

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO  
DE VENTAS”

Caso: (Victory Motors Bolivia)

**Para optar al título de Licenciatura en Ingeniería de Sistemas**  
**Mención: INFORMÁTICA Y TELECOMUNICACIONES**

**Postulante:** Esther Eva Calle Limachi  
**Tutor Metodológico:** M.Sc. Ing. Enrique Flores Baltazar  
**Tutor Especialista:** Ing. Edwin Mamani Viscarra  
**Tutor Revisor:** Lic. Freddy Salgueiro Trujillo

El Alto – Bolivia

2020

*Dedicatoria:*

*Con mucho cariño a mis padres, por ser la principal inspiración y el pilar fundamental en mi educación, por su incondicional apoyo día tras día y por el tiempo dedicado, ha sido posible este trabajo, gracias a ellos.*

## Agradecimientos

En primer lugar dar las gracias a Dios por darme vida y salud, y la familia que tengo. Agradecer a mis padres por dedicarme su tiempo y atención incondicional para salir adelante y a mis hermanos por los consejos.

A mis tutores por la paciencia y los consejos y guiarme a lo largo del desarrollo de este proyecto que sin ellos no habría sido posible.

Agradecer también a la empresa “Victory Motors Bolivia” especialmente al Sr. Diego Tarqui Gerente General por brindarme información y oportunidad de elaborar el proyecto.

## RESUMEN

Actualmente con el avance de la ciencia y tecnología contar con un sistema automatizado es una necesidad en cualquier entidad, por tal motivo se ha propuesto el sistema de información para el control de inventario de ventas para la empresa Victory Motors.

Los sistemas de información han sido una ayuda importante en el manejo de información, es así que tomando el caso de la empresa Victory Motors, el cual genera datos ofreciendo los productos/vehículos a los clientes realizando las ventas.

Es así que el presente proyecto tiene como objetivo desarrollar e implementar sistema de información para el control de inventario de ventas para la empresa "Victory Motors Bolivia", que le permita administrar la información de manera eficiente, agilizando y mejorando los procesos que se realizan en la empresa.

Para el análisis y diseño de la aplicación web se utilizó la metodología UWE UML (UML-Based Web Engineering) para la construcción y el diseño. UWE es el proceso de desarrollo para aplicaciones web, basada en las técnicas de UML.

Para evaluar la calidad del software se utilizó la norma ISO 9126, para la seguridad a la empresa se utilizó la norma 27000 y para la seguridad de la información se recurrió a la norma 27002 como finalmente para la estimación de costo de producto se usó COCOMO basado en el peso o líneas de código.

La arquitectura MVC (Modelo, Vista, Controlador) ayuda a desarrollo a mantener ordenado los aspectos visuales de a lógica de negocio, PHP es el lenguaje de programación elegido para crear la aplicación del lado del servidor, Laravel 5.7 es un framework que utiliza la arquitectura MVC y MySQL como gestor de Base de Datos.

## ÍNDICE DE CONTENIDO

	Pág.
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES .....	2
1.2.1. ANTECEDENTES DE LA EMPRESA .....	2
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES.....	2
1.3. PLANTEAMIENTO DEL PROBLEMA .....	4
1.3.1. PROBLEMA PRINCIPAL.....	4
1.3.2. PROBLEMAS SECUNDARIOS .....	5
1.4. OBJETIVOS.....	5
1.4.1. OBJETIVO PRINCIPAL.....	5
1.4.2. OBJETIVOS SECUNDARIOS .....	5
1.5. JUSTIFICACIÓN .....	6
1.5.1. JUSTIFICACIÓN TÉCNICA.....	6
1.5.2. JUSTIFICACIÓN ECONÓMICA.....	6
1.5.3. JUSTIFICACIÓN SOCIAL .....	6
1.6. METODOLOGÍA .....	6
1.6.1. MÉTODO DE INGENIERÍA WEB: UWE .....	6
1.6.2. MÉTODOS DE PRUEBAS DE SOFTWARE.....	7
1.6.3. MÉTRICAS DE CALIDAD DE SOFTWARE.....	7
1.6.4. MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE .....	7
1.6.5. SEGURIDAD DEL SOFTWARE .....	8
1.7. HERRAMIENTAS.....	8
1.7.1. LENGUAJE DE PROGRAMACIÓN .....	8
1.7.2. SISTEMA GESTOR DE BASE DE DATOS .....	9
1.7.3. SERVIDOR WEB.....	9
1.8. LÍMITES Y ALCANCES.....	9
1.8.1. LÍMITES .....	9
1.8.2. ALCANCES.....	10
1.9. APORTES.....	11
2.1. DATO.....	12
2.2. INFORMACIÓN.....	12
2.3. SISTEMA .....	12
2.4. SISTEMA DE INFORMACIÓN .....	12

2.5.	INVENTARIO .....	14
2.5.1.	TIPOS DE INVENTARIOS.....	14
2.5.1.1.	INVENTARIOS SEGÚN EL MOMENTO.....	14
2.5.1.2.	CONTROL DE INVENTARIOS.....	15
2.5.1.3.	MÉTODO DE CONTROL DE INVENTARIOS .....	16
2.6.	INGENIERÍA DE REQUERIMIENTOS .....	16
2.6.1.	REQUERIMIENTOS FUNCIONALES .....	17
2.6.2.	REQUERIMIENTOS NO FUNCIONALES.....	17
2.6.3.	CARACTERÍSTICAS DE UN REQUERIMIENTO .....	17
2.7.	INGENIERÍA DE SOFTWARE.....	18
2.8.	INGENIERÍA WEB .....	18
2.9.	METODOLOGÍA PARA EL DESARROLLO DEL SOFTWARE.....	19
2.9.1.	METODOLOGÍA UWE BASADA EN UML (UWE).....	19
2.9.1.1.	CARACTERISTICAS DE UWE. ....	19
2.9.1.2.	FASES DE LA METODOLOGÍA UWE .....	20
2.9.1.3.	FASE DE ANÁLISIS DE REQUISITOS .....	20
2.9.1.4.	FASE DE CONTENIDO .....	21
2.9.1.5.	FASE DE NAVEGACIÓN .....	22
2.9.1.6.	FASE DE DISEÑO DE PRESENTACIÓN .....	23
2.10.	LENGUAJE DE MODELADO UNIFICADO (UML) .....	24
2.10.1.	DIAGRAMAS DE CASOS DE USO .....	24
2.11.	ARQUITECTURA CLIENTE/SERVIDOR .....	27
2.12.	MODELO VISTA CONTROLADOR (MVC) .....	28
2.13.	HERRAMIENTAS DE DESARROLLO DE SOFTWARE .....	29
2.13.1.	LENGUAJE PHP .....	29
2.13.2.	FRAMEWORK LARAVEL.....	30
2.13.2.1.	ORM .....	31
2.13.2.2.	BLADE .....	31
2.13.2.3.	COMPOSER.....	32
2.13.3.	BASE DE DATOS.....	32
2.13.4.	SISTEMA GESTOR DE BASE DE DATOS .....	32
2.13.4.1.	MySQL.....	33
2.13.5.	SERVIDOR WEB APACHE .....	34
2.14.	PRUEBAS DE SOFTWARE.....	34

2.14.1.	PRUEBAS DE CAJA NEGRA.....	34
2.14.2.	PRUEBAS DE CAJA BLANCA .....	35
2.15.	MÉTRICAS DE CALIDAD DE SOFTWARE ESTABLECIDAS POR LA NORMA ISO 9126 .....	36
2.15.1.	FUNCIONALIDAD .....	37
2.15.2.	EFICIENCIA .....	38
2.15.3.	CONFIABILIDAD .....	38
2.15.4.	USABILIDAD .....	39
2.15.5.	MANTENIBILIDAD .....	40
2.15.6.	PORTABILIDAD .....	41
2.16.	PUNTO FUNCIÓN.....	42
2.17.	METODO DE ESTIMACION DE COSTOS DE SOFTWARE .....	46
2.17.1.	MODELO CONSTRUCTIVO DE COSTOS (COCOMO) .....	46
2.17.1.1.	CARACTERÍSTICAS GENERALES .....	46
2.17.1.2.	INCONVENIENTES .....	46
2.17.1.3.	MODELOS DE ESTIMACIÓN .....	47
2.17.1.4.	MODELO BÁSICO.....	48
2.17.1.5.	MODELO INTERMEDIO .....	49
2.17.1.6.	ATRIBUTOS .....	49
2.17.1.7.	MODELO DETALLADO .....	52
2.18.	SISTEMA DE GESTION DE SEGURIDAD DE LA INFORMACION ISO/IEC 27000 .....	52
2.19.	SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACION ISO– 27002 53	
3.1.	ANALISIS DE REQUERIMIENTOS.....	57
3.1.1.	ANALISIS DE LA SITUACIÓN ACTUAL .....	58
3.1.2.	REQUERIMIENTOS DEL SISTEMA.....	58
3.1.2.1.	REQUERIMIENTOS FUNCIONALES .....	59
3.1.2.2.	REQUERIMIENTOS NO FUNCIONALES .....	60
3.1.3.	ACTORES IDENTIFICADOS.....	61
3.1.4.	MODELO DE CASO DE USO DEL SISTEMA. ....	62
3.1.4.1.	DIAGRAMA DE CASO DE USO GENERAL DEL SISTEMA. ....	62
3.1.4.2.	DESCRIPCION DEL CASO DE USO GENERAL .....	63
3.1.4.3.	CASO DE USO DEL SISTEMA.....	65
3.2.	MODELO DE CONTENIDO .....	70

3.3.	MODELO DE NAVEGACIÓN .....	71
3.4.	MODELO DE PRESENTACIÓN.....	74
3.5.	MODELO DE IMPLEMENTACIÓN.....	76
3.6.	PRUEBAS DEL SISTEMA.....	88
3.6.1.	PRUEBA DE CAJA NEGRA .....	88
3.6.2.	PRUEBAS DE CAJA BLANCA .....	91
4.1.	MÉTRICAS DE CALIDAD DE SOFTWARE.....	95
4.1.1.	FUNCIONALIDAD .....	95
4.1.2.	USABILIDAD .....	98
4.1.3.	CONFIABILIDAD .....	99
4.1.4.	MANTENIBILIDAD.....	101
4.1.5.	PORTABILIDAD .....	101
4.2.	SISTEMA DE GESTION DE SEGURIDAD DE LA INFORMACION ISO-27002 102	
4.2.1.	SEGURIDAD DE BASE DE DATOS.....	102
4.2.2.	SEGURIDAD LÓGICA.....	102
4.2.3.	SEGURIDAD FÍSICA.....	103
4.2.4.	SEGURIDAD ORGANIZATIVA.....	103
4.2.5.	SEGURIDAD DEL SISTEMA.....	103
4.2.6.	SEGURIDAD DE LA APLICACIÓN.....	104
5.1.	ANÁLISIS DE COSTOS.....	105
5.1.1.	RESUMEN DEL ANALISIS DE COSTO .....	108
6.1.	CONCLUSIONES.....	109
6.1.1.	TOMANDO LOS OBJETIVOS .....	109
6.1.1.1.	TOMANDO EL OBJETIVO PRINCIPAL .....	109
6.1.1.2.	TOMANDO LOS OBJETIVOS SECUNDARIOS .....	109
6.2.	RECOMENDACIONES .....	110
	Bibliografía.....	111

## ÍNDICE DE FIGURAS

	Pág.
<b>Figura N° 2.1.</b> Capas de la ingeniería de software .....	18
<b>Figura N° 2. 2.</b> Capas de la ingeniería de software .....	20
<b>Figura N° 2. 3.</b> Modelo de caso de Uso.....	21
<b>Figura N° 2. 4.</b> Diagrama de Contenido.....	21
<b>Figura N° 2. 5.</b> Diagrama de navegación UML .....	22
<b>Figura N° 2. 6.</b> Modelo de Presentación.....	23
<b>Figura N° 2. 7.</b> Sistema de Casos de Uso .....	24
<b>Figura N° 2. 8.</b> Representación gráfica de caso de uso.....	25
<b>Figura N° 2. 9.</b> Actor que inicia el caso de uso .....	25
<b>Figura N° 2. 10.</b> Características de los actores .....	26
<b>Figura N° 2. 11.</b> Ejemplo de Casos de Uso .....	26
<b>Figura N° 2. 12.</b> Modelo Cliente-Servidor .....	28
<b>Figura N° 2. 13.</b> Modelo Cliente-Servidor .....	28
<b>Figura N° 2. 14.</b> Gestor de Dependencias de PHP.....	32
<b>Figura N° 2. 15.</b> Norma ISO/IEC 9126 .....	36
<b>Figura N° 2. 16.</b> Cálculo de puntos de función. ....	43
<b>Figura N° 2. 17.</b> Escala de Valores .....	44
<b>Figura N° 2. 18.</b> Proceso de cálculo de puntos de función. ....	45
<b>Figura N° 3. 1.</b> Organigrama de la empresa .....	57
<b>Figura N° 3. 2.</b> Diagrama de casos de uso General .....	62
<b>Figura N° 3. 3.</b> Diagrama de casos de uso Cliente.....	65
<b>Figura N° 3. 4.</b> Diagrama de casos de uso Vendedor.....	66
<b>Figura N° 3. 5.</b> Diagrama de casos de uso Almacén .....	68
<b>Figura N° 3. 6.</b> Diagrama de casos de uso Gestionar Usuario. ....	69
<b>Figura N° 3. 7.</b> Diagrama de contenido del Sistema.....	71
<b>Figura N° 3. 8.</b> Diagrama de Navegación del Sistema Admin.....	72
<b>Figura N° 3. 9.</b> Diagrama de Navegación del Sistema Vendedor .....	72
<b>Figura N° 3. 10.</b> Diagrama de Navegación del Sistema Almacén .....	73
<b>Figura N° 3. 11.</b> Diagrama de Actividad Autenticación Usuario .....	73
<b>Figura N° 3. 12.</b> Modelo de Autenticación .....	74

<b>Figura N° 3. 13.</b>	<b>Modelo de presentación interface Admin.....</b>	<b>75</b>
<b>Figura N° 3. 14.</b>	<b>Modelo de Presentación Nueva Venta .....</b>	<b>75</b>
<b>Figura N° 3. 15.</b>	<b>Interface de Autenticación de Usuarios. ....</b>	<b>76</b>
<b>Figura N° 3. 16.</b>	<b>Pantalla Interfaz de Menú.....</b>	<b>77</b>
<b>Figura N° 3. 17.</b>	<b>Interfaz del Listado de Menú del Sistema .....</b>	<b>77</b>
<b>Figura N° 3. 18.</b>	<b>Interfaz Perfil de la Empresa .....</b>	<b>78</b>
<b>Figura N° 3. 19.</b>	<b>Interfaz Permisos a los Usuarios .....</b>	<b>78</b>
<b>Figura N° 3. 20.</b>	<b>Interfaz Rol y Categorías de Usuarios .....</b>	<b>78</b>
<b>Figura N° 3. 21.</b>	<b>Interfaz Nuevo Rol/Categoría .....</b>	<b>79</b>
<b>Figura N° 3. 22.</b>	<b>Interfaz Almacén .....</b>	<b>79</b>
<b>Figura N° 3. 23.</b>	<b>Interfaz Nuevo Almacén .....</b>	<b>79</b>
<b>Figura N° 3. 24.</b>	<b>Interfaz Usuario .....</b>	<b>80</b>
<b>Figura N° 3. 25.</b>	<b>Interfaz Nuevo Usuario.....</b>	<b>80</b>
<b>Figura N° 3. 26.</b>	<b>Interfaz Persona .....</b>	<b>81</b>
<b>Figura N° 3. 27.</b>	<b>Interfaz Nueva Persona .....</b>	<b>81</b>
<b>Figura N° 3. 29.</b>	<b>Interfaz Nuevo Grupo .....</b>	<b>82</b>
<b>Figura N° 3. 30.</b>	<b>Interfaz Cliente .....</b>	<b>83</b>
<b>Figura N° 3. 31.</b>	<b>Interfaz Nuevo Cliente .....</b>	<b>83</b>
<b>Figura N° 3. 32.</b>	<b>Interfaz Producto .....</b>	<b>84</b>
<b>Figura N° 3. 33.</b>	<b>Interfaz Nuevo Producto .....</b>	<b>84</b>
<b>Figura N° 3. 34.</b>	<b>Interfaz Categoría.....</b>	<b>85</b>
<b>Figura N° 3. 35.</b>	<b>Interfaz Nueva Categoría .....</b>	<b>85</b>
<b>Figura N° 3. 36.</b>	<b>Interfaz Ventas .....</b>	<b>86</b>
<b>Figura N° 3. 37.</b>	<b>Interfaz Nueva Venta.....</b>	<b>86</b>
<b>Figura N° 3. 38.</b>	<b>Interfaz Agregar Producto a la Nueva Venta .....</b>	<b>87</b>
<b>Figura N° 3. 39.</b>	<b>Interfaz Stock .....</b>	<b>87</b>
<b>Figura N° 3. 40.</b>	<b>Interfaz Nuevo Stock .....</b>	<b>87</b>
<b>Figura N° 3. 41.</b>	<b>Gráfico general.....</b>	<b>90</b>
<b>Figura N° 3. 42.</b>	<b>Diagrama de uso general .....</b>	<b>91</b>
<b>Figura N° 3. 43.</b>	<b>Diagrama de Flujo General .....</b>	<b>92</b>
<b>Figura N° 3. 44.</b>	<b>Grafo de Flujo .....</b>	<b>93</b>

## ÍNDICE DE TABLAS

	Pág.
<b>Tabla N° 2. 1.</b> Estructura de Navegación.....	22
<b>Tabla N° 2. 2.</b> Modelo de Presentación .....	23
<b>Tabla N° 2. 3.</b> Constantes para el cálculo de costes Modelo Básico.....	48
<b>Tabla N° 2. 4.</b> Constantes para el cálculo de costes Modelo Intermedio .....	49
<b>Tabla N° 2. 5.</b> Atributos para el cálculo de costes.....	51
<b>Tabla N° 3. 1.</b> Categoría de Funciones.....	58
<b>Tabla N° 3. 2.</b> Requerimientos Funcionales.....	59
<b>Tabla N° 3. 3.</b> Requerimientos No Funcionales .....	60
<b>Tabla N° 3. 4.</b> Especificación de los actores de Caso de Uso.....	63
<b>Tabla N° 3. 5.</b> Caso de Uso Gestionar Productos.....	63
<b>Tabla N° 3. 6.</b> Caso de Uso Gestionar Usuarios.....	64
<b>Tabla N° 3. 7.</b> Caso de Uso Consultar Almacén .....	64
<b>Tabla N° 3. 8.</b> Caso de Uso Gestionar las Ventas .....	64
<b>Tabla N° 3. 9.</b> Especificación de los casos de uso Cliente.....	65
<b>Tabla N° 3. 10.</b> Especificación de los casos de uso Vendedor. ....	67
<b>Tabla N° 3. 11.</b> Especificación de los casos de uso Almacén .....	68
<b>Tabla N° 3. 12.</b> Especificación de los casos de uso Gestionar usuario.....	69
<b>Tabla N° 3. 13.</b> Especificación de los casos de uso Generar reportes.....	70
<b>Tabla N° 3. 14.</b> Interfaz del Registro de un Nuevo Producto.....	88
<b>Tabla N° 3. 15.</b> Interfaz de la Nueva Venta.....	89
<b>Tabla N° 4. 1.</b> Factores de Ponderación.....	95
<b>Tabla N° 4. 2.</b> Valores de ajuste de Complejidad.....	96
<b>Tabla N° 4. 3.</b> Escala de Punto Función .....	97
<b>Tabla N° 4. 4.</b> Ajuste de preguntas .....	99
<b>Tabla N° 4. 5.</b> Gestión de Comunicación y Operaciones .....	102



**CAPÍTULO I**  
**MARCO PRELIMINAR**

## **1.1. INTRODUCCIÓN**

Los continuos avances de la informática y las telecomunicaciones, además del crecimiento global en la web, han ayudado a la difusión de la información alrededor de todo el mundo y a su acceso de manera instantánea, provocando que la información se convierta en una de las herramientas más importantes para el crecimiento de cualquier organización en el mundo. Las tecnologías de la información, actualmente son elementos fundamentales para la superación y desarrollo de un país. Por eso, los países desarrollados basan su crecimiento en la aplicación y la programación estratégica de las herramientas computacionales y han definido políticas que los inducirán a su permanencia en el dinamismo mundial de los próximos años.

Actualmente la empresa Victory Motors Bolivia, se dedica a ofrecer a los clientes, vehículos de distintas marcas, que a su vez son productos importados, y que todo el proceso de la información es realizada manualmente. Como ser el proceso de la venta y el registro de los productos. Tarea que a su vez se vuelve tediosa y consume parte del tiempo que podría aprovecharse en actividades de actualización y el control de la información, además de no contar con un sistema de inventarios que mantenga actualizadas las cantidades de información.

Con el objetivo de mejorar aspectos como la venta y el ingreso y/o actualización de los nuevos productos que llegan a la empresa, entre otros, se elaborará el sistema de información para el control de inventarios de ventas.

Para la operatividad de desarrollar el sistema de información de control de inventarios, se utilizara la siguiente metodología UWE UML (UML-Based Web Engineering) y las herramientas de desarrollo PHP, y para la gestión de base de datos se opta por MySQL y como servidor Xampp.

Con el fin de obtener resultados satisfactorios como para que los usuarios puedan realizar sus tareas, inventariar los productos, entre otros. Es necesario desarrollar el sistema de información web que faciliten la búsqueda y recuperación datos de los productos, aplicando la tecnología web.

## **1.2. ANTECEDENTES**

### **1.2.1. ANTECEDENTES DE LA EMPRESA**

La empresa “Victory Motors Bolivia” fue fundada en el año 2014, el cual está ubicada en la avenida Juan Pablo II entre cruz papal y el TAM de la ciudad de El Alto, perteneciendo a “Red Carga S.R.L” está dedicada a ofrecer a los clientes, vehículos de distintas marcas, que a su vez son productos importados, aspirando ser líderes en el mercado con el objetivo de ser reconocidos a nivel nacional, cuenta con más de cinco años de experiencia en este rubro, motivo por el cual se ha mantenido considerablemente las ventas, la forma de llevar el control del registro del producto que entran al almacén, así como el inventario de dichos productos son elaborados manualmente y las ventas son realizadas en hojas de cálculo.

#### **Misión.-**

Ser la empresa líder en la comercialización de vehículos y prestación de servicios post ventas, brindando soluciones a la medida de nuestros clientes. Ser reconocida por la calidad humana y profesional de nuestra gente, tanto por clientes.

#### **Visión.-**

Ser la mejor opción del mercado en venta de vehículos y servicios posteriores.

### **1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES**

#### **Proyectos similares internacional:**

[Sánchez López, Maricela, 2011]

**Sistema de Información para el Control de Inventarios del Almacén del Its.** El objetivo general fue el desarrollo Diseñar, desarrollar e implementar un sistema de información que permita llevar el registro de artículos, el control de las entradas y

salidas de los mismos, así como todas aquellas actividades requeridas en el almacén del Instituto Tecnológico de Saltillo, la Universidad Autónoma del Noreste, otros Institutos Tecnológicos generalmente tienen sistemas de información para llevar el control de inventarios del almacén general que suministra toda clase de artículos de equipo de cómputo, artículos de oficina, limpieza, entre otros. En el caso del ITS la forma de operar y llevar control de los artículos que entran y salen del almacén, así como del inventario de dichos artículos se tienen en formatos en Word y Excel donde se registran todos los eventos de requisiciones de materiales, de los diferentes departamentos. Se tiene una bitácora interna del registro de artículos ordenados por fechas, y por el folio de cada una de las facturas con la descripción de los artículos que se tienen en el inventario de dicho almacén.

Universidad: [Instituto Tecnológico de Aguascalientes, México].

[Yudy Esmeralda Silva Rodriguez, Danna Andrea Riaño Borrero Julio de 2012]

**Prototipo de Control de Inventarios de Producción de La Empresa Calzado Zanelli.** El objetivo general fue desarrollar prototipo funcional Web para gestionar el manejo de los costos de producción, la programación de producción y la gestión del inventario para la empresa Calzado Zanelli. Identificando las deficiencias que existían en el registro manual que se realizaba para el manejo de costos de producción, inventario de productos y comportamiento de las ventas estableciendo los requerimientos de acuerdo a las necesidades del cliente. Modelando el aplicativo a través de UML, diseñando modelo de datos, definir los módulos y perfiles de usuario de la aplicación Web, analizar y diseñar los módulos que se implementaran. Universidad: [Universidad Ean Facultad de Ingeniería Programa de Ingeniería de Sistemas Bogotá D.C.].

### **Proyectos similares nacionales:**

[Jesus Hermogenes Valle Quispe, marzo de 2009]

**Sistema de Control y Seguimiento de Inventario de Fármacos.** El objetivo general fue desarrollar una alternativa de solución a estos problemas de crecimiento a través del desarrollo de un sistema de control y seguimiento de inventario de fármacos esta herramienta permitirá un adecuado control del

inventario que cumpla con todos los requerimientos cambiantes de la clínica y de los usuarios a través de registros de transacciones, informes que apoyen a los inventarios físicos, elaboración de reportes de clientes, proveedores, ítems y otros que van de acuerdo a las exigencias de las áreas involucradas.

Universidad: [Universidad Mayor de San Andrés].

[Sarco Mendoza, Monica, 2017]

**Sistema de control de compra, venta e inventarios Caso: Empresa Protec.** El presente proyecto tiene como objetivo implementar un sistema para optimizar el control eficiente de ventas, compras e inventario de la empresa descrita anteriormente, de forma que los gerentes hagan un control adecuado sobre estos procesos. El proyecto fue desarrollado utilizando la metodología ágil XP (programación extrema) en sus distintas fases como son: planificación, diseño, desarrollo y pruebas. El diseño de los procesos e interfaz fue realizado bajo la metodología de modelado Webml que cuenta con diversos esquemas para la representación gráfica de estos procesos. La calidad del sistema se la realizó bajo el estándar ISO 9126 que evalúa aspectos como usabilidad, funcionalidad, confiabilidad, mantenibilidad y portabilidad, proporcionando una evaluación tras la implementación del sistema. Por último, los objetivos del proyecto han sido alcanzados satisfactoriamente de manera que se implementó un sistema que permita tener un control productivo a través de las compras, ventas e inventarios.

Universidad: [Universidad Mayor de San Andrés].

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

#### **1.3.1. PROBLEMA PRINCIPAL**

La empresa “Victory Motors Bolivia” realiza el control y seguimiento acerca de las ventas, almacenes, registro a los clientes, control de stock solo con la ayuda de hojas de cálculo o planillas estos procesos son elaborados manualmente, lo que genera problemas como perdida y desactualización de la información, obteniendo demora en la búsqueda y elaboración de informes.

### **1.3.2. PROBLEMAS SECUNDARIOS**

- ✓ El registro manual de la información de ventas, genera dificultad y pérdida de tiempo.
- ✓ El proceso de la venta es lento, porque es manual, generando retraso y en ocasiones errores de cálculo, para sacar el total de la venta.
- ✓ Presentando demora en la búsqueda de información acerca del stock y existencia del producto.
- ✓ Falta de control de los responsables, lo que ocasiona una mala administración de la información.
- ✓ Retraso en la entrega de informe o reportes con la información de los productos existentes en el almacén.

Por lo tanto nos planteamos la siguiente interrogante.

¿De qué manera, coadyuvaría el sistema de información para el control de inventarios de ventas, en el manejo de información para la venta de productos en la empresa Victory Motors Bolivia?

### **1.4. OBJETIVOS**

#### **1.4.1. OBJETIVO PRINCIPAL**

Desarrollar e implementar el sistema de información para el control de inventario de venta, como una herramienta en la administración y la optimización de la información en la empresa Victory Motors Bolivia de la ciudad de El Alto.

#### **1.4.2. OBJETIVOS SECUNDARIOS**

- ✓ Realizar un diagnóstico de la situación actual de la empresa para identificar que procesos se deben automatizar.
- ✓ Diseñar el sistema de inventario para realizar el registro de toda la información de las ventas.
- ✓ Optimizar el tiempo de la búsqueda de los datos existentes para mejorar la organización de la información.

- ✓ Crear niveles de usuario para la administración del sistema.
- ✓ Realizar el modulo para generar reportes con información sobre los productos en almacén.

## **1.5. JUSTIFICACIÓN**

### **1.5.1. JUSTIFICACIÓN TÉCNICA**

Se justifica técnicamente porque actualmente la empresa Victory Motos Bolivia cuenta con la infraestructura y recursos como el hardware y software para poder implementar el sistema de control de inventario de ventas lo que hace que el proyecto sea más factible.

### **1.5.2. JUSTIFICACIÓN ECONÓMICA**

Con el sistema de información de control de inventario, permitirá que la empresa pueda optimizar en sus principales tareas, permitiendo al personal de la empresa un tiempo de ahorro considerable realizando consultas desde su oficina minimizando el material de escritorio.

También se justifica económicamente por el desarrollo del sistema porque se utilizarán herramientas libres como, lenguaje de programación PHP, Motor de Base de Datos MySQL y el Servidor Web Apache.

### **1.5.3. JUSTIFICACIÓN SOCIAL**

El sistema de información beneficiara a la empresa tanto al personal directivo y administrativo, permitiendo búsquedas de la información rápida y oportuna para una buena toma de decisiones, el sistema también formara parte del beneficio para los clientes, ya que a través de ello pueden observar los productos.

## **1.6. METODOLOGÍA**

### **1.6.1. MÉTODO DE INGENIERÍA WEB: UWE**

UWE, desarrollado por Nora Koch, es un método de desarrollo de aplicaciones Web basado en UML estándar, basada en el entorno Orientada a Objetos.

## **UWE UML (UML Based Web Engineering)**

Además de la notación UML, utilizan el proceso unificado de desarrollo de software (Rational Unified Process o RUP) como metodología para realizar aplicaciones web, por lo cual el proceso es iterativo e incremental. La estrategia de diseño UWE se basa en modelos que se construyen durante la fase de análisis, principalmente el modelo conceptual y el modelo de procesos. (Valle, 2009).

### **1.6.2. MÉTODOS DE PRUEBAS DE SOFTWARE**

Las pruebas son de gran importancia en la garantía de calidad de software, los objetivos para realizar una prueba son:

- ✓ Detectar un error en el sistema.
- ✓ Tener un buen caso de prueba.
- ✓ Descubrir un error no descubierto antes.

Los métodos de prueba del software de caja negra y caja blanca tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores. (Torrez, 2010).

### **1.6.3. MÉTRICAS DE CALIDAD DE SOFTWARE**

Las métricas de calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software, con el propósito de lograr una mayor confiabilidad, mantenibilidad, facilidad y prueba de uso. (Filoltrani, 2007).

### **1.6.4. MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE**

Método de estimación de costos COCOMO nivel intermedio para estimar el costo de software.

El modelo constructivo de costos (COCOMO), por su acrónimo en inglés (COConstructive COSt MOdel) es un modelo matemático de base empírica utilizada para estimación de costos de software. (Boehm, 1981).

#### **1.6.5. SEGURIDAD DEL SOFTWARE**

La seguridad del software se enfoca en la identificación y valoración de los riesgos potenciales que pueden afectar al software negativamente y hacer que falle todo un sistema. Si los riesgos pueden identificarse tempranamente en el proceso de ingeniería del software, puedan especificarse características de diseño del software que eliminarán o controlarán los riesgos potenciales.

#### **1.7. HERRAMIENTAS**

Entre los lenguajes de programación y gestores de base de datos más utilizados para el desarrollo de proyectos son los siguientes:

##### **1.7.1. LENGUAJE DE PROGRAMACIÓN**

Como lenguaje de programación se utilizará PHP porque satisface con los requerimientos del sistema.

**PHP** se caracteriza por ser un lenguaje gratuito y multiplataforma. Además de su posibilidad de acceso a muchos tipos de bases de datos, también es importante destacar su capacidad de crear páginas dinámicas, así como la posibilidad de separar el diseño del contenido de una web. En PHP también encontramos frameworks potentes y muy usados como Laravel o Symfony. Una de las ventajas de PHP es su facilidad de uso, otra gran ventaja es su disponibilidad, ya que lo encontramos en la mayoría de alojamientos web de todo tipo. (Lerdorf, 2018).

##### **Framework Laravel**

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP evitando el código espagueti. Laravel tiene como objetivo ser un framework

que permita el uso de una sintaxis, muy potente y tiene una interfaz elegante y divertida de usar. (Outwel, 2011).

### **1.7.2. SISTEMA GESTOR DE BASE DE DATOS**

Para sistema gestor de base de datos se utilizará MYSQL, que maneja grandes volúmenes de datos.

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto para entornos de desarrollo web. (COBO, 2005).

### **1.7.3. SERVIDOR WEB**

Como servidor web se utilizará Apache que es multiplataforma que tiene un soporte intenso carga de trabajo. **XAMPP**, es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestionamiento de bases de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script: php y perl. (Raffino, 2019).

## **1.8. LÍMITES Y ALCANCES**

### **1.8.1. LÍMITES**

En la actualidad muchas empresas enfrentan problemas de diversa naturaleza con distintos grados de complejidad, que no necesariamente pueden ser solucionados a través de un sistema informático, el presente sistema de inventario se limitara a:

- ✓ La actualización y restricción de los datos será solo por el personal autorizado.
- ✓ Personas no autorizadas no tendrán acceso al sistema.
- ✓ El sistema de información para el control de inventario de venta no está proyectado para realizar transacciones monetarias o pagos vía web.
- ✓ El sistema se concentra en funciones específicas como ser: venta y almacén.

- ✓ Automatizar los datos a partir de la implementación del sistema de información para el control de inventario, se aclara que solo será respecto al área de ventas.

### **1.8.2. ALCANCES**

Para dar cumplimiento a los requerimientos de la empresa se realizará el sistema de información que permitirá al personal tener un acceso a los siguientes módulos que se detalla a continuación:

- ✓ Inicio
  - ❖ Contendrá estadísticas y el movimiento acerca de las ventas.
- ✓ El módulo de administrador.
  - Submódulos.
    - ❖ Menú contenido de los Submodulos.
    - ❖ Permisos, donde se asigna un permiso al usuario.
    - ❖ Nivel de acceso, donde se le asigna el nivel de accesibilidad al usuario con el rol/categoría.
    - ❖ Almacén, donde ingresan la cantidad de productos/vehículos y son registrados.
    - ❖ Empresa, donde se registra a la empresa.
    - ❖ Usuario, registro de usuarios para interactuar con el sistema.
- ✓ El módulo Grupo Menú.
  - Submódulos.
    - ❖ Persona, donde se registra a la persona
    - ❖ Grupo Menú, donde se registra al grupo de menú para los módulos.
- ✓ El módulo Productos.
  - Submódulos.
    - ❖ Cliente, donde se realiza el registro del nuevo cliente
    - ❖ Producto, donde son registrados los productos/vehículos para agregar a almacenes y stock
    - ❖ Categoría, ingresando la nueva categoría del producto/vehículo

- ❖ Ventas, modulo troncal para la venta
- ❖ Stock, se registra el stock
- ✓ El módulo de reportes
  - Submódulos.
    - ❖ Repostes nos muestra los reportes

## **1.9. APORTES**

El aporte para la empresa Victory Motors es el desarrollo e implementación del sistema de información para el control de inventario de venta, la cual será de gran ayuda para la empresa tanto para su administración como para su crecimiento empresarial en el área de venta y almacén.

También se brinda con este proyecto será automatizar sus procesos rutinarios, minimizar y optimizar tiempos de ejecución generando información que coadyuve a la buena toma de decisiones



## **CAPÍTULO II**

### **MARCO TEÓRICO**

### **2.1. DATO**

Un dato es el elemento primario de la información conformado por símbolos (letras y/o números) o variable (cuantitativa y/o cualitativa) que reunidos pueden tener significación. Solo o aislado el dato no posee relevancia, pero utilizado en las premisas de un razonamiento puede llevarnos a una conclusión.

