

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



**PROYECTO DE GRADO**

**“SISTEMA DE CONTROL DE COMPRA, VENTA E INVENTARIOS”**  
**CASO: EMPRESA PROTEC**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: MONICA SARCO MENDOZA**  
**TUTOR METODOLÓGICO: LIC. MIGUEL TOLEDO PAZ**  
**ASESOR: LIC. JAVIER REYES PACHECO**

**LA PAZ – BOLIVIA**  
**2017**



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



**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 DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## **DEDICATORIA**

---

A Dios, por guiarme cada pasó que doy, por la perseverancia, esfuerzo y capacidad que puso en mí, por mostrarme día a día su inmenso amor y misericordia.

A mis padres Sixto y María, por su constante apoyo en todo momento, por la paciencia y confianza.

A mis hermanos Jorge, Rudy, Ronald, Luis y Yesica, por su apoyo intelectual y moral.

A todas aquellas personas que fueron y son una pieza fundamental en mi vida.

## **AGRADECIMIENTOS**

Agradezco a Dios por darme la capacidad, la perseverancia y las fuerzas en todo el recorrido de la carrera y para poder concluir este proyecto, por el amor incondicional que me brinda cada día, por las oportunidades que me da para ser una mejor persona.

A mi familia, por el apoyo incondicional que me brindan y por la confianza que pusieron en mí, por haberme inculcado valores.

Al Docente tutor Lic. Miguel Toledo Paz, quien con su profesionalismo y experiencia acompañó en el desarrollo del presente proyecto, por todo su apoyo, paciencia, guía y consejos, los cuales dieron empuje al desarrollo y conclusión de este proyecto.

Al Lic. Javier Reyes Pacheco, por su tiempo, consejos, guía, experiencia y comprensión, los cuales fueron indispensables en el desarrollo y conclusión de este proyecto.

A la Lic. Bhylenia Yhasmyna Ríos Miranda, quien con su gran calidad de persona, profesionalismo y experiencia acompañó en el desarrollo del presente proyecto, gracias por la paciencia, por el apoyo que me brindo en todo momento.

Finalmente, a todos mis amigos y amigas que me han dado ánimos para seguir adelante.

## **RESUMEN**

Hoy en día los sistemas de información se han convertido en la herramienta imprescindible de toda empresa, ya sea grande o mediana, necesita para desarrollarse y crecer. La gran cantidad de información que manejan las entidades crea la necesidad de automatizar no solamente la información, sino también los procesos de negocio de las compañías.

La Empresa Protec, va creciendo constantemente y consigo la cantidad de información que maneja en sus distintas áreas organizacionales. Protec requiere tener el control diario de compra, venta y por defecto requieren tener el control de los estados de inventarios.

El presente proyecto tiene como objetivo implementar un sistema para optimizar el control eficiente de ventas, compras e inventario de la empresa descrita anteriormente, de forma que los gerentes hagan un control adecuado sobre estos procesos.

El proyecto fue desarrollado utilizando la metodología ágil XP (programación extrema) en sus distintas fases como son: planificación, diseño, desarrollo y pruebas. El diseño de los procesos e interfaz fue realizado bajo la metodología de modelado Webml que cuenta con diversos esquemas para la representación gráfica de estos procesos.

La calidad del sistema se la realizó bajo el estándar ISO 9126 que evalúa aspectos como usabilidad, funcionalidad, confiabilidad, mantenibilidad y portabilidad, proporcionando una evaluación tras la implementación del sistema.

Por último, los objetivos del proyecto han sido alcanzados satisfactoriamente de manera que se implementó un sistema que permita tener un control productivo a través de las compras, ventas e inventarios.

## **ABSTRACT**

Today information systems have become the essential tool of any company, whether large or medium, needs to develop and grow. The large amount of information that entities manage creates the need to automate not only the information, but also the business processes of the companies.

The Company Protec, is growing constantly and I get the amount of information that manages in its different organizational areas. Protec requires having the daily control of purchase, sale and default require to have control of the inventory statements.

The present project aims to implement a system to optimize the efficient control of sales, purchasing and inventory of the company described above, so that managers make adequate control over these processes.

The project was developed using agile XP methodology (extreme programming) in its different phases such as: planning, design, development and testing. The design of the processes and interface was done under the modeling methodology Webml that has several schemes for the graphic representation of these processes.

The quality of the system was performed under the ISO 9126 standard that evaluates aspects such as usability, functionality, reliability, maintainability and portability, providing an evaluation after the implementation of the system.

Finally, the objectives of the project have been satisfactorily achieved in a way that implemented a system that allows a productive control through purchases, sales and inventories.

# ÍNDICE

<b>CAPÍTULO I .....</b>	
Marco Introductorio .....	1
1.1      Introducción .....	1
1.2      Antecedentes .....	2
1.2.1      Antecedentes institucionales .....	2
1.2.2      Trabajos 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 específicos .....	5
1.5      Justificación .....	5
1.5.1      Justificación social .....	5
1.5.2      Justificación Técnica .....	6
1.5.3      Justificación Económica .....	6
1.6      Alcances y límites .....	6
1.6.1      Delimitación Temática.....	6
1.6.2      Delimitación Espacial .....	7
1.6.3      Delimitación Temporal .....	7
1.7      Aportes .....	7
1.7.1      Práctico .....	7
1.7.2      Teórico .....	7
1.8      Metodología .....	8
1.8.1      Metodología de Investigación .....	8
1.8.2      Metodología de Desarrollo de Software Ágil XP .....	8
<b>CAPÍTULO II .....</b>	
<b>MARCO TEÓRICO .....</b>	<b>9</b>
2.1      Introducción .....	9
2.2      Ingeniería de Software .....	9

2.3	Metodologías de desarrollo .....	9
2.3.1	Metodologías Ágiles.....	10
2.3.2	Modelo – Vista - Controlador .....	10
2.3.2.1	MVC en aplicaciones web.....	11
2.4	Extreme Programming (Programación Extrema) .....	11
2.4.1	Objetivos de XP .....	11
2.4.2	Características de XP .....	12
2.4.3	Valores de XP .....	12
2.4.4	Proceso de desarrollo de XP .....	13
2.4.5	Principios de XP .....	13
2.4.6	Prácticas básicas de XP .....	14
2.4.7	Roles de XP .....	15
2.4.8	Fases de la metodología XP .....	16
2.4.8.1	Fase I - Planificación del proyecto .....	16
2.4.8.2	Fase II –Diseño .....	19
2.4.8.3	Fase III - Codificación .....	20
2.4.8.4	Fase IV - Pruebas .....	21
2.5	Ingeniería Web .....	22
2.5.1	Modelado Webml .....	23
2.5.2	Objetivos principales de webml .....	23
2.5.3	Características .....	24
2.5.4	Diseño en WebML .....	25
2.5.4.1	Modelado de Estructural .....	25
2.5.4.2	Modelado de Hipertexto .....	25
2.5.4.3	Unidades de contenido .....	26
2.5.4.4	Diagramas de Composición .....	26
2.6	Análisis costo beneficio .....	27
2.6.1	Cocomo II .....	27
2.6.2	Modelos de estimación .....	28
2.6.2.1	Puntos función .....	28
2.7	Inventarios .....	30



2.7.1	Tipos de Inventarios .....	31
2.7.2	Técnicas de valuación para salida de almacenes .....	32
2.7.2.1	Método primero en entrar primero en salir (PEPS) .....	32
2.8	Compra .....	33
2.9	Venta .....	33
2.10	Herramientas de desarrollo de software .....	33
<b>CAPÍTULO III .....</b>		
<b>MARCO APLICATIVO .....</b>		<b>36</b>
3.1	Introducción .....	36
3.2	Fases de la metodología XP .....	37
3.2.1	Fase I - Planificación .....	37
3.2.1.1	Análisis de situación actual .....	38
3.2.1.2	Obtención de requerimientos .....	39
3.2.1.3	Historias de Usuario .....	40
3.2.1.4	Planificación de iteraciones .....	46
3.2.1.5	Tarjetas de Tarea .....	47
3.2.1.5.1	Primera iteración .....	48
3.2.1.5.2	Segunda iteración .....	49
3.2.1.5.3	Tercera iteración .....	50
3.2.1.5.4	Cuarta iteración .....	51
3.2.1.6	Plan de iteración .....	52
3.2.1.7	Plan de entregas .....	52
3.2.2	Fase II - Diseño .....	53
3.2.2.1	Modelo Estructural .....	53
3.2.2.1.1	Diagrama de clases .....	54
3.2.2.1.2	Diagrama de componentes .....	54
3.2.2.2	Tarjetas de CRC (clase-responsabilidad-colaboración) .....	60
3.2.3	Fase III - Codificación .....	61
3.2.3.1	Primera iteración .....	61
3.2.3.2	Segunda iteración .....	62
3.2.3.3	Tercera iteración .....	63

3.2.3.4	Cuarta iteración .....	64
3.2.4	Modelo Entidad/ Relación .....	65
3.2.5	Fase IV - Pruebas .....	66
3.2.5.1	Pruebas de aceptación .....	66
<b>CAPÍTULO IV .....</b>		
<b>CALIDAD Y SEGURIDAD .....</b>		<b>69</b>
4.1	Introducción .....	69
4.2	Calidad de Software .....	69
4.2.1	Técnica ISO 9126 .....	69
4.2.1.1	Usabilidad .....	69
4.2.1.2	Funcionalidad .....	71
4.2.1.3	Confiabilidad .....	74
4.2.1.4	Mantenibilidad .....	75
4.2.1.5	Portabilidad .....	76
4.2.1.6	Resultados .....	77
4.3	Seguridad de Software .....	77
4.3.1	Seguridad a nivel de usuario .....	77
4.3.2	Seguridad a nivel de Base de Datos .....	77
4.3.3	Seguridad a nivel de aplicación .....	77
<b>CAPÍTULO V .....</b>		
<b>ANÁLISIS COSTO BENEFICIO .....</b>		<b>79</b>
5.1	Introducción .....	79
5.2	Costo del Sistema .....	79
5.2.1	Cocomo II .....	79
5.2.2	Costo del Software Desarrollado .....	80
5.2.3	Costo de implementación del sistema .....	83
5.2.4	Costo de Elaboración del Sistema .....	83
5.2.5	Costo total del Software .....	83
5.3	Beneficio .....	83
5.3.1	Cálculo del Valor Actual Neto (VAN) .....	83
5.3.2	La Tasa Interna de Retorno TIR .....	85

5.4	Costo / Beneficio .....	86
<b>CAPÍTULO VI .....</b>		
	Conclusiones y Recomendaciones .....	87
6.1	Conclusiones .....	87
6.2	Recomendaciones .....	88
	BIBLIOGRAFÍA .....	89
	ANEXOS .....	92

## ÍNDICE DE FIGURAS

Figura 2.1: Ciclo de Vida del XP .....	16
Figura 2.2: Modelo de Estructura .....	25
Figura 2.3: Modelo de Hipertexto .....	26
Figura 2.4: Unidades de Contenido .....	26
Figura 2.5: Ejemplo uso de Unidad .....	27
Figura 3.1: Metodología XP reformulada .....	37
Figura 3.2: Diagrama de contexto del sistema .....	53
Figura 3.3: Diagrama de clases .....	54
Figura 3.4: Diagrama de componentes- ABM de producto.....	55
Figura 3.5: Modelo de Presentación- ABM de producto.....	55
Figura 3.6: Diagrama de componentes – ABM de Categoría .....	56
Figura 3.7: Modelo de Presentación – ABM de Categoría.....	56
Figura 3.8: Diagrama de componentes – Registro de Ventas .....	57
Figura 3.9: Modelo de Presentación – Registro de Ventas.....	57
Figura 3.10: Diagrama de componentes – Registro de Compra.....	58
Figura 3.11: Modelo de Presentación – Registro de Compra.....	58
Figura 3.12: Diagrama de componentes – Inventario.....	59
Figura 3.13: Modelo de Presentación – Inventario.....	59
Figura 3.14: Pantalla - ABM de Productos .....	62
Figura 3.15: Pantalla - ABM directorio de Clientes.....	62
Figura 3.16: Pantalla - Panel de Ventas.....	63
Figura 3.17: Pantalla – Resumen de Ventas .....	63
Figura 3.18: Pantalla – ABM de Proveedores .....	64
Figura 3.19: Pantalla – Panel de compras o Reabastecimiento .....	64
Figura 3.20: Pantalla – Inventario .....	64
Figura 3.21: Pantalla – Reporte de Inventario.....	65
Figura 3.22: Pantalla – Reporte de Venta.....	65
Figura 3.23: Modelo Entidad / Relación .....	65
Figura 5. 1: Calculo del TIR.....	86

## ÍNDICE DE TABLAS

Tabla 2.1: Modelo propuesto para una historia de usuario.....	17
Tabla 2.2: Modelo propuesto para una tarea de ingeniería.....	18
Tabla 2.3: Tarjeta CRC.....	20
Tabla 2.4: Modelo propuesta para una prueba de aceptación.....	22
Tabla 3.1: Historia de Usuario 1.....	41
Tabla 3.2: Historia de Usuario 2.....	42
Tabla 3.3: Historia de Usuario 3.....	42
Tabla 3.4: Historia de Usuario 4.....	42
Tabla 3.5: Historia de Usuario 5.....	43
Tabla 3.6: Historia de Usuario 6.....	43
Tabla 3.7: Historia de Usuario 7.....	44
Tabla 3.8: Historia de Usuario 8.....	44
Tabla 3.9: Historia de Usuario 9.....	45
Tabla 3.10: Historia de Usuario 10.....	45
Tabla 3.11: Resumen de Historia de Usuario .....	46
Tabla 3.12: Historia de Usuario de la Primera Iteración .....	46
Tabla 3.13: Historia de Usuario de la Segunda Iteración .....	47
Tabla 3.14: Historia de Usuario de la Tercera Iteración.....	47
Tabla 3.15: Historia de Usuario de la Cuarta Iteración .....	47
Tabla 3.16: Tarjeta de Tarea 1.1 de Historia de Usuario 1.....	48
Tabla 3.17: Tarjeta de Tarea 2.1 de Historia de Usuario 2.....	48
Tabla 3.18: Tarjeta de Tarea 3.1 de Historia de Usuario 3.....	49
Tabla 3.19: Tarjeta de Tarea 4.1 de Historia de Usuario 4.....	49
Tabla 3.20: Tarjeta de Tarea 5.1 de Historia de Usuario 8.....	50
Tabla 3.21: Tarjeta de Tarea 6.1 de Historia de Usuario 5.....	50
Tabla 3.22: Tarjeta de Tarea 7.1 de Historia de Usuario 6.....	51
Tabla 3.23: Tarjeta de Tarea 8.1 de Historia de Usuario 7.....	51
Tabla 3.24: Tarjeta de Tarea 9.1 de Historia de Usuario 9.....	51
Tabla 3.25: Tarjeta de Tarea 10.1 de Historia de Usuario 10.....	52
Tabla 3.26: Plan de Iteración.....	52
Tabla 3.27: Plan de Entrega.....	52
Tabla 3.28: Verificación de ingreso de usuario.....	66
Tabla 3.29: Verificación de registro de productos .....	67
Tabla 3.30: Verificación de módulo de ventas .....	68
Tabla 3.31: Verificación de módulo de compra o abastecimiento .....	68

Tabla 4.1: Encuesta de usabilidad del sistema .....	70
Tabla 4.2: Conteo de parámetros del PF.....	72
Tabla 4.3: Tabla de ajuste de complejidad .....	73
Tabla 4.4: Resultados .....	77
Tabla 5.1: Coeficiente a y c y los exponentes b y d .....	80
Tabla 5.2: Conversión de puntos de función siguiente.....	81
Tabla 5.3: Costos de elaboración del proyecto.....	83
Tabla 5.4: Costo total de productos de software .....	83
Tabla 5.5: Cantidad nominal por año .....	84
Tabla 5.6: Interpretación del VAN .....	85

## **MARCO INTRODUCTORIO**

### **1.1 INTRODUCCIÓN**

Hoy en día los sistemas de información se han convertido en la herramienta imprescindible de toda empresa, ya sea grande o mediana, necesita para desarrollarse y crecer. La gran cantidad de información que manejan las empresas crea la necesidad de automatizar no solamente la información, sino también los procesos de negocio de las compañías.

Las redes informáticas es uno de los medios más eficientes para mantener la información actualizada ya que pueden compartir información entre varias computadoras y usuarios simultáneamente y de esta forma disponer de información oportuna que ayudará al mejor funcionamiento de la empresa. En la actualidad las empresas públicas y privadas necesitan tener medios tecnológicos para mantener su información de manera óptima y oportuna para la toma de decisiones.

Si no se tiene el seguimiento de control de compras, venta e inventario puede causar la quiebra o pérdidas en la empresa o negocio, por la mala administración.

En la administración de una pequeña empresa si lo vemos más de cerca nos daremos cuenta que también se necesita de un sistema de control de compras, venta e inventario, como se dijo anteriormente existe mucho flujo de información, procesos de entrada y salida de datos que son muy importantes a la hora de tomar decisiones.

Mediante el presente proyecto se dio una solución favorable a los problemas de administración de información de las compras, ventas e inventarios, desarrollando un sistema, de manera que al momento de realizar la toma de decisiones, los gerentes propietarios cuenten con información clara, precisa, actualizada e instantánea para, de esta forma, satisfacer la demanda de sus clientes. Además, el sistema agiliza la obtención de reportes de ventas mensuales de la Empresa Protec.

Finalmente, se demostró que tener disponibilidad de información inmediata permite controlar de manera eficiente el ingreso y egreso del inventario, la cual es una de las

necesidades que justifica el desarrollo de una solución informática ya que además proporcionará mayor información actualizada del funcionamiento de la empresa, útil para la asistencia a los procesos administrativos, consultas, reportes, control, entre otros.

