

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

SISTEMA PARA LA GESTIÓN DE VENTAS E INVENTARIO
CASO: IMPORTADORA SOLUCIONES MEDICAS LIFEMED
S.R.L.

Proyecto de Grado para obtener el Título de Licenciatura en Informática
Mención Ingeniería de Sistemas Informáticos

POSTULANTE: LA FUENTE CHOQUE JOHOVANA
TUTOR: LIC. JOSE LUIS ZEBALLOS ABASTO

LA PAZ – BOLIVIA
2021

HOJA DE CALIFICACIONES
UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

Proyecto de grado:

SISTEMA PARA LA GESTIÓN DE VENTAS E INVENTARIO

CASO: IMPORTADORA SOLUCIONES MEDICAS LIFEMED S.R.L.

Presentado por: Johovana La Fuente Choque

Para optar el grado Académico de Licenciada en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral: 80

Nota Literal: Ochenta

Ha sido: Sobresaliente

Director de la carrera de Informática: Lic. Nancy Imelda Orihuela Sequeiros

Tutor: Lic. José Luis Zeballos Abasto

Presidente de Tribunal: Lic. Brigida Alexandra Carvajal Blanco

Tribunal: Lic. Jhonny Roberto Felipez Andrade

Tribunal: P.Ph.D. Yohoni Cuenca Sarzuri

Tribunal: M.Sc. Aldo Ramiro Valdez Alvarado



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS, EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARÁ EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

RESUMEN

El presente proyecto de grado fue desarrollado para la implementación de un SISTEMA PARA LA GESTIÓN DE VENTAS E INVENTARIO para la IMPORTADORA SOLUCIONES MEDICAS LIFEMED S.R.L. dicha empresa está dedicada a la venta de equipos e insumos médicos.

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 óptimo y automatizado de las ventas, facturación e inventarios y que conlleva a mejorar la gestión de la información para la toma de decisiones en el negocio.

Palabras claves: Ventas, Facturación, Inventario, gestión, control.

El proyecto fue desarrollado bajo el enfoque de la Metodología Ágil XP que nos permite crear en cada etapa un prototipo cada vez mejorado hasta cumplir con las expectativas del cliente.

Dedicatoria

A Dios por regalarme vida, aun en estos días difíciles

A mis padres quienes son la principal inspiración de superación y fortaleza, a mi esposo Carlos(†) y mis hijas Nicole y Gianna por su apoyo y comprensión, a mis hermanos por su apoyo constante y confianza para seguir adelante.

Agradecimiento

A Dios por darme vida para llegar y disfrutar de la culminación de mi carrera.

A Lic. Grover Alex Rodríguez Ramírez, por su apoyo, dedicación y guía en el desarrollo y conclusión del presente proyecto, por su calidad humana y amabilidad.

A mi Docente Tutor Lic. José Luis Zeballos Abasto, por sus oportunas revisiones, correcciones y exigencias hacia el presente proyecto; para que el mismo llegue a su culminación de manera satisfactoria.

A mi hermano Alex La Fuente Choque, por enseñarme a soñar y que los sueños se pueden hacer realidad, siempre te llevare en mi corazón.

A los dueños de la empresa LIFEMED por su colaboración, confianza y apoyo en el transcurso del desarrollo de este proyecto.

A todos los docentes de la carrera de informática por brindarme los conocimientos para mi vida profesional, y a mis amigos que me acompañaron durante toda la carrera.

Gracias.

jolafu88@gmail.com

Contenido

1.	ANTECEDENTES GENERALES	1
1.1	INTRODUCCIÓN	1
1.2	ANTECEDENTES	2
1.2.1	ANTECEDENTES INSTITUCIONALES	2
1.2.2	ANTECEDENTES DE PROYECTOS SIMILARES	3
1.3	PLANTEAMIENTO DEL PROBLEMA	4
1.3.1	PROBLEMA PRINCIPAL.....	4
1.3.2	PROBLEMAS SECUNDARIOS.....	4
1.4	OBJETIVOS	5
1.4.1	OBJETIVO GENERAL	5
1.4.2	OBJETIVOS SECUNDARIOS	5
1.5	JUSTIFICACIONES	6
1.5.1	JUSTIFICACIÓN SOCIAL	6
1.5.2	JUSTIFICACIÓN ECONÓMICA	6
1.5.3	JUSTIFICACIÓN TECNOLÓGICA.....	7
1.6	ALCANCES	7
1.7	APORTES	8
1.7.1	APORTE PRÁCTICO.....	8
1.7.2	TEÓRICO.....	8
1.8	METODOLOGÍA.....	9
1.8.1	MÉTODO SISTEMÁTICO	9
2.	MARCO TEORICO.....	10
2.1	MARCO INSTITUCIONAL.....	10

2.1.1	RESUMEN DE LA INSTITUCION	10
2.1.2	ORGANIGRAMA.....	10
2.1.3	SISTEMA ACTUAL.....	11
2.2	METODOLOGÍAS ÁGILES.....	12
2.2.1	METODOLOGÍA XP.....	12
2.2.2	FASES PARA LA METODOLOGIA XP	15
2.3	TEMAS RELACIONADOS CON EL SISTEMA.....	22
2.3.1	COMPRA.....	22
2.3.2	VENTA	22
2.3.3	CONCEPTOS BÁSICOS DE INVENTARIO.....	23
2.3.4	FACTURACIÓN	23
2.4	TECNOLOGÍA DE SOFTWARE.....	23
2.4.1	LENGUAJE PHP	23
2.4.2	GESTORES DE BASE DE DATOS MYSQL.....	24
2.5	CALIDAD DE SOFTWARE ISO 9126.....	26
2.5.1	FUNCIONALIDAD	27
2.5.2	CONFIABILIDAD	28
2.5.3	USABILIDAD.....	29
2.5.4	EFICIENCIA.....	30
2.5.5	CAPACIDAD DE MANTENIMIENTO	31
2.5.6	PORTABILIDAD	33
2.5.7	CALIDAD EN USO	34
2.6	ESTUDIO DE COSTO Y BENEFICIO COCOMO.....	35
2.7	SEGURIDAD	36

2.7.1	FÍSICO.....	37
2.7.2	LÓGICO.....	38
3.	MARCO APLICATIVO.....	40
3.1	ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA	40
3.2	FASE I – PLANIFICACIÓN.....	40
3.2.1	HISTORIAS DE USUARIO.....	40
3.2.2	PLANIFICACIÓN DE ITERACIONES.....	46
3.2.3	PLANIFICACION DE TAREAS	48
3.3	FASE II - DISEÑO.....	54
3.3.1	IDENTIFICACIÓN DE LOS ROLES Y TAREAS DE LOS USUARIOS.....	54
3.3.2	MODELO DE CASO DE USO DEL NEGOCIO.....	55
3.3.3	MODELO DE ESTRUCTURA.....	56
3.4	FASE III – CODIFICACIÓN	57
3.5	FASE IV – PRUEBAS	60
4.	MÉTRICAS DE CALIDAD.....	63
4.1	INTRODUCCIÓN	63
4.2	CALIDAD DE SOFTWARE.....	63
4.2.1	USABILIDAD.....	63
4.2.2	FUNCIONALIDAD	65
4.2.3	CONFIABILIDAD	67
4.2.4	EFICIENCIA.....	68
4.2.5	CAPACIDAD DE MANTENIMIENTO	69
4.2.6	PORTABILIDAD	70
5.	EVALUACIÓN DE COSTO Y BENEFICIO	71

5.1	INTRODUCCIÓN	71
5.2	COSTO DEL SISTEMA	71
5.2.1	COSTO DEL SOFTWARE DESARROLLADO.....	72
5.2.2	COSTO DE IMPLEMENTACIÓN DEL SISTEMA	75
5.2.3	COSTO DE ELABORACIÓN DEL SISTEMA	75
5.2.4	COSTO TOTAL DEL SISTEMA.....	75
5.3	BENEFICIO	75
5.3.1	CÁLCULO DEL VALOR ACTUAL NETO (VAN).....	76
5.4	LA TASA INTERNA DE RETORNO (TIR).....	77
5.5	COSTO / BENEFICIO.....	79
6.	<i>SEGURIDAD DE SISTEMA</i>	80
6.1	SEGURIDAD DE SISTEMA OPERATIVO	80
6.2	SEGURIDAD DE BASE DE DATOS.....	80
6.3	SEGURIDAD SOFTWARE DE APLICACIÓN	81
7.	<i>CONCLUSIONES Y RECOMENDACIONES.....</i>	82
7.1	CONCLUSIONES.....	82
7.2	RECOMENDACIONES	83

INDICE DE ILUSTRACIONES

Ilustración 2.1 Diagrama Organizacional Empresa LIFEMED S.R.L.	11
Ilustración 2.2 Fases de la Metodología XP	15
Ilustración 2.3 Diagrama de casos de uso.....	21
Ilustración 2.4 Diagrama de clases	22
Ilustración 2.5 Esquema del funcionamiento de las páginas PHP	24
Ilustración 2.6 Modelo de ISO 9126.....	26
Ilustración 2.7 Funcionalidad	28
Ilustración 2.8 Confiabilidad	29
Ilustración 2.9 Usabilidad.....	30
Ilustración 2.10 Eficiencia	31
Ilustración 2.11 Capacidad de mantenimiento.....	32
Ilustración 2.12 Portabilidad.....	33
Ilustración 2.13 Calidad de uso.....	34
Ilustración 3.1 Caso de uso negocio – adición de producto.....	54
Ilustración 3.2 Caso de uso negocio – venta de producto.....	54
Ilustración 3.3 Diagrama Entidad Relación.....	55
Ilustración 3.4 Diagrama de clases	56
Ilustración 3.5 Pantalla de ingreso al sistema	57
Ilustración 3.6 Pantalla de registrar producto	58
Ilustración 3.7 Pantalla de registrar a un Cliente	58
Ilustración 3.8 Pantalla de registrar venta.....	59

INDICE DE TABLAS

Tabla 3.1 Historia de usuario 1	40
Tabla 3.2 Historia de usuario 2.....	40
Tabla 3.3 Historia de usuario 3.....	41
Tabla 3.4 Historia de usuario 4.....	41
Tabla 3.5 Historia de usuario 5.....	42
Tabla 3.6 Historia de usuario 6.....	42
Tabla 3.7 Historia de usuario 7.....	43
Tabla 3.8 Historia de usuario 8.....	43
Tabla 3.9 Historia de usuario 9.....	44
Tabla 3.10 Historia de usuario 10.....	44
Tabla 3.11 Historias de usuario de la Primera Iteración.....	46
Tabla 3.12 Historias de usuario de la Primera Iteración.....	46
Tabla 3.13 Historias de usuario de la Segunda Iteración.....	47
Tabla 3.14 Historias de usuario de la Tercera Iteración.....	47
Tabla 3.15 Historias de usuario de la Cuarta Iteración.....	47
Tabla 3.16 Tarjeta de Tarea 1 de Historia de usuario 1.....	48
Tabla 3.17 Tarjeta de Tarea 2 de Historia de usuario 2.....	48
Tabla 3.18 Tarjeta de Tarea 3 de Historia de usuario 3.....	49
Tabla 3.19 Tarjeta de Tarea 4 de Historia de usuario 4.....	49
Tabla 3.20 Tarjeta de Tarea 5 de Historia de usuario 8.....	50
Tabla 3.21 Tarjeta de Tarea 6 de Historia de usuario 5.....	50
Tabla 3.22 Tarjeta de Tarea 7 de Historia de usuario 6.....	51

Tabla 3.23 Tarjeta de Tarea 8 de Historia de usuario 7.....	51
Tabla 3.24 Tarjeta de Tarea 9 de Historia de usuario 9.....	52
Tabla 3.25 Tarjeta de Tarea 10 de Historia de usuario 10.....	52
Tabla 3.26 Plan de entregas.....	52
Tabla 3.27 Roles de usuarios del sistema.....	53
Tabla 3.28 Verificación de Usuarios para ingreso al sistema.....	60
Tabla 3.29 Verificación de ingreso de productos al sistema.....	60
Tabla 3.30 Verificación de registro de compra.....	61
Tabla 3.31 Verificación de proceso de venta.....	61
Tabla 4.1 Encuesta de Usabilidad del Sistema.....	63
Tabla 4.2 Conteo de parámetros de PF del sistema.....	64
Tabla 4.3 Ajuste de complejidad PF.....	65
Tabla 4.4 Factores de Eficiencia.....	68
Tabla 5.1 Conversión de puntos de función.....	71
Tabla 5.2 Conductores de Coste.....	72
Tabla 5.3 Modelo COCOMO.....	72
Tabla 5.4 Costos de elaboración del proyecto.....	74
Tabla 5.5 Costos total del sistema.....	74
Tabla 5.6 Flujo de caja proyectado a 5 años.....	75
Tabla 5.7 Interpretación del VAN.....	76
Tabla 5.8 Cálculo de TIR en hoja de Excel.....	77

1. ANTECEDENTES GENERALES

1.1 INTRODUCCIÓN

Todo sector u organización grande y pequeña tienden a depender de herramientas tecnológicas, para hacer más rápido el procesamiento de información. La más requerida e imprescindible herramienta en muchas empresas o microempresas es la computadora, es una herramienta muy importante en la vida diaria con la cual se ofrece grandes oportunidades.

La importadora Lifemed trabaja con una cantidad de información y por la cual comienza a necesitar la implementación de un software que agilice y organice mejor el desempeño así facilitara que la entidad de la empresa sea competitiva y eficiente. Muchas compañías de nueva creación tienen la oportunidad de cambiar las prácticas y procesos de negocio tradicionales que imponen altos niveles de estructura organizacional y costos; se debe comprender que el secreto de los negocios consiste en racionalizar los procesos comerciales para mejorar la eficiencia operativa de la empresa.

En la actualidad la importadora Soluciones Medicas Lifemed S.R.L. necesita contar con estos medios tecnológicos para reducir los procesos ya que se genera pérdidas de tiempo, aumento de costo de operación, atención al cliente por consecuencia reducción de ganancias. La importadora también carece de los constantes cambios tecnológicos que les permite un mayor nivel de competencia por tal razón que las tareas requieren un cambio tecnológico. El proyecto hace referencia al desarrollo de una aplicación para ayudar al control de registro de ventas e inventarios para la empresa Importadora Soluciones Medicas Lifemed S.R.L.

