar icon.
- Comentario**: A text area with a character count '0 / 1000'.
- Estado**: A toggle switch currently set to 'Activo' (Active).

At the bottom of the form are two buttons: 'Cancelar' (Cancel) and 'Guardar' (Save).

Figura 34. Vista programada del formulario de registro de productos (Elaboración propia)

En esta vista el usuario tiene acceso a un formulario que le permite agregar un producto a su lista de productos existentes, cuenta con una opción de cancelar para volver atrás en cualquier momento.

### 5.6.1.3. Controlador

```
ProductosController.cs* X
Tarium Web API Tarium_Web_API.Controllers.ProductosController
8
9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class ProductosController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public ProductosController(TariumMainDB_Context context)
18         {
19             _context = context;
20         }
21
22         [HttpGet]
23         public async Task<ActionResult<IEnumerable<Producto>>> GetProductos()
24         {
25             return Ok(_context.Productos.ToList());
26         }
27
28         [HttpGet("{id}")]
29         public async Task<ActionResult<Producto>> GetProducto(int id)
30         {
31             return Ok(_context.Productos.FirstOrDefault(p => p.Id == id));
32         }
33
34         [HttpPut("{id}")]
35         public async Task<IActionResult> PutProducto(int id, Producto producto)
36         {
37             _context.Productos.Update(producto);
38             await _context.SaveChangesAsync();
39             return Ok(producto);
40         }
41
42         [HttpPost]
43         public async Task<ActionResult<Producto>> PostProducto(Producto producto)
44         {
45             _context.Productos.Add(producto);
46             await _context.SaveChangesAsync();
47             return Ok(producto);
48         }
49
50         [HttpDelete("{id}")]
51         public async Task<IActionResult> DeleteProducto(int id)
52         {
53             _context.Productos.Remove(_context.Productos.FirstOrDefault(p => p.Id == id));
54             await _context.SaveChangesAsync();
55             return Ok();
56         }
57
58         private bool ProductoExists(int id)
59         {
60             return _context.Productos.Any(p => p.Id == id);
61         }
62     }
63 }
```

Figura 35. Controlador de la entidad productos (Elaboración propia)

Este es el controlador Web API de para el manejo de las peticiones web de los productos, la forma en la que se encuentra desarrollada no permite utilizar el protocolo HTTP/HTTPS para conocer la intención de la petición; siendo GET utilizada para obtener la lista de productos o un producto específico al enviarle el identificador principal del producto; PUT para actualizar la información de algún producto; POST para agregar un producto a la lista de la base de datos; y DELETE para eliminar un producto específico de la base de datos.

## 5.6.2. Desarrollo del Menú

De acuerdo al diseño modelado del sistema, existen dos tipos de usuario distintos que interactúan con la aplicación web; el primero es el usuario administrador, encargado de gestionar las informaciones base del sistema; el segundo es el usuario transaccional, encargado del registro de entradas y salidas de los productos de la empresa. A continuación se muestran los distintos accesos relacionados a cada tipo de usuario.

### 5.6.2.1. Usuario Administrador

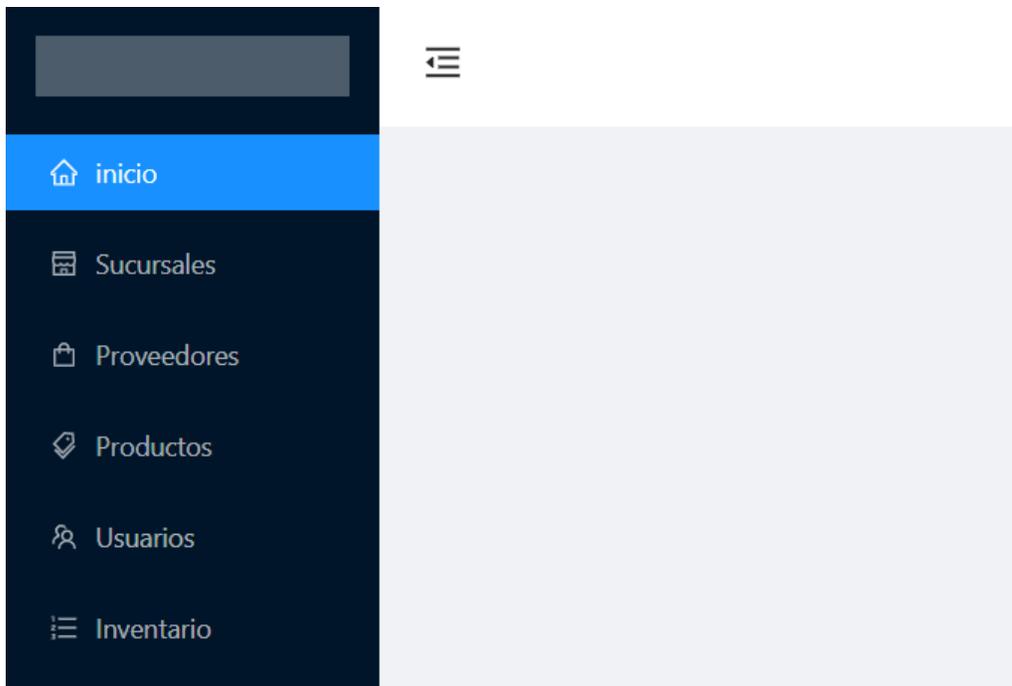


Figura 36. Menú del usuario administrador (Elaboración propia)

Este menú muestra las opciones que el usuario administrador necesita para acceder a las funciones de gestión de recursos del sistema de inventario.

### 5.6.2.2. Usuario Transaccional

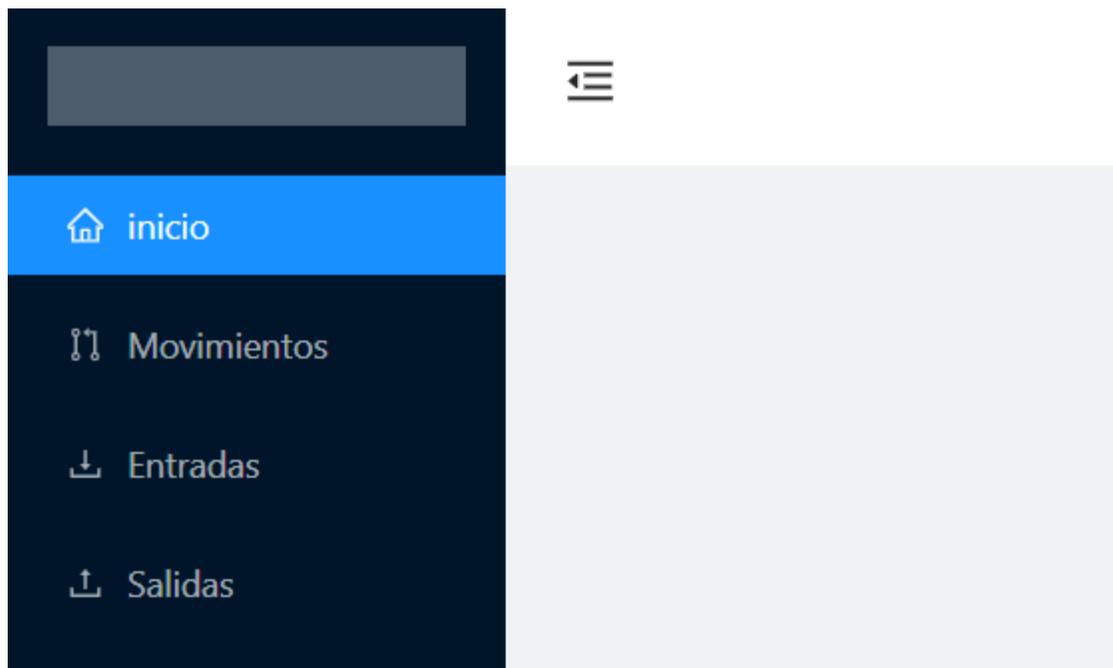


Figura 37. Menú del usuario transaccional (Elaboración propia)

En este menú se le da acceso al usuario transaccional al manejo de las entradas y salidas de productos del sistema, para así tener registro de los movimientos de inventario y utilizar estos datos para automatizar los procesos.

## Conclusión

En este capítulo se programó la arquitectura y marcos de trabajo necesarios para el desarrollo completo de la solución. Se mostró las dos aplicaciones principales que forman la solución y cómo interactúan entre ellas para cumplir con los requerimientos necesarios como para implementar esta solución de manera exitosa.

Una vez este proceso de desarrollo haya concluido, se procede a desplegar todas las partes de la solución en servidores en la nube para que se encuentre disponible para los usuarios que deseen utilizar sus funcionalidades.

En el siguiente capítulo se profundizará más sobre cómo desplegar y mantener esta solución de forma exitosa.

# Capítulo VI: Implementación de la Aplicación

## Introducción

El despliegue de software son todas las actividades que hacen que un sistema de software esté disponible para su uso. El proceso de implementación general consiste en varias actividades interrelacionadas con posibles transiciones entre ellas.

La actividad de publicación se deriva del proceso de desarrollo completado y, a veces, se clasifica como parte del proceso de desarrollo en lugar del proceso de implementación. Este proceso incluye todas las operaciones para preparar un sistema para el compilado y la transferencia a los sistemas informáticos en los que se ejecutará en producción. Por lo tanto, a veces implica la determinación de los recursos necesarios para que el sistema funcione con un rendimiento y planificación tolerables y/o documente las actividades posteriores de este proceso.

La adopción de la nube aumenta a medida que el desarrollo ágil, la implementación rápida y la escala ilimitada se convierten en la nueva normalidad para los clientes de todos los sectores, tamaños y geografías. En la segunda evaluación de Gartner, que abarca tanto la infraestructura en la nube como los servicios de plataforma (IaaS y PaaS o "CIPS"), AWS es evaluado como líder y se sitúa en la posición más alta en ambos ejes de medición, capacidad de ejecución e integridad de visión.

## **6.1. Selección del Proveedor de Servicios de Nube**

AWS (Amazon Web Services) es el líder indiscutible de prestaciones de servicios de nube (IaaS, PaaS, SaaS, etc.), cuenta con miles de servicios activos a disposición del cliente en sus distintas modalidades, es muy estable y robusto.

Windows Azure, perteneciente a la empresa Microsoft es uno de los principales competidores, quienes cuentan con servicios de nube muy estables y flexibles. Su plataforma de administración es bastante intuitiva y cuenta con muchas empresas (incluso muchas locales) que son Gold Partners y ofrecen sus servicios para acompañarte en el proceso de transición hacia la nube.

Google los líderes actuales en cuanto a innovación, son apreciados positivamente por los clientes, siguen vistos como una empresa de hoy y con permanencia en el mañana, ofrecen un abanico amplio de servicios en su nube, los cuales son creados y usados internamente por la propia empresa. Son creadores de diversas tecnologías que se han vuelto estándares de la nube.

Oracle continúa estando entre los grandes, sobre todo lidera en el mundo de las empresas que usaban localmente sus productos, ERP y Base de datos, quienes han ido poco a poco migrando a la nube.

IBM es una de las empresas de mayor permanencia y experiencia en tecnología

(Software y Hardware) durante los últimos 70 años, cuentan con una nube rica y variados servicios a disposición de empresas de distintos tamaños y necesidades.

Luego de esta breve caracterización de los principales proveedores, decidimos optar por Windows Azure por su integración con los IDE que se seleccionaron para este proyecto, por su flexible herramienta de devops, por ser los proveedores del SGDB, por sus costos flexibles, lo cual permite a todo publico adaptarse a sus necesidades y por su esquema de elasticidad.



Figura 38. Principales proveedores de servicios de nube, de gartner.com, 2021

## **6.2. Selección del Dimensionamiento para el Servidor Web**

Azure Container Instances ofrece la forma más rápida y sencilla de ejecutar un contenedor en Azure, sin tener que provisionar ninguna máquina virtual ni alguna otra herramienta, se trata tan sólo de su aplicación, en un contenedor, que se ejecuta en la nube. Con Azure Container Instances, puede ejecutar fácilmente contenedores con un solo comando.

Se ha seleccionado esta opción de despliegue por su garantía de ponerse en marcha en poco tiempo, mientras se reducen los costos de infraestructura gracias a la facturación por segundo y los tamaños de máquina.

Por otro lado, se cuenta con escalabilidad según la demanda, ya que cuando este aumenta se habilita otra máquina con recursos extras para satisfacer la demanda y se paga por lo consumido.

Visite [Azure \(Windows Azure, 2021\)](#) para más detalles de cómo configurar paso a paso el servicio.

### **6.3. Selección del Dimensionamiento para la Base de Datos**

Al igual que el caso anterior la misma estará alojada en Azure Container Instances (Windows Azure, 2021). El Gestor de Base de Datos es Oracle MySQL, en su versión 8.0, ejecutando en un Sistema Operativo Linux.

El pool de conexiones se ha configurado para soportar la media de las peticiones concurrentes que estarán recibiendo diariamente.

## **6.4. Selección y Configuración del WAF Como Servicio**

Un Firewall de aplicaciones web (WAF) proporciona protección centralizada para las aplicaciones web. WAF defiende los servicios web frente a vulnerabilidades comunes. Mantiene su servicio de alta disponibilidad para los usuarios y le ayuda a cumplir los requisitos de cumplimiento.

Un WAF es una solución global y centralizada. Se implementa en ubicaciones de perímetro de red del proveedor de nube en todo el mundo. Las aplicaciones web habilitadas para WAF inspeccionan todas las solicitudes entrantes entregadas por dicho Firewall en el perímetro de red.

Un WAF evita los ataques malintencionados cerca de los orígenes de ataques antes de que entren en la red virtual. El usuario obtiene protección a gran escala sin sacrificar el rendimiento. En el caso de Azure, una directiva WAF se vincula fácilmente a cualquier perfil de Front Door de la suscripción. Las nuevas reglas se implementan en cuestión de minutos, por lo que puede responder rápidamente a los cambios en los patrones de amenazas.

Se ha seleccionado Azure Front Door como WAF para la aplicación que estamos desplegando. Se contrata el servicio y se coloca por delante de nuestra aplicación para evitar que lleguen a la misma los ataques del OWASP Top Ten, sin necesidad

de que el equipo de desarrollo se deba cargar la responsabilidad de tener que implementar estas validaciones en su código, retrasando el entregable y dedicándose a algo que no es su mayor área de conocimiento y que puede ser delegada.

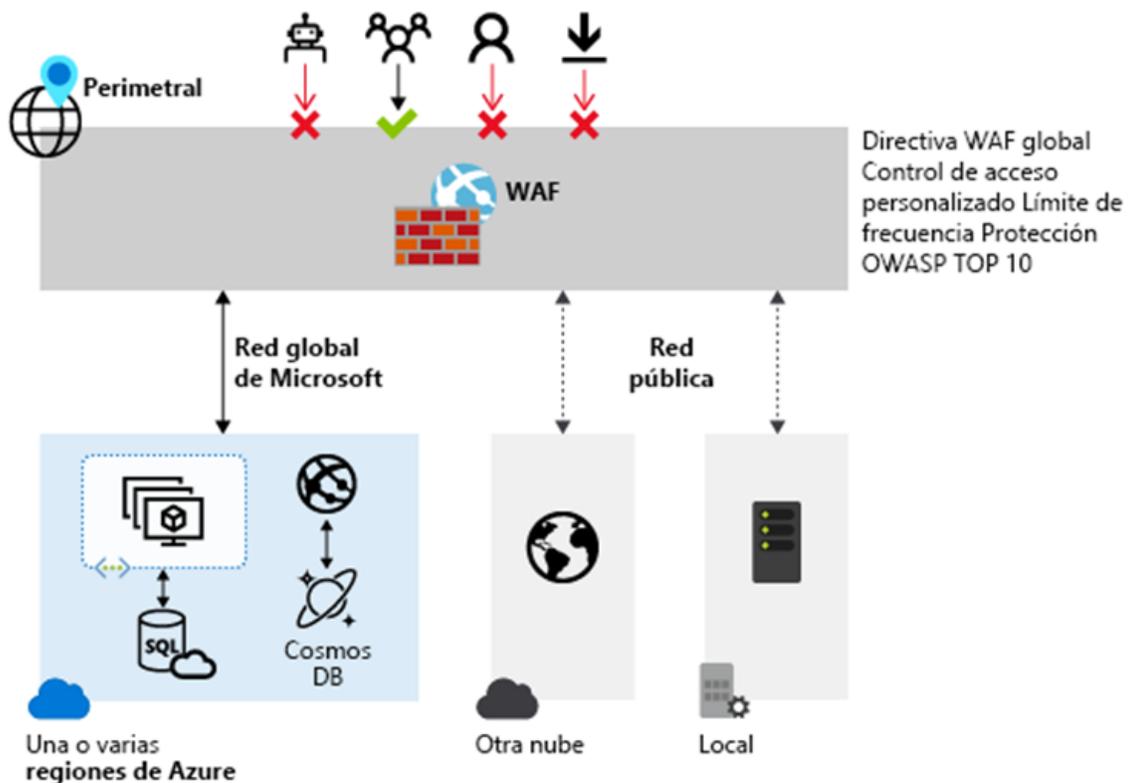


Figura 39. Arquitectura WAF de Azure, recuperado de [azure.microsoft.com](https://azure.microsoft.com), 2021

## **6.5. Despliegue de la Aplicación en la Nube**

El despliegue se hará siguiendo las directrices de dockers y kubernetes pautadas por Azure como proveedor de nube y su integración con los agentes de estas tecnologías (Docker Hub Repository, entre otros), a fin de simplificar el proceso de puesta en producción y garantizar que lo mismo que teníamos en desarrollo es lo mismo que tendremos en producción, sin fallas en la configuración ni cambios en el ambiente / plataforma subyacente.

### **6.5.1. Ventajas de Docker**

- Flexibilidad para admitir microservicios y cargas de trabajo de aplicaciones tradicionales.
- Compatibilidad con contenedores Linux y Windows Server.
- Control de acceso basado en roles (RBAC) granular, compatibilidad con el protocolo de acceso a directorio ligero (LDAP) e integración con Azure Active Directory.
- Modelo de seguridad de extremo a extremo para una cadena de suministro más segura.
- Administración y operación basadas en una interfaz de usuario gráfica integrada.

## 6.5.2. Ventajas de la Integración de Azure

- Balanceadores de carga auto provisionados y auto configurados.
- Experiencia de usuario administrador de Azure familiar.
- Red de Azure y configuración de instancias seguras y de sencilla administración.
- Profunda integración con la infraestructura de Azure subyacente: se obtienen funcionalidades nativas de la infraestructura sin configuración adicional.

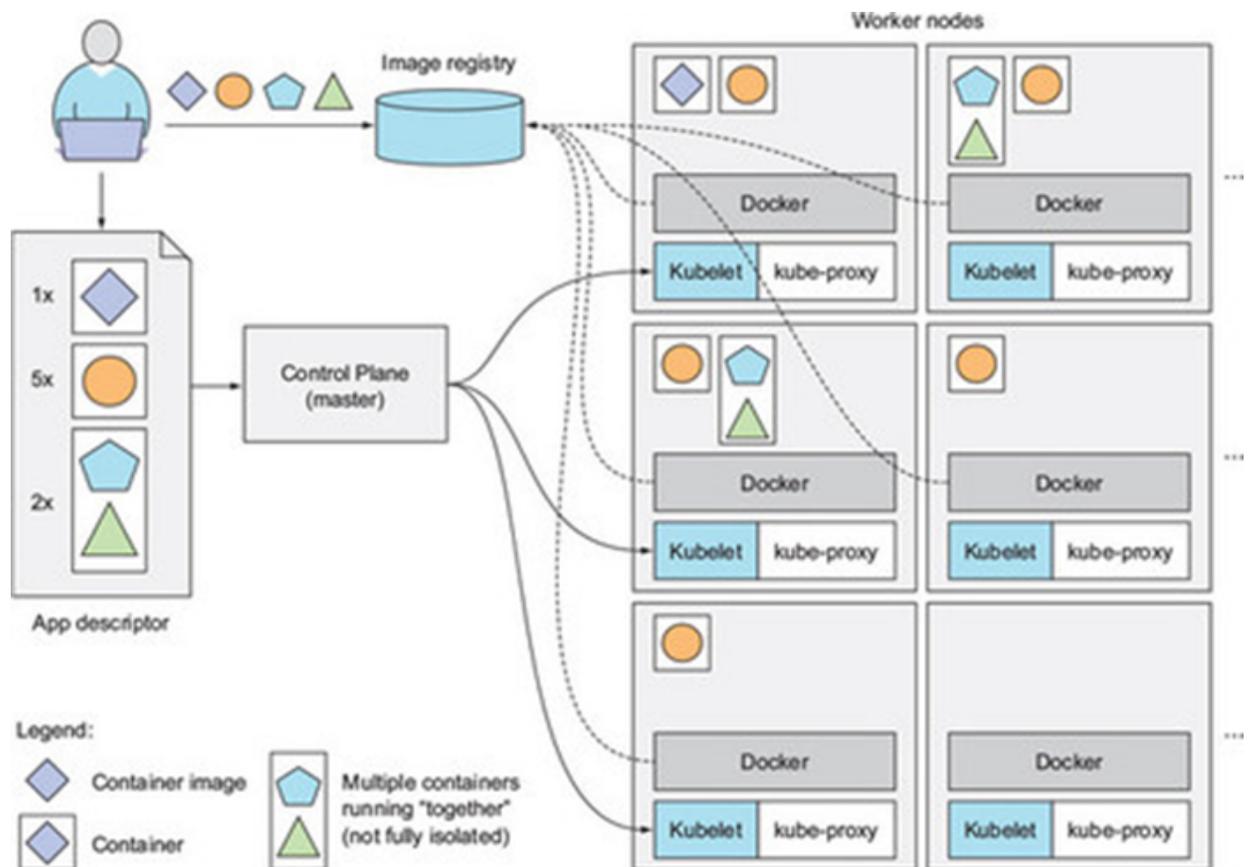


Figura 40. Docker y Kubernetes, recuperada de <https://www.xataka.com/>, 2021

## **6.6. Plan de Implementación y de Reverso**

Es correcto que elaboremos un plan de implementación en el cual se detallen cada uno de los pasos que irán dando para implementar la aplicación web en la nube, en orden y sin omisión de aspectos relevantes. Se debe agregar también que aspecto garantiza que la tarea fue concluida de forma exitosa, a fin de que sean verificados en el trayecto.

