



**UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA DE INGENIERÍA EN TELEINFORMÁTICA**

**TRABAJO DE TITULACIÓN
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TELEINFORMÁTICA**

**ÁREA
TECNOLOGÍA DE LAS TELECOMUNICACIONES**

**TEMA
“SISTEMA DE INVENTARIO PARA LA EMPRESA
SOLUDESIGN MULTIPLATAFORMA.”**

**AUTORA
TORRES NEIRA EMILY NICOLE**

**DIRECTORA DEL TRABAJO:
ING. TELECOM. TRUJILLO BORJA XIMENA FABIOLA, MG.**

GUAYAQUIL, SEPTIEMBRE 2022



ANEXO XI.- FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN

FACULTAD: INGENIERÍA INDUSTRIAL
CARRERA: INGENIERÍA EN TELEINFORMÁTICA



| | | | |
|--|---|-----------------|-----|
| REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA | | | |
| FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN | | | |
| TÍTULO: | “SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGN MULTIPLATAFORMA.” | | |
| AUTORA: | TORRES NEIRA EMILY NICOLE | | |
| TUTORA: REVISOR: | ING. TELECOM. TRUJILLO BORJA XIMENA FABIOLA, MG ING. TELECOM. VEINTIMILLA ANDRADE JAIRO GEOVANNY, MG | | |
| INSTITUCIÓN: | UNIVERSIDAD DE GUAYAQUIL | | |
| UNIDAD/FACULTAD: | FACULTAD DE INGENIERÍA INDUSTRIAL | | |
| MAESTRÍA/ESPECIALIDAD: | | | |
| GRADO OBTENIDO: | INGENIERIA EN TELEINFORMÁTICA | | |
| FECHA DE PUBLICACIÓN: | 27 DE SEPTIEMBRE DEL 2022 | No. DE PÁGINAS: | 124 |
| ÁREAS TEMÁTICAS: | TECNOLOGÍA DE LAS TELECOMUNICACIONES | | |
| PALABRAS CLAVES/ KEYWORDS: | Sistema de inventario, stock, raspberry, proceso, inventariado. Inventory system, stock, raspberry, process, inventory | | |
| RESUMEN: <p>En el presente trabajo se detalla el cumplimiento de cada uno de los requerimientos que fueron solicitados por la empresa SOLUDESING para la implementación de un sistema de inventario, el sistema consta con 8 módulos, el módulo de ingreso de pedido en el cual se permite realizar una solicitud de artículos a los diferentes proveedores ingresados, una vez que se ingresa un pedido y llega a la bodega de la empresa se realiza su ingreso y automáticamente el sistema presenta las existencias de los artículos en el módulo inventario, así también para realizar el egreso de un producto es necesario emitir una orden de egreso que mediante el código del producto busca dentro del stock y los descuenta una vez emitida dicha orden, permitiendo al usuario administrador contar con un apartado que presenta los reportes de cada uno de los ingresos y egresos de artículos con un fácil acceso según lo que fue solicitado, adicionalmente se implementó un módulo que consiste en el reconocimiento mediante el código del producto que permite establecer si un artículo pertenece al stock de la compañía o no ha sido registrado en el sistema ni en el inventario. Las recomendaciones y conclusiones que se indican son específicamente para la mejora de futuros trabajos o investigaciones con el fin de que el sistema sea asequible a empresas que tengan como actividad principal la compra, venta y distribución de artículos.</p> | | | |
| ABSTRACT: <p>This work details the fulfillment of each of the requirements that were requested by SOLUDESING company for the implementation of an inventory system, the system consists of 8 modules, the order entry module which allows to make a request for items to different suppliers entered, once an order is registered and arrives at the warehouse of the company it is entered and automatically the system displays the stock of items in the inventory module, Also, in order to dispatch a product, it is necessary to issue a dispatch order that searches within the stock by means of the product code and deducts it once the order is issued, allowing the administrator user to have a section that shows the reports of each of the incoming and outgoing items with an easy access</p> | | | |

according to what was requested, additionally a module was implemented that consists of the recognition by means of the product code that allows establishing if an item belongs to the company's stock or has not been registered in the system or in the inventory. The recommendations and conclusions that are indicated are specifically for the improvement of future works or investigations with the purpose of making the system accessible to companies whose main activity is the purchase, sale and distribution of articles.

| | | |
|------------------------------|--|--|
| ADJUNTO PDF: | SI (X) | NO |
| CONTACTO CON AUTOR/ES: | Teléfono: 0960201537 | E-mail: emily.torresn@ug.edu.ec |
| CONTACTO CON LA INSTITUCIÓN: | Nombre: Ing. Ind. Ramón Maquilón Nicola, Mg. | |
| | Teléfono: 042-658128 | |
| | E-mail: direccionti@ug.edu.ec | |



**ANEXO XII.- DECLARACIÓN DE AUTORÍA
Y DE AUTORIZACIÓN DE LICENCIA
GRATUITA INTRANSFERIBLE Y NO EXCLUSIVA
PARA EL USO NO COMERCIAL DE LA OBRA
CON FINES NO ACADÉMICOS**



**FACULTAD: INGENIERÍA INDUSTRIAL
CARRERA: INGENIERÍA EN TELEINFORMÁTICA**

LICENCIA GRATUITA INTRANSFERIBLE Y NO COMERCIAL DE LA OBRA CON
FINES NO ACADÉMICOS

Yo **TORRES NEIRA EMILY NICOLE** con C.C. No. **0941096372**, certifico que los contenidos desarrollados en este trabajo de titulación, cuyo título es **“SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGN MULTIPLATAFORMA.”** son de mi absoluta propiedad y responsabilidad, en conformidad al Artículo 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN*, autorizo la utilización de una licencia gratuita intransferible, para el uso no comercial de la presente obra a favor de la Universidad de Guayaquil.

Emily Torres Neira

TORRES NEIRA EMILY NICOLE

C.C.: 0941096372



**ANEXO VII.- CERTIFICADO PORCENTAJE
DE SIMILITUD
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Habiendo sido nombrado **ING. TELEC. TRUJILLO BORJA XIMENA FABIOLA, MG.** tutor del trabajo de titulación certifico que el presente trabajo de titulación ha sido elaborado por **TORRES NEIRA EMILY NICOLE**, con mi respectiva supervisión como requerimiento parcial para la obtención del título de INGENIERO EN TELEINFORMÁTICA.

Se informa que el trabajo de titulación: SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGNMULTIPLATAFORMA, ha sido orientado durante todo el periodo de ejecución en el programa Antiplagio quedando el 10% de coincidencia.



Link:

https://ev.turnitin.com/app/carta/en_us/?u=1133713395&lang=en_us&s=1&o=1904783411



Firmado electrónicamente por:

**XIMENA FABIOLA
TRUJILLO BORJA**

**ING. TELEC. TRUJILLO BORJA XIMENA FABIOLA, MG.
DOCENTE TUTOR
C.C. 0603375395
FECHA: 20 DE SEPTIEMBRE DEL 2022**



Universidad de Guayaquil

**ANEXO VI. - CERTIFICADO DEL DOCENTE-TUTOR
DEL TRABAJO DE TITULACIÓN
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 20 de septiembre del 2022

Sra.

Ing. Annabelle Lizarzaburu Mora, MG.

Directora de Carrera Ingeniería en Teleinformática / Telemática

**FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE
GUAYAQUIL**

Ciudad. -

De mis consideraciones:

Envío a Ud. el Informe correspondiente a la tutoría realizada al Trabajo de Titulación SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGN MULTIPLATAFORMA de la estudiante TORRES NEIRA EMILY NICOLE, indicando que ha cumplido con todos los parámetros establecidos en la normativavigente:

- El trabajo es el resultado de una investigación.
- El estudiante demuestra conocimiento profesional integral.
- El trabajo presenta una propuesta en el área de conocimiento.
- El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se adjunta el certificado de porcentaje de similitud y la valoración del trabajo de titulación con la respectiva calificación.

Dando por concluida esta tutoría de trabajo de titulación, **CERTIFICO**, para los fines pertinentes, que el estudiante está apto para continuar con el proceso de revisión final.



Firmado electrónicamente por:

**XIMENA FABIOLA
TRUJILLO BORJA**

Atentamente,

**ING. TELEC. TRUJILLO BORJA XIMENA FABIOLA, MG.
DOCENTE TUTOR
C.C. 0603375395
FECHA: 20 DE SEPTIEMBRE DEL 2022**



Universidad de Guayaquil

**ANEXO VIII.- INFORME DEL DOCENTE REVISOR
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 22 de septiembre de 2022

Sr (a).

Ing. Annabelle Lizarzaburu Mora, MG.

Director (a) de Carrera Ingeniería en Teleinformática / Telemática

**FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE
GUAYAQUIL**

Ciudad. -

De mis consideraciones:

Envío a Ud. el informe correspondiente a la REVISIÓN FINAL del Trabajo de Titulación **“SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGN MULTIPLATAFORMA”** del estudiante **TORRES NEIRA EMILY NICOLE**. Las gestiones realizadas me permiten indicar que el trabajo fue revisado considerando todos los parámetros establecidos en las normativas vigentes, en el cumplimiento de los siguientes aspectos:

Cumplimiento de requisitos de forma:

El título tiene un máximo de 8 palabras.

La memoria escrita se ajusta a la estructura establecida.

El documento se ajusta a las normas de escritura científica seleccionadas por la Facultad. La investigación es pertinente con la línea y sublíneas de investigación de la carrera.

Los soportes teóricos son de máximo 5 años.

La propuesta presentada es pertinente.

Cumplimiento con el Reglamento de Régimen Académico:

El trabajo es el resultado de una investigación.

El estudiante demuestra conocimiento profesional integral.

El trabajo presenta una propuesta en el área de conocimiento.

El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se indica que fue revisado, el certificado de porcentaje de similitud, la valoración del tutor, así como de las páginas preliminares solicitadas, lo cual indica el que el trabajo de investigación cumple con los requisitos exigidos.

Una vez concluida esta revisión, considero que el estudiante está apto para continuar el proceso de titulación. Particular que comunicamos a usted para los fines pertinentes.

Atentamente,



Firmado electrónicamente por:
**JAIRO GEOVANNY
VEINTIMILLA
ANDRADE**

ING. TELECOMUNICACIONES. VEINTIMILLA ANDRADE JAIRO GEOVANNY, MG

C.C:0922668025

FECHA: 22 DE SEPTIEMBRE DE 2022

Dedicatoria

A Dios por cada una de las veces en que todo parecía incierto me mostro el camino y me permitió llegar hasta aquí.

A mi mamá que ha estado apoyándome en cada uno de mis aciertos y derrotas, y le dedicare cada uno de mis logros siempre, porque sin su apoyo no podría haber llegado a donde estoy.

A mi hermana Jenyffer Torres quien ha sido mi apoyo fundamental en cada una de mis etapas de vida, y siempre encuentra la forma de encaminarme.

A mi hermana Jessica Torres por su apoyo incondicional, por creer en mí y motivarme a ser mejor.

Agradecimiento

A Dios por brindarme sabiduría y fortaleza para lograr cada una de mis metas.

A mis padres, mis hermanas, mis hermanos y a cada una de las personas que a lo largo de mi carrera aportaron un granito de arena ya sea económico o motivacional.

A mis amigos Denisse Diaz, Fabian Baque, Heidy Camba, Gabriela Freire y Álvaro Acosta, a quienes conocí a lo largo de la carrera y siempre creyeron en mí y confiaron que en algún momento me convertiría en ingeniera.

Índice del contenido

| N° | Descripción | Pág. |
|----|--------------|------|
| | Introducción | 17 |

Capítulo 1

| | | |
|--|-------------|---|
| | El problema | 4 |
|--|-------------|---|

| N° | Descripción | Pág. |
|--------|---|------|
| 1.1 | Problema de investigación | 18 |
| 1.1.1. | Planteamiento del problema | 18 |
| 1.2 | Formulación del problema de investigación | 19 |
| 1.3 | Sistematización del problema de investigación | 20 |
| 1.4 | Hipótesis | 20 |
| 1.5 | Operacionalización de las variables | 20 |
| 1.6 | Justificación de la investigación. | 21 |
| 1.7 | Objetivos de la Investigación. | 22 |
| 1.8 | Alcance de la investigación | 22 |

Capítulo II

Marco teórico

| N° | Descripción | Pág. |
|-----|--------------------------|------|
| 2.1 | Antecedentes del estudio | 23 |
| 2.2 | Fundamentación Teórica | 24 |
| 2.3 | Marco Legal | 30 |

Capítulo III

Metodología

| N° | Descripción | Pág. |
|-----|--|------|
| 3.1 | Diseño de la propuesta | 33 |
| 3.2 | Situación Actual de la empresa | 34 |
| 3.3 | Análisis y Resultados de la entrevista con el propietario de la compañía | 34 |
| 3.4 | Requerimientos solicitados por el dueño de la compañía | 35 |
| 3.5 | Diseño de la propuesta | 35 |

| | | |
|------|---------------------------------------|-----|
| 3.6 | Implementación del proyecto | 35 |
| 3.7 | Definición de System Backlog | 39 |
| 3.8 | Esquema de Conexión del Sistema | 42 |
| 3.9 | Dimensionamiento del Hardware | 42 |
| 3.10 | Flujo de Procesos dentro del Software | 43 |
| 3.11 | Pruebas de Funcionamiento | 54 |
| 3.12 | Costos de Elaboración | 65 |
| 3.13 | Análisis de los resultados | 65 |
| 3.14 | Conclusiones | 66 |
| 3.15 | Recomendaciones | 52 |
| | Anexos | 18 |
| | Bibliografía | 123 |

Índice de Tablas

| N° | Descripción | Pág. |
|-----------|---------------------------------|-------------|
| 1 | Operacionalización de Variables | 20 |
| 2 | Comparación entre servidores | 38 |
| 3 | System Backlog Sprints | 39 |
| 4 | Dimensionamiento del Hardware | 42 |
| 5 | Costos de Elaboración | 65 |

Índice de figuras

| Nº | Descripción | Pág. |
|-----------|--|-------------|
| 1 | Mapa modelado del Sistema de Inventario | 36 |
| 2 | Arquitectura del sistema | 42 |
| 3 | Flujo de procesos del Módulo Ajustes – Detalles de la compañía | 44 |
| 4 | Flujo de procesos del Módulo Ajustes – Ajustes Generales | 45 |
| 5 | Flujo de procesos del Módulo Ajustes Financiar | 45 |
| 6 | Flujo de procesos del Módulo Ajustes – Preferencias | 46 |
| 7 | Flujo de procesos del Módulo Ajustes – Imprimir Códigos de Barra | 46 |
| 8 | Flujo de procesos del Módulo clientes | 47 |
| 9 | Flujo de procesos del Módulo Inventario | 48 |
| 10 | Flujo de procesos del Módulo Validar Ítems | 49 |
| 11 | Flujo de procesos del Módulo Egresos | 50 |
| 12 | Flujo de procesos del Módulo Proveedores | 51 |
| 13 | Flujo de procesos del Módulo Pedidos | 52 |
| 14 | Flujo de procesos del Módulo Informes | 53 |
| 15 | Inicio de sesión del software de Inventario | 54 |
| 16 | Inicio de Pantalla de usuario administrativo del Sistema de Inventario | 54 |
| 17 | Creación de usuario en el Sistema de Inventario | 55 |
| 18 | Selección de roles de usuario en el Sistema de Inventario | 55 |
| 19 | Pantalla que muestra el listado de Ítems del Sistema de Inventario | 56 |
| 20 | Ingreso de un artículo al sistema | 56 |
| 21 | El código del producto no es editable | 57 |
| 22 | Ingreso del precio de los productos | 57 |
| 23 | Ingreso de precio de compra del pedido | 58 |
| 24 | Realización de orden de pedido | 58 |
| 25 | Listado de órdenes de pedido | 59 |
| 26 | Registro del ingreso del pedido | 59 |
| 27 | Ingreso de las cantidades de productos recibidos en el pedido | 60 |

| | | |
|----|--|----|
| 28 | Visualización del stock de los productos | 60 |
| 29 | Registro de una orden de egreso | 61 |
| 30 | Registro de una orden de egreso manual | 61 |
| 31 | Registro de un egreso | 62 |
| 32 | Listado de las ordenes de egreso | 62 |
| 33 | Requerimiento de acceso a la cámara | 63 |
| 34 | Reconocimiento de códigos de barra mediante cámara | 63 |
| 35 | Reconocimiento de códigos de barra de códigos que no existen | 64 |
| 36 | Reconocimiento de códigos de barra mediante cámara | 64 |
| 37 | Reconocimiento de códigos de barra de códigos que si existen | 65 |



ANEXO XIII.- RESUMEN DEL TRABAJO DE TITULACIÓN (ESPAÑOL)

FACULTAD: INGENIERÍA INDUSTRIAL
CARRERA: INGENIERÍA EN TELEINFORMÁTICA



“SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGN MULTIPLATAFORMA.”

Autor: Torres Neira Emily Nicole

Tutor: Ing. Telecom. Trujillo Borja Ximena Fabiola, Mg.

Resumen

En el presente trabajo se detalla el cumplimiento de cada uno de los requerimientos que fueron solicitados por la empresa SOLUDESING para la implementación de un sistema de inventario, el sistema consta con 8 módulos, el módulo de ingreso de pedido en el cual se permite realizar una solicitud de artículos a los diferentes proveedores ingresados, una vez que se ingresa un pedido y llega a la bodega de la empresa se realiza su ingreso y automáticamente el sistema presenta las existencias de los artículos en el módulo inventario, así también para realizar el egreso de un producto es necesario emitir una orden de egreso que mediante el código del producto busca dentro del stock y los descuenta una vez emitida dicha orden, permitiendo al usuario administrador contar con un apartado que presenta los reportes de cada uno de los ingresos y egresos de artículos con un fácil acceso según lo que fue solicitado, adicionalmente se implementó un módulo que consiste en el reconocimiento mediante el código del producto que permite establecer si un artículo pertenece al stock de la compañía o no ha sido registrado en el sistema ni en el inventario. Las recomendaciones y conclusiones que se indican son específicamente para la mejora de futuros trabajos o investigaciones con el fin de que el sistema sea asequible a empresas que tengan como actividad principal la compra, venta y distribución de artículos.

Palabras Claves: Sistema de inventario, stock, raspberry, proceso, inventariado.



**ANEXO XIV.- RESUMEN DEL TRABAJO DE
TITULACIÓN (INGLÉS)**



**FACULTAD: INGENIERÍA INDUSTRIAL
CARRERA: INGENIERÍA EN TELEINFORMÁTICA**

‘INVENTORY SYSTEM FOR SOLUDESIGN MULTIPLATFORM COMPANY.’

Author: Torres Neira Emily Nicole

Advisor: TE Trujillo Borja Ximena Fabiola, Mg.

Abstract

This work details the fulfillment of each of the requirements that were requested by SOLUDESING company for the implementation of an inventory system, the system consists of 8 modules, the order entry module which allows to make a request for items to different suppliers entered, once an order is registered and arrives at the warehouse of the company it is entered and automatically the system displays the stock of items in the inventory module, Also, in order to dispatch a product, it is necessary to issue a dispatch order that searches within the stock by means of the product code and deducts it once the order is issued, allowing the administrator user to have a section that shows the reports of each of the incoming and outgoing items with an easy access according to what was requested, additionally a module was implemented that consists of the recognition by means of the product code that allows establishing if an item belongs to the company's stock or has not been registered in the system or in the inventory. The recommendations and conclusions that are indicated are specifically for the improvement of future works or investigations with the purpose of making the system accessible to companies whose main activity is the purchase, sale and distribution of articles.

Keywords: Inventory system, stock, raspberry, process, inventory.

Introducción

El presente trabajo se centraliza en el desarrollo de una solución informática mediante el cual se va a realizar un proceso de inventario para la empresa SOLUDESING, teniendo en cuenta que este proceso se desarrolla de forma manual actualmente lo cual representa diversas amenazas como la pérdida de información y de activos con los que cuenta la compañía, por lo cual se presenta una solución que domine todo lo imprescindible para el adecuado funcionamiento del registro de información y etapas del proceso de inventariado.

Mediante la información recaudada por los autores Peña y Silva se establece que, en cuanto a las debilidades en el control de almacenes que afectan el control de las estructuras de inventario resultan del hecho de no controlar y resguardar bien la mercadería. Así mismo, una gestión de información ineficiente puede afectar, en términos de la falta precisa de conocimiento de los niveles de inventario en cualquier momento, estadísticas críticas para trazar compras, producción y distribución. Por otro lado, un punto débil en la gestión de las fuentes financieras puede hacer que las compras no sean posibles al no contar con el capital financiero para ejecutarlas y la carencia de control de administración no permite medir las estrategias en busca de perfeccionamientos. (Peña & Silva, 2016).

El propósito del desarrollo e implementación del software para la empresa SOLUDESING se enfoca en cubrir el funcionamiento de cada una de las etapas del registro del inventario, adicionalmente se añadió la identificación de serie de productos para poder establecer si un código visualizado mediante una cámara conectada en el sistema corresponde a uno previamente ingresado en el stock de la empresa, presentando un mensaje que señala las características del producto en caso de que si pertenezca al stock ingresado o un mensaje de error para cada una de las series que no se encuentran registradas en el sistema.

Capítulo I

Diseño de la Investigación

1.1 Problema de investigación.

1.1.1. Planteamiento del problema.

En toda actividad económica, física o tecnológica la cual preste sus servicios, bienes o productos es necesario llevar un control contable y de inventario para medir sus ventas, ingresos, egresos, ganancias para asegurarse de una buena rotación, flujo y movimiento de mercadería.

En todas las regiones comerciales, el control de inventario desempeña una posición crucial. Por tal motivo, se debe conocer cada uno de los factores que intervienen en el mal manejo de los inventarios, como elemento sustancial para obtener eficiencia extra dentro de la corporación empresarial comercial. (Cèpedes, Paz , Fèlix Jimenez, Leonardo Pèrez, & Yaite Pèrez, 2017) Por ende, cualquier empresa que maneje como actividad principal la venta y distribución de artículos, registrando todas sus actividades que desempeñan en documentación física representan un riesgo potencial de pérdida de información por ser obsoleto, que a su vez retrasa las actividades y la optimización del tiempo que dedican los empleados en el ingreso de artículos a la bodega de la empresa y el inventario por causa de un mal manejo.

En la actualidad las diferentes instituciones deben manejar una eficiente revisión en todos sus módulos para lograr un incremento de la eficacia en la labor de la misma. Mediante la realización de un inventario de bienes de un almacén se requiere una técnica sustentable para disminuir errores que conlleven a ingresar más de una vez un artículo idéntico, no agregar a otros productos que no se encuentren registrados dentro del stock, detectar robos o extravíos. (Delgado, Rivero , & Lince, 2019). En el ejercicio de las actividades diarias de una empresa es necesario llevar un control adecuado de la mercadería con la que se cuenta en las bodegas de las instalaciones, ya que esto permite evitar el extravío de artículos lo que representa un riesgo vital del capital con el que se cuenta, y a su vez en caso de que se presente algún robo de mercadería se permita establecer de forma rápida el cálculo en el que estaría valorada la pérdida para que la empresa pueda tomar las medidas pertinentes en esta situación.

En un registro de inventario se debe tener un mejor control de la información que es ingresada en la documentación de la empresa, por lo cual se debe evitar cometer errores de

registro por parte de los empleados, y encontrar la manera de limitar que el personal cometa el grave error de ingresar más de una vez un producto que ya cuenta con un código único, teniendo en cuenta que una vez que un producto es registrado en el stock y ya posee un código se debe agregar existencias y no asignar un nuevo código para un producto, debido a que se trata de un producto de las mismas características y especificaciones previamente almacenado en la bodega de la empresa.

Por ende, al realizar un mal almacenamiento de los diferentes productos que se manejan en el stock y se encuentran en bodega, puede generar el extravío de mercancía dentro de las mismas instalaciones de la empresa, ya que se encuentran erróneamente ubicadas en un área diferente de la bodega a la que pertenecen, esto se puede suscitar debido a que un empleado por un descuido genere una mala categorización de los productos electrónicos.

La empresa SOLUDESING se dedica a la compra y venta de artículos electrónico, manejando el registro de inventario de forma manual, lo que representa riesgos para la empresa y la información que maneja, al no contar con un respaldo de información se hace casi imposible rastrear el flujo de actividades que se deben seguir para el proceso de inventario. Al no contar con los medios disponibles para realizar un informe de cada uno de los movimientos del producto es casi imposible manejar un control de ingreso y egreso de artículos que pertenecen al stock de la compañía.

1.2 Formulación del problema de investigación.

¿Cuáles son los riesgos que con lleva tener un inventariado manual para la empresa SOLUDESING?

¿Cuán importante es registrar de forma adecuada el ingreso de mercadería en el inventario de la empresa?

¿Qué riesgos implican no cumplir con las fases del proceso de registro de inventario que se manejan en la empresa?

¿Qué es lo que implica registrar de forma errónea la etiqueta que corresponde a un producto en particular?

1.3 Sistematización del problema de investigación.

1.3.1 Variable Dependiente:

Reducir los problemas presentados por el mal manejo de inventario.

1.3.2 Variable Independiente:

Desarrollo y adaptación de un software de inventario de artículos electrónicos que atienda las necesidades de la empresa SOLUDESING.

1.4 Hipótesis.

Desarrollando un sistema de inventario que se ajuste a las necesidades de la empresa SOLUDESING se reduce cada uno de los problemas suscitados dentro del manejo de inventario como lo son el mal registro de información, lo cual representa un peligro para el manejo de existencias de productos dentro del inventario.

1.5 Operacionalización de las variables

Tabla N°1 Operacionalización de Variables

| Variable | Dimensión | Indicador |
|--|------------|--|
| Reducir los problemas presentados por el mal manejo de inventario. | Productos | El usuario debe realizar el ingreso del producto en el sistema. |
| | Inventario | Tener un control de cada uno de los productos que ingresan y salen del stock. |
| Desarrollo y adaptación de un software de inventario de artículos electrónicos que atienda las necesidades de la empresa SOLUDESING. | Diseño | El software debe ser intuitivo para que el usuario realice de forma rápida el flujo del inventario. |
| | Contenido | Se deben cubrir cada una de las etapas del inventariado, registrando de forma correcta el ingreso y egreso de productos. |

Elaboración: Torres Neira Emily.

1.6 Justificación de la investigación.

Para una compañía dedicada a la adquisición y comercio de productos se vuelve necesario almacenar y registrar de forma correcta cada uno de los productos con los que cuenta, ya que, al no hacerlo se pueden presentar problemas como un mal etiquetado de productos, es decir que un producto cuente con un código de barra que no pertenece al artículo ingresado y registrado, esto puede causar que el personal pierda tiempo valioso en buscar el producto al que corresponde dicho código de barra, esto se debe a que al registrar de forma manual el inventario pueden existir errores de confusión por parte del personal de bodega que se encarga del etiquetado de los productos.

Para realizar un correcto inventariado de información es necesario determinar las fases que se manejan dentro del proceso del mismo para la empresa que es sujeto de estudio, una vez definidas las fases que con lleva realizar el inventario se debe proveer una herramienta capaz de cumplir satisfactoriamente cada una de estas, por lo cual una vez que se deje de registrar información de forma manual se logra llevar un control de cada uno de los procesos registrados evitándose problemas de etiquetación y de mal almacenamiento de productos dentro de las instalaciones de la compañía. (Tomala , 2019).

Es de vital importancia que cualquier empresa que maneje un stock de artículos tenga a disposición una forma de rastrear y evidenciar el movimiento de los artículos dentro de la empresa, ya que esto permite tener el conocimiento necesario a los administradores de cómo se encuentra la situación de la compañía.

Actualmente para las empresas se sugiere que es mucho más beneficioso y competitivo contar con un acceso a equipos que no generen grandes costos, esto se refiere a que permitan aumentar la productividad y eficiencia de los métodos que desempeñan. (Delgado, Rivero , & Lince, 2019). Mientras que para una empresa que no cuente con una automatización de sus procesos, se considera casi imposible poder garantizar que su ejercicio económico presente ganancias representativas a la compañía, ya que, al contar con procesos de manejo obsoletos retrasan sus actividades y se convierte en debilidades frente a la competencia.

1.7 Objetivos de la Investigación.

1.7.1 Objetivo General.

Implementar un sistema de inventario digital para la empresa SOLUDESIGN.

1.7.2 Objetivos Específicos.

1. Determinar la situación actual de la empresa y sus requerimientos en el manejo de inventario.
2. Diseñar un sistema de inventario digital que cumpla con los requerimientos de la empresa.
3. Realizar las pruebas de funcionalidad del sistema implementado.

1.8 Alcance de la investigación

Establecer la situación actual de la empresa, con esto se permite desarrollar un sistema de inventario que se ajuste a las necesidades previamente establecidas, evitando el mal registro de productos y mitigando el incorrecto etiquetado de los productos existentes en la bodega, presentando reportes de manejos de la compra y venta de equipos electrónicos. Y como medida adicional se presenta un reconocimiento de las etiquetas ingresadas en el sistema para determinar si algún producto dentro del stock que se encuentra en bodega cuenta con una etiqueta que no pertenece a la empresa, lo que puede darse debido al mal ingreso o impresión de un código de barra en alguno de los diferentes equipos o productos electrónicos con los que se cuenta.

Capítulo II

Marco Teórico

2.1 Antecedentes del estudio

Amazon es el portal de ventas en línea más grande a nivel mundial. Inicialmente, permitía agregar sus productos desde un supervisor, o mediante la forma de ingresar documentos de carga y enviarlos a su sitio web, pero en 2009 lanzaron una API pública para permitir la descentralización de estas cargas.

En Amazon, todos los productos se distribuyen en partes estratégicas de los diferentes almacenes para que puedan estar siempre disponibles a la vista de los usuarios. Esto se basa totalmente en la teoría que determina que no es necesario que todos los artículos similares estén juntos. Amazon hace posible que esta sea una buena manera de conocer todos los productos y dónde están ubicados mediante un software inteligente llamado Quiet Logistics que geolocaliza cada objeto. Con esto es posible reconocer en cada momento qué mercancías están disponibles y cuáles han sido compradas.

Este operador se basa principalmente en el intercambio verbal cliente-servidor con los servidores de Amazon, en el que, a través de llamadas HTTP y mensajes XML, todo tipo de carga que se transformó en factible manualmente y, además, nuevas alternativas que consisten en adquirir reseñas sobre precios, productos, ventas o competencia. Estas capacidades brindan acceso a las instalaciones logísticas de Amazon, en las que el proveedor puede optar por tomar su catálogo (pagando una tarifa de garaje).

Por este motivo, Amazon ofrece una gran cantidad de documentación, que permite la decisión de los servicios. Estas llamadas, están encapsuladas en sus sobres, y cumplen con las regulaciones de tamaño y proveedor, utilizando un dispositivo de Throtting para rechazar paquetes al mismo tiempo que recibe demasiados del mismo vendedor. (Gutiérrez, 2016).

Alvendi es un software libre, sin límites de hechos o caducidad para el control de existencias, ventas y compras. Permite generar reportes impresos o en vista previa, incluye opciones que permiten exportar en archivo Excel, funciona con el dispositivo operativo Windows. Este software es fácil de aplicar ya que tiene una interfaz definitivamente intuitiva, y al mismo tiempo un software efectivo en el manejo de ingresos y compras con

diversas características y, en concreto, informes de mercancías, clientes, proveedores y movimientos de compra o venta, resumida y en detalle. (Rivero, Delgado , & Ernesto, 2019).

MercadoLibre es un sistema web que opera basándose en el comercio electrónico y permite a sus usuarios acceder a través del enlace www.Mercadolibre.Com. La plataforma está diseñada para brindar a los usuarios mecanismos efectivos para comprar, vender, pagar, acumular, poner en el mercado y consultar a través de internet, y está diseñada de acuerdo con las características específicas de América Latina, un área con una población de más de seiscientos millones de seres humanos. Los Marketplace se han convertido en eje y motor para la mejora de ofertas complementarias en las áreas de pagos, envíos, gestión de redes, entre otras. Estas tendencias contribuyen al desarrollo de la experiencia de usuario quienes reciben una solución más completa y robusta y su recurrencia, mejorando la monetización de las transacciones. Las ofertas complementarias se han convertido en unidades empresariales comerciales en sí mismas y compiten con los líderes locales en todas sus industrias.