El presente proyecto trata sobre el control de las ventas y los inventarios de la empresa Importadora Soluciones Medicas Lifemed S.R.L. esto para su buena administración de los productos que masivamente se manipulan sin algún registro previo ocasionando un desorden

dentro de la empresa, se ha visto que la sistematización de procesos de manipulación de registro de datos acerca de ventas realizadas e inventariados dentro de una empresa es vital, ya que esto produce un riesgo menor en el caso de pérdida de información relevante a los casos de venta e inventariado.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INSTITUCIONALES

La Importadora Soluciones Medicas Lifemed S.R.L., es una empresa cuyo principal objetivo es la comercialización de equipos e insumos médicos para la salud.

La Importadora Soluciones Medicas Lifemed S.R.L. actualmente realiza sus tareas rutinarias de recopilar, transmitir, registrar y revisar información en sus sucursales las cuales se llevan a cabo de forma manual.

El proceso de ventas lo registran en cuadernos, así también la facturación es de forma manual usando talonarios impresos. No contemplando que los registros en papel son propensos a perderse o dañarse. Las facturas que realizan de manera digital, también no las toman en cuenta todas lo que dificulta el control de las ventas e incluso para cobros a los clientes.

En resumen, el control que hace la Importadora Soluciones Medicas Lifemed S.R.L. en los procesos de compras, ventas e inventarios, consistía en el registro de informes de las ventas y compras de productos que son centralizados a fin de mes y gestión, de forma manual esto, por supuesto, implicaba la inversión de mucho tiempo y esfuerzo, quedando sujeto a errores humanos que no garantizaban la fiabilidad de la información, ya que en la actualidad el problema de la pandemia hizo reducir el personal en las empresas y también los horarios de trabajo, se debe hacer teletrabajo y la mejor manera es tener todo en orden, para no ofrecer algún producto que ya no esta disponible.

1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES

Se desarrollan diversos Proyectos de Grado similares, una investigación preliminar evidencia que en la carrera de Informática de la Universidad Mayor de San Andrés existen algunos proyectos que se relacionan de alguna manera con el actual trabajo de investigación tales como:

- Título: “Sistema en plataforma mixta para el control ventas e inventarios con código qr”

Caso: “importadora LU.CE.R.”

Resumen: El presente documento contiene el proceso de desarrollo de un sistema de plataforma mixta para el control de ventas e inventarios, así saber el ingreso y egreso de productos, donde los mismos comprenden de filtros, lubricantes y separadores de aceite, estos tendrán la característica de tener pegado un código QR con la intención de capturar la información del mismo de forma rápida, cómoda y eficientemente desde un dispositivo móvil, de acuerdo a ello poder distribuir de manera eficiente los pedidos realizados por la empresa distribuidora LU.CE.R. Logrando la mejora de sus ingresos y eficiencia.. (Calle, 2018)

- Título: Control de Ventas e Inventarios

Caso: “COIMPE & LTDA.”

Resumen: Este proyecto está dirigido a la empresa COIMPE & LTDA., que tiene como área de trabajo la importación de piedras preciosas de diferentes países del mundo. El presente trabajo se vale de algunas técnicas de investigación para capturar la información y el tratamiento de la misma, de esta manera se delimita el ámbito de trabajo. (Aliaga, 2001)

- Título: Sistema de Control De Ventas e Inventarios Para Almacenes De Aluminios Utilizando Dispositivos Móviles

Caso: TECNICA DE ALUMINIO, VIDRIO Y SERVICIOS (TALVISER)”

Resumen: El diseño e implementación del sistema de control de inventarios, distribución y ventas mediante inventarios perpetuos, para desarrollar este proyecto se utilizó la metodología R.U.P. además de utilizar la modelo inventarios sin déficit, este modelo tiene como base el mantener un inventario sin falta de productos para desarrollar las actividades de cualquier empresa , además la métrica de software , en tanto métrica de calidad del software esta basa en la ISO 9126 el cual proporciona un esquema para la evaluación de la calidad del software y así refinarlo en un periodo determinado. (Gutierrez, 2015)

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 PROBLEMA PRINCIPAL

El problema principal en la Importadora Soluciones Medicas Lifemed S.R.L.es que el control de compra, ventas e inventario son realizados de forma manual que resulta insuficiente ante las necesidades de la empresa. La cantidad de información sobre las compras, las ventas diarias y la disposición de material ha crecido tanto que resulta complicada llevar un buen control al respecto ya que los mismos se anotan en talonarios y la pérdida de información se presenta con frecuencia en estas situaciones. Es por esta forma de procesar la información, que la empresa tiene problema y dificultad al momento de la toma de decisiones.

1.3.2 PROBLEMAS SECUNDARIOS

Entre las principales deficiencias que se tiene en las empresas se pueden nombrar las siguientes:

- Demora en la venta a los clientes ellos lo realizan de manera manual, en la tardanza de la entrega de los artículos.
- No tener toda la lista de precios de los productos.

- No dispone de información oportuna referente a la compra de productos.
- La búsqueda de artículos y precios es manual, causando un trabajo lento al momento de realiza las ventas
- Los procesos de movimientos de artículos en el almacén central y el sub almacén son lentos, esto hace que se pierda tiempo en reponer los mismos y se pierdan ventas.
- Reportes de las ventas registradas en desorden, causando equivocaciones.
- Se requiere reportes fuera de oficina, no obtenidas a tiempo por no tener un medio para obtenerlas al instante.
- No se tiene un registro actualizado de las ventas realizadas de los vendedores.
- La presentación de los productos se lo realiza en forma presencial o con envío de archivos, así también los pagos.

¿Cómo se puede mejorar el control de ventas e inventarios, de modo que se evite demoras en el proceso de facturación, registro de inventarios; y así generar reportes actualizados de ventas?

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Implementar un sistema para mejorar el control de ventas de la empresa Importadora Soluciones Medicas Lifemed S.R.L., para evitar demoras y así agilizar el proceso de facturación, registro de inventarios y generar reportes actualizados de ventas.

1.4.2 OBJETIVOS SECUNDARIOS

- Elaborar un módulo de ventas.
- Realizar un módulo para registros de los productos.

- Elaborar un módulo de compras.
- Realizar un módulo de inventarios.
- Efectuar un módulo que registre proveedores.
- Desarrollar un módulo que genere reportes de información del proceso de ventas e inventarios.

1.5 JUSTIFICACIONES

La existencia de normas y lineamientos específicos para el control de ventas e inventarios del almacén en las entidades privadas y/o públicas, hace necesario la creación de procedimientos acorde al reglamento interno de la institución.

1.5.1 JUSTIFICACIÓN SOCIAL

El presente proyecto es desarrollado con la finalidad de poder dar un óptimo control de ventas e inventarios a la empresa Importadora Soluciones Medicas Lifemed S.R.L., una alternativa de solución a la problemática de la empresa al momento de realizar registros adecuados de sus productos de importación.

En cuanto al personal de la importadora de igual forma serán beneficiados, pues hará que los procesos sean más sencillos y rápidos. Igualmente, los clientes serán beneficiados ya que recibirán una atención sin demoras.

1.5.2 JUSTIFICACIÓN ECONÓMICA

Este sistema reducirá los errores más comunes tanto en el llenado como en la elaboración de informes de ventas y facturación, reduciendo el costo de compra de papel, talonarios de facturas, demás papelería. Además, cuenta con el hardware necesario para implantar este sistema, por lo cual no significará un gasto adicional.

La implementación del sistema se realizará con software libre, lo que reducirá el costo de desarrollo notablemente, al no comprar licencias.

1.5.3 JUSTIFICACIÓN TECNOLÓGICA

El sistema a desarrollar, se realiza por la necesidad que tiene la importadora, ya que no cuenta con un buen control de ventas, así poder optimizar los servicios que presta el mismo.

Los componentes técnicos cómo tecnológicos que se están empleando en el presente desarrollo de software son computadoras, lenguajes de programación tales como php, mysql, html que impulsarán no sólo al mejor tratamiento de información si no a la innovación tecnológica que nos permitirá diferenciarnos de las otras empresas.

1.6 ALCANCES

Este proyecto será implementado en el área de ventas de la empresa Importadora Soluciones Medicas Lifemed S.R.L., para lo cual los alcances del aplicativo serán los siguientes:

- Módulo de Administración de usuarios.
- Módulo de Registro de Compras.
- Módulo de ventas
- Módulo de reportes sobre ventas.
- Módulo de inventario de productos

Así mismo, este proyecto será implementado en la casa matriz que se encuentra en la ciudad de La Paz (Bolivia), en la Zona Sopocachi Bajo, Av. Arce #2631, Piso 6, oficina 610.

1.7 APORTES

1.7.1 APORTE PRÁCTICO

El aporte práctico del proyecto está dirigido al campo de negocio que se pretende generar tomando como base el uso de una herramienta sistematizada que le permitirá al usuario mejorar la calidad del servicio en todos los sentidos.

1.7.2 TEÓRICO

La Programación Extrema XP que es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo, con la ayuda de UML para el modelado de proyecto.

Método Promedio Ponderado o Precio Promedio Ponderado (PPP) que se basa en el supuesto de que tanto el costo de ventas como el de los inventarios finales deben evaluarse a un costo promedio

- Costo de los artículos disponibles
- Unidades disponibles para la venta (Stock)

Al igual que PEPS y UEPS, también es un método que se utiliza en el sistema de inventario permanente.

Con este método lo que se hace es determinar un promedio, sumando los valores existentes en el inventario con los valores de las nuevas compras, para luego dividirlo entre el número de unidades existentes en el inventario incluyendo tanto los inicialmente existentes, como los de la nueva compra.

1.8 METODOLOGÍA

1.8.1 MÉTODO SISTEMÁTICO

El enfoque sistemático es una manera de abordar y formular problemas con vistas a una mayor eficacia en la acción, que se caracteriza por concebir a todo objeto como un sistema o componente de un sistema, entendiendo por sistema una agrupación de partes entre las que se establece alguna forma de relación que las articule en la unidad que es precisamente el sistema. (Alquiles & Ferreras, 1995)

La metodología de desarrollo para el presente proyecto es XP (Extreme Programming), cuyo objetivo es conseguir la máxima satisfacción del cliente en forma rápida y eficiente ante los cambios de requisitos, en las actividades, análisis y diseño.

La programación extrema tiene las siguientes características:

- Comunicación constante entre el cliente y el equipo de desarrollo.
- Respuesta rápida a los cambios constantes.
- La planificación es abierta con un cronograma de actividades flexible.
- El software que funciona está por encima de cualquier otra documentación.
- Los requisitos del cliente y el trabajo del equipo del proyecto son los principales

factores de éxito del mismo. (Beck, 1999)

2. MARCO TEORICO

2.1 MARCO INSTITUCIONAL

2.1.1 RESUMEN DE LA INSTITUCION

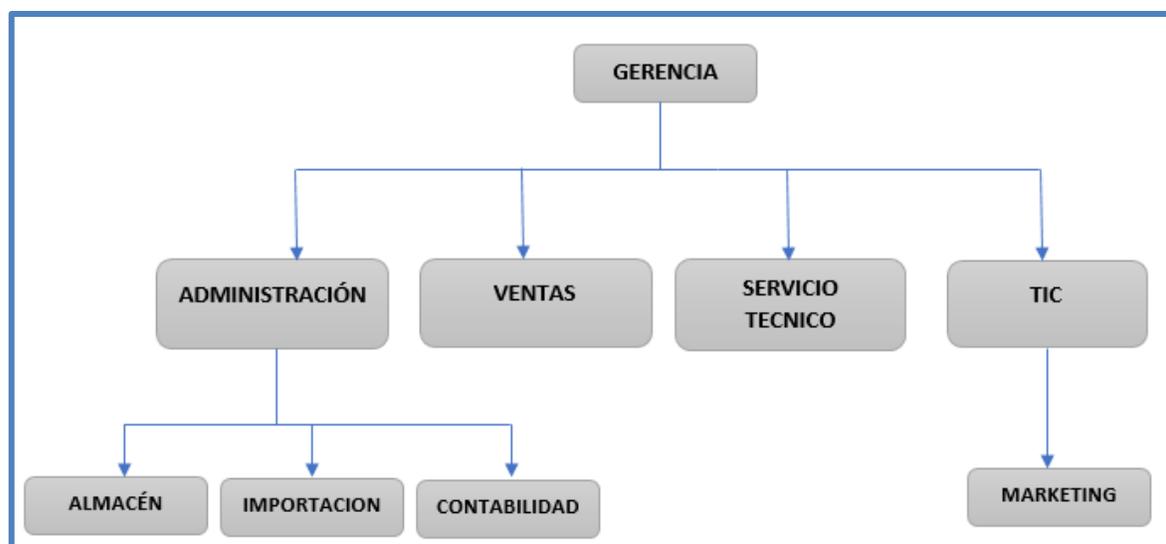
Importadora Soluciones Medicas Lifemed S.R.L., es una empresa cuyo principal objetivo es la comercialización de equipos e insumos médicos para la salud en todo el territorio boliviano y tiene como casa Matriz la sucursal en la ciudad de La Paz.

Misión: Somos una empresa cuyo principal objetivo es la comercialización de equipos e insumos médicos para la salud, fue creada como una respuesta al creciente desarrollo tecnológico y la necesidad de satisfacer demandas cada vez más exigentes. Contamos con personal calificado, en cada una de las áreas para mejorar las oportunidades y la calidad de la salud en nuestro país, comprometidos con la atención oportuna, suministrando soluciones y apoyo tecnológico a las diferentes Especialidades.

Visión: Ser una empresa de excelencia para el equipamiento de hospitales y productos relacionados a la salud para la población en general, haciéndonos referencia boliviana, facilitando y garantizando toda compra directa a través de una atención oportuna, eficiente y beneficiosa al cliente final.

2.1.2 ORGANIGRAMA

Importadora Soluciones Medicas Lifemed S.R.L. presenta el siguiente organigrama, como se muestra en la Ilustración 2.1:



*Ilustración 2.1 Diagrama Organizacional Empresa LIFEMED S.R.L.
Fuente: (Lifemed, 2017)*

Importadora Soluciones Medicas Lifemed S.R.L. actualmente realiza sus tareas rutinarias de recopilar, transmitir, registrar y revisar información en sus almacenes, se llevan a cabo de forma manual.