## **1.2 ANTECEDENTES**

### **1.2.1 ANTECEDENTES INSTITUCIONALES**

Protec es una empresa unipersonal, con autonomía administrativa y patrimonio independiente, constituida por el pleno inventario de sus activos. La misma que está ubicada en la Zona Central Calle Ingavi No. 684 esq. Alto de la Alianza.

La Empresa Protec, va creciendo constantemente y consigo la cantidad de información que maneja en sus distintas áreas organizacionales. Esta empresa tiene un variado stock de materiales que distribuye como empresa entre ellas están los tubos de oxígeno medicinal (reguladores med, humidificador, canuta nasal, etc.), extintores, tubos CO2, señalizaciones en acrílicos y otros materiales de seguridad industrial, ferretería en general.

Los procesos de compra y venta son realizados de forma manual la cual resulta insuficiente ante las necesidades de la empresa. La cantidad de información sobre las compras mensuales, las ventas diarias ha crecido tanto que dificulta llevar un buen control al respecto y la pérdida de información se presenta con frecuencia en estas situaciones. Es por esta forma de procesar la información, que la empresa tiene problemas y dificultades al momento de la toma de decisiones. Al no contar con información de las ventas totales mensuales, los productos más vendidos o los productos que ya se agotaron.

Al momento de hacer compras a sus proveedores, la verificación de la existencia de los productos implica demora y un arduo trabajo al identificarlos en extensas listas de detalle de la factura, ya que esto también se realizaba de forma manual.

En resumen, el control que hace la Empresa Protec 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.

### **1.2.2 TRABAJOS SIMILARES**

En gestiones pasadas ya se han realizado distintos sistemas de control de ventas, compras e inventarios, los cuales han sido desarrollados según las necesidades de empresas específicas.

Por ejemplo, de entre los muchos que existen en la Carrera de Informática de la Universidad Mayor de San Andrés, se pueden citar:

- Sistema web de control de compras, venta e inventarios caso LIBRERÍA DE LA ASOCIACIÓN CRISTIANA PAN DE VIDA de Raquel Condori Palomeque en el año 2015. Este proyecto se desarrolla un sistema que permite informatizar los procesos de compra, venta e inventarios de la LIBRERÍA PAN DE VIDA, haciendo uso de la metodología de Desarrollo Ágil XP y modelado con Diseño Conceptual de Aplicaciones Web (WebML).
- Sistema web de seguimiento de ventas y cobranzas para la agencia de viajes “Cosmos Travel and Services S.R.L.” de Luis Alfredo Colmena Vargas en el año 2015. En el proyecto, se diseña y desarrolla un sistema en plataforma web que permite hacer control y seguimiento de las ventas y cobranzas de los servicios ofertados por la agencia de viajes Cosmos Travel and Services S.R.L., haciendo uso de la metodología de Desarrollo Ágil XP y modelado con el Diseño Conceptual de Aplicaciones Web (WebML).
- Sistema de entradas y salidas e inventarios caso BOLITAL S.R.L. de Claudia Chiri Honorio en el año 2014. Este proyecto para su desarrollo se utilizó la metodología orientada a objetos RUP y haciendo uso del lenguaje modelo UML.
- Sistema web de compras, ventas e inventario para la empresa EDDYMAR de Deysi Vanessa Rojas Laguna en el año 2014. En este proyecto se diseña y desarrolla un sistema que permite hacer el control de las compras, ventas e inventario de la

empresa EDDYMAR, haciendo uso de la metodología de Desarrollo Ágil XP y modelado con el Diseño Conceptual de Aplicaciones Web (WebML).

- Sistema web de control de inventarios y cobranzas con alarma temprana para Forever Living Bolivia S.R.L. de Daniela Valverde Valero en el año 2013. En este proyecto se diseña y desarrolla un sistema en plataforma web para la empresa Forever Living Bolivia S.R.L., haciendo uso de la metodología de Desarrollo Ágil XP y modelado con el Diseño Conceptual de Aplicaciones Web (WebML).
- Sistema de información de compras e inventario SAMA de Choque Chambilla Raúl Francisco en el año 2009. Para el desarrollo de proyecto se optó por utilizar en el proceso de investigación el Método Científico y para el análisis y diseño para sistemas se utilizó la metodología de análisis estructurado.

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

#### **1.3.1 PROBLEMA PRINCIPAL**

El problema principal en la Empresa Protec 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 ésta forma de procesar la información, que la empresa tiene problema y dificultad al momento de la toma de decisiones.

¿Cómo se pueden mejorar el control de compra, venta e inventario de la Empresa Protec?

#### **1.3.2 PROBLEMAS SECUNDARIOS**

- 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.

- No dispone de información exacta y oportuna sobre los inventarios de los productos.
- No tener toda la información necesaria de todos los proveedores de la empresa.
- Inexistencia de reportes de ventas e inventarios por día, mes que sean rápidas y claras con una descripción de todo lo que se vendió, no se las realiza.

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

Implementar un sistema para optimizar el control eficiente de compras, ventas e inventario, para la Empresa Protec.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Elaborar un módulo de ventas.
- Realizar un módulo para registros de los artículos.
- 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 JUSTIFICACIÓN**

### **1.5.1 JUSTIFICACIÓN SOCIAL**

El sistema ayuda a la empresa a facilitar el acceso a la información de las compras, ventas e inventarios esto hace que la empresa tenga mayor competitividad en el mercado.

En cuanto al personal de la empresa 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 TÉCNICA**

Los componentes técnicos como 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.

### **1.5.3 JUSTIFICACIÓN ECONÓMICA**

El proyecto permitirá incrementar los beneficios económicos de la Empresa Protec con la mejora en el manejo eficiente de la información, un mejor control de su inventario y un mejor flujo de información a través del sistema, reduciendo de esta manera la pérdida de dinero en la compra de talonarios, papeles, etc..

El desarrollo del sistema no tendrá costo, ya que se emplearán herramientas de software libre y código abierto, aprovechando estos recursos al máximo para obtener un producto de calidad.

## **1.6 ALCANCES Y LIMITES**

### **1.6.1 DELIMITACIÓN TEMÁTICA**

El presente Proyecto de Grado, permite ejercer mayor control de los procesos de compras, ventas e inventarios.

Tiene los siguientes alcances:

- Módulo de administración de usuarios.
- Módulo de ventas.
- Módulo de compras.
- Módulo de inventarios.
- Módulo de proveedores.
- Módulo de productos.

### **1.6.2 DELIMITACIÓN ESPACIAL**

El sistema se implementara para la empresa Protec, que se encuentra en la Zona Central de la ciudad de La Paz Calle Ingavi No. 684 esq. Alto de la Alianza, se espera que tenga un alcance nacional.

### **1.6.3 DELIMITACIÓN TEMPORAL**

El periodo de tiempo a ser tomado en cuenta será a partir de la puesta en marcha del sistema.

## **1.7 APORTES**

### **1.7.1 PRÁCTICO**

El sistema ayudará a la empresa a realizar mayor control de las compras, ventas e inventarios, de igual manera los registros de los productos donde nos servirá para optimizar el tiempo y eficiencia con respecto al manejo de la información. Permitiendo centralizar la información.

El desarrollo del sistema permite cubrir las necesidades para organizar los procesos rutinarios, minimizar tiempos, un mejor control y seguimiento que se debe realizar, generando información que ayude la fácil y correcta toma de decisiones en la empresa.

### **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.

Así mismo el sistema contribuirá con la aplicación del modelado web: WebML que es la notación visual para el diseño de aplicaciones web. Provee especificaciones gráficas formales para un proceso de diseño completo que puede ser asistido por herramientas de diseño visuales.

## **1.8 METODOLOGÍA**

### **1.8.1 METODOLOGÍA DE INVESTIGACIÓN**

En el presente proyecto se usará el Método Científico para desarrollar y definir las tareas investigativas. El método científico está compuesto por las siguientes etapas:

- Observación
- Inducción, Hipótesis, Prueba de Hipótesis.
- Demostración o Refutación (antítesis) de la hipótesis.

### **1.8.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE ÁGIL XP**

Para el desarrollo de este proyecto de grado, en el modelado, diseño y desarrollo del sistema, se hará uso de una metodología de desarrollo de software ágil, en este caso la metodología XP (Extreme Programming, 2015) y para el modelado de datos se utilizará el Diseño Conceptual de Aplicaciones Web (WebML).



## **MARCO TEÓRICO**

### **2.1 INTRODUCCIÓN**

En este capítulo se describirá los principios y conceptos básicos para la implementación del proyecto, así como de las tecnologías usadas para el desarrollo del sistema, sin embargo no se puede ofrecer una teoría completa acerca de las metodologías, técnicas y herramientas que se utilizará para la elaboración del proyecto, sin embargo se presentara una base para la fácil comprensión de la misma.

### **2.2 INGENIERÍA DE SOFTWARE**

Según Pressman, la ingeniería del software es una disciplina o área de la informática o ciencia de la computación, que ofrece técnicas y métodos para desarrollar y mantener software de calidad.

Según Sommerville, para muchas personas el software son solo programas de computadora, sin embargo nos comenta que son todos aquellos documentos asociados a la configuración de datos que se necesita para hacer que estos programas operen de manera adecuada. Estos productos de software se desarrollan para algún cliente en particular o para un mercado en general. Para el diseño y desarrollo de proyectos de software se aplican metodologías, modelos y técnicas que permiten resolver los problemas. En los años 50 no existían metodologías de desarrollo, el desarrollo estaba a cargo de los propios programadores. De ahí la importancia de contar con analistas y diseñadores que permitieran un análisis adecuado de las necesidades que se deberían de implementar.

### **2.3 METODOLOGÍAS DE DESARROLLO**

En el entendido de que el desarrollo de software no es una actividad sencilla, el que existan diversas propuestas metodológicas para hacerlo, comprueban la afirmación. Por una parte figuran propuestas tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los productos, las herramientas y notaciones que se usarán; por otra, figuran aquellas que le dan mayor valor

al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo de software. Es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito (Laboratorio Nacional de Calidad del Software, 2009).

### **2.2.1 METODOLOGÍAS ÁGILES**

Las metodologías ágiles son adaptables en lugar de predictivas, es decir que para las metodologías ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio. Están orientadas a la gente y al proceso. El objetivo de los métodos “clásicos” es definir un proceso que funcionará bien independientemente de quien lo utilice. Las metodologías ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. (Gomez, Aranda , & Fabrega, 2010)

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Entre estas se hallan: XP, SCRUM, DSDM y Crystal

### **2.3.2 MODELO-VISTA-CONTROLADOR**

Este es un patrón de arquitectura de las aplicaciones software. Separa la lógica de negocio de la interfaz de usuario y facilita la evolución por separado de ambos aspectos, incrementando su eficacia debido a que permite reutilización y flexibilidad. Es utilizado en múltiples frameworks.

Bajo este patrón destacan un modelo, varias vistas y varios controladores. Las vistas y los controladores suelen estar estrechamente relacionados, ya que estos tratan los eventos que se producen en la interfaz gráfica, lo cual ofrece un panorama de flexibilidad al



desarrollador a momento de encarar sus actividades (Mestras, 2015).

#### **2.3.2.1 MVC EN APLICACIONES WEB**

- Para Vista: Control sobre la página HTML generada.
- Para el Controlador: Generación de código que obtiene datos dinámicamente y genera el contenido HTML.
- Para el Modelo en sí: La información es almacenada en una base de datos o en XML junto con las reglas de negocio que transforman esa información según las acciones de los usuarios.

#### **2.4 EXTREME PROGRAMMING (PROGRACIÓN EXTREMA)**

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 de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo (Joskowicz, 2011).

##### **2.4.1 OBJETIVOS DE XP**

Consisten en:

- Establecer las mejores prácticas de Ingeniería de Software en los desarrollos de proyectos.

- Mejorar la productividad de los proyectos.
- Garantizar la calidad del Software desarrollado, haciendo que este supere las expectativas del cliente.

#### 2.4.2 CARACTERÍSTICAS DE XP

Según Bustamante y Rodríguez de 2014 las características de XP son las siguientes:

- Se diferencia de las metodologías principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- Se aplica de manera dinámica durante el ciclo de vida del software.
- Es capaz de adaptarse a los cambios de requisitos.
- Los individuos e interacciones son más importantes que los procesos y herramientas.

#### 2.4.3 VALORES DE XP

XP se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito (Joskowicz, 2011).

- **Comunicación:** Muchos de los problemas que existen en proyectos de software se deben a problemas de comunicación entre las personas. La comunicación permanente es fundamental dado que la documentación es escasa, el diálogo cara a cara entre desarrolladores, gerentes y el cliente es el medio básico de comunicación.
- **Simplicidad:** XP apuesta a la sencillez en su máxima expresión, en el diseño, en el código, en los procesos y demás. La sencillez es esencial para que todos puedan entender el código, y se trata de mejorar mediante recodificaciones continuas.
- **Retroalimentación:** La retroalimentación debe funcionar en forma permanente. El cliente debe brindar retroalimentación de las funciones desarrolladas, de manera de poder tomar sus comentarios para la próxima iteración y para comprender sus necesidades.

- **Coraje:** Ante problemas serios en el diseño, o en cualquier otro aspecto, se debe tener el coraje suficiente como para encarar su solución, sin importar que tan difícil sea. Si es necesario cambiar completamente parte del código, hay que hacerlo, sin importar cuanto tiempo se ha invertido previamente en el mismo.

#### **2.4.4 PROCESO DE DESARROLLO 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 (Robles y Ferrer, 2002).

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 restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

#### **2.4.5 PRINCIPIOS DE XP**

Los desarrollos de software ágil basados en XP generalmente adoptan los principios del “Agile Manifesto”, iniciativa de 17 desarrolladores de software reunidos en Utah, Estados Unidos, para lograr un equilibrio en el que la documentación sea concreta, útil y no burocrática, pero reconociendo sus limitaciones en el actual mundo en permanente cambio (Joskowicz, 2011).

Los principios que fundamentan esta metodología se traducen en que:

- Se valora a los individuos y las interacciones sobre los procesos y las herramientas.
- Se valora a las aplicaciones que funcionan sobre la documentación exhaustiva.
- Se valora la colaboración del cliente sobre las negociaciones contractuales.
- Se valora la respuesta al cambio sobre el seguimiento de un plan.

#### **2.4.6 PRACTICAS BÁSICAS DE XP**

El proceso que recomiendan los autores de XP es el siguiente: Identificar el principal problema del proceso de desarrollo actual. Escoger la práctica que ayuda a resolver ese problema y aplicarla. Cuando ese haya dejado de ser un problema, escoger el siguiente. Se recomienda trabajar estas prácticas en par (Robles, 2002):

1. El juego de la planificación (the planning game)
2. Pequeñas entregas (small releases)
3. Metáfora (metaphor)
4. Diseño simple (simple design)
5. Pruebas (testing)
6. Refactorización (refactoring)
7. Programación por parejas (pair programming)
8. Propiedad colectiva (collective ownership)
9. Integración continua (continous integration)
10. 40 horas semanales (40-hour week)
11. Cliente en casa (on-site costumer)
12. Estándares de codificación (coding standards)

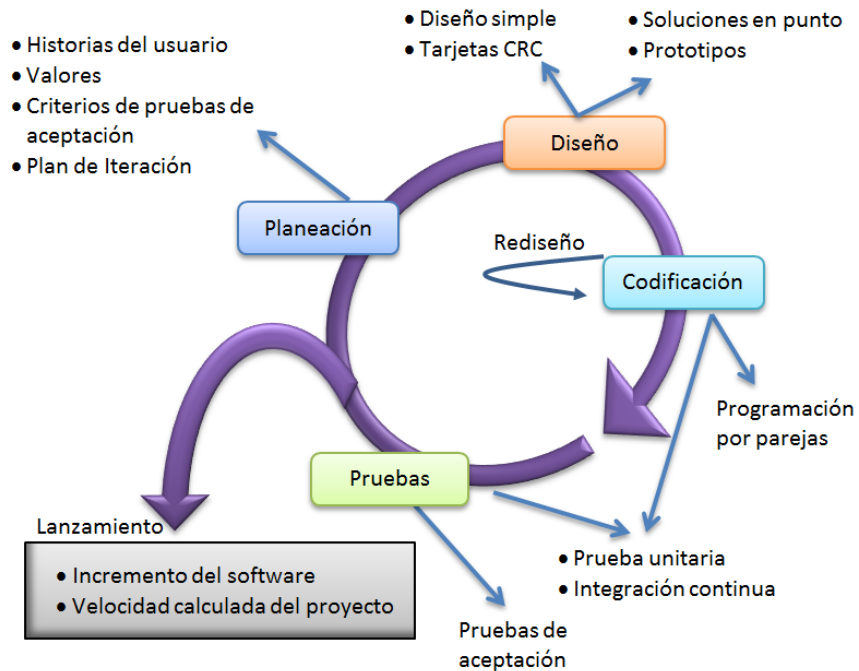
#### 2.4.7 ROLES DE XP

En este apartado se describen los roles de acuerdo con la propuesta original de Beck. Los roles originales fueron: Programador, Encargado de pruebas, Encargado de Seguimiento, Entrenador, Consultor y Gestor. (Loraine, 2012)

- **Programador:** es el encargado de escribir las pruebas unitarias y producir el código del sistema. Es responsable de las decisiones técnicas y de construir el sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- **Cliente:** escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración buscando aportar mayor valor al negocio.
- **Encargado de pruebas (Tester):** ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker):** proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
- **Entrenador (Coach):** es responsable del proceso global. Es el líder del equipo y quien toma las decisiones importantes.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- **Gestor (Big boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es coordinación.

## 2.4.8 FASES DE LA METODOLOGÍA DE XP

XP abarca fases que ocurren en el contexto de cuatro actividades del marco de trabajo, las mismas que se muestran en la figura 2.1 y posteriormente se hace la descripción fase por fase.



