



**UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS  
APLICADAS  
CARRERA DE INGENIERÍA EN SISTEMAS  
COMPUTACIONALES.**

**TEMA**

**ESTUDIO DE LA METODOLOGÍA MSF “MICROSOFT SOLUTION  
FRAMEWORK” APLICADA AL DESARROLLO DE UN SISTEMA DE  
FACTURACIÓN “SOFTWARE LIBRE” PARA LA UNIÓN DE  
PAPELERÍAS DE LA CIUDAD DE IBARRA.**

**APLICATIVO**

**SISTEMA DE FACTURACIÓN Y CONTROL DE INVENTARIOS PARA  
LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA**

**Autor:** Carlos Julio Landázuri Ortiz

**Director:** Econ. Winston Oviedo.


**Ibarra – Ecuador**

**2013**



## CERTIFICACIÓN

Certifico que la Tesis: **ESTUDIO DE LA METODOLOGÍA MSF “MICROSOFT SOLUTION FRAMEWORK” APLICADA AL DESARROLLO DE UN SISTEMA DE FACTURACIÓN “SOFTWARE LIBRE “PARA LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA.** Ha sido realizada en su totalidad por el señor: Carlos Julio Landázuri Ortiz portador de la cédula de identidad número: 0401593421.



.....

Econ. Winston Oviedo.

**Director de Tesis**

## CERTIFICACIÓN

Ibarra, 29 de abril de 2013

Señores

UNIVERSIDAD TÉCNICA DEL NORTE

Presente

De mis consideraciones.-

Siendo auspiciante del proyecto de tesis del Egresado CARLOS JULIO LANDÁZURI ORTIZ con CI: 0401593421 quien desarrolló su trabajo con el tema **ESTUDIO DE LA METODOLOGÍA MSF “MICROSOFT SOLUTION FRAMEWORK” APLICADA AL DESARROLLO DE UN SISTEMA DE FACTURACIÓN “SOFTWARE LIBRE “PARA LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA.** Certifico que el sistema de facturación y control de inventarios “Control Paper Store” ha sido instalado, probado y puesto en uso de acuerdo a los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado: CARLOS JULIO LANDÁZURI ORTIZ. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa/institución.

El egresado CARLOS JULIO LANDÁZURI ORTIZ puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,

Lic. Alicia Ortiz

PROPIETARIA PAPELANDIA.



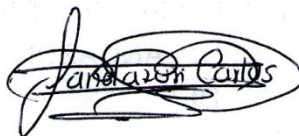
UNIVERSIDAD TÉCNICA DEL NORTE

**CESIÓN DE DERECHOS DE AUTOR  
DEL TRABAJO DE INVESTIGACIÓN A  
FAVOR DE LA UNIVERSIDAD TÉCNICA  
DEL NORTE**

Yo, CARLOS JULIO LANDÁZURI ORTIZ con cédula de identidad Nro. 0401593421, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“ESTUDIO DE LA METODOLOGÍA MSF “MICROSOFT SOLUTION FRAMEWORK” APLICADA AL DESARROLLO DE UN SISTEMA DE FACTURACIÓN “SOFTWARE LIBRE “PARA LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA”** que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.



.....  
Nombre: CARLOS JULIO LANDAZURI ORTIZ

Cédula: 0401593421

Ibarra a los 29 días del mes de abril del 2013.



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**BIBLIOTECA UNIVERSITARIA**  
**AUTORIZACIÓN DE USO Y PUBLICACIÓN**  
**A FAVOR DE LA UNIVERSIDAD**  
**TÉCNICA DEL NORTE**

## 1. IDENTIFICACIÓN DE LA OBRA

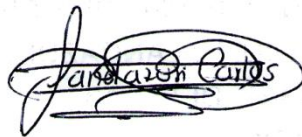
La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

<b>DATOS DE CONTACTO</b>	
<b>CEDULA DE IDENTIDAD</b>	0401593421
<b>APELLIDOS Y NOMBRES</b>	LANDÁZURI ORTIZ CARLOS JULIO
<b>DIRECCIÓN</b>	JUAN HERNADEZ 3-82 Y JAIME ROLDÓZ
<b>EMAIL</b>	<a href="mailto:cjlandazuri@hotmail.com">cjlandazuri@hotmail.com</a>
<b>TELÉFONO</b>	062643210
<b>MOVIL</b>	0993623258
<b>DATOS DE LA OBRA</b>	
<b>TITULO</b>	ESTUDIO DE LA METODOLOGÍA MSF "MICROSOFT SOLUTION FRAMEWORK" APLICADA AL DESARROLLO DE UN SISTEMA DE FACTURACIÓN "SOFTWARE LIBRE "PARA LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA
<b>AUTOR</b>	CARLOS JULIO LANDÁZURI ORTIZ
<b>FECHA</b>	29 DE ABRIL DEL 2013
<b>PROGRAMA</b>	PREGRADO
<b>TITULO POR EL QUE</b>	INGENIERÍA EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR</b>	ECON. WINSTON OVIEDO

## 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, CARLOS JULIO LANDÁZURI ORTIZ, con cédula de identidad Nro. 0401593421, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.

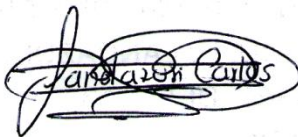


.....  
Nombre: CARLOS JULIO LANDAZURI ORTIZ  
Cédula: 0401593421 .....

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 29 días del mes de abril del 2013

A handwritten signature in black ink, appearing to read 'Landazuri Carlos', enclosed within a circular scribble.

.....  
Nombre: CARLOS JULIO LANDAZURI ORTIZ  
Cédula: 0401593421



## DEDICATORIA

A Dios por brindarme su guía y poner en mi camino las herramientas necesarias para alcanzar cada una de mis metas, por concederme salud, perseverancia y sobre todo por la fortaleza que fue el motor que me hace llegar a esta meta tan importante para mí.

A mis padres porque con su ejemplo de sabiduría y rectitud han formado mi carácter basado en la enorme admiración hacia ellos. Gracias por su confianza y apoyo incondicional.

A mi madre el pilar de mi vida, que con su ejemplo de trabajo y honradez formaron en mí un ser humano con valores y principios los mismos que pongo en práctica en cada paso de mi vida. Gracias por ser esa fuerza que me alienta en los momentos de debilidad, por ser la mano firme que me encaminó cuando erré y la primera en alegrarse por mis logros.

A mis hermanos Marcelo, Raúl, Raquel por su ejemplo de esfuerzo, trabajo y dedicación, a mis sobrinos Shalom, Marcel, Israel y Nathalia.

A mis amigos Marita, Gina, Roger, Jessy, Lupita, Edison, Alexandra y Nancy por tantos años de sincera amistad por alentarme a culminar mis metas y por extender su mano cuando necesite su ayuda. De Sobre manera a Marita por tu paciencia y ayuda desinteresada y por tantos momentos inolvidables de amistad.

## **AGRADECIMIENTO**

A la mi querida y gloriosa Universidad Técnica del Norte por permitirme formar parte de su alumnado, por formarme cada día de nuevos conocimientos y moldar profesionales con principios y valores.

Al Economista y director del Proyecto el Econ. Winston Oviedo, que con sus acertadas apreciaciones guió este proyecto.

A los docentes de la UTN en especial de la Facultad de Ingeniería en Sistemas por la enorme labor que realizan día con día al compartir sus conocimientos y experiencias con nosotros.

A mis amigos y compañeros de la Universidad por haber compartido conmigo momentos de lucha y constancia, gracias por tantos buenos recuerdos que jamás se van a olvidar.

A mi familia por su apoyo incondicional por su amor y por ser el apoyo que siempre necesito.

A todas las personas que me ayudaron de una u otra manera a culminar esta etapa de la vida.

## INTRODUCCIÓN

DECLARACIÓN.....	i
CERTIFICACIÓN.....	ii
CESIÓN DE DERECHOS.....	iii
IDENTIFICACIÓN DE LA OBRA .....	iv
AUTORIZACIÓN A FAVOR DE LA UNIVERSIDAD.....	v
CONSTANCIAS.....	vi
DEDICATORIA.....	vii
AGRADECIMIENTO.....	viii
TABLA DE CONTENIDOS.....	ix
RESUMEN .....	xv
SUMARY.....	xvi

## TABLA DE CONTENIDOS

<b>1</b>	<b>INTRODUCCION .....</b>	<b>1</b>
1.1	Tema:.....	1
1.2	Justificación .....	1
<b>2</b>	<b>MARCO TEÓRICO .....</b>	<b>3</b>
2.1	CAPÍTULO I: Procesos de desarrollo de Software.....	3
2.1.1	Conceptos generales.....	4
2.1.2	Modelos de procesos de software .....	5
2.1.3	Modelos del ciclo de procesos más populares .....	6
2.1.4	Modelo codificar y corregir .....	6
2.1.5	Modelo en cascada .....	7
2.1.6	Desarrollo Evolutivo.....	8
2.1.7	Desarrollo Formal de Sistemas .....	9
2.1.8	Desarrollo Incremental .....	10
2.1.9	Desarrollo en Espiral .....	12

2.1.10	Desarrollo Basado en Reutilización .....	13
2.1.11	El ciclo de vida y los procesos .....	14
2.1.12	Calidad en el Software .....	21
2.1.13	La calidad en ingeniería de software.....	22
2.1.14	La calidad desde el punto de vista de las ISO .....	23
2.1.15	Diferentes aspectos que intervienen en la medición de calidad.....	29
2.1.16	Metodología de desarrollo.....	31
2.1.17	Clasificación de las Metodologías .....	35
2.1.18	Metodologías Estructuradas.....	35
2.1.19	Metodologías de desarrollo Ágiles vs Tradicionales .....	39
2.1.20	Metodologías de desarrollo de software más populares .....	39
2.1.21	Metodología de desarrollo de software XP .....	40
2.1.22	Metodología de desarrollo de software SCRUM .....	46
2.1.23	Metodología de desarrollo Crystal Clear .....	48
2.1.24	Metodología de desarrollo ASD .....	51
2.1.25	Rational Unified Process (RUP).....	54
2.2	<i>CAPITULO II: SOFTWARE LIBRE</i> .....	58
2.2.1	Orígenes del Software Libre.....	58
2.2.2	Evolución del Software Libre.....	59
2.2.3	Características del Software Libre. ....	59
2.2.4	Ventajas del software Libre .....	60
2.2.5	Desventajas del Software Libre.....	61
2.2.6	El software propietario .....	61
2.2.7	Análisis comparativo Software libre vs Software propietario.....	63
2.2.8	Copyright, copyleft y patentes.....	64
2.2.9	Tipos de licencias.....	66
2.2.10	Comparativa de las diferentes licencias del software libre.....	70
2.2.11	Aplicación práctica de una licencia libre.....	71
2.2.12	Software Libre en la Actualidad de América Latina.....	72
2.2.13	Software libre en la actualidad de Estados Unidos .....	74
2.2.14	Software Libre en Ecuador.....	76
2.2.15	Seguridades de Software Libre .....	78
2.2.16	Seguridad en el software libre vs el software propietario .....	79
2.2.17	Modelo de negocio de proyectos de Software Libre .....	80
<b>3</b>	<b>METODOLOGÍA.....</b>	<b>82</b>
3.1	Selección de la Metodología Microsoft Solution Framework.....	82
3.1.1	MSF Orígenes.....	82
3.1.2	Versiones MSF.....	83
3.1.3	Principios Fundamentales.....	88
3.1.4	Las disciplinas de MSF .....	90
3.1.5	MSF Modelos de procesos y aplicaciones .....	91
3.1.6	Desarrollo ágil de aplicaciones.....	95

3.1.7	Casos de éxito.....	96
3.1.8	MSF y el Software Libre .....	97
3.1.9	MSF y Microsoft Operations Framework.....	100
3.1.10	Modelo Orientado a roles.....	102
3.1.11	Fases de la metodología MSF. ....	106
<b>4</b>	<b>APLICACIÓN DE LA METODOLOGÍA MSF 3.0.....</b>	<b>119</b>
4.1	<i>Estrategia visión y alcance</i> .....	119
4.1.1	Documentos a entregar en la fase de visión y alcance .....	119
4.1.2	Visión de “Control Paper Store” .....	134
4.1.3	Alcance de Control Paper Store.....	140
4.1.4	Planificación .....	151
4.1.5	Desarrollo de Control Paper Store.....	188
4.1.6	Estabilización de Control Paper Store.....	197
4.1.7	Implementación de Paper Control Store .....	202
<b>5</b>	<b>CONCLUSIONES .....</b>	<b>204</b>
<b>6</b>	<b>RECOMENDACIONES.....</b>	<b>206</b>
<b>7</b>	<b>BIBLIOGRAFÍA: .....</b>	<b>207</b>
<b>8</b>	<b>ANEXOS .....</b>	<b>211</b>
	<b>MANUAL DE INSTALACIÓN “CONTROL PAPER STORE”.....</b>	<b>212</b>
	<b>MANUAL DE USUARIO “CONTROL PAPER STORE”.....</b>	<b>220</b>

## ÍNDICE DE FIGURAS

Figura 1:	Diagrama genérico del desarrollo Evolutivo Incremental .....	11
Figura 2:	Gráfico Modelo en Espiral.....	13
Figura 3	Ciclo de Vida de los Procesos. ....	15
Figura 4	Metodologías orientadas a procesos. ....	36
Figura 5	Historia y versiones de MSF.....	84
Figura 6	Representación gráfica del modelo en cascada. ....	92
Figura 7	Representación gráfica del modelo en espiral. ....	93
Figura 8	Modelo propuesto pos MSF.....	93
Figura 9	Ciclos de Microsoft Operation Framework. ....	101
Figura 10	: Requerimientos y solución de Arquitectura.Net. ....	109
Figura 11	Gráfico de especificación de roles en la fase de visión.....	121

Figura 12: Caso de Uso Ingreso al Sistema. ....	153
Figura 13: Caso de Uso Facturación. ....	156
Figura 14: Caso de Devolución de Mercadería. ....	158
Figura 15 Caso de Uso Administración de productos.....	161
Figura 16 Caso de Uso Kardex. ....	162
Figura 17: Diagrama de la Arquitectura Multicapa.....	167
Figura 18 Diagrama de la Arquitectura MVC. ....	168
Figura 19 Diseño de la base de datos AppFacturaciónInventarios.....	178
Figura 20: Modelo de factura.....	190
Figura 21 Pantalla de control de acceso.....	191
Figura 22 Pantalla del menú principal.....	191
Figura 23 Pantalla de “Empresa” sistema Control Paper Store.....	192
Figura 24 Pantalla “Usuarios” sistema Control Paper Store.....	192
Figura 25 Pantalla “Categorías” Sistema Control Paper Store.....	192
Figura 26 Pantalla “Productos” Sistema Control Paper Store.....	193
Figura 27: Pantalla “Proveedores” Sistema Control Paper Store.....	193
Figura 28: Pantalla “Anulación Facturas” Sistema Control Paper Store.....	193
Figura 29: Pantalla “Clientes” Sistema Control Paper Store.....	194
Figura 30: Pantalla “Facturación” Sistema Control Paper Store.....	194
Figura 31: Pantalla “Reportes” Sistema Control Paper Store.....	195

## ÍNDICE DE TABLAS

Tabla 1 Ventajas y desventajas del Modelo Codificar y Corregir.....	7
Tabla 2: Cuadro Sinóptico Ventajas y Desventajas del Modelo Cascada. ....	8
Tabla 3 Mapa Conceptual Ventajas Y Desventajas Modelo Espiral. ....	12
Tabla 4 Planificación de la Gestión del Proyecto.....	15
Tabla 5: Identificación de la Necesidad en los Ciclos de Vida.....	16
Tabla 6 Proceso de diseño en los ciclos de vida de los procesos.....	17
Tabla 7 Proceso de implementación en los ciclos de vida de los procesos.....	18
Tabla 8 Proceso de instalación en los ciclos de vida de los procesos.....	18
Tabla 9 Los procesos de mantenimiento y retiro en los ciclos de vida. ....	19
Tabla 10 Procesos de validación en los ciclos de vida de los procesos. ....	19
Tabla 11 Procesos de configuración en los ciclos de vida de los procesos.....	20
Tabla 12 Los procesos de documentación y de formación en los procesos.....	21
Tabla 13 Términos comunes utilizados en la calidad de software.....	23
Tabla 14 Estándares a evaluar según la Norma ISO 9003.....	25
Tabla 15 Estándares a evaluar según la Norma ISO 9003.....	26
Tabla 16 Parámetros más relevantes de la norma ISO 9126.....	27
Tabla 17 Procesos del ciclo de vida del estándar 12207.....	28
Tabla 18 Clasificación de las metodologías de desarrollo de software. ....	35
Tabla 19 Tabla comparativa de metodologías ágiles vs tradicionales.....	39

Tabla 20 Iteración de artefactos en ASD. ....	53
Tabla 21 Artefactos utilizados en RUP. ....	57
Tabla 22 Ventajas y desventajas software propietario. ....	62
Tabla 23 Análisis comparativo Software libre vs Software propietario. ....	63
Tabla 24 Cuadro explicativo de las libertades del software libre. ....	64
Tabla 25 Estructura MSF V4.0. ....	88
Tabla 26 Modelo MSF con los ciclos y las iteraciones. ....	94
Tabla 27 Fases de MSF y la documentación por entregar. ....	95
Tabla 28 Tareas principales del Administrador del proyecto en MSF. ....	103
Tabla 29 Tareas principales del Arquitecto de software en MSF. ....	104
Tabla 30 Tareas principales del Jefe del proyecto en MSF. ....	104
Tabla 31 Tareas principales del Desarrollador. ....	105
Tabla 32 Tareas principales del Tester en MSF. ....	106
Tabla 33 Tareas en la fase de Visión en MSF. ....	108
Tabla 34 Entregables en la fase de Visión en MSF. ....	108
Tabla 35 : Fases de los procesos de planeación en MSF. ....	110
Tabla 36: Fases en el diseño conceptual de la etapa de planeación. ....	111
Tabla 37 Tareas de la fase de planificación de MSF. ....	115
Tabla 38 Documentación a entregar en la fase de estabilización. ....	117
Tabla 39 Documentación a entregar en la etapa de implementación. ....	118
Tabla 40 Documentación a entregar en la Fase de visión y alcance. ....	119
Tabla 41 Construcción del equipo de desarrollo. ....	123
Tabla 42 Tabla descriptiva de los responsables del proyecto. ....	123
Tabla 43 Descripción de riesgos del proyecto. ....	129
Tabla 44 Análisis de riesgos del proyecto. ....	130
Tabla 45 Rango de probabilidades con su porcentaje. ....	130
Tabla 46 Impacto de riesgos con su valoración. ....	131
Tabla 47 Tabla de exposición de riesgo del proyecto por valor. ....	131
Tabla 48 Descripción módulo de administración. ....	140
Tabla 49 Descripción Módulo de Inventarios. ....	141
Tabla 50 Descripción Módulo de Facturación. ....	142
Tabla 51 Descripción General del sistema. ....	143
Tabla 52 Caso de uso de ingreso al sistema. ....	152
Tabla 53 Caso de uso Categorías de productos. ....	154
Tabla 54 Caso de Uso Facturación. ....	155
Tabla 55 Caso de uso Devolución de Mercadería. ....	157
Tabla 56 Administración de Proveedores. ....	159
Tabla 57 Caso de uso Administración de productos. ....	160
Tabla 58 Caso de Uso Kardex. ....	162
Tabla 59: Administración de Clientes. ....	163
Tabla 60 Caso de uso de Reportes. ....	164
Tabla 61 Diagrama de la arquitectura MVC. ....	169
Tabla 62 : Definición de términos referentes a MSF. ....	175
Tabla 63 Estructura de la tabla categoría. ....	179

Tabla 64 Estructura de la tabla cliente.....	179
Tabla 65 estructura de la tabla detalle factura.....	180
Tabla 66 Estructura de la tabla Maestro factura. ....	181
Tabla 67 Estructura de la tabla módulos. ....	181
Tabla 68 Estructura de la tabla Kardex.....	182
Tabla 69 Estructura de la tabla Nota de crédito.....	182
Tabla 70 Estructura para la tabla proveedor.....	183
Tabla 71 Estructura de la tabla Permisos. ....	183
Tabla 72 Estructura de la tabla Productos.....	184
Tabla 73 Estructura de la Tabla Proveedor. ....	185
Tabla 74 Estructura de la tabla Transacción. ....	185
Tabla 75 Estructura de la tabla Usuarios.....	186
Tabla 76 Pruebas unitarias por módulo.....	200



## RESUMEN

La Universidad Técnica del Norte pensando en ayudar a la colectividad Ibarreña promueve la creación de proyectos orientados a las necesidades de su entorno. Es así como el desarrollo de un sistema de facturación y control de inventarios cubrirá los requerimientos de al menos 10 locales de papelerías que conforman la Unión de Papelerías de la ciudad de Ibarra los cuales contarán con el sistema de Software Libre "Control Paper Store". El mismo que es desarrollado bajo el lineamiento de la metodología de desarrollo Microsoft Solution Framework cumpliendo estándares y lineamientos de calidad de Software.

El diseño y desarrollo de software exige el cumplimiento de parámetros especificados en los requerimientos, pero cumpliendo con los procesos de una metodología probada y garantizada solo así se puede tener un producto de calidad que satisfaga las necesidades del cliente y cumpla con parámetros de calidad de software.

Para el diseño de un sistema de facturación se estudia el objetivo principal que es llevar el control de la mercadería existente en un negocio de papelería y la actividad de facturar cumpliendo con las exigencias del Servicio de Rentas Internas del Ecuador SRI.

La finalidad del sistema es que los clientes que realizan una adquisición cuenten con un documento dónde se registre su compra y los propietarios de los negocios tengan pleno control de su mercadería así como de las ventas.

En el desarrollo del sistema se eligió a MSF como metodología ya que adapta tanto el modelo cascada y el modelo espiral, se junta las ventajas de estos dos modelos para tener una metodología totalmente práctica y personalizada para entregar soluciones tecnológicas con menos personas, menos riesgos y con resultados de calidad y de impacto comercial.

Los objetivos de MSF son:

- ✓ Alinear objetivos empresariales y tecnológicos.
- ✓ Trazar correctamente los roles, responsabilidades y objetivos del proyecto.
- ✓ Establecer puntos iterativos, estableciendo puntos de control.
- ✓ Establecimiento oportuno de los riesgos.
- ✓ Respuestas oportunas y efectivas a cambios no esperados.

## SUMMARY

The North Technical University in an effort to help community, promotes the creation of projects oriented to the needs of your environment. Is how the development of an invoicing system and inventory control, covers the requirements of at least ten stationery stores that forming the Union of Stationers of Ibarra city, which will have the free software system called "Control Paper Store".

The same that is prepared under the guidelines of the development methodology Microsoft Solution Framework, complying software quality standards.

The design and development of software requires compliance of specified parameters, the requirements but complying with the processes of a proven and guaranteed only way you can have a quality product that satisfies customer needs and comply with software quality parameters.

For the design of an invoicing system studies the main objective, which is that of take control of the existing merchandise in business stationery and invoicing activity complying with the requirements of the Internal Revenue Service of Ecuador SRI:

The purpose of the system is that the customers who make an acquisition have a document where they register their purchase and business owners have control over their products and sales.

In the development of the system was chosen as a methodology MSF since adapts both the cascade model and the spiral model, joins the advantages of these two models for have a completely practical and customizable methodology for delivering technology solutions with less people, fewer risks, with quality results and business impact.

MSF's objectives are:

- ✓ Align business and technology objectives.
- ✓ Rightly divide the roles, responsibilities and objectives of the project.
- ✓ Establish interactive points, creating checkpoints.
- ✓ Timely establishment of the risks.
- ✓ Timely and effective responses to unexpected changes.

# **1 INTRODUCCION**

## **1.1 TEMA:**

ESTUDIO DE LA METODOLOGÍA MSF “MICROSOFT SOLUTIONFRAMEWORK” APLICADA AL DESARROLLO DE UN SISTEMA DE FACTURACIÓN “SOFTWARE LIBRE” PARA LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA.

## **1.2 JUSTIFICACIÓN**

La Universidad Técnica del Norte en su afán de promover proyectos que ayuden al desarrollo del Cantón. Después del análisis, en el que se enfatizó la utilización de programas informáticos que sistematicen las actividades comerciales se pensó en el diseño y desarrollo de un sistema de facturación que cubra los requerimientos de los locales de Unión de Papelerías de la ciudad de Ibarra específicamente será implantado en la papelería “PAPELANDIA” y luego de su prueba de funcionamiento estará a disposición de cualquiera de los miembros de la Unión de Papelerías de la ciudad de Ibarra.

El sistema será desarrollado con herramientas de software libre PHP y MySQL y tendrá licencia GPL que permite su libre uso, utilización y modificación del código fuente lo que ayuda a los propietarios ya que no pagan licencias por las herramientas.

El sistema será capaz de llevar el control de facturación del local comercial, se registrará en la base de datos la mercadería existente y se ingresarán las compras realizadas, es decir total control de los inventarios. Después de registrar una venta en la base de datos se imprimirá la factura autorizada por el SRI, la misma que será entregada al cliente. El proyecto se lo desarrollará de tal manera que la utilización para el usuario sea fácil y tendrá una interfaz amigable, además se podrá realizar consultas como ventas diarias, mensuales, anuales, reportes de un producto específico en el inventario y un reporte de cuáles son los productos próximo agotarse para que los propietarios puedan abastecerse.

## **OBJETIVOS:**

### ***Objetivo General***

Aplicar la metodología para el desarrollo de software “Microsoft Solution Framework”, en el desarrollo de un sistema de “Software Libre” que lleva el control de facturación e Inventarios para Unión de Papelerías de la ciudad de Ibarra.

### ***Objetivos Específicos***

1. Estudiar los procesos notables en el desarrollo de software, sus modelos y aspectos relevantes a tomar en cuenta en una metodología.
2. Realizar un análisis sobre “Software Libre” y las ventajas de su implementación, así como un estudio detallado sobre licencias GNU y validez.
3. Analizar la metodología para desarrollo de software “Microsoft Solution Framework” examinando su modelo y fases de desarrollo.
4. Aplicar la metodología MSF en el desarrollo del sistema de Facturación e inventarios, entregar e instalar el producto finalizado en su totalidad en la papelería “Papelandia” que forma parte de la Unión de papelerías de la ciudad de Ibarra. Realizando el original escrito del proyecto final recopilando la documentación de todas las fases de la metodología.
5. Analizar si tanto el software libre como software propietario actualmente pueden ir de la mano en un mismo proyecto.

## **2 MARCO TEÓRICO**

### **2.1 CAPÍTULO I: Procesos de desarrollo de Software**

#### **Introducción**

Software es un conjunto de secuencias lógicas ordenadas que son indispensables para la realización de procesos específicos, se lo conoce como el equipamiento lógico de un sistema informático a diferencia con los componentes físicos que se los define como hardware.

#### **Historia de los procesos de software**

El desarrollo de software estructurado tuvo sus inicios hacia los años de 1900 como contraparte a los métodos muy deliberados o estrictos en su realización, es decir no se tomaba en cuenta procesos adecuados lo que originaba que no se cumplan los objetivos del sistema ni las expectativas del cliente. Es así que en la década de 1960 el desarrollar a gran escala sistemas de negocios estaba sujeto a una época de grandes cambios tecnológicos. Se pensó en la idea de continuar con el desarrollo de los sistemas informáticos pero alrededor de etapas o ciclos de vida ya que hasta entonces se utilizaba solo el proceso originado del uso del modelo cascada el mismo que era visto muy lento e inconsistente y traía muchas veces resultados no esperados por eso la necesidad de nuevas formas de desarrollo que arrojaran trabajos eficientes.<sup>1</sup>

---

<sup>1</sup> Wikipedia (2009), Software Conceptos y definiciones. En: <http://es.wikipedia.org/wiki/Software>

## 2.1.1 Conceptos generales

### ➤ **Tarea**

Son funciones que cumplen una tarea específica como un algoritmo, arrojando un solo resultado y no un conjunto de acciones al mismo tiempo.

### ➤ **Proceso o Procedimiento**

Las tareas en los procesos de Software son los pasos o procedimientos ordenados que permiten al desarrollador o programador crear programas informáticos, usando algunas alternativas conocidas como lenguajes de programación los que incluyen: editores de texto, compiladores, intérpretes, enlazadores, depuradores entre otros.

### ➤ **Metodología**

Con la necesidad de mejoras en los procesos para el desarrollo de software nacen las llamadas metodologías de desarrollo de software que no son más que un framework o una base usada para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. El Framework para metodología consta de:

- Una base o enfoque de procesos de software previamente analizados y aceptados según su propia filosofía.
- Modelos, métodos y tareas para guiar el proceso de software.

### ➤ **Técnica**

Planear una aplicación como una operación global para descomponerla en operaciones más sencillas, detalladas y específicas.<sup>2</sup>

### ➤ **Herramienta**

Denominadas también CASE son ayudas o soportes a una tarea específica dentro de las actividades del desarrollo de software.

### ➤ **Producto**

Un producto de software es cualquier trabajo de desarrollo que se puede ofertar en un mercado tecnológico, cliente o negocio para satisfacer una necesidad. El producto de software es parte de la mezcla de marketing de la empresa, junto al precio, distribución, promoción y capacitación.

## **2.1.2 Modelos de procesos de software**

Los procesos utilizados en el desarrollo de software implican la utilización de estándares al momento de realizar una aplicación, que va desde el momento que surge la necesidad hasta el momento que se entrega el producto final. Ningún ciclo de software propone un modelo concreto, ni la forma de cómo realizar las diferentes actividades de cada proceso.

### **Definición**

Los modelos de procesos de software son simplificaciones de las actividades más importantes al momento de desarrollar un nuevo producto de software, por eso decimos que un modelo de software es una representación de alto nivel de un proceso de software.

---

<sup>2</sup> Universidad Nacional de Educación a Distancia (2010), Técnicas generales de diseño de software. En: <http://www.issi.uned.es/is/Disenio.pdf>

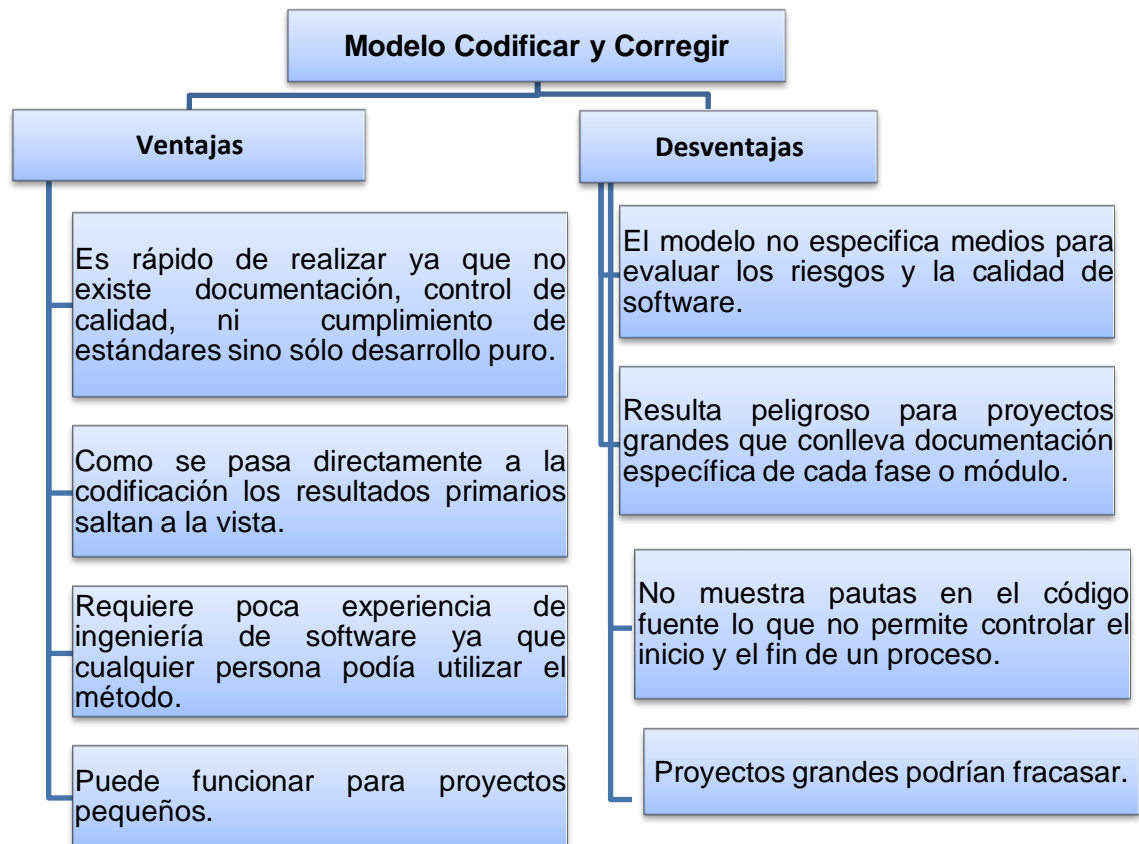
### **2.1.3 Modelos del ciclo de procesos más populares**

Los modelos de software describen y especifican fases o procesos encadenados entre ellos. Tenemos muchos modelos de procesos ya que un modelo es mejor que otro dependiendo de las características de cada uno y la finalidad de su utilización.

### **2.1.4 Modelo codificar y corregir**

Este modelo es muy utilizado pero no tan útil ya que no cuenta con una especificación formal. Si no se ha utilizado un método formal quiere decir que se está utilizando el método codificar y corregir. Este modelo empieza con una idea general de lo que se parte hacia una combinación de diseño, código o depuración y también métodos de pruebas que sirven para verificar la correcta funcionalidad del mismo.

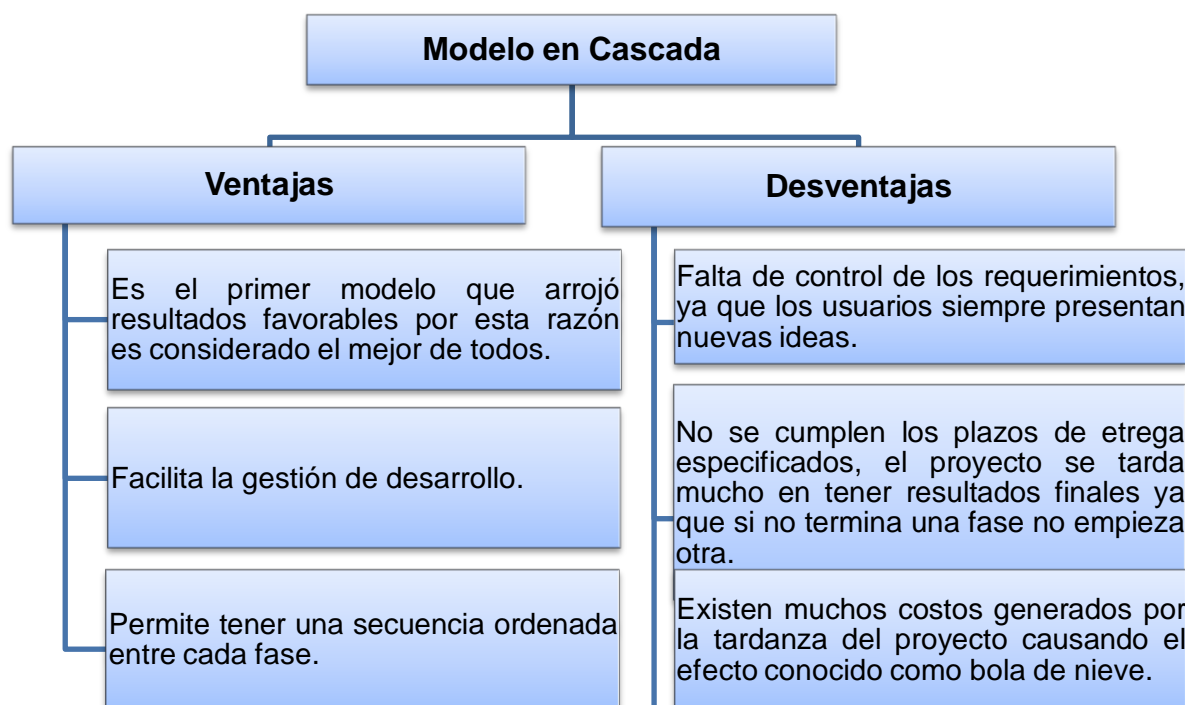




**Tabla 1 Ventajas y desventajas del Modelo Codificar y Corregir.**  
Elaborado: Carlos Landázuri.

### 2.1.5 Modelo en cascada

Poyce en 1970 fue el primero en utilizarlo, el mismo que lo llamaba un ciclo de vida básico o modelo secuencial ya que como indica este nombre es la ejecución secuencial de una serie de fases. Cada fase genera documentación que debe ser aprobada. Una fase no puede empezar sin concluir la anterior.



**Tabla 2: Cuadro Sinóptico Ventajas y Desventajas del Modelo Cascada.**  
 Elaborado: Carlos Landázuri.

### 2.1.6 Desarrollo Evolutivo

También conocido como modelo lineal secuencial está diseñado para el desarrollo en línea recta. Este modelo asume que se tiene que entregar un sistema completo, una vez que el sistema lineal haya terminado se entrega un prototipo, es muy útil para el usuario permitiéndole tener una idea clara del sistema y de sus requerimientos.

Los modelos evolutivos son interactivos. Se caracterizan por la forma que permiten desarrollar versiones cada vez más completas de software.

### Ventajas:

- El método evolutivo tiene la gran ventaja de reconocer cambios en los requerimientos permitiendo la satisfacción del cliente.
- Propone una solución apropiada para los problemas encontrados por el usuario sin que esto retrase el tiempo de entrega.
- Se obtiene productos lo más parecidos a los requerimientos del cliente.
- Se puede realizar un crecimiento en el sistema con nuevas funcionalidades ya que el sistema es en una parte escalable.

### Desventajas:

- No facilita la integración de aplicaciones que han sido desarrolladas como sistemas independientes.
- Facilita la posibilidad de que, en la medida que se evoluciona, pero no se pueda echar marcha atrás si algún proceso estuviera erróneo.
- Pueden ocurrir que el software nuevo es un reemplazo incremental de un subsistema dentro de un gran sistema existente. Si el sistema existente está con módulos mal realizados, entonces es obvia la dificultad en hacer que la nueva versión se acople con facilidad al resto.<sup>3</sup>

## **2.1.7 Desarrollo Formal de Sistemas**

Este modelo se basa en transformaciones formales de los requisitos hasta llegar a un programa ejecutable. Se conforma por dos fases globales:

- Especificación (incluyendo validación)
- Transformación.

---

<sup>3</sup> Wikipedia (2010) Modelo de desarrollo evolutivo. En: <http://es.wikimedia.org/modelosdesarrollo>

La especificación formal y ejecutable es la prima base que tiene el cliente para tener una idea global del proyecto así la especificación es validada mediante un prototipo del sistema. Posteriormente, a través de transformaciones formales la especificación se convierte en la implementación del sistema, en el último paso de transformación se obtiene una implementación en un lenguaje de programación determinado.

El desarrollo formal de sistemas permite demostrar la corrección del mismo durante el proceso de transformación, y así las pruebas que verifican la correspondencia con la especificación no son necesarias. Este método es atractivo sobre todo para sistemas donde hay requisitos de seguridad y confiabilidad importantes.

Requiere desarrolladores especializados y experimentados en este proceso para llevarse a cabo.

### **2.1.8 Desarrollo Incremental**

El modelo incremental es un modelo de tipo evolutivo que está basado en muchos ciclos cascada realimentados, aplicados repetidamente con una filosofía interactiva.<sup>4</sup>

---

<sup>4</sup>Wikipedia (2010), Desarrollo formal de sistemas. En: [http://es.wikipedia.org/wiki/Software#Modelos\\_evolutivos](http://es.wikipedia.org/wiki/Software#Modelos_evolutivos)



**Figura 1: Diagrama genérico del desarrollo Evolutivo Incremental**  
**Fuente: Fundamentos de sistemas<sup>5</sup>**

Ventajas:

- Con un modelo incremental se reduce el tiempo de desarrollo inicial ya que se considera una funcionalidad parcial.
- Se entrega rápidas partes operativas de software al cliente
- Proporciona ventajas del modelo cascada reduciendo sus desventajas sólo al ámbito de cada modelo incremental.
- Resulta más sencillo corregir errores y acortar el tamaño de incrementos no necesarios.
- Requiere una correcta planeación administrativa y técnica lo que ayuda a tener un correcto control del proyecto.

