

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS

ESCUELA PROFESIONAL DE INGENIERIA EN COMPUTACION E INFORMATICA



**TESIS PARA OPTAR EL TITULO PROFESIONAL DE
INGENIERO EN COMPUTACION E INFORMATICA**

*“DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA
METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA
EMPRESA AGRO MARKET PERU S.A.C”*

Autor: Bach. Altamirano Fernández Jorge Luis

Asesor: Ing. Terán Santa Cruz Franklin Edinson

LAMBAYEQUE, 19 DE DICIEMBRE 2017

*DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,”*

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS

ESCUELA PROFESIONAL DE INGENIERIA EN COMPUTACION E INFORMATICA

***“DESARROLLO DE UN SISTEMA DE INFORMACION
HACIENDO USO DE LA METODOLOGIA XP PARA LA
GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA
EMPRESA AGRO MARKET PERU S.A.C.”***

TESIS PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO EN COMPUTACION E INFORMATICA

BACH. ALTAMIRANO FERNÁNDEZ JORGE LUIS
AUTOR

ING. TERÁN SANTA CRUZ FRANKLIN EDINSON
ASESOR

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS

ESCUELA PROFESIONAL DE INGENIERIA EN COMPUTACION E INFORMATICA

***“DESARROLLO DE UN SISTEMA DE INFORMACION
HACIENDO USO DE LA METODOLOGIA XP PARA LA
GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA
EMPRESA AGRO MARKET PERU S.A.C.”***

TESIS PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO EN COMPUTACION E INFORMATICA

APROBADO POR EL JURADO:

M.SC. ING. VALDIVIA SALAZAR CARLOS
PRESIDENTE

ING. CHAYÁN COLOMA ALEJANDRO
SECRETARIO

ING. FUENTES ADRIANZÉN DENNY
VOCAL

DEDICATORIA

Al creador del universo por darme la vida y la fortaleza para continuar, con toda la humildad de mi corazón dedico mi trabajo de investigación en primer lugar a Dios.

Por aquellos que soñaron con un mejor porvenir para su generación, aquellos que trabajaron de sol a sol para sustentar mi formación, por aquellos que forjaron en mí el deseo de superación. Con todo mi amor y respeto dedico este trabajo a mis queridos padres, Juanito Altamirano y Cesarina Fernández.

A mi esposa Carmencita por mostrarme siempre su amor, apoyo incondicional y la motivación constante por que sea una mejor persona y profesional.

Bach. Jorge Luis Altamirano Fernández

AGRADECIMIENTO

Agradezco a Dios por guiarme en todo mi camino y dame las fuerzas para superar los obstáculos y dificultades a lo largo de toda mi vida

A mi asesor por el apoyo incondicional en la elaboración del presente proyecto de tesis.

Al personal de la empresa Agro Market Perú S.A.C por dedicarme un espacio de su tiempo y proporcionarme la información necesaria para lograr los objetivos de ésta investigación.

A todos mis maestros que me formaron y me transmitieron sus conocimientos a lo largo de mi formación profesional.

A todas las personas que apoyaron directa e indirectamente en el desarrollo de este proyecto.

RESUMEN

La investigación planteada en esta tesis está enfocada en el desarrollo de un sistema de información para mejorar los procesos de ventas, compras y almacén de la empresa Agro Market Perú S.A.C, para lo cual se tuvo que investigar cómo se realizan sus procesos y así plantear una solución informática.

Para tal fin haremos uso de la metodología XP (Xtreme Programming), el potente lenguaje de programación Java con el paradigma de programación orientada a objetos y haciendo uso del sistema gestor de bases de datos MySQL.

El presente informe Tesis se planteó el desarrollo de un sistema de información para la empresa Agro Market Perú S.A.C con el objetivo de gestionar los procesos de ventas, compras y almacén, logrando un posicionamiento competitivo en el ámbito regional y satisfacer las necesidades de sus clientes.

Para el desarrollo del sistema de información se realizó varios procedimientos como la recopilación de la información, revisión de archivos físicos de la empresa y entrevistas con el personal involucrado en los procesos. Con dicha información recopilada se planteó las soluciones a la problemática.

La tesis planteada posee un tipo de investigación Tecnológica Formal, se analizó en función a dos variables (Independiente y Dependiente), y el planteamiento de hipótesis, aplicada porque utilizaré programas en el desarrollo del sistema de información.

PALABRAS CLAVE: Sistema de información, metodología XP, Java, bases de datos, gestionar procesos.

ABSTRACT

The research presented in this thesis is focused on the development of an information system to better the sales, purchasing and warehouse processes of the company Agro Market Peru SAC, for which it was necessary to investigate how its processes are carried out and to propose a Computer solution.

For this purpose we will use XP methodology (Xtreme Programming), the powerful Java programming language with the paradigm of object oriented programming and making use of the system gestures of MySQL databases.

The present thesis report included the development of an information system for Agro Market Perú S.A.C with the objective of managing the sales, purchasing and warehousing processes, achieving a competitive position in the regional area and meeting the needs of its customers.

For the development of the information system, a number of procedures were carried out, such as the collection of information, review of the company's physical files and interviews with personnel involved in the processes. With this information collected, solutions to the problem were raised.

The thesis presented has a type of Formal Technological research, was analyzed according to two variables (Independent and Dependent), and the hypothesis approach, applied because I will use programs in the development of the information system.

KEY WORD: System information, XP methodology, Java, databases, manage processes.

INDICE GENERAL

DEDICATORIA	4
AGRADECIMIENTO	5
RESUMEN	6
ABSTRACT	7
INDICE GENERAL.....	8
INDICE DE TABLAS	12
INDICE DE FIGURAS	14
INTRODUCCIÓN	16
CAPITULO I	17
DATOS GENERALES DE LA ORGANIZACIÓN	17
1.1.- DESCRIPCIÓN DE LA ORGANIZACIÓN	18
1.2.-MISIÓN, VISIÓN Y OBJETIVOS DE LA ORGANIZACIÓN	18
1.2.1. MISIÓN	18
1.2.2. VISIÓN	18
1.2.3. OBJETIVOS.....	18
1.3.- ESTRUCTURA ORGÁNICA	19
CAPITULO II	20
PROBLEMÁTICA DE LA INVESTIGACIÓN	20
2.1.- REALIDAD DE LA PROBLEMÁTICA	21
2.1.1. PLANTEAMIENTO DEL PROBLEMA	21
2.2.- FORMULACIÓN DE PROBLEMA.....	23
2.3.- JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN	23
2.3.1 JUSTIFICACIÓN ECONÓMICA.....	23
2.3.2 JUSTIFICACIÓN TECNOLÓGICA	23
2.3.3 JUSTIFICACIÓN ACADÉMICA.....	24
2.3.4 JUSTIFICACIÓN OPERATIVA	24
2.3.5 IMPORTANCIA	24
2.4.- OBJETIVOS DE LA INVESTIGACIÓN	25
2.4.1. OBJETIVO GENERAL.....	25
2.4.2. OBJETIVOS ESPECÍFICOS.....	25
2.5.- LIMITACIONES DE LA INVESTIGACIÓN	26
CAPITULO III	27

*DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,*

MARCO METODOLÓGICO.....	27
3.1.- TIPO DE INVESTIGACIÓN	28
3.2. HIPÓTESIS	28
3.3.- VARIABLES.....	28
3.3.1. VARIABLE INDEPENDIENTE.....	28
3.3.2. VARIABLE DEPENDIENTE	28
3.4.- DISEÑO Y CONTRASTACIÓN DE HIPÓTESIS	28
CAPÍTULO IV	29
MARCO TEORICO.....	29
4.1.- ANTECEDENTES DE LA INVESTIGACIÓN	35
4.1.1. ANTECEDENTES EN EL CONTEXTO INTERNACIONAL.....	35
4.1.2. ANTECEDENTES EN EL CONTEXTO NACIONAL	36
4.1.3. ANTECEDENTES EN EL CONTEXTO LOCAL	37
4.2.- BASE TEÓRICA	38
4.2.1 RESUMEN LAS METODOLOGÍAS ÁGILES MÁS RELEVANTES.....	38
4.2.1.1 SCRUM.....	38
4.2.1.2 CRYSTAL METHODOLOGIES:.....	39
4.2.1.3 DYNAMIC SYSTEMS DEVELOPMENT METHOD (DSDM)	39
4.2.1.4 ADAPTIVE SOFTWARE DEVELOPMENT (ASD).....	39
4.2.1.5 FEATURE-DRIVEN DEVELOPMENT (FDD).....	40
4.2.1.6 METODOLOGIA RUP	40
4.2.1.7 PROGRAMACIÓN EXTREMA (XP).....	40
4.2.2 CRITERIOS DE SELECCIÓN DE LA METODOLOGÍA EMPLEADA.....	41
4.2.2.1 FORMULACION Y DEFINICION DE CRITERIOS.....	41
4.2.2.1.1 PRESUPUESTO DISPONIBLE	41
4.2.2.1.2 TAMAÑO DEL PROYECTO	42
4.2.2.1.3 TIEMPOS LIMITADOS DE ENTREGA	42
4.2.2.1.4 NECESIDAD DE DOCUMENTACION.....	43
4.2.2.1.5 PERSONAL NECESARIO	43
4.2.2.1.6 ADAPTABILIDAD, RESPUESTA A CAMBIOS	44
4.2.2.1.7 IMPOSIBILIDAD DEL CLIENTE.....	44
4.2.2.2 CUANTIFICACION DE LOS CRITERIOS.....	45
4.2.3 METODOLOGÍA XP.....	47
4.2.3.1 INTRODUCCIÓN	47
4.2.3.2 CONCEPTOS BÁSICOS	48

*DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,”*

4.2.3.3 VENTAJAS DE LA METODOLOGÍA XP	49
4.2.3.4 PRINCIPIOS BÁSICOS	50
4.2.3.5 ROLES DE LA METODOLOGÍA XP	53
4.2.3.6 ARTEFACTO DE LA METODOLOGÍA XP	54
4.2.3.7 FASES DE LA METODOLOGÍA XP	58
4.2.3.8 MEJORES PRÁCTICAS UTILIZANDO LA METODOLOGÍA XP	71
4.2.4 BASE DE DATOS MySQL	71
4.2.4.1 INTRODUCCIÓN	71
4.2.4.2 PRINCIPALES CARACTERÍSTICAS	73
4.2.5 LENGUAJE DE PROGRAMACIÓN JAVA	74
4.2.5.1 INTRODUCCIÓN	74
4.2.5.2 PRINCIPALES CARACTERÍSTICAS	75
4.2.5.3 ENTORNO DE DESARROLLO TÍPICO EN JAVA	80
4.2.6 HERRAMIENTAS DE DESARROLLO	81
4.2.6.1 MySQL WORKBENCH	81
4.2.6.2 NETBEANS IDE	83
4.2.6.3 SERVIDOR XAMPP	83
4.2.6.4 IREPORT	84
4.3.- CONCEPTOS Y DEFINICIONES	86
4.3.1 DEFINICIÓN DE LOS PROCESOS	86
4.3.1.1 PROCESO DE VENTA	86
4.3.1.2 PROCESO DE COMPRAS	86
4.3.1.3 PROCESO DE ALMACÉN	87
4.3.2 SISTEMA	87
4.3.3 SISTEMA DE INFORMACIÓN	87
4.3.4 BASE DE DATOS	87
4.3.5 PROCESO	88
4.3.6 COMPILADOR	88
4.3.7 JAVA DEVELOPMENT KIT (JDK)	88
CAPITULO V	89
DESARROLLO DE LA PROPUESTA	89
5.1. CONFIGURACIONES INICIALES PARA DESARROLLAR EL SISTEMA DE INFORMACIÓN	90
5.1.1 BASES DE DATOS	90
5.1.2 SOFTWARE	93
5.2. PLANIFICACIÓN Y DISEÑO	96

*DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,”*

5.2.1	ITERACIÓN 1	96
5.2.2	ITERACIÓN 2	104
5.2.3	ITERACIÓN 3	108
5.2.4	ITERACIÓN 4	114
5.2.5	ITERACIÓN 5	118
5.2.5	BASE DE DATOS	123
5.3.-	IMPLEMENTACIÓN DE FUNCIONALIDADES	124
5.3.1	REALIZAR UNA VENTA	124
5.3.1.1	CODIFICACION	124
5.4	SEGURIDAD EN EL SISTEMA	130
5.5	PRUEBAS DE ACEPTACIÓN DEL USUARIO FINAL	134
CAPITULO VI	136
COSTO Y BENEFICIOS	136
6.1.-	ANALISIS DE COSTOS Y BENEFICIOS	137
6.1.1.	COSTO DE SOFTWARE	137
6.1.2.	COSTO DE PERSONAL	137
6.1.3.	COSTO DE SERVICIO Y MATERIALES	138
6.1.4.	COSTOS DE HARDWARE	138
6.1.5.	COSTOS DE IMPLEMENTACIÓN	139
6.1.6.	COSTO DE MANTENIMIENTO	139
6.1.7.	RESUMEN DE COSTOS	139
6.2	RECUPERACION DE LA INVERSION	140
6.2.1	EVALUACION DEL PROYECTO EN BASE AL PERIODO DE RECUPERACION	140
6.2.2	EVALUACIÓN DEL PROYECTO EN BASE AL VALOR ACTUAL NETO.....	141
6.2.3	EVALUACIÓN DEL PROYECTO EN BASE A LA TASA DE RETORNO DE LA INVERSIÓN....	141
6.3.-	BENEFICIOS	142
6.3.1.	BENEFICIOS TANGIBLES.....	142
6.3.2.	BENEFICIOS INTANGIBLES	143
CONCLUSIONES	144
RECOMENDACIONES	146
CAPÍTULO VII	148
REFERENCIAS BIBLIOGRÁFICAS	148
BIBLIOGRAFIA	149
ANEXOS	151

INDICE DE TABLAS

<i>Tabla 1 Modelo de historia de usuario</i>	56
<i>Tabla 2 Modelo de ficha para tareas</i>	57
<i>Tabla 3 Modelo nombre de la clase</i>	57
<i>Tabla 4 Gestión de Empleado</i>	97
<i>Tabla 5 Diseño de interfaz gestión de empleado</i>	97
<i>Tabla 6 Mantenimiento de gestión de empleado</i>	98
<i>Tabla 7 Gestión de proveedor</i>	98
<i>Tabla 8 Diseño de interfaz gestión de proveedor</i>	99
<i>Tabla 9 Mantenimiento de gestión de proveedor</i>	99
<i>Tabla 10 Gestión de cliente</i>	100
<i>Tabla 11 Diseño de interfaz gestión de cliente</i>	100
<i>Tabla 12 Mantenimiento de gestión de cliente</i>	101
<i>Tabla 13 Tarjeta CRC Empleado</i>	101
<i>Tabla 14 Tarjeta CRC proveedor</i>	101
<i>Tabla 15 Tarjeta CRC cliente</i>	102
<i>Tabla 16 Gestión de tipos de Usuarios</i>	104
<i>Tabla 17 Diseño de interfaz gestión de usuarios y privilegios</i>	104
<i>Tabla 18 Mantenimiento de gestión de tipos de usuarios</i>	105
<i>Tabla 19 Gestión de privilegios de tipo de usuario</i>	105
<i>Tabla 20 Diseño de interfaz gestión de privilegios de usuario</i>	106
<i>Tabla 21 Mantenimiento de gestión de privilegios de Tipo de usuario</i>	106
<i>Tabla 22 Tarjeta CRC Tipo Usuario</i>	107
<i>Tabla 23 Gestión de Categorías de productos</i>	108
<i>Tabla 24 Diseño de interfaz gestión de categoría de productos</i>	109
<i>Tabla 25 Mantenimiento de Categorías de productos</i>	109
<i>Tabla 26 Tarjeta CRC Categoría</i>	109
<i>Tabla 27 Gestión de productos</i>	110
<i>Tabla 28 Diseño de interfaz Gestión de productos</i>	110
<i>Tabla 29 Diseño Interfaz Gestión de productos</i>	111
<i>Tabla 30 Tarjeta CRC Producto</i>	111
<i>Tabla 31 Gestión de compras</i>	114
<i>Tabla 32 Diseño de interfaz Gestión de Compras</i>	114
<i>Tabla 33 Mantenimiento Gestión de Compras</i>	115
<i>Tabla 34 Gestión Anular compras</i>	115
<i>Tabla 35 Diseño de interfaz Gestión de Anular compras</i>	116
<i>Tabla 36 Mantenimiento Gestión Anular compras</i>	116
<i>Tabla 37 Tarjeta CRC Compras</i>	117
<i>Tabla 38 Gestión de Ventas</i>	118

<i>Tabla 39</i>	<i>Diseño de interfaz de Gestión de Ventas</i>	119
<i>Tabla 40</i>	<i>Mantenimiento Gestión de ventas</i>	119
<i>Tabla 41</i>	<i>Gestión de Anular de ventas</i>	120
<i>Tabla 42</i>	<i>Diseño de interfaz Gestión Anular Ventas</i>	120
<i>Tabla 43</i>	<i>Mantenimiento Gestión Anular Ventas</i>	121
<i>Tabla 44</i>	<i>Tarjeta CRC Ventas</i>	121
<i>Tabla 45</i>	<i>Software de desarrollo</i>	137
<i>Tabla 46</i>	<i>Sistema operativo</i>	137
<i>Tabla 47</i>	<i>Costo de personal</i>	137
<i>Tabla 48</i>	<i>Costo de materiales</i>	138
<i>Tabla 49</i>	<i>Costo de servicios</i>	138
<i>Tabla 50</i>	<i>Costo de Hardware</i>	138
<i>Tabla 51</i>	<i>Costos de implementación</i>	139
<i>Tabla 52</i>	<i>Costo de mantenimiento</i>	139
<i>Tabla 53</i>	<i>Resumen de costos</i>	139
<i>Tabla 54</i>	<i>Monto Inversión Inicial</i>	140
<i>Tabla 55</i>	<i>Cuadro de Ahorro con uso del Sistema</i>	140
<i>Tabla 56</i>	<i>Evaluación en base al VAN</i>	141
<i>Tabla 57</i>	<i>Evaluación en base al TIR</i>	141

INDICE DE FIGURAS

Figura 1 Estructura Orgánica	19
Figura 2 Criterios de selección de la metodología empleada	46
Figura 3 Principios de la Metodología XP	53
Figura 4 Fases de la metodología XP	58
Figura 5 Entorno de desarrollo típico en Java	80
Figura 6 Entorno MySQL Workbench	90
Figura 7 Vista Modelo en Entorno MySQL Workbench	91
Figura 8 Entorno Servidor Local XAMPP.....	92
Figura 9 Entorno phpMyAdmin	92
Figura 10 Entorno de desarrollo NetBeans	93
Figura 11 Configuración del controlador MySQL JDBC Driver	94
Figura 12 Cadena de conexión a la Base de Datos	95
Figura 13 Entorno de iReport	95
Figura 14 Interface principal del sistema	96
Figura 15 Registro de empleados	102
Figura 16 Registro de proveedor	103
Figura 17 Registro de cliente	103
Figura 18 Creación de roles según tipo de Usuario.....	107
Figura 19 Registro de Tipo de Usuario	108
Figura 20 Registrar/Modificar Categorías	112
Figura 21 Registrar Categoría	112
Figura 22 Registrar/Modificar Producto.....	113
Figura 23 Registrar producto	113
Figura 24 Registrar Compra	117
Figura 25 Anular Compra	118
Figura 26 Realizar venta	122
Figura 27 Anular venta	122
Figura 28 Base de Datos	123
Figura 29 Formulario de Ventas	124
Figura 30 Método listarClientePorParametro	125
Figura 31 Procedimiento almacenado SP_S_clientePorParametro	126
Figura 32 FrmBuscarCliente_Venta	126
Figura 33 Seleccionar Tipo Documento (Boleta, Factura, Ticket...) Desde Combobox.....	127
Figura 34 btnAgregarProducto	128
Figura 35 Generar Venta	129
Figura 36 Visualizar o imprimir comprobante de venta.....	129

Figura 37 Comprobante de venta.....	130
Figura 38 Número aleatorios para panel de contraseña de acceso.....	131
Figura 39 Vista principal Sistema.....	131
Figura 40 Login para acceso al sistema	132
Figura 41 Acceso al sistema.....	132
Figura 42 Interfaz de almacén	133
Figura 43 Tipos de usuarios registrados.....	133
Figura 44 Tipos de funciones asignadas por tipo de usuario	134

INTRODUCCIÓN

El proyecto de investigación lleva por título Desarrollo de un sistema de información haciendo uso de la metodología XP, para la gestión de ventas, compras y almacén de la empresa Agro Market Perú S.A.C.

Consta de 7 capítulos donde en cada uno de ellos se desarrolla la solución a la problemática.

Los capítulos son Datos Generales de la Organización, Problemática de la Investigación, objetivos, justificación, Marco Metodológico, Marco Teórico, Desarrollo de la Propuesta y Aspecto Administrativo.

Capítulo I, se hace referencia a la información general de la empresa donde se llevará a cabo el trabajo de investigación como su misión, visión, objetivos, estructura orgánica etc.

Capítulo II, hablaremos de la problemática donde basaremos nuestra investigación, planteamiento del problema y su justificación e importancia.

Capítulo III, describe el marco metodológico, tipo de investigación a utilizar en el proyecto, contrastación de hipótesis.

Capitulo IV, marco teórico, herramientas de desarrollo, software y arquitectura.

Capítulo V, describe el paso desarrollo de la propuesta, configuraciones iniciales, base de datos, entorno de desarrollo, planificación y diseño, iteraciones

Capítulo VI, realizaremos un análisis de costos y beneficios, recuperación de la inversión.

Conclusiones.

Recomendaciones.

Capitulo VII, Bibliografía y linkografía utilizada como referencia en la investigación y desarrollo.

Anexos

CAPITULO I

DATOS GENERALES DE LA ORGANIZACIÓN

I: DATOS GENERALES DE LA ORGANIZACIÓN

1.1.- DESCRIPCIÓN DE LA ORGANIZACIÓN

Agro Market Perú S.A.C, fundada en el año 2005 y establecida en la ciudad de Olmos - Lambayeque. Es una empresa dedicada a la producción y comercialización de productos alimenticios y de primera necesidad al por mayor y menor.

1.2.-MISIÓN, VISIÓN Y OBJETIVOS DE LA ORGANIZACIÓN

1.2.1. MISIÓN

Ser líder en la comercialización de productos de consumo, satisfaciendo la mayoría de necesidades de nuestros clientes por encima de sus expectativas, brindándole productos de muy buena calidad y a precios muy económicos, para lograr relaciones comerciales de largo plazo, mediante un servicio rápido eficaz y de alternativas óptimas.

1.2.2. VISIÓN

Ser una empresa líder con un buen nivel de competencia local y nacional, libre de errores y malos entendidos, generando mayores beneficios para nuestros colaboradores y sus familias, para así asegurar la permanencia del negocio en el mercado.

1.2.3. OBJETIVOS

- El objetivo principal es el de poder expandirse a nivel nacional.
- Ofrecer siempre a los clientes una compra que supere sus expectativas.
- Tener siempre un buen surtido para mayor satisfacción del cliente.
- Mantener siempre un orden para mayor comodidad y acogida del cliente.
- Ofrecer una atención de alta calidad.

1.3.- ESTRUCTURA ORGÁNICA

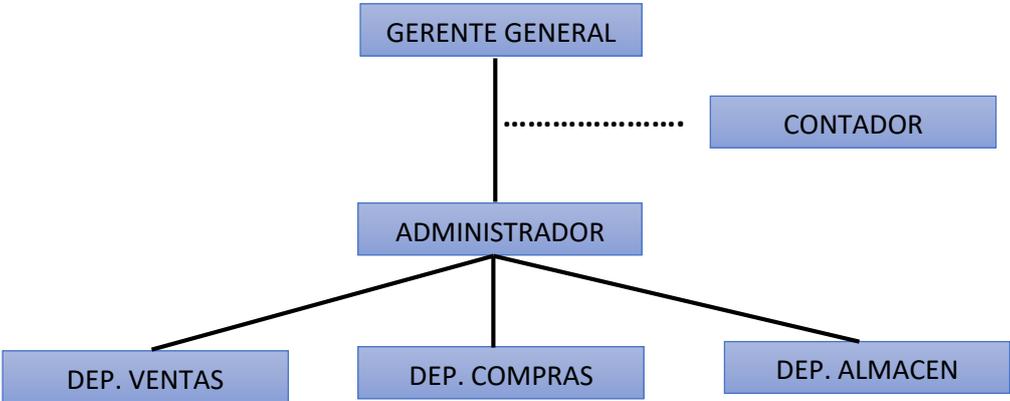


Figura 1 Estructura Orgánica
Fuente: Elaboración Propia

CAPITULO II

PROBLEMÁTICA DE LA INVESTIGACIÓN

II: PROBLEMÁTICA DE LA INVESTIGACIÓN

2.1.- REALIDAD DE LA PROBLEMÁTICA

2.1.1. PLANTEAMIENTO DEL PROBLEMA

A nivel mundial se ha reconocido que las Tecnologías de la Información y las Comunicaciones (TIC) tienen repercusiones en prácticamente todos los aspectos de nuestras vidas. El rápido avance de estas tecnologías brinda oportunidades sin precedentes para alcanzar niveles más elevados de desarrollo.