2.1.3 SISTEMA ACTUAL

El proceso de ventas lo registran en cuadernos, así también la facturación es de forma manual usando talonarios impresos. No contemplando que los registros en papel son propensos a perderse o dañarse.

Hay que tomar en cuenta que para realizar un arqueo de venta diaria se debe revisar todas las facturas de ese día, sumarlas una a una para poder obtener esa información, sin olvidar que en cualquier momento se puede pasar por alto una factura, produciendo datos erróneos, y al momento de registrar esos datos en los arqueos diarios para realizar estadísticas de la oficina se produce un desequilibrio contable y difícil de corregir.

2.2 METODOLOGÍAS ÁGILES

Existen numerosas propuestas de metodología para desarrollar software.

Tradicionalmente estas metodologías se centran en el control del proceso, estableciendo rigurosamente las actividades, herramientas y notaciones al respecto, dado estas reglas estas metodologías se caracterizan por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. En este escenario, las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico.

Los objetivos de las metodologías ágiles, entre los cuales se destaca la preferencia de algunos valores por sobre otros, por ejemplo: Individuos e interacciones, sobre procesos y herramientas, Software operativo, sobre documentación extensiva Y Colaboración con el cliente, sobre negociación de Contratos. (Letelier & Patricio, 2006)

2.2.1 METODOLOGÍA XP

Extreme Programming es una metodología ágil para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo, centrada en potenciar las relaciones interpersonales como clave para el éxito del desarrollo de software. Está diseñada para el desarrollo de aplicaciones que requieran un grupo de programadores pequeño, donde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Beck, 1999)

2.2.1.1 VALORES DE LA METODOLOGÍA XP

Los Valores originales de la programación extrema según (Bustamante D., 2014) son: simplicidad, comunicación, retroalimentación, coraje y respeto, donde se detallan a continuación:

- **Simplicidad:** Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases.
- **Comunicación:** La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor cuanto más simple sea y también la comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.
- **Retroalimentación (FEEDBACK):** Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Meses de trabajo pueden tirarse

por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo.

➤ **Coraje o Valentía:** Muchas de las prácticas implican valentía, una de ellas es siempre diseñar y programar para hoy y no para mañana. La valentía permite a los desarrolladores que se sientan cómodos con reconstruir su código cuando sea necesario. También significa persistencia: donde el programador puede permanecer sin avanzar en un problema complejo por un día entero, y luego lo resolverá rápidamente al día siguiente, sólo si es persistente.

2.2.1.2 EL PROCESO DE DESARROLLO EN XP

La programación extrema parte del caso habitual de una compañía que desarrolla software, generalmente software a medida, en la que hay diferentes roles: un equipo de gestión, un equipo de desarrolladores y los clientes. La relación con el cliente es totalmente diferente a lo que se ha venido haciendo en las metodologías tradicionales que se basan fundamentalmente en una fase de captura de requisitos previa al desarrollo y una fase de validación posterior al mismo. Para que el presente proyecto se ejecute debe seguir pasos donde el desarrollo es la pieza clave de todo el proceso XP. Todas las tareas tienen como objetivo realizar el desarrollo a la máxima velocidad, sin interrupciones y siempre en la dirección correcta. El proceso de desarrollo tiene los siguientes pasos:

- ✓ El cliente define el valor de negocio a implementar.
- ✓ El programador estima el esfuerzo necesario para su implementación.
- ✓ El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- ✓ El programador construye ese valor de negocio.

- ✓ Vuelve al paso 1. (Robles G., 2002)

2.2.2 FASES PARA LA METODOLOGIA XP

XP abarca fases que ocurren en el contexto de 4 actividades del marco de trabajo, las mismas que se muestra en la Ilustración 2.2 y posteriormente se hace la descripción fase por fase.

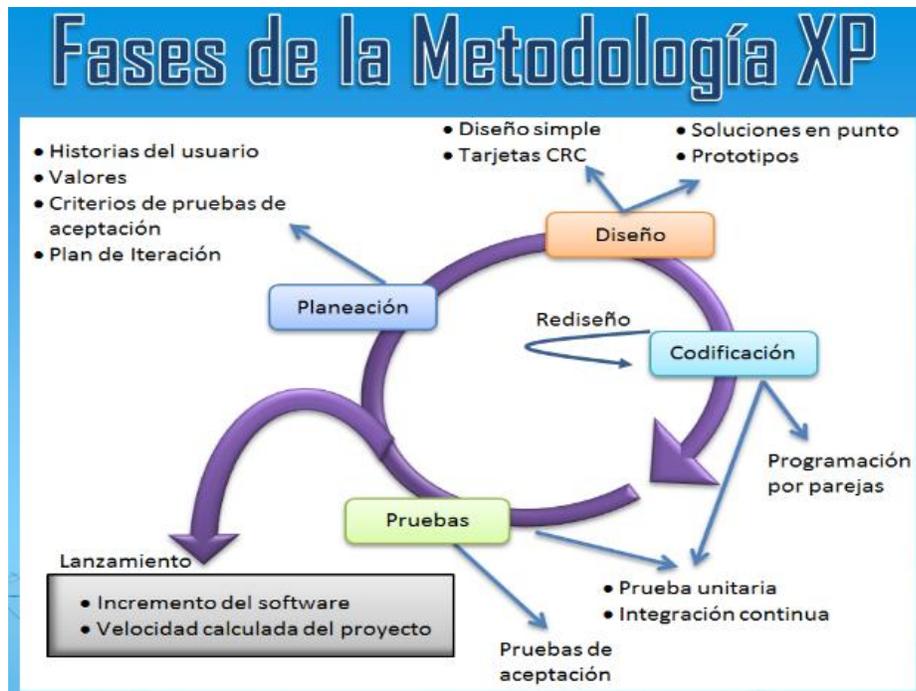


Ilustración 2.2 Fases de la Metodología XP
Fuente: Bustamante y Rodríguez, 2014

2.2.2.1 FASE I – PLANIFICACION DEL PROYECTO

XP plantea la planificación como un permanente dialogo entre la parte empresarial y técnica del proyecto, en la que los primeros decidirán el alcance, la prioridad, la 21 composición de las versiones y la fecha de las mismas. En cuanto a los técnicos son los responsables de estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las consecuencias de determinadas decisiones, de organizar el trabajo u finalmente de realizar la planificación detallada dentro de cada versión. (Beck, 1999).

➤ Historias de usuario: El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso, pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, entre otros. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas. (Beck, 1999)

➤ Plan de entregas: Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Un Release planes una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después de un Release plan tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado. (Beck, 1999)

➤ Iteraciones: Todo proyecto que siga la metodología X.P. se ha de dividir en iteraciones de aproximadamente 1 a 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Plan de entregas" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores. Para cada iteración se define un módulo al conjunto de historia de usuario que se van a implementar, al final de cada iteración se tiene la entrega de un producto, el cual debe superar las pruebas de aceptación que establece el cliente para dar cumplimiento a los requisitos las tareas que no se vean cubiertas por el producto deberán ser tomadas en cuenta para la siguiente iteración. (Beck, 1999)

➤ La Velocidad del Proyecto: Es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo Release Plan. La velocidad de proyecto se usa para determinar cuántas historias de usuario puede ser implementada antes de una fecha dada (tiempo), o cuanto tiempo es necesario para llevar a cabo un conjunto de historias (alcance). Cuanto se realiza una planificación por alcance se divide el número total de semanas entre la velocidad del proyecto para determinar cuántas iteraciones estarán disponibles.

- Programación en Parejas: La metodología X.P. aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad
- Reuniones Diarias: Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.
- Entregas Pequeñas: Al final de cada iteración una entrega de avances del producto, los cuales deberán ser completamente funcionales y estas se caracterizan por ser frecuentes.
- Roles XP: Los roles se asignan de acuerdo al punto 7.2.1.3 anteriormente definida.
- Ajustar a XP: Todos los proyectos tienen características específicas por lo cual XP puede ser modificado para ajustarse bien al proyecto en cuestión. Al iniciar el proyecto se debe aplicar XP tal como es sin embargo no se debe dudar en modificar aquellos aspectos en que no funcione. (Beck, 1999).

2.2.2.2 FASE II - DISEÑO

En XP se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio, el segundo motivo dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo. (Beck, 1999)

- **Diseños Simples:** La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible.
- **Glosarios de Términos:** Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.
- **Riesgos:** Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.
- **Funcionabilidad extra:** Nunca se debe añadir funcionalidad extra al programa, aunque se piense que en un futuro será utilizada.
- **Re factorizar:** Re factorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Re factorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento.
- **Metáfora del sistema:** Se trata de plasmar la arquitectura del sistema en una “historia” con la cual se le da al grupo de desarrollo una misma visión sobre el proyecto además de brindarles un primer vistazo muy completo de los nuevos integrantes del grupo para hacer su adaptación más rápida.
- **Tarjetas CRC (Clase-Responsabilidad-Colaboración):** La principal funcionalidad que tiene estas tarjetas, es ayudar a dejar el pensamiento procedimental para incorporarse el enfoque orientado a objetos. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. En la siguiente figura se muestra cómo se distribuye estas informaciones.

➤ **No Solucionar antes de tiempo:** En XP solo se analiza lo que se desarrollara en la iteración actual, olvidando por completo cualquier necesidad que se pueda presentar a futuro. (Echeverry, 2007)

2.2.2.3 FASE III - CODIFICACION

La codificación es un proceso que se realiza en forma paralela al diseño. La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente, facilita su comprensión y escalabilidad.

➤ **Cliente Siempre Presente:** Uno de los requerimientos de XP es que el cliente este siempre presente y disponible. No solamente para solucionar las dudas del grupo de desarrollo, debería ser parte de este.

➤ **Codificar Primero La Prueba:** Cuando se crea la primera prueba, se ahorra mucho tiempo elaborando el código que la haga pasar, siendo menor el tiempo de hacer ambos procesos que crear el código solamente.

➤ **No trabajar más de 40 horas semanales: trabajar** horas extras absorbe al espíritu y la motivación del equipo. Aquellos proyectos que requiera horas extras para acabarse a tiempo pueden convertirse en un problema en lugar de esto se utilizara las conocidas reuniones también es una mala idea incorporar nueva gente al proyecto. (Echeverry, 2007)

2.2.2.4 FASE IV -PRUEBAS

Es imprescindible en la metodología XP los test para comprobar el funcionamiento exitoso de los códigos que se vayan implementando.

El uso de los test en XP es el siguiente:

✓ Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

- ✓ Hay que someter al test las distintas clases del sistema omitiendo los métodos más triviales.
- ✓ Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- ✓ Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- ✓ Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- ✓ Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

2.2.2.5 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, como se muestra en la Ilustración 2.3.

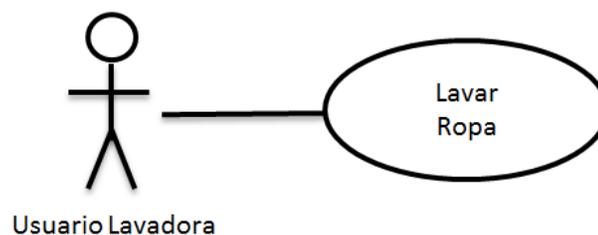


Ilustración 2.3 Diagrama de casos de uso
Fuente: "Aprendiendo UML en 24 Horas". Schmuller, J. (2004).

2.2.2.6 DIAGRAMA DE CLASES

Una clase es una categoría o grupo de cosas que tienen atributos y acciones similares, como por ejemplo la clase lavadora, cuenta con los atributos marca, modelo, número de serie,

capacidad. También posee ciertos comportamientos o métodos tal como: agregar ropa, agregar detergente y sacar ropa, como se puede ver en la Ilustración 2.4.

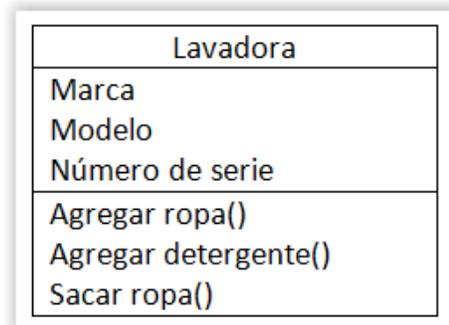


Ilustración 2.4 Diagrama de clases

Fuente: "Aprendiendo UML en 24 Horas". Schmuller, J. (2004). p. 28

2.3 TEMAS RELACIONADOS CON EL SISTEMA

2.3.1 COMPRA

Las compras son actividad altamente calificada y especializada. Deben ser analíticas y racionales para lograr objetivos de una acertada gestión de adquisiciones que resume en adquirir productos y servicios en cantidad, calidad, precio, momento sitio proveedor justo y adecuado, buscando la máxima rentabilidad de la empresa y una motivación para que el proveedor desee seguir realizando negocios con su cliente. (Arana, 2014)

2.3.2 VENTA

Tiene múltiples definiciones dependiendo del contexto en el que se maneje. La venta es el intercambio de servicios y productos. Desde el punto de vista contable y financiero, la venta es el monto total cobrado por productos o servicios prestados. En cualquier situación, las ventas son el corazón de cualquier negocio y actividad fundamental. (Arana, 2014)

2.3.3 CONCEPTOS BÁSICOS DE INVENTARIO

Los inventarios son un puente de unión entre la producción y las ventas. En una empresa manufacturera el inventario equilibra la línea de producción si algunas maquinas operan a diferentes volúmenes de otras, pues una forma de compensar este desequilibrio es proporcionando inventarios temporales o bancos. Los inventarios de materias primas, productos semiterminados y productos terminados absorben la holgura cuando fluctúan las ventas. Estos tienden a proporcionar un flujo constante de producción, facilitando su programación. Los inventarios de materia dan flexibilidad al proceso de compra de la empresa. Sin ellos en la empresa existen una situación “de la mano a la boca”, comprándose la materia prima estrictamente necesaria para mantener el plan de producción, es decir comprando y consumiendo.

