

# UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



## FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

### CARRERA DE COMPUTACIÓN

**Tema:** "Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX"

Trabajo de Integración Curricular previo a la obtención del título de Ingenieras en Ciencias de la Computación

**AUTOR:** Martínez Martínez Esteban Wladimir

**TUTOR:** Msc. Jorge Humberto Miranda Realpe

Tulcán, 2023

## CERTIFICADO DEL TUTOR

Certificamos que la estudiante Esteban Wladimir Martínez Martínez con el número de cédula 1004095632 ha desarrollado el trabajo de Integración Curricular: "Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva.



f.....

MSc: Miranda Jorge

**TUTOR**

Tulcán, febrero de 2023

## AUTORÍA DE TRABAJO

El presente trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniera en la Carrera de Computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales.

Yo, Esteban Wladimir Martínez Martínez con cédula de identidad número 1004095632 declaro: que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.



f.....

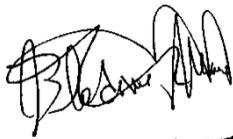
Esteban Wladimir Martínez Martínez

**AUTOR**

Tulcán, febrero de 2023

## **ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

Yo, Esteban Wladimir Martínez Martínez declaro ser autora de los criterios emitidos en el Trabajo de Integración Curricular: "Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX" y se examine expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.



f.....

Esteban Wladimir Martínez Martínez

**AUTOR**

Tulcán, febrero de 2023

## **AGRADECIMIENTO**

A Dios por cada instante de vida que bondadosamente me brinda. A mis padres con su amor, paciencia y compañía han sido un pilar fundamental en mi formación personal y quienes han sabido compartirme desinteresadamente sus valores y fortaleza en cada etapa de mi vida. A mi esposa por el apoyo incondicional, y quienes me ayudan a esforzarme cada día para ser una mejor persona. A mi Tutor de tesis MSc: Jorge Humberto Miranda Realpe quien ha sido una guía para la culminación de la tesis, quien ha sabido compartir generosamente su conocimiento y experiencia profesional.

Esteban Wladimir

## **DEDICATORIA**

Dedico a mis padres en su totalidad quienes con su amor, apoyo y comprensión incondicional estuvieron siempre con una palabra de aliento en tiempos difíciles y que han sido incentivo y pilares fundamentales en mi vida, A Dios por ser mi guía y fortaleza todos estos años, y bendecir mi camino siempre.

Esteban Wladimir

## ÍNDICE

RESUMEN .....	13
ABSTRACT .....	14
INTRODUCCIÓN.....	15
I. EL PROBLEMA .....	16
1.1. PLANTEAMIENTO DEL PROBLEMA.....	16
1.2. FORMULACIÓN DEL PROBLEMA.....	17
1.3. JUSTIFICACIÓN .....	17
1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN.....	18
1.4.1. Objetivo General .....	18
1.4.2. Objetivos Específicos .....	18
1.4.3. Preguntas de Investigación .....	18
II. FUNDAMENTACIÓN TEÓRICA.....	19
2.1. ANTECEDENTES DE LA INVESTIGACIÓN .....	19
2.2. MARCO TEÓRICO .....	20
2.2.1. Automatización de procesos administrativos .....	20
2.2.2 Sistemas de procesamiento de pedidos .....	20
2.2.3. Sistema de ventas.....	21
2.2.4. Sistema Control de Inventario.....	22
2.2.5 Modelos de inventarios .....	23

2.2.6. Ingeniería de software .....	26
2.2.7 Metodologías de desarrollo ágil .....	27
2.2.8 Modelos tradicionales.....	27
2.2.7. Sistemas Informáticos .....	37
2.2.8. Lenguajes de programación.....	38
2.2.9. Bases de datos.....	44
III. METODOLOGÍA.....	47
3.1. ENFOQUE METODOLÓGICO .....	47
3.1.1. Enfoque .....	47
3.1.2. Tipo de Investigación .....	47
3.2. IDEA A DEFENDER .....	48
3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES .....	49
3.4. MÉTODOS UTILIZADOS .....	51
3.5.1. Población .....	51
IV. RESULTADOS Y DISCUSIÓN .....	53
4.1. RESULTADOS .....	53
4.2. PROPUESTA <sup>59</sup>	
V. CONCLUSIONES Y RECOMENDACIONES .....	91
5.1. CONCLUSIONES .....	91
5.2. RECOMENDACIONES .....	92
VI. REFERENCIAS BIBLIOGRÁFICAS .....	93
VII. ANEXOS .....	95



## ÍNDICE DE TABLAS

<b>Tabla 1</b>	Cuadro comparativo de metodologías. ....	28
<b>Tabla 2</b>	Metodologías de desarrollo .....	29
<b>Tabla 3</b>	Gestión en la incertidumbre del proyecto .....	31
<b>Tabla 4</b>	Comparativa de metodologías de desarrollo .....	35
<b>Tabla 5</b>	Lenguaje de programación.....	42
<b>Tabla 6</b>	comparativa de bases de datos relacionales y no relacionales .....	46
<b>Tabla 7</b>	Variable independiente .....	49
<b>Tabla 8</b>	Variable dependiente .....	50
<b>Tabla 9</b>	Recursos de Software .....	62
<b>Tabla 10</b>	Tipo de recursos.....	62
<b>Tabla 11</b>	Factibilidad Económica .....	63
<b>Tabla 12</b>	Identificación de Procesos .....	65
<b>Tabla 14</b>	Diagrama de flujo reparación de contraseña; <b>Error! Marcador no definido.</b>	

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> El modelo en cascada.....	33
<b>Figura 2.</b> Basado en prototipos .....	34
<b>Figura 3.</b> Fases de desarrollo rápido de aplicaciones .....	36
<b>Figura 4.</b> Representación porcentual pregunta 1 .....	53
<b>Figura 5.</b> Representación porcentual pregunta 2.....	54
<b>Figura 6.</b> Representación porcentual pregunta 3.....	54
<b>Figura 7.</b> Representación porcentual pregunta 4.....	55
<b>Figura 8.</b> Representación porcentual pregunta 5.....	56
<b>Figura 9.</b> Representación porcentual pregunta 6.....	56
<b>Figura 10.</b> Representación porcentual pregunta 7 .....	57
<b>Figura 11.</b> Representación porcentual pregunta 8.....	57
<b>Figura 12.</b> Representación porcentual pregunta 9 .....	58
<b>Figura 13.</b> Representación porcentual pregunta 10.....	58
<b>Figura 14.</b> Representación porcentual pregunta 11 .....	59
<b>Figura 15.</b> Organigrama .....	61
<b>Figura 16.</b> Diagrama de flujo ingreso al sistema.....	66
<b>Figura 17.</b> Diagrama de flujo reparación de contraseña .....	66
<b>Figura 18.</b> Diagrama de flujo registro de ventas .....	67
<b>Figura 19.</b> Diagrama de flujo registro de ventas .....	67
<b>Figura 20.</b> Diagrama del flujo registro de cliente .....	68
<b>Figura 21.</b> Diagrama de flujo registro de categorías .....	68
<b>Figura 22.</b> Diagrama de flujo de artículos.....	69
<b>Figura 23.</b> Diagrama de flujo registro de proveedores.....	69
<b>Figura 24.</b> Diagrama de flujos registro de ingreso .....	70
<b>Figura 25.</b> Diagrama de flujo consulta de ventas .....	70
<b>Figura 26.</b> Diagrama de flujo consulta de ingresos.....	71
<b>Figura 27.</b> Diagrama de flujo clientes.....	71
<b>Figura 28.</b> Diagrama de flujo clientes.....	72
<b>Figura 29.</b> Caso de uso Usuario de sistema.....	72
<b>Figura 30.</b> Caso de uso Usuario de sistema.....	73
<b>Figura 31.</b> Caso de uso categorías.....	73
<b>Figura 32.</b> Caso de uso productos.....	74

<b>Figura 33.</b> Caso de uso proveedores.....	<b>¡Error! Marcador no definido.</b>
<b>Figura 34.</b> Caso de uso ingresos .....	75
<b>Figura 35.</b> Caso de uso consultas compras.....	75
<b>Figura 36.</b> Caso de uso consulta ingreso .....	76
<b>Figura 37.</b> Caso de uso barras .....	76
<b>Figura 38.</b> Diseño de base de datos .....	77
<b>Figura 39.</b> Login principal .....	77
<b>Figura 40.</b> Menú de opciones.....	78
<b>Figura 41.</b> Inicio de sistema .....	78
<b>Figura 42.</b> Lista categorías .....	79
<b>Figura 43.</b> Registro de productos .....	79
<b>Figura 44.</b> Interfaz de crear ingresos. ....	80
<b>Figura 45.</b> Interfaz de proveedores .....	80
<b>Figura 46.</b> formulario ventas.....	81
<b>Figura 47.</b> Interfaz de factura. ....	81
<b>Figura 48.</b> Componente inicio de sesión .....	82
<b>Figura 49.</b> Componente menú .....	82
<b>Figura 50.</b> Componente menú .....	82
<b>Figura 51.</b> Componente categoría.....	83
<b>Figura 52.</b> Componente productos .....	83
<b>Figura 53.</b> Componentes ventas .....	84
<b>Figura 54.</b> Componente proveedores .....	84
<b>Figura 55.</b> Componentes venta .....	85
<b>Figura 56.</b> Componentes venta .....	85
<b>Figura 57.</b> Fase de transición .....	86
<b>Figura 58.</b> Lista de productos con su stock y su respectivo semáforo .....	86
<b>Figura 59.</b> Existencias del producto a vender .....	87
<b>Figura 60.</b> Alerta en el caso que intente ejecutar.....	87
<b>Figura 61.</b> Base imponible de impuesto el %.....	88
<b>Figura 62.</b> Configuración para manejar los impuestos y el porcentaje.....	88

## ÍNDICE DE ANEXOS

<b>Anexo 1.</b> Informe de Turnitin.....	95
<b>Anexo 2.</b> Certificado de Abstrac .....	96
<b>Anexo 3.</b> Certificado del Abstrac .....	97
<b>Anexo 4.</b> Acta de predefensa.....	98
<b>Anexo 5.</b> Rubrica de la predefensa .....	99
<b>Anexo 6.</b> Certificado de cumplimiento .....	100

## RESUMEN

La presente investigación con el tema "Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX", se realizó un amplio estudio de los procesos de gestión de inventarios y ventas, donde el principal objetivo fue desarrollar una de aplicación informática para el proceso de control de inventarios y ventas, el cual sea capaz de dar cumplimiento con este propósito se utilizó un enfoque mixto conjuntamente con la investigación de campo, exploratoria, descriptiva y documental, mismos que contribuyeron a la recolección de información acerca de los procedimientos que se siguen para el registro y control de inventarios y ventas, una encuesta dirigida a los trabajadores de la empresa y una entrevista dirigida a la propietaria de la empresa que permitieron conocer la información de forma detallada e identificar factores importantes sobre el manejo y los problemas principales que se presentan al momento de realizar el proceso de inventarios y ventas dentro de la empresa, orientando de manera exitosa el contexto de estudio con el fin de ofrecer una solución al problema planteado. Con los resultados obtenidos se logró desarrollar un prototipo de aplicación informática para la gestión de inventarios enfocado al entorno web, para su desarrollo se utilizó la metodología RAD, gracias a la flexibilidad de sus fases se utilizó los componentes e instrumentos que aportaban más información, la programación fue realizada JavaScript la arquitectura cliente servidor para guardar la información MongoDB para realizar la interfaz.

**Palabras claves:** Inventario, MongoDB Aplicación informática, P, Herramienta, JavaScript

## ABSTRACT

The present investigation is titled "Development of a computer system for the Inventory control of the company RIVATEX"; carried out an extensive study of the inventory and sales management processes, in which the main objective was to develop a computer application for the control process. Therefore, to fulfill this purpose, a mixed approach was applied together with field, exploratory, descriptive, and documentary research, which contributed to the collection of information about the procedures followed for the registration and Inventory control. Moreover, a survey was applied to the workers of the company, and an interview for the owner of the company allowed knowing the information in detail and identifying factors about the management and the main problems that arise at the time of carrying out the process of inventories and sales within the company, successfully guiding the study context to offer a solution to the problem. Based on the results, a prototype of a computer application for inventory management focused on the web environment was developed. Due to the flexibility of its phases, the RAD methodology was used for its development. Likewise, components and instruments that provided more information were incorporated; programming was performed using JavaScript server-client architecture to save MongoDB data.

**Keywords:** Inventory, MongoDB, Application Software, Tool, JavaScript.

## INTRODUCCIÓN

Actualmente las empresas buscan resaltar y mantenerse en un mercado cada vez más exigente y competitivo, teniendo que adaptarse a las necesidades de la empresa. El desarrollar un sistema informático de procesos de venta e inventario de la empresa logren maximizar el servicio ofrecido por los clientes.

El sistema de venta e inventario busca mantener el correcto almacenamiento, registro eficiente y el precio de los productos. Se decidió realizar esta investigación ya que la empresa RIVATEX, tiene varias bodegas para almacenamiento de productos, teniendo en cuenta que la empresa despacha grandes cantidades de productos por día, se observó la necesidad de desarrollar un sistema informático para las medidas necesarias para mejorar el servicio de la empresa.

Este proyecto busca establecer una correcta administración, venta y control de inventarios, además de contar con la información correcta de los inventarios, agilizar el flujo de productos terminado, detectar errores de los inventarios, detectar posibles robos en la entrada y salida de productos.

Con el desarrollo de este sistema informático se pretende organizar los productos de la empresa RIVATEX, generando satisfacción en los clientes internos y externos garantizando la calidad de los productos y su despacho.

## **I. EL PROBLEMA**

### **1.1. PLANTEAMIENTO DEL PROBLEMA**

En la actualidad existen empresas donde se presentaron inconvenientes al momento de realizar el proceso de inventario de bienes disponibles, como en el caso presentado en la investigación que se realizó en Bogotá- Colombia en la empresa comercializadora de Frutas El Puerto Ltda., donde realiza sus procesos de forma manual (ventas e inventarios) lo que produce un tiempo prolongado en el registro de información y puede generar problemas de inconsistencias entre los cálculos realizados e inventarios reales, sin mencionar el gasto monetario que genera llevar los procesos a través de papel y lápiz. (Ramírez, 2017).

A nivel nacional, específicamente en la ciudad de Ambato se realizó implementación de un sistema de inventarios comercializadora Arturo CALLE S.A.S. se dónde se determinó la inexistencia de un registro de inventarios actualizado, el cual provocaba, dificultad a la hora de buscar información de un dispositivo, ya que se debe buscar en varios documentos y además se han presentado pérdidas de datos en varias oportunidades. Otros problemas que se encuentran actualmente es que no se guarda un registro de los lugares donde ha sido asignado un dispositivo, sólo se almacena el lugar donde se encuentra actualmente, además existen riesgos de guardar información errónea, duplicar valores, asignaciones en ubicaciones inexistentes de la empresa, entre otros inconvenientes que se presentan por no utilizar una herramienta adecuada. (Peláez, 2017).

La empresa RIVATEX es una industria especializada en compra y venta de prendas para uso comercial ubicada en la ciudad de Quito Provincia de Pichincha, La situación problemática que presenta en la actualidad es que no cuenta con un control de inventario y registro venta adecuado en el negocio. Provocando con ello, dificultad al momento de realizar la búsqueda de información de alguna prenda, demora al entregar el producto, errores al escribir, incluso en ocasiones se presentaba pérdidas de datos, no se puede saber con exactitud qué cantidad de ventas diarias realizadas



Los procesos que se emplean actualmente en la empresa se manejan a través de hojas físicas, mismas que disminuyen de forma exponencial sobre control sobre la mercadería; desorganización que, por un lado, pérdidas económicas totales; y, por otro lado, insatisfacción en los clientes por demora en el despacho de las prendas al realizar el proceso de manera manual.

## **1.2. FORMULACIÓN DEL PROBLEMA**

¿De qué modo ayuda el desarrollo de un sistema informático para la gestión de procesos de venta e inventario de la empresa RIVATEX?

## **1.3. JUSTIFICACIÓN**

Existen algunos elementos que impiden que las empresas utilicen eficientemente los sistemas de gestión con los que cuentan, algunos de ellos son: falta de tiempo para utilizar el sistema, falta de habilidades en el manejo el sistema, falta de soporte funcional y técnico especializado cuando se presenta un problema con el uso del sistema. (Barco y Jiménez, 2017, p.16)

El presente proyecto de investigación pretende sistematizar los procesos de ventas e inventario a través de una propuesta tecnológica que brinde información actualizada para que los usuarios accedan a ella mediante una aplicación y puedan realizar sus ventas con el fin de evitar demora, confusiones e insatisfacción en los clientes.

La investigación es importante porque se desarrolló una solución tecnológica como alternativa de mejora en los procesos que realiza la empresa y de esta manera lograr que el propietario tenga un conocimiento exacto tanto de los insumos que se tiene en stock como de las ganancias diarias, semanales o mensuales.

Como beneficiarios directos de esta investigación se tiene a la propietaria y empleados los cuales serán los que hagan uso del sistema y como beneficiarios indirectos están los clientes ya que al momento de hacer uso de los servicios que presta la empresa estos recibirán una mejor atención.

Finalmente, el trabajo de investigación es factible porque la recolección de información fue asequible, la dueña de la empresa estuvo a completa disponibilidad del investigador para la aplicación de una entrevista y la definición de requerimientos de software que sirvan para el desarrollo del sistema.

## **1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN**

### **1.4.1. Objetivo General**

“Desarrollar un sistema informático para los procesos de venta e inventario en la empresa RIVATEX

### **1.4.2. Objetivos Específicos**

- Fundamentar bibliográficamente la actual investigación a través de medios virtuales y físicos para sustentar teóricamente la investigación.
- Elaborar un marco metodológico para la investigación de herramientas tecnológicas afines para procesos de ventas e inventario
- Determinar los procesos de control de ventas e inventarios para el desarrollo del sistema informático.
- Establecer una propuesta de un sistema informático para la gestión de venta e inventario en la empresa "RIVATEX" en la ciudad de Quito.

### **1.4.3. Preguntas de Investigación**

- ¿Cómo la fundamentación bibliográfica ayudará a sustentar teóricamente la investigación?
- ¿La elaboración del marco metodológico permitirá obtener información de los procesos ventas e inventario para el desarrollo del sistema informático?
- ¿Cómo se puede determinar los requerimientos relacionados para la gestión de inventario?
- ¿Cómo la propuesta de una solución informática ayudará a la gestión de ventas e inventario?

## II. FUNDAMENTACIÓN TEÓRICA

### 2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Para reforzar el estudio se ha seleccionado las investigaciones con mayor relevancia para el tema planteado, estos trabajos fueron extraídos de revistas indexadas y repositorios digitales de instituciones de educación superior.

Como primer antecedente se tiene un estudio realizado por (Colin Scully, 2017), cuyo tema es "A computer system for inventory management of lighting products", esta investigación describe los problemas y beneficios involucrados en la implementación de un sistema de control de inventario por computadora en una empresa de fabricación de productos de iluminación obteniendo como resultado que el uso del sistema permite facilitar al personal la realización de las diferentes actividades evitando de esta manera ocasionar retrasos en la ejecución de las diferentes operaciones.

Como segundo antecedente se tiene un estudio realizado por (Martínez y Rocha, 2019) titulado "Implementación de un sistema de control de inventario en la Ferretería Benjumea y Benjumea ubicada en el municipio de Cerete Córdoba", esta investigación tiene como objetivo principal el poder brindar solución a los principales problemas que se presentan en dicha empresa mediante la implementación de un sistema informático para el control de inventarios permitiendo minimizar gastos y brindar una mejor atención.

Por otro lado (Estrada y Steffano, 2017), afirma: En su trabajo de investigación titulado "Estándar de Procedimientos como Mejora de Control Interno del Inventario de una Empresa" plantea como objetivo general el diseño de patrones de procedimientos para optimizar el control de inventario usados en una empresa importadora. En este estudio mencionaremos la importancia de obtener la información relacionada con el manejo de control de inventario llevado por la empresa: desarrollar una propuesta de control de inventario, para el manejo de las compras de materia prima. Este estudio favorece a la investigación planteada,

proporcionando conocimientos sobre el almacenaje adecuado en las bodegas, el tratamiento de alimentos perecibles, el método de valoración de las salidas de inventarios.

## **2.2. MARCO TEÓRICO**

### **2.2.1. Automatización de procesos administrativos**

La automatización de procesos actualmente juega un papel sumamente importante en las compañías, ya que la administración del conocimiento y la gestión tecnológica influyen cada vez más en el nivel competitivo que tienen. En el entorno global que vivimos es más importante que nunca para las organizaciones simplificar las operaciones para que los gerentes y directivos se puedan enfocar más tiempo en actividades que generen ingresos. Sin importar si se trata de una mediana empresa o un gran corporativo, el definir las actividades por roles y áreas para automatizar los procesos, permite hacer más eficiente la operación logrando así reducir costos. (Zapata, 2019)

La automatización de procesos administrativos es el uso de sistemas que busca hacer más fácil, efectivo y eficiente el funcionamiento de la empresa RIVATEX. En general todas las acciones que realiza una persona adentro de una organización tienen como soporte un proceso, que puede o no estar formalizado, indicando cómo fluye la información dentro de la misma.

### **2.2.2 Sistemas de procesamiento de pedidos**

Pérez (2019) afirma: Los sistemas de procesamiento de pedidos implican la configuración de ventas, planeación y ejecución del embarque, el control de inventarios, facturación, la interacción con el cliente y la asignación y programación de rutas. Los procesos de negocios respaldados por estos sistemas son tan esenciales para la operación de la empresa que en ocasiones a los sistemas de procesamiento de pedidos se les conoce como la sangre vital de la organización. El sistema de gestión de pedidos es una solución digital que permite el control y seguimiento en tiempo real de la información y los procesos de gestión de pedidos. Esto es gracias a una plataforma que tiene una visión amplia en áreas como inventarios ventas y pedidos

### **2.2.3. Sistema de ventas**

Pérez (2019) afirma: Introduce los datos básicos necesarios para proceder el pedido de un cliente. Los pedidos pueden llegar a través del correo o mediante un sistema por teléfono a un grupo de vendedores, llegar a través de transacciones de intercambio electrónico de datos, directamente de la computadora de un cliente o de una red de área amplia, o el cliente lo puede ordenar de forma directa por internet mediante una forma de captura de datos en el sitio Web. (p.23.)

Las escalas de calidad de servicios electrónicos existentes miden principalmente el comportamiento de compra electrónica. Como resultado, estas escalas no cubren todas las facetas de la evaluación de la calidad de un consumidor. Aplicamos un modelo de proceso de transacción a las interacciones de servicios electrónicos para incluir características de calidad de servicio electrónico. Construimos una escala basada en el proceso de transacciones para medir la calidad del servicio con base en este marco fundamental que captura todas las etapas del proceso de prestación de servicios electrónicos.

Para tener éxito financiero, debe adquirir nuevos consumidores y retener a los antiguos. Se dice comúnmente que cuando los clientes tienen una experiencia positiva con una empresa, se lo dicen a una o dos personas más, sin embargo, cuando tienen una experiencia negativa, pueden decírselo a decenas o incluso a cientos de personas.

Algunas empresas emplean un sistema de interacción con el cliente para rastrear y registrar las interacciones del cliente con la organización a fin de mantener felices a los clientes existentes. Estos sistemas están diseñados para aumentar la lealtad del consumidor.

Los controles de ventas son simplemente los procedimientos que sigue un gerente de la empresa RIVATEX para asegurarse de que su equipo esté haciendo las cosas correctas en el momento adecuado. Estos sistemas de control mejoran el desempeño y el bienestar de los representantes de ventas. Hay dos tipos de controles de ventas: basados en el comportamiento y basados en resultados.

El control de ventas e inventarios comprende el proceso de administrar las mercancías que se guardan en almacén, con el objetivo de obtener información de la entrada y salida de productos, buscando el ahorro de costes.

## **Facturación**

En la facturación, se lleva un control de todos los ingresos y egresos financieros de una empresa. Se realiza un seguimiento de todos los movimientos de dinero, tanto de entradas como de salidas, para así poder llevar un control preciso de las finanzas de la empresa. La facturación permite llevar un registro de todas las operaciones económicas realizadas por la empresa, lo cual es muy útil a la hora de hacer un análisis de los ingresos y egresos, así como de las ganancias y pérdidas. De esta manera define que existe dos métodos de facturación el primero mediante métodos tradicionales donde se suele utilizar una gran cantidad.

Las facturas deben ser de acuerdo con la solicitud o necesidad del comprador, además, la información debe ser clara donde los datos se vean de forma perfecta y clara, actualizada según la normativa tributaria, se deben realizar facturas electrónicas porque beneficia a ambos, tanto a la empresa como al comprador, debemos detallar las especificaciones requeridas por el comprador si desea emitir la factura en el momento de la entrega, de acuerdo con los pedidos o 27 cuentas por cobrar y al final del mes de la factura. Deben evaluarse las comodidades del comprador. (Rodríguez, 2017)

### **2.2.4. Sistema Control de Inventario**

Pérez (2019) afirma: Para cada artículo tomado, se crea una transacción y se envía al sistema de control de inventario, junto con el número de referencia y el monto proporcionado. Esto cambia los registros de existencias para reflejar el número real de cada artículo de existencias que está disponible. Como resultado, cuando los empleados de entrada de pedidos evalúan el nivel de un producto, tienen la información más actualizada. (p.26-27.)

La automatización de procesos administrativos es la aplicación de un grupo de herramientas de software y hardware, que conforman un sistema de información buscando aumentar su eficiencia con una información actualizada y disponible a todo momento desde cualquier parte del mundo.

El procedimiento de control de inventario se dificulta cuando hay un número significativo de artículos. Para resolver este desafío, las empresas RIVATEX dividen sus registros en algunas categorías y aplican procedimientos de control y comparaciones de productos de cada grupo. Se han propuesto muchos enfoques de agrupación en esta área.

El control de inventario es el proceso sistemático de monitorear y regular los niveles de inventario para garantizar que las empresas tengan el nivel correcto de existencias disponibles para satisfacer la demanda de los clientes. El control efectivo del inventario puede ayudar a la empresa RIVATEX a evitar los desabastecimientos, minimizar el exceso de inventario y optimizar los niveles de existencias para reducir los costos.

Según (Ramírez y Ramírez, 2018) la importancia del control de inventarios radica en que permite a las organizaciones y a los administradores medir y evaluar el rendimiento de las actividades y de los procesos, también permite a las organizaciones y a los administradores corregir las actividades y procesos que no están funcionando de la manera prevista.

#### Importancia de inventarios

La teoría de inventarios se deriva del modelo de cantidad financiera de pedido (conocido por sus siglas EOQ). Actualmente, proporciona una aproximación de la estrategia de inventario óptima en varios escenarios del mundo real. Sin embargo, en el caso de un sistema de almacenamiento degradado con un impacto económico significativo, la instalación de productos con una vida útil infinita puede conducir a políticas de almacenamiento óptimas. (Asencio, et. al 2017, p. 4)

#### **2.2.5 Modelos de inventarios**

**Modelo ABC.** - Para muchos el ABC es considerado como un sistema, pero en realidad se trata de un método de costeo de la producción basado en las actividades realizadas en su elaboración. El método se diseñó para las empresas de manufactura, sin embargo, gracias a los buenos resultados en el manejo de este su aplicación se extiende a las empresas de servicios. (Allauca, 2018).

El ABC es un sistema de costos basado en actividades que permite clasificar los productos o artículos para fijarles un determinado nivel de control de existencias y reducir los costos en el manejo de inventarios. Es una aplicación del principio de Pareto que manifiesta que hay unos pocos críticos y un gran número de triviales. El sistema ABC clasifica los artículos del inventario en tres grupos, en este caso la empresa RIVATEX hace uso del modelo ABC ya que clasifica los productos dependiendo del movimiento mensual.

**Modelo JUST IN TIME.** -El objetivo, en este caso es reducir o eliminar en gran medida el inventario requerido en un proceso de producción. Es un sistema en el que se dispone de los inventarios sólo en los momentos en que se necesitan. Condiciones:

1. El proceso de producción es repetitivo se produce un mismo producto una y otra vez.
2. Se puede controlar la escasez de insumos para la producción, con continuidad en el trabajo. Ello es debido al diseño de la producción; permite tener siempre disponible el requerimiento necesario.
3. El proveedor cumple a tiempo en la entrega
4. Se aplica una administración con calidad total, tal que las partes que llegan de los proveedores y que salen de una estación de trabajo a otra funcionan según lo especificado.
5. La demanda del producto final terminado jala las demandas de las demás partes. En contraste, cuando las partes individuales se conforman como inventarios de trabajo en proceso.

**Modelos Probabilísticos.** -Para trabajar estos modelos se requieren los siguientes supuestos:

- La demanda no es constante.
- La demanda se puede definir por una distribución de probabilidad.
- No son posibles los descuentos por cantidad.
- Se evitan las roturas de inventario.
- Mantener un nivel de servicio adecuado frente a la incertidumbre de la demanda.
- El nivel de servicio es el complemento de la probabilidad que se agoten las existencias.