**Figura 2.1: Ciclo de vida de XP**  
Fuente: (Monroy, 2013)

### 2.4.8.1 FASE I – PLANIFICACIÓN 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 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 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. 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. El modelo de Historia de Usuario por Kent Beck es el siguiente:

HISTORIA DE USUARIO	
<b>NUMERO:</b> 1	<b>NOMBRE HISTORIA DE USUARIO</b>
<b>USUARIO:</b> Autor	<b>ITERACION ASIGNADA</b>
<b>PRIORIDAD EN NEGOCIO:</b> Alta/Media/Baja)	<b>PUNTOS ESTIMADOS</b>
<b>RIESGOS DE DESARROLLO:</b> (Alta/Media/Baja)	
<b>DESCRIPCION:</b> Breve descripción de lo que el sistema, constan de 3 ó 4 líneas	
<b>OBSERVACIONES:</b>	

**Tabla 2.1: Modelo propuesto para una historia de usuario**  
Fuente: (Beck, 1999)

Las historias de usuario tienen tres aspectos:

- **Tarjeta:** En ella se almacena suficiente información para identificar y detallar la historia.
- **Conversación:** Cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera).
- **Pruebas de aceptación:** Permite confirmar que la historia ha sido implementada.

**Tarjetas de Tareas:** En la siguiente tabla 2.2 se muestra como se distribuye la información de Task Card.

TAREA DE INGENIERÍA	
NUMERO DE TAREA	HISTORIA DE USUARIO(Nro. y Nombre)
Nonmbre Tarea	
Tipo de Tarea	
Tipo de Tarea:Desarrollo/Correccion/Mejora	
Fecha inicio	Fecha Fin
Programador Responsable	
Descripción	

**Tabla 2.2: Modelo propuesto para una tarea de ingeniería**

Fuente: Anaya y Plaza, s.a.

- **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. Este "Release plan", en inglés, es 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.
- **Iteraciones:** El proyecto se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" 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.
- **Velocidad del proyecto:** 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 Plan de Entregas.

- **Programación en pareja:** La metodología XP 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.

#### 2.4.8.2 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.

- **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 e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.
- **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.
- **Funcionalidad extra:** Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.
- **Refactorizar:** Es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado.

**Tarjetas C.R.C. (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 muestra cómo se distribuye estas informaciones.

NOMBRE DE LA CLASE	
RESPONSABILIDADES	COLABORADORES

**Tabla 2.3: Tarjeta CRC**  
Fuente: (Echeverry, Delgado, 2007)

### 2.4.8.3 FASE III – CODIFICACIÓN

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

- **Reuniones con el cliente:** Como ya se mencionó anteriormente, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.
- **Tests:** Se deben hacer pruebas unitarias al código comprobando que los resultados cumplan con las historias de usuario.
- **Desarrollo colectivo:** XP sugiere que todos los programadores estén implicados en todas las tareas, de manera que cualquiera pueda modificar o mejorar el código ajeno.
- **Estándares de codificación:** En esta fase deben seguirse estándares de codificación de manera que el código se mantenga ordenado, consistente y de manera que sea de fácil comprensión y escalabilidad.

#### 2.4.8.4 FASE IV – PRUEBAS

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos 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 a 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.

- Crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa.

XP enfatiza muchos aspectos relacionados con las pruebas, clasificándolas en diferentes tipos, funcionalidades específicas, indicando quien, cuando y como deben ser implementadas y ejecutadas (Echeverry y Delgado, 2007).

- Pruebas Unitarias: Estas pruebas se aplican a todos los métodos no triviales de todas las clases de proyecto con la condición que no se liberara ninguna clase que tenga asociada su correspondiente paquete de pruebas.
- Pruebas de Aceptación: También llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario. Las pruebas de aceptación son pruebas de caja negra, que representa un resultado esperado de determinada transacción con el sistema.

CASO DE PRUEBAS DE ACEPTACION	
Código	Historia de usuario(Nro. y Nombre)
Nombre	
Descripción	
Condiciones de ejecución	
Entrada/Pasos de ejecución	
Resultado Esperado	
Evaluación de la prueba	

Tabla 2.4: Modelo propuesta para una prueba de aceptación  
Fuente: Anaya y Plaza, s.a.

## 2.5 INGENIERÍA WEB

La Ingeniería Web es una versión adaptada del enfoque de la Ingeniería de Software que propone una estructura ágil, pero disciplinada, para construir sistemas y aplicaciones

basados en la web con calidad industrial (Pressman & Lowe, Web Engineering: A Practitioner's Approach, 2008). Este concepto implica la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente de aplicaciones de alta calidad accesibles por Internet, así como su operación y evolución (Jarquín, 2015)

Actualmente existen millones de sitios web que cubren diferentes necesidades de la vida, donde la de comunicación resulta ser la más importante. La aparición de los servicios web acrecentó el problema de la interoperabilidad entre el intercambio de datos y su procesamiento en contextos heterogéneos relativos a la navegación entre contenido hipertextual y multimedia en un entorno de desarrollo global y distribuido cada vez más flexible y donde se materializan constantemente nuevas oportunidades de negocio, situación que ameritaba su tratamiento por parte de una rama científica exclusiva, derivada de la Ingeniería de Software, cuyo objeto de estudio principal se denomina Aplicación Web.

### **2.5.1 MODELADO WEBML**

WebML es un lenguaje modelado de alto nivel para la especificación de aplicaciones web. En esta aproximación, se propone la especificación de la aplicación Web en base a cuatro perspectivas: modelo estructural, modelo del hipertexto, modelo de presentación y modelo de personalización. Define también un proceso interactivo, con las siguientes etapas: recolección de requisitos, diseño de hipertexto, diseño de presentación, diseño de usuarios y grupos, diseño de personalización. (Ceri y Piero, 2000).

WebML es un lenguaje de modelado gráfico utilizado para apoyar las actividades del diseño de sitios Web. Provee gráficos, formalismos, especificaciones y diseño de procesos apoyados por herramientas gráficas. Define varios tipos de diagramas: de estructura, composición y navegación. (Carmona, 2008).

### **2.5.2 OBJETIVOS PRINCIPALES DE WEBML**

Al proporcionar especificaciones gráficas mediante un proceso de diseño integral



desarrollado con herramientas de diseño visual, sus principales objetivos (Stefano Ceri, 2012) consisten en:

- Expresar la estructura de una aplicación: Soportar una colección de conceptos que posibiliten un diseño de alto nivel, proveyendo especificaciones gráficas para producir una descripción abstracta de la WebApp.
- Prestar múltiples vistas de un mismo contenido.
- Separar el contenido de la información de su composición en las páginas, la navegación y la presentación, que puede ser definido y desarrollado de forma independiente.
- Almacenar meta-información: Registrar información recopilada durante el proceso de diseño de un repositorio que pueda utilizarse durante el tiempo de vida útil de la solicitud de generación dinámica de la página web.
- Permitir la especificación de las operaciones de manejo de datos para actualizar el contenido del sitio o la interacción con servicios externos.
- Modelar usuarios y comunidades.

### **2.5.3 CARACTERÍSTICAS**

Sus principales características (Gonzales Alvarán, Reyes Gamboa, & Vásquez Echavarría, 2010) son:

- Combina técnicas de modelado ER con UML
- Se basa en la distribución de nodos en los niveles del hipertexto sobre las páginas del nivel de presentación.
- Dispone de una herramienta CASE que facilita la creación de páginas web en jsp WebRatio.
- Para el modelado estructural no se exige ninguna herramienta específica.
- Se enfoca en el diseño de la interfaz. Para esto provee una serie de estereotipos que pueden ser implementados usando XML.
- Presenta una extensión del módulo de navegación de UML para el manejo de proyectos de WebApps.

## 2.5.4 DISEÑO EN WEBML

El WEBML describe el alto nivel de un sitio Web en distintas dimensiones ortogonales: su contenido de datos modelo estructural, las páginas que lo componen modelo de composición, la topología de vínculos entre páginas modelo de navegación, el diseño y los requisitos gráficos para la renderización de la página modelo de presentación, así como las características de personalización para la entrega “uno a uno” del contenido modelo de personalización (Stefano Ceri, 2012), las cuales son interdependientes entre sí, tal como se describe (Gonzales Alvarán, Reyes Gamboa, & Vásquez Echavarría, 2010) a continuación:

### 2.5.4.1 MODELADO ESTRUCTURAL

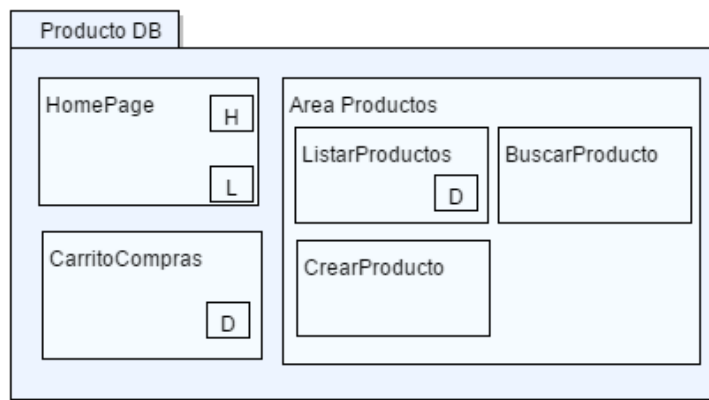
Cuando se trabaja con WebML el proceso de desarrollo comienza con la descripción conceptual del sistema, en la cual, utilizando herramientas CASE para modelado - como UML, DIA, Enterprise Architect, se representa la estructura estática del sistema mediante la definición de entidades o contenedores de datos y sus relaciones.



**Figura 2.2: Modelo de Estructura**  
Fuente: Elaboración Propia

### 2.5.4.2 MODELO HIPERTEXTO

En este modelo se describen los diferentes hipertextos que van a ser publicados en el sitio web. Cada uno de estos define una vista del sitio y su descripción se realiza mediante dos modelos: el de composición, que define las páginas que componen la estructura del hipertexto, así como el contenido de éstas; y el de navegación, que describe cómo se podrá navegar a través de ellas, especificando los vínculos o enlaces entre páginas y entre unidades de una misma página.

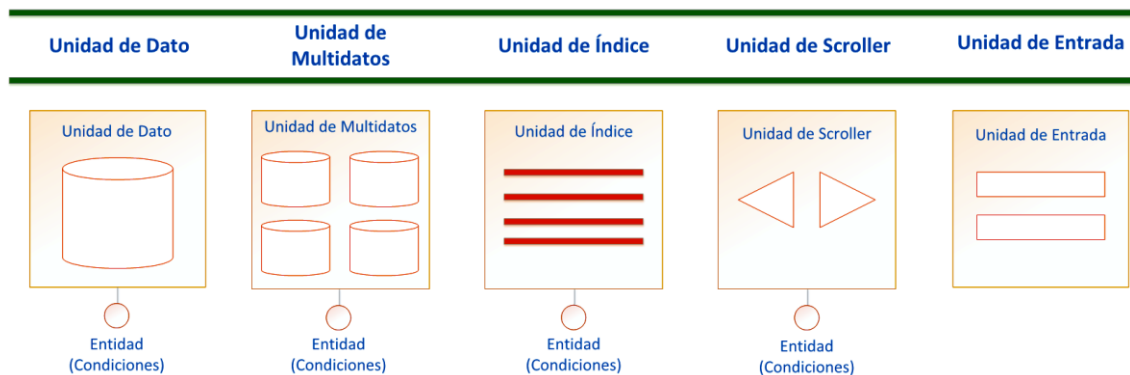


**Figura 2.3: Modelo de Hipertexto**

Fuente: Elaboración Propia

### 2.5.4.3 UNIDADES DE CONTENIDO

Las Unidades de contenido ayudan a modelar el funcionamiento o los procesos que incluyen las páginas del sitio web. WebML consta de 5 componentes de Unidades de Contenido, como se aprecia en el siguiente gráfico:



**Figura 2.4: Unidades de contenido**

Fuente: (Web Modeling Language, s.f.)

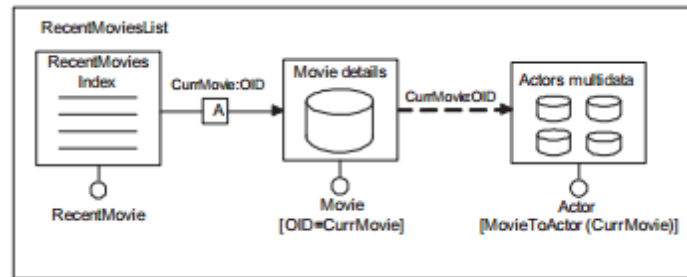
### 2.5.4.4 DIAGRAMAS DE COMPOSICIÓN

Se encarga de definir las funciones de los objetos que componen las páginas web, al mismo tiempo define el flujo de navegación entre páginas. Uno de los aportes más interesantes de WebML es que ofrece una herramienta CASE que permite aplicar las técnicas propuestas y conseguir los resultados sistemáticamente.



La utilización de estos componentes se analiza en el siguiente ejemplo. Se tiene un listado de películas donde:

- A: Indica que la lista jala automáticamente información de la base de datos.
- Flecha Puntuada: Jala información de una tabla implícita, en este caso los actores que están relacionados con la película.



**Figura 2.5: Ejemplo uso de Unidad**

Fuente: (Brambilla, Comai, Fraternali, & Matera, 2013)

## 2.6 ANÁLISIS COSTO BENEFICIO

### 2.6.1 COCOMO II

COCOMO (COConstructive COst MOdel) es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software. Está asociado a los ciclos de vida modernos. El modelo original COCOMO ha tenido mucho éxito pero no puede emplearse con las prácticas de desarrollo software más recientes tan bien como con las prácticas tradicionales. COCOMO II apunta hacia los proyectos software de los 90 y de la primera década del 2000, y continuará evolucionando durante los próximos años. Sus objetivos son:

- Estimar el tiempo y el coste del software de acuerdo con los ciclos de vida utilizados en los 90 y en la primera década del 2000.
- Proporcionar un marco analítico cuantitativo y un conjunto de herramientas y técnicas para la evaluación de los efectos de la mejora tecnológica del software en costes y tiempo del ciclo de vida software.

En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto. Los factores de escala incluyen la parte de escala producida a medida que un proyecto de software incrementa su tamaño, COCOMO II posee tres modelos:

- **Composición de Aplicación:** Se emplea en desarrollo de software durante la etapa del prototipado.
- **Diseño Temprano:** Se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto.
- **Post-Arquitectura:** Se aplica en la etapa de desarrollo, después de definir la arquitectura del sistema, y en la etapa de mantenimiento.

Cada uno de estos modelos está orientado a sectores específicos del mercado de desarrollo de software y a las distintas etapas del desarrollo de software.

## 2.6.2 MODELOS DE ESTIMACIÓN

En la estimación del tamaño de software, COCOMO II utiliza tres técnicas:

- Puntos objeto
- Puntos fusión no ajustados y ajustados
- Líneas de código fuente

Además se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reuso, reingeniería, conversión y mantenimiento.

Para el presente proyecto se eligió el modelo de Puntos Función, el cual se define a continuación:

### 2.6.2.1 PUNTOS FUNCIÓN

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura.

Los puntos función están basados en la información disponible de las etapas tempranas del ciclo de vida del desarrollo de software. COCOMO II considera solamente UFP (Puntos Función no ajustados).

$$\mathbf{FP = UFP \times TCF}$$

Donde:

- **UFP:** Puntos Función no Ajustados
- **TCF:** Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes elementos:

- ✓ Archivos Lógicos Internos (ILF, del inglés Internal Logical Files): Grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de las Entradas Externas.
- ✓ Archivos de Interfaz Externos (EIF, del inglés External Interface Files): Grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones. Un caso de uso que como parte de alguna de sus secuencias de pasos indique que el sistema debe consultar información de alguna base de datos externa y mantenida por otro sistema.
- ✓ Entradas Externas (EI del inglés External Inputs): Un proceso elemental en el que ciertos datos cruzan la frontera del sistema desde afuera hacia adentro.
- ✓ Salidas Externas (EO, del inglés External Outputs): Un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (que se calculan a partir de otros datos) cruzan la frontera del sistema desde adentro hacia afuera. Adicionalmente, las Salidas Externas pueden actualizar un Archivo Lógico Interno.
- ✓ Consultas Externas (EQ, del inglés External Inquiries): Un proceso elemental con componentes de entrada y de salida donde un Actor del sistema rescata datos de uno

o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos). Dentro de éste tipo de transacción entran los listados y las búsquedas de los sistemas.

## **2.7 INVENTARIO**

El término “inventario” tiene dos definiciones (Real Academia Española , 2015), la primera como “asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden y precisión” y la segunda referida a “papel o documento en que están escritas dichas cosas”, significando entonces el conjunto de bienes (en este caso de una empresa) que se encuentra sometido a diferentes procedimientos de registros para su adecuado control.

En el ámbito comercial, el inventario se representa en un esquema de ventas donde se registran las operaciones que se producen desde que el cliente efectúa un pedido a las instalaciones hasta que se realiza su entrega:

- Se produce el pedido de uno o varios artículos a nuestras instalaciones.
- Se verifica el pedido en las instalaciones, caso contrario de no existir, se pide autorización para buscar en almacén.
- Se acepta el pedido.
- Se procesa la cancelación de dicho pedido.
- Se recepciona el pedido por parte del cliente.

En el esquema de aprovisionamiento, se realizan las siguientes operaciones que nos permitirán abastecernos de material de ventas:

- Al notar la falta de algún material se solicita el abastecimiento al proveedor.
- Se recepciona el pedido y se realiza el envío a los centros de venta.
- Se contabiliza la cantidad que ingresó.