Desventajas:

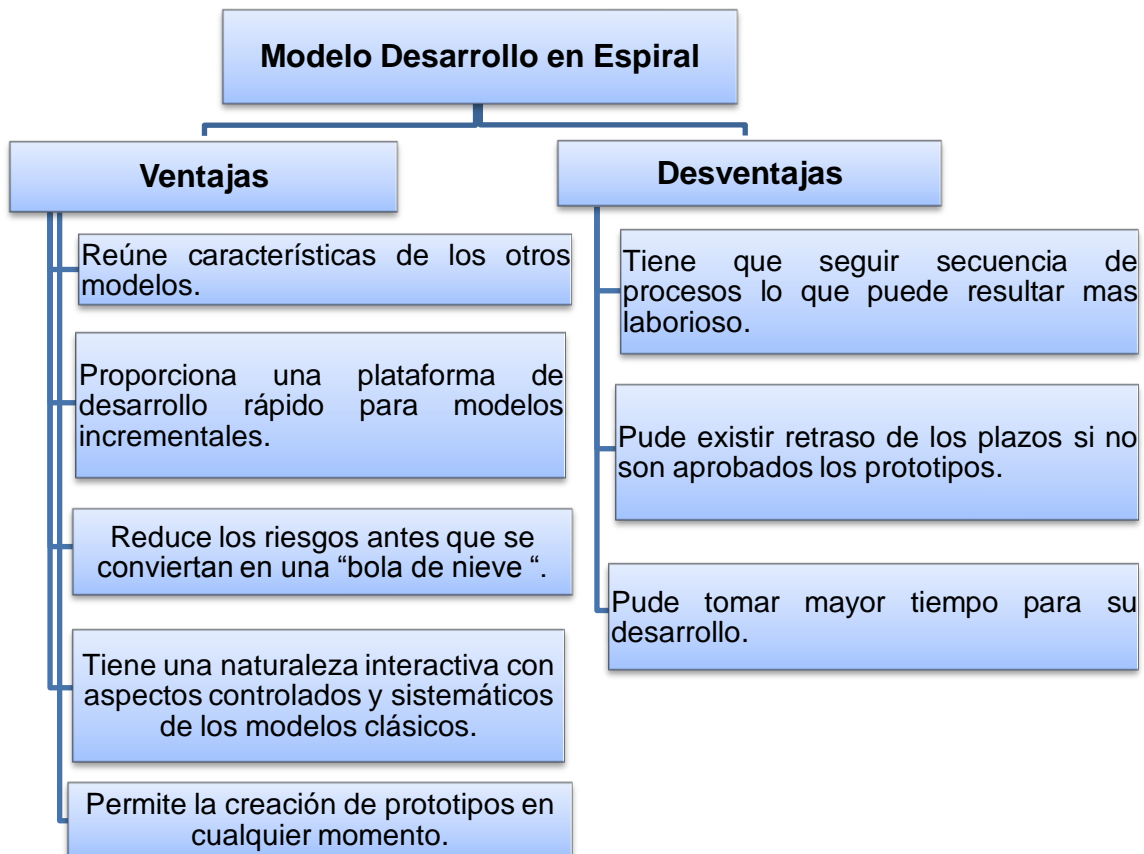
- No es recomendable en sistemas de tiempo real o de alta seguridad ya que tiene mucho índice de riesgo.
- Toma mucha tiempo para su planeación.
- Se necesita tener requerimientos claros del proyecto.

<sup>5</sup>Ing. Software (2009), DIAPOSITIVAS-UNIDAD 1. FUNDAMENTOS DE SISTEMAS. En: <http://ingsoftware072301.obolog.com/>

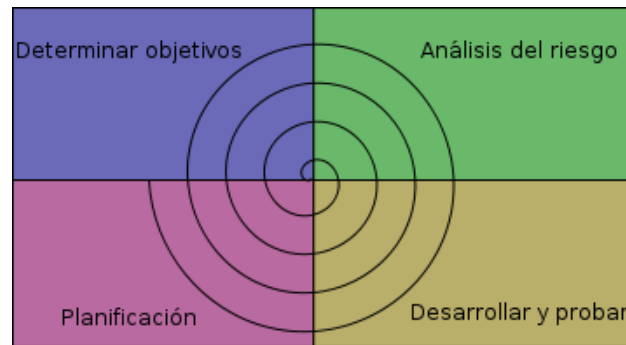
## 2.1.9 Desarrollo en Espiral

El desarrollo en espiral fue utilizado por primera vez por Barry Boehm en 1988. Actualmente es el enfoque más realista en la ingeniería de software para sistemas grandes.

Las actividades de este modelo como su nombre indica son en espiral, cada bucle o iteración representa un conjunto de actividades que siempre deben cumplirse para seguir con la siguiente etapa.



**Tabla 3 Mapa Conceptual Ventajas Y Desventajas Modelo Espiral.  
Elaborado: Carlos Landázuri.**



**Figura 2: Gráfico Modelo en Espiral**  
**Fuente: Modelos de desarrollo de software Wikipedia 6**

### 2.1.10 Desarrollo Basado en Reutilización

Este método se lo identifica también con la palabra “abstracción” que se refiere a la técnica de un párrafo de código fuente pueda ser utilizado en la construcción de otro programa.

El método basado en la reutilización puede verse en las bibliotecas de los compiladores donde se agrupan varias operaciones para facilitar el desarrollo de programas nuevos. Para que el código existente se pueda reutilizar, debe definir alguna forma de comunicación que permita llamar a los otros métodos y rutinas.

#### Ventajas:

- Reducción notable de código fuente.
- Mayor claridad en la estructura de un proyecto
- Claridad en las especificaciones de cada módulo, método, o clase.

#### Desventajas:

- Se requiere personas que dominen a la perfección el lenguaje de desarrollo.
- Puede provocar pérdida de tiempo tratar de desarrollar solo con este método.

<sup>6</sup> Wikipedia (2010). Desarrollo en Espiral. En: [http://es.wikipedia.org/wiki/Desarrollo\\_en\\_espiral](http://es.wikipedia.org/wiki/Desarrollo_en_espiral)

## 2.1.11 El ciclo de vida y los procesos

Los procesos primarios del ciclo de vida son la base para crear un producto de software. Las características de estos son:

- ✓ Son complejos de entender si no se los realiza de manera adecuada.
- ✓ No hay que confundirlos con procesos de producción.
- ✓ Muchas veces son improvisado por circunstancias impredecibles.
- ✓ No son procesos de ingeniería “pura”.
- ✓ Dependen demasiado de los desarrolladores.
- ✓ Diseño y producción no están claramente separados, presupuestos, calendarios, calidad no pueden ser planificados de forma fiable, de ahí nació la necesidad de las metodologías.
- ✓ Están basados en descubrimientos dependientes de la comunicación, coordinación y cooperación en los marcos de trabajo definidos.
- ✓ Los entregables generan nuevos requerimientos.
- ✓ Los costes del cambio del software no suelen reconocerse.
- ✓ El éxito depende de la implicación del usuario y de la coordinación de muchos roles (ventas, desarrollo técnico, cliente, etc.).



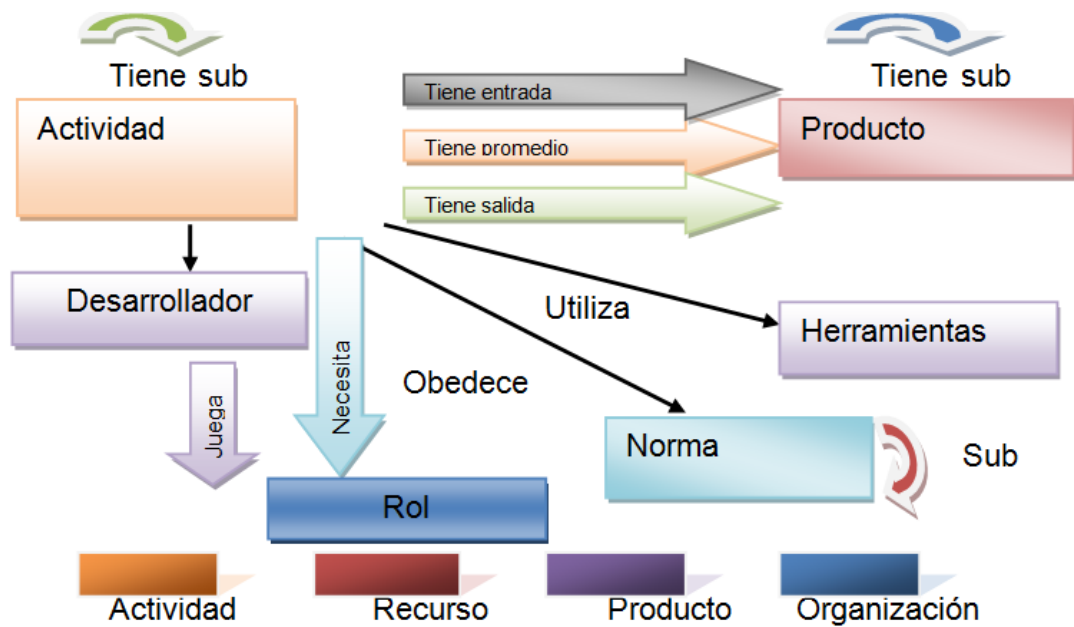


Figura 3 Ciclo de Vida de los Procesos.  
Elaborado: Carlos Landázuri.

## La planificación de la gestión del proyecto

Las tareas a realizar dentro de un proyecto requieren de una planificación estructurada y específica para cada miembro del equipo.

Planificación de la Gestión del Proyecto.	
Actividades.	Construcción de un mapa de actividades a realizar para el modelo ya elegido, tomando en cuenta la asignación de recursos, la definición del proyecto y la planificación de gestión.
Documentos de la planificación.	Plan de gestión.
Técnicas a Utilizar.	Cronogramas, diagramas, informes, estadísticas, simulaciones.

Tabla 4 Planificación de la Gestión del Proyecto.  
Elaborado: Carlos Landázuri.

## **La identificación de la necesidad**

Es el punto de partida de un proyecto donde se establece las necesidades específicas, para poner en marcha el desarrollo del producto que después de las evaluaciones dé la vialidad al mismo.

Identificación de la necesidad.	
<b>Actividades</b>	Buscar la vialidad del proyecto. Alza de requerimientos. Estudio de vialidad.
<b>Documentos de salida.</b>	Informe de necesidades físicas, informe de requerimientos, documento de riesgos y soluciones.
<b>Técnicas a Utilizar.</b>	Ganancia de conocimientos, análisis del costo beneficio, diagramas de flujo.

**Tabla 5: Identificación de la Necesidad en los Ciclos de Vida.  
Elaborado: Carlos Landázuri.**

## **El proceso de especificación de los requisitos**

En la utilización de proceso de software es necesario tener un modelo claro y preciso que permita reunir los requisitos para el desarrollo del producto. La meta de todo esto es la totalidad de los requisitos de software. El análisis se lo realiza en base a los requerimientos del cliente, la idea del producto final, la descomposición de datos.

Las características o condiciones que tienen los programas para satisfacer las necesidades se denominan requisitos los mismos que pueden ser funcionales si especifican las funciones que deben realizar, de rendimiento si una característica numérica y de interfaces si especifican las características y propiedades de las interfaces.

## El proceso de diseño

Es la base para obtener un producto de calidad que satisfaga los requisitos de software. El diseño comprende cuatro tipos de actividades: el diseño de datos que modela las estructuras necesarias para el desarrollo, el modelo arquitectónico que define las relaciones entre las estructuras del programa y el procedimental que transforma estructuras en descripción del software. El diseño de interfaces define los mecanismos de interacción del usuario con el programa.

Los diseños modulares, reducen el riesgo de cambios permitiendo un desarrollo en paralelo de diferentes etapas (módulos) del proyecto que tienen como objetivo la cohesión y mínima interacción con los otros módulos.

	<b>Proceso de Diseño.</b>
<b>Actividades.</b>	Desarrollo del diseño arquitectónico, analizando el flujo de información, diseño de la base de datos, diseño de interfaces desarrollo de algoritmos
<b>Documentos de salida.</b>	Detalle del diseño de software, de la arquitectura del software, flujo gramas de la información y descripción de la base de datos.
	<b>Procesos de diseño</b>
<b>Técnicas a Utilizar.</b>	Diseño de base de datos estructurado. Diálogo de interface. Técnicas orientadas a objetos modelo clase objeto. Diagrama de módulos UML y Casos de uso.

**Tabla 6 Proceso de diseño en los ciclos de vida de los procesos**  
Elaborado: Carlos Landázuri

## El proceso de implementación

<b>Procesos de implementación.</b>	
<b>Actividades</b>	Crear los datos prueba en la BDD, generar el código fuente, documentar, iniciar la integración de módulos.
<b>Documentos de salida.</b>	Documentación del sistema según la metodología, plan de integración.
<b>Técnicas a Utilizar.</b>	Ganancia de conocimientos, análisis del costo beneficio, diagramas de flujo.

Tabla 7 Proceso de implementación en los ciclos de vida de los procesos.  
Elaborado: Carlos Landázuri.

## El proceso de instalación

Una vez realizados todos los pasos anteriores y ya teniendo una estructura final del sistema y las pruebas de funcionamiento se procede a realizar la instalación del producto de acuerdo a los requerimientos del cliente y previa aceptación del mismo.

<b>Proceso de instalación.</b>	
<b>Actividades</b>	Planificación, instalación de software, cargar base de datos, pruebas de funcionamiento.
<b>Documentos de salida.</b>	Informe de instalación.

Tabla 8 Proceso de instalación en los ciclos de vida de los procesos.  
Elaborado: Carlos Landázuri.

## Los procesos de mantenimiento y retiro

Con los errores detectados se realiza una depuración del sistema tratando de corregir las fallas detectadas con las mejoras solicitadas y cambios necesarios. Es un ciclo de vida en la aplicación pero con un software ya existente como iteraciones de desarrollo.

Los mantenimientos pueden ser correctivos, por defectos encontrados, adaptativos y de mejoras.

<b>Procesos</b>	<b>Mantenimiento</b>	<b>Retiro</b>
<b>Actividades</b>	Re aplicar el ciclo de vida.	Notificar al usuario los cambios y retiro del sistema.
<b>Documento de salida</b>	Orden de mantenimiento. Recomendaciones.	Pan de contingencia de la información, plan de retiro, informe.

**Tabla 9 Los procesos de mantenimiento y retiro en los ciclos de vida.  
Elaborado: Carlos Landázuri.**

### **El proceso de validación**

Las tareas recomendadas para realizar un correcto proceso de validación son: pruebas de verificación, revisiones y auditoria; e incluye las tareas de validación y pruebas de validación que se realizan durante el ciclo de vida del software.

Las pruebas consisten en ejecutar el sistema con algunos datos de entrada y producir resultados que puedan ser comprobados.

<b>Proceso de validación.</b>	
<b>Actividades</b>	Planificar y ejecutar tareas de verificación y validación. Recoger y analizar los datos que servirán como parámetros de validación. Planificar las pruebas, ejecutar las pruebas.
<b>Documentos de salida</b>	Plan de acción, plan de pruebas, especificación de las pruebas, resultados de las pruebas.
<b>Técnicas a Usar</b>	Pruebas de simulación de procesos. Revisiones formales y auditorias.

**Tabla 10 Procesos de validación en los ciclos de vida de los procesos.  
Elaborado: Carlos Landázuri.**

## **El proceso de la gestión de la configuración**

Dentro del ciclo de vida del software está el proceso de configuración que conlleva la gestión de cambios cuyo objetivo es el control de los cambios producidos y el control de los mismos.

<b>Proceso de configuración.</b>	
<b>Actividades.</b>	Análisis y planificación previo antes de configuración, realizar el control de la configuración y un informe del estado del mismo.
<b>Documentos de salida.</b>	Plan de acción de la configuración, orden de cambio, informe de cambio.

**Tabla 11 Procesos de configuración en los ciclos de vida de los procesos.  
Elaborado: Carlos Landázuri.**

## **Los procesos de desarrollo de la documentación y de formación**

Esta fase nos permite organizar toda la información concerniente al proyecto realizado, con el fin de mantener los documentos para los desarrolladores y los usuarios.

Para que el sistema sea funcional debe proporcionar al usuario las instrucciones y guías necesarias para que puedan orientarse acerca del uso del sistema y de las limitaciones. También es importante la capacitación del usuario, de los desarrolladores y el soporte técnico.

Procesos	Documentación	Formación
<b>Actividades</b>	Planificar e implementar la documentación, producir y distribuir la documentación.	Planificar el programa de formación. Desarrollar materiales de formación. Validar e implementar el programa.
<b>Documento de salida</b>	Documentación general.	Plan de formación.

**Tabla 12 Los procesos de documentación y de formación en los procesos.**  
Elaborado: Carlos Landázuri.

### **La selección de un ciclo de vida**

El ciclo de vida para el desarrollo de un sistema va de la mano con las características del producto a obtener, a partir de los requisitos del desarrollo especificado.

Una vez se tiene claro el sistema a desarrollar se realizará una adaptación de los procesos vistos anteriormente en forma general y realizar la matriz de actividades, partiendo del mapa de actividades. A partir de esta matriz se podrá tener una estimación del tiempo que tomará finalizar el proyecto, además se podrá tomar en cuenta el costo de cada actividad y del proyecto global.

#### **2.1.12 Calidad en el Software**

Calidad de software son todas las características inherentes que puede tener un producto que se puedan asegurar y controlar en un sistema basado en estándares, con funcionalidad y rendimiento que satisfagan las necesidades del cliente.

### **2.1.13 La calidad en ingeniería de software**

Todas las metodologías y herramientas tienen un único fin producir software de gran calidad.

#### **Definición**

Algunos personajes como Deming definen la calidad como la conformidad con requisitos y la confiabilidad de su correcto funcionamiento. Pero la norma ISO 8402 define la calidad como la totalidad de características de un producto o servicio que le confieren su amplitud para satisfacer unas necesidades expresadas o implícitas.<sup>7</sup> La calidad del software se la puede definir como el conjunto de cualidades que caracterizan y determinan la utilidad de un sistema y su existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema a otro. La calidad del software se puede medir después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.<sup>8</sup>

#### **El aseguramiento de la calidad:**

Aseguramiento de la calidad: Son las tareas que realizadas de forma planificada y sistemática deberían proporcionar la confianza adecuada para que un producto o servicio cumpla con las expectativas y requerimientos del cliente. Aseguramiento de la calidad de software: Conjunto de actividades

---

<sup>7</sup> Calidad de software Fuente: [www.iso.com.es](http://www.iso.com.es)

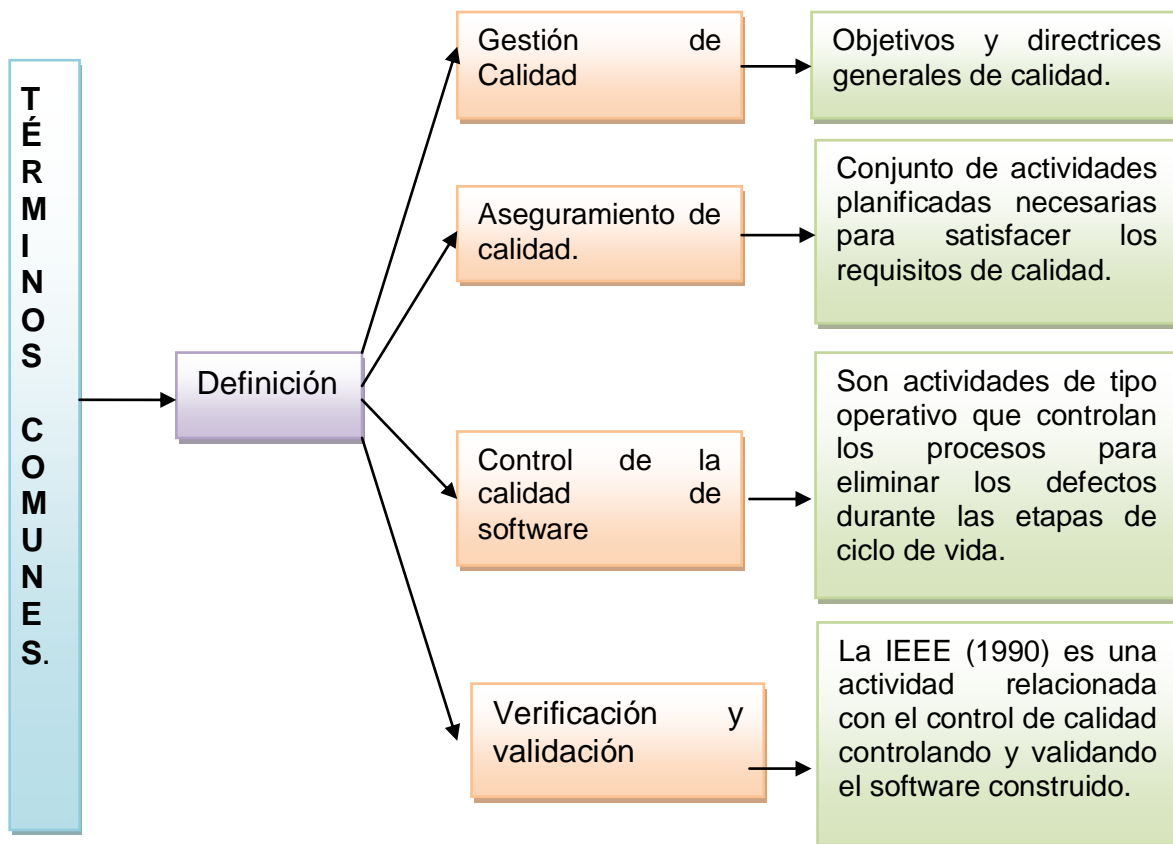
<sup>8</sup>Oscar M. Fernández, Delba García y Alfa Beltrán. (2005). Un enfoque actual sobre la calidad del software. En: [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)



planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfaga los requisitos dados de calidad total.<sup>9</sup>

### 2.1.14 La calidad desde el punto de vista de las ISO

Las normas ISO son un conjunto de estándares en las que se apoya el sistema de calidad de una empresa. La norma ISO 9000 define al sistema como la estructura de organización de procedimientos necesarios para la gestión de calidad. Entre las normas que definen y aseguran la calidad de los productos de software, están las de la familia ISO 9000(ISO 9000.1991) Esta familia de normas específica el diseño para aplicaciones industriales en general existe una versión 9000-3, específica para productos lógicos.



**Tabla 13 Términos comunes utilizados en la calidad de software.**  
Realizado: Carlos Landázuri.

<sup>9</sup>Monografías (2011). Gestión de calidad del software. En: <http://www.monografias.com/trabajos59/calidad-software/calidad-ssoftware2.shtml#xcalidadsoft>

## **La Norma ISO 9003**

La norma ISO 9003 también conocida como ISO/IEC 9000-3:2004 fue propuesta en el comité técnico de tecnología de la información y el subcomité de ingeniería de software de sistemas. Esta norma tiene como finalidad establecer las pautas para aplicar la Norma ISO 9001 la misma que establece un sistema de calidad aplicando los procesos de adquisición, provisión, desarrollo, operación y mantenimiento y servicios de ayuda relacionados.

Es muy poco probable obtener buenos resultados en un proyecto de desarrollo de software en organizaciones de tamaño mediano, ya que no se toma en cuenta algunos temas como la Administración de la Configuración o Planeación de Proyectos que no están presentes en las normas de la familia ISO 9000, mas si en la ISO 9003. Esta característica implica que para ciertos productos y servicios, la especificación de requerimientos contenida en las normas de ISO 9001 no sea la suficiente para asegurar la calidad, justificando la necesidad de otras normas o guías más específicas como complemento. ISO 9000-3 no impone un modelo de ciclo de vida específico. Asimismo, no provee métodos específicos para evaluar la capacidad de aseguramiento de la calidad de una organización. Por lo tanto, puede ser combinado con otros enfoques más específicos, como por ejemplo el modelo Espiral de Boehm.

En resumen, esta norma supone un cambio para que los ambientes tecnológicos más especializados avancen, especialmente los enfocados al diseño y desarrollo de software estableciendo pautas si bien no tan específicas pero que sirven como guía para garantizar el éxito del proyecto.

Parámetros a evaluar	
<b>Calidad de Diseño</b>	Exactitud: Que el software cumpla con las especificaciones y se ajuste a los objetivos planteados.
	Mantenibilidad: Facilidad de localización de una falla de software dentro de un espacio seleccionado.
	Verificabilidad: Facilidad para comprobar el rendimiento y funciones de software en base a los requerimientos.
	Eficiencia: Podemos medirla considerando el alcance del producto final con respecto a los recursos del sistema que utiliza es decir hacer más con menos recursos del sistema.

**Tabla 14 Estándares a evaluar según la Norma ISO 9003.  
Elaborado: Carlos Landázuri**

Parámetros a evaluar	
<b>Calidad de Desempeño</b>	Integridad: Capacidad de protección del sistema a la intrusión de usuarios
	Integridad: Capacidad de protección del sistema a la intrusión de usuarios no autorizados.
	Fiabilidad: Comprobación de las funciones que tiene el sistema con respecto a los requerimientos.
	Usabilidad: Se refiere a la capacidad de los usuarios para comprender y usar el sistema.
	Comprobabilidad: Facilidad que tiene el producto final para la realización de pruebas en comprobación de su correcto funcionamiento.

	Parámetros a evaluar
<b>Calidad de adaptación.</b>	Expansibilidad: Se refiere al esfuerzo para ampliar el funcionamiento del proyecto y el rendimiento luego de dichos cambios.
	Flexibilidad: Esfuerzo que hacer posible el cambio de funciones o datos para satisfacer los requerimientos funcionales.
	Portabilidad: Capacidad de cambio de un entorno a otro ya sea del software o de otra plataforma.
	Reusabilidad: Facilidad para usar el software o sus componentes en otros sistemas y aplicaciones.
	Interoperabilidad: Relativo al esfuerzo necesario para acoplar el software en una plataforma diferente.
	Intra-Operabilidad: Esfuerzo necesario para las comunicaciones entre componentes del mismo sistema.

**Tabla 15 Estándares a evaluar según la Norma ISO 9003**  
**Elaborado: Carlos Landázuri.**

## **El Estándar ISO 9126**

El ISO/IEC 9126 es un estándar internacional que permite evaluar la calidad del producto software. Este estándar actualmente se encuentra siendo guiado y supervisado por el ISO/IEC 25000:2005 y una serie de estándares agrupadas dentro del proyecto de “Software Product Quality Requirement and Evaluation”. El ISO/IEC 9126 contiene un modelo de medición y un modelo calidad que permiten llevar a cabo la evaluación de la calidad de los productos software.

Actualmente el estándar está conformado de 4 partes que dirigen: un modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. La primera parte agrupa y define un conjunto de características y sub características para determinar la calidad de un producto estas características se muestran en la siguiente tabla.

Características que se evalúan en el Estándar ISO 9126:

<b>Funcionalidad</b>	<b>Eficiencia</b>
Condiciones que cumplan con los requerimientos específicos del cliente ✓ Adecuación. ✓ Exactitud. ✓ Operatividad. ✓ Seguridad. ✓ Conformidad.	Característica con la cuál se garantiza el rendimiento apropiado.  ✓ Tiempo de respuesta. ✓ Utilización de Recursos. ✓ Conformidad.
<b>Fiabilidad.</b>	<b>Mantenibilidad.</b>
Capacidad del producto para mantener el nivel de rendimiento especificado. ✓ Madurez ✓ Tolerancia a fallos ✓ Recuperabilidad ✓ Conformidad.	Capacidad del software para poder ser modificado. Dichas modificaciones pueden ser correcciones, o mejoras de adaptación al sistema. ✓ Fácil de analizar. ✓ Fácil de cambiar ✓ Estabilidad.
<b>Usabilidad.</b>	<b>Portabilidad</b>
Característica del producto para ser entendido, utilizado y agradable al usuario. ✓ Fácil de comprender ✓ Fácil de manejar. ✓ Operatividad ✓ Conformidad. ✓ Atractivo.	Capacidad del producto para ser transferido de un lugar a otro. ✓ Adaptabilidad ✓ Fácil de instalar. ✓ Conformidad.

**Tabla 16 Parámetros más relevantes de la norma ISO 9126**  
 Elaborado: Carlos Landázuri.

El estándar en una segunda parte define las métricas externas necesarias para evaluar con respecto a las características y sub características mencionadas en la primera parte. En la tercera parte el estándar examina las métricas internas necesarias para estimar las características de calidad de un nuevo sistema que quiera contar con parámetros de dicho estándar.

La última parte, estructura las métricas para definir la calidad en uso de un producto software.

## **Estándar ISO 12207**

Este establece un marco de referencia común para los procesos del ciclo de vida del software, con una terminología bien definida, que puede ser referenciada por la industria del software”.

Esta nos presenta las siguientes características:

- Define los procesos, actividades y tareas que constituyen cada actividad presentes en la adquisición, suministro, desarrollo, operación y mantenimiento del software.
- Según esta norma, un proceso es un conjunto de actividades interrelacionadas que transforman entradas en salidas.
- Un proceso define quién, qué, cuándo, y cómo, para alcanzar un determinado objetivo.

Procesos que se debe tomar en cuenta en la utilización de este estándar.

### Procesos del ciclo de vida del estándar 12207

<b>Procesos Principales</b>	<b>Organizacionales</b>	<b>Soporte</b>
<ul style="list-style-type: none"><li>•Adquisición</li><li>•Suministro</li><li>•Desarrollo</li><li>•Explotación</li><li>•Mantenimiento</li></ul>	<ul style="list-style-type: none"><li>•Gestión</li><li>•Infraestructura</li><li>•Mejora</li><li>•Recursos Humanos</li><li>•Gestión de Activos</li><li>•Gestion de Reutilización</li><li>•Dominio</li></ul>	<ul style="list-style-type: none"><li>•Documentación.</li><li>•Gestión de Configuración</li><li>•Aseguramineto de calidad</li><li>•Verificación</li><li>•Validación</li><li>•Revisión Conjunta</li><li>•Auditoria</li><li>•Resolución de problemas</li><li>•Usabilidad</li><li>•Evaluación del producto.</li></ul>

**Tabla 17 Procesos del ciclo de vida del estándar 12207**

**Elaborado: Carlos Landázuri.**

## **Métricas de calidad del software**

El concepto de métrica de calidad de software es el término que describe muchos y muy variados casos de medición. Siendo una métrica una medida estadística (no cuantitativa como en otras disciplinas ejemplo física) que se aplica a todos los aspectos de calidad de software, los cuales deben ser medidos desde diferentes puntos de vista como el análisis, construcción, funcional, documentación, métodos, proceso, usuario, entre otros.<sup>10</sup>

### **2.1.15 Diferentes aspectos que intervienen en la medición de calidad**

Para medir la calidad de software es necesario conocer quiénes intervienen en la calidad del producto de software.

- Cliente
- Empresa o los empresarios.
- Personal técnico programadores.

#### **➤ Calidad Interna**

Esta es medible a partir de algunos aspectos internos como el código fuente, dentro de este existen conceptos como el de McCall (1977) establece el concepto de calidad en la subdivisión de las propiedades más sencillas de medir y evaluar como la facilidad de uso, integridad, fiabilidad, corrección, facilidad de prueba y facilidad de mantenimiento en esta descomposición del concepto de calidad se desprenden tres usos importantes para el usuario que son:<sup>11</sup>

---

<sup>10</sup>Monografías (2010). Ingeniería del Software. En: <http://www.monografias.com/trabajos15/ingenieria-software/ingenieria-software2.shtml>

<sup>11</sup>Pressman, Roger (2011). Métricas Técnicas. En: <http://itzamna.bnct.ipn.mx:8080/dspace>

- Características de operación
- Capacidad para soportar cambios
- Adaptabilidad de nuevos entornos.

### ➤ **Calidad externa**

En el momento de que el producto este finalizado entran algunos parámetros de medición de calidad como los que proponen las normas ISO o la IEEE esta última la IEEE 1061 propone un modelo de medición muy parecido al de McCall y la norma ISO 9126 (ISO,1991) establece un modelo propio, similar al de McCall pero también tomando en cuenta factores externos ya que se evalúa cuando el sistema ya está en la fase de prueba.

En la década del ochenta, se comenzó a usar modelos particulares de evaluación para cada empresa o proyecto, implantándose el concepto de calidad relativa.

Gilb (1988) propone la creación de una especificación de requisitos de calidad a redactar conjuntamente el usuario y los analistas, determinando así la lista de características que definan la calidad de cada aplicación. Este enfoque se ha asociado a la filosofía QFD (Quality Function Deployment), o el despliegue de la función de la calidad que se aplica al ámbito de la gestión de la calidad industrial y el que se han basado modelos posteriores. Otros modelos son los de Basili y Rombach (1988) proponen el paradigma GQM, (objetivo–pregunta– métrica o goal-question- metric) para evaluar la calidad de cada proyecto.



➤ **Calidad en uso.**

Este aspecto es realizado por el usuario cuando el sistema ya está en uso. Si el sistema cumple con las necesidades del cliente y efectúa los requerimientos establecidos al inicio del proyecto se puede decir que el sistema cumple con los parámetros de calidad por parte de los usuarios.

### **2.1.16 Metodología de desarrollo**

Una metodología de desarrollo de software se refiere a un framework que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. El framework para metodología de desarrollo de software consiste en:

- Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software.
- Herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

Estos Frameworks son a menudo vinculados a algún tipo de organización, que apoya el uso y promueve la metodología. La metodología es a menudo documentada en algún tipo de documentación formal.<sup>12</sup>

---

<sup>12</sup> Wikipedia (2010). Metodología de desarrollo de software. En: [http://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)

## **Introducción e importancia**

Las metodologías de desarrollo comprende un grupo de elementos ordenadamente relacionados para conseguir un fin su importancia radica en que nos sirven para especificar cómo dividir un proyecto en etapas, que actividades o tareas debemos realizar en cada etapa, artefactos que generan, entradas salidas, secuencias de las tareas, restricciones, etc.

Las metodologías nos ayudan a describir quien (roles), que (método), cuando (proceso) y como (técnica). Hay unos pasos y procedimientos que deben seguirse para el desarrollo de software:

- Como se debe dividir un proyecto en etapas.
- Que tareas se llevan a cabo en cada etapa.
- Heurísticas para llevar a cabo dichas tareas.
- Que salidas se producen y cuando se deben producir.
- Que restricciones se aplican.
  
- Que herramientas se van a usar.
- Como se gestiona y controla un proyecto.

## **Evolución de las metodologías de desarrollo**

Al inicio del desarrollo del software los primeros desarrolladores no contaban con una metodología definida que sirviera de guía para desarrollar software, es decir que el desarrollo era artesanal y convencional. La problemática se enfocaba en falta de control en el desarrollo de un proyecto lo que arrojaba resultados no esperados.

Estas limitaciones originaron una serie de problemas y por ende la necesidad de una guía que garantizara la calidad total del proyecto. La programación estructurada, aparece en los años sesenta y setenta de la mano con el desarrollo científico y empresarial. Tiene como punto de partida el uso de normas de estructuras de datos y control. En los setenta la programación estructurada comienza a utilizar fases de diseño y gestión de construcción de software.

Uno de los principales problemas con los que se enfrentaban los desarrolladores de software de aquellos tiempos es que contaban con un código desarrollado monolíticamente lo que dificultaba encontrar los errores y las soluciones de los mismos, es por esta razón que en las publicaciones de los autores Myers (1975) Constantine (1975) y Page-Jones (1980), se manifiesta que debe haber un componente básico o módulo estructural pasando después a la normalización de los módulos de el programa para por último hacer un control final de producto. En este período ya se comienzan a tratar términos como “calidad de los programas”.

La base de la programación y el diseño estructurados, es un análisis del problema usando el diseño *top-down* o *descendente*, donde la parte primordial esta en las especificaciones funcionales. Se compone de diagramas, con textos de referencia de los mismos, y con independencia para que se puedan leer en forma parcial lo que permite tener una referencia de cada etapa.

También, se puede destacar, que ha habido una evolución en cuanto al modelado de sistemas en tiempo real, modelado de datos y estudio de eventos. El modelo orientado objetos aparece más tarde, se refiere a los procesos y los datos encapsulándolos como módulos de acuerdo a la información y el procesamiento.

Aparece el Smalltalk, con énfasis en la abstracción de datos, considerando a los problemas como un conjunto de objetos de datos a los que se les adicionaba un conjunto de operaciones, pero se fundamenta en la abstracción, la modularidad y el ocultamiento de la información, derivados del diseño estructurado.

En los ochenta, aparecen el C++ y el object C y más tarde el Lenguaje ADA, del Departamento de Defensa (DoD) de los Estados Unidos para la definición y mejora de tecnologías basada en este lenguaje.

En general para los desarrollos de las metodologías orientadas al objeto se tomaron de los conceptos de técnicas estructuradas.

### **Características de las Metodologías.**

Existen varias características que debe tener una metodología de desarrollo, estas varían de acuerdo al modelo propuesto por cada una, la mayoría utilizan parámetros que nos lleven a cumplir con estándar de calidad Microsoft Solution Framework utiliza la mayoría de las enunciadas a continuación:

- Reglas predefinidas.
- Determinación de los pasos del ciclo de vida.
- Verificaciones en cada etapa.
- Planificación y control.
- Comunicación efectiva entre desarrolladores y usuarios.
- Flexibilidad: aplicación en un amplio espectro de casos.
- Soporte de herramientas automatizadas.
- Que permita definir mediciones que indiquen mejoras.
- Que permita modificaciones.
- Que soporte reusabilidad del software.

## 2.1.17 Clasificación de las Metodologías

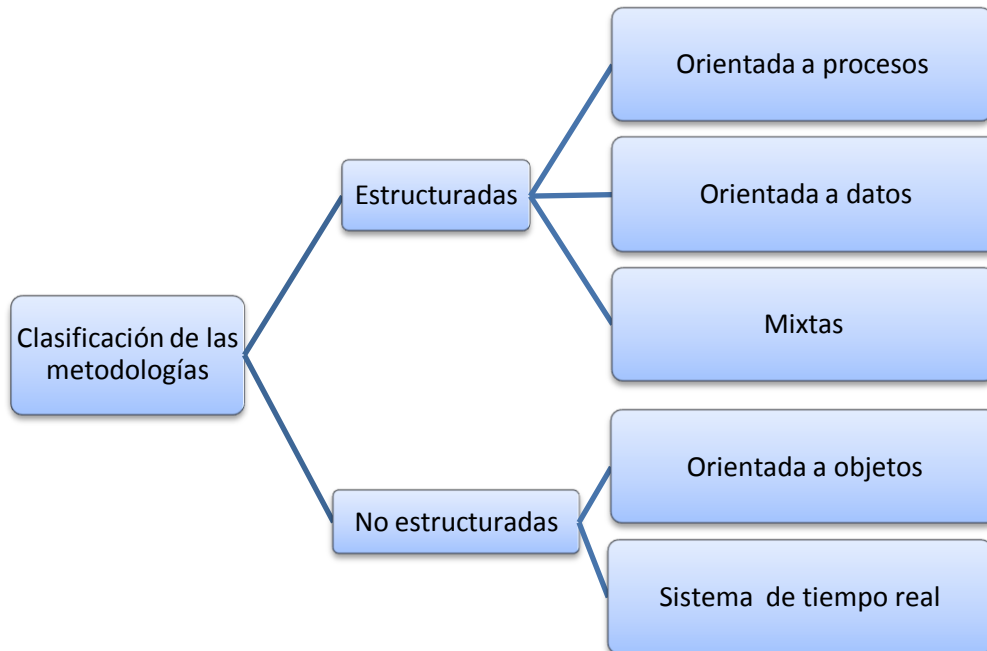


Tabla 18 Clasificación de las metodologías de desarrollo de software.

Elaborado: Carlos Landázuri.

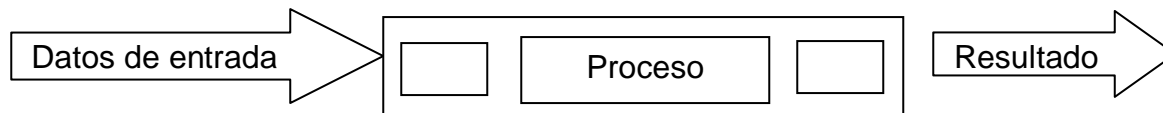
## 2.1.18 Metodologías Estructuradas

Los problemas en la estructura y composición de un proyecto pueden ser dirigidos por una metodología de desarrollo estructurada ya que esta permite una descomposición funcional de problemas en unidades más pequeñas que se relacionan entre sí, las mismas que pueden ser representadas en flujos de datos y de manera jerárquica con la finalidad de tener una visión más clara del proyecto.