Es necesario conocer las siguientes variables a trabajar:

- $d$  = Tasa de demanda diaria.
- $L$  = Plazo de entrega del pedido.
- $p$  = Tasa de producción diaria.
- ROP = Punto de pedido o reorden.
- CT = Costo total por unidad de tiempo.
- $\mu$  = Demanda media.
- $\sigma$  = Desviación estándar.



- $Z$  = Distribución normal.

**Modelo de compra sin déficit.** -Para trabajar este modelo se supone una tasa de producción continua, lo cual permite hacer una reposición del inventario constante durante el tiempo de producción. En este modelo en particular, se deduce que el artículo no será producido sino comprado o que se necesita un material auxiliar utilizado en la producción, pero este elemento es comprado. (Arauka, 2018). Este modelo es también conocido como modelo de cantidad de pedido o lote económicos (EOQ); es uno de los modelos de inventario más antiguo y conocido, y está basado en las siguientes hipótesis:

- La demanda es constante y conocida.
- El plazo de entrega es constante y conocido.
- El pedido llega en un solo lote y todo de una vez.
- Los costos por ordenar un pedido y los costos de mantenimiento son constantes y conocidos.
- No son posibles los descuentos por cantidad.
- Se evitan las roturas de inventario.
- No se permite diferir demanda al futuro.

**Modelo de compra con déficit.** -Para trabajar este modelo se requieren los siguientes supuestos:

Según Mata, (2019). mencionó que los siguientes supuestos son necesarios para asegurar el normal funcionamiento de este modelo hipotético:

- La demanda es constante y conocida.
- La tasa de producción es constante y conocida.
- El pedido llega en un sólo lote y todo de una vez.
- Los costos por ordenar un pedido, los costos de mantenimiento y los costos de penalización y fijos son constantes y conocidos.
- No son posibles los descuentos por cantidad.
- Se permite diferir demanda al futuro.
- La reposición del inventario se hace instantáneamente.

Es necesario conocer las siguientes variables a trabajar:

- $Q$  = Cantidad óptima a comprar por pedido.
- $D$  = Demanda por unidad de tiempo.
- $C_o$  = Costo por ordenar el pedido.

- Cm = Costo de mantener una unidad por año.
- CTO = Costo total por ordenar un pedido.
- CTM = Costo total de mantenimiento.
- Cv = Costo variable por unidad.
- Cp = Costo unitario de penalización por unidad de tiempo.
- d = Tasa de demanda diaria.
- p = Tasa de producción diaria.
- Ct = Costo total promedio por unidad de tiempo.
- CT = Costo total por unidad de tiempo.

La cantidad óptima para pedir se obtiene de la siguiente formula:

$$Q = \sqrt{(2DCo Cp + Cm)/(CmCp)}$$

### **2.2.6. Ingeniería de software**

Es un área de la informática que detalla métodos y técnicas para el desarrollo y mantenimiento de software o programas informáticos que brindan una solución a problemas de todo tipo para satisfacer las necesidades de los usuarios. (Braude y Bernstein, 2017, p.2)

La ingeniería de software incorpora el bosquejo del proyecto en la empresa RIVATEX, el desarrollo del software, las pruebas y posterior poner en funcionamiento del sistema. El desarrollo de software requiere de un proceso planeado y estandarizado para generar un producto de alta calidad tanto en la documentación como en el aplicativo final.

Para desarrollar un software y usar en una institución o empresas no es un trabajo fácil. El objetivo de un buen software es que posea una mayor efectividad en cuanto a costos y eficaz al momento del manejo de este, optimizando recursos, para ello existen técnicas que inciden en distintas dimensiones para el proceso de desarrollo. En el proceso se manejan diferentes modelos para definir un conjunto de acciones, métodos, prácticas y modificaciones, que los desarrolladores utilizan al construir y mantener software Maldonado, A., Margain, L., F, Álvarez, F., y Benítez, E., (2018).

García, F. (2018) muestra que un modelo de desarrollo de software es una representación indeterminada de las actividades a realizar en el desarrollo. Para esto se han establecido metodologías y modelos para desarrollar un software en

un tiempo determinado y recursos que dispone una empresa o entidad. Estos procesos poseen una fase de requisitos.

En ese sentido, en el campo de la informática se puede sostener que la ingeniería de software define técnicas y procedimientos para el desarrollo y la conservación de programas informáticos orientados a brindar soluciones a diferentes inconvenientes para satisfacer las necesidades de las partes interesadas.

La visión del desarrollo de software como actividad de programación, teniendo en cuenta el análisis y diseño previo, así como otras actividades de integración y validación posteriores. La distribución de series temporales de todas estas actividades se denomina ciclo de desarrollo de la aplicación.

### **2.2.7 Metodologías de desarrollo ágil**

Esta metodología es un grupo de métodos y procedimientos que permiten la ejecución justa e inaugurada de cada una de las ocupaciones en el periodo de vida de un plan de desarrollo. Estos métodos son combinaciones de modelos de procesos comunes, es un enfoque metódico para administrar, administrar y ejecutar un proyecto para lograr el éxito. (Maida y Pacienza, 2015, p. 12).

Los métodos ágiles, pueden adaptarse al entorno de cada equipo o plan que se dividen en proyectos más pequeños según a una nómina organizada de cualidades. Cada proyecto se trata de forma autónoma y formar un subgrupo de funciones en un breve ciclo de 2 a 6 semanas.

### **2.2.8 Modelos tradicionales**

Este modelo de desarrollo suministra documentación detallada del proyecto, incluido el control y la planificación, y describe una metodología engorrosa debido a la gran cantidad de actividades y fases para cumplir con los objetivos contables. Plan de trabajo para la primera fase del proyecto. En el momento del desarrollo del software.

**Tabla 1** Cuadro comparativo de metodologías.

Agiles	Tradicionales
Están dispuestas para efectuar modificaciones	Renuencia a hacer cambios
Se elaboran en pequeños grupos	Se ejecutan en grandes grupos
Posee escasos principios, poco controlado	Proceso controlado y muchas normas
Pocos roles	Tienen más roles
desarrollos adaptables	Proceso rígido
Interacción con el cliente	Poca comunicación con el cliente
Orientado a proyectos pequeños	Orientados a proyectos de cualquier tamaño.

El esquema de metodologías agiles permite ejecutar de manera adaptable, flexible y tener constante interacción con el cliente, permitiendo la elaboración de un proyecto exitoso, afirmando a corto plazo el desarrollo de proyectos.

### **Metodología RAD**

Originalmente propuesto por Martín y conocido en la comunidad de ingeniería de software como RAD o Rapid Application, este enfoque involucra a un pequeño grupo de programadores con herramientas que generan automáticamente código de sintaxis de alto nivel como entrada.

**Tabla 2** Metodologías de desarrollo

Criterio	RAD	SCRUM	XP
Características	Permite el Desarrollo interactivo, realización de prototipos	Se basan en un proceso incremental	Basado en flexibilidad y adaptabilidad
Revisión	Cuando se finaliza el proceso y permite reutilizar funciones ya desarrolladas se realizan las pruebas.	Necesitan de revisión diarias y se planifica el trabajo	Se debe integrar mínimo una vez al día
Objetivo	Aplicación en menor tiempo y bajo costo.	Indicado para proyectos complejos y obtener resultados pronto.	Trabajo en grupo
Equipos	Interacción constante con el cliente y programadores completos.	Se dividen las actividades entre los integrantes del equipo	Programación en parejas

**Fuente:**( Rodriguez,2021)

Para los autores (Singh y Peláez, 2020), el uso de este tipo de metodologías ofrece algunas ventajas:

- Amplia la flexibilidad, de que los programadores realicen los cambios que sean precisos de manera inmediata en el desarrollo.
- Estas interacciones acortan el periodo de desarrollo al tiempo que agilizan la entrega.

- Reduce la probabilidad de cometer errores, ya que fomenta la reutilización de códigos y con ello minimiza la programación manual.
- Incrementa la proporción de complacencia de los clientes producto de su elevado nivel de colaboración y coordinación entre partes; es decir, desarrolladores-clientes-usuarios.
- Limita la probabilidad de factores sorpresas producto de la temprana entre las fases del desarrollo del software.

La metodología ágiles, son elegidas aquellas de Rápida Aplicación (RAD), este modelo me permite satisfacer las necesidades del cliente (empresa RIVATEX), estar en constante comunicación con la propietaria y empleado, así mismo me posibilita ahorrar tiempo y recursos en el desarrollo del programa y obtener un aplicativo eficiente y de alta calidad.

El tipo de desarrollo rápido de aplicaciones corresponde a una técnica para desarrollo de software que prioriza las entregas e interacciones rápida.

**Tabla 3** Gestión en la incertidumbre del proyecto

<b>Enfoque</b>	<b>Modelos tradicionales</b>	<b>Modelos ágiles</b>
Inicio del proyecto	Se precisa todo a partir del inicio del proyecto	Reestructurar código fuente
Compromiso	El beneficiario (cliente) delega su compromiso	Colaboración e iteraciones
Acuerdo del proyecto	Contratos exactos	Método justo a tiempo
Apertura del proyecto	Responsabilidad asignada	Responsabilidad adquirida del cliente y el compromiso
Dimensión del proyecto	Grandes	Pequeños
Énfasis	Orientada a procesos	Orientada a personas
Ciclos	Existencia de Roles	Carencia de roles
<b>Gestión del cambio del proyecto</b>		
Habilidad del proyecto	Se resiste al cambio de técnicas	Los métodos pueden cambiar (cliente)
Comprensión del proyecto	Deficiencia en la retroalimentación ante dificultades	Aprendizaje continuo
Cumplimiento del proyecto	Desacuerdo entre lo solicitado	Se desarrolla en base a las necesidades.
Competencia	Inexperiencia en la situación real del mercado	Se añade un valor extra.

### **Gestión de los equipos**

Evaluación del proyecto	Evaluación individual	Flujo de valor continuo
Planificación	Escasa motivación del proyecto	El cliente está implicado en el proyecto
Conocimiento	Personal con conocimientos básicos	Compromiso de personal
Equipo de trabajo	Grandes equipos. (2 o más personas)	Pequeños equipos. (1 persona)

---

**Fuente:**( Pillago,2020)

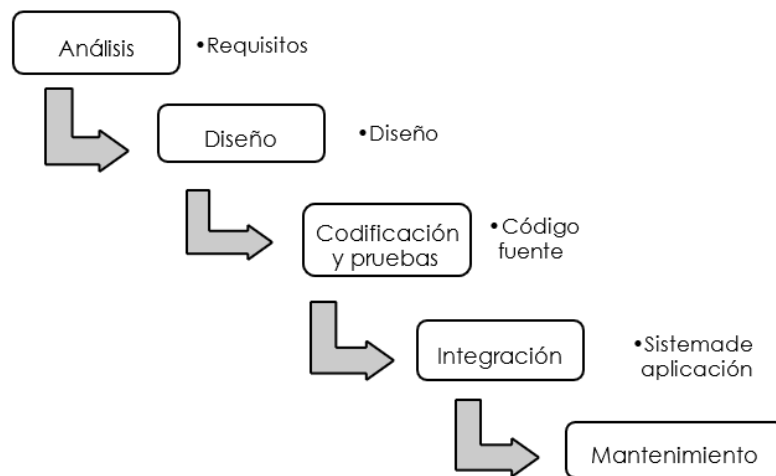


Maida, E., y Paciencia, J. (2017) manifiestan que los modelos habituales son denominados metodologías pesadas por el gran número de actividades y etapas al momento de desarrollar el software, este modelo lleva una documentación absoluta del proyecto la planificación y el control al igual para llegar a desempeñar el objetivo en la fase inicial del proyecto. Asimismo, los modelos tradicionales no se acomodan a los cambios por los requisitos que no puede predecir. Estos modelos son:

- Cascada
- Basado en prototipos
- Modelo de desarrollo rápido de aplicaciones (RAD)

El modelo en cascada Zumba, J. (2018) indica que es un modelo de proceso secuencial en donde desempeña un conjunto de etapas que se obtiene una tras otra. Para el desarrollo del software se debe cumplir las fases siguiendo un flujo de ejecución de arriba hacia abajo como una cascada.

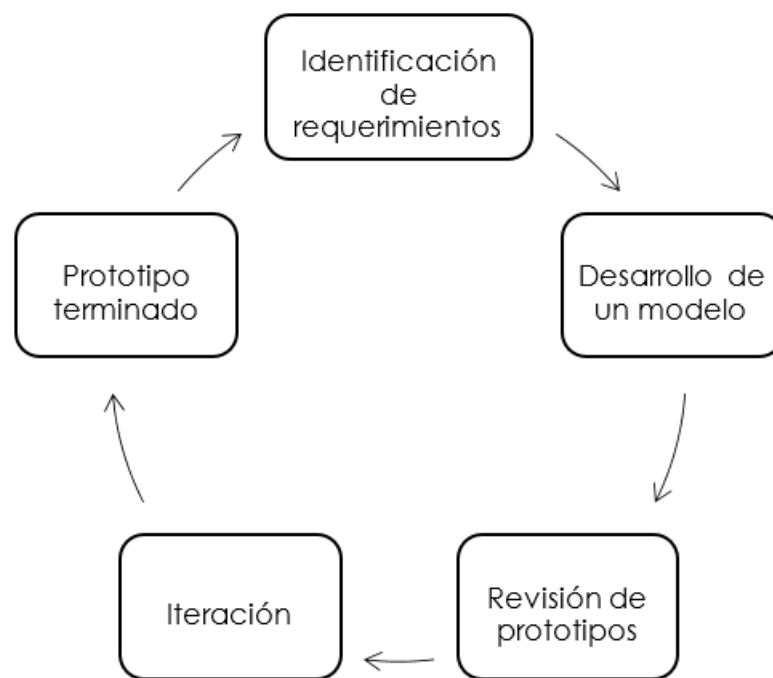
#### Fases del modelo en cascada



**Figura 1.** El modelo en cascada

**Fuente:** (Sandoval, 2020)

Basado en prototipos conocido como modelo de desarrollo tradicional o cascada, se identifica los requerimientos de la empresa RIVATEX mediante la construcción de un prototipo para exponer al cliente, obtener críticas y retroalimentación con el propósito de mejorar hasta que el cliente este satisfecho, el diseño del prototipo va evolucionando. Para la construcción del prototipo se debe realizarlo en corto tiempo, usando los programas apropiados y no se debe invertir muchos recursos a partir de que este sea aprobado.



**Figura 2.** Basado en prototipos

**Fuente:**(Narvaez,2019)

**RAD.** Es un modelo de desarrollo rápido de aplicaciones según Maida, E., y Pacienza, J. (2018) indica que fue desarrollado por James Martin en 1980, como sus siglas se refiere (Rapid Application Development) conocida como el diseño rápido de prototipos que admite construir sistemas de alta calidad, en corto tiempo regularmente de sesenta a noventa días y con un coste bajo de inversión. Se indica que el desarrollo rápido de aplicaciones (RAD) es uno de los métodos de desarrollo más conocidos porque domina técnicas de desarrollo ágil, los diseñadores y desarrolladores pueden utilizar el conocimiento y los

descubrimientos logrados durante el proceso para dar forma al diseño totalmente del software

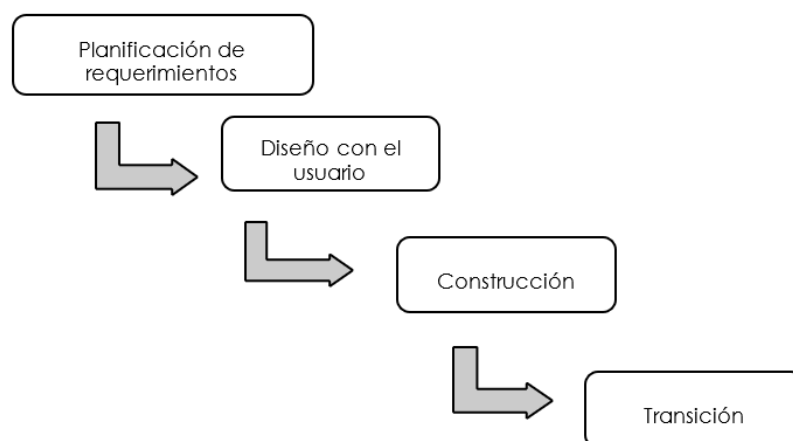
**Tabla 4** Comparativa de metodologías de desarrollo

<b>Criterio</b>	<b>RAD</b>	<b>SCRUM</b>	<b>XP</b>
Características		Se basan en un proceso incremental	Basado en flexibilidad y adaptabilidad
Revisión	Permite el Desarrollo interactivo, realización de prototipos	Necesitan de revisión diarias y se planifica el trabajo	Se debe integrar mínimo una vez al día
Objetivo	Cuando se finaliza el proceso y permite reutilizar funciones ya desarrolladas se realizan las pruebas. Aplicación en menor tiempo y bajo costo.	Indicado para proyectos complejos y obtener resultados pronto.	Trabajo en grupo
Equipos	Interacción constante con el cliente y programadores completos.	Se dividen las actividades entre los integrantes del equipo	Programación en parejas

Para los autores (Singh y Peláez, 2020), el uso de este tipo de metodologías ofrece algunas ventajas:

- Amplia la flexibilidad, de que los programadores realicen los cambios que sean precisos de manera inmediata en el desarrollo.
- Estas interacciones acortan el periodo de desarrollo al tiempo que agilizan la entrega.
- Reduce la probabilidad de cometer errores, ya que fomenta la reutilización de códigos y con ello minimiza la programación manual.
- Incrementa la proporción de complacencia de los clientes producto de su elevado nivel de colaboración y coordinación entre partes; es decir, desarrolladores-clientes usuarios.
- Limita la probabilidad de factores sorpresas producto de la temprana entre las fases del desarrollo del software.

### Fases de desarrollo rápido de aplicaciones



**Figura 3.**Fases de desarrollo rápido de aplicaciones

**Fuente:** (Gomez,2018)

- **Planificación de requerimientos.** – Hace referencia al análisis de la información alcanzada, por el método de investigación propuesto, la propietaria y el encargado de bodega del almacén de la empresa RIVATEX, realizaran un análisis de todos los procesos que se ejecutan cuando se compra o venden la mercadería y poder automatizarlo en la aplicación informática de gestión.

- **Diseño con el usuario.** –Durante esta fase se mejora el fruto del modelo de gestión con un conjunto de actividades para determinar las características y funcionalidades de la aplicación.
- **Construcción.** – Al contar con el diseño básico, gran porcentaje del proyecto está hecho: codificar cada interfaz propuesta con las necesidades de los clientes. Convierte todo lo diseñado en la fase anterior en el código fuente para el lenguaje de programación establecido.
- **Transición.** -Se realizan ensayos prácticos de la aplicación en cada interfaz, proceso que se puede repetir en reiteradas ocasiones durante esta fase con la finalidad de verificar que el producto final funciona correctamente.
- **Se ejecutan pruebas** contra la funcionalidad de cada módulo y cada botón contenido en el programa para determinar qué funcionalidad se redirige o ejecuta. Esto sucede durante el desarrollo del programa.

### 2.2.7. Sistemas Informáticos

Los sistemas informáticos son encargados de gestionar la entrada y salida de datos para su almacenamiento y procesamiento. También puede ser un grupo de controladores programables o microcontroladores utilizados para abordar uno o más desafíos específicos. Estos sistemas se componen de dos partes: el procesamiento de la información, que se conoce como denominación informática, y el segundo componente, que facilita el ingreso de datos para su procesamiento y la salida de resultados, los cuales se conocen como denominaciones periféricas. Guillem, J (2018). Para decirlo de otra manera, un sistema informático debe estar compuesto por hardware y software para ser considerado un sistema. Los periféricos, el OS del dispositivo y sus aplicaciones son ejemplos de componentes de hardware y software de un SI.

El procesamiento de los datos almacenados es la función principal de un sistema informático, son adaptables a prácticamente cualquier industria o actividad económica, y pueden emplearse casi sin restricciones por su utilidad.

Los sistemas de administración de computadoras no estaban en la lista de las principales prioridades para los propietarios de pequeñas empresas. Solo las organizaciones importantes tenían el beneficio de tener un sistema que les permitía

llevar a cabo actividades administrativas, y debido a que los costos de desarrollar e implementar sistemas eran extremadamente costosos en ese momento, un mayor acceso a nuevas herramientas tecnológicas se volvió cada vez más complicado de conseguir.

### **2.2.8. Lenguajes de programación**

Costa (2018) afirma: Podemos definir el lenguaje de programación como el modo de expresión permite dar una serie de instrucciones al ordenador en un formato lo más parecido al humano. Hay que distinguir dos partes en el lenguaje de programación, lo que se denomina sintaxis del lenguaje y la semántica. Por sintaxis entendemos el conjunto de las construcciones del lenguaje que consideramos correctas en cuanto a su forma, mientras que la semántica es ese mismo conjunto de construcciones que consideramos correctas en cuanto al significado. (p.10.)

Gortázar, F., Martínez, R., y Fresno, V. (2016) manifiestan que los lenguajes de programación es un conjunto de símbolos, reglas sintácticas y semánticas definiendo su estructura, expresiones que permiten construir aplicaciones. Cada lenguaje tiene sus enunciados verbales e instrucciones para realizar programas. Dentro de los más conocidos tenemos los siguientes: java, HTML, PHP, Microsoft Visual Studio, Java Script. El lenguaje es de amplio alcance, ha sido desarrollado y aplicado eficazmente en áreas tan diversas como microprogramación, teoría de conmutación, investigación de operaciones, recuperación de información, teoría de clasificación, estructura de compiladores, procedimientos de búsqueda y traducción de idiomas. El lenguaje permite un alto grado de formalismo útil. Se basa en gran medida en una extensión sistemática de un pequeño conjunto de operaciones básicas a vectores, matrices y árboles, y en una familia de operaciones de selección flexibles controladas por vectores lógicos. Las ilustraciones se extraen de una variedad de aplicaciones.

**JavaScript.** - Lenguaje de programación que realiza actividades complejas en el desarrollo de páginas web y presenta tres enfoques claros: Es útil, práctico y está disponible en cualquier navegador. No requiere de compilación ya que funciona del lado del cliente y tiene la ventaja de ser emigrado en cualquier página web y puede ser ejecutado en cualquier navegador sin la necesidad de instalar otro programa para ser visualizado Livias, N. (2018).

Se eligió este lenguaje de programación porque además de ejecutarse del lado del servidor también puede correr del lado del cliente por ende se puede acelerar el proceso del desarrollo del software. A continuación, nombraré algunas de sus características:

- Lenguaje del lado del cliente y del lado del servidor
- Lenguaje orientado a objetos
- De tipado débil o no tipado
- De alto nivel
- Lenguaje interpretado y no compilado
- Mas utilizado por los programadores en el mundo

Este lenguaje es multiparadigma en la actualidad es muy popular en casi todos los desarrolladores y es estandarizado por ECMAScript (siglas en inglés European Computer Manufacturers Association Script) a continuación sus ventajas más relevantes:

- Tiene una curva de aprendizaje muy corta a comparación con otros lenguajes de programación.
- Es muy rápido y se puede incrustar con un script en cualquier página independiente
- Se ejecuta en el lado del cliente por lo tanto va a reducir la carga de nuestro servidor web
- Crea interfaces dinámicas y muy amigables, se puede desarrollar menú, botones, formularios y efectos.
- Es multiplataforma se puede ejecutar en muchos sistemas operativos sin problemas y al ser un lenguaje estándar funciona en todos los navegadores.
- Del lado del servidor utiliza el motor v8 de Google
- Tiene actualizaciones frecuentemente

**Node.js.** - Es un marco del lado del servidor útil para crear aplicaciones rápidas y altamente escalables, es una plataforma que se basa en v8, el tiempo de ejecución de JavaScript que impulsa el navegador Chrome diseñado por Google. Node.js está diseñado para ser ideal para aplicaciones de E / S intensivas que

utilizan la arquitectura impulsada por eventos sin bloqueo. (Maldonado, 2018, pág. 206)

Se eligió este lenguaje de programación ya que se puede realizar funciones sincrónicas, se usa más comúnmente para procesos asincrónicos y llama a eventos con una devolución de invocación registrada para manejar el retorno de la función mientras construye una aplicación.

El próximo evento o función en su aplicación podría estar en cola para su ejecución mientras espera la devolución. El evento de retorno de llamada de la primera función es ejecutado y manejado por la invocación a la función. En Node.js, se detalla este procesamiento impulsado por eventos.

**Api Rest.** Es una interfaz de programación (API) que utiliza HTTP verbos como GET, PUT, POST, y DELETE para operar de datos. También conocidas como servicios web RESTFUL, las API RESTFUL se basan en el enfoque Representational State Transfer (REST), un estilo arquitectónico que permite a los desarrolladores manipular datos. (Ariel, 2016)

**Ionic.** – Según (Manda, 2018) Es una herramienta multiplataforma de código abierto que admite desarrollar aplicaciones híbridas que funcionan en diferentes plataformas móviles como Android, IOS y Windows sin esfuerzo y aumentando la facilidad de desarrollo. El marco iónico mejora el aspecto y el efecto de la interfaz de usuario de la aplicación híbrida con el uso de HTML5 y CSS.

Se eligió este framework porque con esta plataforma, puede crear aplicaciones móviles utilizando tecnologías web como HTML, CSS, JavaScript, Angular y Typecript. Ionic es una colección de componentes que habilitan la funcionalidad de la plataforma móvil, ofrece una variedad de comportamientos además de la función, proporciona un rendimiento excelente al tiempo que requiere cambios mínimos de DOM.

**Angular.** – ha evolucionado y se ha ido adaptando a las nuevas tecnologías con la aparición de la versión 5 en 2017, y con la versión 6 en 2018. Aun así, la mayoría de las aplicaciones web actuales utilizan la versión 2, que fue la última versión de Angular 1. La primera versión de Angular, AngularJS, se publicó en 2009 por el equipo de Google. Fue uno de los primeros frameworks MVC para JavaScript, y su popularidad ha ido en aumento con el tiempo. AngularJS tiene una larga lista de características interesantes, y conocerlas puede ser útil para comprender los



principios de Angular y la forma en que funciona el framework. (QUALITY DEVS, 2019) Según Manuel José Gonçalves (2021), Angular es una solución integral, lo que significa que no necesitas nada más para construir una aplicación. Con Angular CLI, puedes crear un proyecto, agregar nuevos comportamientos, crear una biblioteca y crear un servidor de desarrollo.

Por otro lado, permite una mayor eficiencia en el código y un mejor funcionamiento en general de la aplicación. Ya que Angular es un framework de código abierto, cualquier persona puede crear una librería de código que pueda ser utilizada por los demás, lo que facilita la creación de aplicaciones.

Características de angular

Las características clave de Angular son:

Estructura de aplicaciones MVC

- Directivas
- Enrutamiento
- Datos bidireccionales
- Modelos
- Plantillas
- Servicios
- Inyección de dependencias
- Estabilidad

**Tabla 5** Lenguaje de programación

**Cuadro comparativo de lenguajes de programación**

Lenguaje	Paradigma	Características	Ventajas	Desventajas
C#	Parte de la plataforma net y está orientado a objetos.	Sintaxis fácil de usar, multiplataforma	Tipos de datos, atributos, control de versiones	Requiere un mínimo de 4 GB para su instalación.
HTML	Se utiliza para páginas web	Es multiplataforma, puede ser creado y editado, etiquetas con un inicio y un fin.	Permite describir hipertexto, despliegue rápido, fácil de aprender y admite todos los exploradores	Estático, la interpretación puede cambiar en cada navegador.
JavaScript	Orientado a objetos	Es dinámico, simple, maneja objetos, sintaxis fácil de usar y es multiplataforma.	Lenguaje muy sencillo de usar, compatible, es muy versátil y se ejecuta en manera híbrida.	En el Front-End sus códigos son visibles, sus opciones 3D son limitadas.
Python	Orientado a objetos	Lenguaje interpretado, no compilado, es multiplataforma.	Simplificado y rápido, flexible, ordenado, limpio y es portable.	Hosting, librerías de terceros y suelen ser lentos.

PHP	Orientado a objetos	Libre y abierto,
Genera páginas web dinámicas, mucha oferta de trabajo.	entorno de desarrollo rápido y fácil de configurar.	El código no puede ser ocultado eficientemente, dejan abierta muchas brechas de seguridad.

---

**Fuente:**( Guality,2019)

### 2.2.9. Bases de datos

Actualmente se utilizan dos modelos de datos principales para representar el conocimiento y la información en los sistemas informáticos. Los modelos de bases de datos, especialmente las bases de datos relacionales han sido líderes en las últimas décadas, permitiendo que la información se almacene y consulte de manera eficiente. Por otro lado, las ontologías han aparecido como una alternativa a las bases de datos en aplicaciones que requieren un significado más 'enriquecido'. Sin embargo, existe controversia sobre la mejor técnica de modelado de información, ya que ambos modelos presentan características similares. (Carmen, 2018)

Las bases de datos son el producto de la necesidad humana de almacenar información, es decir, para guardar la información para poder acceder a ella más tarde. En este sentido, la llegada de la informática ha proporcionado un elemento digital indispensable para almacenar grandes cantidades de datos en un espacio físico limitado, gracias a su capacidad.

**Oracle.** Es una herramienta cliente/servidor para la gestión de base de datos, usada principalmente por las grandes empresas a nivel mundial que administran y controlan una gran cantidad de datos con el propósito de reducir costos y los riesgos asociados a la pérdida de información Alvarado, J., Naranjo, J., y Chaves, P. (2019).