Mercado Libre ha crecido con fuerza en los últimos años en un contexto macroeconómico destructivo y logró crecer hasta ser el principal consolidador de la oferta y la demanda en la región. La valoración de la empresa se basa en su fuerte papel agresivo en el mercado y la resiliencia de su versión comercial, y puede cerrar con el paso de los años en la medida en que continúa con su método de entorno logrando un mayor control en la experiencia de compra y sigue reinventando su negocio y generando ingresos a sus usuarios. (Herrera, 2015).

2.2 Fundamentación Teórica

2.2.1 Software

El software es lo que da ayuda operativa al hardware, es decir, es la mejora lógica y algorítmica de la máquina informática que permite su funcionamiento. Un programa de software es un sistema, utilidad o máquina, que permite el funcionamiento de una computadora, cajeros automáticos, teléfonos inteligentes, sistema electrónico, entre otros. El hardware es la cuestión física y el software es el operativo. (Gòmez Sebastià & Moraleda, 2020).

2.2.2 Sistema Web

Son ficheros que tienen la posibilidad de guardar en un website mediante el cual se permite el ingreso usando el protocolo del lenguaje. Para usar una aplicación web a partir de un aparato en especial, basta con contar con un portal con acceso a internet introducido para notar su funcionamiento. (Pantaleo & Rinaudo, 2015).

2.2.3 Base de Datos

Se consideran almacenes de información, ya que, se caracterizan por contener una gran proporción de información relacionada entre sí lo cual permite tener acceso de forma fácil y accesible. Están formadas de mínimo una tabla que guarda datos. Cada una de la tabla tiene secciones y filas. Cada una de los apartados mantienen un fragmento de información acerca de cada elemento que es necesario conservar, cada fila de la tabla es la composición un registro. (Cordero, 2017).

2.2.3.1 Particularidades de las bases de datos

- Información sin redundancia de datos.
- Alta fiabilidad de información.
- Mejor organización de la información.
- Ingreso concurrente por diferentes clientes.
- Estabilidad de ingreso y revisión.

2.2.3.2 Tipos de Bases de Datos

En las diversas tipologías de base de datos, se encuentran:

- MySQL: con una estructura GPL basada en servidor. Se usa ampliamente en diversas regiones de redes y sistemas, a pesar de que su composición no se recomienda lo suficiente para grandes volúmenes de información. (Rollet, 2015).

Las funciones más notorias de MySQL son:

- ✓ Soporte de servicios
- ✓ Librerías.
- ✓ Averiguación por escrito.
- ✓ Caché de investigaciones (para incrementar el rendimiento).

- PostgreSQL: Son cercos de base de datos de alta calidad. Controla un gran conjunto de datos y generalmente se usa en intranets.
- Access: se desarrolla con la ayuda de la empresa Microsoft, se complementa de una forma menos complicada con el ambiente gráfico de Visual Basic, actualmente sigue operativa y se localiza dentro del paquete principal de Office.
- Microsoft SQL Server: Es un diagrama que se fundamenta totalmente en SQL, que incluye su propio alcance de registros y su DBMS interno.

2.2.3.3 Tipos de datos

Los diferentes lenguajes implementan diversos tipos que tienen la posibilidad de cambiar de nombre. Entre los más frecuentes se puede definir a:

- ✓ Numérico: se puede encontrar números completos "sin decimales" y "decimales" legítimos.
- ✓ Booleanos: poseen 2 condiciones: positivo "Sí" e inexistente "No".
- ✓ Fechas: se registran las fechas para disponer de un reverso de este, su uso puede ser variado entre los cuales puede integrar logs de servicios.
- ✓ Alfanumérico: tiene imágenes y letras. Extensión delimitada (255 caracteres).
- ✓ Auto incrementables: son numéricos enteros que se aumentan.

2.2.3.4 Consistencia de datos

El control de las redundancias de datos reduce sustancialmente el riesgo de irregularidades. En el caso de que la información se guarde solo una vez, cualquier actualización debe realizarse mejor una única vez. El marco en sí mismo puede hacer frente a la verificación de que todos los duplicados son honestos y sólidos. (Oddone & Chernobilsky, 2016).

2.2.3.5 Mejora en la integridad de datos

La honestidad de la base de datos se refiere a la legitimidad y consistencia de los registros almacenados. La honestidad generalmente se comunica a través de obstáculos o principios de los que no se puede abusar. (Fritz, Montejano, Garcia, & Bast, 2016).

2.2.4 Lenguajes de Programación

Tienen estrictos patrones de sintaxis, retratos y frases específicas se utilizan para recopilar un software de computadora portátil. Consiguen usarse para hacer aplicaciones que manipulan la conducta mecánica y genuina de un dispositivo, para formular cálculos con exactitud o como una forma de comunicación humana. (Arias, 2017).

Está hecho de varias representaciones pictóricas y dictados sintácticos y semánticos que representan su composición y el ritmo de sus factores. La técnica a través de la cual se crea, analiza, inquiriere, acopla (si es crítico) y conserva el código fuente de una aplicación de computadora se conoce como desarrollo. Además, por la implementación de tácticas aptas, por los avances que la acompañan:

- Optimización genuina del sistema para atender una dificultad en específico.
- Escribir la lógica del desarrollo usando un lenguaje específico (codificación del desarrollo).
- Experimentar e averiguar el programa.
- Optimización del expediente.

2.2.4.1 Codificación o Implementación

La codificación es la tarea de componer el diseño del problema (como lo indica el buen juicio del diagrama de flujo o pseudocódigo), en una progresión de direcciones factor por punto de uso.

La disposición de direcciones compuestas para un programa se conoce como código de suministro y está escrito en un lenguaje de programación que puede ser de etapa baja, media o alta.

2.2.4.1 Herramientas de Desarrollo de Software

Las instrumentales de desarrollo son paquetes de computadora que se utilizan para generar diferentes archivos con su asistencia. (Pantaleo & Rinaudo, 2015). Conforme el incremento del tiempo se hizo imprescindible debido a varios de las instrucciones que se generan son más complicados para realizar un uso más amistoso con el cliente.

2.2.4.1.1 Sublime Text

Este software permite tener diversos documentos abiertos a través de vistas, incluso brindando la comodidad de usar distintos documentos. Además, la codificación puede tolerar un variado conjunto de lenguajes de programación, es decir, una sección que permite moverse de inmediato. Este software tiene autoguardado, numerosas posibilidades de personalización, cuenta con una respetable variedad de máquinas para intercambiar el desarrollo y transformar las obligaciones.

2.2.4.1.2 DbForge Studio para SQL Server

Mencionando otro de los equipos para desarrollar un programa, este DbForge Studio es un ámbito protegido para la mejora, administración, informes de registros y, a menudo, el examen de SQL Server.

2.2.4.2 Herramientas para programación

2.2.4.2.1 Framework Laravel

Laravel se define como la composición de código abierto que se genera mediante aplicaciones web y gestiones con PHP 5. Permitiendo implementar una composición de lenguaje delicada y explícita para producir código de forma inmediata y consintiendo una mayor proporción de funciones.

2.2.4.2.2 HTML

La composición de programación para disponer de HTML es la implementación de un ordenador de escritura, por ejemplo, el excelente provisto por Windows (Bloc de notas), por MS Office (Word), en general es realizado con mayor programa de software superior para suministrar el marcado y alineación de registros.

Dichos instrumentos tienen la oportunidad de ser visualizados a través de visitantes o navegadores de páginas de la red en Internet, lo que incluye Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, entre otros. Además, está Dynamic HTML (DHTML), que es una optimización de HTML Form 4. Zero que admite realizar reajustes ejemplificar, recomendaciones que salen del sitio web mensaje por mensaje o el efecto de mejora en la rotación de páginas. promoviendo la moda. Y página web. (Fossati, 2018).

2.2.4.2.3 JavaScript

El lenguaje JavaScript tuvo sus inicios en el año 1996 a través de la empresa Netscape, la cual lo introdujo como una innovación en la programación web en su Navegador Netscape Navigator 2.0, dicho navegador se encargaba de ser un intermediario y que ejecutaría las sentencias de JavaScript para un mejor desempeño, esta inclusión fue de gran ayuda en la animación y comunicación del lenguaje HTML, este lenguaje con el pasar del tiempo se volvió indispensable en cualquier aplicación Web, tanto fue su crecimiento en todos los navegadores web hoy lo poseen.

JavaScript como tal es un lenguaje establecido en los navegadores web, este lenguaje no necesita compilar o empaquetar los archivos, se pueden ejecutar y probar a través de cualquier navegador web disponible, este lenguaje se basa en crear bibliotecas de entorno que son de uso múltiple en aplicaciones web, este lenguaje fue inspirado en JAVA, pero no tienen otra similitud más que el nombre, pero son dos lenguajes completamente diferentes. (Luna, 2019).

2.2.4.3 Herramientas de Hardware para la implementación del sistema

2.2.4.3.1 Raspberry Pi 4

Es una nueva edición que incluye 4 GB de RAM y se permite conectar hasta dos dispositivos de visualización de video 4K al mismo tiempo mediante puertos micro HDMI. Adicionalmente dos puertos USB 2.0, la placa también tiene puertos USB 3.0 que permite realizar trasposos de datos y se carguen más rápido mediante USB. La placa Modelo B es una mini computadora del tamaño de poco similar de una tarjeta de crédito. Puede tener una pequeña computadora portátil completa organizada para usar.

2.2.4.3.2 Cámara de Video

Una videocámara o cámara de video es una herramienta normalmente transportable que permite grabar imágenes y sonidos, convirtiéndolos en indicadores eléctricos que pueden ser reproducidos por una herramienta seleccionada.

2.2.4.3.3 Lector de Código de Barra

El descifrador o escáner de código de barras se define como un artefacto informático que utiliza un láser integrado para estudiar un código de barras y generar los registros (la numeración) que representa el código, es decir no una foto. Así desaparece la tediosa tarea

de digitar los códigos y se disminuye sensiblemente el tiempo y los recursos invertidos en ella. Además, con este dispositivo se asegura que no haya errores que puedan poner en peligro la trazabilidad de la muestra.

2.3 Marco Legal

2.3.1 Leyes que rigen en la implementación del software

2.3.1.1 Ley de Uso de Software Libre en el Ecuador

Según el Art.2 se establece que se considera como Software Libre a aquellas aplicaciones portátiles que pueden ser utilizadas y distribuidas sin restricciones, que permiten acceder a sus códigos fuente y que sus programas pueden ser mejorados. Por otro lado, en el Art.4 se establece que el uso de programas de software propietarios (no libres) está autorizado solo mientras no se encuentre un recurso de software libre que satisfaga los requerimientos del usuario, o mientras sea propenso a la seguridad nacional, o esté se encuentre en un punto sin retorno es decir se convierta en obsoleto. Con esto se determina en el Art. 5 de la ley presentada que ya sea un software libre o propietario debe atender cada una de las necesidades presentadas por los usuarios que adquieren el mismo para cumplir con sus actividades diarias.

En el desarrollo de esta propuesta se considera que este software es de tipo propietario ya que, es desarrollado únicamente con el fin de atender las necesidades de la empresa SOLUDESING y el código programado es único y no se encuentra a disposición de usuarios para que realicen modificaciones, para poder realizar alguna modificación en el software se debe tener las credenciales de acceso al servidor de bajo costo con el que cuenta este software. Estos artículos son mencionados y especificados con el fin de evidenciar el tipo de software que es presentado a lo largo del desarrollo de esta propuesta.

2.3.1.2 Ley de Propiedad Intelectual

2.3.1.2.1 Sección V - Disposiciones especiales sobre ciertas obras Párrafo primero de los programas del ordenador

Mediante el Art. 28 se detalla que la implementación de un programa en un ordenador se considera una obra literaria y se deben proteger como tales, esa salvaguardia se da independientemente de que hayan sido incorporadas en un ordenador o en cualquier manera que constituya una forma de expresión, así sea en forma comprensible por el ser humano

(código fuente) o en forma comprensible por máquina (código objeto), ya sean por programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y generalmente, esos recursos que conformen la composición secuencian y organización del programa. Mientras que, en el Art. 29 se define que el titular de un programa de ordenador, es decir el individuo natural o jurídica que toma la idea y responsabilidad de la ejecución de la obra. Se considerará titular al individuo cuyo nombre conste en la obra o sus copias de manera común.

Esto se menciona ya que, al ser un software de uso privado pertenece únicamente a la empresa utilizada como sujeto de estudio, esta solución puede ser modificada y editada en base a las necesidades que presenta la empresa.

2.3.1.3 Código orgánico de la economía social de los conocimientos creatividad e innovación

2.3.1.3.1 Acceso y soberanía del conocimiento en entornos digitales e informáticos

Según el Art. 39 se especifica que la entrada al entendimiento independiente debe ser seguro en espacios digitales e informáticos, por medio de las tecnologías de la información y comunicaciones elaboradas en plataformas compatibles entre sí; así como el despliegue en infraestructura de telecomunicaciones, el desarrollo de contenidos y aplicaciones digitales y la apropiación de tecnologías, conforman un factor transversal de la economía social de los conocimientos, la creatividad y la innovación y es imprescindible para poder hacer la satisfacción de necesidades y el efectivo goce de derechos.

Es importante presentar una solución informática que sea segura y amigable con los usuarios, que toda la información se encuentre a disposición y sea enlazada entre sí, para que se encuentre disponible en cada uno de los apartados que forman parte del sistema.

2.3.1.4 Código Orgánico Integral Penal

Mediante el Art. 234 se establece que el derecho de paso no autorizado en un ambiente de telecomunicaciones, informático, o telemático será sancionado mientras que el individuo que sin aprobación, consienta total o parcialmente a un dispositivo portátil o método telemático o de telecomunicaciones o permanezca en él contra su voluntad de quien tiene el derecho legítimo o de aprovecharse ilegítimamente del derecho de acceso al terminar,

ajustar un portal que se encuentre en internet, desviar o redirigir visitantes del sitio de digital o de comandos de voz u brindar ofertas que esos métodos no brindan generalmente fuera de pagar a los proveedores de servicios válidos o autorizados, podría ser penado con despojo de independencia de tres a cinco años.

Dado el caso de que algún usuario acceda sin las debidas autorizaciones al código del software o intente realizar ataques maliciosos será sancionado según lo que se especifica en la ley previamente detallada.

Capítulo III

Diseño de la Propuesta

3.1 Diseño de la propuesta

3.1.1 Enfoque de la Investigación

Efectuando el progreso de la investigación se tiene en cuenta los siguientes enfoques significativos:

- Investigación Aplicada
- Investigación Cualitativa
- Investigación Cuantitativa

3.1.1.1 Investigación Aplicada

Se trata de una investigación aplicada ya que se recopila la información respecto a una empresa en particular, se recolecta los requerimientos iniciales de la misma y se desarrolla e implementan dentro del sistema cada uno de los requerimientos y soluciones informáticas que permitan mejorar de manera eficiente el progreso de las operaciones de la compañía.

3.1.1.2 Investigación Cuantitativa

Mediante la utilización de esta técnica se permitirá coleccionar y examinar la información de la diferencia a como se manejaban los procesos antes en la empresa de forma manual, midiendo en cuanto a tiempo lo que se tardaba el personal de la compañía en realizar los procesos de inventariado y la diferenciación en cuanto le toma registrar la información necesaria para los procesos que son automatizados dentro del sistema de inventario.

3.1.1.3 Investigación Cualitativa

Se establece una reunión con el dueño de la empresa para determinar los requerimientos que necesita la compañía y establecer la situación actual, mediante preguntas objetivas con el fin de diseñar un enfoque, diseño de la propuesta e implementación de la solución informática.

3.2 Situación Actual de la empresa

La empresa SOLUDESING inició sus actividades en el norte de la localidad de Guayaquil en una pequeña oficina, dedicándose inicialmente a la compra y venta de artículos electrónicos bajo pedido, poco a poco incremento sus ventas lo que conllevó que contaran con una bodega principal donde se almacenan cada uno de los electrónicos con los que cuenta la empresa.

Esta compañía tiene como actividad comercial la venta y distribución de equipos electrónicos, llevando a cabo sus registros de inventario en documentación física, lo cual genera un retraso en el desempeño de sus actividades diarias en el negocio.

Esta empresa no posee un control adecuado de la mercadería con la que cuenta, por lo cual su principal requerimiento es una herramienta que le permita llevar a cabo el registro de cada una de sus mercaderías para optimizar el tiempo y recursos con los que cuenta.

Esta empresa posee en su stock productos electrónicos que en su mayoría son diferentes marcas y modelos de teléfonos celulares, distribuyendo también cámaras, video cámaras, computadoras, teclados, consolas de video juego, entre otros. El personal de la compañía que se dedica a realizar el inventario es dos personas encargada de realizar pedidos a los diferentes proveedores que trabajan con la empresa, una persona encargada de revisar que el pedido realizado llegue completo y de forma satisfactoria a la empresa y posteriormente registrarla en el inventario manual que manejan actualmente para ingresarlas en el stock, dos bodegueros que son los encargados de ubicar los productos dentro de la bodega principal, dos personas dedicadas a vender y registrar la salida de los diferentes producto en las ordenes de ingreso que se manejan en papel, una persona encargada de supervisar que se cumplan de forma satisfactoria todos los proceso, el dueño de la empresa que se dedica a dar seguimiento de todo el personal y los diferentes procesos que se realizan en las actividades de la compañía, y una persona que se dedica a llevar la contabilidad de la empresa.

3.3 Análisis y Resultados de la entrevista con el propietario de la compañía.

En esta entrevista se estableció la situación inicial de la compañía, es decir como manejan cada uno de los procesos necesarios para cumplir de forma satisfactoria el proceso de inventario de la compañía, estableciendo cada uno de los requerimientos necesarios para la implementación del software desarrollado. Verificación de las preguntas realizadas en la entrevista en el Anexo I.

Como resultado de la entrevista se realizó un listado de cada uno de los scripts que son necesarios para la elaboración de cada uno de los módulos que se manejaran dentro del sistema de inventario para el correcto funcionamiento, es necesario contar con un módulo que enliste cada uno de los productos ingresados en el sistema, que se visualice las existencias y posea una foto característica del artículo, adicionalmente como el dueño de la empresa indico que se generan problemas por la errónea etiquetación de artículos se

desarrolla una herramienta que permita reconocer mediante una cámara si el código de barra pertenece a un artículo registrado en el stock para determinar si ese producto se encuentra etiquetado correctamente, el sistema también debe contar con un módulo de pedidos ya que es necesario registrar el ingreso de artículos electrónicos a la compañía y así también un módulo de egresos para constatar la salida del producto mediante las ventas que se realizan, y por último, un módulo en el que se presenten informes de cada una de las ingresos y egresos de los productos y las ganancias que generan a la compañía.

3.4 Requerimientos solicitados por el dueño de la compañía.

En la reunión se establecieron los siguientes requerimientos:

- Poder realizar el registro del pedido solicitado a los proveedores que trabajan generalmente con la compañía.
- Se debe poseer un instrumento que autorice el ingreso cada uno de los productos con una categoría y generar un código de barra único para el mismo.
- Contar con un inventario que especifique que productos se tiene en el stock, cual fue el costo del producto, a cuanto se vende al por mayor y menor.
- Realizar registros de órdenes de egreso del producto y que se vayan descontando del stock una vez que los clientes realicen la compra.

3.5 Diseño de la propuesta

La solución planteada para la empresa SOLUDESING es el desarrollo de una aplicación web que atienda las necesidades de la compañía en cada una de las fases del proceso que con lleva realizar el inventario. Para la implementación de esta propuesta se debe realizar la adquisición de una raspberry pi 4 que funciona como un servidor donde se instala la información que permite conectarse a cada uno de los usuarios mediante cualquier navegador desde cualquier terminal electrónico sin distinción del sistema operativo con el que cuente.

3.6 Implementación del proyecto

Este apartado se compone de la fase de ejecución de los diversos requisitos previamente obtenidos en la entrevista con el dueño de la empresa, implementando soluciones mediante la programación que permiten automatizar cada uno de sus procesos en su gran mayoría.

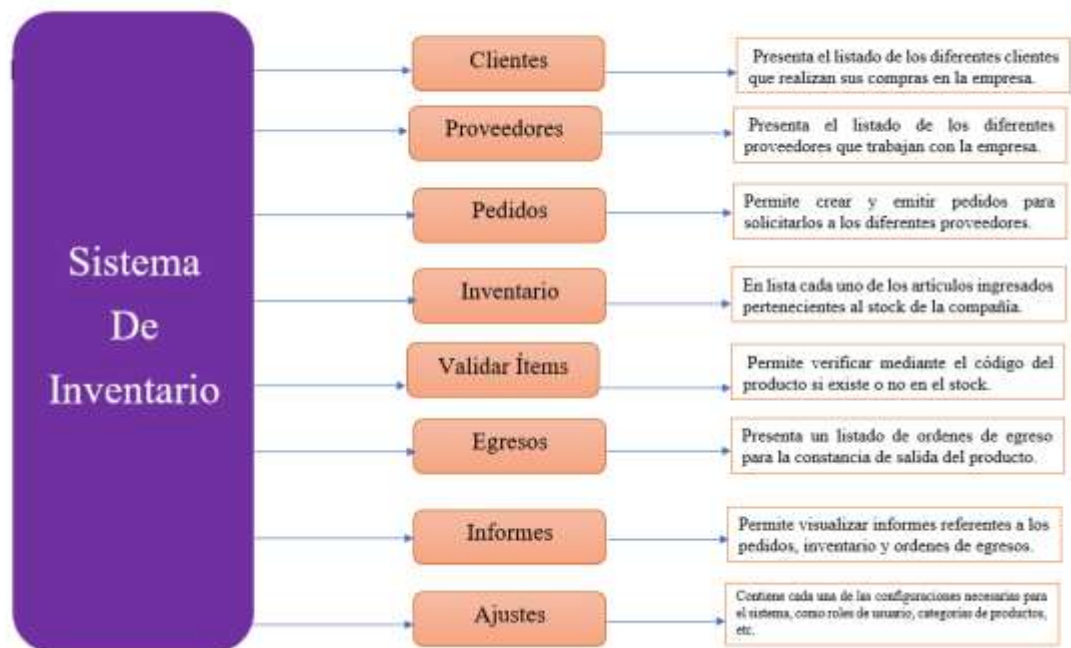


Figura 1 Mapa modelado del Sistema de Inventario. Desarrollado a partir del software.
Elaborado por Torres Neira Emily.

3.6.1 Módulo Clientes

Contiene un listado de cada uno de los clientes que han sido ingresados en el sistema para realizar el registro de una orden de egreso, permitiendo agregar nuevos clientes en caso de ser necesario.

3.6.2 Módulo Proveedores

Contiene un listado de cada uno de los proveedores que han sido ingresados en el sistema, permitiendo agregar nuevos proveedores en caso de ser necesario.

3.6.3 Módulo Pedidos

En este apartado se permite crear pedidos para llevar un registro de cada uno de los productos solicitados a los diferentes proveedores, es necesario registrar una orden de pedido ya que al realizar uno se estima que una cantidad de productos llegue pero debido a diversos motivos como que los proveedores no tengan a disposición el artículo solicitado se deba retrasar la entrega de ese producto en particular, por ello el sistema cuenta con la opción de pedido, y una vez que llegue a la bodega de la empresa los artículos son registrados en la opción ingreso de pedido donde se realiza el ingreso del producto al stock de artículos de la compañía.

3.6.4 Módulo Inventario

Permite ingresar cada uno de los productos que estarán disponibles en el stock de la compañía, asignándoles un código que debe ser único dentro del sistema, una categoría, precio de compra, costos de comercio mayorista y minorista, una foto identificativa del artículo y automáticamente se presenta un listado de cada uno de los diferentes artículos que son distribuidos por la empresa. En la parte superior de la vista del sistema se cuenta con un apartado que detalla el total de artículos ingresados, el valor de los costos del stock y el valor de ganancia que se tendrá al realizar las ventas de estos artículos.

3.6.5 Módulo Validar Ítems

En esta opción del sistema se permite reconocer mediante una cámara conectada al dispositivo enlazado con el software, si el código del producto ingresado se encuentra dentro del stock de los artículos o no, en caso de que el código del producto si este registrado en el sistema se presenta un mensaje que contiene el nombre del producto y serie, caso contrario se presenta un mensaje de error indicando que el código visualizado mediante la cámara no pertenece ni ha sido registrado en el sistema.

3.6.6 Módulo Egresos

En este apartado se deben registrar los artículos que han sido vendidos para llevar un control de cada uno de los productos que han salido del stock de la compañía, una vez ingresado mediante la pistola lectora de código de barras la serie de un producto se realiza la disminución del stock del artículo en el módulo de inventario.

3.6.7 Módulo Informes

Se presentan los diferentes informes relacionados con los productos que se tienen en stock, un reporte de cada una de las órdenes de egreso mostrando un gráfico representativo de las ganancias obtenidas, un informe de compras que representa cada uno de los pedidos realizados y un reporte de los diferentes usuarios que fueron establecidos en el software.

3.6.8 Módulo Ajustes

En este apartado se realiza la configuración de cada uno de los módulos del sistema, se configura las bodegas pertenecientes a la compañía, los diferentes roles asignados a cada usuario lo que delimita su acceso a la información contenida dentro del sistema, los datos de la compañía como logo, nombre de la empresa, correo institucional. En este apartado se

permite imprimir los códigos de barra que pertenecen a los artículos que se encuentran en el stock de la empresa. También se permite ingresar nuevos roles de usuario y establecer cada uno de los miembros del equipo, cuando se encuentre en sesión el usuario administrador.

3.6.9 Servidor Web

Maquina física con una construcción Linux para disminuir coste del software, el cual se conforma con un bando de datos y asistencia Apache HTTP fundamentales para la presentación en red del software de inventario.

Tabla N. 2 *Comparación entre servidores*

| Características | Hosting GoDaddy | HP ML30 | Raspberry pi 4 |
|-------------------------|--------------------------------|--------------------------------|-------------------------------|
| CPU | Procesador Quad Core 1 núcleo. | Intel Xeon E3-1220v6 Quad Core | Quad core Cortex-A72 (ARM v8) |
| RAM | 512 Mb | 4 Gb | 2 GB |
| Espacio en disco | 100 Gb | 1 Tb | 128 Gb |
| Bases de datos | 10 x 1 GB | Ilimitado | Ilimitado |
| MySQL | | compartido | compartido |

Elaboración: Torres Neira Emily.

Luego de realizar la comparación de servidores, se establece que la raspberry pi 4 es el mejor servidor en relación calidad precio, ya que lo que se propone en esta solución informática es un servidor de bajo costo que satisfaga las necesidades de la empresa SOLUDESING.

3.6.10 Metodología Utilizada en el desarrollo del proyecto

La metodología por usar es SCRUM, es un modelo de despliegue e implantación que optimiza los recursos reduciendo el tiempo necesario para completar las fases que lo componen. Para ello se necesita conocer las 5 fases en el ciclo de desarrollo de la metodología Scrum.

1. **Concepto:** Se define los requerimientos del personal y se asigna al equipo que se encargara del desarrollo.
2. **Especulación:** Se realizarán los requerimientos obtenidos del personal y se definirán los alcances del desarrollo del software, entre el equipo se definen los costos y un cronograma para las reuniones futuras.
3. **Exploración:** Se realizan los ajustes solicitados por el cliente.

4. **Revisión:** El equipo hace una revisión del proyecto y ejecuta el plan de pruebas.
5. **Cierre:** Se entrega una versión del proyecto según la fecha acordada para la revisión del cliente, siendo susceptible a cambios.

3.6.10.1 Etapas de la metodología del proyecto

En este apartado se indica los procesos adecuados que se deben seguir para un correcto desarrollo obteniendo los resultados requeridos en el tiempo establecido. Para lo cual, las etapas que se deben de considerar para el desarrollo del proyecto son: análisis, diseño, desarrollo, pruebas e instalación.

Para el desarrollo del proyecto, es necesario la implementación de una metodología que permita llevar de manera organizada un control en los procesos, por tal razón se utiliza la metodología SCRUM.

Se realizarán varias fases las cuales son:

- PLANIFICACION DEL BACKLOG
- DESARROLLO DE SPRINTS
- REVISION DE SPRINTS

3.7 Definición de System Backlog

Se detalla un listado en donde de manera ordenada se presentan cada uno de los requerimientos que se necesitan para el desarrollo del sistema de inventario. Siendo esta la fuente que podría ayudar en caso de realizar cambios durante el desarrollo del proyecto.

Tabla N. 3 *System Backlog – Sprints*

| System Backlog – Sprints | | | | |
|--------------------------|--|--------|-------|--|
| N° | REQUERIMIENTO | SPRINT | HORAS | OBJETIVO |
| R1 | Arquitectura del sistema | 1 | 8 | Establecer la arquitectura de la propuesta. |
| R2 | Permitir el registro de los usuarios por parte del administrador | 1 | 5 | Validar los datos: nombres y correo. |
| R3 | El administrador otorga los roles a los usuarios | 1 | 4 | Control sobre el acceso a las configuraciones del sistema. |

| | | | | |
|------------|--|---|---|---|
| R4 | Configuración de página principal del sistema. | 1 | 5 | Crear interfaz amigable con los usuarios. |
| R5 | Configuración de reportes | 1 | 5 | Llevar el control de los diferentes módulos establecidos en el sistema. |
| R6 | Configuración del módulo inventario | 2 | 5 | Crear interfaz para la revisión de los productos configurados. |
| R7 | Configuración del módulo clientes | 2 | 5 | Crear interfaz para la revisión de los clientes configurados. |
| R8 | Configuración del módulo proveedores | 2 | 5 | Crear interfaz para la revisión de los proveedores configurados. |
| R9 | Configuración del módulo pedidos | 2 | 5 | Crear interfaz para la revisión de las ordenes de pedidos y pedidos realizados. |
| R10 | Configuración del módulo egresos | 2 | 5 | Crear interfaz para la revisión de las ordenes de egresos y egresos realizados. |
| R11 | Configuración del módulo validar ítems | 2 | 5 | Crear interfaz para la revisión de las ordenes de pedidos y pedidos realizados. |
| R12 | Desarrollo de interfaz de usuario. | 2 | 8 | Permitir al usuario el acceso al sistema. |
| R13 | Desarrollo de interfaz del usuario. | 2 | 5 | Permitir que el usuario pueda modificar sus datos. |

| | | | | |
|------------|---|----|---|---|
| R14 | Desarrollo de interfaz de los reportes. | .3 | 5 | Mantener el control de los entradas y salidas de productos en el sistema. |
| R15 | Desarrollo de interfaz del producto. | .3 | 5 | Permitir al administrador la configuración del producto. |
| R16 | Desarrollo de interfaz del validar ítem. | .3 | 5 | Permitir al administrador visualizar si un ítem existe en el sistema o no. |
| R17 | Desarrollo de interfaz de los proveedores. | .3 | 5 | Permitir al administrador la configuración de los proveedores. |
| R18 | Desarrollo de interfaz de los clientes. | .3 | 5 | Permitir al administrador la configuración de los clientes. |
| R18 | Desarrollo de interfaz de las configuraciones (mantenimientos del sistema). | 3 | 5 | Otorgar acceso al administrador a los mantenimientos de configuración del sistema |

***Elaboración:** Torres Neira Emily.*

3.8 Esquema de Conexión del Sistema

Como se observa en la siguiente imagen el sistema utiliza como servidor una raspberry pi 4 que mediante la conexión a internet permite que realicen la conexión de hasta 20 usuarios en el sistema mediante el IP de la raspberry.