La capacidad de las TIC para reducir muchos obstáculos tradicionales, especialmente el tiempo, la distancia, posibilitan el uso potencial de estas tecnologías en beneficio de millones de empresas y personas en el mundo.

Las Tecnologías de la Información y las Comunicaciones (TICs), están inundando el mundo moderno con implicaciones en cada una de las ramas de la sociedad actual. Como podemos ver la sociedad de hoy día se adapta perfectamente a las tecnologías de la información y las comunicaciones.

Las TIC son un fenómeno que ha invadido todos los sectores de la vida, desde el trabajo hasta el ocio, los procesos de enseñanza y aprendizaje que se realizan en los diferentes niveles de educación, la economía porque permiten generar riqueza a distancia y en red superando las fronteras geográficas y políticas. Han impuesto también un cambio en las relaciones laborales, económicas, culturales y sociales, y un cambio en la forma de pensar de los propios individuos.

El uso de las TIC en nuestro país está fundamentado en sencillas operaciones relacionadas principalmente a la facturación, cobranza y muy poco en procesos de gestión de negocio. Aún con esta situación, se podrían identificar ciertas medidas y servicios que podrían impulsar el desarrollo de nuevas soluciones para la gestión del negocio, tales como: el uso de Internet como fuente de información en cada uno de

los diferentes sectores; la implantación de la banca electrónica a nivel general; el desarrollo de la comunicación con la administración pública, el uso de software para gestión de negocios, entre otros.

Las TIC en el departamento de Lambayeque han ido creciendo a con el transcurrir del tiempo por el incremento de competitividad entre empresas de diferentes rubros, para poder administrar, controlar y gestionar de una mejor manera sus recursos para lograr el éxito anhelado. La adopción e implantación de tecnologías en las grandes empresas es importante, ya que muestra principalmente el camino a seguir por las pequeñas y medianas empresas en el comportamiento frente al uso de las tecnologías de información y comunicaciones

Actualmente en la empresa Agro Market Perú S.A.C existen sistemas manuales que conllevan a la lentitud de sus procesos, tanto a la hora de atender a sus clientes como al momento de solicitar informes detallados.

Existen pequeñas hojas de cálculo en Excel, las cuales son utilizadas para solucionar procesos de las áreas de ventas, compras y almacén. Es por ello que el proceso de ventas presenta demoras al momento de solicitar varios pedidos a la vez. En oportunidades, se vende un producto que no está en stock y generando insatisfacción en los clientes.

El proceso de compras no tiene un informe detallado de cuál es el stock de almacén, lo cual conlleva a que se sobrecarguen de algunos productos y les falten otros.

El proceso de distribución presenta demoras de tiempo al momento de registrar la entrega de productos a sus clientes, así como el proceso de almacén tiene dificultades para llevar un buen control de los ingresos y salidas de los productos.

La lentitud de sus procesos manuales se está incrementando proporcionalmente al crecimiento de la empresa.

El tiempo de atención al cliente ha ido incrementándose conforme el número de clientes ha ido creciendo, por lo que

concluimos que la empresa requiere implementar un sistema de información.

Según la información proporcionada por la empresa Agro Market Perú S.A.C, los reclamos de parte de los clientes que se sienten insatisfechos por las demoras han ido aumentando considerablemente, por lo que es urgente buscar alternativas de solución.

2.2.- FORMULACIÓN DE PROBLEMA

¿Qué características debe tener el desarrollo de un Sistema de Información que permita mejorar los procesos de gestión de ventas, compras y almacén de la empresa Agro Market Perú S.A.C.?

2.3.- JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN

2.3.1 JUSTIFICACIÓN ECONÓMICA

Una vez implementado el sistema de información, permitirá a la empresa Agro Market Perú S.A.C agilizar sus procesos de compras, ventas y almacén, permitiendo el ahorro de mano de obra en personal de almacén y ventas.

2.3.2 JUSTIFICACIÓN TECNOLÓGICA

En un mundo globalizado donde las tecnologías de información y comunicación (TIC), brindan oportunidades para alcanzar niveles más elevados de desarrollo, es por ello que es de vital importancia utilizar tecnologías de información y comunicación adecuadas para el procesamiento y transmisión de los datos que se gestionarán en el sistema de información.

La empresa Agro Market Perú, convertirá su emprendimiento en una empresa competitiva insertada en el mercado actual, a raíz de los cambios en la economía mundial y la globalización, los datos relativos a todo el proceso productivo de una compañía se han vuelto uno de los elementos fundamentales para lograr el éxito comercial por ello la empresa Agro Market Perú S.A.C no es ajeno a estos cambios, razón fundamental para implementar un sistema informático de ventas, compras y almacén.

2.3.3 JUSTIFICACIÓN ACADÉMICA

Para justificar los conocimientos adquiridos durante la vida universitaria y su relación con la práctica, la Universidad Nacional Pedro Ruiz Gallo de Lambayeque, exige un trabajo de investigación para obtener el título de Ingeniero en computación e Informática.

Para cumplir con esta exigencia, se desarrollará un sistema de información para dar la solución al problema expuesto anteriormente, para así poner en práctica los conocimientos adquiridos, así como también ganar experiencia en la planificación y desarrollo de proyectos de investigación.

Para el desarrollo e implementación del sistema de información es necesario aplicar conocimientos de bases de datos relacionales(MySQL), uso del lenguaje de programación Java y conectividad de computadoras mediante una red LAN. Esto nos permitirá poner en práctica los conocimientos adquiridos en materia de bases de datos, programación y redes informáticas.

En conclusión, el desarrollo de este proyecto de investigación me permitirá poner en práctica los conocimientos adquiridos durante todos los ciclos de mi formación universitaria, y obtener la experiencia en elaborar proyectos de investigación.

2.3.4 JUSTIFICACIÓN OPERATIVA

Para el desarrollo del proyecto de investigación, se tienen las facilidades necesarias y la colaboración de la empresa respecto a la recopilación de información para la elaboración e implementación del sistema.

2.3.5 IMPORTANCIA

Con el fin de mejorar la productividad y el rendimiento de la empresa Agro Market Perú S.A.C, es fundamental implementar un sistema de información para el control de ventas, compras y almacén, que brinden eficiencia y eficacia de la gestión de dichos procesos.

Con la implementación de este sistema de información en la empresa Agro Market Perú S.A.C, obtendremos grandes

ventajas, permitiéndonos agilizar los procesos de ventas, compras, almacén y atención al cliente, incrementar la capacidad de organización y competitividad de la empresa, mejorando los reportes de ventas diarias, el control de sus productos para poder realizar una mejor compra y venta de los mismos, el sistema proporcionará información clave para la toma de decisiones; esta información será sencilla, clara, expedita, veraz, precisa, consistente y fácil de analizar e interpretar.

2.4.- OBJETIVOS DE LA INVESTIGACIÓN

2.4.1. OBJETIVO GENERAL

Desarrollar un Sistema de información haciendo uso de la Metodología XP, para la empresa Agro Market Perú S.A.C.

2.4.2. OBJETIVOS ESPECÍFICOS

1. Analizar las fases de desarrollo utilizados por la metodología XP para la correcta aplicación de sus pasos en el desarrollo del sistema.
2. Capturas de los requerimientos de los procesos de ventas, compras y almacén.
3. Analizar la situación actual de los procesos de ventas, compras y almacén de la empresa Agro Market Perú S.A.C, específicamente las bases de datos y sistemas manuales de ventas.
4. Modelar la base de datos y la arquitectura básica del sistema de información aplicando los recursos disponibles para desarrollar la fase de elaboración.
5. Desarrollar un sistema de información que permita gestionar los procesos de ventas, compras y almacén de la empresa Agro Market Perú S.A.C.
6. Realizar la evaluación económica del proyecto

2.5.- LIMITACIONES DE LA INVESTIGACIÓN

Durante el desarrollo de la presente investigación se encontró las siguientes limitaciones:

- Deficiente acceso a la información de los procesos de ventas, compras y almacén.
- Desconocimiento de los procesos por parte del personal encargado de las áreas de ventas, compras.
- Poco apoyo al momento de la recopilación de la información por parte del personal de la empresa.
- No se contó con copias de información e informes por parte de la empresa por no contar con los recursos necesarios para costear dicho material.

CAPITULO III

MARCO METODOLÓGICO

III: MARCO METODOLÓGICO

3.1.- TIPO DE INVESTIGACIÓN

Investigación Tecnológica Formal

3.2. HIPÓTESIS

La demostración de la hipótesis planteada en el presente proyecto será mediante un Diseño No Experimental.

3.3.- VARIABLES

3.3.1. VARIABLE INDEPENDIENTE

Sistema de Información

3.3.2. VARIABLE DEPENDIENTE

Gestión de Compras

Gestión de Ventas

Gestión de Almacén

3.4.- DISEÑO Y CONTRASTACIÓN DE HIPÓTESIS

Mediante el uso del Gestor de bases de datos MySQL se logrará el correcto manejo de toda la información de los procesos de ventas, compras y almacén de la empresa Agro Market Perú S.A.C.

El uso del lenguaje de programación Java nos permitirá desarrollar un sistema potente para gestionar de manera eficiente los procesos de la empresa.

CAPÍTULO IV

MARCO TEORICO

IV: MARCO TEÓRICO

El sistema de Información se desarrollará con una arquitectura de tres capas, haciendo uso de la metodología XP, el lenguaje de programación Java, con el paradigma de programación orientada a objetos y utilizará una base de datos relacional MySQL. Es por eso que este capítulo se referirá a cada uno de estos temas.

(Kendall & Kendall, 2011)

A) SISTEMA DE INFORMACIÓN

“Es un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades de una empresa o negocio”.

Partiendo de esta definición, indica que las organizaciones con éxito utilizan la información como instrumento eficaz para la administración y han adquirido sistemas de información que responden a las necesidades de las personas.

Actualmente la necesidad de información en las organizaciones es mucha y su existencia es vital para alcanzar éxito, los sistemas han evolucionado en su uso, comenzando con la automatización de procesos operativos de las organizaciones como apoyo a este nivel para brindar información que sirva de base en el proceso de toma de decisiones.

Es importante tener en cuenta que un sistema de información necesita justificar su implementación desde el punto de vista costo/beneficio, partiendo de la concepción del valor que se le otorgue a la información dentro de una organización. Los beneficios se pueden medir en el ámbito intangible y tangible de acuerdo a la organización, que pretende prestar un servicio a la ciudadanía.

Los sistemas de información se desarrollan para distintos fines, dependiendo de las necesidades de los usuarios humanos y la empresa.

A.1) TIPOS DE SISTEMA INFORMACIÓN

El propósito de un sistema de información, puede ser muy amplio, todo depende de las necesidades de la organización. Existen distintos tipos de sistemas de información, entre los que destacan los siguientes:

A.1.1) SISTEMAS DE PROCESAMIENTO DE TRANSACCIONES

Se define como transacción un suceso que implica o afecta a una organización, y que está compuesta por datos referentes a ellas y que son de importancia para la organización.

Estos sistemas se encargan del procesamiento de los datos referentes a las transacciones, además de permitir la automatización de tareas y procesos operativos. La información que se obtiene como salida es utilizada posteriormente por los funcionarios de nivel operativo de la organización en la toma de decisiones.

Las razones para el procesamiento de las transacciones son:

- **CLASIFICACIÓN**, implica agrupar todos los datos de acuerdo con características comunes.
- **OPERACIONES DE CÁLCULO**, consiste en realizar alguna operación para obtener resultados útiles.
- **ORDENAMIENTO**, consiste en disponerlos de alguna forma o secuencia, facilita el procesamiento y la búsqueda.
- **SÍNTESIS**, reduce los datos en información breve y concisa.
- **ALMACENAMIENTO**, permite el registro de todas y cada una del suceso que afectan a la organización.

A.1.2) SISTEMAS DE INFORMACIÓN ADMINISTRATIVA

(Kendall & Kendall, 2011)

Los sistemas de información administrativa (MIS) no sustituyen a los sistemas de procesamiento de transacciones; más bien, todos los sistemas MIS incluyen el procesamiento de transacciones. Los MIS son sistemas de información computarizados que funcionan debido a la decidida interacción entre las personas y las computadoras.

Al requerir que las personas, el software y el hardware funcionen en concierto, los sistemas de información administrativa brindan soporte a los usuarios para realizar un espectro más amplio de tareas organizacionales que los

sistemas de procesamiento de transacciones, incluyendo los procesos de análisis y toma de decisiones.

Para acceder a la información, los usuarios del sistema de información administrativa comparten una base de datos común; ésta almacena tanto los datos como los modelos que permiten al usuario interactuar con ellos, interpretarlos y aplicarlos. Los sistemas de información administrativa producen información que se utiliza en el proceso de toma de decisiones. También pueden ayudar a integrar algunas de las funciones de información computarizadas de una empresa.

A.1.3) SISTEMAS DE SOPORTE DE DECISIONES

(Kendall & Kendall, 2011)

Son sistemas de información que tienen como propósito auxiliar al usuario con las decisiones únicas que no se repiten y que no tienen una estructura definida. Además de estar hechos a la medida de la persona o grupo que los usa en comparación con los Sistemas de información Gerencial.

El propósito de estos sistemas es el de responder correctamente a condiciones inesperadas y propias de la información. Esto permite que sean empleados en niveles altos de la organización.

A.1.4) SISTEMAS DE INFORMACIÓN GERENCIAL

(Kendall & Kendall, 2011)

Los Sistemas de Información Gerencial, también llamados Sistemas de Reportes de Gerencia, se dedican al apoyo de decisiones siempre que los requerimientos de información sean identificados, esto es que la información que necesita para la toma de decisiones haya sido analizada anteriormente, y que esta misma decisión pueda tomarse nuevamente.

Estos sistemas pueden extraer la información necesaria de cualquier parte de la organización, por lo que la información

necesaria ya se tiene almacenada al ser procesada por un sistema de transacciones.

A.1.5 SISTEMAS EXPERTOS E INTELIGENCIA ARTIFICIAL

(Kendall & Kendall, 2011)

La inteligencia artificial (AI) puede ser considerada como el campo dominante de los sistemas expertos. La idea general de la AI ha sido desarrollar equipos que se comporten de manera inteligente. Dos ramas de investigación de la AI son:

- La comprensión del lenguaje natural
- El análisis de la habilidad para razonar un problema y llegar a una conclusión lógica.

Los sistemas expertos utilizan las metodologías de razonamiento de la AI para resolver los problemas que los usuarios de negocios (y otros tipos de usuarios) les presentan.

Los sistemas expertos son una clase muy especial de sistema de información que ha demostrado su utilidad comercial gracias a la disponibilidad extendida de hardware y software como las computadoras personales (PC) y las interfaces de sistemas expertos. Un sistema experto (también conocido como sistema basado en el conocimiento) captura y utiliza en forma efectiva el conocimiento de uno o varios expertos humanos para resolver un problema específico al que una organización se enfrenta. Cabe mencionar que a diferencia de los sistemas DSS, que en última instancia dejan la decisión a la persona encargada de la toma de decisiones, un sistema experto selecciona la mejor solución para un problema o una clase específica de problemas.

Los componentes básicos de un sistema experto son la base de conocimiento, un motor de inferencia que conecta al usuario con el sistema mediante el proceso de consultas en lenguajes, como el lenguaje de consulta estructurado (SQL), y la interfaz de usuario. Las personas conocidas como ingenieros del conocimiento capturan la experiencia de los expertos, crean un sistema computacional que incluye este conocimiento y después lo implementan.

(Elmasri & Navathe, 2002)

B) CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN

El ciclo de vida de un sistema de información comprende las siguientes etapas:

- ✓ **PLANIFICACIÓN:**
 - Ámbito del proyecto
 - Estudio de viabilidad
 - Análisis de riesgos
 - Estimación
 - Planificación temporal
 - Asignación de recursos.

- ✓ **ANÁLISIS (¿qué?):**
 - Elicitación de requerimientos (funcionales y no funcionales)
 - Modelado de datos y de procesos
- ✓ **DISEÑO (¿cómo?):** Estudio de alternativas y diseño arquitectónico
 - Diseño de la base de datos
 - Diseño de las aplicaciones
- ✓ **IMPLEMENTACIÓN:** Adquisición de componentes, creación e integración de los recursos necesarios para que el sistema funcione.
- ✓ **PRUEBAS:** Pruebas de unidad, pruebas de integración, pruebas alfa, pruebas beta, test de aceptación.
- ✓ **INSTALACIÓN / DESPLIEGUE**
- ✓ **USO / MANTENIMIENTO**

C) ARQUITECTURA DE SISTEMA DE INFORMACIÓN

(IEEE 1471-2000)

- ✓ La arquitectura del Software es la organización fundamental del sistema que incluye a sus componentes, sus relaciones entre ellos, el ambiente y los principios que dictan su diseño y evolución.

- ✓ Involucra un conjunto de decisiones significativas acerca de la organización del sistema
- ✓ Selección de sus elementos estructurales y sus interfaces
- ✓ Comportamiento, especificado en función de la colaboración de los elementos
- ✓ Composición de sub-sistemas más grandes a partir de elementos estructurales y elementos con comportamiento

La arquitectura de software también involucra

- ✓ Funcionalidad
- ✓ Usabilidad
- ✓ Tolerancia a cambios
- ✓ Performance
- ✓ Reutilización
- ✓ Restricciones económicas y tecnológicas (equilibrio)
- ✓ Aspectos estéticos

4.1.- ANTECEDENTES DE LA INVESTIGACIÓN

4.1.1. ANTECEDENTES EN EL CONTEXTO INTERNACIONAL

- **RODRÍGUEZ, M. (2005)**, en su trabajo de investigación para grado, titulado: “DESARROLLO DE UN SISTEMA DE INFORMACIÓN AUTOMATIZADO PARA LA FACTURACIÓN Y EL CONTROL DE VENTAS DE LA EMPRESA 3M COMPUTACIÓN, C.A.

CONCLUSIÓN: los adelantos tecnológicos y los sistemas automatizados que hoy en día manejamos son herramientas que le permiten a la empresa contar con información veraz y oportuna.

Tal como lo explica Rodríguez, el sistema propuesto, ayudó a la solución de fallas del sistema Premium Sofá, la cual se realizó para gestionar las actividades de la empresa aportando una herramienta eficaz para llevar el control de los procesos administrativos de la empresa. La investigación se inclinó en la metodología descrita por el autor Jonás Montilva, metodología estructurada o ciclo de vida de un sistema de información, el cual proporcionó un aporte considerable para el tratamiento de la metodología en el análisis y aplicación de cada fase de investigación.

- **SÁNCHEZ MINDY (2011)**, “DESARROLLO DE UN SISTEMA WEB BASADO EN ADO.NET PARA EL CONTROL DE LA GESTIÓN DE INSCRIPCIÓN Y LOGÍSTICA DE CURSOS EN LA EMPRESA ENFOQUE DIRECTO APLICADO CONSULTORES, C.A.”

CONCLUSIÓN: Recalamos que el trabajo desarrollado cubrió las expectativas de la empresa, y que esta manifestó que realizará las gestiones administrativas necesarias para procurar los recursos económicos que requiere a efecto de implementarla en su plataforma tecnológica.

4.1.2. ANTECEDENTES EN EL CONTEXTO NACIONAL

- **PEZO LINARES, JOSÉ (2012)**, “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA MEJORAR EL PROCESO DE VENTAS EN LA TIENDA SEÑOR DE AYABACA DE LA CIUDAD DE TARAPOTO, 2012”.

CONCLUSIÓN: El diseño modular empleado que tiene el sistema facilita la administración y entendimiento haciendo más fácil la integración de otros módulos o componentes para su crecimiento, con ello también cabe recalcar que el diseño multiplataforma que se integre fácilmente a cualquier plataforma de hardware y software.

- **GUZMÁN SILVA, J (2008)**, “DISEÑO Y OPTIMIZACIÓN DEL PROCESO DE GESTIÓN Y EJECUCIÓN DE LA VENTA MAYORISTA PARA UNA EMPRESA TIPO HOME IMPROVEMENT”.

- **CONCLUSIÓN:** Lo que ha permitido entre estas sinergias se encuentra el hecho que el análisis de Venta Cruzada sea una extensión de la Minería de Datos, que se realiza con el apoyo de una empresa de prestigio como lo es Venta. Por otro lado, los Modelos de Optimización demuestran tener, a partir de una muestra pequeña pero representativa de acuerdo a su variedad y niveles de compra, una cercanía con el comportamiento real de los clientes.

Es importante hacer notar en este punto de comparación que la realización de la tercera etapa de optimización (en el sentido de aprovechar de mejor manera los recursos de fuerza de venta para mejorar los resultados tanto en los ingresos como en la calidad percibida por el cliente) marcaría

una diferencia importante, pues es la base del crecimiento de los ingresos aprovechando de mejor manera las necesidades de los clientes, incluso generando en ellos necesidades que serán satisfechas inmediatamente por la empresa, incluso antes que al cliente se le presenten como urgencias por consideraciones no son tomadas en el momento correcto, sino que son sugeridas por el vendedor.

4.1.3. ANTECEDENTES EN EL CONTEXTO LOCAL

- **LLACCHUA, M. (2007)**, “DISEÑO DE UN SISTEMA DE COMERCIALIZACION PARA EL SUPERMERCADO MINIMARKET TITO’S”.

CONCLUSIÓN: El diseño modular que tiene el sistema facilita la administración y el entendimiento del mismo haciendo más la integración de otros módulos o componentes para su crecimiento con ello también cabe recalcar que el diseño multiplataforma hace que se integre fácilmente a cualquier plataforma de hardware y software. El uso de metodología de desarrollo RUP, conjuntamente con el lenguaje UML y el manejo de los conceptos de la programación orientada a objetos, propiciaron que el desarrollo del sistema sea entendible, sostenible, incremental. Más aún, el uso de un entorno de desarrollo rápido como es Netbeans para JAVA, ha hecho que nuestro desarrollo sea concluido en el tiempo previsto.

- **BALAREZO PAREDES, BRALLAN (2012)**, “DESARROLLO DE UN SISTEMA DE INFORMACIÓN DE REGISTRO DE PEDIDOS PARA VENTAS USANDO DISPOSITIVOS MÓVILES”.

Conclusión: El desarrollo del presente proyecto representará una gran ayuda a las empresas comercializadoras, debido a que contarán con un sistema que les permita realizar la gestión de pedidos de una forma más rápida, confiable y adicionalmente se tendrá la posibilidad de explotar la información registrada a través de reportes que sirvan de apoyo para la toma de decisiones.

4.2.- BASE TEÓRICA

La Metodología a utilizar tiene por característica fundamentales el desarrollo iterativo e incremental, aquí los requerimientos y soluciones evolucionan mediante la colaboración e intervención de grupos relacionados con el proyecto en desarrollo.

Estas metodologías enfatizan los siguientes valores:

- ✓ El individuo y la gente de desarrollo es el principal factor de éxito de un proyecto de software sobre el proceso y herramientas a usarse.
- ✓ Desarrollar un software funcional, antes que producir documentación buena, ésta última debe producirse en el caso de ser necesaria de forma inmediata, así mismo estos documentos deben ser cortos y centrarse en lo fundamental.
- ✓ La interacción constante entre el cliente y el equipo de desarrollo, más que la negociación de un contrato. Este detalle será la pauta que marque la marcha del proyecto y asegure su éxito.
- ✓ Responder los cambios que se susciten a lo largo del proyecto antes que seguir estrictamente una planificación.

4.2.1 RESUMEN LAS METODOLOGÍAS ÁGILES MÁS RELEVANTES

(Florez Marin)2014

4.2.1.1 SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

4.2.1.2 CRYSTAL METHODOLOGIES:

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

4.2.1.3 DYNAMIC SYSTEMS DEVELOPMENT METHOD (DSDM)

Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

4.2.1.4 ADAPTIVE SOFTWARE DEVELOPMENT (ASD)

Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes

sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

4.2.1.5 FEATURE-DRIVEN DEVELOPMENT (FDD)

Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.

4.2.1.6 METODOLOGIA RUP

El proceso unificado es uno de los marcos de trabajo más utilizados en la actualidad para el desarrollo de software. Es orientado a objetos y se guía mediante tres conceptos principales. Es dirigida por casos de uso. Se centra en la arquitectura. Es iterativo e incremental. Fases de desarrollo: Inicio, elaboración, construcción, transición.