El plan debe incluir el personal involucrado, su rol, tareas asignadas y en qué momento del proceso el mismo entra en juego.

De igual forma se debe incluir una sección del plan o un documento aparte, según su extensión, en el cual se detallen los pormenores ameritados para desinstalar la aplicación, si el proceso de implementación fue fallido por alguna razón. Es decir, se debe crear una especie de “lista de verificación” con los pasos a dar para desmontar o desinstalar la aplicación, incluyendo responsables, orden exacto de cada paso, entre otros.

## 6.7. Prueba de Estrés de la Aplicación Desplegada

Para garantizar que la aplicación ya desplegada funcione adecuadamente antes de comenzar a comercializarla y hacer pública su URL, se recomiendan realizar pruebas de estrés de la misma, las cuales procuran llevar la plataforma a su límite, para ver cuántos usuarios concurrentes la misma soporta realizando sus actividades habituales.

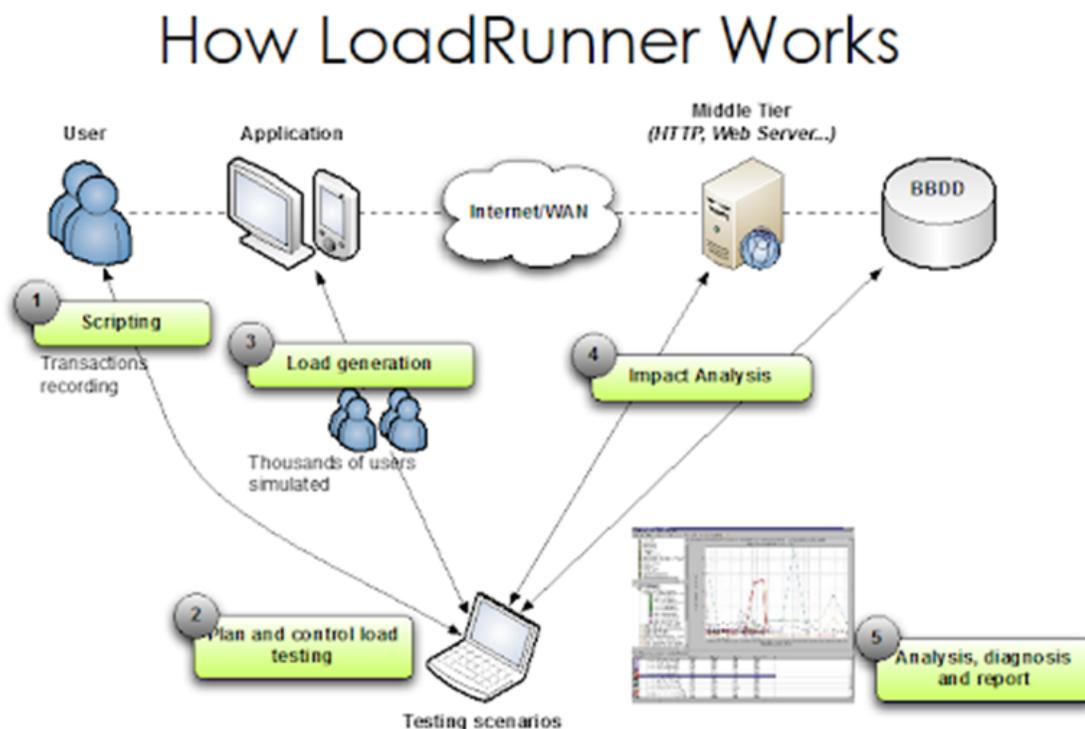


Figura 41. Pruebas de estrés, recuperado de <https://testeandosoftware.com/>, 2021

Estas pruebas también permiten identificar cuál es el “talón de Aquiles” de los recursos, es decir, saber cual es el recurso (memoria, procesador, disco, etc.) que

más rápidamente se agotará cuando estemos al tope máximo de usuarios. De igual forma permitirá confirmar que nuestra estrategia de escalabilidad / elasticidad funciona adecuadamente, pues se habilitará un nuevo contenedor hasta llegar al máximo configurado para ir saciando la demanda en aumento.

Para no tener que instalar y configurar una herramienta open source como Jmeter, se contratará el servicio de LoadRunner, se le configura la cantidad de usuario que se desean emular, la URL de la aplicación a llamar y los servicios a consumir (previamente configurados) y al finalizar se analizan los resultados y se realizan los ajustes correspondientes. Para más detalles véase (LoadRunner, 2021)

## 6.8. Revisión Post-Implementación

La post implementación asegura un buen trabajo. Los ejecutivos y miembros del equipo del proyecto, al igual que el equipo de desarrollo y de SQA llegan al final de la implementación, dispuestos a tirar la toalla y acabar con el dolor, el riesgo y la perturbación creada por el despliegue. No es extraño que la última cosa que pase por sus mentes sea trabajar para determinar los resultados reales después de la implementación.

Es importante realizar una lista en frío antes de llegar a esta etapa para evaluar en el entorno de producción:

- Que los resultados funcionales son los adecuados.
- Que en términos de performance y seguridad la aplicación se comporta según lo esperado.
- Que la aplicación quedó debidamente configurada y accesible.
- Realice una prueba de WPO usando herramientas como GTMetrix para garantizar que los tiempos de carga estén bien en producción.

## Conclusión

En este capítulo se estuvieron ponderando distintas modalidades de implementación del producto de software en la nube, así como los proveedores líderes actuales de servicios de nube pública, todo esto con la finalidad de seleccionar cada opción tratada en cada punto particular de este último apartado del trabajo de grado.

No es de extrañar que "la nube" sea la opción de preferencia por encima de la premisa y que la adopción de modelos "serverless", en el cual la administración descansa en un tercero, experto dicho sea de paso, en esas plataformas, en lugar de cargarse a la empresa desarrolladora, conocedora de su dominio de negocio, de su aplicación y de las tecnologías particulares que utiliza.

Ya que ha certificado que todo está listo, inicie su estrategia de SEO, su publicidad en las redes sociales, visite puerta a puerta a las PyMES identificadas como candidatas a usar su aplicación y venda su producto.



## **V. CONCLUSIONES**

## Conclusiones

El Trabajo de Grado presentado tuvo como objetivo principal presentar una propuesta de desarrollo e implementación de un sistema de inventario por suscripción para las PyMES de Santo Domingo bajo la arquitectura SaaS en la nube, con la finalidad de agilizar el proceso de gestión de inventario, eficientizar los tiempos de respuestas y ahorrar el gasto que genera administrarlo con los métodos convencionales.

El sistema ofrece a las PyMES la información necesaria para tomar mejores decisiones de negocio al momento de gestionar y reabastecer su inventario. Además, la plataforma cuenta con un sistema de predicción de demanda adaptativa que ayudará a los usuarios a organizar la existencia de cada producto de forma más eficiente.

Al realizar la investigación sobre el proceso de gestión de inventario de las PyMES de Santo Domingo se dio a conocer que estas no suelen contar con herramientas adecuadas para administrar correctamente este proceso, y la mayoría de estas estuvieron de acuerdo en tener una herramienta dedicada a este proceso específicamente.



## **VI. RECOMENDACIONES**

## Recomendaciones

A continuación, se muestran recomendaciones para el desarrollo e implementación de esta solución:

- Se recomienda realizar respaldos de la información de la base de datos del sistema semanalmente de forma mínima o contratar este servicio al proveedor de la nube seleccionado en almacenamiento económico.
- Se recomienda verificar los respaldos de la información de la base de datos luego de crearlos.
- Se recomienda crear una sección de ayuda al usuario con documentación útil sobre el uso de la aplicación.
- Se recomienda realizar campañas publicitarias utilizando redes sociales para aumentar el alcance de la aplicación.
- Se recomienda encriptar los datos de la base de datos ya que puede contener información sensible de los usuarios.
- Se recomienda utilizar una función de hashing para almacenar las credenciales de los usuarios en la base de datos.
- Se recomienda que la Web API sea desarrollada con el framework ASP.NET, y .NET Core 5.
- Se recomienda que la aplicación web sea construida bajo el modelo SPA utilizando ReactJS.
- Se recomienda tener una gestión adecuada de los riesgos a tomar en cuenta

durante el desarrollo e implementación del proyecto.

- Se recomienda utilizar Selenium IDE para la automatización de las pruebas del sistema.
- Se recomienda utilizar Google PageSpeed Insight para certificar la aplicación web cuando se publique.
- Se recomienda configurar adecuadamente el límite de elasticidad de la solución definiendo hasta cuántas máquinas se podrá crecer según aumente la demanda de los clientes, estos parámetros deben ser reevaluados periódicamente.
- Se recomienda dar seguimiento al uso de la aplicación por parte de los clientes, a fin de identificar oportunidades de capacitación, garantizar el uso continuo de la misma y la captación de nuevos clientes, mediante referencia de los actuales.
- Se recomienda desarrollar un módulo administrativo para los dueños de la aplicación donde adicional al punto anterior, se puede enviar sugerencias en línea a los clientes conectados de forma masiva o individual, hacerles sugerencias y motivar el uso de sus aplicaciones, en base a sus hábitos recordarles tareas previas realizadas por ellos en fechas o días de la semana similares, entre otras.



## VII. BIBLIOGRAFÍAS

## Bibliografías

- Benítez Flores, C. R., Granda Ayabaca, D. M., & Jaramillo Alba, J. A. (2019). LA COMPUTACIÓN EN LA NUBE EN LOS ESPACIOS EDUCATIVOS (Instituto Tecnológico Superior Jubones, Ed.). *Uso adecuado de las fuentes de información*, 2(1), 51-58. <https://doi.org/10.51247/st.v2i1.67>
- contenido, W., Romero, A., & Membreño, A. (2008). *Comparación de los métodos de valuación de inventarios en una economía con alta tasa de inflación* (UNAN-RUCFA, Ed.). UNAN-RUCFA. [https://d1wqtxts1xzle7.cloudfront.net/55263555/Comparacion\\_de\\_Metodos\\_de\\_Valuacion\\_de\\_Inventarios-with-cover-page-v2.pdf?Expires=1636671746&Signature=F4gMDUMknlbXvc3AL7jVX9Se6BG2nKlcAVcsFiqRA8owzjVXBTZV2BP7MVwPXfa59EPutXil3rWgnjZw9ZO8Y1qCS2Oc8IAI1eAPIzv3Mj](https://d1wqtxts1xzle7.cloudfront.net/55263555/Comparacion_de_Metodos_de_Valuacion_de_Inventarios-with-cover-page-v2.pdf?Expires=1636671746&Signature=F4gMDUMknlbXvc3AL7jVX9Se6BG2nKlcAVcsFiqRA8owzjVXBTZV2BP7MVwPXfa59EPutXil3rWgnjZw9ZO8Y1qCS2Oc8IAI1eAPIzv3Mj)
- Durán, Y. (2012, Ene-Jun). Administración del inventario: elemento clave para la optimización de las utilidades en las empresas (Universidad de Los Andes, Ed.). *Visión gerencial*, 11(1), 55-78. <https://www.redalyc.org/pdf/4655/465545892008.pdf>
- Figueroa, A. A. (2009). MeISE: Metodología de Ingeniería de Software Educativo. *Revista Revista Internacional Internacional Internacional de Educación en Ingeniería*, 2(1), 3-5. [https://d1wqtxts1xzle7.cloudfront.net/33528424/Abud.pdf?1398173105=&response-content-disposition=inline%3B+filename%3DP\\_a\\_g\\_i\\_n\\_a\\_MeISE\\_Metodologia\\_de\\_Ingenieria.pdf&Expires=1626320241&Signature=WCLiA6ve9-uNEq5BGFwDUPFsQ5lukWVB9-sfdfEjlg5qnP9uRI-a8s7Yh741ah92](https://d1wqtxts1xzle7.cloudfront.net/33528424/Abud.pdf?1398173105=&response-content-disposition=inline%3B+filename%3DP_a_g_i_n_a_MeISE_Metodologia_de_Ingenieria.pdf&Expires=1626320241&Signature=WCLiA6ve9-uNEq5BGFwDUPFsQ5lukWVB9-sfdfEjlg5qnP9uRI-a8s7Yh741ah92)
- Latorre Pelaez, D. G. (2017). *Implementación De Un Sistema De Inventarios Para El Área De Soporte Técnico En La Empresa Comercializadora Arturo Calle S.A.S.* Universidad

Catolica de Colombia.

<https://repository.ucatolica.edu.co/bitstream/10983/14503/1/DocumentoTrabajoDeGraduacion.pdf>

Laveriano, W. (2010). Importancia del control de inventarios en la empresa. *Actualidad Empresarial*, (198), II-1.

<http://biblioteca.esucomex.cl/RCA/Importancia%20del%20control%20de%20inventarios%20en%20la%20empresa.pdf>

Loja Guarango, J. C. (2015). *Propuesta de un Sistema de Gestión de Inventarios para la Empresa Femarpe CIA, LTDA*. Universidad Politécnica Salesiana Ecuador.

<https://dspace.ups.edu.ec/bitstream/123456789/7805/1/UPS-CT004654.pdf>

Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software* (Universidad Católica Argentina, Ed.). Universidad Católica Argentina.

<https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Nieto, E. (2013). *Diseño de aplicaciones SaaS sobre plataformas de Cloud Computing*.

Universidad Nacional De La Plata.

[http://sedici.unlp.edu.ar/bitstream/handle/10915/46834/Documento\\_completo.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/46834/Documento_completo.pdf?sequence=1&isAllowed=y)

Osorio, C. A. (2013, 06 17). Modelos para el control de inventarios en las pymes. *PANORAMA - Revista semestral especializada en Educación y E-Learning*, 2(6), 4-10.

<https://doi.org/10.15765/pnrm.v2i6.241>

Vera Yáñez, C. M. (2019). *Desarrollo e implementación de un sistema web para el control de inventario y alquiler de maquinarias de la empresa MEGARENT S.A*. Universidad Politécnica Salesiana.

<https://dspace.ups.edu.ec/bitstream/123456789/17525/1/UPS-GT002706.pdf>

Vizcaíno, A., Soto, J. P., Garcia, F., Ruíz, F., & Piattini, M. (2006). Aplicando Gestión del Conocimiento en el Proceso de Mantenimiento del Software Inteligencia Artificial (Asociación Española para la Inteligencia Artificial, Ed.). *Revista Iberoamericana de Inteligencia Artificial*, 10(31), 91- 98. <https://www.redalyc.org/pdf/925/92503110.pdf>



## **VIII. WEBGRAFÍAS**

## Webgrafías

<https://dgii.gov.do/estadisticas/Boletin%20Mipymes/Boletin-MIPYMES-2020.pdf>

<https://www.impulsapopular.com/gerencia/5-sofwarees-para-gestionar-el-inventario-de-tu-pyme/>

<https://www.ar-racking.com/co/actualidad/blog/calidad-y-seguridad-4/tipos-de-stock-e-inventarios-en-un-almacen>

<https://smsecuador.ec/reglas-de-oro-del-control-de-inventarios/>

<https://www.microfocus.com/es-es/products/loadrunner-professional/overview>

<https://azure.microsoft.com/es-es/pricing/details/container-instances/>

<https://www.bmc.com/blogs/solid-design-principles/>

<https://softwareengineering.stackexchange.com/questions/73065/what-are-dry-kiss-solid-etc-classified-as>

<https://garywoodfine.com/what-is-clean-code/>

<https://www.pluralsight.com/blog/software-development/10-steps-to-clean-code>



## **IX. ANEXOS**

# **Anexos**

## **Anexo 1**

DECANATO DE INGENIERIA E INFORMATICA

“ANTEPROYECTO DE TESIS”

**“PROPUESTA DE DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE  
INVENTARIO POR SUSCRIPCIÓN PARA PYMES BAJO LA ARQUITECTURA SAAS  
EN LA NUBE, SANTO DOMINGO, 2021”**

**ESTUDIO DE CASO**

Anteproyecto de Tesis para optar por el título de Ingeniero (a) de Software

**SUSTENTANTE**

Emerson Pichardo García 2015-0437

**ASESOR**

Juan Pablo Valdez

**ANTEPROYECTO DE TESIS**

Santo Domingo, D.N.

# Índice

<b>Introducción</b>	<b>6</b>
<b>Justificación</b>	<b>7</b>
<b>Delimitación del tema y planteamiento del problema</b>	<b>8</b>
Descripción del problema	8
Antecedentes	8
Actualidad	8
Formulación del problema	8
Sistematización del problema	8
<b>Objetivos</b>	<b>9</b>
Objetivo general	9
Objetivos específicos	9
<b>Marco teórico referencial</b>	<b>9</b>
Marco teórico	9
Marco conceptual	11
<b>Hipótesis</b>	<b>13</b>
Hipótesis de segundo grado	13
<b>Diseño metodológico: metodología y técnicas de investigación cualitativa y/o cuantitativa</b>	<b>13</b>
Tipos de Investigación	13
Explicativa	13
Descriptiva	13
Analítica	14
Técnicas de recolección de datos	14
Observación	14
Entrevista	14
Encuesta	14
<b>Fuentes de documentación (fuentes bibliográficas primordiales sobre el tema)</b>	<b>15</b>
<b>Esquema preliminar de contenido del trabajo de grado</b>	<b>17</b>

## **2 - Introducción**

Este trabajo tiene como finalidad presentar el análisis, diseño, desarrollo e implementación de un sistema de software que supla de forma efectiva la necesidad de un sistema de inventario en la nube para las PyMES dominicanas.

Este trabajo consta de la justificación y delimitación de la investigación a realizar, los objetivos tanto generales como específicos, el marco teórico y conceptual, hipótesis, diseño metodológico y un esquema preliminar para ser trabajado en una tesis de grado.

Mediante este proyecto se brindará una solución viable y efectiva a un sector muy amplio del comercio dominicano, utilizando los mejores estándares en la industria del desarrollo e implementación del software.

### **3 - Justificación**

Para el primer semestre del 2020 en la República Dominicana se encontraban registradas 1.5 millones de PyMES de las cuales el 46.7% pertenecen al sector comercio y el 14.9% al sector industrias (Ministerio de Industria, Comercio y Mipymes, 2020) por lo que 924,000 empresas, equivalentes al 61.6% de las PyMES, se encuentran involucradas en actividades comerciales que requieren de la administración de inventario de productos, y no todas estas empresas cuentan con un sistema de software apropiado para manejar adecuadamente este proceso o simplemente no cuentan con la infraestructura necesaria.

El presente proyecto brinda el análisis, diseño, desarrollo e implementación para una solución de software que permita a PyMES tener acceso a una aplicación en la nube para la administración de su inventario a través de una suscripción adaptada a sus necesidades y a sus posibilidades económicas.

# **4 - Delimitación del tema y planteamiento del problema**

## **4.1 - Descripción del problema**

### **4.1.1 - Antecedentes**

Los inventarios son una parte fundamental en el proceso de comercialización de productos, por lo que su buena administración es crucial para el desarrollo exitoso de cualquier empresa; no obstante, no todas las PyMES tienen acceso a una gestión adecuada de su inventario.