### **2.2. INFORMACIÓN**

Información se denomina al conjunto de datos, ya procesados y ordenados para su comprensión, que aportan nuevos conocimientos a un individuo o sistema sobre un asunto, materia, fenómeno o ente determinado. Por otra perspectiva nos indica que la información es un recurso que otorga significado o sentido a la realidad, ya que mediante códigos y conjuntos de datos, de origen a los modelos de pensamientos humano.

### **2.3. SISTEMA**

Sistema es conjunto ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí. Los sistemas reciben (entrada) como ser datos, (proceso) es lo que transforma una entrada en salida, y (salida) es el resultado al procesar la entrada, es decir información. (Bertalanffy, 1940).

### **2.4. SISTEMA DE INFORMACIÓN**

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

**Entrada de Información** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáners, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

**Almacenamiento de información** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

**Procesamiento de Información** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.

**Salida de Información** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. (Peralta, 2008).

Es necesario conocer los conceptos sobre inventarios, control de inventarios y el método, es de mucha importancia para el siguiente proyecto.

## 2.5. INVENTARIO

El inventario es el conjunto de artículos o mercancías que se acumulan en el almacén pendientes de ser utilizados en el proceso productivo o comercializados. Otra definición de inventario vinculada al ámbito económico es la relación ordenada de bienes de una organización.

El concepto inventario o stock resulta muy importante en las empresas con el propósito de que las demandas de los consumidores sean atendidas para que no se vea interrumpido el proceso productivo ante la falta de materias primas. Pueden considerarse como una herramienta reguladora que mantiene el equilibrio entre los flujos reales de entrada y los de salida. (Garcia, 2017).

### 2.5.1. TIPOS DE INVENTARIOS

El inventario puede ser de distintas maneras en función de una serie de parámetros:

- **Inventario de materias primas:** está compuesto por aquellos materiales con los que se fabrican los productos, pero que aún no han sido procesados.
- **Inventario de productos en proceso de fabricación:** son los bienes comprados por las compañías industriales. Su cuantificación se realiza por la cantidad de materiales, gastos de fabricación y mano de obra.
- **Inventario de productos terminados:** los distintos bienes comprados por las compañías industriales, los cuales se transforman con el propósito de ser comercializados como artículos elaborados.
- **Inventario de suministros de fábrica:** los materiales con los que se fabrican los productos, pero que no pueden ser cuantificados con exactitud.

#### 2.5.1.1. INVENTARIOS SEGÚN EL MOMENTO

- **Inventario inicial** Se realiza cuando inicias con las operaciones de la empresa. En términos contables, se refiere al inventario que refleja tu saldo antes de que adquieras inventario adicional o que vendas el existente en un determinado periodo.

- **Inventario final** Se realiza al cierre de un periodo contable, es de vital importancia, ya que está incluido en el balance general de la empresa, por lo que aporta información sobre los recursos financieros disponibles al término de este.

#### **2.5.1.2. CONTROL DE INVENTARIOS**

El control de inventario se refiere a todos los procesos que coadyuvan al suministro, accesibilidad y almacenamiento de productos en la organización para minimizar los tiempos y costos relacionados: es un mecanismo a través del cual, la organización administra de manera eficiente el movimiento y almacenamiento de mercancía, así como el flujo de información y recursos que resultan de ello. Involucra distintos aspectos, pero en términos generales se subdivide en lo correspondiente a gestión y optimización. La gestión se encarga de mantener la productividad en las operaciones relacionadas con la administración del inventario, mientras que la optimización se ocupa de incrementar las ganancias de la empresa provenientes del uso y manejo. (Castro, 2018).

Las empresas dedicadas a la compra y venta de mercancías, por ser ésta su principal función y la que dará origen a todas las restantes operaciones, necesitarán de una constante información resumida y analizada sobre sus inventarios, lo cual obliga a la apertura de una serie de cuentas principales y auxiliares relacionadas con esos controles. Entre estas cuentas podemos nombrar las siguientes:

- Inventario (inicial)
- Compras
- Devoluciones en compra
- Gastos de compras
- Ventas
- Devoluciones en ventas
- Inventario (final).

### 2.5.1.3. MÉTODO DE CONTROL DE INVENTARIOS

Existen diversos métodos para el control de inventarios, los cuales varían de acuerdo con las características y necesidades de cada empresa. No obstante, todos se asemejan en que están orientados a la obtención de utilidades.

Gran parte de las PYMES adolece de una correcta administración de sus inventarios, ya sea por desconocimiento o por percibir esta actividad como un gasto. La realidad es que con un eficiente control y manejo de mercancías se evitan desperdicios, se identifican oportunidades, se conocen los costes de producción y se fijan precios competitivos. Todo esto se refleja en ganancias y crecimiento para la empresa. El método considerable y manejable de la empresa es:

**Método PEPS:** este es el sistema de Primeras Entradas, Primeras Salidas, llamado en inglés FIFO (First in, First Out). Se basa en identificar el precio de coste del stock se basa en el coste que tuvieron al llegar al almacén.

### 2.6. INGENIERÍA DE REQUERIMIENTOS

La ingeniería de requerimientos cumple un papel fundamental en el proceso de producción de software, ya que se enfoca un área fundamental: la definición de lo que se desea producir. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver un problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos (RE).

El término requerimiento no se utiliza de una manera constante en la industria de software. En algunos casos, un requerimiento es simplemente una declaración abstracta de alto nivel de servicio, sirve como una base sólida en el proceso de desarrollo de software, por lo que antes de pasar a tardar los aspectos referentes a la administración adecuada a los requerimientos. (Someville, 2011).

### **2.6.1. REQUERIMIENTOS FUNCIONALES**

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole.

Como también los requerimientos funcionales, relacionados con funciones del negocio, datos que deben ingresarse en las pantallas del sistema (interfaz gráfica), los relacionados con control de acceso o emisión de reportes requeridos por entes reguladores.

### **2.6.2. REQUERIMIENTOS NO FUNCIONALES**

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

Los requerimientos no funcionales se pueden clasificar según su finalidad, como por ejemplo requerimientos de negocio, requerimientos originados en aspectos regulatorios, de seguridad, entre otros.

### **2.6.3. CARACTERÍSTICAS DE UN REQUERIMIENTO**

Un requerimiento debe cumplir ciertos criterios y características:

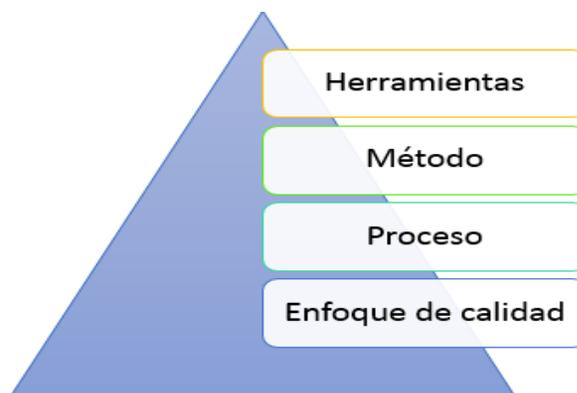
- **Único:** El requerimiento debe poder ser interpretado inequívocamente de una sola manera.
- **Verificable:** Su implementación debe poder ser comprobada. El test debe dar como resultado CORRECTO o INCORRECTO.
- **Claro:** Los requerimientos no deben contener terminología innecesaria. Deben ser establecidos de forma clara y simple.

- **Viable** (realista y posible): El requerimiento debe ser factible según las restricciones actuales de tiempo, dinero y recursos disponibles.
- **Necesario**: Un requerimiento no es necesario si ninguno de los interesados necesita el requerimiento o bien si la retirada de dicho requerimiento no tiene ningún efecto.

## 2.7. INGENIERÍA DE SOFTWARE

La Ingeniería del Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software. (Roger S. Pressman, Ingeniería del software, 2010).

**Figura N° 2.1.** Capas de la ingeniería de software



**Fuente:** (Pressman, Roger S., Ingeniería del Software, 2010).

## 2.8. INGENIERÍA WEB

La ingeniería web es un área que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad.

La ingeniería Web toma prestado muchos de los conceptos y principios básicos de la ingeniería del software, dando importancia a las mismas actividades técnicas y de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque

disciplinado para el desarrollo de un Sistema basado en computadora. (Del Valle Rodriguez, 2015).

## **2.9. METODOLOGÍA PARA EL DESARROLLO DEL SOFTWARE**

Una metodología de software es un enfoque, una manera de interpretar la realidad o la disciplina en cuestión, que en este caso particular correspondería a la Ingeniería de Software. De hecho, la metodología destinada al desarrollo de software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializada. (Gomez, 2017).

### **2.9.1. METODOLOGÍA UWE BASADA EN UML (UWE)**

UML-Based Web Engineering (UWE) es una metodología de desarrollo de aplicaciones web, utilizada en la ingeniería web, dedicado a la sistematización y personalización, es decir realizar sistemas adaptativos. Debemos también destacar las características relevantes del proceso UWE como la utilización del paradigma orientado a objetos, su orientación al usuario.

UWE es un proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, las fases coinciden con las propuestas en el Proceso Unificado de Modelado y UML pero adaptada a la web.

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre una metodología de diseño sistemática, basada en las técnicas de UML. El proceso de desarrollo en cuatro fases principales: la fase de captura de requisitos, la fase de análisis y diseño de contenido, la fase de modelo de navegación por último la fase de implementación. (Minguez Sanz & Garcia Morales, 2007).

#### **2.9.1.1. CARACTERÍSTICAS DE UWE.**

Las principales características en los que se fundamenta UWE son los siguientes:

- Una de las características de UWE es el uso de una notación estándar, para todos los modelos Lenguaje de modelado unificado UML.

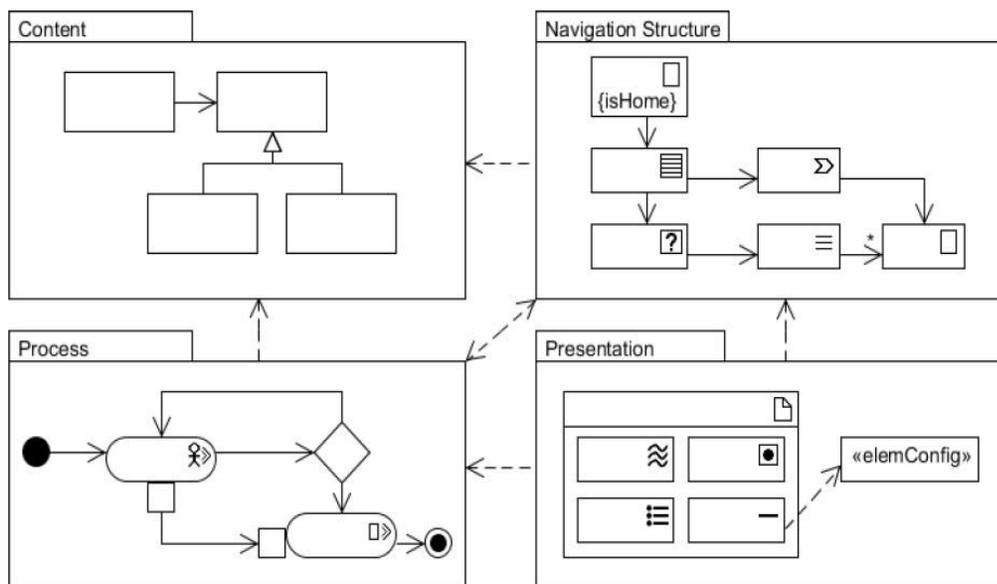
- Definición de métodos: UWE presenta una definición de los pasos para la construcción de los diferentes modelos.
- Especificación de Restricciones: en la metodología UWE, se recomienda el uso de restricciones en su desarrollo.

### 2.9.1.2. FASES DE LA METODOLOGÍA UWE

UWE es una metodología dirigida o enfocada al modelado de aplicaciones Web, ya que está basada estrictamente en UML, esta metodología nos garantiza que sus modelos sean fáciles de entender para los que manejan UML.

En la siguiente figura podemos ver una vista general de UWE, con las fases que tiene como.

**Figura N° 2. 2.** Capas de la ingeniería de software

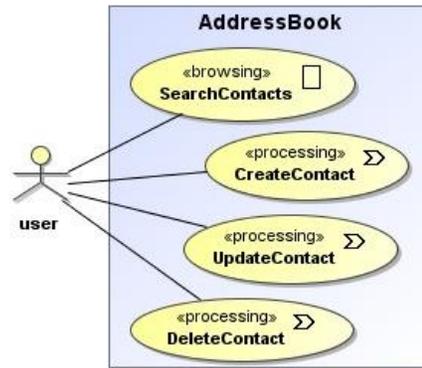


**Fuente:** (UWE-Engineering).

### 2.9.1.3. FASE DE ANÁLISIS DE REQUISITOS

La Fase de Análisis de Requerimientos realiza la captura de los mismos mediante diagramas de casos de uso acompañado de la documentación que detalla.

**Figura N° 2. 3.** Modelo de caso de Uso

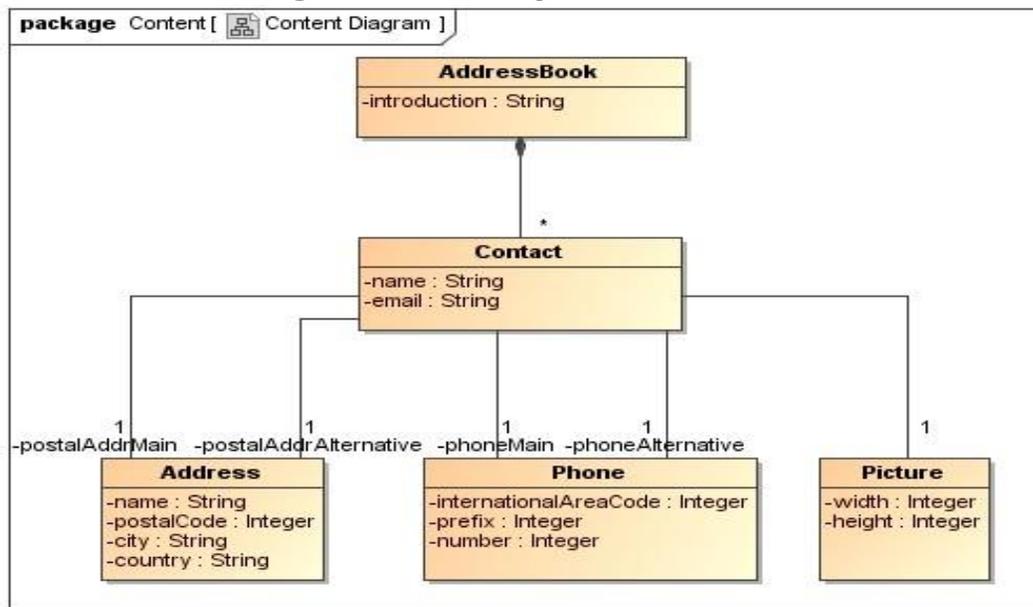


Fuente: (UWE-Engineering).

#### 2.9.1.4. FASE DE CONTENIDO

Caracterizado por un modelo de dominio, que utiliza los requisitos que se detallan en los casos de uso. En esta etapa se representa el dominio del problema con un diagrama de clases de UML, que permiten determinar, métodos y atributos. El propósito de este diagrama es construir un modelo del dominio que intenta no considerar el paseo de la navegación, la presentación y los aspectos de interacción. Aspectos que se analizarán en los pasos respectivos de navegación y presentación de la planificación.

**Figura N° 2. 4.** Diagrama de Contenido



Fuente: (UWE-Engineering).

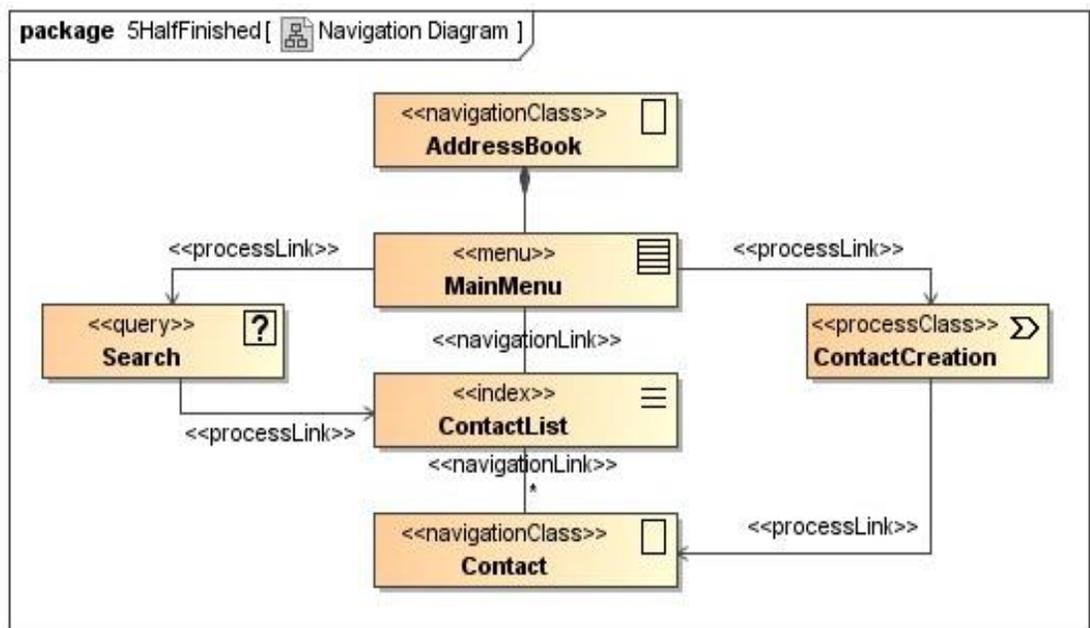
### 2.9.1.5. FASE DE NAVEGACIÓN

Basado en el diagrama de la fase conceptual, donde se especifica los objetos que serán visitados dentro de la aplicación web y la relación entre los mismos.

Su objetivo principal es representar el diseño y estructura de las rutas de navegación al usuario para evitar la desorientación en el proceso de navegación.

Este modelo se destaca en el marco de UWE como el más importante, ya que representa elementos estáticos, a la vez que se pueden incorporar lineamiento semántico de referencia para las funcionalidades dinámicas de una aplicación Web.

**Figura N° 2. 5.** Diagrama de navegación UML



**Fuente:** (UWE-Engineering).

**Tabla N° 2. 1.** Estructura de Navegación

Nombres de estereotipos y sus iconos	
□ clase de navegación	☰ menú
☰ índice	? pregunta
☞ visita guiada	⤷ clase de proceso
☞ nodo externo	

**Fuente:** (UWE-Engineering).

### 2.9.1.6. FASE DE DISEÑO DE PRESENTACIÓN

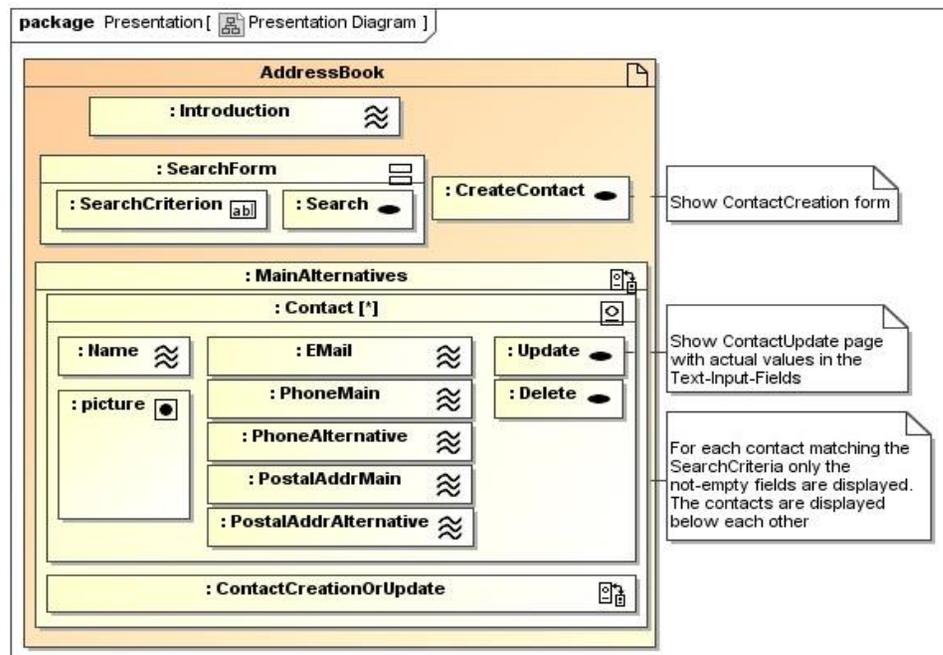
El diagrama de esta fase representa los objetos de navegación y elementos de acceso, por ejemplo en que marco o ventana se encuentra el contenido y que será reemplazado cuando se accione un enlace. En la siguiente imagen podremos observar un ejemplo de un diagrama de presentación mediante UWE.

**Tabla N° 2. 2. Modelo de Presentación**

Nombres de estereotipos y sus iconos	
 grupo de presentación	 página de presentación
 texto	 entrada de texto
 ancla	 fileUpload
 botón	 imagen
 formulario	 componente de cliente
 alternativas de presentación	 selección

Fuente: (UWE-Engineering).

**Figura N° 2. 6. Modelo de Presentación**



Fuente: (UWE-Engineering).

## 2.10. LENGUAJE DE MODELADO UNIFICADO (UML)

Lenguaje Unificado de Modelado “UML”, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Se lo puede definir como un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (Roger S. Pressman, Lenguaje modelao unificado, 2010).

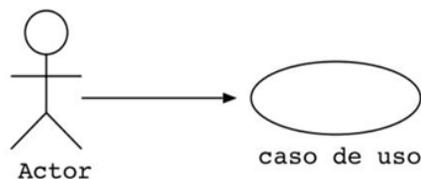
### 2.10.1. DIAGRAMAS DE CASOS DE USO

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

- **Símbolos de los casos de uso**

**Sistema:** El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del Sistema

**Figura N° 2. 7.** Sistema de Casos de Uso



**Fuente:** (Pressman, R., 2010).

**Caso de uso:** Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.

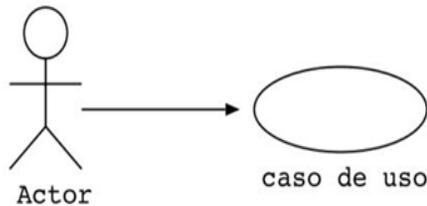
**Figura N° 2. 8.** Representación gráfica de caso de uso



**Fuente:** (Pressman, R., 2010).

**Actor:** Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectoras. Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.

**Figura N° 2. 9.** Actor que inicia el caso de uso



**Fuente:** (Pressman, R., 2010).

**Por ejemplo:** Un actor puede ser un empleado, pero también puede ser un cliente en la tienda de la empresa. Incluso cuando es la misma persona en el mundo real, se representa como dos símbolos distintos en un diagrama de caso de uso, ya que la persona interactúa con el sistema en distintos roles.

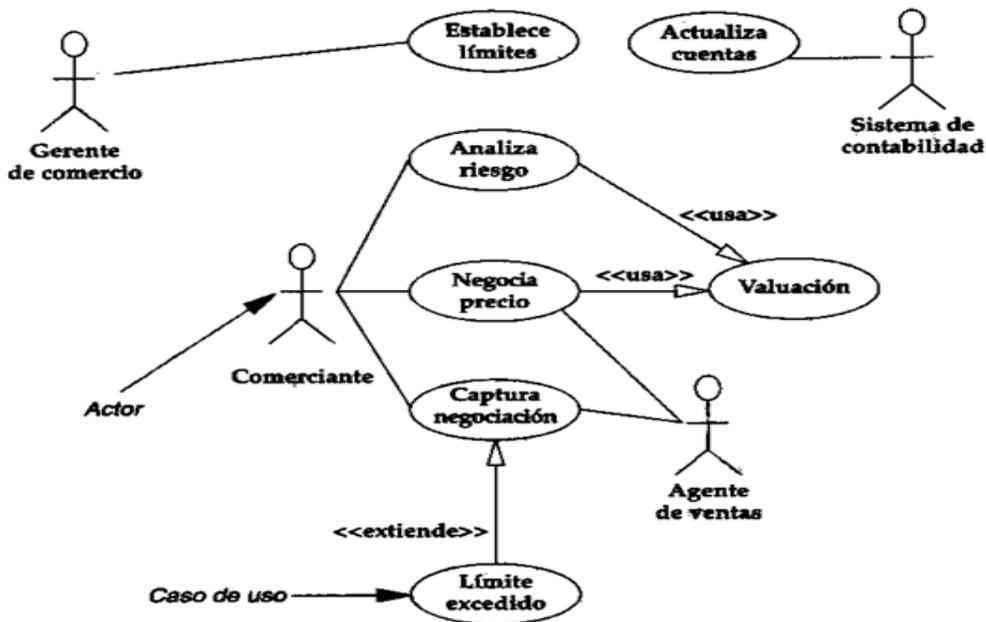
**Figura N° 2. 10.** Características de los actores



**Fuente:** (Pressman, R., 2010).

**Relaciones:** Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas “incluir” o “extender.” Una relación “incluir” indica que un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación “extender” indica opciones alternativas para un cierto caso de uso.

**Figura N° 2. 11.** Ejemplo de Casos de Uso



**Fuente:** (Pressman, R., 2010).

## **2.11. ARQUITECTURA CLIENTE/SERVIDOR**

En la arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo.

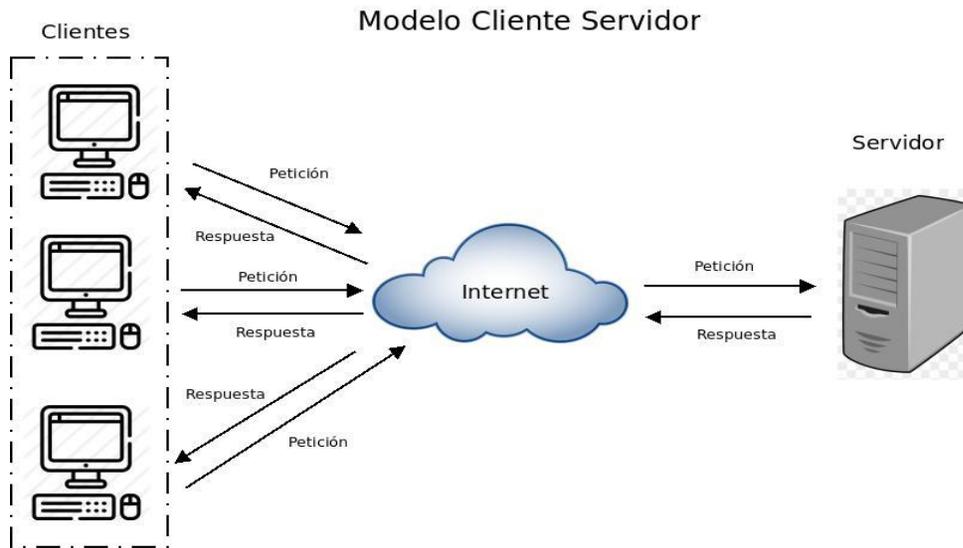
Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es el Internet.

### **Ventajas del Modelo Cliente/Servidor**

Existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas mucho más baratas que las requeridas por una solución centralizada, basada en sistemas grandes (mainframes). Facilita la integración entre sistemas diferentes y comparte información, permitiendo por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables el usuario.

- Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos bajo este esquema tienen una mayor y más intuitiva con el usuario.
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- Contribuye además a proporcionar a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información. (Schiaffarino, 2015).

**Figura N° 2. 12. Modelo Cliente-Servidor**



**Fuente:** (Andrés Schiaffarino, 2015)

## 2.12. MODELO VISTA CONTROLADOR (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

**Figura N° 2. 13. Modelo Cliente-Servidor**



**Fuente:** (Uriel Hernández, 2015)

**Modelo:** Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc.

**Vistas:** Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

**Controlador:** Se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista. (Hernández, 2015).

## **2.13. HERRAMIENTAS DE DESARROLLO DE SOFTWARE**

### **2.13.1. LENGUAJE PHP**

PHP es un lenguaje de programación muy potente que, junto con html, permite crear sitios web dinámicos. Php se instala en el servidor y funciona con versiones de Apache, Microsoft IIS, Netscape Enterprise Server y otros.

La forma de usar php es insertando código php dentro del código html de un sitio web. Cuando un cliente (cualquier persona en la web) visita la página web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador. (Lerdort, 1995).

#### **Características de PHP**

- ✓ Más rápido que ASP
- ✓ Es un lenguaje multiplataforma
- ✓ Lenguaje más fácil y potente
- ✓ Integración perfecta más de ocho servidores HTTP
- ✓ Diseño modular de fácil ampliación
- ✓ Licencia abierta
- ✓ Acceso a 20 tipos de Base de Datos.

## 2.13.2. FRAMEWORK LARAVEL

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma amigable y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

### Características:

- ✓ Sistema de ruteo, también RESTful
- ✓ Blade, Motor de plantillas
- ✓ Peticiones Fluent
- ✓ Eloquent ORM
- ✓ Basado en Composer
- ✓ Soporte para el caché
- ✓ Soporte para MVC
- ✓ Usa componentes de Symfony
- ✓ Adopta las especificaciones PSR-2 y PSR-4

Laravel incluye un sistema de mapeo de datos relacional llamado Eloquent ORM que facilita la creación de modelos. Este ORM se funda en patrón active record y su funcionamiento es muy sencillo. Es opcional el uso de Eloquent, pues también dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la creación de modelos.

Laravel incluye de paquete un sistema de procesamiento de plantillas llamado Blade. Este sistema de plantillas favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace mucho más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura. Por ejemplo, en vez pintar la vista usando el código PHP.

### 2.13.2.1. ORM

Mapeo Objeto-Relacional o como se conocen comúnmente, ORM (del inglés Object Relational Mapping). Son, aquellos que permite convertir los datos de tus objetos en un formato correcto para poder guardar la información en una base de datos (mapeo) creándose una base de datos virtual donde los datos que se encuentran en nuestra aplicación, quedan vinculados a la base de datos y tiene operación corresponde con una sentencia SQL.

- ✓ Crear: INSERT
- ✓ Obtener: SELECT
- ✓ Actualizar: UPDATE
- ✓ Borrar: DELETE

### 2.13.2.2. BLADE

Blade de Laravel, permite una sintaxis mucho más reducida en su escritura. Blade nos provee muchas ventajas, además de modularizar la vista de una forma presentable también nos permite usar estructuras de control y Básicamente: `{{ }}` es un sustituto de `<?php ?>` que por un lado es más corto y fácil de escribir, por otro usar etiquetas PHP (`<? ?>`) se “mezcla” con las etiquetas del HTML, así que usar motores como Blade es más “limpio” por así decirlo.

Blade sólo reemplaza `{{ * }}` por `<?php echo * ?>`, es decir, todo el código PHP que quieras escribir es válido dentro de `{{ }}`, por ejemplo:

- **Código:**

```
<h1>Hello {{ strtoupper($name) }}</h1>
```

### 2.13.2.3. COMPOSER

Composer es un manejador de dependencias, no un gestor de paquetes. Pero es cierto que trata con paquetes y librerías, la instalación siempre es local para cualquier proyecto, las librerías se instalan en un directorio por defecto (normalmente es /vendor). Composer es capaz de instalar las librerías que requiere tu proyecto con las versiones que necesiten. También es capaz de resolver eso y descargar todo lo necesario para su funcionamiento. (Otwell, 2018).

Figura N° 2. 14. Gestor de Dependencias de PHP



Fuente: (Taylor Otwell, 2018).

### 2.13.3. BASE DE DATOS

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros.

### 2.13.4. SISTEMA GESTOR DE BASE DE DATOS

Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar,

modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones.

### **Características gestores de bases de datos**

- ✓ Permite crear y gestionar base de datos de forma fácil, cómoda y rápida.
- ✓ Tiene gran flexibilidad para el trabajo con base de datos relacional.
- ✓ Ofrece una interfaz gráfico agradable.

#### **2.13.4.1. MySQL**

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido , MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP).

### **Características**

- ✓ Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- ✓ Transacciones y claves foráneas.
- ✓ Conectividad segura.
- ✓ Replicación.
- ✓ Búsqueda e indexación de campos de texto.

### **2.13.5. SERVIDOR WEB APACHE**

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI.

#### **Ventajas**

- ✓ Instalación/Configuración. Software de código abierto.
- ✓ Coste. El servidor web Apache es completamente gratuito.
- ✓ Funcional y Soporte. Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.
- ✓ Multi-plataforma. Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- ✓ Rendimiento. Capacidad de manejar más de un millón de visitas/día.
- ✓ Soporte de seguridad SSL y TLS.

### **2.14. PRUEBAS DE SOFTWARE**

#### **2.14.1. PRUEBAS DE CAJA NEGRA**

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase

de errores diferente que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- ✓ Funciones incorrectas o faltantes.
- ✓ Errores de interfaz.
- ✓ Errores en las estructuras de datos o en el acceso a bases de datos externas.
- ✓ Errores de comportamiento o rendimiento.
- ✓ Errores de inicialización y terminación.

A diferencia de las pruebas de caja blanca, que se realizan tempranamente en el proceso de pruebas, la prueba de caja negra tiende a aplicarse durante las últimas etapas de la prueba. Puesto que, a propósito, la prueba de caja negra no considera la estructura de control, la atención se enfoca en el dominio de la información. (Pressman, 2010)

#### **2.14.2. PRUEBAS DE CAJA BLANCA**

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

- ✓ Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- ✓ Revisen todas las decisiones lógicas en sus lados verdadero y falso
- ✓ Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- ✓ Revisen estructuras de datos internas para garantizar su validez. (Pressman, 2010).

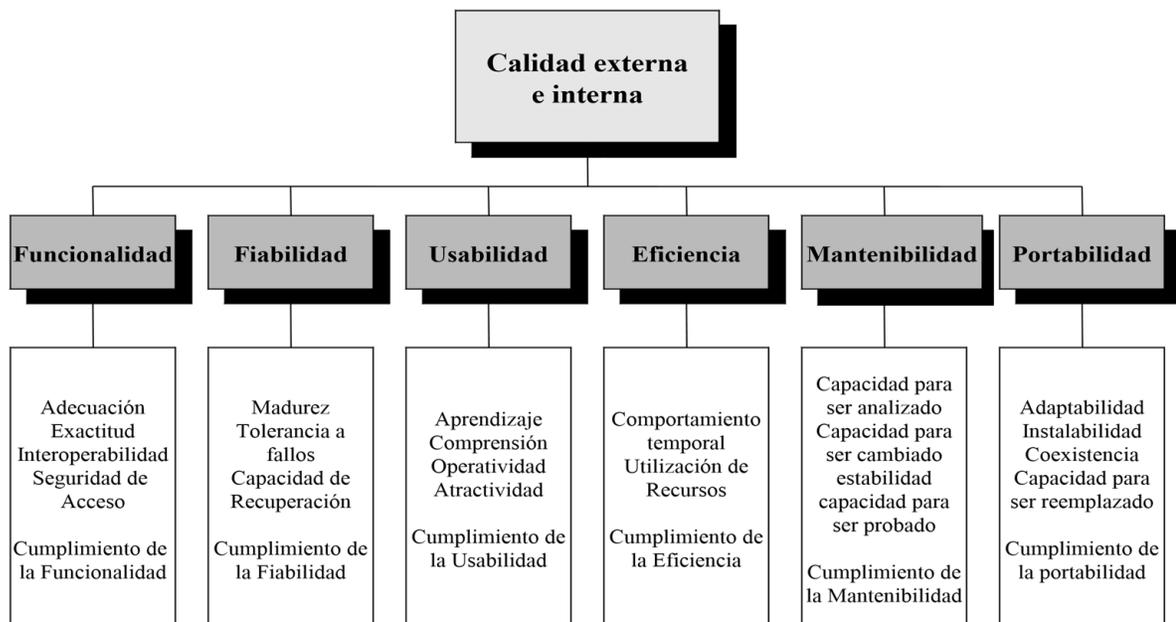
## 2.15. MÉTRICAS DE CALIDAD DE SOFTWARE ESTABLECIDAS POR LA NORMA ISO 9126

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del software. Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad.