### 2.7.1 TIPOS DE INVENTARIOS

Existen diferentes tipos de inventarios que se pueden realizar en un determinado tiempo y en una determinada ocasión.

- Inventarios finales: se realizan cada vez que se cierra el periodo fiscal, habitualmente el 31 de diciembre.
- Inventarios periódicos: se realizan cada determinado tiempo dentro de una empresa;
- Inventarios iniciales: se registran todos los bienes de la empresa; solo se documentan los bienes existentes en el o en los días de elaboración. Por lo general se elabora al inicio del periodo contable, que suele ser el 1 de enero.
- Inventarios de liquidación legal y política en su proceso.
- Inventarios de productos en proceso de fabricación: incluyen los bienes que ha adquirido una empresa de tipo manufacturera o industrial y están en proceso aún de manufactura; se cuantifican a través de la cantidad de materiales, de la mano de obra o de los gastos de fabricación, aplicables a la fecha de cierre.
- Inventarios de materias primas: incluyen los materiales que se requieren para la elaboración de los productos y que aún no se han procesado de ninguna manera.
- Inventarios de suministros de fábrica: incluyen todos los materiales que se utilizan para la elaboración de los productos, pero que no se pueden cuantificar exactamente; por ejemplo, pintura, lija, clavos, lubricantes.
- Inventarios de producción en proceso.
- Inventarios de productos terminados: los de los productos que adquieren las empresas manufactureras o industriales y después modifican o transforman para venderlos como productos elaborados por ellos.
- Inventarios de materiales y suministros; Son aquellos materiales con los que se elaboran los productos realizados en una empresa, pero que no pueden ser

cuantificados de una manera exacta. En las cuales pueden ser: Pintura, lija, clavos, lubricantes, entre muchos otros.

## **2.7.2 TÉCNICAS DE VALUACIÓN PARA SALIDA DE ALMACENES**

Los métodos de valuación o métodos de valuación de inventarios son técnicas utilizadas con el objetivo de seleccionar y aplicar una base específica para evaluar los inventarios en términos monetarios. La valuación de inventarios es un proceso vital cuando los precios unitarios de adquisición han sido diferentes.

Existen numerosas técnicas de valoración de inventarios, sin embargo las comúnmente utilizadas por las organizaciones en la actualidad son:

- Identificación específica
- Primeros en entrar Primeros en salir – PEPS
- Últimos en entrar Primeros en salir – UEPS
- Costo promedio constante o promedio ponderado

Estas técnicas son las más utilizadas y conocidas se tiene muchas pero mencionaremos la más importante para el proyecto.

### **2.7.2.1 METODO PRIMERO EN ENTRAR PRIMERO EN SALIR (PEPS)**

Este método identificado también como “PEPS”, se basa en el supuesto de que los primeros artículos en entrar al almacén son los primeros en salir de él.

Se ha considerado conveniente este método por que da lugar a una evaluación del inventario concordado con la tendencia de los precios; puesto que se presupone que el inventario está integrado por las compras más recientes y esta valorizado a los costos también más recientes, la valorización sigue entonces la tendencia del mercado.

Método PEPS, tipo de inventario perpetuo que detalla por medio de la tarjeta de control de inventario, las salidas y entradas de las mercancías. Establece que la primera mercancía que se compra es la primera en venderse o salir.



En tales condiciones, el costo está absorbiendo materiales a precio más reciente, que son los más bajos. El objetivo final de esta técnica es que las utilidades sean más conservadoras.

En cuanto se agota el producto más antiguo, con su correspondiente costo de adquisición. En inventario tiende a quedar valorado al costo de adquisición más reciente. Considerada que la primera unidad adquirida, son las primeras surtidas al ser vendidas. La existencia en el inventario corresponde a las compras más recientes.

## **2.8 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)..

## **2.9 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).

La definición que se tomara es que las ventas es un cambio de productos y servicios por dinero. Desde el punto de vista legal, se trata de la transferencia del derecho de posesión de un bien a cambio de dinero.

## **2.10 HERRAMIENTAS DE DESARROLLO DE SOFTWARE**

MySQL.- Es un sistema de gestión de base de datos SQL Open Source, esto significa que cualquiera puede usar y modificar el software, ya que cuenta con una licencia GLP (GNU General Public License). La base de datos de MySQL es relacional, esto quiere decir que almacena los datos en tablas separadas en lugar de poner todos los datos en un gran

almacén. La parte SQL de MySQL es sinónimo de “Structured Query Language”. SQL es el lenguaje estandarizado más común utilizando para acceder a bases de datos. MySQL consiste en un servidor multihilo y multiusuario.

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

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's (Interfaz de Programación de Aplicaciones) en gran cantidad de lenguajes (C, C++,Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- HTML (Lenguaje de Marcas de Hipertexto).- Es el lenguaje de mercado predominante para la construcción de páginas web, Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo javascript), el cual puede efectuar el comportamiento de navegadores web y otros procesadores de HTML.
- PHP (Hypertext Preprocessor).- PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de líneas de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.



PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web y en casi todos los sistemas operativos y plataformas sin costo alguno. Sus principales características son:

- Es un lenguaje multiplataforma.
  - Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
  - Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
  - (Lenguaje de marcas de hipertexto), es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
  - Permite las técnicas de programación Orientada a objetos.
  - Biblioteca nativa de funciones sumamente amplia e incluida.
- CODEIGNITER.- Codeigniter es un framework para el desarrollo de aplicaciones en php que utiliza el MVC. Permite a los programadores web mejorar la forma de trabajar y hacerlo a mayor velocidad. Al igual que cualquier framework está pensado para gente que tiene un dominio, al menos medio, del lenguaje de programación php.

Sus principales ventajas son:

- Separación de la lógica y arquitectura de la web, el MVC.
- Cualquier servidor que soporte PHP+MYSQL sirve para codeigniter.
- Facilidad de edición del código ya creado.
- Reutilización de código, desarrollo ágil.
- Las páginas se procesan muy rápido, el núcleo de codeigniter es bastante ligero.

Programa *DIA* versión 2.4, para Diagrama de clases, Entidad – Relación.

*Web ratio*, para el diagrama de componentes.

*Balsamiq Mockup 3*, para el modelo de presentación.

## MARCO APLICATIVO

### 3.1 INTRODUCCIÓN

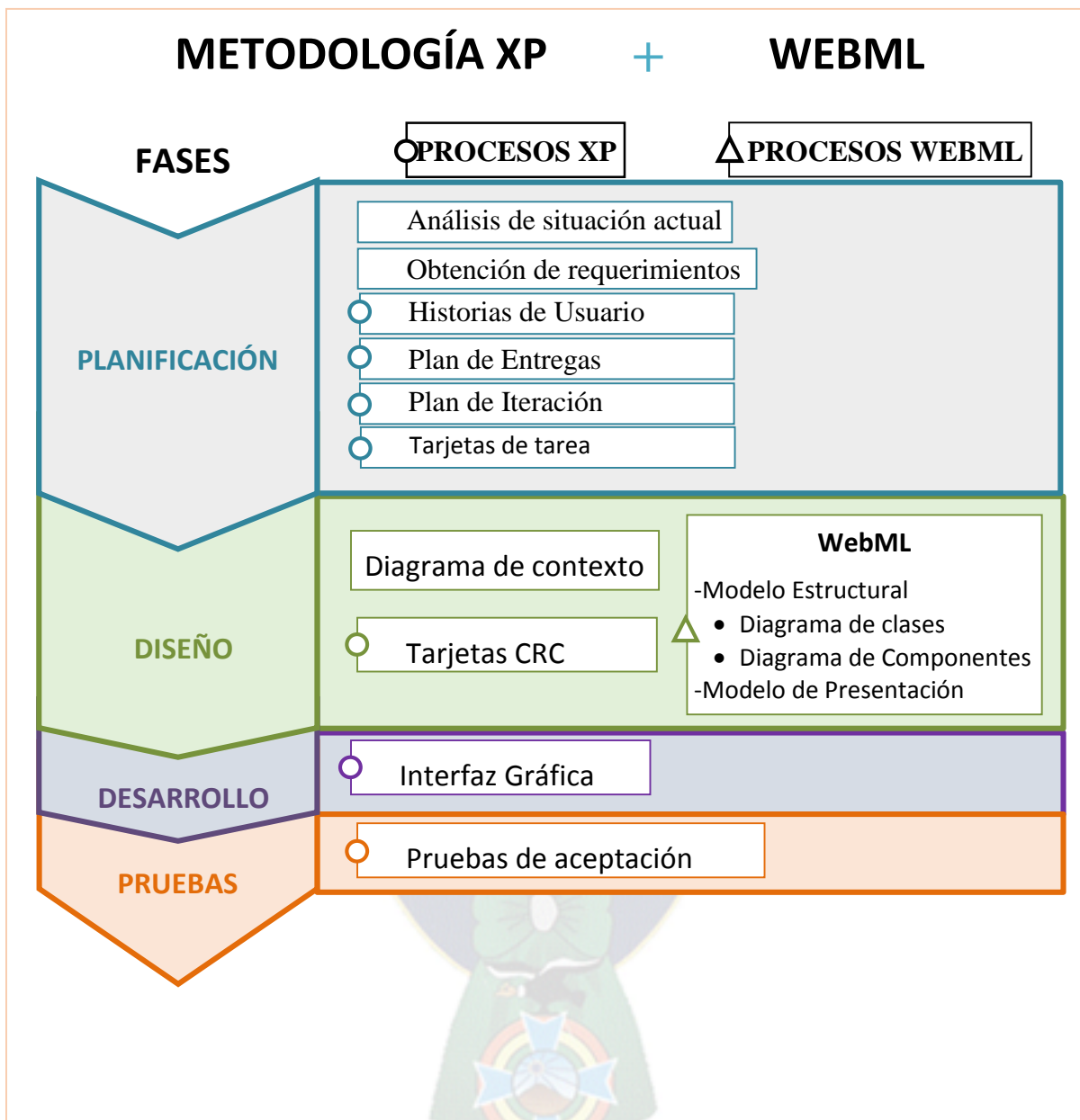
En este capítulo se detalla la forma de organización y métodos de trabajo del sistema, se hará uso de las metodologías y herramientas descritas anteriormente, las mismas que nos servirán para el desarrollo del sistema y todos sus módulos.

Al comenzar con el desarrollo del sistema se observó que la comunicación con el equipo de trabajo era muy importante, desde el cliente que forma parte del equipo haciendo conocer la forma del trabajo actual, explicando paso a paso el procedimiento en el área de ventas y captar las necesidades de los mismos.

El análisis de los requerimientos resultó un tanto moroso, debido a la ambigüedad con la que se manejan varios aspectos como ser las peticiones del cliente. También influye la cantidad de producto que compra un cliente, además depende del tipo de producto, quedando sujeto a la categoría del producto.

Por estas razones se hará el uso de la metodología XP, donde claramente pone en comunicación directa y continua a clientes y desarrolladores, este aspecto fue aplicado ya que se mantuvieron reuniones frecuentes con el personal del cliente, quienes se integraron al proyecto para establecer prioridades y resolver dudas. Dicha metodología fue elegida por que responde de manera favorable a un ambiente de requerimientos dinámicamente cambiantes. Además, aborda los riesgos del proyecto y las prácticas, están creadas para mitigar los riesgos y elevar las posibilidades de éxito.

En el esquema que se muestra a continuación se ve como la metodología de desarrollo ágil programación extrema XP, interactuara con el modelo WebML. XP ayudará al desarrollo del proyecto atraves de las iteraciones, donde cada iteración comprenderá las fases de planificación, diseño, desarrollo y pruebas. Interactuando con la fase de diseño del modelo WebML haciendo uso de los modelos de datos, modelo estructural.



**Figura 3.1: Metodología XP reformulada**  
Fuente: Elaboración propia

## 3.2 FASES DE LA METODOLOGÍA XP

### 3.2.1 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.

### **3.2.1.1 ANALISIS DE SITUACIÓN ACTUAL**

Se tuvieron reuniones con los responsables de la Empresa Protec, en las cuales permitió la observación de sus procesos cotidianos. El producto de estos encuentros dio lugar al siguiente análisis de la situación actual en la que se encuentra Protec.

- El personal que está encargado en la venta la realiza manualmente el registro de los artículos vendidos, los registra en un talonario de facturas.
- Para realizar la venta no se tiene un registro de los productos existentes y la búsqueda es realizada de forma manual.
- Los precios de los productos no están normalizados porque no existe un registro de precios para el personal de venta.
- Se cuenta con una lista de precios de los productos, la cual está elaborada de forma manual y se contrasta a simple vista cada vez que se modifica.
- Al finalizar el día lo registrado de la venta no concuerda con el dinero que se encuentra en caja.
- En cada compra, la verificación de la existencia del producto es realizada de forma manual comparándolo con una lista en un documento físico.
- Hasta no hacer el proceso de obtención de precios, el producto no puede salir a exposición y ser vendido.
- Los precios finales están registrados en un documento físico, la cual es utilizada como guía de precios.
- No se registran los productos entrantes o salientes por tanto no existe un control

del inventario existente.

- No se tiene un registro o control de todos los productos existentes o del flujo que existe de entrantes o salientes de productos. Por esta situación sufren una serie de problemas ya que no tienen a disposición esta información.

### 3.2.1.2 OBTENCIÓN DE REQUERIMIENTOS

A continuación, se muestra cada uno de los requerimientos que fueron descritos por el cliente y los futuros usuarios del sistema:

- **Registrar información de artículos nuevos para la empresa:** El administrador debe registrar el ingreso de artículos nuevos para la empresa.
- **Contacto directo con el vendedor:** El sistema debe proporcionar un contacto directo con el vendedor de la empresa, mediante la introducción de usuario y contraseña que se le fue asignado por el administrador luego ingresará al sistema.
- **Administrar usuarios:** Todo sistema debe contar con este módulo, donde el usuario pueda administrar los ajustes básicos del sistema así como permitir la asignación de un rol a un usuario.
- **Administración de clientes:** Crear, listar, modificar y eliminar registros de clientes.
- **Clasificar los productos en categorías:** Crear, listar, modificar y eliminar registros de las categorías, y a la vez permitir adicionar uno o más categorías a cada producto.
- **Registro de ventas:** Las ventas se deben registrar cada vez que se realiza una venta de productos.
- **Registrar datos de proveedores:** Se debe realizar el registro de los datos personales de todos los proveedores que trabajan con la empresa.

- **Registro de compras o abastecimiento relacionando a los proveedores:** Hacer el registro correspondiente de cada compra relacionando al proveedor. Para las compras se requiere seguir lo siguiente:
  - Hacer el registro de la cantidad de cada producto, el cual debe ser adicionado al stock ya existente en el caso de que el producto sea nuevo, se deberá ir al registro de productos.
- **Relación directa del inventario con las compras y ventas:** En cada transacción de compra o venta, el inventario debe incrementar o disminuir en el ítem respectivo.
- **Entorno grafico amigable al usuario:** El usuario deberá adaptarse fácilmente al uso del sistema.

### 3.2.1.3 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:

- **Prioridad:** De las historias de usuarios existen algunas con mucha importancia para su desarrollo y de acuerdo a conversaciones con todas las personas que tendrán acceso al sistema, tiene tres grados de prioridad para el desarrollo e implementación de las historias de usuario que son alta, media y baja.
- **Riesgos de desarrollo:** Es el riesgo que existe al desarrollar de forma inadecuada la solución de las historias de usuarios, se tiene tres grados de riesgos de los cuales son alto, medio y bajo.
- **Iteración asignada:** Es el número de iteración en el cual se espera poder implementar la historia de usuario, el tiempo promedio de entrega en cada iteración es aproximadamente tres semanas.
- **Punto estimado:** Es el tiempo promedio en semanas de desarrollo los cuales se miden en la escala de 1 a 3 semanas de desarrollo aproximadamente, según la



complejidad de cada iteración.

Después de establecer los puntos de medida de la historias de usuario a continuación se describirán cada una de estas, junto a una descripción de las mismas.

**H.U. de productos:** El sistema debe registrar la información necesaria de los productos con la que cuenta la empresa, lo cual es realizado por el administrador. Esta historia de usuario tiene una prioridad alta. El riesgo de desarrollo es alto debido a que este objeto es la base del sistema y es preciso definirlo.

Por otro lado 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	
<b>Número:1</b>	<b>NOMBRE:</b> Productos
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada: 1</b>
<b>Prioridad:</b> Alta	
<b>Descripción:</b> El registro de productos nuevos y actualización de productos ya existentes es indispensable para el sistema, se lo realizará de acuerdo a las características de los artículos de la empresa tomando los datos necesarios de los artículos los cuales son: código del producto, nombre del producto, categoría del producto, descripción del producto, precio de entrada, precio de salida, unidad del producto y otros.	

**Tabla 3.1: Historia de Usuario 1**  
Fuente: Elaboración propia

HISTORIA DE USUARIO	
<b>Número:2</b>	<b>NOMBRE:</b> Categorías
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:1</b>
<b>Prioridad:</b> Medio	

**Descripción:** El sistema debe tener la opción de crear, listar, modificar y eliminar registros de categorías.

**Tabla 3.2: Historia de Usuario 2**

Fuente: Elaboración propia

HISTORIA DE USUARIO	
<b>Número:3</b>	<b>NOMBRE:</b> Clientes
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:</b> 1
<b>Prioridad:</b> Baja	
<b>Descripción:</b> El sistema debe tener la opción de crear, listar, modificar y eliminar registros de clientes.	

**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	
<b>Número:4</b>	<b>NOMBRE:</b> Ventas
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:</b> 2
<b>Prioridad:</b> Alta	
<b>Descripción:</b> En este módulo se realizará el registro de las ventas de los productos teniendo en cuenta los datos necesarios para la operación así como: 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	
<b>Número:5</b>	<b>NOMBRE:</b> Proveedores
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:3</b>
<b>Prioridad:</b> Alta	
<b>Descripción:</b> El Sistema debe tener la opción de administrar los proveedores con los siguientes datos: nombre, apellido, dirección, email, teléfono.	