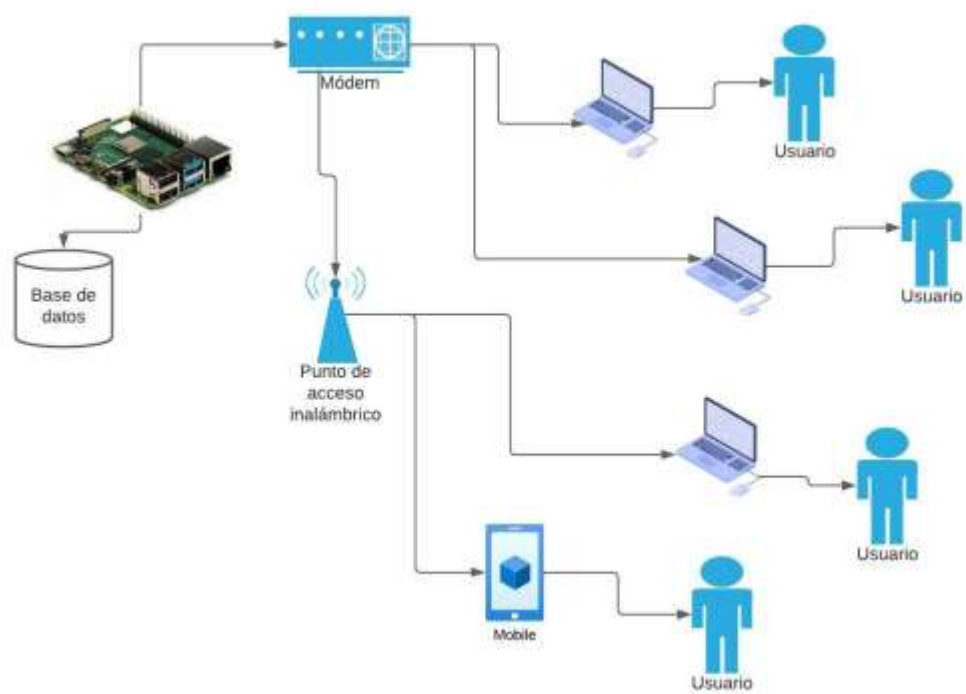


Figura 2 Arquitectura del Sistema. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.9 Dimensionamiento del Hardware

Se presenta una tabla que abarca los detalles de cuanta información necesita el software y cuanta información se puede almacenar en la raspberry como servidor.

Tabla N°4 Dimensionamiento del Hardware

| Raspberry Pi 4 | Sistema de Inventario |
|-------------------------|---|
| Sistema operativo Linux | Se conecta a cualquier sistema operativo, ya que es un sistema web. |
| Memoria RAM de 4 GB | X (no necesita memoria RAM) |

128 GB

El sistema necesita 20 GB para almacenar los diferentes archivos de información distribuidos de la siguiente forma:

- archivos de programación de software: 3 GB
 - base de datos: 17 GB
-

Elaboración: Torres Neira Emily.

3.10 Flujo de Procesos dentro del Software

A continuación, se detalla el flujo que con lleva cada uno de los procesos dentro del sistema.

3.10.1 Módulo Ajustes

En este apartado se realizan cada una de las configuraciones necesarias para que el sistema funciones correctamente, se debe establecer la información referente a la empresa, se debe seguir un flujo para detallar la información de la compañía y determinar los roles de ingreso de los diferentes usuarios que acceden al sistema.

A continuación, se establece el flujo que se debe cumplir para registrar cada uno de los detalles de la compañía dentro del sistema, los ajustes generales donde se establecen las categorías de productos que se manejan, cada uno de los impuestos que se cobran mientras los clientes hacen una compra, así como también los días en los que deben cancelar los clientes una compra, los métodos de pago y el tipo de moneda que se acepta en la venta de artículos, también se configura el formato de fecha, el número de filas presentadas por páginas, y por último, imprimir n cantidades de códigos de barras correspondientes a los diferentes productos ingresados en el stock del sistema de inventario.

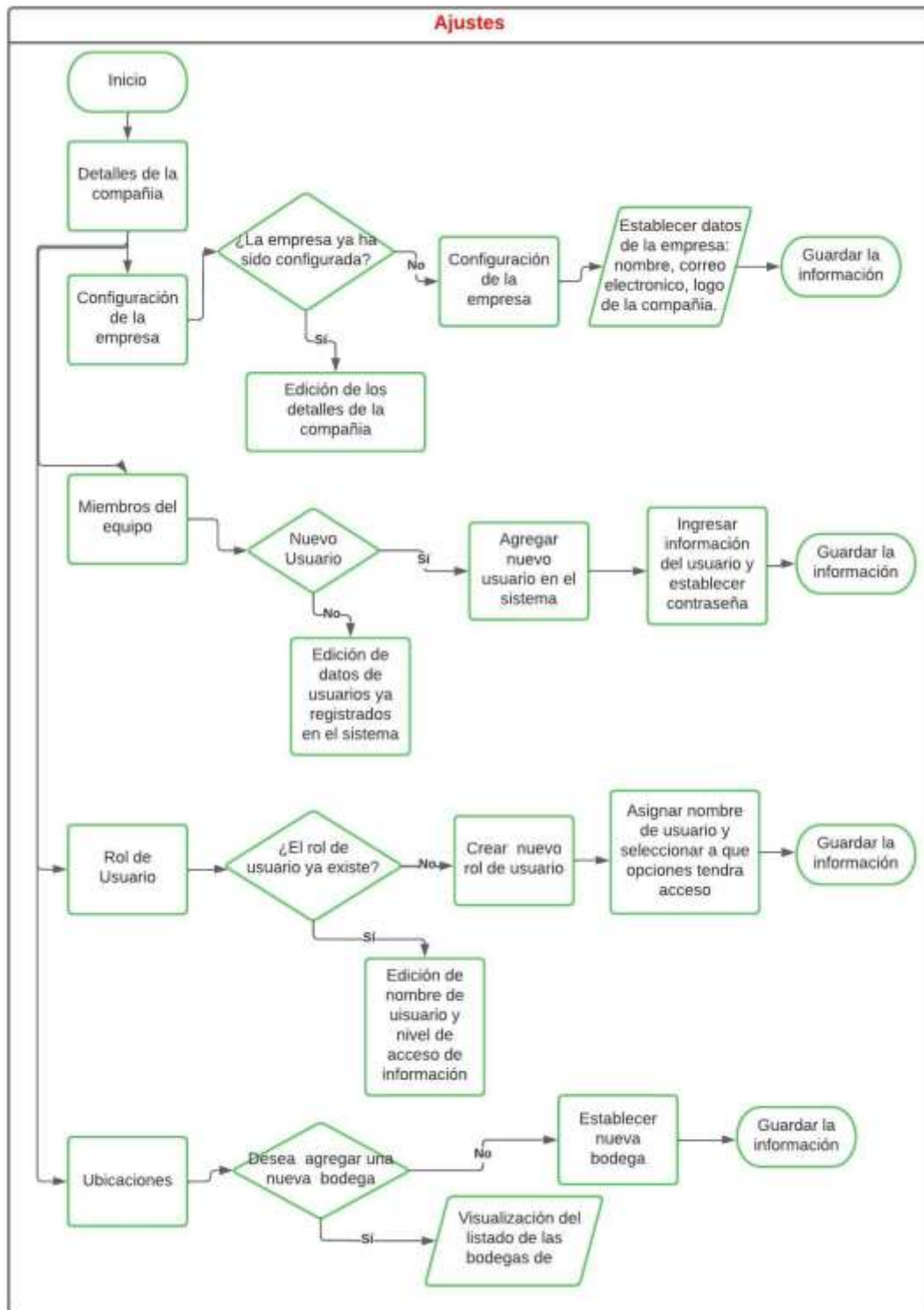


Figura 3 Flujo de procesos del Módulo Ajustes – Detalles de la Compañía del Sistema de Inventario.
Imagen creada por la autora. Elaborado por Torres Neira Emily.

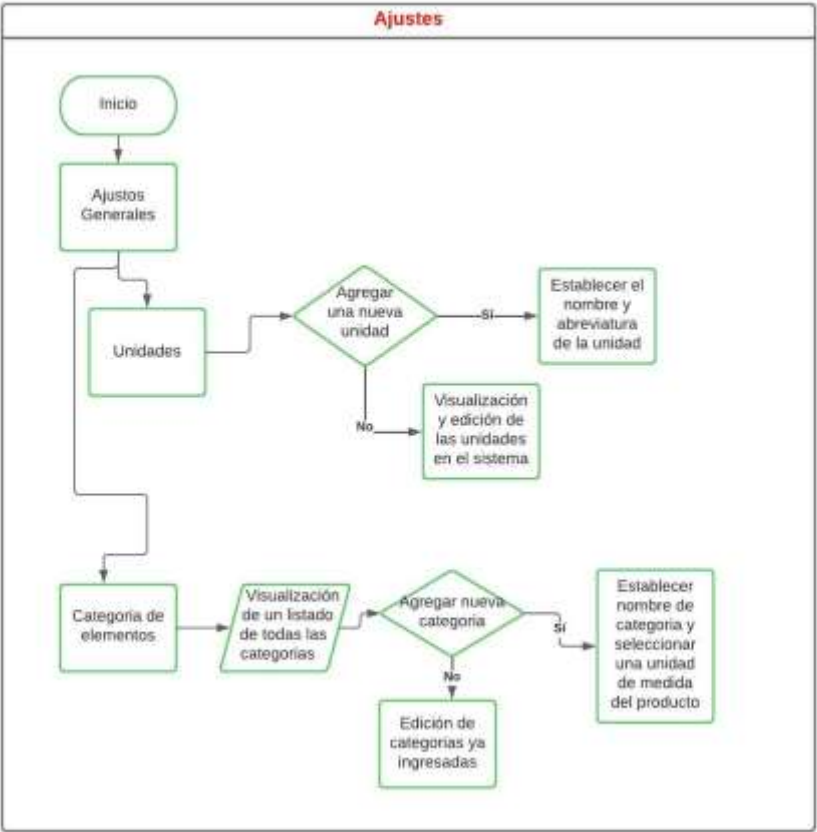


Figura 4 Flujo de procesos del Módulo Ajustes – Ajustes Generales del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

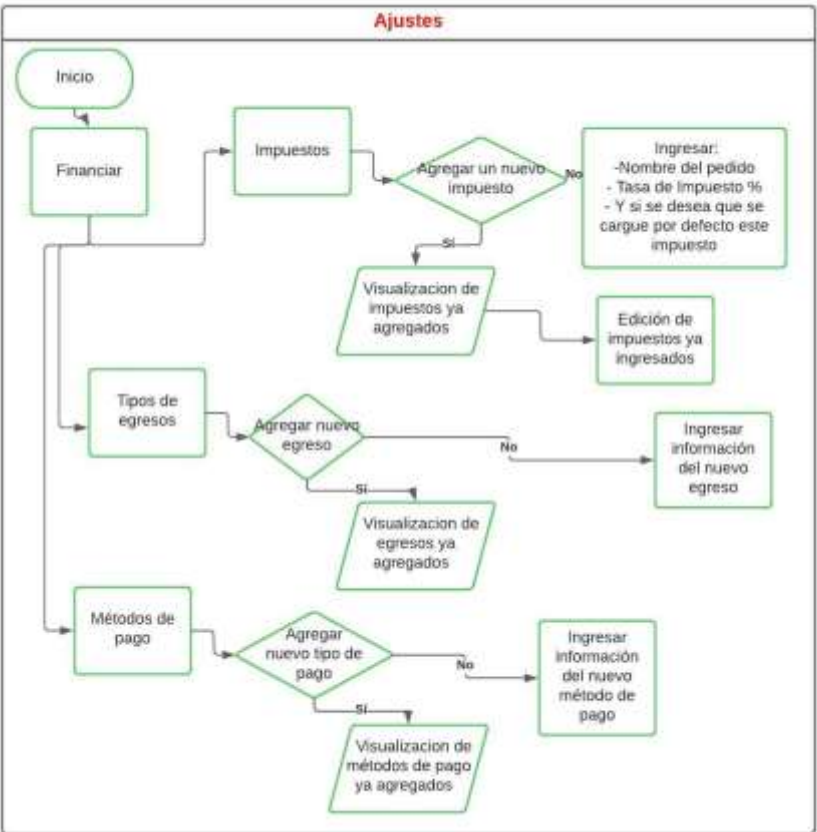


Figura 5 Flujo de procesos del Módulo Ajustes – Financiar del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

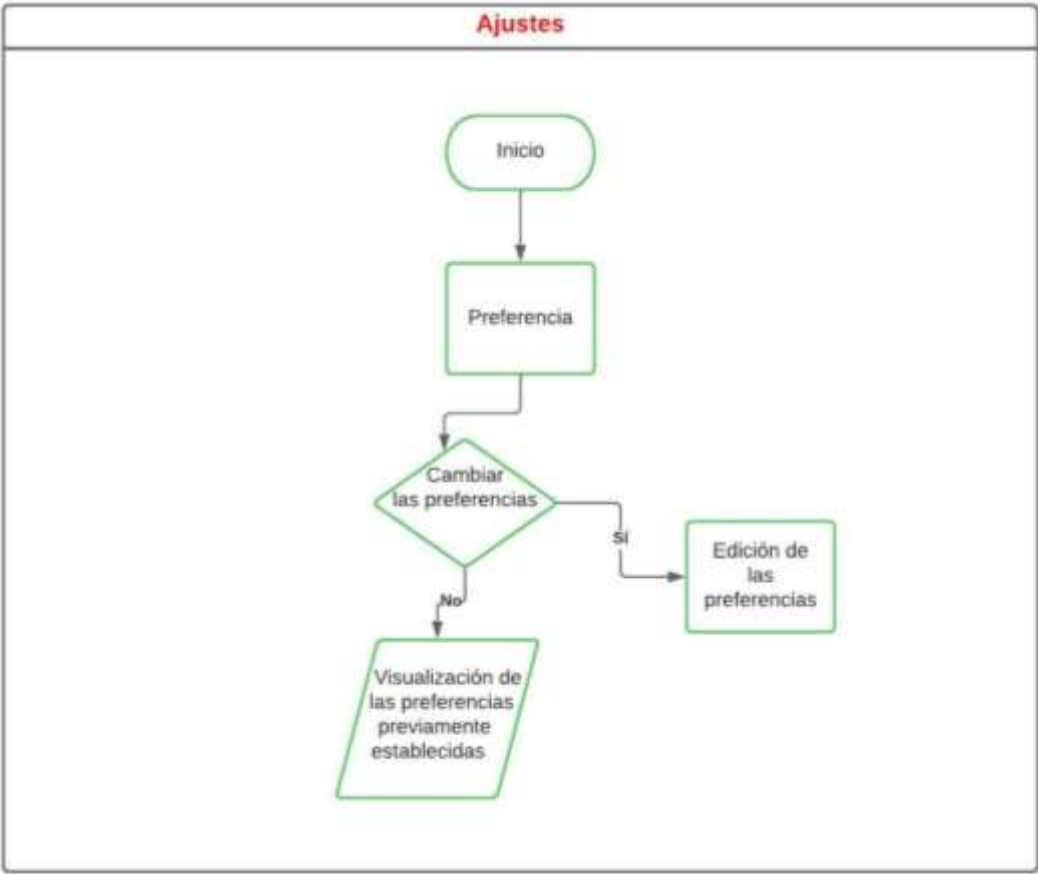


Figura 6 Flujo de procesos del Módulo Ajustes – Preferencia del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

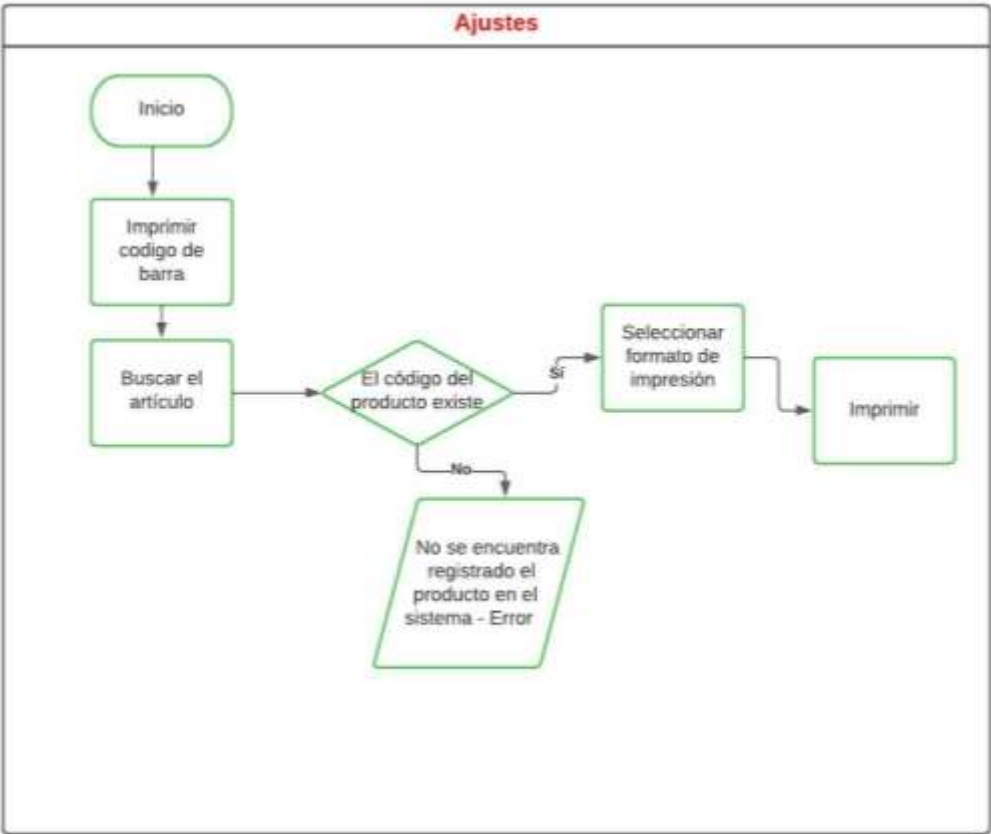


Figura 7 Flujo de procesos del Módulo Ajustes – Imprimir Código de Barra del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.2 Módulo Clientes

Al ingresar a este módulo se carga un listado de cada uno de los clientes ingresados en el sistema, en caso de que algún cliente requiera una orden de egreso de un producto y no se encuentre registrado en el sistema este apartado permite realizar la creación de clientes en la opción agregar nuevo cliente.

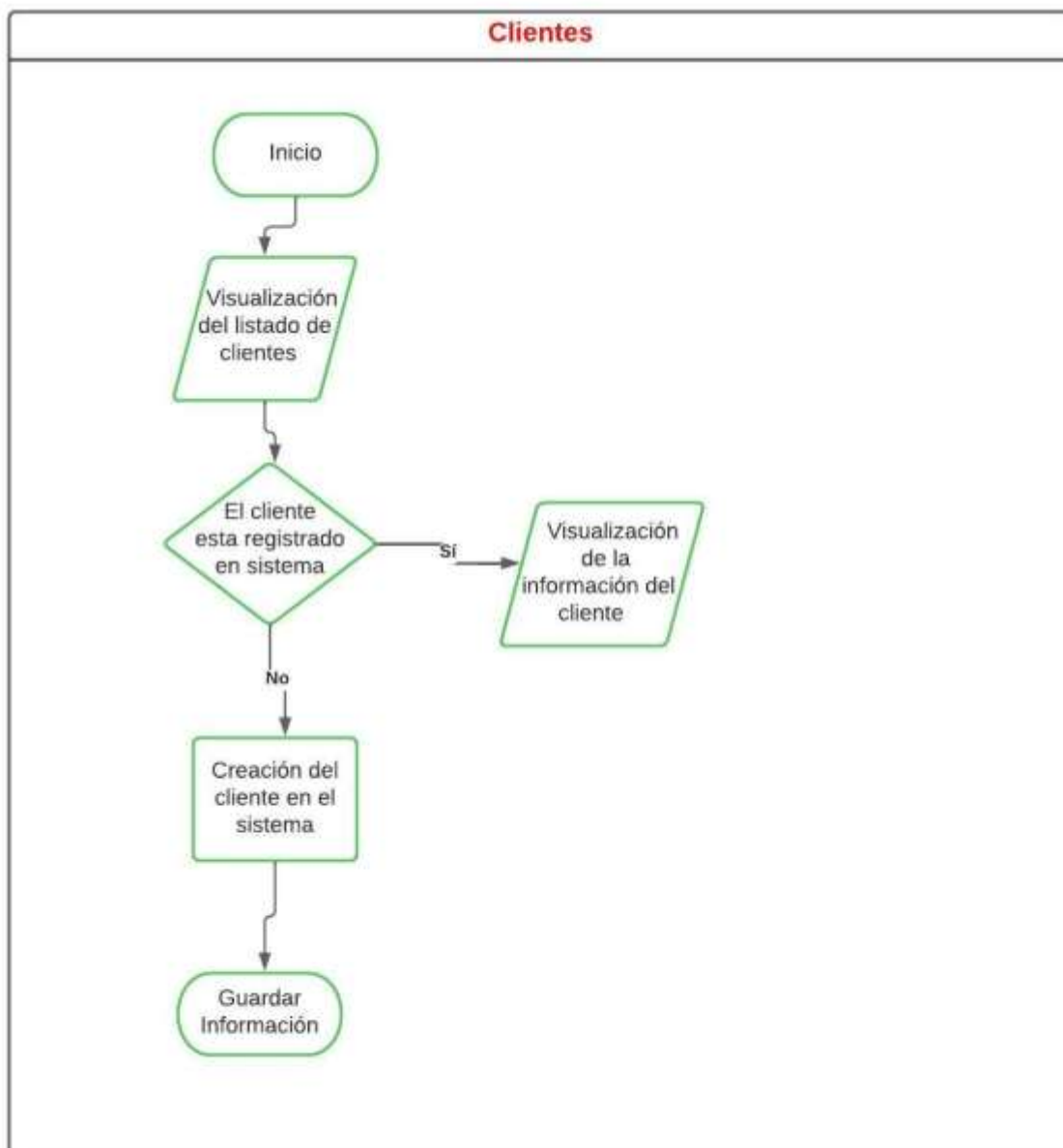


Figura 8 Flujo de procesos del Módulo Clientes del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.3 Módulo Inventario

Se visualiza un listado de cada uno de los artículos electrónicos registrados en el sistema, adicionalmente se permite ingresar nuevos artículos en el sistema, es necesario establecer un código único en el sistema para el producto ingresado, así como también establecer el valor en el que fue comprado el artículo y el costo al que será vendido.

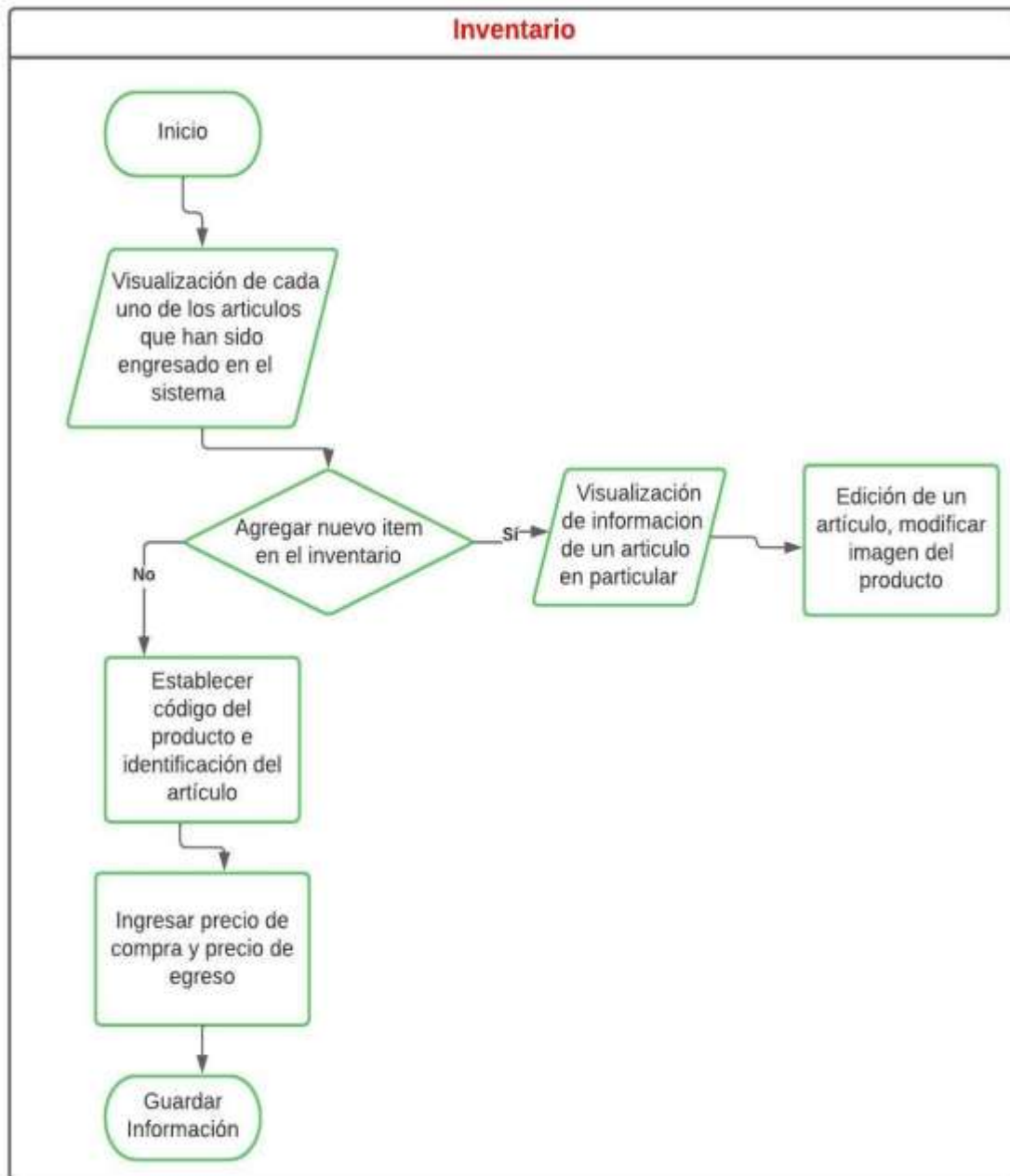


Figura 9 Flujo de procesos del Módulo Inventario del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.4 Módulo Validar Ítems

Al ingresar en este módulo inmediatamente se presenta un mensaje para aceptar la conexión con la cámara con la que cuente el dispositivo con el cual se encuentra iniciada sesión en el sistema, al conectarse la cámara debe poner al frente la etiqueta del producto que se desea validar, una vez que se presiona el botón buscar se presentara un mensaje que hace referencia a si la serie ingresada pertenece o no al sistema.

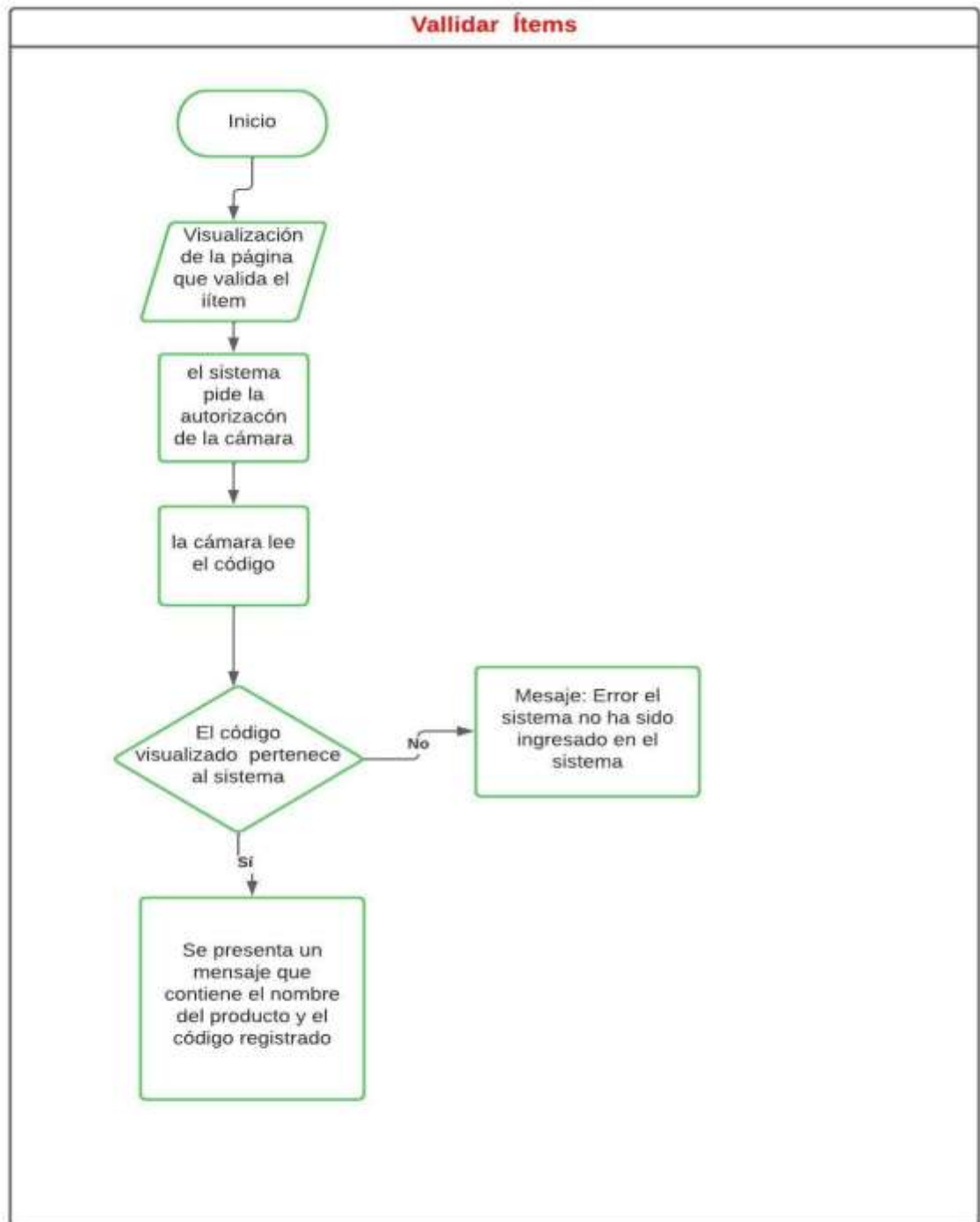


Figura 10 Flujo de procesos del Módulo Validar Ítems del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.5 Módulo Egreso

Permite visualizar cada una de las ordenes ingresadas en el sistema para constatar el egreso de un producto y poder realizar la disminución en el inventario del artículo que fue vendido por la empresa.

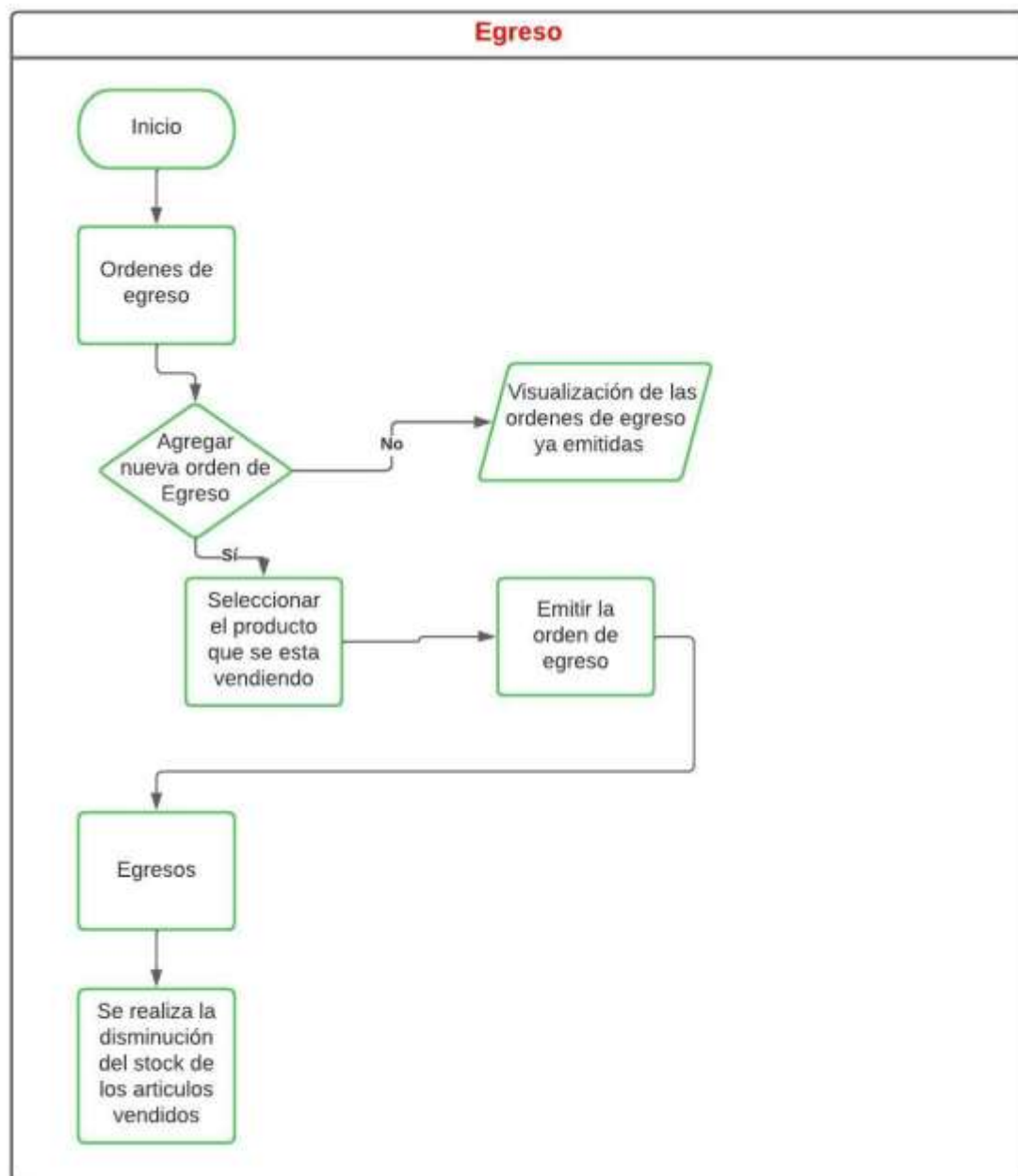


Figura 11 Flujo de procesos del Módulo Egresos del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.6 Módulo Proveedores

Permite la visualización del listado de cada uno de los proveedores registrados en el sistema, así como también permite registrar nuevos proveedores en el sistema a los cuales se realizará una compra después.