El primer objetivo del equipo de proyecto es medir errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad. (Roger S. Pressman, Calidad en la industria del Software Norma ISO-9126, 2012).

**Figura N° 2. 15.** Norma ISO/IEC 9126



**Fuente:** (Roger S. Pressman, 2012).

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software. Los modelos de calidad para el software se describen así:

### 2.15.1. FUNCIONALIDAD

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. A continuación se muestra la característica de Funcionalidad y las subcaracterísticas que cubre: La funcionalidad se divide en cinco criterios:

**Adecuación:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

**Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

**Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.

**Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados.

**Conformidad de la funcionalidad:** La capacidad del software de cumplir los estándares referentes a la funcionalidad.

Para calcular los puntos de función se utiliza la siguiente ecuación.

$$PF = \text{Conteo Total} * \left( 0,65 + 0,01 * \sum fi \right)$$

Donde:

Punto Función: Medida de Funcionalidad entregada a la aplicación como valor de normalización.

Conteo Total: Es el nivel de complejidad del sistema respecto al usuario.

0,01: Factor de conversión, es decir un error del 1%.

0,65: Valor mínimo del ajuste de la complejidad.

$\Sigma Fi$ : Es la sumatoria de los valores de ajuste complejidad.

### **2.15.2. EFICIENCIA**

La eficiencia del software es la forma del desempeño adecuado, de acuerdo al número de recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros. La eficiencia se divide en tres criterios:

**Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta, realizar la misma función pero con más cantidad de registros.

**Utilización de recursos:** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

**Conformidad de eficiencia:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

### **2.15.3. CONFIABILIDAD**

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En este caso a confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida. La confiabilidad se divide en cuatro criterios:

**Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacia, o cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.

**Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

**Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

**Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad. Formula

$$F(t) = f * e^{(-\mu*t)}$$

Donde:

$f$ , funcionalidad del sistema.

$\mu$ , Probabilidad de error que puede tener el sistema.

$t$ , tiempo que dura una gestión en el sistema.

En el inicio de ejecución  $t_0=0$ , lo que significa el tiempo inicial el cual dará inicio el funcionamiento del sistema.

El término en el cual el sistema trabaja sin falla está en la siguiente ecuación:

$$P(T \leq t) = F(t) \quad \text{Probabilidad de fallas.}$$

$$P(T \leq t) = 1 - F(t) \quad \text{Probabilidad de trabajo sin fallas.}$$

#### **2.15.4. USABILIDAD**

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los

usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. La usabilidad se divide en cinco criterios:

**Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

**Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

**Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.

**Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

**Conformidad de uso:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad. Lo cual se calcula con la siguiente formula:

$$FU = \left( \frac{\sum Xi}{n} \right) * 100$$

#### **2.15.5. MANTENIBILIDAD**

La capacidad de mantenibilidad es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales. El mantenimiento se divide en cinco criterios:

**Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

**Cambiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.

**Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.

**Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

**Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento. Se lo determina con la siguiente formula:

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$$

Donde:

*Mt*, número de módulos de la versión actual.

*Fa*, numero de módulos en la versión actual que se han añadido.

*Fc*, numero de módulos en la versión actual que se han cambiado.

*Fd*, numero de módulos de la versión anterior que se han borrado en la versión actual.

#### **2.15.6. PORTABILIDAD**

La capacidad que tiene el software para ser trasladado de un entorno a otro. La portabilidad se divide en cinco criterios:

**Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

**Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.

**Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.

**Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.

**Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad. (Roger S. Pressman, Calidad en la industria del Software Norma ISO-9126, 2012).

## 2.16. PUNTO FUNCIÓN

La técnica de medición del tamaño en punto-función consiste en asignar una cantidad de puntos a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos, siempre tratando de considerarlo desde el punto de vista del usuario. (Pressman R. S., Punto Función, 2013)

Los valores de los dominios de información se definen de la forma siguiente:

**Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

**Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

**Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.

**Número de archivos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).

**Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad. Las organizaciones que utilizan métodos de puntos de función desarrollan criterios para determinar si una entrada en particular es simple, media o compleja. No obstante la determinación de la complejidad es algo subjetiva.

**Figura N° 2. 16.** Cálculo de puntos de función.

Parámetros de medición	Cuenta	Factor de ponderación			=	Cuenta
		Simple	Medio	Complejo		
Número de entradas de usuario	<input type="text"/>	× 3	4	6	=	<input type="text"/>
Número de salidas de usuario	<input type="text"/>	× 4	5	7	=	<input type="text"/>
Número de peticiones de usuario	<input type="text"/>	× 3	4	6	=	<input type="text"/>
Número de archivos	<input type="text"/>	× 7	10	15	=	<input type="text"/>
Número de interfaces externas	<input type="text"/>	× 5	7	10	=	<input type="text"/>
Cuenta total	→					<input type="text"/>

**Fuente:** (Roger S. Pressman, 2013).

Para calcular puntos de función, se utiliza la relación siguiente:

$$PF = Cuenta - Total * [065 + 0,01 * \sum(fi)] \quad (1)$$

Donde cuenta-total es la suma de todas las entradas obtenidas de la Figura N° 2.16  
 Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las siguientes preguntas:

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?
6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
8. ¿Se actualizan los archivos maestros de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidas en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Cada una de las preguntas anteriores es respondida usando una escala con rangos desde 0 (no importante o aplicable) hasta 5 (absolutamente esencial).

**Figura N° 2. 17.** Escala de Valores

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

**Fuente:** (Roger S. Pressman, 2013).

Los valores constantes de la ecuación (1) y los factores de peso que se aplican a las cuentas de los dominios de información se determinan empíricamente. Una vez que se han calculado los puntos de función, se utilizan de forma análoga a las líneas de código como forma de normalizar las medidas de productividad, calidad y otros atributos del software.

Una vez calculado los puntos de función se usan de forma analógica a las líneas de código como medida de la productividad, calidad y otros productos del software.

**Productividad** = PF / persona-mes

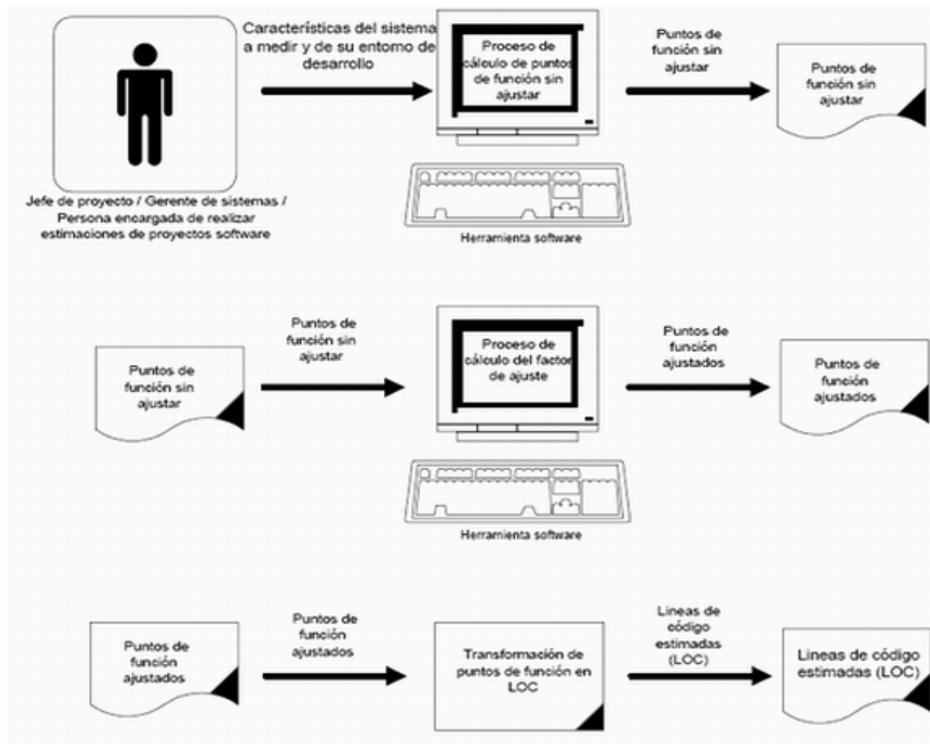
**Calidad** = Errores / PF

**Costo** = Dólares / PF

**Documentación** = Pags. Doc / PF

El proceso de cálculo y obtención de puntos de función y su posible transformación a líneas de código se esquematiza en la figura:

**Figura N° 2. 18.** Proceso de cálculo de puntos de función.



**Fuente:** (Roger S. Pressman, 2013).

## **2.17. METODO DE ESTIMACION DE COSTOS DE SOFTWARE**

### **2.17.1. MODELO CONSTRUCTIVO DE COSTOS (COCOMO)**

El Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés COConstructive COst MOdel) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981).

#### **2.17.1.1. CARACTERÍSTICAS GENERALES**

Pertenece a la categoría de modelos estimadores basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en función de la cantidad de líneas de código, principalmente.

Se presentan tres niveles: básico, intermedio y detallado.

#### **2.17.1.2. INCONVENIENTES**

- ✓ Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.
- ✓ Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
- ✓ Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
- ✓ Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.
- ✓ La medición por líneas de código no es válida para orientación a objetos.

- ✓ Utilizar este modelo puede resultar un poco complicado, en comparación con otros métodos (que también sólo estiman).

### 2.17.1.3. MODELOS DE ESTIMACIÓN

Las ecuaciones que se utilizan en los tres modelos son:

- ✓  $E = a(Kl)^b * m(X)$ , en persona/mes
- ✓  $Tdev = c(E)^d$ , en meses
- ✓  $P = E/Tdev$ , en personas

donde:

- ✓ E es el esfuerzo requerido por el proyecto, en persona-mes
- ✓ Tdev es el tiempo requerido por el proyecto, en meses
- ✓ P es el número de personas requerido por el proyecto
- ✓ a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo
- ✓ Kl es la cantidad de líneas de código, en miles.
- ✓ m(X) Es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- ✓ Modo orgánico, un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- ✓ Modo semilibre o semiencajado, corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- ✓ Modo rígido o empotrado, el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema

a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

#### 2.17.1.4. MODELO BÁSICO

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

**Tabla N° 2. 3.** Constantes para el cálculo de costes Modelo Básico

MODO	A	b	C	d
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semi – Orgánico</b>	3.00	1.12	2.50	0.35
<b>Empotrado</b>	3.60	1.20	2.50	0.33

**Fuente:** Ingeniería de Software (R. Pressman, 2010)

Estos valores son para las fórmulas:

- ✓ Personas necesarias por mes para llevar adelante el proyecto

$$(MM) = a * (Kl^b)$$

- ✓ Tiempo de desarrollo del proyecto

$$(Tdev) = c * (MM^d)$$

- ✓ Personas necesarias para realizar el proyecto

$$(CosteH) = MM/Tdev$$

- ✓ Costo total del proyecto

$$(CosteM) = CosteH * \text{Salario medio entre los programadores y analistas}$$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del

esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.

#### 2.17.1.5. MODELO INTERMEDIO

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula son:

**Tabla N° 2. 4.** Constantes para el cálculo de costes Modelo Intermedio

<b>MODO</b>	<b>a</b>	<b>B</b>
<b>Orgánico</b>	3.20	1.05
<b>Semi – Orgánico</b>	3.00	1.12
<b>Empotrado</b>	2.80	1.20

**Fuente:** Ingeniería de Software (R. Pressman, 2010)

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste.

#### 2.17.1.6. ATRIBUTOS

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy bajo - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

- De software
  - ✓ RELY: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).
  - ✓ DATA: tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación:  $D/K$  donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
  - ✓ CPLX: representa la complejidad del producto.
  
- De hardware
  - ✓ TIME: limitaciones en el porcentaje del uso de la CPU.
  - ✓ STOR: limitaciones en el porcentaje del uso de la memoria.
  - ✓ VIRT: volatilidad de la máquina virtual.
  - ✓ TURN: tiempo de respuesta requerido.
  
- De personal
  - ✓ ACAP: calificación de los analistas.
  - ✓ AEXP: experiencia del personal en aplicaciones similares.
  - ✓ PCAP: calificación de los programadores.
  - ✓ VEXP: experiencia del personal en la máquina virtual.
  - ✓ LEXP: experiencia en el lenguaje de programación a usar.
  
- De proyecto
  - ✓ MODP: uso de prácticas modernas de programación.
  - ✓ TOOL: uso de herramientas de desarrollo de software.

- ✓ SCED: limitaciones en el cumplimiento de la planificación.

El valor de cada atributo, de acuerdo a su calificación, se muestra en la siguiente tabla:

**Tabla N° 2. 5.** Atributos para el cálculo de costes

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	

Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

**Fuente:** Ingeniería de software (R. Pressman, 2010).

### 2.17.1.7. MODELO DETALLADO

Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.

Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema.

### 2.18. SISTEMA DE GESTION DE SEGURIDAD DE LA INFORMACION ISO/IEC 27000

Es un estándar para la seguridad de información, aprobada y publicada en octubre de 2005 por ISO. Especifica los requisitos necesarios para establecer, mantener y mejorar un sistema de Gestión de la seguridad de la información (SGSI).SGSI también llamada Sistema de Gestión de la Seguridad de la Información, conjunto de políticas y procedimiento que normaliza la gestión de la seguridad de la información de toda una organización, debe garantizar la disponibilidad, confidencialidad e integridad de la información.

## **2.19. SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACION ISO-27002**

La ISO 27002, es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información.

Se encuentra estructurada en 14 capítulos que describen las áreas que se deben considerar para garantizar la seguridad de la información de las que se dispone.

### **1. Políticas de Seguridad de la Información**

Dentro de este capítulo se hace hincapié en la importancia que ocupa la disposición de una adecuada política de seguridad, aprobada por la dirección, comunicada a todo el personal, revisada de forma periódica y actualizada con los cambios que se producen en el interior y en el exterior.

### **2. Organización de la Seguridad de la Información**

Los controles indicados en este capítulo buscan estructurar un marco de seguridad eficiente tanto mediante los roles, tareas, seguridad, etc. como en los dispositivos móviles.

### **3. Seguridad relativa a los recursos humanos**

Si analizamos los incidentes de seguridad que se producen en una organización nos daremos cuenta de que la gran mayoría de estos tienen su origen en un error humano. Se debe concienciar y formar al personal de los términos de empleo de la información en el desarrollo de sus actividades y la importancia que tiene la información en el desarrollo de sus actividades.

### **4. Gestión de activos**

Se centra en la atención en la información como activo y en cómo se deben establecer las medidas adecuadas para guardarlos de las incidencias, quebras en la seguridad y en la alteración no deseada.

## **5. Control de acceso**

Controlar quien accede a la información dentro de un aspecto relevante. Al fin y al cabo no todas las personas de una organización necesitan acceder para realizar su actividad diarias a todos los datos, sino que tendremos roles que necesitan un mayor acceso y otros con un acceso mucho más limitado. Para poder marcar las diferencias, se deben establecer todos los controles como registro de los usuarios, gestión de los privilegios de acceso, etc.

## **6. Criptografía**

En el caso de que estemos tratando la información sensible o crítica puede ser interesante utilizar diferentes técnicas criptográficas para proteger y garantizar su autenticidad, confidencialidad e integridad.

En criptografía, **MD5** (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. Uno de sus usos es el de comprobar que algún archivo no haya sido modificado.

## **7. Seguridad física y del entorno**

La seguridad no es solo a nivel tecnológico sino también físico, es decir, una simple labor de no dejar las pantallas e impresoras en zonas que sean fácilmente accesibles, por parte del personal externo los documentos con los que se están trabajando no sólo nos permitirán gestionar de forma adecuada la seguridad sino que se acabarán convirtiendo en hábitos que nos aportan eficiencia en la gestión.

## **8. Seguridad de las operaciones**

Tiene un marcado componente técnico entrado en todos los aspectos disponibles como la protección del software malicioso, copias de seguridad, control de software en explotación, gestión de vulnerabilidad, etc.

## **9. Seguridad de las comunicaciones**

Partiendo de la base de que la gran mayoría de los intercambios de información y de datos en distintas escalas se llevan a cabo mediante las redes sociales, garantizar la seguridad y proteger de forma adecuada los medios de transmisión de estos datos clave.

## **10. Adquisiciones, desarrollo y mantenimiento de los sistemas de información**

La seguridad no es un aspecto de un área en concreto, ni de un determinado proceso, no que es general, abarca toda la organización y tiene que estar presente como elemento transversal clave dentro del ciclo de vida del sistema de gestión.

## **11. Relación de proveedores**

Cuando se establecen las relaciones con terceras partes, como puede ser proveedores, se deben establecer medidas de seguridad pudiendo ser muy recomendable e incluso necesario en determinados casos.

## **12. Gestión de incidentes de seguridad de la información.**

No podemos hablar de controles de seguridad sin mencionar un elemento clave, los incidentes en seguridad. Y es que, estar preparados para cuando estos incidentes ocurran, dando una respuesta rápida y eficiente siendo la clave para prevenirlos en el futuro.

## **13. Aspectos de seguridad de la información para la gestión de la continuidad de negocio.**

No sabemos lo que necesitábamos un dato hasta que lo hemos perdido. Sufrir una pérdida de información relevante y no poder recuperarla de laguna forma puede poner en peligro la continuidad de negocio de la organización.

## **14. Cumplimiento**

No podemos hablar de seguridad de la información, sin hablar de legislación, normas y políticas aplicables que se encuentre relacionadas con este campo y con las que conviven en las organizaciones. Debemos tener presente que ocupan un enorme lugar en cualquier sistema de gestión y deben garantizar que se cumple y que están actualizados con los últimos cambios siendo esencial para no llevarnos sorpresas desagradables. (Camelo, 2010).

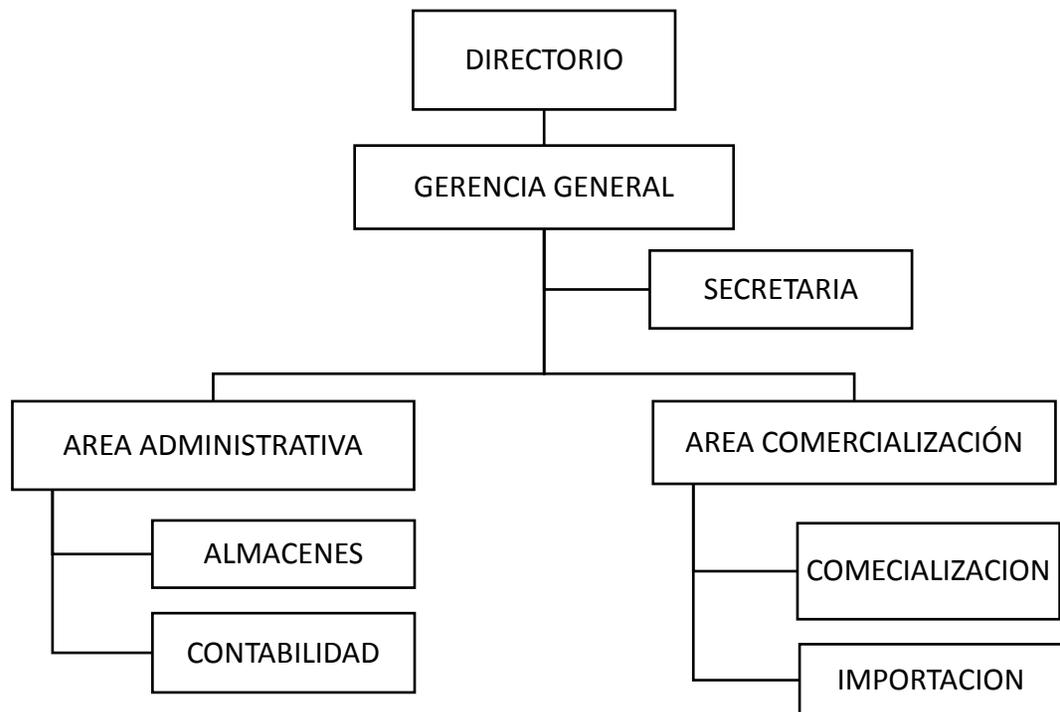


**CAPÍTULO III**  
**MARCO APLICATIVO**

3.1. ANALISIS DE REQUERIMIENTOS

En esta fase entramos a la etapa de análisis y diseño, y el comportamiento actual, donde identificamos a los actores que interactúan con los requerimientos del sistema lo primero es enfocarse en modelo de negocio y luego nos enfocaremos en los casos de uso de negocio. A continuación la organización de la empresa Victory Motors Bolivia.

Figura N° 3. 1. Organigrama de la empresa



Fuente: Elaboración Victory Motors Bolivia (Diego Tarqui, 2014)

### 3.1.1. ANALISIS DE LA SITUACIÓN ACTUAL

La empresa Victory Motors Bolivia se dedica a la venta de productos importados, como ser vehículos de diferentes marcas y modelos y que todos estos productos son almacenados en el almacén e inventariados, es manipulada de manera manual con el apoyo de hojas en calculo: Excel consumiendo el tiempo para los encargados de las diferente áreas, debido al aumento de demanda de la empresa, es necesario contar con un sistema de manera ordenada y que permita administrar las funciones y las actividades relacionadas a la venta.

### 3.1.2. REQUERIMIENTOS DEL SISTEMA

El método de la ingeniería de requerimientos es el que permite identificar los servicios y restricciones que forman un sistema.

Como también se encarga de la realización de actividades en el intento de entender las necesidades exactas de los usuarios y formar sus funciones y acciones que serán útil para el desarrollo del sistema.

A continuación detallamos algunas de las especificaciones del requerimiento del usuario.

- ✓ Registro de ingreso de productos a la empresa
- ✓ Registro de la venta del producto
- ✓ Verificación de la existencia de productos para su venta
- ✓ Obtener reportes de los movimientos realizados de los productos.

La descripción del funcionamiento del sistema actual puede apreciarse en las siguientes tablas:

**Tabla N° 3. 1.** Categoría de Funciones

<b>Categoría de la Función</b>	<b>Significado</b>
<b>Evidente</b>	Debe realizarse, y que los usuarios deben estar conscientes lo que se ha elaborado.

<b>Oculto</b>	Debe realizarse aunque no sea visible para los usuarios. Son guardados en un mecanismo constante de almacenamiento.
<b>Superflua</b>	Opcional es su inclusión no resulte de forma significativa en costo ni en otras funciones.

**Fuente:** (Elaboración Propia)

### 3.1.2.1. REQUERIMIENTOS FUNCIONALES

Estos requerimientos se detallan a continuación:

**Tabla N° 3. 2.** Requerimientos Funcionales

Ref.	Función	Categoría
<b>Ref. 1</b>	El sistema, debe tener seguridad en al acceso de la información del sistema.	Evidente
<b>Ref. 2</b>	Administración de usuarios, debe ser posible la creación, modificación y eliminación de la información de las cuentas de usuarios. Así también poderlos deshabilitar según se requiera. Dichas operaciones solo deberá ser realizado por el gerente o el administrador.	Evidente
<b>Ref. 3</b>	Registro de productos, se deben registrar los productos como ser las características código, nombre, modelo, descripción, categoría y precio de venta y entrada, imagen.	Evidente
<b>Ref. 4</b>	Registro de clientes, debe ser posible tener un registro de los clientes y sus datos, así mismo poderlos modificarlos o eliminar.	Evidente
<b>Ref. 5</b>	Venta, el sistema debe der capaz de generar una nota de venta con los productos solicitados por el cliente y sus	Evidente

	precios correspondientes, así mismo debe ser posible modificar la venta durante el proceso de negociación.	
<b>Ref. 6</b>	Asignación de roles, para una mejor administración del sistema, se debe contar con distintos tipos de usuario que tendrán diferentes acciones dentro del sistema, los principales son el vendedor, administrador o gerente.	Evidente
<b>Ref. 7</b>	Descarga de reportes, será posible descarga la información generada por el módulo de reportes en archivos de formato PDF, los cuales serán de gran importancia y de utilidad para la presentación de informes.	Evidente
<b>Ref. 9</b>	Cierre del sistema, el sistema deberá ser capaz de cerrar.	Oculto

**Fuente:** (Elaboración Propia)

### 3.1.2.2. REQUERIMIENTOS NO FUNCIONALES

Estas funciones se detallan en la siguiente tabla:

**Tabla N° 3. 3.** Requerimientos No Funcionales

<b>Ref.</b>	<b>Función</b>	<b>Categoría</b>
<b>Ref. 1</b>	El sistema debe ser accesible desde cualquier lugar y cualquier hora, lo que requiere un navegador ya sea Chrome, Mozilla, etc.	Evidente
<b>Ref. 2</b>	El sistema debe asegurar que los datos estén protegidos del acceso no autorizado.	Evidente
<b>Ref. 3</b>	El sistema debe contar con manual de usuario estructurados actualmente.	Evidente
<b>Ref. 4</b>	El sistema debe proporcionar mensajes de error que sean informativos y orientadas al usuario final.	Evidente
<b>Ref. 5</b>	El sistema debe poseer un diseño “ <i>Responsive</i> ” a fin de garantizar la visualización deben múltiples dispositivos tabletas y teléfonos inteligentes.	Evidente

**Fuente:** (Elaboración Propia)

### **3.1.3. ACTORES IDENTIFICADOS**

Una vez obtenida el estudio preliminar de la situación actual de la empresa, se pudo observar e identificar a los actores. Los actores son usuarios del sistema los cuales interactúan aportar y reciben información del sistema.

Detallando a los actores del sistema:

#### **Gerencia General (Administrador).**

Este actor es el que está informado del movimiento de la empresa. Cuando es necesario puede realizar diferentes niveles de tareas dando soluciones a los errores que presenta la gerencia como también los problemas técnicos.

Esta persona tiene que estar informado teniendo el control y seguimiento de las actividades de la empresa y sus funciones son:

- ✓ Asignar tarea a cada miembro y supervisando su funcionamiento.
- ✓ Aprueba el movimiento de la empresa acerca de las ventas.

#### **Área administrativa (Jefe de Almacén).**

Es encargado del movimiento de los productos que se encuentra dentro el almacén.

- ✓ Realiza el registro de los productos que entran y salen del almacén.
- ✓ Controla el stock y el reporte de inventarios.

#### **Área de comercialización (Contabilidad).**

Sus funciones se detallan de esta manera:

- ✓ Realiza el registro del nuevo cliente
- ✓ El vendedor verificar la existencia del producto solicitado por el cliente, luego genera la nota de venta correspondiente y si el cliente la acepta, la venta registrándola y emitiendo la nota venta. En el caso de que el producto no exista en almacenes, el vendedor informe al administrador de la existencia para realizar la importación.

### 3.1.4. MODELO DE CASO DE USO DEL SISTEMA.

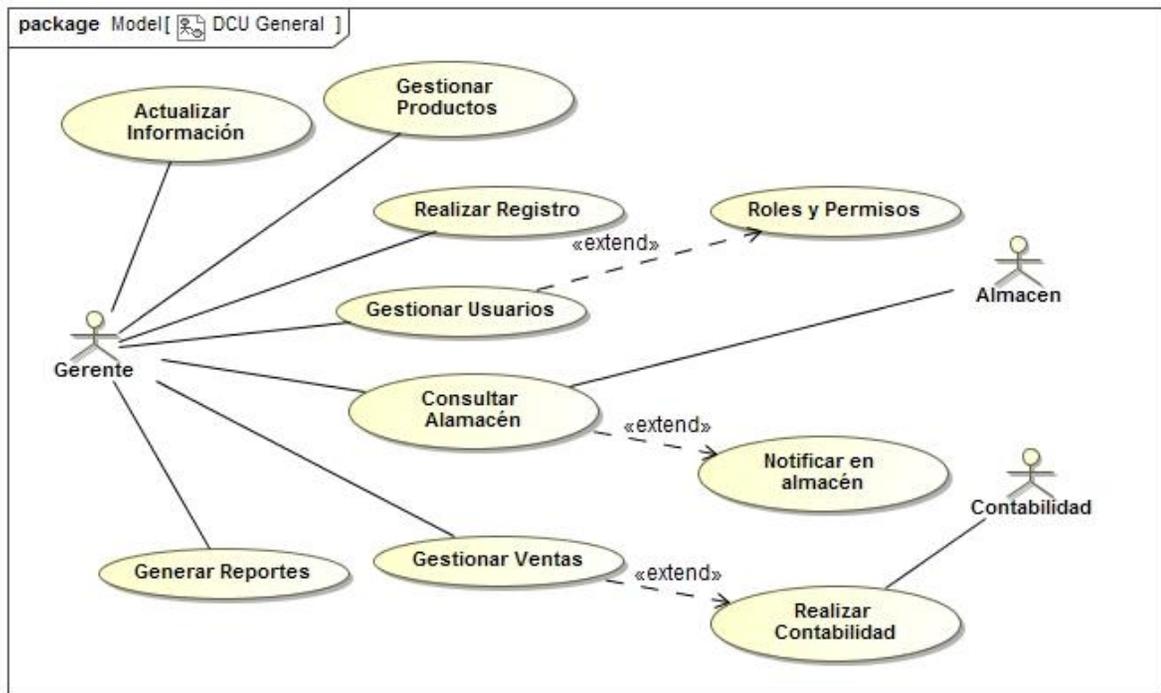
El caso de uso nos ayuda a capturar los requisitos de acuerdo según el usuario necesita una función como también es una representación gráfica del sistema.

En continuidad se muestra el análisis de requerimientos del sistema mediante el diseño de caso de uso con la presentación de las actividades de los actores dentro el sistema.

#### 3.1.4.1. DIAGRAMA DE CASO DE USO GENERAL DEL SISTEMA.

En este presente diagrama se muestra el movimiento interno de la empresa como los actores interactúan con el sistema.

**Figura N° 3. 2.** Diagrama de casos de uso General



**Fuente:** (Elaboración Propia).

**Tabla N° 3. 4.** Especificación de los actores de Caso de Uso

<b>ACTIVIDADES DE CADA AUTOR INTERACTUANDO CON EL SISTEMA</b>	
 <b>Gerente</b> <b>General/Administrador</b>	<ul style="list-style-type: none"> <li>✓ Gestiona Usuarios</li> <li>✓ Realiza el registro de los productos.</li> <li>✓ Realiza el registro del ingreso de los productos.</li> <li>✓ Registra a los clientes.</li> <li>✓ Realiza las ventas de los productos.</li> <li>✓ Genera reportes de las ventas.</li> <li>✓ Administra el almacén y el stock</li> </ul>
 <b>Almacén</b>	<ul style="list-style-type: none"> <li>✓ Administra el almacén.</li> <li>✓ Registra el ingreso de los productos.</li> <li>✓ Registra el stock del producto.</li> </ul>
 <b>Contabilidad</b>	<ul style="list-style-type: none"> <li>✓ Realiza la venta.</li> <li>✓ Registra los pedidos del producto.</li> <li>✓ Genera reportes de las ventas.</li> </ul>

**Fuente:** (Elaboración Propia).

### 3.1.4.2. DESCRIPCION DEL CASO DE USO GENERAL

En esta etapa vamos a describir y comprender la funcionalidad del sistema.

**Tabla N° 3. 5.** Caso de Uso Gestionar Productos.

<b>Nombre</b>	<b>Gestionar productos</b>
<b>Actores</b>	Almacén y gerente
<b>Descripción</b>	<b>Eventos Actor</b>
	Realiza las siguientes operaciones de gestión como ser: altas, bajas, modificaciones o búsquedas.

**Fuente:** (Elaboración Propia).

**Tabla N° 3. 6.** Caso de Uso Gestionar Usuarios.

<b>Nombre</b>	<b>Gestionar usuarios.</b>
<b>Actor</b>	Administrador, gerente
<b>Descripción</b>	<b>Eventos Actor</b>
	Se realiza las operaciones de gestión tales como: altas, bajas o modificación.

**Fuente:** (Elaboración Propia).

**Tabla N° 3. 7.** Caso de Uso Consultar Almacén

<b>Nombre</b>	<b>Consultar en almacén.</b>
<b>Actor</b>	Gerente y almacén.
<b>Descripción</b>	<b>Eventos Actor</b>
	Para poder responder la petición del almacén realizamos la búsqueda (nombre del producto).
	Una vez hecha la búsqueda se observar los datos existentes en el almacén.

**Fuente:** (Elaboración Propia).

**Tabla N° 3. 8.** Caso de Uso Gestionar las Ventas

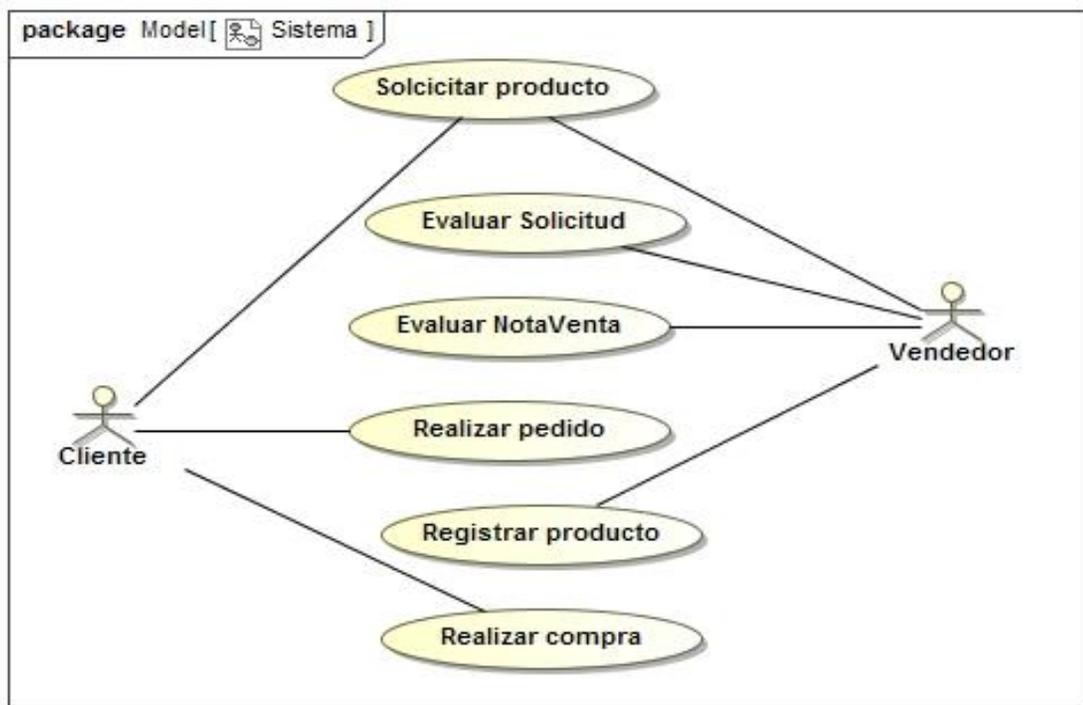
<b>Nombre</b>	<b>Realizar compra de un producto</b>
<b>Actor</b>	Gerente, Contabilidad
<b>Descripción principal</b>	<b>Eventos Actor</b>
	Se realiza la petición del producto.
	Se hace la búsqueda y emite una nota de venta.
	Evaluación del documento para su aceptación
	Se registra la venta y se emite una nota de venta correspondiente.
	Se registra el nuevo producto en almacén.