**Tabla 3.5: Historia de Usuario 5**

Fuente: Elaboración propia

**H.U. de abastecimiento:** 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	
<b>Número:6</b>	<b>NOMBRE:</b> Compra o Abastecimiento
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:3</b>
<b>Prioridad:</b> Alta	
<b>Descripción:</b> El administrador o el personal de venta 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	
<b>Número:7</b>	<b>NOMBRE:</b> Inventarios
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:4</b>
<b>Prioridad:</b> Alta	
<b>Descripción:</b> Se desarrollará el módulo de inventario debido a que se necesita tener los reportes de todos los módulos del sistema. Este módulos 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.	

**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	
<b>Número :8</b>	<b>Nombre:</b> Usuarios
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:2</b>
<b>Prioridad:</b> Alta	
<b>Descripción:</b> Se desarrollará el módulo de usuarios para el ingreso al sistema, se debe ingresar al sistema usando un nombre de usuario y contraseña.	

**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	
<b>Número :9</b>	<b>Nombre:</b> Reporte de Ventas
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:</b> 4
<b>Prioridad:</b> Alta	
<b>Descripción:</b> 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	
<b>Numero :10</b>	<b>Nombre:</b> Reporte de inventario
<b>Autor:</b> Monica Sarco Mendoza	<b>Iteración asignada:</b> 4
<b>Prioridad:</b> Alta	
<b>Descripción:</b> 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

La siguiente tabla muestra un resumen de las Historias de Usuario y su implementación.

Nº	Nombre	Prioridad	Riesgo	Puntos estimados	Iteración
1	Productos	Alta	Alto	2	1
2	Categorías	Medio	Bajo	1	1
3	Clientes	Baja	Medio	1	1
4	Venta	Alta	Alto	3	2
5	Proveedores	Baja	Bajo	1	3
6	Compra o Abastecimiento	Alta	Alto	3	3
7	Inventarios	Alta	Medio	2	4
8	Usuarios	Media	Bajo	1	2
9	Reporte de ventas	Baja	Bajo	2	4
10	Reporte de Inventario	Baja	Bajo	2	4

**Tabla 3.11: Resumen Historia de Usuario**

Fuente: Elaboración propia

### 3.2.1.4 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.

#### ➤ Primera Iteración

Nro.	Historia de Usuario
H1	Productos
H2	Categorías
H3	Clientes

**Tabla 3.12: Historias de Usuario de la Primera Iteración**

Fuente: Elaboración Propia

#### ➤ Segunda Iteración

Nro.	Historia de Usuario
H4	Ventas
H8	Usuarios

**Tabla 3.13: Historias de Usuario de la Segunda Iteración**  
Fuente: Elaboración Propia

➤ **Tercera Iteración**

Nro.	Historia de Usuario
H5	Proveedores
H6	Compra o Abastecimiento

**Tabla 3.14: Historias de Usuario de la Tercera Iteración**  
Fuente: Elaboración Propia

➤ **Cuarta Iteración**

Nro.	Historia de Usuario
H7	Inventarios
H9	Reporte de ventas
H10	Reporte de inventario

**Tabla 3.15: Historias de Usuario de la Cuarta Iteración**  
Fuente: Elaboración Propia

### 3.2.1.5 TARJETAS DE TAREA

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 “ABM” que significa Altas, Bajas y Modificaciones, son el conjunto de funciones básicas que todo módulo debería tener.

### 3.2.1.5.1 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:

- ✓ H1: Productos
- ✓ H2: Categorías
- ✓ H3: Clientes

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

TAREA	
<b>Numero de Tarea:</b> 1.1	<b>Numero de Historia:</b> 1
<b>Nombre de Tarea:</b> Crear ABM de Productos	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 01-02-17	<b>Fecha Fin:</b> 14-02-17
<b>Descripción:</b> Crear funciones para crear, modificar, eliminar y listar el objeto Producto, sus datos básicos son: Código, imagen, nombre, precio de entrada, precio de salida, categorías, mínima y activo.	

**Tabla 3.16: Tarjeta de Tarea 1.1 de Historia de Usuario 1**

Fuente: Elaboración propia

TAREA	
<b>Numero de Tarea:</b> 2.1	<b>Numero de Historia:</b> 2
<b>Nombre de Tarea:</b> Crear ABM de Categorías	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 14-02-17	<b>Fecha Fin:</b> 20-02-17
<b>Descripción:</b> Crear funciones para crear, modificar, eliminar y listar el objeto Categoría, su dato básico es: Nombre de categoría.	

**Tabla 3.17: Tarjeta de Tarea 2.1 de Historia de Usuario 2**

Fuente: Elaboración propia

TAREA	
<b>Numero de Tarea:</b> 3.1	<b>Numero de Historia:</b> 3
<b>Nombre de Tarea:</b> Crear ABM de Clientes	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 21-02-17	<b>Fecha Fin:</b> 27-02-17
<b>Descripción:</b> Crear funciones para crear, modificar, eliminar y listar el objeto Cliente, sus datos básicos son: Nombre completo, dirección, email, teléfono.	

**Tabla 3.18: Tarjeta de Tarea 3.1 de Historia de Usuario 3**

Fuente: Elaboración propia

### 3.2.1.5.2 SEGUNDA ITERACIÓN

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

- ✓ H4: Ventas
- ✓ H8: Usuarios

Esta área permite almacenar datos de la venta realizada que serán mostrados al personal de venta en un resumen, de los que podrá ver cada producto con la descripción de la venta realizada.

En esta iteración se calcula un alto riesgo, ya que el módulo de Ventas es de análisis complejo.

TAREA	
<b>Numero de Tarea:</b> 4.1	<b>Numero de Historia:</b> 4
<b>Nombre de Tarea:</b> Registro de ventas	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 01-03-17	<b>Fecha Fin:</b> 20-03-17
<b>Descripción:</b> En este módulo tendremos primeramente un buscador de productos por nombre o por código para realizar la venta correspondiente del producto solicitado, seguidamente mostrará los resultados de búsqueda donde se adicionara la cantidad de productos a vender, luego nos 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 y finalizar la venta.	

**Tabla 3.19: Tarjeta de Tarea 4.1 de Historia de Usuario 4**

Fuente: Elaboración propia



TAREA	
<b>Numero de Tarea:</b> 5.1	<b>Numero de Historia:</b> 8
<b>Nombre de Tarea:</b> Usuario	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 21-03-17	<b>Fecha Fin:</b> 31-03-17
<b>Descripción:</b> Crear las funciones para registrar nuevo usuario, modificar y listar. A la vez, se validarán los datos ingresados. Debe tener como campos básicos como: Nombre, apellido, nombre de usuario, email, contraseña.	

**Tabla 3.20: Tarjeta de Tarea 5.1 de Historia de Usuario 8**

Fuente: Elaboración propia

### 3.2.1.5.3 TERCERA ITERACIÓN

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

- ✓ H5: Proveedores
- ✓ H6: Compras o abastecimiento

En este módulo se registra a todos los proveedores que brindan un servicio a la empresa.

TAREA	
<b>Numero de Tarea:</b> 6.1	<b>Numero de Historia:</b> 5
<b>Nombre de Tarea:</b> Diseño de formulario de registro de proveedores	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 03-04-17	<b>Fecha Fin:</b> 10-04-17
<b>Descripción:</b> Se crea un formulario que permita introducir el nombre del proveedor, con toda la descripción de los datos del proveedor además se comprobará que estos datos sean almacenados y guardados correctamente en la base de datos.	

**Tabla 3.21: Tarjeta de Tarea 6.1 de Historia de Usuario 5**

Fuente: Elaboración propia

TAREA	
<b>Numero de Tarea:</b> 7.1	<b>Numero de Historia:</b> 6
<b>Nombre de Tarea:</b> Crear función de ingreso de productos a inventarios	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 11-04-17	<b>Fecha Fin:</b> 30-04-17



**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 7.1 de Historia de Usuario 6**

Fuente: Elaboración propia

#### 3.2.1.5.4 CUARTA ITERACIÓN

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

- ✓ H7: Inventarios
- ✓ H9: Reporte de ventas
- ✓ H10: Reporte de inventario

TAREA	
<b>Numero de Tarea:</b> 8.1	<b>Numero de Historia:</b> 7
<b>Nombre de Tarea:</b> Desarrollo del módulo de inventarios	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 01-05-17	<b>Fecha Fin:</b> 15-05-17
<b>Descripción:</b> Se mostrará un listado de todos los productos existentes con una descripción necesaria de cada producto.	

**Tabla 3.23: Tarjeta de Tarea 8.1 de Historia de Usuario 7**

Fuente: Elaboración propia

TAREA	
<b>Numero de Tarea:</b> 9.1	<b>Numero de Historia:</b> 9
<b>Nombre de Tarea:</b> Reporte de ventas	
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha Inicio:</b> 16-05-17	<b>Fecha Fin:</b> 29-05-17
<b>Descripción:</b> Desarrollar un módulo que muestre las ventas realizadas en un rango de fecha que será variable.	

**Tabla 3.24: Tarjeta de Tarea 9.1 de Historia de Usuario 9**

Fuente: Elaboración propia

TAREA	
Numero de Tarea: 10.1	Numero de Historia: 10
Nombre de Tarea: Reporte de inventario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 30-05-17	Fecha Fin: 13-06-17
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 10.1 de Historia de Usuario 10**

Fuente: Elaboración propia

### 3.2.1.6 PLAN DE ITERACIÓN

Iteraciones	Historia de Usuario	Duración	Fecha inicio	Fecha fin
Primera	Producto	4 semanas	01/02/2017	28/02/2017
	Categoría			
	Cientes			
Segunda	Ventas Usuarios	4 semanas	01/03/2017	31/03/2017
Tercera	Proveedor Compra o Abastecimiento	4 semanas	03/04/2017	28/04/2017
Cuarta	Inventarios Reporte de ventas Reporte de Inventarios	6 semanas	01/05/2017	13/06/2017

**Tabla 3.26: Plan de iteración**





Fuente: Elaboración propia

### 3.2.1.7 PLAN DE ENTREGAS

Nº	Semanas/Fases	Enero				Febrero				Marzo					Abril				Mayo				Junio				
		1	2	3	4	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4	1	2	3	4	5
1	Planificación																										
2	Diseño																										
3	Desarrollo																										
4	Pruebas																										

**Tabla 3.27: Plan de entrega**

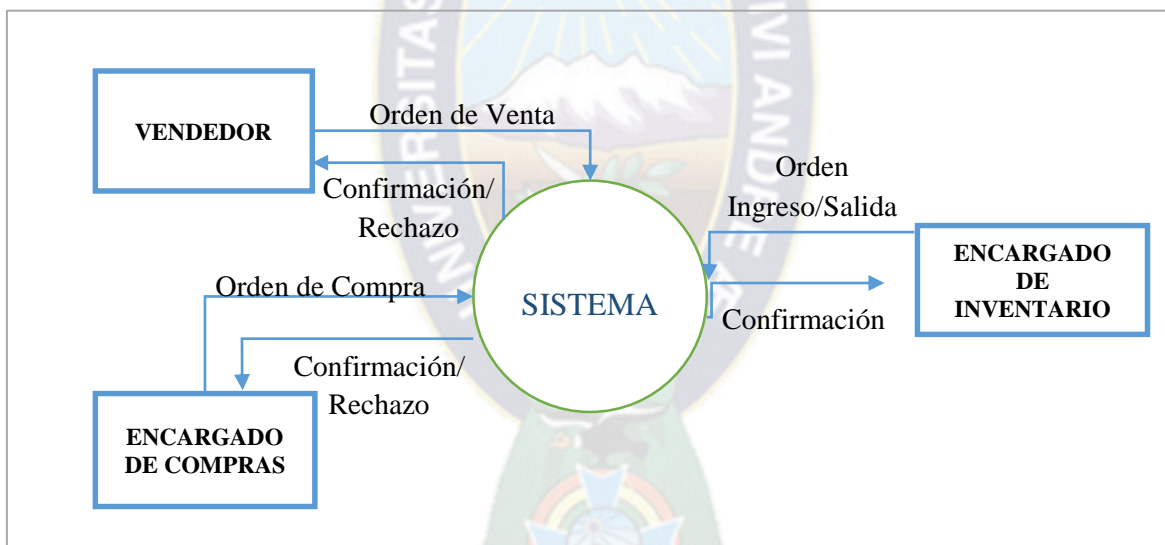
Fuente: Elaboración propia

				
Planificación Global	1ra Iteración	2da Iteración	3ra Iteración	4ta Iteración
3 semanas	4 semanas	4 semanas	4 semanas	6 semanas

### 3.2.2 FASE II - DISEÑO

En esta fase se presentan 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 WebML.

Previamente, se elaboró el diagrama de contexto del sistema. Éste identifica las entidades externas que tienen vínculo con el sistema y que se interrelacionan de alguna forma con el sistema, así como los flujos de datos entre cada entidad y el sistema.



**Figura 3.2: Diagrama de contexto del sistema**

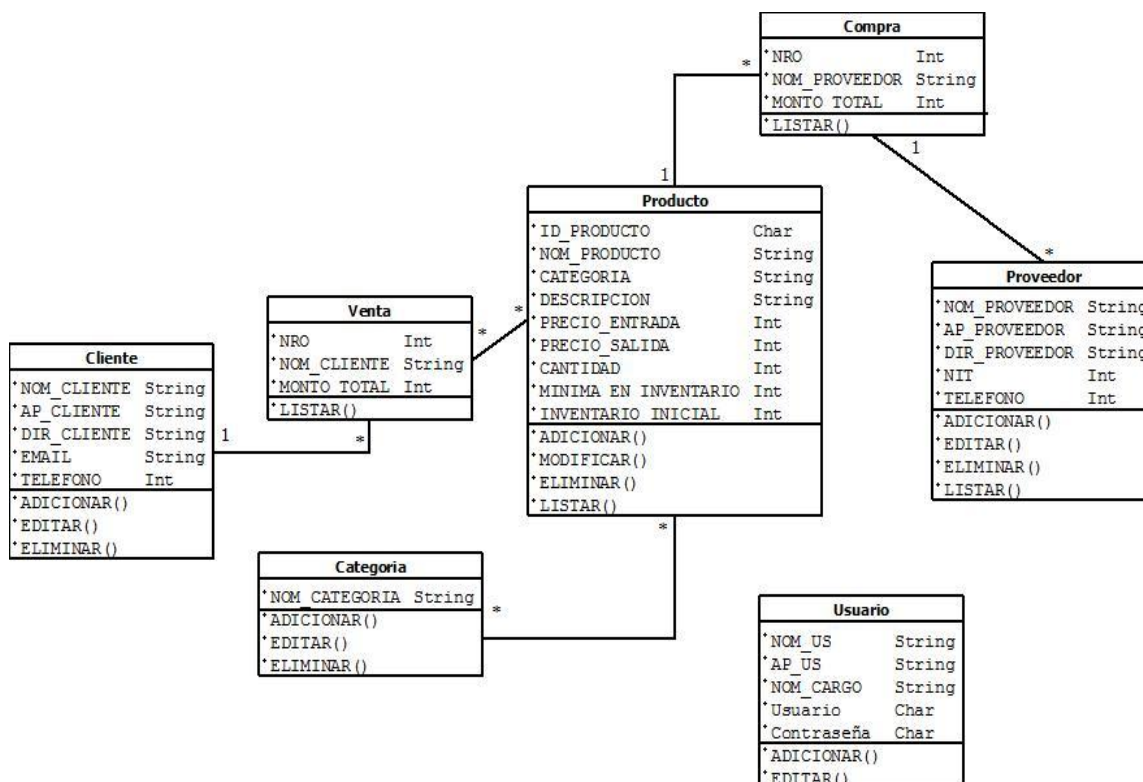
Fuente: Elaboración propia

#### 3.2.2.1 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.

### 3.2.2.1.1 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. Será la guía al momento de crear las tarjetas CRC.



**Figura 3.3: Diagrama de clases**

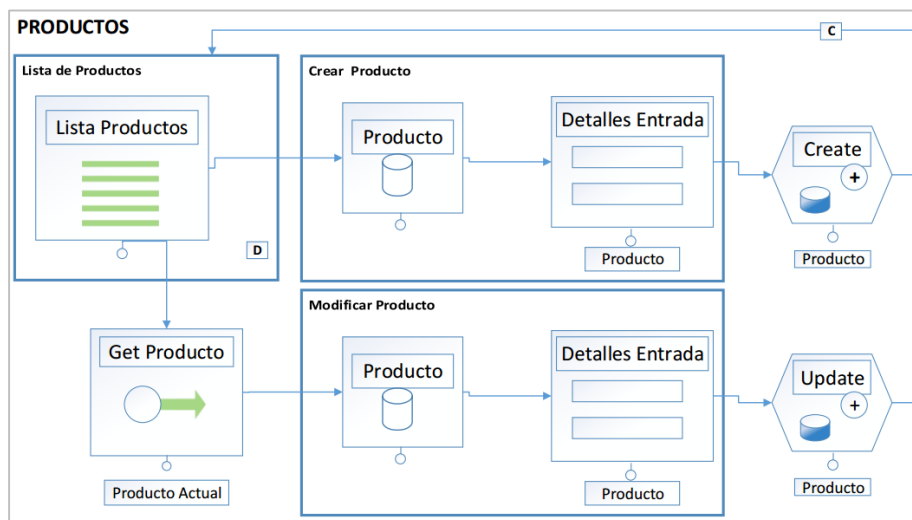
Fuente: Elaboración propia

### 3.2.2.1.2 DIAGRAMA DE COMPONENTES

Un diagrama de componentes representa como un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes.