Flujos de trabajo: Modelado del negocio, requerimientos, análisis y diseño, implementación, pruebas, despliegue, gestión del proyecto, configuración y control de cambios, entorno

4.2.1.7 PROGRAMACIÓN EXTREMA (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo.

4.2.2 CRITERIOS DE SELECCIÓN DE LA METODOLOGÍA EMPLEADA

4.2.2.1 FORMULACION Y DEFINICION DE CRITERIOS

La selección de los marcos de trabajo a evaluar se realizó bajo la premisa de frecuencia de uso, documentación existente y efectividad de la metodología.

RUP es el marco de trabajo tradicional más utilizado actualmente y sobre el cual existe la mayor cantidad de documentación.

En cuanto a metodologías ágiles, XP y SCRUM resaltan por su gran adaptabilidad frente a cambios y sus drásticas reducciones de los tiempos de desarrollo. Si bien el uso de Crystal, ASD, DSDM, FDD su uso no es muy común en la actualidad, se seleccionó este marco de trabajo debido a su gran similitud tanto con metodologías tradicionales, como ágiles, permitiendo evaluar la alguna posibilidad.

A continuación, se realizará la discriminación de criterios que servirán como base para una futura calificación cuantitativa, de manera que se a claramente visible la diferencia en el uso de una u otra metodología de desarrollo.

(Florez Marin)

4.2.2.1.1 PRESUPUESTO DISPONIBLE

A la hora de llevar a cabo un proyecto es de vital importancia realizar un estimado del presupuesto que se va a destinar a este.

Los costos de implementación de cada metodología varían, dados los requerimientos específicos que cada una de ellas posee.

Teniendo en cuenta las investigaciones hechas en los anteriores capítulos del proyecto podemos observar los diferentes presupuestos que cada metodología puede tener por los

diferentes recursos, artefactos y personal que éstas requieren para su desarrollo.

RUP puede llegar a ser la metodología más costosa, dependiendo del tamaño del proyecto. ASD, DSDM, FDD puede implicar costos muy bajos para un producto de buena calidad.

Por otro lado, tanto XP como SCRUM por ser metodologías ágiles no demandan muchos gastos en cuestiones de personal y recursos para el desarrollo de los proyectos.

(Florez Marin)

4.2.2.1.2 TAMAÑO DEL PROYECTO

Las metodologías tradicionales van enfocadas principalmente hacia proyectos grandes que conlleven desarrollos a largo plazo. RUP es una metodología pesada, orientada a los casos de uso y con estándares que facilitan el desarrollo ordenado para proyectos grandes, sin embargo, en proyectos relativamente pequeños puede ocasionar algunos sobrecostos. ASD, Crystal, DSDM y FDD son metodología cuya efectividad se ve reducida en proyectos de tamaño demasiado grande. XP y SCRUM están orientadas principalmente a proyectos no demasiado extensos como el que estamos se va a desarrollar.

(Florez Marin)

4.2.2.1.3 TIEMPOS LIMITADOS DE ENTREGA

Todo proyecto, independiente de su tamaño, se ve sujeto a limitaciones de tiempo, las cuales pueden llegar a marcar la diferencia entre la selección de una metodología ágil o una tradicional.

Las metodologías ágiles se caracterizan por tener tiempos cortos de diseño e

implementación por sus cortas iteraciones. En contraposición, las metodologías tradicionales poseen una mejor organización a la hora de la división del trabajo, conllevando iteraciones más prolongadas. XP es una metodología diseñada para realizar entregas viables en tiempos relativamente cortos. RUP requiere una cantidad mayor de tiempo para sus iteraciones, en comparación con una metodología ágil.

(Florez Marin)

4.2.2.1.4 NECESIDAD DE DOCUMENTACION

Para diversos equipos de trabajo, dependiendo de su tamaño y organización, se hace necesaria la creación de documentos con una mayor o menor profundidad. No todas las empresas requieren documentación exhaustiva sobre su software o los procesos para llevarlo a cabo. La creación de manuales de usuario es opcional dentro de algunas empresas.

Tanto XP como SCRUM carecen del manejo de una documentación formal para el desarrollo de los proyectos, la única documentación que estas 2 metodologías ofrecen es el código resultado de las diferentes iteraciones. RUP es una metodología orientada a la creación de múltiples documentos de apoyo para los diversos procesos.

4.2.2.1.5 PERSONAL NECESARIO

Existen diferentes tamaños de proyecto, cada uno con sus requerimientos de personal, dado el software y hardware necesario, la necesidad de un equipo interdisciplinario y la coordinación requerida entre cada área del desarrollo.

XP cuenta con numerosos roles para el control de los procesos en la diferentes iteraciones y el

número de personas puede aumentar dependiendo del tamaño del grupo de programadores, aun así, el total de miembros no sobrepasa los 15. SCRUM es la metodología que menos personal necesita ya que posee muy pocos roles y su tamaño crece dependiendo del grupo de programación que puede ir de 5 a 10 personas. RUP es una metodología con roles definidos para grupos grandes de programadores.

4.2.2.1.6 ADAPTABILIDAD, RESPUESTA A CAMBIOS

No todo proyecto se ve sujeto a cambios repentinos durante su planeación, pero todos deberían poseer un mecanismo de respuesta ante estos. La posibilidad de que ocurra un cambio repentino varía de acuerdo al tipo de proyecto y sus cualidades.

XP y SCRUM como metodologías ágiles responden a uno de los valores en los cuales son basados y esto es la flexibilidad que poseen en respuesta a los cambios que se pueden presentar en el desarrollo del proyecto. Las metodologías tradicionales como RUP son sensibles a cambios, principalmente en etapas avanzadas del proyecto.

4.2.2.1.7 IMPOSIBILIDAD DEL CLIENTE

El cliente es la parte más importante en el desarrollo de cualquier proyecto de software, dado que es él quien provee todos los requerimientos, especificaciones y detalles del proyecto que se va a llevar a cabo, por lo tanto, la disponibilidad de tiempo del cliente para el proyecto es un tema a poner en consideración.

SCRUM como metodología ágil tiene al cliente como parte del equipo llamándolo Product Owner, el cual participa constantemente de las correcciones y observaciones en las

diferentes entregas de los Sprints. XP como parte de su estructura de desarrollo posee en sus prácticas como fundamental la participación del cliente no solamente como apoyo a los desarrolladores, sino formando parte del grupo. RUP es un modelo incremental e iterativo, de manera que la interacción con el cliente se realiza principalmente en etapas tempranas de planeación y desarrollo.

(Kniberg)

4.2.2.2 CUANTIFICACION DE LOS CRITERIOS

Teniendo en cuenta los criterios analizados en el anteriormente, procedemos a asignar valores a cada metodología según su grado de cumplimiento con los criterios existentes.

La evaluación será realizada utilizando números enteros en el rango del 01 al 05, asignando un valor objetivo, dependiendo directamente de la necesidad específica del proyecto a desarrollar.

Los valores asignados reflejarán la aplicabilidad o no de la metodología en el proyecto de investigación, siendo 05 el valor que indica el mejor cumplimiento del criterio analizado respecto a las prácticas utilizadas en el marco de trabajo. En caso contrario, un valor 01 implica que tales prácticas resultan contraproducentes para el correcto desarrollo del proyecto.

En la Figura 02 se puede observar la asignación de tales valores, de manera que se puedan referenciar fácilmente las metodologías y sus ventajas en ciertos factores que pueden considerarse decisivos a la hora de tomar decisiones.

*DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,*

	ASD	Crystal	DSDM	FDD	RUP	SCRUM	XP
Presupuesto disponible	1	2	1	1	1	4	5
Tamaño del proyecto	3	3	3	4	5	3	4
Tiempos limitados de entrega	1	1	1	1	1	4	5
Necesidad de documentación	2	3	3	2	5	3	4
Personal Necesario	4	3	3	3	4	4	5
Adaptabilidad, respuesta a cambios	3	3	3	3	3	5	5
Imposibilidad del cliente	3	3	4	3	3	4	5
Total	17	18	18	17	22	27	33

*Figura 2 Criterios de selección de la metodología empleada
Fuente: Diseño Propio*

Tomando en consideración la tabla de comparación de metodologías ágiles y tradicionales presentada, donde se basa en parámetros como: Presupuesto disponible, tamaño del proyecto, tiempos limitados de entrega, necesidad de documentación, personal necesario para el desarrollo, adaptabilidad con respuesta a cambios, imposibilidad del cliente o participación de este, y características específicas de cada metodología, por ejemplo: Participación del cliente, adaptabilidad, entre otras.

Se concluye que la metodología que se ajusta para el desarrollo del Sistema de Información es la metodología de Programación Extrema (XP), que además posee abundante información en internet que facilitará la ejecución del proyecto.

4.2.3 METODOLOGÍA XP

(Extreme Programming (XP), 2006)

4.2.3.1 INTRODUCCIÓN

La metodología XP fue creada para desarrollar sistemas de corto plazo por Kent Beck en 1999; el éxito de esta metodología es el uso de una excelente comunicación, su sencillez y el permitir interactuar con el cliente.

“Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar”. - Kent Beck

Las características más importantes de la metodología XP son:

- Actualmente es utilizada en Ingeniería de Software para desarrollar sistemas a corto plazo.
- Permite desarrollar proyectos sin dar mucha importancia a la documentación.
- Realiza la codificación y pruebas de forma paralela y no secuencial, así como permite la colaboración y comunicación directa de los clientes, otorgando respuestas inmediatas ante la eventualidad de presentarse algún cambio.
- Dispone de herramientas informáticas y no informáticas para desarrollar por completo un proyecto de desarrollo Web.
- Verifica que el desarrollo del proyecto funcione por medio de prototipos. Por regla

general estos se deben producir cuando sean realmente necesarios.

(Bustamante & Rodriguez, 2014)

4.2.3.2 CONCEPTOS BÁSICOS

La metodología XP está fundamentada en los siguientes valores que permiten facilitar el trabajo para realizar la implementación de la aplicación.

- **COMUNICACIÓN**
Mantener un intercambio de información entre el cliente y los programadores durante todo el desarrollo del sistema, para evitar dejar de lado puntos importantes que al inicio fueron presentados.
- **SIMPLICIDAD**
Desarrollar un sistema que sea ágil, unificado y de fácil comprensión para cuando se precise realizar actualizaciones, en caso de ser necesario, no se requiera empezar de cero.
- **RETROALIMENTACIÓN**
Visualizar el desarrollo del sistema a tiempo, para ir verificando si todo está correcto o si se debe realizar cambios; esto depende de las necesidades del cliente.
- **TENACIDAD**
Es el valor más importante en razón de que a través de éste podemos cumplir los tres valores anteriores, pues se debe poseer: valor para la comunicación, decisión para la simplicidad y enfatizar la retroalimentación.

La metodología XP toma en cuenta cuatro variables muy importantes:

- **COSTE**
Los cambios a realizar no afectan el tiempo de entrega.
- **TIEMPO**
Debe ser el menor posible pero siempre y cuando cumpla todos los requerimientos especificados durante el desarrollo del sistema.
- **CALIDAD**
Poseer una mayor calidad en el menor tiempo posible.
- **ÁMBITO**
Especificado por medio de los programadores.

(Bustamante & Rodriguez, 2014)

4.2.3.3 VENTAJAS DE LA METODOLOGÍA XP

Las ventajas que se pueden obtener al utilizar la metodología XP son las siguientes:

- Potenciar el trabajo en grupo durante el desarrollo de la aplicación.
- Entregar periódicamente avances del sistema.
- Realizar pruebas continuas hasta obtener la aplicación final.
- Los cambios que se realizan en la aplicación no deben aumentar con el tiempo.
- Realizar un desarrollo de manera eficiente, sencilla y clara; para facilitar el rendimiento y su actualización.
- Da lugar a una programación sumamente organizada.
- Ocasiona eficiencias en el proceso de planificación y pruebas.
- Cuenta con una tasa de errores muy pequeña.
- Propicia la satisfacción del programador.

- Fomenta la comunicación entre los clientes y los desarrolladores.
- Facilita los cambios.
- Permite ahorrar mucho tiempo y dinero.
- Puede ser aplicada a cualquier lenguaje de programación.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.
- La XP es mejor utilizada en la implementación de nuevas tecnologías.

4.2.3.4 PRINCIPIOS BÁSICOS

La metodología XP utiliza 12 principios básicos que se dividen en cuatro categorías: Retroalimentación a escala fina, Proceso continuo, Entendimiento compartido y Bienestar del programador.

- **RETROALIMENTACIÓN A ESCALA FINA**
 - **PRINCIPIO DE PRUEBAS:** Lo primero que se debe hacer es establecer un período de pruebas de aceptación del programa, en el cual se definirán las entradas y salidas del sistema. Básicamente se define lo que debe hacer el software desarrollado. Como si fuese una caja negra.
 - **PLANIFICACIÓN:** El cliente (o su representante) escribirá sus necesidades para definir concretamente las actividades que el sistema debe realizar. En esta fase se creará un documento que contendrá historias de usuario que forman el plan de liberación, el cual define los tiempos de entrega de la aplicación para poder recibir feedback por parte del cliente.

- **CLIENTE EN EL SITIO:** El cliente (o su representante) deberá formar parte del equipo de desarrollo. Se le dará poder para determinar los requisitos de la aplicación, definir la funcionalidad y dar prioridad a determinadas cosas. Gracias a esto, habrá una fuerte interacción con los programadores, disminuyendo así el tiempo de comunicación y la cantidad de documentación a redactar. El cliente estará con el equipo durante todo el proceso de desarrollo del proyecto.

- **PROGRAMACION EN PAREJAS:** Este punto junto con el anterior son los más radicales de esta metodología. Consiste en escribir código en parejas compartiendo una sola máquina. Según los experimentos ya realizados sobre este método, se producen mejores y más consistentes aplicaciones a igual o menor coste.

- **PROCESO CONTINUO**
 - **INTEGRACIÓN CONTINUA:** Consiste en implementar progresivamente las nuevas características del software. En lugar de crear versiones estables en función de una planificación previamente realizada, los programadores reúnen su código y reconstruyen el proyecto varias veces al día si hace falta.

 - **REFACTORIZACIÓN:** Mediante la constante eliminación de código duplicado y/o ineficiente los equipos de programación mejoran el diseño del sistema. El código se evalúa continuamente para ofrecer la mayor calidad posible.

- **ENTREGAS PEQUEÑAS:** El producto es evaluado en un ambiente real mediante la colocación de un sistema sencillo en producción el cual se actualizará rápidamente, es decir, cada 2 semanas (3 como máximo) el software será puesto en producción.
- **ENTENDIMIENTO COMPARTIDO**
 - **DISEÑO SIMPLE:** El mejor programa será aquel que cumpla con los requisitos y sea más simple. Es importante proporcionar un software que cubra las necesidades de un cliente. Ni más ni menos.
 - **METÁFORA:** expresa la visión evolutiva del proyecto y define los objetivos del sistema mediante una historia.
 - **POSESIÓN COMBINADA DEL CÓDIGO:** El código tiene propiedad compartida. Nadie es propietario de nada, ni siquiera de lo que ha desarrollado. Todos los programadores son "dueños" de todo el código. Según esta metodología, cuantos más programadores hayan trabajando en una parte de código, menos errores tendrá.
 - **ESTÁNDAR DE PROGRAMACIÓN:** define las reglas para escribir y documentar código, además de cómo se comunican las diferentes piezas de código desarrolladas por diferentes equipos. El objetivo de esto es que parezca que el código ha sido escrito por una única persona.
- **BIENESTAR DEL PROGRAMADOR**
 - **SEMANA DE 40 HORAS:** Los programadores cansados escriben peor código. Es importante minimizar las horas extras y

mantener a los programadores frescos y descansados. De esta manera, se generará mejor código. Si es necesario hacer horas extras, quiere decir que el proyecto está mal planificado.

A continuación, la figura 03 presenta los principios básicos de la metodología XP.

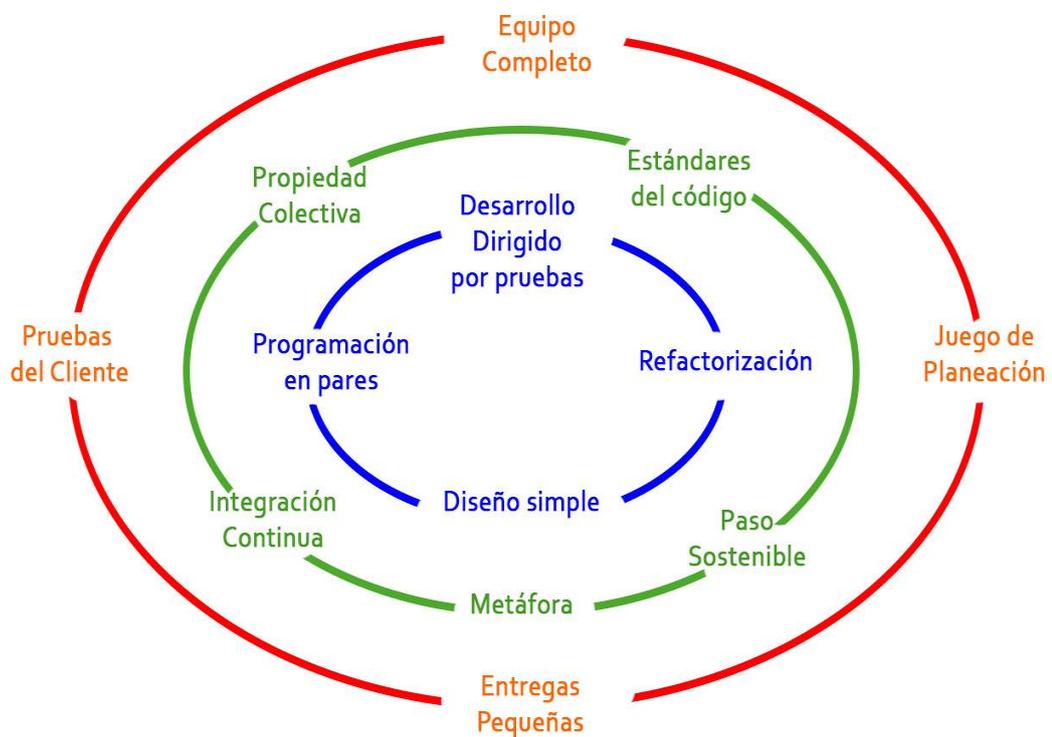


Figura 3 Principios de la Metodología XP
Fuente: Diseño propio

4.2.3.5 ROLES DE LA METODOLOGÍA XP

Los roles que posee la metodología XP permite a los desarrolladores organizar y distribuir el trabajo de manera más equilibrada y eficiente.

Estos roles se pueden resumir de la siguiente manera:

- **PROGRAMADOR**
Responsable de construir, analizar, programar, tomar decisiones y realizar pruebas del sistema.
- **JEFE DE PROYECTO**
Responsable de coordinar, gestionar y administrar las reuniones para considerar las condiciones de cómo avanza el proyecto.
- **CLIENTE**
Persona que debe especificar qué construir, cuándo y dónde realizar las pruebas.
- **ENCARGADO DE PRUEBAS**
Delegado de ayudar al cliente para que las pruebas sean realizadas y superadas.

- **RASTREADOR**
Responsable de obtener datos históricos; encargados de observar sin molestar durante el desarrollo del sistema.
- **ENTRENADOR**
Facultado de visualizar el proceso y desarrollo del sistema, desde un segundo plano.

Cuando se utilizan los roles de la metodología XP se trata al cliente como parte del equipo de desarrolladores, lo que evita la redundancia de información, y permite que se realicen pruebas constantes de la aplicación. Otro beneficio de esta inclusión es que se efectúan cambios mientras avanza el desarrollo según los requerimientos solicitados evitando pérdidas de tiempo.

(Bustamante & Rodriguez, 2014)

4.2.3.6 ARTEFACTO DE LA METODOLOGÍA XP

Los artefactos de la metodología XP son: Historias de usuario, iteración, tarjeta CRC.

A) HISTORIA DE USUARIO:

Las historias de usuario explican de una manera simple y entendible los requisitos de la empresa ya

que son escritas por los clientes involucrados en el desarrollo utilizando un lenguaje común (sin tecnicismos).

Para su elaboración se utilizan tarjetas de papel o a través de una computadora. En estas se plasman los requisitos del usuario con lo que el desarrollador podrá visualizar el ámbito y fijar prioridades para el desarrollo de la aplicación, así como generar estimaciones, consecuencias y definir procesos a ser realizados.

Por medio de las historias de usuario se puede formalizar de manera casi exacta la estimación del tiempo que llevará realizar el desarrollo de la aplicación. Permiten proporcionar la estimación de riesgo de cada requisito y verificar si se cumple correctamente con lo solicitado cuando se realicen las pruebas.

También ayudan a definir las funciones que realizan los usuarios cuando utilizan el software y determinar quiénes deben poseer acceso ilimitado y limitado para evitar errores.

Cada una de las historias debe ser limitada y clara para ser implementada por los desarrolladores máximo en tres semanas; si se requiere más tiempo esta debe ser dividida entre dos personas para obtener un mejor desempeño.

Al inicio es común que no se especifiquen todas las historias de usuario, pero estas se incrementarán al realizar las iteraciones. Se debe tomar en cuenta que para cada historia de usuario debe existir una prueba de aceptación.

En la tabla 01 se visualiza el modelo de ficha para realizar las historias de usuario.

Historia de usuario	
Número:	Usuario:
Nombre historia de usuario:	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados:	Iteración asignada:
Descripción:	
Observación:	

*Tabla 1 Modelo de historia de usuario
Fuente: Ron E. Jeffries, 1999-2009*

B) ITERACIÓN

Las iteraciones permiten realizar un análisis más exhaustivo antes de implementar una historia de usuario; de esta forma el desarrollador en conjunto con el cliente deben realizar las fichas de tareas para obtener la información necesaria para un mejor desarrollo.

Las iteraciones son útiles para verificar el avance del proyecto; entre menos errores este tenga se puede observar un gran avance en el mismo, tanto en calidad como en tiempo. Dentro de cualquier desarrollo se establece como mínimo tres interacciones.

En la tabla 02 se visualiza el modelo de ficha para realizar las tareas.

Tarea	
Número tarea:	Nombre historia:
Programador Responsable:	
Tipo de tarea: Desarrollo/Corrección/ Mejora/Otra	Puntos estimados:
Descripción	

*Tabla 2 Modelo de ficha para tareas
Fuente: Ron E. Jeffries, 1999-2009*

C) TARJETAS CRC (TARJETAS, RESPONSABILIDADES Y COLABORACIÓN)

Una tarjeta CRC representa una clase en la programación orientada a objetos.

Son utilizadas por la metodología XP para diseñar el sistema uniendo todas las funcionalidades. Ayudan a definir actividades mientras se está diseñando el sistema.

Como se visualiza en la tabla 03 la tarjeta CRC representa: la clase a la que pertenece el objeto, las responsabilidades que deben realizar el objeto y las colaboraciones con otras clases para conocer cómo se comunican entre ellas y cumplen con cada responsabilidad del objeto.

Modelo Nombre de la Clase	
Responsabilidades (Métodos o atributos de la clase)	Colaboradores (Clases con las que existe comunicación para realizar las responsabilidades)

*Tabla 3 Modelo nombre de la clase
Ron E. Jeffries, 1999-2009*

Las ventajas de crear las tarjetas CRC son las siguientes:

- Permiten contribuir al diseño de todo el equipo del proyecto.
- Permiten reducir el modo de pensar procedural y apreciar la tecnología de objetos.
- Las tarjetas CRC son simples, fáciles de comprender y flexibles de realizarlas.
- Se visualiza en forma resumida todo el funcionamiento de una clase.

4.2.3.7 FASES DE LA METODOLOGÍA XP

En la figura 12 se observa las fases de la metodología XP para la creación de la aplicación.