2.3.4 FACTURACIÓN

Básicamente consiste en: el contribuyente envía datos de una o varias transacciones (fecha, nombre, documento de identificación e importe) para que la Agencia Tributaria complete la información de las facturas (N.º de autorización, N.º de factura, fecha límite de emisión y código de control), posteriormente devuelve al contribuyente a través de comprobante electrónico, para su emisión y entrega al cliente. (Globaleinvoicing.com, 2019)

2.4 TECNOLOGÍA DE SOFTWARE

2.4.1 LENGUAJE PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y

mucha documentación. Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente.

Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente, ver Ilustración 2.5.

El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de correo electrónico, subida de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales. (Suehring S., 2009)

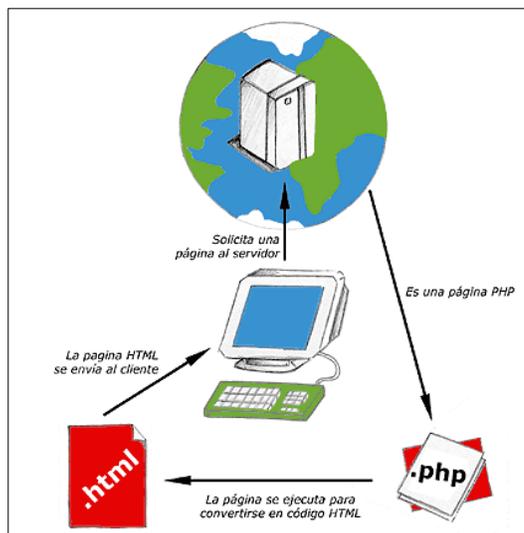


Ilustración 2.5 Esquema del funcionamiento de las páginas PHP
Fuente: DesarrolloWeb

2.4.2 GESTORES DE BASE DE DATOS MYSQL

MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de

MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. Y así ocurre. MySQL ha pasado de ser una "pequeña" base de datos a una completa herramienta y ha conseguido superar a una gran cantidad de bases de datos comerciales (lo que ha asustado a la mayor parte de los proveedores comerciales de bases de datos).

MySQL fue creada por la empresa sueca MySQL LAB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQLAB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario ya que, de no ser así, se vulneraría la licencia GPL. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (Thomson, 2003)

Las principales características de este gestor de bases de datos son las siguientes:

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multi-hilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

2.5 CALIDAD DE SOFTWARE ISO 9126

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad.

Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo, el deseo de un buen mantenimiento) que también pueden implicar una falta de calidad. Las métricas se utilizan para supervisar y controlar un proyecto de software. El estándar ISO 9126, ha sido desarrollado en un intento de identificar los atributos de calidad para el software, se utilizara el punto función para este proyecto, como se muestra en la Ilustración 2.6. El estándar identifica los siguientes atributos de calidad. (Pressman, 2010)



Ilustración 2.6 Modelo de ISO 9126

2.5.1 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. Como se ve en la Ilustración 2.7. A continuación, se muestra la característica de Funcionalidad y las sub características que cubre: (Medina, 2017)

- **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.

$$PF = \text{Cuenta total} * (X + \text{Min}(Y) * \sum Fi)$$

Donde:

PF: Medida de la funcionalidad.

X: Confiabilidad del proyecto, varía entre el 1% a 100%

Min (y): Error mínimo aceptable al de la complejidad, el margen de error es igual a 0.01.

Fi: Son los valores de ajuste de la complejidad, donde $i=1$ a $i=14$.



Ilustración 2.7 Funcionalidad

Fuente: ISO/IEC 2007

2.5.2 CONFIABILIDAD

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. Como se ve en la Ilustración 2.8. En este caso a la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida. (Medina, 2017)

- **Madurez** - Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.
- **Recuperabilidad** - Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- **Tolerancia a errores** - Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.

- **Conformidad de Confiabilidad** - La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.

La función a continuación muestra el nivel de confiabilidad del sistema:

$$F(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

Se ve el trabajo hasta que se observa un fallo en un instante t, la función es la siguiente:

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de no hallar una falla: $P(T > t) = 1 - F(t)$



Ilustración 2.8 Confiabilidad

Fuente: ISO/IEC 2007

2.5.3 USABILIDAD

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios. Como se ve en la Ilustración 2.9. (Medina, 2017)

- **Aprendizaje**- Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- **Comprensión** - Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.

- **Operatividad** - Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.
- **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.



Ilustración 2.9 Usabilidad

Fuente: ISO/IEC 2007

2.5.4 EFICIENCIA

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas. Como se ve en la Ilustración 2.10. (Medina, 2017)

- **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.



Ilustración 2.10 Eficiencia

Fuente: ISO/IEC 2007

2.5.5 CAPACIDAD DE MANTENIMIENTO

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software. Como se ve en la Ilustración 2.11. (Medina, 2017)

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

- **Cambialidad:** 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.
- **Portabilidad:** La capacidad que tiene el software para ser trasladado de un entorno a otro.

$$IMS = \frac{Mt - (Fc + Fa + FE)}{Mt}$$

Dónde:

Mt: Número de módulos total de la versión actual

Fc: Número de módulos de la versión actual que se cambiaron.

Fa: Número de módulos de la versión actual que se añadieron.

FE: Número de módulos de la versión anterior que se eliminaron en la versión actual.



Ilustración 2.11 Capacidad de Mantenimiento

Fuente: ISO/IEC 2007

2.5.6 PORTABILIDAD

Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra. Como se ve en la Ilustración 2.12. (Medina, 2017)

- **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.
- **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 softwares, 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.
- **Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.



Ilustración 2.12 Portabilidad

Fuente: ISO/IEC 2007

2.5.7 CALIDAD EN USO

Conjunto de atributos relacionados con la aceptación por parte del usuario final y Seguridad.

Como se ve en la Ilustración 2.13. (Medina, 2017)

- **Eficacia** - Atributos relacionados con la eficacia del software cuando el usuario final realiza los procesos.
- **Productividad** - Atributos relacionados con el rendimiento en las tareas cotidiana realizadas por el usuario final.
- **Seguridad** - Atributos para medir los niveles de riesgo.
- **Satisfacción** - Atributos relacionados con la satisfacción de uso del software



Ilustración 2.13 Calidad en Uso

Fuente: ISO/IEC 2007

2.6 ESTUDIO DE COSTO Y BENEFICIO 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 costos¹ 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"

Las ecuaciones que se utilizan en los tres modelos son:

$$E = a (KLDC)^b * m(X), \text{ en persona-mes}$$

$$Tdev = c(E)^d, \text{ en meses}$$

$$P = E/Tdev, \text{ en personas}$$

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.

2.7 SEGURIDAD

El área de la informática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta y especialmente, la información contenida o circulante es la seguridad informática o seguridad de tecnologías de la información. Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información.

2.7.1 FÍSICO

- **Control de la red:** Los puntos de entrada en la red son generalmente el correo, las páginas web y la entrada de ficheros desde discos, o de ordenadores ajenos, como portátiles. Mantener al máximo el número de recursos de red solo en modo lectura, impide que ordenadores infectados propaguen virus. En el mismo sentido se pueden reducir los permisos de los usuarios al mínimo. Se pueden centralizar los datos de forma que detectores de virus en modo batch puedan trabajar durante el tiempo inactivo de las máquinas. Controlar y monitorizar el acceso a Internet puede detectar, en fases de recuperación, cómo se ha introducido el virus.
- **Protección física de acceso a las redes:** Independientemente de las medidas que se adopten para proteger a los equipos de una red de área local y el software que reside en ellos, se deben tomar medidas que impidan que usuarios no autorizados puedan acceder. Las medidas habituales dependen del medio físico a proteger. A continuación, se enumeran algunos de los métodos, sin entrar al tema de la protección de la red frente a ataques o intentos de intrusión desde redes externas, tales como Internet.
- **Redes cableadas:** Las rosetas de conexión de los edificios deben estar protegidas y vigiladas. Una medida básica es evitar tener puntos de red conectados a los switches. Aun así, siempre puede ser sustituido un equipo por otro no autorizado con lo que hacen falta medidas adicionales: norma de acceso 802.1x, listas de control de acceso por MAC addresses, servidores de DHCP por asignación reservada, etc.
- **Redes inalámbricas:** En este caso el control físico se hace más difícil, si bien se pueden tomar medidas de contención de la emisión electromagnética para circunscribirla a aquellos lugares que consideremos apropiados y seguros. Además, se consideran medidas de

calidad el uso del cifrado (WPA, WPA v.2, uso de certificados digitales, etc.), contraseñas compartidas y, también en este caso, los filtros de direcciones MAC, son varias de las medidas habituales que cuando se aplican conjuntamente aumentan la seguridad de forma considerable frente al uso de un único método.

2.7.2 LÓGICO

2.7.2.1 RESPALDO DE INFORMACIÓN

La información constituye el activo más importante de las empresas, pudiendo verse afectada por muchos factores tales como robos, incendios, fallas de disco, virus u otros. Desde el punto de vista de la empresa, uno de los problemas más importantes que debe resolver es la protección permanente de su información crítica. La medida más eficiente para la protección de los datos es determinar una buena política de copias de seguridad o backups. Este debe incluir copias de seguridad completa (los datos son almacenados en su totalidad la primera vez) y copias de seguridad incrementales (solo se copian los ficheros creados o modificados desde el último backup). Es vital para las empresas elaborar un plan de backup en función del volumen de información generada y la cantidad de equipos críticos.

Un buen sistema de respaldo debe contar con ciertas características indispensables:

- **Continuo:** El respaldo de datos debe ser completamente automático y continuo. Debe funcionar de forma transparente, sin intervenir en las tareas que se encuentra realizando el usuario.
- **Seguro:** Muchos softwares de respaldo incluyen cifrado de datos, lo cual debe ser hecho localmente en el equipo antes del envío de la información.
- **Remoto:** Los datos deben quedar alojados en dependencias alejadas de la empresa.

➤ **Mantenimiento de versiones anteriores de los datos:** Se debe contar con un sistema que permita la recuperación de, por ejemplo, versiones diarias, semanales y mensuales de los datos.

2.7.2.2 PROTECCIÓN CONTRA VIRUS

Los virus son uno de los medios más tradicionales de ataque a los sistemas y a la información que sostienen. Para poder evitar su contagio se deben vigilar los equipos y los medios de acceso a ellos, principalmente la red.

3. MARCO APLICATIVO

3.1 ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA

En el sistema estamos usando la metodología eXtreme Programming (XP), desarrollando mediante las fases que nos señala, con la ayuda de herramientas del lenguaje unificado (UML) para el diseño.

3.2 FASE I – PLANIFICACIÓN

En esta fase se mostrará la forma de trabajo actual mediante las Historias de Usuarios, además de los requerimientos del sistema se definirán todas las tareas que serán necesarias para poder desarrollar esto mediante las Tarjetas de Tareas y por último se realizará el Plan de Entregas que contendrá las iteraciones a realizar para el desarrollo del presente proyecto.

3.2.1 HISTORIAS DE USUARIO

A partir del conjunto de requerimientos, se pudo construir junto al cliente un conjunto de historias de usuario, las mismas que cuentan con prioridades, riesgos e iteraciones que son descritos a continuación:

H.U. de productos: El sistema debe registrar la información necesaria de los productos con la que cuenta la empresa, la información del producto debe ser actualizada constantemente para que no exista errores de alguna forma por ejemplo el cambio de precio ya sea por el incremento o el decremento del precio.

HISTORIA DE USUARIO	
Número: 1	Nombre: Producto
Iteración asignada: 1	Prioridad: Alta
<p>Descripción:</p> <p>El registro de productos nuevos y actualización de productos ya existentes es indispensable para el sistema, los datos necesarios de los artículos los cuales son: código del producto, nombre del producto, descripción del producto, precio de entrada, precio de salida, cantidad del producto.</p>	

Tabla 3.1 Historia de usuario 1

Fuente: Elaboración propia

HISTORIA DE USUARIO	
Número: 2	Nombre: Categoría
Iteración asignada: 1	Prioridad: Media
<p>Descripción:</p> <p>El sistema debe tener la opción de crear, listar, modificar y eliminar registros de categorías.</p>	

Tabla 3.2 Historia de usuario 2

Fuente: Elaboración propia

HISTORIA DE USUARIO	
Número: 3	Nombre: Clientes
Iteración asignada: 1	Prioridad: Baja
Descripción: El sistema debe tener la opción de crear, listar, modificar y eliminar registros de clientes. Donde se tiene los datos de: Nombre del Cliente, Apellidos del cliente, Número de identificación del cliente, Email y teléfono.	

Tabla 3.3 Historia de usuario 3

Fuente: Elaboración propia

H.U. de ventas: El sistema debe registrar las ventas que se realizan de acuerdo a los productos con los que cuenta para lo cual habrá un buscador de productos por nombre o por código para posteriormente realizar las ventas. El mismo que será atendido por el personal de venta o el administrador.

HISTORIA DE USUARIO	
Número: 4	Nombre: Ventas
Iteración asignada: 2	Prioridad: Alta
Descripción: En este módulo se realizará el registro de las ventas de los productos teniendo en cuenta los datos necesarios para la operación, así como: Código de usuario que realiza la venta, id de venta, subtotal, descuento, total, fecha.	

Tabla 3.4 Historia de usuario 4

Fuente: Elaboración propia

H.U. de proveedores: El sistema debe registrar los datos de los proveedores con los que tiene contacto la empresa para tener más información de todos los productos que proveen, también para el control de compras o abastecimiento de los artículos faltantes o nuevos.

HISTORIA DE USUARIO	
Número: 5	Nombre: Proveedores
Iteración asignada: 3	Prioridad: Alta
Descripción: El sistema debe tener la opción de administrar los proveedores de la importadora, se manejarán los siguientes datos: Nombre de proveedor, país del proveedor, dirección y teléfono.	

Tabla 3.5 Historia de usuario 5

Fuente: Elaboración propia

H.U. de Compras: Para la empresa el abastecimiento es muy importante por la que dependerá su funcionamiento adecuado, para la toma de decisiones, el control ya que está relacionado con el inventario, con proveedores, con compras, con ventas por tal razón es esencial para la empresa.

HISTORIA DE USUARIO	
Número: 6	Nombre: Compras
Iteración asignada: 3	Prioridad: Alta
Descripción: El administrador es quien realiza las compras, o el personal de almacén podrá realizar el registro de los datos de los productos que ingresan y el stock que ingresa.	