<b>Descripción secundario y/o alternativo</b>	Una vez registrada el producto está listo para su posterior venta.
---	--

**Fuente:** (Elaboración Propia).

### 3.1.4.3. CASO DE USO DEL SISTEMA.

**Figura N° 3. 3.** Diagrama de casos de uso Cliente



**Fuente:** (Elaboración Propia).

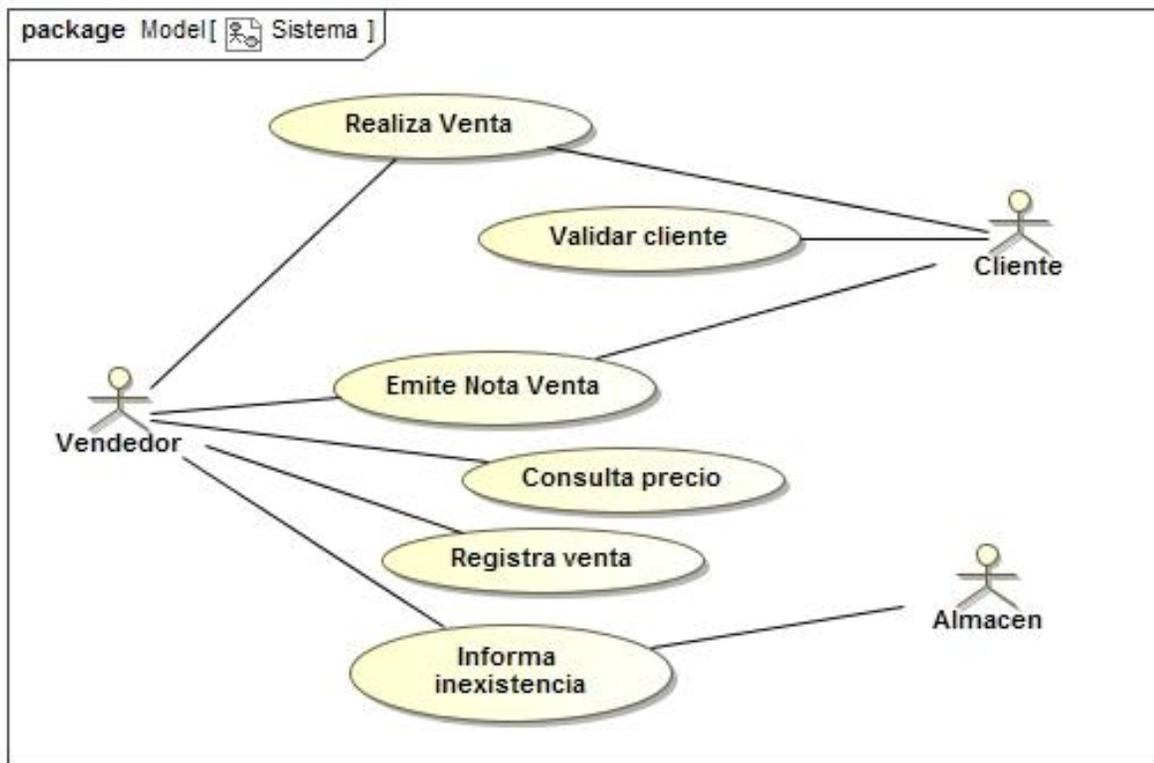
**Tabla N° 3. 9.** Especificación de los casos de uso Cliente

<b>Nombre</b>	Realizar venta de un producto	
<b>Actor</b>	Cliente ,Vendedor	
<b>Descripción principal</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	Se realiza la petición del producto.	Se muestra la vista de búsqueda de productos.
	Se hace la búsqueda y emite una nota de venta.	Genera cotización para imprimirla.

	Evaluación del documento para su aceptación	
	Se registra la venta y se emite una nota de venta correspondiente.	Se presenta el formulario de registro de venta.
Descripción secundario y/o alternativo	Se hace un pedido de productos.	
	Espera de llegada del producto para su posterior registro de venta.	Presenta el formulario de registro de venta.

Fuente: (Elaboración Propia).

Figura N° 3. 4. Diagrama de casos de uso Vendedor



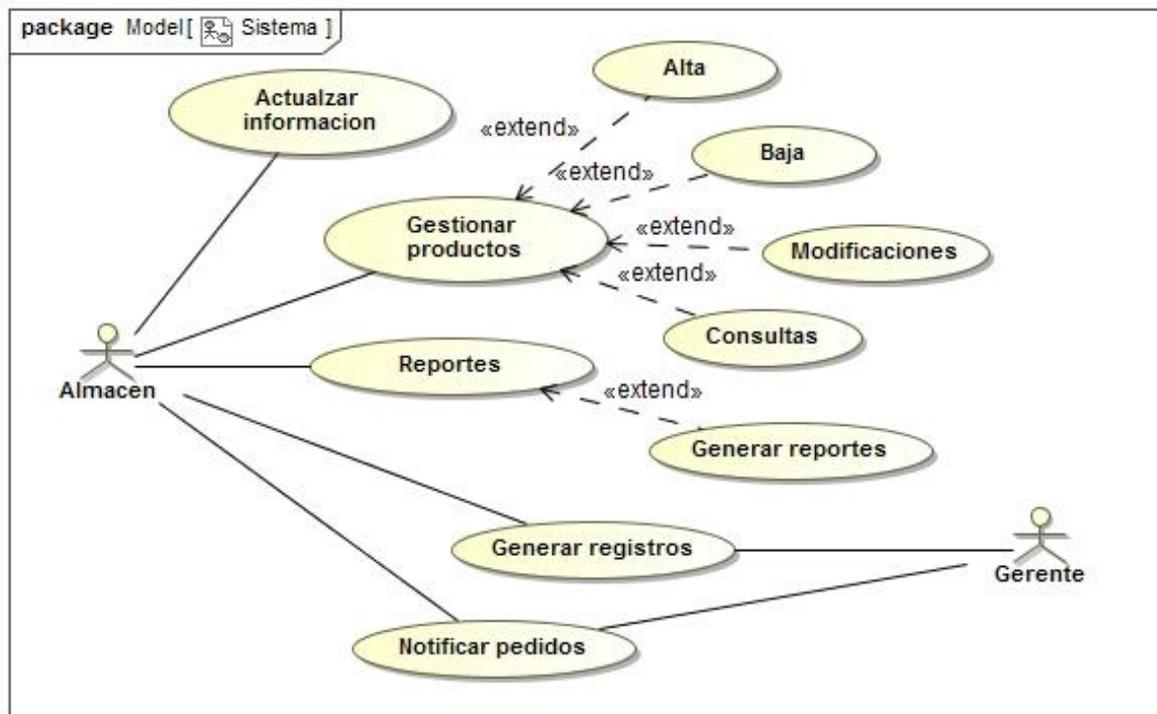
Fuente: (Elaboración Propia).

**Tabla N° 3. 10.** Especificación de los casos de uso Vendedor.

<b>Nombre</b>		
Consultar almacén, registrar venta, emisión de nota de venta.		
<b>Actor</b>	Vendedor, cliente y almacén	
<b>Descripción principal</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	Para poder responder la petición del cliente realizamos la búsqueda (nombre del producto).	Se muestra la vista de búsqueda de productos.
<b>Descripción secundario y/o alternativo</b>	Una vez hecha la búsqueda se elabora la nota de venta con todos los datos del cliente y se registra la venta.	Genera cotización para imprimirla que muestra la nota de venta.
	Se solicita un producto para importar o si no existe en almacenes.	
	Espera de llegada del producto para su posterior registro de venta.	Presenta el formulario de registro de venta.

**Fuente:** (Elaboración Propia).

**Figura N° 3. 5.** Diagrama de casos de uso Almacén



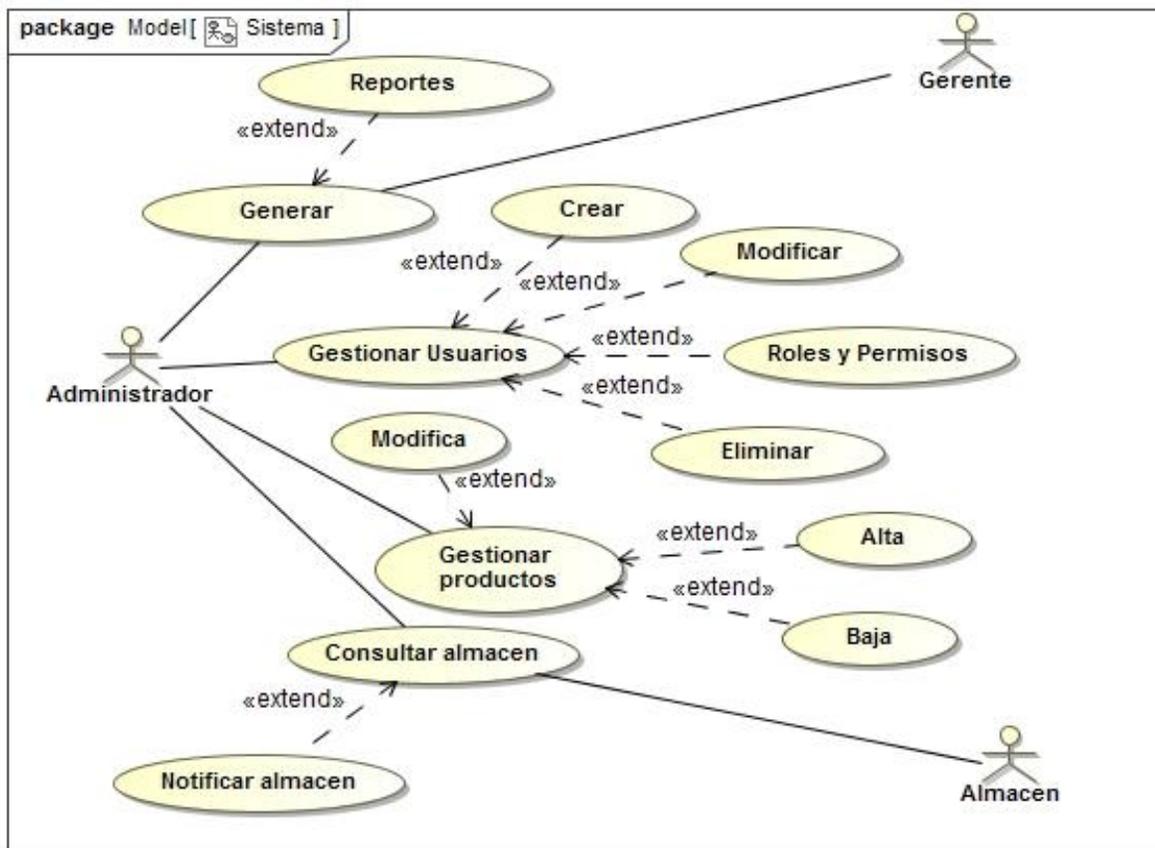
**Fuente:** (Elaboración Propia).

**Tabla N° 3. 11.** Especificación de los casos de uso Almacén

Nombre	Gestionar productos	
Actor	Almacén, gerente	
Descripción principal	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	Realiza las siguientes operaciones de gestión como ser: altas, bajas, modificaciones o búsquedas.	Se muestra la gestión de productos.
Descripción secundario y/o alternativo	Recibe un mensaje de error al realizar una operación incorrecta.	Muestra un mensaje de error.

**Fuente:** (Elaboración Propia).

**Figura N° 3. 6.** Diagrama de casos de uso Gestionar Usuario.



**Fuente:** (Elaboración Propia).

**Tabla N° 3. 12.** Especificación de los casos de uso Gestionar usuario

Nombre	Gestionar usuarios.	
Actor	Administrador, gerente y almacén	
Descripción principal	Eventos Actor	Eventos Sistema
	Se realiza las operaciones de gestión tales como: altas, bajas, modificaciones o eliminación.	Se muestra la vista de gestión de cuentas de usuario.
Descripción secundario y/o alternativo	Recibe el mensaje de error al realizar una inadecuada operación.	Recibe el mensaje de error al realizar y se registra.

**Fuente:** (Elaboración Propia).

**Tabla N° 3. 13.** Especificación de los casos de uso Generar reportes

Nombre		Generar reportes.	
Actor	Administrador, gerente y almacén		
Descripción principal	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
	Se ingresa al módulo de reportes.	Se muestra la vista de gestión de reportes.	
	Ingresa y busca el tipo de reporte y proporciona la información requerida.	Muestra el formulario para la emisión del reporte elegido.	
Descripción secundario y/o alternativo	Recibe el mensaje de error al realizar una mala operación.	Recibe el mensaje de error al realizar y se registra.	

**Fuente:** (Elaboración Propia).

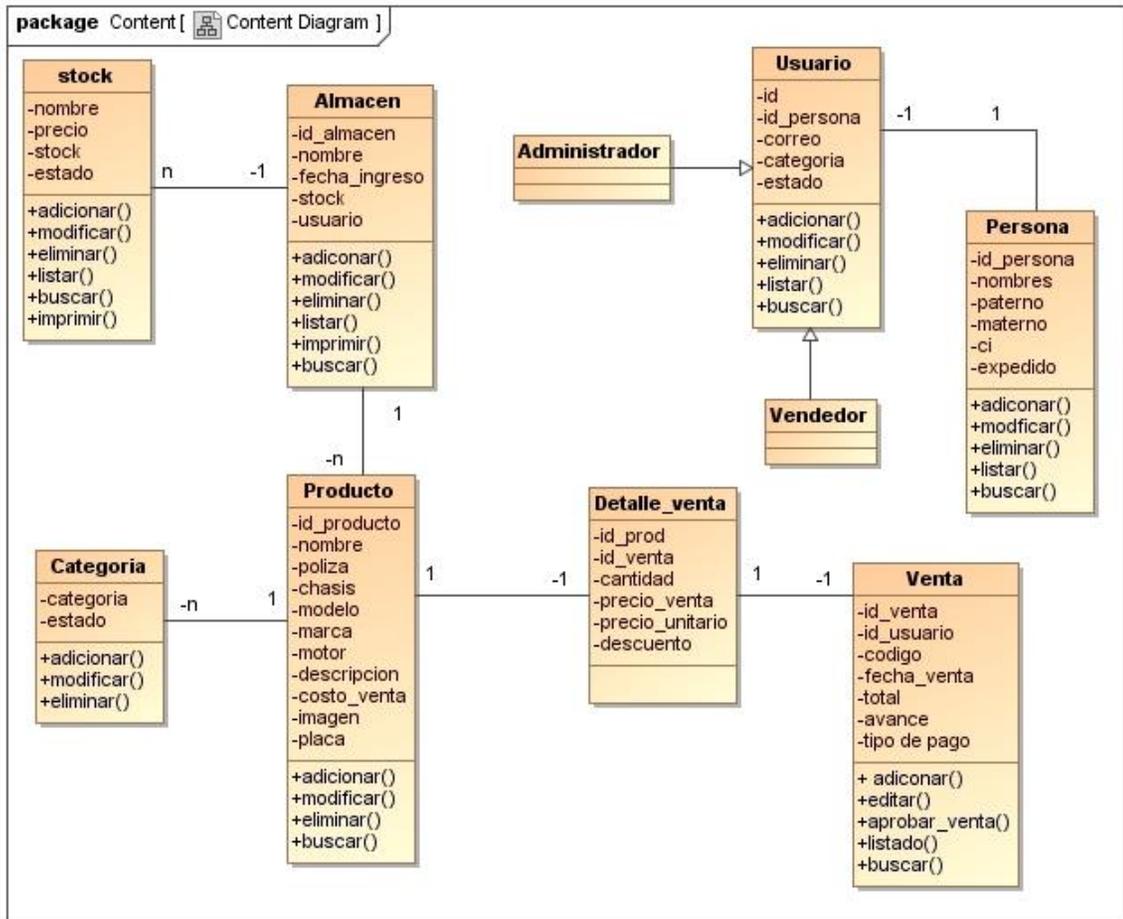
### 3.2. MODELO DE CONTENIDO

Contiene elementos estructurales que proporcionan una importante visión de los requisitos de contenido para una webApp. Además incluye todas las clases de análisis: entidades visibles para el usuario que se crean o manipulan conforme con la web app. El modelo de contenido se deriva a partir de un examen cuidadoso de los casos.

- ✓ Definición de objetos de contenido: puede ser una descripción textual de un producto. Los objetos de contenido se extraen de los casos de uso al examinar la descripción del escenario para referencias directas o indirectas al contenido.

En este presente modelo de contenido nos muestra la relación existente en la empresa:

**Figura N° 3. 7.** Diagrama de contenido del Sistema



**Fuente:** (Elaboración Propia).

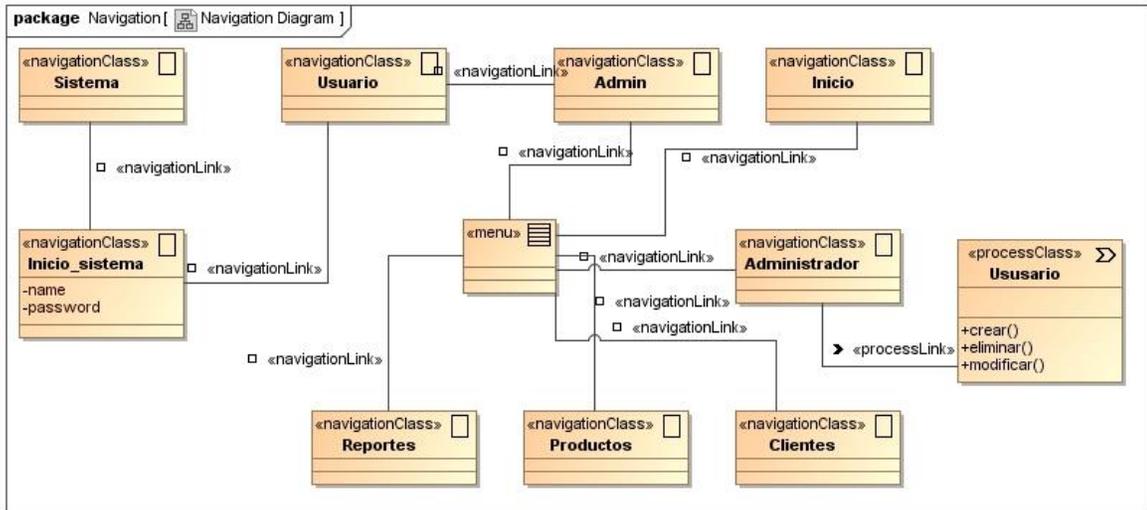
### 3.3. MODELO DE NAVEGACIÓN

El modelo de navegación describe los caminos que para cada meta debe recorrer el usuario en una página web.

Por lo general, el producto tendrá más de un usuario, y cada usuario puede tener varios objetivos, por lo que los modelos como este pueden ser muy extensa.

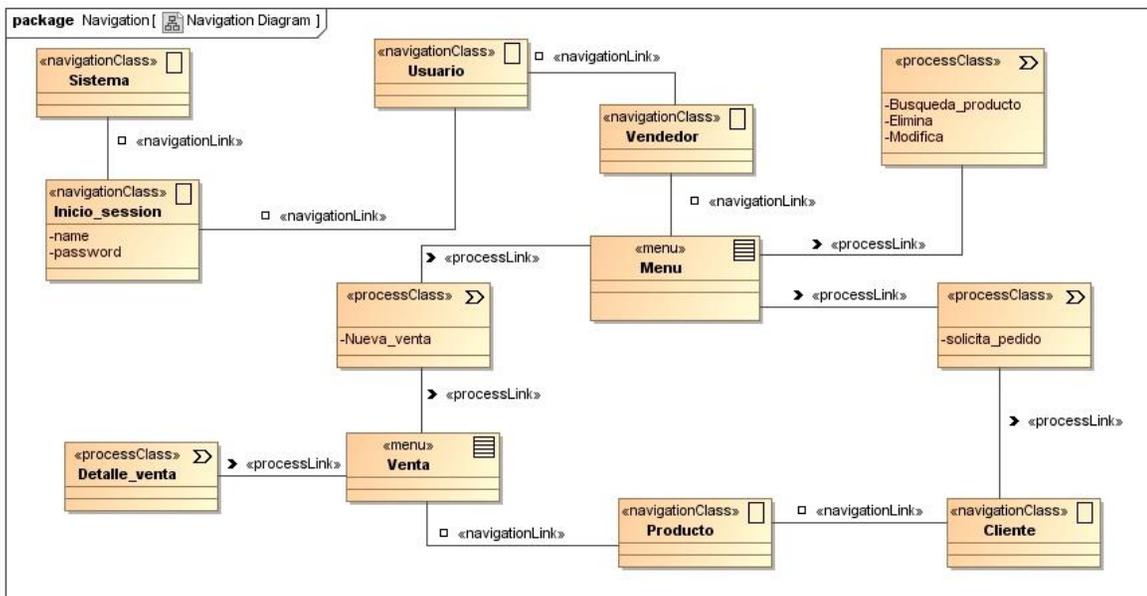
La creación de un modelo de navegación es también una oportunidad para revisar cualquier contenido existente que pueda tener la web y entender el contexto en el que se va a crear la estructura.

**Figura N° 3. 8. Diagrama de Navegación del Sistema Admin**



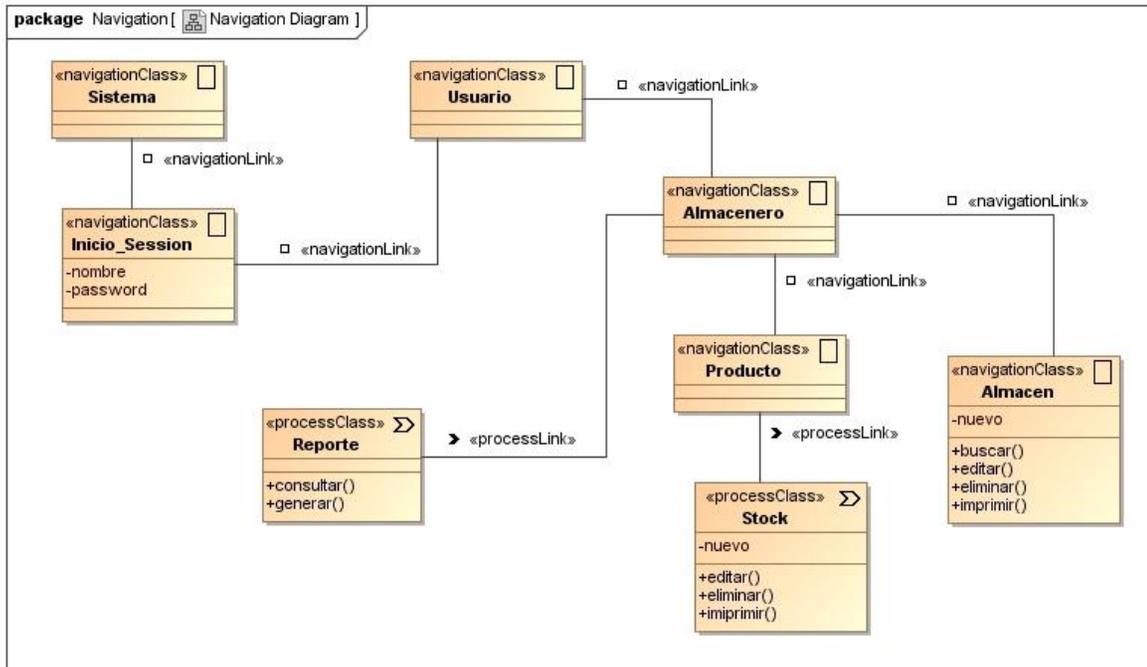
Fuente: (Elaboración Propia).

**Figura N° 3. 9. Diagrama de Navegación del Sistema Vendedor**



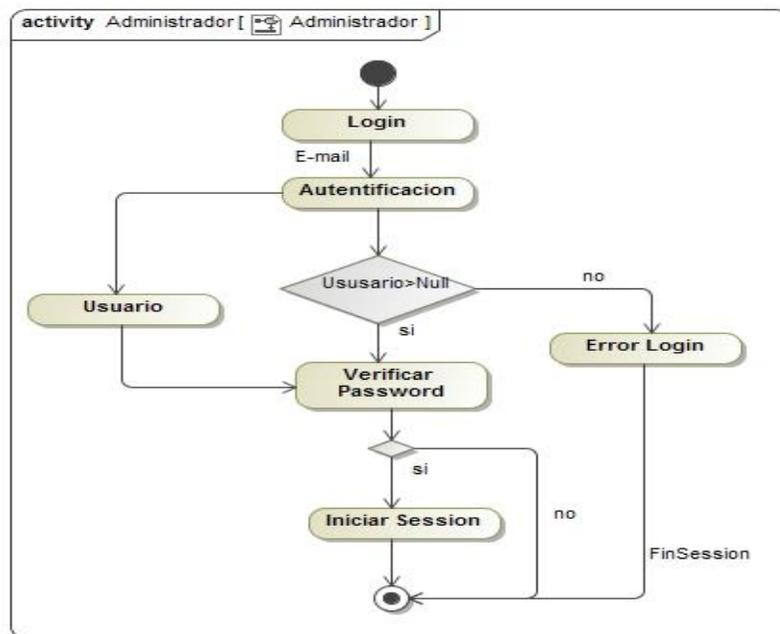
Fuente: (Elaboración Propia).

**Figura N° 3. 10.** Diagrama de Navegación del Sistema Almacén



Fuente: (Elaboración Propia).

**Figura N° 3. 11.** Diagrama de Actividad Autenticación Usuario

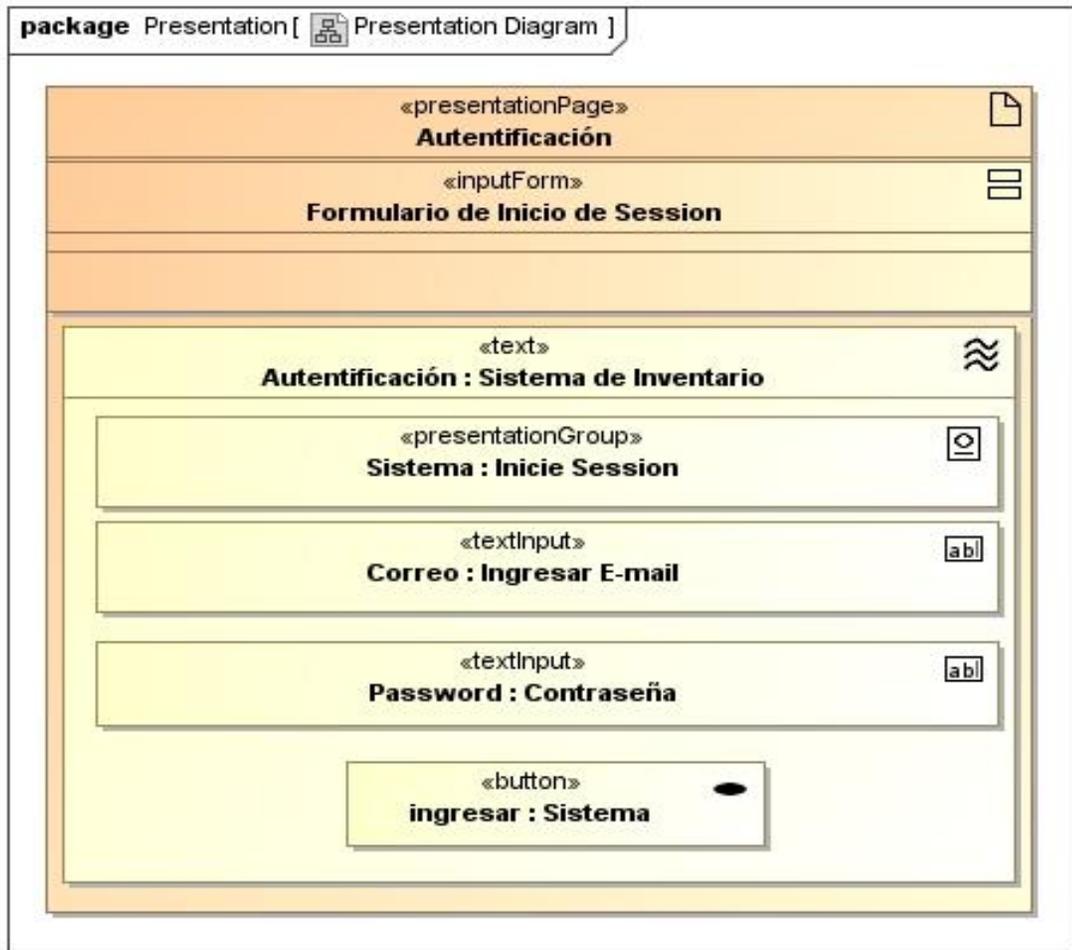


Fuente: (Elaboración Propia).

### 3.4. MODELO DE PRESENTACIÓN

En la Figura N°3.12. Nos muestra el modelo de autenticación del usuario para ingresar al sistema para interactuar con los módulos que contiene el sistema de información.

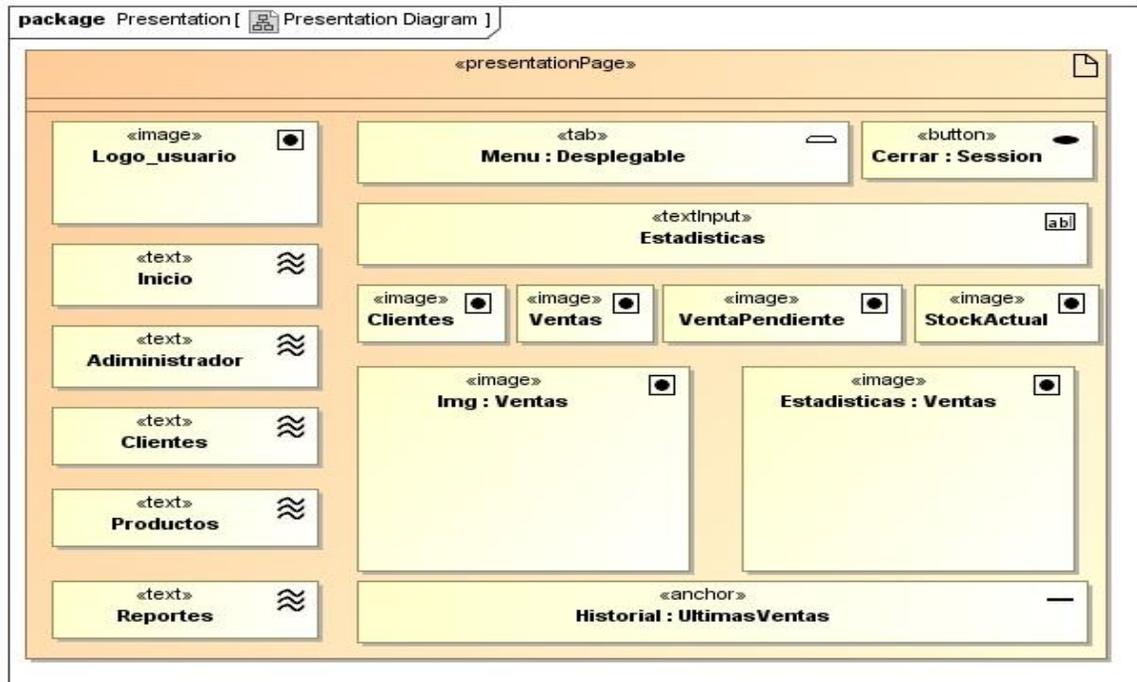
Figura N° 3. 12. Modelo de Autenticación



Fuente: (Elaboración Propia).

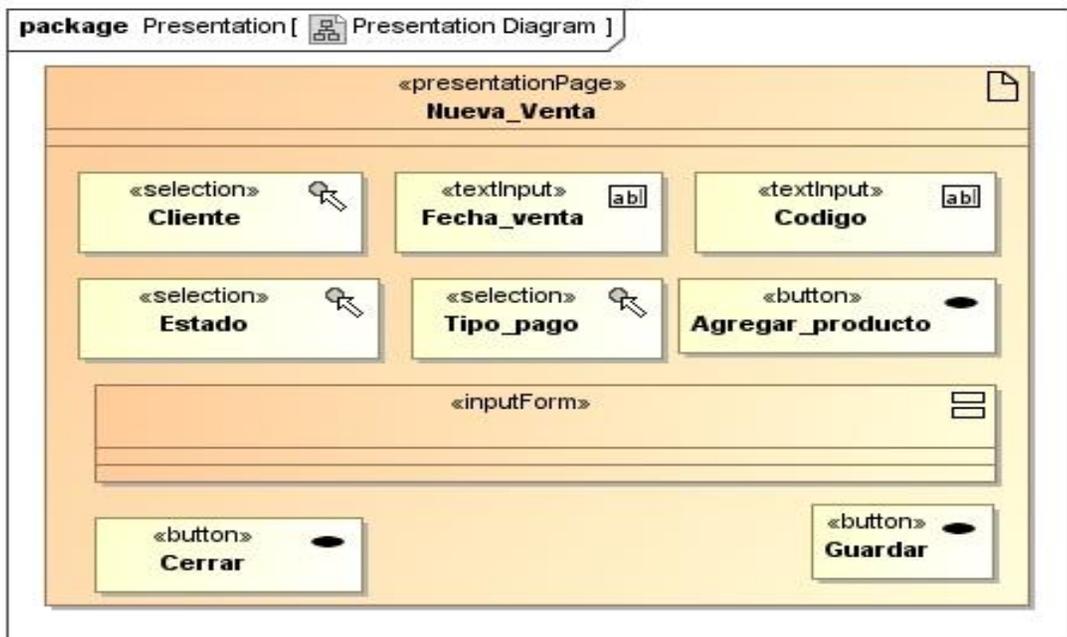
En la Figura N°3.13 nos muestra en el modelo de presentación del contenido y bienvenida al sistema a través del ingreso al Sistema por el usuario. Como se va observar los módulos y las estadísticas.

Figura N° 3. 13. Modelo de presentación interface Admin



Fuente: (Elaboración Propia).

Figura N° 3. 14. Modelo de Presentación Nueva Venta



Fuente: (Elaboración Propia).

### 3.5. MODELO DE IMPLEMENTACIÓN

En la etapa de modelo de implementación consta del desarrollo del sistema y sus interfaces que la componen, tomando en cuenta el diseño elaborado en UWE. A continuación la autenticación del sistema con la encriptación Crypt con la medida de seguridad.

**Figura N° 3. 15.** Interface de Autenticación de Usuarios.

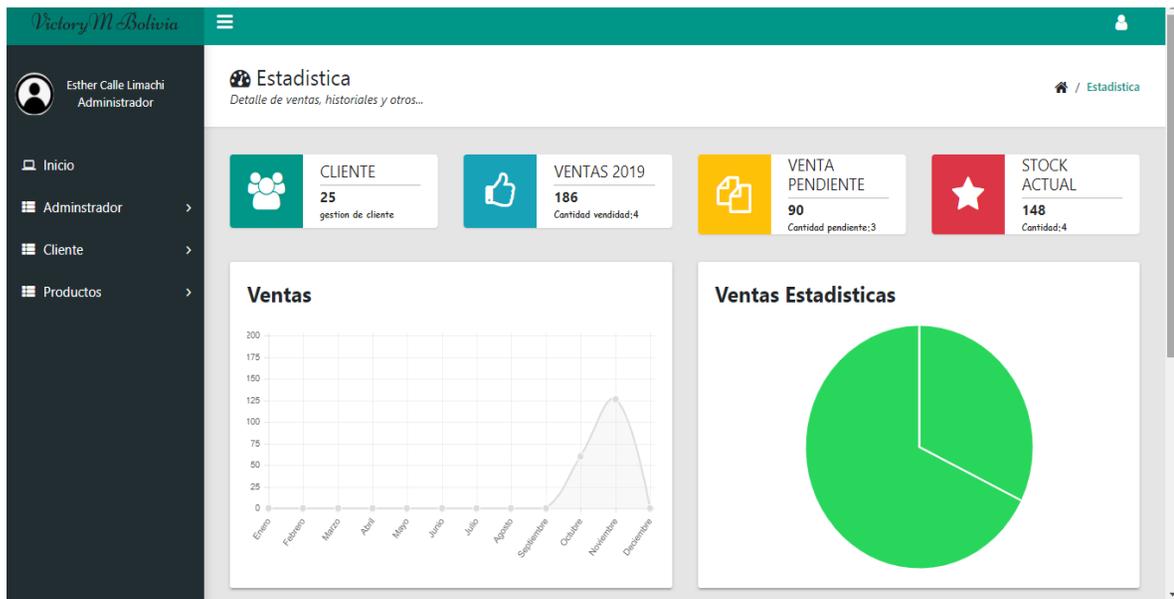


The image shows a login interface for a system titled "SISTEMA DE INVENTARIO". The interface features a teal header with the system name in white. Below the header is a white login form titled "Inicie Session" with a user icon. The form contains two input fields: "CORREO" and "PASSWORD". Below the fields is a teal button with a right-pointing arrow and the text "INGRESAR".