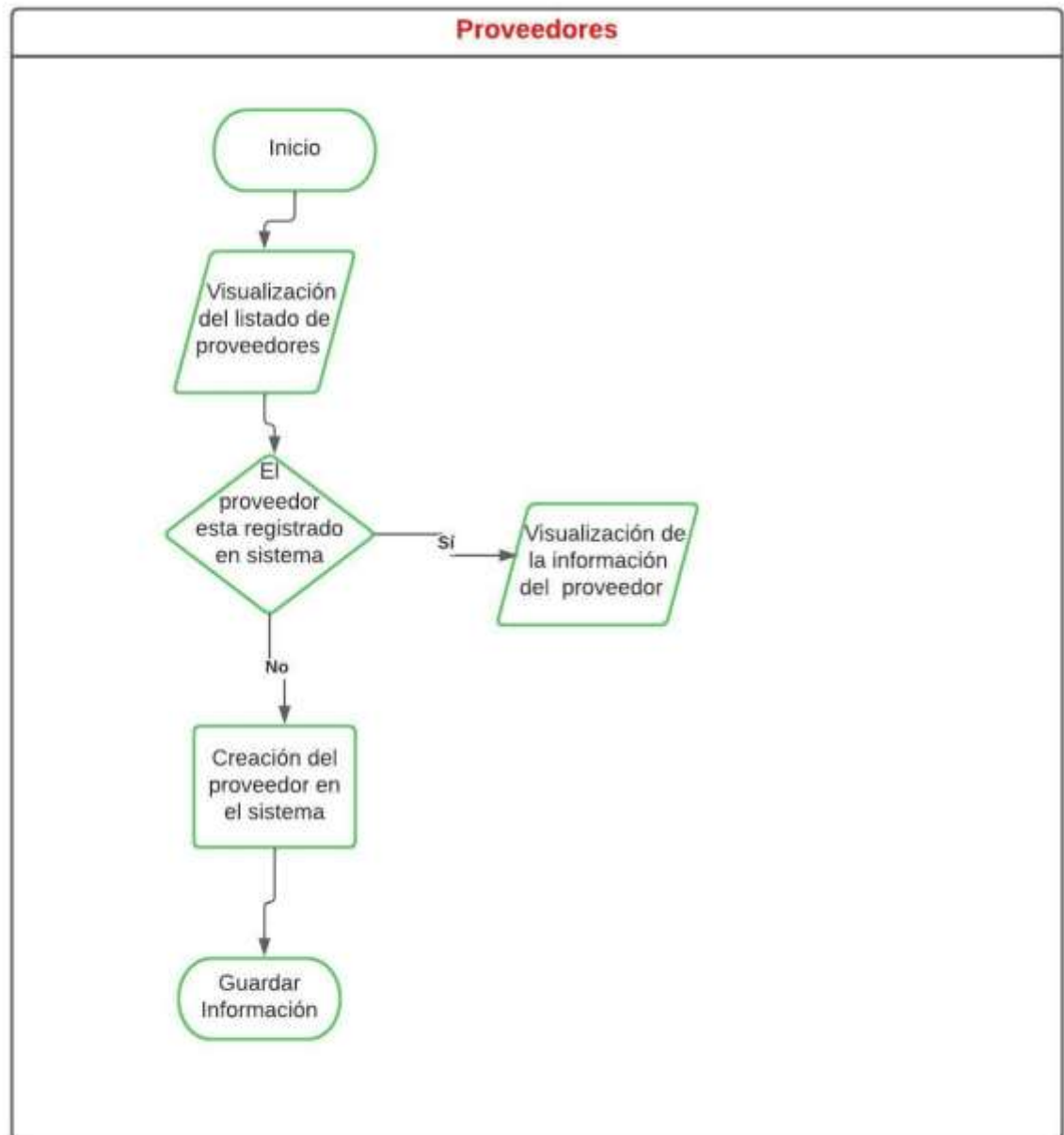


Figura 12 Flujo de procesos del Módulo Proveedores del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.7 Módulo Pedidos

En este apartado se registran cada una de las ordenes de pedidos solicitados a los diferentes proveedores que distribuyen artículos a la empresa, una vez realizada la orden del pedido se debe ingresar el pedido ya que pueden llegar los productos con fallas o incompletos entonces una vez que llegue el pedido a la bodega de la empresa es ingresado en el stock del sistema.

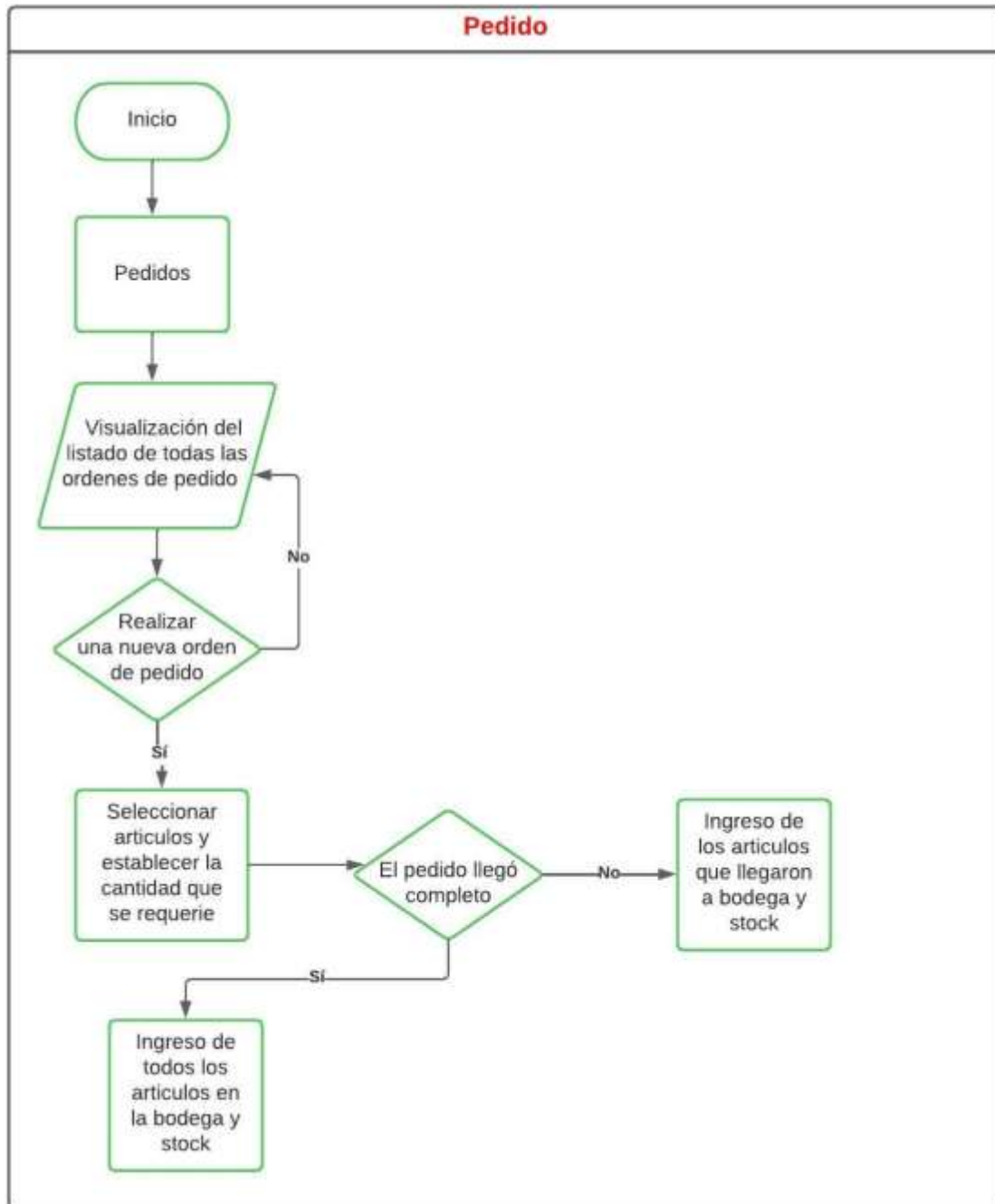


Figura 13 Flujo de procesos del Módulo Pedidos del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.10.8 Módulo Informes

Recopila información de cada uno de los módulos del sistema y presenta diversos informes como de los artículos con los que cuenta la empresa, los costos de ventas y compras de los diversos productos que se manejan en el stock y los miembros que pertenecen al equipo es decir un informe de cada uno de los usuarios que ingresan al sistema.

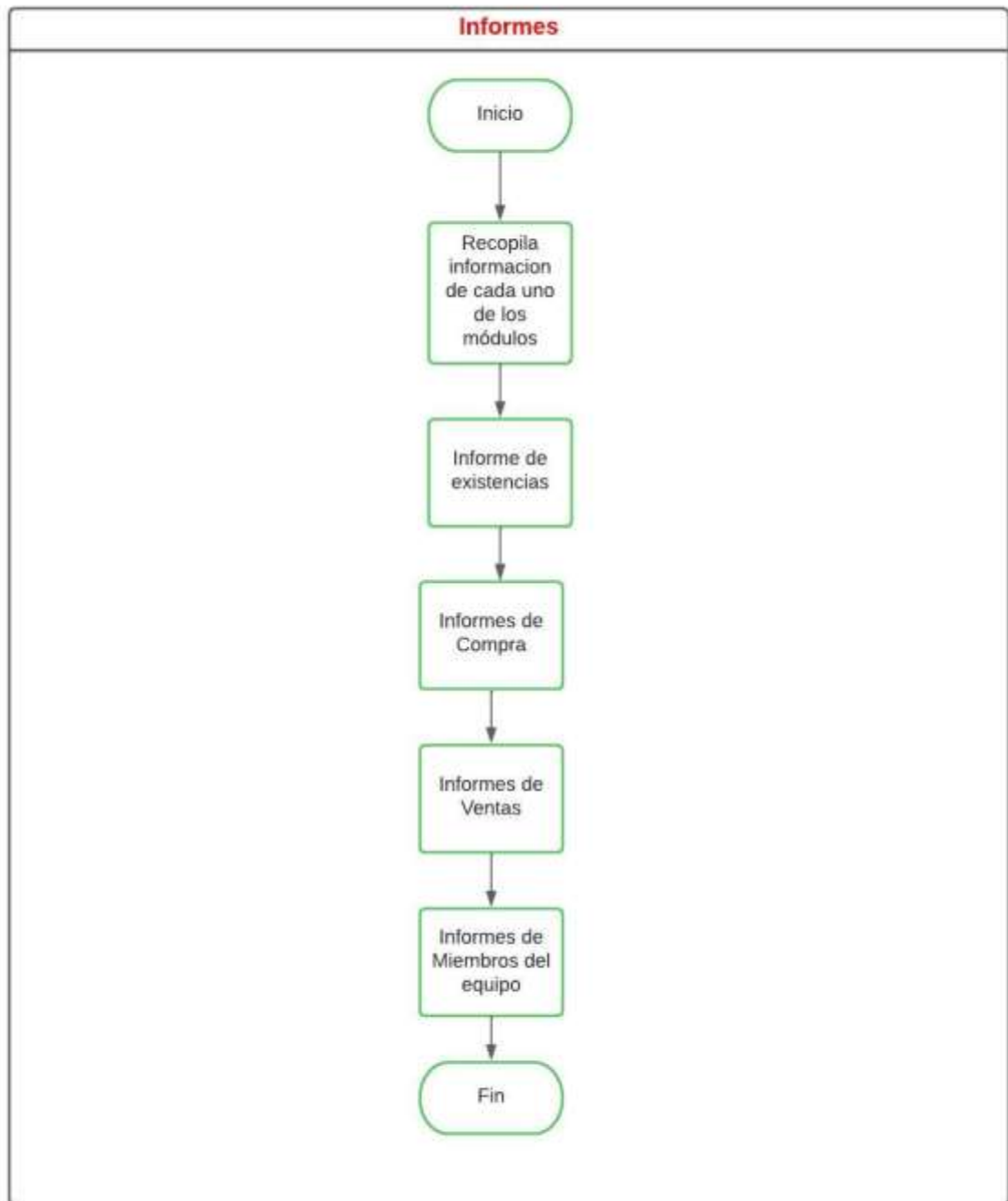


Figura 14 Flujo de procesos del Módulo Informes del Sistema de Inventario. Imagen creada por la autora. Elaborado por Torres Neira Emily.

3.11 Pruebas de Funcionamiento

En este bloque se demuestra la utilidad del sistema implementado y el correcto funcionamiento del mismo.

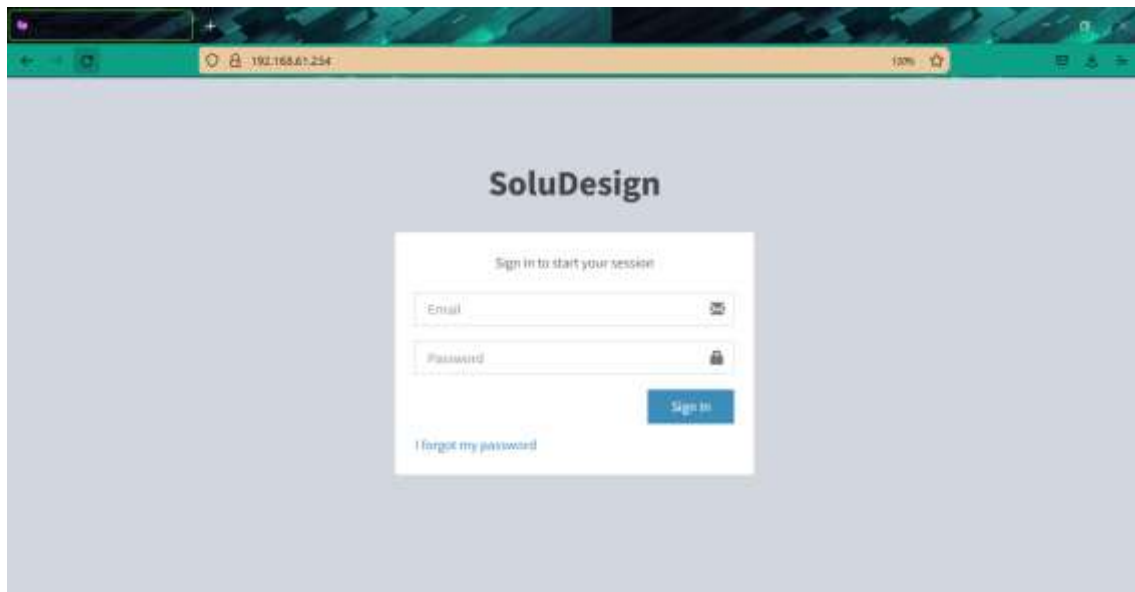


Figura 15 Inicio de sesión del software de Inventario. Elaborado por Torres Neira Emily.

Mediante la visualización de la imagen se presenta el acceso al software, al que se obtiene el ingreso digitando un correo electrónico y una contraseña personalizada, la información de la contraseña se encuentra cifrada dentro de un hasher, los diversos miembros mantienen una tipología, el cual permite tener variantes en el acceso para realizar una experiencia más satisfactoria y eficiente para cada empleado.

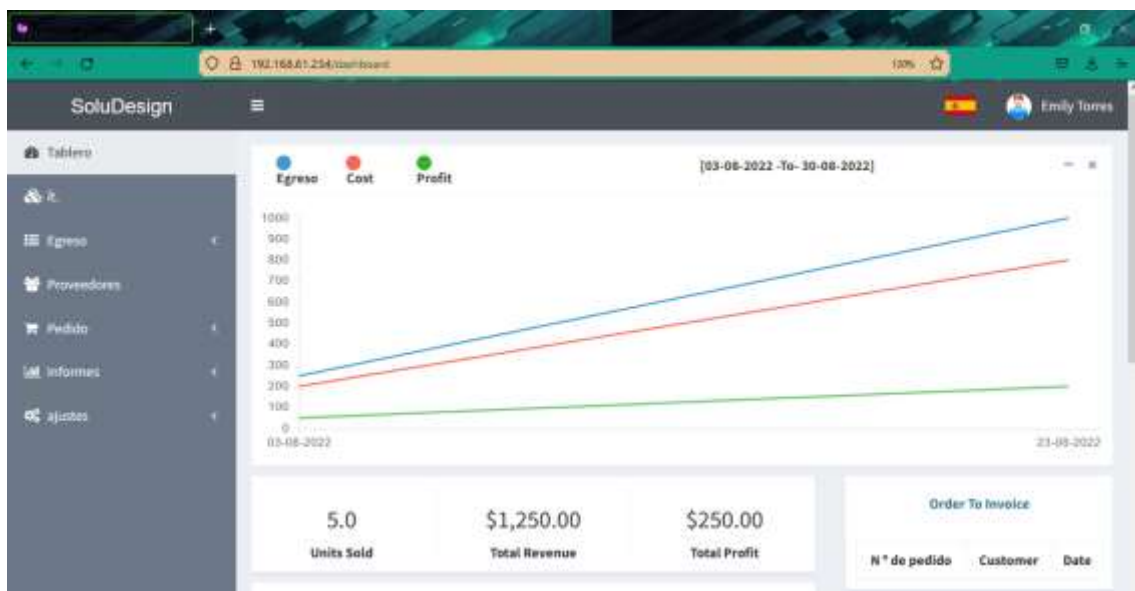


Figura 16 Inicio de Pantalla de usuario administrativo del Sistema de Inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

En la imagen anterior se puede visualizar un listado que contiene las diferentes opciones útiles en el sistema, la opción presentada corresponde al rol administrador, por eso se permite la opción de ajustes e informes que son de relevancia para la compañía.

Ingreso de un nuevo usuario en el sistema, es necesario ingresar el nombre, correo, establecer una contraseña y seleccionar un tipo de rol de usuario que será asignado al nuevo perfil que esta siendo creado.

The screenshot displays the 'SoluDesign' web application. On the left is a dark sidebar with navigation icons and labels: 'Tablero', 'Clientes', 'Iniciar Item', 'Egresos', 'Proveedores', 'Pedidos', 'Informes', and 'Ajustes'. Below these is a 'Detalles de la compañía' section with links for 'Ajustes generales', 'Financiar', 'Plantilla de correo electrónico', and 'Preferencia'. The main content area is titled 'Usuario Crear página'. It contains a form with the following fields: 'Nombre completo' (text input), 'correo electrónico' (text input), 'Contraseña' (password input), 'Teléfono' (text input), and 'Rol del usuario' (a dropdown menu currently showing 'Seleccionar uno'). There are 'Cancelar' and 'Enviar' buttons at the bottom of the form. A 'Manage Company Settings' panel is visible on the left side of the main area, with options for 'Configuración de la empresa', 'Miembros del equipo', 'Rol del usuario', and 'Ubicaciones'. The footer includes 'Copyright © 2022. All rights reserved.' and 'Version 1.7.8'.

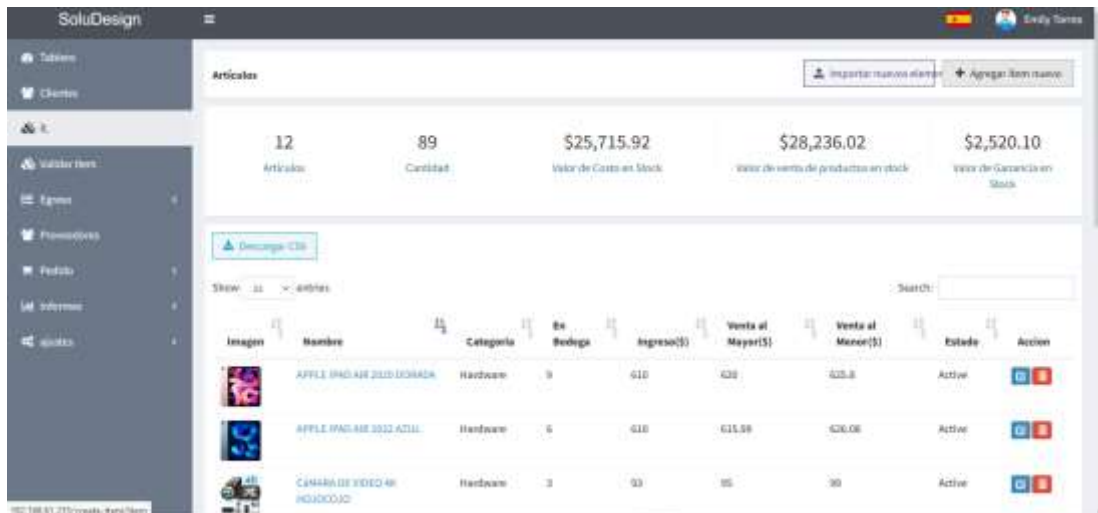
Figura 17 Creación de usuario en el Sistema de Inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

El nivel de acceso al sistema lo determina el usuario administrador quien es la persona que debe crear cada uno de los diferentes roles que serán manejados en el sistema.

This screenshot is similar to the previous one, showing the 'Usuario Crear página' form. However, the 'Rol del usuario' dropdown menu is now open, displaying three options: 'Seleccionar uno', 'orden', and 'bodega'. The rest of the interface, including the sidebar, settings panel, and other form fields, remains the same as in the previous image.

Figura 18 Selección de roles de usuario en el Sistema de Inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

En esta pantalla se presenta un listado de cada uno de los productos ingresados en el sistema, los cuales incrementan su stock una vez que ingresa el pedido del artículo en el inventario.












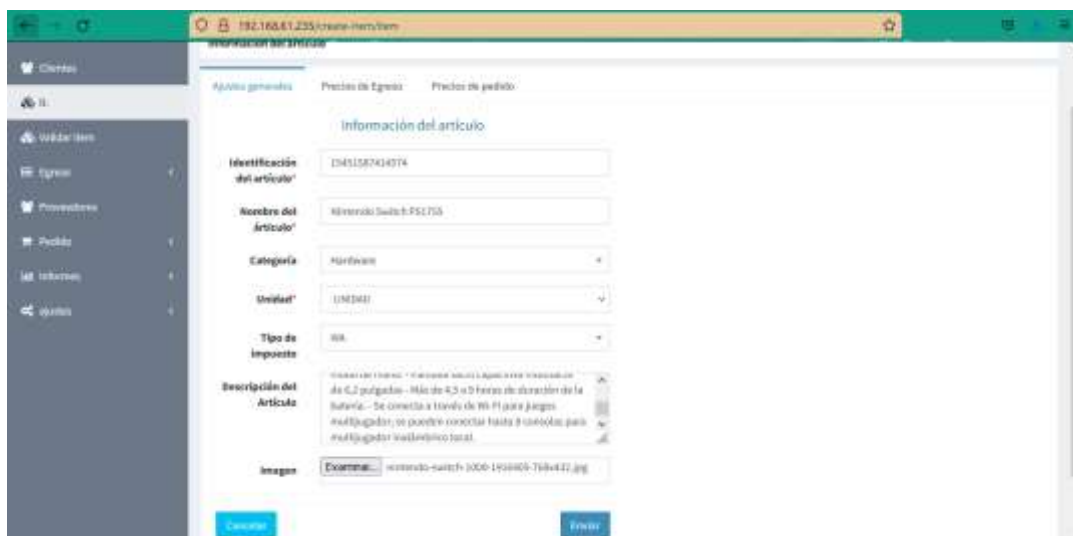
| Imagen | Nombre | Categoría | En Bodega | Ingreso(\$) | Venta al Mayor(\$) | Venta al Menor(\$) | Estado | Acción |
|---|------------------------------|-----------|-----------|-------------|--------------------|--------------------|--------|---|
|  | APPLE IPAD AIR 2020 DONADA | Hardware | 9 | 610 | 620 | 625.8 | Active |   |
|  | APPLE IPAD AIR 2020 ACTIVO | Hardware | 6 | 610 | 615.99 | 626.08 | Active |   |
|  | CÁMARA DE VIDEO 4K HD1000020 | Hardware | 3 | 93 | 95 | 99 | Active |   |

Figura 19 Pantalla que muestra el listado de Ítems del Sistema de Inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

La siguiente imagen muestra cada uno de los campos que deben ser completados al ingresar un producto en el sistema, es necesario que se ingrese un código al producto el cual es único y no se permite realizar la edición una vez que se da clic en el botón enviar. Así como también se establecen las características del producto y se permite agregar una imagen identificativa del artículo.



Información del artículo

Identificación del artículo: 2945158743474

Nombre del artículo: Monitorio Switch-F51755

Categoría: Hardware

Unidad: UNIDAD

Tipo de impuesto: IVA

Descripción del artículo: Monitorio Switch - Máximo de 4.3 o 3 veces de duración de la batería. Se conecta a través de Wi-Fi para cargar. Multifunción: se puede conectar hasta 8 conexiones para multifunción inalámbrica total.


Imagen: 

Figura 20 Pantalla que muestra el ingreso de un artículo al Sistema de Inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

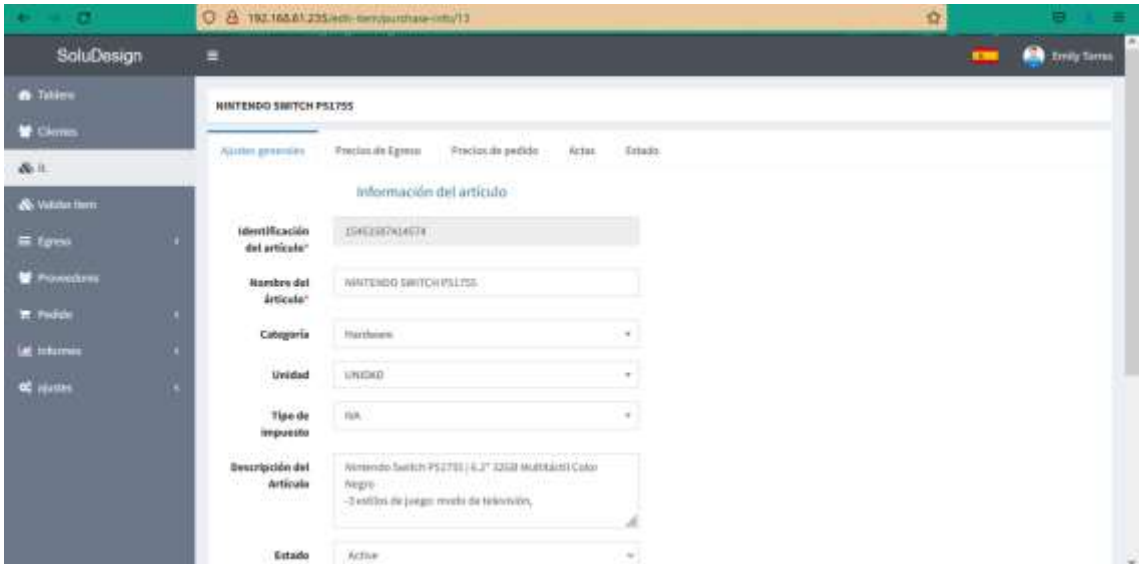


Figura 21 Pantalla que muestra que el código no se modifica. Imagen tomada del software. Elaborado por Torres Neira Emily.

Al guardar la información principal de un artículo se debe establecer el precio al cual será vendido el producto al por mayor y al por menor, ya que esto permite precargar los valores al momento de emitir una orden de egreso.

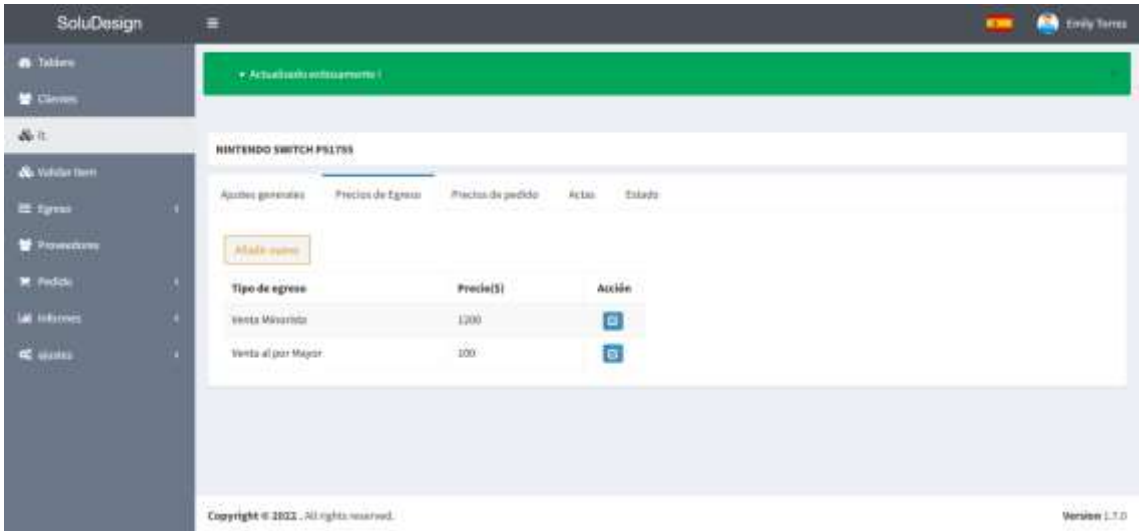


Figura 22 Pantalla donde se ingresa el precio de los productos. Imagen tomada del software. Elaborado por Torres Neira Emily.

En la siguiente pantalla se presenta una vista donde se debe ingresar el precio al cual fue comprado el producto, para establecer un informe que permite determinar las ganancias por cada uno de los artículos adquiridos.

Figura 23 Pantalla donde se ingresa el precio de compra de los productos. Imagen tomada del software. Elaborado por Torres Neira Emily.

Se debe realizar una orden de pedido para realizar la adquisición de artículos, esta se envía a los proveedores para solicitar los artículos.

Figura 24 Pantalla donde se realiza la orden de pedido. Imagen tomada del software. Elaborado por Torres Neira Emily.