### **4.1.2 - Actualidad**

La falta de opciones de software para administración de inventario en el mercado actual de República Dominicana ha llevado a muchas PyMES a: gestionar su inventario con aplicaciones que no se encuentran diseñadas explícitamente para esta tarea, utilizar métodos manuales, o directamente manejar el proceso de forma empírica. La utilización de estos métodos no convencionales generan un descontrol en los registros de inventario, que conlleva a pérdidas como: tiempo, productos, dinero, clientes potenciales, y finalmente la falta de interés en organizar este proceso.

## **4.2 - Formulación del problema**

¿Cómo analizar, diseñar e implementar un sistema de inventario por suscripción para PyMES bajo la arquitectura SaaS en la nube?

## **4.3 - Sistematización del problema**

- ¿Cuáles son los requerimientos no funcionales y funcionales de la propuesta?
- ¿Cómo diseñar y analizar arquitectura de la solución?

- ¿Cómo diseñar, desarrollar e implementar una aplicación web de la propuesta?
- ¿Cómo diseñar, analizar, desarrollar e implementar una base de datos para la propuesta?
- ¿Cómo diseñar, analizar, desarrollar e implementar una RESTful web API para la propuesta?
- ¿Cuánto costará la implementación de la propuesta descrita?
- ¿Cuál es el costo de suscripción anual más balanceado para la propuesta en cuestión?
- ¿Cómo mejorará el proceso de inventario de una PyME con el uso de esta herramienta?

## 5 - Objetivos

### 5.1 - Objetivo general

Desarrollar e implementar un sistema de inventario por suscripción para PyMES de Santo Domingo bajo la arquitectura SaaS en la nube.

### 5.2 - Objetivos específicos

- Evaluar la factibilidad de la implementación de la propuesta
- Modelar un sistema de inventario acorde a las necesidades de las PyMES
- Desarrollar la propuesta de software y elaborar una guía de implementación de la misma
- Identificar las mejores prácticas de implementación de software en la nube

## 6 - Marco teórico referencial

### 6.1 - Marco teórico

- *David Guillermo Latorre Pelaez, Universidad Catolica de Colombia, Implementación De Un Sistema De Inventarios Para El Área De Soporte Técnico En La Empresa Comercializadora Arturo Calle S.A.S., 2017.* Este trabajo de grado presenta la

implementación de un sistema de inventarios diseñado para el área de soporte técnico de la empresa Comercializadora Arturo Calle S.A.S. El proyecto se realizó con base en los lineamientos de la metodología de diseño de software Extreme Programming y se desarrolla en un ciclo de cuatro fases (exploración, planificación, iteración y puesta en producción). La implementación del sistema se dio como solución a los problemas evidenciados en la forma como se llevaba el registro de los dispositivos tecnológicos de la empresa. Información errónea, pérdida de datos y falta de control eran algunos de los inconvenientes que afectaban los procesos en la gestión de incidencias en el área de soporte técnico. En este documento se muestran los pasos que se siguieron para desarrollar el proyecto y los entregables que se generaron en cada una de las fases de la metodología. Finalmente se darán las conclusiones que se obtuvieron al implantar el sistema de inventarios diseñado.

- ***Jessica Carolina Loja Guarango, Universidad Politécnica Salesiana Ecuador, Propuesta de un Sistema de Gestión de Inventarios para la Empresa Femarpe CIA, LTDA., 2015.*** El primer capítulo abarca la fundamentación científica describiendo de manera teórica, conceptual, y viene a constituir el sustento de la investigación. En este capítulo se abordan los temas relacionados a los inventarios, ya que es de vital importancia para la realización de este trabajo final el conocimiento de estos temas. Tiene temas como objetivos, tipos de inventarios, métodos de costeo, sistemas de inventarios y técnicas para el control del inventario. En el segundo capítulo se adentra en conocer más de las empresas, en este capítulo se describe la historia de la empresa, misión, visión, objetivos, también se realiza una matriz FODA, situación actual de la empresa, el control actual de los inventarios, se describe la estructura de la bodega. En el capítulo tres se realizan las propuestas de sistema de gestión para los inventarios, en este capítulo se da una propuesta de implementación de las 5 S japonesas con las cuales se podrá incluir orden, limpieza, higiene, estandarización y crear una cultura de autodisciplina en sus empleados, otra propuesta para administración es llevar toda la documentación necesaria y una base de datos eficiente de sus inventarios.

- **Christian Marcelo Vera Yáñez, Universidad Politécnica Salesiana sede Guayaquil Ecuador, Desarrollo e implementación de un sistema web para el control de inventario y alquiler de maquinarias de la empresa MEGARENT S.A., 2019.** En este trabajo de investigación se dio a conocer que la empresa MEGARENT S.A no contaba con un sistema que le permitiera llevar el control del proceso de alquiler de maquinarias e inventarios, en ese momento el proceso de alquiler de la empresa era tedioso tanto para el cliente como para la empresa debido a la cantidad de pedidos diarios e información que se debe manejar para el alquiler, generando inconformidad por retrasos a clientes y empleados. Dichos factores permitieron proponer un proyecto que les de la facultad de obtener una mejora significativa en sus procesos y optimización de recursos, haciendo uso de las nuevas tecnologías para el mejor manejo de la información de clientes y maquinarias. El sistema ofrece a los usuarios agilidad en los procesos. El sistema es desarrollado e implementado con PHP 7 como lenguaje de front-end, Bootstrap como framework de maquetación del sitio, entre otros. La arquitectura del aplicativo fue implementada bajo el framework Codeigniter en su versión 3 con el conocido patrón M.V.C. que brinda al desarrollador la facilidad de separar la lógica del diseño y lograr estandarización del código.

## 6.2 - Marco conceptual

- **Inventario:** “Se puede definir como el registro documental de los bienes tangibles que se tienen para la venta en el curso ordinario del negocio o para ser consumidos en la producción de bienes o servicios para su posterior comercialización” (Laveriano, 2010, II-1)
- **IaaS:** “Es el servicio más básico del modelo de servicio de cloud computing. Ofrece recursos de cómputo como CPU, memoria, almacenamiento mediante máquinas virtuales, las cuales son creadas utilizando un administrador de máquinas virtuales.” (Nieto, 2013, 10)

- **PaaS:** “Este servicio ofrece herramientas de software orientado a desarrolladores de aplicaciones. Permite abstraer al desarrollador de las particularidades de la infraestructura proveyendo una capa de más alto nivel para administrarla. Desde el punto de vista del desarrollador de aplicaciones, la plataforma funciona como un middleware entre su aplicación y la infraestructura cloud del proveedor.” (Nieto, 2013, 10)
- **SaaS:** ...“Es el servicio que permite al usuario utilizar aplicaciones que están corriendo en una infraestructura cloud. Estas aplicaciones son accesibles desde diferentes dispositivos a través de un browser. El consumidor de este servicio no administra ni tiene control sobre la infraestructura cloud subyacente, como ser la red, los servidores, sistemas operativos, almacenamiento, ni siquiera sobre las características técnicas de la aplicación.” (Nieto, 2013, 11)
- **PyMES:** Empresas de 10 a 151 empleados; y ventas de entre RD\$9,118,990.74 a RD\$230,254,516.23 (Impuestos Internos, n.d.)
- **Nube:** “La nube es una combinación de recursos tecnológicos como redes, servidores y aplicaciones en un espacio común en el que las empresas y las personas pueden arrendar una serie de servicios y almacenamiento de acuerdo a sus necesidades específicas“ (Benítez Flores et al., 2019, 52)
- **Ciclo de desarrollo de software:** El ciclo de desarrollo de software contempla las siguientes etapas: etapa conceptual, análisis y diseño inicial, plan de iteraciones, diseño computacional, desarrollo y fase de despliegue (Figuroa, 2009, 3-5)

## **7 - Hipótesis**

“Si las PyMES tuvieran acceso a una aplicación diseñada para la gestión de inventario que además sea asequible, entonces se mejoraría la comercialización de productos en Santo Domingo y la eficiencia operativa de las mismas”

### **7.1 - Hipótesis de segundo grado**

La utilización de métodos no convencionales para la administración de inventario generan un descontrol en este proceso, que conlleva a pérdidas como: tiempo, productos, dinero, clientes potenciales, y finalmente la falta de interés en organizar este proceso.

## **8 - Diseño metodológico: metodología y técnicas de investigación cualitativa y/o cuantitativa**

### **8.1 - Tipos de Investigación**

#### **8.1.1 - Explicativa**

Estudiaremos a fondo las problemáticas ya expuestas, con la finalidad de obtener un mejor entendimiento de estos y así poder tener resultados concluyentes a la hora de diseñar una solución.

#### **8.1.2 - Descriptiva**

Durante el desarrollo de este proyecto se utilizará la metodología de investigación descriptiva, mediante la cual definiremos, analizaremos y planteamos las implicaciones más relevantes sobre las problemáticas que enfrentan las PyMES al intentar administrar sus inventarios con métodos poco convencionales.

### **8.1.3 - Analítica**

Agotaremos un exhaustivo proceso de análisis para determinar, no solo la raíz de los problemas que enfrentan los usuarios, también para encontrar la forma más simple y eficaz de resolverlos a través de nuestra propuesta.

## **8.2 - Técnicas de recolección de datos**

### **8.2.1 - Observación**

Con la técnica de observación captamos el flujo de personas que interactúan en los procesos de inventariado, así como los diversos métodos que las PyMES tienen para gestionarlo.

### **8.2.2 - Entrevista**

Realizaremos entrevistas a los empleados de diversas PyMES que estén involucrados en el proceso de inventariado para conocer este proceso con exactitud.

### **8.2.3 - Encuesta**

Realizaremos encuestas a los empleados de diversas PyMES que sean parte del proceso de inventario para entender a fondo las dificultades que se presentan en el mismo.

## 9 - Fuentes de documentación (fuentes bibliográficas primordiales sobre el tema)

- Benítez Flores, C. R., Granda Ayabaca, D. M., & Jaramillo Alba, J. A. (2019). LA COMPUTACIÓN EN LA NUBE EN LOS ESPACIOS EDUCATIVOS (Instituto Tecnológico Superior Jubones, Ed.). *Uso adecuado de las fuentes de información*, 2(1), 51-58. <https://doi.org/10.51247/st.v2i1.67>
- Impuestos Internos. (n.d.). *MIPYMES*. MIPYMES. <https://dgii.gov.do/contribuyentesRegistrados/mipymes/Paginas/default.aspx>
- Latorre Pelaez, D. G. (2017). *Implementación De Un Sistema De Inventarios Para El Área De Soporte Técnico En La Empresa Comercializadora Arturo Calle S.A.S*. Universidad Católica de Colombia. <https://repository.ucatolica.edu.co/bitstream/10983/14503/1/DocumentoTrabajoDeGrado.pdf>
- Laveriano, W. (2010). Importancia del control de inventarios en la empresa. *Actualidad Empresarial*, (198), II-1. <http://biblioteca.esucomex.cl/RCA/Importancia%20del%20control%20de%20inventarios%20en%20la%20empresa.pdf>
- Loja Guarango, J. C. (2015). *Propuesta de un Sistema de Gestión de Inventarios para la Empresa Femarpe CIA, LTDA*. Universidad Politécnica Salesiana Ecuador. <https://dspace.ups.edu.ec/bitstream/123456789/7805/1/UPS-CT004654.pdf>
- Ministerio de Industria, Comercio y Mipymes. (2020, 06 17). *SITUACIÓN ECONÓMICA Y DE MERCADO DE LAS MIPYMES EN REPÚBLICA DOMINICANA POR LA CRISIS DEL COVID 19*. INFORMEPNUDMIPYMESMod17062020. Retrieved 07 11, 2021, from [https://dominicanrepublic.un.org/sites/default/files/2020-07/INFORME\\_Encuesta\\_Mipy](https://dominicanrepublic.un.org/sites/default/files/2020-07/INFORME_Encuesta_Mipy)

mes\_PNUDMICM\_17\_6\_2020-comprimido.pdf#:~:text=En%20Rep%C3%BAblica%20Dominicana%20existen%20casi,porcentaje%20al%20PIB%20de%2038.6%25.

Nieto, E. (2013). *Diseño de aplicaciones SaaS sobre plataformas de Cloud Computing*.

Universidad Nacional De La Plata.

[http://sedici.unlp.edu.ar/bitstream/handle/10915/46834/Documento\\_completo.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/46834/Documento_completo.pdf?sequence=1&isAllowed=y)

Vera Yáñez, C. M. (2019). *Desarrollo e implementación de un sistema web para el control de inventario y alquiler de maquinarias de la empresa MEGARENT S.A.* Universidad Politécnica Salesiana.

<https://dspace.ups.edu.ec/bitstream/123456789/17525/1/UPS-GT002706.pdf>

Figuroa, A. A. (2009). MeISE: Metodología de Ingeniería de Software Educativo. *Revista Revista Internacional Internacional Internacional de Educación en Ingeniería*, 2(1), 3-5.

[https://d1wqtxts1xzle7.cloudfront.net/33528424/Abud.pdf?1398173105=&response-content-disposition=inline%3B+filename%3DP\\_a\\_g\\_i\\_n\\_a\\_MeISE\\_Metodologia\\_de\\_Ingenieria.pdf&Expires=1626320241&Signature=WCLiA6ve9-uNEq5BGFwDUPFsQ5lukWVB9-shdfEjlg5qnP9uRI-a8s7Yh741ah92](https://d1wqtxts1xzle7.cloudfront.net/33528424/Abud.pdf?1398173105=&response-content-disposition=inline%3B+filename%3DP_a_g_i_n_a_MeISE_Metodologia_de_Ingenieria.pdf&Expires=1626320241&Signature=WCLiA6ve9-uNEq5BGFwDUPFsQ5lukWVB9-shdfEjlg5qnP9uRI-a8s7Yh741ah92)

# 10 - Esquema preliminar de contenido del trabajo de grado

Presentación

Dedicatoria

Agradecimientos

Tabla De Contenidos

Introducción

## 1. **CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA**

Introducción

**1.1. Formulación del problema**

**1.2. Justificación de la investigación**

**1.3. Situación problemática**

**1.4. Problema**

**1.5. Objeto**

**1.6. Campo**

**1.7. Objetivos de investigación**

1.7.1. Objetivos generales

1.7.2. Objetivos específicos

**1.8. Aportes de la investigación**

**1.9. Hipótesis**

**1.10. Tipo de investigación**

1.10.1. Descriptiva

1.10.2. Explicativa

**1.11. Métodos de investigación**

1.11.1. Deductivo

1.11.2. Síntesis

1.11.3. Análisis

1.11.4. Método de observación

**1.12. Técnicas e instrumentos**

1.12.1. Entrevistas

1.12.2. Cuestionarios

1.12.3. Consulta de expertos

**1.13. Población y muestra**

## **2. CAPÍTULO II: DISEÑO METODOLÓGICO**

Introducción

### **2.1. El software en general**

### **2.2. Desarrollo de aplicaciones móvil**

2.2.1. Historia

2.2.2. Concepto

2.2.3. Tipos de aplicaciones móvil

2.2.4. Clasificación de las aplicaciones móvil

2.2.5. Ventajas de las aplicaciones móvil

2.2.6. Tecnologías de desarrollo móvil

### **2.3. Elementos de una aplicación de inventario**

### **2.4. Metodologías de desarrollo de software**

### **2.5. Scrum**

## **3. CAPÍTULO III: ANÁLISIS DE LA SITUACIÓN ACTUAL EN EL USO DE TECNOLOGÍAS Y PROCESOS PARA REALIZAR INVENTARIOS EN REPÚBLICA DOMINICANA**

Introducción

### **3.1. Evaluación de los métodos y procesos actuales**

### **3.2. Análisis de las debilidades de los procesos y métodos**

### **3.3. Recursos tecnológicos existentes**

## **4. CAPÍTULO IV: ESTUDIO DE LA FACTIBILIDAD DE IMPLEMENTACIÓN DE UNA APLICACIÓN PARA GESTIÓN DE INVENTARIO**

Introducción

### **4.1. Estudio de la factibilidad**

4.1.1. Objetivo del estudio

4.1.2. Políticas que favorecen o limitan el desarrollo del proyecto

4.1.3. Estudio de mercado

4.1.4. Software sustitutos o similares

4.1.5. Requerimientos de personal y costos

4.1.6. Aspectos técnicos

## **4.2. Análisis de los resultados**

- 4.2.1. Encuestas
- 4.2.2. Tratamiento de la información
- 4.2.3. Presentación y análisis de los resultados
- 4.2.4. Análisis general

## **4.3. Beneficios y recomendaciones**

- 4.3.1. Beneficio

# **5. CAPÍTULO V: ANÁLISIS Y DISEÑO**

Introducción

## **5.1. Descripción del proyecto**

## **5.2. Modelo del sistema**

## **5.3. Sesión de requerimientos**

- 5.3.1. Requisitos funcionales
- 5.3.2. Requisitos no funcionales

## **5.4. Casos de uso**

- 5.4.1. Listado de los casos de uso

## **5.5. Diagramas**

- 5.5.1. Diagrama UML
- 5.5.2. Diagramas de flujo
- 5.5.3. Diagrama de clase
- 5.5.4. Diagrama de secuencia

## **5.6. Diseño de la interfaz gráfica de la aplicación**

## **5.7. Arquitectura de la aplicación**

## **5.8. Seguridad de la aplicación**

- 5.8.1. Controles para garantizar la seguridad de la información

# **6. CAPÍTULO VI: DESARROLLO E IMPLEMENTACIÓN**

Introducción

## **6.1. Metodología de desarrollo del software a utilizar**

- 6.1.1. Scrum

## **6.2. Tecnologías para utilizar**

## **6.3. Pruebas de software**

## **6.4. Implementación**

## **6.5. Manual de usuario**

## **7. CONCLUSIÓN Y RECOMENDACIONES**

Introducción

### **7.1. Conclusión**

### **7.2. Recomendaciones**

### **7.3. Anexos**

7.3.1. Anexo A: Glosario

7.3.2. Anexo B: Anteproyecto

### **7.4. Bibliografía**

## Anexo 2

### Encuesta realizada a los Propietarios De las PyMES

- 1) Tipos de productos / servicios suplidos por su empresa
  - a) Medicamentos
  - b) Productos de belleza
  - c) Productos dietéticos
  - d) Productos de Limpieza
  - e) Artículos ferreteros
  - f) Otros (Accesorios, dulces, bebidas alcohólicas...)
- 2) Indicar los días picos (los días de mayor concurrencia) dentro de su proceso de inventario
  - a) Lunes
  - b) Martes
  - c) Miércoles
  - d) Jueves
  - e) Viernes
  - f) Sábado
  - g) Domingo
- 3) ¿Con qué frecuencia realizan pedidos para reabastecerse? (Marque del 1 al 5, donde 1 = Diario, 2= 2 veces x semana, 3= Semanal, 4= quincenal y 5 = mensual)
  - a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
- 4) ¿Dentro de cuál rango de edad se encuentra usted?
  - a) 18-25 años
  - b) 26-35 años
  - c) 36-45 años
  - d) 46-61 años
  - e) 62 o más
- 5) ¿Son sus productos perecederos?
  - a) Si
  - b) No
- 6) ¿Qué tan frecuente es que una persona se vaya del establecimiento por no encontrar un producto deseado? (Marque del 1 al 5, donde 1 = Muy Inusual y 5 = Muy Frecuente)

- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
- 7) ¿En qué tanda suelen realizar con más frecuencia el conteo físico de productos (toma física)?
- a) Matutina
  - b) Vespertina
  - c) Nocturna
- 8) ¿Cuál es el tiempo promedio que tarda una persona desde que inicia la búsqueda de un producto hasta que lo localiza en el establecimiento?
- a) Menos de 1 minuto
  - b) Entre 1 y 10 minutos
  - c) Más de media hora
- 9) ¿Cuál es el % de clientes que entiendes se retiran disgustado del establecimiento (por tardanza, por expectativas distintas en cuanto al embalaje, porque no encontró su producto, porque el producto que había no era exactamente el que buscaba, etc.)?
- a) Menos de 10%
  - b) 10%
  - c) 20%
  - d) 30%
  - e) 40%
  - f) 50%
  - g) 60%
  - h) 70%
  - i) 80%
  - j) 90%
  - k) Más de 90%
- 10) ¿Ha sido objeto de robo tu empresa por empleados que sustraen artículos?
- a) Si
  - b) No
  - c) Tal vez
- 11) ¿Conoce del abastecimiento “Justo a tiempo”?
- a) Si
  - b) No
- 12) ¿Sabe de cual proveedor es un artículo que haya sido devuelto por estar defectuoso o por que se haya deteriorado antes de tiempo?
- a) Si
  - b) No

13) ¿Sabes en qué han aumentado o disminuido el costo de tus artículos comprados?