Las plataformas compatibles por este sistema gestor son Windows, Linux, Mac Os X, BSD y Unix. Para su utilización se debe instalar la herramienta servidor y cargar la base de datos con Oracle Designer, Developer PL/SQL. En la cual Designer es una herramienta que conecta a la base de datos, Developer permite crear formularios compilarlos y ejecutarlos mientras que PL/SQL gestiona a la base de datos.

Es una solución de administración de bases de datos cliente servidor que es utilizada principalmente en empresas en todo el mundo para administrar y controlar cantidades masivas de datos con el fin de disminuir los costos y riesgos asociados con la pérdida de datos se encuentran entre los métodos compatibles con este sistema de gestión. Para utilizarlo, deberá instalar la herramienta del servidor y cargar la base de datos con Oracle.

**Microsoft SQL.** - Es un sistema que sirve para la gestión de base de datos (RDBMS), ha sido creado por Microsoft basado en el modelo relacional siendo su principal lenguaje de consulta el Transact-SQL permitiendo desarrollar, implementar y administrar soluciones tanto locales como en la nube. Sus principales características son:

- Administración de otros servidores de datos.
- Soporte de procedimientos almacenados.
- Uso de comandos DDL Y DML.
- Soporte de transacciones.

La base de datos se ejecuta en modo servidor, solo se puede instalar en un servidor Windows, proporciona procesamiento de datos, estructura y servicios de acceso, y se puede usar desde aplicaciones o herramientas de cliente como SQL Server Management Studio o SQLCmd, o herramientas de terceros. Puede usar conectores como ODBC, ADO.NET u OLE-DB para ejecutar comandos TSQL en SQL Server.

**Bases de datos NoSQL.** -Las bases de datos no relacionales se las utiliza para creación de redes sociales, sitios de comercio electrónico que almacenan y procesan grandes volúmenes de datos, debido a que tiene un alto nivel de concurrencia y esta optimizado para almacenar cantidades masivas de datos. Las bases de datos NoSQL son dinámicas y permiten insertar, editar o eliminar datos a menudo, se utilizan en tiendas donde sus productos se van actualizando por algo más novedoso o si algún producto se acaba se lo reemplaza por otro (Pérez, 2020).

De acuerdo a lo antes mencionado una base de datos no relacional tiene una concurrencia mayor a las bases de datos relacionales, nos permite agregar, editar y eliminar productos en tiempo real con manejo de grandes volúmenes de información, es importante porque la empresa cuenta con varios productos que se quiere dar a conocer a los clientes con sus respectivos detalles para que realicen sus pedidos, además estas bases de datos tiene una escalabilidad horizontal que permite en crecimiento de datos extenso y técnicamente no tiene límites, se podrían agregar cuantos servidores sean necesarios.

**Mongodb.** -. Es una base de datos orientada a documentos. Esto quiere decir que, en lugar de guardar los datos en registros, guarda los datos en colecciones. Estos son almacenados en BSON, que es una representación binaria de JSON. Es fácil de

aprender y usar y proporciona a los desarrolladores todas las funcionalidades que necesitan para satisfacer los requisitos más complejos a cualquier escala. Se proporcionan drivers para más de diez lenguajes de programación. (Hurtado, 2019)

Se eligió este motor de base de datos ya que es muy útil en entornos que requieren escalabilidad. Con las opciones de copiar y dividir, que son fáciles de configurar, podemos tener un sistema de capas horizontales sin demasiados problemas.

**Tabla 6** comparativa de bases de datos relacionales y no relacionales

<b>Características</b>	<b>Bases de datos relacionales</b>	<b>Bases de datos no relacionales</b>
Rendimiento	Bajo	Alto
Disponibilidad	Bueno	Bueno
Escalabilidad	Alto (más costoso)	Alto
Almacenamiento de datos	Bueno para BBDD de mediano a gran tamaño	Optimizado para cantidades masivas de datos
Concurrencia de lectura y escritura	Rápido	Lento
Flexibilidad	Bueno	Bajo

### **III. METODOLOGÍA**

#### **3.1. ENFOQUE METODOLÓGICO**

##### **3.1.1. Enfoque**

El presente proyecto utilizo un enfoque mixto debido a que abarca al enfoque cualitativo y cuantitativo mismo que incluye las características de cada uno de ellos, es un proceso que recolecta, vincula y analiza datos cualitativos y cuantitativos en una investigación, utiliza evidencias de datos numéricos, textuales, verbales, simbólicos, visuales entre otras que representa un conjunto de métodos sistemáticos, empíricos y críticos de la investigación.

Se utilizo el enfoque cualitativo porque en base a este se describirá las cualidades y características del sistema, permitirá la obtención de información y levantamiento de requerimientos para el desarrollo de la solución tecnológica.

Se utilizo el enfoque cuantitativo debido a que se realizará un análisis estadístico utilizando a la encuesta como técnica de recolección de información, además permitirá cuantificar los indicadores de la gestión de ventas e inventario.

##### **3.1.2. Tipo de Investigación**

###### **3.1.2.1 Investigación de Campo**

Se utilizó esta investigación debido a que la correlación entre los investigadores y el fenómeno de estudio es directa, además su aplicación sirvió para obtener información relevante de los objetos de estudio a través de encuestas a los trabajadores del departamento de administración de la empresa REVATEX en la ciudad de Quito.

###### **3.1.2.2 Investigación Descriptiva**

Este tipo de investigación ayuda a determinar las características del trabajo, así podemos identificar los aspectos más importantes acerca del uso de las herramientas tecnológicas en las empresas. En la empresa RIVATEX se pudo identificar que el mayor uso de las herramientas es Excel, Word es decir que si tienen conocimiento en manejar las herramientas tecnológicas.

### **3.1.2.3 Investigación Documental**

Se ha tomado este tipo de investigación ayudándonos con fuentes secundarias como textos, libros, artículos científicos y en su gran parte todos los recursos alojados en la web han facilitado la percepción de los fenómenos de estudio y han amplificado la visión sobre el desarrollo de sistemas.

### **3.1.2.4 Investigación Exploratoria.**

La investigación exploratoria se empleó porque era necesario profundizar en el tema de estudio que ha sido poco explorado por otros investigadores. Además, ha generado nuevos datos que pueden ser utilizados en futuros trabajos investigativos.

## **3.2. IDEA A DEFENDER**

El desarrollo de un sistema informático permite automatizar los procesos de venta e inventario en la empresa RIVATEX en la ciudad de Quito.



### 3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

**Tabla 7** Variable independiente

Variable	Definición	Dimensión	Indicador	Técnica	Instrumento
<b>Aplicación informática</b>	"Estos son programas que permiten la interacción entre el usuario y la computadora, permitiéndoles realizar varios tipos de trabajo" (Chirimelli, 2017)	Integración y control	Cantidad de usuarios	Encuesta	Cuestionario
		Seguridad	Nivel de seguridad	Diseño de interfaces	
		Registros	Forma de proteger los datos	Requerimientos	
		Información	Respaldo de información		
			Registro de ventas		
			Cantidad de ventas		

**Tabla 8** Variable dependiente

Variable	Definición	Dimensión	Indicador	Técnica	Instrumento
<b>procesos de venta e inventario</b>	Moreno, (2017) afirma que: "La gestión de inventario se define como conjunto de existencias que posee una organización para tener planificado, organizado el inventario que posee".	Control de Inventarios	Ingreso de artículos	Encuesta	cuestionario
		Control de ventas	Reportes	Diseño de	
		Manejo de ventas	Facturación Módulo	interfaces	

### **3.4. MÉTODOS UTILIZADOS**

#### **3.4.2 Método Inductivo**

El método inductivo se caracteriza porque permite analizar las variables de estudio partiendo del análisis de datos obtenidos, los autores Hernández, Zapata y Mendoza, (2018) aseguran que: este método aborda un área más específica hacia una conclusión macro de la investigación, se lo maneja en estudios tácticos apoyándose en la experiencia del área estudiada, de esta manera se puede analizar similitudes de tema a tratar.

##### **3.4.2.3 Método de Investigación Acción.**

Vargas (2018) menciona que: la investigación-acción tiene como objetivo cambiar una situación actual, no se trata de estudiar una realidad, sino de cambiarla, de modo que se debe intervenir en la realidad en la que se encuentra el investigador. La investigación-acción tiene como objetivo cambiar una situación actual, no se trata de estudiar una realidad, sino de cambiarla, de modo que se debe intervenir en la realidad en la que se encuentra el investigador. Por otro lado, la investigación-acción establece una retroalimentación constante de la forma de intervenir de la realidad a estudiar, es decir, la realidad es el espejo de la investigación, de forma que la investigación-acción no se puede concebir sin la intervención del investigador, ya que éste es quien lleva a cabo.

Este método permitió el desarrollo de una propuesta de inventarios y ventas que genere una respuesta al problema analizado y de esta manera con estudios posteriores de implementación se podrá contribuir a la transformación de la realidad actual en la empresa RIVATEX.

### **3.5. ANÁLISIS ESTADÍSTICO**

#### **3.4.1 Método deductivo**

El método deductivo se caracteriza porque este método permite empezar a establecer una teoría, a partir de un análisis de la información recopilada. De acuerdo con Hernández et al. (2018) se caracteriza porque se parte de lo particular a lo general.

Este método es utilizado en la etapa de análisis de la información, aquí se analiza la información recopilada en la etapa de recolección de datos y se establecen las posibles hipótesis que pueden dar como respuesta al problema del estudio.

### **3.5.1. Población**

La población está comprendida por 10 personas, donde se aplica la encuesta a la propietaria y los empleados, fuentes principales para la recolección de información teniendo en cuenta que al ser una población pequeña.

## **3.5 TÉCNICAS E INSTRUMENTOS**

### **3.5.2 Encuesta**

Se define como una técnica cuantitativa que se realiza sobre una muestra, utiliza métodos estandarizados con el objetivo de conseguir resultados cuantitativos, además se ejecuta mediante el estudio de características objetivas y subjetivas, así mismo el recabar información de cualquier tipo de población, este estándar permite que sea menos complejo el tratamiento de los datos en el análisis estadístico (Gil, 2016).

Se realizará con el fin de obtener la información concisa sobre la situación actual de la empresa para dar soluciones prioritarias de la propuesta.

A través de esta técnica de recolección información realizaremos preguntas abiertas a la propietaria y los empleados de la empresa con el objetivo de recopilar datos de manera puntual y organizada para la identificación de todos los procesos que se llevan a cabo actualmente en la gestión de ventas e inventarios específicos a tratarse en esta investigación.

En nuestra investigación, utilizamos fichas de caracterización basadas en la NORMA ISO 9001:2015 para sistemas de gestión de calidad con el objetivo de conocer el estado actual de la empresa como se encuentra realizando los procesos de ventas e inventarios, nos permitirá analizar, diseñar y operacionalizar los procesos administrativos que se encuentran en la empresa.

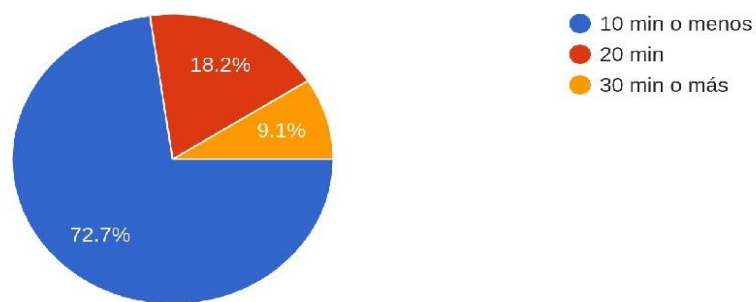
## IV. RESULTADOS Y DISCUSIÓN

### 4.1. RESULTADOS

#### 4.1.1 Análisis e Interpretación de resultados

Para realizar el análisis se obtuvieron los datos de una encuesta aplicada a los trabajadores de la empresa "RIVATEX" ubicada en la ciudad de Quito, las preguntas fueron planteadas a través de la herramienta de formularios de Google para facilitar la tabulación y obtención de gráficos estadísticos.

#### 1. ¿Aproximadamente cuánto tiempo se demora en efectuar un registro de venta?



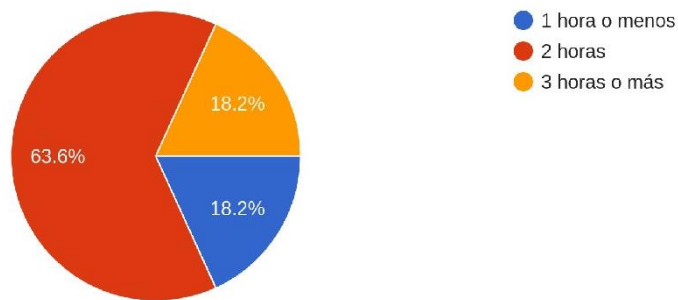
**Figura 4.**Representación porcentual pregunta 1

#### **Análisis:**

Los resultados logrados muestran que la mayoría de los empleados al realizar un registro de ventas se tarda 10 min o menos, mientras que el resto indica que se tarda 20 min para la facturación, con los datos observados se puede manifestar que los

empleados ejecutan un registro de ventas tradicional en un tiempo aceptable, cumpliendo los objetivos de la empresa y la expectativa del cliente.

**2. ¿Aproximadamente cuánto tiempo se tarda en realizar un registro de inventario?**

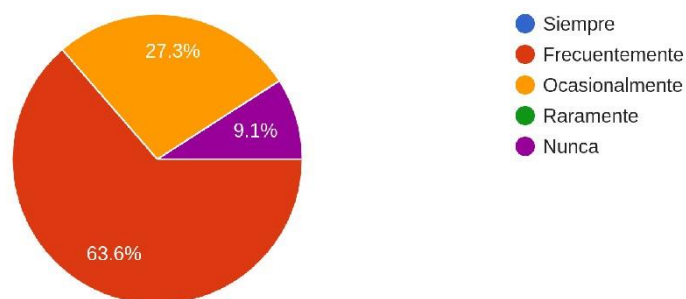


**Figura 5.**Representación porcentual pregunta 2

**Análisis:**

Los resultados obtenidos muestran que la mayoría de los empleados al realizar el registro de inventario se tarda 2 horas, mientras el restante afirma que se puede tardar entre 1 hora o menos y de 3 horas o más, con los datos observados se puede llegar a la conclusión que los empleados requieren más tiempo para realizar el inventario causando una mayor inversión en esta área.

**3. ¿Al momento de realizar el registro de inventarios existe duplicidad en los valores ingresados?**

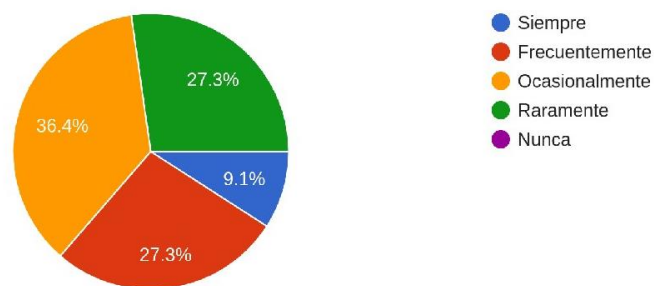


**Figura 6.**Representación porcentual pregunta 3

### **Análisis:**

En los resultados obtenidos la mayoría de los empleados indican que al realizar el registro de inventario existe duplicidad frecuentemente, mientras que el restante afirma que no existe duplicidad o duplicidad ocasionalmente, llegando a la conclusión que hay un gran porcentaje de irregularidades al momento de realizar el inventario, afectando el grado de cumplimiento laboral.

#### **4. ¿El control de inventarios que aplica la empresa le ha causado faltante y sobrante del producto para la venta?**

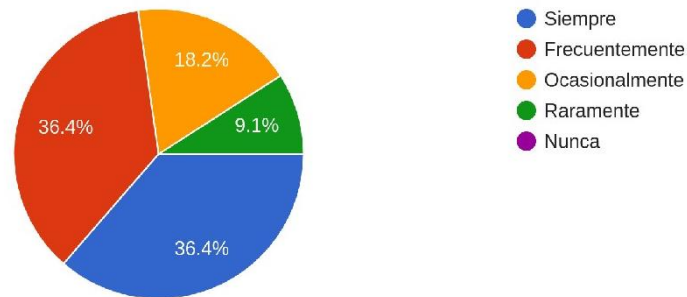


**Figura 7.**Representación porcentual pregunta 4

### **Análisis:**

Con los resultados obtenidos muestra que la mayoría de los empleados afirma que ocasionalmente existen faltantes o sobrantes de productos para la venta, por otro lado, el restante indica que es frecuente y raramente, llegando a la conclusión que los procesos actuales que tiene la empresa no son eficientes por ende en ocasiones tienen que repetir el proceso más de una vez causando inconvenientes al momento de registrar los inventarios.

**5. ¿Cree Ud. que el actual manejo de inventarios que tiene la empresa le permite hacer frente a la oferta y demanda?**

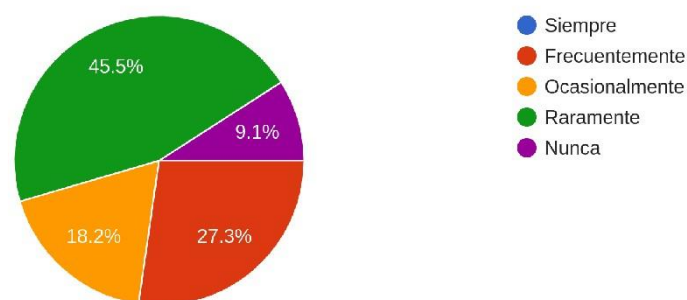


**Figura 8.**Representación porcentual pregunta 5

**Análisis:**

El mayor porcentaje de los encuestados indican que siempre y frecuentemente la empresa enfrenta a la oferta y demanda, mientras que el menor porcentaje afirma que la empresa no logra enfrentar los inconvenientes de la oferta y demanda lo que quiere decir que la empresa debe analizar a fondo estas variaciones para enfrentar la oferta y demanda.

**6. ¿Los clientes actuales han tenido reclamos por equivocaciones en las facturas emitidas?**



**Figura 9.**Representación porcentual pregunta 6

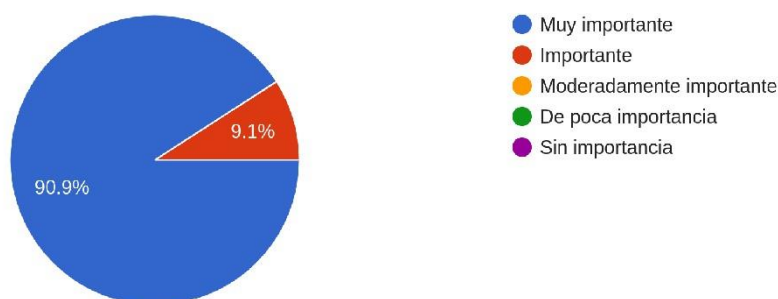
**Análisis:**

La mayoría de los encuestados respondieron que raramente hay equivocaciones de facturas emitidas, el restante afirma que es frecuente y ocasionalmente, llegando a la conclusión que no existe dificultad con los pedidos lo que significa que la empresa no tiene mayores inconvenientes al momento de realizar las



facturas, teniendo una buena comunicación con los clientes y cumpliendo con los objetivos de la empresa.

**7. ¿Considera Ud. Necesario o conveniente que la empresa implemente un sistema informático?**

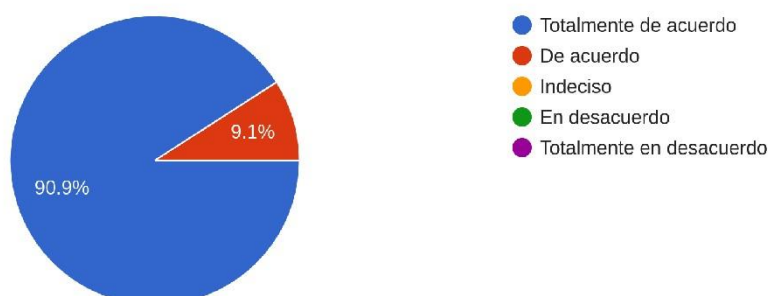


**Figura 10.**Representación porcentual pregunta 7

**Análisis:**

La mayoría de los encuestados indican que es muy importante la implementación de un sistema informático, llegando a la conclusión que si es necesario la implementación de este sistema de esa manera se lograra mejorar las condiciones y las actividades que realiza de la empresa.

**8. ¿Piensa Ud. que la implementación de este sistema informático ayudara a resolver los inconvenientes que tiene la empresa como: atención, ventas e inventario?**



**Figura 11.**Representación porcentual pregunta 8

### Análisis:

El nivel de aceptación del personal de la empresa para la implementación de este sistema informático fue aceptable demostrando que están totalmente de acuerdo con la implementación del sistema informático, llegando a la conclusión que es necesario la implementación de este sistema informático para poder brindar un mejor servicio y lograra mejorar el nivel de satisfacción del cliente.

### 9. ¿Considera que las facturas serian emitidas de mejor manera y con mejor presentación mediante la entrega de una factura generada por el sistema en comparación con el actual?

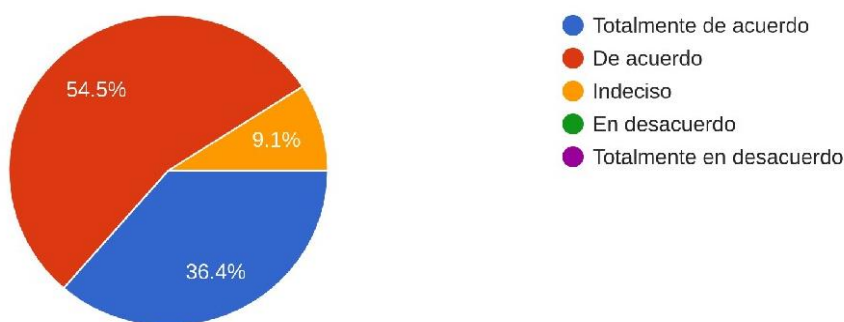


Figura 12.Representación porcentual pregunta 9

### Análisis:

Según la encuesta realizada al personal de la empresa RIVATEX, se puede observar que la mayoría están de acuerdo con las facturas que serían emitidas por el sistema, llegando a la conclusión que, si se considera que al entregar las nuevas facturas tendrán mejor presentación, una manera más eficiente y ordenada al momento de brindar el servicio que se ofrece.

### 10. ¿Cree Ud. ¿Que el sistema que se va a implementar pueda desempeñar las funciones de venta e inventario requerido por la empresa?

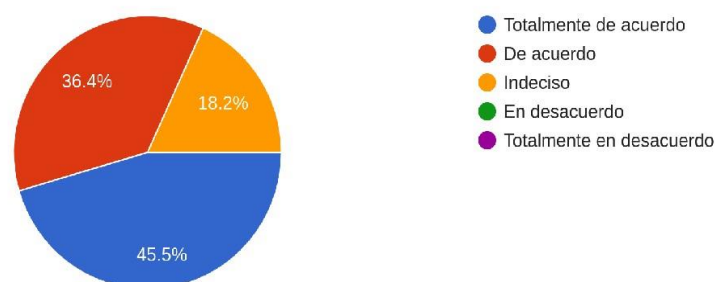
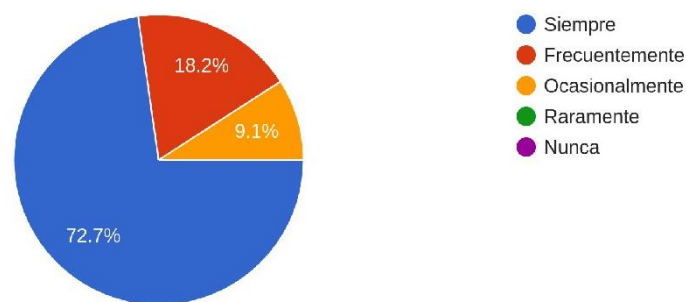


Figura 13.Representación porcentual pregunta 10

### **Análisis:**

La mayoría de los encuestados indican que están totalmente de acuerdo con la implementación del sistema para desempeñar las funciones de venta e inventario requerido por la empresa, y la minoría de los encuestados se encuentran indecisos, llegando a la conclusión que es importante el desarrollo del sistema para brindar una mejor atención al cliente y obtener el cumplimiento laboral deseado.

### **11. ¿Cree usted que con el desarrollo de un sistema se tendrá un mejor control y seguimiento de los clientes?**



**Figura 14.**Representación porcentual pregunta 11

### **Análisis:**

El nivel de aceptación de la mayoría del personal de la empresa para desarrollo de un sistema para mejorar el control y seguimiento de los clientes considerando que es importante, tomando todo esto como referencia se puede deducir que con la ayuda del sistema es posible mejorar el control y seguimiento de los clientes con datos accesibles y actualizados en todo momento.

## **4.2. PROPUESTA**

### **4.2.1 Introducción**

En primer lugar se realizó un estudio de factibilidad para determinar la viabilidad del desarrollo de la aplicación obteniendo un resultado positivo ya que la empresa cuenta con los recursos necesarios para la realización esta actividad, en segundo lugar para el desarrollo del proyecto se tomó a la metodología de desarrollo RAD debido a que con esta metodología se puede realizar una planificación ordenada, a partir de la fase de planificación se llevó a cabo todas las fases correspondientes a esta metodología.

Determinamos que la mejor solución es el desarrollo de una aplicación web que integre módulo de ventas e inventarios la misma que fue desarrollada utilizando las herramientas tecnológicas necesarias.

#### **4.2.2 Estudio de Factibilidad.**

##### **4.2.2.1. Factibilidad Organizacional.**

###### **Aspectos generales de la organización.**

Institución: PRIVATEX

**Ubicación geográfica:** Quito - Duchicela N4-14 y Geovanny Calles, Quito, Ecuador

**Área:** Departamento de administración y ventas

**Sistema:** Sistema informático para los procesos de venta e inventario de la empresa RIVATEX

**Objeto social:** Venta de productos

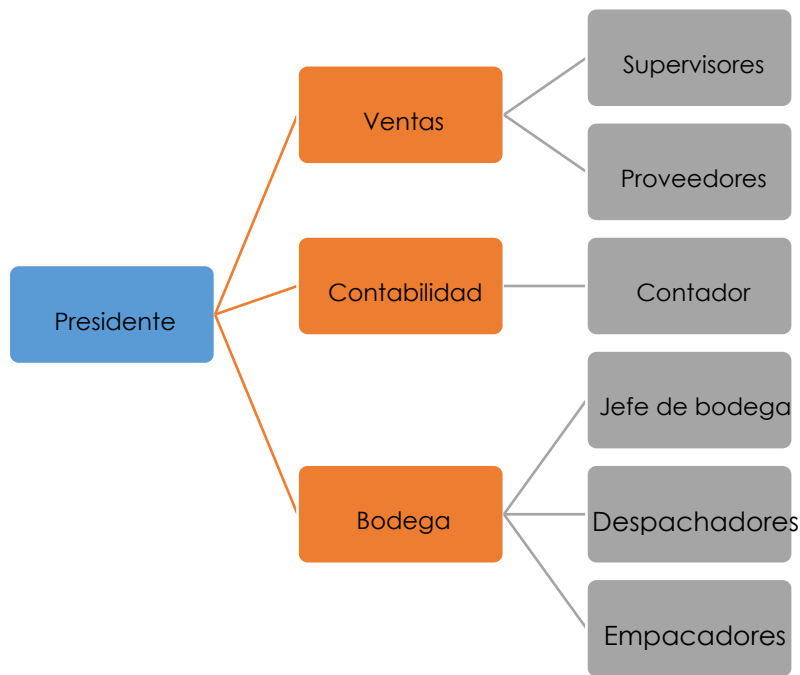
###### **Misión**

Garantizar de servicios de calidad de sus servicios que brinda, mediante la aplicación de procesos técnicos y eficientes, que promuevan y preserve la seguridad y confianza de los ciudadanos.

###### **Visión**

La Empresa RIVATEX será un referente a nivel nacional, a través de la innovación permanente y la mejora continua de sus procesos con enfoque en la rentabilidad social, la calidad de los servicios y la modernización.

###### **Organigrama**



**Figura 15.**Organigrama

#### **4.2.1.2 Factibilidad Técnica.**

Para el desarrollo de este proyecto se elaboró una lista de los recursos que serán utilizados, tales como el hardware y software.

El sistema informático para los procesos de venta e inventario se va a desarrollar con la tecnología de Node.js, Express.js, Angular, JavaScript, MongoDB estos recursos fueron elegidos por ser Open Source y se integran fácilmente al momento de programar, esto es un beneficio al no generar ningún costo en la investigación, además son herramientas que cuentan con un gran apoyo por parte de la comunidad.

#### **Recursos software**

**Tabla 9.**Recursos de Software

Tipo	Nombre	Descripción	Cantidad recurso	Recurso
Software	JavaScript	Lenguaje de Programación		1
	Express.js	Framework Back-end		1
	Node.js	Entorno de desarrollo integrado		1
	Angular	Framework Front-end		1
	Mongodb	Sistema gestor de base de datos		1
	Git y GitHub	Sistema de control de versiones		1
	Microsoft Office	Herramienta ofimática		2

**Fuente:** (Hernandez,2019)

El investigador cuenta con los equipos necesarios para el desarrollo del proyecto, además de una conexión a internet lo que facilita la comunicación de trabajo, la organización dispone de recursos económicos para contratar un servidor virtual (VPS), se concluye que los recursos son aptos y existe una factibilidad técnica.