Con estos diagramas definiremos las unidades y contenidos de las páginas web que son parte de la interfaz del usuario.

La siguiente figura 3.4 muestra el **diagrama de componentes** de Producto, donde se tiene la opción de crear un nuevo registro o modificar un registro actual, una vez modificado o creado, el proceso vuelve a la pantalla de listado de productos.



**Figura 3.4: Diagrama de componentes – ABM de producto**

Fuente: Elaboración propia

- **MODELO DE PRESENTACIÓN**

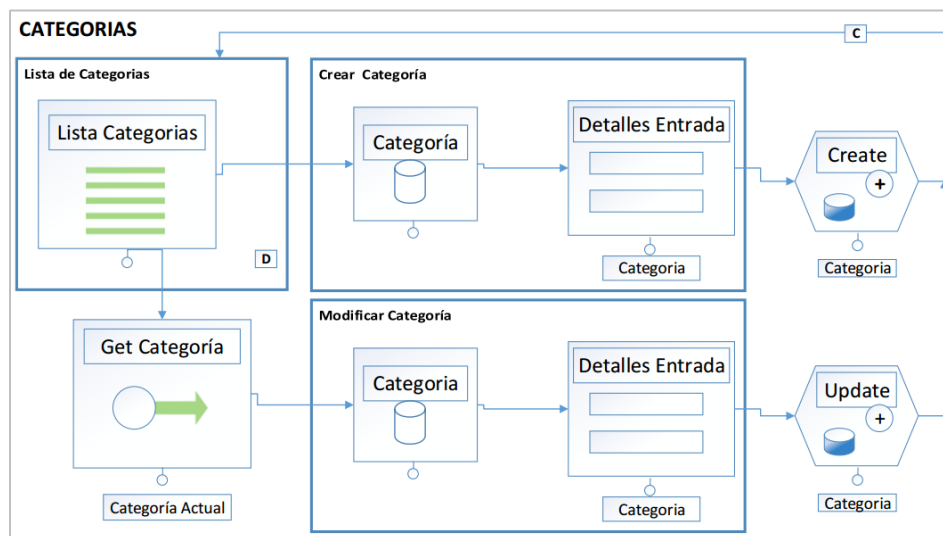
La figura 3.5 muestra el modelo de presentación de producto con las opciones de editar, eliminar y agregar nuevo producto.

Codigo	Nombre	PrecioEntrada	PrecioSalida	Categoria	Minima
1	Casco	\$30	\$50	MaterialBombero	3
2	Botas	\$70	\$110	MaterialBombero	5
3	ExtintorQ	\$250	\$350	Extintores	7

**Figura 3.5: Modelo de Presentación – ABM de producto**

Fuente: Elaboración propia

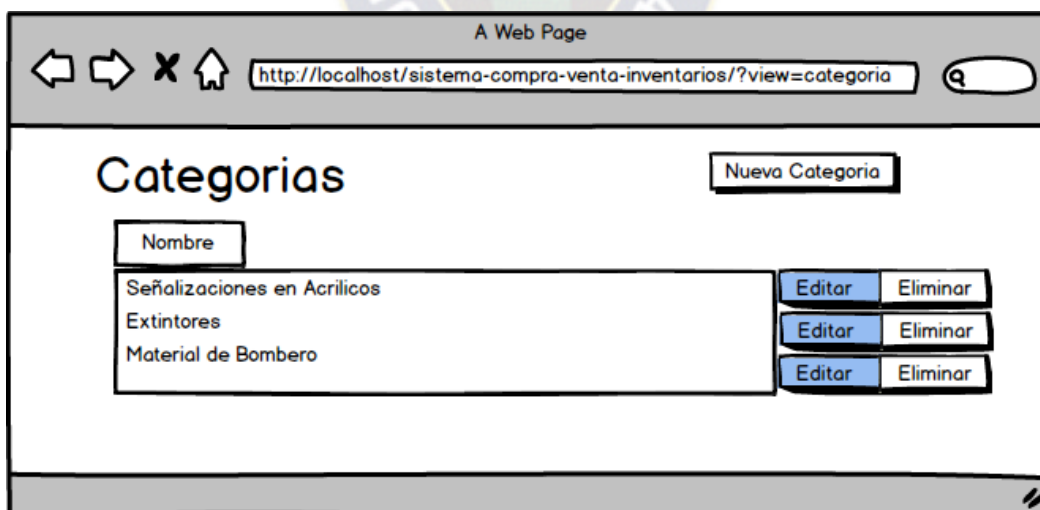
La Figura es un ABM básico y se repite para la mayoría de los objetos en el sistema, esto puede apreciarse en el siguiente **diagrama de componentes**, el cual describe el funcionamiento y componentes del ABM de Categorías:



**Figura 3.6: Diagrama de componentes – ABM de categoría**

Fuente: Elaboración propia

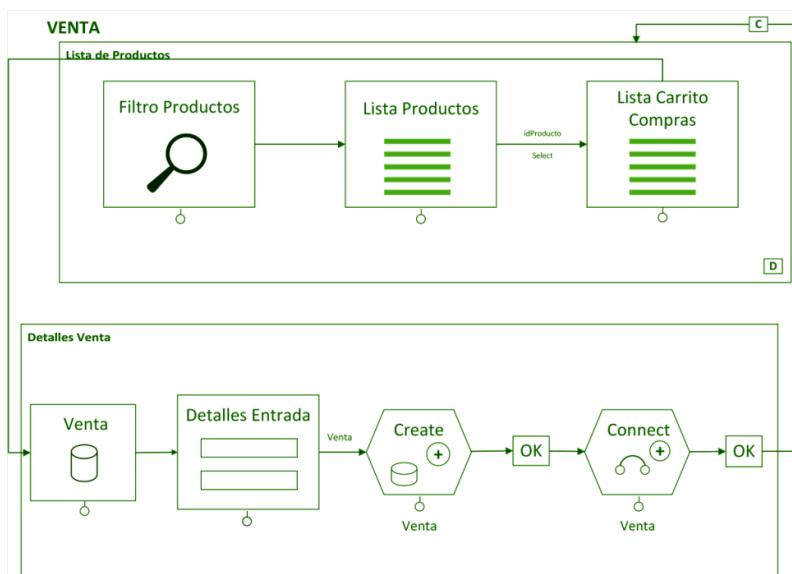
- **Modelo de Presentación**



**Figura 3.7: Modelo de Presentación – ABM de categoría**

Fuente: Elaboración propia

El siguiente diagrama muestra la composición del proceso de registro de una venta. Se tiene primeramente un filtro que tiene lista productos como resultados de un criterio de búsqueda, los componentes de esta lista pueden ser añadidos a otra lista considerada el carrito de búsqueda. Una vez seleccionados los productos se procede a poner datos de la compra y se registra la compra.



**Figura 3.8: Diagrama de componentes – Registro de Ventas**

Fuente: Elaboración propia

- **Modelo de Presentación**

**Figura 3.9: Modelo de Presentación – Registro de Venta**

Fuente: Elaboración propia

El siguiente diagrama muestra la composición del proceso de registro de una compra. Como página por defecto se visualiza una lista de productos que fue importada, el dato de proveedor ya fue elegido, en la siguiente pantalla se tiene un formulario donde deben confirmarse las cantidades recibidas, luego se procede a crear la compra ingresando los productos a inventario.

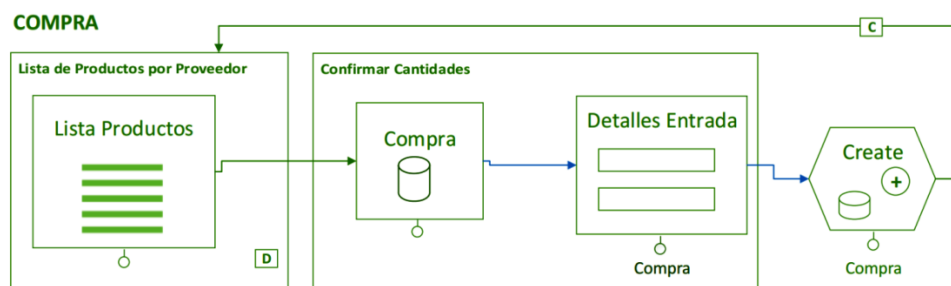


Figura 3.10: Diagrama de componentes – Registro de Compra

Fuente: Elaboración propia

- **Modelo de Presentación**

A Web Page

http://localhost/sistema-compra-venta-inventarios/?view=re

## Reabastecer Inventario

Buscar producto por nombre o por código

...

Buscar

### Lista de Reabastecimiento

Código	Cantidad	Unidad	Producto	PrecioUnitario	PrecioTotal
4	3	5	Casco	\$30	\$90

Cancelar

### Resumen

Proveedor:

Efectivo:

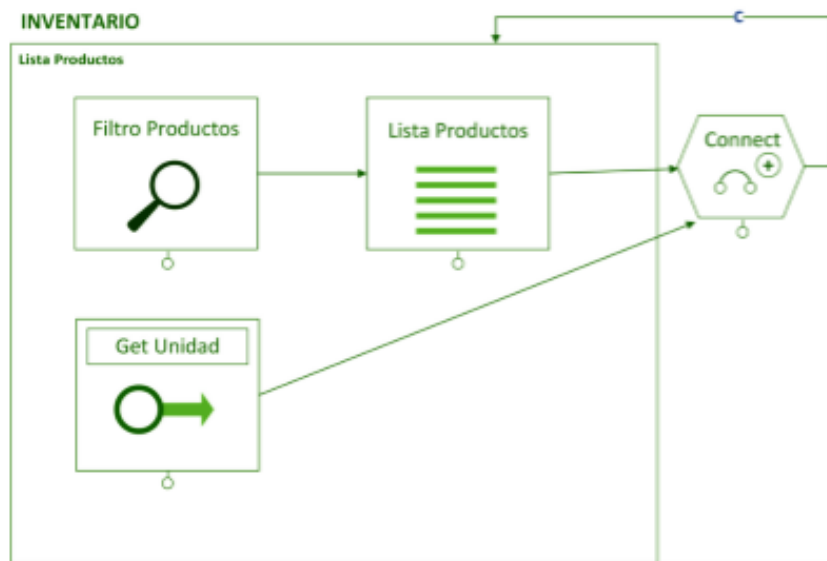
Cancelar ProcesarReabastecimiento

Figura 3.11: Modelo de Presentación – Registro de Compra

Fuente: Elaboración propia



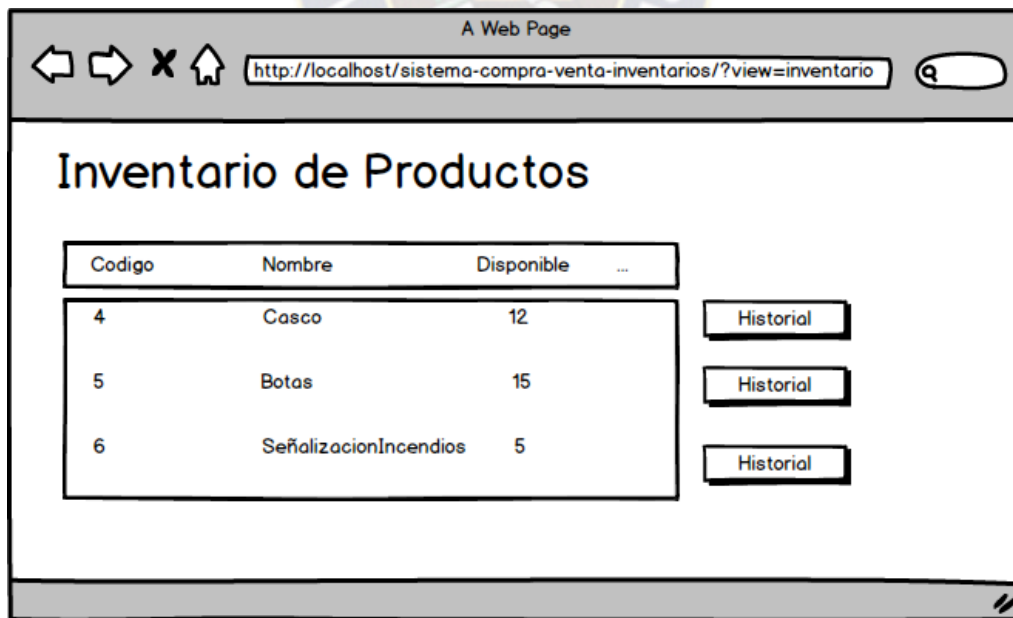
El siguiente diagrama muestra la composición del reporte de los estados de los productos. Puede apreciarse que para filtrar la lista de productos de inventario.



**Figura 3.12: Diagrama de componentes – Inventario**

Fuente: Elaboración propia

- **Modelo de Presentación**



**Figura 3.13: Modelo de Presentación – Inventario**

Fuente: Elaboración propia

### 3.2.2.2 TARJETAS CRC (CLASE- RESPONSABILIDAD – COLABORACIÓN)

Para el diseño de las tarjetas CRC se necesita identificar que se debe hacer y sobre quien se deben realizar las acciones.

A continuación se muestra el diseño de las tarjetas CRC para las principales historias de usuario:

- La clase Producto

PRODUCTO	
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"><li>• Crea, modifica, lista y elimina registros de productos</li><li>• Calcula el precio del producto con descuento</li></ul>	<ul style="list-style-type: none"><li>• Autor</li><li>• Categoría</li><li>• Base de datos de producto</li></ul>

- Clase Proveedor

PROVEEDOR	
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"><li>• Nuevo Proveedor</li><li>• Actualiza Proveedor</li><li>• Eliminar Proveedor</li><li>• Listar Proveedor</li></ul>	<ul style="list-style-type: none"><li>• Registro de Proveedores</li><li>• Base de datos de Proveedores</li></ul>

- Clase Venta

VENTA	
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"><li>• Crea registros de ventas</li><li>• Descuenta ítems del inventario</li><li>• Calcula total de la venta</li><li>• Calcula descuentos</li></ul>	<ul style="list-style-type: none"><li>• Producto</li><li>• Cliente</li></ul>

- Clase Compra o Abastecimiento

<b>COMPRA</b>	
<b>RESPONSABILIDADES</b>	<b>COLABORADORES</b>
<ul style="list-style-type: none"> <li>• Crea registros de compras</li> <li>• Agrega ítems al inventario</li> <li>• Calcula total de la compra</li> </ul>	<ul style="list-style-type: none"> <li>• Producto</li> <li>• Proveedor</li> </ul>

- Clase Inventarios

<b>INVENTARIOS</b>	
<b>RESPONSABILIDADES</b>	<b>COLABORADORES</b>
<ul style="list-style-type: none"> <li>• Crea registros de inventarios</li> <li>• Modifica inventarios</li> </ul>	<ul style="list-style-type: none"> <li>• Producto</li> </ul>

### 3.2.3 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 mostraran el proceso de desarrollo y capturas de pantalla de las historias de usuario más importantes.

#### 3.2.3.1 PRIMERA ITERACIÓN

A continuación se muestran las capturas de pantallas más relevantes, el resto se mostrara en anexos, que son el resultado de la implementación de las tareas de la primera iteración.

La siguiente figura 3.14 muestra el Altas Bajas Modificaciones de Producto, se tiene en el panel un listado de los productos que se tienen registrados, se tiene la opción ABC es decir se crea o agrega un nuevo producto, se elimina un producto y se modifica. Además cuenta con la opción de poder descargar en un documento Word el reporte de productos.

Codigo	Imagen	Nombre	Precio Entrada	Precio Salida	Categoria	Minima	Activo
001		Casco	\$ 30.00	\$ 50.00	Material de Bombero	5	✓
002		Botas	\$ 70.00	\$ 110.00	Material de Bombero	2	✓
003		Señalización de Incendios	\$ 60.00	\$ 100.00	Señalizaciones	8	✓
004		Extintore-Polvo-Químico	\$ 250.00	\$ 350.00	Extintores	4	✓
005		Señalamientos Restrictivos	\$ 50.00	\$ 80.00	Señalizaciones	6	✓
006		Chaleco de vialidad	\$ 110.00	\$ 200.00	Seguridad Industrial	7	✓
007		Conos de Señalización para trafico	\$ 30.00	\$ 60.00	Seguridad Industrial	7	✓

**Figura 3.14: Pantalla– ABM Producto**

Fuente: Elaboración propia

La siguiente figura 3.15 muestra el Altas Bajas Modificaciones de Cliente, se tiene en el panel un listado de los clientes que se tienen registrados, se tiene la opción ABC es decir se crea o agrega un nuevo cliente, se elimina un cliente y se modifica. Además cuenta con la opción de poder descargar en un documento Word el reporte de clientes.

Nombre completo	Direccion	Email	Telefono
Carlos Fernandez	av/ America z/ 3 de mayo	carlos@gmail.com	72505661
Claudio Apaza	av/ Tejada Sorsano	caludioapaza@gmail.com	22637884
Juan Perez	av/ las americas z/ central # 1250	juanperez@gmail.com	22347443
Pedro Durán	z/villa fatima	pedroduran@gmail.com	72403356

**Figura 3.15: Pantalla– ABM de directorio de clientes**

Fuente: Elaboración propia

### 3.2.3.2 SEGUNDA ITERACIÓN

La figura 3.16 muestra la ventana de venta donde se puede observar que cuenta con un buscador de productos por nombre o código para realizar la venta correspondiente, muestra

un listado de venta y resumen donde se tiene la opción de seleccionar un cliente ya registrado quien está realizando la compra.