- a) Si
- b) No

14) ¿Usas algún método para costear tu inventario (UEPS, PEPS, promedio)?

- a) Si
- b) No

## Anexo 3

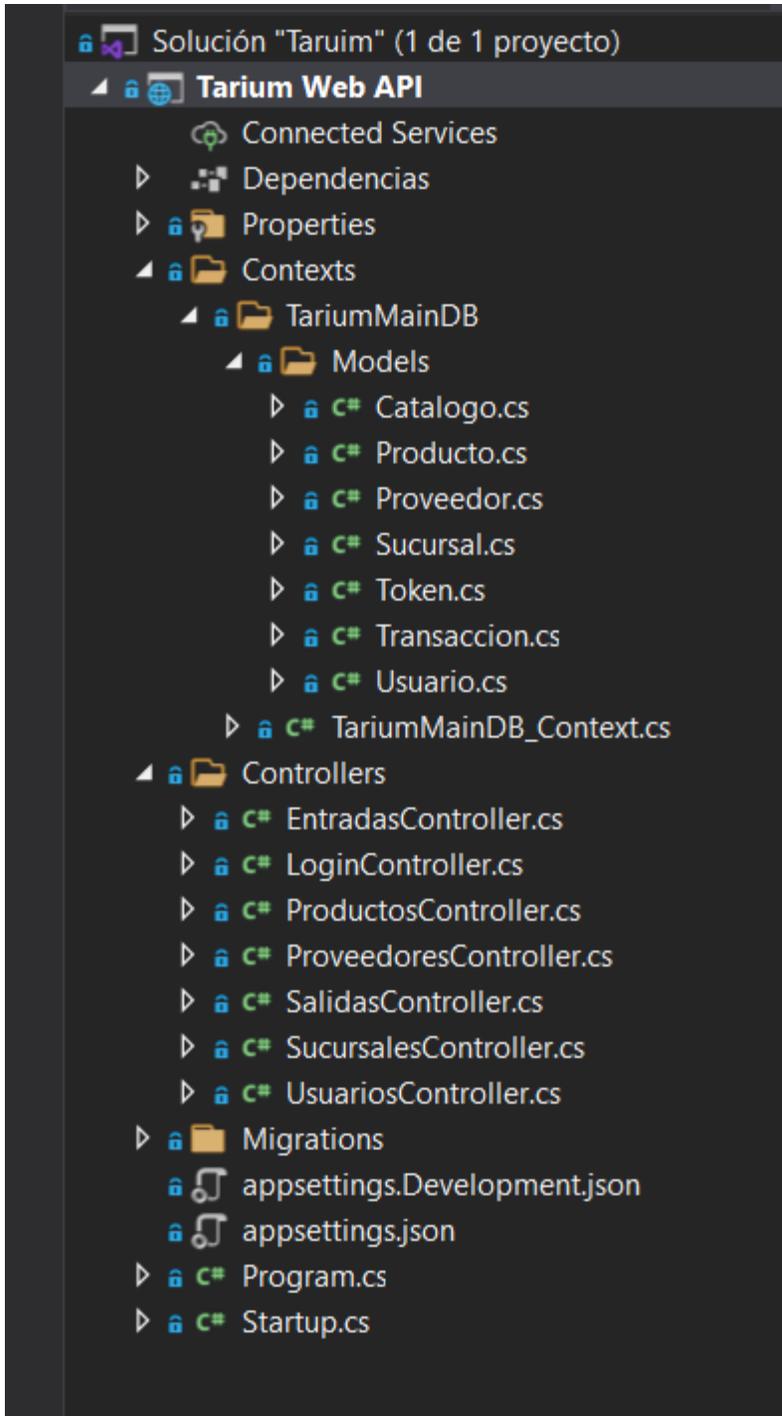
### Encuesta Realizada a Empleados Operativos de las PyMES

1. ¿Cuál es su género?
  - a. Femenino
  - b. Masculino
  - c. Prefiero no decirlo
2. ¿Dentro de cuál rango de edad se encuentra usted?
  - a. 18-25
  - b. 26-35
  - c. 36-40
  - d. 41-45
  - e. 46-61
  - f. 62 o más
3. ¿En qué municipio te ubicas?
  - a. Santo Domingo Este
  - b. Santo Domingo Oeste
  - c. Santo Domingo Norte
  - d. Boca Chica
  - e. Los Alcarrizos
  - f. Pedro Brand
  - g. San Antonio de Guerra
  - h. Distrito Nacional
4. ¿Con qué frecuencia asistes a la empresa a la que laboras en este periodo de pandemia?
  - a. Cada día
  - b. Interdiario
  - c. Cada semana
  - d. Cada mes
  - e. Otra
5. ¿Dónde está ubicada la empresa para la cual laboras?
  - a. Santo Domingo Este
  - b. Santo Domingo Oeste

- c. Santo Domingo Norte
  - d. Boca Chica
  - e. Los Alcarrizos
  - f. Pedro Brand
  - g. San Antonio de Guerra
  - h. Distrito Nacional
6. ¿Cuánto dinero consume en transporte, llamadas al mantener contacto con otros empleados u empresas mientras gestiona el inventario? (Los montos están representados en pesos dominicanos)
- a. No gasto dinero en transporte / llamadas
  - b. 0-30
  - c. 31-50
  - d. 51-100
  - e. 101-150
  - f. 151-200
  - g. 201-250
  - h. 251-300
  - i. 300-500
  - j. 500 o más
7. ¿Cuál de las siguientes opciones es la principal razón por la cual usted entiende prudente que la empresa adquiriera una aplicación para gestionar el inventario?
- a. Mejorar la eficiencia operativa
  - b. Disminución de costos
  - c. Poder realizar la gestión desde el hogar
  - d. Lentitud actual del proceso
  - e. Mucho uso de papel
  - f. Otra

## Anexo 4

### Código fuente de la Web API



```
Tarium Web API
└─ Tarium_Web_API.Contexts.TariumMainDB.TariumMainDB_Context
  1  using Microsoft.EntityFrameworkCore;
  2  using Tarium_Web_API.Contexts.TariumMainDB.Models;
  3
  4  namespace Tarium_Web_API.Contexts.TariumMainDB
  5  {
  6      31 referencias
  7      public class TariumMainDB_Context : DbContext
  8      {
  9          6 referencias
 10         public DbSet<Sucursal> Sucursales { get; set; }
 11         5 referencias
 12         public DbSet<Usuario> Usuarios { get; set; }
 13         1 referencia
 14         public DbSet<Token> Tokens { get; set; }
 15         6 referencias
 16         public DbSet<Proveedor> Proveedores { get; set; }
 17         6 referencias
 18         public DbSet<Producto> Productos { get; set; }
 19         10 referencias
 20         public DbSet<Catalogo> Catalogos { get; set; }
 21         14 referencias
 22         public DbSet<Transaccion> Transacciones { get; set; }
 23
 24         0 referencias
 25         public TariumMainDB_Context(DbContextOptions<TariumMainDB_Context> options) : base(options) { }
 26
 27         0 referencias
 28         protected override void OnModelCreating(ModelBuilder modelBuilder)
 29         {
 30             // Use Fluent API to configure
 31
 32             // Map entities to tables
 33             modelBuilder.Entity<Sucursal>().ToTable("Sucursales");
 34             modelBuilder.Entity<Usuario>().ToTable("Usuarios");
 35             modelBuilder.Entity<Token>().ToTable("Tokens");
 36             modelBuilder.Entity<Proveedor>().ToTable("Proveedores");
 37             modelBuilder.Entity<Producto>().ToTable("Productos");
 38             modelBuilder.Entity<Catalogo>().ToTable("Catalogos");
 39             modelBuilder.Entity<Transaccion>().ToTable("Transacciones");
 40         }
 41     }
 42 }
```

```

29 modelBuilder.Entity<Transaccion>().ToTable("Transacciones");
30
31 // Configure Primary Keys
32 modelBuilder.Entity<Sucursal>().HasKey(sucursal => sucursal.Id).HasName("PK_Sucursales");
33 modelBuilder.Entity<Usuario>().HasKey(usuario => usuario.Id).HasName("PK_Usuarios");
34 modelBuilder.Entity<Token>().HasKey(token => token.Hash).HasName("PK_Tokens");
35 modelBuilder.Entity<Proveedor>().HasKey(proveedor => proveedor.Id).HasName("PK_Proveedores");
36 modelBuilder.Entity<Producto>().HasKey(producto => producto.Id).HasName("PK_Productos");
37 modelBuilder.Entity<Catalogo>().HasKey(catalogo => catalogo.Id).HasName("PK_Catalogos");
38 modelBuilder.Entity<Transaccion>().HasKey(transaccion => transaccion.Id).HasName("PK_Transacciones");
39
40 // Configure indexes
41 modelBuilder.Entity<Sucursal>().HasIndex(sucursal => sucursal.Nombre).IsUnique().HasDatabaseName("UNQ_Sucursales_Nombre");
42 modelBuilder.Entity<Usuario>().HasIndex(usuario => usuario.Nombre).HasDatabaseName("IND_Usuarios_Nombre");
43 modelBuilder.Entity<Usuario>().HasIndex(usuario => usuario.Correo).IsUnique().HasDatabaseName("UNQ_Usuarios_Correo");
44 modelBuilder.Entity<Producto>().HasIndex(producto => producto.SKU).IsUnique().HasDatabaseName("UNQ_Productos_SKU");
45 modelBuilder.Entity<Producto>().HasIndex(producto => producto.CodigoBarra).HasDatabaseName("IND_Productos_CodigoBarra");
46
47 // Configure columns
48 modelBuilder.Entity<Sucursal>().Property(sucursal => sucursal.Id).HasColumnType("int").UseMySQLIdentityColumn().IsRequired();
49 modelBuilder.Entity<Sucursal>().Property(sucursal => sucursal.Nombre).HasColumnType("nvarchar(500)").IsRequired();
50 modelBuilder.Entity<Sucursal>().Property(sucursal => sucursal.Tipo).HasColumnType("nvarchar(50)").IsRequired();
51 modelBuilder.Entity<Sucursal>().Property(sucursal => sucursal.Estado).HasColumnType("nvarchar(50)").IsRequired();
52
53 modelBuilder.Entity<Usuario>().Property(usuario => usuario.Id).HasColumnType("int").UseMySQLIdentityColumn().IsRequired();
54 modelBuilder.Entity<Usuario>().Property(usuario => usuario.Nombre).HasColumnType("nvarchar(100)").IsRequired();
55 modelBuilder.Entity<Usuario>().Property(usuario => usuario.Correo).HasColumnType("nvarchar(100)").IsRequired();
56 modelBuilder.Entity<Usuario>().Property(usuario => usuario.Clave).HasColumnType("nvarchar(500)").IsRequired();
57 modelBuilder.Entity<Usuario>().Property(usuario => usuario.Rol).HasColumnType("int").IsRequired();
58
59 modelBuilder.Entity<Token>().Property(token => token.Hash).HasColumnType("nvarchar(500)").IsRequired();
60 modelBuilder.Entity<Token>().Property(token => token.Id_Usuario).HasColumnType("int").IsRequired();
61
62 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Id).HasColumnType("int").UseMySQLIdentityColumn().IsRequired();
63 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Nombre).HasColumnType("nvarchar(100)").IsRequired();
64 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Contacto).HasColumnType("nvarchar(200)").IsRequired();
65 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Estado).HasColumnType("nvarchar(50)").IsRequired();
66 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Comentario).HasColumnType("nvarchar(1000)");

```

```

65 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Estado).HasColumnType("nvarchar(50)").IsRequired();
66 modelBuilder.Entity<Proveedor>().Property(proveedor => proveedor.Comentario).HasColumnType("nvarchar(1000)");
67
68 modelBuilder.Entity<Producto>().Property(producto => producto.Id).HasColumnType("int").UseMySQLIdentityColumn().IsRequired();
69 modelBuilder.Entity<Producto>().Property(producto => producto.SKU).HasColumnType("nvarchar(100)").IsRequired();
70 modelBuilder.Entity<Producto>().Property(producto => producto.CodigoBarra).HasColumnType("nvarchar(100)");
71 modelBuilder.Entity<Producto>().Property(producto => producto.Nombre).HasColumnType("nvarchar(200)").IsRequired();
72 modelBuilder.Entity<Producto>().Property(producto => producto.FechaVencimiento).HasColumnType("datetime");
73 modelBuilder.Entity<Producto>().Property(producto => producto.Id_Proveedor).HasColumnType("int");
74 modelBuilder.Entity<Producto>().Property(producto => producto.Comentario).HasColumnType("nvarchar(1000)");
75 modelBuilder.Entity<Producto>().Property(producto => producto.Estado).HasColumnType("nvarchar(50)").IsRequired();
76
77 modelBuilder.Entity<Catalogo>().Property(catalogo => catalogo.Id).HasColumnType("int").UseMySQLIdentityColumn().IsRequired();
78 modelBuilder.Entity<Catalogo>().Property(catalogo => catalogo.Id_Sucursal).HasColumnType("int").IsRequired();
79 modelBuilder.Entity<Catalogo>().Property(catalogo => catalogo.Id_Producto).HasColumnType("int").IsRequired();
80 modelBuilder.Entity<Catalogo>().Property(catalogo => catalogo.Cantidad).HasColumnType("int").IsRequired();
81
82 modelBuilder.Entity<Transaccion>().Property(transaccion => transaccion.Id).HasColumnType("int").UseMySQLIdentityColumn().IsRequired();
83 modelBuilder.Entity<Transaccion>().Property(transaccion => transaccion.Tipo).HasColumnType("nvarchar(50)").IsRequired();
84 modelBuilder.Entity<Transaccion>().Property(transaccion => transaccion.Id_Producto).HasColumnType("int").IsRequired();
85 modelBuilder.Entity<Transaccion>().Property(transaccion => transaccion.Cantidad).HasColumnType("int").IsRequired();
86 modelBuilder.Entity<Transaccion>().Property(transaccion => transaccion.Costo).HasColumnType("decimal").IsRequired();
87
88 // Configure relationships
89 modelBuilder.Entity<Token>()
90     .HasOne<Usuario>()
91     .WithMany()
92     .HasPrincipalKey(usuario => usuario.Id)
93     .HasForeignKey(token => token.Id_Usuario)
94     .OnDelete(DeleteBehavior.NoAction)
95     .HasConstraintName("FK_Tokens_Usuarios");
96 modelBuilder.Entity<Producto>()
97     .HasOne<Proveedor>()
98     .WithMany()
99     .HasPrincipalKey(proveedor => proveedor.Id)
100    .HasForeignKey(producto => producto.Id_Proveedor)
101    .OnDelete(DeleteBehavior.NoAction)