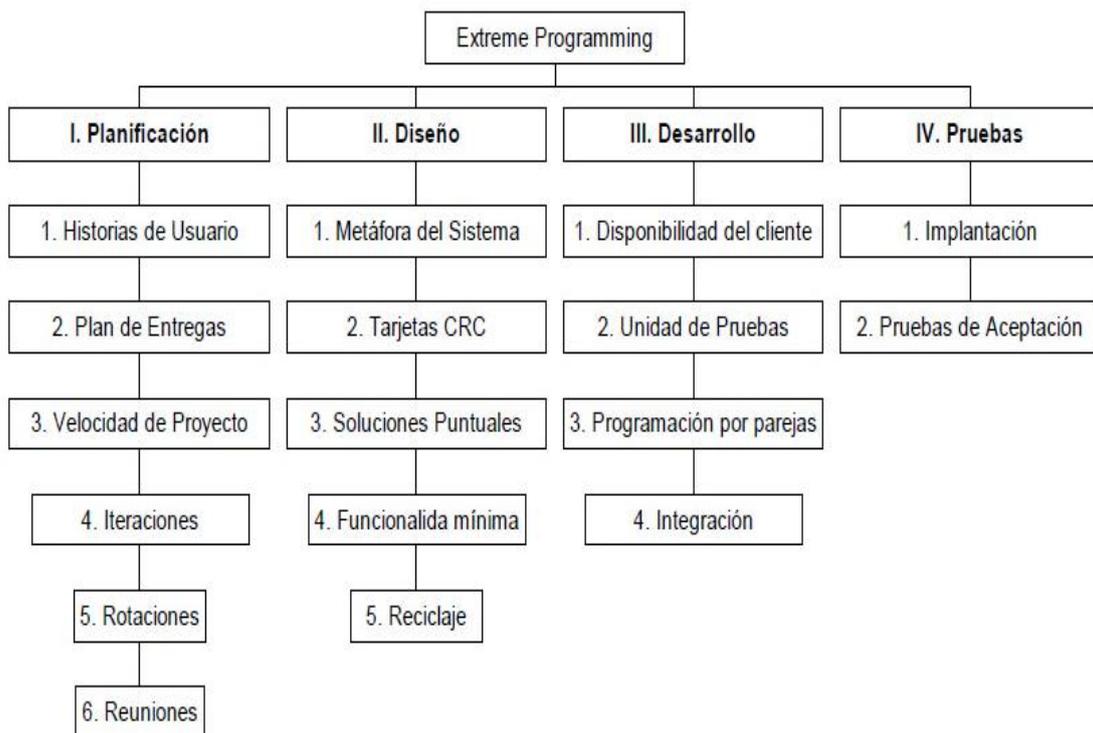


Figura 4 Fases de la metodología XP

Fuente: BATALLA, Líber, *Extreme Programming (XP)*, 2006

I) PLANIFICACIÓN

Permite obtener y recolectar la información necesaria para conocer lo que el cliente requiere; se debe estar al tanto de los objetivos del proyecto, el tiempo que tomará desarrollar el sistema, personas involucradas en el desarrollo, presentación de prototipos y la evaluación de calidad.

Esta fase lo componen las siguientes etapas:

a) Historias de Usuario

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc.

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.

b) Plan de entregas

Consiste en facilitar un cronograma como resultado de la información obtenida de las historias de usuario, que debe ser analizado por todos los actores del proyecto. Dicho plan puede ser modificado o actualizado de ser requerido, para lo cual se debe realizar una nueva reunión entre los actores.

Este cronograma tiene que especificar el tiempo establecido para la implementación de las historias de usuario, prioridad de desarrollo de cada una y especificar la presentación de pequeñas versiones del proyecto.

Por medio del plan de entregas se debe obtener los siguientes factores:

- **Objetivos:** Clasificar las historias de usuario.
- **Tiempo:** Desarrollar y publicar los prototipos.
- **Personas involucradas:** Especificar el número de personas necesarias.
- **Evaluaciones:** Verificar la calidad del proyecto.

c) 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".

d) Iteraciones

Todo proyecto que siga la metodología X.P. 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 "Plan de entregas" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

e) Rotaciones

Permite mantener un lazo de comunicación con todos los actores del proyecto para que los involucrados conozcan el diseño y la codificación del sistema en su totalidad.

Esta característica permite que una persona rote por todas las funcionalidades durante el desarrollo del sistema, y dado que se crean patrones de codificación todos los individuos involucrados son aptos de entender el funcionamiento del mismo. Así la metodología XP trata de evitar cuellos de botella o códigos que no

pueden ser reutilizados por otros programadores.

f) Reuniones

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.

II) DISEÑO

Fase en la que se crea estándares y patrones para realizar la codificación; de esta forma se obtendrá un código más eficiente, con calidad y comprensión para todos los involucrados en el desarrollo. Esta fase lo determinan los siguientes componentes:

a) Metáfora del Sistema

En la metodología XP es importante crear una metáfora entre los desarrolladores y el cliente. La metáfora consiste en narrar cómo funcionaría el sistema en el momento de la entrega final. Permite conocer el alcance, propósito y es utilizada como una guía para desarrollar el sistema.

“Una metáfora para el sistema es una historia que todo el mundo puede contar acerca de cómo el sistema funciona”.
Kent Beck.

Para un mejor desarrollo de la metáfora se debe especificar nombres (se debe definir las nomenclaturas de clases y métodos del sistema) y analizar posibles problemas que puedan generarse.

Las ventajas de crear una metáfora son las siguientes:

- Conocer la mayoría o todo lo que se va a implementar.
- Permite entender de forma clara el objetivo del sistema.
- El cliente pueda hablar cómodamente sobre el funcionamiento del sistema.
- Los desarrolladores conocen el mismo lenguaje y si hay nuevos desarrolladores les permite adaptarse más rápido y sin dificultades.
- Permite implementar los patrones de diseño que podrían ser utilizados durante el desarrollo del sistema.

b) Tarjetas CRC

Las tarjetas Clase – Responsabilidad - Colaboración su técnica consiste en dibujar una tarjeta por cada clase u objeto, y dividirla en tres zonas:

- En la parte superior, el nombre de la clase.
- Debajo, en la parte izquierda, las responsabilidades de dicha clase. Son sus objetivos, a alto nivel.
- A la derecha de las responsabilidades, los colaboradores, que son otras clases que ayudan a conseguir cumplir a esta con sus responsabilidades

c) Soluciones Puntuales

La metodología XP utiliza las soluciones puntuales llamadas Spike, para

encontrar posibles respuestas a problemas (técnico, arquitectura del software, estimación de tiempo) que pueden aparecer durante el desarrollo del sistema.

Estos pequeños programas son utilizados para probar y evaluar soluciones; son considerados desechables, ya que una vez utilizados y valorados los resultados obtenidos de su ejecución se los puede descartar.

La metodología XP da preferencia a la solución de los problemas de manera automática y no manual; de esta forma se trata de disminuir el riesgo y aumentar la confiabilidad en el desarrollo de alguna historia de usuario.

d) Funcionalidad mínima

La metodología XP indica que la funcionalidad extra consume recursos, representa una pérdida de tiempo y quizás nunca se utilice esas funcionalidades.

Por esta razón se da prioridad a una eficiente programación, focalizando en quien realiza la tarea programada para cada día, creando el código necesario y verificando que realice su trabajo correctamente. De esta forma, se evita crear código que quizás no sea utilizado. Por lo tanto 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.

e) Reciclaje

Es una actividad que utiliza la metodología XP para obtener un mejor control cuando se programa.

Consiste en eliminar la redundancia de código, simplificar y crear una estructura fácil de comprender y modificar para futuros cambios.

Es necesario revisar el código cada vez que se termina una iteración, tomando en cuenta los siguientes puntos:

- Desplazamiento fácil en el sistema. (Si/No)
- Existen varios objetos que realizan lo mismo. (Si/No)
- Existe suficiente espacio para el contenido del sistema. (Si/No)
- Existe código de funciones que fueron eliminadas. (Si/No)
- Comunica la visión principal u original del sistema. (Si/No)

Por lo tanto, 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.

III) DESARROLLO

La metodología XP indica que el código expresa tú interpretación del problema, así podemos utilizar el código para comunicar, para hacer más tus ideas, y por tanto para aprender y mejorar.

a) Disponibilidad del cliente

Para la metodología XP es necesario crear una comunicación fluida, personal y directa entre el cliente y el equipo de trabajo en todas las fases del proyecto; de este modo se reduce el tiempo de desarrollo del sistema y evita malos entendidos por cuanto se trata de una comunicación oral y no escrita.

Las funciones que realiza un cliente son:

- Realizar las historias de usuario y especificar el plan de entrega.
- Validar el sistema cuando se realicen las pruebas funcionales.
- Toma de decisiones sobre las funciones que realiza el sistema.
- Responder dudas, determinar prioridades, especificar mejoras y resolver posibles problemas.
- Tener un compromiso de trabajo en equipo.
- Estar presente antes de desarrollar cada historia de usuario y comunicar más detalladamente su funcionamiento.
- Determinar cuando el sistema puede ser colocado en producción.

Por medio de la comunicación permanente se puede prevenir a tiempo cualquier situación o funcionalidad que no es deseable o necesaria para el proyecto, sin necesidad de esperar a estar en el ciclo de desarrollo para comenzar a realizar las correcciones, que a la postre puedan ser consideradas muy complicados.

b) Unidad de pruebas

La metodología XP recomienda crear las pruebas de unidad antes del código y de la implementación del sistema, para obtener un conocimiento global del comportamiento del sistema.

Estas pruebas deben ser ejecutadas cada vez que exista una modificación en el sistema. Como resultado se obtiene un código conciso y simple ya que se implementa únicamente las características necesarias que requiere el sistema.

Todas las características que tiene el sistema tienen que ser probadas constantemente, lo que permite adaptarse a nuevos cambios.

Las pruebas de unidad tienen las siguientes características:

- Se crean una vez y se ejecutan constantemente durante todo el desarrollo del proyecto.
- Verifican el funcionamiento correcto evitando redundancia.
- Al principio se obtiene una prueba y función simple; se van agregando nuevos requerimientos dependiendo de

las etapas en la que se encuentre hasta obtener toda la funcionalidad requerida.

- Una funcionalidad se la determina como finalizada cuando paso correctamente todas las pruebas de unidad.

c) Programación por parejas

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.

d) Integración

El objetivo de la metodología XP fomenta que la programación siempre debe ser en pareja, para desarrollar el código de manera continua, rotar funciones y escribir con los mismos estándares.

Existen dos tipos de integraciones que utiliza la metodología XP, que son:

- **Integración continua**

Se realiza la integración de código cada vez que esté listo por los desarrolladores, para luego ser probado; esta debe ser realizada, de ser posible, varias

veces en un día. Aquí todos trabajan con la última versión del proyecto.

Al utilizar este tipo de integración se detecta o evade de manera eficiente los posibles problemas de compatibilidad que pueden surgir.

- **Integración secuencial**

Es recomendable realizar la integración de código en una máquina para evitar posibles problemas o redundancia en el código; ya que los desarrolladores trabajan en paralelo en donde se debe elegir únicamente a una pareja de desarrolladores que integren, prueben y realicen cambios en el código.

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:

a) Implantación

- 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.

- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican.
- Test de aceptación: Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

b) Prueba de aceptación

Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.

4.2.3.8 MEJORES PRÁCTICAS UTILIZANDO LA METODOLOGÍA XP

- La metodología XP puede reutilizar el código al desarrollar un sistema unificado y entendible para todos los programadores, así como disponer de la información organizada y evitar errores o redundancias.
- La metodología XP es muy utilizada para desarrollar aplicaciones pequeñas o donde la aplicación deba ser actualizada constantemente (en este aspecto es muy eficiente); pero en una aplicación grande está metodología se vuelve inconveniente.
- La metodología XP especifica que se debe realizar pruebas constantes para asegurar que la aplicación está realizando todas las funciones requeridas por el cliente y que no existan desviaciones en la aplicación, lo que permite reducir el riesgo de fallas que se puedan producir en ésta.

4.2.4 BASE DE DATOS MySQL

4.2.4.1 INTRODUCCIÓN

(Gilfillan, 2003)

Una base de datos relacional es un conjunto ordenado de datos, los cuales normalmente están almacenados en uno o más ficheros. Los datos están estructurados en tablas, y estas pueden tener referencias cruzadas. La existencia de estas referencias o relaciones es lo que da el nombre de relacional a este tipo de bases de datos.

Por ejemplo, una base de datos puede almacenar información acerca de los clientes de una compañía.

La base de datos estaría compuesta de una tabla de clientes (nombre, dirección, etc.), una tabla con los productos que la empresa ofrece, y finalmente, una tabla con los pedidos de la empresa. A través de la tabla de pedidos sería posible acceder a los datos de las otras dos tablas (por ejemplo, a través de los identificadores de cliente y producto).

Como ejemplos de sistemas de bases de datos relacionales tenemos MySQL, Oracle, PostgreSQL, Microsoft SQL Server, IBM DB2, etc.

Estos sistemas proporcionan herramientas que permiten almacenar, procesarlos, analizar y ordenar de manera eficiente los datos. Además, todo esto ha de ser posible que funcione sobre una red, por lo que normalmente hablaremos de un servidor de bases de datos.

Como contraposición a las bases de datos relacionales tenemos las bases de datos orientadas a objetos. Estas pueden almacenar objetos individuales sin estar estructurados en tablas. Estas bases de datos permiten acceder directamente a objetos definidos en el entorno de un lenguaje de programación (PHP, C++, Java). A pesar de que ha habido una serie de tendencias hacia bases de datos relacionales en los últimos años, estas solo han encontrado pequeños nichos de mercado.

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y

los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

4.2.4.2 PRINCIPALES CARACTERÍSTICAS

- ✓ Escrita en C/C++.
- ✓ Funciona en múltiples plataformas: GNU/Linux y Windows, pero también en AIX, Amiga, FreeBSD
- ✓ Es multihilo y puede aprovechar la disponibilidad de multiprocesadores.
- ✓ Es software libre (licencia GPL) y también está disponible bajo licencia comercial (por ejemplo, para integración).
- ✓ Tiene distintos motores de almacenamiento (tipos de tablas), algunos con soporte transaccional y otro no.
- ✓ Es muy, muy rápida... si se utilizan motores de almacenamiento no transaccionales.
- ✓ Gran soporte del estándar SQL.
- ✓ Numerosas opciones de conexión (PHP, ODBC, sockets TCP/IP, JDBC, API de C/C++)
- ✓ Soporte multilinguaje.
- ✓ Fácil de instalar, usar y administrar.
- ✓ Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.

4.2.5 LENGUAJE DE PROGRAMACIÓN JAVA

4.2.5.1 INTRODUCCIÓN

(Deitel, 2012)

Java es un tipo de lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995.

A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

4.2.5.2 PRINCIPALES CARACTERISTICAS

(Schildt, 2009)

Ninguna discusión sobre la historia de Java está completa sin tener en cuenta las cualidades que describen a Java. Aunque las razones fundamentales de la invención de Java fueron la portabilidad y la seguridad, existen otros factores que también desempeñaron un papel importante en el modelado de la forma final del lenguaje. Las consideraciones clave fueron resumidas por el equipo de Java en la siguiente lista de términos:

a) **Simple**

Java fue diseñado con la finalidad de que su aprendizaje y utilización resultaran sencillos para el programador profesional. Contando con alguna experiencia en programación es fácil dominar Java. Si ya se comprenden los conceptos básicos de programación orientada a objetos, aprender Java será aún más sencillo. Lo mejor de todo, si se tiene experiencia programando con C++, cambiar a Java requiere sólo un poco de esfuerzo. La mayoría de los programadores de C/C++ no tienen prácticamente ningún problema al aprender Java porque Java hereda la sintaxis y muchas de las características orientadas a objetos de C++.

b) Orientado a Objetos

Aunque influido por sus predecesores, Java no fue diseñado para tener un código compatible con cualquier otro lenguaje. Esto dio la libertad al equipo de Java de partir de cero. Una consecuencia de esto fue una aproximación clara, pragmática y aprovechable de los objetos. Java ha tomado prestadas muchas ideas de entornos de orientación a objetos de las últimas décadas, logrando un equilibrio razonable entre el modelo purista “todo es un objeto” y el modelo pragmático “mantente fuera de mi camino”. El modelo de objetos en Java es sencillo y de fácil ampliación, mientras que los tipos primitivos como los enteros, se mantienen como “no objetos” de alto rendimiento.

c) Robusto

El ambiente multiplataforma de la Web es muy exigente con un programa, ya que éste debe ejecutarse de forma fiable en una gran variedad de sistemas. Por este motivo, la capacidad para crear programas robustos tuvo una alta prioridad en el diseño de Java. Para ganar fiabilidad, Java restringe al programador en algunas áreas clave, con ello se consigue encontrar rápidamente los errores en el desarrollo del programa. Al mismo tiempo, Java lo libera de tener que preocuparse por las causas más comunes de errores de programación. Como Java es un lenguaje estrictamente tipificado, comprueba el código durante la compilación. Sin embargo, también comprueba el código durante la ejecución. De hecho en Java es imposible que se produzcan situaciones en las que aparecen a menudo errores difíciles de localizar. Una característica clave de

Java es que se conoce que el programa se comportará de una manera predecible en diversas condiciones.

Para comprender la robustez de Java, consideremos dos de las causas de fallo de programa más importantes: la gestión de memoria y las condiciones de excepción no controladas (errores en tiempo de ejecución). La gestión de la memoria puede convertirse en una tarea difícil y tediosa en los entornos de programación tradicionales.

Por ejemplo en C/C++ el programador debe reservar y liberar la memoria dinámica en forma manual. Esto puede ocasionar problemas, ya que en ocasiones los programadores olvidan liberar memoria que ha sido reservada previamente o, peor aún, intentan liberar memoria que otra parte de su código todavía está utilizando.

Java elimina virtualmente este problema, ya que se encarga en lo interno tanto de reservar la memoria como de liberarla. De hecho, la liberación es completamente automática, ya que Java dispone del sistema de recolección de basura que se encarga de los objetos que ya no se utilizan. En los entornos tradicionales, las excepciones surgen, a menudo, en situaciones tales como la división entre cero, o "archivo no encontrado", y se deben gestionar mediante construcciones torpes y difíciles de leer. En esta área, Java proporciona la gestión de excepciones orientada a objetos. En un programa de Java correctamente escrito, todos los errores de ejecución pueden y deben ser gestionados por el programa.

d) Multihilo

Java fue diseñado para satisfacer los requisitos del mundo real, de crear programas en red interactivos. Para ello, Java proporciona la programación multihilo que permite la escritura de programas que hagan varias cosas simultáneamente. El intérprete de Java dispone de una solución elegante y sofisticada para la sincronización de múltiples procesos que permiten construir fácilmente sistemas interactivos. El método multihilo de Java, de utilización sencilla, permite ocuparse sólo del comportamiento específico del programa, en lugar de pensar en el sistema multitarea.

e) Arquitectura neutral

Una cuestión importante para los diseñadores de Java era la relativa a la longevidad y portabilidad del código. Uno de los principales problemas a los que se enfrentan los programadores es que no tienen garantía de que el programa que escriben hoy podrá ejecutarse mañana, incluso en la misma máquina. Las actualizaciones de los sistemas operativos y los procesadores, y los cambios en los recursos básicos del sistema, conjuntamente, pueden hacer que un programa funcione mal.

Los diseñadores de Java tomaron decisiones difíciles en el lenguaje y en el intérprete Java en un intento de cambiar esta situación. Su meta fue "escribir una vez; ejecutar en cualquier sitio, en cualquier momento y para siempre". Ese objetivo se consiguió en gran parte.

f) Interpretado y de alto rendimiento

Como antes se ha descrito, Java permite la creación de programas que pueden ejecutarse en

diferentes plataformas por medio de la compilación en una representación intermedia llamada código bytecode. Este código puede ser interpretado en cualquier sistema que tenga un intérprete Java. Como ya se explicó el bytecode fue cuidadosamente diseñado para que fuera fácil de traducir al código nativo y poder conseguir así un rendimiento alto utilizando la característica de JIT. Los intérpretes de Java que proporcionan esta característica no pierden ninguna de las ventajas de un código independiente de la plataforma.

g) Distribuido

Java fue ideado para el entorno distribuido de Internet, ya que gestiona los protocolos TCP/IP. De hecho, acceder a un recurso utilizando un URL no es muy distinto a acceder a un archivo. Java soporta invocación remota de métodos (RMI, por sus siglas en inglés). Esta característica permite a un programa invocar métodos de objetos situados en computadoras diferentes a través de la red.

h) Dinámico

Los programas de Java se transportan con cierta cantidad de información que se utiliza para verificar y resolver el acceso a objetos en el tiempo de ejecución. Esto permite enlazar el código dinámicamente de una forma segura y viable. Esto es crucial para la robustez del entorno de Java, en el que pequeños fragmentos de bytecode pueden ser actualizados dinámicamente en un sistema que está ejecutándose.

4.2.5.3 ENTORNO DE DESARROLLO TÍPICO EN JAVA

Por lo general, los programas en Java pasan a través de cinco fases: Edición, Compilación, Carga, Verificación, Ejecución.

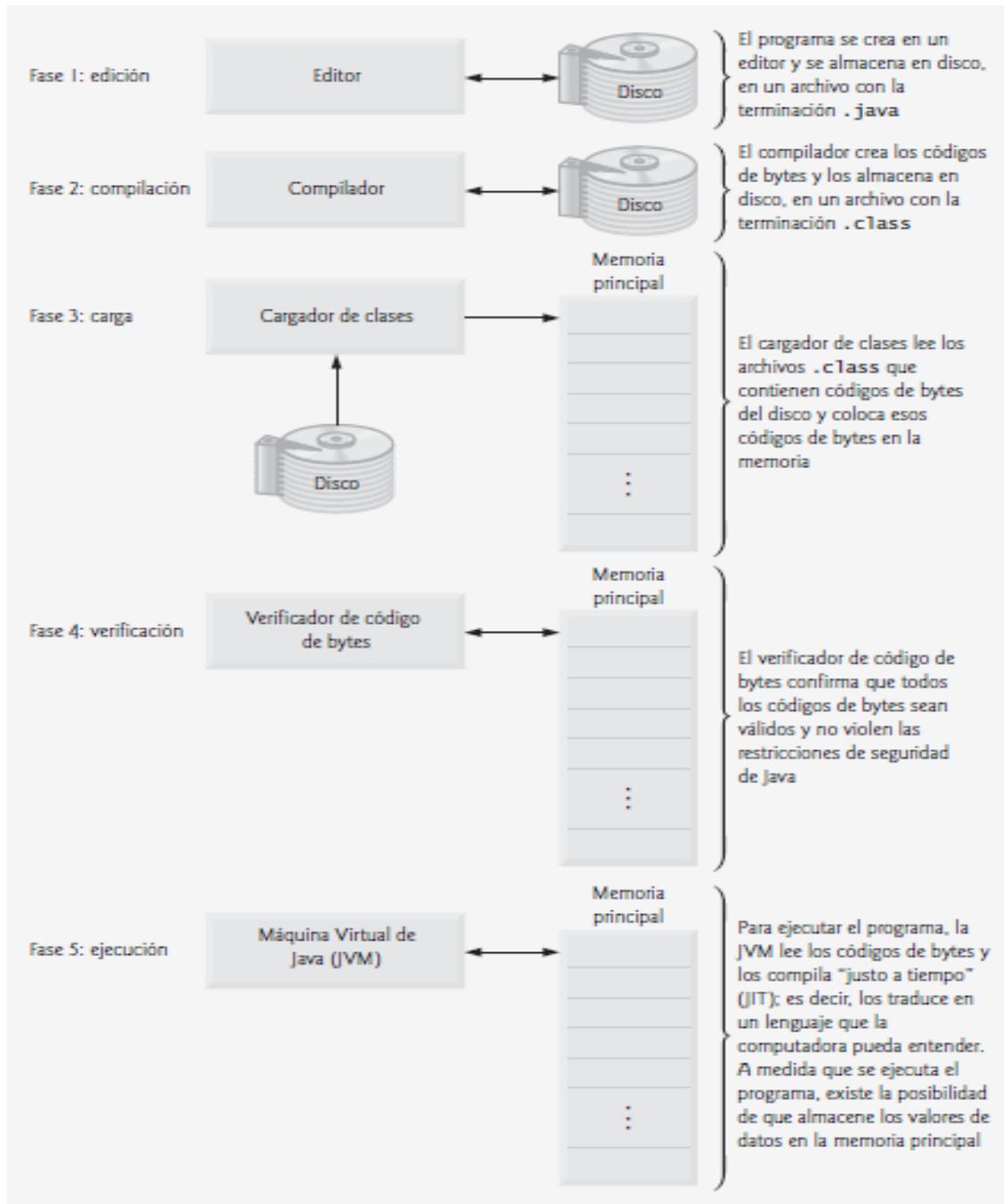


Figura 5 Entorno de desarrollo típico en Java

Fuente: (Deitel, 2012)

4.2.6 HERRAMIENTAS DE DESARROLLO

4.2.6.1 MySQL WORKBENCH

(Oracle, Oracle - MySQL, 2016)

a) Definición

MySQL Workbench es una herramienta visual unificada para diseñadores de base de datos, desarrolladores y DBAs. MySQL Workbench ofrece modelado de datos, desarrollo de SQL y herramientas completas de administración para la configuración de servidores, administración de usuarios, backup y mucho más.