Los proyectos implementados utilizando estas metodologías, utilizan mucho la manipulación de datos de ficheros o bases de datos y gestión documental siempre separando los datos de los procesos basándose en ciclos de vida en cascada.

Su principio fundamental es examinar el sistema desde las funciones y tareas, mientras que en las metodologías orientadas a objetos modelan el sistema examinando el dominio del problema como un conjunto de objetos que interactúan entre sí.

### ➤ **Orientadas a Procesos**



**Figura 4 Metodologías orientadas a procesos.**

**Elaborado: Carlos Landázuri.**

Se funda en el modelo básico: entrada/proceso/salida. Enfoca la parte del proceso. Generalmente las especificaciones se basan en:

- Diagrama de flujo de datos.
- Diccionario de datos.
- Especificaciones de procesos
- Realizar diagrama de flujo de datos del sistema.
- Realizar diagrama de estructuras.
- Evaluar el diseño.
- Preparar el diseño para implantación.

## ➤ **Orientadas a Datos**

Con la aparición de los procedimientos de la Orientación a Objetos surgieron métodos, procesos y metodologías específicas como OMT (Object Modeling Technique), RUP o Microsoft Solution Framework, entre otras.

Ante la necesidad de normalizar el proceso de desarrollo se empezaron a tomar en cuenta las metodologías en las que priman las fases, actividades y tareas antes que a las personas: lo más importante es el rol que juega la persona dentro del proceso de desarrollo. Esto se ha justificado históricamente argumentando que el desarrollo de software es una actividad compleja que debe estar dirigida por un proceso y debe llevarse a cabo en grupos.

Para cubrir todas las fases se necesitan distintos y diversos perfiles, que no se encuentran en una sola persona. El rol se asigna dependiendo de la experiencia, formación y capacidades de cada individuo.

Los roles de los implicados pueden ser similares entre todos pero cada uno cumple una función específica en todos los procesos: administrador de proyecto, analista, diseñador, programador, asegurador de calidad, documentalista, ingeniero de validación y verificación, administrador de la configuración. En un nuevo proyecto no siempre es necesario que estén todos los actores del desarrollo eso varía de acuerdo a la complejidad de cada proyecto.

Los roles de cada perfil está compuesto por los objetivos y las actividades que se necesitan para cumplir con dicho objetivo, para esto interactúan entre ellos y las herramientas a utilizar, además del perfil de las personas en ese rol y un plan de trabajo.

Una de las ventajas de estos procesos es que el intercambio de un recurso es relativamente fácil: cuanto más identificadas se tengan las tareas que realiza un rol, más fácil es formar a una persona en esas tareas específicas.

### ➤ **Jerárquicas**

La estructura de datos del programa tiene un nivel jerárquico que se deriva según los datos del programa y las importancia de cada uno de ellos. Para el diseño de este modelo se tiene en cuenta la estructura de datos de entrada y salida estructurarlas en forma jerárquicas y después desarrollar una lógica de desarrollo que se ajuste a la estructura.

### ➤ **No Jerárquicas**

Las metodologías no jerárquicas utilizan primero una arquitectura de la información que sería para cualquier tipo de proyectos y una forma de hacer coincidir esta arquitectura con los objetivos. Después vendría un análisis donde se trazan los requisitos del sistema y por último un diseño donde ya se enfoca en el sistema requerido por el usuario pero que esté acorde a la tecnología existente.

### ➤ **Mixtas para Sistemas de Tiempo Real**

Se basan en un control de los procesos de la información más que en los datos en sí, priorizando procesos de comunicación entre tareas y acceso simultáneo de datos iguales.



## 2.1.19 Metodologías de desarrollo Ágiles vs Tradicionales

Metodologías Ágiles	Metodologías Tradicionales
Con una filosofía heurísticas provenientes de prácticas de producción de código	Se basan en normas y estándares de desarrollo
Permiten cambios de especificaciones en el proyecto en desarrollo.	Resistencia a cambios.
No existe un contrato tradicional o en algunos casos es muy flexible.	Siempre existe un contrato inicial.
El cliente es parte del equipo de producción.	El cliente interactúa solo con el jefe del proyecto.
Pueden ser equipos pequeños.	Grupos grandes y viene estructurados.
Pocos artefactos.	Más artefactos.
Arquitectura de software no muy utilizada.	La arquitectura del software es imprescindible.

Tabla 19 Tabla comparativa de metodologías ágiles vs tradicionales.

Elaborado: Carlos Landázuri.

## 2.1.20 Metodologías de desarrollo de software más populares

La finalidad de las metodologías desde sus inicios por el año de 1960 es ayudar a los profesionales con pautas para realizar y documentar cada etapa del desarrollo. Existen metodologías que han quedado atrás porque no funcionan, están obsoletas desde casi todos los puntos de vista de los entendidos entre estos se pueden mencionar:

- Desarrollo de sistemas de Jackson (JSD), de los años 80.
- Structured System Analysis and Design Method (SSADM). También de los 80. Muy popular en Europa, ya que tiene su origen el Reino Unido entre otras.

Sólo algunas metodologías tradicionales han sido revisadas y adaptadas y su funcionalidad suele ser utilizada en proyectos muy innovadores como RUP, MSF es por eso que para esta investigación se han tomado en cuenta las metodologías más populares.

### **2.1.21 Metodología de desarrollo de software XP**

La programación extrema o Extreme Programming (XP) fue creada por el año de 1999. Es el más destacado de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de otras metodologías porque pone más énfasis en la adaptabilidad. XP propone cambios de requisitos cuando el desarrollo ya está en marcha como un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.<sup>13</sup>

La programación extrema reúne características positivas de otras metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

---

<sup>13</sup>Schenone, Marcelo (2004). Diseño de una Metodología Ágil de Desarrollo de Software. En: <http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>

## **Características**

Las características fundamentales del método son:

- Desarrollo iterativo e incremental permitiendo hacer mejoras y cambios de acuerdo a los nuevos requerimientos.
- Pruebas unitarias continuas: Incluyendo pruebas de regresión, escribiendo el código de la prueba antes de la codificación.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Refactorización del código, es decir, rescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento.
- Responder a los cambios antes que seguir estrictamente un plan. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que se revisan e informan para posibles cambios a realizar.
- Simplicidad en el código: La mejor manera de desarrollo para que las cosas funcionen como lo establecido.
- La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.
- La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer.

## **Roles XP**

Para desarrollar un sistema con la metodología XP es necesario tener claro los actores que intervienen en este proyecto.

- Programador.

Encargado de la ejecución de las pruebas unitarias además de la producción del código del sistema, existiendo plena comunicación entre los programadores y otros miembros del equipo.

- Cliente.

El cliente es un miembro más del equipo, ya que él debe estar de acuerdo con las pruebas funcionales antes de realizar la implementación. Además, asigna la prioridad a tareas de usuario y decide cuales son las de mayor prioridad con lo que va aportar mayor valor al negocio ya que está representando a varias personas que se verán afectadas por el sistema.<sup>14</sup>

- Encargado de pruebas.

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los avances del proyecto.

- Encargado de seguimiento (Tracker).

Es el encargado del seguimiento del proyecto. Es el responsable de conseguir resultados en el equipo y de las herramientas de soporte para pruebas.<sup>15</sup>

---

<sup>14</sup>Canós, J., Letelier, P. y Penadés, M. (2008). Programación Extrema  
[http://www.xelphos.com.ar/xop\\_ma\\_xp.html](http://www.xelphos.com.ar/xop_ma_xp.html)

<sup>15</sup>Canós, J., Letelier, P. y Penadés, M. (2008). Programación Extrema  
[http://www.xelphos.com.ar/xop\\_ma\\_xp.html](http://www.xelphos.com.ar/xop_ma_xp.html)

- Entrenador (Coach).

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

- Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

- Gestor (Big boss)

Su labor esencial es de coordinación. Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas.

## ➤ **Artefactos de XP**

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

- Exploración

En esta primera etapa se planea rasgos generales del proyecto de acuerdo a las experiencias de los clientes.

Al mismo tiempo el equipo de desarrollo analiza las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran

las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- Planificación de la Entrega.

En esta fase el cliente establece la prioridad de cada requerimiento y los desarrolladores realizan una estimación de las necesidades para el desarrollo de cada uno de ellos. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses.

La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

- a. Iteraciones.

Esta fase está formada por planes de entrega, los mismos que están compuestos por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo los requerimientos del cliente que comprometan la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué requerimientos se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

- Producción.

Se caracteriza por el requerimiento de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

- Mantenimiento.

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

- Muerte del Proyecto.

Es cuando el cliente no tiene más necesidades para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.<sup>16</sup>

---

<sup>16</sup>Metodología XP (2010). Ciclo de Vida de un Proyecto. En: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>

## 2.1.22 Metodología de desarrollo de software SCRUM

Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum.<sup>17</sup>

### ➤ Características

Scrum recoge algunas prácticas reconocidas como buenas para el desarrollo ágil del software y las utiliza para crear su metodología entre estas tenemos:

- Desarrollo de Software iterativo e incremental. En Scrum, la planificación del proyecto maneja los requerimientos del cliente, y se realiza un análisis de los mismos en función del valor que proporcionan al cliente y su coste de desarrollo.
- Orientación a personas: En Scrum tanto el equipo como el cliente tienen responsabilidad y autoridad, porque forman un equipo multidisciplinar.
- Planificación con tiempo, tareas y personas. Cada iteración del producto, es asignada por tareas y tiempos para cada miembro, las cuales son detectadas por el equipo en conjunto, siendo así más fiables, ya que se basan en las experiencias e información de todos los miembros del equipo.

---

<sup>17</sup> Wikipedia (2010). Scrum. En: <http://es.wikipedia.org/wiki/Scrum>



- Control del progreso del proyecto: En Scrum, es el equipo el que se compromete a informar diariamente del progreso de las tareas que el mismo se ha asignado en las reuniones, informando del avance de las mismas así como de los problemas que vayan surgiendo.
- Gestión de cambios: Los cambios están permitidos y aceptados en el inicio de cada iteración, considerándose como algo natural, y permitiendo al cliente presentarlos una vez demostrados los resultados de cada iteración.
- Retrospectivas y análisis post mortem, las retrospectivas se hacen al final de cada iteración permitiendo que un mejor aprendizaje por parte del equipo, que puede ser de esta manera más productivo a la hora de desarrollar el proyecto.<sup>18</sup>

## ➤ **Roles Principales SCRUM**

- **Product Owner**

Es el responsable de cuidar los intereses de cada uno de los participantes en especial del cliente, se encarga de entrevistas para tener claros los requerimientos, estima el financiamiento inicial y se preocupa de que se cumplan los objetivos.

- **Scrum Master o Facilitador**

Es el líder del equipo, responsable de todos los procesos, debe enseñar la metodología Scrum a cada integrante implicado en el proyecto, preocupándose de poner la metodología en práctica.

---

<sup>18</sup>Penadés, M<sup>a</sup> (2009). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). En : <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

- Equipo de desarrollo.

Tienen la responsabilidad de entregar el producto finalizado. Pueden ser equipos pequeños con integrantes de 3 a 9 personas con las virtudes para realizar el análisis, diseño, desarrollo, pruebas, documentación, etc.

- Roles Auxiliares.

No tienen un rol formal, sus funciones son ocasionales por ejemplo capacitadores o profesionales involucrados con objetivos específicos del proyecto.

### **2.1.23 Metodología de desarrollo Crystal Clear**

Crystal Clear es una familia de metodologías con un “código genético” común. El nombre Crystal clasifica a un proyecto según su dimensión, la complejidad y los recursos que van a utilizar. Esta metodología hace menos énfasis en la documentación y da mayor prioridad a las versiones del proyecto que ya puedan ser probadas, no cuenta con una metodología específica sino alianza la que más se acomode al tipo de proyecto modificando los procedimientos si así es necesario. Cada metodología encaja en una parte diferente del módulo del proyecto ya que no considera lo mismo un proyecto donde intervienen 40 personas que en uno donde intervengan 6, eso puede ser una pérdida de dinero y demás recursos.

## ➤ **Características:**

- La comunicación es la parte fundamental de todo su equipo de trabajo y del cliente.
- Entrega frecuente: Consiste en entregar software a los clientes con frecuencia, pudiendo ser una entrega diaria semanal o mensualmente.
- Comunicación osmótica: Diálogos frecuentes con el equipo del proyecto discutiendo los temas a tratar con los entendidos de cada fase.
- Mejora reflexiva: Tomarse un pequeño tiempo para hacer un análisis global para analizar al desarrollo del trabajo haciendo: cotejas de notas, reflexionar y discutir.
- Foco. Saber lo que se está haciendo y tratar de cumplir con los objetivos trazados en el lapso establecido.
- Exploración de 360°: verificar el valor de negocios y del proyecto, los requerimientos, el modelo, la tecnología, las técnicas y los procesos.
- Victoria temprana: Uno de los objetivos es realizar pequeños triunfos inicial es lo que se traduciría como prototipos funcionales a que aspirar a una gran victoria tardía es decir la finalización completa del sistema.
- Arquitectura incremental: La arquitectura debe evolucionar en etapa, en caso de corregir errores se mantiene el sistema en ejecución mientras se modifica.

## ➤ **Roles**

- Patrocinador

Realiza la misión del proyecto, traza los objetivos y se encarga de establecer los objetivos con las prioridades de cada uno, además se encarga de conseguir los recursos para la totalidad del proyecto.

- Usuario Experto

Está ligado a las tareas del experto en negocios, realiza los casos de uso. Debe conocer a perfección el funcionamiento del sistema y su estructura para sugerir cambios en la parte visual y de operación.

- Diseñador Principal

Se encarga de la arquitectura del sistema, debe ser un profesional de Nivel 3 con plenos conocimientos en metodologías.

- Diseñador Programador.

Es el encargado de producir, junto con el diseñador principal, los borradores de pantallas, el modelo común de dominio, las notas y diagramas de diseño, el código fuente, el código de migración, las pruebas y el sistema empaquetado.

- Experto en Negocios.

Junto con el usuario experto produce la lista de actores-objetivos y el archivo de casos de uso y requerimientos. Debe conocer las reglas y políticas del negocio.

- Coordinador

Realiza un mapa general del proyecto, un plan de entrega, y un listado de riesgos que podría retrasar el trabajo del equipo.

- Verificador

Produce el reporte de bugs. Puede ser un programador específico para este cargo o puede hacerlo un conjunto del equipo total.

- Escritor

Redacta el manual de usuario unificando de la información documentada por el equipo de trabajo.

### **2.1.24 Metodología de desarrollo ASD (Desarrollo adaptable de software)**

Esta metodología fue desarrollada en 1990 es ideal para un proyecto dónde los cambios sean muy frecuentes ya que esta se adapta al cambio porque no hay un ciclo de planificación diseño y construcción sino más bien un ciclo basado en colaborar y aprender.

Como en la mayoría de metodologías ASD su ciclo reconoce los cambios y errores propios del desarrollo de software.

#### **➤ Características**

- Iterativo: el cliente interactúa con el equipo de trabajo para hacer conocer sus requerimientos y si el proyecto ya está en marcha las necesidades de cambio.
- Orientado a los componentes de software: La funcionalidad que el producto va a tener, características y estas adaptarlas a la tecnología existente más que a las tareas en las que se va a alcanzar dicho objetivo.

- Tolerante a los cambios: Una de las más importantes características que se adapta al cambio sin dejar que esto paralice la elaboración del producto.
- Guiado por los riesgos: La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

## ➤ Roles

- Líder

Encargado de todo el grupo de trabajo tiene que hacer cumplir los objetivos y métricas del proyecto.

- Desarrollo.

Su misión es la de guiar al grupo en la construcción del ERP y CRM y asegurar la calidad del producto.

- Planeación

Es el encargado de ofrecer soporte y guía al grupo de trabajo así mismo como dar seguimiento al proyecto.

- Calidad

La persona a cargo de esta fase tiene que hacer cumplir los estándares de calidad de software, todas las actividades deben sujetarse al plan de calidad plateado.

- Soporte

La persona encargada del soporte tiene a su cargo la administración de las herramientas de implementación de ERP o CRP además de administrar los procesos y sistemas de configuración.

### ➤ **Artefactos**

<b>Fase de Colaboración.</b>	<ul style="list-style-type: none"> <li>• Se centra aquí la mayoría del proyecto.</li> <li>• Grupo socializa experiencias ocurridas en el transcurso.</li> <li>• Todos aportar al proyecto con ideas.</li> </ul>
<b>Fase de Calidad del producto.</b>	<ul style="list-style-type: none"> <li>• Se pone en práctica los parámetros que miden la calidad de software .</li> <li>• Se verifica que se cumplan los parámetros de calidad.</li> </ul>
<b>Fase de calidad del producto vista por los desarrolladores.</b>	<ul style="list-style-type: none"> <li>• Se evalúa la calidad pero de un punto de vista técnico.</li> <li>• Adhesión a las normas y objetivos conforme a la arquitectura.</li> </ul>
<b>Fase de gestión del rendimiento.</b>	<ul style="list-style-type: none"> <li>• Evaluación para medir resultados del equipo.</li> <li>• Ideas para mejorar los logros.</li> </ul>
<b>Fase de situación del proyecto.</b>	<ul style="list-style-type: none"> <li>• Es el punto de partida para la construcción de la siguiente serie de características.</li> </ul>

**Tabla 20 Iteración de artefactos en ASD.**

**Elaborado: Carlos Landázuri.**

## 2.1.25 Metodología Ágil de desarrollo Rational Unified Process (RUP)

RUP es un proceso de la ingeniería de software que brinda un enfoque para la realización de tareas y asignar responsables en la elaboración de un sistema de software, enfocándose en los casos de uso, manejo de riesgos y manejo de la arquitectura.

La utilización de esta metodología garantiza mejores resultados en la productividad del equipo porque cada miembro puede acceder a la base de datos de conocimiento, además del lenguaje, la visión y los objetivos.

### ➤ Características

- Forma disciplinada de asignar tareas: Se asigna un responsable que organiza quién hace qué, cuándo y cómo.
- Permite implementar las mejores prácticas en Ingeniería de Software en cuanto a procesos y la documentación de cada uno.
- Desarrollo iterativo: Cada fase el proyecto tiene responsables y fechas de entrega.
- Administración de requisitos: El acta de requerimientos permite tener una correcta planeación del proyecto y de los implementos que se utilizarán en su desarrollo.
- Uso de arquitectura basada en componentes: Es una rama de la Ingeniería de software en la cual se trata con énfasis la descomposición del software en componentes funcionales. Esta descomposición permite convertir componentes pre-existentes en piezas más grandes de software.<sup>19</sup>

---

<sup>19</sup>Scribd (2009). Arquitectura Basada En Componentes. En: <http://es.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>



- Control de cambios: Al realizar el proyecto en módulos esto permite que los casos que se presenten solo afecten al módulo implicado y el resto del proyecto no se detenga.
- Modelado visual del software: Se refiere a la capacidad de utilizar herramientas de modelo visual como UML y crear los casos de uso.
- Verificación de la calidad del software. De acuerdo a los parámetros de calidad especificados al inicio del proyecto se verifica que el producto final cumpla con los parámetros establecidos.

## ➤ Roles

- Roles Analistas :

Los analistas cumplen un rol vital en el proceso de desarrollo, son responsables de investigar, planear, coordinar y recomendar opciones de software para cumplir los requerimientos del usuario. Un analista de sistemas para ser exitoso debe adquirir cuatro habilidades: Analítica, Técnica, Gerencial e Interpersonal.

- Roles Desarrolladores :

Su función consiste en trasladar las especificaciones del software en una aplicación, para ello es necesario primero diseñar una arquitectura la cual guie la construcción de software describiendo en general el cómo se construirá una aplicación de software mediante el uso de diagramas.

- Arquitecto de Software:

Es responsable de definir la arquitectura que guiará el desarrollo y la refinación continua de la misma en cada iteración, además de definir los lineamientos generales para el diseño e implementación, para probar aspectos riesgosos desde el punto de vista técnico.

- Desarrollador:

Este rol es responsable del código fuente del programa, tiene las tareas de desarrollar, depurar, mantener y documentar el código del programa, además de elaborar y ejecutar las pruebas unitarias sobre el código.<sup>20</sup>

➤ **Roles Gestores**

- Líder del proyecto

Este rol se encarga de establecer las condiciones de trabajo, dirigir y asignar recursos, coordina las interacciones con los clientes y usuarios finales, planifica las iteraciones, planifica y asigna el trabajo.

- Mentor

Es una persona muy ligada con el proceso de desarrollo de software, conoce las prácticas utilizadas y él porque de su uso, apoya a los equipos de trabajo mediante la revisión de los artefactos generados.

---

<sup>20</sup> UPS (2008). Roles que intervienen en RUP. En: <http://dspace.ups.edu.ec/bitstream/123456789/425/8/CAPITULO6.pdf>

➤ **Roles de Apoyo:**

Son personas interesadas en que sus necesidades sean satisfechas con la elaboración del software, estos roles son muy importantes para definir el alcance y los requerimientos del proyecto, entre ellos se encuentran el cliente, diseñador gráfico documentador técnico etc.

➤ **Artefactos**

<b>Fase de Incepción:</b>	<b>Fase de elaboración:</b>	<b>Fase de construcción:</b>	<b>Fase de Transición</b>
<ul style="list-style-type: none"><li>•Se redacta el documento de visión .</li><li>• Se establecen los objetivos generales tomando en cuenta la viabilidad del producto.</li><li>•Delimitando sus operaciones.</li></ul>	<ul style="list-style-type: none"><li>•Realización de los casos de uso.</li><li>•Lista de riesgos y sus estrategias.</li><li>• Calendario de actividades del grupo de trabajo</li><li>• Costo estimado del proyecto.</li></ul>	<ul style="list-style-type: none"><li>•Se establece la arquitectura del sistema con la capacidad de cumplir con los requerimientos funcionales.</li><li>•Comparación con objetivos</li><li>•Realización de un sistema completo.</li></ul>	<ul style="list-style-type: none"><li>•Se genera una versión beta de la aplicación.</li><li>•Se verifica los objetivos .</li><li>•Se establecen correcciones y adecuaciones</li></ul>

**Tabla 21 Artefactos utilizados en RUP.**

**Elaborado: Carlos Landázuri.**

## **2.2 CAPITULO II: SOFTWARE LIBRE**

### **Introducción**

La evolución informática en los últimos años ha generado un gran auge en lo relacionado al desarrollo de software y con ellos la existencia de licencias, permisos y derechos de autor entre otros. Al igual movimientos y estatutos que defienden el derecho al libre acceso del código fuente del software adquirido.

El software que una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente se lo considera como “SOFTWARE LIBRE”. El software libre, la mayoría de veces está disponible al público de forma gratuita o en algunos casos el valor a pagar es por la capacitación de la utilización del sistema más no por las líneas de código, ya que esto último no es obligatorio. No hay que confundir el término “software libre” con el de “software gratuito” ya que es gratuito pero muchas veces no garantiza los derechos de modificar o redistribuir el sistema.

### **2.2.1 Orígenes del Software Libre**

El software libre tiene sus inicios en los años 60 y 70 cuando el software era considerado como un valor agregado de los computadores, en esa época era común que los desarrolladores compartan la información de sus aplicaciones ya que todas estaban destinadas a lo mismo hacer más fácil el funcionamiento de los ordenadores, pero a finales de los 70 las compañías comenzaron a poner restricciones a los usuarios por el uso de sus programas (licencias). Estas licencias eran pasadas por alto para instituciones universitarias y algunas empresariales.

## 2.2.2 Evolución del Software Libre

En los años 80 el escenario empezó a cambiar ya que el uso de los computadores se volvió más común, los sistemas operativos competían en ser más agradables para el usuario, con ello la obligación de los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

Si un usuario encontraba algún error en la aplicación su deber era informar a la empresa de origen del sistema y ellos se encaraban de dar solución. En 1984 Richard Stallman el fundador de GNU se encontró con el problema de necesitar crear una aplicación que le informe el estado de su impresora y al pedir el código de configuración, se encontró con una negativa rotunda, esto fue el hecho que llevo a Stallman a involucrarse con el tema de código libre y más tarde formó la fundación Free Software Foundation (FSF) aquí mismo se creó la definición de “free software” y el concepto de “copyleft”.<sup>21</sup>

En el 2007 el 1 de junio En la Carta Iberoamericana de Gobierno Electrónico, aprobada por la IX Conferencia Iberoamericana de Ministros de la Administración Pública y Reforma del Estado, realizada en Chile, se recomienda el uso de estándares abiertos y Software Libre para desarrollo y explotación de aplicaciones informáticas.

## 2.2.3 Características del Software Libre.

- Su principal característica se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.
- Respeto la libertad de los usuarios sobre su producto adquirido.

---

<sup>21</sup> Wikipedia (2010). Historia del Software Libre. En: [http://www.wikipedia.com/Software\\_libre/wikki.htm](http://www.wikipedia.com/Software_libre/wikki.htm)

- El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios.
- Software libre no quiere decir "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente.
- El "software gratis" incluye en ocasiones el código fuente pero no es libre si no cumple con los principios del software libre.
- Es confundible software libre con "software de dominio público", pero este último no requiere de licencias a diferencia del software libre que posee una licencia GNU.
- El software libre no tiene nada que ver con términos comunes como piratería, regalar o gratis.

#### **2.2.4 Ventajas del software Libre**

- Actualmente la mayoría de aplicaciones ya traen capacidad de ser instaladas y utilizadas en plataformas (Linux, Windows, Mac Os).
- El precio de las aplicaciones es mucho menor, o incluso podrían ser gratuitas.
- Libertad de copia.
- Libertad de modificación y mejora.
- Libertad de uso con cualquier fin.
- Libertad de redistribución.
- Facilidad a la hora de traducir una aplicación en varios idiomas.
- Mayor seguridad y fiabilidad.
- El usuario no depende del autor del software.

## 2.2.5 Desventajas del Software Libre

- Conflicto en formatos de archivos de office de Microsoft si no se tiene la precaución de guardar los archivos con un formato open office se pueden perder datos.
- Mayores costos en lo referente a instalación, migración, o interoperabilidad ya que eso requiere una capacitación en el personal porque el software libre todavía forma parte de "algo nuevo", ello supone afrontar un costo.
- La instalación de la mayoría de aplicaciones bajo Linux pueden llegar a ser complicadas de instalar especialmente para personas que no conocen de sistemas.
- Inexistencia de respaldo por parte del autor, es decir en la mayoría de casos no existe ayuda por parte del fabricante o usos de garantías.
- Interfaces gráficas menos amigables desde los sistemas operativos cuyo manejo comparado con otros sistemas operativos es menos atractivo.
- Poca estabilidad y flexibilidad en el campo de multimedia y juegos.
- Es menos compatible con otras plataformas.

## 2.2.6 El software propietario

El software propietario también llamado privativo, privado, de código cerrado o software no libre, es cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo.

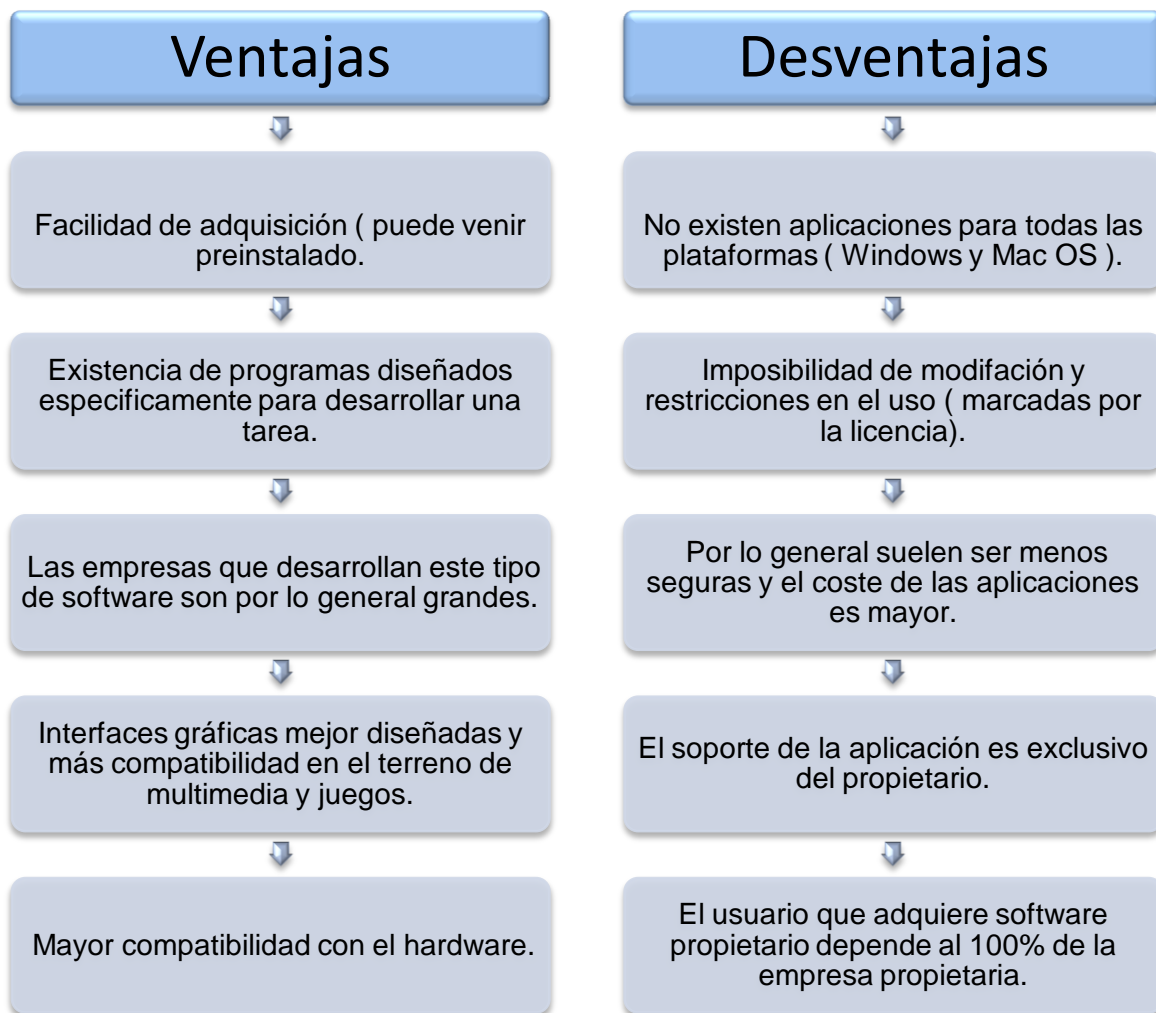
En los años 60 se creó el primer sistema operativo UNIX en los laboratorios Bell cuyo código fuente fue proporcionad para mejoras a grupos científicas que pudieran aportar con ideas. Años más tarde ya se empezó a restringir el acceso con la creación de políticas que reglamentaban esto con lo que dio inicio al código cerrado.<sup>22</sup>

---

<sup>22</sup> Wikipedia (2010). Software Propietario. En: <http://www.wikipedia.com/software-propietario.mht>

En los años 70 se empezó a separar lo que era software del hardware y con esto surgió ya la necesidad de que cada aplicación creada tenga una forma de garantizar los derechos de autoría de sus creadores.

➤ **Ventajas y desventajas**



**Tabla 22 Ventajas y desventajas software propietario.**

**Elaborado: Carlos Landázuri.**



## 2.2.7 Análisis comparativo Software libre vs Software propietario

<b>Costos</b>	<ul style="list-style-type: none"><li>•Software libre tiene un bajo costo de adquisición y libre uso a diferencia del software propietario que tiene un valor mayor sobre todo en licencias</li></ul>
<b>Objetivos</b>	<ul style="list-style-type: none"><li>•La filosofía del software libre, tiene como objetivo principal compartir la información, el software propietario tiene una visión mas bien comercial.</li></ul>
<b>Garantias</b>	<ul style="list-style-type: none"><li>•El software libre no tiene garantía proveniente del autor a diferencia del privativo que ofrece garantía del producto y seguimiento de errores.</li></ul>
<b>Interfaces</b>	<ul style="list-style-type: none"><li>•Las interfaces gráficas de usuario (GUI) y la multimedia apenas se están estabilizando en el SL aunque ahora interfaces como KDE, GNOME son ya lo suficientemente estables para el uso cotidiano pero todavía no se puede decir que se supera la tecnología amigable del software propietario.</li></ul>
<b>Tecnología</b>	<ul style="list-style-type: none"><li>•El usuario debe tener nociones de programación para instalar algún programa pero en software propietario esta función puede ser muy sencilla de realizar además que las tecnologías son mas compatibles con este último.</li></ul>
<b>Hardware</b>	<ul style="list-style-type: none"><li>•El software libre requiere menores requisitos de hardware y menor durabilidad de las soluciones.</li></ul>

Tabla 23 Análisis comparativo Software libre vs Software propietario.

Elaborado Carlos Landázuri.

## ➤ **Libertades del software libre**

El software libre propone las siguientes libertades:

La libertad de usar el programa, con cualquier propósito.

La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.

La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.

La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

**Tabla 24 Cuadro explicativo de las libertades del software libre**

**Elaborado: Carlos Landázuri**

### **2.2.8 Copyright, copyleft y patentes**


Entre los derechos de autor más importantes mencionaremos los siguientes:

#### ➤ **Copyright**

El símbolo del copyright “©”, es usado para indicar que una obra está sujeta al derecho de autor que se refiere al derecho que tiene una persona por la creación de una obra literaria, artística o científica tanto publicada o que todavía no se haya publicado siempre y cuando estos derechos no hayan expirado.

Una de las ventajas de utilizar el Copyright es que los trabajos de muchos artistas están reconocidos y no se pierde en el anonimato y la mayoría de veces obtiene ganancias económicas. En cuanto a los usuarios existe la ventaja de que cualquier reclamo existe un responsable frente al cual se puede establecer una acción legal.

### ➤ **Copyleft**

El símbolo del copyleft es el símbolo del copyright invertido, viendo hacia la izquierda “”Es utilizado como la contrapartida del símbolo del copyright, sin embargo no posee reconocimiento legal.<sup>23</sup>

Se refiere a un grupo de licencias de derechos de autor en lo concerniente al software, la literatura, la música y el arte pero sin restringir el derecho de hacer y redistribuir copias de un trabajo determinado, permitiendo que otros puedan continuar el proceso de ampliar y mejorar su trabajo.

### ➤ **Patentes de software**

La patente de software es el derecho que protege a la propiedad intelectual de un sistema. El software de computadores se compone de dos partes uno es el componente escrito o código y otro los componentes técnicos o algoritmos.

Richard Stallman señala las diferencias entre el copyright y las patentes. El copyright regula las condiciones de expresión de una obra, no protege ninguna idea, las patentes solo protegen las ideas y el uso de las ideas.

El copyright se aplica automáticamente. Las patentes son publicadas por una oficina de patentes de cada país como respuesta a una solicitud.

---

<sup>23</sup>Copyleft. (2006). Software Libre Y Software Propietario. En: [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0121-86972008000200007](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-86972008000200007)

## 2.2.9 Tipos de licencias

Se entiende por seguridad de software a la autorización que da un autor de software a los interesados para ejercer el producto de forma legal. Existen muchas formas de licencias ya que cada una forma parte de un acuerdo entre el autor y el licenciatarario.<sup>24</sup>

### ➤ **Licencias GPL**

Es una de las más utilizadas, es una licencia que especifica que el autor conserva todos los derechos de autor “copyright” es decir que tiene los derechos de redistribuir y modificar bajo los términos ya estipulados pero asegurando que las nuevas versiones modificadas del software permanezcan con los parámetros de GNU y GLP y con esto se asegura que el nuevo producto no sea puesto bajo una licencia GLP. En caso de que el sistema modificado conste de dos partes de códigos diferentes con dos tipos de licencia si la una es GLP el producto final deberá tener una licencia GNU GLP.<sup>25</sup>

### ➤ **Licencias AGPL**

La Licencia Pública General de Affero es un tipo de licencia copyleft que se deriva de GNU, está hecha para asegurar que un sistema cumpla con la comunidad GNU en caso de estar en un servidor de red.

Esta licencia hace que si el software se ejecuta para ofrecer servicios de una red de ordenadores debe tener también una licencia AGPL.

---

<sup>24</sup> Wikipedia (2010). Libertades Del Software Libre. En: [http://www.wikipedia.com/Software\\_libre/wikki.htm](http://www.wikipedia.com/Software_libre/wikki.htm)

<sup>25</sup> GNU (2007) Licencias En El Software Libre. En: <http://www.gnu.org/copyleft/gpl.html>

## ➤ **Licencias estilo BSD**

Esta licencia muy conocida por su posición “aún más libre” que la GPL básicamente una licencia del tipo BSD no tiene Copyleft, por lo que es posible hacer versiones modificadas no libres.

La oposición entre la licencia BSD y la GPL es clara en ese punto: la primera considera que si el software es libre no debe imponer ninguna restricción en su distribución aunque esto signifique que aunque alguien use el software para su propio beneficio y no comparta el código, la GPL ve en esto un problema: el software libre termina favoreciendo a quienes no les importa la libertad de los usuarios.

Las licencias BSD no eran consideradas libres por GNU y en 1999, esta cláusula fue eliminada y a partir de allí la licencia BSD se la puede encontrar en proyectos antiguos. Hoy en día muchos proyectos grandes usan licencias del tipo BSD o inspiradas en su filosofía. Por ejemplo NetBSD usa una licencia BSD original.

## ➤ **Licencia X11**

Es una licencia de software libre simple y permisiva sin Copyleft pero compatible con la GNU GPL. XFree86 usa la misma licencia. A veces se le llama la licencia del “MIT”, pero ese término es engañoso puesto que el MIT ha utilizado muchas licencias para su software. <sup>26</sup>

---

<sup>26</sup> Licencias De Software Libre (2009) [http://www.maestrosweb.com/Licencias libres de Software \(II\) \\_ Maestros del Web.htm](http://www.maestrosweb.com/Licencias%20libres%20de%20Software%20(II)%20_Maestros%20del%20Web.htm)

### ➤ **Licencia Pública de Mozilla (MPL)**

La licencia de la Fundación Mozilla cumple completamente con la definición de software de código abierto de la Open Source Initiative (OSI) y con las cuatro libertades del software libre enunciadas por la Free Software Foundation (FSF). Sin embargo la MPL deja abierto el camino a una posible reutilizar de forma comercial no libre del software, si el usuario así lo desea, sin restringir la reutilización del código ni la liberación bajo la misma licencia.

### ➤ **Licencia CDDL**

El Desarrollo Común y Licencia de Distribución, en español es una licencia Open Source. Es una de las nueve licencias más populares, mundialmente usadas o con fuertes comunidades, siendo Open Solaris el desarrollo más importante que la implementa. La Free Software Foundation afirma que se trata de una licencia libre y que es incompatible con GNU GPL.

### ➤ **Licencia de la Fundación Apache**

Existen tres versiones de la licencia Apache siendo la 2.0 la más empleada esta versión es considerada una licencia de Software Libre. Incorpora ciertas condiciones extra relacionadas con patentes: exige incluir un permiso de uso de patentes por parte del autor poseedor de las patentes y además puede invalidar la licencia por problemas de patentes.

## ➤ **Licencias libres para documentación**

Esta licencia puede consistir en manuales, documentación de código y todo lo que el desarrollador considere importante para usar y modificar su programa.

La cantidad de licencias de documentación libre es significativamente menor que el de licencias de software.

## ➤ **GNU FDL (Free Documentation License).**

La licencia GNU FDL es la más extendida, básicamente su aplicación determina que la obra en cuestión pueda copiarse, modificarse y redistribuirse.

Al igual que la GNU GPL no hace especificaciones sobre el uso comercial y es del tipo Copyleft. La GNU FDL permite definir secciones invariantes dentro del texto, las cuales se deberán preservar sin cambios en las modificaciones y obras subsecuentes. También crea incompatibilidades con otras licencias libres, como las de Creative Commons. Esto es justificado por los defensores de este tipo de licencia por la necesidad de impedir que terceras partes mejoren el documento, y se apropien de él.