```

```
Tarium Web API Tarium_Web_API.Contexts.TariumMainDB.TariumMainDB_Context
101     .onDelete(DeleteBehavior.NoAction)
102     .HasConstraintName("FK_Productos_Proveedores");
103     modelBuilder.Entity<Catalogo>()
104     .HasOne(catalogo => catalogo.Producto)
105     .WithMany()
106     .HasPrincipalKey(producto => producto.Id)
107     .HasForeignKey(catalogo => catalogo.Id_Producto)
108     .onDelete(DeleteBehavior.NoAction)
109     .HasConstraintName("FK_Catalogos_Productos");
110     modelBuilder.Entity<Sucursal>()
111     .HasMany(sucursal => sucursal.Catalogos)
112     .WithOne()
113     .HasPrincipalKey(sucursal => sucursal.Id)
114     .HasForeignKey(catalogo => catalogo.Id_Sucursal)
115     .onDelete(DeleteBehavior.Cascade)
116     .HasConstraintName("FK_Catalogos_Sucursales");
117     modelBuilder.Entity<Transaccion>()
118     .HasOne(transaccion => transaccion.Sucursal)
119     .WithMany()
120     .HasPrincipalKey(sucursal => sucursal.Id)
121     .HasForeignKey(transaccion => transaccion.Id_Sucursal)
122     .onDelete(DeleteBehavior.NoAction)
123     .HasConstraintName("FK_Transacciones_Sucursal");
124     modelBuilder.Entity<Transaccion>()
125     .HasOne(transaccion => transaccion.Producto)
126     .WithMany()
127     .HasPrincipalKey(producto => producto.Id)
128     .HasForeignKey(transaccion => transaccion.Id_Producto)
129     .onDelete(DeleteBehavior.NoAction)
130     .HasConstraintName("FK_Transacciones_Productos");
131     }
132 }
133 }
```

```
Catalogo.cs  + X
Tarium Web API  Tarium_Web_API.Contexts.TariumMainDB.M
1 namespace Tarium_Web_API.Contexts.TariumMainDB.M
2 {
3     10 referencias
4     public class Catalogo
5     {
6         2 referencias
7         public int Id { get; set; }
8
9         10 referencias
10        public int Id_Sucursal { get; set; }
11
12        8 referencias
13        public int Id_Producto { get; set; }
14
15        7 referencias
16        public int Cantidad { get; set; }
17
18        3 referencias
19        public Producto Producto { get; set; }
20    }
21 }
```

```
Producto.cs  [X]
Tarium Web API  Tarium_Web_API.Contexts.TariumMainDB.Models.Pr
1  using System;
2
3  namespace Tarium_Web_API.Contexts.TariumMainDB.Models
4  {
5      public class Producto
6      {
7          public int Id { get; set; }
8
9          public string SKU { get; set; }
10
11         public string CodigoBarra { get; set; }
12
13         public string Nombre { get; set; }
14
15         public DateTime? FechaVencimiento { get; set; }
16
17         public int? Id_Proveedor { get; set; }
18
19         2 referencias
20         public Proveedor Proveedor { get; set; }
21
22         1 referencia
23         public string Comentario { get; set; }
24
25         1 referencia
26         public string Estado { get; set; }
27     }
28 }
```

```
Proveedor.cs  [X]
Tarium Web API  Tarium_Web_API.Contexts.TariumMainDB.
1 namespace Tarium_Web_API.Contexts.TariumMainDB.Mod
2 {
3     public class Proveedor
4     {
5         public int Id { get; set; }
6
7         public string Nombre { get; set; }
8
9         public string Contacto { get; set; }
10
11        public string Estado { get; set; }
12
13        public string Comentario { get; set; }
14    }
15 }
16
```

```
Sucursal.cs  # X
Tarium Web API  Tarium_Web_API.Contexts.TariumMainDB.Models.Sucursal
1  using System.Collections.Generic;
2
3  namespace Tarium_Web_API.Contexts.TariumMainDB.Models
4  {
5      15 referencias
6      public class Sucursal
7      {
8          10 referencias
9          public int Id { get; set; }
10
11         2 referencias
12         public string Nombre { get; set; }
13
14         1 referencia
15         public string Tipo { get; set; }
16
17         1 referencia
18         public string Estado { get; set; }
19
20         6 referencias
21         public ICollection<Catalogo> Catalogos { get; set; }
22     }
23 }
```

```
Token.cs  [X]
Tarium Web API  Tarium_Web_API.Contexts.TariumMainDB.I
1 namespace Tarium_Web_API.Contexts.TariumMainDB.I
2 {
3     public class Token
4     {
5         public string Hash { get; set; }
6
7         public int Id_Usuario { get; set; }
8     }
9 }
10
```

```
Transaccion.cs  + X
Tarium Web API  Tarium_Web_API.Contexts.TariumMainDB.
1 namespace Tarium_Web_API.Contexts.TariumMainDB.Mod
2 {
3     20 referencias
4     public class Transaccion
5     {
6         12 referencias
7         public int Id { get; set; }
8         5 referencias
9         public string Tipo { get; set; }
10        7 referencias
11        public int Id_Sucursal { get; set; }
12        8 referencias
13        public int Id_Producto { get; set; }
14        9 referencias
15        public int Cantidad { get; set; }
16        1 referencia
17        public decimal Costo { get; set; }
18
19        5 referencias
20        public Producto Producto { get; set; }
21        5 referencias
22        public Sucursal Sucursal { get; set; }
23    }
24 }
```

```
Usuario.cs [X]
Tarium Web API Tarium_Web_API.Contexts.TariumMainDB
1 namespace Tarium_Web_API.Contexts.TariumMainDB
2 {
3     18 referencias
4     public class Usuario
5     {
6         7 referencias
7         public int Id { get; set; }
8
9         3 referencias
10        public string Nombre { get; set; }
11
12        4 referencias
13        public string Correo { get; set; }
14
15        3 referencias
16        public string Clave { get; set; }
17
18        2 referencias
19        public int Rol { get; set; }
20    }
21 }
```

```
EntradasController.cs  X
Tarium Web API  Tarium_Web_API.Controllers.EntradasController
9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class EntradasController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public EntradasController(TariumMainDB_Context context)
18         {
19             _context = context;
20         }
21
22         [HttpGet]
23         public async Task<ActionResult<IEnumerable<Transaccion>>> GetTransacciones()
24         {
25             return Ok(_context.Transacciones.ToList());
26         }
27
28         [HttpGet("{id}")]
29         public async Task<ActionResult<Transaccion>> GetTransaccion(int id)
30         {
31             return Ok(_context.Transacciones.FirstOrDefault(x => x.Id == id));
32         }
33
34         [HttpPut("{id}")]
35         public async Task<IActionResult> PutTransaccion(int id, Transaccion transaccion)
36         {
37             return Ok(_context.Transacciones.Update(transaccion));
38         }
39
40         [HttpPost]
41         public async Task<ActionResult<Transaccion>> PostTransaccion(Transaccion transaccion)
42         {
43             return Ok(_context.Transacciones.Add(transaccion));
44         }
45
46         [HttpDelete("{id}")]
47         public async Task<IActionResult> DeleteTransaccion(int id)
48         {
49             return Ok(_context.Transacciones.Remove(id));
50         }
51
52         private bool TransaccionExists(int id)
53         {
54             return _context.Transacciones.Any(x => x.Id == id);
55         }
56     }
57 }
```

```
LoginController.cs  X
Tarium Web API  Tarium_Web_API.Controllers.LoginController
1  using Microsoft.AspNetCore.Mvc;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Threading.Tasks;
5  using Tarium_Web_API.Contexts.TariumMainDB;
6  using Tarium_Web_API.Contexts.TariumMainDB.Models;
7
8  namespace Tarium_Web_API.Controllers
9  {
10     [Route("api/[controller]")]
11     [ApiController]
12     public class LoginController : ControllerBase
13     {
14         private readonly TariumMainDB_Context _context;
15
16         public LoginController(TariumMainDB_Context context) {...}
17
18         [HttpPost]
19         public async Task<ActionResult<UsuarioDTO>> PostToken(Usuario usuario) {...}
20
21         public class UsuarioDTO
22         {
23             public string Hash { get; set; }
24             public string Nombre { get; set; }
25             public int Rol { get; set; }
26         }
27     }
28 }
```

```
ProductosController.cs
Tarium Web API
Tarium_Web_API.Controllers.ProductosController

9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class ProductosController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public ProductosController(TariumMainDB_Context context)
18         {
19             _context = context;
20         }
21
22         [HttpGet]
23         public async Task<ActionResult<IEnumerable<Producto>>> GetProductos()
24         {
25             return Ok(_context.Productos.ToList());
26         }
27
28         [HttpGet("{id}")]
29         public async Task<ActionResult<Producto>> GetProducto(int id)
30         {
31             return Ok(_context.Productos.FirstOrDefault(p => p.Id == id));
32         }
33
34         [HttpPut("{id}")]
35         public async Task<IActionResult> PutProducto(int id, Producto producto)
36         {
37             _context.Productos.Update(producto);
38             await _context.SaveChangesAsync();
39             return Ok(producto);
40         }
41
42         [HttpPost]
43         public async Task<ActionResult<Producto>> PostProducto(Producto producto)
44         {
45             _context.Productos.Add(producto);
46             await _context.SaveChangesAsync();
47             return Ok(producto);
48         }
49
50         [HttpDelete("{id}")]
51         public async Task<IActionResult> DeleteProducto(int id)
52         {
53             var producto = _context.Productos.FirstOrDefault(p => p.Id == id);
54             if (producto != null)
55             {
56                 _context.Productos.Remove(producto);
57                 await _context.SaveChangesAsync();
58             }
59             return Ok();
60         }
61
62         private bool ProductoExists(int id)
63         {
64             return _context.Productos.Any(p => p.Id == id);
65         }
66     }
67 }
```

```
ProveedoresController.cs  + X
Tarium Web API  Tarium_Web_API.Controllers.ProveedoresController
9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class ProveedoresController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public ProveedoresController(TariumMainDB_Context context) ...
18
19         [HttpGet]
20         public async Task<ActionResult<IEnumerable<Proveedor>>> GetProveedores() ...
21
22         [HttpGet("{id}")]
23         public async Task<ActionResult<Proveedor>> GetProveedor(int id) ...
24
25         [HttpPut("{id}")]
26         public async Task<IActionResult> PutProveedor(int id, Proveedor proveedor) ...
27
28         [HttpPost]
29         public async Task<ActionResult<Proveedor>> PostProveedor(Proveedor proveedor) ...
30
31         [HttpDelete("{id}")]
32         public async Task<IActionResult> DeleteProveedor(int id) ...
33
34         private bool ProveedorExists(int id) ...
35     }
36 }
```

```
SalidasController.cs + X
Tarium Web API Tarium_Web_API.Controllers.SalidasController
9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class SalidasController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public SalidasController(TariumMainDB_Context context)
18         {
19             _context = context;
20         }
21
22         [HttpGet]
23         public async Task<ActionResult<IEnumerable<Transaccion>>> GetTransacciones()
24         {
25             return Ok(_context.Transacciones.ToList());
26         }
27
28         [HttpGet("{id}")]
29         public async Task<ActionResult<Transaccion>> GetTransaccion(int id)
30         {
31             return Ok(_context.Transacciones.FirstOrDefault(x => x.Id == id));
32         }
33
34         [HttpPut("{id}")]
35         public async Task<IActionResult> PutTransaccion(int id, Transaccion transaccion)
36         {
37             if (transaccion.Id != id)
38                 return BadRequest();
39
40             _context.Transacciones.Update(transaccion);
41             await _context.SaveChangesAsync();
42             return Ok(transaccion);
43         }
44
45         [HttpPost]
46         public async Task<ActionResult<Transaccion>> PostTransaccion(Transaccion transaccion)
47         {
48             if (transaccion == null)
49                 return BadRequest();
50
51             _context.Transacciones.Add(transaccion);
52             await _context.SaveChangesAsync();
53             return Ok(transaccion);
54         }
55
56         [HttpDelete("{id}")]
57         public async Task<IActionResult> DeleteTransaccion(int id)
58         {
59             var transaccion = _context.Transacciones.FirstOrDefault(x => x.Id == id);
60             if (transaccion == null)
61                 return BadRequest();
62
63             _context.Transacciones.Remove(transaccion);
64             await _context.SaveChangesAsync();
65             return Ok();
66         }
67
68         private bool TransaccionExists(int id)
69         {
70             return _context.Transacciones.Any(x => x.Id == id);
71         }
72     }
73 }
```

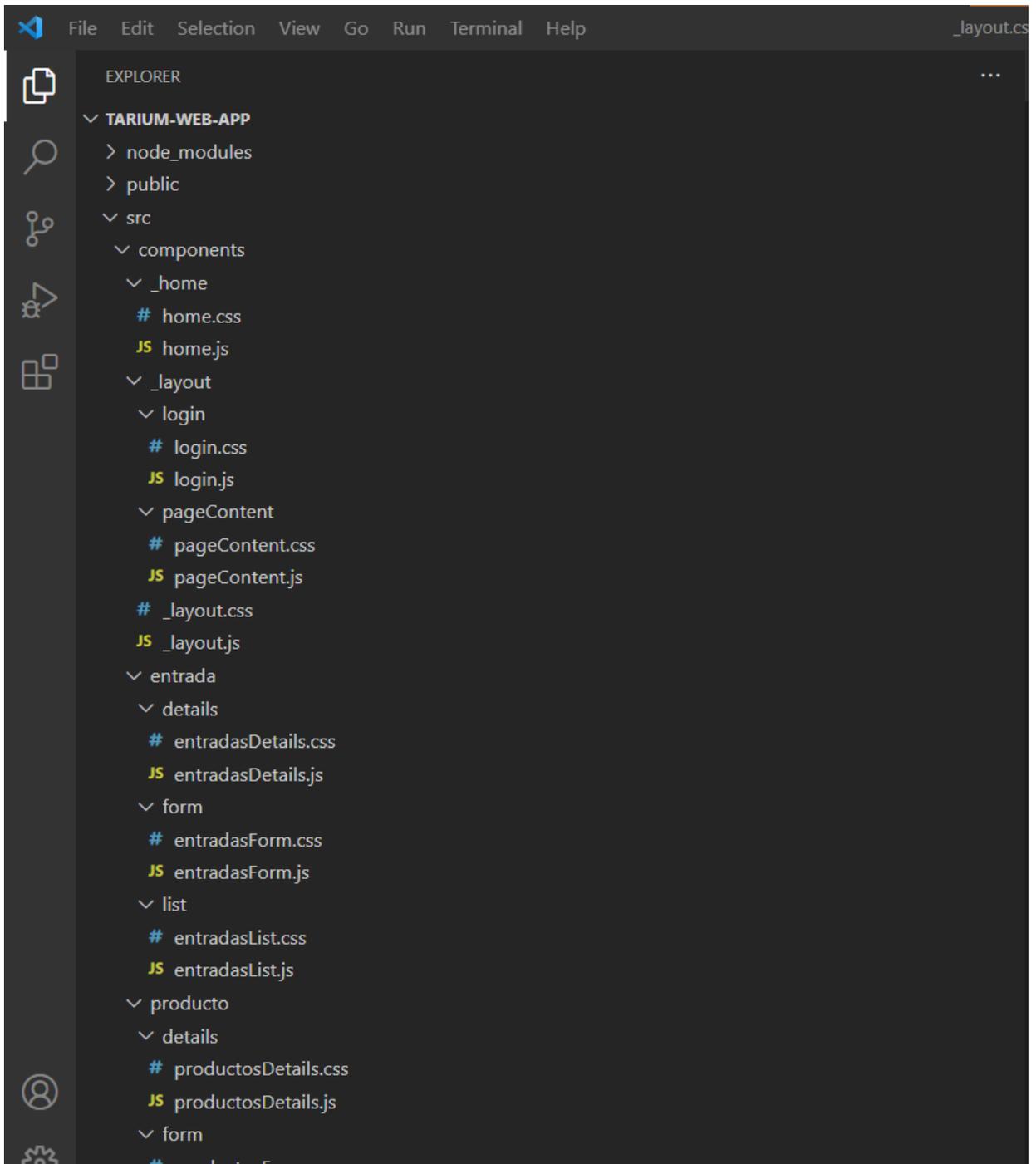
```
SucursalesController.cs [X]
Tarium Web API Tarium_Web_API.Controllers.SucursalesController
9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class SucursalesController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public SucursalesController(TariumMainDB_Context context)
18         {
19             _context = context;
20         }
21
22         [HttpGet]
23         public async Task<ActionResult<IEnumerable<Sucursal>>> GetSucursales()
24         {
25             return Ok(_context.Sucursales.ToList());
26         }
27
28         [HttpGet("{id}")]
29         public async Task<ActionResult<Sucursal>> GetSucursal(int id)
30         {
31             return Ok(_context.Sucursales.FirstOrDefault(s => s.Id == id));
32         }
33
34         [HttpPut("{id}")]
35         public async Task<ActionResult> PutSucursal(int id, Sucursal sucursal)
36         {
37             return Ok(_context.Sucursales.Update(sucursal));
38         }
39
40         [HttpPost]
41         public async Task<ActionResult<Sucursal>> PostSucursal(Sucursal sucursal)
42         {
43             return Ok(_context.Sucursales.Add(sucursal));
44         }
45
46         [HttpDelete("{id}")]
47         public async Task<ActionResult> DeleteSucursal(int id)
48         {
49             return Ok(_context.Sucursales.Remove(id));
50         }
51
52         private bool SucursalExists(int id)
53         {
54             return _context.Sucursales.Any(s => s.Id == id);
55         }
56     }
57 }
```

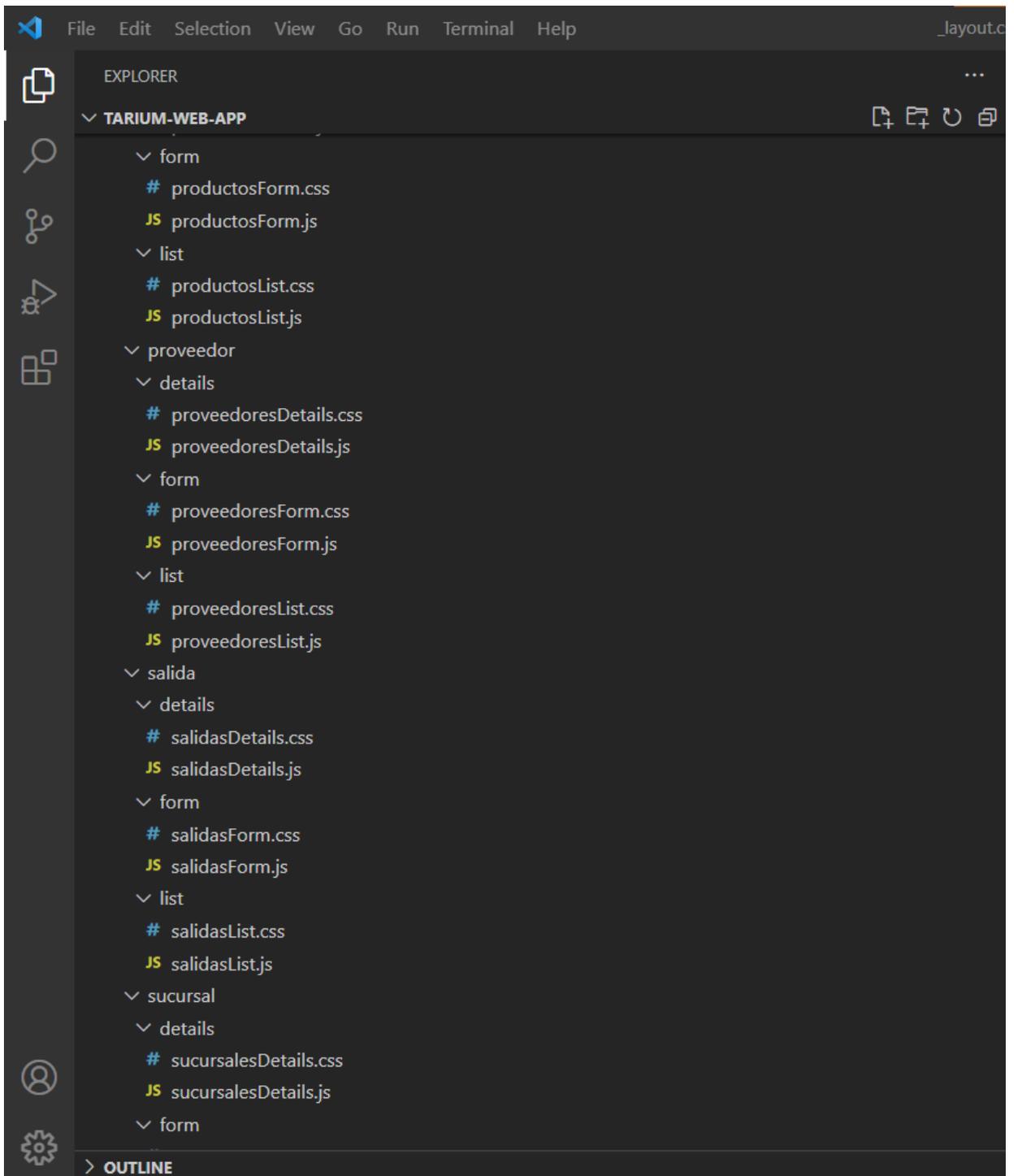
```
UsuariosController.cs  X
Tarium Web API  Tarium_Web_API.Controllers.UsuariosController

9 namespace Tarium_Web_API.Controllers
10 {
11     [Route("api/[controller]")]
12     [ApiController]
13     public class UsuariosController : ControllerBase
14     {
15         private readonly TariumMainDB_Context _context;
16
17         public UsuariosController(TariumMainDB_Context context) ...
18
19         [HttpGet]
20         public async Task<ActionResult<IEnumerable<Usuario>>> GetUsuario() ...
21
22         [HttpGet("{id}")]
23         public async Task<ActionResult<Usuario>> GetUsuario(int id) ...
24
25         [HttpPut("{id}")]
26         public async Task<IActionResult> PutUsuario(int id, Usuario usuario) ...
27
28         [HttpPost]
29         public async Task<ActionResult<Usuario>> PostUsuario(Usuario usuario) ...
30
31         private bool UsuarioExists(int id) ...
32     }
33 }
```

## Anexo 5

### Código fuente de la aplicación Web SPA





```

  ▾ form
    # sucursalesForm.css
    JS sucursalesForm.js
  ▾ list
    # sucursalesList.css
    JS sucursalesList.js
  # _layout.css
  JS index.js
  .gitignore
  {} package-lock.json
  {} package.json
  ⓘ README.md
  ⚡ README.old.md

```

**> OUTLINE**

File Edit Selection View Go Run Terminal Help

EXPLORER

- ▾ **TARIUM-WEB-APP**
  - > node\_modules
  - > public
  - ▾ src
    - ▾ components
    - > \_home
    - > \_layout
    - > entrada
    - > producto
    - > proveedor
    - > salida
    - > sucursal
    - # \_layout.css
    - JS index.js
    - .gitignore
    - {} package-lock.json
    - {} package.json
    - ⓘ README.md
    - ⚡ README.old.md

```

10 components > .layout > login > loginjs > Login > useEffect callback
11 import React, { useState, useEffect } from 'react';
12 import { useParams, useNavigate } from 'react-router-dom';
13 import { Typography, Form, Input, Divider, Button, message } from 'antd';
14 import './login.css';
15
16 const [ Title ] = Typography;
17 const [ Item ] = Form;
18 const [ Password ] = Input;
19
20 function Login(props) {
21   const [ id ] = useParams();
22   let navigate = useNavigate();
23   const [form] = Form.useForm();
24
25   let [loading, setloading] = useState(false);
26
27   useEffect(() => {
28     localStorage.removeItem('usuario');
29   }, []);
30
31   const onFinish = (values) => {
32     setloading(true);
33
34     const config = { method: 'POST', headers: { 'Content-Type': 'application/json' }, body: JSON.stringify(values) };
35
36     fetch('https://localhost:44386/api/login', config)
37       .then(response => {
38         if (response.ok) {
39           response.json().then(usuario => {
40             localStorage.setItem('usuario', JSON.stringify(usuario));
41             navigate('/');
42           });
43         } else {
44           response.json().then(error => {
45             if (error.status === 404) {
46               message.error('Credenciales incorrectas');
47             } else {
48               message.error(`${error.status}: ${error.title}`);
49             }
50             setloading(false);
51           });
52         }
53       })
54       .catch(error => {
55         message.error(error.message);
56         setloading(false);
57       });
58   });
59
60   return (
61     <div className="login-container">
62       <div className="login-form">
63         <title level={3} style={{ marginBottom: '24px', textAlign: 'center' }}>Inicio de sesión</title>
64
65         <Form
66           form={form}
67           layout="vertical"
68           onFinish={onFinish}
69           autoComplete="off"
70         >
71           <Item label="Correo" name="correo"
72             rules={[
73               {
74                 required: true,
75                 message: 'Este campo es requerido'
76               }
77             ]}
78           >
79             <input disabled={loading} />
80           </Item>
81
82           <Item label="Contraseña" name="clave"
83             rules={[
84               {
85                 required: true,
86                 message: 'Este campo es requerido'
87               }
88             ]}
89           >
90             <password disabled={loading} />
91           </Item>
92
93           <Divider />
94
95           <Item style={{ textAlign: 'center' }}>
96             <React.StrictMode>
97               <button type="primary" size="large" htmlType="submit" loading={loading}>
98                 Iniciar sesión
99               </button>
100             </React.StrictMode>
99           </Item>
101         </Form>
102       </div>
103     </div>
104   );
105 }
106
107 export default Login;