Una vez registra la orden de pedido, se presenta en el listado de órdenes, en la cual se debe dar clic en el id identificativo del pedido que ya ha llegado a la bodega de la empresa para realizar el ingreso del producto al stock de la compañía.

| Egreso # | Nombre del proveedor | Cantidad Solicitada | Cantidad Recibida | Total | Fecha de orden | Acciones |
|----------|--------------------------------|---------------------|-------------------|------------|----------------|----------|
| PO-0014 | VLADIMIR AUGUSTO PERALTA MORAL | 1 | 0 | \$1,108.80 | 07-08-2022 | [Iconos] |
| PO-0013 | VLADIMIR AUGUSTO PERALTA MORAL | 3 | 3 | \$342.80 | 02-08-2022 | [Iconos] |
| PO-0005 | ANDREA ESTRELLA HIDALGO | 3 | 3 | \$1,540.00 | 17-08-2022 | [Iconos] |
| PO-0012 | VLADIMIR AUGUSTO PERALTA MORAL | 8 | 8 | \$2,410.20 | 16-08-2022 | [Iconos] |
| PO-0006 | VLADIMIR AUGUSTO PERALTA MORAL | 4 | 3 | \$438.64 | 16-08-2022 | [Iconos] |
| PO-0010 | ANDREA ESTRELLA HIDALGO | 1 | 1 | \$1,685.60 | 15-08-2022 | [Iconos] |
| PO-0008 | VLADIMIR AUGUSTO PERALTA MORAL | 18 | 18 | \$483.84 | 11-08-2022 | [Iconos] |
| PO-0007 | VLADIMIR AUGUSTO PERALTA MORAL | 9 | 9 | \$6,148.80 | 10-08-2022 | [Iconos] |
| PO-0003 | Martin Cabeza | 18 | 18 | \$2,240.00 | 09-08-2022 | [Iconos] |
| PO-0002 | VLADIMIR AUGUSTO PERALTA MORAL | 6 | 6 | \$4,096.20 | 08-08-2022 | [Iconos] |
| PO-0011 | VLADIMIR AUGUSTO PERALTA MORAL | 3 | 3 | \$2,867.20 | 06-08-2022 | [Iconos] |
| PO-0003 | Martin Cabeza | 6 | 4 | \$1,688.00 | 03-08-2022 | [Iconos] |
| PO-0001 | Juanita | 18 | 18 | \$5,360.00 | 03-08-2022 | [Iconos] |

Figura 25 Listado de las ordenes de pedido. Imagen tomada del software. Elaborado por Torres Neira Emily.

Proveedor: VLADIMIR AUGUSTO PERALTA MORAL
Egreso no: PO-0014
Fecha: 07-08-2022
Ubicación: Norte de Guayaquil

| Descripción | Cantidad | Tarifa(\$) | Impuesto(%) | Cantidad |
|------------------------|----------|------------|-------------|------------|
| Nintendo Switch PS1756 | 1 | 990.00 | 12 | 5990.00 |
| Cantidad total | | | | 1 |
| Sub Total | | | | \$990.00 |
| Plus Tax(12%) | | | | \$118.80 |
| Gran total | | | | \$1,108.80 |

Nota:

Figura 26 Registra el ingreso del pedido en el sistema. Imagen tomada del software. Elaborado por Torres Neira Emily.

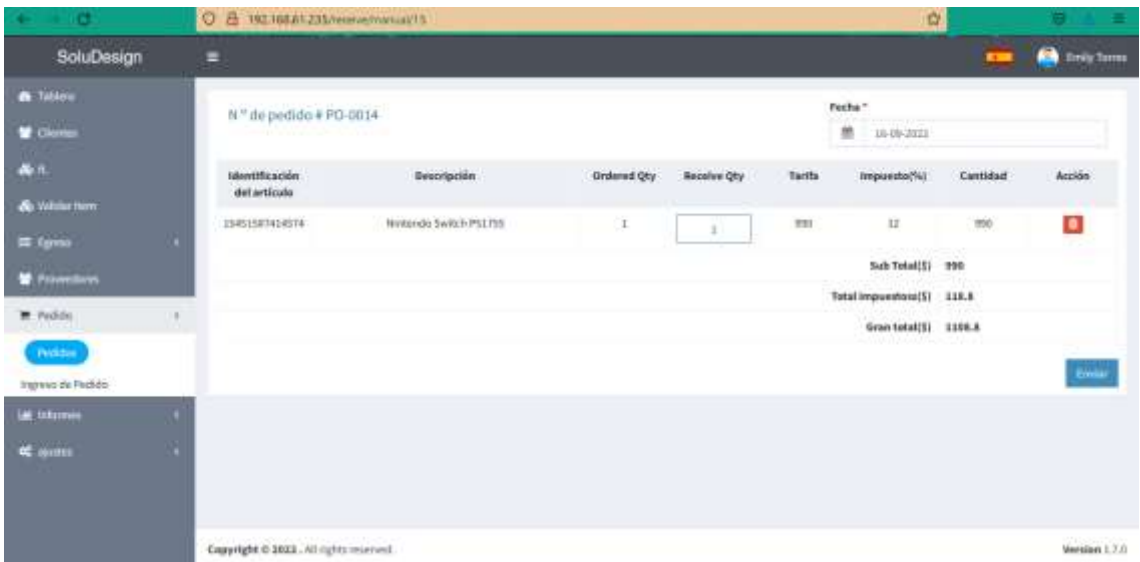


Figura 27 Ingreso de las cantidades de productos que llegan a la bodega. Imagen tomada del software. Elaborado por Torres Neira Emily.

Cuando se termina de realizar el proceso de pedido en el apartado Ítem se visualiza la cantidad de existencias de los productos que fueron adquiridos mediante la solicitud de pedido a proveedores.



Figura 28 Visualización del stock de productos en bodega. Imagen tomada del software. Elaborado por Torres Neira Emily.

Así también para realizar un egreso es necesario registrar una orden de egreso y luego registrar el egreso mediante el pistoleo del código de los productos que son vendidos a los clientes.

SoluDesign

Cliente: Eduardo | Sucursal del Cliente: Eduardo | Lugar: Norte de Guayaquil | Fecha: 16-09-2022

Método de pago: Efectivo | Tipo de Egreso: Venta Minorista | Referencia: 50-0004

Añadir artículo

| Descripción | Cantidad | Tarifa(\$) | Impuesto(%) | Impuesto(\$) | Descuento(\$) | Cantidad(\$) | Acción |
|-----------------------------|----------|------------|---------------|--------------|---------------|--------------|--------|
| CELULAR REDMI 10C 128GB/4GB | 2 | 250 | Impuesto 0(0) | 0 | | 500 | |

Sub Total(\$): 500
Total Impuestos(\$): 0
Gran total(\$): 500

Nota: Descripción...

Figura 29 Registro de una orden de egreso. Imagen tomada del software. Elaborado por Torres Neira Emily.

SoluDesign

Órdenes de egreso

Fecha de orden: 16-09-2022 | Ubicación: Norte de Guayaquil

SoluDesign | Cédula: Eduardo | Egreso: Eduardo

Guayaquil, Guayas | Sucursal: Guayaquil | Guayaquil, Activo

Eduardo, 098593 | Activo, Guayaquil | EC, 23406 | EC, 1233

| Descripción | Cantidad | Tarifa(\$) | Impuesto(%) | Descuento(%) | Cantidad(\$) |
|-----------------------------|----------|------------|-------------|--------------|--------------|
| CELULAR REDMI 10C 128GB/4GB | 2 | 250.00 | 0.00 | 0.00 | 500.00 |

Cantidad total: 2
Sub Total: \$500.00
Gran total: \$500.00
Pagado: \$0.00
Debido: \$500.00

Id * de pedido: 50-0004

Egreso: No se encontró Egreso!

Pagos: No se ha encontrado ningún pago!

Boletín: Ningún envío encontrado!

Figura 30 Registro de una orden de egreso. Imagen tomada del software. Elaborado por Torres Neira Emily.

The screenshot shows the 'Registro de un egreso' (Record an expense) form in the SoluDesign software. The form includes the following fields:

- N° de pedido #**: SO-0004
- Cliente ***: Eduardo
- Secursal del Cliente ***: Eduardo
- Lugar ***: Norte de Guayaquil
- Fecha ***: 16-09-2022
- Método de pago ***: Efectivo
- Plazo de pago ***: Cash
- Referencia ***: INV-0000

The table below shows the items recorded:

| Identificación del artículo | Descripción | Cantidad | Tarifa | Impuesto(%) | Descuento(%) | Cantidad | Acción |
|-----------------------------|-----------------------------|----------|--------|-------------|--------------|----------|--------|
| 001 | CELULAR HONOR 10C 128GB/4GB | 2 | 250 | 0% | 0 | 500 | |

Sub Total: 500
Mas impuestos(0%): 0
Gran total: 500

Figura 31 Registro de un egreso. Imagen tomada del software. Elaborado por Torres Neira Emily.

Al realizar un egreso debe establecer si ya fue pagado en efectivo los productos comprados o especificar el método de pago que realizaran los clientes.

The screenshot shows the 'Facturas' (Invoices) list in the SoluDesign software. The table displays the following data:

| Egreso | N° de pedido | Nombre del cliente | Precio total | Monto de pago | Estado | Fecha de la factura | Acción |
|----------|--------------|--------------------|--------------|---------------|--------|---------------------|--------|
| INV-0001 | SO-0004 | Eduardo | \$500.00 | \$500.00 | Pagado | 16-09-2022 | |
| INV-0002 | SO-0001 | Eduardo | \$1,120.00 | \$1,120.00 | Pagado | 23-09-2022 | |
| INV-0003 | SO-0001 | Eduardo | \$280.00 | \$280.00 | Pagado | 03-09-2022 | |

Figura 32 Listado de cada uno de los egresos. Imagen tomada del software. Elaborado por Torres Neira Emily.

Para realizar la validación de un ítem es necesario que se encuentre una cámara conectada al dispositivo que esta enlazado con el sistema, al ingresar en la opción validar ítems se presenta un mensaje en el que se debe permitir el uso de la cámara.

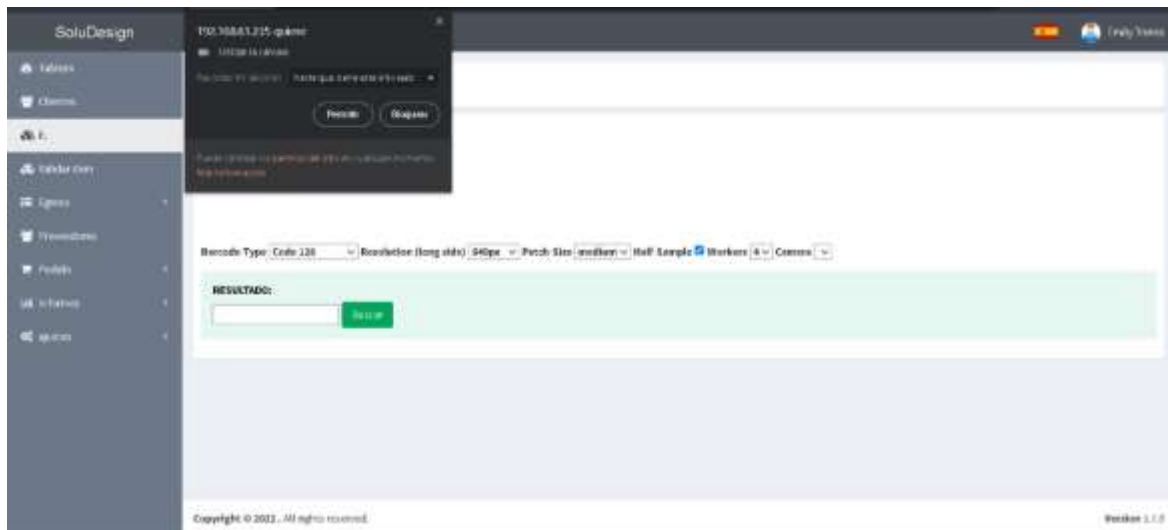


Figura 33 Pantalla que requiere el acceso a la cámara. Imagen tomada del software. Elaborado por Torres Neira Emily.

Una vez que se enlaza el sistema con la cámara se realiza el reconocimiento de los diferentes códigos de barra visualizados.

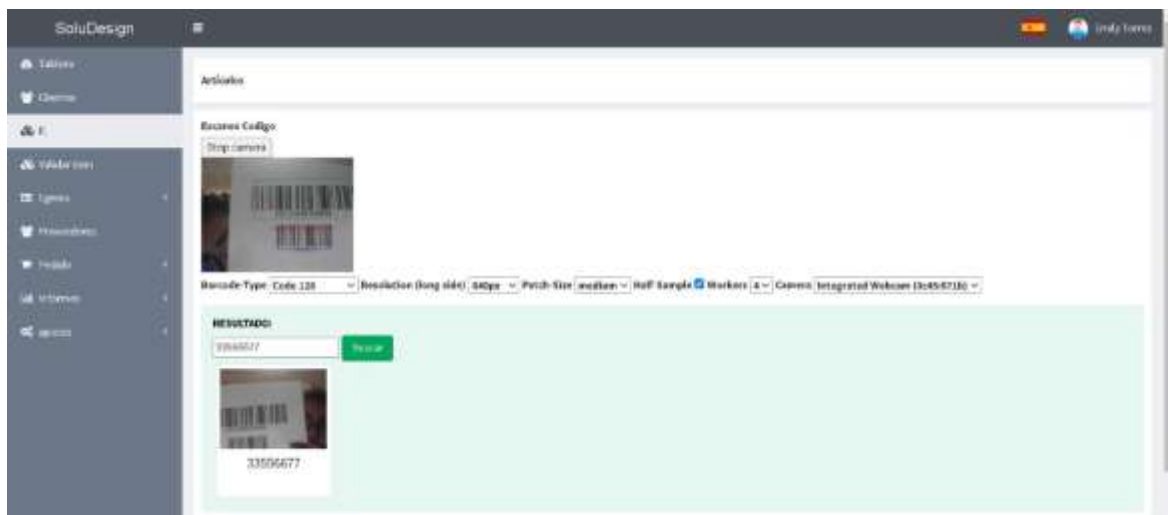


Figura 34 Reconocimiento mediante cámara de códigos de barra. Imagen tomada del software. Elaborado por Torres Neira Emily.

Al buscar un código de barra que no está registrado en el sistema se visualiza el siguiente mensaje en pantalla.

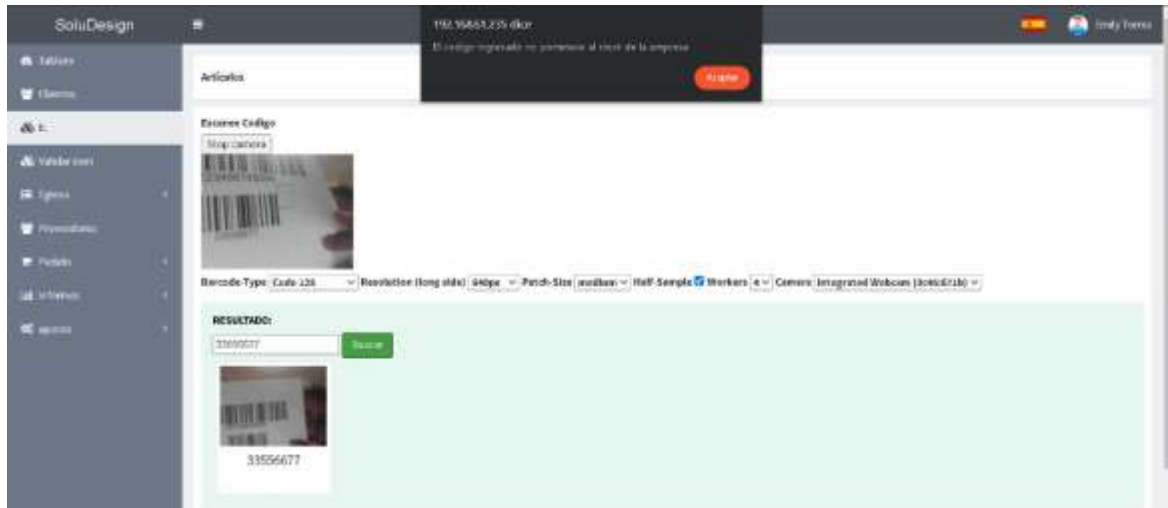


Figura 35 Reconocimiento mediante cámara de códigos de barra de códigos que no existen en el inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

Mientras que, al buscar un código que si ha sido registrado en el sistema se presenta un mensaje que contiene el nombre del producto y el código.

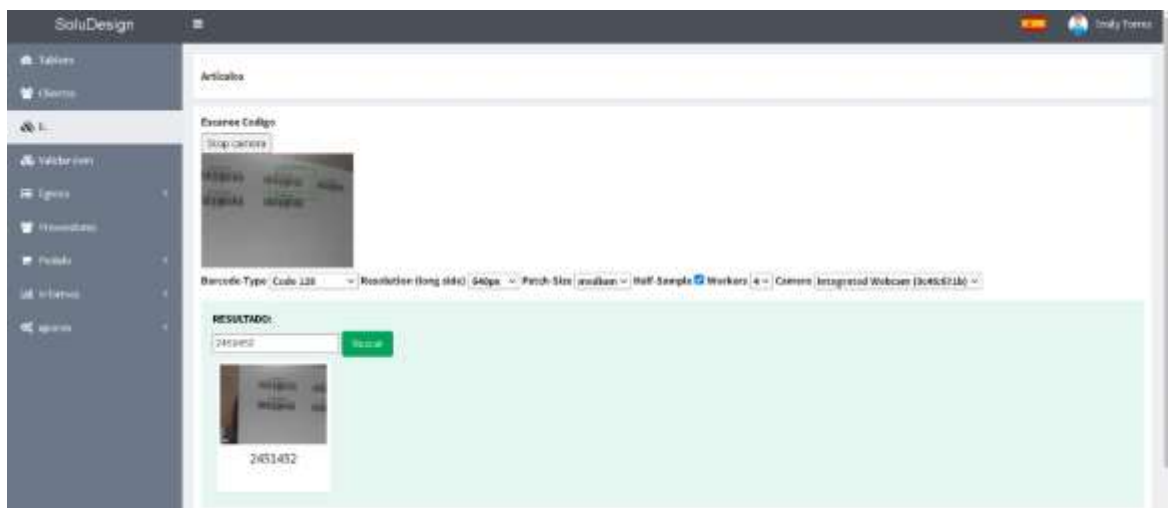


Figura 36 Reconocimiento mediante cámara de códigos de barra. Imagen tomada del software. Elaborado por Torres Neira Emily.

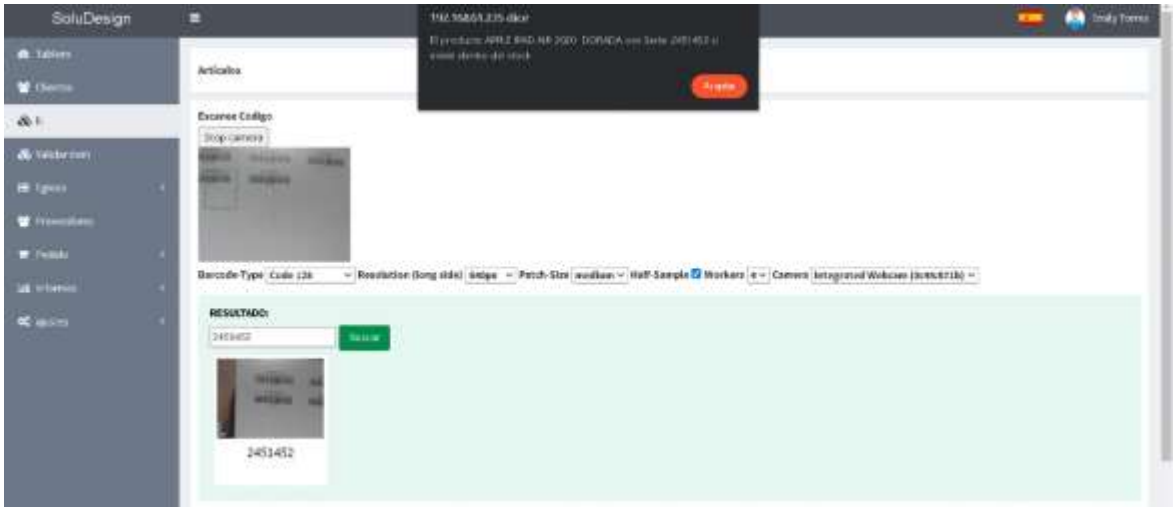


Figura 37 Reconocimiento mediante cámara de códigos de barra de códigos que si existen en el inventario. Imagen tomada del software. Elaborado por Torres Neira Emily.

3.12 Costos de Elaboración

A continuación, se detalla el coste de ejecución del desarrollo web, dejando en constancia que la codificación no se considera con precio ya que este fue implementado dentro de la tesis para cumplir con requerimientos solicitados por la empresa, los precios incluirán hardware, se realizó con código autónomo para la disminución de precios.

Tabla N°5 Costos de elaboración

| Elemento | Costo |
|----------------------------|-----------------|
| Raspberry Pi 4 | \$170.00 |
| Memoria Micro SD de 128 Gb | \$ 25.00 |
| Cargador Micro USB \$ 5.00 | \$ 7.00 |
| Total | \$202.00 |

Elaboración: Torres Neira Emily.

3.13 Análisis de los resultados

El software esta implementado con módulos que permitan realizar el flujo de inventario de forma correcta en la que se cumplan cada una de sus etapas, para realizar el incremento de existencias en el inventario de productos es necesario ingresar un pedido y recibirlo una vez en que llegan a las bodegas de la compañía los artículos son ingresados en el sistema y automáticamente se aumenta la cantidad de stock. Como función adicional se presenta un identificador de código de productos que mediante la conexión de una cámara web o de video enlazada al dispositivo en el que se ha iniciado sesión en el sistema, permite verificar

si la serie ingresada pertenece a algún artículo ingresado en el sistema o nunca ha sido ingresado en el inventario, se presentan mensajes con el nombre del producto en caso de que el ingreso del código sea correcto, mientras que, al tratarse de un código incorrecto se presenta un mensaje que dicho código no ha registrado en el software de la empresa SOLUDESING.

3.14 Conclusiones

Se coordinó una cita con el dueño de la compañía en la cual se estableció la situación inicial de la compañía y como se llevaba a cabo el proceso de inventario de forma manual. Mediante la determinación y conocimiento de la situación actual de la compañía, se estableció cual es el flujo que se debe seguir en el funcionamiento del sistema implementado, cumpliendo de forma satisfactoria cada uno de los requerimientos en listados. Se logro constatar que la herramienta desarrollada para el proceso de inventariado facilita el ingreso de la información y cumple con el flujo de procesos correctamente.

Mediante la integración de un reconocimiento vía cámara se programó una función JavaScript que permite determinar si un código de barra de alguna etiqueta pertenece al sistema o no, al ser la cámara un dispositivo de entrada ingresa la información del código de barra y es leído por el sistema y a continuación se presenta un mensaje que determina si la información ingresada es correcta o no, presentando un mensaje con el nombre y la serie ligada al producto o un mensaje donde indica que el artículo no se encuentra registrado en el sistema.

La raspberry pi 4 siendo un implemento usado en la aplicación de las ciencias aplicadas, también se convierte en un servidor de bajo costo, que representa un valor agregado en comparación en costos-beneficios ya que existen otros servidores que su alquiler representan costos elevados, mientras que, utilizando la raspberry como servidor se logra obtener una amplitud integral hasta 256Gb de almacenamiento interno. Teniendo en cuenta que este servidor soporta hasta 10 usuarios conectados de forma simultánea, es de fácil traslado y transporte en caso de verse en la necesidad de reestablecer el sitio donde se encuentra el servidor.

3.15 Recomendaciones

Realizar capacitaciones para que el personal se encuentre preparado para la correcta utilización del software, ya que con esto se logra obtener una utilización eficaz del uso del

sistema, ya que al no hacerlo se presentarían problemas en el ingreso y manejo de la información.

Para que cada uno de los usuarios administradores posean acceso a la opción de validación de ítems se recomienda que se migre la información a un servidor que contenga una IP publica y adquirir un dominio, ya que actualmente por cuestiones de permisos solo el usuario admin posee acceso a esta opción.

ANEXOS

Anexo 1

Evidencia de la entrevista con el dueño de la empresa



Trabajo previo a la obtención del Título Entrevista al Dueño de la empresa objeto de estudio

TEMA: SISTEMA DE INVENTARIO PARA LA EMPRESA SOLUDESIGN MULTIPLATAFORMA.

Fecha de la entrevista realizada: 17 - Agosto - 2022

Nombre del dueño de la empresa: Andres Abad Nieto

Preguntas:

1. ¿Cuáles son las actividades que desempeña la empresa?

La empresa Soludesign se dedica a la compra y venta de artículos electrónicos como celulares, tablets, computadores, video cámara, etc.

2. ¿Cuál es la cantidad de empleados con los que cuenta la empresa y a qué se dedica uno de ellos?

2 empleados que se dedican a la compra de artículos, 1 persona encargada de la contabilidad, 1 supervisor que se encarga del registro de artículos, 2 bodegaos que etiquetan y clasifican el producto, el gerente general, 2 vendedores que realizan la entrega del producto.

3. ¿Existe un grupo de fases que se deben cumplir en el registro del inventario que maneja actualmente la empresa (Especifíquelo)?

- ① Realizan un pedido de la mercadería que se necesita
- ② Verificación del que el pedido solicitado llegue completo
- ③ Registro los productos que se encuentran en stock
- ④ Tienen el registro de los diferentes productos que han sido vendidos al público.
- ⑤ Identificación de los productos con los que se cuenta.

Figura Preguntas y Respuestas referentes a la entrevista realizada al dueño de la Empresa. Elaborado por Torres Neira Emily.

4. ¿Cuáles considera que son los riesgos que se presentan en la forma de llevar a cabo el inventario de manera manual como lo realiza actualmente la empresa?

Pérdida de tiempo.
Se puede extrañar la información.
No se tiene un control de la mercadería con la que se cuenta.

5. ¿Cómo considera usted que deba funcionar el sistema? ¿Le gustaría agregar algún otro requerimiento que sea necesario y encuentre viable para el correcto desempeño del sistema?

Se cumpla con cada una de las fases que maneja
actualmente de forma manual y visualizar un informe
de los procesos que realiza la empresa.

Dueño de la Empresa Soludesign

Estudiante Entrevistador

Figura Preguntas y Respuestas referentes a la entrevista realizada al dueño de la Empresa. Elaborado por Torres Neira Emily.

Anexo II

Configuración de .env

```
APP_ENV=local

APP_DEBUG=true

APP_KEY=base64:rc6uhGqFugSskgVZeVagqu85sLtjpUbO3B0BUNI41AA=

APP_URL=http://localhost


DB_CONNECTION=mysql

DB_HOST=localhost

DB_PORT=3306

DB_DATABASE=emily|

DB_USERNAME=root

DB_PASSWORD=


CACHE_DRIVER=file
```

Figura Configuración del .env. Elaborado por Torres Neira Emily.

Anexo III

Programación de CompanyController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests;
use DB;
use Session;
use App\Http\Start\Helpers;

class CompanyController extends Controller
{
    public function __construct(){
        /**
         * Set the database connection. reference app\helper.php
         */
        //selectDatabase();
    }
    /**
     * Display a listing of the Company.
     *
     * @return Company List page view
     */
    public function index()
    {
        $data['menu'] = 'setting';
        $data['sub_menu'] = 'general';
        $data['list_menu'] = 'company';
        $data['companyData'] = DB::table('company')->get();

        return view('admin.company.company_list', $data);
    }

    public function setDatabase($host='localhost', $username='root',
$password='', $database, $user_password,$email,$name)
    {
        \Config::set('database.connections.tenant.host', $host);
        \Config::set('database.connections.tenant.username', $username);
        \Config::set('database.connections.tenant.password', $password);
        \Config::set('database.connections.tenant.database', $database);

        \Config::set('database.default', 'tenant');
        $db = DB::reconnect('tenant');
```



```

emptyDatabase();

\Artisan::call('migrate');
\Artisan::call('db:seed');

$userData['real_name'] = $name;
$userData['email'] = $email;
$userData['password'] = $user_password;
$userData['created_at'] = date('Y-m-d H:i:s');
DB::table('users')->insert($userData);

}

/**
 * Show the form for creating a new Company.
 *
 * @return Company create page view
 */
public function create()
{
    $data['menu'] = 'setting';
    $data['sub_menu'] = 'general';
    $data['list_menu'] = 'company';

    return view('admin.company.company_add', $data);
}

/**
 * Store a newly created Company in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return redirect Company List page view
 */
public function store(Request $request)
{
    $this->validate($request, [
        'name' => 'required|min:3',
        'host' => 'required',
        'db_user' => 'required',
        'real_name' => 'required',
        'email' => 'required',
        'db_name' => 'required',
        'user_pass' => 'required',
    ]);

    $data['name'] = $request->name;
    $data['host'] = $request->host;
    $data['db_user'] = $request->db_user;

```

```

        $data['db_password'] = $request->db_password;
        $data['db_name'] = $request->db_name;
        $data['default'] = 'No';
        $user_pass = \Hash::make($request->user_pass);
        $data['created_at'] = date('Y-m-d H:i:s');

        $conn = dbConnect($request->host, $request->db_user,$request->db_password,$request->db_name);

        if ($conn == false) {
            return back()->withInput()->withErrors(['email' => trans('message.error.db_error')]);
            exit;
        }

        $database = \DB::table('company')->select('db_name')->where('db_name',$request->db_name)->first();
        if (!empty($database)) {
            return back()->withInput()->withErrors(['email' => trans('message.error.company_error')]);
            exit;
        }

        $id = \DB::table('company')->insertGetId($data);
        $status = $this->setDatabase($request->host, $request->db_user,$request->db_password,$request->db_name,$user_pass,$request->email,$request->real_name);

        if (!empty($id)) {
            \Session::flash('success',trans('message.success.save_success'));
        };
        return redirect()->intended('company');
    } else {
        return back()->withInput()->withErrors(['email' => "Invalid Request !"]);
    }
}

/**
 * Display the specified Company.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)

```

```

{
    //
}

/**
 * Show the form for editing the specified Company.
 *
 * @param int $id
 * @return Company edit page view
 */
public function edit($id)
{
    $data['menu'] = 'setting';
    $data['sub_menu'] = 'general';
    $data['list_menu'] = 'company';
    $data['companyData'] = \DB::table('company')->where('company_id',
$idid)->first();

    return view('admin.company.company_edit', $data);
}

/**
 * Update the specified Company in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return redirect Company List page view
 */
public function update(Request $request, $id)
{
    $this->validate($request, [
        'name' => 'required|min:3',
    ]);

    $data['name'] = $request->name;
    $data['default'] = $request->default;
    $data['updated_at'] = date('Y-m-d H:i:s');
    if ($data['default'] == 'Yes') {

        $c_id = \DB::table('company')->select('company_id')-
>where('default', '=', 'Yes')->first();

        \DB::table('company')->where('company_id', $c_id->company_id)-
>update(['default' => 'No']);
    }

    \DB::table('company')->where('company_id', $id)->update($data);
}

```

```

        \Session::flash('success',
trans('message.success.update_success'));
        return redirect()->intended('company');
    }

    /**
     * Remove the specified Company from storage.
     *
     * @param int $id
     * @return redirect Company List page view
     */
    public function destroy($id)
    {
        if (isset($id)) {
            $record = \DB::table('company')->where('company_id', $id)-
>first();
            if ($record) {
                \DB::table('company')->where('company_id', '=', $id)-
>delete();

                \Session::flash('success',trans('message.success.delete_suc
cess'));
                return redirect()->intended('company');
            }
        }
    }
}

```

Programación de UserController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Model\Purchase;
use App\Model\Orders;
use App\Model\Sales;
use App\Model\Payment;
use App\Model\Shipment;
use Auth;
use DB;
use Session;
use App\Http\Start\Helpers;

```

```

class UserController extends Controller
{
    public function __construct() {
        /**
         * Set the database connection. reference app\helper.php
         */
        //selectDatabase();
    }
    /**
     * Display a listing of the Users.
     *
     * @return User List page view
     */
    public function index()
    {
        $id = Auth::user()->id;
        $data['menu'] = 'setting';
        $data['sub_menu'] = 'company';
        $data['list_menu'] = 'users';

        $data['userData'] = DB::table('users')
            -
            >leftjoin('roles','roles.id','=','users.role_id')
            ->select('users.*','roles.name as role')
            ->orderBy('id', 'desc')
            ->get();

        return view('admin.user.user_list', $data);
    }

    /**
     * Show the form for creating a new User.
     *
     * @return User create page view
     */
    public function create()
    {
        $data['menu'] = 'setting';
        $data['sub_menu'] = 'company';
        $data['list_menu'] = 'users';

        $data['roleData'] = DB::table('roles')->get();
        return view('admin.user.user_add', $data);
    }

    /**
     * Store a newly created User in storage.
     */
}