## 2.2.10 Comparativa de las diferentes licencias del software libre.

Licencia	Descripción	Compatible con GPL	Certificación OPEN SOURCE
BSD Modificada	Simple, libre, abierta	Si	Si
BSD Original	Permisiva, sin copyleft, con cláusula de advertencia	No	No
GNU Public (GPL)	Libre, abierta, con copyleft	Si	Si
GNU Reducida(LGPL)	GPL sin copyleft, permite enlazar con módulos no libres.	Si	Si
Mozilla Public (MPL)	Libre, copyleft limitado, no enlazable con GPL	No	Si
Netscape Public (NPL)	Como MPL pero puede usar código propietario	No	Si
Apache Software	Libre ya abierta con patentes	No	Si
Common Development and Distribution (CDDL)	Libre sin copyleft con patentes y propiedad intelectual	No	Si
Eiffel Forum (EFL)	Libre y abierta versión 1 no compatible con GPL	V2	Si
Expat	Libre simple, permisiva y sin copyleft	Si	Si
IBM Public	Libre, con patentes	No	Si
Intel Open	Libre ya no se usa	Si	Si
PHP	Libre sin copyleft	No	Si
Open LDAP	Libre, permisiva, sin copyleft	V2	No

**Tabla 24** Tabla comparativa de las diferentes licencias del software.  
Fuente: LICENCIAS DE SOFTWARE.



## 2.2.11 Aplicación práctica de una licencia libre.

Para aplicar concretamente una licencia en el software libre, no todas se aplican exactamente de la misma forma, pero en general el procedimiento es similar. En cada archivo que compone el código fuente de nuestro software deberemos agregar la nota del Copyright, algo como: "Copyright 2012 Carlos Landázuri".

Siempre debemos usar la palabra "Copyright", nunca alguna de sus "traducciones" (como "Derecho de Autor" o "Derecho de Copia"). El símbolo "©" puede estar incluido si así lo deseamos, no es obligatorio, también podríamos usar "(C)".

El año especificado debe ser aquel en el que liberamos dicha versión. A medida que vamos liberando nuevas versiones en los años siguientes, la nota legal deberá hacer referencia a cada uno: "Copyright 2007 2008 2009 Carlos Landázuri".

También debemos agregar en cada archivo fuente una nota estableciendo que está permitida la copia bajo los términos de la GNU GPL. Este es el texto a incluir.

Junto con el código fuente debe incluir una copia de la licencia completa, en nuestro caso la GNU GPL. Este archivo debe ser texto plano y usualmente es nombrado como LICENSE o COPYING. El texto de la licencia debe ser en inglés (las traducciones no son oficiales).

Como lo dicho al comienzo del artículo, no hay ninguna necesidad legal de registrar el software en la entidad de Copyright o Derechos de Autor de su país. La sola distribución hace que su software "obtenga" Copyright. El registro ante la entidad solo cobra sentido ante una confrontación legal o violación de la licencia de su software.

En el caso de la GNU GPL, la FSF nos ofrece nombrarla como titular de nuestro Copyright. De esa forma ellos se encargan de hacer valer la licencia en caso de violación, sobre todo en el contexto legal de los Estados Unidos. Esta posibilidad es muy usada por aquellos desarrolladores que no tienen posibilidades, conocimiento o interés en hacerse cargo de las cuestiones legales de su software, pero quieren hacerlo libre.

## 2.2.12 Software Libre en la Actualidad de América Latina.

En la mayoría de países de América latina sus Constituciones Legislativas no mencionan leyes concisas para el uso y protección del Software libre es así el caso de nuestro país que en el artículo 28 de la constitución vigente menciona que los programas de ordenador están considerados obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador como código fuente ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.<sup>27</sup>

Para corroborar lo anterior el artículo 29.dice que es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo se pruebe lo contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación.

El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo.

Muchos autores reconocen pues que el software no es patentable es decir sujeto a propiedad industrial pero sí le reconocen derecho de autor es decir, sujeto a propiedad intelectual. Este es el caso de la legislación española, pero no de la americana. La única garantía para el software libre garantice su

---

<sup>27</sup>Leyes del Ecuador. (2009)En: <http://www.cibersociedad.net/textos/textos.php>

legalidad es que existan leyes que así lo garanticen pero el software libre no está citado en ninguna constitución.

El software libre es el último eslabón de la cadena: un producto que sólo está sujeto a titularidad y cuya licencia permite el libre uso y distribución. Dada la "excentricidad" de este planteamiento no es de extrañar la falta de legislación

No basta con acceder y utilizar un programa para considerar que la licencia es aceptada. Es necesario el registro para que tal licencia tenga validez, por ello es tan necesario el que el software libre esté perfectamente registrado y con el copyright vigente. Es casi imposible con la ley en la mano perseguir un uso abusivo de un programa GPL. De hecho las infracciones a dicha licencia son resueltas por los usuarios de la red en forma de boicot al infractor lo cual suele ser mucho más efectivo que la actuación legal.<sup>28</sup>

### **¿A cuánto ascienden las pérdidas?**

Una investigación realizada en 2004 por la BSA encontró que en el Ecuador el 70% del software instalado era pirata y el perjuicio económico ascendía a 13 millones de dólares. Las cifras del último estudio de BSA de 2011, muestran un descenso de 3% llegando al 67% de software ilegal; sin embargo, la pérdida económica aumentó a 79 millones de dólares, sólo en Ecuador. Esto se debe a que si bien ahora se instala menos software pirata, también se incrementa el número de computadoras en el país.<sup>29</sup>

---

<sup>28</sup> Instituto de Capacitación Jurídica (2010). Legislación del software libre. En:<http://www.cetid.abogados.ec>

<sup>29</sup> Canal News (2011). Piratería informática en Ecuador. En:<http://www.canalnews.ec/index.php/noticias/it-cifras/458-pirateria-informatica-en-ecuador.html>.

### **2.2.13 Software libre en la actualidad de Estados Unidos**

En Estados Unidos, un tribunal reconoció los derechos de autor de un creador de software gratuito. De esta manera, se admitió que el inventor tenga la facultad de restringir el uso del programa, en caso de que alguien se lo quiera apropiar ilícitamente. Los magistrados estimaron que la ausencia del intercambio de dinero en el otorgamiento de una licencia de uso, no debería negar la existencia de consideraciones económicas.

Un Tribunal Federal de Apelaciones de Estados Unidos, falló a favor del reconocimiento de los derechos de autor del software gratuito. De esta manera, se determinó que los diseñadores de programas informáticos que difunden gratuitamente los códigos programadores de sus invenciones, pueden iniciar reclamos judiciales por violación de los derechos de autor si alguien se apropia indebidamente del material.

Con este pronunciamiento, se intentó determinar el alcance y el control que pueden ejercer los creadores de este tipo de software, una vez distribuida gratuitamente en la llamada comunidad de "fuente abierta", donde los desarrolladores de programas informáticos de código abierto comparten las claves de su software, permitiendo que los usuarios lo cambien o mejoren y luego lo redistribuyan.

El Software libre brinda supuestamente, libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. En base a ello, los problemas surgen cuando alguien hace uso ilícitamente de ese código difundido gratuitamente, incluyéndolo en sus productos con fines de venta sin generar ninguna contribución a los creadores o usuarios del producto.

Por lo que en respuesta a esto, el tribunal determinó que el creador un programa informático disponible para descargas públicas y gratuitas, podía controlar el uso futuro de su trabajo, según las leyes de derechos de autor.

Los magistrados de la Corte de Apelaciones del Circuito Federal en Washington afirmaron que "tradicionalmente, los propietarios de derechos de autor vendían su material registrado a cambio de dinero", pero que la ausencia del intercambio de dinero en el otorgamiento de una licencia de uso en la fuente abierta no debería negar, sin embargo, que no existan consideraciones económicas.

De esta manera, se revocó el pronunciamiento de la instancia anterior, que había denegado el reclamo preliminar de Robert Jacobsen, un aficionado que creó un programa para maquetas de trenes y que estaba disponible para su descarga gratuita. El damnificado, había iniciado el proceso por violación de derechos de autor contra unos desarrolladores de programas comerciales, argumentando que no siguieron los términos de la licencia.

El mismo, alegó que su creación fue violada, a raíz de denunciar que una compañía copió material de su página en Internet y lo incorporó a un paquete de software con fines económicos. Como consecuencia de esto, solicitó una prohibición judicial de uso en contra de la empresa, que elabora un producto de competencia.

En primera instancia, se había rechazado el reclamo, al estimar que la reclamación de derechos de autor no era aplicable, pero que el solicitante podía denunciar ruptura de los términos de contrato. Sin embargo, Jacobsen apeló el pronunciamiento, en base a la consideración de que las posibles compensaciones previstas por la ley federal son mucho mayores que las contempladas por la ley de contratos, por lo que continuó requiriendo que se le

reconozcan sus derechos sobre la obra, reclamos que fueron escuchados en el tribunal de alzada.

El derecho de autor protege la manifestación de ideas expresadas en obras que presenten originalidad o individualidad, y comprende derechos exclusivos de carácter patrimonial y moral. Este tipo de derechos, nace con el acto mismo de la creación, sin necesidad de registración. Sin embargo, en Ecuador, dada las contiendas que se pueden generar sobre la titularidad de la obra, existe el registro en la Dirección Nacional del Derecho de Autor, distinguiendo entre obras publicadas y obras no publicadas, siendo el registro obligatorio para las primeras y voluntario para las segundas.<sup>30</sup>

#### **2.2.14 Software Libre en Ecuador**

El Gobierno Constitucional del Economista Rafael Correa promueve el uso de Software Libre como política de Gobierno.

La Subsecretaría de Informática es responsable de elaborar y ejecutar planes, políticas y reglamentos para el uso de Software Libre en el Ecuador se definen como políticas: la utilización de estándares abiertos, la minimización de compra de licencias propietarias, la contratación de servicios en proyectos informáticos, la reutilización del software y el uso preferencial de programas navegadores como medios de acceso.

En Ecuador El Decreto presidencial 1014 del 10 de abril de 2008, decreta "Establecer como política pública para las Entidades de la Administración Pública Central la utilización de Software Libre en sus sistemas y equipamientos informáticos.

---

<sup>30</sup>Diario Judicial (2009). Artículo demanda sobre Software Libre EEUU. En: [http://www.diariojudicial.com/contenidos/2008/08/20/noticia\\_0007.html](http://www.diariojudicial.com/contenidos/2008/08/20/noticia_0007.html)

## ➤ **Software libre en la administración pública**

Actualmente existen una serie de países como Alemania, Argentina, Brasil, Cuba, Chile, China, Ecuador España, Francia, México y Venezuela en los cuales sus políticas han mostrado apoyo al software libre ya sea con normas que apoyan su libre distribución o también exigiendo su utilización en los sectores públicos.

Existen muchas ventajas que hacen que las naciones actualmente opten por considerar la utilización del software libre como una política de estado por ejemplo:

- Al existir autonomía tecnológica se puede adaptar una aplicación según las necesidades de las distintas dependencias, y todas esas modificaciones deberán realizarse siguiendo los requisitos exigidos por el modelo del Software Libre.
- Se establecen “estándares abiertos”. Esto beneficia la integración de sistemas y el intercambio de información.
- Seguridad: El hecho de hacer públicos los códigos de los programas favorece a la seguridad de los mismos. La seguridad debe basarse en la transparencia y el software privativo, oculta estos aspectos
- El software libre es más fácil de auditar.
- La libertad que genera el software libre los gobiernos lo pueden considerar como una forma de democracia.
- La utilización de software libre puede significar un ahorro en lo referente a licencias.

## 2.2.15      **Seguridades de Software Libre**

### **Introducción.**

La mayoría de usuarios de un sistema informático o de una aplicación ignoran muchas veces factores que tienen que ver con la seguridad informática en especial en el software libre: En un entorno con seguridades los usuarios deberían tener que realizar tareas que pueden resultar tediosas las mismas que ayudan a tener sus equipos seguros como por ejemplo el cambio periódico de contraseñas en lo referente a información personal y otros factores como en el software libre, como la instalación de virus que pueden atacar a su equipo otras formas de ataque a la seguridad pueden ser:

- Acceso a virus troyanos por la descarga y ejecución de ficheros en servidores, en principio, confiables, por parte del usuario y la mayoría de veces puede la instalación pasar desapercibida por los usuarios lo que se conoce como virus troyanos.
- Difusión de virus por correo electrónico lograda gracias a la malversación por parte del virus del programa utilizado porque el usuario activa el virus inadvertidamente creyendo que se trata de otra cosa.
- Explotación de una vulnerabilidad de un servicio que se está ofreciendo a través de Internet. Como por ejemplo un servidor web donde una carpeta compartida con otros miembros de la red local y quizá un virus que haya en sus ordenadores pueden copiar archivos.

Existen diferentes Sistemas Operativos con compromisos de seguridad como Windows 95/98/ y las diferentes versiones de Windows Server los mismos diseñados para ser un Sistemas Operativos con altos niveles de seguridad.

El conjunto de versiones de Linux, que como la mayoría de los UNIX, fue concebido para facilitar al interoperabilidad entre múltiples usuarios y múltiples



máquinas no son sistemas enfocados en la seguridad, por lo que se podría pensar que Linux no es un SO seguro pero estas afirmaciones no son ciertas. Linux "tiene la capacidad de ser extremadamente seguro" además que la mayoría de virus no son enfocados a sus sistemas operativos lo que le dan una ventaja adicional.

### **2.2.16 Seguridad en el software libre vs el software propietario**

No se puede establecer que el software libre no es más seguro que el propietario, ni el propietario lo es más que el software libre ya que no existe un parámetro exacto que mida estos niveles de seguridad. El que un determinado software sea seguro no depende de si se distribuye junto con su código fuente (una de las diferencias entre estos tipos de software). Un software es seguro cuando ha sido bien desarrollado y se utiliza de forma correcta, y esto es independiente de la forma bajo la que se distribuya. Sin embargo, el software libre es más transparente que el software propietario, ya que permite comprobar que fue desarrollado de forma correcta.

Otra de las ventajas del software libre es que está basado en estándares abiertos, es decir cualquier empresa puede crear un programa que maneje la información que genera en ese software. De ese modo no se produce una dependencia tecnológica hacia una determinada empresa. Siempre es el usuario quien elige el programa con el que manejará sus datos, pudiendo cambiar su elección cuando lo desee, ya que la información estará almacenada en formatos estándar, que pueden ser manejados por otros programas diferentes al nuestro. Un ejemplo de esto son los ficheros jpg o png: cada usuario puede utilizar el programa que más le guste para ver las imágenes almacenadas en estos formatos, ya que son formatos públicos.<sup>31</sup>

---

<sup>31</sup> José Ángel de Busto (2007). Seguridad Informática De Software Libre Vs Software Privativo. En: <http://www.kriptopolis.org/node/1435>

## **2.2.17 Modelo de negocio de proyectos de Software Libre**

Cuando se piensa en el desarrollo de software como negocio se piensa en el modo de financiamiento el valor de venta entre muchos otros aspectos, todos estos se ven involucrados en un modelo de negocio de software.

Con la aparición del software libre se creó una nueva oportunidad como un modelo de negocio. Aunque el auge del software tuvo mayor énfasis en buscar la oportunidad de negocio en la venta de licencias y software producido, se pensaría que en el software libre no existe tal capacidad de negocio pero esto se lo enfoca en el servicio mejoramiento y capacitación de las herramientas lo que conlleva un sin número de oportunidades por explotar.

### ➤ **Oportunidad de Negocio como producto**

- Desarrollo de software con mejoramiento adicional, redistribución y servicios posventa.
- Creación de mercado a través de la entrega de software libre con el fin de establecer posición para facturas, ventas y servicios.
- Actualización de hardware.

### ➤ **Oportunidad de negocio como servicio**

- Consultoría y asesoría
- Actualización de sistemas y mejoramiento.
- Capacitación y entrenamiento.
- Soporte instalación e integración.

## ➤ **Venta de software libre**

En el mundo del software libre es difícil cobrar licencias de uso, pero no imposible. En general, no hay nada en las definiciones de software libre que impida que una empresa cree un producto y sólo lo distribuya a quien pague una cierta cantidad. Por ejemplo, un determinado productor podría decidir distribuir su producto con una licencia libre, pero sólo a quien le pague (de forma similar a como se hace en el mundo clásico del software propietario), esto también tendría que ver con el tipo de licencia que el sistema haya sido liberado ya que el propietario de este sistema podría hacer lo mismo con el nuevo sistema mejorado.

## **3 METODOLOGÍA**

### **3.1 Selección de la Metodología Microsoft Solution Framework**

#### **Introducción**

Microsoft Solution Framework (MSF) es una metodología de desarrollo de software que lleva a cabo los planes de acción en el desarrollo teniendo un enfoque sintetizado y claro, orientado a los proyectos tecnológicos, basado en un conjunto de modelos, principios, conceptos y orientaciones. MSF es un modelo de procesos que consiste en un conjunto de ciclos vinculados en distintas interacciones con aprendizaje continuo.

Solamente basándose en una metodología firme y probada como MSF se garantizará el éxito de un proyecto. MSF ha sido probado por muchos años ya que su implementación asegura soluciones prácticas, tangibles, escalables y cumpliendo con el tiempo establecido. MSF es una metodología adaptable a cualquier tipo de proyectos y su utilización es aplicable incluso en el desarrollo de sistemas de software libre.

Gracias a la gran acogida del Internet en el mundo, hoy en día es posible compartir y acceder a información necesaria, es por esto que la metodología propuesta por Microsoft se presenta como una solución viable para el desarrollo de proyectos de software.

#### **3.1.1 MSF Orígenes**

En los inicios de MSF data de los años 90, Microsoft vio la necesidad de encontrar una metodología de desarrollo que sea compatible con sus necesidades es así que comenzó a recopilar las mejores prácticas de procesos de desarrollo de software, no con la intención de establecer una metodología, sino con la idea de tener una colección de prácticas individuales de lo que

funciona, aplicables dentro de determinados contextos es decir que no se busca realizar una metodología nueva sino crear una basada en las experiencias y prácticas más populares de las ya existentes.

Así es como surge Microsoft Solution Framework que no es sólo una metodología en sí, sino el conjunto de prácticas que de acuerdo al contexto de proyecto y a otros factores que intervienen como el tamaño del equipo, frecuencia de entregas, entre otros brinda una herramienta que guía el proceso. De tal manera para cada proyecto se podrán seleccionar aquellas prácticas que realmente agreguen valor al proceso de ahí el concepto de framework.

Cada práctica se compone de una secuencia de actividades para documentación de los procesos que sirven para describir entradas, tareas, verificaciones, validaciones y criterios de salida de la misma forma como se la haría los procesos de forma sistemática es decir en un proyecto se describen los requerimientos, quiénes va realizar qué actividad en cuanto tiempo y los resultados que se van a obtener.

### **3.1.2 Versiones MSF**

Para lograr catalogar a un sistema de software como exitoso Microsoft propone y muestra una guía para obtener un acertado diseño, desarrollo y funcionamiento de un nuevo producto de software.

Este conocimiento es el resultado de las experiencias ganadas dentro de Microsoft con el testimonio de sus desarrolladores y clientes en proyectos de grande desarrollo de software y de prestación de servicios, la experiencia de los grandes consultores de Microsoft y la globalización de la tecnología en los últimos años.

Microsoft en su recolección de las mejores propuestas como metodologías, posee varias versiones de las cuáles 3.0 y 4.0 son las más relevantes y las que más se comercializaron. Su versión 1.0 se introdujo por primera vez por Microsoft en 1993 pero su uso era exclusivo de los desarrolladores de la empresa Microsoft, aquí se sumaron prácticas y métodos “sueños”, de ahí derivaron: Patterns & Practices Group y MOF (Microsoft Operations Framework). En el año de 1995 se creó “Dynamics” que proponía un grupo de reglas para lograr estructurar los proyectos de software. MSF en su V2” fue creada en 1997 era la primera vez que se empezó a utilizar la palabra framework en vez de metodología ya que sus plantillas eran ajustables a las necesidades del cada proyecto. Para la V2.5 ya se estructuro MSF como la recopilación de las mejores técnicas utilizadas para la estructura de los proyectos de software pero cuando se pensó en el lanzamiento comercial de MSF se lo hizo con su versión 3.0

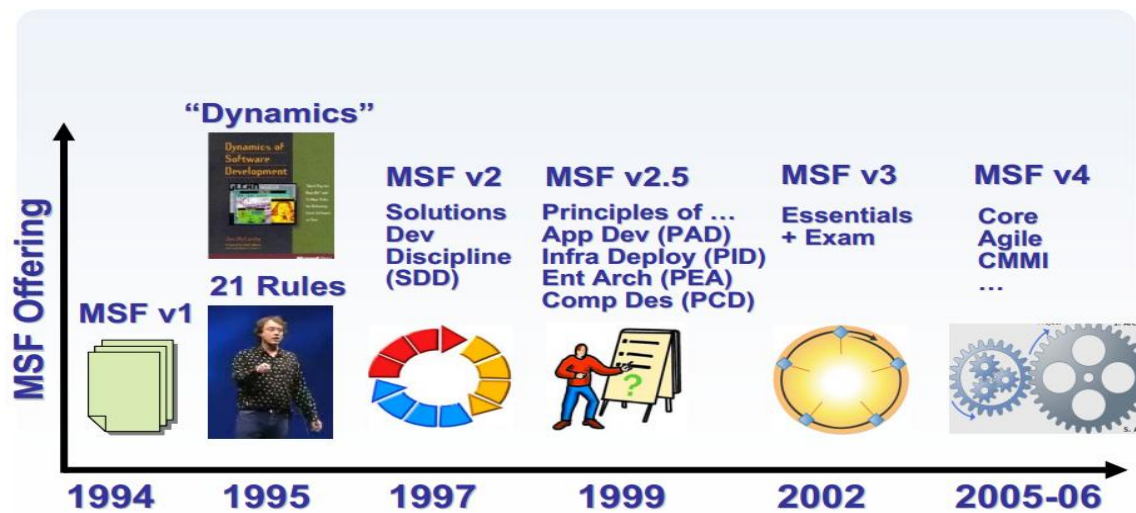


Figura 5 Historia y versiones de MSF  
Fuente: MSF en 60 minutos <sup>32</sup>

<sup>32</sup> Grafico historia MSF

Fuente: <http://www.slideworld.com/slideshows.aspx/Microsoft-Solution-Framework-for-Agile-Software-De-ppt-17113>

### ➤ **MSF versión 3.0**

MSF en su versión 3.0 provee un conjunto de modelos, principios y lineamientos para diseñar y desarrollar soluciones empresariales de manera que todos los elementos de un proyecto (como: la gente, procesos, y herramientas) puedan ser administrados apropiadamente. MSF también provee prácticas probadas para planear, diseñar, desarrollar e implementar soluciones empresariales exitosamente.

Este proceso es flexible y se puede adaptar al diseño y desarrollo de una amplia gama de proyectos de una empresa. Sus principios son aspectos que se debe tener muy en cuenta si lo que queremos es calidad en el modelo y en el producto.

#### Comunicaciones abiertas.

- **Visión compartida:** Un documento de visión en donde todos los miembros tengan un fin común.
- **Fortalecer el equipo:** Capacitación a los miembros, un aspecto que los otros modelos no lo hacen.
- **Responsabilidades:** establecer claramente las responsabilidades personales y las compartidas.
- **Agregar valor:** Dar valor al cliente, al darle productos con funcionalidad, esto quiere decir que MSF es un proceso versionado.
- **Ágil:** Algo muy raro en un proceso prescriptivo, pero de gran ayuda ya que es posible hacer cambios.
- **Calidad:** Como esto cuesta trabajo, se lo ve como una inversión para llegar a la calidad.
- **Aprender experiencias.**

MSF esta subdividido en 2 modelos y 3 disciplinas:

### Modelo de Equipo.

El modelo de equipo es un enfoque desde el punto de vista de la estructura de su personal y las actividades con el fin de asegurar la calidad y el éxito. Existe un grupo de trabajo en MSF que cumplen tareas específicas

Como detallamos:

- ❖ Administración: Jefe de Proyecto.
- ❖ Arquitectura: Arquitecto.
- ❖ Desarrollo: Desarrollador
- ❖ Prueba: Ingeniero de pruebas
- ❖ Operaciones de lanzamiento: Jefe de lanzamiento.
- ❖ Experiencia de Usuario: Analista de negocios.
- ❖ Administración de Productos: Analista de Negocios.

### Modelo de Proceso:

- ❖ Disciplina de proyectos: Un equipo para administrar los principios fundamentales del modelo.
- ❖ Disciplina de riesgos: Equipo para identificar prioridades, tomar decisiones y controlar emergencias.
- ❖ Disciplina de la preparación: Se diagnostica el nivel de conocimiento de los participantes.



## ➤ **MSF versión 4.0**

Utiliza framework descriptivo similar en muchos aspectos a MSF 3.0, pero la gran diferencia es incluye dos metodologías:

MSF para el desarrollo de Aplicaciones Ágiles y MSF para el proceso de mejora CMM. Al igual que la versión anterior define un equipo de trabajo, pero la ventaja es que aumenta la agilidad, presenta dos principios adicionales: una mayor vinculación estrecha con los clientes y que todos los productos sean entregables.

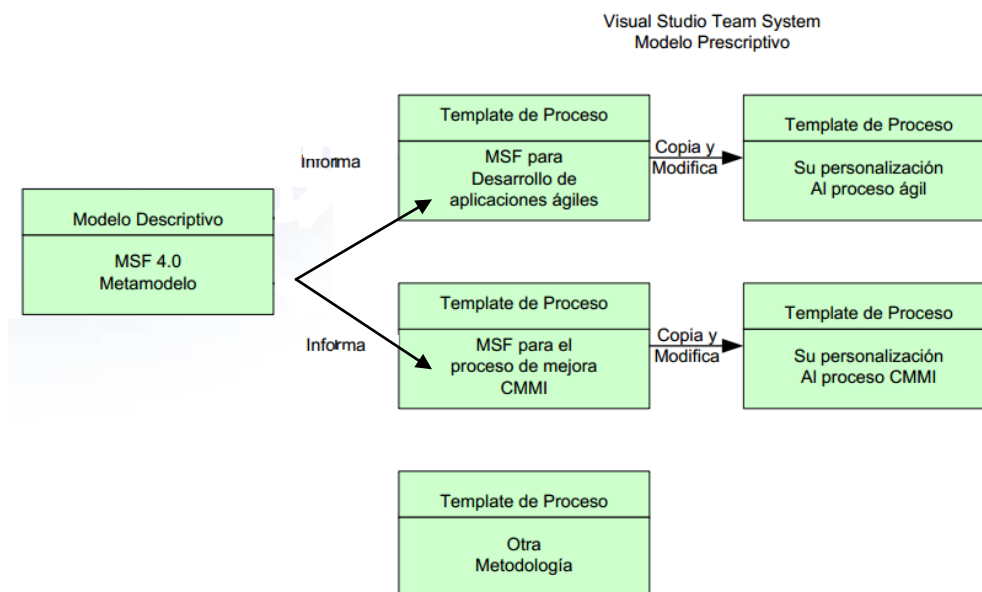
- ❖ Actualmente versión 4.0, aplicada de lleno en VS2005 Team System.
- ❖ MSF V4.0 presentan meta modelos.
- ❖ Contiene plantillas para otros modelos:
  - ✓ Desarrollo de Aplicaciones Ágiles
  - ✓ Desarrollo con proceso de mejora CMMI.

### Ventajas

- La ventaja principal es que al ser un modelo desarrollado por Microsoft se puede tener mayor soporte y mantenimiento.
- Los usuarios finales están más acostumbrados con este producto, por el posicionamiento que Microsoft tiene en el mercado.
- Sirve para grandes y pequeños proyectos.
- Cabe recalcar que MSF no se parece al RUP en algunas definiciones (principalmente en la cuestión de los cambios).
- Brinda un marco de trabajo más compacto y amigable que a diferencia de la 3.0 podría ser un poco más difícil de identificar.

### Desventajas:

- La principal desventaja es que se torna un trabajo bastante largo, ya que para cada fase se debe documentar profundamente todo lo que se haga, pero no deja de ser un modelo que tiene buenos resultados.
- Nos vemos obligados a trabajar en herramientas de Microsoft ya que la herramienta está sujeta al marco de trabajo de las herramientas.



**Tabla 25 Estructura MSF V4.0.**  
Elaborado: Patricia Scalzone<sup>33</sup>

### **3.1.3 Principios Fundamentales.**

MSF aplica los principios propios de esta metodología entre estos podemos mencionar:

- Responsabilidad clara y compartida

<sup>33</sup>Scalzone (2009). MSF. En:  
[http://www.naturastock.com/rsdotnet/iic3140/materia/iic3140\\_04.pdf](http://www.naturastock.com/rsdotnet/iic3140/materia/iic3140_04.pdf)

El Modelo de equipo MSF se basa en brindar guías para implementar cada una de las etapas del desarrollo de un proyecto ya que ninguna persona puede realizar todas las actividades el objetivo es conseguir un sistema de calidad y el éxito de un sistema se garantiza con el trabajo en grupo.

Sin embargo, el cliente necesita una sola fuente de información es por esto que existe un responsable que se encarga de la toma de decisiones y de socializar con el cliente los avances del proyecto. Cada miembro del equipo es responsable del éxito general del proyecto y de la calidad de la solución y se espera que contribuya con ideas y observaciones que se deriven de su conocimiento incluso en áreas en las que no es responsable de una manera personal.

- Los equipos fortalecidos y productivos.

Para que un equipo de trabajo obtenga resultados satisfactorios, se anima a sus miembros a comprometerse para esperar el éxito en cada una de las áreas que se pusieron a su cargo. Igualmente, el cliente tiene la certeza que sus requerimientos se cumplirán en el tiempo establecido sabiendo que cualquier retraso o cambio debe comunicarse lo más pronto posible.

Un equipo de trabajo MSF para mejorar sus resultados debe brindar a sus miembros la confianza que éstos necesitan para cumplir su cometido. Así mismo los jefes del equipo deben tener una función de ayuda hacia el equipo ofreciéndoles al mismo tiempo ayuda y asistencia. La supervisión del progreso se distribuye entre los distintos miembros del grupo y se convierte en una función de ayuda en vez de en una función de control.

### **3.1.4 Las disciplinas de MSF**

Se denominan disciplinas a todas las competencias de las diferentes áreas del conocimiento no tecnológico que son importantes de todas las funciones del modelo MSF. En la actualidad MSF reconoce algunas disciplinas fundamentales que son:

➤ **Planeación seguimiento y control de cambios**

Es la sincronización de las labores a realizar por el equipo de trabajo trazando los procedimientos y los parámetros que se van a utilizar para hacer un seguimiento de los cambios.

➤ **Administración de los ámbitos**

Se establecen los aspectos generales del proyecto la definición, división del ambiente del proyecto.

➤ **Administración de la programación**

Es la generación de programas partiendo de las apreciaciones del equipo a través de secuencias de tareas, adecuación de recursos a las tareas, aplicación de técnicas de estadísticas.

➤ **Administración de costos**

Aquí se establece una estimación de los recursos a utilizar, costos de recursos a utilizar y un informe sobre el progreso de los factores de riesgo de costos.

➤ **Administración de recursos humanos**

Planeamiento de disponibilidad de los actores, motivación del equipo, creación de los grupos de trabajo, solución de conflictos.

➤ **Administración de las comunicaciones**

Se planea la forma de hacer conocer al cliente el avance del proyecto trazando las fechas de reuniones y plazos de entrega.

➤ **Administración de riesgos**

Facilitación y dirección del proceso de administración de los riesgos del equipo mantenimiento de la documentación sobre los riesgos.

➤ **Administración de calidad**

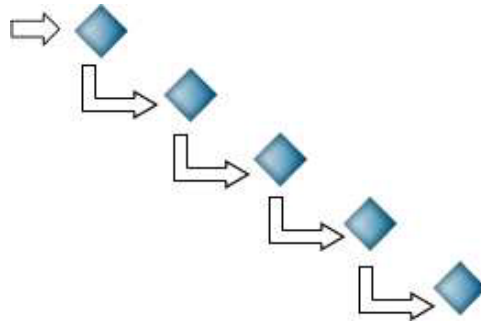
Planeamiento de la calidad, determinación de los estándares que vayan a usarse, documentación de los criterios de calidad o norma a utilizar.

### **3.1.5 MSF Modelos de procesos y aplicaciones**

Microsoft Solution Framework no se basa como la mayoría de metodologías en un mismo modelo, MSF adapta tanto el modelo cascada y el modelo espiral MSF junta las ventajas de estos dos modelos para tener una metodología totalmente práctica.

## ➤ **Modelo Cascada como parámetro para MSF**

El modelo en cascada se representa por el gráfico de la figura.



**Figura 6 Representación gráfica del modelo en cascada.**

**Fuente: Wikipedia**

En la figura 1, se representa cada etapa del proyecto con un rombo. El modelo en cascada como analizamos en el capítulo 1 toma como parámetro inicial de cada interacción la finalización completa de su antecesor, es decir que para comenzar una etapa del proyecto deber estar finalizada por completo.

Así mismo, una vez que se ha pasado a la siguiente interacción no se admite un retroceso a la fase anterior.

Esto funciona para proyectos donde los requerimientos de los usuarios se pueden definir de forma precisa, los mismos que no son modificables del proyecto en una fase inicial. Estas etapas cerradas, facilitan el planeamiento y asignación de recursos para cada etapa. Sin embargo se complica si los requerimientos tomados en un inicio se modifican en cualquier otra etapa.

➤ **Modelo Espiral como parámetro para MSF**



**Figura 7** Representación gráfica del modelo en espiral.  
Fuente: Wikipedia.

El modelo en espiral a diferencia del modelo en cascada no establece actividades claras y definidas dentro del desarrollo del proyecto pero está abierto a cambios en los requerimientos en cualquier fase del proyecto. Es decir se trata de un desarrollo a la que va de la mano con los requerimientos, ya que el cliente provee retroalimentación en cualquier etapa del proyecto.

Esto puede ser muy práctico cuando se necesita un desarrollo rápido en proyectos sumamente pequeñas, pero deja abierta la posibilidad de fracaso en el proyecto por los siguientes aspectos: La primera, es no tener una planeación clara de los recursos necesarios para el desarrollo del proyecto. La segunda, no se puede tener un pleno control en cada una de las etapas de desarrollo aunque se esté trabajando con un equipo pequeño, las funciones de cada actor no están claras.

➤ **Modelo propuesto por MSF.**

MSF propone un modelo novedoso que toma todas las características de los modelos señalados anteriormente tomando sus ventajas y mejorando sus desventajas.



**Figura 8** Modelo propuesto pos MSF.  
Fuente Wikipedia.



Tabla 26 Modelo MSF con los ciclos y las iteraciones.  
Fuente: Carlos Landázuri.

Este modelo interpreta cada rombo como una etapa finalizada con documentación entregable el mismo que puede ser modificada sin que las demás etapas tengan que detenerse. MSF propone un modelo abierto como el espiral que permite volver a las etapas anteriores del proyecto pero para esto ya se tiene puntos de control específicos que permiten tener control del proyecto en desarrollo.

Cada rombo representa una etapa de la metodología visión, planeación, desarrollo, estabilización, implementación. De esta forma se colocan entregables específicos que indican si una etapa está terminada, (aunque puede ser abierta en caso necesario, ya que el modelo permite volver a etapas previas). La siguiente tabla muestra los entregables específicos para cada una de las etapas:



Etapa del proyecto	Documento por entregar.
<b>Estrategia y alcance</b>	Documento de visión y alcance.
<b>Planificación</b>	Documento del plan del proyecto.
<b>Desarrollo</b>	Alcance completado.
<b>Estabilización.</b>	Reléase aprobado.
<b>Implementación.</b>	Proyecto en marcha.

Tabla 27 Fases de MSF y la documentación por entregar.

Elaborado: Carlos Landázuri.

### 3.1.6 Desarrollo ágil de aplicaciones.

Son métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requerimientos y soluciones evolucionan mediante la colaboración de grupo organizado y capaz de hacer varias tareas a la vez.<sup>34</sup>

#### Introducción.

El desarrollo ágil de aplicaciones, es un modelo de procesos cuya finalidad es la de ocupar la menor cantidad de recursos posibles al momento de desarrollar una aplicación, esto no significa por ningún motivo un desarrollo desorganizado, forzado o poco predictivo. Este procedimiento se lo utiliza especialmente en proyectos pequeños o medianos en donde el tiempo es el recurso más valioso.

<sup>34</sup> Wikipedia (2010). Desarrollo ágil de Software. En: [http://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)

Existen dos tipos de modelos de procesos base, el adaptativo y el predictivo. El modelo predictivo intenta predecir al detalle las tareas a ser cumplidas en la planificación total del proyecto. Partiendo de esta idea, planifica al detalle todos los pasos necesarios para el desarrollo del proyecto, desde el principio hasta el final. Sin embargo, esto trae el problema, de que cambios en la dirección del proyecto pueden traer grandes complicaciones, y esto es muy usual en proyectos de desarrollo, especialmente en medianos o pequeños, donde los requerimientos y necesidades se van viendo en el proceso de producción del sistema.

El tipo de modelo adaptativo, es por el contrario muy flexible. Predice una visión global del proyecto, así como los costos del mismo, pero deja abierto cambios en la dirección, debido a su función iterativa.

En este tipo de modelos, se planifica de a poco, y se va sacando nuevas versiones del proyecto por cada funcionalidad nueva agregada. Se trata de segmentar el proyecto en proyectos más pequeños, que igualmente seguirán los pasos de envisionamiento, planificación, desarrollo, pruebas e implementación para concluir con la funcionalidad modular y pasar luego a la siguiente funcionalidad o iteración. La desventaja en proyectos grandes es que este tipo de segmentación e iteración puede ser difícil de controlar, la ventaja es que genera aplicaciones mucho más acordes a los requerimientos del cliente, en un tiempo mucho menor.

### **3.1.7 Casos de éxito.**

#### ➤ **Banco Pichincha.**

El Banco Pichincha, ha venido trabajando con MSF (Microsoft Solution Framework) como marco de trabajo para el desarrollo de los proyectos.

Los proyectos eran administrados de manera individual usando Project Professional y la información generada de los proyectos almacenada en un servidor de archivos, esto representaba varios retos:

- ❖ Al momento de publicar la información.
  - ❖ Para hacer el seguimiento y análisis de desviación del avance por parte los diferentes interesados en los proyectos.
  - ❖ La actualización del cumplimiento de las tareas ejecutadas por parte de los miembros del equipo de trabajo.
- 
- Banco industrial de Venezuela.

Buscando la independencia BIV y Microsoft conformaron un equipo de trabajo para desarrollar un nuevo Internet Banking a la altura de las cambiantes necesidades de la entidad, y acorde con la dinámica del sector financiero.