```

JS pageContent.js X

src > components > \_layout > pageContent > JS pageContent.js > ...

```
1 import React from 'react';
2 import { useNavigate } from "react-router-dom";
3 import { PageHeader, Layout } from 'antd';
4 import './pageContent.css'
5
6 const { Content } = Layout;
7
8 function PageContent(props) {
9   const { title, subtitle, extra } = props;
10  const navigate = useNavigate();
11
12  return (<>
13    <PageHeader
14      className="site-layout-pageheader"
15      ghost={false}
16      onBack={() => navigate(-1)}
17      title={title}
18      subTitle={subtitle}
19      extra={extra}
20    />
21
22    <Content className="site-layout-background">{props.children}</Content>
23  </>)
24 }
25
26 export default PageContent;
```

```

JS _layout.js X
src > components > _layout > JS _layout.js > _Layout > [0] transactionalMenu
1  import React, { useState } from 'react';
2  import { Link, useNavigate } from "react-router-dom";
3  import { Layout, Menu, Typography, Row, Col, Space, Divider, Badge, Dropdown } from 'antd';
4  import {
5      MenuUnfoldOutlined,
6      MenuFoldOutlined,
7      BellOutlined,
8      UserOutlined,
9      PoweroffOutlined,
10     HomeOutlined,
11     ShopOutlined,
12     ShoppingOutlined,
13     TagsOutlined,
14     TeamOutlined,
15     OrderedListOutlined,
16     PullRequestOutlined,
17     DownloadOutlined,
18     UploadOutlined
19 } from '@ant-design/icons';
20 import './_layout.css';
21
22 const { Header, Sider } = Layout;
23 const { Text } = Typography;
24
25 function _Layout(props) {
26     let [collapsed, setCollapsed] = useState(false);
27     let navigate = useNavigate();
28
29     function toggle() {
30         setCollapsed(!collapsed);
31     }
32
33     const menu = (
34         <Menu>
35             <Menu.Item key="0">
36                 <Space size="large">
37                     <Text type="danger"><PoweroffOutlined /></Text>
38                     <a href="/login">
39                         <Text type="danger">Cerrar sesión</Text>
40                     </a>
41                 </Space>
42             </Menu.Item>
43         </Menu>
44     );
45
46     const adminMenu = (
47         <Menu theme="dark" mode="inline" defaultSelectedKeys={['1']}>
48             <Menu.Item key="1" icon={ <HomeOutlined /> }>
49                 <Link to="/">inicio</Link>
50             </Menu.Item>
51             <Menu.Item key="2" icon={ <ShopOutlined /> }>
52                 <Link to="/sucursales">Sucursales</Link>
53             </Menu.Item>
54             <Menu.Item key="3" icon={ <ShoppingOutlined /> }>
55                 <Link to="/proveedores">Proveedores</Link>
56             </Menu.Item>
57             <Menu.Item key="4" icon={ <TagsOutlined /> }>
58                 <Link to="/productos">Productos</Link>
59             </Menu.Item>
60             <Menu.Item key="5" icon={ <TeamOutlined /> }>
61                 <Link to="/usuarios">Usuarios</Link>
62             </Menu.Item>
63             <Menu.Item key="6" icon={ <OrderedListOutlined /> }>
64                 <Link to="/inventario">Inventario</Link>
65             </Menu.Item>
66         </Menu>
67     );
68
69     const transactionalMenu = (

```

```

15 _layout.js X
src > components > _layout > JS _layout.js > _Layout > [6] transactionalMenu
67
68
69
70 const transactionalMenu = (
71   <Menu theme="dark" mode="inline" defaultSelectedKeys={['1']}>
72     <Menu.Item key="1" icon={<HomeOutlined />}>
73       <Link to="/">Inicio</Link>
74     </Menu.Item>
75     <Menu.Item key="2" icon={<PullRequestOutlined />}>
76       <Link to="/movimientos">Movimientos</Link>
77     </Menu.Item>
78     <Menu.Item key="3" icon={<DownloadOutlined />}>
79       <Link to="/entradas">Entradas</Link>
80     </Menu.Item>
81     <Menu.Item key="4" icon={<UploadOutlined />}>
82       <Link to="/salidas">Salidas</Link>
83     </Menu.Item>
84   </Menu>
85 );
86
87 return (
88   <Layout>
89     <Sider trigger={null} collapsible collapsed={collapsed}>
90       <div className="logo" />
91       {JSON.parse(localStorage.getItem('usuario'))?.rol === 1 ? adminMenu : transactionalMenu}
92     </Sider>
93     <Layout className="site-layout">
94       <Header className="site-layout-background">
95         <Row justify="space-between">
96           <Col span={8}>
97             {React.createElement(collapsed ? MenuUnfoldOutlined : MenuFoldOutlined, {
98               className: 'header-icon trigger',
99               onClick: toggle,
100             })}
101           </Col>
102           <Col span={8} style={{ textAlign: 'right', paddingRight: '16px' }}>
103             <Space split={<Divider type="vertical" />}>
104               <Badge dot>
105                 <BellOutlined className="header-icon" />
106               </Badge>
107               <Dropdown overlay={menu}>
108                 <span className="header-icon"><UserOutlined />
109                 <span style={{ padding: '0px 24px 0px 6px' }}>{JSON.parse(localStorage.getItem('usuario'))?.nombre || 'Desconocido'}</span>
110               </Dropdown>
111             </Space>
112           </Col>
113         </Row>
114       </Header>
115       {props.children}
116     </Layout>
117   </Layout>
118 )
119 }
120
121 export default _Layout;

```

```

15 entradasDetails.js X
src > components > entrada > details > JS entradasDetails.js > EntradasForm
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './entradasDetails.css';
7
8 const { Item } = Descriptions;
9
10 function EntradasForm(props) {
11   const { id } = useParams();
12   let navigate = useNavigate();
13   let [loading, setLoading] = useState(true);
14   let [data, setData] = useState(true);
15
16   useEffect(() => {
17     if (id) {
18       fetch('https://localhost:44306/api/entradas/${id}', { method: 'GET' })
19         .then(response => response.json())
20         .then(_data => {
21           setData(_data);
22           setLoading(false);
23         });
24     } else {
25       message.error('id no encontrado');
26       setLoading(false);
27     }
28   }, []);
29
30   return (
31     <PageContent title="Entradas" subtitle="Detalles">
32       <Descriptions title={`Transacción #${id}`} column={1} bordered>
33         <Item label="Id">{data.id}</Item>
34         <Item label="Tipo">{data.tipo}</Item>
35         <Item label="Sucursal">{data?.sucursal?.nombre}</Item>
36         <Item label="Producto">{data?.producto?.nombre}</Item>
37         <Item label="Cantidad">{data.cantidad}</Item>
38         <Item label="Costo">{'$' + parseFloat(data.costo).toFixed(2).replace(/(\d)(?=(\d{3})+)?!(\d))/g, '$1,')}</Item>
39       </Descriptions>
40
41       <Divider />
42
43       <Space split={(<Divider type="vertical" />)}>
44         <Button icon={<ArrowLeftOutlined />} onClick={() => navigate(-1)}>
45           Volver
46         </Button>
47         <Button type="primary" icon={<EditOutlined />} onClick={() => navigate(`/entradasForm/${id}`)}>
48           Editar
49         </Button>
50       </Space>
51     </PageContent>
52   )
53 }
54
55 export default EntradasForm;

```

```

# EntradasForm.js X
src > components > entrada > form > # entradasForm.js > @ EntradasForm
1 import React, { useState, useEffect } from 'react';
2 import { useParams, useNavigate } from 'react-router-dom';
3 import { Row, Typography, Col, Form, Select, InputNumber, Space, Divider, Button, message } from 'antd';
4 import { CloseCircleOutlined, SaveOutlined } from 'ant-design-icons';
5 import PageContent from './../_layouts/pageContent/pageContent';
6 import './entradasForm.css';
7
8 const { Title } = Typography;
9 const { Item } = Form;
10
11 function EntradasForm(props) {
12   const { id } = useParams();
13   let navigate = useNavigate();
14   const [form] = Form.useForm();
15
16   let [loading, setloading] = useState(true);
17   let [pure, setpure] = useState([]);
18   let [sucursales, setsucursales] = useState([]);
19   let [productos, setProductos] = useState([]);
20
21   useEffect(() => {
22     if (id) {
23       Promise.all([
24         fetch('https://localhost:44386/api/sucursales', { method: 'GET' }).then(response => response.json()),
25         fetch('https://localhost:44386/api/entradas?id=' + id, { method: 'GET' }).then(response => response.json())
26       ])
27         .then(responses => {
28           setSucursales(responses[0]);
29           setProductos(responses[1].map(sucursal => { return { label: sucursal.nombre, value: sucursal.id } }));
30           setProductos(responses[1].filter(sucursal => sucursal.id !== responses[0].id_sucursal).map(sucursal => sucursal.catalogos[0].map(catalogo => catalogo.producto).map(producto => { return { label: producto.nombre, value: producto.id } }));
31           form.setFieldsValue(responses[1]);
32           setloading(false);
33         })
34       )
35     } else {
36       fetch('https://localhost:44386/api/sucursales', { method: 'GET' })
37         .then(response => response.json())
38         .then(data => {
39           setpure(data);
40           setsucursales(data.map(sucursal => { return { label: sucursal.nombre, value: sucursal.id } }));
41           setloading(false);
42         })
43     }
44   }, []);
45
46   const onFinish = (values) => {
47     setloading(true);
48     let link, method;
49     if (id) {
50       link = 'https://localhost:44386/api/entradas?id=' + id;
51       method = 'PUT';
52       values.id = parseInt(id);
53     } else {
54       link = 'https://localhost:44386/api/entradas';
55       method = 'POST';
56     }
57
58     values.tipo = 'Entrada';
59     const config = { method, headers: { 'Content-Type': 'application/json' }, body: JSON.stringify(values) };
60
61     fetch(link, config)
62       .then(response => {
63         if (response.ok) {
64           message.success('Entrada guardada');
65           navigate(-1);
66         } else {
67           response.json().then(error => {
68             message.error({ error: status: 1, error: title });
69             setloading(false);
70           })
71         }
72       })
73       .catch(error => {
74         message.error(error.message);
75         setloading(false);
76       });
77   };
78
79   const onChange = (value) => {
80     form.setFieldsValue({ id_producto: undefined });
81     setProductos(pure.filter(sucursal => sucursal.id !== value).map(sucursal => sucursal.catalogos[0].map(catalogo => catalogo.producto).map(producto => { return { label: producto.nombre, value: producto.id } }));
82   }
83 }

```



```

# EntradasList.js X
src > components > entrada > list > # EntradasList.js > EntradasList > @eliminar
1 import React, { useState, useEffect } from 'react';
2 import { Link } from 'react-router-dom';
3 import { Space, Button, Tag, Popconfirm, message, Row, Col, Input, Table } from 'antd';
4 import { ZoomInOutlined, PlusCircleFilled, EditOutlined, DeleteOutlined, SearchOutlined } from '@ant-design/icons';
5 import PageContent from '../../_layout/pageContent/pageContent';
6 import './EntradasList.css';
7
8 let searchTimeout;
9
10 function EntradasList() {
11   const columns = [
12     {
13       title: 'id',
14       dataIndex: 'id',
15       key: 'id',
16       render: (text, record) => {
17         return (
18           <Link to={`/entradasDetails/${record.id}`}>
19             <ZoomInOutlined style={{padding: '0px 6px'}}/>
20             {text}
21           </Link>
22         )
23       }, {
24         title: 'Sucursal',
25         dataIndex: 'sucursal',
26         key: 'sucursal',
27         render: (_, record) => record.sucursal.nombre
28       }, {
29         title: 'Producto',
30         dataIndex: 'producto',
31         key: 'producto',
32         render: (_, record) => record.producto.nombre
33       }, {
34         title: 'Cantidad',
35         dataIndex: 'cantidad',
36         key: 'cantidad'
37       }, {
38         title: 'Acciones',
39         key: 'acciones',
40         width: '8px',
41         render: (_, record) => (
42           <Space size="large">
43             <Link to={`/entradasForm/${record.id}`}>
44               <Button type="primary" size="small" icon={EditOutlined} /> ghost
45               Editar
46             </Button>
47             </Link>
48
49             <Popconfirm
50               title="¿Desea eliminar esta entrada?"
51               onConfirm={() => eliminar(record.id)}
52               okText="Si"
53               cancelText="No"
54               icon={DeleteOutlined style={{color: 'red'}} />
55             >
56               <Button type="dashed" size="small" icon={DeleteOutlined} /> danger
57               Eliminar
58             </Button>
59             </Popconfirm>
60           </Space>
61         ),
62       }
63     ];
64
65     let [data, setData] = useState([]);
66     let [filteredData, setFilteredData] = useState([]);
67     let [loading, setLoading] = useState(true);
68
69     useEffect(() => {
70       fetch('https://localhost:44386/api/entradas', { method: 'GET' })
71         .then(response => response.json())
72         .then(_data => {
73           _data.map(d => { d.key = d.id; return d; })
74           setData(_data);
75           setFilteredData(_data);
76           setLoading(false);
77         });
78     }, []);
79
80     const eliminar = (id) => {
81       setLoading(true);
82       fetch('https://localhost:44386/api/entradas/${id}', { method: 'DELETE' })
83         .then(response => {
84           if (response.status === 204) {

```

```
JS entradasList.js X
src > components > entrada > list > JS entradasList.js > EntradasList > [60] eliminar
81
82   fetch(`https://localhost:44386/api/entradas/${id}`, { method: 'DELETE' })
83     .then(response => {
84       if (response.status === 204) {
85         setData(data => data.filter(_data => _data.id !== id));
86         setFilteredData(filteredData => filteredData.filter(_data => _data.id !== id));
87         message.warning('Entrada eliminada');
88       } else {
89         message.error('Error al eliminar');
90       }
91     })
92     .setLoading(false);
93   });
94 }
95
96 const onSearch = (event) => {
97   clearTimeout(searchTimeout);
98
99   searchTimeout = setTimeout(() => {
100     const text = event.target.value;
101
102     setFilteredData(data.filter(_data => {
103       return (
104         _data.id.toLowerCase().includes(text.toLowerCase()) ||
105         _data.sucursal.nombre.toLowerCase().includes(text.toLowerCase()) ||
106         _data.producto.nombre.toLowerCase().includes(text.toLowerCase()) ||
107         _data.cantidad.toString().toLowerCase().includes(text.toLowerCase())
108       );
109     }));
110   }, 300);
111 }
112
113 return (
114   <PageContent title="Entradas" subtitle="Listado">
115     <Row justify="space-between" align="middle" style={{marginBottom: '24px'}}>
116       <Col span={10}>
117         <Link key="1" to="/entradasForm">
118           <Button type="primary" size="large" icon={PlusCircleFilled} />
119           Agregar entrada
120         </Button>
121       </Link>
122     </Col>
123     <Col span={6}>
124       <Input
125         placeholder="Buscar..."
126         onChange={onSearch}
127         allowClear
128         suffix={SearchOutlined style={{fontSize: 16, color: '#1890ff'}} />
129     </Input>
130   </Col>
131 </Row>
132
133   <Table columns={columns} dataSource={filteredData} loading={loading} />
134 </PageContent>
135 );
136 }
137
138 export default EntradasList;
```

```

JS productosDetails.js X
src > components > producto > details > JS productosDetails.js > ProductosForm > useEffect() callback > then() callback
1 import React, { useState, useEffect } from 'react';
2 import { useParams, useNavigate } from 'react-router-dom';
3 import { Descriptions, Tag, Space, Divider, Button, message } from 'antd';
4 import { ArrowLeftOutlined, EditOutlined } from '@ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './productosDetails.css';
7 import moment from 'moment';
8
9 const { Item } = Descriptions;
10
11 function ProductosForm(props) {
12   const { id } = useParams();
13   let navigate = useNavigate();
14   let [loading, setLoading] = useState(true);
15   let [data, setData] = useState(true);
16
17   useEffect(() => {
18     if (id) {
19       fetch('https://localhost:44386/api/productos/${id}', { method: 'GET' })
20         .then(response => response.json())
21         .then(_data => {
22           _data.fechaVencimiento = moment(_data.fechaVencimiento).format('DD MMM YYYY');
23           setData(_data);
24           setLoading(false);
25         });
26     } else {
27       message.error('id no encontrado');
28       setLoading(false);
29     }
30   }, []);
31
32   return (
33     <PageContent title="Productos" subtitle="Detalles">
34       <Descriptions title={`Producto #${id}`} column={2} bordered>
35         <Item label="Id">{data.id}</Item>
36         <Item label="SKU">{data.sku}</Item>
37         <Item label="Nombre" span={2}>{data.nombre}</Item>
38         <Item label="Proveedor">{data.proveedor?.nombre || '-'}</Item>
39         <Item label="Código de barra del proveedor">{data.codigoBarra || '-'}</Item>
40         <Item label="Fecha de vencimiento" span={2}>{data.fechaVencimiento || '-'}</Item>
41         <Item label="Comentario" span={2}>{data.comentario || '-'}</Item>
42         <Item label="Estado" span={2}>{<Tag color={data.estado === 'Activo' ? 'success' : 'error'}>{data.estado}</Tag>}</Item>
43       </Descriptions>
44
45       <Divider />
46
47       <Space split={<Divider type="vertical" />}>
48         <Button icon={<ArrowLeftOutlined />} onClick={() => navigate(-1)}>
49           Volver
50         </Button>
51         <Button type="primary" icon={<EditOutlined />} onClick={() => navigate(`/productosForm/${id}`)}>
52           Editar
53         </Button>
54       </Space>
55     </PageContent>
56   )
57 }
58
59 export default ProductosForm;

```

```

# productosForm.js X
src > components > producto > form > # productosForm.js > ...
1 import React, { useState, useEffect } from 'react';
2 import { useParams, useNavigate } from "react-router-dom";
3 import { Row, Typography, Col, Form, Input, Select, Switch, DatePicker, Space, Divider, Button, message } from 'antd';
4 import { CloseCircleOutlined, SaveOutlined } from '@ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './productosForm.css';
7 import moment from 'moment';
8
9 const { Title } = Typography;
10 const { Item } = Form;
11 const { TextArea } = Input;
12
13 function ProductosForm(props) {
14   const { id } = useParams();
15   let navigate = useNavigate();
16   const [form] = Form.useForm();
17
18   let [loading, setLoading] = useState(true);
19   let [proveedores, setProveedores] = useState([]);
20
21   useEffect(() => {
22     if (id) {
23       Promise.all([
24         fetch('https://localhost:44386/api/proveedores', { method: 'GET' }).then(response => response.json()),
25         fetch('https://localhost:44386/api/productos/${id}', { method: 'GET' }).then(response => response.json())
26       ])
27         .then(responses => {
28           setProveedores(responses[0].map(proveedor => { return { label: proveedor.nombre, value: proveedor.id } }));
29           responses[1].estado = responses[1].estado === 'Activo' ? true : false;
30           responses[1].fechaVencimiento = moment(responses[1].fechaVencimiento);
31           form.setFieldsValue(responses[1]);
32           setLoading(false);
33         });
34     } else {
35       fetch('https://localhost:44386/api/proveedores', { method: 'GET' })
36         .then(response => response.json())
37         .then(data => {
38           setProveedores(data.map(proveedor => { return { label: proveedor.nombre, value: proveedor.id } }));
39           setLoading(false);
40         });
41     }
42   }, [id]);
43
44   const onFinish = (values) => {
45     setLoading(true);
46
47     let link, method;
48
49     if (id) {
50       link = 'https://localhost:44386/api/productos/${id}';
51       method = 'PUT';
52       values.id = parseInt(id);
53     } else {
54       link = 'https://localhost:44386/api/productos';
55       method = 'POST';
56     }
57
58     values.estado = values.estado ? 'Activo' : 'Inactivo';
59     const config = { method, headers: { 'Content-Type': 'application/json' }, body: JSON.stringify(values) }
60
61     fetch(link, config)
62       .then(response => {
63         if (response.ok) {
64           message.success('Producto guardada');
65           navigate(-1);
66         } else {
67           response.json().then(error => {
68             message.error(`${error.status}: ${error.title}`);
69             setLoading(false);
70           });
71         }
72       })
73       .catch((error) => {
74         message.error(error.message);
75         setLoading(false);
76       });
77   };
78
79   return (
80     <PageContent title="Productos" subtitle="Formulario">
81       <Row>
82         <Col span=8>
83           <Title level=5 style={{ marginBottom: '24px' }}><id ? 'Editar producto' : 'Agregar producto'></Title>
84