Tabla 3.6 Historia de usuario 6

Fuente: Elaboración propia

H.U. de inventarios: En esta área se podrá mostrar un reporte de todos los productos existentes, cantidades disponibles con su respectiva descripción.

HISTORIA DE USUARIO	
Número: 7	Nombre: Inventarios
Iteración asignada: 4	Prioridad: Alta
<p>Descripción:</p> <p>Se desarrollará el módulo de inventario debido a que se necesita tener los reportes de todos los módulos del sistema. Este módulo mostrará los productos existentes, cantidades disponibles, además habrá un historial de cada producto donde se podrá obtener las entradas, las salidas, la disponibilidad y las fechas y horas de las acciones realizadas.</p>	

Tabla 3.7 Historia de usuario 7

Fuente: Elaboración propia

H.U. Usuarios: Las adiciones y ediciones del personal por el administrador son prioridades para la empresa pues el vendedor debe estar adicionado por el mismo.

HISTORIA DE USUARIO	
Número: 8	Nombre: Usuarios
Iteración asignada: 2	Prioridad: Alta
<p>Descripción:</p> <p>Se desarrollará el módulo de usuarios para el ingreso al sistema, se debe ingresar al sistema usando un nombre de usuario y contraseña.</p>	

Tabla 3.8 Historia de usuario 8

Fuente: Elaboración propia

H.U. Reporte de ventas: En esta área se podrá mostrar un reporte de las ventas realizadas en la empresa.

HISTORIA DE USUARIO	
Número: 9	Nombre: Reporte de Ventas
Iteración asignada: 4	Prioridad: Alta
Descripción: El sistema debe proveer un reporte donde se puedan visualizar las ventas realizadas en un periodo determinado.	

Tabla 3.9 Historia de usuario 9

Fuente: Elaboración propia

H.U. Reporte de inventarios: En esta área se podrá mostrar un reporte de las operaciones realizadas como ser entradas y salidas de productos.

HISTORIA DE USUARIO	
Número: 10	Nombre: Reporte de Inventarios
Iteración asignada: 4	Prioridad: Alta
Descripción: El sistema debe proveer un reporte donde se puedan visualizar las entradas y salidas de productos en un periodo determinado.	

Tabla 3.10 Historia de usuario 10

Fuente: Elaboración propia

3.2.2 PLANIFICACIÓN DE ITERACIONES

Partiendo de las historias de usuarios anteriores se realiza una planificación en 4 iteraciones basándose en el tiempo y procurando agrupar la funcionalidad relacionada en la misma iteración.

Iteraciones	Historia de Usuario	Duración	Fecha Inicio	Fecha Fin
Primera	Productos Categoría Clientes	3 semanas	01/03/2021	19/03/2021
Segunda	Ventas Usuarios	3 semanas	22/03/2021	09/04/2021
Tercera	Proveedores Compras	3 semanas	12/04/2021	30/04/2021
Cuarta	Inventarios Reporte de ventas Reporte de inventarios	4 semanas	03/05/2021	28/05/2021

Tabla 3.11 Historias de usuario de la Primera Iteración

Fuente: Elaboración propia

➤ Primera Iteración

Nro.	HISTORIA DE USUARIO
1	Productos
2	Categoría
3	Clientes

Tabla 3.12 Historias de usuario de la Primera Iteración

Fuente: Elaboración propia

➤ **Segunda Iteración**

Nro.	HISTORIA DE USUARIO
4	Ventas
8	Usuarios

Tabla 3.13 Historias de usuario de la Segunda Iteración

Fuente: Elaboración propia

➤ **Tercera Iteración**

Nro.	HISTORIA DE USUARIO
5	Proveedores
6	Compras

Tabla 3.14 Historias de usuario de la Tercera Iteración

Fuente: Elaboración propia

➤ **Cuarta Iteración**

Nro.	HISTORIA DE USUARIO
7	Inventarios
9	Reporte de ventas
10	Reporte de Inventarios

Tabla 3.15 Historias de usuario de la Cuarta Iteración

Fuente: Elaboración propia

3.2.3 PLANIFICACION DE TAREAS

En esta área se hace uso de Tarjetas de Tareas, las cuales tienen el objetivo de definir de forma concreta las tareas a realizar para implementar una historia de usuario. Se hará un análisis más profundo en las historias de usuario. Sus campos más relevantes son:

- ✓ **Tipo de tarea:** Existen varios tipos de tarea entre los cuales están el de desarrollo, corrección, mejora u otro.

- ✓ **Puntos estimados:** Son las semanas de desarrollo.

- ✓ **Fecha de Inicio y Fin:** Las fechas en la que se planea iniciar y terminar las actividades de la tarea.

De aquí en adelante se hará referencia al término “A B M” que significa Altas, Bajas y Modificaciones, son el conjunto de funciones básicas que todo módulo debería tener.

3.2.3.1 PRIMERA ITERACIÓN

En la primera iteración se implementarán los Objetos Base del sistema, estos objetos son los que serán consultados por otras funciones, procesos e incluso por otros objetos a lo largo del desarrollo del sistema, es por eso que su implementación es primordial para el desarrollo y pruebas de otros módulos. Las historias de Usuario de esta iteración son:

- ✓ 1: Productos

- ✓ 2: Clientes

En este caso, se tiene que uno de los objetos base es Productos; esta tabla será consultada por los módulos de compra, venta y muchas otras tablas.

TAREA 1	
Nombre de tarea: Crear A B M de productos	Número de Historia: 1
Tipo de tarea: Desarrollo	
Fecha Inicio: 01/03/2021	Fecha Fin: 05/03/2021
Descripción: Crear funciones para crear, modificar, eliminar y listar el objeto Producto, sus datos básicos son: Código de producto, nombre, precio de entrada, precio de salida, Cantidad y mínima en inventario.	

Tabla 3.16 Tarjeta de Tarea 1 de Historia de usuario 1

Fuente: Elaboración propia

TAREA 2	
Nombre de tarea: Crear A B M de	Número de Historia: 2
Tipo de tarea: Desarrollo	
Fecha Inicio: 08/03/2021	Fecha Fin: 12/03/2021
Descripción: Crear funciones para crear, modificar, eliminar y listar el objeto Categoría, su dato básico es: Nombre de categoría.	

Tabla 3.17 Tarjeta de Tarea 2 de Historia de usuario 2

Fuente: Elaboración propia

TAREA 3	
Nombre de tarea: Crear A B M de Clientes	Número de Historia: 3
Tipo de tarea: Desarrollo	
Fecha Inicio: 15/03/2021	Fecha Fin: 19/03/2021
Descripción: Crear funciones para crear, modificar, eliminar y listar el objeto Cliente, sus datos básicos son: Nombre completo, Número de identificación, email y teléfono.	

Tabla 3.18 Tarjeta de Tarea 3 de Historia de usuario 3

Fuente: Elaboración propia

3.2.3.2 SEGUNDA ITERACIÓN

En esta iteración se pretende implementar las siguientes historias de usuarios:

- ✓ 4: Ventas

- ✓ 8: Usuarios

TAREA 4	
Nombre de tarea: Registro de Ventas	Número de Historia: 4
Tipo de tarea: Desarrollo	
Fecha Inicio: 22/03/2021	Fecha Fin: 30/03/2021
Descripción: En este módulo tendremos primeramente un buscador de productos por nombre o por código para realizar la venta correspondiente del producto solicitado, mostrará la lista de venta además del resumen donde se podrá seleccionar el cliente que realiza la compra, el descuento que se puede realizar y el efectivo monto total a pagar, finalizar la venta e imprimir la factura.	

Tabla 3.19 Tarjeta de Tarea 4 de Historia de usuario 4

Fuente: Elaboración propia

TAREA 5	
Nombre de tarea: Crear A B M de Usuarios	Número de Historia: 8
Tipo de tarea: Desarrollo	
Fecha Inicio: 31/03/2021	Fecha Fin: 09/04/2021
Descripción: Crear las funciones para registrar nuevo usuario, modificar y listar. A la vez, se validarán los datos ingresados, tiene datos como: Nombre, apellido, código de usuario y contraseña.	

Tabla 3.20 Tarjeta de Tarea 5 de Historia de usuario 8

Fuente: Elaboración propia

3.2.3.3 TERCERA ITERACIÓN

En esta iteración se pretende implementar las siguientes historias de usuarios:

- ✓ 5: Proveedores, se muestra en la tabla 3.21
- ✓ 6: Compras, se muestra en la tabla 3.22

TAREA 6	
Nombre de tarea: Crear A B M de Proveedores	Número de Historia: 5
Tipo de tarea: Desarrollo y diseño de Formulario de registro de Proveedores	
Fecha Inicio: 12/04/2021	Fecha Fin: 16/04/2021
Descripción: Crear un formulario que permita introducir el nombre del proveedor, país del proveedor, dirección y teléfono.	

Tabla 3.21 Tarjeta de Tarea 6 de Historia de usuario 5

Fuente: Elaboración propia

TAREA 7	
Nombre de tarea: Crear Compras	Número de Historia: 6
Tipo de tarea: Desarrollar formulario para ingreso de productos a inventarios	
Fecha Inicio: 19/04/2021	Fecha Fin: 30/04/2021
Descripción: Crear las funciones y procedimientos necesarios para ingresar productos al inventario. También, estas funciones deben validar si el ítem ingresado es nuevo o no, en el caso de ser nuevo el sistema debe crear un nuevo registro en Registro de Producto, caso contrario solo debe añadir la cantidad nueva a la cantidad que ya había anteriormente. Esto se sabrá mediante un buscador de productos por nombre o código para abastecer productos.	

Tabla 3.22 Tarjeta de Tarea 7 de Historia de usuario 6

Fuente: Elaboración propia

3.2.3.4 CUARTA ITERACIÓN

En esta iteración se pretende implementar las siguientes historias de usuario:

- ✓ 7: Inventarios, tabla 3.23
- ✓ 9: Reporte de ventas, tabla 3.24
- ✓ 10: Reporte de inventario, tabla 3.25

TAREA 8	
Nombre de tarea: Módulo de inventarios	Número de Historia: 7
Tipo de tarea: Desarrollo	
Fecha Inicio: 03/05/2021	Fecha Fin: 12/05/2021
Descripción: Se mostrará un listado de todos los productos existentes con una descripción necesaria de cada producto.	

Tabla 3.23 Tarjeta de Tarea 8 de Historia de usuario 7

Fuente: Elaboración propia

TAREA 9	
Nombre de tarea: Reporte de Ventas	Número de Historia: 9
Tipo de tarea: Desarrollo	
Fecha Inicio: 10/05/2021	Fecha Fin: 14/05/2021
Descripción: Desarrollar un módulo que muestre las ventas realizadas en un rango de fecha, que será variable.	

Tabla 3.24 Tarjeta de Tarea 9 de Historia de usuario 9

Fuente: Elaboración propia

TAREA 10	
Nombre de tarea: Reporte de Inventarios	Número de Historia: 10
Tipo de tarea: Desarrollo	
Fecha Inicio: 17/05/2021	Fecha Fin: 28/05/2021
Descripción: Desarrollar un módulo que muestre las entradas y salidas de productos realizadas en un rango de fecha que será variable.	

Tabla 3.25 Tarjeta de Tarea 10 de Historia de usuario 10

Fuente: Elaboración propia

3.2.3.5 PLAN DE ENTREGAS

N°	Semanas/ Fases	Febrero				Marzo				Abril				Mayo			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Planificación																
2	Diseño																
3	Desarrollo																
4	Pruebas																

Tabla 3.26 Plan de entregas

Fuente: Elaboración propia

Este plan de entregas que se muestra en la tabla 3.26 se presentó al cliente.

3.3 FASE II - DISEÑO

En esta fase se presenta diseños simples como sugiere la metodología XP. Para lograr una mejor comprensión de la funcionalidad del sistema y el diseño debe ser sencillo. El diseño de los modelos está basado en el lenguaje UML.

3.3.1 IDENTIFICACIÓN DE LOS ROLES Y TAREAS DE LOS USUARIOS

El sistema será utilizado por el personal de la importadora, quienes tendrán acceso a una cuenta personal, a continuación, se detalla un listado de roles que desempeñan los usuarios.

ROL	DESCRIPCIÓN
 ADMINISTRADOR	Persona que se encarga de hacer pedidos para la importación y recepcionar los productos que ingresan a la importadora.
 ENCARGADO DE ALMACEN	Es la persona encargada de verificar la existencia o no de los productos pedidos, quien este encargado del Stock en general.
 CAJERO	Persona encargada de registrar las ventas a los diferentes clientes, encargándose de la facturación.

Tabla 3.27 Roles de usuarios del sistema

Fuente: Elaboración propia

3.3.2 MODELO DE CASO DE USO DEL NEGOCIO.

Para una mejor descripción de la actividad realizada que tiene el sistema actual de la importadora, veamos las ilustraciones 3.1 y 3.2 a continuación.