Codigo	Cantidad	Unidad	Producto	Precio Unitario	Precio Total
9	3	5	Chaleco de seguridad	\$ 200	\$ 600
5	3	5	Botas	\$ 110	\$ 330
4	1	5	Casco	\$ 50	\$ 50

Figura 3.16: Pantalla– Panel de Ventas  
Fuente: Elaboración propia

Codigo	Cantidad	Nombre del Producto	Precio Unitario	Total
9	3	Chaleco de seguridad	\$ 200.00	\$ 600.00
5	3	Botas	\$ 110.00	\$ 330.00
4	1	Casco	\$ 50.00	\$ 50.00

Figura 3.17: Pantalla– Resumen de Ventas  
Fuente: Elaboración propia

### 3.2.3.3 TERCERA ITERACIÓN

Nombre completo	Direccion	NIT	Telefono
Esteban Mercado	av/ las americanas # 1250	7422332222	73604821
Juan Alvarado	Av/Buenos Aires o'3 de mayo	126629028	22839003
Mauricio Copa	Av/Buenos Aires o'3 de mayo	1452662773	73624593
Rene Jimenez	av/ America 2ª Villa Fatima	763324444	2232866

**Figura 3.18: Pantalla– ABM de Proveedores**  
Fuente: Elaboración propia

**Reabastecer Inventario**

Buscar producto por nombre o por código:

**Lista de Reabastecimiento**

Código	Cantidad	Unidad	Producto	Precio Unitario	Precio Total	
4	3	5	Casco	\$ 90	\$ 90	<input type="button" value="Cancelar"/>
5	3	6	Botas	\$ 70	\$ 210	<input type="button" value="Cancelar"/>
7	1	5	Extintore-Polvo-Químico	\$ 250	\$ 250	<input type="button" value="Cancelar"/>
8	2	5	Señalamientos Restrictivos	\$ 50	\$ 100	<input type="button" value="Cancelar"/>

**Resumen**

Proveedor:

Efectivo:

Subtotal: \$ 566

IVA: \$ 85

Total: \$ 650

**Figura 3.19: Pantalla– Panel de compras o Reabastecimiento**  
Fuente: Elaboración propia

### 3.2.3.4 CUARTA ITERACIÓN

**Inventario de Productos**

Página 1 de 1

Código	Nombre	Disponible	
4	Casco	12	<input type="button" value="Historial"/>
5	Botas	15	<input type="button" value="Historial"/>
6	Señalización de Incendios	6	<input type="button" value="Historial"/>
7	Extintore-Polvo-Químico	8	<input type="button" value="Historial"/>
8	Señalamientos Restrictivos	10	<input type="button" value="Historial"/>
9	Chaleco de visibilidad	1	<input type="button" value="Historial"/>
11	Conos de Señalización para tráfico	11	<input type="button" value="Historial"/>

Límite:

**Figura 3.20: Pantalla– Inventario**  
Fuente: Elaboración propia

**Reporte de Ventas**

Id	Subtotal	Descuento	Total	Fecha
14	\$ 980.00	\$ 0.00	\$ 980.00	2017-06-27 02:22:29
10	\$ 100.00	\$ 0.00	\$ 100.00	2017-06-24 18:47:24
8	\$ 220.00	\$ 20.00	\$ 200.00	2017-06-05 09:45:16

**Total de ventas: \$ 1,280.00**

**Figura 3.21: Pantalla– Reporte de Inventarios**  
Fuente: Elaboración propia

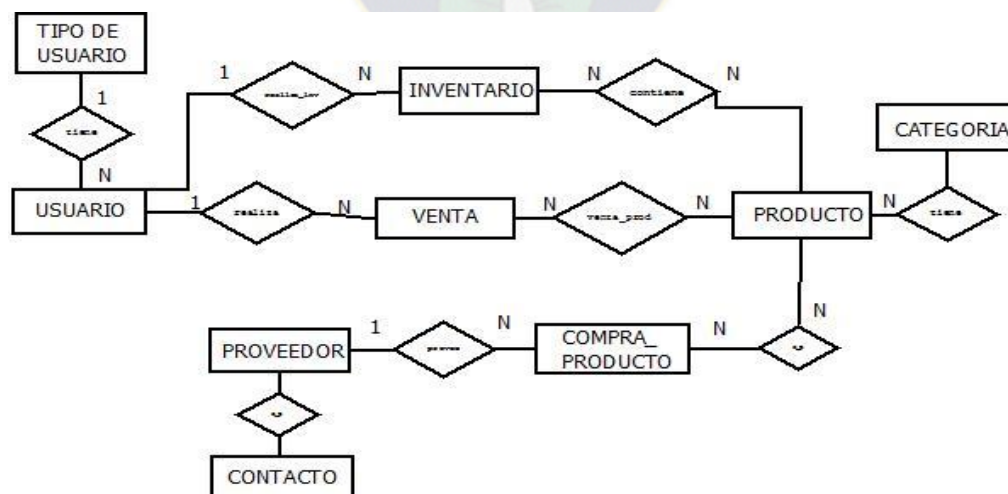
Id	Subtotal	Descuento	Total	Fecha
14	\$ 980.00	\$ 0.00	\$ 980.00	2017-06-27 02:22:29
10	\$ 100.00	\$ 0.00	\$ 100.00	2017-06-24 18:47:24
8	\$ 220.00	\$ 20.00	\$ 200.00	2017-06-05 09:45:16
<b>Total de ventas: \$ 1,280.00</b>				

**Figura 3.22: Pantalla– Reporte de Ventas**  
Fuente: Elaboración propia

### 3.2.4 MODELO ENTIDAD / RELACIÓN

El Modelo Entidad-Relación es un método que permite diseñar esquemas para su implementación en un gestor de Base de Datos. El modelo se representa a través de diagramas y está compuesto por varios elementos.

Este modelo será la base de la estructura del sistema, será la plataforma para definir el comportamiento de los procesos y funciones que compongan la capa del controlador y definirá los formularios de registros de la capa de la vista. Para obtener el Modelo Relacional nos basamos en el Modelo Entidad-Relación.



**Figura3.23: Modelo Entidad- Relación**  
Fuente: Elaboración propia



### 3.2.5 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.

#### 3.2.5.1 PRUEBAS DE ACEPTACIÓN

Este tipo de pruebas fueron realizadas para cada entrega del software en las distintas iteraciones que se tuvo, ya que fueron definidas en el reverso de cada historia de usuario.

*Adición y edición de Usuario:* Esta prueba busca encontrar todo tipo de errores que se generen durante el proceso de registro de un usuario nuevo en el sistema, ya sea en el llenado del formulario de registro con datos no válidos o también por la falta de datos obligatorios.

PRUEBA DE ACEPTACIÓN	
Número: 1	Historia de Usuario: H8: Usuarios
Nombre: Verificación del ingreso de usuario a sistema	
<b>Descripción:</b> <ul style="list-style-type: none"><li>• Controlar que el nombre de usuario sea único.</li><li>• Controlar que no se permita registrar el formulario de Usuario Nuevo, si es que los campos obligatorios no están presentes o llenados como corresponde.</li><li>• Validar los campos numéricos para evitar errores en el sistema.</li><li>• Mostrar un mensaje de confirmación una vez que se haya hecho el registro correctamente.</li></ul>	
<b>Entrada/ Pasos de Ejecución:</b> El gerente, el administrador y encargado de ventas podrán ingresar al sistema previa autenticación.	
<b>Resultado Esperado:</b> El sistema responde al ingreso de datos, usuario y password, ingresando al menú principal del sistema, para empezar a interactuar con el mismo.	
Evaluación de la Prueba: Aceptada	

**Tabla 3.28: Verificación de ingreso de usuario**

Fuente: Elaboración propia

*Administración de Productos:* Esta prueba busca encontrar todo tipo de errores que se generen durante el proceso de administración de productos, ya que los datos de precios, descuentos y categorías se configuran en este módulo y estos datos serán utilizados al momento de hacer transacciones de ventas o compras.

PRUEBA DE ACEPTACIÓN	
<b>Número: 2</b>	<b>Historia de Usuario:</b> H1: Productos
<b>Nombre:</b> Verificación de registro de productos	
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Verificar que los precios de los productos sean correctos.</li> <li>• Controlar que los campos obligatorios sean debidamente llenados, mostrando alertas y señalando los campos que deben ser ingresados.</li> <li>• Validar los campos numéricos para evitar errores en el sistema.</li> <li>• Mostrar un mensaje de confirmación una vez que se haya hecho el registro correctamente.</li> </ul>	
<b>Entrada/ Pasos de Ejecución:</b> El personal autorizado para el registro de productos proporcionando datos de ingreso de productos al sistema, el sistema actualiza la información dentro de la base de datos.	
<b>Resultado Esperado:</b> El sistema proporciona reportes propios y generales sobre los productos registrados.	
<b>Evaluación de la Prueba:</b> Aceptada	

**Tabla 3.29: Verificación de registro de productos**

Fuente: Elaboración propia

*Administración de ventas:* Esta prueba busca encontrar todo tipo de errores que se generen durante el proceso de ventas guardado en la base de datos, ya sea para la validez de los datos almacenados y la edición, si es posible, de algún dato guardado incorrectamente.

PRUEBA DE ACEPTACIÓN	
<b>Número: 3</b>	<b>Historia de Usuario:</b> H3: Ventas
<b>Nombre:</b> Verificación del módulo de ventas	
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Controlar que la venta realizada funcione correctamente mostrando los datos registrados.</li> <li>• Controlar que las ventas ejecuten el proceso de salida de inventario de cada producto</li> </ul>	

<p>que es vendido.</p> <ul style="list-style-type: none"> <li>• Controlar que los campos obligatorios sean debidamente llenados.</li> <li>• Validar los campos numéricos para evitar errores en el sistema.</li> <li>• Mostrar un mensaje de confirmación una vez que se haya el registro correctamente.</li> </ul>
<p><b>Entrada/ Pasos de Ejecución:</b></p> <p>El personal encargado de ventas alimenta el sistema con datos diarios acerca de las cantidades de productos vendidos, el sistema almacena esta información en la base de datos.</p>
<p><b>Resultado Esperado:</b></p> <p>El sistema registra la venta realizada.</p>
<p><b>Evaluación de la Prueba:</b> Aceptada</p>

**Tabla 3.30: Verificación de módulo de ventas**

Fuente: Elaboración propia

*Administración de compras o abastecimiento:* Esta prueba busca encontrar todo tipo de errores que se generen durante el proceso de gestión de compras. Ya que la compra implica el ingreso de artículos a inventario, se debe tener mucho cuidado con la información que está siendo ingresada a la base de datos.

PRUEBA DE ACEPTACIÓN	
<b>Número: 4</b>	<b>Historia de Usuario:</b> H6: Compra o abastecimiento
<b>Nombre:</b> Verificación del módulo de compra o abastecimiento	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>• Verificar el ingreso de los artículos a inventario de cada producto comprado.</li> <li>• Controlar que los campos obligatorios sean debidamente llenados, mostrando alertas y señalando los campos que deben ser ingresados</li> <li>• Validar los campos numéricos para evitar errores en el sistema.</li> <li>• Mostrar un mensaje de confirmación una vez que se haya el registro correctamente.</li> </ul>	
<p><b>Resultado Esperado:</b></p> <p>El sistema registra la compra realizada.</p>	
<b>Evaluación de la Prueba:</b> Aceptada	

**Tabla 3.31: Verificación de módulo de compra o abastecimiento**

Fuente: Elaboración propia

## **CALIDAD Y SEGURIDAD**

### **4.1 INTRODUCCIÓN**

En el presente capítulo se realizará y se comprobará la calidad al igual que la seguridad del software haciendo uso de 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 usara la técnica ISO 9126.

#### **4.2.1 TÉCNICA 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

##### **4.2.1.1 USABILIDAD**

La usabilidad consiste de un conjunto de atributos que permite evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema, es decir realizar una serie de preguntas que permiten ver cuán sencillo, fácil de aprender y manejar es para los usuarios. Esta comprensión por parte de los usuarios con relación al sistema evalúa los siguientes pasos:

- Entendimiento
- Aprendizaje
- Operabilidad

- Atracción
- Conformidad de uso

En la siguiente tabla se observa estos criterios en niveles de porcentajes a los que llego el sistema en cuanto a su comprensibilidad, para el usuario, y posteriormente se da el porcentaje final de usabilidad del sistema. Se realizan encuestas a los usuarios finales sobre el manejo, la comprensión y la facilidad de aprender el sistema para medir la usabilidad según la siguiente tabla 4.1:

PREGUNTAS	RESPUESTAS		PORCENTAJES
	SI	NO	
¿El acceso al sistema es complicado?	0	10	100%
¿Son comprensibles las respuestas del sistema?	1	9	90%
¿Son complicadas los procesos que realiza el sistema?	10	0	100%
¿El sistema tiene interfaces entendibles?	9	1	90%
¿La interfaz del sistema es agradable a la vista?	10	0	100%
¿Son satisfactorias las respuestas que el sistema devuelve?	8	2	80%
¿El sistema reduce su tiempo de trabajo?	9	1	90%
¿El sistema satisface las necesidades que usted requiere?	9	1	90%
¿Utiliza el sistema con facilidad?	9	1	90%
<b>PROMEDIO</b>			<b>92%</b>

**Tabla 4.1: Encuesta de usabilidad del sistema**

Fuente: Elaboración propia

De acuerdo a los datos obtenidos en la tabla de usabilidad, se concluye que el sistema tiene una usabilidad del 92%. Que de cada 100 personas que lleguen a usar el sistema 92

personas indican que el sistema es fácil de manejar y es comprensible.

#### 4.2.1.2 FUNCIONALIDAD

La funcionalidad examina si el sistema satisface los requisitos funcionales esperados. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario. Se cuantifica el tamaño y la complejidad del sistema en términos de las funciones de usuario, puede ser valorado mediante el Punto Función. Se basa en la contabilización de cinco parámetros los cuales se desarrollan a continuación:

- **Número de entradas de usuario:** Se refiere a cada entrada de control del usuario que proporciona diferentes datos al sistema.
- **Número de salidas de usuario:** Se refiere a cada salida de información que proporciona el sistema al usuario, entre estos están: informes, pantallas, mensajes de errores, entre otros.
- **Número de peticiones de usuario:** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salidas interactivas
- **Número de Archivos:** Se cuenta archivos maestro lógico, estos pueden ser: grupo lógico de datos que sean parte de una base de datos, o un archivo independiente.
- **Número de interfaces externas:** Se cuenta las interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema.

Para calcular el Punto Función se utilizará la siguiente fórmula:

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

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

Parámetro	Cuenta	Factor de ponderación	Subtotal
Entradas de Usuario	18	*4	72
Salidas de Usuario	20	*4	80
Peticiones de Usuario	21	*3	63
Archivo lógico interno	18	*10	180
Archivo de interfaz Externa	0	*5	0
<b>Total</b>			<b>395</b>

**Tabla 4.2: Conteo de parámetros del PF**

Fuente: Elaboración propia

Cada organización que utiliza métodos de puntos desarrolla criterios para determinar si una entrada es denominada simple o compleja. Los valores de Fi, se obtiene de los resultados de la siguiente tabla, bajo las ponderaciones descritos en la escala.

IMPORTANCIA	0	1	2	3	4	5
ESCALAS	Sin importancia	Incremental	Moderado	Medio	Significativo	Esencial
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
2.- ¿Se requiere comunicación de datos especializadas para transferir información a la aplicación u obtenerlas de ellas?						X
3. ¿Existe funciones de procesos distribuidos?				X		



4. ¿Es crítico el rendimiento?				X		
5.- ¿El sistema web será ejecutado en el S.O. Actual?					X	
6.- ¿Se requiere una entrada interactiva para el sistema?					X	
7.- ¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?						X
8.- ¿Se actualiza los archivos de forma interactiva?						X
9.- ¿Son complejas las entradas, salidas, los archivos o las peticiones?				X		
10.- ¿Es complejo el procesamiento interno del sistema?						X
11.- ¿Se ha diseñado el código para ser reutilizado?						X
12.- ¿Están incluidas en el diseño la conversión y la instalación?				X		
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?				X		
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?						X
<b>Cuenta total</b>	<b><math>\Sigma(f_i)=58</math></b>					

**Tabla 4.3: Tabla de ajuste de complejidad**

Fuente: Elaboración propia

Para el ajuste se utiliza la ecuación:

$$PF = \text{cuenta total} * (\text{grado de confiabilidad} + \text{Tasas de Error} * \Sigma(f_i))$$

$$PF = 395 * (0.65 + 0.01 * 58)$$

$$PF \text{ obtenida} = 485.85$$

Para el ajuste se utiliza la ecuación para hallar el punto función ideal al 100% de los

factores que seria 70:

$$PF = 395 * (0.65 + 0.01 * 70)$$

PF ideal = 533.25

Calculando del % de funcionalidad real:

PF real = PF obtenida / PF ideal

$$\text{FUNCIONALIDAD} = 485.85/533.25*100$$

$$\text{FUNCIONALIDAD} = 91\%$$

Interpretando, el sistema tiene una funcionalidad o utilidad del 91% para la empresa, lo que indica que él es sistema cumple con los requisitos funcionales de forma satisfactoria.

#### 4.2.1.3 CONFIABILIDAD

La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos usados, es decir, representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la privacidad de que un sistema presente fallas:

- **Comportamiento con respecto al tiempo:** Atributos de software relativos a los tiempos de respuesta y de procesamiento de los datos.
- **Comportamiento con respecto a Recursos:** Atributos software relativo a la cantidad de recursos usados y la duración de su uso en la realización de funciones.

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:

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

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

Dónde:

Funcionalidad = 0.91

$\lambda = 0.01$  (es decir 1 error en cada 6 ejecuciones)

t=12 meses.