**Fuente:** (Elaboración Propia).

Cuando el usuario ingresa al sistema autenticado inmediatamente se dirige a la pantalla principal del sistema como se ve en la Figura N°3.16

**Figura N° 3. 16. Pantalla Interfaz de Menú**



**Fuente:** (Elaboración Propia).

En la Figura N°3.17 podemos observar el listado del menú del Sistema

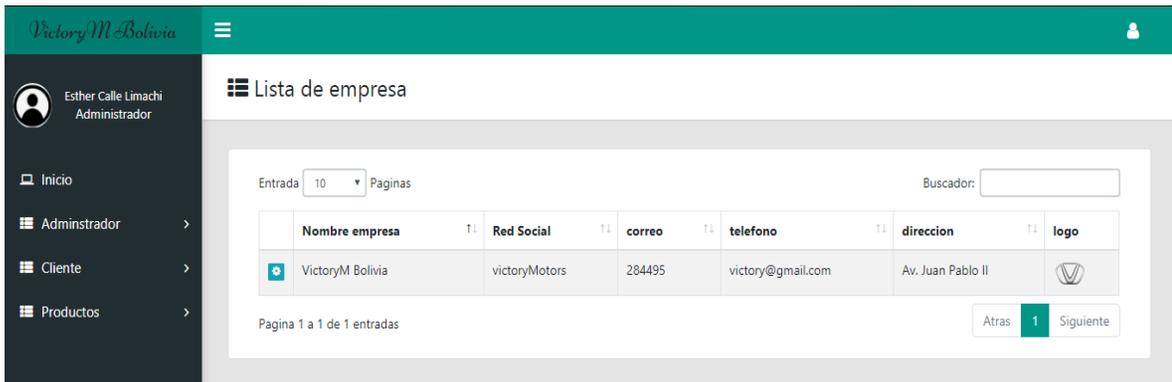
**Figura N° 3. 17. Interfaz del Listado de Menú del Sistema**

The interface includes a search bar and pagination controls (Entrada: 10, Paginas). A 'Nueva Menu' button is visible in the top right corner.

Nombre menu	titulo	Url / Ruta	Icons
almacen	Almacen	almacen	fa-table
cliente	Lista Cliente	cliente	fa-table
compras	Compras	compras	fa-table
empresa	Empresa	empresa	fa-table
fabrica	Listar Fabrica	fabrica	fa-table
grupo	Grupo Menu	grupo	fa-table
inventario	Stock	inventario	fa-table
menu	Menu	menu	fa-table
permiso	Permisos	permiso	fa-table

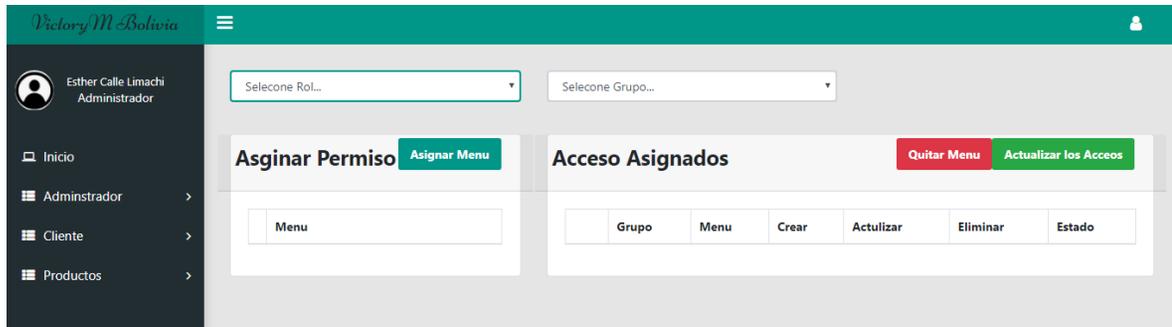
**Fuente:** (Elaboración Propia).

**Figura N° 3. 18. Interfaz Perfil de la Empresa**



**Fuente:** (Elaboración Propia).

**Figura N° 3. 19. Interfaz Permisos a los Usuarios**



**Fuente:** (Elaboración Propia).

**Figura N° 3. 20. Interfaz Rol y Categorías de Usuarios**



**Fuente:** (Elaboración Propia).

**Figura N° 3. 21.** Interfaz Nuevo Rol/Categoría

Registrar Nuevo Rol / Categoría

Nombre rol Descripción

Agregar Rol

**Fuente:** (Elaboración Propia).

**Figura N° 3. 22.** Interfaz Almacén

Victoria M Bolivia

Esther Calle Limachi Administrador

Lista de Almacén

Nueva Almacen Imprimir Almacen

Entrada 10 Paginas Buscador:

	Nombre Producto	Fecha	stock	Usuario
+	Kinglong	2019-11-09	50	esther
+	Kingstar	2019-11-07	80000	jessica
+	susuki	2019-11-07	63	jessica

Pagina 1 a 3 de 3 entradas

Atras 1 Siguiente

**Fuente:** (Elaboración Propia).

**Figura N° 3. 23.** Interfaz Nuevo Almacén

Registrar Nuevo Almacen

Nombre Producto Fecha de Almacen Stock

Selecciona producto.. fecha venta

Agregar Almacen

**Fuente:** (Elaboración Propia).

**Figura N° 3. 24. Interfaz Usuario**

Victory M Bolivia

Esther Calle Limachi  
Administrador

Lista de Usuario Nueva Usuario

Entrada 10 Paginas Buscador:

	Usuario	Persona	Correo	Categoria	Estado
<input type="checkbox"/>	esther	Esther	esther@gmail.com	Administrador	Activo
<input type="checkbox"/>	jessica	jessica	admin@gmail.com	Administrador	Activo
<input type="checkbox"/>	pruebas	cliente	pruebas@gmail.com	Administrador	Activo

Pagina 1 a 3 de 3 entradas Atras 1 Siguiente

**Fuente:** (Elaboración Propia).

**Figura N° 3. 25. Interfaz Nuevo Usuario**

Registrar Nuevo Usuario

Nombre usuario

Correo

Persona

Rol

Contraseña

Agregar Usuario

**Fuente:** (Elaboración Propia).

Figura N° 3. 26. Interfaz Persona

Victory M Bolivia

Esther Calle Limachi  
Administrador

Lista de Persona

Nueva Persona

Entrada: 10 Paginas Buscador:

	Nombre Persona	Apellido Paterno	Apellido Materno	Cedula de Identidad	Expedido
	Esther	Calle	Limachi	78946152	LP
	jessica	moner	clemente	666446	LP
	luis	conde	delgado	78946	OR
	pablo	mita	vaes	147852	CBBA

Pagina 1 a 4 de 4 entradas

Atras 1 Siguiente

Fuente: (Elaboración Propia).

Figura N° 3. 27. Interfaz Nueva Persona

Registrar Nuevo Persona

Nombre completo paterno materno

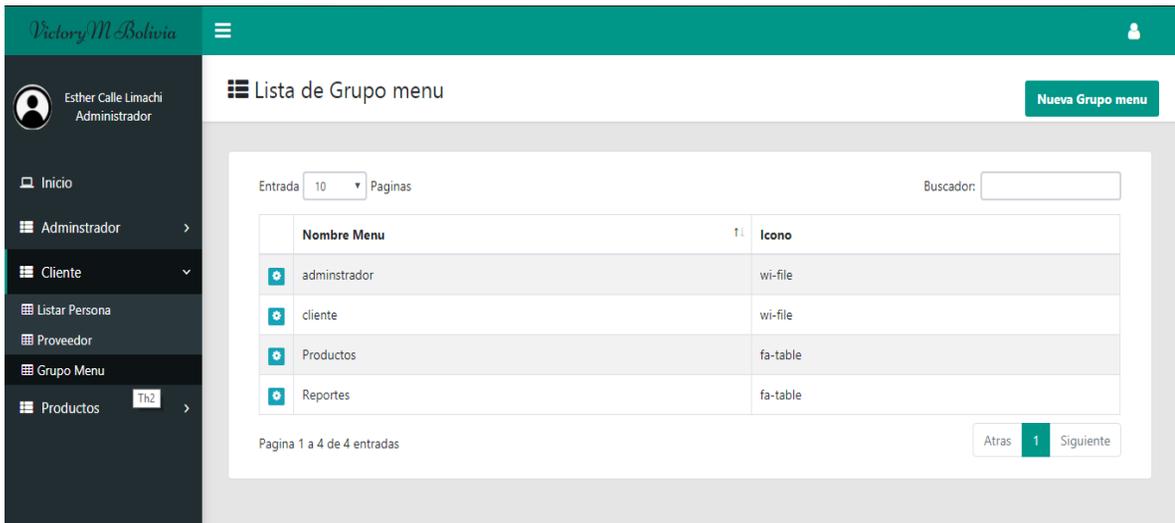
Cedula de Idetidad Expedido

Seleccione Expedido...

Agregar Persona

Fuente: (Elaboración Propia).

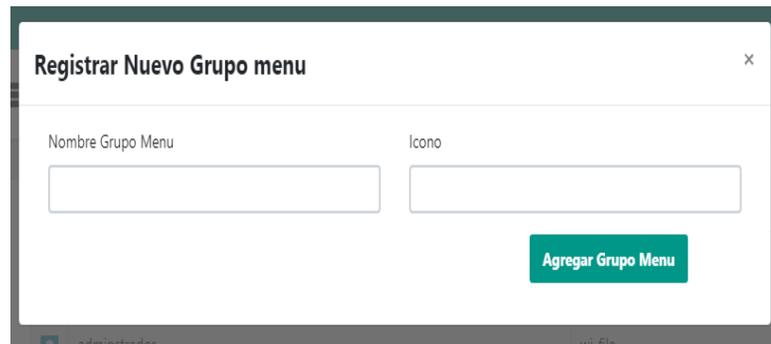
**Figura N° 3. 28.** Interfaz Grupo de Menú



*Figura N° 3. 28*

**Fuente:** (Elaboración Propia).

**Figura N° 3. 29.** Interfaz Nuevo Grupo



**Fuente:** (Elaboración Propia).

**Figura N° 3. 30. Interfaz Cliente**

Victory M Bolivia

Esther Calle Limachi  
Administrador

Lista de Cliente Nueva Cliente

Entrada 10 Paginas Buscador:

	Nombre del cliente	Telfeno	correo	direccion
+	Alvaro Medrano	78946546	alvi@hotmail.com	C. Hyunay
+	Diego Hunca	78941256	huanca@hotmail.com	C. Cochabamba N°45
+	Luis Conde	78946546	lucho@yahoo.es	C. Los Alamos N°1
+	wilmer	79464413	quispe@hotmail.com	C. Almirante Grau

Pagina 1 a 4 de 4 entradas Atras 1 Siguiente

**Fuente:** (Elaboración Propia).

**Figura N° 3. 31. Interfaz Nuevo Cliente**

Registrar Nuevo Cliente

Nombre completo cliente

Telefono

Correo

Direccion

Agregar Cliente

**Fuente:** (Elaboración Propia).

Figura N° 3. 32. Interfaz Producto

Esther Calle Limachi  
Administrador

Inicio

Administrador

Cliente

Productos

Lista de Producto

Nueva Producto

Entrada 10 Paginas Buscador:

	Codigo	Nombre del producto	modelo	Fabrica	descripcion	costo de venta	logo	estado
	C- 456	apv2019	susuki	Empresasss	auto importado desde europa	35500.00		Activo
	C-123	kinglong2019	Kinglong	foton	auto con 14 asientos comodoss	243950.00		Activo
	C-789	king2017	Kingstar	hander	Minibus con 8 asientos super comodoss	28900.00		Activo

Pagina 1 a 3 de 3 entradas

Atras 1 Siguiente

Fuente: (Elaboración Propia).

Figura N° 3. 33. Interfaz Nuevo Producto

Registrar Nuevo producto

Codigo

Nombre producto

modelo

descripcion

fabrica

Costo

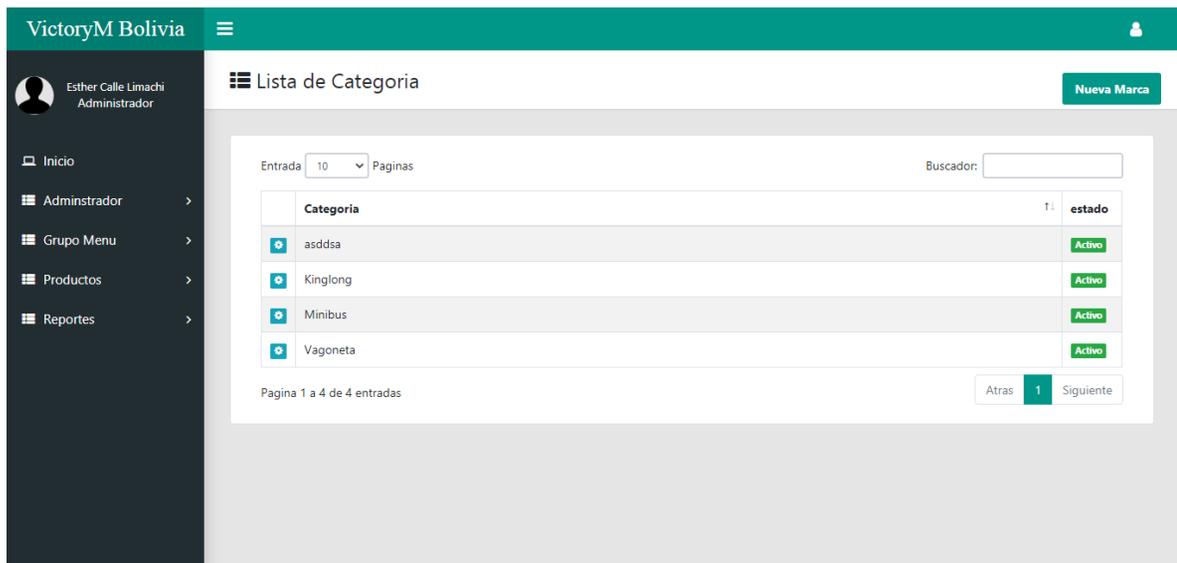
Subir logo

Logo

Actualizar

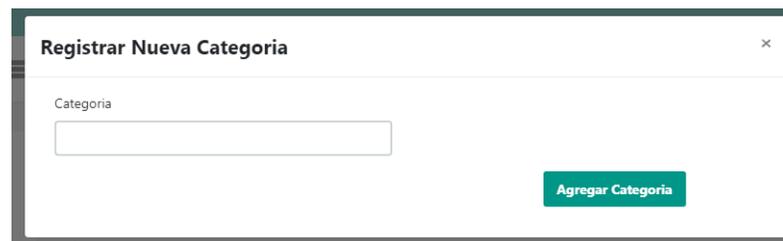
Fuente: (Elaboración Propia).

**Figura N° 3. 34. Interfaz Categoría**



**Fuente:** (Elaboración Propia).

**Figura N° 3. 35. Interfaz Nueva Categoría**



**Fuente:** (Elaboración Propia).

**Figura N° 3. 36. Interfaz Ventas**

Lista de Ventas

Realiza Nueva venta

Entrada: 10 Paginas Buscador:

Codigo	Nombre Cliente	Fecha venta	Total	Avance	tipo Pago	Vendedor
12-12	rewtttttt	2019-10-16	24.00	cerrado	efectivo	jessica
12-12-12	rewtttttt	2019-10-15	24.00	cerrado	cheque	jessica
123	ewq	2019-10-26	36.00	pendiente	efectivo	jessica
456	ewq	2019-11-07	30.000.00	pendiente	efectivo	jessica
56-89	rewtttttt	2019-10-16	12.00	cerrado	efectivo	jessica
78-788	wilmer	2019-10-07	24.00	pendiente	efectivo	jessica
C- 456	Diego Hunca	2019-11-09	14.000.00	pendiente	efectivo	jessica
9999	ewq	2019-11-06	24.00	conclada	efectivo	jessica
V1	pruebas tes	2019-11-07	126.000.00	cerrado	cheque	jessica

Página 1 de 9 Entradas

Atras 1 Siguiente

Fuente: (Elaboración Propia).

**Figura N° 3. 37. Interfaz Nueva Venta**

Nueva Venta

Cliente: Diego Hunca Fecha de Venta: 2019-11-09 Codigo: C-456

Estado: Pendiente Tipo de Pago: Efectivo

Agregar Producto

Accion	Codigo	Cant.	Descripcion	Precio Unit.	Precio Total
	C-789	2	Kingstar	7000.00	14000

Total: 14,000.00 Bs

cerrar Guardar

Fuente: (Elaboración Propia).

**Figura N° 3. 38.** Interfaz Agregar Producto a la Nueva Venta

**Agregar Producto**

busca producto

Fabrica Cantidad Precio

busca producto cantidad fecha venta

Stock: 00

**Agregar Producto**

**Fuente:** (Elaboración Propia).

**Figura N° 3. 39.** Interfaz Stock

*Victoria M Bolivia*

Esther Calle Limachi  
Administrador

**Lista de Stock** **Nueva Stock** **Imprimir Stock**

Entrada 10 Paginas Buscador:

	Nombre Producto	Precio	stock	Estado
	Kingstar	7000.00	20	Activo
	susuki	48000.00	48	Activo

Pagina 1 a 2 de 2 entradas **Atras** 1 **Siguiente**

**Fuente:** (Elaboración Propia).

**Figura N° 3. 40.** Interfaz Nuevo Stock

**Registrar Nuevo Stock**

Nombre producto Precio Stock

Selecciona producto..

**Agregar Stock**

**Fuente:** (Elaboración Propia).

### 3.6. PRUEBAS DEL SISTEMA

#### 3.6.1. PRUEBA DE CAJA NEGRA

Las pruebas de caja negra se centran principalmente en lo que “se quiere” de un módulo, es una manera de encontrar casos específicos en ese modulo que atiendan a su especificación. Las pruebas de caja negra son, pruebas funcionales dedicadas a “mirar” en el exterior de lo que se prueba. Estas pruebas se denominan de varias formas, pruebas de caja “opaca”, pruebas de entrada/salida, pruebas inducidas por datos.

A continuación detallaremos los que se tomaran en cuenta Registro de un producto, Proveedor y Venta de un producto para realizar la prueba de caja negra.

**Tabla N° 3. 14.** Interfaz del Registro de un Nuevo Producto

Prueba N°1 Interfaz de Registro de un Producto			
PRUEBA	ENTRADA O ACCIÓN DEL USUARIO	RESULTADO DEL SISTEMA	¿RESULTADO?
P1.	E-mail y contraseña del usuario correctos.	Si el sistema acepta los datos ingresados correctamente entonces puede ingresar al menú o Página Principal.	SI
P2.	E-mail y contraseña del usuario incorrectos.	El sistema automáticamente muestra el mensaje de error al ingreso al sistema con el usuario y contraseña.	SI
P3.	Elegir en el menú desplegable El registro de un producto.	El sistema debe de adicionar el nuevo registro del producto.	SI

<b>P4.</b>	Ingresar los datos del Producto como ser: código, nombre del Producto, Modelo, descripción, categoría, costo y el logo.	El sistema tiene el botón de la decisión de guardar los datos ingresados.	SI
------------	---	---	----

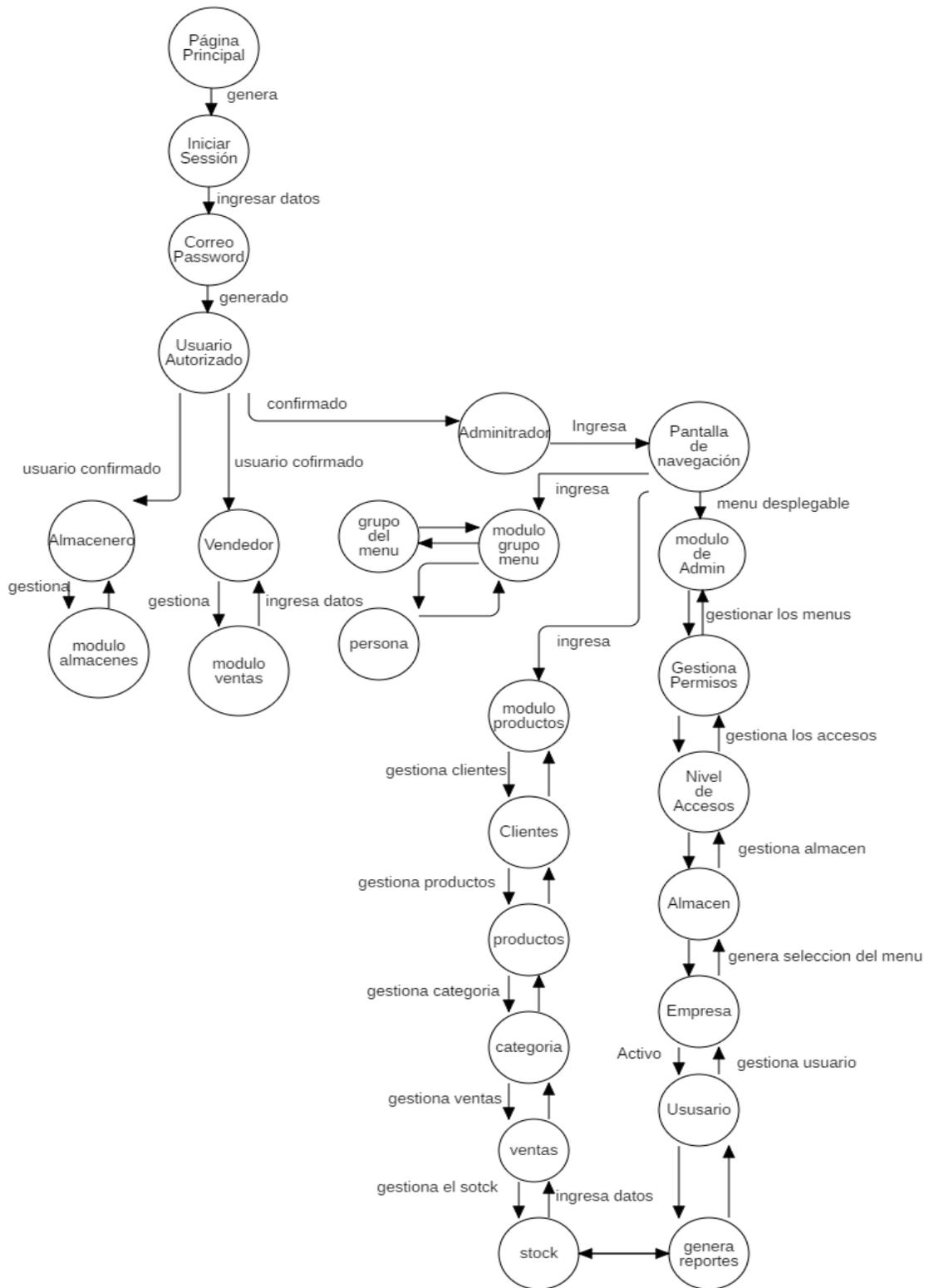
**Fuente:** (Elaboración Propia).

**Tabla N° 3. 15.** Interfaz de la Nueva Venta

<b>Prueba N°1 Interfaz de Venta de un Producto</b>			
<b>PRUEBA</b>	<b>ENTRADA O ACCIÓN DEL USUARIO</b>	<b>RESULTADO DEL SISTEMA</b>	<b>¿RESULTADO?</b>
<b>P1.</b>	El usuario debe de escoger en el menú desplegable la opción de la venta de un producto.	Ingresar al menú o a la interfaz de la venta.	SI
<b>P2.</b>	Ingresar al interfaz de ventas para ingresar los datos.	El sistema contiene el interfaz del formulario para la nueva venta.	SI
<b>P3.</b>	Los datos que deben contener son: cliente, fecha de venta, código, estado, tipo de pago, el botón agregar producto, con las acciones de cerrar y guardar los datos.	El sistema debe de adicionar la nueva venta para preparar la nota de venta.	SI

**Fuente:** (Elaboración Propia).

**Figura N° 3. 41. Gráfico general**



**Fuente:** (Elaboración Propia).

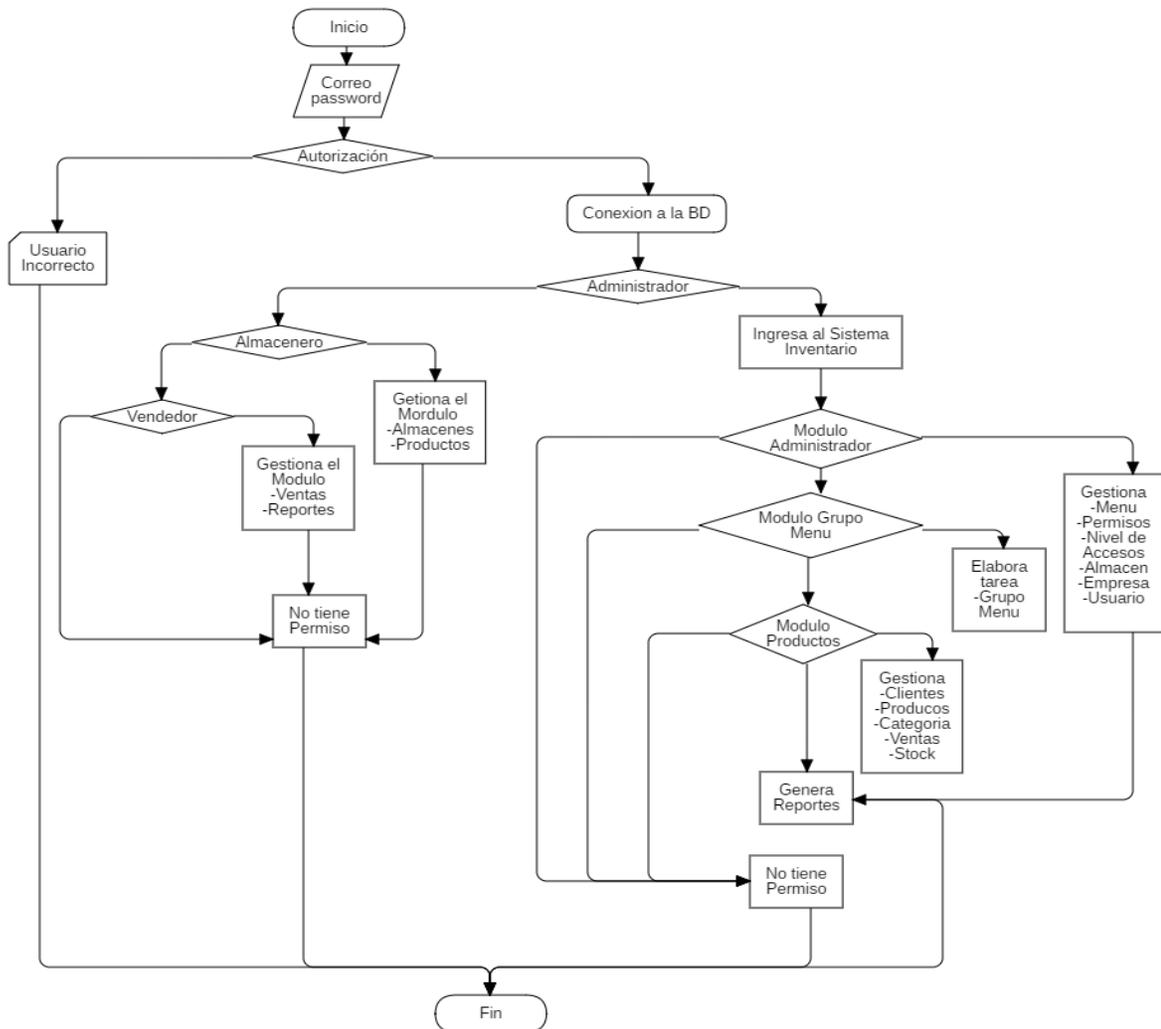
### 3.6.2. PRUEBAS DE CAJA BLANCA

En este tipo de prueba, observamos todo el código con la noción de probar el desenvolvimiento del sistema recorrido por cada uno de los casos presentados por los algoritmos que se utilizaron en la codificación, es decir son casos de prueba que se aplica al código fuente.

En las pruebas de caja blanca se consideran lo siguiente:

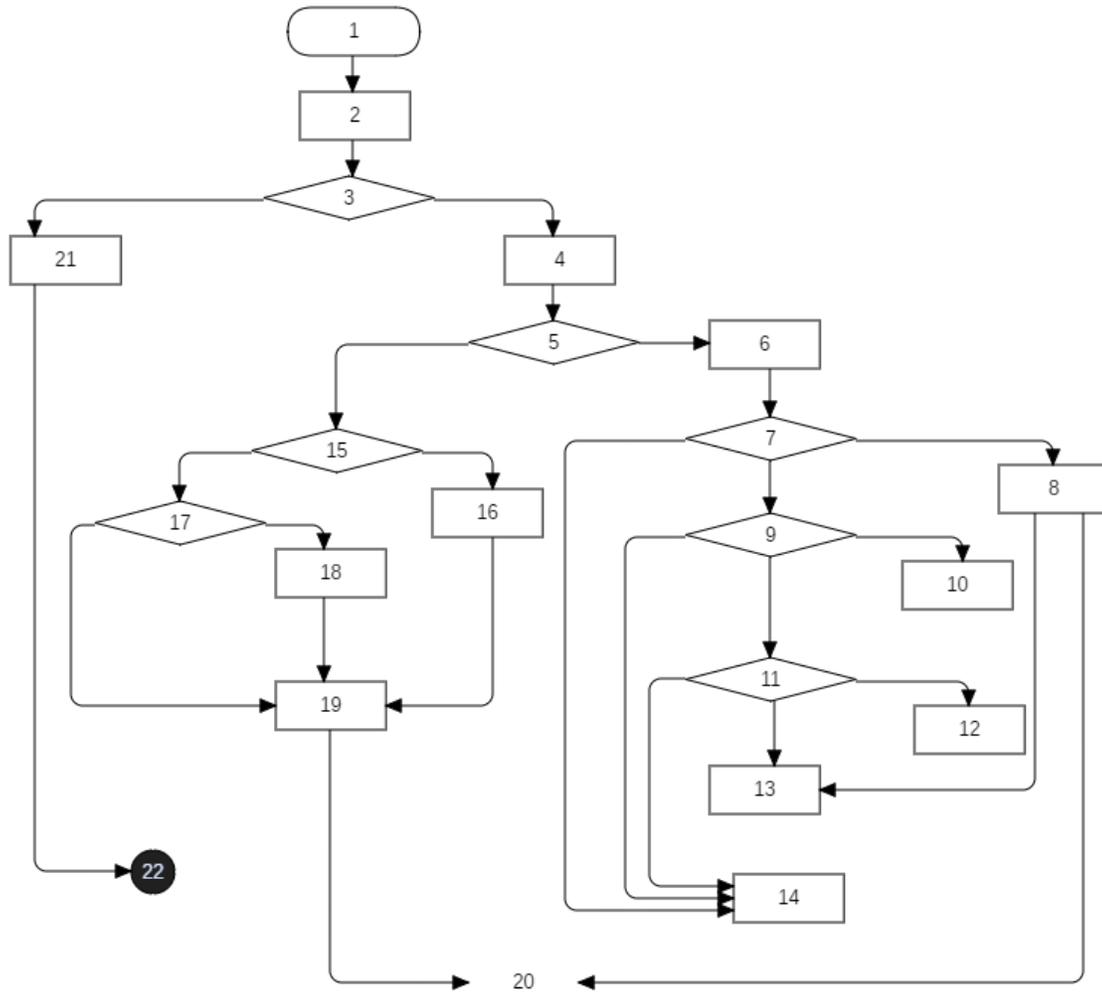
- ✓ Se realizaran las pruebas utilizando el conocimiento del funcionamiento interno del código.

**Figura N° 3. 42.** Diagrama de uso general



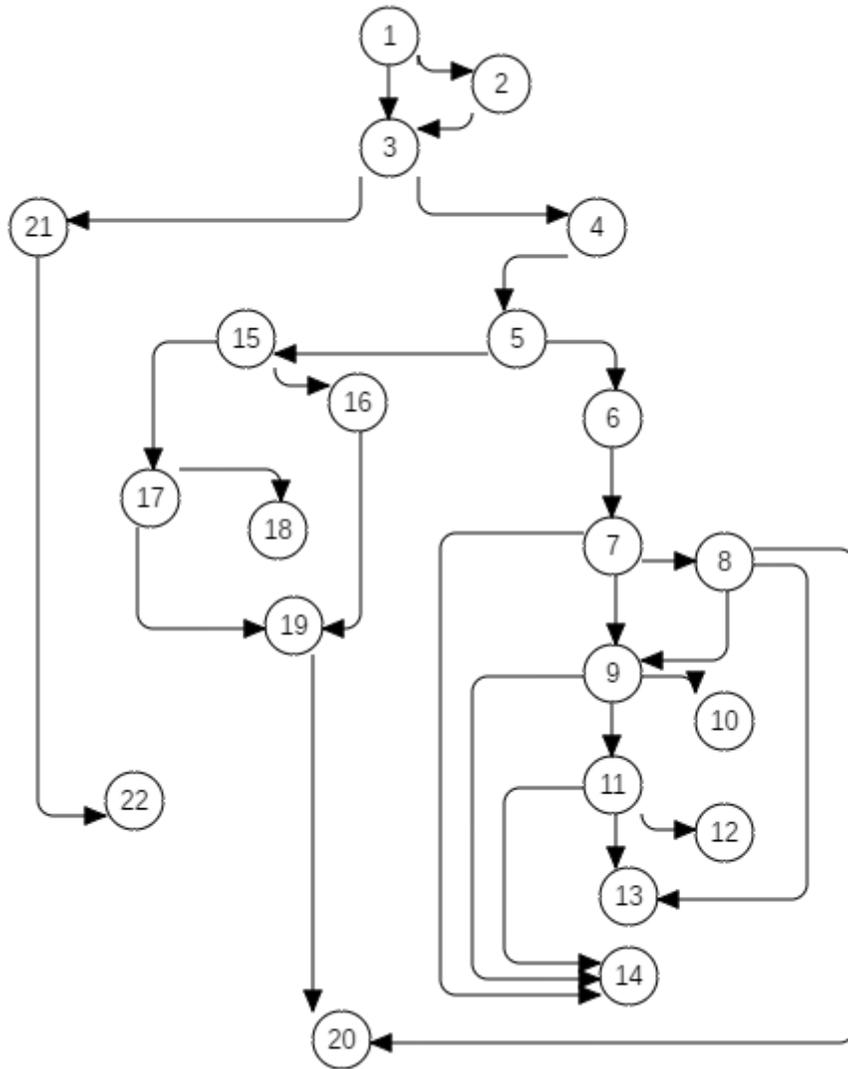
**Fuente:** (Elaboración Propia).

**Figura N° 3. 43. Diagrama de Flujo General**



**Fuente:** (Elaboración Propia).

**Figura N° 3. 44.** Grafo de Flujo



**Fuente:** (Elaboración Propia).

Calculo de la complejidad ciclomática:

La complejidad ciclomática  $V(G)$  para un gráfico de flujo  $G$  se define como

$$V(G) = A - N + 2$$

Dónde:  $A$  es el número de aristas del grafo = 28

$N$  el número de nodos del grafo = 22

$$V(G) = 28 - 22 = 6 + 2 = 8$$

El valor de  $V(G) = 8$  nos indica que son ocho los casos de pruebas que deben de ejecutarse y diseñar para garantizar que se cubren las sentencias del programa.

**Sacamos caminos independientes:**

Camino 1: 1 – 3 – 21 – 22

Camino 2: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 14

Camino 3: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 14

Camino 4: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 13

Camino 5: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 20

Camino 6: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 11 – 14

Camino 7: 1 – 2 – 3 – 4 – 5 – 15 – 16 – 19 – 20

Camino 8: 1 – 2 – 3 – 4 – 5 – 15 – 17 – 19 – 20



**CAPÍTULO IV**  
**CALIDAD Y SEGURIDAD**

**CAPITULO IV**  
**CALIDAD Y SEGURIDAD**

**4.1. MÉTRICAS DE CALIDAD DE SOFTWARE**

En este capítulo se realizará la métrica de calidad de software mediante la norma ISO 9126 con las características que son: la funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

**4.1.1. FUNCIONALIDAD**

La funcionalidad no se puede medir directamente, por la cual tomaremos el factor de ponderación para hallar el punto función que cuantifica el tamaño y la complejidad del sistema en términos de las funciones del usuario.