✓ Adición de Producto

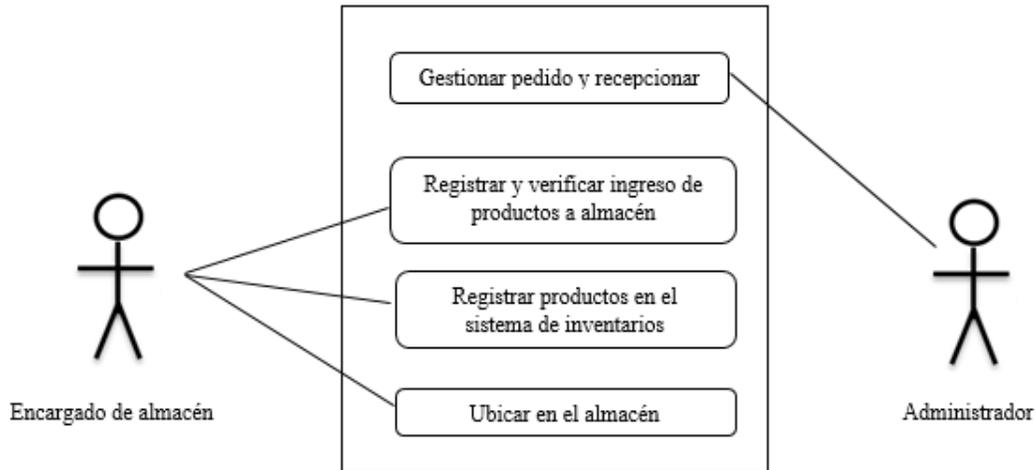


Ilustración 3.1 Caso de uso negocio - Adición de producto

Fuente: Elaboración propia

✓ Venta de producto

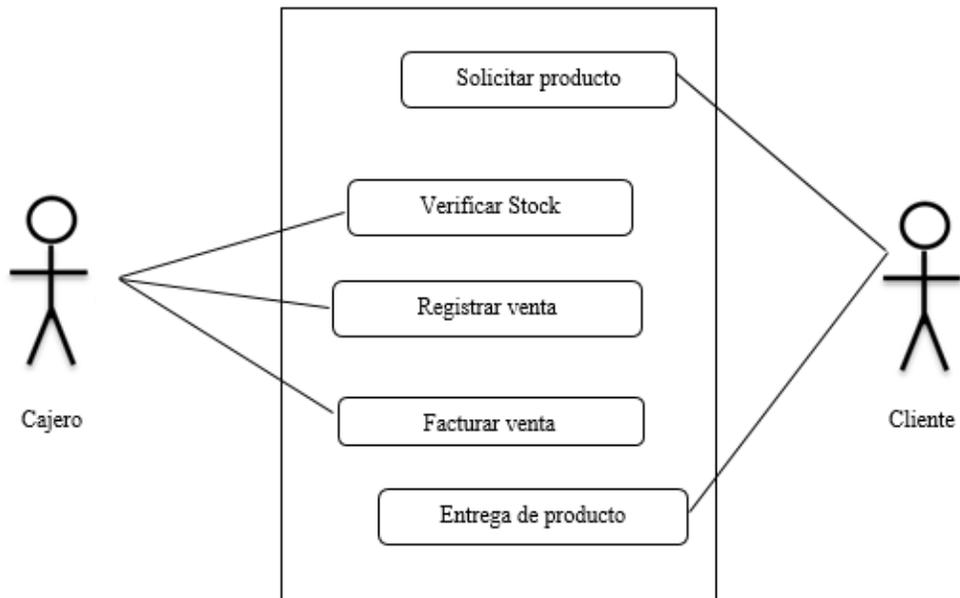


Ilustración 3.2 Caso de uso negocio – Venta de producto

Fuente: Elaboración propia

3.3.3 MODELO DE ESTRUCTURA

El modelo de estructura, como su nombre indica, tiene el objetivo de representar la estructura de sistema a nivel de las tablas que lo componen, la interrelación entre ellas y el comportamiento a nivel funcional entre estas.

3.3.3.1 DIAGRAMAS DE RELACIÓN DE ENTIDADES

Los ERD son representaciones visuales de bases de datos que muestran cómo se relacionan los elementos dentro de una base de datos. Un ERD está compuesto por dos tipos de objetos: entidades y relaciones. Las relaciones tienen terminaciones de línea especiales llamadas cardinalidades que describen cómo dos elementos de la base de datos interactúan entre sí, como se muestra en la ilustración 3.3

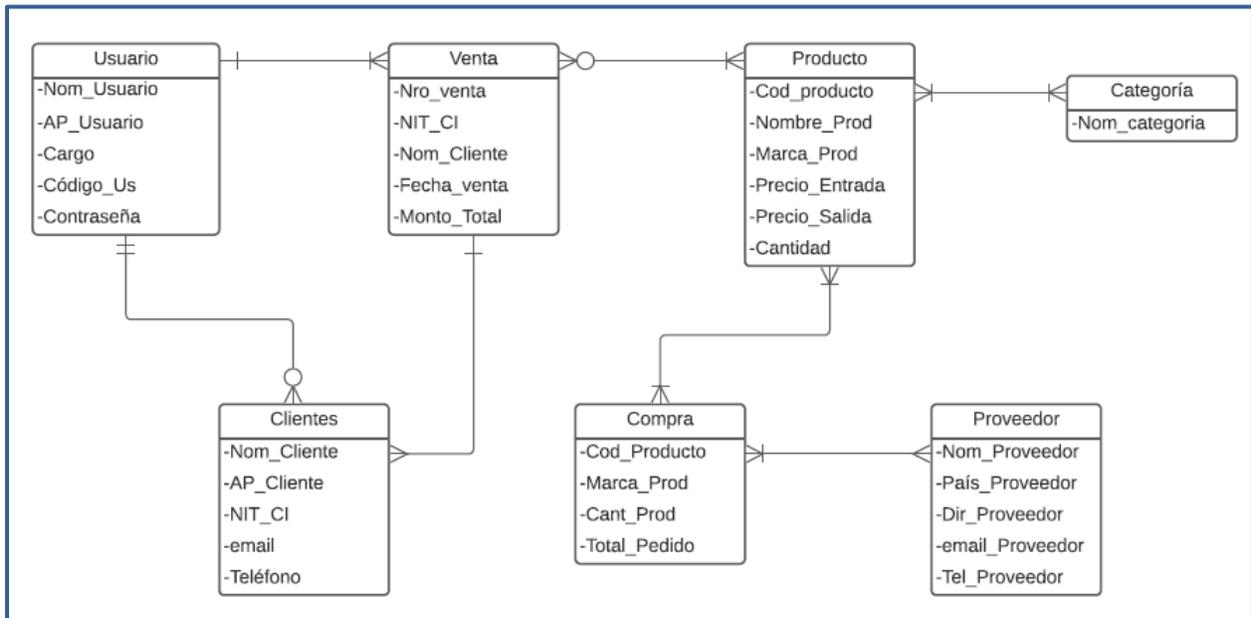


Ilustración 3.3 Diagrama Entidad Relación

Fuente: Elaboración propia

3.3.3.2 DIAGRAMA DE CLASES

Un diagrama de clases es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones o métodos, y las relaciones entre objetos del sistema, como podemos ver en la ilustración 3.4 que se puede apreciar a continuación.

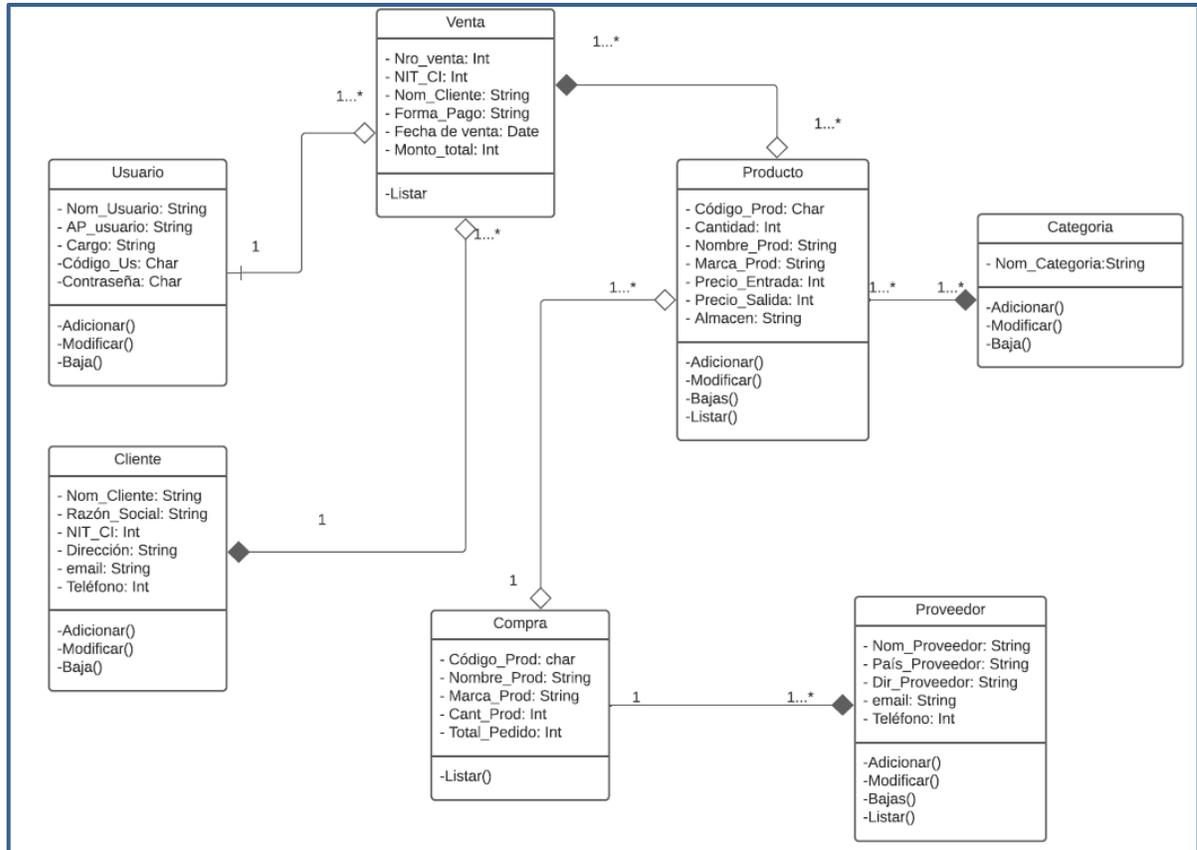


Ilustración 3.4 Diagrama de Clases

Fuente: Elaboración propia

3.4 FASE III – CODIFICACIÓN

En esta fase se realiza la programación del sistema acorde al plan de entrega realizadas anteriormente, teniendo en cuenta todas las características que se presentaron y diseñaron.

A continuación, se mostrará el proceso de desarrollo y capturas de pantalla de las historias de usuario más importantes.

La ilustración 3.5 muestra la pantalla de acceso al sistema, donde el usuario coloca su Código de usuario y contraseña, estos datos son validados y se ingresa a la pantalla principal.

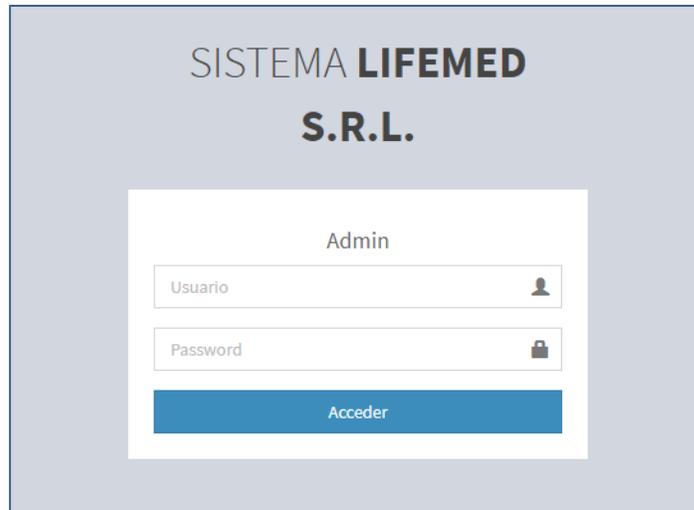


Ilustración 3.5 Pantalla de Ingreso al sistema

Fuente: Elaboración propia

La ilustración 3.6 muestra la pantalla para ingresar un nuevo producto a la base de datos, donde se debe llenar los datos, posteriormente estos productos pueden ser enlistados en un reporte de inventarios, el Costo se refiere al precio_entrada, Precio de venta es el Precio_salida.

Nuevo Artículo

Tipo

Imagen Ningún archivo seleccionado

Nombre*

Categoría

Descripción

Costo(Bs)*

Precio de Venta (Bs)*

Mínima en inventario:

Inventario inicial:

Fecha de caducidad:

Ilustración 3.6 Pantalla de registrar Producto

Fuente: Elaboración propia

La ilustración 3.7 muestra la pantalla para ingresar un nuevo Cliente, donde se ingresan sus datos para que se registre en la base de datos.

Nuevo Cliente

Nombre Cliente *

Razon Social *

NIT/CI *

Direccion

Email

Telefono

* Campos obligatorios

Ilustración 3.7 Pantalla de registrar a un Cliente

Fuente: Elaboración propia

La ilustración 3.8 muestra la pantalla para registrar venta de productos, donde se ingresan los datos del cliente y de acuerdo a la cantidad y precio de cada producto, se realiza el cálculo del monto total que debe cancelar el cliente. Todas las ventas del día, semana o mes se pueden imprimir, generando un reporte de ventas.

The screenshot shows a web interface for recording a sale. On the left, a table titled 'Lista de venta' contains one row with the following data:

Cantidad	Producto	Precio Venta	Precio Total	Total a Cancelar	
5	TAZA BLANCA DE 6OZ	Bs 6.00	Bs 30.00	Bs 30.00	✖ Quitar

On the right, the 'Resumen' section includes:

- Buttons for 'Principal' and 'Extra'.
- 'Almacen' field with value 'CASA MATRIZ' and a '+ Cliente Nuevo' button.
- 'NIT/CI' field with a text input.
- 'Razón Social' field with a text input containing 'RAZÓN SOCIAL'.
- 'Pago' dropdown set to 'Pagado' and 'Entrega' dropdown set to 'Entregado'.
- 'Forma de pago' dropdown set to 'Efectivo'.
- 'Efectivo Bs' input field with value '30.00'.
- A summary table:

Subtotal	Bs 26.10
IVA (13%)	Bs 3.90
Total	Bs 30.00
- Buttons at the bottom: '✖ Cancelar' and '\$ \$ Finalizar Venta'.

Ilustración 3.8 Pantalla de registrar venta

Fuente: Elaboración propia

3.5 FASE IV – PRUEBAS

Esta fase de la Programación Extrema es una de las más importantes, ya que nos permite verificar junto al cliente si se pudieron atender los requerimientos específicos en las historias de usuario. También sirve como retroalimentación para ver que historias de usuario fueron implementadas en versiones anteriores y necesitan ser modificadas, mejoradas o simplemente descartadas.

Se realizará la prueba de módulos más relevantes del sistema, estas pruebas se realizan junto a los usuarios, para que ellos manipulen el sistema y den su visto bueno, como se muestra en

las siguientes tablas: 3.28 Prueba de Usuario, 3.29 Prueba de Producto, 3.30 Prueba de Compra y 3.31 prueba de venta.