```

```

* @param \Illuminate\Http\Request $request
* @return User List page view
*/
public function store(Request $request)
{
    $this->validate($request, [
        'email' => 'required|unique:users,email',
        'real_name' => 'required',
        'password' => 'required',
        'role_id' => 'required',
    ]);

    $data['email'] = $request->email;
    $data['user_id'] = $request->real_name;
    $data['real_name'] = $request->real_name;
    $data['password'] = \Hash::make($request->password);
    $data['role_id'] = $request->role_id;
    $data['phone'] = $request->phone;
    $data['created_at'] = date('Y-m-d H:i:s');

    $id = DB::table('users')->insertGetId($data);

    DB::table('role_user')->insertGetId(['user_id'=>$id,'role_id'=>$request->role_id]);

    if (!empty($id)) {
        \Session::flash('success',trans('message.success.save_success'))
    );
        return redirect()->intended('users');
    }
}

/**
 * Display the specified User.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified User.
 *
 * @param int $id
 * @return User edit page view

```

```

    */
    public function edit($id)
    {
        $data['menu'] = 'setting';
        $data['sub_menu'] = 'company';
        $data['list_menu'] = 'users';

        $data['roleData'] = DB::table('roles')->get();
        $data['userData'] = DB::table('users')->where('id', '=', $id)-
>first();

        return view('admin.user.editProfile', $data);
    }

    /**
     * Update the specified User in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return User List page view
     */
    public function update(Request $request, $id)
    {
        $this->validate($request, [
            'real_name' => 'required',
            'role_id' => 'required',
        ]);

        $data['user_id'] = $request->real_name;
        $data['real_name'] = $request->real_name;
        $data['role_id'] = $request->role_id;
        $data['phone'] = $request->phone;
        $data['updated_at'] = date('Y-m-d H:i:s');

        $pic = $request->file('picture');

        if (isset($pic)) {
            $upload = 'public/uploads/userPic';

            $pic1 = $request->pic;
            if ($pic1 != NULL) {
                $dir = public_path("uploads/userPic/$pic1");
                if (file_exists($dir)) {
                    unlink($dir);
                }
            }
        }
    }

```

```

        $filename = $pic->getClientOriginalName();
        $pic = $pic->move($upload, $filename);
        $data['picture'] = $filename;
    }

    DB::table('users')->where('id', $id)->update($data);
    DB::table('role_user')->where('user_id', $id)-
>update(['role_id'=>$request->role_id]);

    \Session::flash('success',trans('message.success.update_success'));
    return redirect()->intended("edit-user/$id");
}

/**
 * Remove the specified User from storage.
 *
 * @param int $id
 * @return User List page view
 */
public function destroy($id)
{
    if(isset($id)) {
        $record = \DB::table('users')->where('id', $id)->first();
        if($record) {

            \DB::table('users')->where('id', '=', $id)->delete();

            \Session::flash('success',trans('message.success.delete_suc
cess'));

            return redirect()->intended('users');
        }
    }
}

/**
 * Validate email address while creating a new User.
 *
 * @return true or false
 */
public function validEmail()
{
    $email = $_POST['email'];
    $v = DB::table('users')->where('email', '=', $email)->first();
    if (!empty($v)) {
        echo 1;
    } else {
        echo 0;
    }
}

```



```

}

/**
 * Show and manage user profile CRUD operation
 *
 * @return User profile page view
 */
public function profile()
{
    $id = Auth::user()->id;
    $data['menu'] = 'setting';
    $data['sub_menu'] = 'profile';
    $data['header'] = 'profile';
    $data['breadcrumb'] = 'profile';
    $data['userData'] = DB::table('users')->where('id', '=', $id)-
>first();
    $data['roleData'] = DB::table('roles')->get();

    return view('admin.user.editProfile', $data);
}

/**
 * show user change password operation
 *
 * @return change password page view
 */
public function changePassword($id)
{
    $data['menu'] = 'NULL';
    $data['header'] = 'profile';
    $data['breadcrumb'] = 'change/password';
    $data['userData'] = DB::table('users')->where('id', '=', $id)-
>first();

    return view('admin.user.change_password', $data);
}

/**
 * Change user password operation perform
 *
 * @return change password page view
 */

public function updatePassword(Request $request, $id)
{
    $this->validate($request, [
        'old_pass' => 'required',
        'new_pass' => 'required',
    ]);
}

```

```

]);

$v = DB::table('users')->where('id', '=', $id)->first();

$data['password'] = \Hash::make($request->new_pass);
$data['updated_at'] = date('Y-m-d H:i:s');

if (\Hash::check($request->old_pass, $v->password)) {
    DB::table('users')->where('id', $id)->update($data);
    \Session::flash('success', 'Password Update successfully !');
    return redirect()->intended("change-password/$id");
} else {
    return back()->withInput()->withErrors(['email' => "Old
Password is Wrong !"]);
}

}

public function userPurchaseOrderList($id){

    $data['menu'] = 'report';
    $data['sub_menu'] = 'member-report';
    $data['po_status'] = 'active';
    $data['user_id'] = $id;
    $data['supplier'] = $supplier = 'all';
    $data['location'] = $location = 'all';

    $fromDate = DB::table('purch_orders')->select('ord_date')->
->where('person_id', $id)->orderBy('ord_date', 'asc')->first();
    $toDate = DB::table('purch_orders')->select('ord_date')->
->where('person_id', $id)->orderBy('ord_date', 'desc')->first();

    $data['from'] = $from = isset($fromDate->ord_date) ?
formatDate($fromDate->ord_date): formatDate(date("d-m-Y", strtotime("-1
months"))));
    $data['to'] = $to = isset($toDate->ord_date) ? formatDate($toDate->
ord_date) : formatDate(date('d-m-Y'));

    $data['user'] = DB::table('users')->where('id', $id)->first();

    $data['supplierList'] = DB::table('suppliers')
->select('supplier_id', 'supp_name')
->get();

    $data['locationList'] = DB::table('location')
->select('loc_code', 'location_name')

```

```

        ->get();
        $data['purchData'] = (new Purchase)-
>getAllPurchOrderById($from, $to, $supplier, $location, $id);

        if(isset($_GET['btn'])){

            $data['supplier'] = $supplier = $_GET['supplier'];
            $data['location'] = $location = $_GET['location'];
            $data['from'] = $from = $_GET['from'];
            $data['to'] = $to = $_GET['to'];

            $data['purchData'] = (new Purchase)-
>getAllPurchOrderById($from, $to, $supplier, $location, $id);
        }

        return view('admin.userdetail.po-orders', $data);
    }

    public function userSalesOrderList($id){

        $data['so_status'] = 'active';
        $data['menu'] = 'report';
        $data['sub_menu'] = 'member-report';

        $data['customer'] = $customer = 'all';
        $data['location'] = $location = 'all';
        $data['user_id'] = $id;

        $fromDate = DB::table('sales_orders')->select('ord_date')-
>where('person_id',$id)->orderBy('ord_date','asc')->first();
        $data['from'] = $from = isset($fromDate->ord_date) ?
formatDate($fromDate->ord_date): formatDate(date("d-m-Y", strtotime("-1
months"))));
        $data['to'] = $to = formatDate(date('d-m-Y'));

        $data['user'] = DB::table('users')->where('id',$id)->first();

        $data['customerList'] = DB::table('debtors_master')
            ->select('debtor_no','name')
            ->get();

        $data['locationList'] = DB::table('location')
            ->select('loc_code','location_name')
            ->get();

        $data['salesData'] = (new Orders)->getAllSalseOrderById($from,
$to, $customer, $location, $id);
    }

```

```

        if(isset($_GET['btn'])) {

            $data['customer'] = $customer = $_GET['customer'];
            $data['location'] = $location = $_GET['location'];
            $data['from'] = $from = $_GET['from'];
            $data['to'] = $to = $_GET['to'];

            $data['salesData'] = (new Orders)-
>getAllSalseOrderByUserId($from, $to, $customer, $location, $id);
        }

        return view('admin.userdetail.so-orders', $data);
    }

    public function userSalesInvoiceList($id){

        $data['invoice'] = 'active';
        $data['menu'] = 'report';
        $data['sub_menu'] = 'member-report';
        $data['user_id'] = $id;

        $data['customer'] = $customer = 'all';
        $data['location'] = $location = 'all';

        $data['user'] = DB::table('users')->where('id',$id)->first();

        $fromDate = DB::table('sales_orders')->select('ord_date')-
>where('person_id',$id)->orderBy('ord_date','asc')->first();
        $data['from'] = $from = isset($fromDate->ord_date) ?
formatDate($fromDate->ord_date): formatDate(date("d-m-Y", strtotime("-1
months"))));
        $data['to'] = $to = formatDate(date('d-m-Y'));

        $data['customerList'] = DB::table('debtors_master')
            ->select('debtor_no','name')
            ->get();

        $data['locationList'] = DB::table('location')
            ->select('loc_code','location_name')
            ->get();

        $data['salesData'] = (new Sales)->getAllSalseInvoiceByUserId($from,
$to, $customer, $location, $id);

        if(isset($_GET['btn'])) {

            $data['customer'] = $customer = $_GET['customer'];

```

```

        $data['location'] = $location = $_GET['location'];
        $data['from'] = $from = $_GET['from'];
        $data['to'] = $to = $_GET['to'];

        $data['salesData'] = (new Sales)-
>getAllSalseInvoiceByUserId($from, $to, $customer, $location, $id);
    }

    return view('admin.userdetail.invoices', $data);

}

public function userTransferList($id){

    $data['transfer'] = 'active';
    $data['menu'] = 'report';
    $data['sub_menu'] = 'member-report';
    $data['user_id'] = $id;
    $data['source'] = $source = 'all';
    $data['destination'] = $destination = 'all';

    $data['user'] = DB::table('users')->where('id',$id)->first();

    $fromDate = DB::table('stock_transfer')->select('transfer_date')-
>where('person_id',$id)->orderBy('transfer_date','asc')->first();

    $data['from'] = $from = isset($fromDate->transfer_date) ?
formatDate($fromDate->transfer_date): formatDate(date("d-m-Y", strtotime("-
1 months"))));
    $data['to'] = $to = formatDate(date('d-m-Y'));

    $data['locationList'] = DB::table('location')
        ->select('loc_code','location_name')
        ->get();

    $data['list'] = (new Shipment)->getAllStockTransferByUserId($from,
$to, $source, $destination, $id);

    //d($data,1);
    if(isset($_GET['btn'])){

        $data['source'] = $source = $_GET['source'];
        $data['destination'] = $destination = $_GET['destination'];
        $data['from'] = $from = $_GET['from'];
        $data['to'] = $to = $_GET['to'];
        $data['list'] = (new Shipment)-
>getAllStockTransferByUserId($from, $to, $source, $destination, $id);
    }
}

```

```

    }

    return view('admin.userdetail.transfer', $data);

}

public function userPaymentList($id){

    $data['payment'] = 'active';
    $data['menu'] = 'report';
    $data['sub_menu'] = 'member-report';
    $data['user_id'] = $id;
    $data['customer'] = $customer = 'all';
    $data['user'] = DB::table('users')->where('id',$id)->first();

    $fromDate = DB::table('payment_history')->select('payment_date')-
>where('person_id',$id)->orderBy('payment_date','asc')->first();

    $data['from'] = $from = isset($fromDate->payment_date) ?
formatDate($fromDate->payment_date): formatDate(date("d-m-Y", strtotime("-1
months"))));

    $data['to'] = $to = formatDate(date('d-m-Y'));

    $data['customerList'] = DB::table('debtors_master')
        ->select('debtor_no','name')
        ->get();

    $data['paymentList'] = (new Payment)->getAllPaymentByUserId($from,
$to, $customer, $id);

    if(isset($_GET['btn'])){

        $data['customer'] = $customer = $_GET['customer'];
        $data['from'] = $from = $_GET['from'];
        $data['to'] = $to = $_GET['to'];

        $data['paymentList'] = (new Payment)-
>getAllPaymentByUserId($from, $to, $customer, $id);
    }

    return view('admin.userdetail.payments', $data);

}

/**
 * Display a listing of the Users.
 */

```

```

    * @return User List page view
    */
    public function memberReport()
    {
        $id = Auth::user()->id;
        $data['menu'] = 'report';
        $data['sub_menu'] = 'member-report';
        $data['userData'] = DB::table('users')
            -
        >leftjoin('roles','roles.id','=','users.role_id')
            ->select('users.*','roles.name as role')
            ->orderBy('id', 'desc')
            ->get();

        return view('admin.user.team_member', $data);
    }
}

```

Programación de ItemController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Model\Item;
use Image;
use Session;
use Excel;
use Input;
use DB;
use Validator;
use App\Model\Report;
use App\Http\Start\Helpers;
use Datatables;
use App\DataTables\ItemDataTable;
use PhpParser\Node\Expr\FuncCall;

class ItemController extends Controller
{
    public function __construct(Report $report)
    {
        /**
         * Set the database connection. reference app\helper.php

```

```

        */
        //selectDatabase();
        $this->report = $report;
    }
    /**
     * Display a listing of the Item.
     *
     * @return Item list page view
     */
    public function index(ItemDataTable $dataTable)
    {

        $data['menu'] = 'item';
        $data['header'] = 'item';
        $qtyOnHand = 0;
        $costValueQtyOnHand = 0;
        $retailValueOnHand = 0;
        $profitValueOnHand = 0;
        $mac = 0;

        $data['itemData'] = DB::table('item_code')->where(['deleted_status'
=> 0])->count();
        $data['itemQuantity'] = DB::table('stock_moves')-
>select(DB::raw('SUM(qty) as total_item'))->first();
        $itemList = $this->report->getInventoryStockOnHand('all', 'all');
        //d($itemList,1);
        foreach ($itemList as $key => $item) {

            $qtyOnHand += $item->available_qty;

            if ($item->received_qty != 0) {
                $mac = $item->cost_amount / $item->received_qty;
            }

            $costValueQtyOnHand += $item->available_qty * $mac;
            $retailValueOnHand += $item->available_qty * $item-
>retail_price;
            $profitValueOnHand += (($item->available_qty * $item-
>retail_price) - ($item->available_qty * $mac));
        }
        $data['qtyOnHand'] = $qtyOnHand;
        $data['costValueQtyOnHand'] = $costValueQtyOnHand;
        $data['retailValueOnHand'] = $retailValueOnHand;
        $data['profitValueOnHand'] = $profitValueOnHand;

        //return view('admin.item.item_list', $data);
        return $dataTable->render('admin.item.view', $data);
    }

```



```

public function validate_item(ItemDataTable $dataTable)
{
    $data['menu'] = 'item';
    $data['header'] = 'item';
    $qtyOnHand = 0;
    $costValueQtyOnHand = 0;
    $retailValueOnHand = 0;
    $profitValueOnHand = 0;
    $mac = 0;

    $data['itemData'] = DB::table('item_code')->where(['deleted_status'
=> 0])->count();
    $data['itemQuantity'] = DB::table('stock_moves')-
>select(DB::raw('SUM(qty) as total_item'))->first();
    $itemList = $this->report->getInventoryStockOnHand('all', 'all');
    //d($itemList,1);
    foreach ($itemList as $key => $item) {

        $qtyOnHand += $item->available_qty;

        if ($item->received_qty != 0) {
            $mac = $item->cost_amount / $item->received_qty;
        }

        $costValueQtyOnHand += $item->available_qty * $mac;
        $retailValueOnHand += $item->available_qty * $item-
>retail_price;
        $profitValueOnHand += (($item->available_qty * $item-
>retail_price) - ($item->available_qty * $mac));
    }
    $data['qtyOnHand'] = $qtyOnHand;
    $data['costValueQtyOnHand'] = $costValueQtyOnHand;
    $data['retailValueOnHand'] = $retailValueOnHand;
    $data['profitValueOnHand'] = $profitValueOnHand;

    //return view('admin.item.item_list', $data);
    return $dataTable->render('admin.item.validate', $data);
}

/**
 * Show the form for creating a new Item.
 *
 * @return Item create page view
 */
public function create($tab)
{

```

```

$item_stock_id = Session::get('stock_id');
$data['menu'] = 'item';

$unit = DB::table('item_unit')->get();
$unit_name = array();
foreach ($unit as $value) {
    $unit_name[$value->id] = $value->name;
}

$data['unit_name'] = $unit_name;

$data['locData']      = DB::table('location')->get();
$data['taxTypes']     = DB::table('item_tax_types')->get();
$data['saleTypes']    = DB::table('sales_types')->get();
$data['categoryData'] = DB::table('stock_category')->get();
$data['unitData']     = DB::table('item_unit')->get();
$data['suppliers']    = DB::table('suppliers')->get();

if (!empty($item_stock_id)) {
    $data['stock_id'] = strtoupper($item_stock_id);
}

$data['tab'] = $tab;

return view('admin.item.item_add', $data);
}

/**
 * Store a newly created Item in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return redirection Item list page view
 */
public function store(Request $request)
{
    $this->validate($request, [
        'stock_id' => 'required|unique:item_code,stock_id',
        'description' => 'required',
        'tax_type_id' => 'required',
        'units' => 'required',
        'item_image' => 'mimes:jpeg,bmp,png'
    ]);

    $data['stock_id'] = strtoupper($request->stock_id);
    $data['description'] = strtoupper($request->description);
    $data['category_id'] = $request->category_id;
    $data['created_at'] = date('Y-m-d H:i:s');

```

```

    $pic = $request->file('item_image');

    if (isset($pic)) {
        $destinationPath = public_path('/uploads/itemPic/');
        $filename = $pic->getClientOriginalName();
        $img = Image::make($request->file('item_image')->getRealPath());

        $img->resize(400, 400, function ($constraint) {
            $constraint->aspectRatio();
        })->save($destinationPath . '/' . $filename);

        $data['item_image'] = $filename;
    }

    $id = DB::table('item_code')->insertGetId($data);

    if (!empty($id)) {
        Session::put('stock_id', strtoupper($request->stock_id));

        $data2['stock_id'] = strtoupper($request->stock_id);
        $data2['description'] = $request->description;
        $data2['long_description'] = $request->long_description;
        $data2['units'] = $request->units;
        $data2['tax_type_id'] = $request->tax_type_id;
        $data2['category_id'] = $request->category_id;
        $data2['created_at'] = date('Y-m-d H:i:s');

        DB::table('stock_master')->insert($data2);

        $data3[0]['stock_id'] = strtoupper($request->stock_id);
        $data3[0]['sales_type_id'] = 1;
        $data3[0]['price'] = 0;
        $data3[0]['curr_abrev'] = 'USD';

        $data3[1]['stock_id'] = strtoupper($request->stock_id);
        $data3[1]['sales_type_id'] = 2;
        $data3[1]['price'] = 0;
        $data3[1]['curr_abrev'] = 'USD';

        DB::table('sale_prices')->insert($data3);

        $purchaseInfos['stock_id'] = strtoupper($request->stock_id);
        $purchaseInfos['price'] = 0;
        DB::table('purchase_prices')->insert($purchaseInfos);
    }

```

```

        //\Session::flash('success',trans('message.success.save_success
'));
        return redirect()->intended("edit-item/item-info/$id");
    } else {

        return back()->withInput()->withErrors(['email' => "Invalid
Request !"]);
    }
}

/**
 * Store a newly created Item sales price in storage.
 * @param \Illuminate\Http\Request $request
 * @return redirection Item purchasing price
 */
public function storeSalePrice(Request $request)
{
    $this->validate($request, [
        'stock_id' => 'required|unique:sale_prices,stock_id',
        'sales_type_id' => 'required',
        'price' => 'required|numeric'
    ]);
    $data['stock_id'] = strtoupper($request->stock_id);
    $data['sales_type_id'] = $request->sales_type_id;
    $data['price'] = $request->price;
    $data['curr_abrev'] = $request->curr_abrev;
    DB::table('sale_prices')->insert($data);
    return redirect()->intended('create-item/purchase');
}

/**
 * Update Item sales price in storage.
 * @param \Illuminate\Http\Request $request
 *
 */
public function addSalePrice(Request $request)
{
    $this->validate($request, [
        'stock_id' => 'required',
        'sales_type_id' => 'required',
        'price' => 'required|numeric'
    ]);

    $id = $request->item_id;

    $data['stock_id'] = $request->stock_id;
    $data['sales_type_id'] = $request->sales_type_id;
    $data['price'] = $request->price;

```

```

$data['curr_abrev'] = $request->curr_abrev;

$ceck = DB::table('sale_prices')->where([
    ['stock_id', '=', $request->stock_id],
    ['sales_type_id', '=', $request->sales_type_id],
])->first();
if (!empty($ceck)) {
    \Session::flash('fail', "Already added !");
    return redirect()->intended("edit-item/sales-info/$id");
} else {

    DB::table('sale_prices')->insert($data);
    \Session::flash('success',
trans('message.success.update_success'));
    return redirect()->intended('edit-item/sales-info/' . $request-
>item_id);
}
}

public function editSalePrice(Request $request)
{
    //dd($request->id);
    $id = $request->id;

    $saleTypes = DB::table('sales_types')->get();

    $salesTypeName = array();
    foreach ($saleTypes as $value) {
        $salesTypeName[$value->id] = $value->sales_type;
    }

    $data = DB::table('sale_prices')->where('id', $id)->first();

    if (!empty($data)) {
        $v['id'] = $id;
        $v['sales_type_id'] = $salesTypeName[$data->sales_type_id];
        $v['price'] = $data->price;
        $v['status'] = 1;

        echo json_encode($v);
    }
}

public function updateSalePrice(Request $request)
{
    $id = $request->id;

    DB::table('sale_prices')

```

```

        ->where('id', $id)
        ->update(['price' => $request->price]);

        \Session::flash('success',
trans('message.success.update_success'));
        return redirect()->intended('edit-item/sales-info/' . $request-
>item_id);
    }

    public function deleteSalePrice($id, $item_id)
    {
        if (isset($id)) {
            $record = \DB::table('sale_prices')->where('id', $id)->first();
            if ($record) {

                \DB::table('sale_prices')->where('id', '=', $id)->delete();

                \Session::flash('success',
trans('message.success.delete_success'));
                return redirect()->intended('edit-item/sales-info/' .
$item_id);
            }
        }
    }

    /**
     * Store a newly created Item purchase price in storage.
     * @param \Illuminate\Http\Request $request
     * @return redirection Item purchasing price
     */
    public function updatePurchasePrice(Request $request)
    {
        $this->validate($request, [
            'stock_id' => 'required',
            'price' => 'required|numeric',
            // 'supplier_id'=>'required'
        ]);

        $data['price'] = $request->price;
        //$data['supplier_id'] = $request->supplier_id;
        $data['stock_id'] = $request->stock_id;
        $priceInfo = DB::table('purchase_prices')->where('stock_id',
$request->stock_id)->count();
        // d($priceInfo,1);
        if ($priceInfo == 0) {
            DB::table('purchase_prices')->insert($data);
        } else {

```

```

        DB::table('purchase_prices')->where('stock_id', $request-
>stock_id)->update($data);
    }

    \Session::flash('success',
trans('message.success.update_success'));
    return redirect()->intended('edit-item/purchase-info/' . $request-
>item_id);
}
/**
 * Store a newly created Item purchase price in storage.
 * @param \Illuminate\Http\Request $request
 * @return redirection Item purchasing price
 */
public function storePurchasePrice(Request $request)
{
    $this->validate($request, [
        'stock_id' => 'required|unique:purchase_prices,stock_id',
        // 'suppliers_uom' => 'required',
        'price' => 'required|numeric',
        'supplier_id' => 'required'
    ]);

    $data['stock_id'] = strtoupper($request->stock_id);
    $data['suppliers_uom'] = $request->suppliers_uom;
    $data['price'] = $request->price;
    $data['supplier_id'] = $request->supplier_id;
    $data['conversion_factor'] = $request->conversion_factor;
    $data['supplier_description'] = $request->supplier_description;

    \Session::flash('success', trans('message.success.save_success'));
    DB::table('purchase_prices')->insert($data);
    return redirect()->intended('item');
}

/**
 * Display the specified Item.
 *
 * @param int $id
 * @return Item show page view
 */
public function show($id)
{
    $data['menu'] = 'item';
    $data['header'] = 'item';
    $data['breadcrumb'] = 'showitem';
    $loc = DB::table('location')->get();
    $loc_name = array();

```

```

        foreach ($loc as $value) {
            $loc_name[$value->loc_code] = $value->location_name;
        }

        $data['loc_name'] = $loc_name;

        $data['itemData'] = (new Item)->getItemById($id);
        $data['loc_Data'] = (new Item)->getTransaction($data['itemData']->stock_id);
        $data['locData'] = DB::table('location')->get();
        $data['itemHistory'] = DB::table('stock_moves')
            ->where(['stock_id' => $data['itemData']->stock_id])
            ->orderBy('trans_id', 'desc')
            ->take(5)
            ->get();

        return view('admin.item.item_show', $data);
    }

    /**
     * Show the form for editing the specified Item.
     *
     * @param int $id
     * @return Item edit page view
     */
    public function edit($tab, $id)
    {
        $data['menu'] = 'item';
        $data['header'] = 'item';
        $data['breadcrumb'] = 'additem';
        $data['locData'] = DB::table('location')->get();
        $data['taxTypes'] = DB::table('item_tax_types')->get();
        $data['saleTypes'] = DB::table('sales_types')->get();
        $data['categoryData'] = DB::table('stock_category')->get();
        $data['unitData'] = DB::table('item_unit')->get();
        $data['suppliers'] = DB::table('suppliers')->get();

        $loc = DB::table('location')->get();
        $loc_name = array();
        foreach ($loc as $value) {
            $loc_name[$value->loc_code] = $value->location_name;
        }

        $data['loc_name'] = $loc_name;

        $salesTypeName = array();
        foreach ($data['saleTypes'] as $value) {

```



```

        $salesTypeName[$value->id] = $value->sales_type;
    }

    $data['salesTypeName'] = $salesTypeName;

    $data['itemInfo'] = DB::table('item_code')
        ->leftjoin('stock_master', 'stock_master.stock_id', '=',
'item_code.stock_id')
        ->where('item_code.id', $id)
        ->select('item_code.*', 'stock_master.tax_type_id',
'stock_master.units', 'stock_master.long_description')
        ->first();

    $data['salesInfo'] = DB::table('sale_prices')
        ->where('stock_id', $data['itemInfo']->stock_id)
        ->first();
    $data['purchaseInfo'] = DB::table('purchase_prices')
        ->where('stock_id', $data['itemInfo']->stock_id)
        ->first();
    //d($data['purchaseInfo'],1);
    $data['salePriceData'] = DB::table('sale_prices')-
->where('stock_id', $data['itemInfo']->stock_id)->get();

    $data['itemQuantity'] = DB::table('stock_moves')
        ->select(DB::raw('SUM(qty) as total_item'), 'loc_code')
        ->where('stock_id', strtoupper($data['itemInfo']->stock_id))
        ->groupBy('loc_code')
        ->get();

    $data['tab'] = $tab;

    $data['transations'] = DB::table('stock_moves')
        ->where('stock_moves.stock_id', $data['itemInfo']->stock_id)
        ->leftjoin('item_code', 'item_code.stock_id', '=',
'stock_moves.stock_id')
        ->leftjoin('location', 'location.loc_code', '=',
'stock_moves.loc_code')
        ->select('stock_moves.*', 'item_code.description',
'location.location_name')
        ->orderBy('stock_moves.tran_date', 'DESC')
        ->get();
    //d($data['transations'],1);
    return view('admin.item.item_edit', $data);
}

/**
 * Show the form for Copy the specified Item.
 */

```

```

* @param int $id
* @return Item copy page view
*/
public function copy($id)
{
    $data['menu'] = 'item';
    $data['header'] = 'item';
    $data['breadcrumb'] = 'copyitem';
    $data['locData'] = DB::table('location')->get();
    $data['taxTypes'] = DB::table('item_tax_types')->get();
    $data['categoryData'] = DB::table('stock_category')->get();
    $data['unitData'] = DB::table('item_unit')->get();
    $data['itemData'] = (new Item)->getItemById($id);

    return view('admin.item.item_copy', $data);
}

```

```

/**
 * Update the specified Item in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return redirection Item list page view
 */
public function updateItemInfo(Request $request)
{
    $this->validate($request, [
        'stock_id' => 'required',
        'description' => 'required',
        'units' => 'required',
        'item_image' => 'mimes:jpeg,bmp,png'
    ]);

    $data['description'] = $request->description;
    $data['category_id'] = $request->category_id;
    $data['inactive'] = $request->inactive;
    $data['updated_at'] = date('Y-m-d H:i:s');

    $pic = $request->file('item_image');

    if (isset($pic)) {

        $pic1 = $request->pic;
        if ($pic1 != NULL) {
            $dir = public_path("uploads/itemPic/$pic1");
            if (file_exists($dir)) {
                unlink($dir);
            }
        }
    }
}

```

```

    }

    $destinationPath = public_path('/uploads/itemPic/');
    $filename = $pic->getClientOriginalName();
    $img = Image::make($request->file('item_image')->getRealPath());

    $img->resize(400, 400, function ($constraint) {
        $constraint->aspectRatio();
    })->save($destinationPath . '/' . $filename);

    $data['item_image'] = $filename;
}

DB::table('item_code')->where('id', $request->id)->update($data);

$stockMaster['description'] = $request->description;
$stockMaster['long_description'] = $request->long_description;
$stockMaster['units'] = $request->units;
$stockMaster['category_id'] = $request->category_id;
$stockMaster['tax_type_id'] = $request->tax_type_id;
$stockMaster['inactive'] = $request->inactive;
$stockMaster['updated_at'] = date('Y-m-d H:i:s');

DB::table('stock_master')->where('stock_id', $request->stock_id)->update($stockMaster);

\Session::flash('success',
trans('message.success.update_success'));
return redirect()->intended('edit-item/item-info/' . $request->id);
}

/**
 * Remove the specified Item from storage.
 *
 * @param int $id
 * @return redirect Item list page view
 */
public function destroy($id)
{
    if (isset($id)) {
        $record = \DB::table('item_code')->where('id', $id)->first();
        if ($record) {
            DB::table('item_code')->where('id', '=', $id)->update(['deleted_status' => 1]);
            DB::table('stock_master')->where('stock_id', '=', $record->stock_id)->update(['deleted_status' => 1]);
        }
    }
}

```

```

        \Session::flash('success',
trans('message.success.delete_success'));
        return redirect()->intended('item');
    }
}

/**
 * Add item to stock.
 *
 * @param \Illuminate\Http\Request $request
 * @return redirect Item show page view
 */
public function addStock(Request $request)
{
    $p_id = \Auth::user()->id;
    $data['stock_id'] = $request->stock_id;
    $data['type'] = $request->type;
    $data['loc_code'] = $request->loc_code;
    $data['person_id'] = $p_id;
    $data['qty'] = $request->quantity;
    $data['note'] = $request->reference;
    $data['tran_date'] = date('Y-m-d H:i:s');

    $insert_id = DB::table('stock_moves')->insertGetId($data);

    $id = $request->id;

    if (!empty($insert_id)) {
        \Session::flash('success',
trans('message.success.added_success'));
        return redirect()->intended("show-item/$id");
    }
}

/**
 * Remove item to stock.
 *
 * @param \Illuminate\Http\Request $request
 * @return redirect Item show page view
 */
public function removeStock(Request $request)
{
    $p_id = \Auth::user()->id;
    $data['stock_id'] = $request->stock_id;
    $data['type'] = $request->type;
    $data['loc_code'] = $request->loc_code;
    $data['person_id'] = $p_id;

```

```

        $data['qty'] = "-" . $request->quantity;
        $data['note'] = $request->reference;
        $data['tran_date'] = date('Y-m-d H:i:s');

        $id = $request->id;

        $status = $this->qtyValidate($request->loc_code, $request-
>stock_id, $request->quantity);
        //dd($status);

        if ($status['fail'] == 1) {
            $nm = $status['stock'];
            \Session::flash('fail', "You Can not Remove more than $nm
item.");
            return redirect()->intended("show-item/$id");
            exit();
        }
        $insert_id = DB::table('stock_moves')->insertGetId($data);

        if (!empty($insert_id)) {
            \Session::flash('success',
trans('message.success.removed_success'));
            return redirect()->intended("show-item/$id");
        }
    }

    /**
     * Move item from one to another stock.
     *
     * @param \Illuminate\Http\Request $request
     * @return redirect Item show page view
     */
    public function moveStock(Request $request)
    {
        $id = $request->id;
        $p_id = \Auth::user()->id;
        $data['stock_id'] = $request->stock_id;
        $data['type'] = $request->type;
        $data['loc_code'] = $request->loc_code1;
        $data['person_id'] = $p_id;
        $data['qty'] = "-" . $request->quantity;
        $data['note'] = $request->reference;
        $data['tran_date'] = date('Y-m-d H:i:s');