MySQL Workbench está disponible en Windows, Linux y Mac OS X.

b) Características de MySQL Workbench

- Permite a un DBA, desarrollador o diseñador de datos diseñar visualmente, modelar, generar y administrar bases de datos. Incluye todo lo que necesita un modelador de datos para crear modelos complejos de ER, ingeniería directa e inversa y también ofrece características clave para realizar tareas difíciles de gestión de cambios y documentación que normalmente requieren mucho tiempo y esfuerzo.
- Ofrece herramientas visuales para crear, ejecutar y optimizar consultas SQL. El editor de SQL proporciona resaltado de sintaxis de color, auto-completar, reutilización de fragmentos de SQL y el historial de ejecución de SQL. El Panel de conexiones de base de datos permite a los desarrolladores administrar fácilmente las conexiones de base de datos estándar, incluido MySQL

Fabric. El Explorador de objetos proporciona acceso instantáneo al esquema ya los objetos de la base de datos.

- Proporciona una consola visual para administrar fácilmente entornos MySQL y obtener una mejor visibilidad en las bases de datos. Los desarrolladores y los DBA pueden utilizar las herramientas visuales para configurar servidores, administrar usuarios, realizar copias de seguridad y recuperación, inspeccionar los datos de auditoría y ver el estado de la base de datos.
- Proporciona un conjunto de herramientas para mejorar el rendimiento de las aplicaciones MySQL. Los DBA pueden ver rápidamente indicadores clave de rendimiento mediante el Performance Dashboard. Los Informes de rendimiento proporcionan fácil identificación y acceso a puntos de acceso IO, declaraciones SQL de alto costo y más. Además, con un solo clic, los desarrolladores pueden ver dónde optimizar su consulta con el mejor y más fácil de usar Visual Explain Plan.
- Proporciona ahora una solución completa y fácil de usar para migrar Microsoft SQL Server, Microsoft Access, Sybase ASE, PostgreSQL y otras tablas, objetos y datos de RDBMS a MySQL. Los desarrolladores y DBAs pueden convertir rápida y fácilmente aplicaciones existentes para ejecutar en MySQL tanto en Windows como en otras plataformas. La migración también admite la migración de versiones anteriores de MySQL a las últimas versiones.

4.2.6.2 NETBEANS IDE

(Oracle N. , 2016)

NetBeans IDE es una aplicación de código abierto ("open source") diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java.

NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de packs. En NetBeans IDE encontrarás la solución más completa para programar en Java.

NetBeans IDE nos permite desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y web Java, así como aplicaciones HTML5 con HTML, JavaScript y CSS. El IDE también proporciona un gran conjunto de herramientas para desarrolladores de PHP y C / C ++. Es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo.

4.2.6.3 SERVIDOR XAMPP

(Friends, s.f.)

XAMPP es un paquete de instalación independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.

4.2.6.4 IREPORT

Es una herramienta de soporte de la librería JasperReports que ayuda a la edición de los informes complejos mediante un editor gráfico, permitiendo la inclusión de gráficos, imágenes y subinformes. Está desarrollada en código Java y es de libre distribución. Las principales características de la herramienta son las siguientes:

- 100% escrito en Java y además OpenSource y gratuito.
- Maneja el 90% de las etiquetas de JasperReports.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, eclipses, campos de los textfields, cartas, subinformes.
- Soporta JDBC.
- Soporta JavaBeans como origen de datos (estos deben implementar la interfaz JRDataSource).
- Incluye asistentes para crear informes automáticamente.
- Tiene asistentes para generar los subinformes.
- Tiene asistentes para las plantillas.

En definitiva, permite crear cualquier tipo de informe de una manera sencilla y rápida a través de una interfaz gráfica intuitiva, diseñada tanto

para gente que no está familiarizada con esta tecnología, y que desconoce la sintaxis XML de JasperReports, como a usuarios expertos que ya conocían este lenguaje, ahorrándoles tiempo durante el desarrollo de informes muy elaborados.

El resultado de la utilización de iReport es una plantilla que deberá ser rellena con los datos que conformarán el informe. Su diseño puede incluir todo tipo de elementos gráficos. A continuación, se detallan las secciones que componen un reporte en IReport:

- **Title:** Título del reporte. Sólo se muestra en la primera página del mismo
- **Pageheader:** Cabecera de página que puede contener información adicional del reporte
- **Columnheader:** En esta sección se muestran los nombres de los campos a presentar
- **Detail:** En esta sección se despliegan los valores correspondientes a los nombres definidos anteriormente
- **Columnfooter:** Puede presentar valores totales para algún campo de la sección "detail"
- **Pagefooter:** En la parte inferior de cada página para poner fecha, numero de página.
- **Summary:** Sección para incluir gráficos o cálculos totales

Al diseñar un informe podemos seleccionar las etiqueta que consideremos interesantes para nuestro diseño. Todos estos parámetros

constituyen el fichero JRXML, es decir, un diseño de informe que ha de ser compilado por el propio iReport utilizando las clases de JasperReports. Así se consigue un fichero .jasper, el cual servirá de guía para la generación del informe definitivo.

Como todo informe, los datos a mostrar pueden ser estáticos o dinámicos. Por lo tanto iReport admite un gran número de bases de datos, entre ellas se encuentran MySql, Hsql y Oracle. Sin embargo, es interesante realizar un esfuerzo por mantener la separación de las capas, es decir, la capa de persistencia sea independiente de la de presentación, lo que nos facilitará el rápido entendimiento de nuestras aplicaciones y la cómoda realización de modificaciones futuras.

(Junta Andalucía)

4.3.- CONCEPTOS Y DEFINICIONES

4.3.1 DEFINICIÓN DE LOS PROCESOS

4.3.1.1 PROCESO DE VENTA

El ciclo o proceso de venta es la sucesión de pasos que una empresa realiza desde el momento en que intenta captar la atención de un potencial cliente hasta que la transacción final se lleva a cabo.

4.3.1.2 PROCESO DE COMPRAS

Es la secuencia de fases y actividades que transforman, por medio de los recursos disponibles de la organización, una solicitud de un bien o servicio realizada por un departamento en la provisión del bien o la prestación del servicio solicitado por parte del proveedor.

4.3.1.3 PROCESO DE ALMACÉN

Es el proceso de la función logística que trata la recepción, almacenamiento y movimiento dentro de un mismo almacén hasta el punto de consumo de cualquier material – materias primas, semielaborados, terminados, así como el tratamiento e información de los datos generados

4.3.2 SISTEMA

Un sistema es un conjunto de elementos relacionados entre sí y que funcionan como un todo para lograr un objetivo común.

4.3.3 SISTEMA DE INFORMACIÓN

Conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo.

Todos estos elementos interactúan para procesar los datos (incluidos los procesos manuales y automáticos) y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos.

4.3.4 BASE DE DATOS

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. Las bases de datos tradicionales se organizan por campos, registros y archivos.

4.3.5 PROCESO

Es el conjunto de actividades y procedimientos que debe realizar un sistema de información.

4.3.6 COMPILADOR

Es un programa o herramienta encargada de compilar, un compilador toma un texto (código fuente) escrito en un lenguaje de alto nivel y lo traduce a un lenguaje comprensible por las computadoras (código objeto).

4.3.7 JAVA DEVELOPMENT KIT (JDK)

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

CAPITULO V

DESARROLLO DE LA PROPUESTA

V: DESARROLLO DE LA PROPUESTA

5.1. CONFIGURACIONES INICIALES PARA DESARROLLAR EL SISTEMA DE INFORMACIÓN

Para el desarrollo del sistema de información se realizaron diversas configuraciones y descargas de software de internet, para lograr un entorno de desarrollo más adecuado y nos ayude a lograr el objetivo planteado en el desarrollo.

5.1.1 BASES DE DATOS

5.1.1.1 DISEÑO (MODELO RELACIONAL)

Para el desarrollo del sistema de información en primer lugar se tuvo que hacer un análisis de la información a manejar y luego se procedió a diseñar el modelo de la base de datos, este trabajo se realizó con la herramienta MySQL Workbench, se instaló la versión 6.2 de este software.

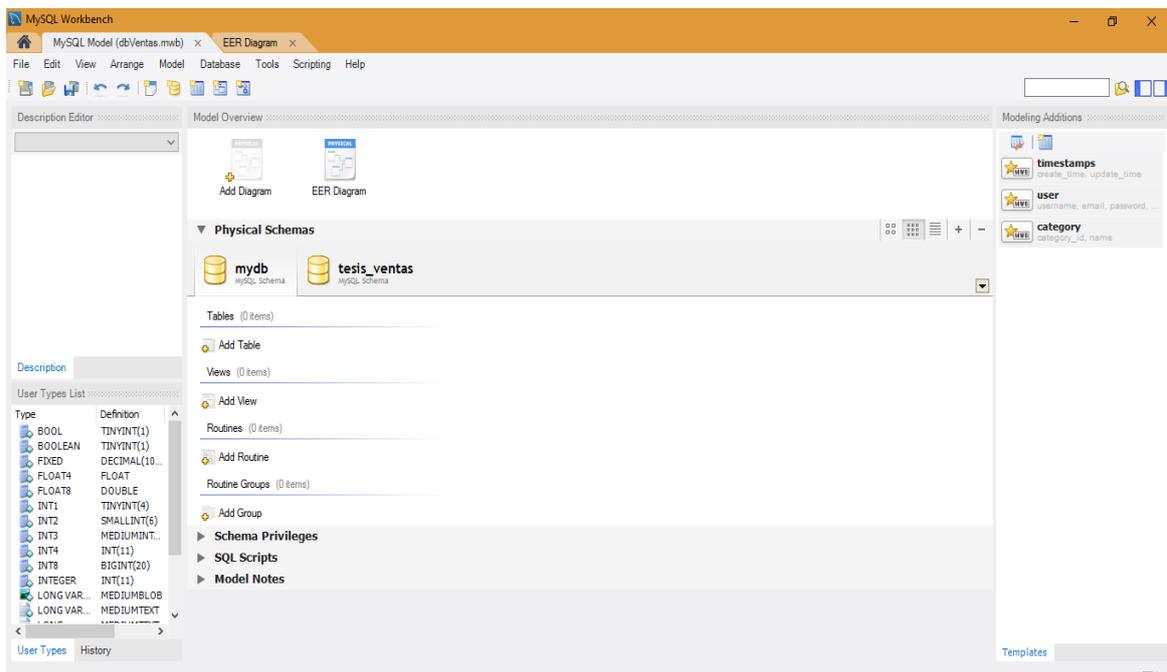


Figura 6 Entorno MySQL Workbench
Fuente: Captura propia

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,”

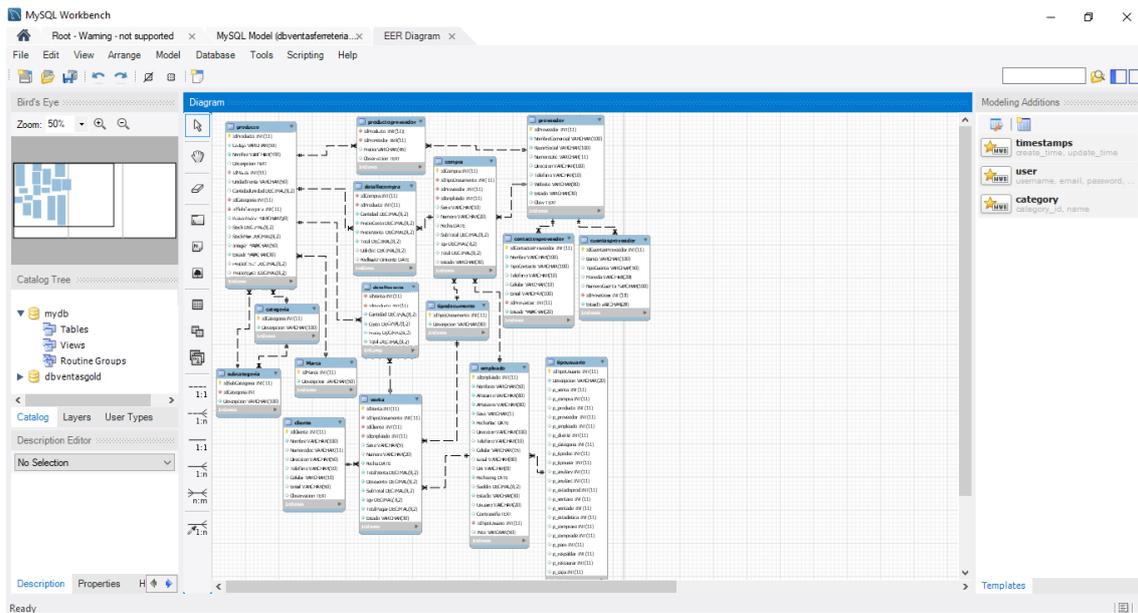


Figura 7 Vista Modelo en Entorno MySQL Workbench
Fuente: Captura propia

5.1.1.2 GESTOR/ADMINISTRADOR DE BASE DE DATOS MySQL (XAMPP)

Para la administración de la base de datos del Sistema se instaló un servidor local XAMPP que incluye en su paquete de instalación el gestor de base de datos phpMyAdmin, el cual utilizaremos en el desarrollo de los procedimientos. Se instaló la versión 3.2.2 de este software.

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,”

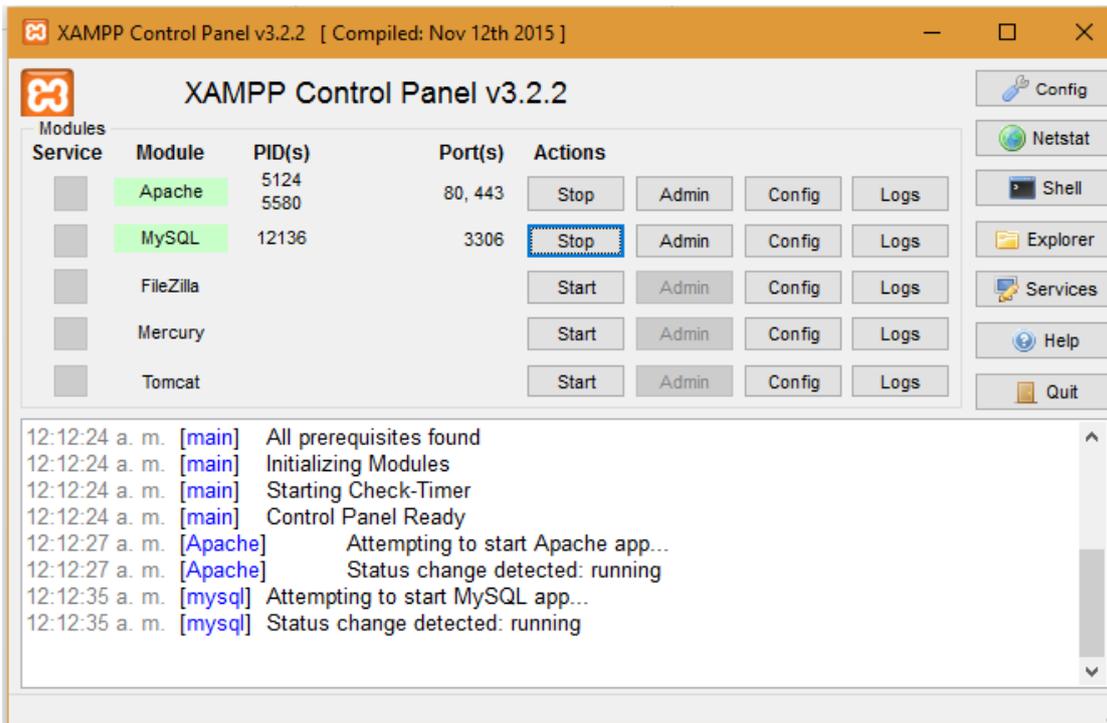


Figura 8 Entorno Servidor Local XAMPP
Fuente: Captura propia

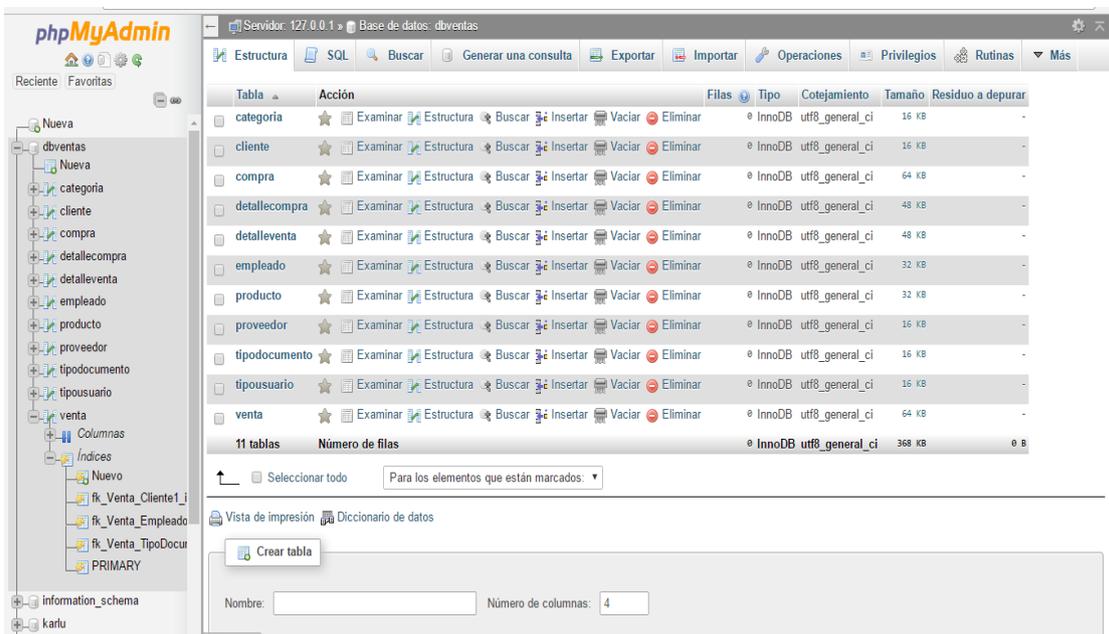


Figura 9 Entorno phpMyAdmin
Fuente: Captura propia

5.1.2 SOFTWARE

Para el desarrollo de sistema se utilizaron diversos programas tanto para la escritura del código fuente, el diseño de formularios de la interface gráfica, complementos de suma importancia para lograr que el espacio de trabajo sea el adecuado y también la arquitectura requerida del sistema.

5.1.2.1 ENTORNO DE DESARROLLO NETBEANS

Software que fue diseñado exclusivamente para trabajar con el lenguaje de programación Java, Netbeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos, estos módulos contienen clases de java escritas para interactuar con las APIs de NetBeans. Se descargó e instaló la versión 8.02 de este programa.

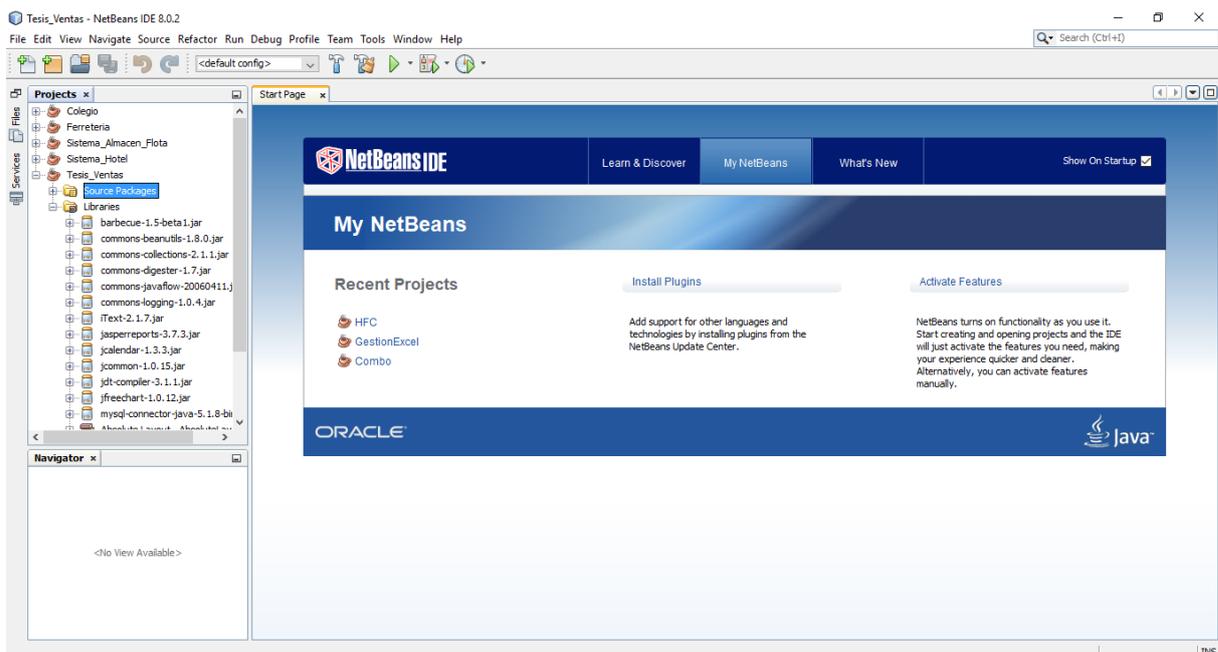
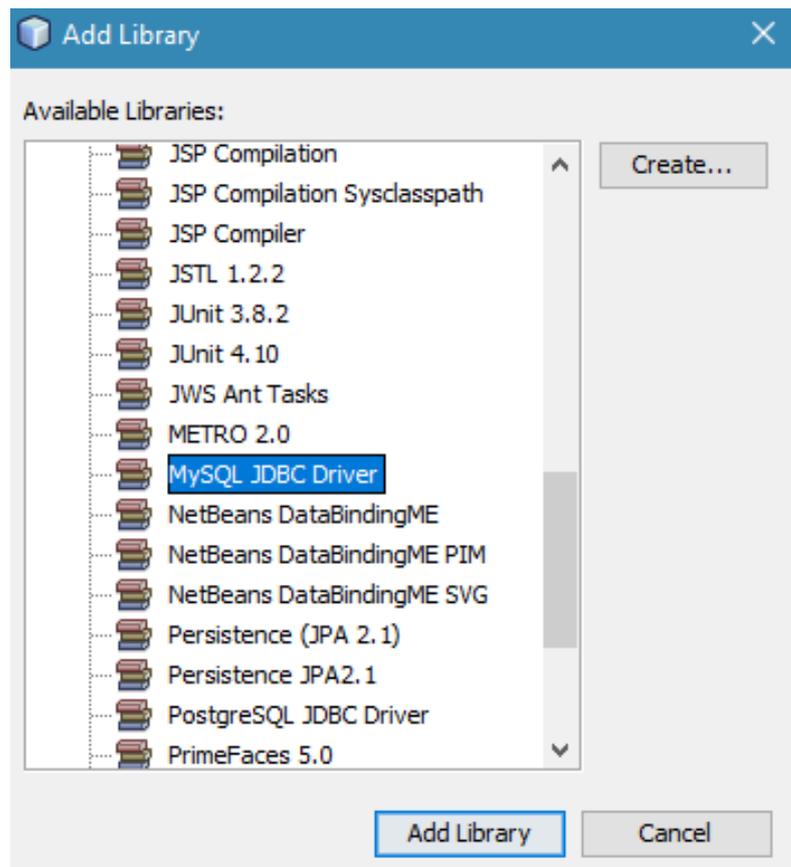


Figura 10 Entorno de desarrollo NetBeans
Fuente: Captura propia

Para conectar nuestro sistema a la base de datos MySQL es necesario el JDBC proporcionado por el mismo NetBeans.

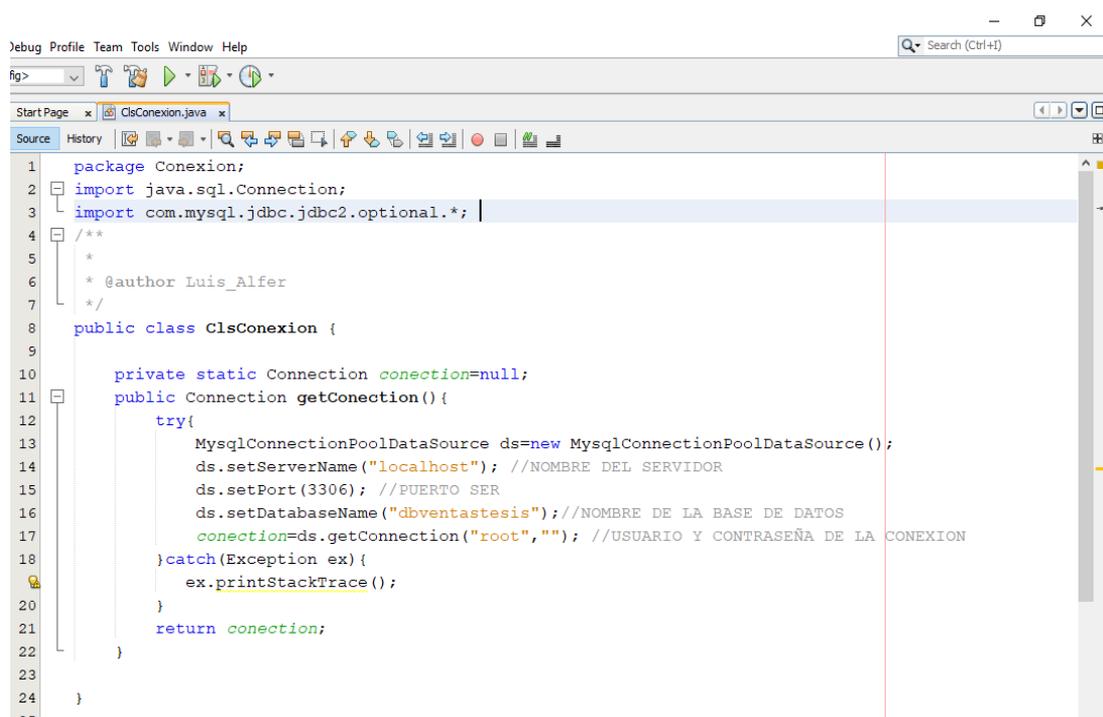


*Figura 11 Configuración del controlador MySQL JDBC Driver
Fuente: Captura propia*

Una vez instalado el controlador MySQL JDBC conectamos nuestra base de datos al entorno de desarrollo Netbeans para empezar el desarrollo del sistema.