Los puntos función se derivan usando una relación empírica basada en medidas contables (directas) del dominio de información del software y en valoraciones cualitativas de la complejidad del software.

**Tabla N° 4. 1.** Factores de Ponderación

Valor de medida de información	Conteo	Factor ponderado			Total
		Simple	Promedio	Complejo	
Entrada de Usuario	28	3	4	6	112
Salida de Usuario	14	4	5	7	70
Peticiones de Usuarios	12	3	4	6	72
Número de Archivos	5	7	10	15	50
Interfaces Externas	1	5	7	10	7
Conteo Total					311

**Fuente:** (Elaboración Propia).

**Tabla N° 4. 2.** Valores de ajuste de Complejidad

Factor de Complejidad	Valoración del factor						Total
	No influye	Incidental	Moderado	Medio	Significativo	Esencial	
¿Requiere el Sistema copias de Seguridad y de recuperación fiables?	0	1	2	3	4	5	5
¿Se requiere comunicación de datos?	0	1	2	3	4	5	4
¿Existen función de procesamiento distribuido?	0	1	2	3	4	5	4
¿Es crítico el rendimiento?	0	1	2	3	4	5	3
¿Sera ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	0	1	2	3	4	5	3
¿Requiere el sistema entrada de datos interactiva?	0	1	2	3	4	5	4
¿Requiere la entrada de datos interactiva que las transacciones se lleven a cabo en múltiples pantallas?	0	1	2	3	4	5	3
¿Se actualizan los archivos maestros de forma interactiva?	0	1	2	3	4	5	4
¿Son complejas las entradas, las salidas los archivos o las peticiones?	0	1	2	3	4	5	3
¿Es complejo el procesamiento interno?	0	1	2	3	4	5	3
¿Se ha diseñado el código para ser utilizado?	0	1	2	3	4	5	5
¿Están incluidas en el diseño la conversión y la instalación?	0	1	2	3	4	5	3
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	0	1	2	3	4	5	3
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente usada por el usuario?	0	1	2	3	4	5	5
<b>Total</b>							<b>52</b>

**Fuente:** (Elaboración Propia).

Para calcular los puntos de función se utiliza la siguiente ecuación.

$$PF = \text{Conteo Total} * (0,65 + 0,01 * \sum fi)$$

Donde:

Punto Función: Medida de Funcionalidad entregada a la aplicación como valor de normalización.

Conteo Total: Es el nivel de complejidad del sistema respecto al usuario.

0,01: Factor de conversión, es decir un error del 1%.

0,65: Valor mínimo del ajuste de la complejidad.

$\sum Fi$ : Es la sumatoria de los valores de ajuste complejidad.

Reemplazando en la ecuación.

$$PF = 311 * (0,65 + 0,01 * 52)$$

$$PF = 363.87$$

Observando la siguiente tabla, el sistema tiene una funcionalidad óptima, ya que el punto función se encuentra en el intervalo mayor a 300.

**Tabla N° 4. 3.** Escala de Punto Función

Escala	Observación
PF>300	Optimo
200<PF<300	Bueno
100<PF<200	Suficiente
PF<100	Deficiente

**Fuente:** (Roger S. Pressman, 2010)

Si calculamos al 100% el nivel de confianza considerado  $\sum Fi = 70$  como el máximo valor de ajuste de complejidad entonces se tiene:

$$PFmax = \text{Conteo Total} * (0,65 + 0,01 * \sum Fi)$$

Reemplazando valores:

$$PFmax = 363 * (0,65 + 0,01 * 70)$$

$$PFmax = 490.05$$

Con el máximo valor de ajuste de complejidad se tiene la funcionalidad real:

$$\text{Funcionalidad} = \frac{PF}{PFmax}$$

$$\text{Funcionalidad} = \frac{363.87}{490.05} = 0,74$$

$$\text{Funcionalidad} = 0,74 * 100 = 74\%$$

Entonces la funcionalidad del sistema es del 74%, esto quiere decir que el sistema tiene 74% de probabilidad de funcionamiento sin riesgo de fallo y con operatividad constante y el 26% de colapso del sistema.

#### **4.1.2. USABILIDAD**

La usabilidad es la facilidad de uso, esta métrica nos muestra el costo de aprender el producto, lo cual se calcula con la siguiente formula:

$$FU = \left( \frac{\sum Xi}{n} \right) * 100$$

**Tabla N° 4. 4. Ajuste de preguntas**

Preguntas	Respuestas		Ponderación
	Si	No	
¿Puede utilizar con facilidad el sistema?	3	0	1
¿Puede controlar operaciones que el sistema solicite?	2	1	0,67
¿Las respuestas del sistema son complicadas?	3	0	1
¿El sistema permitió la retroalimentación de información?	2	1	0,67
¿El sistema cuenta con interfaz agradable a la vista?	3	0	1
¿La respuesta del sistema es satisfactoria?	2	1	0,67
¿Le parece complicadas las funciones del sistema?	1	2	0,67
¿Se hace difícil aprender a manejar el sistema?	2	1	0,67
¿Los resultados que proporciona el sistema facilitan el trabajo?	3	0	1
¿Durante el uso del sistema se produjo errores?	1	2	0,67
Total			8,02

**Fuente:** (Roger S. Pressman, 2010)

Reemplazando en la fórmula:

$$FU = \left(\frac{8,02}{10}\right) * 100 = 80,02$$

Existe un 80,02%, de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

#### **4.1.3. CONFIABILIDAD**

La confiabilidad se define como la probabilidad de que un sistema, activo o competente lleve a cabo su función adecuadamente durante un periodo bajo condiciones operacionales previamente definidas y constantes.

Para calcular la confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y se obtiene muestras, donde se encuentra:

$$F(t) = f * e^{(-\mu * t)}$$

Donde:

$f$ , funcionalidad del sistema.

$\mu$ , Probabilidad de error que puede tener el sistema.

$t$ , tiempo que dura una gestión en el sistema.

En el inicio de ejecución  $t_0=0$ , lo que significa el tiempo inicial el cual dará inicio el funcionamiento del sistema.

El término en el cual el sistema trabaja sin falla está en la siguiente ecuación:

$$P(T \leq t) = F(t) \quad \text{Probabilidad de fallas.}$$

$$P(T \leq t) = 1 - F(t) \quad \text{Probabilidad de trabajo sin fallas.}$$

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla.

Conociendo la funcionalidad del 0,74 del sistema calculamos para el periodo establecido.

$$F(t) = f * e^{\left(-\frac{1}{10} * 20\right)}$$

$$F(t) = 0,74 * e^{\left(-\frac{1}{10} * 20\right)}$$

$$F(t) = 0,1001 * 100$$

$$F(t) = 10\%$$

$$P(T \leq t) = F(t) = P(T \leq t) = 0,100 * 100\% = 10\%$$

$$P(T \leq t) = 1 - F(t) = P(T \leq t) = 1 - 0,100 = 0,9 = 90\%$$

Por lo tanto en un periodo de prueba de 20 días, el sistema muestra un índice de confiabilidad de 90%.

#### 4.1.4. MANTENIBILIDAD

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos que la empresa tenga y los reglamentos que está regida por la misma. El estándar IEE94 sugiere un índice de madurez del software (IMS) que proporciona un indicador en la estabilidad de un producto, se lo determina con la siguiente formula:

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$$

Donde:

*Mt*, número de módulos de la versión actual.

*Fa*, numero de módulos en la versión actual que se han añadido.

*Fc*, numero de módulos en la versión actual que se han cambiado.

*Fd*, numero de módulos de la versión anterior que se han borrado en la versión actual.

Reemplazando los valores:

$$IMS = \frac{[5 - (1 + 0 + 0)]}{5} = 0.8$$

Por tanto se puede decir el nuevo sistema tiene una estabilidad de 80% que indica la facilidad de mantenimiento, el 20% restante es el margen de error correspondiente a los cambios y modificaciones.

#### 4.1.5. PORTABILIDAD

El sistema es portable, si el costo de transportar y adaptar la aplicación a un nuevo entorno es menor al costo de rediseñar el sistema para dicho ambiente, la relación que permite determinar el grado el grado de portabilidad es la siguiente:

Portabilidad la del servidor.

A nivel se sistema operativo el Sistema de Información para el Control de inventario de Ventas, es portable bajo los siguientes operativos de la familia Windows y todas las distribuciones de sistemas operativos libre de Linux.

## **4.2. SISTEMA DE GESTION DE SEGURIDAD DE LA INFORMACION ISO-27002**

La ISO 27002, forma parte de la serie de normas 27000 verifica y evalúa su implementación, en cuanto la seguridad se detallara los siguientes aspectos.

### **4.2.1. SEGURIDAD DE BASE DE DATOS**

Se usó como base de datos MySQL. En cuanto a la forma de resguardo se realiza:

- ✓ Cuando una acción del usuario en el sistema requiere o solicita algunos registros de la base datos, existe una conexión segura para esta acción.
- ✓ Para la seguridad de datos del sistema se tienen registrado de nombre de usuario y contraseña de acceso, según su nivel de acceso pueda realizar actividades en el sistema.

La información en una empresa es muy valiosa, por tanto su resguardo es fundamental, la conexión a la base de datos y el cierre de la conexión es de forma automática.

### **4.2.2. SEGURIDAD LÓGICA**

- ✓ **Gestión de Comunicación y Operaciones**

Los respaldos (Back-up) de la base de datos del sistema, se deberá realizar con precauciones como sigue Tabla N° 4.5.

**Tabla N° 4. 5.** Gestión de Comunicación y Operaciones

<b>DESCRIPCION</b>	<b>DURACION</b>
<b>En periodos de registro de usuarios</b>	1 vez por semana
<b>En periodo de registro del producto en almacenes</b>	1 vez por semana
<b>En periodo de registro del stock del producto.</b>	1 ves por semana
<b>En periodo de registro de las ventas</b>	1 ves por semana

**Fuente:** (Elaboración Propia)

Al usuario Administrador se recomienda cambiar periódicamente el password.

#### **4.2.3. SEGURIDAD FÍSICA**

##### ✓ **Seguridad Física y del Entorno**

Se recomienda realizar, Back-up más de tres copias, estos sean almacenados en distintos lugares.

Los back-up de la base de datos, deberán estar protegidas y tener acceso al personal autorizado.

#### **4.2.4. SEGURIDAD ORGANIZATIVA**

La información del sistema debe tener un nivel de protección, como ser. El manejo de los back-up de acuerdo a las fechas en el que se realizan las copias.

#### **4.2.5. SEGURIDAD DEL SISTEMA**

Para la seguridad del sistema se consideran las siguientes precauciones, los cuales son:

- ✓ Autenticación de los usuarios
- ✓ Manejo de privilegios y tipos de usuarios en el sistema.
- ✓ Manejo de vistas.
- ✓ Integridad y control de datos por url's
- ✓ Encriptación con el algoritmo HASH en las contraseñas o passwords.

Una función hash es método para generar claves o llaves que representen de manera unívoca a un documento o conjunto de datos. Es una operación matemática que se realiza sobre este conjunto de datos de cualquier longitud, y su salida es una huella digital, de tamaño fijo e independiente de la dimensión del documento original. El contenido es ilegible.

Es posible que existan huellas digitales iguales para objetos diferentes, porque una función hash tiene un número de bits definido. En el caso del SHA-1, tiene 160bits, y los posibles objetos a resumir no tienen un tamaño límite. A partir de un hash o huella digital, no podemos recuperar el conjunto de datos originales. Los más conocidos son

el MD5 y el SHA-1, aunque actualmente no son seguros utilizarlos ya que se han encontrado colisiones. Cifrar una huella digital se conoce como firma digital.

Requisitos que deben cumplir las funciones hash:

- ✓ Imposibilidad de obtener el texto original a partir de la huella digital.
- ✓ Imposibilidad de encontrar un conjunto de datos diferentes que tengan la misma huella digital (aunque como hemos visto anteriormente es posible que este requisito no se cumpla).
- ✓ Poder transformar un texto de longitud variable en una huella de tamaño fijo (como el SHA-1 que es de 160bits).
- ✓ Facilidad de empleo e implementación.

#### **4.2.6. SEGURIDAD DE LA APLICACIÓN**

Se desarrolla un módulo de control de acceso al sistema para la restricción del acceso a usuario no autorizado. Este módulo verifica y autoriza a los usuarios por medio de permisos que son otorgados por el Admin del sistema, haciendo uso de las sesiones de PHP. Se realiza el registro del usuario que modifica la información la base de datos, para esto se registra en cada tabla el identificar del usuario que modifica la información.



**CAPÍTULO V**  
**ANÁLISIS DE COSTOS**

### 5.1. ANÁLISIS DE COSTOS

Para calcular el esfuerzo y tiempo requerido para el desarrollo de un producto es necesario primero hallar el KLDC (kilo-líneas de Código).

#### Calculo de KLDC

$$LDC = 11,135$$

$$KLDC = \frac{11,135}{1000}$$

$$KLDC = 11,14$$

Dentro de los modos de desarrollo que define el método, se utilizara el orgánico, debido a que el proyecto tiene una complejidad intermedia, el tamaño no supera a 50 KLDC y las restricciones no son complejas, además COCOMO define tres modelos de estimación, (básico, intermedio y avanzado) que se diferencia en los resultados. Para el presente proyecto se aplicara el modelo intermedio y lo detallamos en la siguiente tabla:

**Tabla N° 5. 1** Coeficiente para medir el costo del software

PROYECTO DE SOFTWARE	A	B	C	D
Orgánico	3.20	1.05	2.50	0.38

**Fuente:** (Roger S. Pressman, 2010).

Factor de ajuste de Esfuerzo, resultado de la multiplicación de los atributos que cuantifican el entorno de un proyecto. Para hallar los valores de FAE, se utilizara siguiente tabla:

**Tabla N° 5. 2.** Atributos y sus valores usados en el cálculo de FAE

Atributos	Valor					
	Muy bajo	Bajo	Normal	Alto	Muy Alto	E. Alto
Atributos de software						
Facilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de Datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de Hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de Personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributo del Proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

**Fuente:** (Elaboración Propia).

$$FAE = 1,15 * 1,00 * 0,85 * 1,11 * 1,00 * 1,00 * 1,07 * 1,00 * 0,82 * 0,86 * 1,00 * 0,95$$

$$* 0,82 * 0,91 * 1,00$$

$$FAE = 0,58$$

### **Cálculo del Esfuerzo de desarrollo**

$$E = a * KLDC^b * FAE$$

$$E = 3,20 * 11,14^{1,05} * 0,58$$

$$E = 23,33 \cong 23 \text{ (personas/mes)}$$

### **Cálculo del Tiempo del desarrollo**

$$T = c * E^d$$

$$T = 2,50 * 23,33^{0,38}$$

$$T = 8,27 \cong 8 \text{ (meses)}$$

### **Cálculo de la Productividad**

$$PR = \frac{LDC}{E}$$

$$PR = \frac{11135}{23,33}$$

$$PR = 477,28 \cong 477 \text{ (LDC/personas mes)}$$

### **Cálculo del Personal promedio**

$$P = \frac{E}{T}$$

$$P = \frac{23,33}{8,27}$$

$$P = 2,82 \cong 3 \text{ (personas)}$$

### **Cálculo del Costo promedio**

$$\text{Costo} = P * \text{Salario} * T$$

$$\text{Costo} = 3 * 500 * 8$$

$$\text{Costo} = 12,000 \text{ \$us}$$

### **5.1.1. RESUMEN DEL ANALISIS DE COSTO**

En conclusión, se requiere 3 personas para programar con un salario de 500 bs, con el tiempo de trabajo de 8 meses, con el coste total del proyecto finalizado de 12,000 \$us, equivalente en Bolivianos 83,640 Bs.



**CAPÍTULO VI  
CONCLUSIONES Y  
RECOMENDACIONES**

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1. CONCLUSIONES

En conclusión tomando a los objetivos planteados en el presente proyecto desarrollado para la empresa Victory Motors Bolivia.

##### 6.1.1. TOMANDO LOS OBJETIVOS

Detallaremos a continuación tomando en cuenta los objetivos pronunciados anteriormente en el Capítulo I.

##### 6.1.1.1. TOMANDO EL OBJETIVO PRINCIPAL

Se cumple al determinar el proyecto con el objetivo general desarrollar e implementar el sistema de información para el control de inventario de ventas para la empresa Victory Motors Bolivia que le permita administrar la información de manera oportuna, agilizando y mejorando los procesos para la venta de un producto la cual fue detallada en el capítulo III.

##### 6.1.1.2. TOMANDO LOS OBJETIVOS SECUNDARIOS

- ✓ Realizar un diagnóstico de la situación actual de la empresa para identificar que procesos se deben automatizar, Se realizó el diagnóstico y elaborando la tarea en su totalidad.
- ✓ Determinar los requerimientos y las necesidades relacionados con el sistema que se desarrollara, Se cumplió con las necesidades de la empresa para poder desarrollar el sistema.
- ✓ Diseñar el sistema de inventario para realizar el registro de toda la información de las ventas, Se cumplió al desarrollar el Sistema de Información.

- ✓ Optimizar el tiempo de la búsqueda de los datos existentes para mejorar la organización de la información, Se elaboró las búsquedas en los módulos necesarios.
- ✓ Implementar el sistema de información para evitar falencias en los registros de productos, y mejorar las ventas, Se cumplió al implementar el sistema.

## **6.2. RECOMENDACIONES**

A partir de este trabajo y en base a las observaciones realizadas en la implementación se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema:

- ✓ Se recomienda al usuario cambiar la contraseña de acceso al sistema periódicamente, para mejorar la seguridad y evitar accesos no autorizados.
- ✓ Para resguardar los datos internos, se recomienda realizar copias de seguridad de la base de datos con el fin de evitar la pérdida de la información.
- ✓ Con respecto al análisis y diseño del sistema, cuando se requiera la ampliación y creación de nuevos módulos, se recomienda primero revisar la documentación para poder tomar una buena decisión, ya que el sistema presenta elementos reutilizables que podrían ser utilizados en los módulos nuevos como el módulo de facturación y entre otros.
- ✓ Se recomienda para trabajos futuros utilizar herramientas similares para que facilite el desarrollo del producto.

## **Bibliografía**

- Bertalanffy, V. (1940). *Concepto Sistema*.
- Boehm, B. (1981). *El Modelo COCOMO*. Obtenido de El Modelo COCOMO:  
<http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm>
- Camelo, L. (2010). *Seguridad Informatica Domonios de la ISO-27002*.
- Castro, J. (01 de 10 de 2018). *Control de Inventario*. Obtenido de Control de Inventario: <https://hipodec.up.edu.mx/blog/que-es-control-inventario>
- COBO, Á. y. (2005). *Sistemas gestor de bases de datos*. melledin: Díaz de Santos.
- Del Valle Rodriguez, A. n. (20 de febrero de 2015). *LA INGENIERÍA WEB*. Obtenido de ingenieriaweb: <https://laingenieriaweb.wordpress.com/la-ingenieria-web/>
- Encargado y empleados. (27 de septiembre de 2016). Entrevista. (esther, Entrevistador)
- Fillottrani, P. R. (2007). <https://www.yumpu.com/>. Obtenido de [https://www.yumpu.com/es/document/read/18214221/calidad-en-el-desarrollo-de-software-modelos-de-calidad-de-](https://www.yumpu.com/es/document/read/18214221/calidad-en-el-desarrollo-de-software-modelos-de-calidad-de)
- Filoltrani, P. R. (2007). Modelos de calidad de software Universidad Nacional del Sur.
- Garcia, I. (2017). *Inventario*.
- Gomez, K. (2017). *Ingenieria para el desarrollo del software*.
- Hernández, U. (2015). *Modelo, Vista y Controlador*.
- Lerdorf, R. (21 de junio de 2018). *www.php.net*. Obtenido de [www.php.net](http://www.php.net):  
<https://www.ecured.cu/PHP>
- Lerdort, R. (1995). *Lenguaje PHP*.
- Manson, L. M. (7 de diciembre de 1999). *monografias.com*. Obtenido de <http://www.monografias.com/trabajos/evoinfosoc/evoinfosoc.shtml>
- Minguez Sanz, D., & Garcia Morales, E. J. (2007). *Metodologia web uwe, METODOLOGÍA UWE BASADA EN UML (UWE)*.

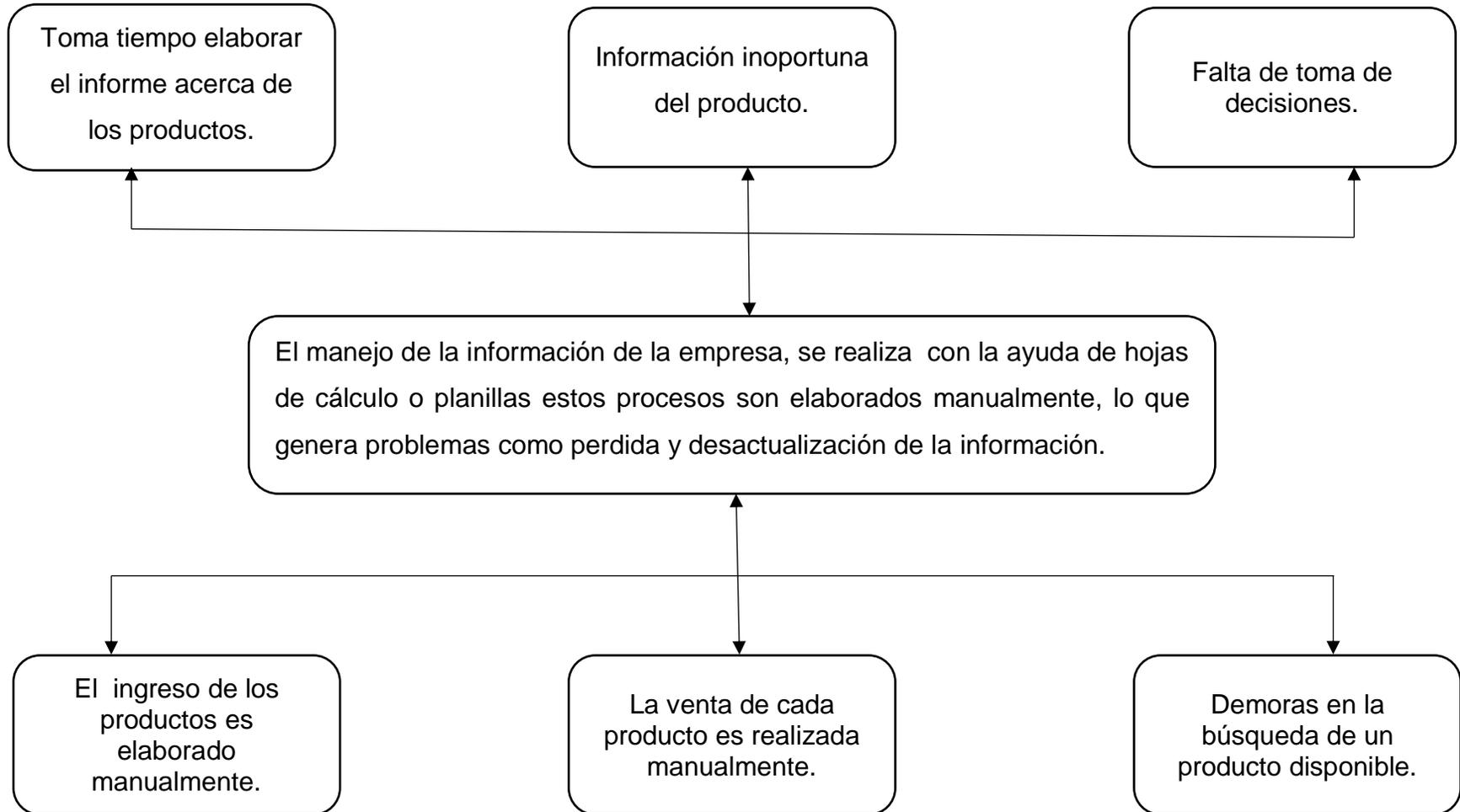
- Otwell, T. (2018). *laravel*. Obtenido de laravel: <https://getcomposer.org/doc/00-intro.md>
- Outwel, T. (19 de abril de 2011). *www.laravel.com*. Obtenido de [www.laravel.com](http://www.laravel.com): <https://laravel.com/docs/6.x>
- Peralta. (2008). *Sistema de informacion*.
- Prentice-Hall. (1981). *El modelo cocomo*.
- Pressman. (2010). *pruebas de caja blanca y negra*. Obtenido de pruebas de caja blanca y negra: <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>
- Pressman, R. S. (2007). *Sistema de informacion almacenamiento entrada proceso y salida*.
- Pressman, R. S. (2013). *Punto Función*. Mexico: Mc Graw Hill.
- Raffino, M. E. (29 de agosto de 2019). Obtenido de <https://concepto.de/servidor-web/>
- Roger S. Pressman, P. (2010). Ingeniería del software. En P. Roger S. Pressman, *Ingeniería del software* (pág. 805). Mexico: S.A. DE C.V.
- Roger S. Pressman, P. (2010). *Lenguaje modelao unificado*. mexico: McGRAW-HILL INTERAMERICANA .
- Roger S. Pressman, P. (2012). *Calidad en la industria del Software Norma ISO-9126*. Mexico: Mc Graw Hill.
- Roger S. Pressman, P. (2012). *Calidad en la industris del Software Norma ISO-9126*. Mexico: Mc Graw Hill.
- Schiaffarino, A. (2015). *Arquitectura Cliente Servidor*.
- Soliz, A. (2012). *Tropiccode.net*. Obtenido de <https://tropiccode.net/seguridad/que-es-y-como-funciona-sha1/>
- Someville. (2011). *Ingeniería de requerimientos*.
- Torrez, D. M. (15 de 06 de 2010). Ingeniería de Software Método de Prueba Universidad Tecnológica de la Mixteca. En D. M. Torrez. Obtenido de <https://www.academia.edu>: [https://www.academia.edu/18053264/Microsoft\\_PowerPoint\\_Tema5\\_Metodos\\_de\\_prueba\\_ppt\\_Modo\\_de\\_compatibilidad](https://www.academia.edu/18053264/Microsoft_PowerPoint_Tema5_Metodos_de_prueba_ppt_Modo_de_compatibilidad)

Valle, A. N. (2009). *Metodología web uwe*. españa: ntics.  
*wikipedia*. (2 de mayo de 2016). Obtenido de <https://es.wikipedia.org/wiki/XAMPP>

# **ANEXOS**

**ANEXO A**  
**ARBOL DE PROBLEMAS**

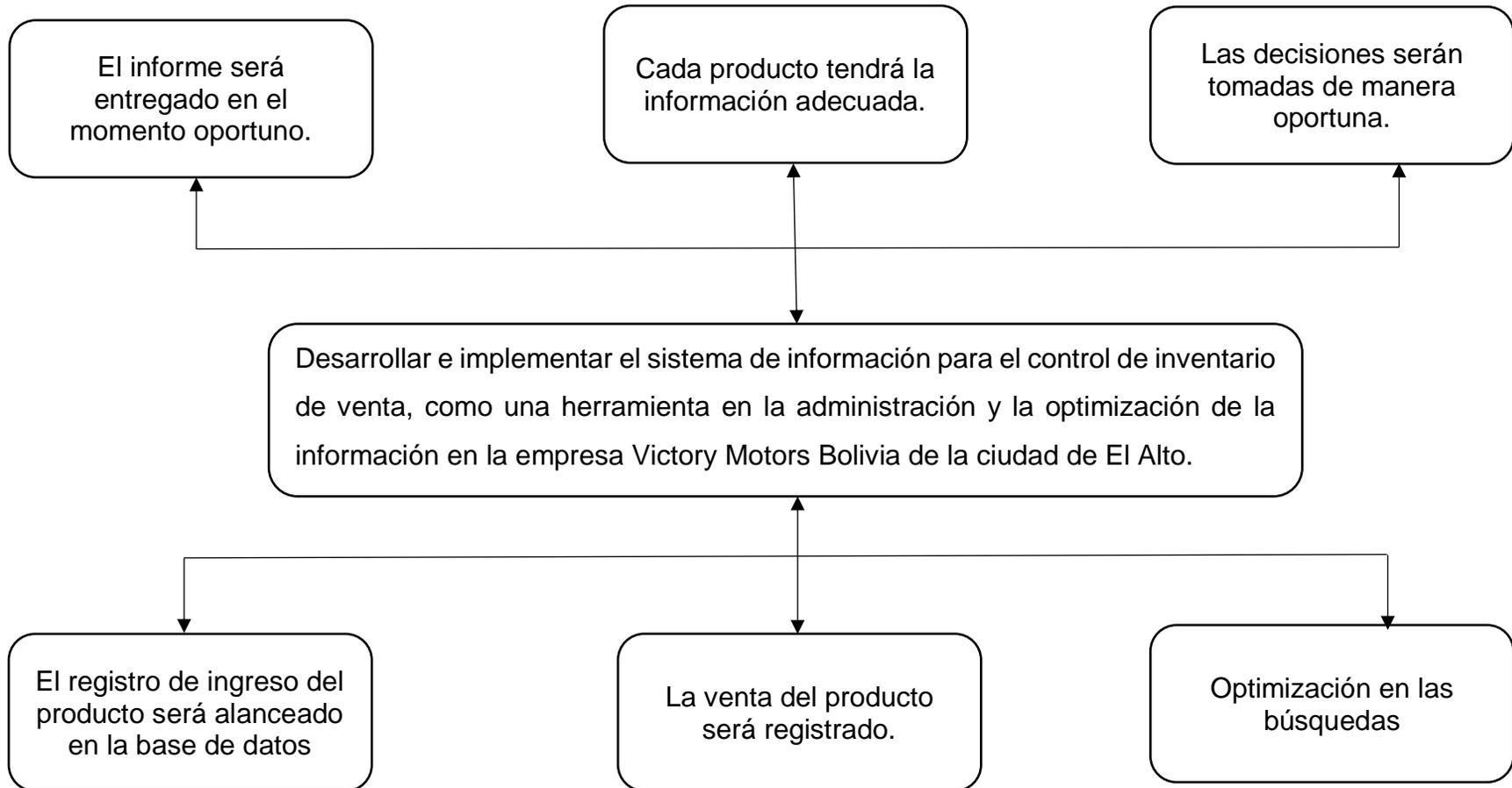
## ÁRBOL DE PROBLEMAS



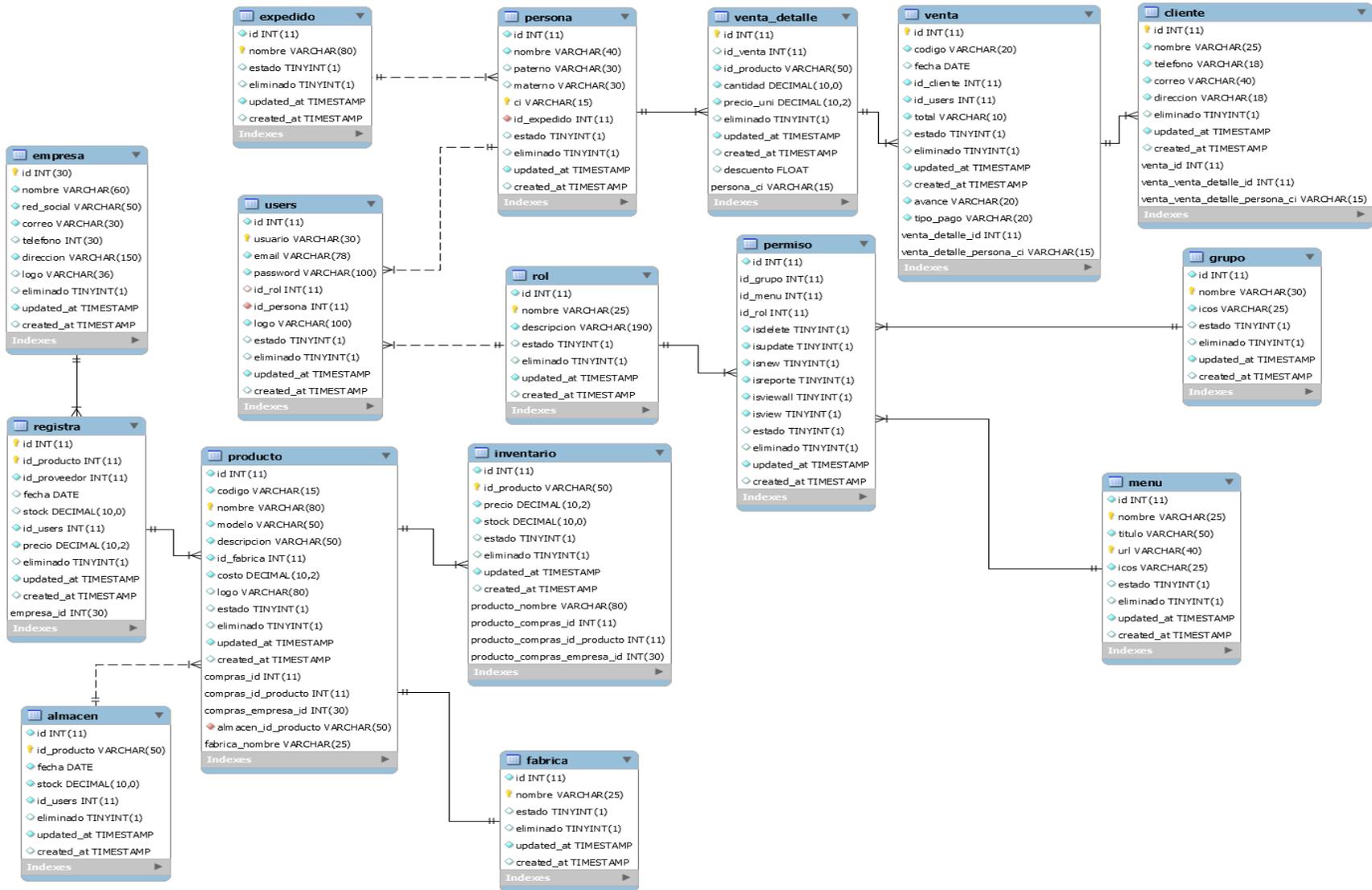
# **ANEXO B**

## **ARBOL DE OBJETIVOS**

## ÁRBOL DE OBJETIVOS



**ANEXO C**  
**MEDLO RELACIONAL DE BASE**  
**DE DATOS**



**ANEXO D**  
**MANUAL DE USUARIO**



VICTORY MOTORS-BOLIVIA  
Av. Juan Pablo II Frente Tam Y Cruz  
Papal  
Contacto: (+591) 70128283

# MANUAL DE USUARIO

SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO DE  
VENTAS

EL ALTO - BOLIVIA

## ÍNDICE DE CONTENIDO

	PÁG.
1. INTRODUCCIÓN .....	1
2. DESCRIPCION DEL SISTEMA .....	1
3. INGRESO AL SISTEMA.....	2
4. SECCIONES DEL SISTEMA.....	3
5. CONFIGURACION DE LA EMPRESA .....	3
6. MENU INICIO.....	4
7. PRODUCTOS .....	5
8. CATEGORIA DEL PRODUCTO.....	6
8. VENTAS.....	6
9. CLIENTES.....	9
10. PERSONAS .....	10
11. GESTION DE USUARIOS .....	10
12. PERMISO .....	11
13. NIVEL DE ACCESO.....	12
14. GRUPO MENU .....	13
15. REPORTES .....	13
16. RECOMENDACIONES .....	14

## 1. INTRODUCCIÓN

El presente sistema, fue desarrollado con el objetivo de automatizar los procesos relacionados a las ventas, y que sea posible administrar la información de manera amigable.