        $status = $this->qtyValidate($request->loc_code1, $request-
>stock_id, $request->quantity);

        if ($status['fail'] == 1) {

```

```

        $nm = $status['stock'];
        \Session::flash('fail', "You Can not Move more than $nm
item.");
        return redirect()->intended("show-item/$id");
        exit();
    }

$insert_id = DB::table('stock_moves')->insert($data);

$data2['stock_id'] = $request->stock_id;
$data2['type'] = $request->type;
$data2['loc_code'] = $request->loc_code2;
$data2['person_id'] = $p_id;
$data2['qty'] = $request->quantity;
$data2['note'] = $request->reference;
$data2['tran_date'] = date('Y-m-d H:i:s');

$insert_id = DB::table('stock_moves')->insert($data2);

$loc = DB::table('location')->get();
$loc_name = array();
foreach ($loc as $value) {
    $loc_name[$value->loc_code] = $value->location_name;
}

$a = $loc_name[$request->loc_code1];
$b = $request->quantity;
$c = $loc_name[$request->loc_code2];

\Session::flash('success', trans('message.success.moved_success'));
return redirect()->intended("show-item/$id");
}

/**
 * Validated item quantity if it can be removed from stock.
 */
public function qtyValidate($loc, $id, $qty)
{
    $v = [];
    $current_stock = (new Item)->stock_validate($loc, $id);

    if ($current_stock->total < $qty) {

        $v['stock'] = $current_stock->total;
        $v['fail'] = 1;
    }
}

```

```

        return $v;
    } else {
        $v['fail'] = 0;
        return $v;
    }
}

/**
 * Validated item quantity if it can be removed from stock. using ajax
 */
public function qtyValidAjax()
{
    $v = [];
    $id = $_POST['mvid'];
    $loc = $_POST['loc1'];
    $qty = $_POST['qty'];

    $current_stock = (new Item)->stock_validate($loc, $id);
    //dd($current_stock);

    if (empty($current_stock)) {

        $v['fail'] = 2;
        echo json_encode($v);
    } elseif (!empty($current_stock) && $current_stock->total < $qty) {

        $v['stock'] = $current_stock->total;
        $v['fail'] = 1;

        echo json_encode($v);
    } else {
        $v['fail'] = 0;
        echo json_encode($v);
    }
}

/**
 * Validated item Id is unique or not
 */
public function stockValidChk()
{
    $st_id = $_POST['stock_id'];
    $v = DB::table('item_code')->where('stock_id', $st_id)->first();

    if (!empty($v)) {
        echo 1;
    } else {
        echo 0;
    }
}

```

```

    }
}

/**
 * Show full transation details
 */
public function showFullDetails($id)
{
    $data['menu'] = 'item';
    $data['header'] = 'item';
    $data['breadcrumb'] = 'historyitem';
    $loc = DB::table('location')->get();
    $loc_name = array();
    foreach ($loc as $value) {
        $loc_name[$value->loc_code] = $value->location_name;
    }

    $data['loc_name'] = $loc_name;
    $data['itemHistory'] = DB::table('stock_moves')
        ->where(['stock_id' => $id])
        ->orderBy('trans_id', 'desc')
        ->get();

    return view('admin.item.item_transition', $data);
}

public function downloadCsv($type)
{
    if ($type == 'csv') {

        $itemdata = (new Item)->getAllItemCsv();
        //d($itemdata,1);
        foreach ($itemdata as $key => $value) {
            $csvdata[$key]['Item ID'] = $value->stock_id;
            $csvdata[$key]['Item Name'] = $value->item_name;
            // $csvdata[$key]['Long Description'] =
            $value['long_description'];
            $csvdata[$key]['Item Category'] = $value->category;
            // $csvdata[$key]['Unit'] = $value['units'];
            // $csvdata[$key]['Tax Type'] = $value['tax_type'];
            $csvdata[$key]['Purchase Price'] = $value->purchashe_price;
            $csvdata[$key]['Retail Price'] = $value->retail_price;
            $csvdata[$key]['Whole Sale Price'] = $value->wholesale_price;
        }
    }
    //d($csvdata,1);
}

```



```

        if ($type == 'sample') {
            $csvdata[0]['Item ID'] = 'DELL';
            $csvdata[0]['Item Name'] = 'Dell desktop computer';
            $csvdata[0]['Long Description'] = 'This is a destop computer';
            $csvdata[0]['Item Category'] = 'Computer';
            $csvdata[0]['Unit'] = 'pc';
            $csvdata[0]['Tax Type'] = 'Normal';
            $csvdata[0]['Purchase Price'] = '50';
            $csvdata[0]['Retail Price'] = '60';
            $csvdata[0]['Wholesale Price'] = '55';

            $type = 'csv';
        }

        return Excel::create('Item_sheet' . time() . '', function ($excel)
use ($csvdata) {

            $excel->sheet('mySheet', function ($sheet) use ($csvdata) {
                $sheet->fromArray($csvdata);
            });
        }->download($type);
    }

    public function import()
    {
        $data['menu'] = 'item';

        return view('admin.item.item_import', $data);
    }

    public function importCsv(Request $request)
    {
        $file = $request->file('import_file');
        // d($file,1);
        $validator = Validator::make(
            [
                'file' => $file,
                'extension' => strtolower($file-
>getClientOriginalExtension()),
            ],
            [
                'file' => 'required',
                'extension' => 'required|in:csv',
            ]
        );

        if ($validator->fails()) {
            return back()->withErrors(['email' => "File type Invalid !"]);
        }
    }

```

```

    }

    if (Input::hasFile('import_file')) {
        $path = Input::file('import_file')->getRealPath();

        $csv = array_map('str_getcsv', file($path));
        //d($csv,1);
        $unMatch = [];
        $header = array("Item ID", "Item Name", "Long Description",
            "Item Category", "Unit", "Tax Type", "Purchase Price", "Retail Price",
            "Wholesale Price");

        for ($i = 0; $i < sizeof($csv[0]); $i++) {
            if (!in_array(trim($csv[0][$i]), $header)) {
                $unMatch[] = trim($csv[0][$i]);
            }
        }

        if (!empty($unMatch)) {

            return back()->withErrors(['email' => "Please Check Csv
Header Name !"]);
        }

        $data = [];
        foreach ($csv as $key => $value) {
            if ($key != 0) {

                $data[$key]['item_id'] =
mb_convert_encoding(trim($value[0]), 'ISO-8859-1');
                $data[$key]['item_name'] =
mb_convert_encoding(trim($value[1]), 'ISO-8859-1');
                $data[$key]['long_description'] =
mb_convert_encoding(trim($value[2]), 'ISO-8859-1');
                $data[$key]['item_category'] =
mb_convert_encoding(trim($value[3]), 'ISO-8859-1');
                $data[$key]['unit'] =
mb_convert_encoding(trim($value[4]), 'ISO-8859-1');
                $data[$key]['tax_type'] =
mb_convert_encoding(trim($value[5]), 'ISO-8859-1');
                $data[$key]['purch_price'] =
mb_convert_encoding(trim($value[6]), 'ISO-8859-1');
                $data[$key]['retail_price'] =
mb_convert_encoding(trim($value[7]), 'ISO-8859-1');
                $data[$key]['whole_price'] =
mb_convert_encoding(trim($value[8]), 'ISO-8859-1');
            }
        }
    }

```

```

$data = array_values($data);
//d($data,1);

$cat_id = array();
$cat     = DB::table('stock_category')->get();

foreach ($cat as $value) {
    $cat_id[$value->description] = $value->category_id;
}

$tax_id = array();
$tax     = DB::table('item_tax_types')->get();

foreach ($tax as $value) {
    $tax_id[$value->name] = $value->id;
}

//d($data,1);
if (!empty($data)) {

    $item = uniqueMultidimArray($data, 'item_id');
    sort($item);
    foreach ($item as $key => $value) {

        $itemCount      = DB::table('item_code')->
>where(['stock_id' => $value['item_id'], 'deleted_status' => 0])->count();
        $categoryCount  = DB::table('stock_category')->
>where('description', '=', $value['item_category'])->count();
        $unitCount      = DB::table('item_unit')->where('name',
'=', $value['unit'])->count();
        $taxCount       = DB::table('item_tax_types')->
>where('name', '=', $value['tax_type'])->count();
        //d($itemCount);
        /*d($value);
        d($itemCount);
        d($categoryCount);
        d($unitCount,1);*/

        if ($itemCount == 0 && $categoryCount > 0 && $unitCount
> 0 && $taxCount > 0) {
            //d($data);
            $itemCode[] = [
                'stock_id'      => $value['item_id'],
                'description' => $value['item_name'],
                'category_id' =>
$cat_id[$value['item_category']]

```

```

];

$stockMaster[] = [
    'stock_id' =>
strtoupper($value['item_id']),
    'description' => $value['item_name'],
    'long_description' =>
$value['long_description'],
    'units' => $value['unit'],
    'tax_type_id' =>
$tax_id[$value['tax_type']],
    'category_id' =>
$cat_id[$value['item_category']]
];

$purchPrice[] = [
    'stock_id' => strtoupper($value['item_id']),
    //'price' => $value['purch_price'],
    'price' => preg_replace('/[\$,]/', '',
$value['purch_price']),
    // 'supplier_id' => 0
];

$salePrice[] = [
    'stock_id' => strtoupper($value['item_id']),
    //'r_price' => $value['retail_price'],
    'r_price' => str_replace(",", "",
$value['retail_price']),
    //'w_price' => $value['whole_price']
    'w_price' => str_replace(",", "",
$value['whole_price'])
];

}
}
// d($purchPrice);
//d($salePrice,1);
//d($itemCode,1);
/*d($itemCode);
d($salePrice);
d($purchPrice,1);*/

if (!empty($salePrice)) {
    $k = 0;
    foreach ($salePrice as $key => $value) {
        $retailPrice[$key]['stock_id'] =
strtoupper($value['stock_id']);
        $retailPrice[$key]['sales_type_id'] = 1;
    }
}

```

```

        $retailPrice[$key]['price'] =
preg_replace('/[\$,]\/', '', $value['r_price']) ? preg_replace('/[\$,]\/',
'', $value['r_price']) : 0;
        $retailPrice[$key]['curr_abrev'] = 'USD';

        $wholePrice[$key]['stock_id'] =
strtoupper($value['stock_id']);
        $wholePrice[$key]['sales_type_id'] = 2;
        $wholePrice[$key]['price'] =
preg_replace('/[\$,]\/', '', $value['w_price']) ? preg_replace('/[\$,]\/',
'', $value['w_price']) : 0;
        $wholePrice[$key]['curr_abrev'] = 'USD';
    }
}

if (!empty($itemCode)) {
    DB::table('item_code')->insert($itemCode);
    DB::table('stock_master')->insert($stockMaster);
    DB::table('purchase_prices')->insert($purchPrice);
    DB::table('sale_prices')->insert($retailPrice);
    DB::table('sale_prices')->insert($wholePrice);

    \Session::flash('success',
trans('message.success.import_success'));
    return redirect()->intended('item');
} else {
    return back()->withErrors(['email' => "Please Check Csv
File !"]);
}
}
}
return back();
}