Nuestra cadena de conexión a la base de datos es la siguiente.

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,



```
1 package Conexion;
2 import java.sql.Connection;
3 import com.mysql.jdbc.jdbc2.optional.*;
4 /**
5  *
6  * @author Luis_Alfer
7  */
8 public class ClsConexion {
9
10     private static Connection conection=null;
11     public Connection getConnection(){
12         try{
13             MysqlConnectionPoolDataSource ds=new MysqlConnectionPoolDataSource();
14             ds.setServerName("localhost"); //NOMBRE DEL SERVIDOR
15             ds.setPort(3306); //PUERTO SER
16             ds.setDatabaseName("dbventastesis");//NOMBRE DE LA BASE DE DATOS
17             conection=ds.getConnection("root","");//USUARIO Y CONTRASEÑA DE LA CONEXION
18         }catch (Exception ex){
19             ex.printStackTrace();
20         }
21         return conection;
22     }
23 }
24 }
```

Figura 12 Cadena de conexión a la Base de Datos
Fuente: Captura propia

Para la creación de reportes en java se utilizará los ya muy conocidos software iReport.

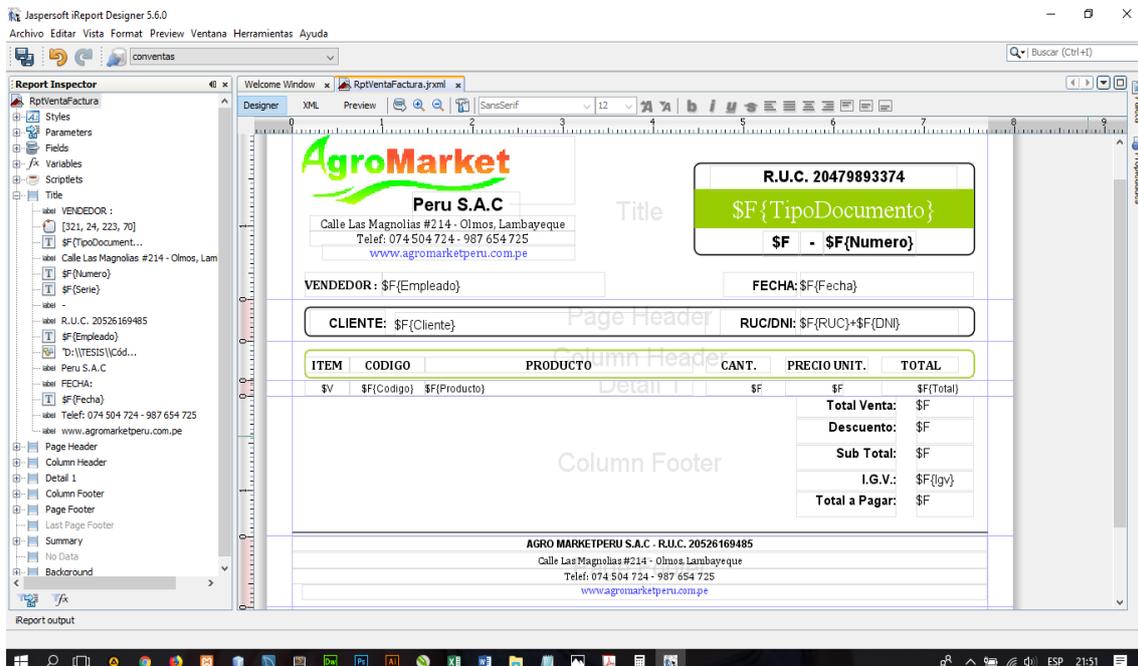


Figura 13 Entorno de iReport
Fuente: Captura propia

Con las configuraciones iniciales e instalación del software necesario, ya estamos listos para empezar a codificar nuestra aplicación.

5.2. PLANIFICACIÓN Y DISEÑO



*Figura 14 Interface principal del sistema
Fuente: Captura propia*

Según lo planteado anteriormente desarrollaremos el sistema de información haciendo uso de la metodología XP, es decir mediante historias de usuarios.

5.2.1 ITERACIÓN 1

5.2.1.1 HISTORIAS DE USUARIOS

La iteración consta de 03 historia de usuario:

- Gestión de Empleado
- Gestión de Proveedor
- Gestión de Cliente

Gestión de empleado	
Número Historia: 01	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de empleado	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 7 (1-10)	Iteración asignada: 1
Descripción: El usuario podrá administrar la creación de empleados. El formulario de Gestión de empleados nos permitirá agregar, modificar y deshabilitar un usuario específico.	
Observación: Para registrar un empleado previamente debemos registrar un tipo de usuario con sus roles o permisos asignados según las funciones que va a desempeñar.	

*Tabla 4 Gestión de Empleado
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 01	Nombre: Diseño de interfaz Gestión de Empleado
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 2
Descripción: Diseñaremos una interfaz donde se pueda registrar, modificar y deshabilitar un empleado que tendrá acceso al sistema.	

*Tabla 5 Diseño de interfaz gestión de empleado
Fuente: Elaboración propia*

Tarea N° 02	
Número de Historia: 01	Nombre: Mantenimiento de Gestión de empleado
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 7
Descripción: Registraremos, modificaremos o cambiaremos de estado un empleado específico, que buscaremos mediante los campos de registro Id, Nombre o Apellido, DNI. Una vez ingresado los datos necesarios o realizado los cambios en los campos deseados el sistema enviará una respuesta y la información se guardará en la base de datos.	

*Tabla 6 Mantenimiento de gestión de empleado
Fuente: Elaboración propia*

Gestión de proveedor	
Número Historia: 02	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de proveedor	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 7 (1-10)	Iteración asignada: 1
Descripción: El usuario podrá administrar en el sistema la creación de proveedores. El formulario de Gestión de proveedores nos permitirá agregar, modificar y deshabilitar un proveedor específico.	
Observación: Ninguna	

*Tabla 7 Gestión de proveedor
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 02	Nombre: Diseño de interfaz Gestión de proveedor
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 2
Descripción: Diseñaremos una interfaz donde se pueda registrar, modificar y deshabilitar un proveedor que tendrá acceso al sistema.	

*Tabla 8 Diseño de interfaz gestión de proveedor
Fuente: Elaboración propia*

Tarea N° 02	
Número de Historia: 02	Nombre: Mantenimiento de Gestión de proveedor
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 7
Descripción: Registraremos, modificaremos o cambiaremos de estado un proveedor específico, que buscaremos mediante los campos de registro Id, Nombre o razón social, DNI y ruc. Una vez ingresado los datos necesarios o realizado los cambios en los campos deseados el sistema enviará una respuesta y la información se guardará en la base de datos.	

*Tabla 9 Mantenimiento de gestión de proveedor
Fuente: Elaboración propia*

Gestión de Cliente	
Número Historia: 03	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de cliente	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 7 (1-10)	Iteración asignada: 1
Descripción: El usuario podrá administrar en el sistema la creación de clientes. El formulario de Gestión de clientes nos permitirá agregar, modificar y deshabilitar un proveedor específico.	
Observación: Ninguna	

*Tabla 10 Gestión de cliente
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 03	Nombre: Diseño de interfaz Gestión de cliente
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 2
Descripción: Diseñaremos una interfaz donde se pueda registrar, modificar y deshabilitar un cliente específico que tendrá acceso al sistema.	

*Tabla 11 Diseño de interfaz gestión de cliente
Fuente: Elaboración propia*

Tarea N° 02	
Número de Historia: 03	Nombre: Mantenimiento de Gestión de cliente
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 7
Descripción: Registraremos, modificaremos un cliente específico, que buscaremos mediante los campos de registro Id, Nombre o razón social, DNI y RUC. Una vez ingresado los datos necesarios o realizado los cambios en los campos deseados el sistema enviará una respuesta y la información se guardará en la base de datos.	

*Tabla 12 Mantenimiento de gestión de cliente
Fuente: Elaboración propia*

Tarjeta CRC Empleado	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar empleado • Modificar empleado • Buscar empleado • Validar Campos • Validar Campos obligatorios 	ClsEntidadEmpleado ClsEmpleado

*Tabla 13 Tarjeta CRC Empleado
Fuente: Diseño propio*

Tarjeta CRC Proveedor	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar proveedor • Modificar proveedor • Deshabilitar proveedor • Buscar proveedor • Validar Campos • Validar Campos obligatorios 	ClsEntidadProveedor ClsProveedor

*Tabla 14 Tarjeta CRC proveedor
Fuente: Elaboración propia*

Tarjeta CRC Cliente	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar cliente • Modificar tipo usuario • Eliminar tipo usuario • Buscar tipo usuario • Validar Campos • Validar Campos obligatorios 	ClsEntidadCliente ClsCliente

Tabla 15 Tarjeta CRC cliente
Fuente: Elaboración propia

5.2.1.2 DISEÑO DE LA ITERACIÓN

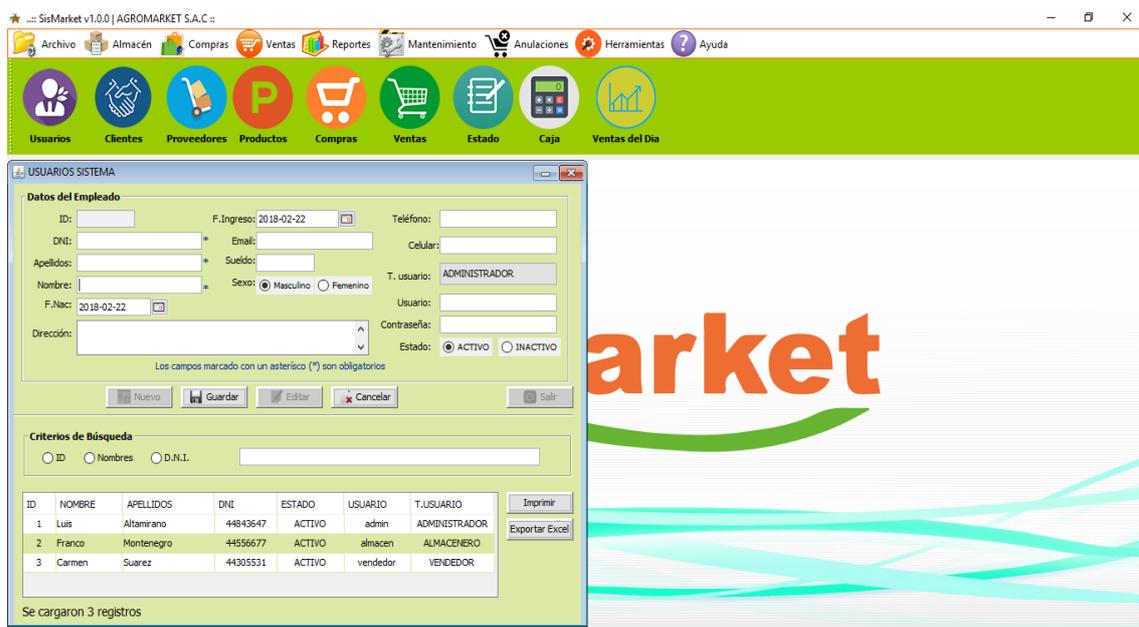


Figura 15 Registro de empleados
Fuente: Elaboración propia

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,”



Figura 16 Registro de proveedor
Fuente: Elaboración propia

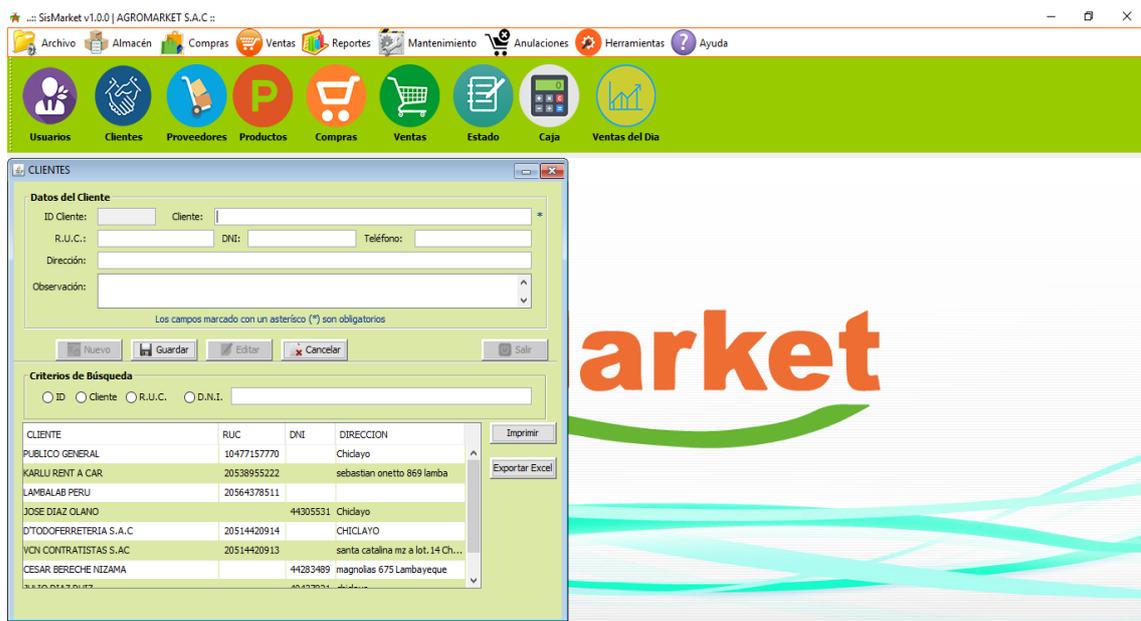


Figura 17 Registro de cliente
Fuente: Elaboración propia

5.2.2 ITERACIÓN 2

5.2.2.1 HISTORIAS DE USUARIOS

La iteración consta de 02 historia de usuario:

- Gestión de tipos de Usuarios
- Gestión de privilegios de tipo de usuario

Gestión de tipos de Usuarios	
Número Historia: 04	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de tipos de Usuarios	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 9 (1-10)	Iteración asignada: 2
Descripción: El usuario podrá administrar la creación de tipos de usuarios. El formulario de Gestión de usuarios permitirá agregar, modificar y deshabilitar un usuario específico.	
Observación: El usuario estará activo siempre y cuando esté laborando en la empresa, de caso contrario estará en estado Deshabilitado.	

Tabla 16 Gestión de tipos de Usuarios

Fuente: Elaboración propia

Tarea N° 01	
Número Historia: 04	Nombre: Diseño de interfaz Gestión de tipos de Usuarios
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 2
Descripción: Diseñaremos una interfaz donde se pueda gestionar el registro de usuarios con acceso al sistema y sus respectivos privilegios de acceso.	

Tabla 17 Diseño de interfaz gestión de usuarios y privilegios

Fuente: Elaboración propia

Tarea N° 02	
Número de Historia: 04	Nombre: Mantenimiento de Gestión de tipos de Usuarios
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 7
Descripción: Registraremos, modificaremos o cambiaremos de estado un usuario específico, que buscaremos mediante los campos de registro Nombre o Apellido, DNI, Id Empleado. Una vez ingresado los datos necesarios o realizado los cambios en los campos deseados el sistema enviará una respuesta y la información se guardará en la base de datos.	

*Tabla 18 Mantenimiento de gestión de tipos de usuarios
Fuente: Elaboración propia*

Gestión privilegios de tipo de usuario	
Número Historia: 05	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de privilegios de tipo de usuario	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 7 (1-10)	Iteración asignada: 2
Descripción: El usuario podrá administrar la creación de los privilegios según el tipo establecido para ciertas funciones. El formulario de Gestión de privilegios permitirá agregar, modificar los privilegios o funciones deseadas.	
Observación: Ninguna	

*Tabla 19 Gestión de privilegios de tipo de usuario
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 05	Nombre: Diseño de interfaz Gestión de Privilegios de tipo de usuario
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 2
Descripción: Diseñaremos una interfaz donde se pueda gestionar la asignación de los permisos y/o funciones para cada tipo de usuario con acceso al sistema.	

*Tabla 20 Diseño de interfaz gestión de privilegios de usuario
Fuente: Elaboración propia*

Tarea N° 02	
Número de Historia: 05	Nombre: Mantenimiento de Gestión de privilegios de tipo de usuario
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 7
Descripción: Asignaremos, modificaremos los privilegios a un tipo de usuario específico, que buscaremos mediante los campos de registro Id tipo de usuario o descripción. Una vez ingresado los datos necesarios o realizado los cambios con los roles deseados el sistema enviará una respuesta y la información se guardará automáticamente en la base de datos.	

*Tabla 21 Mantenimiento de gestión de privilegios de Tipo de usuario
Fuente: Elaboración propia*

Tarjeta CRC Tipo Usuario	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar tipo usuario • Modificar tipo usuario • Eliminar tipo usuario • Buscar tipo usuario • Validar Campos • Validar Campos obligatorios 	ClsEntidadTipoUsuario ClsTipoUsuario

Tabla 22 Tarjeta CRC Tipo Usuario
Fuente: Elaboración propia

5.2.2.2 DISEÑO DE LA ITERACIÓN

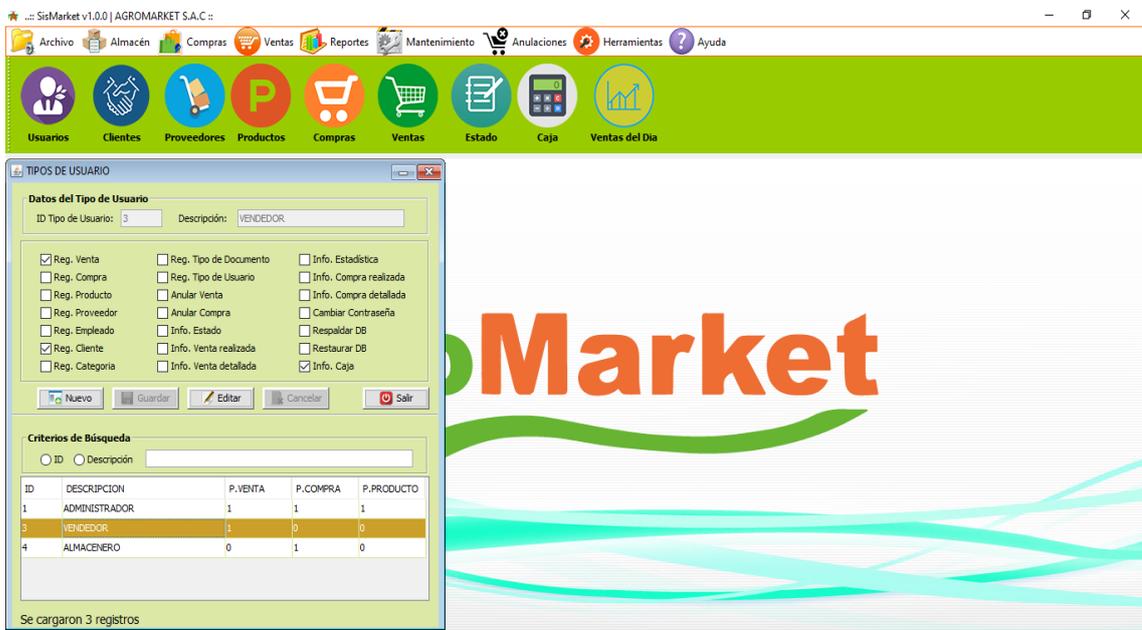


Figura 18 Creación de roles según tipo de Usuario
Fuente: Captura propia

**DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C.,"**

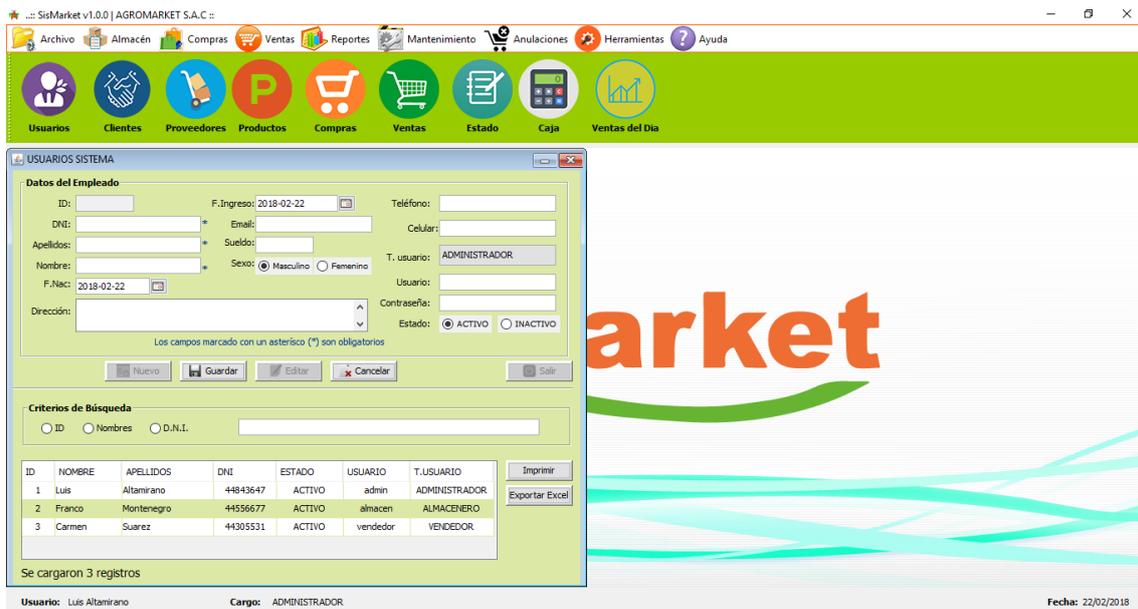


Figura 19 Registro de Tipo de Usuario
Fuente: Captura propia

5.2.3 ITERACIÓN 3

5.2.3.1 HISTORIAS DE USUARIOS

Esta iteración consta de 02 historia de Usuario

- Gestión de Categoría de productos
- Gestión de productos

Gestión de Categoría de productos	
Número de Historia: 06	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de Categoría de productos	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 7 (1-10)	Iteración asignada: 3
Descripción: El usuario podrá gestionar el registro de las diferentes categorías de los productos a registrar en el sistema, para poder así clasificar por categoría los productos comercializados.	
Observación: Ninguna	

Tabla 23 Gestión de Categorías de productos
Fuente: Elaboración propia

Tarea N° 01	
Número Historia: 06	Nombre: Diseño de interfaz Gestión de Categorías de productos
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 2
Descripción: Diseñaremos una interfaz donde se pueda gestionar el registro, modificación y eliminación de las categorías de productos.	

*Tabla 24 Diseño de interfaz gestión de categoría de productos
Fuente: Elaboración propia*

Tarea N° 02	
Número Historia: 06	Nombre: Mantenimiento de Categorías de productos
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 7
Descripción: Pulsaremos en la pestaña “Nuevo/Modificar” desde donde podremos agregar una nueva categoría pulsando el botón “Nuevo” o si la acción es modificar buscaremos por “id” o “descripción” la categoría deseada y pulsaremos botón “Modificar”, rellenaremos el campo descripción y luego pulsaremos el botón “Modificar” y el sistema guardará automáticamente la información en la base de datos.	

*Tabla 25 Mantenimiento de Categorías de productos
Fuente: Elaboración propia*

Tarjeta CRC Categoría	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar categoría • Modificar categoría • Eliminar categoría • Buscar categoría • Validar Campos • Validar Campos obligatorios 	<p>ClsEntidadCategoria ClsCategoria</p>

*Tabla 26 Tarjeta CRC Categoría
Fuente: Elaboración propia*

Gestión de productos	
Número de Historia: 03	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de productos	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 8 (1-10)	Iteración asignada: 3
Descripción: El usuario podrá gestionar el registro o consultas de los diferentes productos en el sistema, para poder así clasificar por categoría los productos comercializados.	
Observación: El producto siempre se guarda con estado inicial activo	

*Tabla 27 Gestión de productos
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 06	Nombre: Diseño de interfaz Gestión de productos
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 3
Descripción: Diseñaremos una interfaz donde se pueda gestionar el registro, modificación y eliminación de los productos.	