PRUEBA 1: DE HISTORIA DE USUARIO 8
Nombre: Verificación de ingreso de Usuarios al sistema
Descripción: <ul style="list-style-type: none"> ✓ Controlar que el nombre de usuario sea único. ✓ Validar los campos numéricos para evitar errores en el sistema. ✓ Mostrar un mensaje de confirmación una vez que se haya hecho el registro correctamente.
Entrada/Ejecución: Los usuarios ingresan con su código y contraseña, se autentica los datos.
Resultado Esperado: El sistema ingresa al menú principal después de validar los datos
Evaluación de la Prueba: Aceptada

Tabla 3.28 Verificación de Usuarios para ingreso al sistema

Fuente: Elaboración propia

PRUEBA 2: DE HISTORIA DE USUARIO 1
Nombre: Verificación de ingreso de Productos
Descripción: <ul style="list-style-type: none"> ✓ Verificar que los precios de los productos sean correctos. ✓ Controlar que los campos obligatorios sean debidamente llenados, mostrando alertas. ✓ Validar los campos numéricos para evitar errores en el sistema.
Entrada/Ejecución: Los usuarios ingresan un producto nuevo, se guarda en la base de datos.
Resultado Esperado: El sistema genera reporte de los productos en inventario.
Evaluación de la Prueba: Aceptada

Tabla 3.29 Verificación de ingreso de productos al sistema

Fuente: Elaboración propia

PRUEBA 3: DE HISTORIA DE USUARIO 6
Nombre: Verificación de compra
Descripción: <ul style="list-style-type: none"> ✓ Verificar el ingreso de los artículos a inventario de cada producto comprado. ✓ Controlar que los campos obligatorios sean debidamente llenados, mostrando alertas. ✓ Validar los campos numéricos para evitar errores en el sistema.
Entrada/Ejecución: el sistema registra la compra de productos.
Resultado Esperado: El sistema genera reporte de los productos en inventario.
Evaluación de la Prueba: Aceptada

Tabla 3.30 Verificación de registro de compra

Fuente: Elaboración propia

PRUEBA 4: DE HISTORIA DE USUARIO 4
Nombre: Verificación de venta
Descripción: <ul style="list-style-type: none"> ✓ Controlar que la venta realizada funcione correctamente mostrando los datos registrados. ✓ Controlar que los campos obligatorios sean debidamente llenados, mostrando alertas. ✓ Validar los campos numéricos para evitar errores en el sistema. ✓ Mostrar un mensaje de confirmación una vez que se haya el registro correctamente.
Entrada/Ejecución: el sistema registra la venta de productos.
Resultado Esperado: El sistema registra la venta realizada, genera reporte de ventas.
Evaluación de la Prueba: Aceptada

Tabla 3.31 Verificación de proceso de venta

Fuente: Elaboración propia

4. MÉTRICAS DE CALIDAD

4.1 INTRODUCCIÓN

En el presente capítulo se realizará y se comprobará la calidad al igual que la seguridad del software haciendo uso de punto función en uno de los estándares para evaluar el proyecto.

4.2 CALIDAD DE SOFTWARE

El desarrollar un software de calidad es el objeto de todo desarrollador, por tanto, se le dedica muchos esfuerzos, pero se debe tomar en cuenta que todo software debe cumplir y/o superar las expectativas del cliente. El presente proyecto utilizara la técnica del Calidad de Software ISO 9126.

El objetivo principal de esta técnica es alcanzar la calidad necesaria para satisfacer las necesidades del cliente. La calidad según esta norma ISO 9126 puede ser medida de acuerdo a los factores:

- ✓ Usabilidad
- ✓ Funcionalidad
- ✓ Confiabilidad
- ✓ Mantenibilidad
- ✓ Portabilidad

4.2.1 USABILIDAD

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios. (Medina, 2017)

- Aprendizaje
- Comprensión
- Operatividad

- Atracción
- Conformidad de uso

PREGUNTAS	RESPUESTAS		PORCENTAJE
	SI (0-10)	NO (0-10)	
1.- ¿El acceso al sistema es complicado?	0	10	100%
2.- ¿Las respuestas del sistema son comprensibles?	9	1	90%
3.- ¿Es fácil de aprender?	10	0	100%
4.- ¿Son comprensibles y satisfactorios los datos que se muestran en el sistema?	9	1	90%
5.- ¿El sistema es de interfaz amigable?	10	0	100%
6.- ¿El sistema satisface los requerimientos de la importadora?	10	0	100%
7.- ¿Es aplicable el sistema en el control de sus ventas?	9	1	90%
PROMEDIO TOTAL			95.71 %

Tabla 4.1 Encuesta de Usabilidad del Sistema

Fuente: Elaboración propia

De acuerdo a los datos obtenidos en la tabla 4.1 de encuesta de usabilidad, indica que el usuario del sistema se encuentra conforme, se concluye que el sistema tiene:

USABILIDAD = 95.71%.

4.2.2 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 sub características que cubre: (Medina, 2017)

- Adecuación
- Exactitud
- Interoperabilidad
- Seguridad
- Conformidad de la funcionalidad

Para calcular el Punto Función se utilizará la siguiente fórmula:

$$PF = \text{Cuenta total} * (X + \text{Min}(Y) * \sum Fi)$$

Donde:

PF: Medida de la funcionalidad.

X: Confiabilidad del proyecto, varía entre el 1% a 100%

Min (y): Error mínimo aceptable al de la complejidad, el margen de error es igual a 0.01.

Fi: Son los valores de ajuste de la complejidad, donde i=1 a i=14.

En la siguiente tabla 4.2 se calcula el punto función, los cuales miden el software desde una perspectiva del usuario, dejando de lado los detalles de codificación.

PARAMETROS	CANTIDAD	FACTOR DE PONDERACIÓN	TOTAL
Adecuación	20	4	80
Exactitud	21	4	84
Interoperabilidad	0	3	0
Seguridad	18	10	180

Conformidad de la funcionalidad	18	3	54
CUENTA TOTAL			398

Tabla 4.2 Conteo de parámetros de PF del sistema

Fuente: Elaboración propia

IMPORTANCIA	0	1	2	3	4	5
ESCALAS	Sin	Incremental	Moderado	Medio	Significativo	Esencial
1.- ¿El sistema requiere respaldo de seguridad?						X
2.- ¿El sistema será utilizado en el entorno de trabajo?						X
3.- ¿La entrada de datos es interactiva?						X
4.- ¿Existe una interfaz amigable con el Usuario?					X	
5.- ¿Se actualiza la BD en forma interactiva?				X		
6.- ¿Son complejas las entradas, salidas o las peticiones?				X		
7.- ¿Es complejo el procesamiento interno?					X	
8.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones, en diferentes empresas?		X				
9.- ¿La aplicación, será fácilmente utilizada por los usuarios?					X	
10.- ¿Se ha diseñado el código para ser reutilizado?						X
11.- ¿Están incluidos en el diseño la conversión y la instalación?				X		
12.- ¿Se ha diseñado el sistema para facilitar los cambios y para ser fácilmente utilizados por el usuario?						X
13.- ¿Será ejecutado el sistema en un entorno existente y fuertemente utilizado?						X
14.- ¿Es crítico el rendimiento?					X	
TOTAL, ΣFi	56					

Tabla 4.3 Ajuste de complejidad PF

Fuente: Elaboración propia

La Sumatoria de escalas de la tabla 4.3 es de 56, con lo cual se calcula el Punto función.

$$PF = \text{Cuenta Total} * (X + \text{Min}(Y) * \Sigma Fi)$$

$$PF=398* (0.65+0.01 *56)$$

Punto Función Obtenido: $PF= 481.6$

Para el ajuste se utiliza la ecuación para hallar el punto función ideal al 100% de los factores que sería 70:

Punto Función Ideal: $PF = 398 *(0.65+ 0.01 * 70)$

$$PF=537.3$$

PF ideal = 537.3

Calculando del % de funcionalidad real:

PF real = PF obtenida / PF ideal

FUNCIONALIDAD = $481.6/537.3*100$

$$**FUNCIONALIDAD = 90%**$$

4.2.3 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 la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida. (Medina, 2017)

- Madurez
- Recuperabilidad
- Tolerancia a errores
- Conformidad de Confiabilidad

La función a continuación muestra el nivel de confiabilidad del sistema:

$$F(t) = (Funcionalidad) * e(-\lambda t)$$

Se ve el trabajo hasta que se observa un fallo en un instante t, la función es la siguiente:

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de no hallar una falla: $P(T > t) = 1 - F(t)$

Donde:

Funcionalidad = 0,90

$\lambda = 0.14$ (1 error cada 7 ejecuciones)

Tomemos un tiempo t de 12 meses

Ahora hallando la confiabilidad del sistema:

$$F(12) = (0,90) * e^{(-0,14*12)}$$

$$F(12) = 0,168$$

Remplazando en la fórmula de no hallar una falla se tiene:

$$P(T > t) = 1 - F(t)$$

$$P(T > t) = 1 - 0,168$$

$$P(T > t) = 0,83$$

Con este resultado podemos decir que la probabilidad que el sistema no presente fallas es de 0,83 y que presente fallas es del 0,17.

CONFIABILIDAD = 83%

Con este resultado se concluye que la aplicación tiene un grado de confiabilidad del 83% durante los próximos 12 meses.

4.2.4 EFICIENCIA

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas. (Medina, 2017)

➤ Comportamiento de tiempos

- Utilización de recursos
- Conformidad de eficiencia

NÚMERO	FACTOR DE AJUSTE	PORCENTAJE OBTENIDO
1	Es de respuesta rápida al utilizar sus funciones	92
2	Tiene rendimiento de acuerdo a los factores que utiliza	90
3	El tiempo de respuesta a sus consultas es adecuado	93
TOTAL		92

Tabla 4.4 Factores de Eficiencia

Fuente: Elaboración propia

EFICIENCIA = 92%

El sistema obtuvo una eficiencia del 92%, como se muestra en la tabla 4.4

4.2.5 CAPACIDAD DE MANTENIMIENTO

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software. (Medina, 2017)

La Mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software.

$$IMS = \frac{Mt - (Fc + Fa + FE)}{Mt}$$

Dónde:

Mt: Número de módulos total de la versión actual (tenemos 8)

Fc: Número de módulos de la versión actual que se cambiaron. (sólo 1)

Fa: Número de módulos de la versión actual que se añadieron. (ninguno)

FE: Número de módulos de la versión anterior que se eliminaron en la versión actual.

(ninguno)

$$IMS=8-(1+0+0)/8=0.88$$

Este resultado establece un **88%**, lo que se interpreta que no requiere de mantenimiento inmediatamente.

4.2.6 PORTABILIDAD

Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra. (Medina, 2017)

- Adaptabilidad
- Facilidad de instalación
- Coexistencia
- Reemplazabilidad
- Conformidad de portabilidad

En cuanto al software, el sistema fue desarrollado en PHP y de la base de datos MySQL, se ejecuta en todos los servidores web, ya que no se necesita realizar una instalación del sistema.

En cuanto a la plataforma el sistema de información está desarrollado para sistemas operativos de la familia Microsoft Windows. También se comprobó que, en los distintos navegadores más usados, se le da una calificación del 100% de portabilidad. El resultado del 100% indica que el desenvolvimiento del sistema es correcto en los distintos navegadores.

$$**PORTABILIDAD = 100%**$$

5. EVALUACIÓN DE COSTO Y BENEFICIO

5.1 INTRODUCCIÓN

La técnica de Análisis de Costo y Beneficio, tiene como objetivo fundamental proporcionar una medida de la rentabilidad de un proyecto, mediante la comparación de los costos previstos con los beneficios esperados en la realización del mismo. Esta técnica se debe utilizar al comparar proyectos para la toma de decisiones.

5.2 COSTO DEL SISTEMA

El costo del sistema desarrollado se hará través de COCOMO II. El costo del sistema se planteará en tres partes: Desarrollo del software, Elaboración del Proyecto y Total del Software.

$$E = a (KLDC)^b * FAE, \text{ en persona-mes}$$

$$Tdev = c(E)^d, \text{ en meses}$$

$$P = E/Tdev, \text{ en personas}$$

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

KLDC, es la cantidad de líneas de código, en miles.

FAE, Es un multiplicador que depende de 15 atributos.

5.2.1 COSTO DEL SOFTWARE DESARROLLADO

Para la determinación del costo del software desarrollado, se hará uso del modelo constructivo de costos COCOMO II.

$$PF=481.60 \text{ (Obtenido)}$$

Para la conversión de los puntos de función a KLDC se convierte los puntos función a miles de líneas de código, teniendo en cuenta lo que se muestra en la siguiente tabla 5.1:

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
ANSI BASIC	5	6464
JAVA	6	53
PL/I	4	80
ANSI COBOL 74	3	107
VISUAL BASIC	7.00	46
ASP	9.00	36
PHP	11.00	29
VISUAL C++	9.50	34

Tabla 5.1 Conversión de puntos de función

Fuente: Pressman 2005

Así pues, tras saber que son 29 LDC por cada PF, el resultado de los KDLC será el siguiente:

$$KLDC = \frac{(PF * \text{Lineas de codigo por cada PF})}{1000}$$

$$KLDC = \frac{(481.6 * 29)}{1000}$$

$$KLDC = 13.96$$

Por tanto, existen 14 líneas de código distribuidas para el proyecto.

CONDUCTORES DE COSTE	VALORACIÓN					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
Fiabilidad requerida del software	0,75	0,88		1,15	1,40	-
Tamaño de la base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1,00	1,15	1,30	1,65
Restricciones del tiempo de ejecución	-	-	1,00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1,00	1,07	1,15	-
Capacidad del analista	1,46	1,19	1,00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	-
Experiencia en S.O. utilizado	1,21	1,10	1,00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1,00	0,95	-	-
Prácticas de programación modernas	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1,00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,23	1,08	1,00	1,04	1,10	-

Tabla 5.2 Conductores de Coste

Fuente: Calculo de costos de producción de software, (Castillo, 2017)

Realizamos la multiplicación de la valoración obtenida de los conductores de coste de la tabla anterior 5.2

$$FAE=1,15*1*0,85*1,11*1*1*1,07*0,86*0,82*0,70*1*0,95*1*0,91*1,08$$

$$FAE= 0,53$$

A continuación, haremos el cálculo del esfuerzo necesario para la programación del sistema.