Para desarrollar el proyecto se utilizó el Modelo de Equipos del marco de trabajo Microsoft Solution Framework (MSF) y Microsoft Operations Framework (MOF), mediante el cual se identificaron los roles principales y sus respectivas responsabilidades en las fases del proyecto, además de contemplar el entrenamiento y soporte post-implantación.<sup>35</sup>

### **3.1.8 MSF y el Software Libre**

Microsoft Solution Framework más que una metodología de desarrollo de software es un enfoque de las mejores prácticas es decir que hasta su versión V3 es un conjunto de documentos con las plantillas o framework “MSF v3

---

<sup>35</sup>Microsoft (2008). Casos De Éxito. En: <http://www.microsoft.com/venezuela/casosdeexito/biv.aspx>

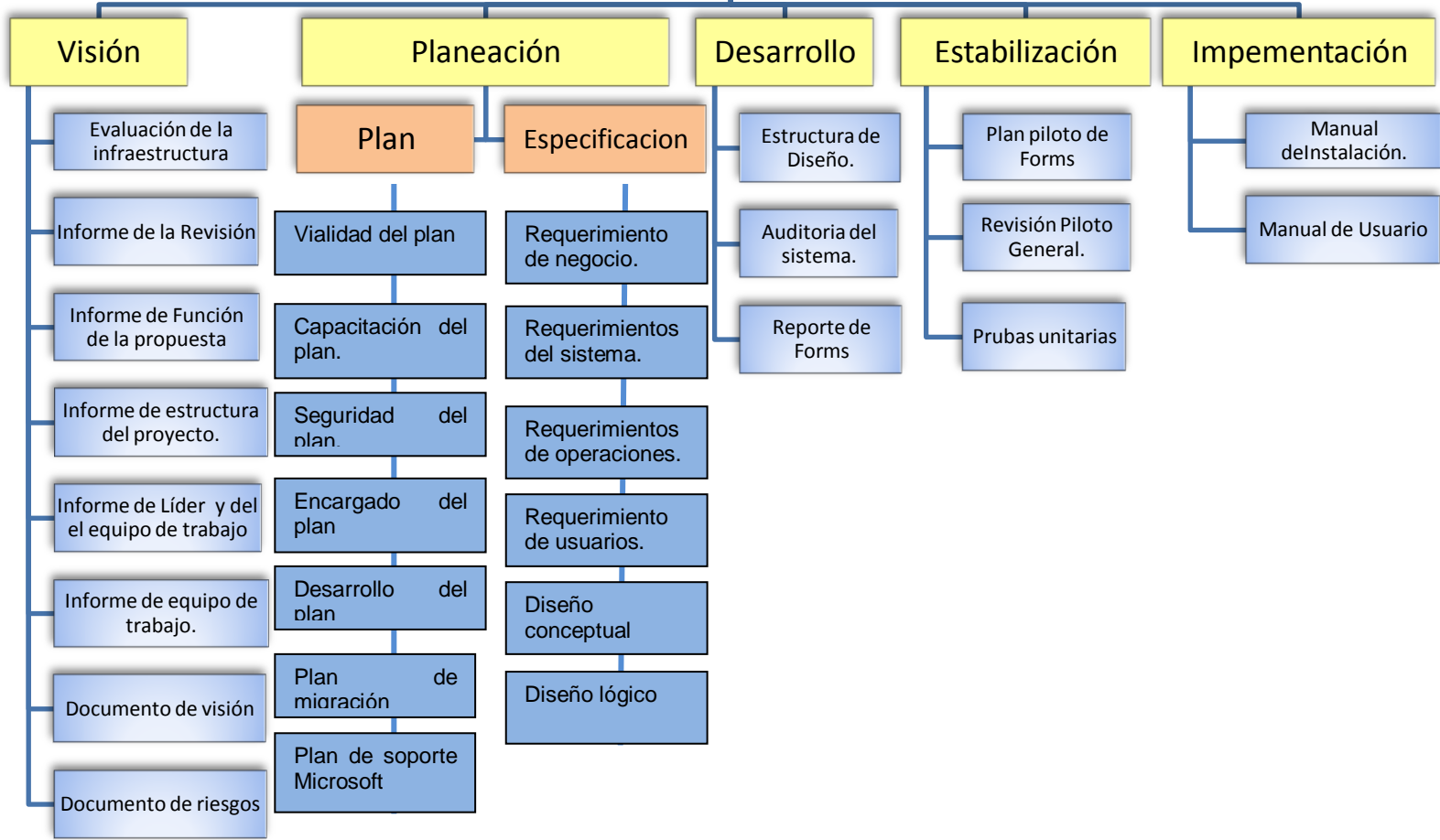
Sample Templates “utilizadas por varios años por el grupo de desarrollo de Microsoft.

MSF hasta su V3 no cuenta con un meta modelo como guía. La utilización de estas plantillas queda a consideración del grupo de trabajo que de acuerdo a sus necesidades utilizan dichos Frameworks como ayuda para documentar cada fase del proyecto.

Es por esto que MSF hasta su versión V3 puede ser utilizada para el desarrollo de cualquier sistema informático incluso sistemas que usen herramientas de software libre.

# MSF V3 Sample Templates

Documentos en la etapa de:



### **3.1.9 MSF y Microsoft Operations Framework**

Microsoft Operation Framework (MOF) ,es un conjunto de conceptos, principios y prácticas cuyo primordial objetivo es elevar al máximo el beneficio del negocio que representa el desarrollo de un nuevo sistema informático , incrementando el nivel de servicios brindados al negocio, optimizando costos y minimizando los riesgos que todo entorno conlleva.

MOF ofrece un marco de referencia que les permite a las organizaciones lograr confiabilidad en los sistemas que ya están en producción y que son de misión crítica, asegurando su disponibilidad y soporte, así como su mantenimiento y administración (Run IT right). Esta herramienta integra los procesos generados por la comunidad, administración, riesgo y cumplimiento de las actividades, revisiones por la dirección a diferencia de Microsoft Solution Framework (MSF) que promueve las mejores prácticas. MOF complementa y se integra en Microsoft Solution Framework (MSF).

MSF es un enfoque por disciplinas para la administración de proyectos tecnológicos basados en prácticas internas de Microsoft, la experiencia del Servicio de soporte técnico de Microsoft con clientes y socios, las prácticas recomendadas del sector para el desarrollo de software y la administración de proyectos. MSF es un enfoque para diseño e implementación de sistemas de TI, mientras que MOF trata la administración diaria de un sistema o un entorno.

La orientación en el Microsoft Operation Framework engloba todas las actividades y procesos que intervienen en la gestión de un servicio de TI: su concepción, desarrollo, operación, mantenimiento, y en última instancia-de su jubilación.

El TI es un conjunto de libros, es una metodología que nos va a ayudar a que las cosas se puedan hacer de una forma más eficiente, ya que lo que propone es que se adopten ciertas métricas y procedimientos que otros proveedores de IT adoptaron y que gracias a ellas son catalogadas como mejores prácticas en la planificación de un proyecto de desarrollo de software.



Figura 9 Ciclos de Microsoft Operation Framework.

Fuente: Microsoft Operations Framework (MOF)<sup>36</sup>

<sup>36</sup> Figura de ciclos de MOF  
Fuente: <http://www.cyquent.ae/Pages/mof.aspx>

### **3.1.10 Modelo Orientado a roles**

El Equipo de Trabajo tiene la finalidad de hacer frente a la planificación y el desarrollo de todo el proyecto involucrando a todo el equipo en las decisiones fundamentales, asegurándose que se cumplan todos los requerimientos del cliente y enfrentar los resultados finales.

#### **➤ Definiciones de Roles**

MSF define los siguientes roles dentro de la estructura organizacional de la metodología:

- **Analista del Negocio**

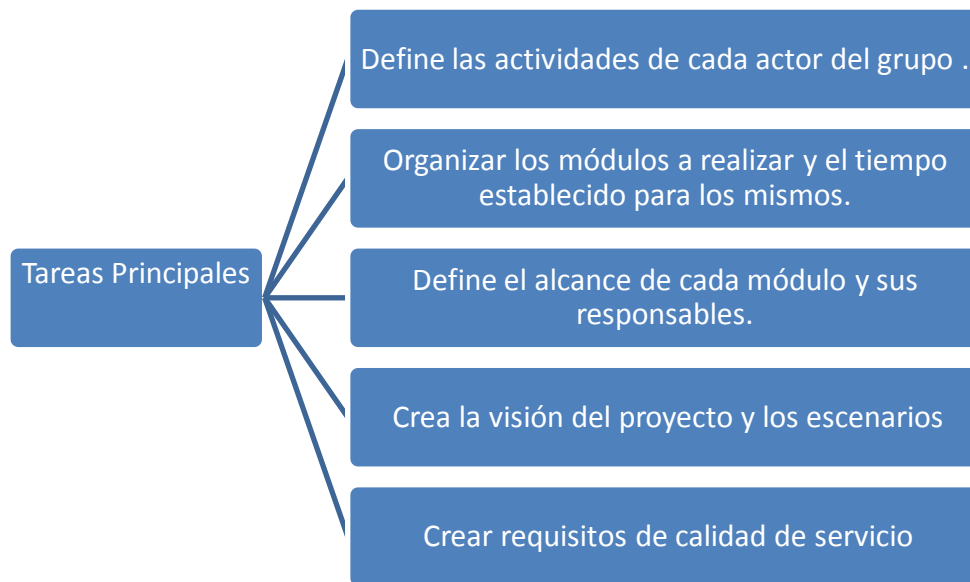
La función principal del Analista del Negocio es receptor, entender y hacer llegar los requerimientos del cliente al resto del equipo, es decir que va a tener una comunicación directa con el cliente o cualquier involucrado que tenga que ver con la visión del proyecto. Luego tendrá que traducir a los actores, escenarios y requerimientos de calidad de servicio, los cuáles van a ser llevados a los desarrolladores para que plasmen en el sistema las necesidades del cliente final.

Este miembro del equipo es también el delegado del manejo de las funcionalidades del sistema y su buen funcionamiento. En lo concerniente al proyecto en si el analista de negocio debe cuidar los intereses del cliente, siendo el que juzgará el proyecto final y el cumplimiento de los requerimientos. En resumen es el intermediario entre el equipo de desarrollo y el usuario final.



- **Administrador del Proyecto**

La finalidad del administrador del proyecto es realizar un cronograma y control del proyecto para que se ajuste al tiempo de entrega y al presupuesto preestablecido tomando en cuenta factores externos y riesgos que pueden intervenir en desarrollo del trabajo. El administrador de proyectos debe cumplir las siguientes funcionalidades:

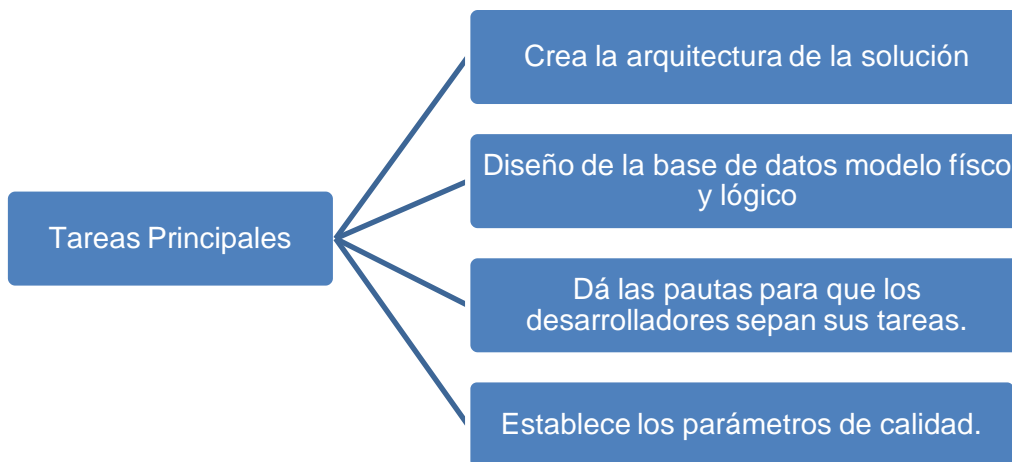


**Tabla 28 Tareas principales del Administrador del proyecto en MSF.**  
**Elaborado: Carlos Landázuri.**

- **Arquitecto**

Su meta es la de garantizar el desarrollo de un proyecto con éxito. Su objetivo principal del arquitecto es diseñar la forma del proyecto de tal manera los desarrolladores se encarguen de una parte específica del proyecto así de esta manera se tiene un total control del proyecto al momento del desarrollo su trabajo es comparable a la realización de los planos de una construcción en el caso de un arquitecto de edificaciones. Su trabajo final es muy importante ya no solo se dan los parámetros para la realización sino que también establece las

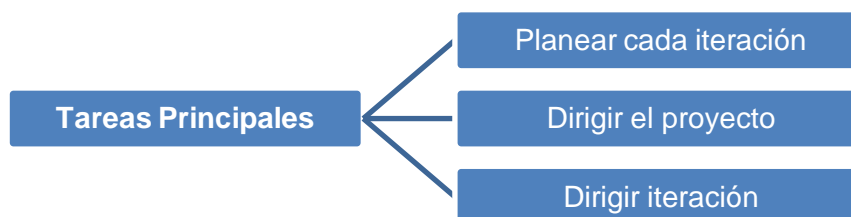
pautas que harán que el proyecto llegue a su realización con éxito. Esto incluye su funcionalidad, mantenimiento y escalabilidad, cumpliendo con los requerimientos. y estableciendo los parámetros de calidad.



**Tabla 29 Tareas principales del Arquitecto de software en MSF.**  
Elaborado: Caros Landázuri.

- **Jefe de Proyecto**

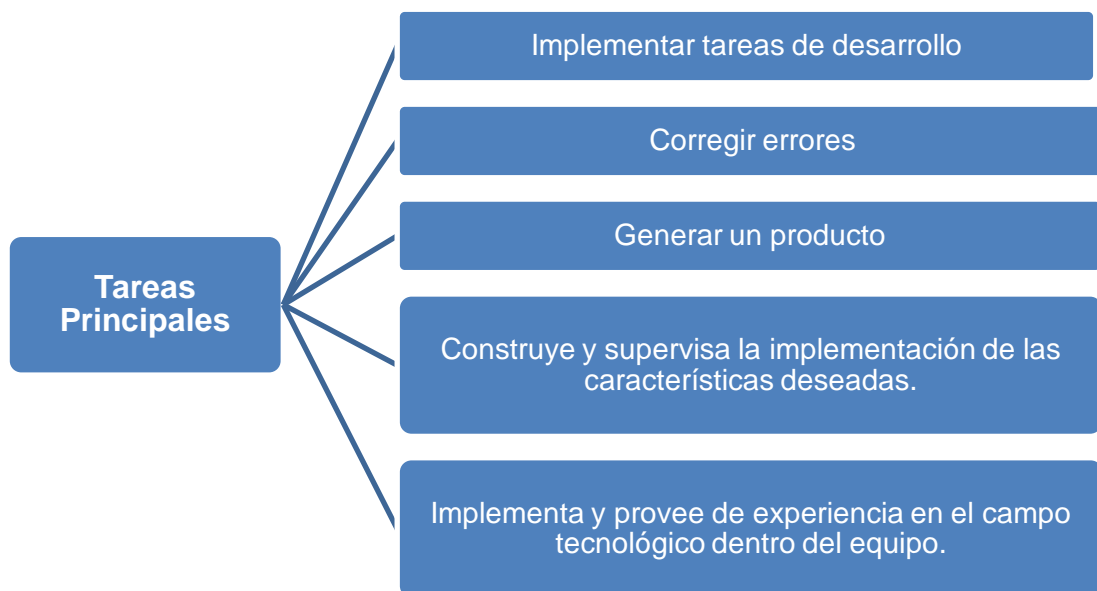
El objetivo principal del jefe de proyecto es entregar un valor de negocio que se ajuste al programa y presupuesto acordados. El jefe proyecto se encarga del planeamiento y programación de tareas, incluido el desarrollo del proyecto y planes de iteración, el control y elaboración de informes sobre el estado del proyecto, y la identificación y mitigación de riesgos.



**Tabla 30 Tareas principales del Jefe del proyecto en MSF.**  
Elaborado: Carlos Landázuri.

- **Desarrollador**

La meta principal del desarrollador es la de implementar la aplicación de acuerdo a la planificación establecida. También se espera que el desarrollador ayude a especificar las características del diseño físico, la estimación de tiempo y esfuerzo para completar cada una de las tareas.

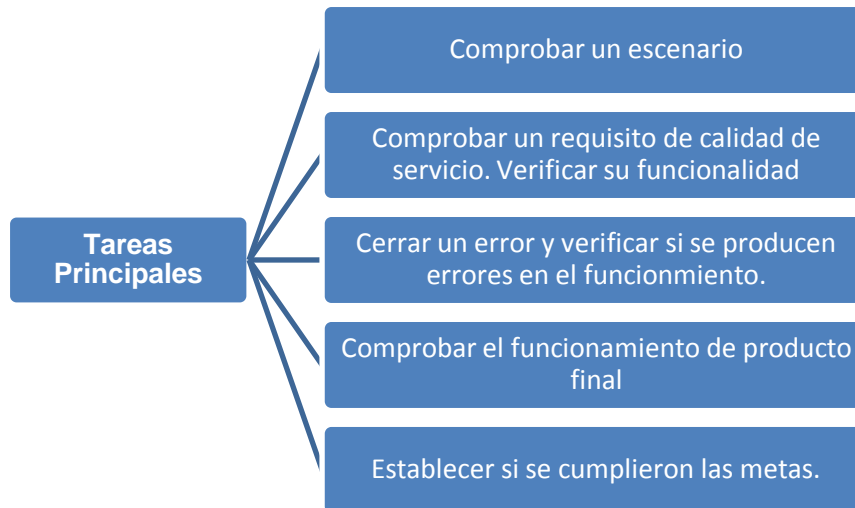


**Tabla 31 Tareas principales del Desarrollador.**  
**Fuente Carlos Landázuri.**

- **Tester**

La meta principal del personal de pruebas es descubrir y comunicar los problemas que podrían influir de forma negativa sobre el valor del producto. El personal de pruebas debe comprender el contexto del proyecto y ayudar a los demás a tomar decisiones estudiadas en función de este contexto. Un objetivo

clave del personal de pruebas es encontrar y comunicar los errores importantes localizados en el producto a través de las pruebas.<sup>37</sup>



**Tabla 32 Tareas principales del Tester en MSF.**  
Elaborado: Carlos Landázuri.

### 3.1.11 Fases de la metodología MSF.

MSF Cuenta en su estructura con las siguientes fases o ciclos:

#### ➤ **Estrategia Visión y Alcance.**

La estrategia y la visión es dar al grupo de trabajo una idea clara de lo que se desea conseguir con el sistema a desarrollar y así mismo al cliente involucrarlo en los aspectos lógicos del negocio mencionando las problemáticas actuales del cliente y exponiendo las soluciones de los mismos.

En esta etapa se consigue trazar los objetivos que el equipo debe cumplir en los lapsos de tiempo establecidos en conformidad con los clientes.

---

<sup>37</sup> Introducción a la Ingeniería del Software (2007). Roles Del Equipo De Trabajo Msf. En: <http://atello334.blogspot.es/tags/MSF/>

- **Objetivos de la Estrategia y Alcance.**

Mediante la visión del proyecto se desea trazar los objetivos principales para los cuales la organización desea satisfacer las necesidades del cliente. En base a esto, se persiguen los siguientes objetivos específicos:

- ❖ Identificar las metas del proyecto.
- ❖ Conocer las expectativas y requerimientos que tiene el cliente para buscar satisfacer sus necesidades.
- ❖ Formar las bases sobre las cuales el equipo desarrollará el proyecto en etapas.
- ❖ Definir el alcance del proyecto, sobre el cual se hará el planeamiento detallado en etapas posteriores
- ❖ Estimar los recursos necesarios para el desarrollo del proyecto.

- **Tareas principales en la fase de visión**

Tarea	Descripción
Creación de los equipos de trabajo	Tomando en cuenta las fortalezas de cada miembro del equipo se realizan los grupos de trabajo.
Realización del cronograma de actividades no definitivo.	Se crea un cronograma de actividades provisional que luego se modificará para planear definitivamente el trabajo de cada etapa de los miembros y las actividades.
Elaboración del documento de visión.	<ul style="list-style-type: none"> <li>• Crear un nombre o referencia para el sistema.</li> <li>• Oportunidad de negocio y visión del proyecto</li> <li>• Se perfilan los actores.</li> <li>• Se definen los recursos a utilizar.</li> <li>• Creación del concepto de la solución donde se muestra la arquitectura del sistema.</li> <li>• Identificación los objetivos del sistema.</li> </ul>

<b>Tarea</b>	<b>Descripción</b>
Creación de los documentos iniciales como casos de uso, escenarios de casos de uso y requisitos.	Modelamiento de una forma general del estado actual del negocio sin llegar a profundizar.
Creación de documentos internos.	Catálogo de actores, catálogo de reglas del negocio y tecnologías posibles a utilizar.

**Tabla 33 Tareas en la fase de Visión en MSF.**  
Elaborado: Carlos Landázuri.

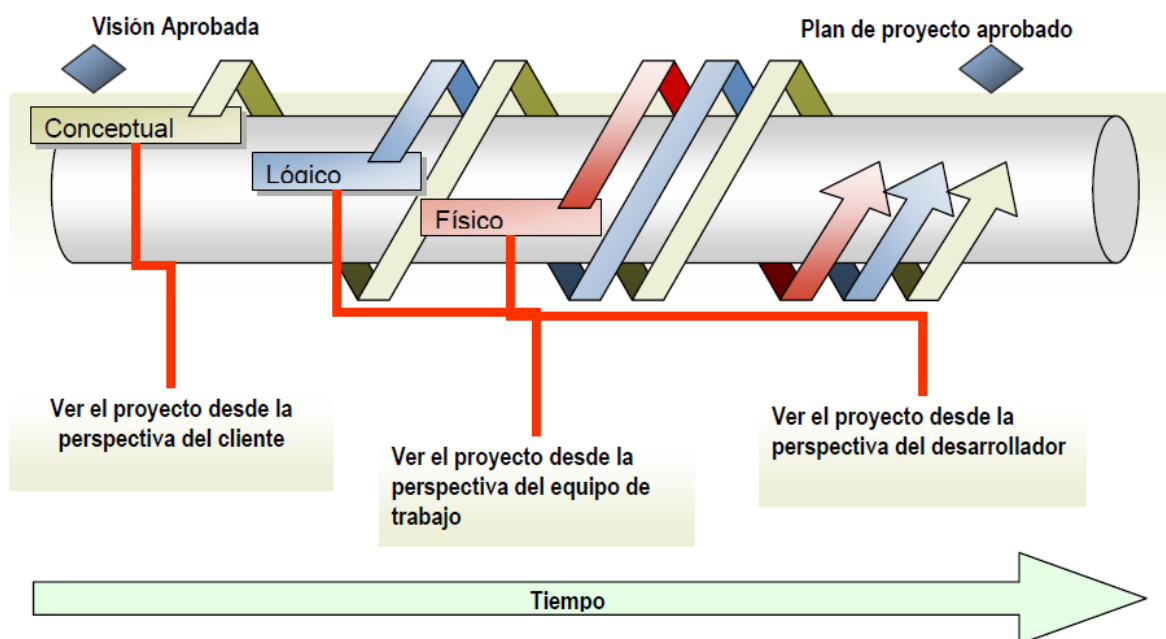
✓ **Entregables de la fase de visión.**

<b>Documento</b>	<b>Descripción</b>
Documento de Visión	Seguir los parámetros de las plantillas del documento de visión, con la posible solución
Documento de estructura del proyecto	Establecer las políticas del proyecto.
Documento inicial de casos de uso, escenarios, reglas etc.	Se redactan a manera general con las finalidad de tener una visión del proyecto ya que es en la planeación donde se pulen estos parámetros
Documentos de análisis de riesgos.	En este documento se destacan los riesgos que pueden poner en peligro al proyecto así como las directivas para disminuir los mismos.

**Tabla 34 Entregables en la fase de Visión en MSF.**  
Elaborado Carlos Landázuri.

➤ **Planificación de un proyecto.**

Esta fase es una fase preliminar antes del desarrollo, la finalidad es aminorar la distancia de que la información necesaria para el desarrollo y la fase de desarrollo. En esta fase quedan más claros los aspectos del proyecto como duración, recursos y características con lo que se mejora la posibilidad de éxito. En la fase de planeación se modela el sistema con diferentes artefactos los cuales van evolucionando desde la fase de diseño físico, no es necesario que una de las fases termine para que comience la otra.



**Figura 10 : Requerimientos y solución de Arquitectura.Net.  
Elaborado: Microsoft Corporation, Curso-Microsoft Oficial –Analizado.**

➤ **Fase de planificación.**

<b>Fases</b>	<b>Concepto</b>
Diseño Conceptual	Solución vista por el cliente
Diseño Lógico	Solución vista desde la perspectiva del trabajo.
Diseño Físico	Solución vista por el desarrollador.

**Tabla 35 : Fases de los procesos de planeación en MSF.  
Elaborado: Carlos Landázuri.**

✓ **Fase de Diseño Conceptual.**

El diseño conceptual es empezar a modelar la solución desde una perspectiva general de tal forma que se pueda discutir con el cliente y los aspectos generales del sistema.

✓ **Objetivos:**

- ❖ Comunicar los requerimientos del sistema.
- ❖ Documentar el problema.
- ❖ Modular la solución.
- ❖ Proveer las bases para planeación del proyecto.
- ❖ Determinar que es lo que se va a entregar.



- **Objetivos del diseño conceptual:**

Es la forma de crear una solución vista desde un panorama general para poder discutir el modelo previo con el cliente. En general los objetivos del diseño conceptual son:

- ❖ Socializar los requerimientos del sistema.
- ❖ Documentar la problemática.
- ❖ Modular una posible solución.
- ❖ Prever las bases para la planeación del proyecto.
- ❖ Establecer que es lo que se va entregar.

- ✓ **Fases del diseño Conceptual**

Fases	Descripción
Investigación	Se responden las dudas del proceso de visión.
Análisis	Se preocupa de corregir los modelos generados inicialmente en el proceso de la Visión, modelos como diagramas de caso de uso, escenarios, diagrama de actividades etc.
Optimización	En esta fase se hace una reingeniería de procesos de negocio tratando de mejorar la productividad.

**Tabla 36: Fases en el diseño conceptual de la etapa de planeación**  
Elaborado: Carlos Landázuri

- ✓ **Documentación necesaria en del diseño conceptual**

- ❖ Diagramas de casos de uso donde se muestran los procesos que va a seguir el sistema y la interacción entre ellos.
- ❖ Escenarios de caso se uso donde se describen textualmente los procesos del sistema.

- ❖ Requerimientos que deben ser bien definidos, factibles, específicos y organizados de forma jerárquica.
- ❖ Diagrama de la arquitectura; un diagrama inicial que se pulirá al final de la planeación.

### ✓ **Fase del diseño lógico.**

En la fase del diseño lógico la documentación de la fase del diseño conceptual, se la desarrolla de una forma más profunda y concreta para esto los documentos, se manejan a nivel interno del equipo de trabajo.

Esta fase el diseño conceptual ayuda a escoger las tecnologías que se utilizarán en la construcción del proyecto en sí pero de una manera independiente del diseño físico.

### Objetivos del diseño lógico.

- ❖ Disminuir la complejidad del sistema a través de modelos de mayor abstracción que los del modelo conceptual.
- ❖ Verificar que los diseños concuerde con los objetivos del proyecto.
- ❖ Facilitar la coordinación entre sistemas.
- ❖ Construir la base para el diseño físico.

### ✓ **Etapas del diseño lógico.**

#### ❖ *Análisis:*

- ✓ Se crea una lista inicial de tecnologías.
- ✓ Se busca los objetos y funciones necesarios para el sistema.
- ✓ Se genera el modelo lógico de objetos.
- ✓ Se bosqueja las interfaces de usuario.

❖ *Optimización:*


Se hace una clasificación del modelo de objetos creados en el análisis buscando que estos sean importantes para lograr discernir lo verdaderamente importante.

✓ **Entregables del diseño lógico.**

- ❖ Lista que se validará en la fase del diseño físico.
- ❖ Modelo dónde se grafica los objetos encontrados con atributos, funcionalidad y relaciones con otros objetos.
- ❖ Diagrama UML donde se evidencia la interrelación entre objetos para cumplir un proceso dado.
- ❖ Tarjetas que muestran la relación entre objetos del sistema.
- ❖ Modelos entidad relación.
- ❖ Bosquejo que validez el diseño físico.

✓ **Diseño Físico**

El diseño físico tiene la funcionalidad de presentar al grupo de trabajo un boceto con la posible forma física que puede tener el sistema, pantallas, formularios, pantallas principales, ventanas a desplegarse etc. Con esto los desarrolladores ya pueden tener una forma mas clara la forma física que el proyecto va a tener.

 **Objetivos del diseño Físico.**

- ❖ Identificar las tecnologías adecuadas para el desarrollo.
- ❖ Transformar los modelos del diseño lógico al diseño físico.
- ❖ Proveer de una línea base para el proceso de desarrollo.
- ❖ Definir cuando el hito de aprobación del plan del proyecto es conseguido.

✓ **Entregables del diseño Físico.**

- ❖ Identificar los requisitos físicos tales como la facilidad de implantación, desempeño, facilidad de uso.
- ❖ Identificar las restricciones físicas tales como la topología de la red, la inversión, seguridades.
- ❖ Resolver conflictos entre requisitos y restricciones.
- ❖ Se crea un modelo preliminar de implementación.
- ❖ Se define los modelos de programación.
- ❖ Se define el modelo físico de las interfaces de usuario.
- ❖ Definir el modelo de las bases de datos.
- ❖ Especificación de técnicas.

➤ **Fase de Planificación.**

Actividades	Descripción
Casos de Uso	Se describe la secuencia de eventos de un actor que usa un sistema para completar un proceso.
Diagrama de Casos de Uso	Muestra las relaciones entre actores y casos de uso del sistema.
Glosario de Términos	Descripción textual de cualquier elemento.
Tecnología a utilizar en el Proyecto.	Enumerar las herramientas que van a ser utilizadas para el desarrollo del sistema.
Diagrama de Secuencia.	Muestran una intersección ordenada de la secuencia de eventos.

Actividades	Descripción
Diagrama de Clases	Muestra la especificación para las clases de Software de aplicación.
Diagrama de bases de datos.	Representación gráfica de las tablas y sus relaciones
Arquitectura de Aplicaciones.	Muestra una descripción de las diferentes capas que serán implementadas para las elaboraciones de la aplicación.
Diagrama de Componentes	Representa un bloque de construcción al modelar aspectos físicos

**Tabla 37 Tareas de la fase de planificación de MSF.**

**Elaborado: Carlos Landázuri.**

➤ **Desarrollo del proyecto.**

En la etapa de desarrollo los programadores utilizan los documentos generados durante todo el proceso y lo traducen a código necesario para crear un producto final que sirva al cliente, siguiendo las especificaciones trazadas en los requerimientos, en la planeación y las pautas el arquitecto del proyecto.

✓ **Guía de proyecto**

En esta fase se entrelazan dos parámetros muy importantes por un lado las especificaciones de requerimientos y el otro el cronograma de la planeación

para lo cual se establecen actividades para cada desarrollador de acuerdo cronograma y al tiempo final de entrega.

➤ **Estabilización del proyecto.**

Esta fase es dónde se pone a prueba el proyecto en ambientes que simulen las actividades comerciales de los negocios de papelería lo más real posible para que de esta manera se corrijan los errores de la aplicación.

✓ **Actividades:**

- En esta etapa del proyecto encontramos la convergencia de errores que no es mas que un punto en el proceso de la resolución de errores dónde se analiza que los errores encontrados a medida que avanza el proceso son menores cada vez.
- La liberación de la versión cero errores, se desarrolla una vez que ya no se encuentren fallas posibles, esto no quiere decir que luego aparezcan otros pero ya el número va a disminuir considerablemente.
- Una vez que el equipo de desarrollo no encuentre errores, el arquitecto analiza el sistema en general y junto con el administrador lanzan una versión beta para ser presentada.
- Con la conclusión del prototipo, se procede a probar el sistema en un ambiente real para analizar los resultados arrojados.

✓ **Documentación a entregar en la fase de Estabilización.**

Documento	Descripción
Documentos de los pilotos	Son reportes sobre cómo ha sido el análisis de las primeras versiones del proyecto.
Versiones para la liberación del proyecto	Son versiones que se entregarán con el código fuente, los ejecutables y scripts de instalación y documentación, además del manual de usuario
Reporte de pruebas y errores	Reportes que ayudarán a la evaluación del proyecto.
Documentación del proyecto	Toda la documentación de cambios del proyecto con un resumen general de cambios.

**Tabla 38 Documentación a entregar en la fase de estabilización.**  
Fuente: Carlos Landázuri.

➤ **Implementación o despliegue del proyecto.**

Al igual que para la fase anterior, en esta fases tampoco se proponen artefactos ya que éstas abarcan las pruebas y la implantación de la solución; es decir, el lanzamiento completo de la solución, aquí se preocupa más de la realización de los manuales de usuario y de la implantación del sistema en el sitio especificado para lo cual se debe preparar el ambiente, se hace la instalación propiamente dicha y se cerciora del funcionamiento adecuado.

✓ **Documentación a entregar en la etapa de Implementación.**

Documento	Descripción
Sistema de Soporte y de evaluación	Más conocido como guía de usuario o manuales de usuario para tener una base de los conocimientos del sistema.
Repositorio de todas las versiones de los documentos y configuraciones	Documentos cuyo objeto es la evaluación
Reporte de cierre de proyecto	Incluyendo una evaluación del cliente que permita cerrar las actividades del proyecto.

**Tabla 39 Documentación a entregar en la etapa de implementación**  
Referencia: Carlos Landázuri



## 4 APLICACIÓN DE LA METODOLOGÍA MICROSOFT SOLUTION FRAMEWORK 3.0 AL SISTEMA DE FACTURACIÓN E INVENTARIOS PARA LA UNIÓN DE PAPELERÍAS DE LA CIUDAD DE IBARRA

### 4.1 Estrategia visión y alcance

#### 4.1.1 Documentos a entregar en la fase de estrategia visión y alcance

Documento a entregar	Persona responsable
Especificación de roles	Carlos Landázuri
Requerimiento de servicio de calidad	Carlos Landázuri
Priorización y lista de riesgos	Carlos Landázuri
Visión del proyecto	Carlos Landázuri
Alcance del proyecto	Carlos Landázuri
Diseño Conceptual	Carlos Landázuri
Definición de arquitectura.	Carlos Landázuri
Diagrama de clases y base de base de datos.	Carlos Landázuri
Documento de Diseño de Interfaz.	Carlos Landázuri
Documento de Pruebas Unitarias	Carlos Landázuri
Implementación, manual de instalación y usuarios.	Carlos Landázuri

**Tabla 40 Documentación a entregar en la Fase de estrategia, visión y alcance.  
Fuente: Carlos Landázuri.**

## **CONTROL PAPER STORE**

Especificación de roles

Autor	Carlos Landázuri
Cargo	Director del Proyecto
Fecha	02-10-2011

## Especificación de roles.

La metodología MSF (Microsoft Solution Framework) adoptada establece los siguientes roles para el equipo de trabajo en el desarrollo de sus proyectos.



**Figura 11 Gráfico de especificación de roles en la fase de visión.  
Elaborado: Carlos Landázuri.**

## Construcción del equipo de desarrollo.

Rol	Responsabilidad
<ul style="list-style-type: none"> <li>Administrador del proyecto</li> </ul>	<p>La función de este rol es la conducción del proyecto, es decir preocuparse del cumplimiento de las especificaciones que se definan y de la entrega a tiempo de los resultados esperados. Definir la arquitectura, asegurar los procesos y los servicios administrativos. Adicionalmente, este rol es responsable de definir las especificaciones funcionales de los servicios a implementar.</p>
<ul style="list-style-type: none"> <li>Desarrollador</li> </ul>	<p>Este rol debe llevar al terreno práctico las especificaciones funcionales, es decir, debe realizar la implantación, desarrollo y completar las funcionalidades del producto tanto en arquitectura como en diseño.</p>
<ul style="list-style-type: none"> <li>Analista de Negocio</li> </ul>	<p>La función principal del Analista del Negocio en la creación de "Control Paper Store " es de receptor entender y hacer llegar los requerimientos de papelería Papelandia y por ende las necesidades de todos los locales que conforman la Unión de papelerías de la ciudad de Ibarra al resto del equipo, es decir que va a tener una comunicación directa con la Sra. Alicia Ortiz y los usuarios de "Papelandia "que van a utilizar el sistema</p> <p>Luego tendrá que traducir a los actores, escenarios y requerimientos de calidad de servicio los cuáles van a ser llevados a los desarrolladores para que plasmen en el sistema las necesidades del cliente final.</p>
<ul style="list-style-type: none"> <li>Tester</li> </ul>	<p>Este rol se encarga de validar la calidad y el correcto funcionamiento de los servicios entregados y su documentación. Este rol comienza su operación en el momento en que comienza el proceso de desarrollo.</p>

Rol	Responsabilidad
<ul style="list-style-type: none"> <li>Arquitecto</li> </ul>	Su meta es la de garantizar el desarrollo de “Control Paper Store” con éxito. Su objetivo principal es de diseñar la forma del proyecto de acuerdo a los requerimientos obtenidos por el Analista del Negocio de tal manera los desarrolladores se encarguen de una parte especifica del proyecto así de esta manera se tiene un total control del proyecto al momento del desarrollo su trabajo es comparable a la realización de los planos de una construcción en el caso de un arquitecto de edificaciones.

**Tabla 41 Construcción del equipo de desarrollo.**  
**Elaborado: Carlos Landázuri**

### **Responsables del proyecto**

A continuación se asignan las personas a cada uno de los roles definidos en la metodología, y adaptados a las necesidades del proyecto en particular. En algunos casos se asigna más de un rol a la misma persona, principalmente debido a la magnitud del proyecto.

El equipo de trabajo y la asignación de roles quedó constituido de acuerdo al siguiente cuadro:

Rol	Responsable
Administrador del proyecto	Carlos Landázuri
Desarrollador	Carlos Landázuri
Analista del Negocio	Carlos Landázuri
Test	Carlos Landázuri
Arquitecto	Carlos Landázuri

**Tabla 42 Tabla descriptiva de los responsables del proyecto.**  
**Fuente: Carlos Landázuri**

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
02-10-2011	Carlos Landázuri	1.0	Documento de especificación de roles
06-10-2011	Carlos Landázuri	1.1	Documento de especificación de roles

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.0	04-10-2011
Carlos Landázuri	1.1	12-10-2011

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Especificación de roles
Autor	Carlos Landázuri
Fecha de creación.	04-10-2011
Última Actualización.	12-10-2011

Nombre	Posición

## **CONTROL PAPER STORE**

Requerimientos del servicio de calidad

Autor	Carlos Landázuri
Cargo	Analista del Negocio
Fecha	07-11-2011

Versión 1.0

## **Requerimientos del servicio de calidad**

Los requerimientos de servicio son:

- Acceso a al sistema por usuarios con una contraseña propia asignada por el usuario administrador, así mismo los permisos y los niveles de acceso a cada usuario nuevo.
- Registro de la mercadería existente en el local como un estado inicial del sistema.
- Ingreso de la mercadería que se compra producto a producto con su referencia de factura.
- Que se registre la mercadería en forma de cajas o unidades y se pueda hacer la transformación (Manejo de bodegas).
- Visualización de los productos y sus movimientos de compra y venta en o lo que se denomina un Kardex
- Permite la creación de facturas con la información de los clientes y el detalle del producto, precio unitario, precio con IVA subtotal y el precio final.
- Registro de los clientes y proveedores con la información principal de cada uno.
- Registro de cuentas por pagar y cuentas por pagar con la fecha de pago, la forma de pago y fechas límites de pagos y de cobros.
- Reportes de compras, ventas, Kardex, usuarios, facturas, proveedores, usuarios.



## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
08-11-2011	Carlos Landázuri	1.0	Registro de la mercadería existente
14-11-2011	Carlos Landázuri	1.1	Acceso al sistema con niveles de usuario
26-02-2013	Carlos Landázuri	1.2	Manejo de bodegas

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.1	15-11-2011
Carlos Landázuri	1.2	28-02-2013

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Requerimiento del servicio de calidad
Autor	Carlos Landázuri
Fecha de creación.	08-11-2011
Última Actualización.	28-02-2013

Nombre	Posición

## **CONTROL PAPER STORE**

Priorización y Lista de riesgos

Autor	Carlos Landázuri
Cargo	Administrador del Proyecto
Fecha	21-11-2011

## Priorización y Lista de riesgos

### Descripción de los riesgos

ID.		Prioridad	Consecuencia	Mitigación
R1	Puede suceder que las aplicaciones resulten complicadas de utilizar para el personal de papelería Papelandia	Riesgo del negocio	Rediseño de nuevos diseños que se acoplen a las necesidades.	Recibir información específica de los requerimientos.
R2	Retraso en cronograma de actividades y costos debidas a requisitos de software	Riesgo del proyecto.	Retrasos en tiempos de entrega estipulados inicialmente.	Capacitación en las herramientas de desarrollo.
R3	Retraso en cronograma de actividades y costos debidas a requisitos de hardware	Riesgo del proyecto.	Retrasos en tiempos de entrega estipulados inicialmente.	Verificación de los equipos existentes y necesarios para cumplir con el proyecto.
R4	Requerimientos especificados no estén de acuerdo a las operaciones comerciales de papelería Papelandia	Riesgo del proyecto	Retraso en la entrega del proyecto	Flujogramas con especificaciones de movimientos comerciales.
R5	Conflictos comerciales entre la asociación de unión de papelerías de la ciudad Ibarra.	Riesgo del proyecto	Confusión sobre los responsables de la asociación.	Socialización sobre el software en cada una de los locales que conforman UPI.
R6	Fallas encontradas por los usuarios originadas en la programación del proyecto	Riesgo de tecnología	Entrega incompleta del proyecto	Trazar dentro del cronograma un tiempo de prueba para el usuario.
R7	Falta de uso del sistema por parte de los usuarios	Riesgo del Negocio	Inoperatividad del sistema	Capacitación a todos los usuarios

**Tabla 43 Descripción de riesgos del proyecto.  
Elaborado: Carlos Landázuri.**

## Análisis de Riesgos.

ID	Probabilidad	Impacto	Prioridad	Exposición al riesgo
R4	Porcentaje 60 % Valor 3	Medio Valor 3	1	9
R6	Porcentaje 40 % Valor 2	Medio Valor 2	2	4
R1	Porcentaje 30 % Valor 1	Medio Valor 2	2	2
R3	Porcentaje 30 % Valor 1	Bajo Valor 1	2	1
R2	Porcentaje 20 % Valor 1	Bajo Valor 1	2	1
R5	Porcentaje 40 % Valor 1	Bajo Valor 1	1	1
R7	Porcentaje 20 % Valor 1	Bajo Valor 1	1	1

**Tabla 44 Análisis de riesgos del proyecto.**  
Elaborado: Carlos Landázuri.

Rango de Probabilidades	Descripción	Valor
1% - 33%	Baja	1
34%- 67%	Media	2
68%-99%	Alta	3

**Tabla 45 Rango de probabilidades con su porcentaje.**  
Elaborado: Carlos Landázuri.

El impacto de riesgo ha sido valorado en función de aspectos como retrasos en la entrega del producto e impacto técnico de acuerdo a los siguientes parámetros:

Impacto	Retraso	Impacto Técnico	Valor
Bajo	1 semana	Ligero efecto en el desarrollo del proyecto	1
Moderado	2 semanas	Moderado efecto en el desarrollo del proyecto	2
Alto	1 mes	Severo efecto en el desarrollo del proyecto	3
Crítico	Mas de un mes	Proyecto no puede ser culminado	4

**Tabla 46 Impacto de riesgos con su valoración.**  
Elaborado: Carlos Landázuri.

La exposición al riesgo ha sido determinada multiplicando la probabilidad del riesgo y el impacto del mismo y se ha categorizado de la siguiente manera:

Exposición al riesgo	Valor	Color
Baja	1 o 2	Amarelo
Media	3 o 4	Verde
Alta	Mayor a 6	Rojo

**Tabla 47 Tabla de exposición de riesgo del proyecto por valor.**  
Elaborado Carlos Landázuri.

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
18-11-2011	Carlos Landázuri	1.2	Riesgos por motivos organizacionales.

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.2	19-11-2011

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Requerimiento del servicio de calidad
Autor	Carlos Landázuri
Fecha de creación.	18-11-2011
Última Actualización.	18-11-2011

Nombre	Posición

## **CONTROL PAPER STORE**

Visión de "Control Paper Store".

Autor	Carlos Landázuri
Cargo	Administrador del Proyecto
Fecha	21-11-2011

### **4.1.2 Visión de “Control Paper Store”**

El objetivo que se persigue Control Paper Store, es dotar al negocio de papelerías con de una solución para el usuario final, para agilizar y automatizar las actividades de su gestión interna, con una herramienta que entregue todas las facilidades necesarias para disminuir los tiempos en sus actividades comerciales y teniendo un pleno control de la mercadería que se compra y la que se vende.

#### **Introducción**

Los locales dedicados actividades comerciales son un ente importante para la economía local del norte del país y más en ciudades como Ibarra que gozan de un prestigio comercial y Turístico

En la ciudad de Ibarra existen aproximadamente 30 papelerías que funcionan como pequeños negocios que se dedican a la venta de útiles escolares y de oficina además de artículos relacionados a la rama. Son muchas familias que dependen de esta actividad en especial en época de regreso a clases donde cada negocio individualmente realiza compras aproximadas a un mínimo de 10000 dólares y solo en esa época obtiene ingresos de aproximadamente 14 000 dólares. Fuente: Papelería “Papelandia”.

La propietaria de la papelería “Papelandia” que forma parte de la Unión de Papelerías de la ciudad de Ibarra, se dedica a la compra y venta de suministros escolares, de oficinas. Adquiere la mercadería de distribuidores que entregan facturas, las mismas que se archivan en lugares físicos. Las ventas se realizan en forma manual y se entrega facturas a los clientes con el detalle de compra.

Los módulos a considerar en este proyecto son los módulos de facturación y control de inventarios para la Papelería “Papelandia” de la ciudad de Ibarra.



## **Enunciado del Problema**

### **Necesidad**

Se necesita evaluar los procesos que siguen los administradores de la papelería “Papelandia” al momento de comprar mercadería, registrar la mercadería que llega, el número de factura y el valor de la compra. Además de los pasos que se realizan al momento de realizar la venta, verificar la existencia del producto, el precio y emitir una factura de la compra.

### **Problema y Motivación**

En la ciudad de Ibarra la mayoría de locales comerciales no manejan sistemas de facturación ni llevan un inventario sistematizado de sus productos este es el caso de Papelería Papelandia que forma parte de la Unión de papelerías de Ibarra la cual la conforman 10 locales de papelerías y solo en una se maneja sistema de facturación e inventarios, cuando analizamos el por qué de esto algunos dueños de los locales saben manifestar que la utilización de sistemas comerciales les parece complicada para su utilización pero sobre todo el costo de la adquisición de un programa no está a su alcance.

Las dificultades actuales de Papelería Papelandia., son principalmente que no cuentan con un sistema que simplifique todas sus actividades comerciales además que organice la información clave para la toma de decisión en gestión comercial, esto provoca excesiva pérdida de tiempo y errores en los procedimientos además de:

- ✓ Carencia de herramienta de análisis y visualización de información en las transacciones diarias
- ✓ Poco control de la mercadería existente y la mercadería que se necesita.
- ✓ Equivocación en los pedidos.
- ✓ Falta de conocimiento en reportes de ventas diarias, mensuales y anuales.

- ✓ Priorización en tareas que un sistema informático podría resumir en segundos.
- ✓ Descontento en los clientes que deben esperar demasiado tiempo por su factura.
- ✓ Errores de cálculo y de sintaxis al momento de realizar las facturas.

Frente al contexto referenciado, se analiza la integración de un sistema que ayude a la planificación y desarrollo del negocio, apoyando el logro de los nuevos desafíos comerciales.

### Oportunidad de negocio.

Luego de que sistema de facturación e inventarios se ha implementado, los negocios de papelerías podrán tener una forma diferente de realizar sus operaciones optimizando tiempo y recursos ya que contarán con pleno control de la mercadería con la que cuentan, los clientes que acuden hacer sus compras, los proveedores de cada producto, las cuentas que están por pagar y si es el caso las deudas de los clientes con cada negocio. De esta forma se sistematizará las operaciones mercantiles ahorrando tiempo y dinero a los propietarios.

### **Meta**

Proporcionar una herramienta estable rápida y segura que permita sistematizar las actividades comerciales de papelerías en la ciudad de Ibarra.

## **Modelo de Ventas**

- Servicios de soporte que permitan asegurar la continuidad operacional y evolución de la solución desarrollada.
- Desarrollo de un conjunto de reportes o vistas sobre los datos para dar respuesta a los requerimientos de los usuarios.

Este corresponde a un “Modelo de Gestión”, orientado al área de ventas permitiendo el análisis encaminado a los clientes de PAPELANDIA, vista desde diversas perspectivas, como son proveedores, clientes, productos, vendedor, etc.

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
25-11-2011	Carlos Landázuri	1.2	Meta
28-11-2011	Carlos Landázuri	1.3	Meta

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.3	30-11-2011

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Visión de "Control Paper Store".
Autor	Carlos Landázuri
Fecha de creación.	21-11-2011
Última Actualización.	28-11-2011

Nombre	Posición

## **CONTROL PAPER STORE**

Alcance del proyecto

Autor	Carlos Landázuri
Cargo	Administrador del Proyecto, Arquitecto, Analista del negocio
Fecha	05-12-2011

Versión 1.0

### 4.1.3 Alcance de Control Paper Store

Control Paper Store nace con la necesidad de aumentar la productividad en las papelerías de la ciudad de Ibarra específicamente los locales que forman parte de la Unión de Papelerías, ya que va a facilitar las operaciones comerciales y el tiempo de generación de información, y así contar con más holgura para dedicar al análisis del negocio.

Nuestro proyecto propone utilizar herramientas de software libre para desarrollar la aplicación, y complementarlas con la plataforma de aplicaciones web para la visualización, optimizando los procesos, y permitiendo contar con herramientas analíticas acordes con las necesidades del usuario.

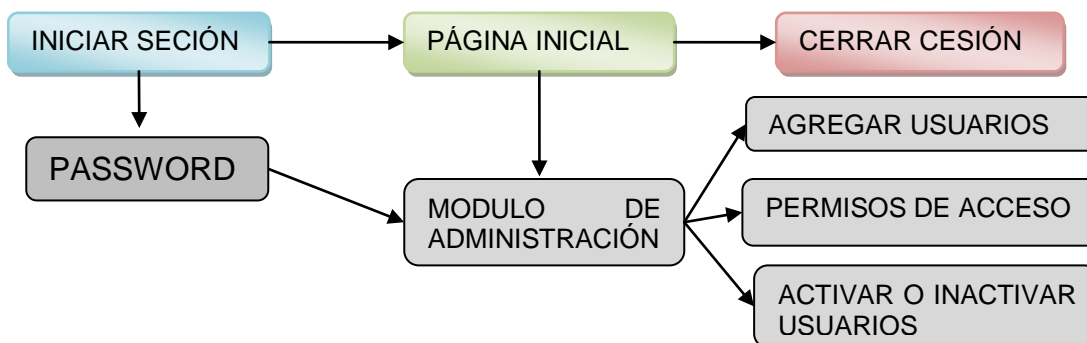
Al final del piloto quedarán definidos lo siguiente:

#### Módulos de Control Paper Store

Después de hecho el análisis con las necesidades del cliente se pensó en la realización de los siguientes módulos o estructuras en Control Paper Store para lo cual se definen los siguientes:

#### ➤ **Módulo de administración:**

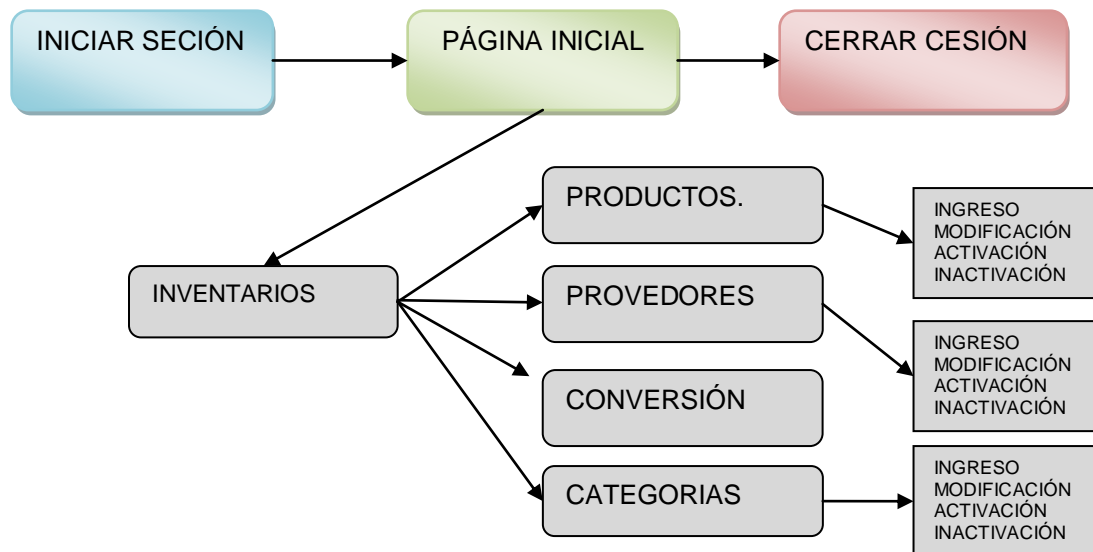
Control Paper Store permite la creación de usuarios que van a tener acceso al sistema, con la creación de niveles de accesibilidad aseguramos que solo el administrador general del sistema conceda los permisos a los demás usuarios del sistema, protegiendo la información de cada local.



**Tabla 48 Descripción módulo de administración.**  
Fuente: Carlos Landázuri.

➤ **Módulo de control de inventarios:**

Una vez implantado el sistema en el local de prueba (Papelandia) se procederá a hacer un inventario físico de la mercadería existente ingresando la información de cada producto, registrando su código de barras, si el producto no cuenta con código de barras se lo registrará con un código único. El sistema una vez empieza a facturar las ventas reducirá la mercadería existente y aumentará su cantidad cada vez que se compre mercadería para el local.

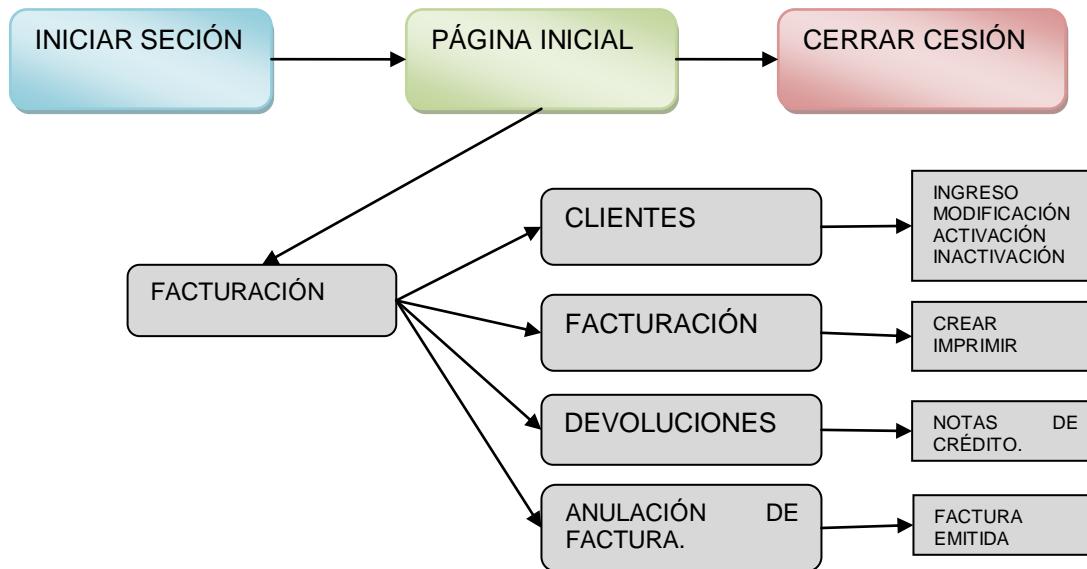


**Tabla 49 Descripción Módulo de Inventarios.**  
Fuente: Carlos Landázuri.

➤ **Módulo de Facturación:**

Control Paper Store tiene como principal objetivo facturar las compras que en una papelería se registran de tal manera que en la factura se imprima al cliente, el valor de cada producto desglosado el impuesto al valor agregado (IVA), en caso de que el producto lleve este impuesto, dará su subtotal y luego el valor

total de la compra, la fecha que realizó la compra, la fecha de validez del documento y los datos del local comercial



**Tabla 50 Descripción Módulo de Facturación.**

**Fuente: Carlos Landázuri.**

### ➤ **Módulo Cuentas por Pagar:**

La mayoría de locales de la Unión de Papelerías trabaja con sus proveedores a través de la modalidad de créditos en especial en temporada escolar es por esto que Control Paper Store permite saber con exactitud la mercadería que se ingresa al inventario de que proveedor proviene el total de las facturas, el numero de factura y la modalidad de crédito tiempo de pago e interés por mora.

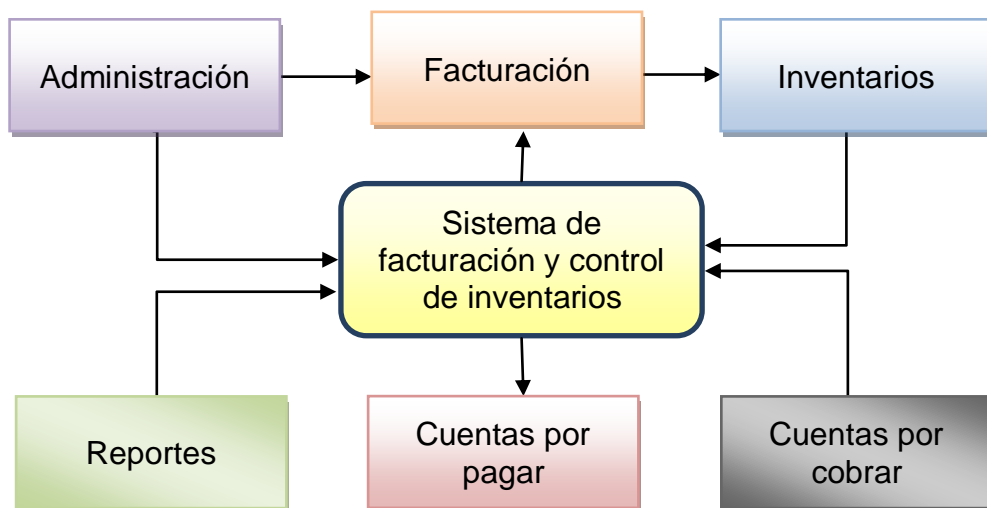


➤ **Módulo de Cuentas por Cobrar:**

En las cuentas por cobrar registra la mercadería concedida a crédito si bien son pequeños negocios y no proveedores se pensó en este módulo por el crecimiento que puede tener cada negocio.

➤ **Módulo de Reportes:**

Aquí se obtiene la información de las facturas almacenadas y emitidas, del kardex actual de la mercadería, información de clientes, proveedores y ventas diarias.



**Tabla 51 Descripción General del sistema Control Paper Store. Fuente: Carlos Landázuri.**

- El sistema será desarrollado con herramientas de software libre PHP y MySQL y tendrá licencia GPL que permite su libre uso, utilización y modificación del código fuente.

- El sistema será capaz de llevar el control de facturación del local comercial.
- Se registrara en la base de datos la mercadería existente y se ingresarán las compras realizadas, es decir total control de los inventarios.
- Los movimientos de la mercadería que sale y que ingresa será visualizada en el kardex.
- Después de registrar una venta en la base de datos se imprimirá en una factura de imprenta la misma que será entregada al cliente.
- El proyecto se lo desarrollará de tal manera que la utilización para el usuario sea fácil y tendrá una interfaz amigable.
- Realizará consultas como ventas diarias, mensuales, anuales, reportes de un producto específico en el inventario y un reporte de cuáles son los productos próximo agotarse para que los propietarios puedan abastecerse.
- Se entregará la documentación final con la metodología y un manual de usuario que ayuden a los usuarios finales a su utilización.

En particular, se proponen las siguientes actividades para resolver la problemática de Papelería Papelandia. Éstos son:

- Dimensionar el hardware necesario.
- Proveer e instalar el software necesario para soportar el Sistema Control Paper Store
- Gestionar y desarrollar el proyecto de implementación del Sistema Control Paper Store, que considera el modelo siguiente:

### ➤ **Solución Conceptual.**

- **Definición de Modelos.**

Microsoft Solution Framework (MSF v4.0) es una flexible e interrelacionada serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Originalmente creado en 1994 para conseguir resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos, se ha convertido posteriormente en un modelo práctico que facilita el éxito de los proyectos tecnológicos.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- **Restricciones del sistema.**

- ❖ El sistema tendrá la capacidad de ser multifuncional pero solo será instalado en un computador.
- ❖ Las facturas serán fabricadas en imprentas el sistema tiene la capacidad de imprimir la factura digital en una factura de imprenta.

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
06-12-2011	Carlos Landázuri	1.3	Alcance por módulos
12-12-2011	Carlos Landázuri	1.4	Solución conceptual
28-02-2013	Carlos Landázuri	1.5	Modulo de conversión

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.3	08-12-2011
Carlos Landázuri	1.4	14-12-2011
Carlos Landázuri	1.5	14-03-2013

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Alcance de Control Paper Store
Autor	Carlos Landázuri
Fecha de creación.	05-12-2011
Última Actualización.	14-03-2013

Nombre	Posición

## **CONTROL PAPER STORE**

Definición de escenarios

Autor	Carlos Landázuri
Cargo	Arquitecto del Proyecto
Fecha	19-12-2011

Versión 1.0

## ➤ **Definición de escenarios.**

Para Paper Control Store se ha definido los siguientes escenarios:

- ❖ Escenario Web Forms en Modelo vista controlador utilizando el IDE NetBeans en su versión 7.0.1

Este marco de escenario debe contener todos los escenarios que permitan interactuar a cada uno de los personajes anteriormente definidos. Para el sistema Paper Control Store se ha definido los siguientes escenarios:

- Escenario del Administración.- En este escenario el administrador posee una interfaz que le permite la creación de usuarios llenado sus datos mas importantes y la asignación de una contraseña, en la edición de usuarios se puede dar de baja a un usuario, asignar un nivel de acceso diferente, el cambio de contraseña o cualquier datos personal, de forma sencilla y rápida.
- Escenario de Inventarios.- En este escenario se definirá un escenario para la creación de grupos o categorías dentro de la mercadería que en las papelerías se maneja. Otro escenario que se muestra es la creación de proveedores con su datos más relevantes y el escenario de productos que contiene sub escenarios ya que aparte de la creación de productos, edición de la información, inactivar si por alguna razón se debe cancelar su venta, aquí encontramos el kardex del sistema dónde se muestran los movimientos de cada producto, las asignación de precios el ingreso de mercadería de forma que el usuario entienda todo el movimiento de los inventarios. También existe el módulo de conversión de mercadería de cajas a unidades cada vez que se necesite.

- Escenario de facturación.- Este escenario se presentan las pantallas para el ingreso de clientes con sus datos personales para que en el escenario de facturación se haga una búsqueda de un cliente y sus datos se carguen y pueda seguir realizando el proceso de facturación.
- Escenario de Cuentas.- Este se desplegará el menú donde indica si es cuanta por cobrar o por pagar cada opción nos muestra formularios con la información correspondiente y el ingreso de las nuevas cuentas.
- Escenario de Reportes.- Despliega los reportes de clientes proveedores, productos, kardex, cuantas por cobrar y pagar.

## Hoja de Revisión y Firma

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
20-12-2012	Carlos Landázuri	1.5	Formulario Usuarios
22-12-2012	Carlos Landázuri	1.6	Formulario Kardex

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.5	21-12-2011
Carlos Landázuri	1.6	22-12-2011

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Definición de escenarios
Autor	Carlos Landázuri
Fecha de creación.	19-12-2011
Última Actualización.	22-12-2011

Nombre	Posición



#### 4.1.4 Planificación

Diseño Conceptual

##### ***CONTROL PAPER STORE***

Casos de Uso

Autor	Carlos Landázuri
Cargo	Arquitecto del Proyecto
Fecha	09-01-2012

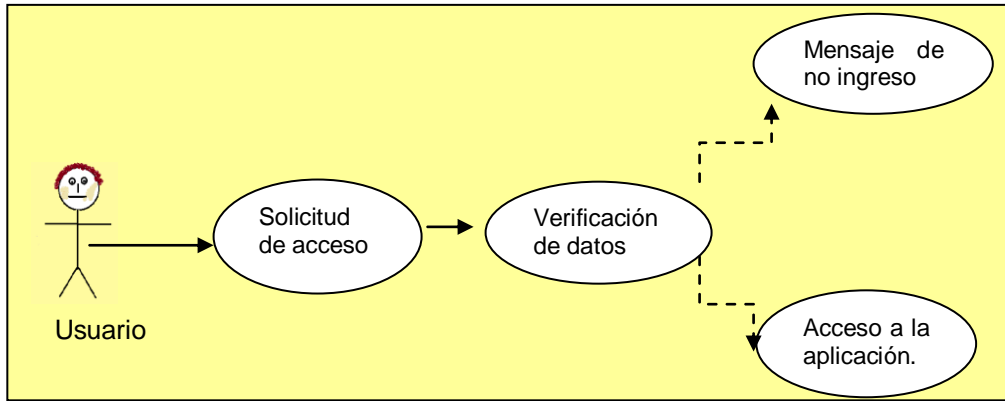
Versión 1.0

➤ **Diseño Conceptual**

• **Casos de Uso.**

<b>Caso de Uso 01:</b> Ingreso al sistema	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Ingreso de usuarios al sistema.	
<b>Visión General:</b> Es el proceso en el cual el usuario ingresa sus datos y se valida dicha información para poder acceder a la aplicación.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
<b>Acción</b>	<b>Respuesta del Sistema</b>
1. Ingreso al sistema se introduce el usuario.	1. Sistema pide el ingreso de un password.
2. Se introduce password	2. Sistema valida la información ingresada y despliega el menú principal.
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Si la información ingresada no coincide con la registrada en la base de datos se genera un mensaje para el usuario indicando que el usuario o contraseña son incorrectos.</li> </ul>	

**Tabla 52** Caso de uso de ingreso al sistema.  
Elaborado: Carlos Landázuri.



**Figura 12: Caso de Uso Ingreso al Sistema.**  
Fuente: Carlos Landázuri.

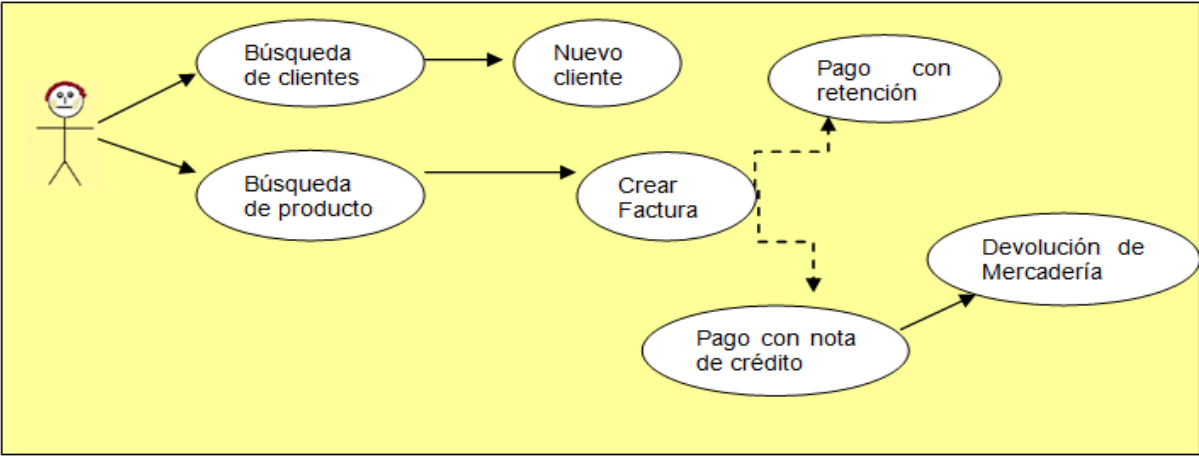
<b>Caso de Uso 02: Administración de Categorías.</b>	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Administrar categorías de productos.	
<b>Visión General:</b> Es el proceso en el cual el usuario que tenga acceso a este pueda crear, editar o inactivar una categoría.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
<b>Acción</b>	<b>Respuesta del Sistema</b>
1. Ingreso en el espacio de búsqueda la categoría a editar o inactivar.	1. El sistema valida la información ingresada y en caso de encontrar despliega un listado de todos las categorías que coinciden con la búsqueda.
2. Ingreso en el botón que indica la creación de una nueva categoría.	2. Despliega los campos a llenar con la información principal de la nueva categoría.
3. Ingreso en el botón que indica la edición de una categoría.	3. Sistema indica el formulario con la información del producto.
4. Usuario llena el formulario con la información de la categoría editada y presiona botón guardar	4. Sistema guarda la información nueva ingresada.
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Si la información ingresada en el buscador no coincide con la registrada en la base de datos se muestra un mensaje que indica que la categoría buscada no existe.</li> <li>• Si la información editada no es correcta o algún campo no se llena se despliega un mensaje indicando que no se pudo guardar la información.</li> </ul>	

**Tabla 53 Caso de uso Categorías de productos.**

**Elaborado: Carlos Landázuri.**

<b>Caso de Uso 03:Facturación</b>	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Facturar las compras realizadas por un cliente.	
<b>Visión General:</b> Realizar los procesos de facturación, detallando las características del producto, el precio unitario, mostrando sub totales, suma por impuestos y el total a pagar.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
<b>Acción</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. Ingreso de Numero de cédula del cliente.</li> <li>2. Si cliente no está registrado de se presiona icono de nueva información.</li> <li>3. Ingreso de del código o nombre del producto.</li> <li>4. Usuario ingresa la cantidad a facturar.</li> <li>5. Usuario guarda la información.</li> <li>6. Una vez verificada la información usuario imprime factura.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema busca si el cliente y carga la información del cliente.</li> <li>2. Sistema despliega pantalla para el ingreso de un nuevo cliente.</li> <li>3. Se despliega los productos que coinciden con la búsqueda.</li> <li>4. Sistema carga detalle del producto con su costo parcial y total de la compra.</li> <li>5. Aparece una ventana con la imagen de la factura que se va imprimir.</li> </ol>
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Si la información ingresada en el buscador no coincide con la registrad en la base de datos se muestra un mensaje que indica que el cliente no está ingresado.</li> <li>• Si al buscar un producto no se despliega nada el sistema muestra un mensaje que indica que no se encontró un producto con esas características.</li> <li>• En la casilla de descuento se ingresa el tipo de descuento que se va aplicar a un cliente.</li> </ul>	

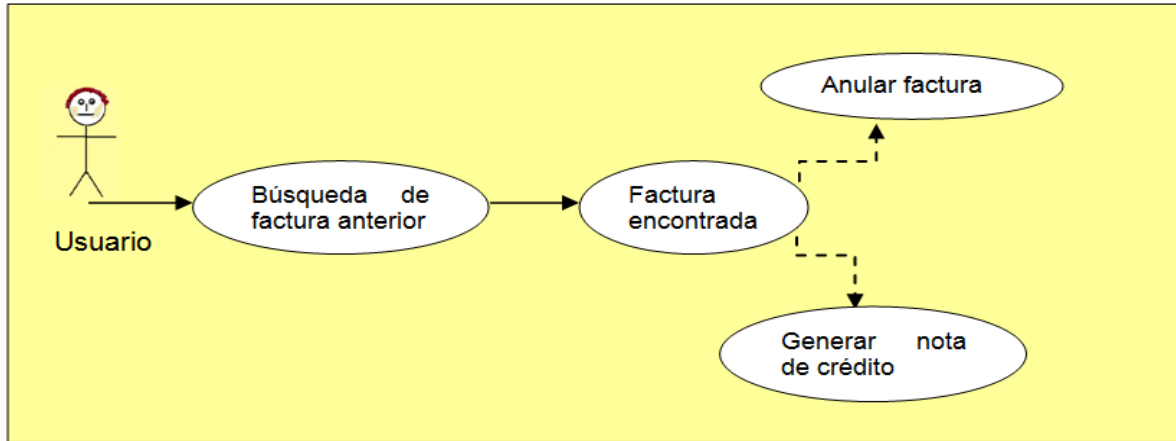
Tabla 54 Caso de Uso Facturación.  
Elaborado: Carlos Landázuri.



**Figura 13: Caso de Uso Facturación.**  
**Elaborado: Carlos Landázuri.**

<b>Caso de Uso 04:</b> Devolución de Mercadería													
<b>Actores:</b> Usuario													
<b>Propósito:</b> Registrar la devolución de una artículo vendido													
<b>Visión General:</b> Realizar el proceso de volver un artículo al inventarió y generar una nota de crédito.													
<b>Tipo:</b> Primario esencial.													
<b>Curso Típico de Eventos:</b>													
<table border="1"> <thead> <tr> <th>Acción</th> <th>Respuesta del Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Ingreso de número de factura existente en la base de datos</td> <td>1. El sistema busca si la factura está registrada y carga la información.</td> </tr> <tr> <td>2. Escogemos si deseamos aplicar una nota de crédito o anular factura.</td> <td>2. Sistema despliega pantalla para ingreso de nota de crédito o anulación de factura.</td> </tr> <tr> <td>3. Usuario escoge opción anular factura</td> <td>3. Se muestra mensaje de factura anulada.</td> </tr> <tr> <td>4. Usuario escoge opción nota de crédito</td> <td>4. Se ingresa una nueva nota de crédito.</td> </tr> <tr> <td>5. Una vez verificada la información usuario imprime nota de crédito</td> <td>5. Aparece una ventana con la imagen de la nota de crédito a imprimir.</td> </tr> </tbody> </table>		Acción	Respuesta del Sistema	1. Ingreso de número de factura existente en la base de datos	1. El sistema busca si la factura está registrada y carga la información.	2. Escogemos si deseamos aplicar una nota de crédito o anular factura.	2. Sistema despliega pantalla para ingreso de nota de crédito o anulación de factura.	3. Usuario escoge opción anular factura	3. Se muestra mensaje de factura anulada.	4. Usuario escoge opción nota de crédito	4. Se ingresa una nueva nota de crédito.	5. Una vez verificada la información usuario imprime nota de crédito	5. Aparece una ventana con la imagen de la nota de crédito a imprimir.
Acción	Respuesta del Sistema												
1. Ingreso de número de factura existente en la base de datos	1. El sistema busca si la factura está registrada y carga la información.												
2. Escogemos si deseamos aplicar una nota de crédito o anular factura.	2. Sistema despliega pantalla para ingreso de nota de crédito o anulación de factura.												
3. Usuario escoge opción anular factura	3. Se muestra mensaje de factura anulada.												
4. Usuario escoge opción nota de crédito	4. Se ingresa una nueva nota de crédito.												
5. Una vez verificada la información usuario imprime nota de crédito	5. Aparece una ventana con la imagen de la nota de crédito a imprimir.												
<b>Cursos Alternativos:</b>													
<ul style="list-style-type: none"> <li>• Si la información ingresada en el buscador no coincide con la registrad en la base de datos se muestra un mensaje que indica que la factura no existe.</li> <li>• En el caso de que el cliente pague su nueva factura con una nota de crédito emitida por el local, el total de la factura se le resta el valor de la nota de crédito y esta se anula.</li> </ul>													

**Tabla 55 Caso de uso Devolución de Mercadería.**  
**Elaborado: Carlos Landázuri.**



**Figura 14: Caso de Devolución de Mercadería.**  
Elaborado: Carlos Landázuri.



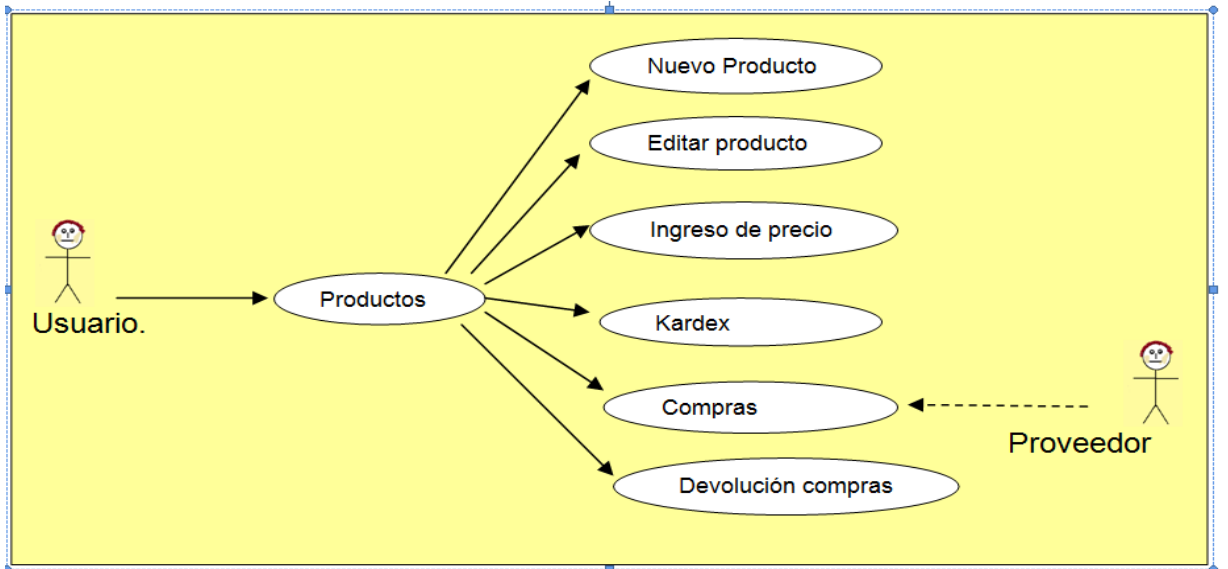
<b>Caso de Uso 05:</b> Administración de Proveedores.	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Administración de proveedores.	
<b>Visión General:</b> Es el proceso en el cual el usuario que tenga acceso a este pueda crear, editar o inactivar un proveedor.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
Acción	Respuesta del Sistema
<ol style="list-style-type: none"> <li>1. Ingreso en el espacio de búsqueda el proveedor a editar o inactivar.</li> <li>2. Ingreso en el botón que indica la creación de un nuevo proveedor.</li> <li>3. Ingreso en el botón que indica la edición de un proveedor.</li> <li>4. Usuario llena el formulario con la información del proveedor ingresado o editado y presiona botón guardar</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema valida la información ingresada y en caso de encontrar despliega un listado de todos los proveedores que coinciden con la búsqueda.</li> <li>2. Despliega los campos a llenar con la información principal de la nueva proveedor.</li> <li>3. Sistema indica el formulario con la información del producto.</li> <li>4. Sistema guarda en la base de datos la información nueva ingresada.</li> </ol>
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Si la información ingresada en el buscador no coincide con la registrada en la base de datos se muestra un mensaje que indica que el proveedor no ha sido registrado.</li> <li>• Si la información nueva o editada no es correcta en algún campo se despliega un mensaje indicando que no se puede guardar la información.</li> </ul>	

**Tabla 56 Administración de Proveedores.**

**Elaborado:** Carlos Landázuri.

<b>Caso de Uso 06:</b> Administración de productos	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Administración de productos.	
<b>Visión General:</b> Es el proceso en el cual el usuario que tenga permisos de acceso pueda crear, editar o inactivar un producto.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
<b>Acción</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. Ingreso en el espacio de búsqueda del producto a editar o inactivar.</li> <li>2. Ingreso en el botón que indica la creación de un nuevo producto.</li> <li>3. Ingreso en el botón que indica la edición de un producto</li> <li>4. Usuario llena el formulario con la información del producto ingresado o editado y presiona botón guardar.</li> <li>5. Ingreso en el botón inactividad de un producto.</li> <li>6. Ingreso a al ícono de precio de un producto.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema valida la información ingresada y en caso de encontrar despliega un listado de todos los productos que coinciden con la búsqueda.</li> <li>2. Despliega los campos a llenar con la información principal del nuevo producto.</li> <li>3. Sistema indica el formulario con la información del producto.</li> <li>4. Sistema guarda en la base de datos la información nueva ingresada.</li> <li>5. Sistema bloquea las búsquedas de dicho producto para evitar su venta.</li> <li>6. El sistema despega una pantalla con su código de barras y el porcentaje de ganancia que se le va a dar a dicho producto.</li> </ol>
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Si la información ingresada en el buscador no coincide con la registrada en la base de datos se muestra un mensaje que indica que el producto no está ingresado.</li> <li>• Si la información nueva o editada no es correcta en algún campo se despliega un mensaje indicando que no se puede guardar la información.</li> </ul>	

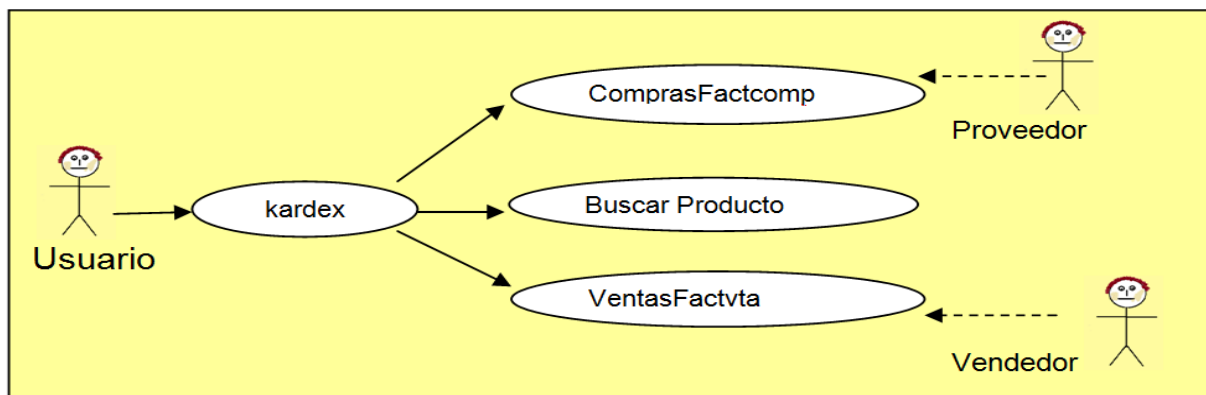
**Tabla 57 Caso de uso Administración de productos.**  
**Fuente: Carlos Landázuri.**



**Figura 15 Caso de Uso Administración de productos.**  
Fuente: Carlos Landázuri.

Caso de Uso 07: Kardex	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Verificación de movimientos de productos en un lapso de tiempo.	
<b>Visión General:</b> Proceso donde se registran los movimientos de un producto específico así se puede saber cuándo se compró dicho producto con qué número de factura, cuándo se lo vendió y el costo promedio de venta.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
Acción	Respuesta del Sistema
1. Ingreso de una fecha específica para consultar los movimientos. 2. Botón "Cancelar" al presionarlo.	1. El sistema despliega todos los movimientos de dicho producto en el lapso de tiempo 2. Sistema nos vuelve a la pantalla de productos.
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>Se puede abandonar la pantalla en cualquier momento con el botón cerrar sesión.</li> </ul>	

**Tabla 58 Caso de Uso Kardex**  
**Elaborado: Carlos Landázuri.**



**Figura 16 Caso de Uso Kardex.**  
**Fuente: Carlos Landázuri.**

<b>Caso de Uso 08:</b> Administración de clientes.	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Administración de clientes.	
<b>Visión General:</b> Es el proceso en el cual el usuario que tenga acceso a este pueda crear, editar o inactivar un clientes.	
<b>Tipo:</b> Primario esencial.	
<b>Curso Típico de Eventos:</b>	
Acción	Respuesta del Sistema
<ol style="list-style-type: none"> <li>1. Ingreso en el espacio de búsqueda del cliente a editar o inactivar.</li> <li>2. Ingreso en el botón que indica la creación de un nuevo cliente.</li> <li>3. Ingreso en el botón que indica la edición de un cliente</li> <li>4. Usuario llena el formulario con la información del cliente ingresado o editado y presiona botón guardar</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema valida la información ingresada y en caso de encontrar despliega un listado de todos los clientes que coinciden con la búsqueda.</li> <li>2. Despliega los campos a llenar con la información principal del nuevo cliente.</li> <li>3. Sistema indica el formulario con la información del cliente.</li> <li>4. Sistema guarda en la base de datos la información nueva ingresada.</li> </ol>
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Si la información ingresada en el buscador no coincide con la registrada en la base de datos se muestra un mensaje que indica que el cliente no está registrado.</li> <li>• Si la información nueva o editada no es correcta en algún campo se despliega un mensaje indicando que no se puede guardar la información.</li> </ul>	

**Tabla 59: Administración de Clientes.**

**Elaborado: Carlos Landázuri.**

<b>Caso de Uso 09: Reportes</b>	
<b>Actores:</b> Usuario	
<b>Propósito:</b> Consultar movimientos del negocio	
<b>Visión General:</b> Realizar consultas de los diferentes movimientos y reportes de ventas entre otros.	
<b>Tipo:</b> Secundario	
<b>Curso Típico de Eventos:</b>	
<b>Acción</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. Ingreso del tipo de reporte requerido en el menú.</li> <li>2. Ingreso fecha de referencia.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema genera los reportes hasta el momento ingresados.</li> <li>2. Sistema genera el reporte de lo ingresado hasta la fecha de referencia.</li> </ol>
<b>Cursos Alternativos:</b>	
<ul style="list-style-type: none"> <li>• Para reportes de facturas se debe ingresar número de factura a consultar.</li> </ul>	

**Tabla 60Caso de uso de Reportes.**

**Elaborado: Carlos Landázuri.**

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
12-01-2012	Carlos Landázuri	1.6	Casos de Uso facturación
14-01-2012	Carlos Landázuri	1.7	Casos de Uso Anulación Facturas.