Hallamos la confiabilidad del sistema:

$$(12)=0.91 * e^{-1/6*12}$$

$$F(12)=0.123$$

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

$$\begin{aligned} P(T>t) &= 1 - F(t) \\ P(T>t) &= 1 - 0.123 \\ P(T>t) &= 0.877 \end{aligned}$$

Con este resultado podemos decir que la probabilidad que el sistema no presente fallas es de 0.88.

$$\textbf{CONFIABILIDAD} = 88\%$$

Para concluir decimos que el sistema tiene un grado de confiabilidad del 88% durante los próximos 12 meses, es decir, que de cada 100 ejecuciones existen 12 fallas de elección del sistema. Por lo tanto, es una aceptación confiable y aceptable de parte del sistema.

#### 4.2.1.4 MANTENIBILIDAD

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.

Se determina la siguiente función (IMS):

$$IMS = Mt - (Fc + Fa + FE) / Mt$$

Dónde:

Mt: Numero de módulos total de la versión actual

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

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

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

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

La interpretación a este resultado establece un **88%**, lo que indica que no requiere de mantenimiento inmediatamente.

***Mantenibilidad=88%***

#### **4.2.1.5 PORTABILIDAD**

La portabilidad se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- **Conformidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.
- **Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo.
- **Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.

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%***

#### 4.2.1.6 RESULTADOS

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente. La calidad está directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

CARACTERÍSTICAS	RESULTADO
Usabilidad	92%
Funcionalidad	91%
Confiabilidad	88%
Mantenibilidad	88%
Portabilidad	100%
<b>Evaluación de calidad total</b>	<b>91.8%</b>

**Tabla 4.4: Resultados**  
Fuente: Elaboración propia

Finalmente, de acuerdo a las escalas de aceptación, indica que los valores de preferencia se encuentran en el rango de 60-100. La calidad del sistema corresponde al 92%, lo que se interpreta como la satisfacción que tiene un usuario al interactuar con el sistema.

**Evaluación de calidad total=92%**

### 4.3 SEGURIDAD DE SOFTWARE

#### 4.3.1 SEGURIDAD A NIVEL DE USUARIO

A nivel de usuario se hará el uso 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.

#### 4.3.2 SEGURIDAD A NIVEL DE BASE DE DATOS

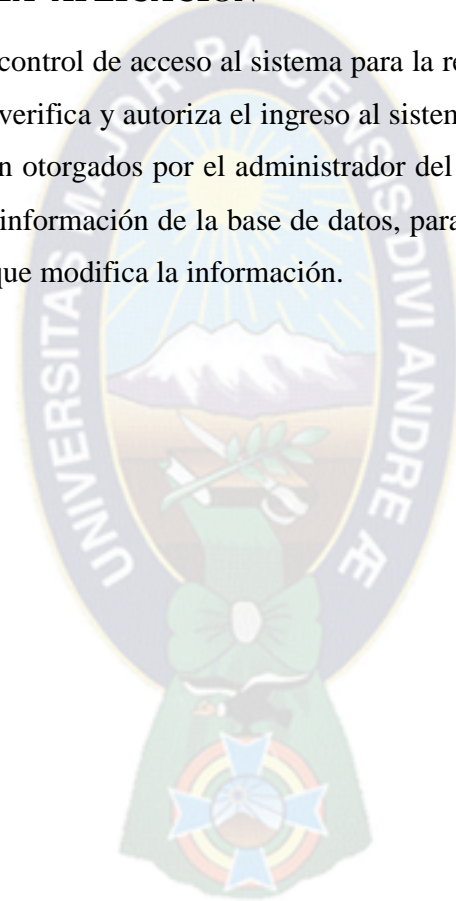
Se usó como base de datos MySQL. En cuanto a la forma de resguardo se realizó los siguientes puntos:

- 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.

#### **4.3.3 SEGURIDAD DE LA APLICACIÓN**

Se desarrolla un módulo de control de acceso al sistema para la restricción del acceso a usuario no autorizado. Este módulo verifica y autoriza 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.





## ANÁLISIS COSTO 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.

#### 5.2.1 COCOMO II

El Modelo de Construcción de Costo COCOMO (Constructive Cost MOdel), es un modelo empírico se utiliza para la estimación de costos de un software. Incluye tres sub-modelos, cada uno ofrece un nivel de detalle y aproximación. En los tres modelos de estimación se utilizan las tres siguientes ecuaciones:

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

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

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

Dónde:

E: esfuerzo requerido por el proyecto, en mes.

D: Tiempo requerido por el proyecto, en meses.

P: Número de personas requerido por el proyecto.

a, b, c y d: constantes con valores definidos, según cada sub-modelo.

KLDC: Cantidad de líneas de código, en miles.



$m(X)$ : multiplicador que depende de 15 atributos.

A la vez cada modelo se divide en modos, estos modos son:

- **Modo orgánico:** es un pequeño grupo de programadores experimentados desarrollando software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- **Modo semi – libre o semi – acoplado:** 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, siendo difícil basarse en la experiencia puesto que puede no haberla.

<b>PROYECTO DE SOFTWARE</b>	<b>ab</b>	<b>bb</b>	<b>cb</b>	<b>db</b>
<i>Orgánico</i>	2.1	1.01	2.2	0.34
<i>Semi – acoplado</i>	3.0	1.12	2.5	0.35
<i>Empotrado</i>	3.6	1.20	2.5	0.32

**Tabla 5.1: Coeficiente a y c y los exponentes b y d**

Fuente: (Pressman, 2002)

### 5.2.2 COSTO DEL SOFTWARE DESARROLLADO

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

$$PF=485.85$$

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:

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
<b>PHP</b>	<b>11.00</b>	<b>29</b>
VISUAL C++	9.50	34

**Tabla 5.2: Conversión de puntos de función siguiente**

Fuente: (Pressman, 2002)

Calculando las líneas de código en la siguiente ecuación tenemos:

$$LDC = PF * Factor\ LDC/PF = 485.85 * 29$$

$$\mathbf{LDC = 14089.65}$$

Para convertirlo a KLDC dividimos LDC entre 1000. Calculando el número de líneas distribuidas en el sistema KLCD se tiene:

$$KLDC = total\ LCD / 1000$$

$$KLDC = 14089.65 / 1000$$

$$\mathbf{KLDC = 14.089}$$

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

A continuación, haremos el cálculo del esfuerzo necesario para la programación del sistema. La ecuación que nos ayudara a hallar el esfuerzo, viene dada de la siguiente manera:

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

Dónde:

E: Es el esfuerzo expresado en personas por mes.

a, b: Son constantes empíricas.

KLDC: Es un número estimado de código fuente en miles distribuidas.

Como nuestro proyecto es del tipo Orgánico utilizaremos  $a = 2.4$  y  $b = 1.05$ . Reemplazando estos valores en la ecuación, tenemos:

$$E = 2.4 * (14.089)^{1.05} = 38.59$$

Calculando D con  $c = 1.05$  y  $d = 0.38$  tenemos:

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

$$D = 1.05 * (38.59)^{0.38}$$

$$\mathbf{D = 4.2}$$

El proyecto deberá tener un desarrollo de aproximadamente 5 meses.

Para calcular el personal requerido, en este caso 1 número de programadores se obtiene con la siguiente fórmula:

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

Reemplazando los datos ya conocidos se tiene:

$$P = 38.59 / 5 = 7.71$$

$$\mathbf{P = 8 [Programadores]}$$

El salario promedio de un programador oscila entre los 2000 y 4000 Bs., en nuestro caso tomaremos un promedio con un valor de 3000 Bs, a partir de este monto podemos calcular el costo del software desarrollado:

Costo del software desarrollado = Número de programadores \* salario de un programador

$$\text{Costo del software desarrollado por persona} = 8 * 3000$$

$$\text{Costo del software desarrollado por persona} = 24000 \text{ Bs}$$

Como el desarrollo de software se lo estima en 5 meses tendremos

$$\text{Costo total de desarrollo} = \text{Costo del software desarrollado} * \text{Número de meses}$$

$$\text{Costo total de desarrollo} = 24000 * 5$$

$$\mathbf{\text{Costo total de desarrollo} = 120000 \text{ Bs}}$$

### 5.2.3 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.

### 5.2.4 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:

DETALLE	IMPORTE (\$us)
Análisis y diseño del proyecto	250
Material de escritorio	55
Internet	80
Otros	20
<b>Total</b>	<b>405</b>

**Tabla 5.3: Costos de elaboración del proyecto**

Fuente: Elaboración propia

### 5.2.5 COSTO TOTAL DEL SOFTWARE

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

DETALLE	IMPORTE (\$us)
Costo de desarrollo	17191.98
Costo de elaboración del software	405
<b>Total</b>	<b>17596.98</b>

**Tabla 5.4: Costo total de producto de software**

Fuente: Elaboración propia

Por lo tanto, el costo total del proyecto es igual 17596.98 \$ su equivalencia en bolivianos es 122826.9 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 CALCULO 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.

$$VAN = \sum \text{Ganancias} / (1+k)^n - I_0$$

Dónde:

VAN: Valor Actual Neto

Ganancias: ingreso del flujo anual

$I_0$ : es el valor del desembolso inicial de la inversión n: número de periodo

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 11%, ya que es la tasa actual de interés de préstamo en las entidades financieras. Para calcular el valor del VAN se tiene lo siguiente:

$$\text{Inversión} = 122826.9 \text{ Bs}$$

$$TD = 11\%$$

Los valores de ganancia esperados se detallan en la siguiente tabla.

Tiempo	1° Año	2° Año	3° Año	4° Año
Flujo de caja neto	42000	43000	44000	45000

**Tabla 5.5: Cantidad nominal por año**

Fuente: Elaboración propia

Entonces para hallar el Van tenemos:

$$VAN = 42000/(1+0.11)^1 + 43000/(1+0.11)^2 + 44000/(1+0.11)^3 + 45000/(1+0.11)^4 - 122826.9$$

$$VAN = 42000/1.11 + 43000/1.23 + 44000/1.36 + 45000/1.52 - 122826.9$$

$$VAN = 37837.84 + 34959.35 + 32352.94 + 29605.26 - 122826.9$$

$$VAN = 11928.49$$

Por lo tanto, la ganancia que se espera recibir del sistema al cabo de 4 años es de 11928.49 Bs.

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

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 5.6: Interpretación del VAN**

Fuente: Elaboración propia

Como el resultado que obtuvimos es de  $VAN = 11928.49$  podemos afirmar que nuestro proyecto es rentable.

### 5.3.2 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 periodo

$k$ : i: tasa de interés de ahorro

$n$ : Es el número de periodos considerados

Entonces para hallar el TIR se necesita la inversión de 122826.9 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, se halla la tasa de descuento:

$$0=42000/(1+TIR)^1+43000/(1+TIR)^2+44000/(1+TIR)^3+45000/(1+TIR)^4-122826.9$$

Con la ayuda de la hoja de cálculo podemos resolver la ecuación y hallar el valor de i.

	A	B	C
12			
13	-122826,9		
14	42000		
15	43000		
16	44000		
17	45000		
18	15%		
19			

**Figura 5.1: Calculo del TIR**

Fuente: Elaboración propia

Lo cual indica que el proyecto está en condiciones de retornar 15% de la inversión cada año, además si la tasa fuera mayor de 15% el proyecto empezaría a no ser rentable. Pues el BNA empezaría a ser menor que la inversión Y si la tasa fuera menor al 11% (como en el caso del VAN donde la tasa es de 11%), 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.

$$TIR \text{ a 4 años} = 15\%$$

Lo cual nos indica que el proyecto sea siempre rentable la tasa de descuento TD no debe ser mayor a 15%

#### 5.4 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:

$$C/B = 122826.9/11928.49$$

$$C/B = 10.29$$

Por tanto, por cada boliviano invertido la empresa tiene una ganancia de 10.29 Bs



## CONCLUSIONES Y RECOMENDACIONES

### 6.1 CONCLUSIONES

Una vez finalizado el desarrollo e implementación del Sistema de control de compra, venta e inventarios para la empresa Protec, 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 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.
- Se facilitó el proceso de compras de los artículos en función de los productos vendidos, esto evita que la empresa compre artículos que no se venden y se llene el stock.
- Se tiene un registro con los datos necesarios de los proveedores con los que mantiene contacto la empresa.
- Se tiene control de la disponibilidad de artículos y de los estados actuales de los inventarios.
- 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.

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 cliente para hacer el control adecuado a dichos procesos. Esto se logró mediante la ejecución de las fases propuestas por la metodología XP.

## 6.2 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.
- Se recomienda realizar dicho sistema orientado aplicaciones móviles.
- Se recomienda la utilización de herramientas de programación brindadas por PHP, debido a la interfaz amigable para el desarrollador.
- Se recomienda utilizar algún framework, ya que este facilita el desarrollo del producto.
- Convendría utilizar la metodología XP en proyectos cortos y medianos, para disminuir el tiempo de desarrollo.



## BIBLIOGRAFÍA

*Apache Shiro*. (2008). Recuperado el 2015, de <http://shiro.apache.org>

Arana J., 2014 “Desarrollo e implementación de un sistema de gestión de ventas de repuestos automotrices en el almacén de auto repuestos eléctricos marcos en la parroquia Posorja cantón Guayaquil, provincia del Guayas”, mención a Ingeniera en Sistemas, Ecuador, Universidad Estatal Península de Santa Elena Facultad de Sistemas y Telecomunicaciones Escuela de Informática Carrera de Informática

Aduviri, P. (2016). *Sistema web de control de ventas e inventarios caso: Michelline* (Proyecto de Grado). Universidad Mayor de San Andrés, La Paz, Bolivia.

Bustamante D., Rodríguez J., 2014, *Metodología Actual Metodología XP*, Barinas, Universidad Nacional Experimental de los Llanos Occidentales Ezequiel Zamora, Ingeniería en Informática.

Beck K., 2002 *una explicación de la programación extrema aceptar el cambio*, Madrid, 189 pag.

Condori, R. (2015). *Sistema web de control de compras, venta e inventarios caso Librería de la Asociación cristiana Pan de Vida* (Proyecto de grado). Universidad Mayor de San Andrés, La Paz, Bolivia.

Chiri, C. (2014). *Sistema de entradas y salidas e inventarios caso BOLITAL S.R.L.* (Proyecto de grado). Universidad Mayor de San Andrés, La Paz, Bolivia.

Ceri S., 2000 Lenguaje de Modelado Web (WebML): un lenguaje para diseñar sitios Web, San Francisco Morgan Kaufman Publisher 2000 216p

Carmona P 2008 *Propuesta y construcción de una ontología para lenguajes de modelado gráfico*, Especialización en Sistemas Universidad Nacional de Colombia Sede Medellín, Escuela de Sistemas

*Extreme Programming*. (2015). Obtenido de <http://www.extremeprogramming.org/rules/crccards.html>

Echeverry I, Delgado I 2007, *Metodología Ágil de Desarrollo de Software* 2007, 305 pag.

Gomez, D., Aranda , E., & Fabrega, J. (3 de Junio de 2010). *Programación Extrema*. Cali, Colombia. Obtenido de <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>

Gonzales Alvarán, L. F., Reyes Gamboa, A. X., & Vásquez Echavarría, G. H. (2010). *Diseño de Aplicaciones Basadas en Arquitecturas Orientadas a Servicios utilizando WebML*. Antioquía: Instituto Internacional de Informática y Sistemas. Recuperado el 10 de Noviembre de 2015, de <http://www.iiis.org/CDs2009/CD2009CSC/CISCI2009/PapersPdf/C553FU.pdf>

Joskowicz, J. (12 de Septiembre de 2011). *Reglas y Prácticas en Xtreme Programing*. Vigo, Madrid, España.

Jarquín, P. S. (2015). *Ingeniería Web*. Publicación PDF en <http://sevillajarquín.udem.edu.ni/>.

Kendall, K. (2012). *Analisis y diseño de sistemas*.

Loraine G. 2012, *Metodologías ágiles y desarrollo basado en conocimiento*, optar el título de Especialista en Ingeniería de Software, Universidad Nacional de La Plata, Facultad de Informática.

Laboratorio Nacional de Calidad del Software. (Marzo de 2009). *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. Madrid, España. Recuperado el 11 de Agosto de 2015

Monroy, E. J. (1 de Diciembre de 2013). *Metodologías de Desarrollo Ágil*. Obtenido de [itamod.blogspot.com](http://itamod.blogspot.com).

Mestras, J. P. (13 de 11 de 2015). *El Patrón Modelo-Vista-Controlador (MVC)*. Obtenido de Universidad Complutense de Madrid - Carrera de Informática: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>

- Pressman Roger, (2006). *Ingeniería de software*. ed. Mexico, McGraw-Hill. 968p.
- Sommerville, (2005). *Ingeniería de software*. Ed. España. Pearson Addison Wesley.
- Stefano Ceri, P. F. (2012). *Web Modeling Language (WebML): a modeling language for designing Web sites*. Milan: Dipartimento di Elettronica e Informazione, Politecnico di Milano. Recuperado el 10 de Noviembre de 2015, de <http://www.webml.org/webml/upload/ent5/1/www9.pdf>
- Penadés, M. d., & Letelier , P. (2004). *Métodologías Ágiles para el desarrollo de software: eXtreme Programming (XP)*. Recuperado el 23 de Julio de 2015, de . <http://www.willydev.net/descargas/masyxp.pdf> .
- Ramírez, J. (2006). *Inventarios*. MARACAIBO.
- Robles, G. (2002). *Software Libre*. Obtenido de <http://es.tldp.org/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.pdf>
- Real Academia Española . (24 de Agosto de 2015). *Diccionario de la Real Academia Española de la Lengua*. Obtenido de DRAE LEMA: <http://lema.rae.es/drae/srv/search?id=oeUHbdnJiDXX2haGF7Pl>
- Rojas, D.(2014). *Sistema web de compras, ventas e inventario para la empresa EDDYMAR*. (Proyecto de Grado). Universidad Mayor de San Andrés, La Paz, Bolivia.
- Science Made Simple*. (2015). Obtenido de [http://www.sciencemadesimple.com/scientific\\_method.html](http://www.sciencemadesimple.com/scientific_method.html)