*Tabla 28 Diseño de interfaz Gestión de productos
Fuente: Elaboración propia*

Tarea N° 02	
Número Historia: 06	Nombre: Mantenimiento Gestión de productos
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 8
<p>Descripción: Pulsaremos en la pestaña “Nuevo/Modificar” desde donde podremos agregar un nuevo producto pulsando el botón “Nuevo”, rellenaremos los campos necesarios o si la acción es modificar buscaremos por “Código de producto”, “nombre”, “descripción” o “categoría” el producto deseado y pulsaremos botón “Modificar”, rellenaremos los campos obligatorios. Luego pulsaremos el botón “Modificar” y el sistema guardará automáticamente la información en la base de datos.</p>	

*Tabla 29 Diseño Interfaz Gestión de productos
Fuente: Elaboración propia*

Tarjeta CRC Producto	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar producto • Modificar producto • Eliminar producto • Buscar producto • Validar Campos • Validar Campos obligatorios 	<p>ClsEntidadProducto ClsProducto</p>

*Tabla 30 Tarjeta CRC Producto
Fuente: Elaboración propia*

5.2.3.2 DISEÑO DE LA ITERACION



Figura 20 Registrar/Modificar Categorías
Fuente: Captura Propia



Figura 21 Registrar Categoría
Fuente: Captura Propia

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C.,"

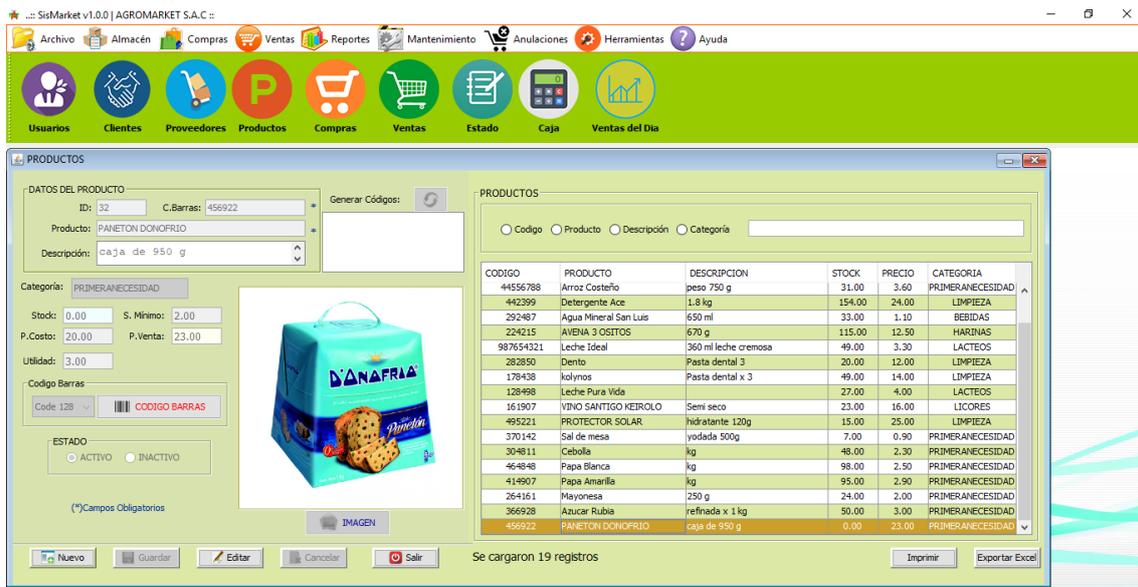


Figura 22 Registrar/Modificar Producto
Fuente: Captura Propia

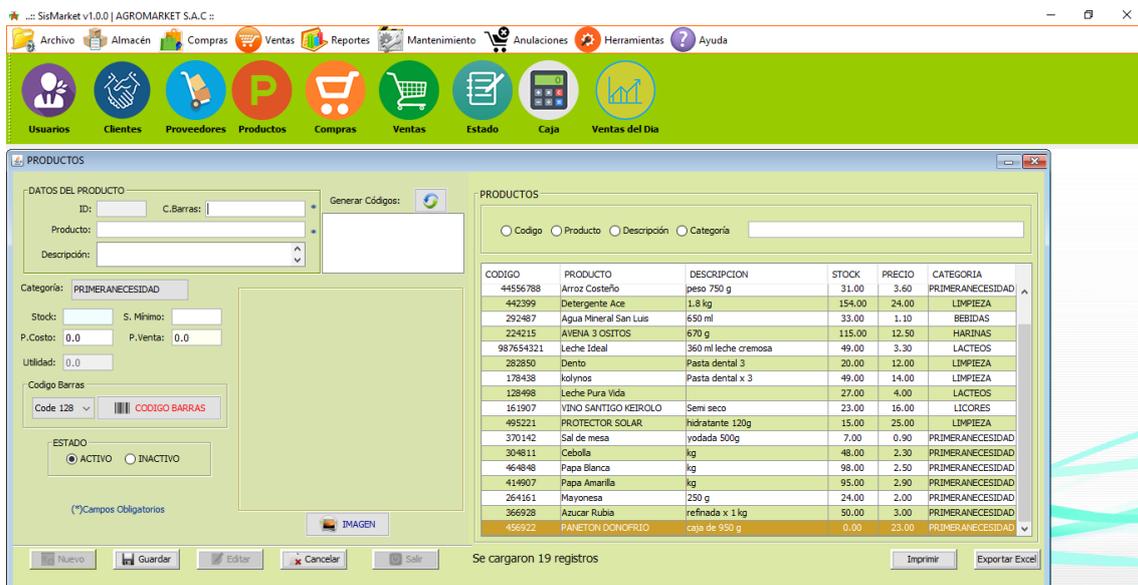


Figura 23 Registrar producto
Fuente: Captura Propia

5.2.4 ITERACIÓN 4

5.2.4.1 HISTORIAS DE USUARIOS

Esta iteración consta de 02 historia de Usuario

- Gestión de Compras
- Gestión Anular compras

Gestión de Compras	
Número de Historia: 07	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de compras	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 8 (1-10)	Iteración asignada: 4
Descripción: El usuario podrá gestionar el registro de compras de los productos necesarios, donde podremos seleccionar el tipo de comprobante recibido por la compra.	
Observación: Para realizar la compra se debe registrar previamente un producto con sus respectivos precios y también un proveedor.	

*Tabla 31 Gestión de compras
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 07	Nombre: Diseño de interfaz Gestión de Compras
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 3
Descripción: Diseñaremos una interfaz donde se pueda gestionar el registro de las compras de los productos.	

*Tabla 32 Diseño de interfaz Gestión de Compras
Fuente: Elaboración propia*

Tarea N° 02	
Número Historia: 07	Nombre: Mantenimiento Gestión de compras
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 8
Descripción: Pulsaremos en el botón “Nuevo” desde donde podremos agregar una nueva compra de los productos deseados, seleccionaremos los campos proveedor y producto con una búsqueda desde el botón “lupa” luego ingresamos los demás datos (tipo de comprobante, fecha, cantidad) de la compra y pulsaremos el botón “Guardar” y el sistema guardará automáticamente la información en la base de datos.	

Tabla 33 Mantenimiento Gestión de Compras

Fuente: Elaboración propia

Gestión Anular Compra	
Número de Historia: 08	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de Anular compras	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 8 (1-10)	Iteración asignada: 3
Descripción: El usuario podrá gestionar la anulación de las compras realizadas.	
Observación: Ninguna	

Tabla 34 Gestión Anular compras

Fuente: Elaboración propia

Tarea N° 01	
Número Historia: 08	Nombre: Diseño de interfaz Gestión de Anular Compras
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 3
Descripción: Diseñaremos una interfaz donde se pueda gestionar la anulación de las compras registradas en el sistema.	

*Tabla 35 Diseño de interfaz Gestión de Anular compras
Fuente: Elaboración propia*

Tarea N° 02	
Número Historia: 08	Nombre: Mantenimiento Gestión Anular compras
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 8
Descripción: Pulsaremos “Anular compra” del menú “Anulaciones”, realizaremos la búsqueda por rango de fechas y tipo de documento del proveedor con una búsqueda desde el botón “Buscar” ubicaremos la compra específica y su detalle pulsando el botón “ver detalle”, luego pulsaremos el botón “Anular” para anular dicha compra y el sistema guardará automáticamente la información en la base de datos.	

*Tabla 36 Mantenimiento Gestión Anular compras
Fuente: Elaboración propia*

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,”

Tarjeta CRC Compras	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar compra • Modificar compra • Anular compra • Buscar compra • Validar Campos • Validar Campos obligatorios 	ClsEntidadCompra ClsCompra

Tabla 37 Tarjeta CRC Compras
Fuente: Elaboración propia

5.2.4.2 DISEÑO DE LA ITERACION



Figura 24 Registrar Compra
Fuente: Captura Propia

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,”



Figura 25 Anular Compra
Fuente: Captura Propia

5.2.5 ITERACIÓN 5

5.2.5.1 HISTORIAS DE USUARIOS

Esta iteración consta de 02 historia de Usuario

- Gestión de Ventas
- Gestión de Anular de venta

Gestión de Ventas	
Número de Historia: 09	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de ventas	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 10 (1-10)	Iteración asignada: 5
Descripción: El usuario podrá gestionar el registro de ventas de los diferentes productos, donde podremos seleccionar el tipo de comprobante solicitado por el cliente al momento de realizar la venta.	
Observación: Para realizar una venta se debe registrar previamente una compra de productos con sus respectivos precios y también un cliente.	

Tabla 38 Gestión de Ventas
Fuente: Elaboración propia

Tarea N° 01	
Número Historia: 09	Nombre: Diseño de interfaz Gestión de Ventas
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 5
Descripción: Diseñaremos una interfaz donde se pueda gestionar las ventas de los productos.	

*Tabla 39 Diseño de interfaz de Gestión de Ventas
Fuente: Elaboración propia*

Tarea N° 02	
Número Historia: 09	Nombre: Mantenimiento Gestión de ventas
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 9
Descripción: Para realizar una venta presionares el botón “Nuevo”, donde se nos habilitarán los campos para rellenar en una venta, buscaremos el cliente con el botón “Lupa”, seleccionaremos los productos con el botón “Lupa”, seleccionaremos el tipo de documento desde la lista desplegable (Boleta, Factura, Ticket), registraremos la fecha e ingresaremos la cantidad deseada del producto seleccionado, luego debemos presionar el botón generar venta y el sistema guardará automáticamente la información en la base de datos y podremos imprimir el comprobante de la venta.	

*Tabla 40 Mantenimiento Gestión de ventas
Fuente: Elaboración propia*

Gestión de Anular de Ventas	
Número de Historia: 10	Usuario: Administrador del Sistema
Nombre historia de usuario: Gestión de ventas	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Puntos estimados: 9 (1-10)	Iteración asignada: 5
Descripción: El usuario podrá gestionar la anulación de las ventas realizadas en el sistema.	
Observación: Ninguna	

*Tabla 41 Gestión de Anular de ventas
Fuente: Elaboración propia*

Tarea N° 01	
Número Historia: 10	Nombre: Diseño de interfaz Gestión de Anular Ventas
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 3
Descripción: Diseñaremos una interfaz donde se pueda gestionar la anulación de las ventas registradas en el sistema.	

*Tabla 42 Diseño de interfaz Gestión Anular Ventas
Fuente: Elaboración propia*

Tarea N° 02	
Número Historia: 10	Nombre: Mantenimiento Gestión Anular Ventas
Programador Responsable: Jorge Luis Altamirano Fernández	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/ Mejora/Otra)	Puntos estimados: 8
Descripción: Pulsaremos “Anular venta” del menú “Anulaciones”, realizaremos la búsqueda por rango de fechas y tipo de documento del cliente con una búsqueda desde el botón “Buscar” ubicaremos la compra específica y su detalle pulsando el botón “ver detalle”, luego pulsaremos el botón “Anular” para anular dicha venta y el sistema guardará automáticamente la información en la base de datos.	

*Tabla 43 Mantenimiento Gestión Anular Ventas
Fuente: Elaboración propia*

Tarjeta CRC Ventas	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Registrar venta • Modificar venta • Anular venta • Buscar venta • Validar Campos • Validar Campos obligatorios 	<p>ClsEntidadVenta ClsVenta</p>

*Tabla 44 Tarjeta CRC Ventas
Fuente: Elaboración propia*

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C.,"

5.2.5.2 DISEÑO DE LA ITERACION



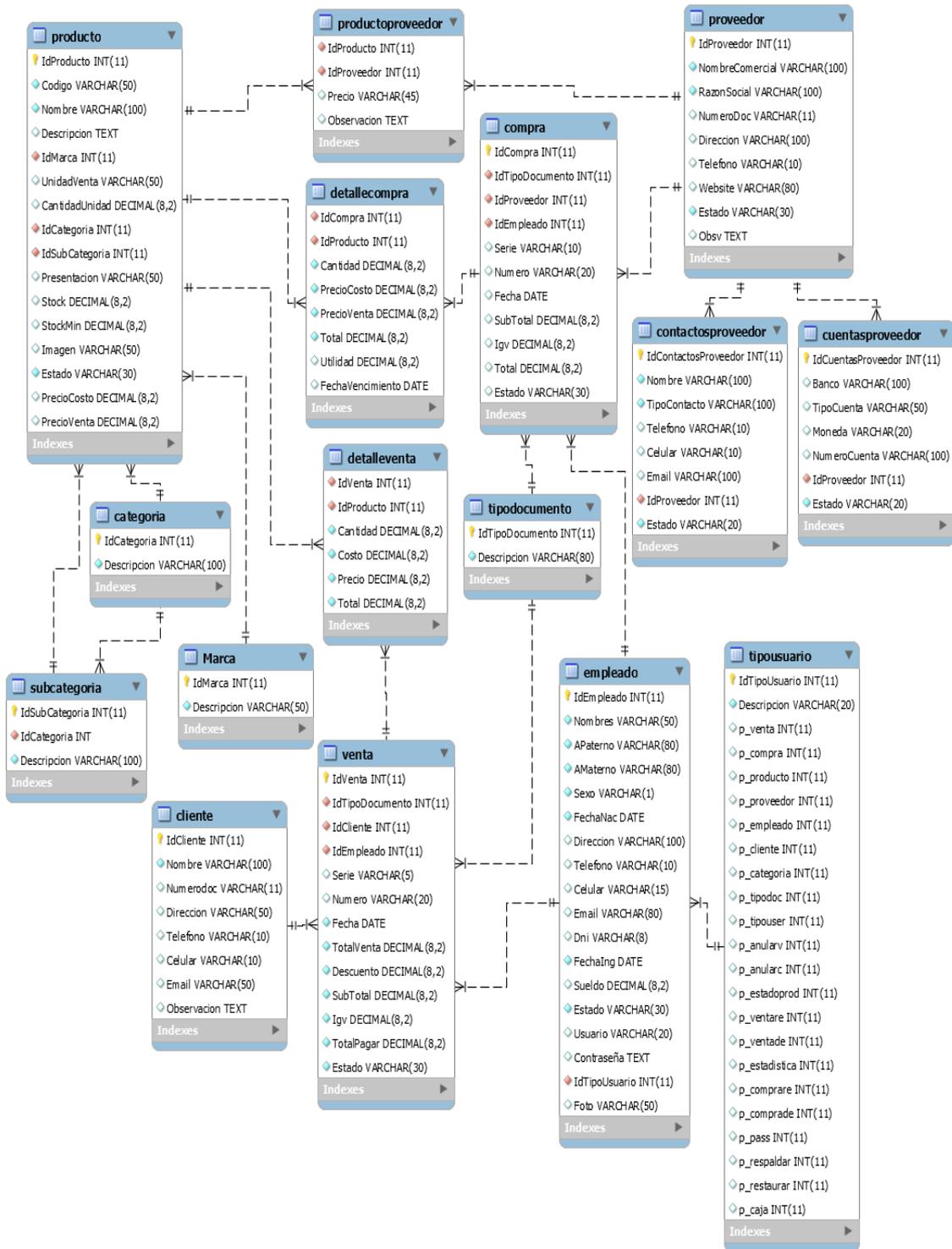
Figura 26 Realizar venta
Fuente: Captura Propia



Figura 27 Anular venta
Fuente: Captura Propia

**DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C.,**

5.2.5 BASE DE DATOS



*Figura 28 Base de Datos
Fuente: Captura Propia*

5.3.- IMPLEMENTACIÓN DE FUNCIONALIDADES

Se detallará el implemento de algunas funcionalidades del sistema para demostrar cómo se ha realizado el trabajo de integración de las diferentes herramientas de desarrollo de software para que esta aplicación sea lo potente y robusta posible.

5.3.1 REALIZAR UNA VENTA

CÓDIGO	PRODUCTO	DESCRIPCIÓN	CANT.	PRECIO	TOTAL
123456789	Leche Gloria	Leches Azul	12.0	3.5	42.0
345567899	Coca Cola	bebida de 1.5 Litros	12.0	4.5	54.0
161907	VINO SANTIGO KEIROLO	Semi seco	12.0	16.0	192.0
292487	Agua Mineral San Luis	650 ml	12.0	1.1	13.2

PAGO CON	VUELTO	VALOR VENTA	DESCUENTO	SUB TOTAL	I.G.V.	TOTAL A PAGAR
0.0	0.0	301.2	0.0	255.25	45.95	301.2

Figura 29 Formulario de Ventas
Fuente: Captura Propia

5.3.1.1 CODIFICACION

BUSCAR CLIENTE:

Hacemos la petición usando la clase Cliente con el método listarClientePorParametro con los criterios de búsqueda definidos en el formulario FrmBuscarCliente Venta y el texto digitado como búsqueda que enviaremos al procedimiento almacenado SP S_clientePorParametro que hemos llamado con la sentencia call desde nuestra base de datos, donde le enviaremos los parámetros para comparar y así devolver el dato solicitado.

```
public ResultSet listarClientePorParametro(String criterio,  
    String busqueda) throws Exception{  
    ResultSet rs = null;  
    try{  
        CallableStatement statement =  
            connection.prepareCall("{call SP_S_ClientePorParametro(?,?)}");  
        statement.setString("pcriterio", criterio);  
        statement.setString("pbusqueda", busqueda);  
        rs = statement.executeQuery();  
        return rs;  
    }catch(SQLException SQLex){  
        throw SQLex;  
    }  
}
```

Figura 30 Método listarClientePorParametro
Fuente: Captura Propia

PROCEDIMIENTO ALMACENADO

SP_S_clientePorParametro

Recibimos los parámetros y realizamos la comparación, dependiendo del criterio que se haya seleccionado en el formulario FrmBuscarCliente Venta (Id, Nombre, RUC, DNI) se realizará la comparación y mostrará los datos del cliente para poder seleccionar para realizar la venta.

**DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,"**

```

385 <----->
386 DELIMITER //
387 CREATE PROCEDURE SP_S_ClientePorParametro
388 (
389     pcriterio varchar(20),
390     pbusqueda varchar(20)
391 )
392 BEGIN
393 IF pcriterio = "id" THEN
394     SELECT c.IdCliente,c.Nombre,c.Ruc,c.Dni,c.Direccion,c.Telefono,c.Obsv FROM cliente AS c WHERE c.IdCliente=pbusqueda;
395 ELSEIF pcriterio = "nombre" THEN
396     SELECT c.IdCliente,c.Nombre,c.Ruc,c.Dni,c.Direccion,c.Telefono,c.Obsv FROM cliente AS c WHERE c.Nombre LIKE CONCAT("%",pbusqueda,"%");
397 ELSEIF pcriterio = "ruc" THEN
398     SELECT c.IdCliente,c.Nombre,c.Ruc,c.Dni,c.Direccion,c.Telefono,c.Obsv FROM cliente AS c WHERE c.Ruc LIKE CONCAT("%",pbusqueda,"%");
399 ELSEIF pcriterio = "dni" THEN
400     SELECT c.IdCliente,c.Nombre,c.Ruc,c.Dni,c.Direccion,c.Telefono,c.Obsv FROM cliente AS c WHERE c.Dni LIKE CONCAT("%",pbusqueda,"%");
401 ELSE
402     SELECT c.IdCliente,c.Nombre,c.Ruc,c.Dni,c.Direccion,c.Telefono,c.Obsv FROM cliente AS c;
403 END IF;
404 END;//
405 DELIMITER &&
406

```

*Figura 31 Procedimiento almacenado SP_S_clientePorParametro
Fuente: Captura Propia*

ID	Nombre o Razón Social	RUC	DNI	Teléfono
2	KARLU RENT A CAR	20538955222		

Se cargaron 1 registros

*Figura 32 FrmBuscarCliente_Venta
Fuente: Captura Propia*

BUSCAR PRODUCTO:

Para buscar y/o seleccionar un producto son los mismos pasos y se utilizan los procedimientos, métodos y pasos que usamos en Buscar cliente.

SELECCIONAR TIPO DOCUMENTO (BOLETA, FACTURA, TICKET...) DESDE COMBOBOX:

Para cargar y luego seleccionar el tipo de documento o tipo de comprobante previamente registrado, usaremos la clase ClsTipoDocumento donde declaramos un ArrayList de la clase ClsEntidadTipoDocumento donde están declarados los metodos get y set, recorreremos el Arraylits haciendo uso del patrón de diseño iterator y el bucle while como mostramos en la Figura 30.

```
void cargarComboTipoDocumento() {
    ClsTipoDocumento tipodocumento=new ClsTipoDocumento();
    ArrayList<ClsEntidadTipoDocumento> tipodocumentos=tipodocumento.listarTipoDocumento();
    Iterator iterator=tipodocumentos.iterator();
    DefaultComboBoxModel DefaultComboBoxModel=new DefaultComboBoxModel();
    DefaultComboBoxModel.removeAllElements();

    cboTipoDocumento.removeAll();
    String fila[]=new String[2];
    intContador=0;

    while(iterator.hasNext()){
        ClsEntidadTipoDocumento TipoDocumento = new ClsEntidadTipoDocumento();
        TipoDocumento=(ClsEntidadTipoDocumento) iterator.next();
        id[intContador]=TipoDocumento.getStrIdTipoDocumento();
        fila[0]=TipoDocumento.getStrIdTipoDocumento();
        fila[1]=TipoDocumento.getStrDescripcionTipoDocumento();
        DefaultComboBoxModel.addElement(TipoDocumento.getStrDescripcionTipoDocumento());
        intContador++;
    }
    cboTipoDocumento.setModel(DefaultComboBoxModel);
}
```

Figura 33 Seleccionar Tipo Documento (Boleta, Factura, Ticket...) Desde Combobox.

Fuente: Captura Propia

Una vez rellenado los campos anteriores presionamos el botón btnAgregarProducto con ícono, “+” para agregar a la lista de productos al JTable, tblDetalleProducto que el cliente va a comprar. Luego

podremos ingresar el monto de dinero con el que el cliente va a cancelar con el botón "Importe" para que el sistema nos calcule el vuelto de este, generamos la venta con el botón "Generar Venta" y el sistema enviará la información a la base de datos para ser agregados.

```
private void btnAgregarProductoActionPerformed(java.awt.event.ActionEvent e)
double stock,cant;

if (!txtCantidadProducto.getText().equals("")) {
    if(txtCantidadProducto.getText().equals("")){
        txtCantidadProducto.setText("0");
        cant=Double.parseDouble(txtCantidadProducto.getText());
    }else{
        cant=Double.parseDouble(txtCantidadProducto.getText());
    }

    if(cant>0){
        stock=Double.parseDouble(txtStockProducto.getText());
        cant=Double.parseDouble(txtCantidadProducto.getText());
        if(cant<=stock){
            int d1=Integer.parseInt(lblIdProducto.getText());
            String d2=txtCodigoProducto.getText();
            String d3=txtNombreProducto.getText();
            String d4=txtDescripcionProducto.getText();
            double d5=Double.parseDouble(txtCantidadProducto.getText());
            double d6=Double.parseDouble(txtCostoProducto.getText());
            double d7=Double.parseDouble(txtPrecioProducto.getText());
            double d8=Double.parseDouble(txtTotalProducto.getText());
            agregardatos(d1,d2,d3,d4,d5,d6,d7,d8);

            CalcularTotal_Pagar();
            CalcularSubTotal();
            CalcularIGV();
            txtCantidadProducto.setText("");
            txtTotalProducto.setText("");

            txtCodigoProducto.setText("");
            txtNombreProducto.setText("");
            txtStockProducto.setText("");
            txtPrecioProducto.setText("");
            txtCodigoProducto.requestFocus();
        }else{
            JOptionPane.showMessageDialog(null, "Stock Insuficiente");
            txtCantidadProducto.requestFocus();
        }
    }else{
        JOptionPane.showMessageDialog(null, "Ingrese Cantidad mayor a 0");
        txtCantidadProducto.requestFocus();
    }
}else{
    JOptionPane.showMessageDialog(null, "Ingrese cantidad");
    txtCantidadProducto.requestFocus();
}
}
```

Figura 34 btnAgregarProducto
Fuente: Captura Propia

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO MARKET PERU S.A.C,”



Figura 35 Generar Venta
Fuente: Captura Propia

Confirmamos la operación para generar la venta y luego presionando el botón “Recibo” podremos elegir si visualizamos el recibo y lo exportamos a formato PDF o lo imprimimos.