**Tabla 10.** Tipo de recursos.

Tipo de recurso	Nombre del recurso	Descripción	Cantidad
Hardware	Equipo computación	de Laptop hp da1500la	1
	Servidor	Disco duro-sata 1TB HDD	1
		Memoria RAM Kingston 16 GB	
		N/A	
	Impresora	Epson L365	1

### 4.2.1.3 Factibilidad Económica.

En el presupuesto del proyecto se tomó en cuenta recursos de hardware, software, talento humano y materiales de oficina.

**Tabla 11.** Factibilidad Económica

Descripción	Cantidad	Costo Real	Costos Costo referencial
		de Hardware	
Equipos de computación	de 1	00,00	1700,00
Impresora	1	00,00	250,00
Servidor aplicaciones (host)	de 1	500,00 c/m	500,00 c/m
<b>Total, de hardware</b>		<b>500,00c/m</b>	<b>2100,00</b>
<b>Costos de software</b>			
JavaScript		00,00	00,00
Node.js		00,00	00,00
Express.js		00,00	00,00
Angular		00,00	00,00
Mongodb		00,00	00,00
<b>Total, de Software</b>		<b>\$ 00,00</b>	<b>\$ 00,00</b>
<b>Talento humano</b>			
Programadores	1	00,00	5000,00
<b>Total, de talento humano</b>		<b>\$ 00,00</b>	<b>\$5000,00</b>
<b>Materiales de oficina</b>			
Internet		360	360
Útiles de oficina		100	100
Impresora	1	00,00	250
Varios		150	150

<b>Total, de materiales de oficina</b>	<b>\$ 610</b>	<b>\$ 860</b>
<b>Subtotal</b>	<b>\$1.110,00</b>	<b>\$7.960,00</b>
10% de imprevistos	\$ 111,00	\$ 796,00
<b>Total</b>	<b>\$1221,00</b>	<b>\$ 8756,00</b>

#### **4.2.1.4 Factibilidad Operativa.**

##### **Situación actual**

En la empresa RIVATEX almacena los registros de los clientes en hojas de cálculo, lo que dificulta su acceso y búsqueda. Lo que genera retraso en la atención al cliente y en caso de algún fallo en el equipo, la información se puede perder porque no se cuenta con métodos de almacenamiento adecuados.

##### **Situación ideal**

El sistema informático para los procesos de venta e inventario de la empresa RIVATEX para la gestión de los procesos administrativo. Lo que permite centralizar la información de los contribuyentes y facilita la búsqueda de datos e historiales de pago.

Por parte de la administración se cuenta con el apoyo necesario y se va a trabajar de forma coordinada con la propietaria para obtener los requerimientos necesarios que cumplan con las funcionalidades específicas del sistema.

#### **4.2.3. Metodología RAD**

##### **4.2.3.1 Fase de planificación de requerimientos**

Una vez realizada la reunión con la propietaria, empleados de la empresa y el autor del proyecto de investigación Martínez Martínez Esteban Wladimir, se llegó a un acuerdo de el desarrollo de un sistema informático de los procesos de registro de ventas e inventarios, el cual posee como objetivo sistematizar los procesos indicados, se toma en cuenta los siguientes requerimientos.

En el área de inventarios de la empresa se registra y guardan la información en carpetas, clasificadas en: registro de entrada materia prima y el stock de producto terminado. En el área de ventas de la empresa se registra e imprime la factura y se



archiva la copia en carpetas. Se determinada los requerimientos funcionales y la prioridad de estos.

Internamente de estos procesos se identificó el siguiente problema: Los procesos que realiza en la empresa se realizan de forma manual, en la cual el proceso de ventas genera molestia debido a las largas aglomeraciones, en el proceso de registro de inventarios tiene problema al momento de realizarlos y la entrega de estos. Para ello es necesario automatizar los procesos de ventas e inventarios con el fin de tener la información de forma segura, disponible y accesible.

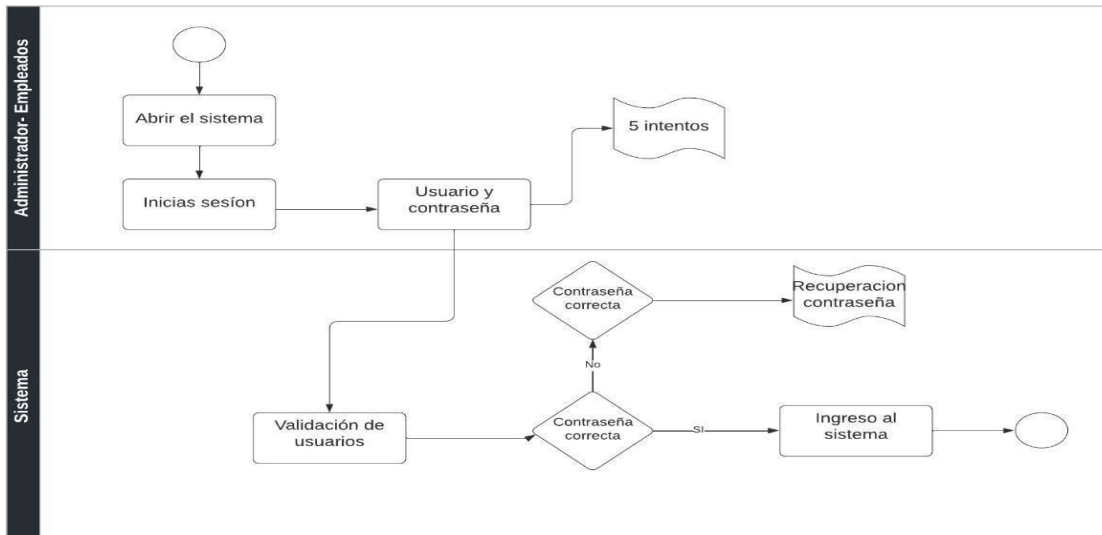
#### **4.2.3.2 Identificación de procesos**

**Tabla 12.**Identificación de Procesos

<b>Procesos</b>	<b>Descripción</b>
<b>Ventas</b>	El cliente se acerca a la empresa y adquiere un producto, los empleados le entregan el mismo y registra la factura, llenando la descripción del producto, cantidad, precio unitario y el costo del IVA, se le entrega al cliente el documento original. Una vez terminado la jornada de labores archivan los documentos en carpetas.
<b>Inventario</b>	Los empleados registran toda la metería prima que ingresa tales como botones, cierres, bordados e hilos. Una vez ingresados se procede a procesar la materia prima a un producto terminado, cuando el de producto esté terminado se lo registra para que esté listo para la venta. Se realiza registro diario de cuantos se vendió que producto más se vendió y se muestra el stock

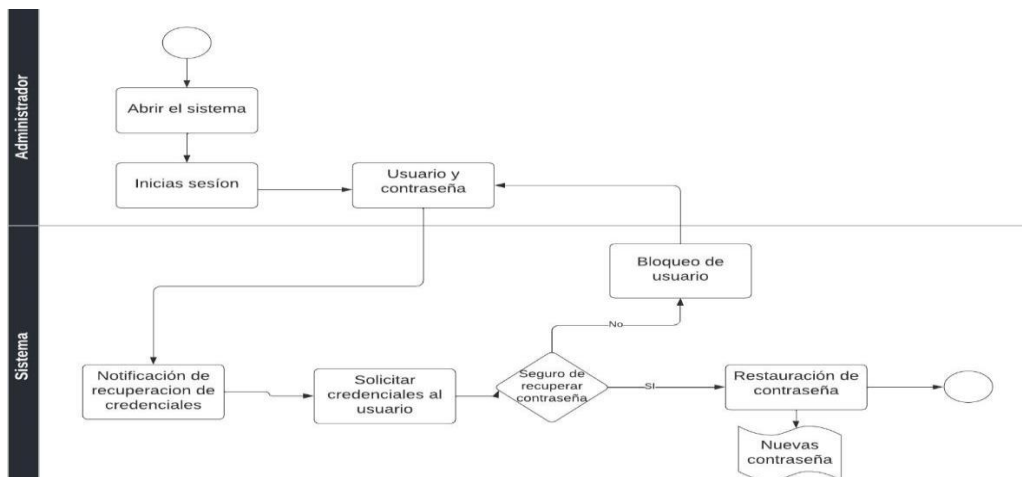
### 4.2.3.3 Fase de diseño de usuario

**Diagrama de flujo ingreso al sistema.** - Para entrar al sistema la propietaria y los empleados deben iniciar sesión con sus respectivas credenciales (usuario y contraseña) caso contrario que ingrese mal sus credenciales habrá una alerta de mensaje de error y no logrará ingresar al sistema.



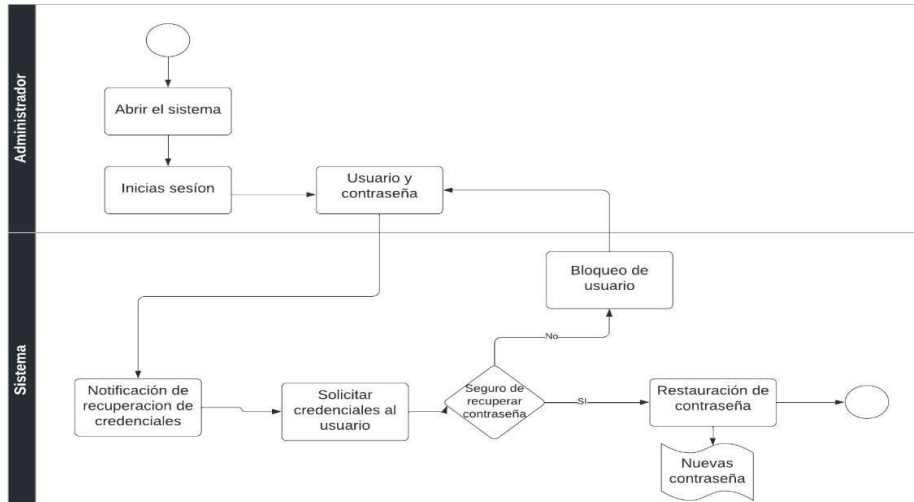
**Figura 16.**Diagrama de flujo ingreso al sistema

**Diagrama de flujo reparación de contraseña.** - Para recuperar la contraseña el administrador ingresará al sistema para rápidamente ver las credenciales del usuario que se olvidó su clave personal. El administrador pedirá la información necesaria para poder crear una nueva contraseña en el sistema.



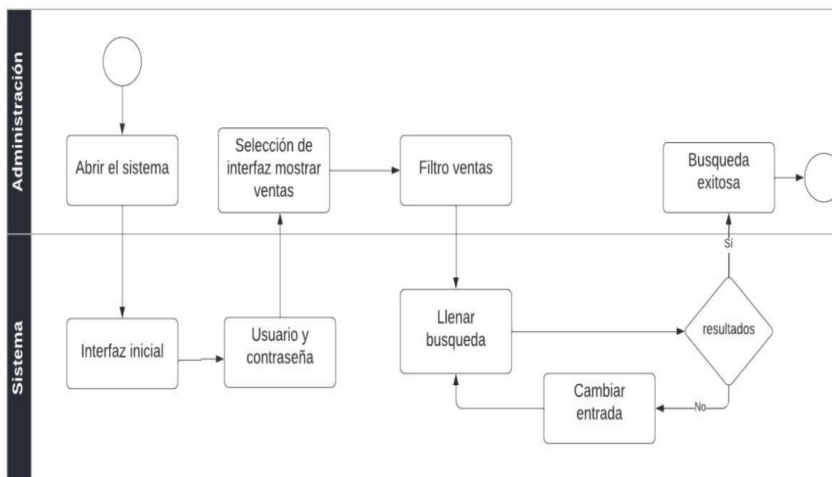
**Figura 17.**Diagrama de flujo reparación de contraseña

**Diagrama de flujo registro de ventas.** - Para el registro de ventas los empleados encargados ingresarán al sistema con sus respectivas credenciales, posterior a eso podrá acceder al módulo de ventas para hacer movimientos los mismos que serán almacenados en una base de datos.



**Figura 18.**Diagrama de flujo registro de ventas

**Diagrama de flujo registro de ventas.** - Para el registro de ventas los empleados encargados ingresarán al sistema con sus respectivas credenciales, posterior a eso podrá acceder al módulo de ventas para hacer movimientos los mismos que serán almacenados en una base de datos.



**Figura 19.**Diagrama de flujo registro de ventas

**Diagrama de flujo registro de clientes.** - El administrador o los encargados del módulo de ventas ingresará al sistema con sus respectivas credenciales, seleccionará la interfaz de clientes para poder ingresar a los mismos y poder hacer un seguimiento de estos y poder efectuar los registros de ventas.

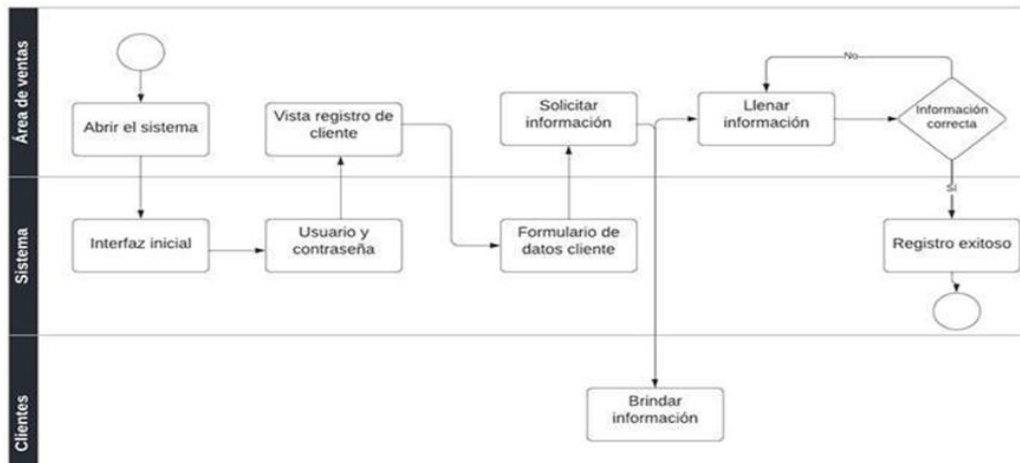


Figura 20. Diagrama del flujo registro de cliente

**Diagrama de flujo registro de categorías.** – los empleados encargados del módulo almacén ingresará al sistema con sus respectivas credenciales, seleccionará la interfaz de categorías para poderlas ingresar o actualizar las diferentes categorías que ofrece la empresa.

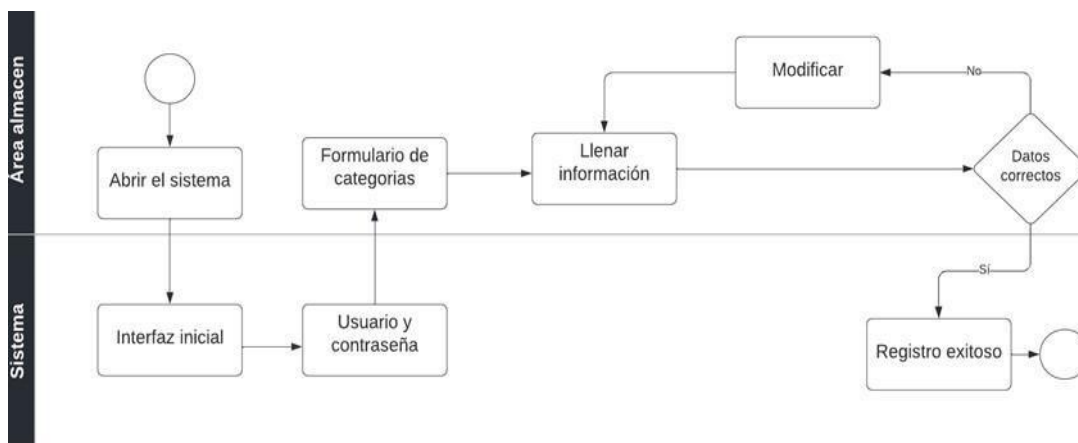
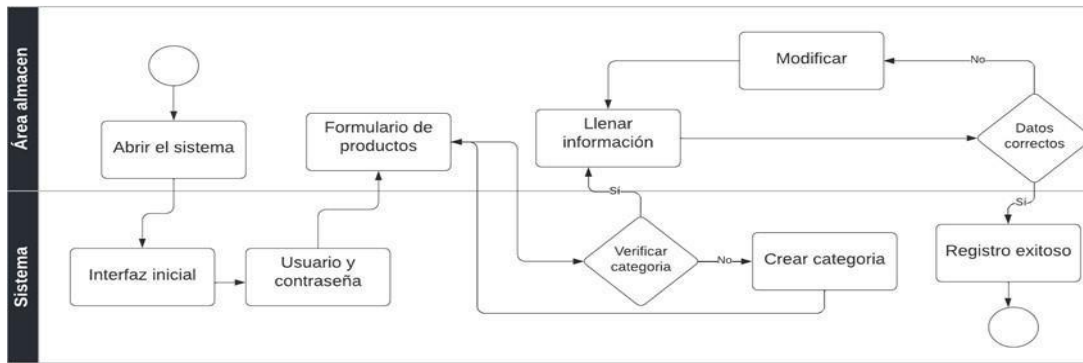


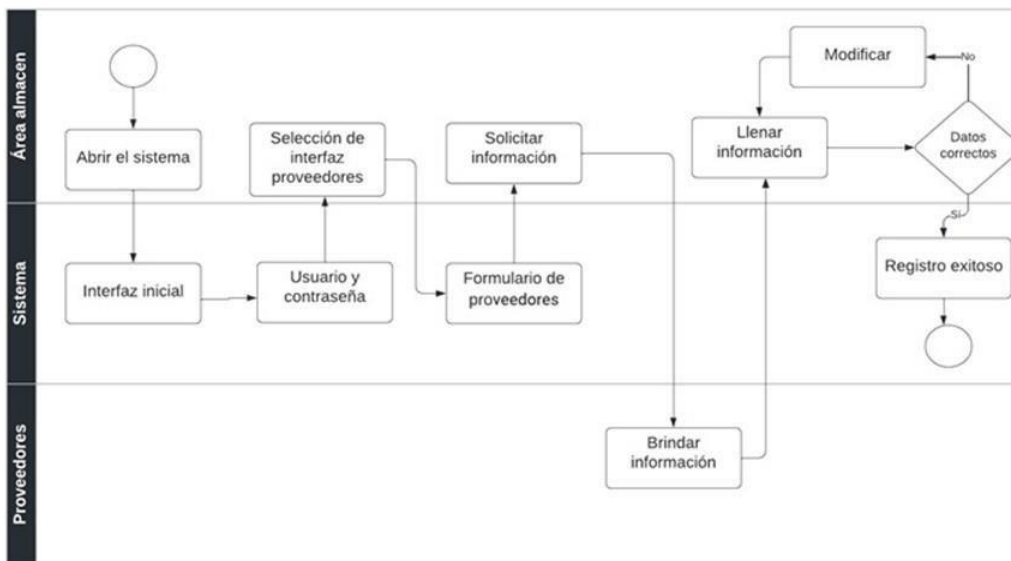
Figura 21. Diagrama de flujo registro de categorías

**Diagrama de flujo registro de artículos.** – los empleados encargados del módulo almacén ingresará al sistema con sus respectivas credenciales, seleccionará la interfaz de artículos para poderlas ingresar o actualizar los diferentes artículos que tiene la empresa.



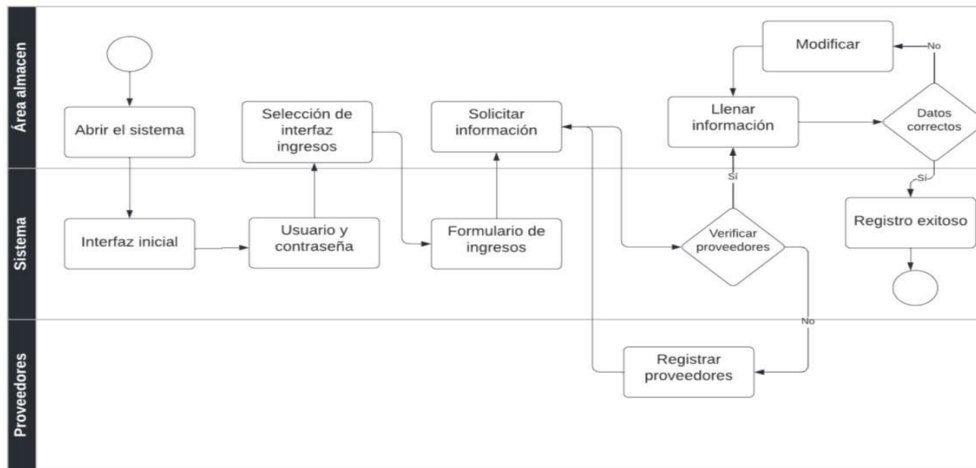
**Figura 22.**Diagrama de flujo de artículos

**Diagrama de flujo registro de proveedores.** – los empleados encargados del módulo proveedores ingresará al sistema con sus respectivas credenciales, seleccionará la interfaz de proveedores para poderlas ingresar o actualizar los diferentes proveedores que tiene la empresa.



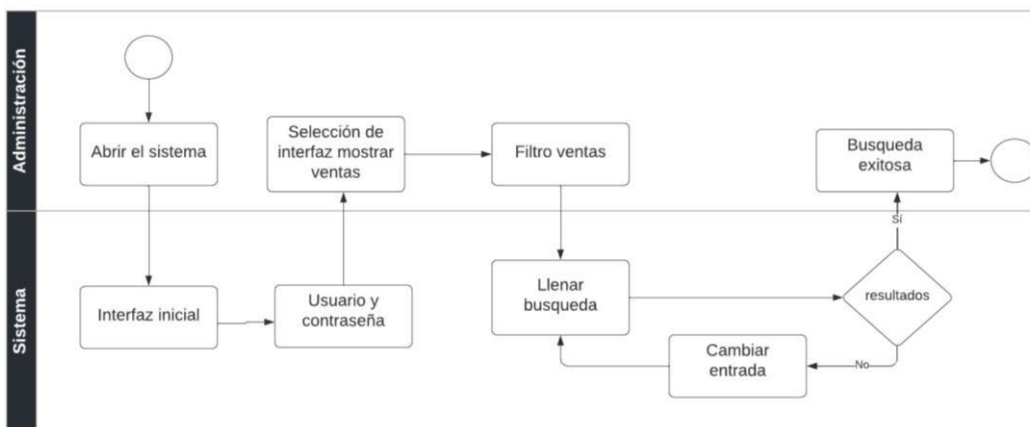
**Figura 23.**Diagrama de flujo registro de proveedores

**Diagrama de flujo registro de ingresos.** – los empleados encargados del módulo de ingreso ingresarán al sistema con sus respectivas credenciales, seleccionará la interfaz de ingreso para poderlas ingresar o actualizar los diferentes ingresos que tiene la empresa.



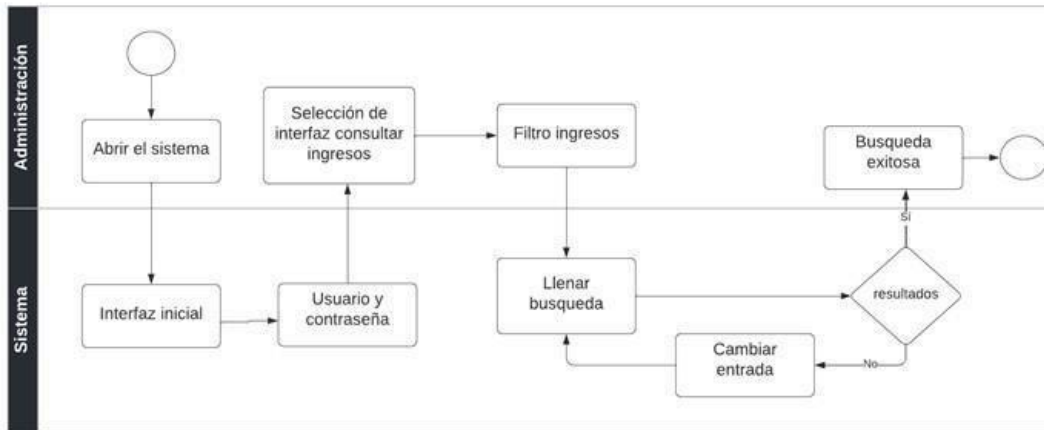
**Figura 24.**Diagrama de flujos registro de ingreso

**Diagrama de flujo consulta de ventas.** – los empleados encargados del módulo de consulta de venta ingresarán al sistema con sus respectivas credenciales, seleccionará la interfaz de este, para poderlas ingresar o actualizar la consulta de ventas de la empresa.



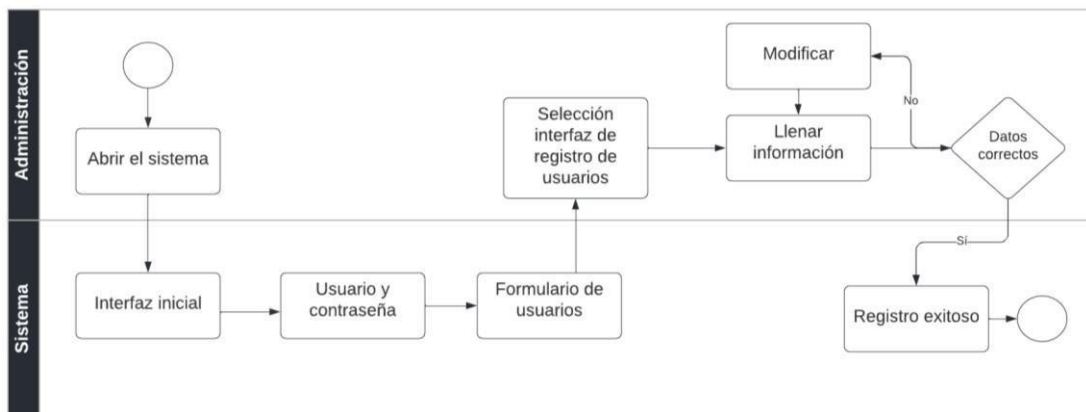
**Figura 25.**Diagrama de flujo consulta de ventas

**Diagrama de flujo consulta de ingresos.** – los empleados encargados del módulo de consulta de ingresos, ingresará al sistema con sus respectivas credenciales, seleccionará la interfaz de este, para poderlas ingresar o actualizar las diferentes consultas de ingresos de la empresa.



**Figura 26.**Diagrama de flujo consulta de ingresos

**Diagrama de flujo consulta de accesos.** – los empleados encargados del módulo de acceso deberán ingresar al sistema con sus respectivas credenciales, seleccionará la interfaz de este, para poderlas unirse o actualizar las consultas de acceso.



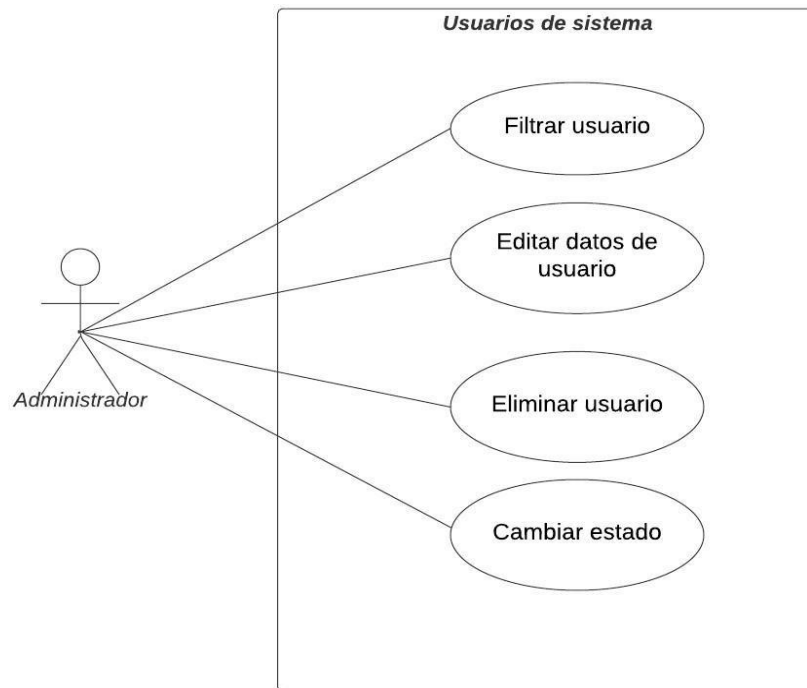
**Figura 27.**Diagrama de flujo clientes

### Casos de uso

Los casos de uso que se mostrara explicaran todas las acciones que realizara el sistema para llevar a cabo los métodos de ventas e inventarios para la empresa RIVATEX.