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.6	13-01-2012
Carlos Landázuri	1.7	15-01-2012

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Casos de Uso
Autor	Carlos Landázuri
Fecha de creación.	09-01-2012
Última Actualización.	15-01-2012

Nombre	Posición

## **CONTROL PAPER STORE**

Definición de arquitectura.

Autor	Carlos Landázuri
Cargo	Arquitecto del Proyecto
Fecha	13-02-2012

Versión 1.0



## ➤ Definición de Proyecto.

### Arquitectura Multicapa:

Para el desarrollo del sistema “Control Paper Store” se pensó en una arquitectura multicapa la misma que permite particionar todo el sistema en distintas unidades o capas funcionales: cliente, presentación, lógica-de-negocio, integración.

La ventaja de utilizar esta arquitectura es asegurar una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica-de-negocios.

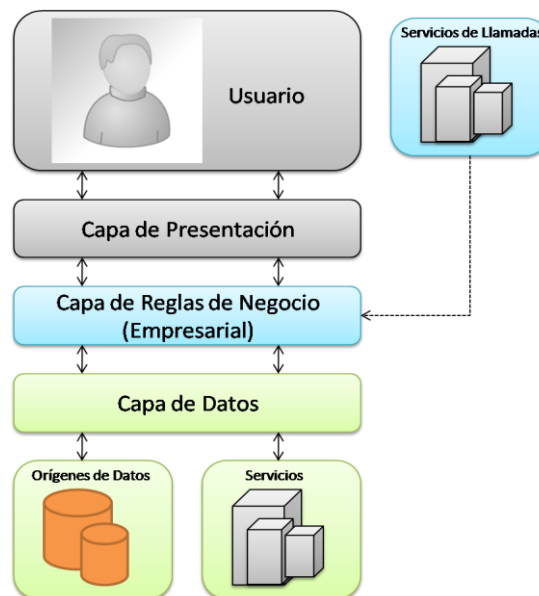
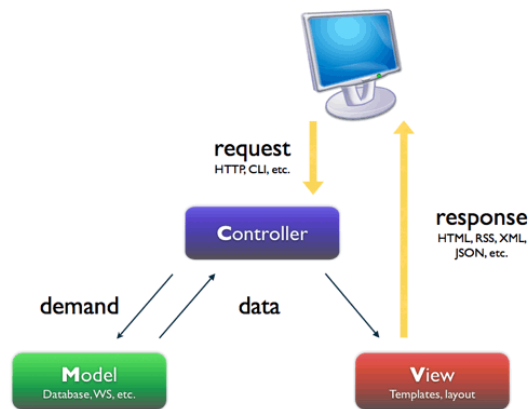


Figura 17: Diagrama de la Arquitectura Multicapa.

Fuente: Wikipedia.



**Figura 18 Diagrama de la Arquitectura MVC.**  
**Fuente: Wikipedia.**

## Arquitectura Modelo Vista Controlador.

El Modelo Vista Controlador (MVC), es una arquitectura de diseño software para separar los componentes de aplicación en tres niveles, interfaz de usuario, lógica de control y lógica de negocio.

MVC es el patrón de diseño arquitectural recomendado para aplicaciones interactivas Java. MVC separa los conceptos de diseño, y por lo tanto reduce la duplicación de código, el centralizamiento del control y hace que la aplicación sea más extensible. Ayuda al equipo de trabajo a enfocarse en sus habilidades principales y a colaborar a través de interfaces claramente definidos. MVC es el patrón de diseño arquitectural para la capa de presentación.

Diagrama general de la arquitectura.

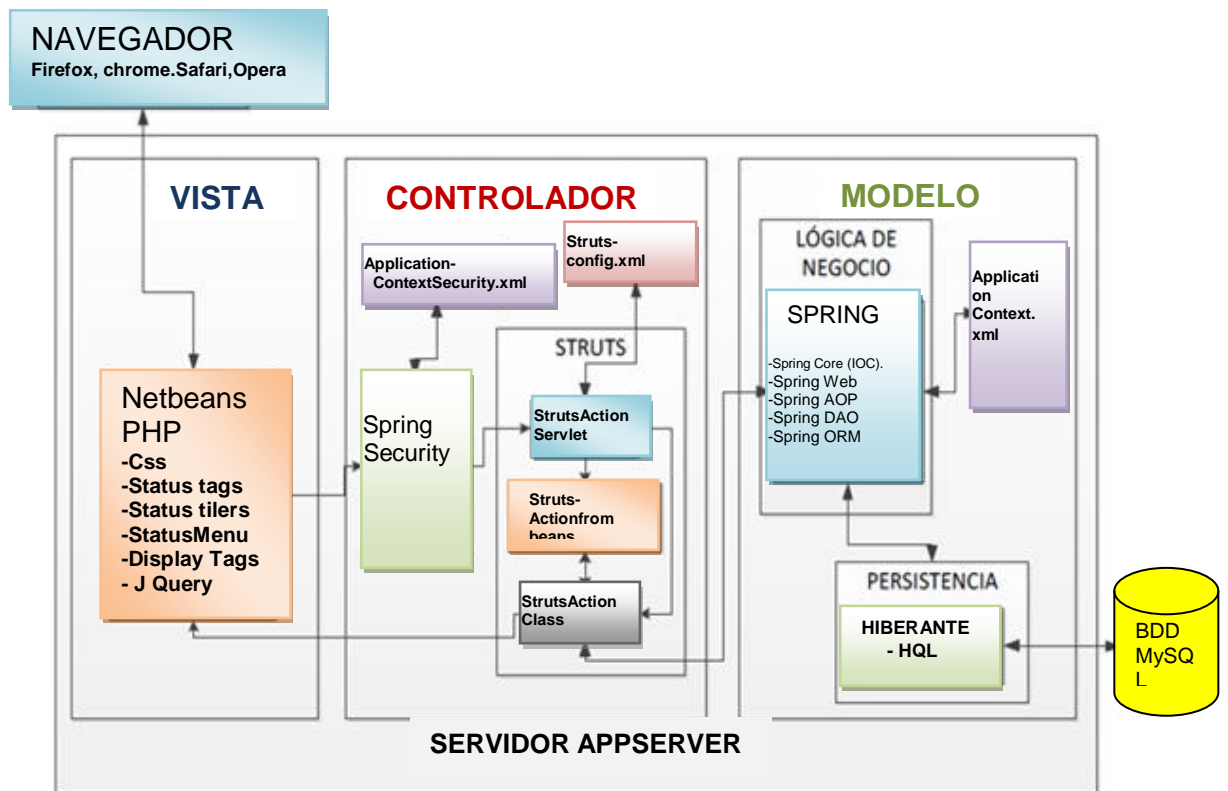


Tabla 61 Diagrama de la arquitectura MVC.

Elaborado: Carlos Landázuri.

## **Capa de Presentación**

Reside en el cliente y es la encargada de presentar la interfaz con la que interactuará el usuario.

## **Capa Lógica**

Reside en un nivel intermedio entre el cliente y el servidor de base de datos, y es la encargada de planificar y priorizar las solicitudes de información.

En este caso, por tratarse de una aplicación Web, utilizaremos un Servidor de Aplicaciones Web.

## **Capa de Datos**

Esta capa reside en el nivel de recursos, el cual está compuesto por tantos recursos como el sistema pretende integrar. Normalmente se aloja en el Servidor de Base de Datos.

## **Flujo de la Información**

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

El usuario interactúa con la interfaz de usuario de alguna forma, por ejemplo pulsando un botón.

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario.
- El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.

- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados cualquier cambio.
- Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
- La interfaz de usuario espera nueva nuevas interacciones del usuario, comenzando el ciclo nuevamente.

### **PHP 5.2.5:**

PHP5.2.5 es un lenguaje de programación de tipo interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

### **Ventajas:**

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

## Apache2

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix, Linux, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

## **MySQL5.0.45**

Es un sistema de base de datos relacional, multihilo y multiusuario, es de uso libre en proyectos, pero si se usa de manera comercial es necesario comprar licencia. Algunas características de este motor de base de datos son:

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Soporte a multiplataforma.
- Procedimientos almacenados.
- Disparadores (triggers).
- Vistas actualizables.
- Soporte a VARCHAR
- INFORMATION\_SCHEMA
- Modo Strict
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle.
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial).
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB.
- Soporte para SSL.
- Sub-SELECTs (o SELECTs anidados).
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin soporte automático para múltiples maestros por esclavo.

## **Netbeans IDE 7.2**

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser

extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes.

Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación.

Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- Administración de ventanas.
- Framework basado en asistentes (diálogo paso a paso).

El IDE de NetBeans es un producto libre y gratuito sin restricciones de uso, soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.



## Definiciones de términos

A continuación se anotan algunas abreviaturas y conceptos que el presente documento se puede mencionar.

Término	Definición
Administrador	Se refiere a la persona que va a manejar el sistema.
Botones	Espacio visual de un sistema donde se especifica una actividad del sistema al presionar el mismo.
Base de Datos	Herramienta informática dónde se almacena la información.
Cliente	Persona que contrata o requiere el sistema informático y el mismo que va a dar la aprobación final.
Control Paper Store	Nombre que se le da al sistema a desarrollar.
Framework:	Es una plataforma donde se desarrolla y se crea un sistema ya que esta ofrece soporte para el mismo. Incluye bibliotecas y un lenguaje interpretado.
Interfaz	Pantalla donde se muestra una serie de aspectos para la manipulación de un sistema.
MSF	Microsoft Solution Framework.
MVC	Modelo Vista Controlador.

**Tabla 62 : Definición de términos referentes a MSF.**  
**Fuente: Carlos Landázuri.**

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
15-02-2012	Carlos Landázuri	1.7	Diagrama de arquitectura
25-02-2012	Carlos Landázuri	1.8	Cambio de versión en las herramientas.

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.7	16-02-2012
Carlos Landázuri	1.8	25-02-2012

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Definición de arquitectura
Autor	Carlos Landázuri
Fecha de creación.	13-02-2012
Última Actualización.	25-02-2012

Nombre	Posición

## **CONTROL PAPER STORE**

Diagrama de clases y base de base de datos.

Autor	Carlos Landázuri
Cargo	Arquitecto del Proyecto
Fecha	27-02-2012

## Diagrama de la base de datos. Base de datos appFacturaciónInventarios.

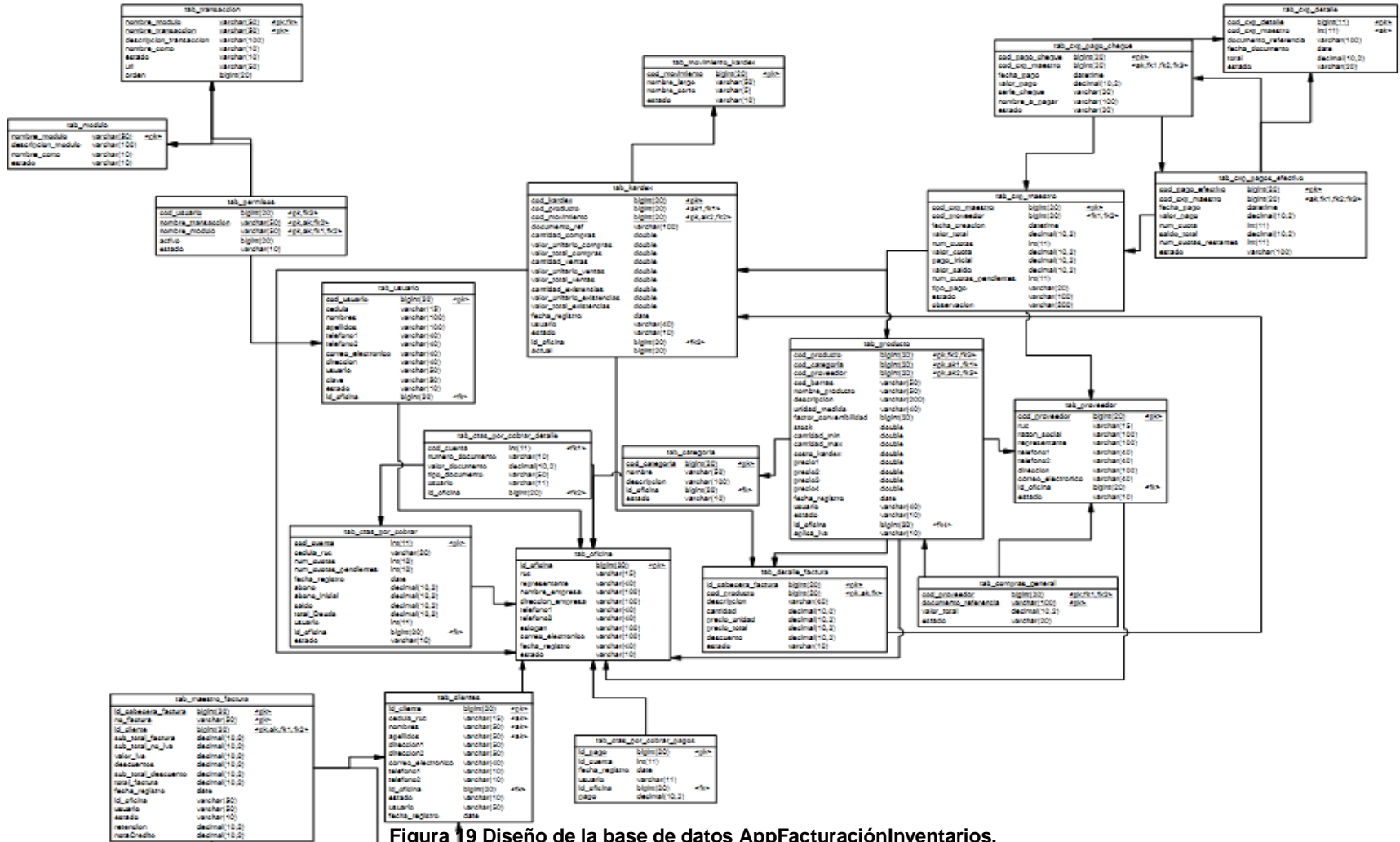


Figura 19 Diseño de la base de datos AppFacturaciónInventarios.  
Fuente: Carlos Landázuri

Estructura de la tabla categoría.

NOMBRE TABLA: tab_categoria	
Nombre Variable	Tipo
cod_categoria	bigint(20)
nombre	varchar(50)
descripcion	varchar(100)
id_oficina	bigint(20)
estado	varchar(10)

**Tabla 63 Estructura de la tabla categoría.**  
Fuente: Carlos Landázuri.

Estructura de la tabla clientes

NOMBRE TABLA: tab_clientes`	
Nombre Variable	Tipo
id_cliente	bigint(20)
cedula_ruc	varchar(15)
nombres	varchar(50)
apellidos	varchar(50)
direccion1	varchar(50)
direccion2	varchar(50)
correo_electronico	varchar(40)
telefono1	varchar(10)
id_oficina	bigint(20)
estado	varchar(10)
usuario	varchar(50)
fecha_registro	date default

**Tabla 64 Estructura de la tabla cliente.**  
Elaborado: Carlos Landázuri.

Estructura de la tabla detalle factura.

NOMBRE TABLA: tab_detalle_factura.	
Nombre Variable	Tipo
id_cabecera_factura	bigint(20)
cod_producto	bigint(20)
descripcion	varchar(40)
cantidad	decimal(10,0)
precio_unidad	decimal(10,2)
precio_total	decimal(10,2)
descuento	decimal(10,2)
estado.	varchar(10)

**Tabla 65** estructura de la tabla detalle factura.  
Elaborado: Carlos Landázuri.

Estructura de la tabla Maestro Factura.

NOMBRE TABLA: tab_maestro_factura	
Nombre Variable	Tipo
id_cabecera_factura	bigint(20)
no_factura	varchar(50)
id_cliente	bigint(20)
sub_total_factura	decimal(10,2)
sub_total_no_iva	decimal(10,2)
valor_iva	decimal(10,2)
descuentos	decimal(10,2)
sub_total_descuento	
total_factura`	decimal(10,2)
fecha_registro	
id_oficina	varchar(50)
usuario	varchar(50)
estado	varchar(10)

**Tabla 66 Estructura de la tabla Maestro factura.**

**Elaborado: Carlos Landázuri.**

Estructura de tabla para la tabla Módulos

NOMBRE TABLA: tab_módulo	
Nombre Variable	Tipo
nombre_modulo	varchar(50)
descripcion_modulo	varchar(100)
nombre_corto	varchar(10)
estado	varchar(10)

**Tabla 67 Estructura de la tabla módulos.**

**Elaborado: Carlos Landázuri.**

Estructura de tabla para la tabla Movimiento de kardex

NOMBRE TABLA: tab_movimiento_kardex	
Nombre Variable	Tipo
cod_movimiento	bigint(20)
nombre_largo	varchar(50)
nombre_corto	varchar(5)
estado	varchar(10)

**Tabla 68 Estructura de la tabla Kardex.**

**Elaborado: Carlos Landázuri.**

Estructura de tabla para la tabla Nota crédito

NOMBRE TABLA: tab_nota_crédito	
Nombre Variable	Tipo
cod_nota_creditono_factura	bigint(20)
nombre_corto	varchar(50)
estado	varchar(5)
motivo_anulacion	varchar(100)
fecha_registro`	
id_oficina	bigint(20)
usuario	varchar(40)
estado	varchar(10)

**Tabla 69 Estructura de la tabla Nota de crédito.**

**Elaborado: Carlos Landázuri**



Estructura de tabla para la tabla proveedor

NOMBRE TABLA: tab_oficina	
Nombre Variable	Tipo
id_oficina	bigint(20)
ruc	varchar(15)
representante	varchar(40)
nombre_empresa	varchar(100)
direccion_empresa	varchar(100)
telefono1	varchar(40)
telefono2	varchar(40)
eslogan	varchar(100)
correo_electronico	varchar(100)
fecha_registro	varchar(40)
estado`	varchar(10)

**Tabla 70 Estructura para la tabla proveedor.**

**Elaborado: Carlos Landázuri.**

Estructura de tabla para la tabla permisos

NOMBRE TABLA: tab_permisos	
Nombre Variable	Tipo
cod_usuario	bigint(20)
nombre_transaccion	varchar(50)
nombre_modulo	varchar(50)
activo	bigint(20)
estado	varchar(10)

**Tabla 71 Estructura de la tabla Permisos.**

**Elaborado: Carlos Landázuri.**

Estructura de tabla para la tabla producto.

NOMBRE TABLA: tab_producto	
Nombre Variable	Tipo
cod_producto	bigint(20)
cod_categoria	bigint(20)
cod_proveedor	bigint(20)
cod_barras	varchar(50)
nombre_producto	varchar(50)
descripcion	varchar(200)
unidad_medida	varchar(40)
factor_convertibilidad	bigint(20)
stock	double default
cantidad_min	double default
cantidad_max	double default
costo_kardex	double default
precio1	double default
precio2	double default
fecha_registro	date
usuario	default}varchar(40)
estado	varchar(10)
id_oficina	bigint(20)
aplica_iva	varchar(10)

**Tabla 72 Estructura de la tabla Productos.**

**Elaborado: Carlos Landázuri.**

Estructura de tabla para la tabla proveedor

NOMBRE TABLA: tab_proveedor	
Nombre Variable	Tipo
cod_proveedor	bigint(20)
ruc	bigint(15)
razon_social	varchar(100)
representante	varchar(100)
telefono1	varchar(40)
telefono2	varchar(40)
direccion	varchar(100)
correo_electronico	varchar(40)
id_oficina`	bigint(20)
estado	varchar(10)

**Tabla 73 Estructura de la Tabla Proveedor.**  
Elaborado: Carlos Landázuri.

Estructura de tabla para la tabla Transacción

NOMBRE TABLA: tab_transaccion	
Nombre Variable	Tipo
nombre_modulo	varchar(50)
nombre_transaccion	varchar(50)
descripcion_transaccion	varchar(100)
nombre_corto	varchar(10)
estado	varchar(10)
url	varchar(50)
orden	bigint(20)

**Tabla 74 Estructura de la tabla Transacción.**  
Elaborado: Carlos Landázuri.

## Estructura de tabla para la tabla Usuario

NOMBRE TABLA: tab_usuario	
Nombre Variable	Tipo
cod_usuario	bigint(20)
cedula	varchar(15)
nombres	varchar(100)
apellidos	varchar(100)
telefono1	varchar(40)
telefono2	varchar(40)
correo_electronico	varchar(40)
direccion	varchar(40)
usuario	varchar(50)
clave	varchar(50)
estado	varchar(10)
id_oficina`	bigint(20)

**Tabla 75 Estructura de la tabla Usuarios.**

**Elaborado: Carlos Landázuri**

## Hoja de Revisión y Firma

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
29-02-2012	Carlos Landázuri	1.8	Modelo entidad relación
17-03-2012	Carlos Landázuri	1.9	Aumento de tablas
12-12-2012	Carlos Landázuri	1.10	Aumento de tablas

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.8	01-03-2012
Carlos Landázuri	1.9	20-03-2012
Carlos Landázuri	1.10	25-03-2012

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Definición de arquitectura
Autor	Carlos Landázuri
Fecha de creación.	27-02-2012
Última actualización	25-03-2012

Nombre	Posición

## 4.1.5 Desarrollo de Control Paper Store

### *CONTROL PAPER STORE*

Documento de Diseño de Interfaz

Autor	Carlos Landázuri
Cargo	Arquitecto del Proyecto
Fecha	05-11-2012

## ➤ **Documento de Diseño de Interfaz.**

### **Diseño de Interfaz.**

Se debe crear una forma de comunicación entre Control Paper Store y el usuario esta se la realiza a través de formularios o pantallas con la información requerida por el usuario, con los campos a llenar para obtener un dato o un grupo de datos específicos. Bajo estos parámetros establecemos que el rendimiento del sistema frente a las necesidades del usuario aquí más que la programación del sistema se analizan aspectos como la funcionalidad de Control Paper Store, la interoperabilidad entre cada módulo, la información ingresada, los formularios que el sistema me devuelve, el fácil manejo para los usuarios, y la funcionalidad en sí de acuerdo a los establecido.

### **Diseño Estético.**

Control Paper Store es una aplicación web que brinda una interfaz gráfica atractiva al usuario el uso es bastante simple a través de comandos enviados por el mouse y el teclado pudiendo comunicar las operaciones realizadas a través de mensajes mostrados en los formularios.

También se necesita un modelo de facturas para la impresión de las ventas por lo que se optó por un diseño dónde los usuarios puedan imprimir en impresoras normales a cartucho o tinta continua con esto evitamos gastos a los propietarios las facturas deben ser impresas con la autorización del SRI de acuerdo a los registros tributarios de cada local de papelerías.

- **Prototipo de la aplicación Web Control Paper Store**

Prototipo de factura.



 <b>LIBRERÍA &amp; PAPELERÍA</b> 001 - 001 <b>Nº 000001506</b>		<b>FACTURA</b> Aut. S.R.L: 1111669671 Landázuri Ortiz Carlos Julio RUC: 0401593421001 Dir.: Eugenio Espejo y Cristobal Colón Telf: 062 609 193 Ibarra - Ecuador		 <b>LIBRERÍA &amp; PAPELERÍA</b> 001 - 001 <b>Nº 000001506</b>		<b>FACTURA</b> Aut. S.R.L: 1111669671 Landázuri Ortiz Carlos Julio RUC: 0401593421001 Dir.: Eugenio Espejo y Cristobal Colón Telf: 062 609 193 Ibarra - Ecuador	
Cliente: Cédula/Ruc                      Teléfono: Dirección                        Fecha:				Cliente: Cédula/Ruc                      Teléfono: Dirección                        Fecha:			
Cantidad	Detalle	P. Unitario	P. Total	Cantidad	Detalle	P. Unitario	P. Total
		SUBTOTAL DESCUENTOS SUB. DSCTO IVA (12%) TOTAL				SUBTOTAL DESCUENTOS SUB. DSCTO IVA (12%) TOTAL	
Responsable                      Cliente				Responsable                      Cliente			
Las oportunidades no son producto de la casualidad, mas bien son resultado del trabajo. (Tonatihu)				Las oportunidades no son producto de la casualidad, mas bien son resultado del trabajo. (Tonatihu)			
<small>LIBRERÍA PAPER STORE - OFICINA GENERAL TEL: 062 609 193 IBARRA - ECUADOR - 2012</small>				<small>LIBRERÍA PAPER STORE - OFICINA GENERAL TEL: 062 609 193 IBARRA - ECUADOR - 2012</small>			

Figura 20: Modelo de factura.

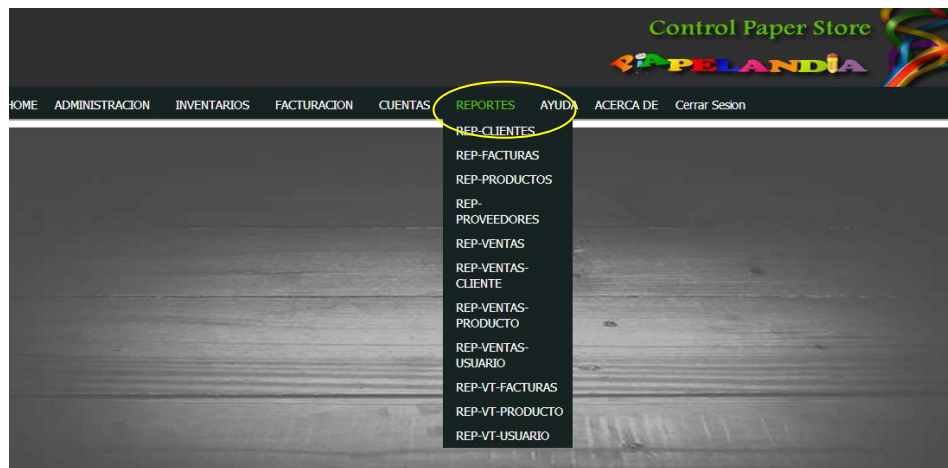
Elaborado: Carlos Landázuri.



Diseño de Pantallas.



**Figura 21** Pantalla de control de acceso.  
Elaborado: Carlos Landázuri.



**Figura 22** Pantalla del menú principal.  
Elaborado: Carlos Landázuri.

Control Paper Store  
PAPELANDIA

HOME ADMINISTRACION INVENTARIOS FACTURACION CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

EMPRESA

USUARIOS

DATOS GENERALES DE LA EMPRESA

GUARDAR SALIR

RUC: 1002562369001

REPRESENTANTE: CARLOS LANDAZURI

NOMBRE EMPRESA: SU PAPELERIA

DIRECCION: LOS CEIBOS

TELF. MOVIL: 0987654321

TELF. CONVENCIONAL: 0623456789

ESLOGAN: TU MEJOR PROVEEDOR EN PAPELES

CORREO: supapeleria@correo.com

Figura 23 Pantalla de “Empresa” sistema Control Paper Store.  
Fuente: Carlos Landázuri.

Control Paper Store  
PAPELANDIA

HOME ADMINISTRACION INVENTARIOS FACTURACION CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

EMPRESA

USUARIOS

Busqueda:

1 - 3 de 3

OPCIONES	CEDULA	EMPLEADO	USUARIO	TELEFONOS	DIRECCION	ESTADO
	1002562369	ADMINISTRADOR ADMINISTRADOR	ADMINISTRADOR	093439788 - 062951355	ADMINISTRADOR	ACTIVO
	2002002002	CARLOS VALENCIA	cvalencia	0987654321 - 0987654321	LA VICTORIA	ACTIVO
	0401593421	CARLOS JULIO LANDAZURI ORTIZ	clandazuri	093623258 - 062609674	IBARRA	INACTIVO

Figura 24 Pantalla “Usuarios” sistema Control Paper Store.  
Fuente: Carlos Landázuri.

Control Paper Store  
PAPELANDIA

HOME ADMINISTRACION INVENTARIOS FACTURACION CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

CATEGORIAS

ADMINISTRACION DE CATEGORIAS

Busqueda:

1 2 1 - 5 de 9

CODIGO	NOMBRE	DESCRIPCION	ESTADO	
	1	PAPELES	TODA CLASE DE PAPELES	A
	2	ESFEROS	ESFEROS	A
	3	CDS	CDS	A
	4	CUADERNIOS	CUADERNIOS	A
	5	LAMINAS ILUSTRATIVAS	LAMINAS ILUSTRATIVAS	A

Figura 25 Pantalla “Categorías” sistema Control Paper Store.  
Fuente: Carlos Landázuri.

ID	BARRAS	NOMBRE	DESCRIPCION	COSTO	STOCK	MIN.	MAX.	UNIDAD	ESTADO
1	1031370445237	PAPEL CARBON	PAPEL CARBON	0.45	33	10	100	DOCENA	A

Figura 26 Pantalla “Productos” Sistema Control Paper Store.  
Fuente: Carlos Landázuri.

CODIGO RUC	RAZON SOCIAL	REPRESENTANTE	DIRECCION	TELEFONOS	ESTADO
1001001001001	SUPER PACO	HECTOR DIAZ	QUITO	0987654321   0987654321	A

Figura 27: Pantalla “Proveedores” Sistema Control Paper Store.  
Fuente: Carlos Landázuri.

CODIGO	NOMBRES	VALOR UNITARIO	CANTIDAD	VALOR TOTAL
1	PAPEL CARBON	0.45	10	4.50

SubTotal: 0.00  
 SubTotalNoIva: 4.50  
 Descuento(%): 0.00  
 Valor descuento: 0.00  
 Iva(12%): 0.00  
 Total: 4.50

Figura 28: Pantalla “Anulación Facturas” Sistema Control Paper Store.  
Fuente: Carlos Landázuri.

Control Paper Store  
PAPELANDIA

HOME ADMINISTRACION INVENTARIOS **FACTURACION** CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

ANULAR FACTURAS  
**CLIENTES**  
FACTURACION

Busqueda:

CEDULA/RUC	NOMBRES	TELEFONOS	DIRECCION	MAIL	ESTADO
2002002002002	ANDRES SALZAR	0987654321 - 0987654321	AV.17 DE JULIO	carlin@hotmail.com	INACTIVO
1002562369001	CRISTHIAN ANDRADE	0987654321 - 0987654321	LOS CEIBOS	cpga@hotmail.com	ACTIVO
1001001001011	PAUL ORDOÑEZ	098712345 - 067897897	YACUCALLE	paul@jhkgasdkj.com	ACTIVO
2002002002	GLORIA VALENCIA	062959592 -	LA VICTORIA	mayu_13@hotmail.com	ACTIVO
3003003003	MARYLIN ANDRADE	062333555 - 092545454	LOS CEIBOS	mary@yahoo.com	ACTIVO

Figura 29: Pantalla “Clientes” Sistema Control Paper Store.  
Fuente: Carlos Landázuri.

Control Paper Store  
PAPELANDIA

HOME ADMINISTRACION INVENTARIOS **FACTURACION** CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

FACTURAR PRODUCTOS

GUARDAR CANCELAR

RUC APELLIDOS A


CEDULA / RUC	NOMBRES	DIRECCION	TELEFONOS	CORREO
1001001001011	ORDOÑEZ PAUL	YACUCALLE	098712345   067897897	paul@jhkgasdkj.com

PRODUCTO: [codigo | nombre]

CODIGO	NOMBRE	STOCK	PRECIO	CANTIDAD	TOTAL
1031370445237	PAPEL CARBON	33	0.45	10	4.5

SubTotal	0
SubTotalNoIva	4.5
Descuento(%)	
Valor descuento	0
Iva(12%)	0
Total	4.5

Figura 30: Pantalla “Facturación” Sistema Control Paper Store.  
Fuente: Carlos Landázuri.

Control Paper Store  


INVENTARIOS    FACTURACION    CUENTAS    **REPORTES**    AYUDA    ACERCA DE    Cerrar Sesión

LISTA

REP-CLIENTES

REP-FACTURAS

REP-PRODUCTOS

REP-PROVEEDORES

REP-VENTAS

REP-VENTAS-CLIENTE

REP-VENTAS-PRODUCTO

REP-VENTAS-USUARIO

REP-VT-FACTURAS

REP-VT-PRODUCTO

REP-VT-USUARIO

COD_CLIENTE	CEDULA	NOMBRES Y APELLIDOS	DIRECCION	TEL	CORREO	OFICINA	FECHA REGISTRO
6	1002562369001	CRISTHIAN ANDRADE	LOS CEIBOS Y DAULE		554321 cpga@hotmail.com	1	2012-11-25
7	1001001001011	PAUL ORDONEZ	YACUCA		7897 paul@hkgasdkj.com	1	2012-02-27
8	2002002002	GLORIA VALENCIA	LA VICTORIA CARLOS 10-26		mayu_13@hotmail.com	1	2012-11-25
9	3003003003	MARYLIN ANDRADE	LOS CEIBOS JUBONES		5454 mary@yahoo.com	1	2012-07-04
10	4004004004	PEDRO VALENCIA	LKJHGF		ghgh@hotmail.com	1	2012-07-04
11	5050505005	ANITA PEREZ	YACUCA		any@hotmail.com	1	2012-07-04
12	6006006006	DANILO JACOME	REDONDI		dany@hotmail.com	1	2012-07-04
13	1003055769	RAFAEL ANDRADE	LA VICTORIA			1	2012-10-14
14	0400159342	CARLOS JULIO LANDAZURI ORTIZ	LAS PALMAS		33210 CJLANDAZURI@HOTMAIL.COM	1	2012-10-15
15	5654789544	LIZBETH OLIVO	LOS CEIBOS		5465 OLIVO@HOTMAIL.COM	1	2012-10-16

**Figura 31: Pantalla “Reportes” Sistema Control Paper Store**  
Fuente: Carlos Landázuri.

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
07-11-2012	Carlos Landázuri	1.10	Diseño de pantallas
27-11-2012	Carlos Landázuri	1.11	Diseño de Facturas

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.10	09-11-2012
Carlos Landázuri	1.11	28-11-.2012

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Definición de arquitectura
Autor	Carlos Landázuri
Fecha de creación.	05-11-2012
Última actualización	28-11-2012

Nombre	Posición

#### 4.1.6 Estabilización de Control Paper Store

##### *CONTROL PAPER STORE*

##### Documento de Pruebas Unitarias

Autor	Carlos Landázuri
Cargo	Arquitecto del proyecto, desarrollador, test
Fecha	09-01-2013

➤ **Pruebas Unitarias.**

Con las pruebas unitarias de cada módulo del sistema para comprobar o verificar el correcto funcionamiento del sistema.