## 2. DESCRIPCION DEL SISTEMA

El sistema de información para el control de inventario contiene los siguientes módulos:

- Inicio
- Administrador
  - Submodulos
    - Menú
    - Permisos
    - Nivel de Accesos
    - Almacén
    - Empresa
    - Usuario
- Grupo menú
  - Submodulos
    - Persona
    - Grupo menú
- Productos
  - Submodulos
    - Cliente
    - Producto
    - Categoría
    - Ventas
    - Stock
- Reportes
  - Submodulo
    - Reportes

Como también, el sistema cuenta con las características:

- Acceso a usuarios autorizados
- Niveles de operatividad, cada tipo de usuario tiene asignado un rol dentro el sistema lo cual restringe los procesos dentro el sistema.
- Información como ser las estadísticas y últimas ventas.
- Descarga de listas y reportes en formato PDF.
- Diseño adaptable, puede ser visualizado a la mayoría de los navegadores web, tanto en ordenadores como en tabletas y celulares.

### 3. INGRESO AL SISTEMA

- Antes de ingresar al sistema, tiene seguir los siguientes:
- Contar con navegador web (Internet Explorer 9.0+, Chrome 40+, Firefox 35.0+), no se garantiza un correcto funcionamiento en otros navegadores.
- Disponer de un usuario y contraseña, autorizado por el administrador.

Ingresar al navegador y nos muestra el siguiente formulario de ingreso:



The image shows a login interface for a system titled 'SISTEMA DE INVENTARIO'. The background is a teal color with the title in white. In the center, there is a white box containing the login form. The form has a header with a user icon and the text 'Inicie Session'. Below this, there are two input fields: 'CORREO' with the value 'admin@gmail.com' and 'PASSWORD' with a masked password '\*\*\*\*\*'. At the bottom of the form is a teal button with a white arrow icon and the text 'INGRESAR'.

Ingresar con datos autorizados y pulse el botón.

INGRESAR

#### 4. SECCIONES DEL SISTEMA



#### Menú principal

En esta sección está disponible el menú desplegable con los módulos que se puede observar respectivamente, acompañado con sus respectivos Submodulos que se estará detallando más adelante.

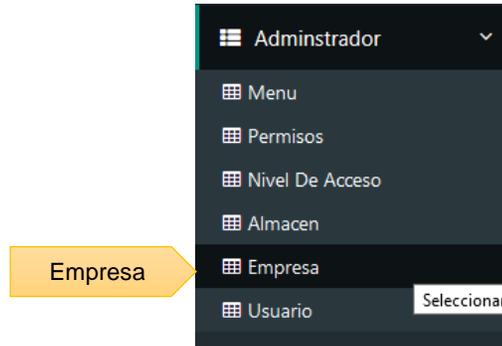
#### Vista principal

En el menú inicio podemos observar el movimiento interno de la empresa más las estadísticas.

En esta sección es el contenedor donde se realizan todas las operaciones como ingreso y consulta de datos.

#### 5. CONFIGURACION DE LA EMPRESA

Vamos a dirigirnos al menú desplegable del Módulo administrador y dirigirse al Submodulo Empresa como nos muestra en la imagen.



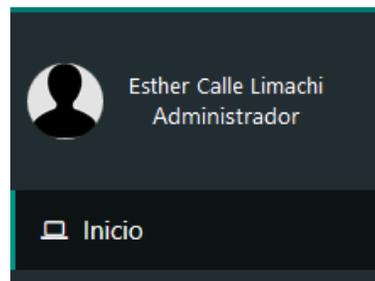
Una vez haciendo click nos muestra la pantalla no dirigimos al botón Nueva Empresa. **Nueva Empresa**

Nos muestra el formulario para registrar a la Empresa.

A form titled 'Registrar Nueva Empresa' is displayed. It has several input fields: 'Nombre Empresa', 'Telefono', 'Red social', 'Correo', 'Direccion', and 'Logo'. Below the 'Logo' field is a 'Subir logo' section with an 'Examinar...' button and a message 'No se ha seleccion...o ningún archivo.'. At the bottom right of the form is a green button labeled 'Actualizar Empresa'.

## 6. MENU INICIO

Es el menú que se muestra al inicio de cada sesión, el las estadísticas y el historial de la ultimas ventas.



Se actualiza con cada registro



## 7. PRODUCTOS

En esta sección podemos detallar todas las operaciones que se puede realizar.

Lista de Producto Nuevo Producto

Entrada 10 Paginas Buscador:

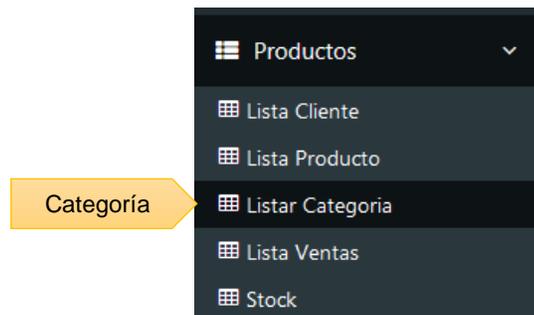
No. Póliza	Marca	Modelo	Chasis	Motor	Categoría	descripcion	costo de venta	logo	estadi
1	dd	dd			Minibus	dd	12458.00		Activo
	Kinglong	2016	9BWCC5X918003018	JDH-09692	Minibus	motor cilindrada 2300cc	17500.00		Activo
178912CVB123	Jincheng	2019	99FGB561674DC	KLM-0521	Minibus	motor cilindrada 2900cc corona montaña	58600.00		Activo

Página 1 a 3 de 3 entradas Atras 1 Siguiente

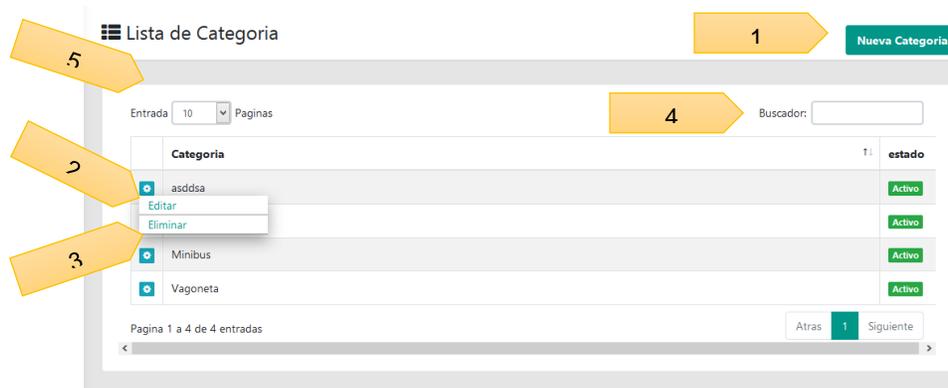
1. Adicionar un nuevo producto.
2. Opción de editar en caso de la incorrecta digitación del dato.
3. Eliminar en caso especial si es requerido.
4. Aquí puede buscar un producto de la lista.
5. configurable para una vista si se registró varios productos.

## 8. CATEGORIA DEL PRODUCTO

En esta sección se encarga de administrar las diferentes categorías de los vehículos. Nos dirigimos al Submodulo Categoría.



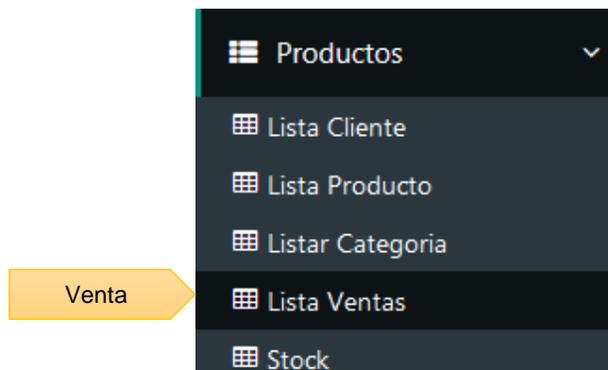
Nos muestra la pantalla de la categoría.



1. Adiciona nueva categoría
2. Editar la categoría en caso que fuese requerido.
3. Eliminar si es que fuese necesario.
4. Buscar por el nombre de categoría de la lista.
5. Configurar el número de listado si es que existiese más datos registrados.

## 8. VENTAS

En esta sección de ventas detallaremos como procede la venta de un vehículo.



Vamos hacer click en nueva venta



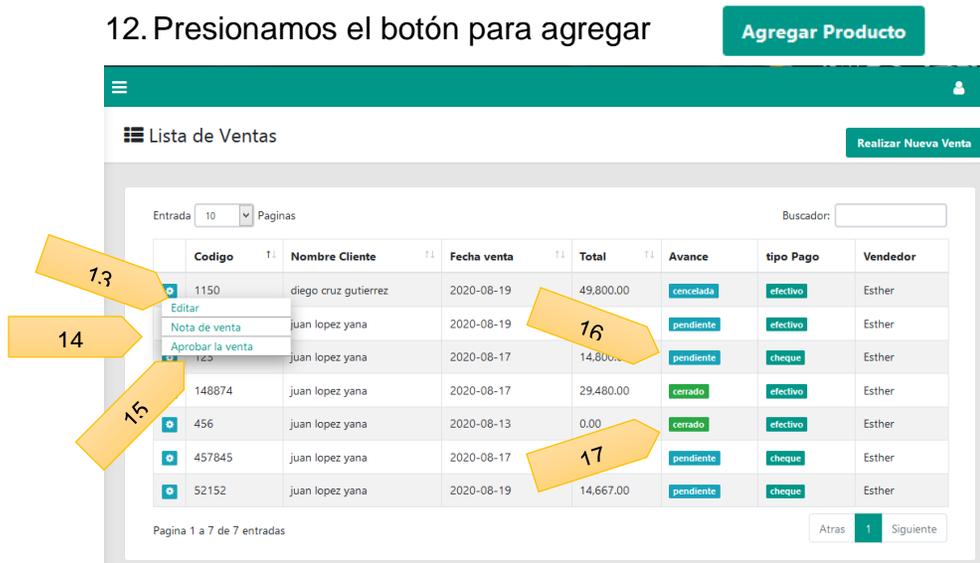
1. Realizamos la nueva venta de la siguiente forma:



1. Para la venta necesitamos agregar a un nuevo cliente y seleccionamos con el menú select.
  2. Ingresamos la fecha de la venta.
  3. Ingresamos el código de la venta.
  4. Seleccionamos el estado de 1) Pendiente y 2) Cancelada.
  5. Seleccionamos el tipo de pago 1) Efectivo 2) Cheque.
  6. Presionamos el botón de Agregar Producto.
- Nos muestra la interfaz del agregado del  vehículo.



7. Guardamos todos los datos registrados.
8. Pero antes como presionamos el botón para agregar nuevo producto a la venta, vamos a buscar el vehículo por el nombre.
9. Seleccionamos la cantidad pero por lo visto la cantidad es uno
10. En esta sección nos muestra el stock por defecto del producto existente en almacenes.
11. En el precio nos genera automáticamente.
12. Presionamos el botón para agregar



13. En la opción editar podemos editar la venta.
14. Generamos la Nota de Venta.
15. Aprobar la nota venta para cerrar la venta.

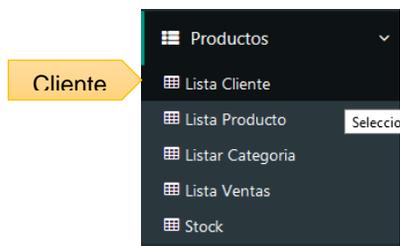


16. Vemos los diferentes tipos de avance como el pendiente, que aún no se cerró la venta.

17. Como también vemos tipo de avance a cerrado sin deudas listo para generar la Nota de Venta y ya no se puede volver a editar porque confirmo cerrar la venta.

### 9. CLIENTES

Nos muestra el listado de los clientes quienes realizaron la compra de un vehículo.



Vamos a ir a nuevo cliente.

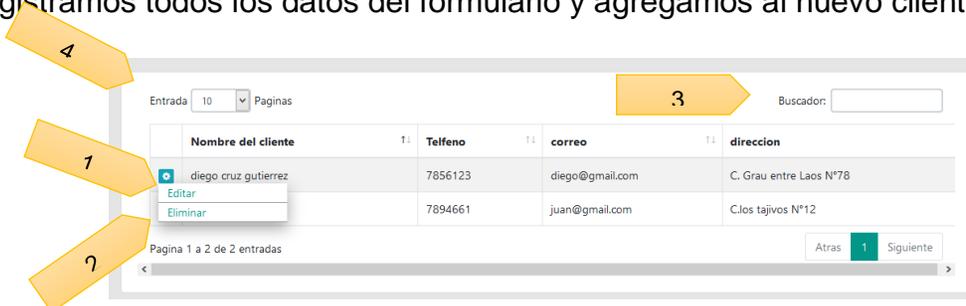


**Registrar Nuevo Cliente** ×

Nombre completo cliente	Correo
<input type="text"/>	<input type="text"/>
Telefono	Direccion
<input type="text"/>	<input type="text"/>

[Agregar Cliente](#)

Registramos todos los datos del formulario y agregamos al nuevo cliente.



1. Editamos a los clientes.
2. Eliminamos si es requerido.
3. Buscamos por el nombre del cliente.

4. Nos facilita la vista del listado de los clientes.

### 10. PERSONAS

Ingresamos a la sección del Submodulo personas.



Luego nos vamos al botón nueva persona



Nos muestra la interfaz para llenar los datos de la nueva persona y luego agregamos.

A form titled 'Registrar Nueva Persona' with fields for 'Nombres', 'Paterno', 'Materno', 'Cédula de Identidad', and 'Expedido'. A green 'Agregar Persona' button is at the bottom right.

Una vez llenado el formulario nos muestra el listado con el menú siguiente:

A table with 5 columns: Nombre Persona, Apellido Paterno, Apellido Materno, Cédula de Identidad, and Expedido. It shows 4 rows of data. Annotations 1, 2, and 3 point to the 'Editar' button, the 'Eliminar' button, and the search bar respectively.

Nombre Persona	Apellido Paterno	Apellido Materno	Cédula de Identidad	Expedido
aaa	sss	dddd	7894512	SC
Esther	Calle	Limachi	66644678	LP
Iola	figueroa	calsina	7894652	CBBA

1. Opción editar a la Persona.
2. Eliminar si fuese necesario.
3. Buscar por el nombre.

### 11. GESTION DE USUARIOS

Vamos al menú desplegable nos vamos gestionar usuarios luego nos vamos a registrar al nuevo usuario haciendo click en siguiente botón.

**Nueva Usuario**

Ingresamos  
datos al  
formulario

**Registrar Nuevo Usuario**

Nombre usuario

Correo

Persona Selecciona Persona...

Rol Selecciona Rol...

Contraseña

**Agregar Usuario**

**Lista de Usuario**

**4**

Buscador:

	Usuario	Persona	Correo	Categoria	Estado
1	Esther	Esther	admin@gmail.com	Administrador	Activo
2	<div style="border: 1px solid #ccc; padding: 2px;"> <p>Editar</p> <p>Eliminar</p> <p>Cambiar Contraseña</p> </div>	valll	jaimito@gmail.com	Cliente	Activo
3	pruebass	cliente	pruebas@gmail.com	Administrador	Activo
	will	cliente	will@gmail.com	Reportes	Activo

Pagina 1 a 5 de 5 entradas

1. Opción del editado del Usuario.
2. Eliminar al usuario.
3. Cambiar la contraseña si en caso se olvidase para ingresar al sistema.
4. Buscar de manera ordenada con el nombre de usuario.

## 12. PERMISO

Nos desplegamos al submenú permiso

Permiso

- Adminstrador
- Menu
- Permisos
- Nivel De Acceso
- Almacen
- Empresa
- Usuario

Y nos muestra la pantalla de los permisos a asignar al usuario correspondiente.

Quitamos menú deseleccionando las cajitas !

Asignamos un nuevo permiso al usuario correspondiente !

Seleccionamos las cajitas haciendo clic !

Grupo	Menu	Crear	Actualizar	Eliminar	Estado
O1	Grupo Menu	persona	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
O2	Grupo Menu	grupo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
O3	Adminstrador	menu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
O4	Adminstrador	permiso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
O5	Adminstrador	rol	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
O6	Adminstrador	almacen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

13. NIVEL DE ACCESO

! Ingresamos al nivel de accesos

Registramos lo datos en siguiente formulario y agregamos el rol para la autorización según usuario.

Registrar Nuevo Rol / Categoría

Nombre rol:

Descripcion:

Agregar Rol

Lista de Rol / Categoría

Entrada 10 Paginas

Buscador:

Nombre Categoría	Descripcion	Estado
Administrador	a	Activo
Reportes	solo reporte	Activo

Página 1 a 3 de 3 entradas

Atras 1 Siguiete

1

2

3

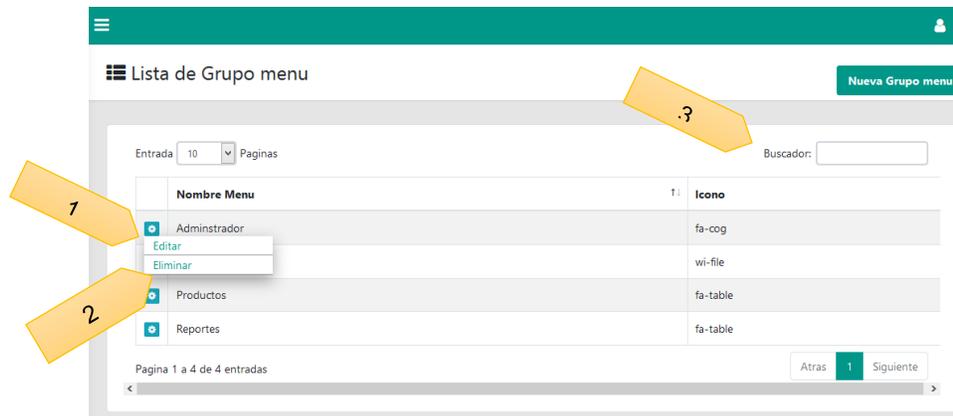
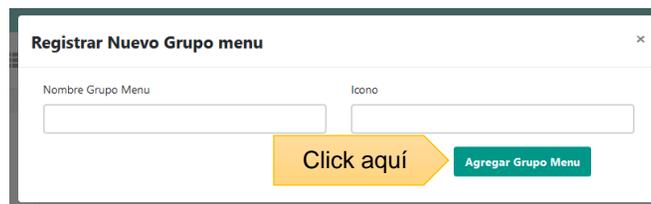
1. Editamos el rol/categoría para dar los niveles de acceso.

2. Eliminamos en caso necesarios.
3. Buscar de forma ordenada por el nombre del rol/categoría.

### 14. GRUPO MENU

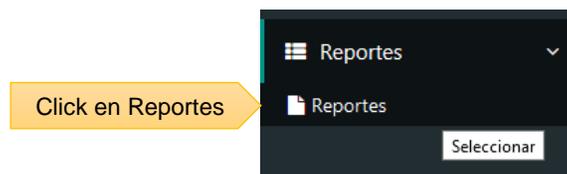


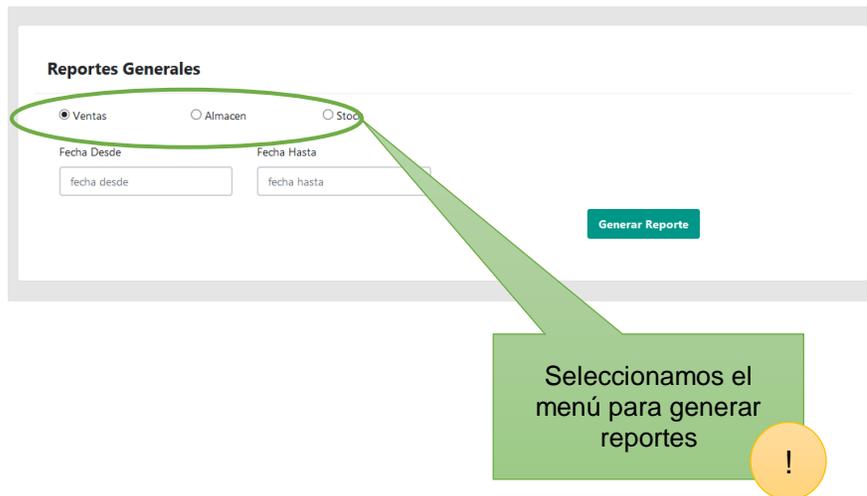
Una vez ingresado nos vamos al botón nuevo, luego nos direcciona al formulario nuevo para registrar con los datos dándole click en el botón agregar Grupo Menú.



1. Editamos el menú.
2. Eliminamos el menú.
3. Buscamos el menú de manera ordenada.

### 15. REPORTEES





## 16. RECOMENDACIONES

- Revise los datos introducidos detalladamente, antes de consolidar cualquier operación, especialmente tener cuidado para la Nota de Venta.
- No divulgue sus credenciales de acceso al sistema y se recomienda cambiar su contraseña si fuese necesario de manera periódica para evitar el ingreso no autorizado.

# **ANEXO E**

# **MANUAL TÉCNICO**



# MANUAL TÉCNICO

SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO DE  
VENTAS

EL ALTO - BOLIVIA

## ÍNDICE CONTENIDO

	PÁG.
INTRODUCCION.....	1
1. REQUERIMIENTOS TÉCNICOS.....	2
REQUERIMIENTOS MÍNIMOS DE SOFTWARE .....	2
2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO.....	2
2.1 PHP .....	2
2.2 MYSQL.....	2
2.3 APACHE.....	3
3. INSTALACIÓN DE WEB SERVER .....	3
4. INSTALACIÓN COMPOSER .....	9
5. CREANDO PROYECTO EN LARAVEL.....	12
5.1. Configurar la base de datos.....	15
5.2. Servidor activo.....	15
6. CASOS DE USO .....	16
7. MODULO DE ADMINISTRACION .....	19

## LISTA DE TABLAS

	PÁG.
<b>Tabla N° 1. 1.</b> Creación de registro de usuario .....	17
<b>Tabla N° 1. 2.</b> Eliminar registro de Usuario .....	17
<b>Tabla N° 1. 3.</b> Modificar Usuario .....	18
<b>Tabla N° 1. 4.</b> Guardar datos en la BD .....	19

## INDICE DE FIGURAS

	<b>PÁG.</b>
<b>Figura N° 1. 1.</b> Xampp instalación .....	4
<b>Figura N° 1. 2.</b> Instalación 1 .....	4
<b>Figura N° 1. 3.</b> Direccionamiento para la Instalación .....	5
<b>Figura N° 1. 4.</b> Iniciar Instalación.....	6
<b>Figura N° 1. 5.</b> Configurar Firewall .....	6
<b>Figura N° 1. 6.</b> Finalizar instalación.....	7
<b>Figura N° 1. 7.</b> Interfaz del servidor configurable.....	8
<b>Figura N° 1. 8.</b> Iniciando el puerto Apache y MySql.....	8
<b>Figura N° 1. 9.</b> Preparando composer para instalar.....	9
<b>Figura N° 1. 10.</b> Pre configuración de composer .....	10
<b>Figura N° 1. 11.</b> Direccionamiento de Composer.....	10
<b>Figura N° 1. 12.</b> Instalando composer .....	11
<b>Figura N° 1. 13.</b> Versión del composer.....	12
<b>Figura N° 1. 14.</b> Instalación del proyecto.....	13
<b>Figura N° 1. 15.</b> Fase de instalación.....	13
<b>Figura N° 1. 16.</b> Instalación finalizada con éxito .....	14
<b>Figura N° 1. 17.</b> Navegación del proyecto AdminInventari.....	14
<b>Figura N° 1. 18.</b> Proyecto creado en la unidad c. ....	15
<b>Figura N° 1. 19.</b> Diagrama CasoUso-Administrador .....	16

## **OBJETIVOS**

Brindar la información necesaria para poder realizar la instalación y configuración del aplicativo.

### **Específicos**

- ✓ Representar la funcionalidad técnica de la estructura, diseño y definición del aplicativo.
- ✓ Definir claramente el procedimiento de instalación del aplicativo.
- ✓ Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
- ✓ Describir las herramientas utilizadas para el diseño y desarrollo del prototipo.

## **INTRODUCCION**

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación del aplicativo creado para la administración de casos de la empresa Victory Motors Bolivia.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación del aplicativo.

## 1. REQUERIMIENTOS TÉCNICOS.

### REQUERIMIENTOS MÍNIMOS DE HARDWARE

- ✓ **Procesador** : Core
- ✓ **Memoria RAM: Mínimo** : 1 Gigabytes (GB)
- ✓ **Disco Duro** : 500Gb.

### REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- ✓ Privilegios de administrador
- ✓ **Sistema Operativo** : Windows 8, Windows 10 en adelante.

## 2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

### 2.1 PHP

Es un Lenguaje de Programación para trabajar páginas WEB ofreciendo la ventaja de combinar con HTML. Las ejecuciones son realizadas en el Servidor y el cliente es el encargado de recibir los resultados de la ejecución. Si el cliente realiza una petición, se ejecuta el intérprete de PHP y se genera el contenido de manera dinámica. Permite conexión con varios tipos de Bases de Datos como: MySql, SQL Server, etc. permitiendo aplicaciones robustas sobre la WEB. Este lenguaje de programación puede ser ejecutado en la gran mayoría de sistemas operacionales y puede interactuar con Servidores WEB populares

### 2.2 MYSQL

Es un manejador de Bases de Datos, el cual permite múltiples hilos y múltiples usuarios, fue desarrollado como software libre.

Aunque se puede usar sobre varias plataformas es muy utilizado sobre LINUX. Es libre para uso en Servidores WEB.

Ofrece ventajas tales como fácil adaptación a diferentes entornos de desarrollo,

Interacción con Lenguajes de Programación como PHP, Java Script y fácil Integración con distintos sistemas operativos.

### **2.3 APACHE**

Es un Servidor WEB desarrollado por el grupo Apache. Su código fuente se puede distribuir y utilizar de forma libre. Está disponible para diferentes plataformas de Sistemas Operativos entre otros Windows, Linux, Mac y NetWare.

Ofrece ventajas tales como independencia de plataforma, haciendo posible el cambio de plataforma en cualquier momento; creación de contenidos dinámicos, permitiendo crear sitios mediante lenguajes PHP.

Además de ser libre su soporte técnico es accesible ya que existe una comunidad que está disponible en foros, canales IRC y servidores de noticias, donde hay gran cantidad de usuarios disponibles para cuando surge algún problema.

## **3. INSTALACIÓN DE WEB SERVER**

Descargar al PC el fichero de instalación xampp-win32-7.2.10-0-VC15-installer - Apache/2.4.34 (Win32) 32b.exe el cuál se encuentra en la página: <https://xampp.uptodown.com/windows> una vez descargado, dar clic sobre él para iniciar la instalación. La instalación se debe hacer con una cuenta de Administrador o con derechos de administrador. Durante la instalación aparecerán las siguientes pantallas:

En la primera ventana aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en "Next".

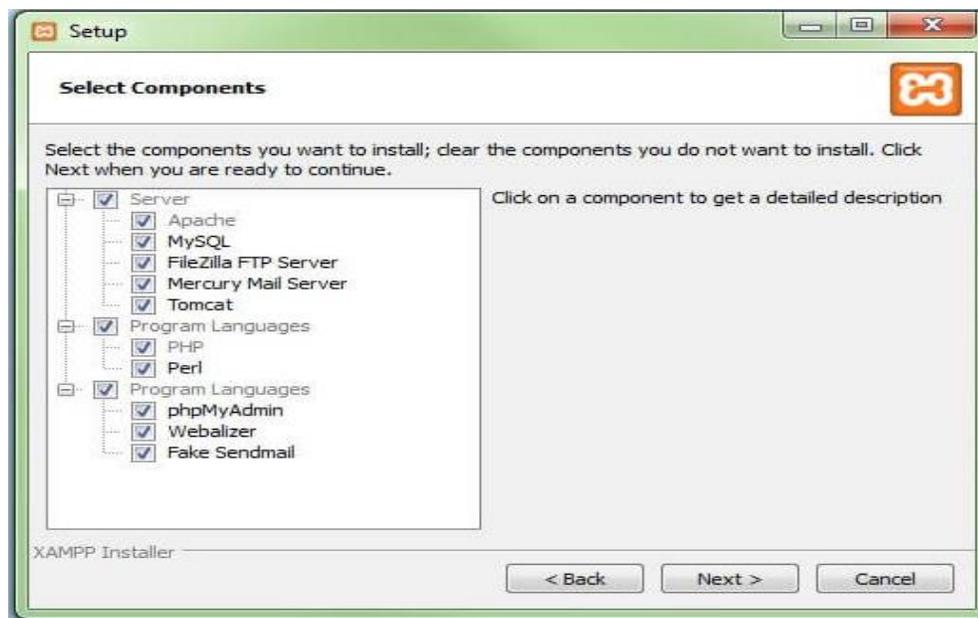
Figura N° 1. 1. Xampp instalación



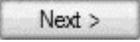
Fuente: (Elaboración Propia).

La siguiente ventana presionar el botón  Instalación y administración para continuar.

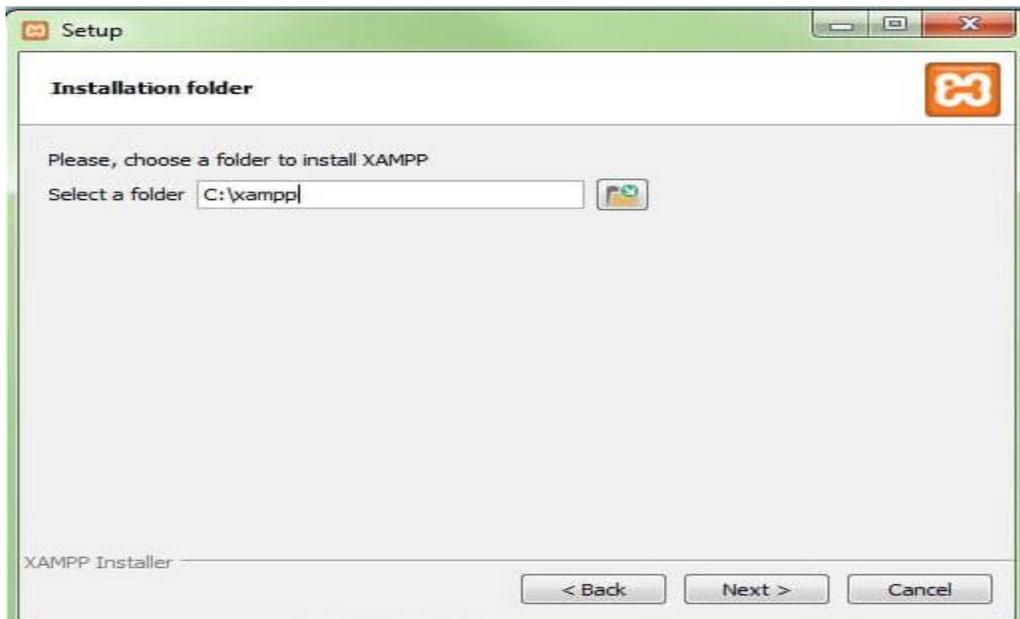
Figura N° 1. 2. Instalación 1



Fuente: (Elaboración Propia).

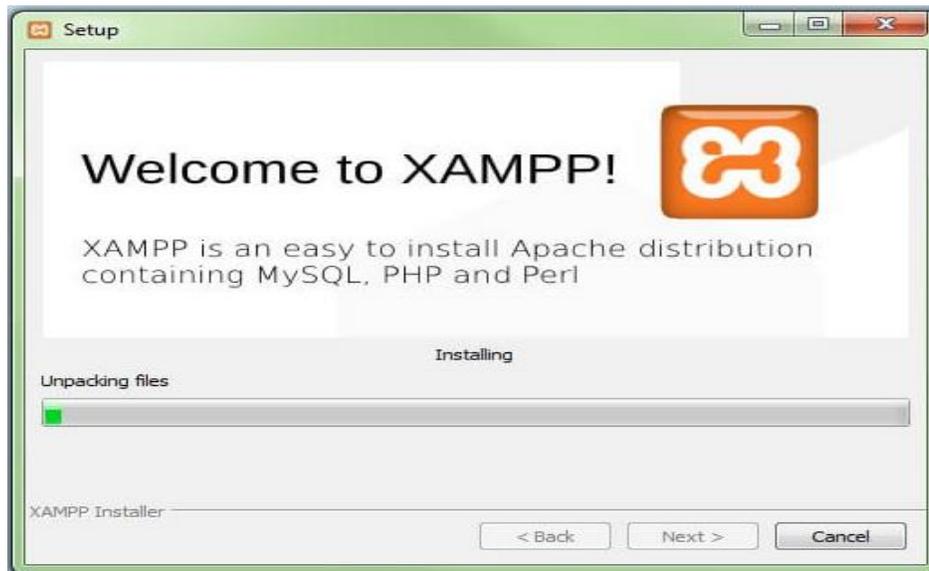
En este paso se escoge el directorio donde se instalará el paquete. Si se ha escogido la configuración estándar se creará una carpeta con el nombre XAMPP en C:\.  Para continuar.

**Figura N° 1. 3.** Direccionamiento para la Instalación



**Fuente:** (Elaboración Propia).

El asistente extrae los componentes seleccionados y los guarda en el directorio escogido en un proceso que puede durar algunos minutos. El avance de la instalación se muestra como una barra de carga de color verde.

**Figura N° 1. 4. Iniciar Instalación.**

**Fuente:** (Elaboración Propia).

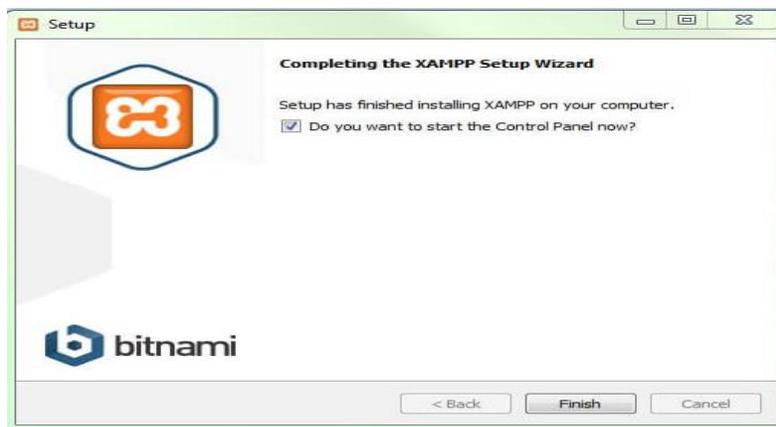
Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública.

**Figura N° 1. 5. Configurar Firewall**

**Fuente:** (Elaboración Propia).

Una vez extraídos e instalados todos los componentes puedes cerrar el asistente con la tecla “Finish”. Para acceder inmediatamente al panel de control solo es necesario marcar la casilla que pregunta si deseamos hacerlo.

**Figura N° 1. 6.** Finalizar instalación

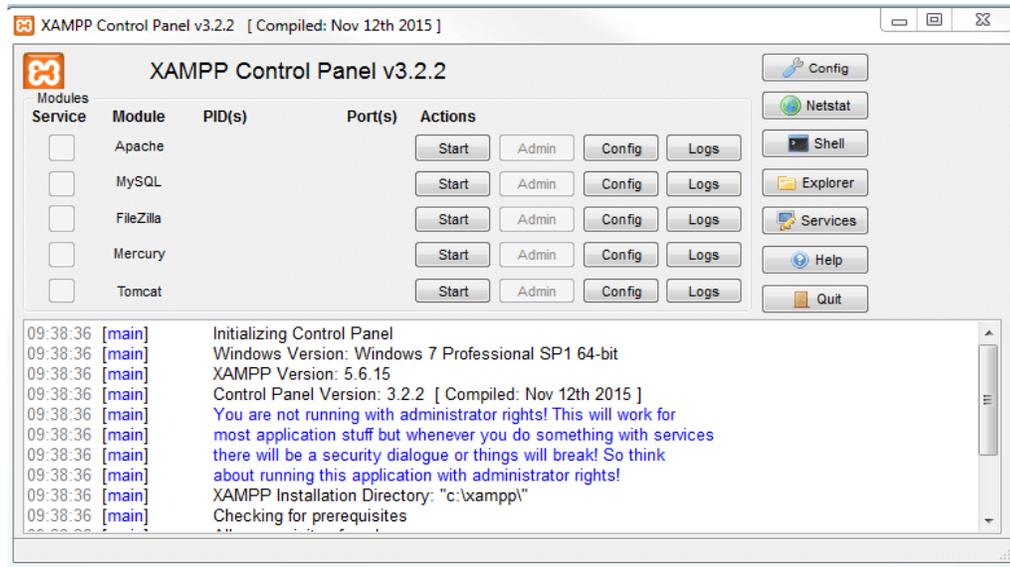


**Fuente:** (Elaboración Propia).

En la interfaz de usuario del panel de control se protocolan todas las acciones y es posible activar o desactivar los módulos por separado con un simple clic. Además, se dispone de diversas utilidades como:

- ✓ Config: para configurar XAMPP así como otros componentes aislados.
- ✓ Netstat: muestra todos los procesos en funcionamiento en el ordenador local
- ✓ Shell: lanza una ventana de comandos UNIX
- ✓ Explorer: abre la carpeta XAMPP en el explorador de Windows
- ✓ Services: muestra todos los servicios en funcionamiento
- ✓ Help: incluye enlaces a foros de usuarios
- ✓ Quit: se usar para salir del panel de control.