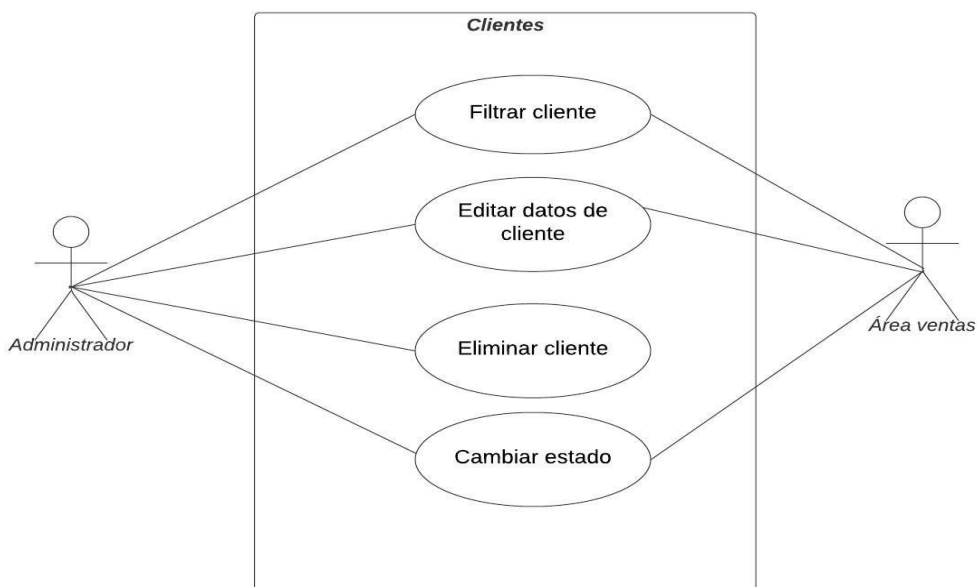
**Caso de uso usuario.** – El administrador tiene acceso a todas las funciones del sistema y puede realizar las siguientes actividades: Filtrar, editar, eliminar y cambiar estado entre otras, en la cual se trasforma en actores primarios del sistema mientras

que los actores secundarios son: Encargados de inventarios y encargados de ventas.



**Figura 28.**Diagrama de flujo clientes

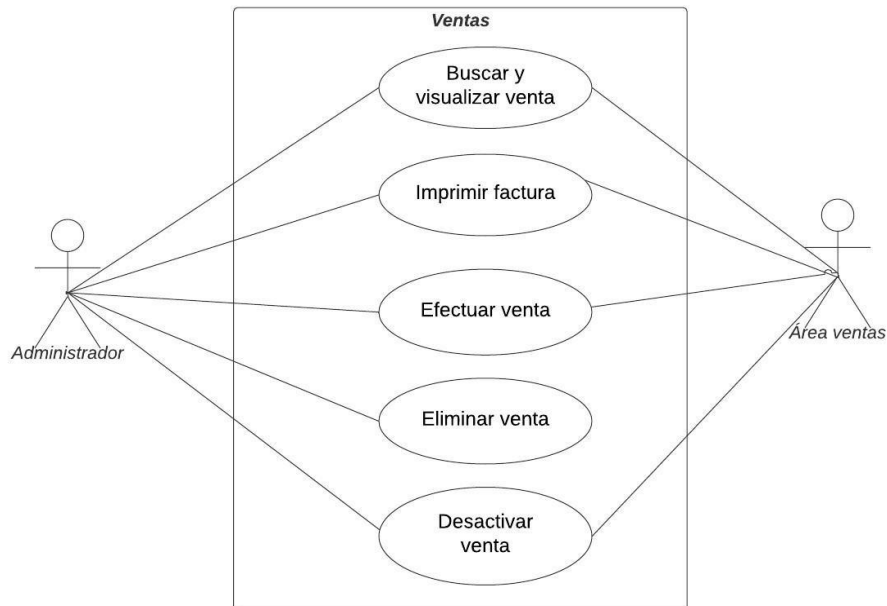
**Caso de uso clientes.** –los empleados encargados de área de ventas pueden acceder al sistema donde realizan las operaciones de: registrar, editar, cambiar de estado a excepción de eliminar cliente.



**Figura 29.**Caso de uso Usuario de sistema

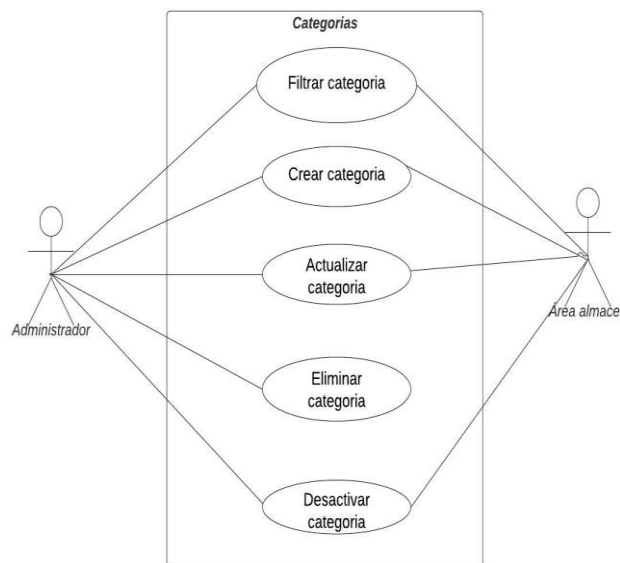


**Caso de uso ventas.** – los empleados encargados de área de ventas pueden acceder al sistema donde realizan las operaciones de: búsqueda de datos, imprimir factura, efectuar venta, desactivar venta a excepción de eliminar venta.



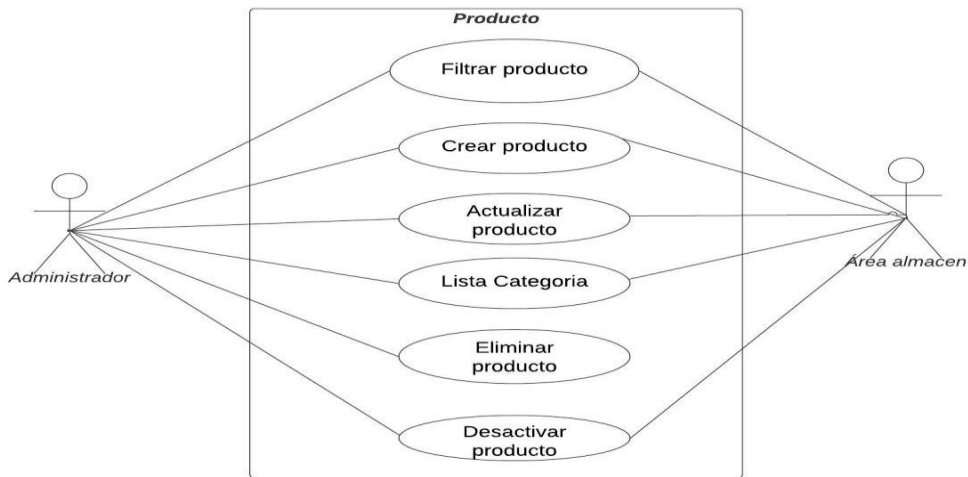
**Figura 30.**Caso de uso Usuario de sistema

**Caso de uso categorías.** – los empleados encargados de área de almacén pueden acceder al sistema donde realizan las operaciones de: búsqueda de datos, crear, actualizar, desactivar excepción de eliminar categoría.



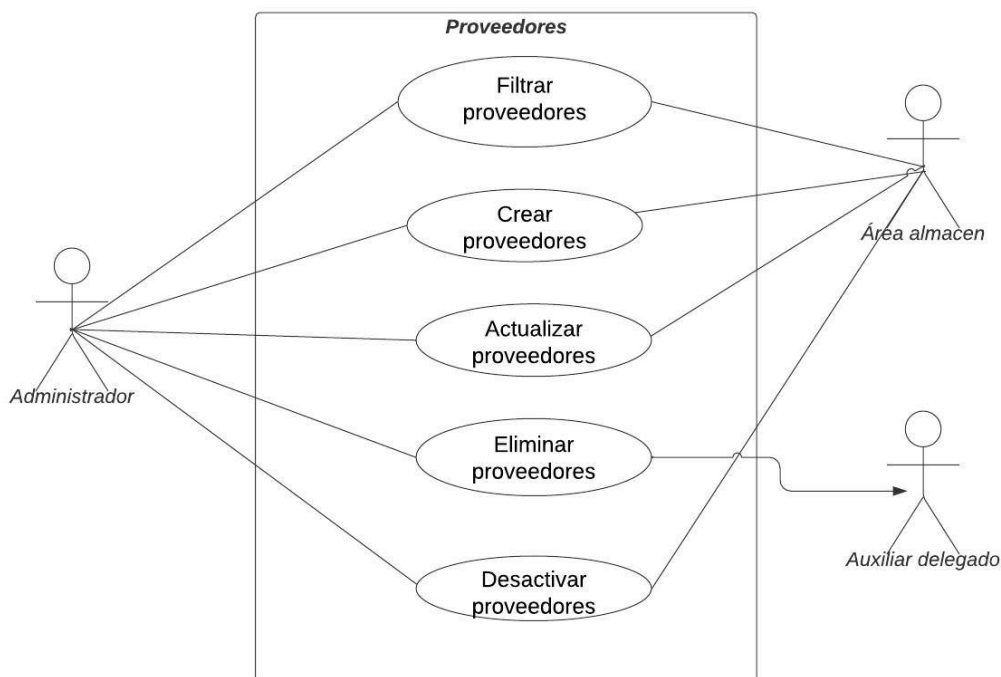
**Figura 31.**Caso de uso categorías

**Caso de uso productos.** – los empleados encargados de área de almacén pueden acceder al sistema donde realizan las operaciones de: búsqueda de producto, crear, actualizar, lista categoría, desactivar producto excepción de eliminar producto.



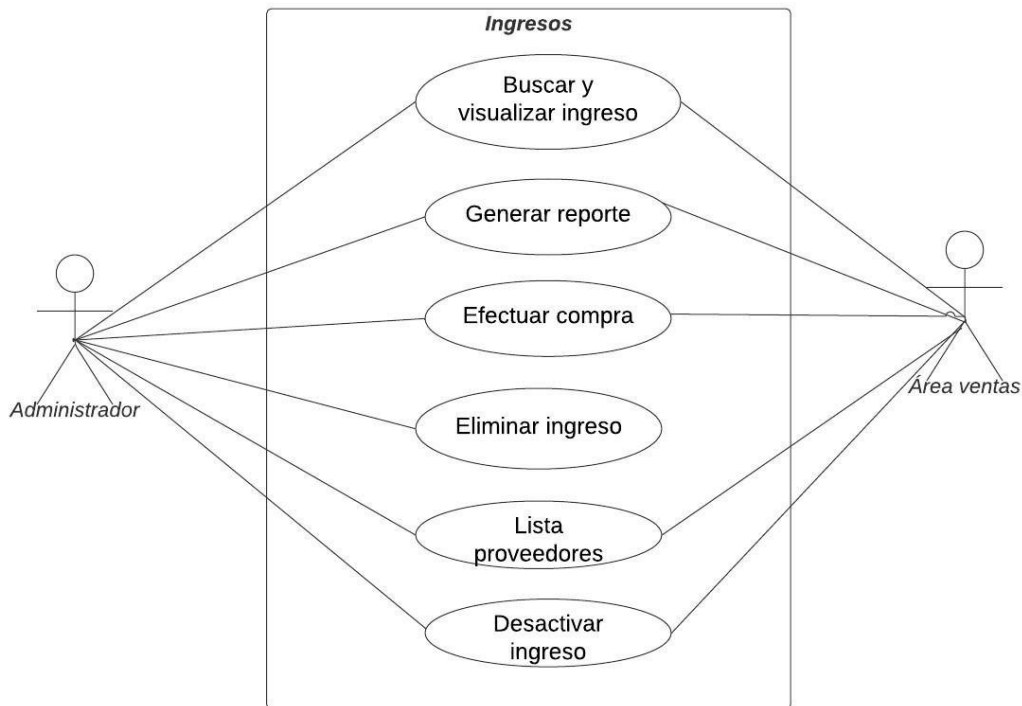
**Figura 32.**Caso de uso productos

**Caso de uso proveedores.** – los empleados encargados de área de almacén pueden acceder al sistema donde realizan las operaciones de: búsqueda de proveedores, crear, actualizar, desactivar excepción de eliminar proveedores.



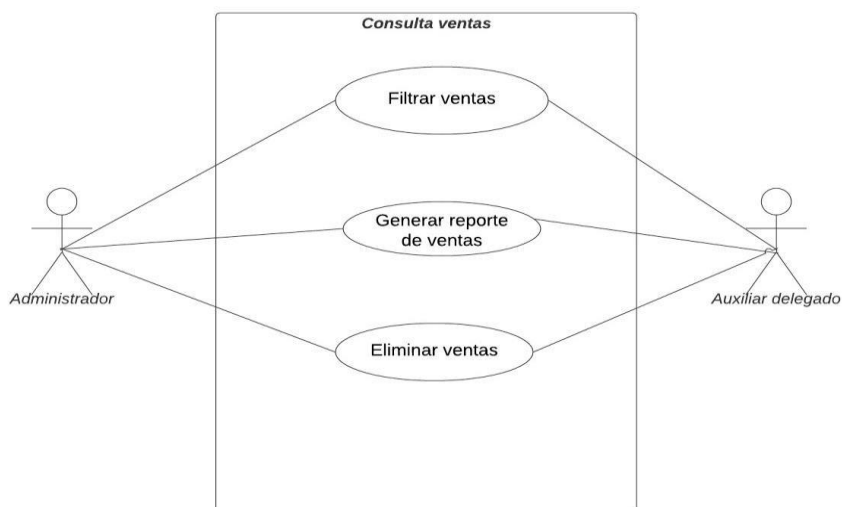
**Figura 33.**Caso de uso productos

**Caso de uso ingresos.** – los empleados encargados de área de almacén pueden acceder al sistema donde realizan las operaciones de: búsqueda de ingresos, generar reporte, efectuar compras, lista de proveedores, desactivar ingresos con la excepción de eliminar ingresos.



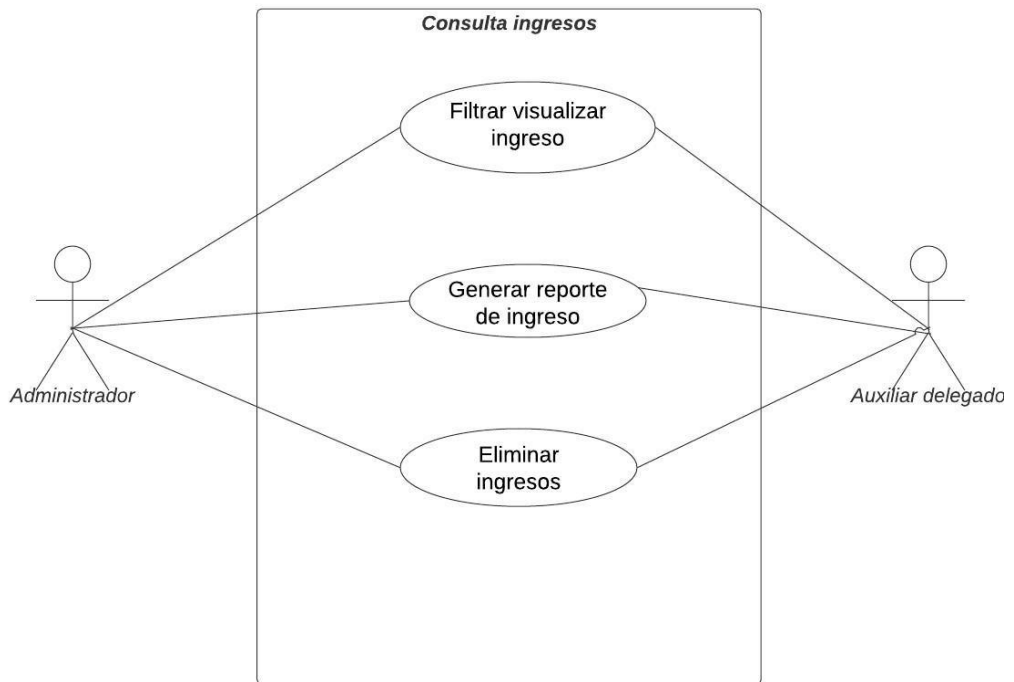
**Figura 34.**Caso de uso ingresos

**Caso de uso consultas compras.** – el auxiliar delegado por el administrador tiene acceso a consulta de las ventas donde pondrá filtrar las ventas de todos los meses, generar y eliminar ventas.



**Figura 35.**Caso de uso consultas compras

**Caso de uso consulta ingreso.** – El auxiliar delegado por el administrador tiene acceso a consulta de ingreso donde pondrá filtrar los ingresos de todos los meses, generar y eliminar ingresos.



**Figura 36.** Caso de uso consulta ingreso

**Caso de uso barras.** – todos los usuarios del sistema pueden observar la estadística de los datos de las ventas generales.



**Figura 37.** Caso de uso barras

## Diseño de base de datos

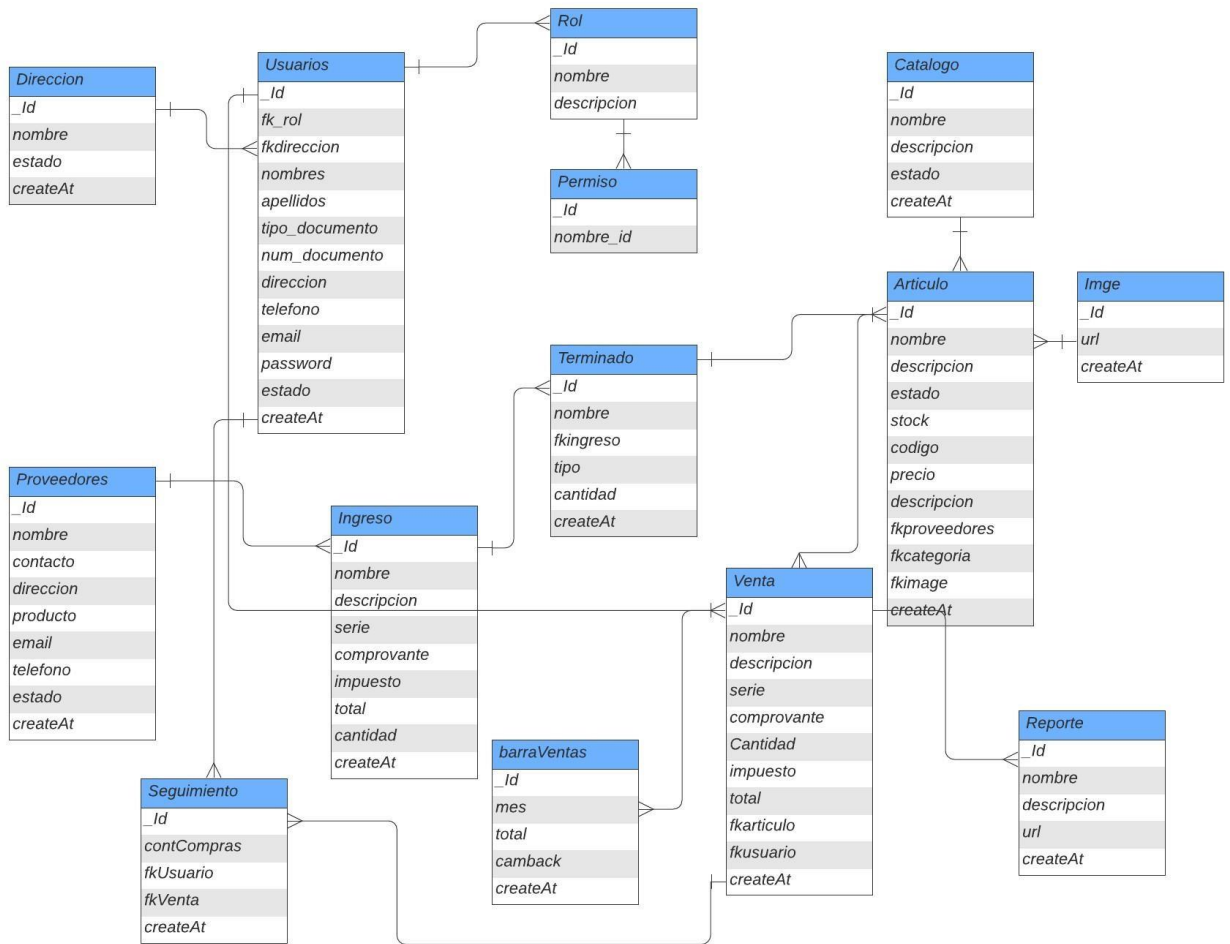


Figura 38. Diseño de base de datos

### 4.2.3.5 Diseño de interfaces

**Interfaz de Login.** Esta interfaz es donde el administrador y empleados encargados de las áreas ventas e inventario logren tener acceso al sistema con sus respectivas credenciales

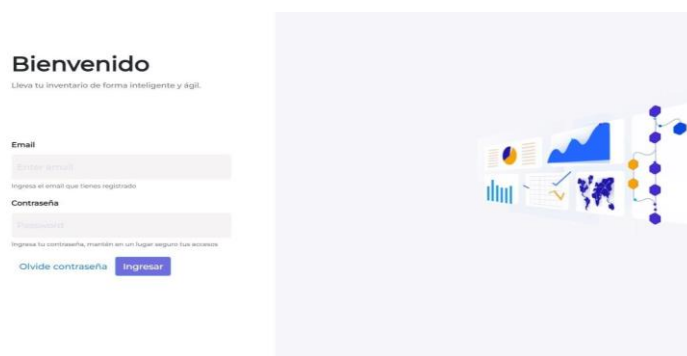


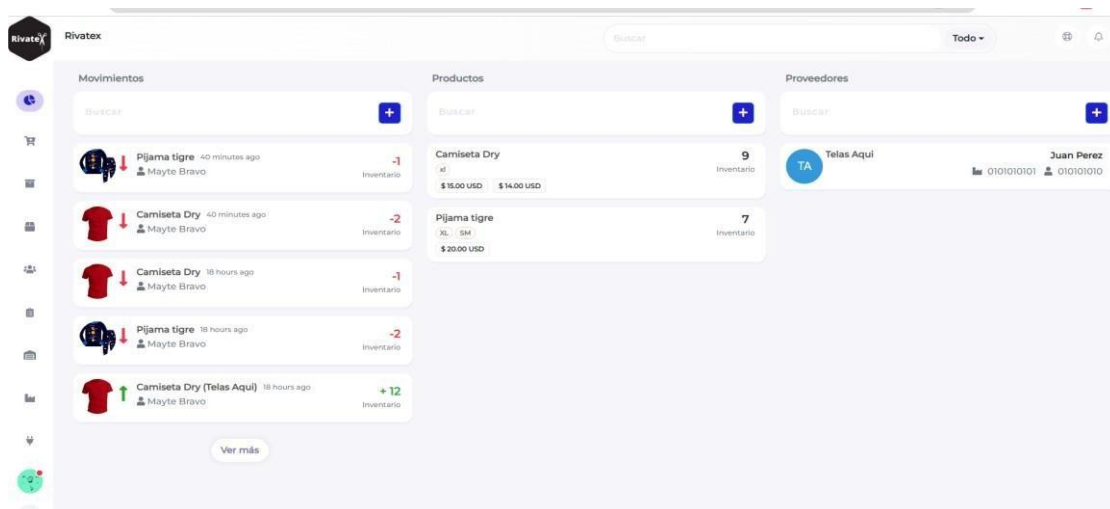
Figura 39. Login principal

**Interfaz de menú.** En esta interfaz podemos apreciar todas las opciones que tiene el sistema para todas las áreas un que su acceso es dependiendo del rol y los permisos que tengan.



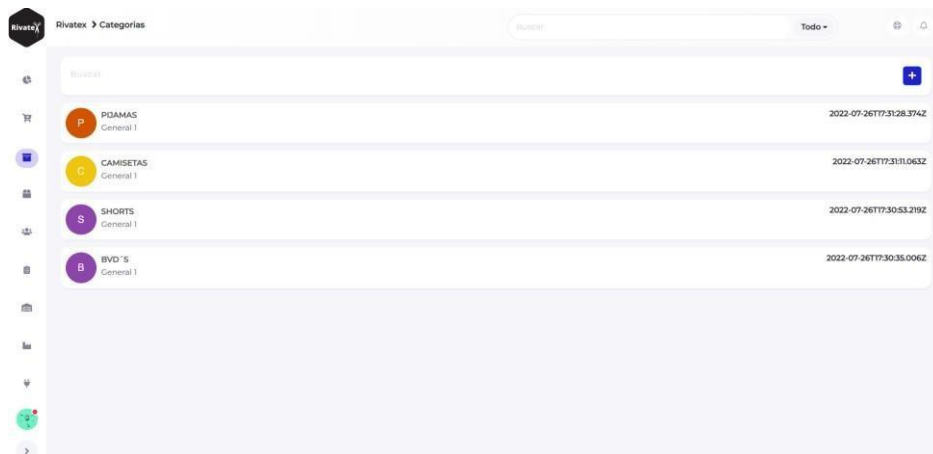
**Figura 40.**Menú de opciones

**Interfaz de inicio.** En esta interfaz se puede observar los datos estadísticos de las ventas de cada mes generado y analizado por el sistema.



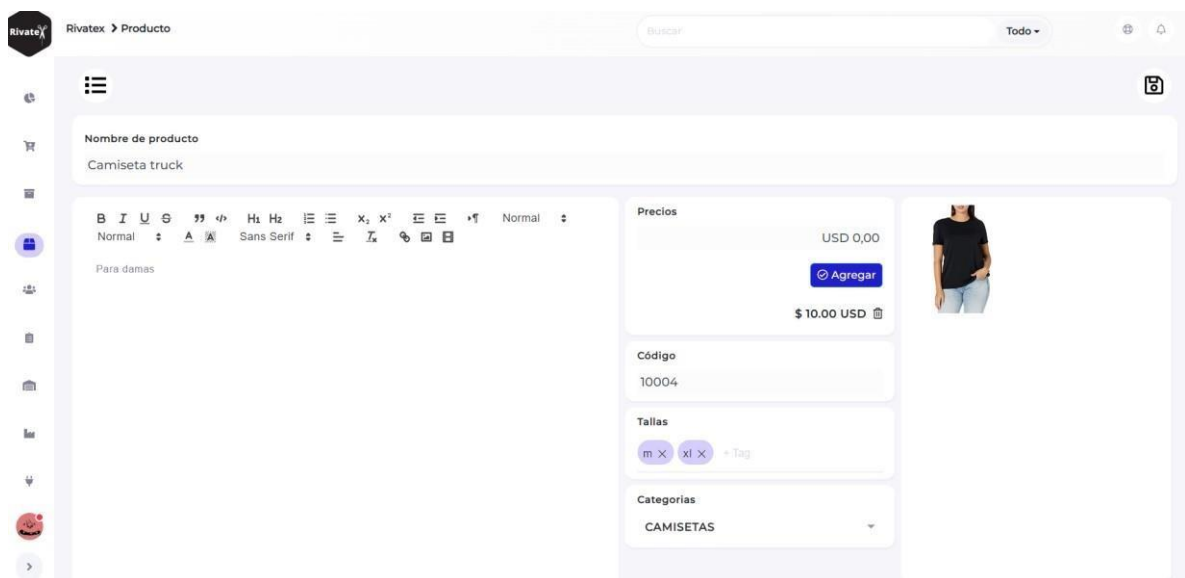
**Figura 41.**Inicio de sistema

**Interfaz de categorías.** En esta interfaz se puede observar toda la lista de categorías existentes con opciones para realizar las operaciones de mantenimiento.