public function validate_search($serie)
{
    $producto = DB::table('item_code')->where('stock_id', $serie)-
>first();
    return view('admin.item.item_unico', ['producto' => $producto]);
}
}

```

Anexo

Manual de usuario

Se presentan un conjunto de screenshot donde se detalla el uso y control del sistema de inventario implementado para la empresa SOLUDESING.

- **Login de Usuario:** Se presenta la pantalla inicial en donde se debe ingresar el usuario y contraseña para loguearse en el sistema.

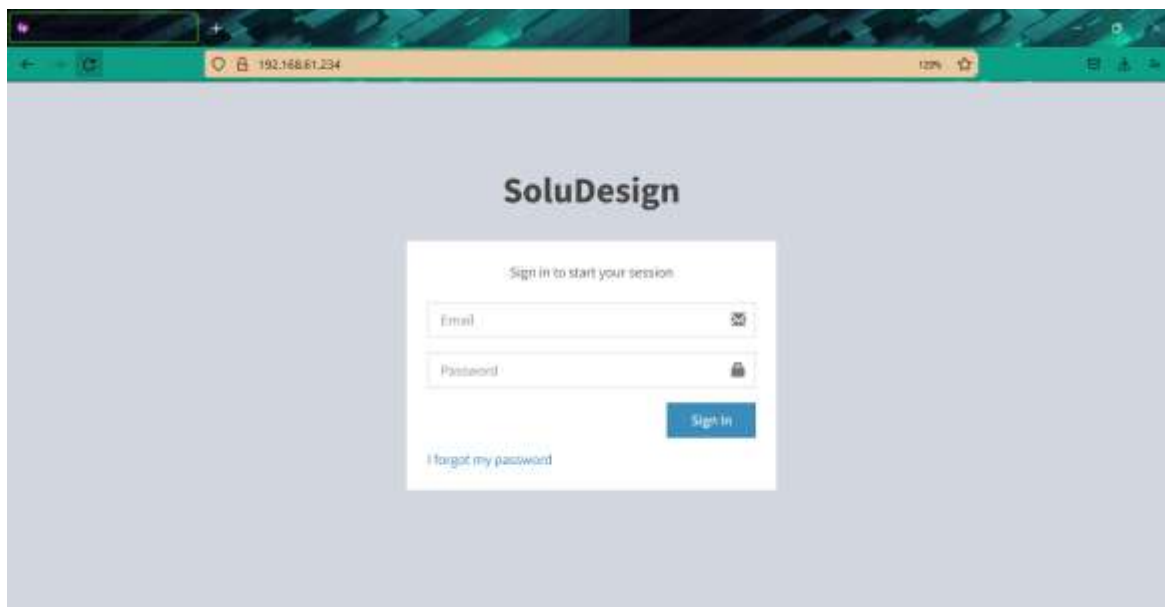


Figura Loguin del sistema de inventario. Elaborado por Torres Neira Emily.

- **Pantalla Principal:** Se presenta un estadístico que contiene los pedidos realizados (Ingresos), Egresos, y Ganancias que han sido registradas en el proceso de inventario de la compañía.

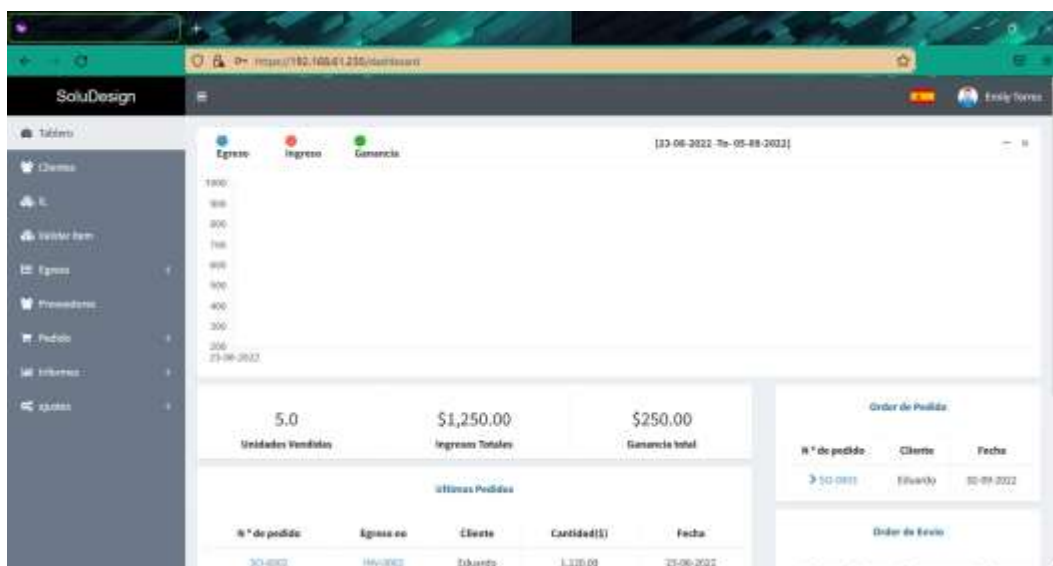


Figura Pantalla principal del sistema de inventario. Elaborado por Torres Neira Emily.

5.0
Unidades Vendidas

\$1,250.00
Ingresos Totales

\$250.00
Ganancia total

Ultimos Pedidos

| N° de pedido | Egreso no. | Cliente | Cantidad(\$) | Fecha |
|--------------|------------|---------|--------------|------------|
| SO-0002 | WV-0002 | Eduardo | 1,120.00 | 23-08-2022 |
| SO-0001 | WV-0001 | Eduardo | 280.00 | 03-08-2022 |

Ultimos Pagos

| Pago No. | N° de pedido | Egreso no. | Cliente | Cantidad(\$) | Fecha de pago |
|----------|--------------|------------|---------|--------------|---------------|
| 0002 | SO-0002 | WV-0002 | Eduardo | 1,120.00 | 23-08-2022 |
| 0001 | SO-0001 | WV-0001 | Eduardo | 280.00 | 03-08-2022 |

Orden de Pedido

| N° de pedido | Cliente | Fecha |
|--------------|---------|------------|
| SO-0001 | Eduardo | 03-08-2022 |

Orden de Envío

| N° de pedido | Cliente | Fecha |
|--------------|---------|------------|
| SO-0002 | Eduardo | 03-08-2022 |
| SO-0001 | Eduardo | 03-08-2022 |
| SO-0002 | Eduardo | 03-08-2022 |

Figura Pantalla principal del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Actualización de datos de la empresa

Para realizar alguna actualización de los datos de la empresa es necesario ingresar en el apartado *Ajustes – Detalles de la Compañía*, ingresando en la opción configuración de la empresa se deben modificar cada uno de los datos informativos que se desean actualizar.

Manage Company Settings

Configuración de la empresa

Detalles de la compañía

Ajustes generales

Financiero

Plantilla de correo electrónico

Preferencia

Impresión código de barras / etiqueta

Configuración de la empresa

Nombre*

SoluDesign

Nombre corto del sitio

SD

Vat Number

2101231

correo electrónico*

atorres@soludesign.co

Teléfono*

123456789

Calle*

Sancta

Ciudad*

Guayaquil

Estado*

Guayas

Zip code

90001

País*

Ecuador

Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Agregar un nuevo usuario en el sistema

Para la creación de un usuario es necesario establecer previamente los roles que tendrán dentro del sistema, al ingresar a la opción *roles de usuario* se deben elegir cada una de las opciones a las que tendrá acceso dentro del sistema el rol que esta siendo creado en ese momento, para que al proceder con la creación de usuarios se seleccione un tipo de rol ya registrado.

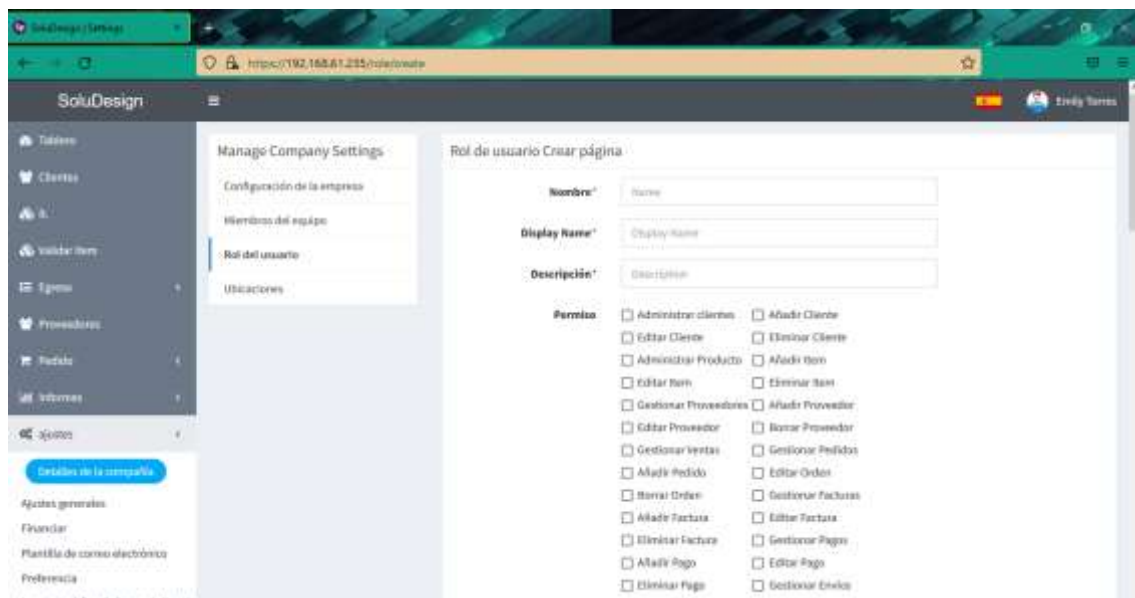


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

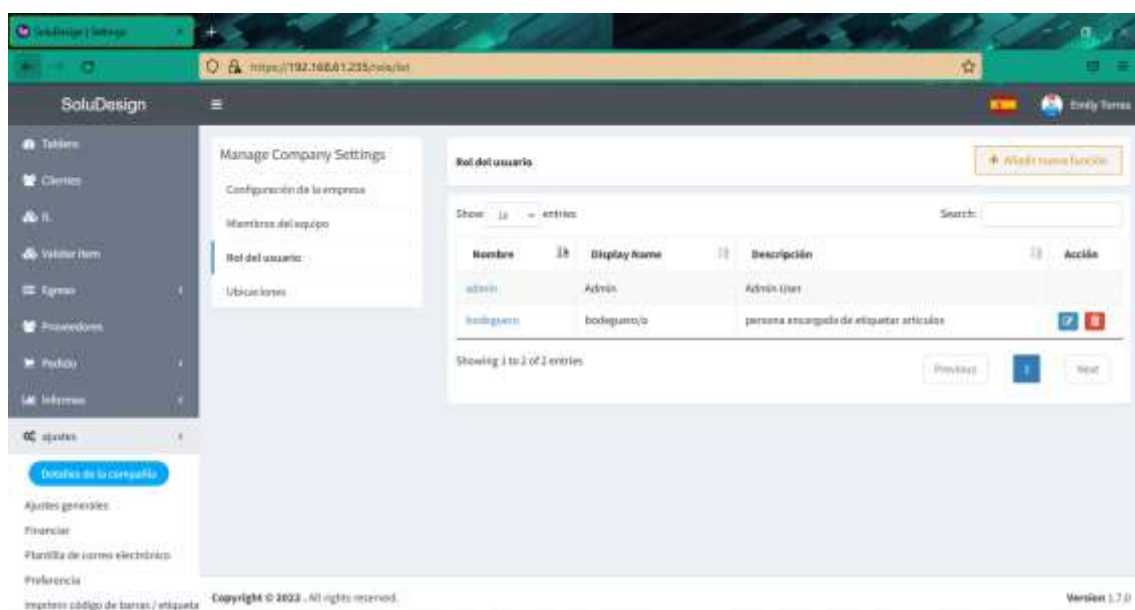


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

Para realizar el ingreso de un nuevo usuario en el sistema es necesario proporcionar un correo y contraseña que posteriormente puede ser actualizada por el usuario, la opción de agregar nuevos miembros de equipo solo está disponible para el usuario administrador del sistema.

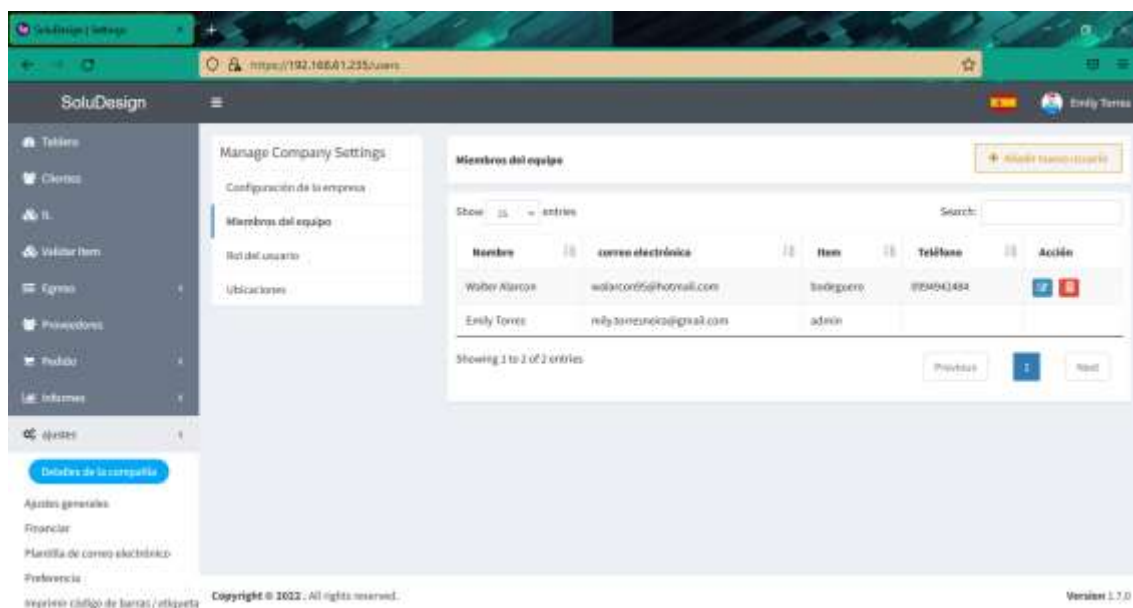


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Creación de categoría de productos

Ingresando al apartado *ajustes generales – categoría elemento* al dar clic en el botón agregar nueva categoría se deben ingresar cada uno de los campos presentados para registrar está en el sistema.

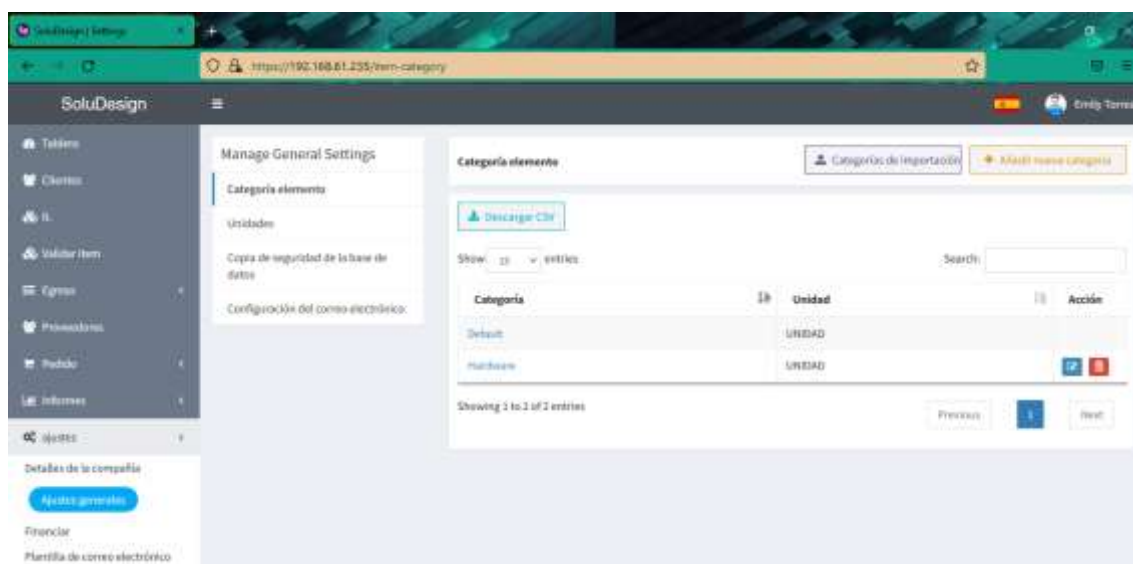


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Creación de Copias de seguridad de las bases de datos

En esta opción se permite realizar copias de seguridad de la información ingresada en el sistema que se encuentra ligada a la base de datos.

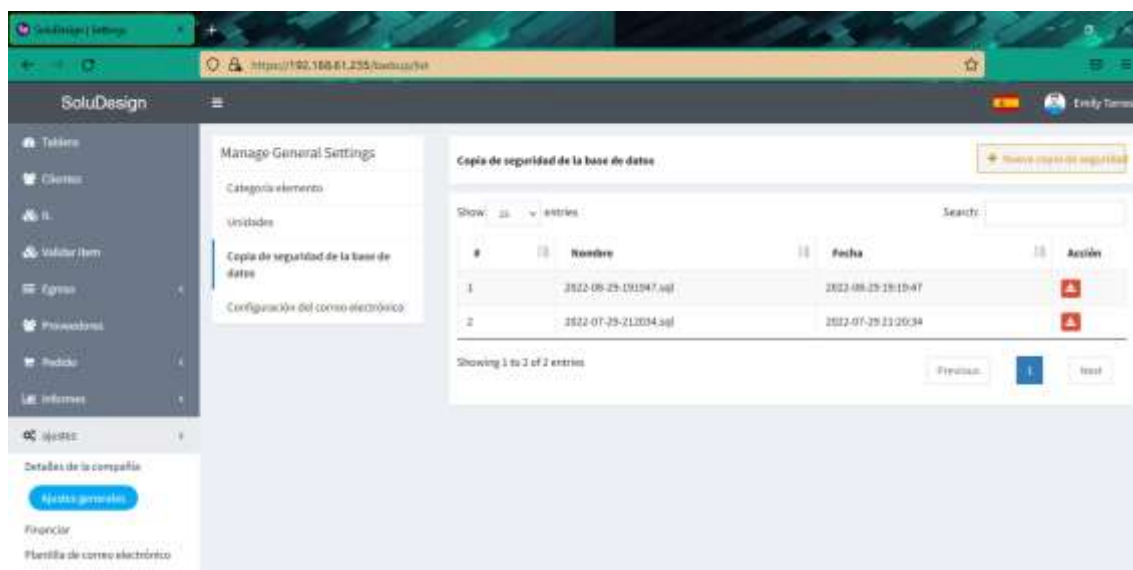


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Establecer impuestos en el sistema

Para ingresar o editar un impuesto que se consideran en las ordenes de egreso es necesario ir al apartado *financiar* - *impuestos*, al dar clic en esta opción se presenta un listado con cada uno de los impuestos establecidos en el sistema permitiendo agregar nuevos en caso de ser requerido por el usuario administrador.

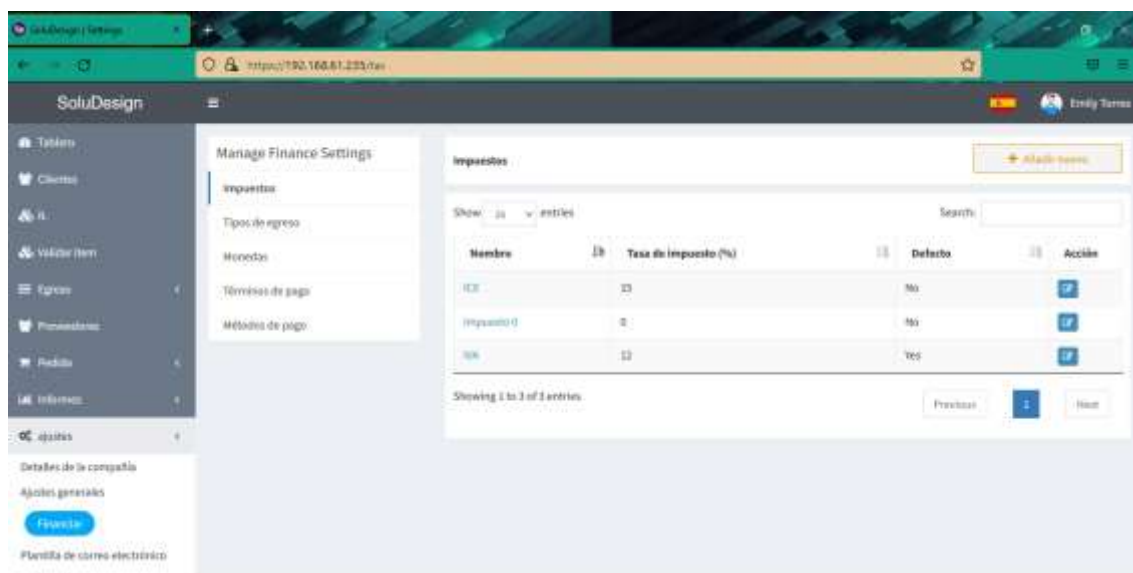


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Imprimir Códigos de Barra

Mediante la búsqueda de un artículo que se encuentre dentro del inventario ingresado en el sistema se permite la impresión de las etiquetas pertenecientes al producto permitiendo que contengan el nombre de la empresa y el nombre del producto.

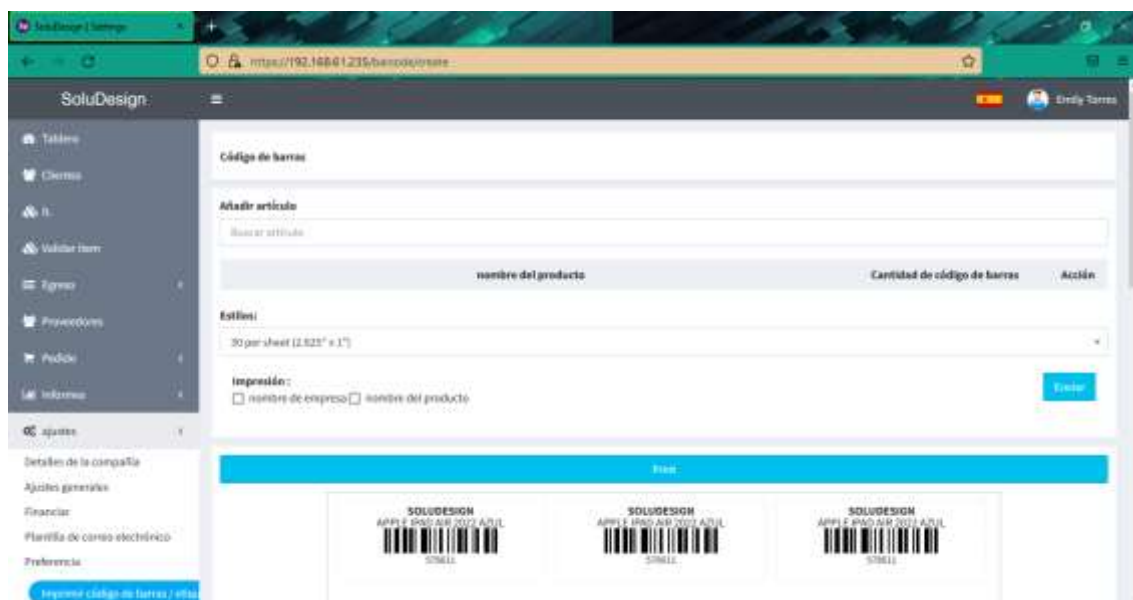


Figura Ajustes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Creación de Clientes

Permite visualizar un listado de cada uno de los clientes registrados en el sistema y a su vez crear un cliente nuevo en caso de ser requerido.

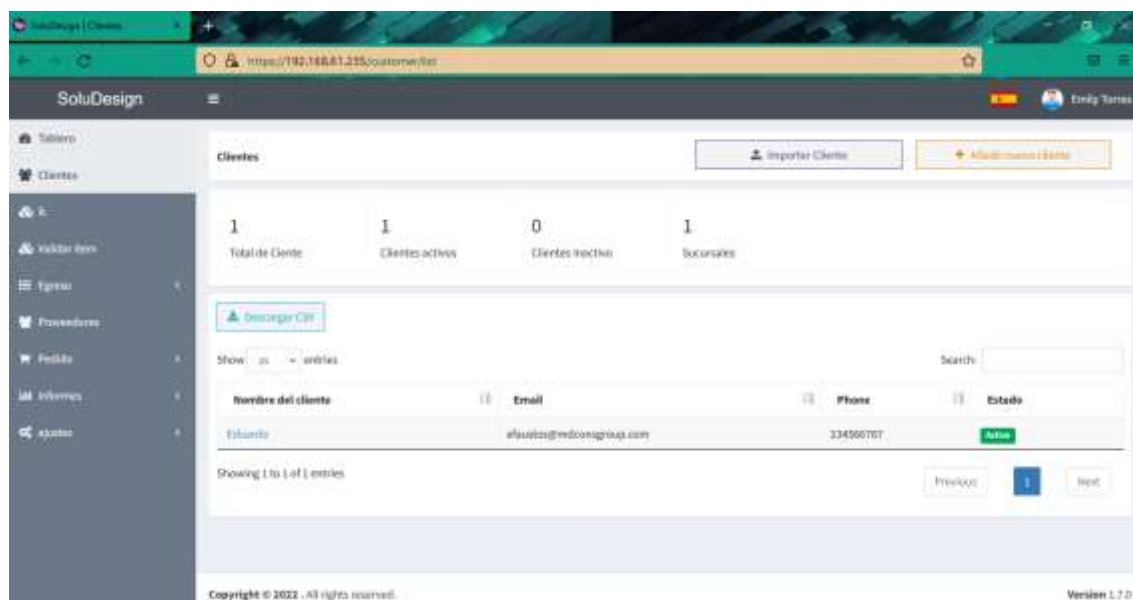


Figura Módulo Clientes del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Módulo Ítems

Se presenta un listado completo de cada uno de los productos que han sido ingresados y registrados en el sistema.

| Imagen | Nombre | Categoría | En Bodega | Ingreso(\$) | Venta al Mayor(\$) | Venta al Menor(\$) | Estado | Acciones |
|--------|------------------------------|-----------|-----------|-------------|--------------------|--------------------|--------|----------|
| | APPLE IPAD 4th 32GB DOCK | Hardware | 8 | 810 | 630 | 625.8 | Activo | |
| | APPLE IPAD 4th 32GB AZUL | Hardware | 8 | 810 | 615.39 | 635.38 | Activo | |
| | CAMARA DE VIDEO DE HD(1080i) | Hardware | 1 | 95 | 95 | 95 | Activo | |
| | CELULAR HOTT 115 WESBO | Hardware | 8 | 310 | 228 | 228 | Activo | |

Figura Módulo It del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Validación de Ítems

Para la utilización de este apartado se requiere tener conectada una cámara web, ya que mediante el uso de esta se permite identificar el código o serie de un producto y se procede a determinar si este se encuentra ingresado en el sistema o no.

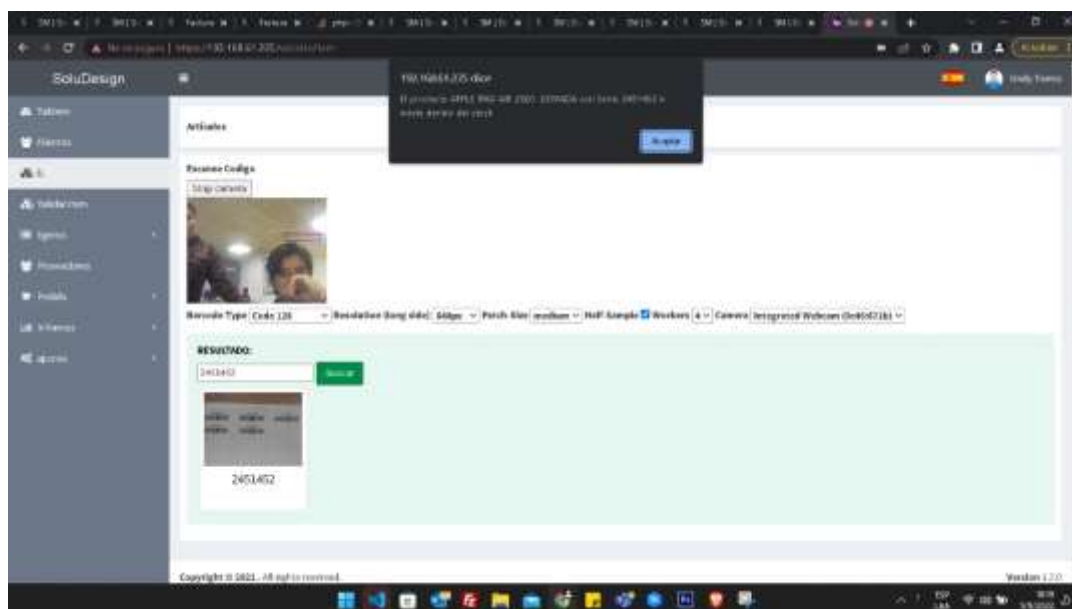


Figura Módulo Validar Ítems del sistema de inventario. Alerta de productos existentes. Elaborado por Torres Neira Emily.

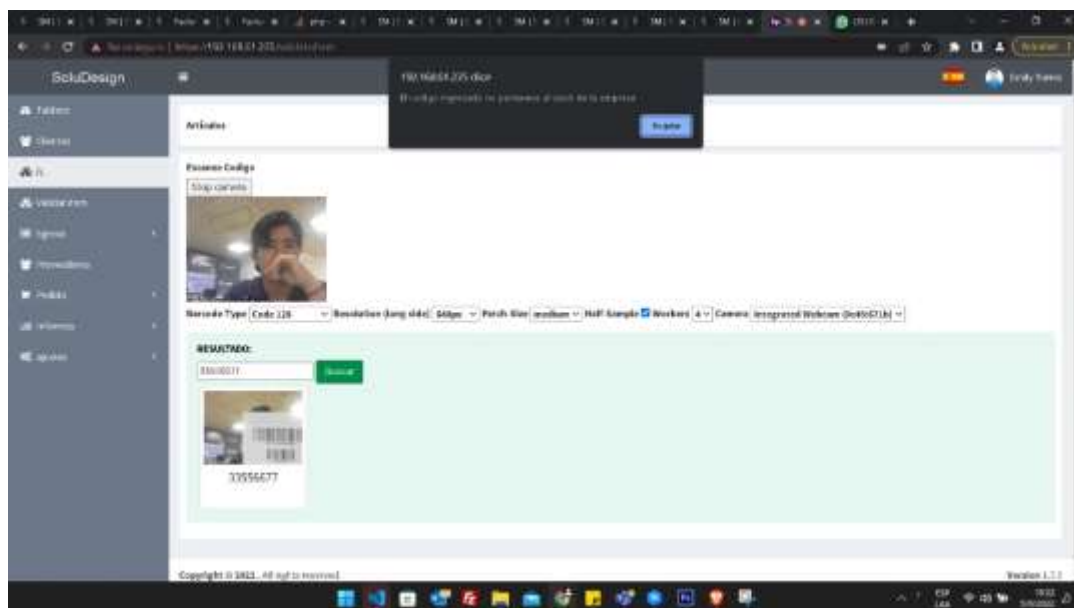


Figura Módulo Validar Ítems del sistema de inventario. Alerta de productos no existentes. Elaborado por Torres Neira Emily.

➤ Registros de Egresos

Permite registrar una orden de egreso para realizar la constancia de salida del producto del inventario del sistema.

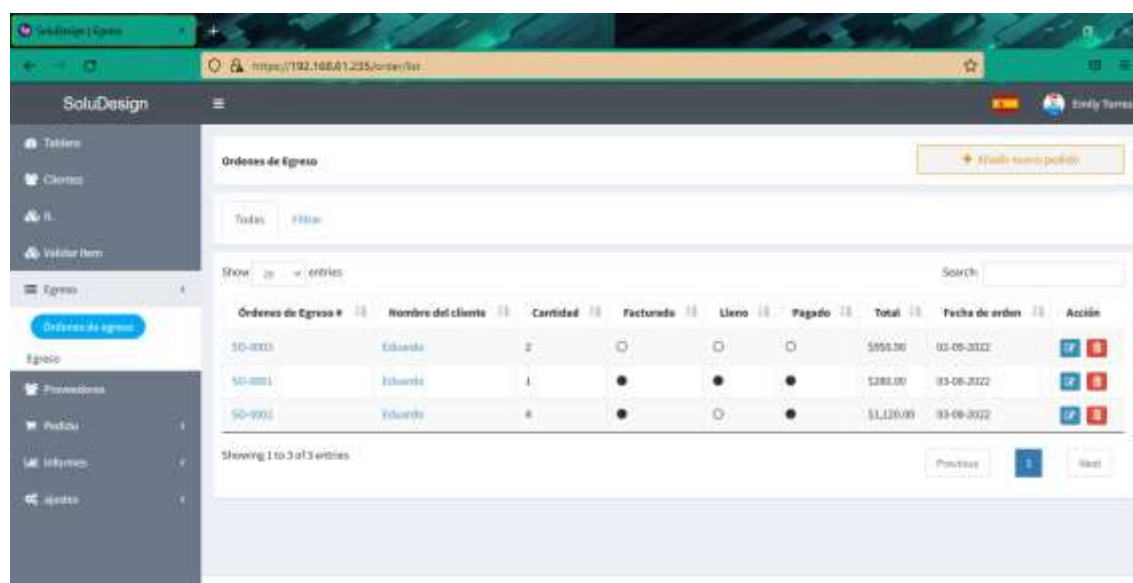


Figura Módulo Egreso del sistema de inventario. Elaborado por Torres Neira Emily.

| Egreso | N° de pedido | Nombre del cliente | Precio total | Monto de pago | Estado | Fecha de la factura | Acción |
|----------|--------------|--------------------|--------------|---------------|--------|---------------------|--------------------|
| 100-0001 | 101-0001 | Eduardo | \$1,120.00 | \$1,120.00 | Pagado | 23-06-2022 | [Iconos de acción] |
| 100-0001 | 101-0001 | Eduardo | \$280.00 | \$280.00 | Pagado | 09-06-2022 | [Iconos de acción] |

Figura Módulo Egreso del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Creación de Proveedores

Muestra el listado de cada uno de los proveedores ingresados en el sistema, permitiendo también agregar un nuevo proveedor en el sistema.

| Nombre del proveedor | Email | Teléfono | Dirección | Estado |
|--------------------------------|-----------------------------|------------|--|--------|
| ADRIANA ESTRELLA REUNGO | ventas.sanantonio@gmail.com | 0989750283 | CALLE ABEL ROMEO CASTILLO Y AV. JUAN TANCA NABENGO, Guayaquil, EC, 593 | Activo |
| GLADHER AGUSTO PERALTA MORALES | gladher_peralta@hotmail.com | 2940337 | Sukari, Guayaquil, Guayaquil, EC, 593 | Activo |
| Maria Taty | maria-taty@hotmail.com | 0960744214 | Alfonsina Lara Etapa , Guayaquil, Guayaquil, EC, 593 | Activo |
| Martin Cobos | martin.cobos16@gmail.com | 0994871389 | Garzota, Guayaquil, Guayaquil, EC, 593 | Activo |
| Juanito | juanito@sotudesign.com | 113445 | Guayaquil, Guayaquil, Activo, EC, 2134 | Activo |

Figura Módulo Proveedores del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Creación de Pedidos

Permite la opción de registrar un pedido realizado a los proveedores que trabajan con la compañía.

| Egreso # | Nombre del proveedor | Cantidad Solicitada | Cantidad Recibida | Total | Fecha de orden | Acción |
|----------|--------------------------------|---------------------|-------------------|------------|----------------|----------|
| PO-0001 | VLADIMIR AUGUSTO PERALTA MORAL | 1 | 1 | \$280.80 | 02-08-2022 | [Iconos] |
| PO-0009 | ANDREA ESTRELLA HIDALGO | 5 | 5 | \$1,540.00 | 17-08-2022 | [Iconos] |
| PO-0012 | VLADIMIR AUGUSTO PERALTA MORAL | 6 | 6 | \$2,418.20 | 16-08-2022 | [Iconos] |
| PO-0008 | VLADIMIR AUGUSTO PERALTA MORAL | 4 | 3 | \$416.64 | 16-08-2022 | [Iconos] |
| PO-0018 | ANDREA ESTRELLA HIDALGO | 7 | 7 | \$1,685.88 | 15-08-2022 | [Iconos] |
| PO-0008 | VLADIMIR AUGUSTO PERALTA MORAL | 18 | 18 | \$483.24 | 11-08-2022 | [Iconos] |
| PO-0007 | VLADIMIR AUGUSTO PERALTA MORAL | 9 | 9 | \$6,345.88 | 10-08-2022 | [Iconos] |
| PO-0013 | Martin Cobos | 10 | 10 | \$2,340.00 | 09-08-2022 | [Iconos] |

Figura Módulo Pedido del sistema de inventario. Elaborado por Torres Neira Emily.

Ingreso de Pedido

Ya que, al realizar un pedido se debe constatar que cada uno de los productos solicitados lleguen a la bodega de la empresa se realiza un ingreso de pedido dentro del sistema.

| Ingreso Pedido # | No. Ingreso # | Nombre del proveedor | Cantidad Recibida | Fecha de Entrega | Acción |
|------------------|---------------|--------------------------------|-------------------|------------------|----------|
| 0001 | PO-0013 | VLADIMIR AUGUSTO PERALTA MORAL | 1 | 02-08-2022 | [Iconos] |
| 0002 | PO-0012 | VLADIMIR AUGUSTO PERALTA MORAL | 6 | 25-08-2022 | [Iconos] |
| 0003 | PO-0011 | VLADIMIR AUGUSTO PERALTA MORAL | 2 | 29-08-2022 | [Iconos] |
| 0004 | PO-0010 | ANDREA ESTRELLA HIDALGO | 7 | 29-08-2022 | [Iconos] |
| 0005 | PO-0009 | ANDREA ESTRELLA HIDALGO | 5 | 29-08-2022 | [Iconos] |
| 0006 | PO-0008 | VLADIMIR AUGUSTO PERALTA MORAL | 18 | 22-08-2022 | [Iconos] |
| 0007 | PO-0007 | VLADIMIR AUGUSTO PERALTA MORAL | 9 | 22-08-2022 | [Iconos] |
| 0008 | PO-0006 | VLADIMIR AUGUSTO PERALTA MORAL | 8 | 22-08-2022 | [Iconos] |

Figura Módulo Pedido del sistema de inventario. Elaborado por Torres Neira Emily.

➤ Informes

Se presentan los diferentes informes que resultan del ingreso de información en el sistema.

- ✓ **Informes existentes a mano** se presenta un informe de cada uno de los productos con los que se cuentan en el stock de la empresa.

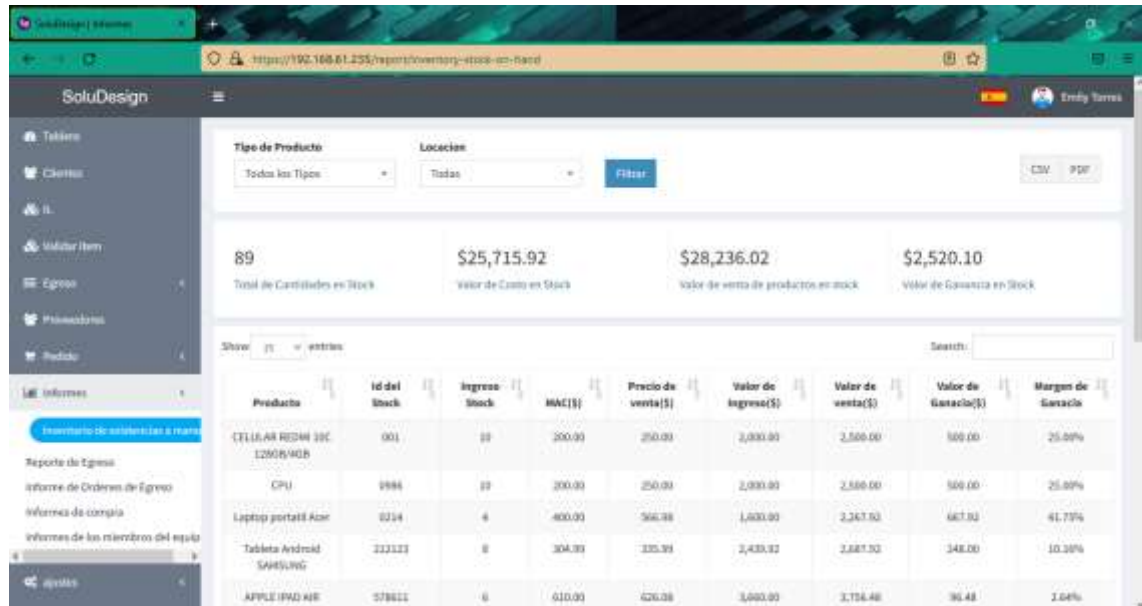


Figura Módulo Informes del sistema de inventario. Elaborado por Torres Neira Emily.

- ✓ **Reporte de Egresos** presenta un listado de cada una de los egresos registrados en el sistema.

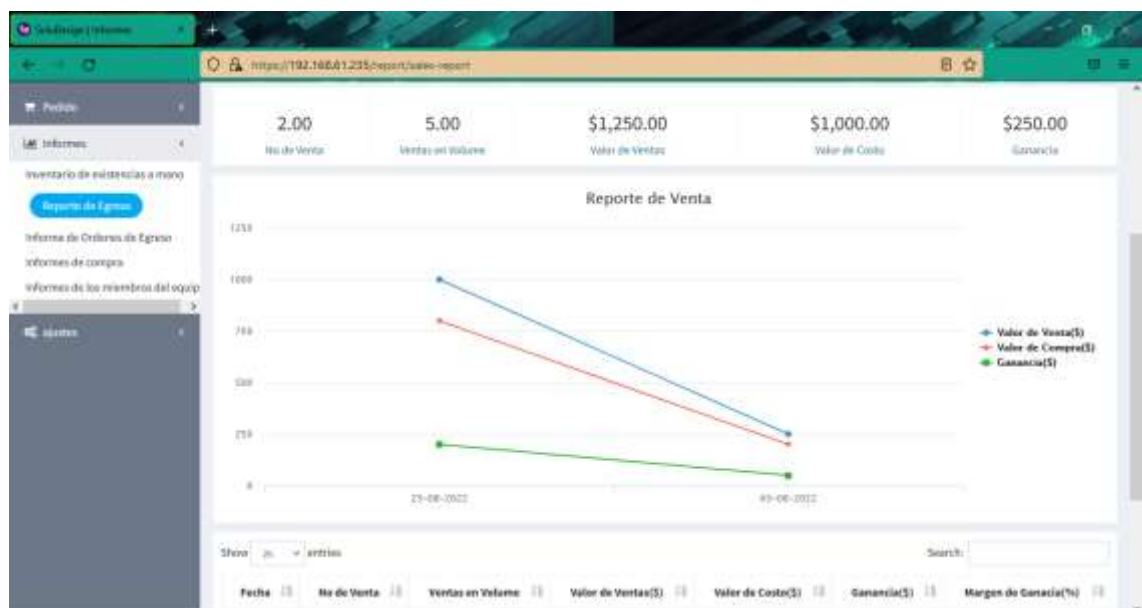


Figura Módulo Informes del sistema de inventario. Elaborado por Torres Neira Emily.

- ✓ **Informes de Ordenes de Egreso** presenta un listado de cada una de las ordenes de egresos registradas en el sistema.

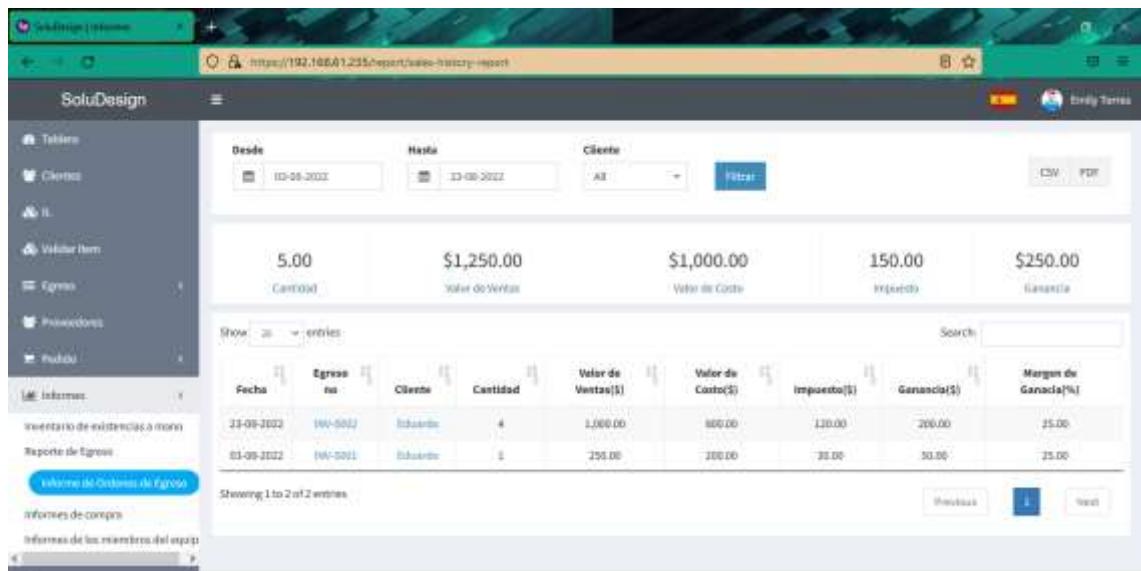


Figura Módulo Informes del sistema de inventario. Elaborado por Torres Neira Emily.

- ✓ **Informes de Ingresos** se presenta un informe de cada uno de los ingresos registrados en el sistema.

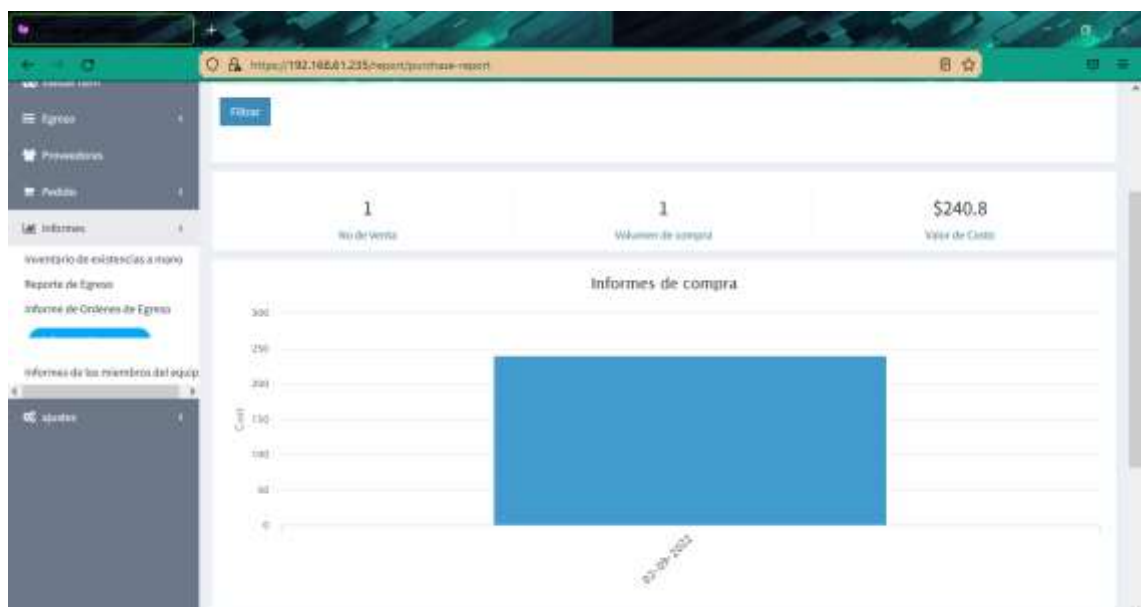


Figura Módulo Informes del sistema de inventario. Elaborado por Torres Neira Emily.

- ✓ **Informes de Miembros del equipo** presenta un listado de cada uno de los usuarios que han sido creados en el sistema.

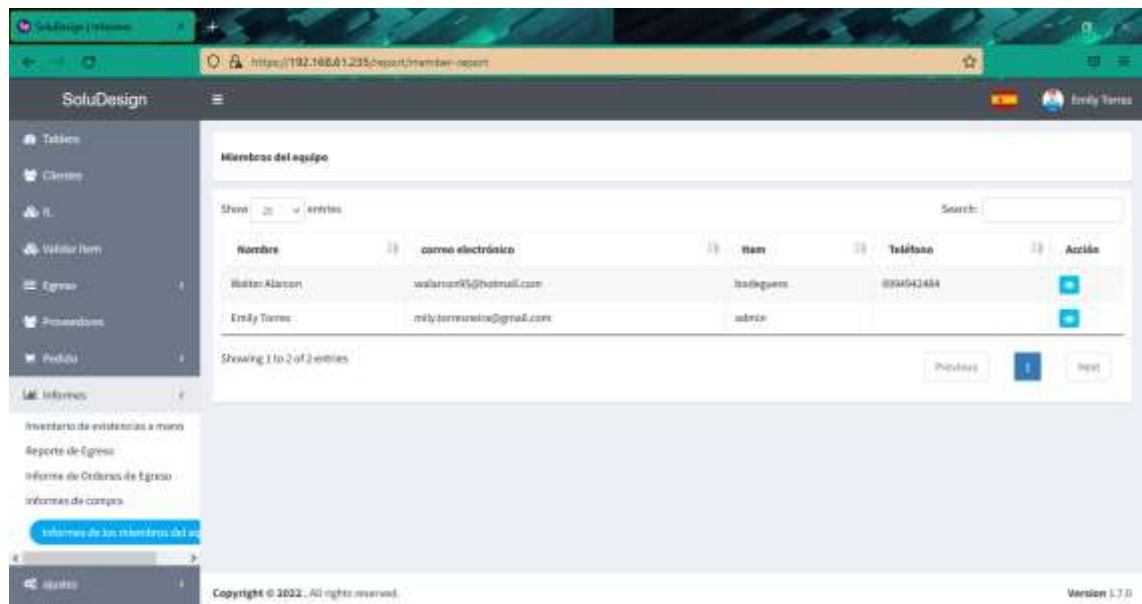


Figura Módulo Informes del sistema de inventario. Elaborado por Torres Neira Emily.

Bibliografía

- Fritz, E., Montejano, G., Garcia, P., & Bast , S. (2016). Integridad de datos. Argentina: In XVIII Workshop de Investigadores en Ciencias de la Computación.
- Agudelo , D., & López, Y. (2018). Dinámica de sistemas en la gestión de inventarios.
- Arias, M. (2017). Aprende Programacion Web con PHP y MySQL: 2ª Edicion. IT campus Academy.
- Cèpedes, N., Paz , J., Fèlix Jimenez, Leonardo Pèrez, & Yaite Pèrez. (2017). La administración de los inventarios en el marco de la administración financiera a corto plazo.
- COIP. (2014). Libro de los Artículos del Código Orgánico Penal. Obtenido de https://tbinternet.ohchr.org/Treaties/CEDAW/Shared%20Documents/ECU/INT_CEDAW_ARL_ECU_18950_S.pdf
- Cordero, M. (2017). Integración Económica Centroamericana: base de datos. Sede Subregional de la CEPAL en México (Estudios e Investigaciones).
- Delgado, S., Rivero , L., & Lince, E. (2019). El uso de software libre en el control de inventarios: caso de estudio. Ciencia Administrativa, 1.
- Gòmez Sebastiàn, & Moraleda, E. (2020). Aproximación a la ingeniería del software. Centro de Estudios Ramon Areces SA.
- Gutièrrez, J. (2016). Integración de un ECM con el sistema de ventas de Amazon. Madrid.
- Herrera, F. (2015). El mito del poder de mercado en el mercado libre. Universidad Peruana de Ciencias Aplicadas (UPC).
- Luna, F. (2019). JavaScript-Aprende a programar en el lenguaje de la web. RedUsers.
- Morales, C. (2021). Estandarización y unificación de sistemas de inventarios en restaurantes. Doctoral dissertation.
- Oddone, M., & Chernobilsky, L. (2016). Utilización de una herramienta informática en el tratamiento de los datos relevados en el estudio. In V Encuentro Latinoamericano de Metodología de las Ciencias Sociales. Centro Interdisciplinario de Metodología de las Ciencias Sociales.

- Pantaleo, G., & Rinaudo, L. (2015). *Ingeniería de software*. Buenos Aires: Alpha Editorial.
- Pantoja, C. (2016). El código de barras como una herramienta para el mejoramiento de los procesos del área de encomiendas y carga de la empresa de transportes Ave Félix, "A.C." de la provincia de Trujillo.
- Peña, O., & Silva, R. (2016). Factores incidentes sobre la gestión de sistemas de inventario en organizaciones venezolanas. *Telos: Revista de Estudios Interdisciplinarios en Ciencias Sociales*, 187-207.
- Rivero, L., Delgado, S., & Ernesto, O. (12 de Julio de 2019). El uso de software libre en el control de inventarios: caso de estudio. *Ciencia Administrativa*, págs. 52-57.
- Rollet, O. (2015). *Aprender a desarrollar un sitio web con PHP y MySQL: ejercicios prácticos y corregidos*. Ediciones ENI.
- Tomala, K. (2019). *Control de inventario y la rentabilidad de la empresa Corpsasso SA, cantón La Libertad, provincia de Santa Elena, año 2018*.