Figura 36 Visualizar o imprimir comprobante de venta
Fuente: Captura Propia

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,”

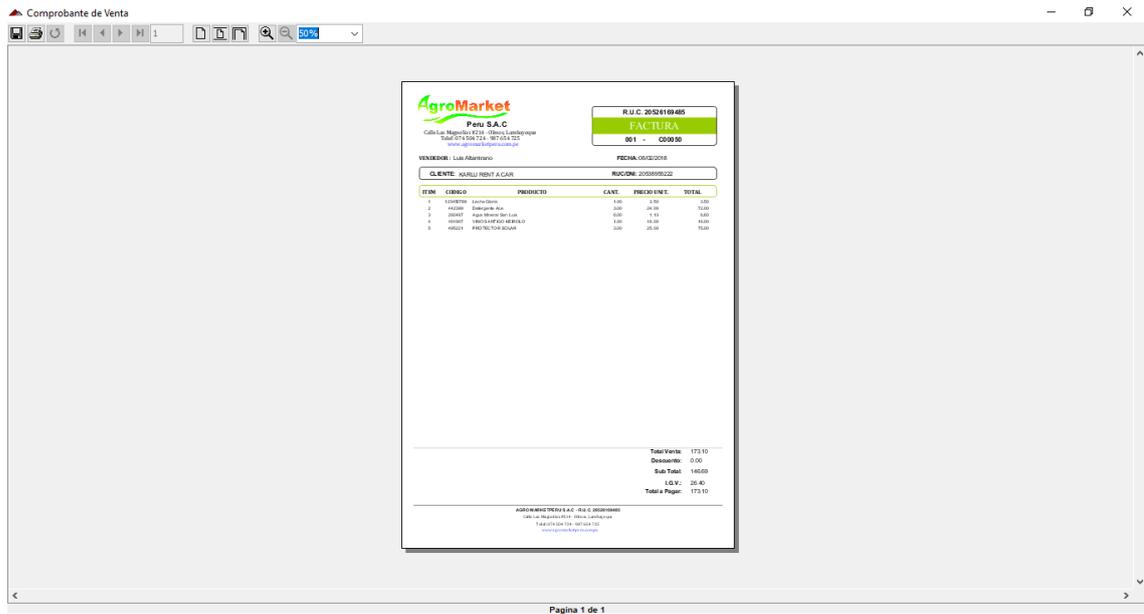


Figura 37 Comprobante de venta
Fuente: Captura Propia

5.4 SEGURIDAD EN EL SISTEMA

Para asegurar un correcto funcionamiento y evitar posibles accesos no permitidos o intrusos al sistema se desarrollará un acceso con clave y contraseña, mediante tipo de usuarios.

- Para la autenticación de usuarios se emplea un panel numérico aleatorio.
- El usuario y contraseña están asociados al Empleado.
- Los privilegios están determinados de acuerdo al tipo de usuario (Administrador, Vendedor, Almacenero...) donde se le dará diversos permisos según el tipo que se registró.

Los usuarios no registrados por un administrador del sistema no podrán acceder al sistema.

DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,”

```
void generarAleatorio(){
    int n=10;int k=n;
    int[] numeros=new int[n];
    int[] resultado=new int[n];
    Random rnd=new Random();
    int res;

    for(int i=0;i<n;i++){
        numeros[i]=numeros[i]+i;
    }

    for(int i=0;i<n;i++){
        res=rnd.nextInt(k);
        resultado[i]=numeros[res];
        numeros[res]=numeros[k-1];
        k--;
    }

    btnD1.setText(String.valueOf(resultado[0]));
    btnD2.setText(String.valueOf(resultado[1]));
    btnD3.setText(String.valueOf(resultado[2]));
    btnD4.setText(String.valueOf(resultado[3]));
    btnD5.setText(String.valueOf(resultado[4]));
    btnD6.setText(String.valueOf(resultado[5]));
    btnD7.setText(String.valueOf(resultado[6]));
    btnD8.setText(String.valueOf(resultado[7]));

    btnD9.setText(String.valueOf(resultado[8]));
    btnD0.setText(String.valueOf(resultado[9]));
}
```

Figura 38 Número aleatorios para panel de contraseña de acceso
Fuente: Captura Propia



Figura 39 Vista principal Sistema
Fuente: Captura Propia

AgroMarket

ACCESO AL SISTEMA SISMARKET V1.0

Tipo Usuario: ADMINISTRADOR

Usuario:

Constraseña:

Teclado:

6	8	3	0
9	5	4	1
2	7	Limpiar	

INICIAR SALIR

Figura 40 Login para acceso al sistema
Fuente: Captura Propia

AgroMarket

Acceso Usuarios

Bienvenido: Sr(a).Luis Altamirano

Aceptar

Teclado:

7	9	1	3
5	8	0	6
2	4	Limpiar	

INICIAR SALIR

Figura 41 Acceso al sistema
Fuente: Captura Propia

**DESARROLLO DE UN SISTEMA DE INFORMACION HACIENDO USO DE LA METODOLOGIA
XP PARA LA GESTIÓN DE VENTAS, COMPRAS Y ALMACEN DE LA EMPRESA AGRO
MARKET PERU S.A.C,”**

Las funciones del sistema y los permisos dependerán del tipo de usuario que inicie sesión, ya que en el menú habrá funciones que estén habilitadas o deshabilitadas según los permisos que se haya definido en el registro de tipo de usuario



*Figura 42 Interfaz de almacén
Fuente: Captura Propia*



*Figura 43 Tipos de usuarios registrados
Fuente: Captura Propia*

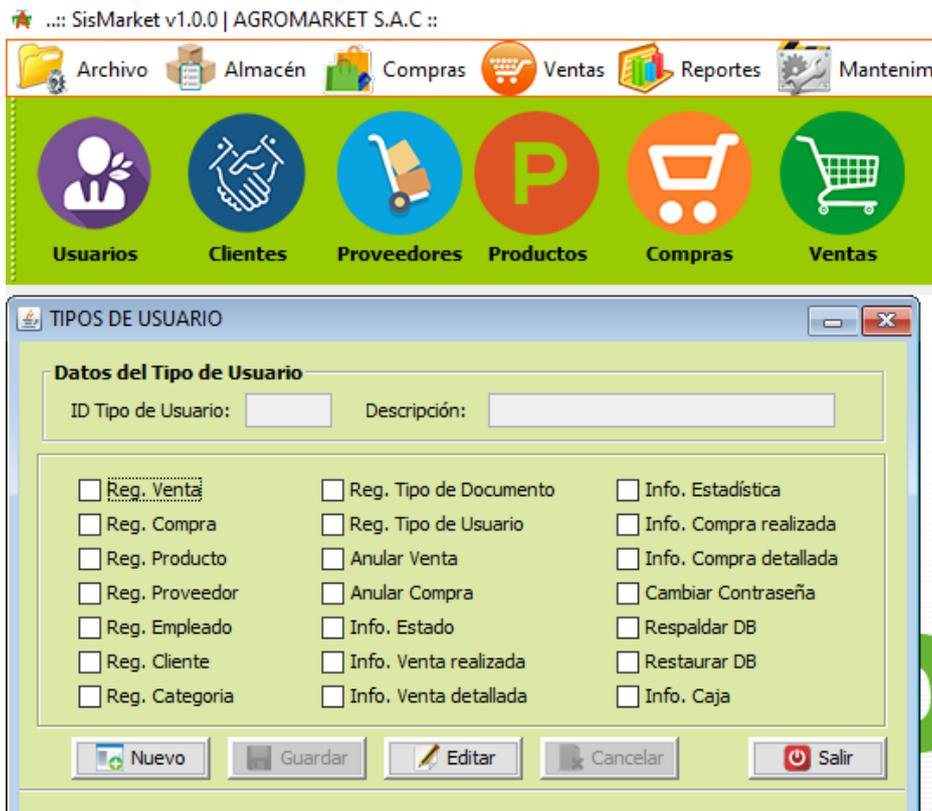


Figura 44 Tipos de funciones asignadas por tipo de usuario
Fuente: Captura Propia

5.5 PRUEBAS DE ACEPTACIÓN DEL USUARIO FINAL

Para realizar Las pruebas de aceptación del sistema de información por parte de usuario final, se instaló en una máquina permitiendo al Gerente general de la empresa Agro Market Perú S.A.C, el Ing, Yolmer Montenegro Sánchez, realizar las pruebas necesarias al sistema para comprobar las interfaces, funcionalidades, reportes que este permite realizar, conocer y controlar sobre el giro de la empresa.

El gerente pudo imprimir el reporte de existencias de mercadería, reporte de los productos más vendidos, los menos vendidos, reporte de clientes, reporte de ventas valoradas.

Luego se hizo la prueba con un vendedor, se simuló un proceso de venta y devolución de un producto. Se registró en el sistema y luego se procedió a hacer la devolución del producto. El sistema registró la salida del producto de almacén y luego el ingreso cuando se procedió a devolverlo.

También se hizo la prueba con el encargado de almacén, quien registró una compra como ingreso a las existencias, Luego simulamos una alerta por stock mínimo de un producto para que se inicie el proceso de adquisición del mismo.

Después de dicho manejo del sistema, el gerente general dio su aprobación y quedó conforme con el desempeño del sistema, dando su conformidad, ver anexo 01.

CAPITULO VI

COSTO Y BENEFICIOS

VI: COSTOS Y BENEFICIOS

6.1.-ANALISIS DE COSTOS Y BENEFICIOS

6.1.1. COSTO DE SOFTWARE

SOFTWARE DE DESARROLLO	SUB TOTAL (S/.)
Netbeans 8.0.2	0.00
MySQL Workbench	0.00
Xampp (phpMyAdmin)	0.00
CA Erwin Data Modeler Community Edition	0.00
TOTAL (S/.)	0.00

Tabla 45 Software de desarrollo

Fuente: Diseño Propio

SISTEMA OPERATIVO	SUB TOTAL (S/.)
Windows 10 Pro	980.00
TOTAL (S/.)	980.00

Tabla 46 Sistema operativo

Fuente: Diseño Propio

6.1.2. COSTO DE PERSONAL

ITEM	SUB TOTAL (S/.)
Movilidad	240.00
Alimentos	1380.00
Analista	3000.00
Administrador Base de Datos	1400.00
Programador	2200.00
TOTAL (S/.)	8220.00

Tabla 47 Costo de personal

Fuente: Diseño Propio

6.1.3. COSTO DE SERVICIO Y MATERIALES

ITEM	CANTIDAD	SUB TOTAL (S/.)
CD	15 unidades	15.00
DVD	15 unidades	22.50
Papel Bond A4	03 millares	78.00
Lapiceros	05 unidades	7.50
Folders	24 unidades	12.00
Sobre Manila A4	12	6.00
	TOTAL (S/.)	141.00

Tabla 48 Costo de materiales
Fuente: Diseño Propio

ITEM	SUB TOTAL (S/.)
Movilidad	240.00
Telecomunicaciones	300.00
Fotocopias	90.00
Impresiones	90.00
Espiralados	15.00
Encuadernación	35.00
Energía eléctrica	360.00
Internet	774.00
TOTAL (S/.)	1904.00

Tabla 49 Costo de servicios
Fuente: Diseño Propio

6.1.4. COSTOS DE HARDWARE

ITEM	SUB TOTAL (S/.)
Laptop	3000.00
Mouse	80.00
TOTAL (S/.)	3080.00

Tabla 50 Costo de Hardware
Fuente: Diseño Propio

6.1.5. COSTOS DE IMPLEMENTACIÓN

ITEM	CANTIDAD	SUB TOTAL (S/.)
Pc core i3	02	1500.00
	TOTAL (S/.)	3 000.00

Tabla 51 Costos de implementación

Fuente: Diseño Propio

6.1.6. COSTO DE MANTENIMIENTO

ITEM	SUB TOTAL (S/.)
Movilidad	50.00
Alimentos	50.00
Administrador Sistema	500.00
TOTAL (S/.)	600.00

Tabla 52 Costo de mantenimiento

Fuente: Diseño Propio

6.1.7. RESUMEN DE COSTOS

ITEM	SUB TOTAL (S/.)
Costo de Software	980.00
Costo de Personal	8220.00
Costo de Servicio y Materiales	2045.00
Costos de Hardware	3580.00
Costos de Implementación	4000.00
Costo de Mantenimiento	600.00
Otros costos	0.00
TOTAL (S/.)	17 925.00

Tabla 53 Resumen de costos

Fuente: Diseño Propio

6.2 RECUPERACION DE LA INVERSION

La recuperación de la inversión en un proyecto de software es difícil de calcular porque las variables son de tipo intangible que este proporciona, pero se puede reflejar en el incremento de la productividad que impulsará el crecimiento de la empresa, mientras que el aumento en rentabilidad te permitirá percibir mayores ingresos y reducir costos.

Se evaluó el proyecto utilizando tres fórmulas financieras PR, VAN y TIR. Cuya interpretación y desarrollo es la siguiente:

INVERSIÓN INICIAL	S/17,925.00
-------------------	--------------------

Tabla 54 Monto Inversión Inicial

Fuente: Diseño Propio

CUADRO DE AHORRO CON EL USO DEL SOFTWARE	
GASTOS MENSUALES	S/
Sueldo del encargado de control de existencias	850.00
Sueldo del encargado de ventas	850.00
SUB TOTAL MENSUAL	1700.00
TOTAL AHORRO ANUAL	20400.00

Tabla 55 Cuadro de Ahorro con uso del Sistema

Fuente: Diseño Propio

6.2.1 EVALUACION DEL PROYECTO EN BASE AL PERIODO DE RECUPERACION

**PR = Periodo último con flujo acumulado negativo +
[valor absoluto del último flujo acumulado negativo/
Valor del flujo de caja en el siguiente periodo]**

$$PR = 0 + [17925/20400]$$

$$PR = 0.88 \text{ Años}$$

Según la evaluación en base al Periodo de recuperación, el proyecto es viable, ya que se recuperará la inversión en menos de un año (0.8 años).

6.2.2 EVALUACIÓN DEL PROYECTO EN BASE AL VALOR ACTUAL NETO

La evaluación del proyecto se está haciendo tomando un periodo de tres años.

Se está tomando una tasa de costo de oportunidad del 5% que es la tasa aceptada en el mercado local.

AÑO	AHORRO ANUAL	DIVIDIR POR EL FACTOR DE ACTUALIZACIÓN	VALOR ACTUAL
1	20400	$(1+0.05)^1$	19428.57
2	20400	$(1+0.05)^2$	18503.40
3	20400	$(1+0.05)^3$	17622.29
TOTAL	61200		55554.26

Tabla 56 Evaluación en base al VAN

Fuente: Diseño Propio

VAN = -Inversión Inicial + VALOR ACTUAL

VAN = -17925+55554.26

VAN = 37 629.26

El **VAN** es mayor a la inversión inicial, por lo que concluimos que es rentable el proyecto

6.2.3 EVALUACIÓN DEL PROYECTO EN BASE A LA TASA DE RETORNO DE LA INVERSIÓN

	MONTO
Inversión Inicial	-17925
Ahorro Año 1	20400
Ahorro Año 2	20400
Ahorro Año 3	20400
TIR	99%

Tabla 57 Evaluación en base al TIR

Fuente: Diseño Propio

Según la evaluación con el método del TIR, vamos a recuperar la inversión con una tasa del 99%, en un periodo de tres años. Por lo que concluimos que el proyecto es viable.

A continuación, se listan los beneficios que una correcta implementación de un sistema de información puede proporcionar al negocio en cuanto a productividad, costos, e ingresos:

Productividad

- Consolidación de información (búsqueda de información en un solo lugar).
- Procesos de negocio sistematizados, controlados, y más eficientes.
- Disminución de documentos físicos, autorizaciones en papel, y movimientos.
- Procesos y cálculos automatizados.

Disminución de costos

- Disminución de tiempos en consolidación y análisis de estados financieros.
- Control de inventarios con menos recursos humanos
- Disminución de tiempos y movimientos en procesos de negocio. (Procesos más ágiles y automatización de tareas).
- Mayor control y decremento de mermas y gastos.
- Eliminación de redundancias (tareas repetidas).

6.3.-BENEFICIOS

6.3.1. BENEFICIOS TANGIBLES

- Reducción de costo laborales
- Mejora en la productividad de los procesos de la empresa
- Reducción en material de apunte de inventarios y ventas
- Mejorar el acceso a la información
- Reducción de tiempo en revisar los informes diarios
- Reducción de los precios de los productos
- Aumento de las utilidades

6.3.2. BENEFICIOS INTANGIBLES

- Aumento de la transparencia en las transacciones diarias
- Rápido acceso a la información para toma de decisiones
- Ahorro de tiempo y esfuerzo en el registro de datos

CONCLUSIONES

CONCLUSIONES

1. La metodología XP nos permitió obtener documentos técnicos y de requerimientos, que nos facilitó la realización del análisis y una acción rápida si es que se presenten algunas modificaciones por parte del usuario, esta es una característica fundamental de la metodología utilizada en el desarrollo.
2. Al utilizar la metodología XP se demostró que es eficiente para ser manejada en equipos de trabajo pequeños o medianos. Eficiente porque facilita al desarrollador el realizar su aplicación sin detenerse en una extensa documentación y diagramación de los modelos como requieren otras metodologías. Así mismo se adapta de muy buena manera a proyectos de corto plazo.
3. La arquitectura del sistema se basó en el modelo de desarrollo de 03 capas (Presentación, Negocio y datos) y programación orientada a objetos, la cual nos permitió una descomposición de las partes que componen el sistema, para así tener un orden adecuado al momento de programar las funcionalidades.
4. La información que se recopiló de la empresa Agro Market nos permitió desarrollar las iteraciones con las historias de usuario, además de definir los requerimientos funcionales y no funcionales.
5. El uso de la encriptación SHA nos permitió darle mayor seguridad al acceso al sistema mediante encriptación de las contraseñas.
6. El uso de iReport nos facilitó la presentación de informes y reportes solicitados por el usuario.
7. Las pruebas de aceptación realizadas nos permiten asegurar el cumplimiento de los requerimientos del usuario.
8. En el desarrollo del proyecto, mediante el uso de software libre nos permitió reducir considerablemente los costos de inversión del proyecto.

RECOMENDACIONES

RECOMENDACIONES

1. Las reuniones con el usuario deben ser frecuentes para mostrarle los avances y proponer mejoras y los cambios en el momento oportuno.
2. Recopilar la información clara para diseñar la base de datos, ya que esta debe estar diseñada y estructurada correctamente, para evitar inconsistencias al momento del desarrollo del proyecto.
3. Los requerimientos deben ser recopilados de manera correcta, el usuario debe ser específico al momento de brindar la información para definir las funcionalidades del proyecto.
4. La configuración del servidor local de base de datos debe ser configurado correctamente al momento de implementar el sistema.
5. Realizar las pruebas finales de testeó con el personal usuario del sistema, para determinar el correcto funcionamiento de este y evitar posibles errores de compilación y ejecución.
6. Brindar capacitación al personal que usará el sistema para su correcto uso y funcionamiento del sistema.

CAPÍTULO VII

REFERENCIAS BIBLIOGRÁFICAS

VII: REFERENCIAS BIBLIOGRÁFICAS

BIBLIOGRAFIA

- ❖ Balarezo Paredes, B. (2012). Desarrollo de un sistema de información de registro de pedidos para ventas usando dispositivos móviles.
- ❖ Barnes, D. (2007). *Programación Orientada a Objetos con Java*.
- ❖ Bustamante, D., & Rodriguez, J. (2014). *Metodología Actual, Metodología XP*. Batrinas.
- ❖ Pezo Linares, J. (2012). Diseño e implementación de un sistema informático para mejorar el proceso de ventas en la tienda Señor de ayabaca de la ciudad de Tarapoto.
- ❖ Pressman, R. (2010). *Ingengeria del software. Un enfoque práctico* (Séptima Edición ed.). México: McGraw- Hill.
- ❖ Rodriguez, J. (2012). *Aprende Java como si estuvieras en Primero*.
- ❖ Rodriguez, M. (2005). Desarrollo de un sistema de información automatizado para la facturación y el control de ventas de la empresa 3M Computación C.A.
- ❖ Sánchez , M. (2011). Desarrollo de un Sistema Web basado en ADO.NET para el control de la gestión de inscripción y logística de cursos en la empresa Enfoque Directo aplicado a consultores, C.A.
- ❖ Schildt, H. (2009). *Java, Manual de Referencia* (séptima Edición ed.). Mexico: McGRAW-HILL INTERAMERICANA EDITORES, S.A.
- ❖ Deitel, P. (2012). *Como Programar en Java* (Octava Edición ed.).
- ❖ Elmasri, R., & Navathe, S. (2002). *Fundamentos de Sistemas de bases de datos* (Tercera Edición ed.). Madrid: Addison-Wesley.
- ❖ Florez Marin, L. (s.f.). FORMULACION DE CRITERIOS PARA LA SELECCION DE METODOLOGIAS . Pereira, Pereira, Risaralda.
- ❖ Gilfillan, I. (2003). *La Biblia de MySQL*. Madrid: Anaya Multimedia.
- ❖ Guzmán Silva, J. (2008). Diseño y Optimización del proceso de ejecución de la venta mayorista para una empresa tipo Home Improvement.
- ❖ Jeffries, R. (2000). *Extreme Programming Installed*. EEUU: EN.

- ❖ Kendall, K., & Kendall, J. (2011). *Análisis y Diseño de Sistemas* (Octava Edición ed.). Mexico: Pearson.
- ❖ Kniberg, H. (s.f.). Scrum and XP from the Trenches. EEUU, EEUU, Estados Unidos de América.
- ❖ Liber, B. (2006). Extreme Programming (XP). *Extreme Programming (XP)*, 2006.
- ❖ Llacchua, M. (2007). Diseño de un sistema de comercialización para el supermercado Minimarket Tito's.

LINKOGRAFIA

- ❖ J, J. (10 de 02 de 2008). *Reglas y prácticas en Extreme Programming*. Obtenido de <http://iie.fing.edu.uy/%7Ejosej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- ❖ Friends, A. (s.f.). *Apache Friends*. Obtenido de <https://www.apachefriends.org/es/index.html>
- ❖ Programming, E. (12 de noviembre de 2017). *extremeprogramming.org*. Obtenido de [extremeprogramming.org:](http://www.extremeprogramming.org/)
- ❖ Oracle. (2015). *MySQL, developer*. Obtenido de <http://ftp.nchu.edu.tw/MySQL/doc/refman/5.0/es/features.html>
- ❖ <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/238>
- ❖ Oracle. (2016). *Oracle - MySQL*. Obtenido de <https://www.mysql.com/products/workbench/>
- ❖ Oracle. (s.f.). *Java Platform SEE8*. Obtenido de Java Platform SEE8: <http://docs.oracle.com/javase/8/docs/api/>
- ❖ Oracle, N. (2016). *NetBeans*. Obtenido de <https://netbeans.org/features/>

ANEXOS

ACTA DE CONFORMIDAD

Conste por el presente documento que en la fecha se realizó las pruebas del proyecto:

“Desarrollo de un Sistema de Información haciendo uso de la Metodología XP para la gestión de ventas, compras y almacén de la empresa Agro Market Peru S.A.C”.

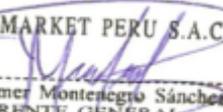
Desarrollado por el señor *Jorge Luis Altamirano Fernández*. Se deja constancia que es de entera satisfacción de la empresa, de acuerdo a las necesidades y objetivos trazados.

Quedó demostrado de que es un proyecto viable y rentable, usando la evaluación basada en la Tasa de retorno de inversión, Valor actual neto y Periodo de recuperación.

Habiéndose culminado el presente trabajo, se firma la presente en señal de conformidad.

Olmos, diciembre del 2017.

Atentamente.

AGRO MARKET PERU S.A.C.

Ing. Yolmer Montenegro Sánchez
GERENTE GENERAL

Anexo 1 Acta de conformidad
Fuente: Elaboración propia