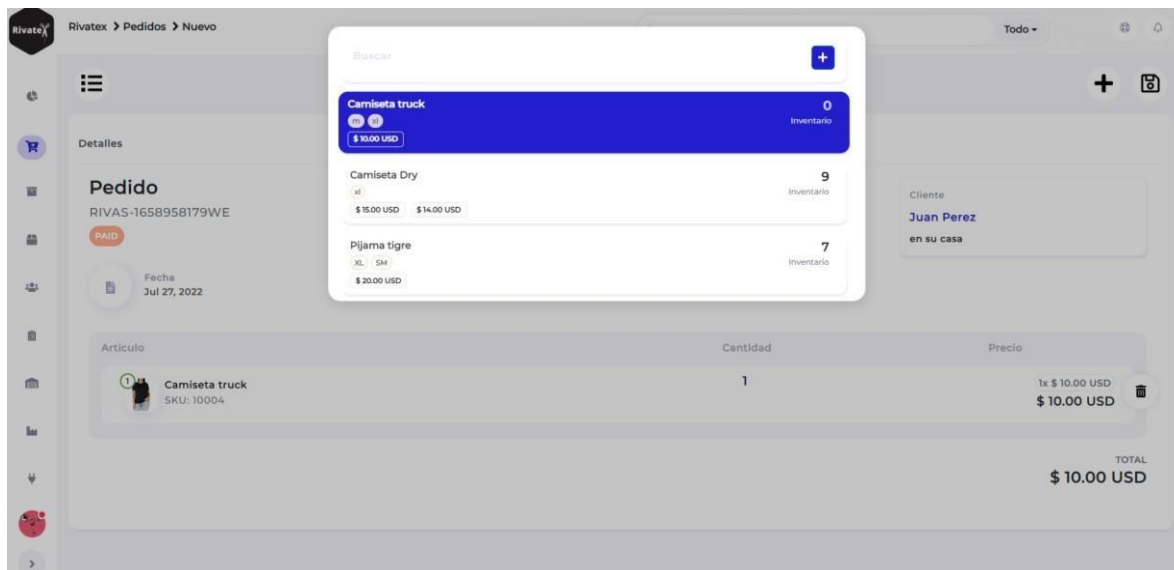
**Figura 42.**Lista categorías

**Interfaz de crear producto.** En esta interfaz se puede realizar el registro de un producto que ofrece la empresa.



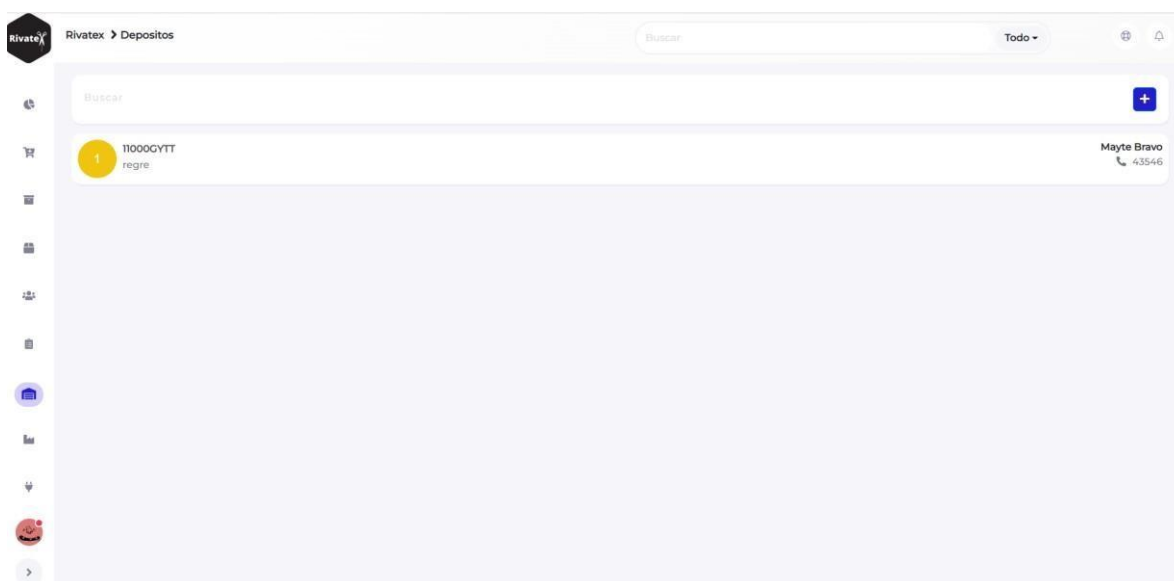
**Figura 43.**Registro de productos

**Interfaz de crear ingresos.** En esta interfaz se puede realizar el registro completo de todos los ingresos o productos de materia prima necesarios para empresa.



**Figura 44.** Interfaz de crear ingresos.

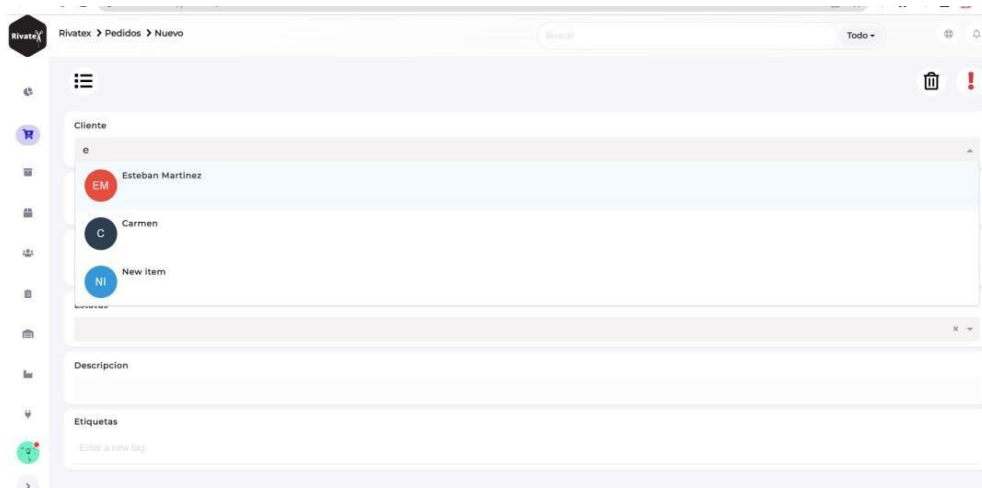
**Interfaz de proveedores.** En esta interfaz se puede observar toda la lista de proveedores con opciones para realizar las operaciones de mantenimiento y agregar un nuevo proveedor.



**Figura 45.** Interfaz de proveedores

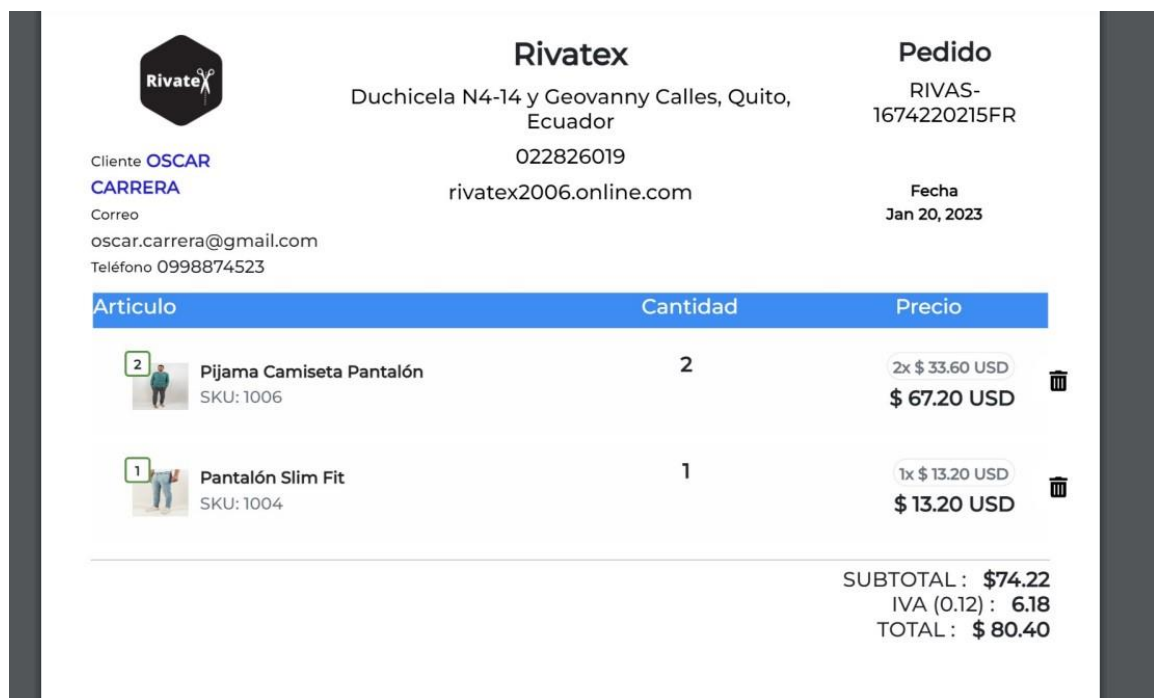
**Interfaz de formulario ventas.** En esta interfaz se puede realizar el registro de las ventas que se efectúan diariamente almacenado de forma ordenada en la base de datos.





**Figura 46.**formulario ventas

**Interfaz de factura.** En esta interfaz se puede imprimir el comprobante de venta es decir la respectiva factura que será entregada y enviada al correo electrónico.



**Figura 47.**Interfaz de factura.

#### 4.2.3.6 fases de construcción

En esta fase se demostrará capturas de pantalla del sistema informático con su respectivo código.

## Componente inicio de sesión

```
3  */
4  exports.login = async (req, res) => {
5    try {
6      const tenant = req.clientAccount;
7      const data = matchedData(req)
8      const user = await findUser(data.email, tenant)
9      await userIsBlocked(user)
10     await checkLoginAttemptsAndBlockExpires(user)
11     const isPasswordMatch = await auth.checkPassword(data.password, user)
12     if (!isPasswordMatch) {
13       utils.handleError(res, await passwordsDoNotMatch(user))
14     } else {
15       // all ok, register access and return token
16       user.loginAttempts = 0
17       await saveLoginAttemptsToDB(user)
18       res.status(200).json(await saveUserAccessAndReturnToken(req, user))
19     }
20   } catch (error) {
21     utils.handleError(res, error)
22   }
23 }
```

Figura 48. Componente inicio de sesión

## Componente menú

```
div class="d-flex flex-column h-100 justify-content-between">
<div>
  <div [routerLink]="['/']">
    <div class="logo">
      <fa-icon *ngIf="!logo" [icon]="faCrown" [style.color]='orange' size="lg"></fa-icon>
      <span><img style="width: 50px !important;" [src]="logo" *ngIf="logo" [alt]="name"></span>
    </div>
  </div>
  <div class="menu-list text-center">
    <ul class="pt-4">
      <ng-container *ngFor="let a of menu">
        <li *ngIf="!a?.disable" class="pl-1 pr-1 pb-1">
          placement="top"
          [tooltip]="tooltTemplate" containerClass="customToolTip"
          [routerLink]="a.route"
          [delay]="115">
            <ng-template #tooltTemplate>{{a?.name}}</ng-template>
            <div class="p-2 icon-wrapper">
              <div class="p-1 sub-wrapper" [routerLinkActive]="['active']">
                <fa-icon class="icon" [icon]="a.icon"></fa-icon>
              </div>
            </div>
          </li>
        </ng-container>
      </ul>
    </div>
  </div>
</div>
```

Figura 49. Componente menú

## Componente estadístico

```
52  async getItemAggregate(aggregate = {}, model, tenant = null) {
53    return new Promise((resolve, reject) => {
54      model.byTenant(tenant).aggregate(aggregate, (err, item) => {
55        itemNotFound(err, item, reject, 'NOT_FOUND')
56        resolve(item.find(a => true))
57      });
58    });
59  },
60
61  /**
62   * Gets item from database by id
63   * @param {string} id - item id
64   * @param model
65   * @param tenant
66   */
67  async getItem(id, model, tenant = null) {
68    return new Promise((resolve, reject) => {
69      model.byTenant(tenant).findById(id, (err, item) => {
70        itemNotFound(err, item, reject, 'NOT_FOUND')
71        resolve(item)
72      });
73    });
74  },
75
76  /**
77   * Gets item from database by id
78   * @param {string} id - item id
79   * @param model
80   * @param tenant
81   */
82  async getItem(id, model, tenant = null) {
83    return new Promise((resolve, reject) => {
84      model.byTenant(tenant).findById(id, (err, item) => {
85        itemNotFound(err, item, reject, 'NOT_FOUND')
86        resolve(item)
87      });
88    });
89  },
90
91  /**
92   * Gets item from database by id
93   * @param {string} id - item id
94   * @param model
95   * @param tenant
96   */
97  async getItem(id, model, tenant = null) {
98    return new Promise((resolve, reject) => {
99      model.byTenant(tenant).findById(id, (err, item) => {
100        itemNotFound(err, item, reject, 'NOT_FOUND')
101        resolve(item)
102      });
103    });
104  },
105
106  /**
107   * Gets item from database by id
108   * @param {string} id - item id
109   * @param model
110   * @param tenant
111   */
112  async getItem(id, model, tenant = null) {
113    return new Promise((resolve, reject) => {
114      model.byTenant(tenant).findById(id, (err, item) => {
115        itemNotFound(err, item, reject, 'NOT_FOUND')
116        resolve(item)
117      });
118    });
119  },
120
121  /**
122   * Gets item from database by id
123   * @param {string} id - item id
124   * @param model
125   * @param tenant
126   */
127  async getItem(id, model, tenant = null) {
128    return new Promise((resolve, reject) => {
129      model.byTenant(tenant).findById(id, (err, item) => {
130        itemNotFound(err, item, reject, 'NOT_FOUND')
131        resolve(item)
132      });
133    });
134  },
135
136  /**
137   * Gets item from database by id
138   * @param {string} id - item id
139   * @param model
140   * @param tenant
141   */
142  async getItem(id, model, tenant = null) {
143    return new Promise((resolve, reject) => {
144      model.byTenant(tenant).findById(id, (err, item) => {
145        itemNotFound(err, item, reject, 'NOT_FOUND')
146        resolve(item)
147      });
148    });
149  },
150
151  /**
152   * Gets item from database by id
153   * @param {string} id - item id
154   * @param model
155   * @param tenant
156   */
157  async getItem(id, model, tenant = null) {
158    return new Promise((resolve, reject) => {
159      model.byTenant(tenant).findById(id, (err, item) => {
160        itemNotFound(err, item, reject, 'NOT_FOUND')
161        resolve(item)
162      });
163    });
164  },
165
166  /**
167   * Gets item from database by id
168   * @param {string} id - item id
169   * @param model
170   * @param tenant
171   */
172  async getItem(id, model, tenant = null) {
173    return new Promise((resolve, reject) => {
174      model.byTenant(tenant).findById(id, (err, item) => {
175        itemNotFound(err, item, reject, 'NOT_FOUND')
176        resolve(item)
177      });
178    });
179  },
180
181  /**
182   * Gets item from database by id
183   * @param {string} id - item id
184   * @param model
185   * @param tenant
186   */
187  async getItem(id, model, tenant = null) {
188    return new Promise((resolve, reject) => {
189      model.byTenant(tenant).findById(id, (err, item) => {
190        itemNotFound(err, item, reject, 'NOT_FOUND')
191        resolve(item)
192      });
193    });
194  },
195
196  /**
197   * Gets item from database by id
198   * @param {string} id - item id
199   * @param model
200   * @param tenant
201   */
202  async getItem(id, model, tenant = null) {
203    return new Promise((resolve, reject) => {
204      model.byTenant(tenant).findById(id, (err, item) => {
205        itemNotFound(err, item, reject, 'NOT_FOUND')
206        resolve(item)
207      });
208    });
209  },
210
211  /**
212   * Gets item from database by id
213   * @param {string} id - item id
214   * @param model
215   * @param tenant
216   */
217  async getItem(id, model, tenant = null) {
218    return new Promise((resolve, reject) => {
219      model.byTenant(tenant).findById(id, (err, item) => {
220        itemNotFound(err, item, reject, 'NOT_FOUND')
221        resolve(item)
222      });
223    });
224  },
225
226  /**
227   * Gets item from database by id
228   * @param {string} id - item id
229   * @param model
230   * @param tenant
231   */
232  async getItem(id, model, tenant = null) {
233    return new Promise((resolve, reject) => {
234      model.byTenant(tenant).findById(id, (err, item) => {
235        itemNotFound(err, item, reject, 'NOT_FOUND')
236        resolve(item)
237      });
238    });
239  },
240
241  /**
242   * Gets item from database by id
243   * @param {string} id - item id
244   * @param model
245   * @param tenant
246   */
247  async getItem(id, model, tenant = null) {
248    return new Promise((resolve, reject) => {
249      model.byTenant(tenant).findById(id, (err, item) => {
250        itemNotFound(err, item, reject, 'NOT_FOUND')
251        resolve(item)
252      });
253    });
254  },
255
256  /**
257   * Gets item from database by id
258   * @param {string} id - item id
259   * @param model
260   * @param tenant
261   */
262  async getItem(id, model, tenant = null) {
263    return new Promise((resolve, reject) => {
264      model.byTenant(tenant).findById(id, (err, item) => {
265        itemNotFound(err, item, reject, 'NOT_FOUND')
266        resolve(item)
267      });
268    });
269  },
270
271  /**
272   * Gets item from database by id
273   * @param {string} id - item id
274   * @param model
275   * @param tenant
276   */
277  async getItem(id, model, tenant = null) {
278    return new Promise((resolve, reject) => {
279      model.byTenant(tenant).findById(id, (err, item) => {
280        itemNotFound(err, item, reject, 'NOT_FOUND')
281        resolve(item)
282      });
283    });
284  },
285
286  /**
287   * Gets item from database by id
288   * @param {string} id - item id
289   * @param model
290   * @param tenant
291   */
292  async getItem(id, model, tenant = null) {
293    return new Promise((resolve, reject) => {
294      model.byTenant(tenant).findById(id, (err, item) => {
295        itemNotFound(err, item, reject, 'NOT_FOUND')
296        resolve(item)
297      });
298    });
299  },
300
301  /**
302   * Gets item from database by id
303   * @param {string} id - item id
304   * @param model
305   * @param tenant
306   */
307  async getItem(id, model, tenant = null) {
308    return new Promise((resolve, reject) => {
309      model.byTenant(tenant).findById(id, (err, item) => {
310        itemNotFound(err, item, reject, 'NOT_FOUND')
311        resolve(item)
312      });
313    });
314  },
315
316  /**
317   * Gets item from database by id
318   * @param {string} id - item id
319   * @param model
320   * @param tenant
321   */
322  async getItem(id, model, tenant = null) {
323    return new Promise((resolve, reject) => {
324      model.byTenant(tenant).findById(id, (err, item) => {
325        itemNotFound(err, item, reject, 'NOT_FOUND')
326        resolve(item)
327      });
328    });
329  },
330
331  /**
332   * Gets item from database by id
333   * @param {string} id - item id
334   * @param model
335   * @param tenant
336   */
337  async getItem(id, model, tenant = null) {
338    return new Promise((resolve, reject) => {
339      model.byTenant(tenant).findById(id, (err, item) => {
340        itemNotFound(err, item, reject, 'NOT_FOUND')
341        resolve(item)
342      });
343    });
344  },
345
346  /**
347   * Gets item from database by id
348   * @param {string} id - item id
349   * @param model
350   * @param tenant
351   */
352  async getItem(id, model, tenant = null) {
353    return new Promise((resolve, reject) => {
354      model.byTenant(tenant).findById(id, (err, item) => {
355        itemNotFound(err, item, reject, 'NOT_FOUND')
356        resolve(item)
357      });
358    });
359  },
360
361  /**
362   * Gets item from database by id
363   * @param {string} id - item id
364   * @param model
365   * @param tenant
366   */
367  async getItem(id, model, tenant = null) {
368    return new Promise((resolve, reject) => {
369      model.byTenant(tenant).findById(id, (err, item) => {
370        itemNotFound(err, item, reject, 'NOT_FOUND')
371        resolve(item)
372      });
373    });
374  },
375
376  /**
377   * Gets item from database by id
378   * @param {string} id - item id
379   * @param model
380   * @param tenant
381   */
382  async getItem(id, model, tenant = null) {
383    return new Promise((resolve, reject) => {
384      model.byTenant(tenant).findById(id, (err, item) => {
385        itemNotFound(err, item, reject, 'NOT_FOUND')
386        resolve(item)
387      });
388    });
389  },
390
391  /**
392   * Gets item from database by id
393   * @param {string} id - item id
394   * @param model
395   * @param tenant
396   */
397  async getItem(id, model, tenant = null) {
398    return new Promise((resolve, reject) => {
399      model.byTenant(tenant).findById(id, (err, item) => {
400        itemNotFound(err, item, reject, 'NOT_FOUND')
401        resolve(item)
402      });
403    });
404  },
405
406  /**
407   * Gets item from database by id
408   * @param {string} id - item id
409   * @param model
410   * @param tenant
411   */
412  async getItem(id, model, tenant = null) {
413    return new Promise((resolve, reject) => {
414      model.byTenant(tenant).findById(id, (err, item) => {
415        itemNotFound(err, item, reject, 'NOT_FOUND')
416        resolve(item)
417      });
418    });
419  },
420
421  /**
422   * Gets item from database by id
423   * @param {string} id - item id
424   * @param model
425   * @param tenant
426   */
427  async getItem(id, model, tenant = null) {
428    return new Promise((resolve, reject) => {
429      model.byTenant(tenant).findById(id, (err, item) => {
430        itemNotFound(err, item, reject, 'NOT_FOUND')
431        resolve(item)
432      });
433    });
434  },
435
436  /**
437   * Gets item from database by id
438   * @param {string} id - item id
439   * @param model
440   * @param tenant
441   */
442  async getItem(id, model, tenant = null) {
443    return new Promise((resolve, reject) => {
444      model.byTenant(tenant).findById(id, (err, item) => {
445        itemNotFound(err, item, reject, 'NOT_FOUND')
446        resolve(item)
447      });
448    });
449  },
450
451  /**
452   * Gets item from database by id
453   * @param {string} id - item id
454   * @param model
455   * @param tenant
456   */
457  async getItem(id, model, tenant = null) {
458    return new Promise((resolve, reject) => {
459      model.byTenant(tenant).findById(id, (err, item) => {
460        itemNotFound(err, item, reject, 'NOT_FOUND')
461        resolve(item)
462      });
463    });
464  },
465
466  /**
467   * Gets item from database by id
468   * @param {string} id - item id
469   * @param model
470   * @param tenant
471   */
472  async getItem(id, model, tenant = null) {
473    return new Promise((resolve, reject) => {
474      model.byTenant(tenant).findById(id, (err, item) => {
475        itemNotFound(err, item, reject, 'NOT_FOUND')
476        resolve(item)
477      });
478    });
479  },
480
481  /**
482   * Gets item from database by id
483   * @param {string} id - item id
484   * @param model
485   * @param tenant
486   */
487  async getItem(id, model, tenant = null) {
488    return new Promise((resolve, reject) => {
489      model.byTenant(tenant).findById(id, (err, item) => {
490        itemNotFound(err, item, reject, 'NOT_FOUND')
491        resolve(item)
492      });
493    });
494  },
495
496  /**
497   * Gets item from database by id
498   * @param {string} id - item id
499   * @param model
500   * @param tenant
501   */
502  async getItem(id, model, tenant = null) {
503    return new Promise((resolve, reject) => {
504      model.byTenant(tenant).findById(id, (err, item) => {
505        itemNotFound(err, item, reject, 'NOT_FOUND')
506        resolve(item)
507      });
508    });
509  },
510
511  /**
512   * Gets item from database by id
513   * @param {string} id - item id
514   * @param model
515   * @param tenant
516   */
517  async getItem(id, model, tenant = null) {
518    return new Promise((resolve, reject) => {
519      model.byTenant(tenant).findById(id, (err, item) => {
520        itemNotFound(err, item, reject, 'NOT_FOUND')
521        resolve(item)
522      });
523    });
524  },
525
526  /**
527   * Gets item from database by id
528   * @param {string} id - item id
529   * @param model
530   * @param tenant
531   */
532  async getItem(id, model, tenant = null) {
533    return new Promise((resolve, reject) => {
534      model.byTenant(tenant).findById(id, (err, item) => {
535        itemNotFound(err, item, reject, 'NOT_FOUND')
536        resolve(item)
537      });
538    });
539  },
540
541  /**
542   * Gets item from database by id
543   * @param {string} id - item id
544   * @param model
545   * @param tenant
546   */
547  async getItem(id, model, tenant = null) {
548    return new Promise((resolve, reject) => {
549      model.byTenant(tenant).findById(id, (err, item) => {
550        itemNotFound(err, item, reject, 'NOT_FOUND')
551        resolve(item)
552      });
553    });
554  },
555
556  /**
557   * Gets item from database by id
558   * @param {string} id - item id
559   * @param model
560   * @param tenant
561   */
562  async getItem(id, model, tenant = null) {
563    return new Promise((resolve, reject) => {
564      model.byTenant(tenant).findById(id, (err, item) => {
565        itemNotFound(err, item, reject, 'NOT_FOUND')
566        resolve(item)
567      });
568    });
569  },
570
571  /**
572   * Gets item from database by id
573   * @param {string} id - item id
574   * @param model
575   * @param tenant
576   */
577  async getItem(id, model, tenant = null) {
578    return new Promise((resolve, reject) => {
579      model.byTenant(tenant).findById(id, (err, item) => {
580        itemNotFound(err, item, reject, 'NOT_FOUND')
581        resolve(item)
582      });
583    });
584  },
585
586  /**
587   * Gets item from database by id
588   * @param {string} id - item id
589   * @param model
590   * @param tenant
591   */
592  async getItem(id, model, tenant = null) {
593    return new Promise((resolve, reject) => {
594      model.byTenant(tenant).findById(id, (err, item) => {
595        itemNotFound(err, item, reject, 'NOT_FOUND')
596        resolve(item)
597      });
598    });
599  },
600
601  /**
602   * Gets item from database by id
603   * @param {string} id - item id
604   * @param model
605   * @param tenant
606   */
607  async getItem(id, model, tenant = null) {
608    return new Promise((resolve, reject) => {
609      model.byTenant(tenant).findById(id, (err, item) => {
610        itemNotFound(err, item, reject, 'NOT_FOUND')
611        resolve(item)
612      });
613    });
614  },
615
616  /**
617   * Gets item from database by id
618   * @param {string} id - item id
619   * @param model
620   * @param tenant
621   */
622  async getItem(id, model, tenant = null) {
623    return new Promise((resolve, reject) => {
624      model.byTenant(tenant).findById(id, (err, item) => {
625        itemNotFound(err, item, reject, 'NOT_FOUND')
626        resolve(item)
627      });
628    });
629  },
630
631  /**
632   * Gets item from database by id
633   * @param {string} id - item id
634   * @param model
635   * @param tenant
636   */
637  async getItem(id, model, tenant = null) {
638    return new Promise((resolve, reject) => {
639      model.byTenant(tenant).findById(id, (err, item) => {
640        itemNotFound(err, item, reject, 'NOT_FOUND')
641        resolve(item)
642      });
643    });
644  },
645
646  /**
647   * Gets item from database by id
648   * @param {string} id - item id
649   * @param model
650   * @param tenant
651   */
652  async getItem(id, model, tenant = null) {
653    return new Promise((resolve, reject) => {
654      model.byTenant(tenant).findById(id, (err, item) => {
655        itemNotFound(err, item, reject, 'NOT_FOUND')
656        resolve(item)
657      });
658    });
659  },
660
661  /**
662   * Gets item from database by id
663   * @param {string} id - item id
664   * @param model
665   * @param tenant
666   */
667  async getItem(id, model, tenant = null) {
668    return new Promise((resolve, reject) => {
669      model.byTenant(tenant).findById(id, (err, item) => {
670        itemNotFound(err, item, reject, 'NOT_FOUND')
671        resolve(item)
672      });
673    });
674  },
675
676  /**
677   * Gets item from database by id
678   * @param {string} id - item id
679   * @param model
680   * @param tenant
681   */
682  async getItem(id, model, tenant = null) {
683    return new Promise((resolve, reject) => {
684      model.byTenant(tenant).findById(id, (err, item) => {
685        itemNotFound(err, item, reject, 'NOT_FOUND')
686        resolve(item)
687      });
688    });
689  },
690
691  /**
692   * Gets item from database by id
693   * @param {string} id - item id
694   * @param model
695   * @param tenant
696   */
697  async getItem(id, model, tenant = null) {
698    return new Promise((resolve, reject) => {
699      model.byTenant(tenant).findById(id, (err, item) => {
700        itemNotFound(err, item, reject, 'NOT_FOUND')
701        resolve(item)
702      });
703    });
704  },
705
706  /**
707   * Gets item from database by id
708   * @param {string} id - item id
709   * @param model
710   * @param tenant
711   */
712  async getItem(id, model, tenant = null) {
713    return new Promise((resolve, reject) => {
714      model.byTenant(tenant).findById(id, (err, item) => {
715        itemNotFound(err, item, reject, 'NOT_FOUND')
716        resolve(item)
717      });
718    });
719  },
720
721  /**
722   * Gets item from database by id
723   * @param {string} id - item id
724   * @param model
725   * @param tenant
726   */
727  async getItem(id, model, tenant = null) {
728    return new Promise((resolve, reject) => {
729      model.byTenant(tenant).findById(id, (err, item) => {
730        itemNotFound(err, item, reject, 'NOT_FOUND')
731        resolve(item)
732      });
733    });
734  },
735
736  /**
737   * Gets item from database by id
738   * @param {string} id - item id
739   * @param model
740   * @param tenant
741   */
742  async getItem(id, model, tenant = null) {
743    return new Promise((resolve, reject) => {
744      model.byTenant(tenant).findById(id, (err, item) => {
745        itemNotFound(err, item, reject, 'NOT_FOUND')
746        resolve(item)
747      });
748    });
749  },
750
751  /**
752   * Gets item from database by id
753   * @param {string} id - item id
754   * @param model
755   * @param tenant
756   */
757  async getItem(id, model, tenant = null) {
758    return new Promise((resolve, reject) => {
759      model.byTenant(tenant).findById(id, (err, item) => {
760        itemNotFound(err, item, reject, 'NOT_FOUND')
761        resolve(item)
762      });
763    });
764  },
765
766  /**
767   * Gets item from database by id
768   * @param {string} id - item id
769   * @param model
770   * @param tenant
771   */
772  async getItem(id, model, tenant = null) {
773    return new Promise((resolve, reject) => {
774      model.byTenant(tenant).findById(id, (err, item) => {
775        itemNotFound(err, item, reject, 'NOT_FOUND')
776        resolve(item)
777      });
778    });
779  },
780
781  /**
782   * Gets item from database by id
783   * @param {string} id - item id
784   * @param model
785   * @param tenant
786   */
787  async getItem(id, model, tenant = null) {
788    return new Promise((resolve, reject) => {
789      model.byTenant(tenant).findById(id, (err, item) => {
790        itemNotFound(err, item, reject, 'NOT_FOUND')
791        resolve(item)
792      });
793    });
794  },
795
796  /**
797   * Gets item from database by id
798   * @param {string} id - item id
799   * @param model
800   * @param tenant
801   */
802  async getItem(id, model, tenant = null) {
803    return new Promise((resolve, reject) => {
804      model.byTenant(tenant).findById(id, (err, item) => {
805        itemNotFound(err, item, reject, 'NOT_FOUND')
806        resolve(item)
807      });
808    });
809  },
810
811  /**
812   * Gets item from database by id
813   * @param {string} id - item id
814   * @param model
815   * @param tenant
816   */
817  async getItem(id, model, tenant = null) {
818    return new Promise((resolve, reject) => {
819      model.byTenant(tenant).findById(id, (err, item) => {
820        itemNotFound(err, item, reject, 'NOT_FOUND')
821        resolve(item)
822      });
823    });
824  },
825
826  /**
827   * Gets item from database by id
828   * @param {string} id - item id
829   * @param model
830   * @param tenant
831   */
832  async getItem(id, model, tenant = null) {
833    return new Promise((resolve, reject) => {
834      model.byTenant(tenant).findById(id, (err, item) => {
835        itemNotFound(err, item, reject, 'NOT_FOUND')
836        resolve(item)
837      });
838    });
839  },
840
841  /**
842   * Gets item from database by id
843   * @param {string} id - item id
844   * @param model
845   * @param tenant
846   */
847  async getItem(id, model, tenant = null) {
848    return new Promise((resolve, reject) => {
849      model.byTenant(tenant).findById(id, (err, item) => {
850        itemNotFound(err, item, reject, 'NOT_FOUND')
851        resolve(item)
852      });
853    });
854  },
855
856  /**
857   * Gets item from database by id
858   * @param {string} id - item id
859   * @param model
860   * @param tenant
861   */
862  async getItem(id, model, tenant = null) {
863    return new Promise((resolve, reject) => {
864      model.byTenant(tenant).findById(id, (err, item) => {
865        itemNotFound(err, item, reject, 'NOT_FOUND')
866        resolve(item)
867      });
868    });
869  },
870
871  /**
872   * Gets item from database by id
873   * @param {string} id - item id
874   * @param model
875   * @param tenant
876   */
877  async getItem(id, model, tenant = null) {
878    return new Promise((resolve, reject) => {
879      model.byTenant(tenant).findById(id, (err, item) => {
880        itemNotFound(err, item, reject, 'NOT_FOUND')
881        resolve(item)
882      });
883    });
884  },
885
886  /**
887   * Gets item from database by id
888   * @param {string} id - item id
889   * @param model
890   * @param tenant
891   */
892  async getItem(id, model, tenant = null) {
893    return new Promise((resolve, reject) => {
894      model.byTenant(tenant).findById(id, (err, item) => {
895        itemNotFound(err, item, reject, 'NOT_FOUND')
896        resolve(item)
897      });
898    });
899  },
900
901  /**
902   * Gets item from database by id
903   * @param {string} id - item id
904   * @param model
905   * @param tenant
906   */
907  async getItem(id, model, tenant = null) {
908    return new Promise((resolve, reject) => {
909      model.byTenant(tenant).findById(id, (err, item) => {
910        itemNotFound(err, item, reject, 'NOT_FOUND')
911        resolve(item)
912      });
913    });
914  },
915
916  /**
917   * Gets item from database by id
918   * @param {string} id - item id
919   * @param model
920   * @param tenant
921   */
922  async getItem(id, model, tenant = null) {
923    return new Promise((resolve, reject) => {
924      model.byTenant(tenant).findById(id, (err, item) => {
925        itemNotFound(err, item, reject, 'NOT_FOUND')
926        resolve(item)
927      });
928    });
929  },
930
931  /**
932   * Gets item from database by id
933   * @param {string} id - item id
934   * @param model
935   * @param tenant
936   */
937  async getItem(id, model, tenant = null) {
938    return new Promise((resolve, reject) => {
939      model.byTenant(tenant).findById(id, (err, item) => {
940        itemNotFound(err, item, reject, 'NOT_FOUND')
941        resolve(item)
942      });
943    });
944  },
945
946  /**
947   * Gets item from database by id
948   * @param {string} id - item id
949   * @param model
950   * @param tenant
951   */
952  async getItem(id, model, tenant = null) {
953    return new Promise((resolve, reject) => {
954      model.byTenant(tenant).findById(id, (err, item) => {
955        itemNotFound(err, item, reject, 'NOT_FOUND')
956        resolve(item)
957      });
958    });
959  },
960
961  /**
962   * Gets item from database by id
963   * @param {string} id - item id
964   * @param model
965   * @param tenant
966   */
967  async getItem(id, model, tenant = null) {
968    return new Promise((resolve, reject) => {
969      model.byTenant(tenant).findById(id, (err, item) => {
970        itemNotFound(err, item, reject, 'NOT_FOUND')
971        resolve(item)
972      });
973    });
974  },
975
976  /**
977   * Gets item from database by id
978   * @param {string} id - item id
979   * @param model
980   * @param tenant
981   */
982  async getItem(id, model, tenant = null) {
983    return new Promise((resolve, reject) => {
984      model.byTenant(tenant).findById(id, (err, item) => {
985        itemNotFound(err, item, reject, 'NOT_FOUND')
986        resolve(item)
987      });
988    });
989  },
990
991  /**
992   * Gets item from database by id
993   * @param {string} id - item id
994   * @param model
995   * @param tenant
996   */
997  async getItem(id, model, tenant = null) {
998    return new Promise((resolve, reject) => {
999      model.byTenant(tenant).findById(id, (err, item) => {
1000        itemNotFound(err, item, reject, 'NOT_FOUND')
1001        resolve(item)
1002      });
1003    });
1004  },
1005
1006  /**
1007   * Gets item from database by id
1008   * @param {string} id - item id
1009   * @param model
1010   * @param tenant
1011   */
1012  async getItem(id, model, tenant = null) {
1013    return new Promise((resolve, reject) => {
1014      model.byTenant(tenant).findById(id, (err, item) => {
1015        itemNotFound(err, item, reject, 'NOT_FOUND')
1016        resolve(item)
1017      });
1018    });
1019  },
1020
1021  /**
1022   * Gets item from database by id
1023   * @param {string} id - item id
1024   * @param model
1025   * @param tenant
1026   */
1027  async getItem(id, model, tenant = null) {
1028    return new Promise((resolve, reject) => {
1029      model.byTenant(tenant).findById(id, (err, item) => {
1030        itemNotFound(err, item, reject, 'NOT_FOUND')
1031        resolve(item)
1032      });
1033    });
1034  },
1035
1036  /**
1037   * Gets item from database by id
1038   * @param {string} id - item id
1039   * @param model
1040   * @param tenant
1041   */
1042  async getItem(id, model, tenant = null) {
1043    return new Promise((resolve, reject) => {
1044      model.byTenant(tenant).findById(id, (err, item) => {
1045        itemNotFound(err, item, reject, 'NOT_FOUND')
1046        resolve(item)
1047      });
1048    });
1049  },
1050
1051  /**
1052   * Gets item from database by id
1053   * @param {string} id - item id
1054   * @param model
1055   * @param tenant
1056   */
1057  async getItem(id, model, tenant = null) {
1058    return new Promise((resolve, reject) => {
1059      model.byTenant(tenant).findById(id, (err, item) => {
1060        itemNotFound(err, item, reject, 'NOT_FOUND')
1061        resolve(item)
1062      });
1063    });
1064  },
1065
1066  /**
1067   * Gets item from database by id
1068   * @param {string} id - item id
1069   * @param model
1070   * @param tenant
1071   */
1072  async getItem(id, model, tenant = null) {
1073    return new Promise((resolve, reject) => {
1074      model.byTenant(tenant).findById(id, (err, item) => {
1075        itemNotFound(err, item, reject, 'NOT_FOUND')
1076        resolve(item)
1077      });
1078    });
1079  },
1080
1081  /**
1082   * Gets item from database by id
1083   * @param {string} id - item id
1084   * @param model
1085   * @param tenant
1086   */
1087  async getItem(id, model, tenant = null) {
1088    return new Promise((resolve, reject) => {
1089      model.byTenant(tenant).findById(id, (err, item) => {
1090        itemNotFound(err, item, reject, 'NOT_FOUND')
1091        resolve(item)
1092      });
1093    });
1094  },
1095
1096  /**
1097   * Gets item from database by id
1098   * @param {string} id - item id
1099   * @param model
1100   * @param tenant
1101   */
1102  async getItem(id, model, tenant = null) {
1103    return new Promise((resolve, reject) => {
1104      model.byTenant(tenant).findById(id, (err, item) => {
1105        itemNotFound(err, item, reject, 'NOT_FOUND')
1106        resolve(item)
1107      });
1108    });
1109  },
1110
1111  /**
1112   * Gets item from database by id
1113   * @param {string} id - item id
1114   * @param model
1115   * @param tenant
11
```