```

# productosForm.js X

src > components > producto > form > # productosForm.js > ...

```
77   };
78
79   return (
80     <PageContent title="Productos" subtitle="Formulario">
81       <Row>
82         <Col span={8}>
83           <Title level={5} style={{ marginBottom: '24px' }}>{id ? 'Editar producto' : 'Agregar producto'}</Title>
84
85           <Form
86             form={form}
87             layout="vertical"
88             onFinish={onFinish}
89             autoComplete="off"
90             initialValues={{
91               estado: true
92             }}
93           >
94             <Item label="SKU" name="sku"
95               rules={[
96                 {
97                   required: true,
98                   message: 'Este campo es requerido',
99                 },
100              ]}
101               hasFeedback>
102               <Input disabled={loading} allowClear />
103             </Item>
104
105             <Item label="Código de barra del proveedor" name="codigoBarra">
106               <Input disabled={loading} allowClear />
107             </Item>
108
109             <Item label="Nombre" name="nombre"
110               rules={[
111                 {
112                   required: true,
113                   message: 'Este campo es requerido',
114                 },
115              ]}
116               hasFeedback>
117               <Input disabled={loading} allowClear />
118             </Item>
119
120             <Item label="Proveedor" name="id_Proveedor">
121               <Select showSearch options={proveedores} allowClear />
122             </Item>
123
124             <Item label="Fecha de vencimiento" name="fechaVencimiento">
125               <DatePicker style={{width: '100%'}} />
126             </Item>
127
128             <Item label="Comentario" name="comentario">
129               <TextArea showCount maxLength={1000} />
130             </Item>
131
132             <Item label="Estado" name="estado" valuePropName="checked">
133               <Switch checkedChildren="Activo" uncheckedChildren="Inactivo" disabled={loading} />
134             </Item>
135
136             <Divider />
137
138             <Item>
139               <Space split={(<Divider type="vertical" />)}>
140                 <Button type="primary" icon={(<CloseCircleOutlined />)} danger ghost onClick={() => navigate(-1)}>
141                   Cancelar
142                 </Button>
143                 <React.StrictMode>
144                   <Button type="primary" htmlType="submit" icon={(<SaveOutlined />)} loading={loading}>
145                     Guardar
146                   </Button>
147                 </React.StrictMode>
148               </Space>
149             </Item>
150           </Form>
151         </Col>
152       </Row>
153     </PageContent>
154   )
155 }
156
157 export default ProductosForm;
```

```

# productosList.js X
src > components > producto > list > # productosList.js > @ ProductosList
1  import React, { useState, useEffect } from 'react';
2  import { Link } from 'react-router-dom';
3  import { Space, Button, Tag, Popconfirm, message, Row, Col, Input, Table } from 'antd';
4  import { ZoomInOutlined, PlusCircleFilled, EditOutlined, DeleteOutlined, SearchOutlined } from 'ant-design/icons';
5  import PageContent from '../../_layout/pageContent/pageContent';
6  import './productosList.css';
7
8  let searchTimeout;
9
10 function ProductosList() {
11   const columns = [
12     {
13       title: 'SKU',
14       dataIndex: 'sku',
15       key: 'sku',
16       render: (text, record) => {
17         return (
18           <Link to={`/${productosDetails}/${record.id}`}>
19             <ZoomInOutlined style={{padding: '0px 6px'}}/>
20             {text}
21           </Link>
22         )
23       },
24     }, {
25       title: 'Nombre',
26       dataIndex: 'nombre',
27       key: 'nombre',
28     }, {
29       title: 'Estado',
30       dataIndex: 'estado',
31       key: 'estado',
32       render: text => {
33         let color = '';
34
35         if (text.toLowerCase() === 'activo') color = 'success';
36         if (text.toLowerCase() === 'inactivo') color = 'error';
37
38         return (
39           <Tag color={color}>{text}</Tag>
40         )
41       },
42     }, {
43       title: 'Acciones',
44       key: 'acciones',
45       width: '0px',
46       render: (_, record) => (
47         <Space size="large">
48           <Link to={`/${productosForm}/${record.id}`}>
49             <Button type="primary" size="small" icon={EditOutlined} /> ghost<br>
50             Editar
51           </Button>
52           </Link>
53
54           <Popconfirm
55             title="¿Desea eliminar este producto?"
56             onConfirm={() => eliminar(record.id)}
57             okText="Si"
58             cancelText="No"
59             icon={DeleteOutlined style={{color: 'red'}} />
60
61             <Button type="dashed" size="small" icon={DeleteOutlined} /> danger<br>
62             Eliminar
63           </Button>
64           </Popconfirm>
65         </Space>
66       ),
67     },
68   ];
69
70   let [data, setData] = useState([]);
71   let [filteredData, setFilteredData] = useState([]);
72   let [loading, setLoading] = useState(true);
73
74   useEffect(() => {
75     fetch('https://localhost:44386/api/productos', { method: 'GET' })
76       .then(response => response.json())
77       .then(_data => {
78         _data.map(d => { d.key = d.id; return d; })
79         setData(_data);
80         setFilteredData(_data);
81         setLoading(false);
82       });
83   }, []);
84
85   const eliminar = (id) => {
86     setLoading(true);

```



JS proveedoresDetails.js X

src > components > proveedor > detalles > JS proveedoresDetails.js > ProveedoresForm

```
1 import React, { useState, useEffect } from 'react';
2 import { useParams, useNavigate } from "react-router-dom";
3 import { Descriptions, Tag, Space, Divider, Button, message } from 'antd';
4 import { ArrowLeftOutlined, EditOutlined } from '@ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './proveedoresDetails.css';
7
8 const { Item } = Descriptions;
9
10 function ProveedoresForm(props) {
11   const { id } = useParams();
12   let navigate = useNavigate();
13   let [loading, setLoading] = useState(true);
14   let [data, setData] = useState(true);
15
16   useEffect(() => {
17     if (id) {
18       fetch(`https://localhost:44306/api/proveedores/${id}`, { method: 'GET' })
19         .then(response => response.json())
20         .then(_data => {
21           setData(_data);
22           setLoading(false);
23         });
24     } else {
25       message.error('id no encontrado');
26       setLoading(false);
27     }
28   }, []);
29
30   return (
31     <PageContent title="Proveedores" subtitle="Detalles">
32       <Descriptions title={`Proveedor #${id}`} column={1} bordered>
33         <Item label="Id">{data.id}</Item>
34         <Item label="Nombre">{data.nombre}</Item>
35         <Item label="Contacto">{data.contacto || '-'}</Item>
36         <Item label="Comentario">{data.comentario || '-'}</Item>
37         <Item label="Estado"><Tag color={data.estado === 'Activo' ? 'success' : 'error'}>{data.estado}</Tag></Item>
38       </Descriptions>
39
40       <Divider />
41
42       <Space split={}<Divider type="vertical" />>
43         <Button icon={<ArrowLeftOutlined />} onClick={() => navigate(-1)}>
44           Volver
45         </Button>
46         <Button type="primary" icon={<EditOutlined />} onClick={() => navigate(`/proveedoresForm/${id}`)}>
47           Editar
48         </Button>
49       </Space>
50     </PageContent>
51   )
52 }
53
54 export default ProveedoresForm;
```

```

JS proveedoresForm.js X
src > components > proveedor > form > JS proveedoresForm.js > ProveedoresForm
1  import React, { useState, useEffect } from 'react';
2  import { useParams, useNavigate } from "react-router-dom";
3  import { Row, Typography, Col, Form, Input, Radio, Switch, Space, Divider, Button, message } from 'antd';
4  import { CloseCircleOutlined, SaveOutlined } from '@ant-design/icons';
5  import PageContent from '../../_layout/pageContent/pageContent';
6  import './proveedoresForm.css';
7
8  const { Title } = Typography;
9  const { Item } = Form;
10 const { TextArea } = Input;
11
12 function ProveedoresForm(props) {
13   const { id } = useParams();
14   let navigate = useNavigate();
15   const [form] = Form.useForm();
16
17   let [loading, setLoading] = useState(true);
18
19   useEffect(() => {
20     if (id) {
21       fetch('https://localhost:44306/api/proveedores/${id}', { method: 'GET' })
22         .then(response => response.json())
23         .then(data => {
24           data.estado = data.estado === 'Activo' ? true : false;
25           form.setFieldsValue(data);
26           setLoading(false);
27         });
28     } else {
29       setLoading(false);
30     }
31   }, []);
32
33   const onFinish = (values) => {
34     setLoading(true);
35
36     let link, method;
37
38     if (id) {
39       link = 'https://localhost:44306/api/proveedores/${id}';
40       method = 'PUT';
41       values.id = parseInt(id);
42     } else {
43       link = 'https://localhost:44306/api/proveedores';
44       method = 'POST';
45     }
46
47     values.estado = values.estado ? 'Activo' : 'Inactivo';
48     const config = { method, headers: { 'Content-Type': 'application/json' }, body: JSON.stringify(values) }
49
50     fetch(link, config)
51       .then(response => {
52         if (response.ok) {
53           message.success('Proveedor guardada');
54           navigate(-1);
55         } else {
56           response.json().then(error => {
57             message.error(`${error.status}: ${error.title}`);
58             setLoading(false);
59           })
60         }
61       })
62       .catch((error) => {
63         message.error(error.message);
64         setLoading(false);
65       });
66   });
67
68   return (
69     <PageContent title="Proveedores" subtitle="Formulario">

```

```

JS proveedoresForm.js X
src > components > proveedor > form > JS proveedoresForm.js > ProveedoresForm
67
68   return (
69     <PageContent title="Proveedores" subtitle="Formulario">
70       <Row>
71         <Col span={8}>
72           <Title level={5} style={{ marginBottom: '24px' }}><id ? 'Editar proveedor' : 'Agregar proveedor'></Title>
73
74           <Form
75             form={form}
76             layout="vertical"
77             onFinish={onFinish}
78             autoComplete="off"
79             initialValues={{
80               estado: true
81             }}
82           >
83             <Item label="Nombre" name="nombre">
84               <Input disabled={loading} allowClear />
85               <FormRules>
86                 {
87                   required: true,
88                   message: 'Este campo es requerido',
89                 }
90               </FormRules>
91             </Item>
92
93             <Item label="Contacto" name="contacto">
94               <Input disabled={loading} allowClear />
95             </Item>
96
97             <Item label="Comentario" name="comentario">
98               <TextArea showCount maxLength={1000} />
99             </Item>
100
101             <Item label="Estado" name="estado" valuePropName="checked">
102               <Switch checkedChildren="Activo" uncheckedChildren="Inactivo" disabled={loading} />
103             </Item>
104
105             <Divider />
106
107             <Item>
108               <Space split={<Divider type="vertical" />}>
109                 <Button type="primary" icon={<CloseCircleOutlined />} danger ghost onClick={() => navigate(-1)}>
110                   Cancelar
111                 </Button>
112                 <React.StrictMode>
113                   <Button type="primary" htmlType="submit" icon={<SaveOutlined />} loading={loading}>
114                     Guardar
115                   </Button>
116                 </React.StrictMode>
117               </Space>
118             </Item>
119           </Form>
120         </Col>
121       </Row>
122     </PageContent>
123   )
124 }
125
126
127 export default ProveedoresForm;

```

```

# proveedoresList.js X
src > components > proveedor > list > # proveedoresList.js > ProvedoresList > useEffect() callback > then() callback
1 import React, { useState, useEffect } from 'react';
2 import { Link } from 'react-router-dom';
3 import { Space, Button, Tag, Popconfirm, message, Row, Col, Input, Table } from 'antd';
4 import { ZoomInOutlined, PlusCircleFilled, EditOutlined, DeleteOutlined, SearchOutlined } from '@ant-design/icons';
5 import PageContent from '../../_layout/pageContent/pageContent';
6 import './proveedoresList.css';
7
8 let searchTimeout;
9
10 function ProvedoresList() {
11   const columns = [
12     {
13       title: 'Nombre',
14       dataIndex: 'nombre',
15       key: 'nombre',
16       render: (text, record) => {
17         return (
18           <Link to={`/proveedoresDetails/${record.id}`}>
19             <ZoomInOutlined style={{padding: '0px 6px'}}/>
20             {text}
21           </Link>
22         )
23       },
24     }, {
25       title: 'Contacto',
26       dataIndex: 'contacto',
27       key: 'contacto',
28       render: text => text || '-'
29     }, {
30       title: 'Estado',
31       dataIndex: 'estado',
32       key: 'estado',
33       render: text => {
34         let color = '';
35
36         if (text.toLowerCase() === 'activo') color = 'success';
37         if (text.toLowerCase() === 'inactivo') color = 'error';
38
39         return (
40           <Tag color={color}>{text}</Tag>
41         )
42       },
43     }, {
44       title: 'Acciones',
45       key: 'acciones',
46       width: '80px',
47       render: (_, record) => (
48         <Space size="large">
49           <Link to={`/proveedoresForm/${record.id}`}>
50             <Button type="primary" size="small" icon={EditOutlined} /> ghost
51             Editar
52           </Link>
53
54           <Popconfirm
55             title="¿Desca eliminar este proveedor?"
56             onConfirm={() => eliminar(record.id)}
57             okText="Si"
58             cancelText="No"
59             icon={DeleteOutlined style={{color: 'red'}} />
60           >
61             <Button type="dashed" size="small" icon={DeleteOutlined} /> danger
62             Eliminar
63           </Button>
64         </Popconfirm>
65       </Space>
66     ),
67   ];
68
69   let [data, setData] = useState([]);
70   let [filteredData, setFilteredData] = useState([]);
71   let [loading, setLoading] = useState(true);
72
73   useEffect(() => {
74     fetch('https://localhost:44386/api/proveedores', { method: 'GET' })
75       .then(response => response.json())
76       .then(_data => {
77         _data.map(d => { d.key = d.id; return d; })
78         setData(_data);
79         setFilteredData(_data);
80         setLoading(false);
81       });
82   }, []);
83
84   const eliminar = (id) => {

```

```

# proveedoresList.js X
src > components > proveedor > list > # proveedoresList.js > ProvedoresList > useEffect() callback > then() callback
84   const eliminar = (id) => {
85     setLoading(true);
86
87     fetch(`https://localhost:44386/api/proveedores/${id}`, { method: 'DELETE' })
88       .then(response => {
89         if (response.status === 204) {
90           setData(data => data.filter(_data => _data.id !== id));
91           setFilteredData(filteredData => filteredData.filter(_data => _data.id !== id));
92           message.warning('Proveedor eliminada');
93         } else {
94           message.error('Error al eliminar');
95         }
96
97         setLoading(false);
98       });
99   }
100
101   const onSearch = (event) => {
102     clearTimeout(searchTimeout);
103
104     searchTimeout = setTimeout(() => {
105       const text = event.target.value;
106
107       setFilteredData(data.filter(_data => {
108         return (
109           _data.nombre.toLowerCase().includes(text.toLowerCase()) ||
110           _data.contacto.toLowerCase().includes(text.toLowerCase()) ||
111           _data.estado.toLowerCase().includes(text.toLowerCase())
112         );
113       }));
114     }, 300);
115   }
116
117   return (
118     <PageContent title="Proveedores" subtitle="Listado">
119       <Row justify="space-between" align="middle" style={{marginBottom: '24px'}}>
120         <Col span={10}>
121           <Link key="1" to="/proveedoresForm">
122             <Button type="primary" size="large" icon={PlusCircleFilled} />
123             Agregar proveedor
124           </Button>
125         </Link>
126       </Col>
127       <Col span={6}>
128         <Input
129           placeholder="Buscar..."
130           onChange={onSearch}
131           allowClear
132           suffix={SearchOutlined style={{fontSize: 16, color: '#1890ff'}} />
133       </Col>
134     </Row>
135
136     <Table columns={columns} dataSource={filteredData} loading={loading} />
137   </PageContent>
138 );
139 }
140
141
142 export default ProveedoresList;

```

```

JS salidasDetails.js X
src > components > salida > details > JS salidasDetails.js > SalidasForm
1 import React, { useState, useEffect } from 'react';
2 import { useParams, useNavigate } from "react-router-dom";
3 import { Descriptions, Tag, Space, Divider, Button, message } from 'antd';
4 import { ArrowLeftOutlined, EditOutlined } from '@ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './salidasDetails.css';
7
8 const { Item } = Descriptions;
9
10 function SalidasForm(props) {
11   const { id } = useParams();
12   let navigate = useNavigate();
13   let [loading, setLoading] = useState(true);
14   let [data, setData] = useState(true);
15
16   useEffect(() => {
17     if (id) { Follow link \(ctrl + click\)
18       fetch(`https://localhost:44306/api/salidas/${id}`, { method: 'GET' })
19         .then(response => response.json())
20         .then(_data => {
21           setData(_data);
22           setLoading(false);
23         });
24     } else {
25       message.error('id no encontrado');
26       setLoading(false);
27     }
28   }, []);
29
30   return (
31     <PageContent title="Salidas" subtitle="Detalles">
32       <Descriptions title={`Transacción #${id}`} column={1} bordered>
33         <Item label="Id">{data.id}</Item>
34         <Item label="Tipo">{data.tipo}</Item>
35         <Item label="Sucursal">{data?.sucursal?.nombre}</Item>
36         <Item label="Producto">{data?.producto?.nombre}</Item>
37         <Item label="Cantidad">{data.cantidad}</Item>
38         <Item label="Costo">{'$' + parseFloat(data.costo).toFixed(2).replace(/(\d)(?=(\d{3})+)?(!\d))/g, '$1,')}</Item>
39       </Descriptions>
40
41       <Divider />
42
43       <Space split={<Divider type="vertical" />}>
44         <Button icon={<ArrowLeftOutlined />} onClick={() => navigate(-1)}>
45           Volver
46         </Button>
47         <Button type="primary" icon={<EditOutlined />} onClick={() => navigate(`/entradasForm/${id}`)}>
48           Editar
49         </Button>
50       </Space>
51     </PageContent>
52   )
53 }
54
55 export default SalidasForm;

```

```

# SalidasFormjs X
src > components > salidas > form > # SalidasFormjs > SalidasForm
1 import React, { useState, useEffect } from 'react';
2 import { useForm, useNavigate } from 'react-router-dom';
3 import { Row, Typography, Col, Form, Select, InputNumber, Space, Divider, Button, message } from 'antd';
4 import { CloseCircleOutlined, SaveOutlined } from '@ant-design/icons';
5 import PageContent from '../../_layout/pageContent/pageContent';
6 import './salidasForm.css';
7
8 const { Title } = Typography;
9 const { Item } = Form;
10
11 function SalidasForm(props) {
12   const { id } = useParams();
13   let navigate = useNavigate();
14   const { form } = useForm();
15
16   let [loading, setLoading] = useState(true);
17   let [pure, setPure] = useState({});
18   let [sucursales, setSucursales] = useState({});
19   let [productos, setProductos] = useState({});
20
21   useEffect(() => {
22     if (id) {
23       Promise.all([
24         fetch('https://localhost:44386/api/sucursales', { method: 'GET' }).then(response => response.json()),
25         fetch('https://localhost:44386/api/salidas/' + id, { method: 'GET' }).then(response => response.json())
26       ])
27         .then(responses => {
28           setPure(responses[0]);
29           setSucursales(responses[0].map(sucursal => { return { label: sucursal.nombre, value: sucursal.id } }));
30           setProductos(responses[0].filter(sucursal => sucursal.id === responses[1].id_sucursal).map(sucursal => sucursal.catalogos[0].map(catalogo => catalogo.producto).map(producto => { return { label: producto.nombre, value: producto.id } }));
31           form.setFieldsValue(responses[1]);
32           setLoading(false);
33         })
34       ) else {
35         fetch('https://localhost:44386/api/sucursales', { method: 'GET' })
36           .then(response => response.json())
37           .then(data => {
38             setPure(data);
39             setSucursales(data.map(sucursal => { return { label: sucursal.nombre, value: sucursal.id } }));
40             setLoading(false);
41           });
42       }
43     }, []);
44
45     const onFinish = (values) => {
46       setLoading(true);
47
48       let link, method;
49
50       if (id) {
51         link = 'https://localhost:44386/api/salidas/' + id;
52         method = 'PUT';
53         values.id = parseInt(id);
54       } else {
55         link = 'https://localhost:44386/api/salidas';
56         method = 'POST';
57       }
58
59       values.tipo = 'Salida';
60       const config = { method, headers: { 'Content-Type': 'application/json' }, body: JSON.stringify(values) };
61
62       fetch(link, config)
63         .then(response => {
64           if (response.ok) {
65             message.success('Salida guardada');
66             navigate(-1);
67           } else {
68             response.json().then(error => {
69               message.error(`${error.status}: ${error.title}`);
70               setLoading(false);
71             });
72           }
73         })
74         .catch(error => {
75           message.error(error.message);
76           setLoading(false);
77         });
78     };
79
80     const onChange = (value) => {
81       form.setFieldsValue({ id_producto: undefined });
82       setProductos(pure.filter(sucursal => sucursal.id === value).map(sucursal => sucursal.catalogos[0].map(catalogo => catalogo.producto).map(producto => { return { label: producto.nombre, value: producto.id } }));
83     }
84