La ecuación que nos ayudara a hallar el esfuerzo, para esta formula utilizaremos los datos de la tabla 5.3 que mostramos a continuación:

PROYECTO SOFTWARE	A	b	c	d
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.2	2.5	0.32

Tabla 5.3 Modelo COCOMO

Fuente: Pressman 2005

$$E = a (KLDC)^b * FAE, \text{ en persona-mes}$$

Como nuestro proyecto es del tipo Semi Acoplado utilizaremos $a = 3.0$ y $b = 1.12$,

Reemplazando estos valores en la ecuación, tenemos:

$$E = 3.0 * (13.96)^{1.12} * 0.53 = 30.45$$

Calculando $Tdev$ con $c = 2.5$ y $d = 0.35$ tenemos:

$$Tdev = c(E)^d$$

$$Tdev = 2.5 * (30.45)^{0.35}$$

$$Tdev = 8.26 \text{ (Serían 8 meses aproximadamente)}$$

Calculando el promedio de personas:

$$P = E/Tdev, \text{ en personas}$$

$$P = 30.45/8.26$$

$$P = 3.68, \text{ (que serían 4 personas)}$$

Según este resultado será necesario un equipo de 4 personas trabajando alrededor de 9 meses.

Ahora pues sabemos que el salario promedio de un programador oscila entre 500 USD y 800 USD, promediando tomaremos 650 USD para calcular el costo del software.

Costo del software desarrollado = Número de programadores*Salario de un programador

Costo del software desarrollado por persona = $4 * 650$ USD

Costo del software desarrollado por persona = 2.600 USD

Como el desarrollo de software se lo estima en 8 meses tendremos:

Costo el desarrollo = Costo del software desarrollado*Número de meses

Costo el desarrollo = $2.600 * 8$

Costo el desarrollo = 20.800 USD

5.2.2 COSTO DE IMPLEMENTACIÓN DEL SISTEMA

Las herramientas de software son PHP y MySQL, son de uso libre y las mismas que son utilizadas por muchas empresas por ser software libre.

5.2.3 COSTO DE ELABORACIÓN DEL SISTEMA

El costo de elaboración de proyecto hace referencia a los gastos que se realizaron a lo largo de las diferentes fases de XP, los mismos se presentan en la siguiente tabla 5.4:

DETALLE	IMPORTE En USD
Análisis y diseño del proyecto	200
Material de escritorio	50
Internet	80
Otros	20
TOTAL	350

Tabla 5.4 Costos de elaboración del proyecto

Fuente: Elaboración propia

5.2.4 COSTO TOTAL DEL SISTEMA

El costo total del sistema se muestra de forma detallada en la siguiente tabla 5.5:

DETALLE	IMPORTE En USD
Costo el desarrollo	20.800
Costo de elaboración del sistema	350
TOTAL	21.150

Tabla 5.5 Costos total del sistema

Fuente: Elaboración propia

Por lo tanto, el costo total del proyecto es igual 21.150 USD su equivalencia en bolivianos es 147.415,50 Bs.

5.3 BENEFICIO

Para evaluar el beneficio se calculará con el método del Valor Actual Neto (VAN) y la Tasa Interna de Retorno (TIR).

5.3.1 CÁLCULO DEL VALOR ACTUAL NETO (VAN)

El VAN es un procedimiento que permite calcular el valor presente de un determinado número de flujos de futuros ingresos y egresos que tendrá el proyecto. La metodología consiste en descontar al momento actual, es decir actualizar mediante una tasa todos los flujos de caja futuros del proyecto, a este valor se le resta la inversión inicial, así el valor obtenido es el valor actual neto del proyecto.

$$VAN = \Sigma [Ganancia / (1+k)^n] - I_0$$

Dónde:

VAN: Valor Actual Neto

Ganancias: ingreso del flujo anual

I₀: es el valor del desembolso inicial de la inversión

k: Tasa de descuento o tasa de interés al préstamo

n: Es el número de periodos considerados

Los valores de ganancia esperados para el presente proyecto se calculan para 4 años, para este caso en particular utilizaremos una tasa de descuento del 10%, ya que es la tasa actual de interés de préstamo en las entidades financieras. Para calcular el VAN se tiene lo siguiente:

$$\text{Inversión} = 147.204,00 \text{ Bs}$$

$$TD = 10\%$$

Lo valores de ganancia esperados se detallan en la siguiente tabla 5.6:

TIEMPO	1er. Año	2do. Año	3er. Año	4to. Año	5to. Año
Flujo De Caja Neto	47000	48000	49000	50000	51000

Tabla 5.6 Flujo de caja proyectado a 5 años

Fuente: Elaboración propia

$$VAN = [47000/(1+0.10)^1 + 48000/(1+0.10)^2 + 49000/(1+0.10)^3 + 50000/(1+0.10)^4 + 51000/(1+0.10)^5] - 147204$$

$$VAN = [42727,27 + 39669,42 + 36814,42 + 34150,67 + 31666,99] - 147204$$

$$VAN = 185028,77 - 147204$$

$$VAN = 37824,77$$

Por lo tanto, la ganancia que se espera recibir del sistema al cabo de 5 años es de 37824,77 Bs.

Para ver si un proyecto es rentable utilizamos la siguiente tabla 5.7

Valor de VAN	Interpretación
$VAN > 0$	El proyecto es rentable
$VAN = 0$	El proyecto es rentable porque ya está incorporada ganancia de la tasa de interés
$VAN < 0$	El proyecto no es rentable

Tabla 5.7 Interpretación del VAN

Fuente: Elaboración propia

Como el resultado que obtuvimos es de $VAN = 37824,77$ Bs, podemos afirmar que nuestro proyecto es rentable.

5.4 LA TASA INTERNA DE RETORNO (TIR)

El TIR es una tasa de descuento TD de un proyecto de inversión para que sea rentable. Cuando el VAN toma un valor igual a 0, k pasa a llamarse TIR. En términos generales, las inversiones más interesantes son aquellas que proporcionan mayor TIR.

Si TIR es inferior a la tasa de descuento de la empresa, la inversión debería ser desestimada.

Si TIR es superior a la tasa de descuento de la empresa la inversión es factible.

$$TIR = -I_0 + Q_1/(1+k) + Q_2/(1+k)^2 + \dots + Q_n/(1+k)^n$$

Dónde:

TIR: tasa interna de retorno

I_0 : es el valor del desembolso inicial de la inversión

n: número de periodos considerados

k: tasa de interés de ahorro

Entonces para hallar el TIR se necesita la inversión de 147204 Bs. Para hallar el TIR se hace uso de la fórmula del VAN, solo hace que el valor de VAN sea igual a 0, a continuación, en la tabla 5.8, se halla la tasa de descuento:

I_0	-147204
1er Año	47000
2do. Año	48000
3er Año	49000
4to. Año	50000
5to. Año	51000
TIR =	19%

Tabla 5.8 Cálculo de TIR en hoja de Excel

Fuente: Elaboración propia

Lo cual indica que el proyecto está en condiciones de retornar 19% de la inversión cada año, además si la tasa fuera mayor de 19% el proyecto empezaría a no ser rentable. Pues el Beneficio Neto Actual empezaría a ser menor que la inversión y si la tasa fuera menor al 10% (como en el caso del VAN donde la tasa es de 10%), a menor tasa, el proyecto es más rentable, pues el BNA es cada vez mayor que la inversión. También se puede mencionar que los beneficios calculados en dinero tienen otros como ser el ahorro del tiempo en el proceso de realizar publicaciones.

TIR a 5 años= 19%

Lo cual nos indica que el proyecto sea siempre rentable la tasa de descuento TD no debe ser mayor a 19%.

5.5 COSTO / BENEFICIO

Para hallar el costo/ beneficio de un proyecto aplicamos la siguiente ecuación: C/B

Reemplazando en la ecuación con los valores hallados anteriormente, tenemos:

$$\begin{aligned} C/B &= \frac{147204}{37824,77} \\ C/B &= 3.89 \end{aligned}$$

Por tanto, por cada boliviano invertido la empresa tiene una ganancia de 3.89 Bs

6. SEGURIDAD DE SISTEMA

El área de la informática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta y especialmente, la información contenida o circulante es la seguridad informática o seguridad de tecnologías de la información. Una vez registrados los productos en el sistema, almacenamos información de mucha importancia para la toma de decisiones por las cuales se deben considerar estrategias de control y políticas de seguridad.

6.1 SEGURIDAD DE SISTEMA OPERATIVO

A nivel de usuario se hará el uso de sesiones:

- ✓ Las sesiones en PHP son herramientas muy usadas en el desarrollo de web permitiendo recordar datos del usuario este logueado en un sitio para que no tenga que ingresar a su password nuevamente cuando vuelve a entrar.
- ✓ El soporte de sesiones en PHP consiste en una manera de guardar ciertos datos a través de diferentes accesos web. Esto permite crear aplicaciones más personalizadas y mejorar las características del sitio web.
- ✓ El servidor o la maquina asignada como tal debe contar con el apoyo de alimentación continua de energía frente a un corte de energía eléctrica. Es decir, un UPS

6.2 SEGURIDAD DE BASE DE DATOS

Se usó como base de datos MySQL. En cuanto a la forma de resguardo se realizó los siguientes puntos:

- ✓ Cuando una acción del usuario en el sistema requiere o solicita algunos registros de la base de datos, existe una conexión segura para esta acción.

✓ Para la seguridad de datos del sistema se tienen registrado 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.

6.3 SEGURIDAD SOFTWARE DE 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 el ingreso al sistema a los usuarios por medio de usuario y contraseña que son otorgados por el administrador del sistema. Se realiza el registro del usuario que modifica la información de la base de datos, para esto se registra en cada tabla el identificador del usuario que modifica la información.

7. CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

Una vez finalizado el desarrollo e implementación del Sistema para la gestión de ventas e inventario para la empresa Importadora Soluciones Medicas Lifemed S.R.L., se ha logrado alcanzar el objetivo principal planteado, bajo los requerimientos de la empresa. Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Se logró diseñar una interfaz amigable y de fácil manipulación para el usuario.
- Se logró mejorar el tiempo empleado en la atención de ventas de productos a los clientes, ya que se realiza este proceso de forma más eficiente y con el mismo se evitará errores en cálculos.
- Se logró mejorar el registro de los productos, se tiene las características detalladas de los mismos y así con estos registros mejorar la atención al cliente final.
- Se facilitó el proceso de compras de los productos en función de los productos vendidos, esto evita que la empresa compre productos que no se venden y se llene el stock.
- Se tiene control de la disponibilidad de productos en existencia.
- Se logró disminuir los tiempos en la generación de reportes tanto de ventas e inventarios, para así tener un mejor control del movimiento de la empresa.
- Se tiene un registro con los datos necesarios de los proveedores con los que mantiene contacto la empresa.
- Se logró disminuir los tiempos en la generación de reportes de ventas e inventarios, así se puede tener un mejor control del movimiento de la empresa.

De esta forma, se alcanzó el objetivo general de lograr la informatización de los procesos de compra, venta e inventario, de manera que la información ahora se encuentra a disposición del usuario para hacer el control adecuado a dichos procesos. Esto se logró mediante la ejecución de las fases propuestas por la metodología XP.

7.2 RECOMENDACIONES

A partir del presente trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Se recomienda para trabajos futuros un sistema de compra, venta e inventarios orientado a aplicaciones Android, para que pueda tener ventas en internet.
- Como ahora ya se esta viendo la facturación en línea se recomienda que se pueda agregar un modulo que sea compatible con el sistema de IN, ya que se trabajara directamente en su plataforma, para que contabilidad no haga el llenado de los libros de venta.

FUENTES DE INFORMACIÓN

Bibliografía

- Aliaga, J. C. (2001). *CONTROL DE VENTAS E INVENTARIOS*. La Paz, Bolivia.
- Alquiles, G., & Ferreras, M. (1995). *LA EDUCACION TECNOLOGICA*. MINISTERIO DE CULTURA Y EDUCACION.
- Arana, J. (2014). “Desarrollo e implementación de un sistema de gestión de ventas de Ecuador: Universidad Estatal Península de Santa Elena Facultad de Sistemas y.
- Beck, K. (1999). *Programación eXtrema explicada: Aceptando el cambio*.
- Bustamante D., R. J. (2014). *Metodología Actual Metodología XP*. Barinas, Venezuela .
- Calle, W. (2018). *SISTEMA EN PLATAFORMA MIXTA PARA EL CONTROL VENTAS E INVENTARIOS CON CÓDIGO QR*”. La Paz, Bolivia.
- Echeverry, D. (2007). *Metodología Agil de Desarrollo de Software* .
- Globaleinvoicing.com*. (06 de Febrero de 2019). Obtenido de Bolivia implementa un nuevo Sistema de Facturación Electrónica a partir de 2019:
<https://globaleinvoicing.com/es/noticias/bolivia-implementa-un-nuevo-sistema-de-facturacion-electronica-partir-de-2019>
- Gutierrez, G. (2015). *SISTEMA DE CONTROL DE VENTAS E INVENTARIOS PARA ALMACENES*. La Paz.
- Larman. (1999). *UML y Patrones Graig Larman* .
- Letelier, T., & Patricio, S. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*”. Carmen, Carmen: Penádes.
- LIFEMED S.R.L. (febrero a Agosto 2021) información Rosmery Sea Quino.

Medina, R. (24 de Julio de 2017). *Rocio Medina Prieto*. Obtenido de BlogSpot:

<http://unidad4rociomp.blogspot.com/2017/07/46.html>

Pressman, R. (2005). *Ingeniería de Software, Un enfoque Práctico*. Madrid, España.

Pressman, R. (2010). *Ingeniería del Software, Un Enfoque Práctico*. Madrid, España.

Robles G., F. J. (2002). *Programación Extrema y Software Libre*. Madrid, España .

Rodríguez, C. (12 de 8 de 2015). *Marketing Digital*. Obtenido de Blog del Máster en

Marketing Directo y Digital de la UPF Barcelona School of Management:

<https://marketingdigital.bsm.upf.edu/e-commerce-comercio-electronico/>

Suehring S., C. T. (2009). *Php 6 MySQL 6 Bible*. Indiana, Estados Unidos : Wiley

Publishing.

Thomson, L. (2003). *Desarrollo Web con PHP y MySQL*.