## Componente categoría

```
7   load = (src: string = '') => {
8     let fields = [
9       `?fields=name`,
10      `&page=${this.page}`
11    ];
12    fields.push(`&limit=${this.limit}`)
13    const q = this.share.parseLoad(src, this.source, fields);
14    this.rest.get(q.join(''))
15      .subscribe(res => {
16        this.viewMore = this.share.nextPage(res);
17        this.data = this.share.parseData(res, this.source);
18      }, error => {
19        (error.status === 401) ? this.cbMode = 'blocked' : null
20      })
21  }
22
23  goTo = () => this.share.goTo(this.source)
24
25  onSrc = (e) => this.load(e)
26
27
```

Figura 51. Componente categoría

## Componente productos

```
<ng-template #customTemplateSrc let-dat='dat'>
  <ul [@listAnimation] *ngIf="dat?.length" class="itemss">
    <li *ngFor="let inside of dat" (click)="emitCbClick(inside)" class="col-12 row ml-0 mr-0">
      <div class="col-md-10 col-lg-10 col-sm-10 pl-0 pr-0">
        <div class="d-flex ">
          <ngx-avatar class="d-none d-md-block" [name]="inside?.name"
            *ngIf="!(inside?.gallery | firstImage:'small')">
          </ngx-avatar>
          <div class="w-100">
            <h6 class="name mr-2 ml-1">{{inside?.name}}</h6>
            <div class="ml-1 mt-1 badge-zone" *ngIf="inside?.tag?.length">
              <span class="badge-tag badge mr-1" *ngFor="let tag of inside?.tag">
                {{tag?.name}}
              </span>
            </div>
            <div class="ml-1 mt-1" *ngIf="inside?.prices?.length">
              <span class="badge-tag-price badge mr-2" *ngFor="let price of inside?.prices">
                {{currencySymbol}} {{price?.amount | number:'1.2-2'}} {{currency}}
              </span>
            </div>
          </div>
        </div>
      </li>
    </ul>
  </div>
```

Figura 52. Componente productos

## Componentes ventas

```
0
1 onSubmit(): void {
2   const method = (this.id) ? 'patch' : 'post';
3   this.rest[method](`purchase${(method === 'patch') ? `/${this.id}` : ''}`, this.form.value)
4     .subscribe(res => {
5       this.router.navigate(['/', 'purchase', res._id])
6     })
7 }
8
9 cbList = () => {
10  this.router.navigate(['/', 'purchase'])
11 }
12
13 cbTrash = () => {
14  this.rest.delete(`purchase/${this.id}`)
15    .subscribe(res => this.cbList())
16 }
17
18 open(data: any = null) {
19   const initialState = {
```

Figura 53. Componentes ventas

## Componente proveedores

```
0   ngOnInit(): void {
1     this.form = this.formBuilder.group({
2       name: ['', Validators.required],
3       manager: ['', Validators.required],
4       address: ['', Validators.required],
5       email: ['', [Validators.required, Validators.email]],
6       phone: ['', Validators.required],
7       trace: ['', Validators.required],
8       description: [''],
9       tag: ['']
10    });
11
12    this.route.params.subscribe(params => {
13      this.id = (params.id === 'add') ? '' : params.id;
14    });
15
16    this.shared.registerUser.subscribe(res => {
17      this.users = [...[res],
18        ...this.users];
19      this.form.patchValue({manager: res})
20    })
21 }
```

Figura 54. Componente proveedores

## Componentes venta

```
3  onRemove(event) {
4      console.log({ action: 'trash', value: event._id });
5      this.cbSwipe({ action: 'trash', value: event._id });
6  }
7
8  parseTotal = () => {
9      try {
10         let total = 0;
11         this.items.forEach((i) => {
12             const prices = i.prices.find((a) => a.amount);
13             total += parseFloat(String(i.qty * prices.amount));
14         });
15         this.total = total;
16         this.subtotal = ((total*12)/(1+12)).toFixed(2)
17         this.ivas = total - parseFloat(this.subtotal)
18     } catch (e) {
19         this.total = 0;
20     }
21 };
```

Figura 55.Componentes venta

## Componente factura

```
public printPDF(){
    var element = document.getElementById('factura');
    html2pdf().from(element).set({
        margin: 0,
        fontSize : 9,
        filename: 'ComprobanteVenta.pdf',
        html2canvas: { scale: 2,useCORS: true },
        jsPDF: { orientation: 'portrait', unit: 'in', format: 'letter', compr
    }).save();
}
public convertToPDF() {
    var quotes = document.getElementById('factura');
    html2canvas(quotes,{allowTaint : true,useCORS: true}).then(function (canvas) {
        var imgData = canvas.toDataURL('image/png');
        var margin = 6;
        var imgWidth = 210;
        var pageHeight = 295;
        var imgHeight = canvas.height * imgWidth / canvas.width;
```

Figura 56.Componentes venta

### 4.2.3.6 Fase de transición

En esta fase se realiza la validación del software con la Unidad Educativa Libertad en donde se justifica el desarrollo del aplicativo web.

Panel de control, se filtran todos los movimientos, ventas y productos.

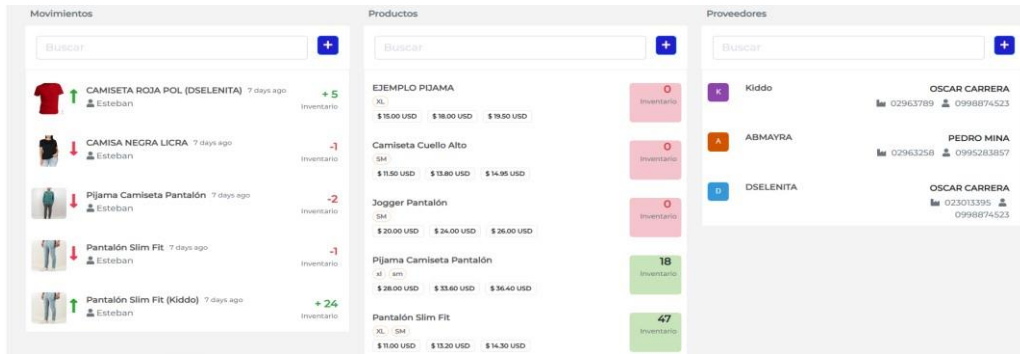


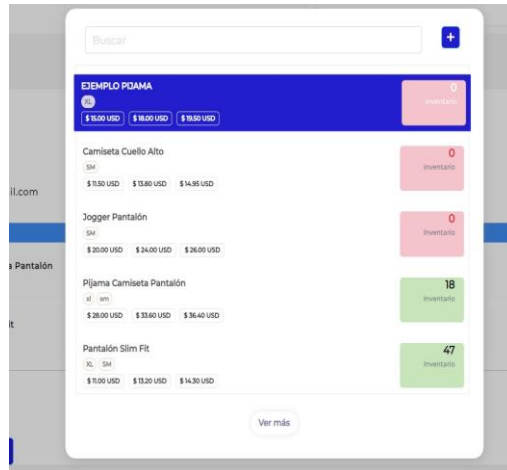
Figura 57. Fase de transición

Lista de productos con su stock y su respectivo semáforo de productos señalando el estado de dichos productos.



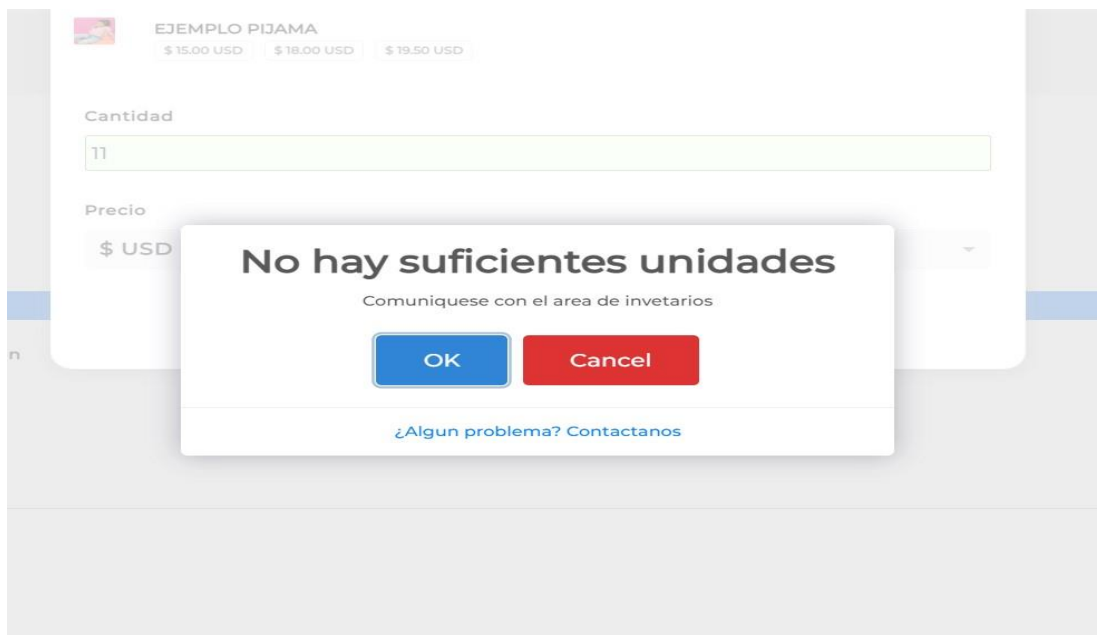
Figura 58. Lista de productos con su stock y su respectivo semáforo

Antes de ejecutar una venta me muestra las existencias del producto a vender para no vender si no hay existencia de producto.



**Figura 59.**Existencias del producto a vender

Se muestra una alerta en el caso que intente ejecutar una venta con el stock es 0 desactivando el botón guardar.



**Figura 60.**Alerta en el caso que intente ejecutar

Los precios se ingresan según la base imponible de impuesto el % de ganancia mínima en cada producto

Precios

USD 0,00

Agregar

\$ 10.55 USD

\$ 12.66 USD

\$ 13.71 USD

Código

**Figura 61.** Base imponible de impuesto el %

Se muestra como es la configuración para manejar los impuestos y el porcentaje de ganancia mínima.

Configuración rápida

Nuestro asistente te guiará para la configuración rápida de tu panel. Solo tomará 1 minuto

Rivate

Nombre de empresa  
RIVATEX

Porcentaje del impuesto  
12

Porcentaje normal de ganancia  
30

Porcentaje de ganancia en productos de descuento  
20

Actualizar

**Figura 62.** Configuración para manejar los impuestos y el porcentaje



## DISCUSIÓN

En el presente proyecto ha tomado en cuenta los objetivos de la investigación, que se orientan en establecer una propuesta de aplicación informática para el proceso de inventarios y ventas en la empresa RIVATEX, el cual se ha tomado en cuenta la problemática de la empresa al realizar el proceso de inventario y venta, también se realizó un estudio para la fundamentación bibliográfica del marco teórico y metodológico para el

desarrollo de la aplicación informática que facilite el manejo de inventarios y ventas.

El uso de la metodología de desarrollo de software (RAD), permitió cumplir con los objetivos de investigación, establecer una propuesta de aplicación informática para el proceso de inventarios y ventas para la empresa RIVATEX, como primer punto fue recolectar los principales requerimientos de la empresa por parte del usuario, realización de un prototipo de las principales vistas que serán parte de la aplicación, las cuales contribuyeron para realizar la codificación de las funcionalidades principales de la aplicación, se realizó pruebas unitarias dentro de la aplicación y pruebas de aceptación con el Administrador, al finalizar con este proceso se obtuvo como resultado el cumplimiento del tercer objetivo específico planteado en esta investigación, el cual permite realizar un inventario y ventas por cada área de conocimiento permitiendo al usuario mantener la información de forma organizada y actualizada de cada uno de los productos de la empresa.

Según (Aizaga,2019) en su proyecto de investigación con el tema "Control De Inventario Para Aumentar La Rentabilidad En La Empresa Lepulunchexpress S. A." considerado un registro de los bienes o materiales que forman parte del activo y del cual podemos obtener la mayor rentabilidad en la empresa.

El inventario es el conjunto de mercancías o artículos que tienen las empresas para comerciar, permitiendo la compra y venta o la fabricación antes de su venta, en un periodo económico determinado; los inventarios forman parte del grupo de activos circulantes de toda organización. Adicionalmente, es uno de los activos más grandes existentes en una empresa, y aparece reflejado tanto en el balance general como en el estado de resultados.

La actual investigación tiene similitud con la investigación citada anteriormente, al momento de realizar las pruebas de aceptación, el cliente se mostró satisfecho con las funcionalidades implementadas en la aplicación informática, ya que esto le permite obtener una información organizada, mencionado que el tiempo de registro, búsqueda de información, generación de reportes y solicitudes se disminuía considerablemente y será beneficioso para la empresa con la demanda de otras empresas similares.

## **V. CONCLUSIONES Y RECOMENDACIONES**

### **5.1. CONCLUSIONES**

- Se documentó bibliográficamente en base a los sistemas informáticos y páginas web permitiendo realizar los antecedentes investigativos, marco teórico y la metodología. Se investigó en libros, artículos científicos, repositorios de tesis y revistas en la cual admitió recopilar la información necesaria para sustentar el proyecto de investigación.
- Con la aplicación de una metodología de investigación se realizó el levantamiento de información que permitió conocer la situación actual del manejo de inventarios y venta la aplicación de un marco metodológico permitió el levantamiento de procesos que se realizaban en la actualidad dentro de la empresa.
- Mediante la aplicación de los instrumentos investigativos se logró recolectar información para determinar los requerimientos y permitir la elaboración del sistema de ventas e inventarios en el cual mejorara la eficiencia en la información, fácil manejo y satisfaciendo las necesidades encontradas en la empresa RIVATEX.
- El prototipo de aplicación informática para la gestión de los procesos administrativos estudiados es fácilmente adaptable a un sistema ya que están relacionados a diversas áreas de la organización y los datos que se generan, pueden estar agrupados y optimizando la facilidad de búsqueda y tratamiento de la información de la empresa.

## **5.2. RECOMENDACIONES**

- Capacitar al personal encargado sobre el manejo respectivo del proceso de ventas e inventarios de los y las demás funcionalidades del sistema informático para poder obtener un mayor rendimiento de los procesos mencionados garantizando la integridad y confidencialidad de los datos.
- Las herramientas de RAD varían, por lo cual se recomienda indagar a profundidad sobre su aplicación en los procesos de desarrollo y escoger los parámetros que más se ajusten a los requerimientos y procesos organizacionales de la empresa teniendo en cuenta la magnitud del proyecto.
- La caracterización de procesos es fundamental en el desarrollo de un sistema informático por esto es importante conocer y hacer uso de varias herramientas y técnicas de investigación que permita caracterizar los procesos de una manera objetiva y ordenada, puesto esta información será la base para continuar con el desarrollo de la herramienta tecnológica.
- Para el buen funcionamiento del sistema es necesario realizar un procedimiento de mantenimiento preventivo de las herramientas tecnológicas para garantizar el buen estado y la duración del siete

## VI. REFERENCIAS BIBLIOGRÁFICAS

- Aarush,G.(2016). *Revista Internacional de Investigación Innovadora en Ciencias y Tecnología de la Computación*.India:Vellore
- Agurto,P.(2017). "El Control De Inventarios Y Su Incidencia En Los Estados Financieros En La Empresa Hdineagros S.A. Obtenido de :  
<http://repositorio.utmachala.edu.ec/bitstream/48000/4623/1/TTUACE-2016-CA-CD00001.pdf>
- Aranibar, N. (2016). *MySQL WorkBench*. Obtenido de  
<http://www.monografias.com/trabajos88/mysql-worckbench/mysqlworckbench.shtml>
- Arias, Á. (2015). *Bases de Datos con MySQL*. Segunda Edición
- Arias, M. (2015). *Aprende Programación Web con PHP y MySQL*. Segunda Edición.
- Ariel, O. (2016). *Association for Computing Machinery*. New York: Memphis Tennessee USA
- Barco, J., & Jiménez, E. (2016). *SISTEMA DE GESTIÓN ACADÉMICA PARA LA INSTITUCIÓN EDUCATIVA*. (Tesis de postgrado). UNIVERSIDAD AUTÓNOMA DE MANIZALES, Manizales, Colombia
- Castillero, O. (2017). *psicologiyamente.com*. Obtenido de  
[psicologiyamente.com:](https://psicologiyamente.com/psicologiyamente.com)  
<https://psicologiyamente.com/miscelanea/tiposde-investigacion>
- Castillo, A. (2016). *Plataforma de inventarios*. Obtenido de  
<https://www.monografias.com/trabajos89/proceso-administracionventas/proceso-administracion-ventas.shtml>

Corzo, P. (2021). *Implementación De Un Sistema Para La Gestión De Ventas E Inventario De Una Empresa De Calzado*. Recopilado de: <https://repositorio.usil.edu.pe/server/api/core/bitstreams/85353a5a-5a65-4b03-a6ba-ab7ca514fab9/content>

Groussard, T. (2011). *Java7 Los fundamentos del lenguaje Java*. Barcelona: ENI.

Lopez, A. (2016). Obtenido de <https://www.heflo.com/es/blog/workflow/automatizacion-oficina/>

Manda, U. (2018). *Ionic Framework con Angular para el desarrollo de aplicaciones híbridas*. Vellore: Revista Internacional de Nuevas Tecnologías e Investigaciones

Mora, N., & Vega, C. (2013). *SISTEMA INFORMÁTICO DE CONTROL DE VENTAS Y PROCESO DE MATRICULACIÓN DE LA UNIDADE EDUCATIVA WENCESLAO RIJAVEC DE LA CIUDAD DECALCETA - CANTÓN BOLÍVAR. (Tesis de grado)*. ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DEMANABÍ MANUEL FÉLIX LÓPEZ, Calceta, Ecuador

Pinargote, M. (2014). *SISTEMA INFORMÁTICO DE GESTIÓN DE VENTAS Y SERVICIOS TÉCNICOS EN LA EMPRESA S-COMPU DEL CANTÓN PEDERNALES*.

Posso, M. (2017). "Modelo de gestión de inventarios para el mejoramiento de la calidad de las empresas. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/7848/1/PG%20598%20TESIS.pdf>


Rouse, M. (2015). *Gestión de proyectos*. Obtenido de <https://es.workmeter.com/blog/gestion-de-proyectos-concepto-beneficios-yfases>

significados.com. (10 de 12 de 2019). Obtenido de <https://www.significados.com/tipos-de-investigacion/>

Sun, H. (2018). *Efficient dynamic analysis for Node.js*. Vienna Austria: Evolución y Reingeniería de Software.

## VII. ANEXOS

### Anexo 1. Informe de Turnitin

 **Informe de Originalidad Turnitin**  
Informe titulación por Esteban Martinez  
Desde proyectos (Titulación)

Índice de similitud	Similitud según fuente
9%	Internet Sources: 11% Publicaciones: 0% Trabajos del estudiante: 3%

Procesado el 06-ene.-2023 09:31 -05  
Identificador: 1989208643  
Número de palabras: 10100

#### fuentes:

- 1 1% match (Internet desde 16-dic.-2022)  
<http://repositorio.upec.edu.ec/bitstream/123456789/1520/1/066-%20CHARRO%20GUACHAN%20JUAN%20CARLOS.pdf>
- 2 1% match (Internet desde 16-dic.-2022)  
<http://repositorio.upec.edu.ec/bitstream/123456789/1511/1/060-%20TIPAN%20VIVEROS%20DARWIN%20FERNANDO.pdf>
- 3 1% match (Internet desde 16-dic.-2022)  
<http://repositorio.upec.edu.ec/bitstream/123456789/1510/1/061-%20CHACHA%20JAIME%20-%20VARGAS%20DIANA.pdf>
- 4 1% match (Internet desde 08-sept.-2022)  
<https://dspace.ups.edu.ec/bitstream/123456789/15717/1/UPS-CT007711.pdf>
- 5 1% match (Internet desde 23-sept.-2022)  
<https://dspace.ups.edu.ec/bitstream/123456789/18172/1/UPS-CT008640.pdf>
- 6 1% match (Internet desde 30-jun.-2018)  
<http://repositorio.uta.edu.ec/bitstream/123456789/6079/1/99%20o.e..pdf>
- 7 1% match (Internet desde 20-dic.-2022)  
<http://repositorio.ug.edu.ec/bitstream/redug/59006/1/SALAZAR%20CASTRO%20ADRIANA%20%281%29.pdf>
- 8 1% match ()  
[Gutierrez Vega, Jacqueline Merry, Jolkeda Loyola, Karina Janet, Lau Chauca De Valencia, Teresa Betty, "Gestión del talento humano por competencias y la calidad de atención de reclamos en la empresa enel distribución Perú S.A.", Universidad Inca Garcilaso de la Vega, 2019](#)

#### texto del trabajo:

4Introducción Actualmente las empresas buscan resaltar y mantenerse en un mercado cada vez más exigente y competitivo teniendo que adaptarse a las

## Anexo 2. Certificado de Abstrac



### UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN AND NATIVE LANGUAGE CENTER

Informe sobre el Abstract de Artículo Científico o Investigación.