```



```

JS salidasList.js X
src > components > salida > list > JS salidasList.js > SalidasList
1 import React, { useState, useEffect } from 'react';
2 import { Link } from "react-router-dom";
3 import { Space, Button, Tag, Popconfirm, message, Row, Col, Input, Table } from 'antd';
4 import { ZoomInOutlined, PlusCircleFilled, EditOutlined, DeleteOutlined, SearchOutlined } from '@ant-design/icons';
5 import PageContent from '../../_layout/pageContent/pageContent';
6 import './salidasList.css';
7
8 let searchTimeout;
9
10 function SalidasList() {
11   const columns = [
12     {
13       title: 'Id',
14       dataIndex: 'id',
15       key: 'id',
16       render: (text, record) => {
17         return (
18           <Link to={`/salidasDetails/${record.id}`}>
19             <ZoomInOutlined style={{padding: '0px 6px'}}/>
20             {text}
21           </Link>
22         )
23       }, {
24         title: 'Sucursal',
25         dataIndex: 'sucursal',
26         key: 'sucursal',
27         render: (_, record) => record.sucursal.nombre
28       }, {
29         title: 'Producto',
30         dataIndex: 'producto',
31         key: 'producto',
32         render: (_, record) => record.producto.nombre
33       }, {
34         title: 'Cantidad',
35         dataIndex: 'cantidad',
36         key: 'cantidad'
37       }, {
38         title: 'Acciones',
39         key: 'acciones',
40         width: '0px',
41         render: (_, record) => (
42           <Space size="large">
43             <Link to={`/salidasForm/${record.id}`}>
44               <Button type="primary" size="small" icon={EditOutlined} /> ghost>
45                 Editar
46             </Button>
47             </Link>
48
49             <Popconfirm
50               title="¿Desea eliminar esta salida?"
51               onConfirm={() => eliminar(record.id)}
52               okText="Sí"
53               cancelText="No"
54               icon={DeleteOutlined} style={{ color: 'red' }} />
55             >
56               <Button type="dashed" size="small" icon={DeleteOutlined} /> danger>
57                 Eliminar
58             </Button>
59             </Popconfirm>
60           </Space>
61         ),
62       ]];
63
64   let [data, setData] = useState([]);
65   let [filteredData, setFilteredData] = useState([]);
66   let [loading, setLoading] = useState(true);
67
68   useEffect(() => {
69     fetch("https://localhost:44306/api/salidas", { method: 'GET' })

```

```
70 .then(response => response.json())
71 .then(_data => {
72   _data.map(d => { d.key = d.id; return d; })
73   setData(_data);
74   setFilteredData(_data);
75   setLoading(false);
76 });
77 }, []);
78
79 const eliminar = (id) => {
80   setLoading(true);
81
82   fetch(`https://localhost:44306/api/salidas/${id}`, { method: 'DELETE' })
83   .then(response => {
84     if (response.status === 204) {
85       setData(data => data.filter(_data => _data.id !== id));
86       setFilteredData(filteredData => filteredData.filter(_data => _data.id !== id));
87       message.warning('Salida eliminada');
88     } else {
89       message.error('Error al eliminar');
90     }
91
92     setLoading(false);
93   });
94 }
95
96 const onSearch = (event) => {
97   clearTimeout(searchTimeout);
98
99   searchTimeout = setTimeout(() => {
100     const text = event.target.value;
101
102     setFilteredData(data.filter(_data => {
103       return (
104         _data.id.toLowerCase().includes(text.toLowerCase()) ||
105         _data.sucursal.nombre.toLowerCase().includes(text.toLowerCase()) ||
106         _data.producto.nombre.toLowerCase().includes(text.toLowerCase()) ||
107         _data.cantidad.toString().toLowerCase().includes(text.toLowerCase())
108       );
109     }));
110   }, 300);
111 }
112
113 return (
114   <PageContent title="Salidas" subtitle="Listado">
115     <Row justify="space-between" align="middle" style={{marginBottom: '24px'}}>
116       <Col span={10}>
117         <Link key="1" to="/salidasForm">
118           <Button type="primary" size="large" icon={}<PlusCircleFilled />>
119             Agregar salida
120           </Button>
121         </Link>
122       </Col>
123       <Col span={6}>
124         <Input
125           placeholder="Buscar..."
126           onChange={onSearch}
127           allowClear
128           suffix={}<SearchOutlined style={{ fontSize: 16, color: '#1890ff' }} />
129         />
130       </Col>
131     </Row>
132
133     <Table columns={columns} dataSource={filteredData} loading={loading} />
134   </PageContent>
135 );
136 }
137
138 export default SalidasList;
```

```

JS sucursalesDetails.js X
src > components > sucursal > details > JS sucursalesDetails.js > SucursalesForm > useEffect() callback > then() callback
4 import { ArrowLeftOutlined, EditOutlined } from 'ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './sucursalesDetails.css';
7
8 const { Item } = Descriptions;
9
10 function SucursalesForm(props) {
11   const { id } = useParams();
12   let navigate = useNavigate();
13   let [loading, setLoading] = useState(true);
14   let [data, setData] = useState({});
15
16   useEffect(() => {
17     if (id) {
18       fetch(`https://localhost:44306/api/sucursales/${id}`, { method: 'GET' })
19         .then(response => response.json())
20         .then(_data => {
21           setData(_data);
22           setLoading(false);
23         });
24     } else {
25       message.error('id no encontrado');
26       setLoading(false);
27     }
28   }, []);
29
30   return (
31     <PageContent title="Sucursales" subtitle="Detalles">
32       <Descriptions title={`Sucursal #${id}`} column={1} bordered>
33         <Item label="Id">{data.id}</Item>
34         <Item label="Nombre">{data.nombre}</Item>
35         <Item label="Tipo">{data.tipo}</Item>
36         <Item label="Productos">{data.catalogos?.length ? data.catalogos.map(catalogo => { return <Tag>{catalogo.producto.nombre}</Tag> } : '-')}</Item>
37         <Item label="Estado">{<Tag color={data.gestado === 'Activo' ? 'success' : 'error'}>{data.gestado}</Tag>}</Item>
38       </Descriptions>
39
40       <Divider />
41
42       <Space split={<Divider type="vertical" />}>
43         <Button icon={<ArrowLeftOutlined />} onClick={() => navigate(-1)}>
44           Volver
45         </Button>
46         <Button type="primary" icon={<EditOutlined />} onClick={() => navigate(`/sucursalesForm/${id}`)}>
47           Editar
48         </Button>
49       </Space>
50     </PageContent>
51   )
52 }
53
54 export default SucursalesForm;

```

```

# SucursalesForm.js X
src > components > sucursal > form > # sucursalesForm.js > SucursalesForm > useEffect() callback > then() callback > catalogos.map() callback
1 import React, { useState, useEffect } from 'react';
2 import { useParams, useNavigate } from 'react-router-dom';
3 import { Row, Typography, Col, Form, Input, Radio, Select, Switch, Space, Divider, Button, message } from 'antd';
4 import { CloseCircleOutlined, SaveOutlined } from '@ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './sucursalesForm.css';
7
8 const { Title } = Typography;
9 const { Item } = Form;
10
11 function SucursalesForm(props) {
12   const { id } = useParams();
13   let navigate = useNavigate();
14   const [form] = Form.useForm();
15
16   let [loading, setloading] = useState(true);
17   let [productos, setProductos] = useState([]);
18
19   useEffect(() => {
20     if (id) {
21       Promise.all([
22         fetch('https://localhost:44306/api/productos', { method: 'GET' }).then(response => response.json()),
23         fetch('https://localhost:44306/api/sucursales/${id}', { method: 'GET' }).then(response => response.json())
24       ])
25         .then(responses => {
26           setProductos(responses[0].map(producto => { return { label: producto.nombre, value: producto.id } }));
27           responses[1].estado = responses[1].estado === 'Activo' ? true : false;
28           responses[1].catalogos = responses[1].catalogos.map(catalogo => catalogo.id_Producto);
29           form.setFieldsValue(responses[1]);
30           setloading(false);
31         });
32     } else {
33       fetch('https://localhost:44306/api/productos', { method: 'GET' })
34         .then(response => response.json())
35         .then(data => {
36           setProductos(data.map(producto => { return { label: producto.nombre, value: producto.id } }));
37           setloading(false);
38         });
39     }
40   }, []);
41
42   const onFinish = (values) => {
43     setloading(true);
44
45     values.catalogos = values.catalogos.map(_catalogo => { return { id_Sucursal: id, id_Producto: _catalogo } });
46
47     let link, method;
48
49     if (id) {
50       link = 'https://localhost:44306/api/sucursales/${id}';
51       method = 'PUT';
52       values.id = parseInt(id);
53     } else {
54       link = 'https://localhost:44306/api/sucursales';
55       method = 'POST';
56     }
57
58     values.estado = values.estado ? 'Activo' : 'Inactivo';
59     const config = { method, headers: { 'Content-Type': 'application/json' }, body: JSON.stringify(values) };
60
61     fetch(link, config)
62       .then(response => {
63         if (response.ok) {
64           message.success('Sucursal guardada');
65           navigate(-1);
66         } else {
67           response.json().then(error => {
68             message.error(`${error.status}: ${error.title}`);
69             setloading(false);
70           });
71         }
72       })
73       .catch((error) => {
74         message.error(error.message);
75         setloading(false);
76       });
77   });
78
79   return (
80     <PageContent title="Sucursales" subtitle="Formulario">
81       <Row>
82         <Col span={8}>
83           <Title level={5} style={{ marginBottom: '24px' }}>{id ? 'Editar sucursal' : 'Agregar sucursal'}</Title>
84

```

```

# sucursalesForm.js X
src > components > sucursal > form > # sucursalesForm.js > SucursalesForm > useEffect() callback > then() callback > catalogos.map() callback
82 <Col span={8}>
83   <Title level={5} style={{ marginBottom: '24px' }}>{id ? 'Editar sucursal' : 'Agregar sucursal'}</Title>
84
85   <Form
86     form={form}
87     layout="vertical"
88     onFinish={onFinish}
89     autoComplete="off"
90     initialValues={{
91       tipo: 'Tienda',
92       estado: true
93     }}
94   >
95     <Item label="Nombre" name="nombre"
96       rules={[
97         {
98           required: true,
99           message: 'Este campo es requerido',
100         },
101       ]}
102       hasFeedback>
103       <Input disabled={loading} allowClear />
104     </Item>
105
106     <Item label="Tipo de sucursal" name="tipo">
107       <Radio.Group buttonStyle="solid" disabled={loading}>
108         <Radio.Button value="Tienda">Tienda</Radio.Button>
109         <Radio.Button value="Almacén">Almacén</Radio.Button>
110       </Radio.Group>
111     </Item>
112
113     <Item label="Productos" name="catalogos">
114       <Select mode="multiple" options={productos} allowClear />
115     </Item>
116
117     <Item label="Estado" name="estado" valuePropName="checked">
118       <Switch checkedChildren="Activo" unCheckedChildren="Inactivo" disabled={loading} />
119     </Item>
120
121     <Divider />
122
123     <Item>
124       <Space split={<Divider type="vertical" />}>
125         <Button type="primary" icon={<CloseCircleOutlined />} danger ghost onClick={() => navigate(-1)}>
126           Cancelar
127         </Button>
128         <React.StrictMode>
129           <Button type="primary" htmlType="submit" icon={<SaveOutlined />} loading={loading}>
130             Guardar
131           </Button>
132         </React.StrictMode>
133       </Space>
134     </Item>
135   </Form>
136 </Col>
137 </Row>
138 </PageContent>
139 )
140 }
141
142 export default SucursalesForm;

```

```

# sucursalesList.js X
src > components > sucursal > list > # sucursalesList.js > SucursalesList > @ columns > render
1 import React, { useState, useEffect } from 'react';
2 import { Link } from 'react-router-dom';
3 import { Space, Button, Tag, Popconfirm, message, Row, Col, Input, Table } from 'antd';
4 import { ZoomInOutlined, PlusCircleFilled, EditOutlined, DeleteOutlined, SearchOutlined } from '@ant-design/icons';
5 import PageContent from '../_layout/pageContent/pageContent';
6 import './sucursalesList.css';
7
8 let searchTimeout;
9
10 function SucursalesList() {
11   const columns = [
12     {
13       title: 'Nombre',
14       dataIndex: 'nombre',
15       key: 'nombre',
16       render: (text, record) => {
17         return (
18           <Link to={`/${sucursalesDetails}/${record.id}`}>
19             <ZoomInOutlined style={{padding: '0px 6px'}}/>
20             {text}
21           </Link>
22         )
23       },
24       {
25         title: 'Tipo',
26         dataIndex: 'tipo',
27         key: 'tipo',
28       },
29       {
30         title: 'Estado',
31         dataIndex: 'estado',
32         key: 'estado',
33         render: text => {
34           let color = '';
35
36           if (text.toLowerCase() === 'activo') color = 'success';
37           if (text.toLowerCase() === 'inactivo') color = 'error';
38
39           return (
40             <Tag color={color}>{text}</Tag>
41           )
42         },
43       },
44       {
45         title: 'Acciones',
46         key: 'acciones',
47         width: '80px',
48         render: (_, record) => (
49           <Space size="large">
50             <Link to={`/${sucursalesForm}/${record.id}`}>
51               <Button type="primary" size="small" icon={EditOutlined} /> ghost
52               Editar
53             </Button>
54             </Link>
55
56             <Popconfirm
57               title="¿Desea eliminar esta sucursal?"
58               onConfirm={() => eliminar(record.id)}
59               okText="Si"
60               cancelText="No"
61               icon={DeleteOutlined} style={{color: 'red'}} />
62
63             <Button type="dashed" size="small" icon={DeleteOutlined} /> danger
64             Eliminar
65             </Button>
66           </Popconfirm>
67         </Space>
68       ),
69     ],
70   ];
71
72   let [data, setData] = useState([]);
73   let [filteredData, setFilteredData] = useState([]);
74   let [loading, setLoading] = useState(true);
75
76   useEffect(() => {
77     fetch('https://localhost:44386/api/sucursales', { method: 'GET' })
78       .then(response => response.json())
79       .then(_data => {
80         _data.map(d => { d.key = d.id; return d; })
81         setData(_data);
82         setFilteredData(_data);
83         setLoading(false);
84       });
85   });
86
87   const eliminar = (id) => {
88     setLoading(true);
89   }
90 }

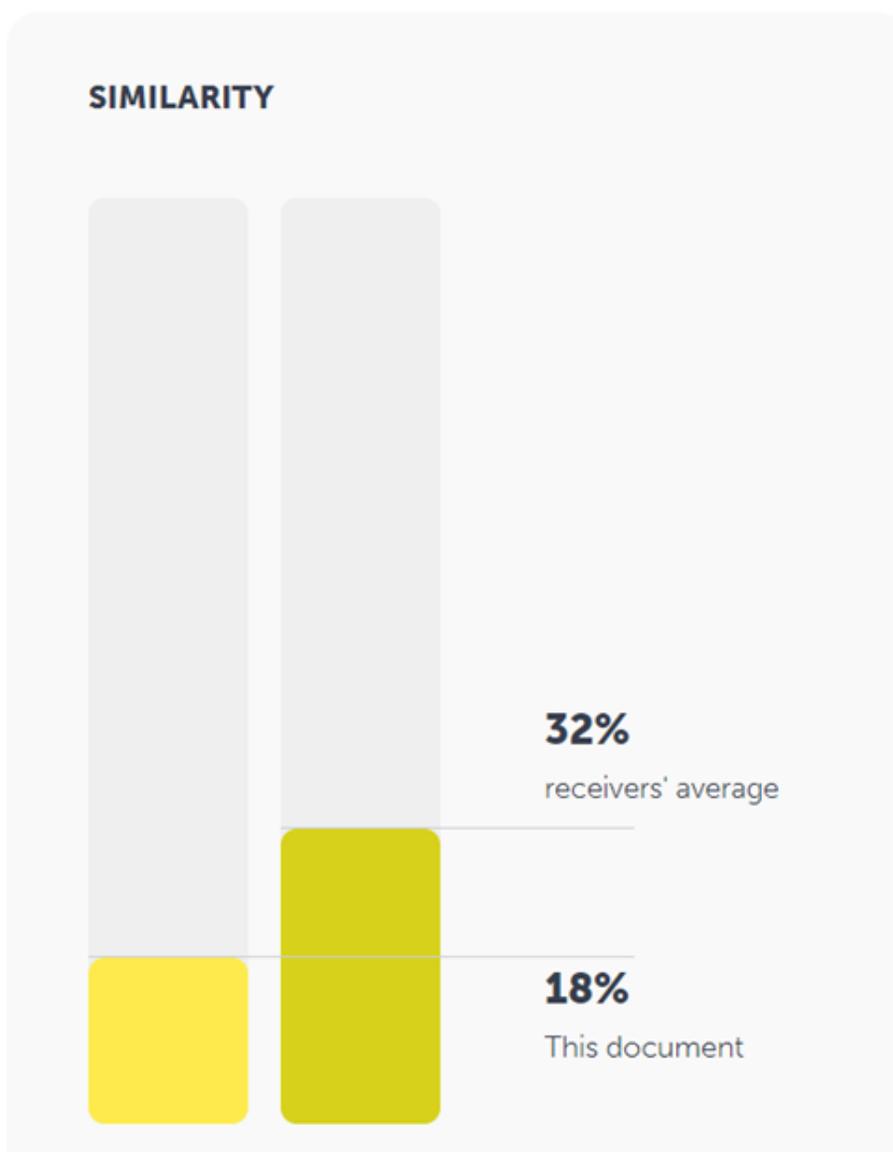
```

```
JS sucursalesList.js X
src > components > sucursal > list > JS sucursalesList.js > SucursalesList > [60] columns > render
84     setLoading(true);
85
86     fetch(`https://localhost:44306/api/sucursales/${id}`, { method: 'DELETE' })
87     .then(response => {
88         if (response.status === 204) {
89             setData(data => data.filter(_data => _data.id !== id));
90             setFilteredData(filteredData => filteredData.filter(_data => _data.id !== id));
91             message.warning('Sucursal eliminada');
92         } else {
93             message.error('Error al eliminar');
94         }
95     })
96     .then(() => {
97         setLoading(false);
98     });
99 }
100
101 const onSearch = (event) => {
102     clearTimeout(searchTimeout);
103
104     searchTimeout = setTimeout(() => {
105         const text = event.target.value;
106
107         setFilteredData(data.filter(_data => {
108             return (
109                 _data.nombre.toLowerCase().includes(text.toLowerCase()) ||
110                 _data.tipo.toLowerCase().includes(text.toLowerCase()) ||
111                 _data.estado.toLowerCase().includes(text.toLowerCase())
112             );
113         }));
114     }, 300);
115 }
116
117 return (
118     <PageContent title="Sucursales" subtitle="Listado">
119         <Row justify="space-between" align="middle" style={{marginBottom: '24px'}}>
120             <Col span={10}>
121                 <Link key="1" to="/sucursalesForm">
122                     <Button type="primary" size="large" icon={PlusCircleFilled} />
123                     Agregar sucursal
124                 </Button>
125             </Link>
126             <Col span={6}>
127                 <Input
128                     placeholder="Buscar..."
129                     onChange={onSearch}
130                     allowClear
131                     suffix={SearchOutlined style={{fontSize: 16, color: '#1890ff'}} />
132             </Input>
133         </Col>
134     </Row>
135
136     <Table columns={columns} dataSource={filteredData} loading={loading} />
137 </PageContent>
138 );
139 }
140
141 export default SucursalesList;
```

## Anexo 6

### Resultados de la revisión anti-plagios

Como se puede notar una parte de los hallazgos tiene que ver con textos de otras tesis asesoradas por el mismo profesor o secciones genéricas donde figuran el nombre de la universidad o contenido de las mismas plantillas.



100%

MATCHING TEXT



Gráfica 12. Representación gráfica de los resultados de la pregunta "Indicar los días picos (los días de mayor concurrencia)

SA

APEC / Tesis Diana\_Winston\_y\_Aquino.pdf

Document: Tesis Diana\_Winston\_y\_Aquino.pdf (D100226225)

Submitted by: jvaldez@unapec.edu.do

Receiver: jvaldez.apec@analysis.arkund.com

76%

MATCHING TEXT



Tipo de Investigación

1.3.1. Investigación Explicativa

Durante el desarrollo del proyecto, estudiaremos a fondo las problemáticas ya expuestas, con la finalidad de profundizar la causa y así obtener un mejor entendimiento de estos y para poder obtener resultados exitosos y concluyentes a la hora de diseñar una solución.

1.3.2. Investigación Descriptiva

En el proceso de elaboración de este trabajo, se utilizará la metodología de investigación descriptiva, mediante la cual definiremos, analizaremos y planteamos las implicaciones más relevantes sobre las problemáticas que enfrentan

76%

MATCHING TEXT



Tipo de Investigación

1.3.1. Investigación Explicativa

Durante el desarrollo del proyecto, estudiaremos a fondo las problemáticas ya expuestas, con la finalidad de profundizar la causa y así obtener un mejor entendimiento de estos y para poder obtener resultados exitosos y concluyentes a la hora de diseñar una solución.

1.3.2. Investigación Descriptiva

En el proceso de elaboración de este trabajo, se utilizará la metodología de investigación descriptiva, mediante la cual definiremos, analizaremos y planteamos las implicaciones más relevantes sobre las problemáticas que enfrentan

78%

MATCHING TEXT



República

Dominicana.

4.1.4. Descripción del Proyecto y Entregables

La implementación del presente proyecto tiene como objetivo ofrecer una

SA

APEC / Trabajo de Grado - Original.pdf

Document: Trabajo de Grado - Original.pdf (D111041694)

Submitted by: jvaldez@unapec.edu.do