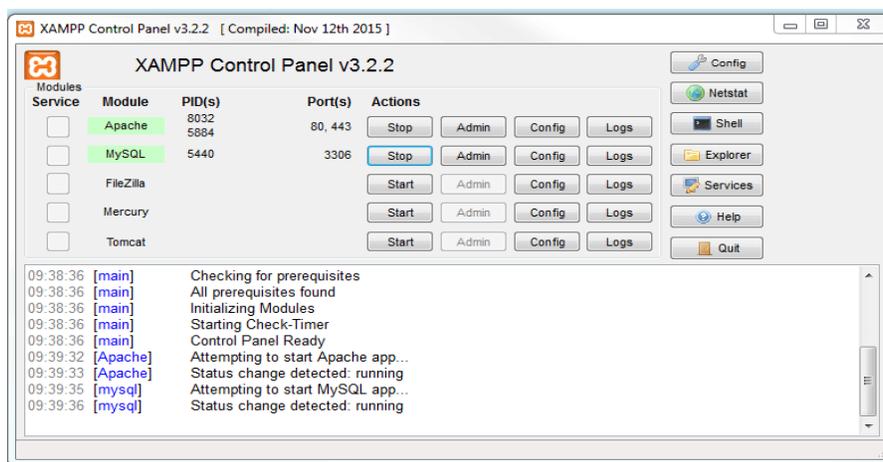
Figura N° 1. 7. Interfaz del servidor configurable



Fuente: (Elaboración Propia).

En la parte superior se pueden iniciar o interrumpir los módulos de XAMPP por separado mediante los comandos “Start” y “Stop” bajo “Actions”. Los módulos que se activaron aparecen marcados en verde.

Figura N° 1. 8. Iniciando el puerto Apache y MySql



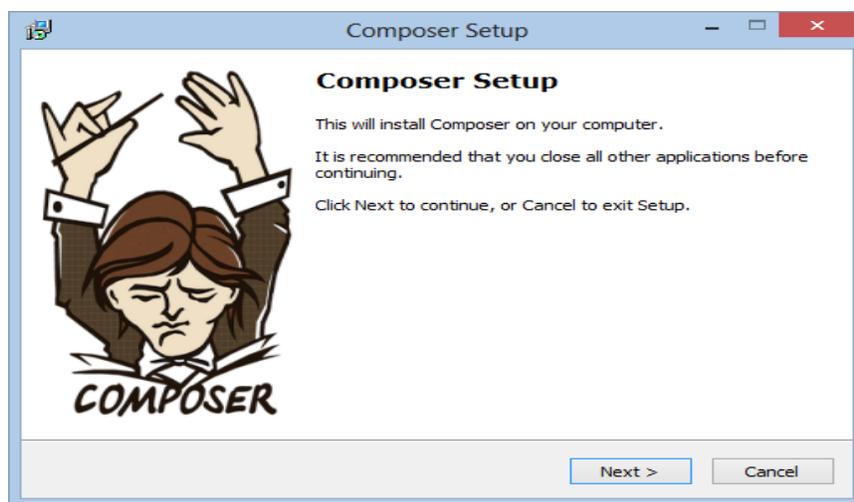
Fuente: (Elaboración Propia).

#### 4. INSTALACIÓN COMPOSER

Composer es una herramienta para gestionar las dependencias en PHP. Te permite declarar las librerías de las cuales tu proyecto depende o necesita y las instala en el proyecto. Para instalar Composer en Windows debemos descargarlo de su página oficial y en la sección Windows Installer, haz click en Composer-Setup.exe.

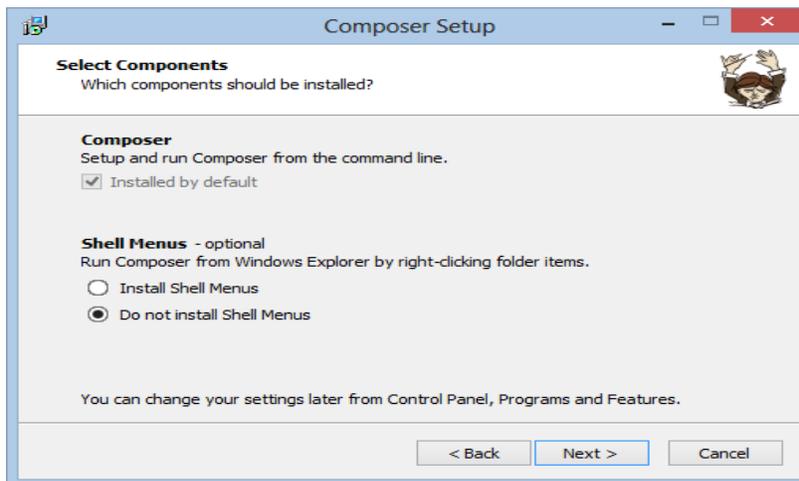
Una vez que la descarga finalice, ejecuta el instalador y haz click en Next.

**Figura N° 1. 9.** Preparando composer para instalar



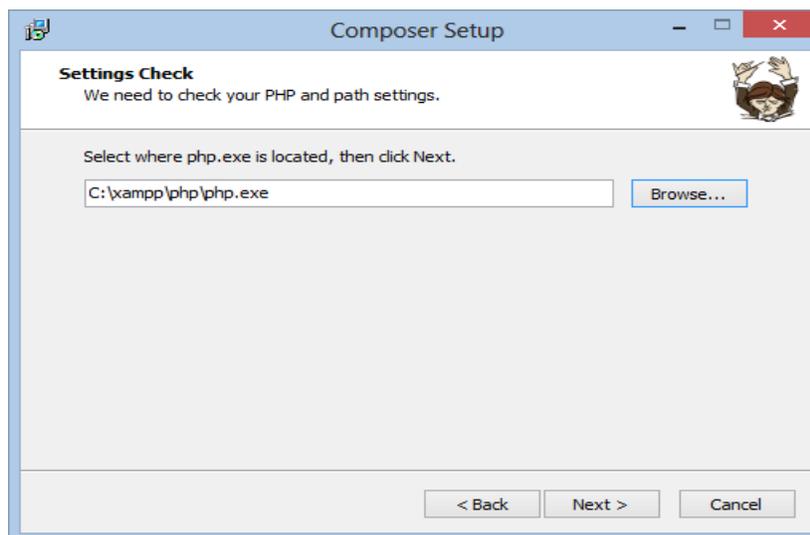
**Fuente:** (Elaboración Propia).

Si quieres administrar tus proyectos mediante el Explorador de Windows puedes seleccionar la opción «Install Shell Menus» aunque lo recomendable es la usar la línea de comandos.

**Figura N° 1. 10.** Pre configuración de composer

**Fuente:** (Elaboración Propia).

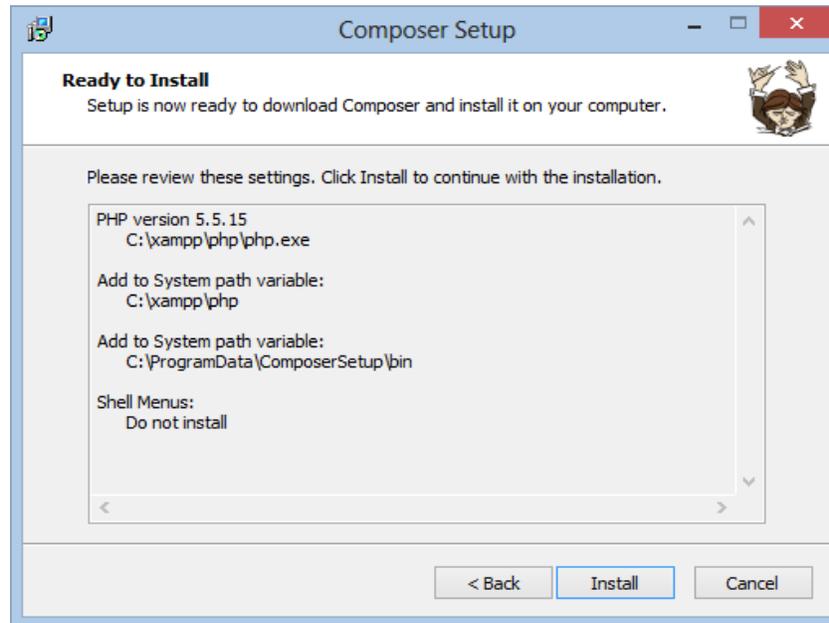
A continuación nos pide que indiquemos la ruta del ejecutable de PHP, en mi caso como estoy trabajando con XAMPP el ejecutable de PHP se encuentra en la ruta C:\xampp\php\ y seleccionas php.exe, luego click en Next.

**Figura N° 1. 11.** Direccionamiento de Composer

**Fuente:** (Elaboración Propia).

En este punto el instalador de Composer nos muestra la configuración de la instalación, simplemente le damos click a Install.

Figura N° 1. 12. Instalando composer



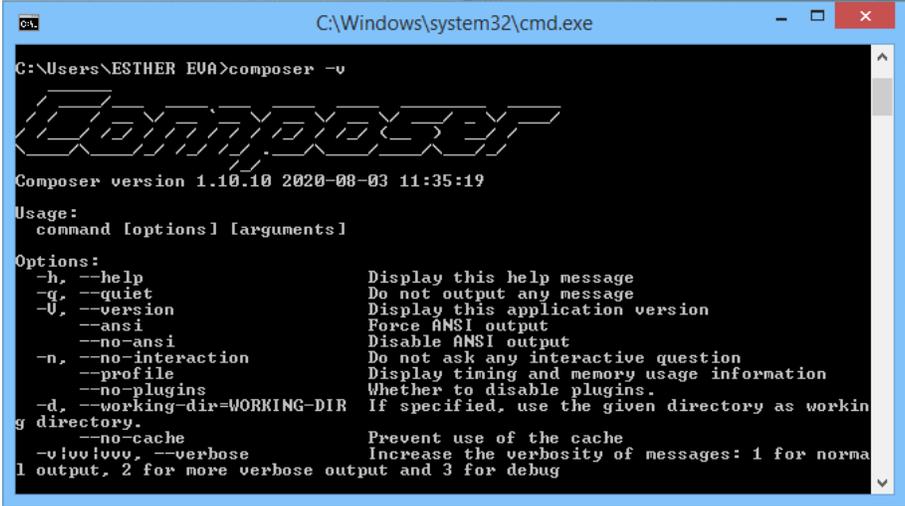
**Fuente:** (Elaboración Propia).

Una vez instalado, aparecerán otras donde simplemente debes hacer click en Next, y posteriormente en Finalizar; después el instalador de Composer habrá puesto en nuestro PATH global la ruta de la carpeta PHP y su propia carpeta Composer. Esto nos permite trabajar desde consola escribiendo sólo php o composer sin necesidad de indicar la ruta del ejecutable. Para ver que todo está en orden vamos a realizar dos pequeñas pruebas, así que es momento de abrir la consola, y teclear:

- ✓ php -v (tecla Enter)
- ✓ composer -version (tecla Enter)

Esto debería devolver la versión de cada uno, como se ve en la siguiente imagen:

Figura N° 1. 13. Versión del composer



```
C:\Windows\system32\cmd.exe
C:\Users\ESTHER EUA>composer -v

Composer version 1.10.10 2020-08-03 11:35:19

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -U, --version         Display this application version
  --ansi               Force ANSI output
  --no-ansi            Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  --profile            Display timing and memory usage information
  --no-plugins         Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache           Prevent use of the cache
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
```

Fuente: (Elaboración Propia).

Con esto ya tenemos Composer instalado y funcionando en Windows, ahora solo nos queda instalar Laravel, veamos cómo hacer esto posible.

## 5. CREANDO PROYECTO EN LARAVEL

Existen dos formas de crear un proyecto con Laravel, la primera es descargando el archivo master desde su repositorio oficial de GitHub y la otra es usando Composer desde la consola que es precisamente lo que haremos. Desde la consola, dirígete al directorio donde guardas tus proyectos web (si usas XAMPP la ruta es C:\xampp\htdocs) y teclea lo siguiente:

✓ `cd C:\xampp\htdocs`

Ahora crearemos el proyecto laravel escribiendo las siguientes palabras mágicas:

✓ `composer create-project laravel/laravel nombre_del_proyecto --prefer-dist`

En mi caso en un arranque de creatividad llamaré a mi proyecto “AdminInventari”

Figura N° 1. 14. Instalación del proyecto

```

C:\Windows\system32\cmd.exe - composer create-project laravel/laravel Adm...
y to composer.json, and updates the composer.lock file.
upgrade Upgrades your dependencies to the latest version according
y to composer.json, and updates the composer.lock file.
validate Validates a composer.json and composer.lock.
why Shows which packages cause the given package to be installed.
why-not Shows which packages prevent the given package from being installed.
C:\Users\ESTHER EVA>cd ..
C:\Users>cd ..
C:\>cd xampp
C:\xampp>cd htdocs
C:\xampp\htdocs>composer create-project laravel/laravel AdminInventari --prefer-dist
Creating a "laravel/laravel" project at "./AdminInventari"
Installing laravel/laravel (v7.25.0)
- Installing laravel/laravel (v7.25.0): Downloading (100%)
Created project in C:\xampp\htdocs\AdminInventari
> @php -r "file_exists('.env') || copy('.env.example', '.env');"

```

Fuente: (Elaboración Propia).

Composer empezará a descargar las librerías necesarias para nuestro proyecto, esto requiere un poco de tiempo.

Figura N° 1. 15. Fase de instalación

```

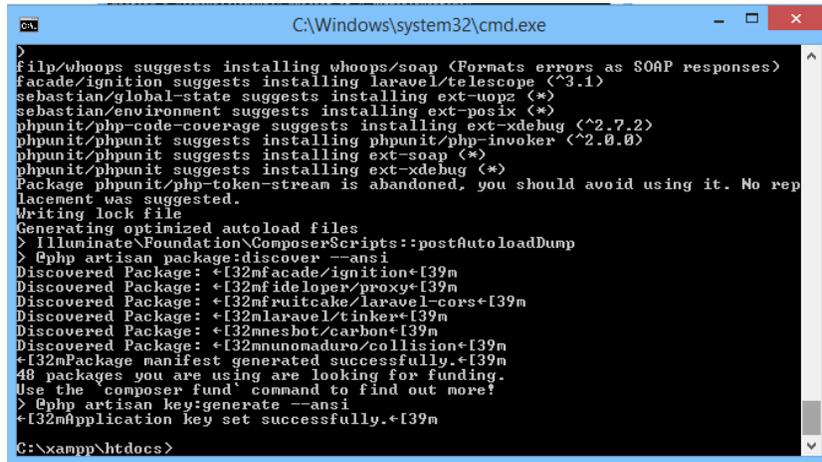
C:\Windows\system32\cmd.exe
- Installing league/commonmark (1.5.4): Downloading (100%)
- Installing dragonmantank/cron-expression (v2.3.0): Downloading (connecting..
Downloading (100%)
- Installing doctrine/inflector (2.0.3): Downloading (100%)
- Installing laravel/framework (v7.25.0): Downloading (100%)
- Installing fideloper/proxy (4.4.0): Downloading (100%)
- Installing asm89/stack-cors (v2.0.1): Downloading (100%)
- Installing fruitcake/laravel-cors (v2.0.1): Downloading (100%)
- Installing ralouphie/getallheaders (3.0.3): Downloading (100%)
- Installing psr/http-message (1.0.1): Downloading (100%)
- Installing guzzlehttp/psr7 (1.6.1): Downloading (100%)
- Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
- Installing guzzlehttp/guzzle (6.5.5): Downloading (100%)
- Installing dnoegel/php-xdg-base-dir (v0.1.1): Downloading (100%)
- Installing nikitic/php-parser (v4.9.0): Downloading (100%)
- Installing psy/psysh (v0.10.4): Downloading (100%)
- Installing laravel/tinker (v2.4.2): Downloading (100%)
- Installing scrivo/highlight.php (v9.18.1.1): Downloading (100%)
- Installing filp/whoops (2.7.3): Downloading (100%)
- Installing facade/ignition-contracts (1.0.1): Downloading (100%)
- Installing facade/flare-client-php (1.3.4): Downloading (100%)
- Installing facade/ignition (2.3.6): Downloading (100%)
- Installing fzaninotto/faker (v1.9.1): Downloading (100%)
- Installing hamcrest/hamcrest-php (v2.0.1): Downloading (100%)
- Installing mockery/mockery (1.3.3): Downloading (100%)
- Installing nunomaduro/collision (v4.2.0): Downloading (100%)
- Installing sebastian/version (2.0.1): Downloading (100%)
- Installing sebastian/type (1.1.3): Downloading (100%)

```

Fuente: (Elaboración Propia).

Si no ocurrió algún problema de conexión a Internet veremos que nuestro proyecto “prueba” se creó correctamente.

Figura N° 1. 16. Instalación finalizada con éxito

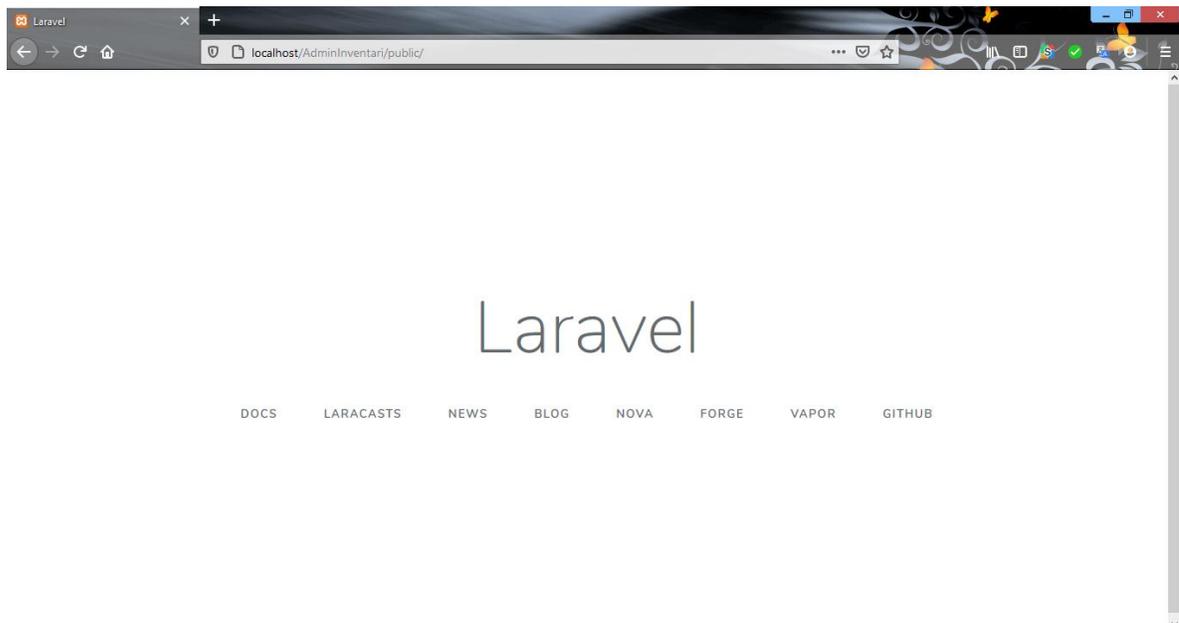


```
C:\Windows\system32\cmd.exe
>
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
facade/ignition suggests installing laravel/telescope (^3.1)
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.7.2)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0.0)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
Package phpunit/php-token-stream is abandoned, you should avoid using it. No replacement was suggested.
Writing lock file
Generating optimized autoload files
> illuminate\Foundation\ComposerScripts::postAutoloadDump
> [32mphp artisan package:discover --ansi
Discovered Package: +[32mfacade/ignition+[39m
Discovered Package: +[32mfideloper/proxy+[39m
Discovered Package: +[32mfruitcake/laravel-cors+[39m
Discovered Package: +[32mlaravel/tinker+[39m
Discovered Package: +[32mnesbot/carbon+[39m
Discovered Package: +[32mnunomaduro/collision+[39m
+ [32mPackage manifest generated successfully.+ [39m
48 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> [32mphp artisan key:generate --ansi
+ [32mApplication key set successfully.+ [39m
C:\xampp\htdocs>
```

Fuente: (Elaboración Propia).

Finalmente para verificar que la creación de nuestro proyecto “AdminInventari” se realizó de manera correcta, accede a [http://localhost/nombre\\_del\\_proyecto/public](http://localhost/nombre_del_proyecto/public) en el navegador de tu preferencia, donde debes ver lo siguiente:

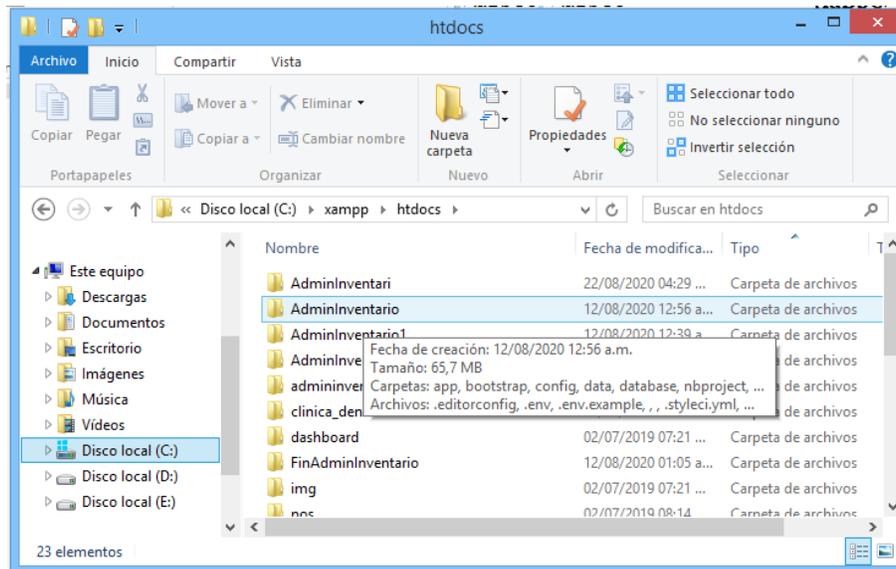
Figura N° 1. 17. Navegación del proyecto AdminInventari



Fuente: (Elaboración Propia).

Luego vamos a la unidad: C:\xampp\htdocs\AdminInventario para observar la carpeta que creo la herramienta composer.

**Figura N° 1. 18.** Proyecto creado en la unidad c.



**Fuente:** (Elaboración Propia).

### 5.1. Configurar la base de datos

```
C:\xampp\htdocs\AdminInventario\,env
```

```
DB_CONNECTION=mysql
```

```
DB_HOST=localhost
```

```
DB_PORT=3306
```

```
DB_DATABASE=inventario
```

```
DB_USERNAME=victorymotors
```

```
DB_PASSWORD=victory123
```

### 5.2. Servidor activo

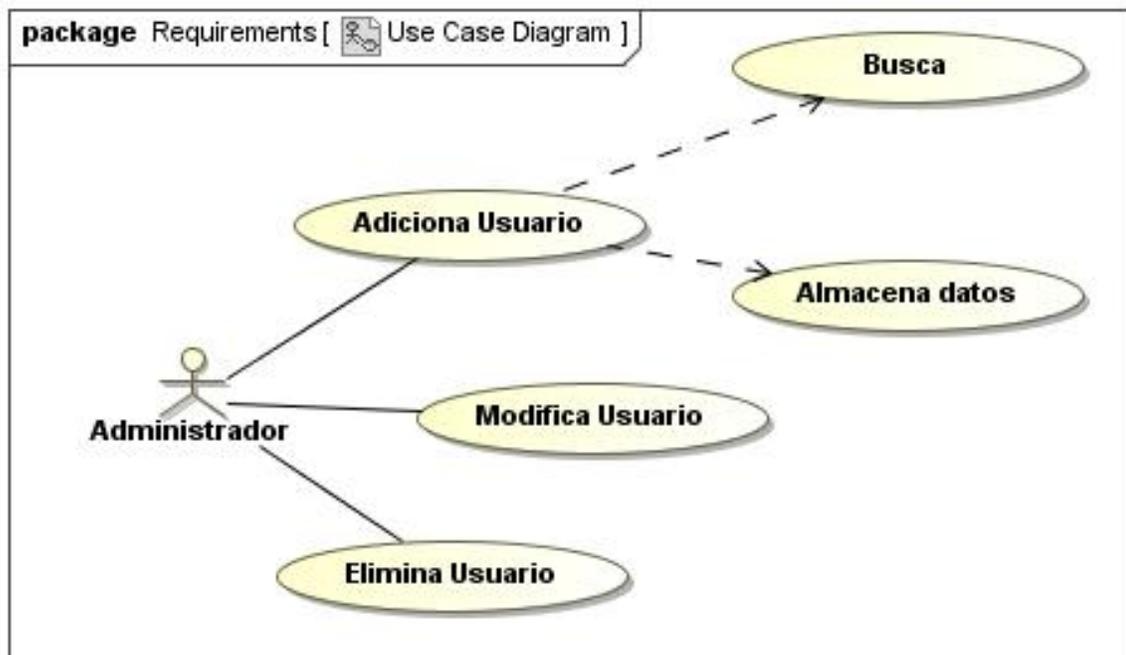
✓ Puerto: 80

## 6. CASOS DE USO

Para el aplicativo se va a definir dos roles o tipos de usuarios que interactuarán con el sistema de información de diferentes modos.

**Administrador:** tendrá la labor de ingresar, eliminar, modificar, copiar y realizar diferentes acciones sobre los registros de las tablas de la base de datos.

Figura N° 1. 19. Diagrama CasoUso-Administrador



Fuente: (Elaboración Propia).

**Solicitantes:** el usuario al que le es asignado este rol tiene acceso logeado al módulo donde podrá registrar datos, consultar actualizaciones de datos previamente registrados y actualizar información.

Tabla N° 1. 1. Creación de registro de usuario

<b>Nombre: Adicionar o Crear Usuarios</b>
<b>Actor(es): Administrador</b>
<b>Descripción:</b> <p style="text-align: center;">Se crean cuentas de usuario para ingreso al sistema por su respectivo rol, directamente en la tabla usuarios</p>
<b>Precondiciones:</b>
<b>Flujo Normal:</b> <ol style="list-style-type: none"> <li>1. El actor ingresa al módulo de administración.</li> <li>2. Antes debe registrar personas con sus datos.</li> <li>3. En la tabla usuarios crea un nuevo registro con los datos el usuario (usuario, persona, correo, categoría y estado).</li> <li>4. Se le asigna usuario y su password.</li> </ol>
<b>Flujo Alternativo: No aplica</b>
<b>Pos condición : 1.</b> <p style="text-align: center;">Cuenta creada</p>

**Fuente:** (Elaboración Propia).

Tabla N° 1. 2. Eliminar registro de Usuario

<b>Nombre: Eliminar Usuario</b>
<b>Actor(es): Administrador</b>
<b>Descripción:</b> <p>Se elimina el registro del usuario determinado en la tabla usuario.</p>
<b>Precondiciones:</b> <ol style="list-style-type: none"> <li>1. El usuario debe existir</li> <li>2. Si el usuario a registrado casos deben eliminarse los registros de las tablas correspondientes</li> </ol>

<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El actor ingresa al módulo de administración.</li> <li>2. En la tabla usuarios busca el usuario por nombre_usuario</li> <li>3. Selecciona y elimina el registro de la tabla</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>1. El actor desiste de la eliminación.</li> <li>2. El actor sale del aplicativo</li> </ol>
<p><b>Pos condición : 2.</b> <b>Cuenta eliminada</b></p>

**Fuente:** (Elaboración Propia).

**Tabla N° 1. 3. Modificar Usuario**

<b>Nombre: Modificar Usuario</b>
<b>Actor(es): Administrador</b>
<p><b>Descripción:</b> Se modifican atributos en la tabla usuario a excepción del campo PK.</p>
<p><b>Precondiciones:</b></p> <ol style="list-style-type: none"> <li>1. El usuario debe existir</li> </ol>
<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El actor ingresa al módulo de administración.</li> <li>2. En la tabla usuarios busca el usuario por código</li> <li>3. Edita y modifica el registro de la tabla</li> </ol>
<b>Flujo Alternativo: No Aplica</b>
<p><b>Pos condición :</b> <b>3. Cuenta modificada</b></p>

**Fuente:** (Elaboración Propia).

Tabla N° 1. 4. Guardar datos en la BD

<b>Nombre: Almacena gestión de registros de usuario</b>
<b>Actor(es): Administrador</b>
<b>Descripción:</b> Cada vez que se realiza alguna acción sobre los registros de la tabla usuarios es almacenada en la base de datos.
<b>Precondiciones:</b> <b>2. Se debe realizar alguna modificación</b>
<b>Flujo Normal:</b> 1. El actor ingresa al módulo de administración. 2. Realiza los cambio respectivos 3. Y verifica que se haya almacenado en la base de datos
<b>Flujo Alternativo: No Aplica</b>
<b>Pos condición :</b> <b>4. Se almacena las diferentes acciones sobre los registros en la base de datos</b>

## 7. MODULO DE ADMINISTRACION

Fue creado con el fin de que facilitar el trabajo al administrador del aplicativo.

A través de este módulo se podrán ver, editar, buscar y borrar registros de las tablas.

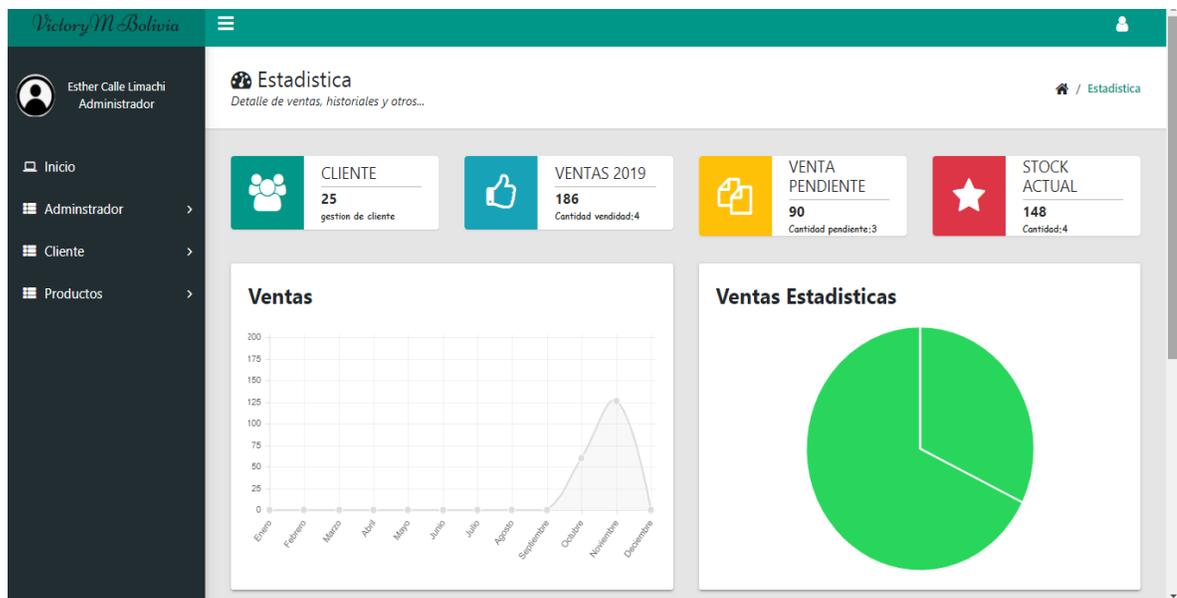
A continuación describimos el procedimiento:

1. se ingresa al módulo de apertura al aplicativo



### Pantalla de Inicio de Sesión

Ingresar datos como el correo y el password, luego ingresa a la pantalla principal de Administrador.



### Pantalla principal a los Módulos del Admin

# **ANEXO F DOCUMENTACIÓN**

---

- ✓ AVAL DE CONFORMIDAD DEL TUTOR METOLÓGICO.
- ✓ AVAL DE CONFORMIDAD DEL TUTOR REVISOR.
- ✓ AVAL DE CONFORMIDAD DEL TUTOR ESPECIALISTA.
- ✓ AVAL DE CONFORMIDAD DE LA EMPRESA VICTORY MOTORS BOLIVIA.

El Alto, agosto de 2020

Señor:  
Ing. David Carlos Mamani Quispe  
**DIRECTOR DE LA CARRERA INGENIERIA DE SISTEMAS**

Presente. –

Ref.- **AVAL DE CONFORMIDAD PROYECTO DE GRADO**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado “SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO DE VENTAS”, que propone la postulante Esther Eva Calle Limachi con CI: 7092335 LP., para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II de acuerdo a reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



Ing. Enrique Flores Baltazar  
TUTOR METODOLOGICO  
TALLER DE LICENCIATURA II

El Alto, agosto de 2020.

**Señor:**  
Msc. Ing. Enrique Flores Baltazar  
**TUTOR METODOLÓGICO TALLER II**

Presente. –

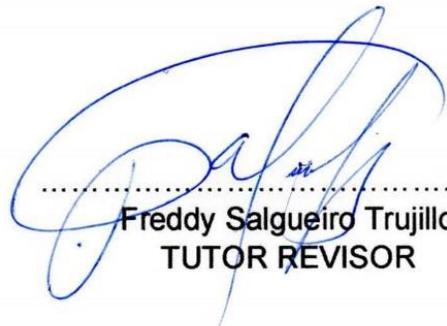
Ref.- **AVAL DE CONFORMIDAD**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO DE VENTAS", que propone la postulante Esther Eva Calle Limachi con CI: 7092335 LP., para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II de acuerdo a reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



.....  
Freddy Salgueiro Trujillo  
TUTOR REVISOR

El Alto agosto de 2020

Señor:  
Ing. Enrique Flores Baltazar  
**TUTOR METODOLÓGICO TALLER II**

Presente. –

Ref.- **AVAL DE CONFORMIDAD**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO DE VENTAS", que propone la postulante Esther Eva Calle Limachi con CI: 7092335 LP., para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II de acuerdo a reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



---

Ing. Edwin Mamani Viscarra  
**TUTOR ESPECIALISTA**



VICTORY MOTORS BOLIVIA

La Paz, Agosto 28 de 2020

Señor:

Ing. David Carlos Mamani Quispe

**DIRECTOR DE LA CARRERA INGENIERIA DE SISTEMAS**

UNIVERSIDAD PÚBLICA DE EL ALTO

Presente.-

**REF. AVAL DE CONFORMIDAD**

Distinguido Ingeniero:

Mediante la presente tengo a bien de comunicarle mi conformidad del proyecto de grado titulado: "SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO DE VENTAS" para la empresa: Victory Motors Bolivia, que propone el postulante Esther Eva Calle Limachi, con cedula de identidad 7092335 LP., para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II de acuerdo a reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



Diego Tarqui Ergueta  
JEFE ADMINISTRATIVO