**Autor:** Martínez Martínez Esteban Wladimir

**Fecha de recepción del abstract:** 10 de febrero de 2023

**Fecha de entrega del informe:** 10 de febrero de 2023

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

#### **Observaciones:**

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según los rubrics de evaluación de la traducción en Inglés, ésta alcanza un valor de 9, por lo cual se valida dicho trabajo.

Atentamente



Ing. Edison Peñañiel Arcos MSc  
Coordinador del CIDEN



Anexo 3. Certificado del Abstrac



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI  
FOREIGN AND NATIVE LANGUAGE CENTER

ABSTRACT- EVALUATION SHEET				
NAME: Martínez Martínez Esteban Wladimir				
DATE: 10 de febrero de 2023				
TOPIC: "Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX"				
MARKS AWARDED		QUANTITATIVE AND QUALITATIVE		
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1 Vera Játiva Edwin Andrés,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED	TOTAL 9		

## Anexo 4. Acta de predefensa



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

### ACTA

DE LA SUSTENTACIÓN ORAL DE LA PREDENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

ESTUDIANTE:	MARTINEZ MARTINEZ ESTEBAN WLADIMIR	CÉDULA DE IDENTIDAD:	1004095632
PERIODO ACADÉMICO:	2022B		
PRESIDENTE TRIBUNAL	JEFFRY ALEX NARANJO CEDEÑO	DOCENTE TUTOR:	MSC. JORGE HUMBERTO MIRANDA REALPE
DOCENTE:	MILTON GABRIEL DEL HIERRO MOSQUERA		
TEMA DEL TIC:	"Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX"		

No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	7,67	Mejorar la redacción del planteamiento del problema, definir macro, meso, micro
2	FUNDAMENTACIÓN TEÓRICA	7,67	
3	METODOLOGÍA	7,67	Mejorar la operacionalización de variables, definir correctamente los indicadores de las variables
4	RESULTADOS	7,67	Incluir el tema de ventas con los diferentes precios que maneja la empresa, definir la configuración inicial en el tema de porcentajes de precios.
5	DISCUSIÓN	7,67	Realizar los filtros o programación necesaria para que no seleccione el precio base del producto, realizar los ajustes necesarios para aplicar los porcentajes en los precios de ganancia y descuentos.
6	CONCLUSIONES Y RECOMENDACIONES	7,67	Añadir otra conclusión y recomendación
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	7,67	Mejorar el tiempo de presentación
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	7,67	Corregir el documento y apegarse al formato

Obteniendo una nota de: 7,37 Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el lunes, 30 de enero de 2023

  
MSC. JEFFRY ALEX NARANJO CEDEÑO  
PRESIDENTE TRIBUNAL

  
MSC. JORGE HUMBERTO MIRANDA REALPE  
DOCENTE TUTOR

  
MSC. MILTON GABRIEL DEL HIERRO MOSQUERA  
DOCENTE

## Anexo 5. Rubrica de la predefensa



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI  
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES



### RÚBRICA DE EVALUACIÓN DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

ESTUDIANTE:	MARTÍNEZ MARTÍNEZ ESTEBAN WLADIMIR	CÉDULA DE IDENTIDAD:	1004695632
PERIODO ACADÉMICO:	2022B	FECHA:	30 de enero de 2023
PRESIDENTE TRIBUNAL:	MSC. JEFFRY ALEX NARANJO CEDEÑO	HORA:	10:30
DOCENTE:	MSC. MILTON GABRIEL DEL HERRO MOSQUERA	DOCENTE TUTOR:	MSC. JORGE HUMBERTO MIRANDA REALPE
TEMA DEL TIC:	"Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX"		

No.	CATEGORÍA	CRITERIO ÓPTIMO DE EVALUACIÓN	CRITERIO ÓPTIMO DE EVALUACIÓN			
			PRESIDENTE	TUTOR	DOCENTE	
SUSTENTACIÓN ORAL - DEFENSA	1	PROBLEMA - OBJETIVOS	Se expone el planteamiento, formulación y justificación, los objetivos son expuestos como sistémicos para alcanzar el objetivo general; las preguntas de investigación aportan a entender lo que se quiere investigar y son coherentes con los objetivos.	7	8	8
	2	FUNDAMENTACIÓN TEÓRICA	Es un marco de referencia para el desarrollo e interpretación de los resultados de la investigación. Los antecedentes investigativos incluidos tienen relación con el tema planteado.	7	8	8
	3	METODOLOGÍA	El estudiante explicó el enfoque de la investigación de manera lógica al análisis estadístico, la población, muestra, técnicas e instrumentos presentados, permitiendo entender que el informe es consistente en resultados y discusión.	7	8	8
	4	RESULTADOS	Se analizó la relación entre las variables de manera cualitativa, cuantitativa y fueron representativas a la profesión. Espuso gráficas, figuras, tablas de frecuencia y contingencia coherentes y de acuerdo a la metodología de investigación. Los datos fueron presentados de forma clara y efectiva a lo observado y no exigen interpretaciones.	7	8	8
	5	DISCUSIÓN	La discusión expuesta y defendida establece la relación de los objetivos propuestos, con los antecedentes de la investigación y el tema.	7	8	8
	6	CONCLUSIONES Y RECOMENDACIONES	Las conclusiones y recomendaciones expuestas, son claras, concisas y acordes a los objetivos y resultados de la investigación.	7	8	8
	7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	El estudiante demostró conocimiento y seguridad del objeto de estudio, Relacionó conceptos y teorías. El vocabulario utilizado fue acorde a la terminología de la profesión con un volumen de voz adecuado. Hizo un uso correcto del tiempo. Utilizó recursos didácticos apropiados.	7	8	8
<b>PROMEDIO SOBRE SIETE</b>			5,37			
DOCUMENTO ESCRITO	8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	El formato, la organización de contenidos, redacción, uso de gramática y ortografía, aplicación de normas de citas y referencias cumplen con el formato de la UFEC.	6	7	7
	<b>PROMEDIO SOBRE TRES</b>			2,00		
			<b>7,37</b>			

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del tribunal de sustentación de la pre-defensa.

Art. 35.- De la aprobación de la pre-defensa del informe final del TIC.- De la aprobación de la pre-defensa del informe final del TIC.- El estudiante deberá obtener una nota mínima de 7/10; el estudiante que no obtenga esta nota mínima, se presentará a un segundo proceso de sustentación, transcurrido el término de 10 días desde la fecha primera.

MSC. JORGE HUMBERTO MIRANDA REALPE  
DOCENTE TUTOR

MSC. JEFFRY ALEX NARANJO CEDEÑO  
PRESIDENTE

MSC. MILTON GABRIEL DEL HERRO MOSQUERA  
DOCENTE

## Anexo 6. Certificado de cumplimiento



Señora Selenita Paspuel propietaria y gerente general de la empresa RIVATEX ubicada en la ciudad de Quito

### CERTIFICO

Que el joven: **ESTEBAN WLADIMIR MARTINEZ MARTINEZ**, portador de la C.C 1004095632, egresado de la Universidad Politécnica Estatal del Carchi (UPEC), en la carrera de Ingeniería en Ciencias de la Computación, ha dado cumplimiento a su Proyecto de investigación denominada "**Desarrollo de un sistema informático para los procesos de venta e inventario de la empresa RIVATEX**" en la empresa, mismo que lo ha culminado satisfactoriamente con todos los requerimientos que la empresa RIVATEX a solicitado, por lo que se extiende al mencionado joven nuestro más sincero agradecimiento por sus aportes brindados que serán de mucha ayuda dentro de la empresa, augurándole toda clase de éxitos en su vida profesional.

Lo que certifico en honor a la verdad, facultando al interesado hacer uso del presente en lo que se estimare conveniente.

Atentamente

  
Selenita Paspuel P.  
**GERENTE GENERAL**

Dirección: ALEJANDRO GALARZA 069-160 Y MANUEL AGUILAR.  
Telefax: (02) 2826-019 Celular: 099 963 2447  
Email: rivatex2006@hotmail.com  
QUITO - ECUADOR

## Anexo 7. Manual de usuario

### Manuel de Usuario

#### Introducción

La aplicación informática para los procesos de venta e inventario de la empresa RIVATEX es una página web, la cual está formada por facturación, manejo de stock y Kardex, el software desarrollado se ha enfocado en los requerimientos de

## Bienvenido

Lleva tu inventario de forma inteligente y ágil.

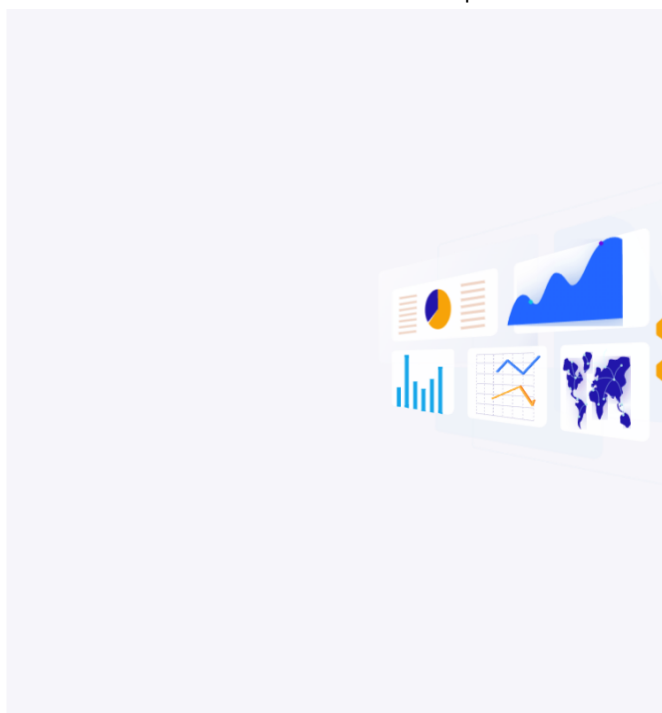
#### Email

Ingresar el email que tienes registrado

#### Contraseña

Ingresar tu contraseña, mantén en un lugar seguro tus accesos

[Olvide contraseña](#)



la empresa en donde se muestra las características principales del sistema.

#### Ingreso al sistema

Para acceder a la aplicación para los procesos de venta e inventario de la empresa RIVATEX, el usuario deberá acceder, donde se mostrará la siguiente ventana.

Aquí podemos observar la ventana donde se puede recuperar la contraseña, debemos de ingresar el correo electrónico activo.

# Bienvenido

Lleva tu inventario de forma inteligente y ágil.

Email

Ingresa el email que tienes registrado

[Login](#)

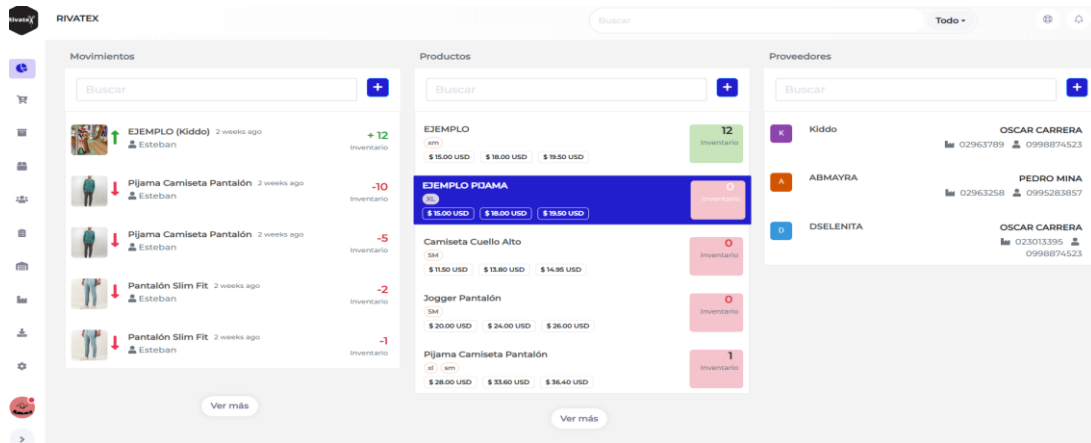
[Recuperar contraseña](#)

## Tenemos el menú de usuario

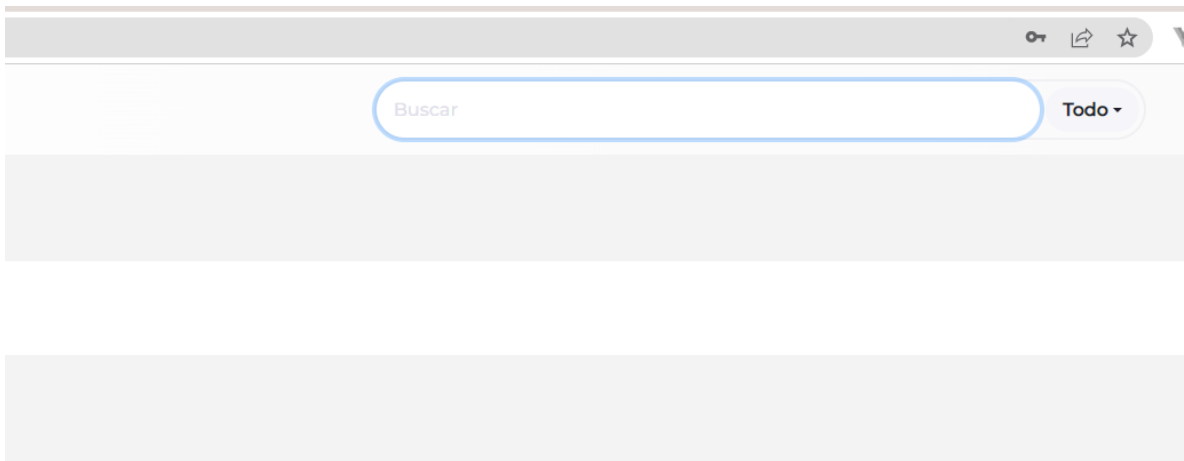
El menú principal redirige a las pantallas de las interfaces principales como: productos categorías ventas, inventarios proveedores, usuarios y configuración general.



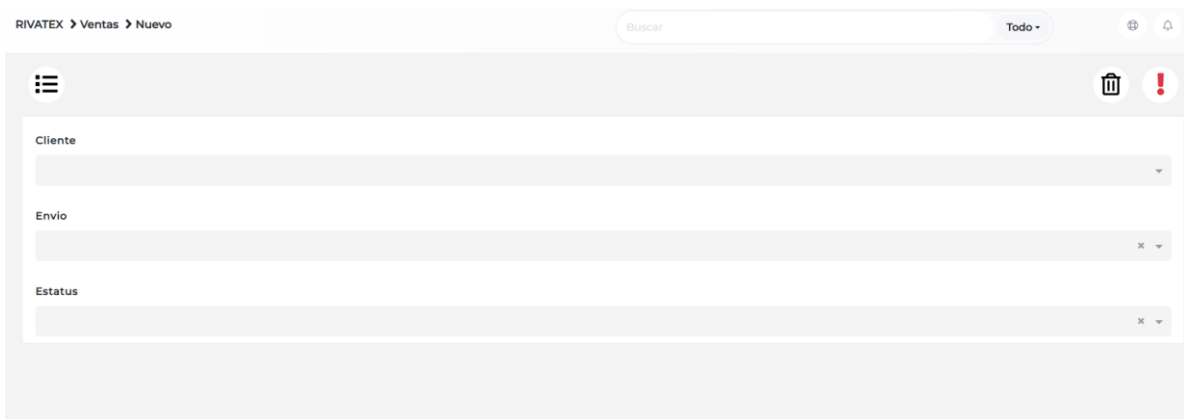
En esta ventana tenemos el panel principal donde se encuentra resumido los movimientos, productos y proveedores



Este componente es útil para buscar cualquier registro en toda la base de datos



Este es formulario para ingresar podemos ingresar una nueva venta, los campos cliente, envió y estado.



En esta ventana podemos agregar productos a la venta donde nos muestra el subtotal, IVA y total.

**Rivatex**  
 Duchicela N4-14 y Geovanny Calles, Quito, Ecuador  
 022826019  
 rivatex2006.online.com

**Pedido**  
 RIVAS-1674220215FR  
 Fecha: Jan 20, 2023

Cliente **OSCAR CARRERA**  
 Correo: oscar.carrera@gmail.com  
 Teléfono: 0998874523

Artículo	Cantidad	Precio
Pijama Camiseta Pantalón SKU: 1006	2	2x \$ 33.60 USD <b>\$ 67.20 USD</b>
Pantalón Slim Fit SKU: 1004	1	1x \$ 13.20 USD <b>\$ 13.20 USD</b>

Aplicar descuento  
**SUBTOTAL : \$67.00**  
**IVA (0.12) : 13.40**  
**TOTAL : \$ 80.40**

En este componente sirve para elegir y filtrar los productos, cada producto contiene el número exacto de productos existentes

Buscar +

**EJEMPLO** 12  
Inventario

xm

\$ 15.00 USD \$ 18.00 USD \$ 19.50 USD

---

**EJEMPLO PIJAMA** 0  
Inventario

XL

\$ 15.00 USD \$ 18.00 USD \$ 19.50 USD

---

**Camiseta Cuello Alto** 0  
Inventario

SM

\$ 11.50 USD \$ 13.80 USD \$ 14.95 USD

---

**Jogger Pantalón** 0  
Inventario

SM

\$ 20.00 USD \$ 24.00 USD \$ 26.00 USD

---

**Pijama Camiseta Pantalón** 1  
Inventario

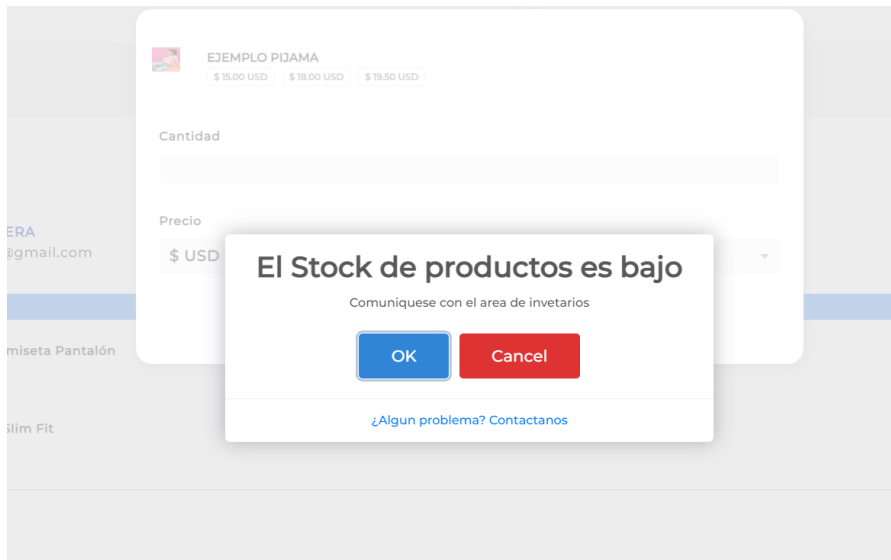
xl sm

\$ 28.00 USD \$ 33.60 USD \$ 36.40 USD

[Ver más](#)



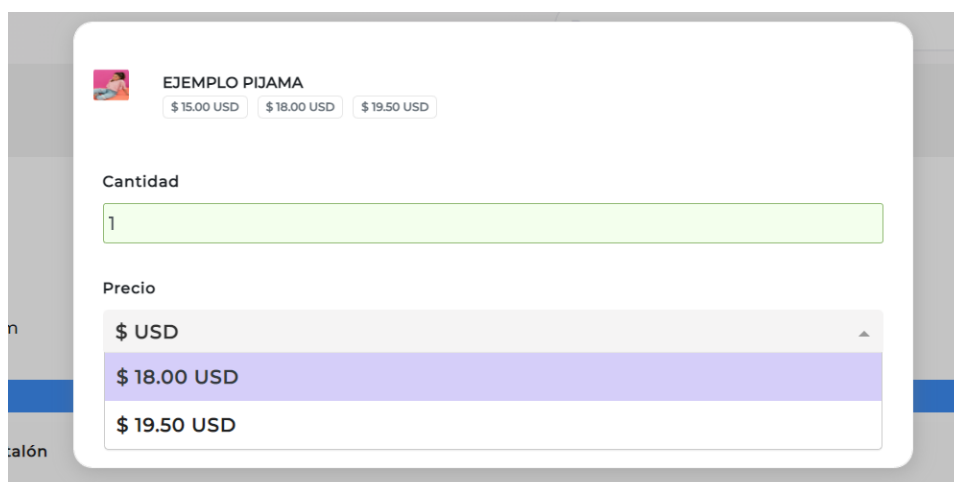
En esta ventana emergente es una alerta que se activa cuando el stock de productos es medio o bajo.



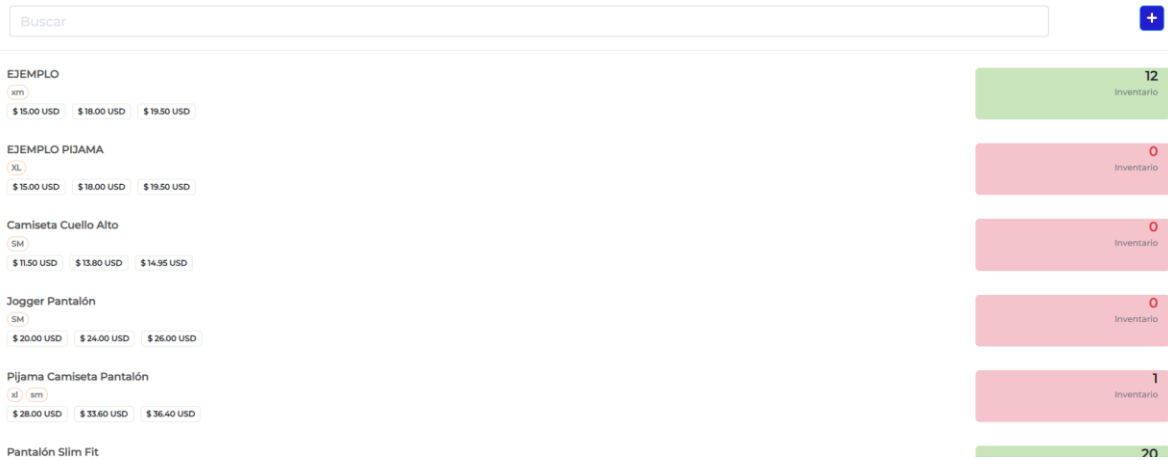
Esta ventana emergente nos indica cuando ya se terminó la cantidad de productos, no nos permite registrar venta.



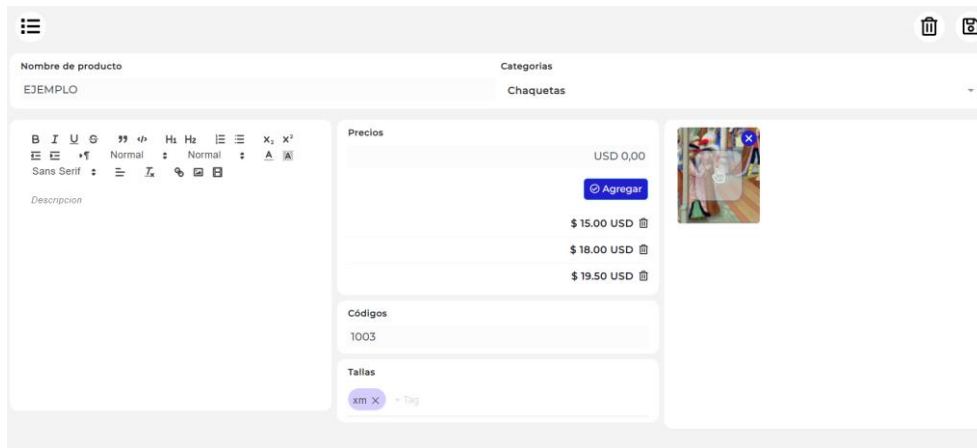
En esta ventana nos muestra a detalle el producto y los diferentes precios que contiene cada artículo y la cantidad.



En esta ventana nos indica la lista de productos con el stock exacto.



En este formulario nos permite registrar un nuevo producto donde podemos colocar la categoría los precios y una imagen.



En esta ventana nos indica la imagen que fue cargada en el servidor.



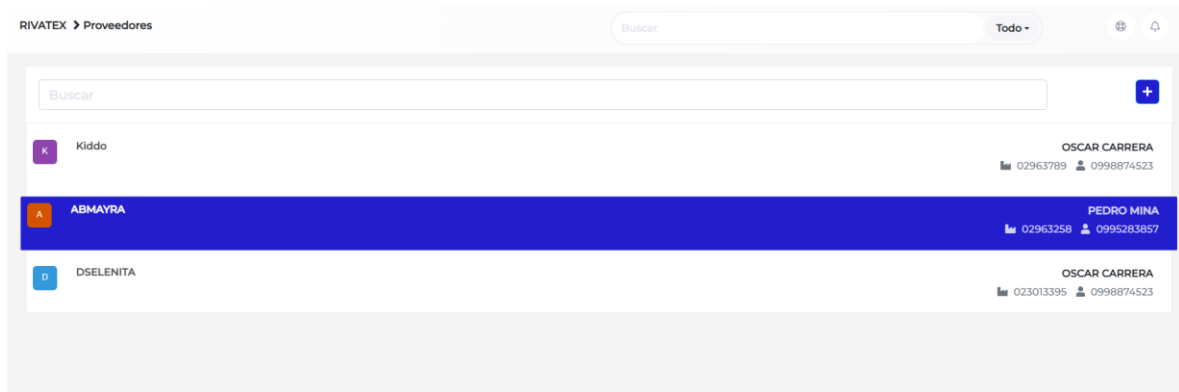
En este formulario nos permite dar de alta a un nuevo usuario, proveedor o usuario de sistema cada cual con un rol específico.

En esta vista nos indica un listado de todas las ubicaciones donde serán almacenados los productos

Icono	Nombre de Bodega	Contacto
BA	BODEGA ALTO	JUAN MENA 0995283857
BB	BODEGA BAJO	JUAN MENA 0995283857

En este formulario nos permite registrar el lugar donde serán almacenados todos los productos que ingresan a la empresa.

Este formulario nos muestra la lista de proveedores donde es posible editar y eliminar.

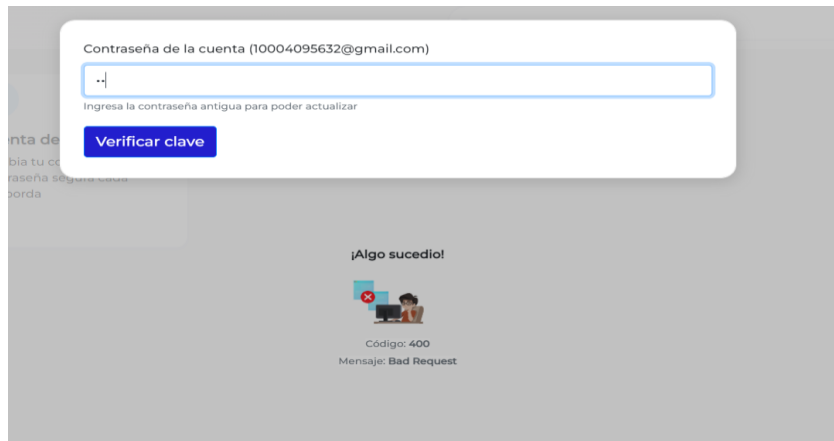


En este formulario nos permite registrar a un nuevo proveedor al sistema.

En esta pantalla nos muestra todas las compras y ventas dándonos la ganancia por mes de cada producto y el stock exacto de productos.

Fecha	Entradas	Salidas	Stock
2023-01-29		-2 \$ 13.20 USD	--
2023-01-29		-1 \$ 13.20 USD	--
2023-01-20		-1 \$ 13.20 USD	--
2023-01-20	24 \$ 11.00 USD		--
<b>RESULTADOS</b>	<b>\$ 264.00 USD</b>	<b>\$ 52.80 USD</b>	<b>20</b> Unidades
<b>TOTAL</b>	<b>\$ 211.20 USD</b>		

En esta ventana emergente podemos actualizar la contraseña periódicamente.



En esta ventana nos permite configurar el nombre, logo, impuesto tributario, porcentaje de ganancia y los estándares de semaforización de productos

#### Configuración rápida

Nuestro asistente te guiará para la configuración rápida de tu panel. Solo tomará 1 minuto



Nombre de empresa

RIVATEX

Porcentaje del impuesto

12

Porcentaje de ganancia sobre el costo

30

Porcentaje descuento

3

Stock semaforo en rojo

5

Stock semaforo en amarillo