MÓDULO	RESULTADOS
<b>INGRESO AL SISTEMA</b>	
Autenticación	<ul style="list-style-type: none"> <li>✓ El sistema reconoce si es un usuario normal o el Administrador correctamente.</li> <li>✓ Verifica usuarios y contraseñas correctamente</li> </ul>
<b>ADMINISTRACIÓN</b>	
Usuarios	<ul style="list-style-type: none"> <li>✓ El registro de usuarios funciona correctamente.</li> <li>✓ La modificación de usuarios funciona correctamente.</li> <li>✓ La activación o inactivación de usuarios funciona correctamente.</li> <li>✓ Los permisos de usuarios funciona correctamente.</li> </ul>
Configurar	<ul style="list-style-type: none"> <li>✓ El rango de facturas guarda la secuencia correctamente.</li> <li>✓ Los parámetros de retención funcionan correctamente.</li> <li>✓ La activación y eliminación de descuentos funciona correctamente.</li> </ul>
<b>INVENTARIOS</b>	
Categorías	<ul style="list-style-type: none"> <li>✓ La creación de categorías funciona correctamente.</li> <li>✓ La modificación de categorías funciona correctamente.</li> <li>✓ La activación o inactivación de categorías funciona correctamente.</li> <li>✓ Conversión de cajas a unidades funciona correctamente.</li> </ul>



Productos	<ul style="list-style-type: none"> <li>✓ La creación de productos funciona correctamente.</li> <li>✓ La modificación de productos funciona correctamente.</li> <li>✓ La activación o inactivación de un producto funciona correctamente.</li> <li>✓ El Kardex de cada producto funciona correctamente</li> <li>✓ El ingreso de mercadería funciona correctamente.</li> <li>✓ La asignación de un porcentaje de ganancia a un producto funciona correctamente.</li> <li>✓ La búsqueda de productos funciona correctamente.</li> </ul>
Proveedores	<ul style="list-style-type: none"> <li>✓ La creación de un proveedor funciona correctamente.</li> <li>✓ La modificación de un proveedor funciona correctamente.</li> <li>✓ La activación o inactivación de un proveedor funciona correctamente.</li> </ul>
<b>FACTURACIÓN</b>	
Anular Facturas	<ul style="list-style-type: none"> <li>✓ La búsqueda de facturas funciona correctamente</li> <li>✓ La anulación de facturas funciona correctamente.</li> <li>✓ La creación de notas de crédito funcionan correctamente.</li> </ul>
Clientes	<ul style="list-style-type: none"> <li>✓ La creación de un cliente funciona correctamente.</li> <li>✓ La modificación de la información de un cliente funciona correctamente.</li> <li>✓ La activación o inactivación de un cliente funciona correctamente</li> </ul>
Facturación	<ul style="list-style-type: none"> <li>✓ La búsqueda de clientes funciona correctamente.</li> <li>✓ La creación de un cliente funciona correctamente.</li> <li>✓ La búsqueda de productos funciona correctamente.</li> <li>✓ El ingreso de productos a la factura funciona correctamente.</li> </ul>

	<ul style="list-style-type: none"> <li>✓ El porcentaje de descuento a una factura funciona correctamente.</li> <li>✓ El descuento del total de la factura por retención funciona correctamente.</li> <li>✓ El descuento del total de la factura por nota de crédito funciona correctamente.</li> </ul>
<b>Cuentas por cobrar y pagar.</b>	
Cuentas por cobrar	<ul style="list-style-type: none"> <li>✓ La búsqueda de clientes funciona correctamente.</li> <li>✓ Los datos del cliente se muestran correctamente.</li> <li>✓ Los datos de cuenta se muestran correctamente.</li> <li>✓ Los datos de pago se muestran y calculan correctamente.</li> <li>✓ La re negociación de una cuenta funciona correctamente.</li> </ul>
<b>Cuentas por pagar</b>	
	<ul style="list-style-type: none"> <li>✓ La búsqueda de proveedor funciona correctamente.</li> <li>✓ La creación de una nueva cuenta funciona correctamente.</li> </ul>
<b>Reportes</b>	
Reportes de Clientes Reporte de Facturas Reporte de Productos Reporte de Proveedores Reporte de Ventas Reporte de Ventas por cliente Reporte de Ventas por producto	<ul style="list-style-type: none"> <li>✓ Reportes de Clientes funciona correctamente</li> <li>✓ Reporte de Facturas funciona correctamente</li> <li>✓ Reporte de Productos funciona correctamente</li> <li>✓ Reporte de Proveedores funciona correctamente</li> <li>✓ Reporte de Ventas funciona correctamente</li> <li>✓ Reporte de ventas por cliente funciona correctamente</li> </ul>

**Tabla 76 Pruebas unitarias por módulo.  
Elaborado: Carlos Landázuri**

## Hoja de Revisión y Firma. Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
12-01-2013	Carlos Landázuri	1.10	Ajuste de formato factura
14-03-2013	Carlos Landázuri	1.11	Modulo de conversión, banner de Uni papelerías

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.11	13-01-2013

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Documento de pruebas unitarias
Autor	Carlos Landázuri
Fecha de creación.	09-01-2013
Última actualización	13-01-2013

Nombre	Posición

## 4.1.7 Implementación de Paper Control Store

### *CONTROL PAPER STORE*

#### ➤ **Manuales de Usuario**

Autor	Carlos Landázuri
Cargo	Arquitecto del proyecto, desarrollador.
Fecha	18-01-2013

En la fase de liberación se presentan los manuales de instalación y de usuario adjunto en los ANEXOS

## Hoja de Revisión y Firma.

### Registro Historial de Cambios

Fecha	Autor	Versión	Referencias del Cambio
21-01-2013	Carlos Landázuri	1.11	Incorporar manual de instalación

### Revisiones

Nombre	Versión Aprobada	Fecha
Carlos Landázuri	1.11	23-01-2013

### Propiedades del Documento

Elemento	Detalle
Título del Documento	Manuales de usuario
Autor	Carlos Landázuri
Fecha de creación.	18-01-2013
Última actualización	23-01-2013

Nombre	Posición

## 5 CONCLUSIONES

1. Se aplicó la metodología de desarrollo Microsoft Solution Framework V 3.0 en el desarrollo del sistema de Software libre “Control Paper Store” ,el mismo que fue creado en el IDE de Java NetBeans 7.2 en lenguaje PHP 5.2.5 con base de datos MySQL 5.0.5 y corriendo en el servidor AppServ (Apache), todas herramientas de Software libre. Ha sido instalado en la Papelería “Papelandia”, este sistema es liberado bajo licencia GNU GPL y queda a disposición de la Unión de Papelerías de la ciudad de Ibarra con los respectivos instaladores de las aplicaciones, documentación y manuales de usuario.
2. Se estudió los procesos de desarrollo de software desde sus inicios haciendo énfasis en lo referente a ciclos de procesos, modelos metodologías de desarrollo y aspectos de calidad en el desarrollo de software, estableciendo los parámetros que originaron la aparición de MSF como metodología para la realización del sistema “Control Paper Store”.
3. Se analizó todo lo referente a Software Libre, investigando el origen, historia y la evolución que ha tenido en los últimos años el software “Open Source”, la filosofía que origino su aparición y las licencias más utilizadas. Bajo esta filosofía se desarrolló un sistema con herramientas que no necesitan pago de licencias esto hizo más factible que la Unión de papelerías cuenten ya con un sistema informático de una manera gratuita logrando sistematizar su negocio sin realizar mayores gastos. El sistema cuenta con una licencia GLP, los acuerdos se pueden analizar en la ayuda del sistema.

4. Se analizó las principales metodologías de desarrollo conociendo más de cada una de ellas para analizar las diferencias y semejanzas con la metodología seleccionada Microsoft Solution Framework conocimos los parámetros establecidos por la metodología para la elaboración del Sistema “Control Paper Store” Se analizó cada fase con sus artefactos, actores y documentación a entregar, pudiendo establecer los siguientes módulos: Administración, Facturación, Productos, Proveedores, Familias de productos, Kardex, Cuentas por Cobrar, Cuentas por Pagar y Reportes. Cada módulo se lo estudió desde la etapa de visión hasta la fase de entrega.
  
5. Mediante la metodología MSF se desarrolló un sistema informático “Control Paper Store” específico para la Unión de Papelerías de la ciudad de Ibarra. El sistema está instalado en la papelería “Papelandia” y realiza las funciones de facturación y control de inventarios. Los usuarios poseen un sistema dónde pueden registrar sus compras, asignar un precio a sus productos, verificar un porcentaje de ganancia. De igual forma los clientes que comparan un producto tienen registro de su compra con una factura reglamentada por el SRI.
  
6. Después de realizar esta investigación pudimos concluir que en la actualidad existen muchas herramientas de software privativo que están a disposición de cualquier usuario es el caso de Microsoft Solution Framework 3.0 que es una herramienta exclusiva de los desarrolladores de Microsoft pero se puede acceder a la documentación en su portal oficial es así que se desarrolló un sistema Control Paper Store con herramientas de software libre, liberado bajo licencia GPL pero desarrollado bajo los parámetros de MSF.

## 6 RECOMENDACIONES

1. Para que la aplicación funcione correctamente es recomendable que los usuarios de Control Paper Store lo ejecuten en un navegador como Internet Explorer versión 7.0 o superior, Mozilla Firefox, Google Chrome y cualquier navegador que tenga compatibilidad con JavaScript.
2. Para la elección de una metodología es recomendable estudiar los modelos de desarrollo para entender su estructura y analizar si está de acuerdo a nuestras necesidades
3. Promover el uso del software libre en sistemas orientados a pequeños negocios de nuestra provincia como papelerías pudiendo disminuir los costos de adquisición y fomentando la sistematización de las actividades comerciales demostrando que el software libre brinda herramientas confiables estables en cualquier ámbito a implementar.
4. MSF es perfecto para proyectos pequeños ofrece una excelente manera de documentar un proyecto aportando una visión clara de lo que se quiere, organizando al equipo de trabajo con tareas específicas y arrojando un producto con parámetros de calidad pero no es recomendable para proyectos grandes ya que la guía para sus documentación es limitada.
5. Se recomienda que los usuarios de Control Paper Store ingresen todas las compras al sistema aun cuando el cliente no pida factura para que no existan errores en los inventarios.
6. Cuando utilicemos una herramienta de software privativo pero que este a disposición de todo el público hacer consultas permanentes de las licencias porque en muchas ocasiones se agregan licencias o parámetros sobre su utilización.



## ➤ **Temas de tesis sugeridos**

- Investigación de Java Cryptography Architecture aplicado al estudio del trazado en la zona limítrofe de la provincia de Imbabura.
- Estudio del framework DHTMLX de HTML5 JavaScript para móviles y dispositivos táctiles orientadas en aplicación móvil.
- Aplicación de “Metodologías de desarrollo Híbridas” en el desarrollo de un sistema para pequeños negocios de la provincia de Imbabura.

## **7 BIBLIOGRAFÍA:**

- Pressman, R. (2010). Ingeniería del Software. España: Mc. Graw Hill, Séptima Edición.
- Somerville, I. (2011). Ingeniería del Software. España: Pearson Educación, Novena Edición.
- Roldán, D. y Valderas, P. (2010). Aplicaciones Web, Un Enfoque Práctico. México: Alfaomega Grupo Editor.
- Corporación de Estatutos y Publicaciones (2010). Ley de Régimen Tributario Interno. Ecuador: Taller de la Corporación de Estatutos y Publicaciones.
- Cohen, D., Lindvall, M. y Costa, P. (2003). Agile Software Development. [En línea]. Estados Unidos: Data and Analysis Center for Software. En: <https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0CDQQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.94.7%26rep%3Drep1%26type%3Dpdf&ei=2DIBUe6yN9HD0AGj7ICwCg&usq=AFQjCNES6h9VbExJrwXOq6aIW2Q7gcARIQ&bvm=bv.41248874,d.dmQ>
- Chipia, J. (2010). Análisis de la importancia del uso de metodologías de desarrollo y métricas de calidad en software en aplicaciones educativas. [En

- línea]. En: <http://www.slideshare.net/JoanFernandoChipia/anlisis-de-la-importancia-del-uso-de-metodologas-de-desarrollo-y-mtricas-de-calidad-de-software-en-aplicaciones-educativas-3565076#>
- Arévalo, M. (2010). MSF ejemplo documento Visión y Alcance. [En línea]En: <http://ebookbrowse.com/memberpostmasters201002-xls-d99793325>
  - MICROSOFT, (2002). Disciplina de Administración de Riesgos v1.1. [En línea] En:<http://dspace.ups.edu.ec/bitstream/123456789/989/3/CAPITULO%20II.pdf>
  - Cataldi, Z. (2000). Metodología de diseño, desarrollo y evaluación de software educativo. [En línea] En:<http://laboratorios.fi.uba.ar/lsi/cataldi-tesisdemagistereninformatica.pdf>
  - Salazar, A. (2006). Inteligencia de Negocios. [En línea]. Ecuador: EPN. En: <http://bibdigital.epn.edu.ec/bitstream/15000/2736/1/CD-0514.pdf>
  - Monografías. (2009). Licencias de Software. [En línea]. En: <http://www.monografias.com/trabajos55/licencias-de-software/licencias-de-software.shtml>
  - Blog (2011). Reglas del Open Source. [En línea]. En: <http://normaitsccsoftwarelibre.blogspot.com/>
  - Herrera, G., Guadalupe, W y Sánchez, Susana. (2006). Software libre vs software propietario. [En línea]. México: Creative Commons. En: <http://www.softwarelibre.cl/drupal/files/32693.pdf>
  - Microsoft Corporation (2002). Disciplina para la administración de proyectos MSF v. 1.1. [En línea]. En: <http://ebookuniverse.net/disciplina-para-administracin-proyectos-msf-pdf-d6750463>
  - Castro, R. (2004). Estructura básica del proceso unificado de desarrollo de software. [En línea]. Colombia: Universidad ICESI. En: [http://www.icesi.edu.co/revistas/index.php/sistemas\\_teleomatica/article/view/933](http://www.icesi.edu.co/revistas/index.php/sistemas_teleomatica/article/view/933).
  - GNU Operating System. (2010). What is free software? [En línea]. En: <http://www.gnu.org/philosophy/free-sw.html.en>
  - Presidencia De La República (2009). Estrategia Para La Implantación De Software Libre En La Administración Pública Central. [En línea]. Ecuador: Subsecretaría de la Información.En:

[http://www.softwarelibre.org.bo/wiki/lib/exe/fetch.php?media=migracion:emslapc\\_v1\\_ecuador.pdf](http://www.softwarelibre.org.bo/wiki/lib/exe/fetch.php?media=migracion:emslapc_v1_ecuador.pdf)

- Sandoval, R. (2007). Microsoft Solutions Framework. Estrategias de Desarrollo del Software. [En línea]. En:<http://repositorio.uc.cl/xmlui/bitstream/handle/123456789/1420/502313.pdf?sequence=1>
- Gattaca (2009). Presentación de Metodología MSF. [En línea]. En:[http://equiposige.blogspot.es/img/Etapas\\_metodologia\\_MSF.pdf](http://equiposige.blogspot.es/img/Etapas_metodologia_MSF.pdf)
- Arias, L. y León, D. (2007). Desarrollo de un Sistema CRM para el Sector de la Capacitación utilizando MSF V 3.1. [En línea]. Ecuador: Escuela Politécnica Nacional. En: <http://bibdigital.epn.edu.ec/bitstream/15000/234/1/CD-0633.pdf>
- Páez, L. (2010). Framework. [En línea]. En:<http://ingenieriasoftwaredos.wikispaces.com/file/view/Framework.pptx>
- Boehm, B. (2006). A Spiral Model of Software Development and Enhancement. [En línea]. En:<http://weblog.erenkrantz.com/~jerenk/phase-ii/Boe88.pdf>
- Wikipedia (2010). Microsoft Solutions Framework [En línea]. En:[http://en.wikipedia.org/wiki/Microsoft\\_Solutions\\_Framework](http://en.wikipedia.org/wiki/Microsoft_Solutions_Framework)
- Benavides, C. (2008). Hojas de Estilo en Cascada. [En línea]. En:<http://www.sidar.org/recur/desdi/mcss/index.php>
- MSDN (2007). Información general sobre Team Foundation Build. [En línea]. En:[http://msdn.microsoft.com/es-es/library/ms181710\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms181710(v=vs.80).aspx)
- Fernández, J. (2003). Seguridad informática y software libre. [En línea]. En:<http://es.tldp.org/Informes/informe-seguridad-SL/informe-seguridad-SL.pdf>
- Innova Empresarial (2005). Metodologías De Desarrollo. [En línea]. En:[http://www.innovaempresarial.com/docs/Metodologia\\_Desarrollo.pdf](http://www.innovaempresarial.com/docs/Metodologia_Desarrollo.pdf)
- Maestro Web (2007). Licencias Libres de Software.[En línea]. En: <http://www.maestrosdelweb.com/editorial/licencias-libres-de-software-ii/>
- Microsoft Corporation (2003). Microsoft Solutions Framework version 3.0 Overview. [En línea]. En:<http://www.microsoft.com/msf>.
- Wikipedia (2010). Licencia de Software [En línea]. En:[http://es.wikipedia.org/wiki/Licencia\\_de\\_software](http://es.wikipedia.org/wiki/Licencia_de_software)

- Rivera, R. (2007). Introducción a la Ingeniería del Software. [En línea]. En:<http://oacosta334.blogspot.es>
- Wikipedia (2010). Metodología de Desarrollo del Software. [En línea]. En:[http://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_softwar\\_e](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_softwar_e)
- Pérez, I., Pérez, R. y Rodríguez, D. (2008). Metodología de Desarrollo del Software. [En línea]. En:[https://docs.google.com/viewer?a=v&q=cache:ao1ocrZUKLoJ:solusoft-g11.googlecode.com/files/Metodologias%2520de%2520desarrollo.pdf+&hl=es&gl=ec&pid=bl&srcid=ADGEESj3OPC7nSTXgKR7mB4TKv9I2Sb-OvZDYcwoKOunhKFJwRUWEAdXru4\\_1bU9j8dZrnfGSHxyziBKHzvRHP2kJmFXN6jRtKIHuUxoyGqp2r0hVxDyeg0ElwXfEgQxmtLEuFeKN&sig=AHIEtbRnt6M8wCVb0e-sO6K8SH4iOQo3Hg](https://docs.google.com/viewer?a=v&q=cache:ao1ocrZUKLoJ:solusoft-g11.googlecode.com/files/Metodologias%2520de%2520desarrollo.pdf+&hl=es&gl=ec&pid=bl&srcid=ADGEESj3OPC7nSTXgKR7mB4TKv9I2Sb-OvZDYcwoKOunhKFJwRUWEAdXru4_1bU9j8dZrnfGSHxyziBKHzvRHP2kJmFXN6jRtKIHuUxoyGqp2r0hVxDyeg0ElwXfEgQxmtLEuFeKN&sig=AHIEtbRnt6M8wCVb0e-sO6K8SH4iOQo3Hg)
- Virrueta, A. (2010). Metodología de Desarrollo del Software. [En línea]. México: Instituto Tecnológico Superior De Apatzingán. En: <http://www.monografias.com/trabajos-pdf4/metodologias-de-desarrollo-software/metodologias-de-desarrollo-software.pdf>
- Chávez, A. (2007). Microsoft Solution Framework (MSF). [En línea]. En:<http://achavez334.blogspot.es/>
- Cruz, S. (2007). Microsoft Solution Framework 3.0 (MSF). [En línea]. En:<http://scruz334.blogspot.es/i2007-11/>
- Monografías (2007). El Software Libre. [En línea]. En:<http://www.monografias.com/trabajos12/elsoflib/elsoflib.shtml>
- Arévalo, M. (2010). Plantillas Microsoft Solution Framework. [En línea]. En:<http://arevalomaria.wordpress.com/2010/12/09/plantillas-microsoft-solution-framework/>
- Díaz, M. (2011). Metodologia Rational Unified Process (RUP) vs Metodologia Extreme Programming (XP). [En línea]. En:<http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>

- Software Libre en el Ecuador (2012). ¿Qué es el software libre?. [En línea]. En:<http://softwarelibre.info/>
- Castañeda, A., Ibarra, C., Mares, R. y Navarrete, M. (2009). Desarrollo de sistemas. Modelo Microsoft Solutions Framework (MSF). [En línea]. México: Instituto Tecnológico Del Valle Del Guadiana. En: [http://chacharaselnido.com/ITVG/Desarrollo%20de%20Sistemas/Unidad\\_2/A/Unidad%202%20-%20MSF%20-%20Grupo%20A.pdf](http://chacharaselnido.com/ITVG/Desarrollo%20de%20Sistemas/Unidad_2/A/Unidad%202%20-%20MSF%20-%20Grupo%20A.pdf)
- SIDAR. (2010). Uso de Hojas de Estilo en Cascada en la revisión. [En línea]. En:<http://www.sidar.org/recur/revisa/usocss/index.php>.

## **8 ANEXOS**

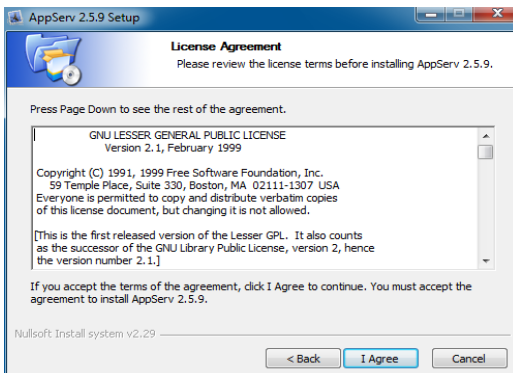
# MANUAL DE INSTALACIÓN “Control Paper Store”.

## 1. Instalación de AppServer.

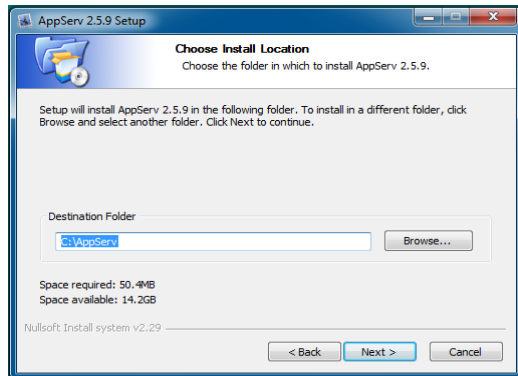
Ejecutamos el instalador y nos aparecerá una pantalla de inicio de instalación.



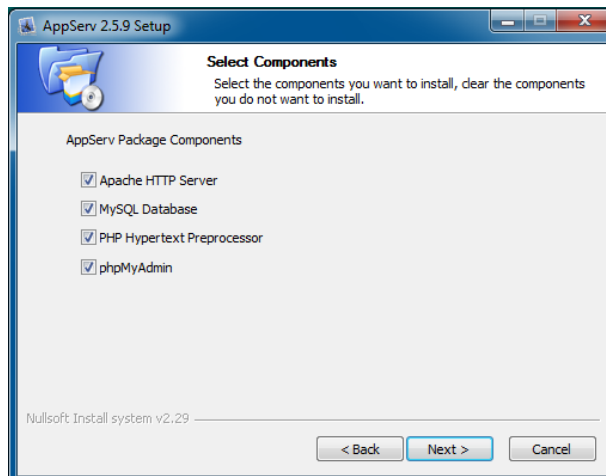
Presionamos “NEXT”



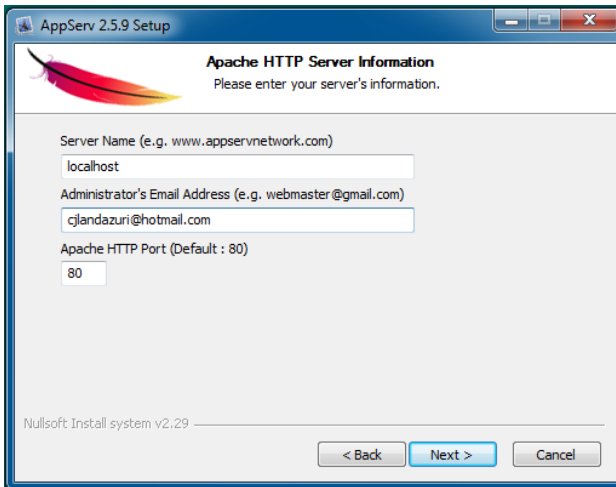
Aceptamos Las condiciones de licencia “I AGREE”



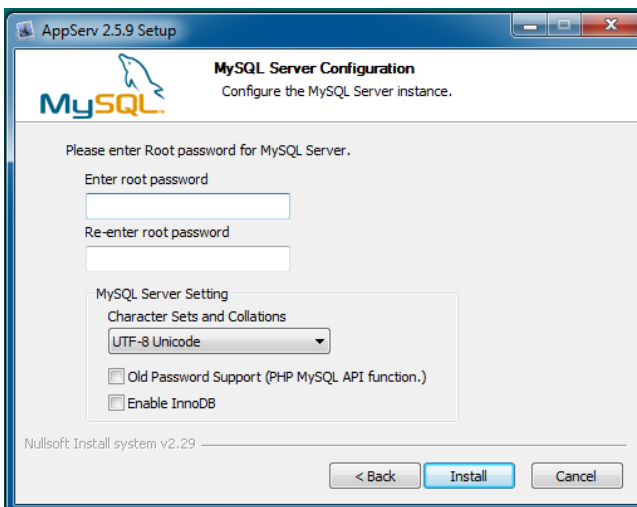
Verificamos la dirección donde se va guardar la carpeta de appserv “**NEXT**”



Seleccionamos los componentes a instalar que para la aplicación se seleccionan todos y presionamos **NEXT**.



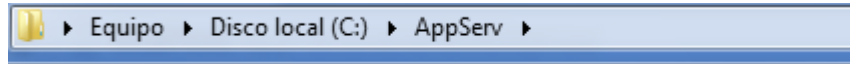
Llenamos lo campos en server name siempre pondremos “**LOCALHOST**” debemos tener en cuenta el puerto el cual se lo puede cambiar en caso que este ocupado por otra aplicación.



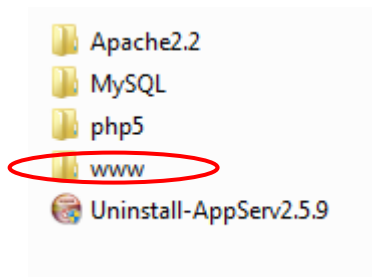
En password ingresamos una contraseña que luego nos va a servir para acceder a la base de datos, una vez puesto una contraseña presionamos “**INSTALL**” .



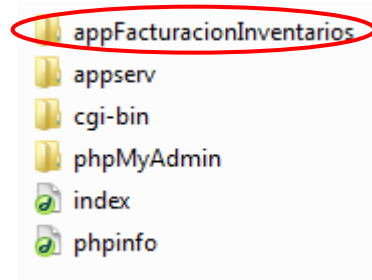
Ya instalada la aplicación nos dirigimos a la dirección donde se instaló el programa.



En la carpeta AppServ encontramos las siguientes carpetas



En la carpeta “**WWW**” debemos pegar nuestro proyecto el cual para mayor facilidad lo hemos llamado “**APPFACTURACIONINVENTARIOS**”.



## CARGAR LA BASE DE DATOS.

Para cargar nuestra base de datos nos dirigimos a la página de **LOCALHOST**



← → ↻ localhost

### The AppServ Open Project - 2.5.9 for Windows


 [phpMyAdmin Database Manager Version 2.10.2](#)  
 [PHP Information Version 5.2.3](#)

[About AppServ Version 2.5.9 for Windows](#)  
AppServ is a merging open source software installer package for Windows includes :

- [Apache Web Server Version 2.2.4](#)
- [PHP Script Language Version 5.2.3](#)
- [MySQL Database Version 5.0.45](#)
- [phpMyAdmin Database Manager Version 2.10.2](#)

- [ChangeLog](#)
- [README](#)
- [AUTHORS](#)
- [COPYING](#)
- <http://www.AppServNetwork.com>

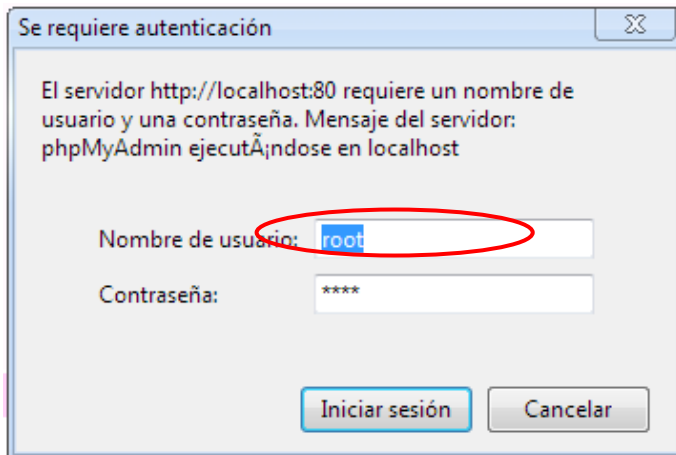
Change Language :  

 [Easy way to build Webserver, Database Server with AppServ :-\)](#)

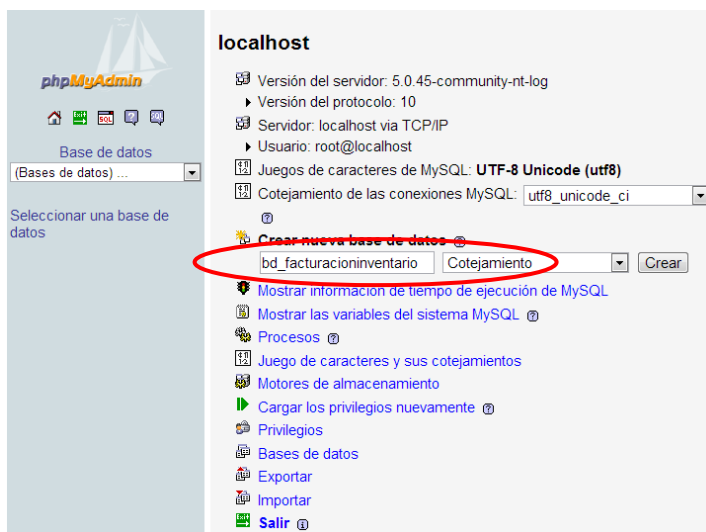
Encontramos la página principal de AppServ Open Project .Aquí nos dirigimos a “PhpMyAdmin”

 [phpMyAdmin Database Manager Version 2.10.2](#)

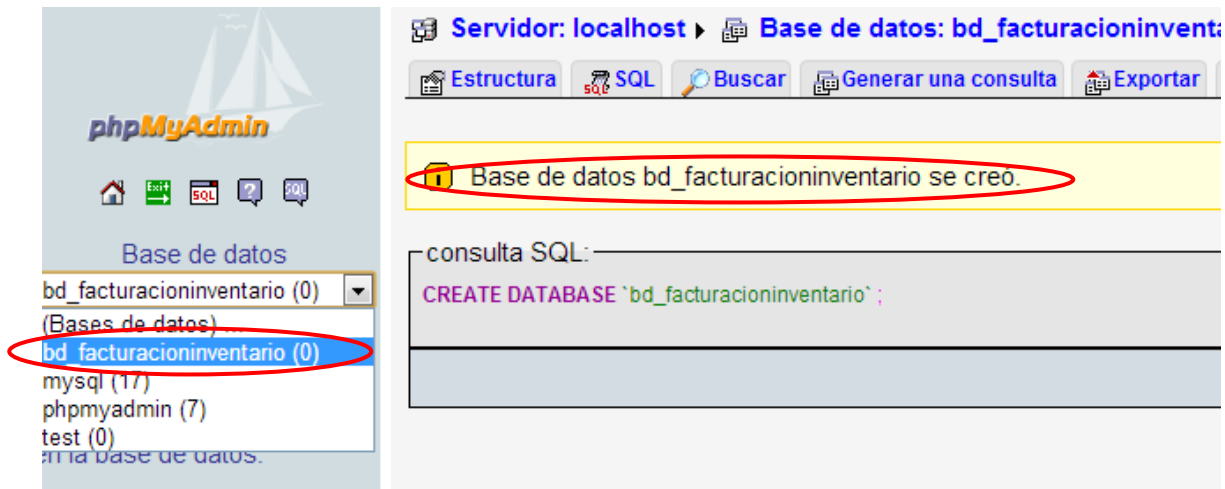
Nos va aparecer una pantalla para ingresar el nombre y la contraseña que asignamos en la instalación del programa.




En la pantalla principal nos vamos a **CREAR NUEVA BASE DE DATOS** y ponemos el nombre de nuestra base de datos” **bd\_facturacioninventario**” y “**CREAR**”



Una vez creada la base nos indica un mensaje de que la base se creó con éxito y podemos ver el nombre de la base de datos en el lado izquierdo.

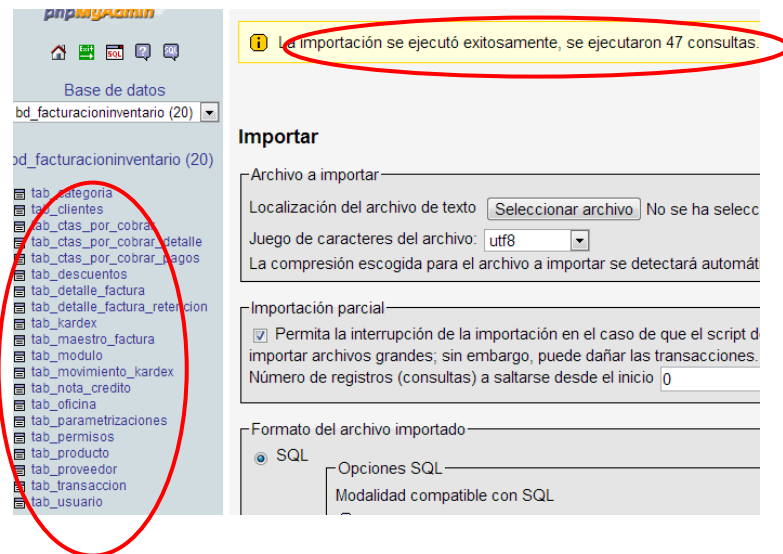


Procedemos a copiar nuestro archivo Zip  bd\_facturacioninventario.sql en una carpeta asegurándonos que este expuesta a daños.

Seleccionamos la base de datos y nos vamos a” **IMPORTAR**” Seleccionamos la base de datos indicada que se encuentra comprimida en el archivo Zip.



Una vez cargada la base de datos con el botón “**SELECCIONAR ARCHIVO**” nos dirigimos a “**CONTINUAR**” y así se cargan las tablas con la información de cada una.



## INGRESO AL SISTEMA

Para el ingreso al sistema nos dirigimos a un navegador lo cuales pueden ser:



Google Chrome.

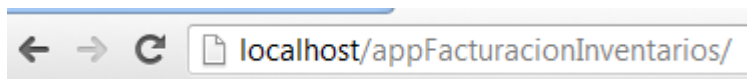
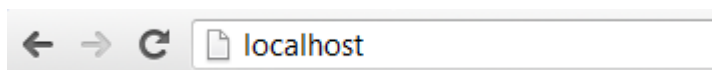


Mozilla Firefox



Internet Explorer

En la ventana que se abra ponemos la dirección de nuestro sistema en el servidor local.



## MANUAL DE USUARIO “Control Paper Store”.

Una vez conectado con el servidor a través del navegador se nos despliega la primera página de **CONTROL DE ACCESO**.



Aquí debemos escribir nuestro nombre de usuario previamente ingresado al sistema y la contraseña adecuada diferenciando mayúsculas y minúsculas.

- a) Si la contraseña es incorrecta se muestra un mensaje de contraseña o usuario incorrectos.
- b) Si la contraseña es correcta se nos despliega el **MENÚ PRINCIPAL**.



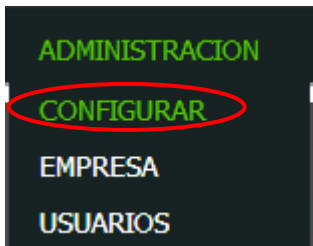
En el menú principal encontramos los diferentes **MODULOS** del sistema podemos estar en cualquier módulo y el botón **HOME** nos regresara siempre al **MENÚ PRINCIPAL**.

## 1. MÓDULO DE ADMINISTRACIÓN.

En este módulo se despliegan las opciones de **CONFIGURAR, EMPRESA, USUARIOS.**



En la opción de **CONFIGURAR** se despliega la pantalla de **ADMINISTRACIÓN DE PARAMETRIZACIONES.**



## ADMINISTRACIÓN DE PARAMETRIZACIONES

### RANGO DE FACTURAS

INICIO	FINAL	ACTUAL	
<input type="text" value="1"/>	<input type="text" value="200"/>	<input type="text" value="150"/>	<input type="button" value="GUARDAR"/>

### PARAMETRIZACIÓN DE (%)RETENCIÓN


(%)RETENCIÓN	<input type="text" value="1"/>	<input type="button" value="GUARDAR"/>
--------------	--------------------------------	--

### PARAMETRIZACIÓN DE DESCUENTOS

NUEVO DESCUENTO

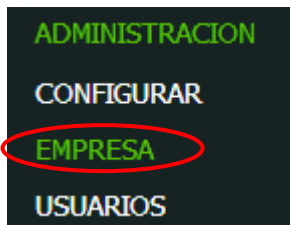
DESCUENTOS EXISTENTES

ELIMINAR	PORCENTAJE
<input type="checkbox"/>	0
<input type="checkbox"/>	5
<input type="checkbox"/>	10
<input type="checkbox"/>	25
<input type="checkbox"/>	50

- En la opción de **RANGO DE FACTURAS** vamos a configurar los factureros que previamente hicimos en una imprenta autorizada por el SRI. Aquí ponemos el rango de nuestras facturas desde la primera hasta la última una vez estemos seguros de la configuración presionamos el botón **GUARDAR**.
- En el botón de **PARAMETRIZACIÓN DE (%) RETENCIÓN** ingresamos el porcentaje de retención que de acuerdo al SRI debemos cobrar en las retenciones. Este valor se debe **GUARDAR** y al momento de facturar con retención se cobrara el valor de la retención de acuerdo a este parámetro ingresado.
- En la opción **PARAMETRIZACIÓN DE DESCUENTOS** tenemos el casillero de “nuevo descuento” aquí ingresamos un porcentaje de descuento al momento de presionar el botón  esta cantidad se guarda en la lista la misma que aparecerá en las facturas para seleccionar, por defecto el descuento será del 0 %.

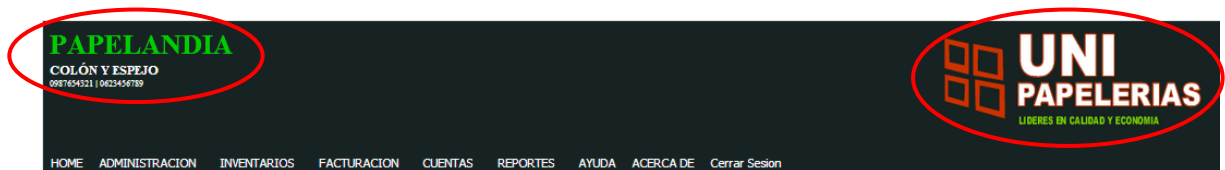


En la opción **EMPRESA** se despliega una pantalla dónde se guarda los datos de cada loca comercial al momento de **GUARDAR** esta información se almacena en la base de datos.

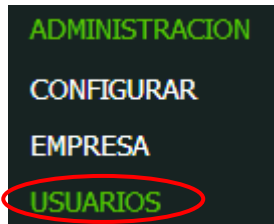


	<a href="#">GUARDAR</a>	<a href="#">SALIR</a>
<b>RUC:</b>	<input type="text" value="1002562369001"/>	
<b>REPRESENTANTE:</b>	<input type="text" value="ALICIA ORTIZ"/>	
<b>NOMBRE EMPRESA:</b>	<input type="text" value="PAPELANDIA"/>	
<b>DIRECCION:</b>	<input type="text" value="LOS CEIBOS"/>	
<b>TELF. MOVIL:</b>	<input type="text" value="0987654321"/>	
<b>TELF. CONVENCIONAL:</b>	<input type="text" value="0623456789"/>	
<b>ESLOGAN:</b>	<input type="text" value="TU MEJOR PROVEEDOR EN PAPELES"/>	
<b>CORREO:</b>	<input type="text" value="supapeleria@correo.com"/>	

Una vez guardada la información los cambios en la información aparecen en el banner principal con el logo de UNI PAPELERÍAS.



En la opción **USUARIOS** vamos a ingresar a la pantalla del listado de usuarios.



USUARIOS						
Busqueda:						
1 - 3 de 3 <span style="float: right;">1 Ir</span>						
OPCIONES	CEDULA	EMPLEADO	USUARIO	TELEFONOS	DIRECCION	ESTADO
	1002562369	ADMINISTRADOR ADMINISTRADOR	ADMINISTRADOR	093439788 - 062951355	ADMINISTRADOR	ACTIVO
	2002002002	CARLOS VALENCIA	cvalencia	0987654321 - 0987654321	LA VICTORIA	ACTIVO
	0401593421	CARLOS JULIO LANDAZURI ORTIZ	clandazuri	093623258 - 062609674	IBARRA	INACTIVO

Aquí puede ver la información de los usuarios que van a tener acceso al sistema. Podemos **ACTIVAR** o **inactivar** a un usuario. En el espacio de búsqueda se encuentra un usuario específico en, si se desea agregar un nuevo usuario presionamos el botón de **NUEVO USUARIO**.





Al agregar un nuevo usuario debemos llenar los campos requeridos, una vez ingresado todos los campos presionamos **GUARDAR**.

NUEVO USUARIO	
CEDULA:	<input type="text"/>
NOMBRES:	<input type="text"/>
APELLIDOS:	<input type="text"/>
TELF. MOVIL:	<input type="text"/>
TELF. CONVENCIONAL:	<input type="text"/>
CORREO:	<input type="text"/>
DIRECCION:	<input type="text"/>
USUARIO:	<input type="text"/>
CLAVE:	<input type="text"/>

Para dar los respectivos permisos de usuario a cada uno en el listado de clientes presionamos el botón de **PERFIL**



Aquí nos muestra las opciones de PRIVILEGIOS EXISTENTES señalamos con el signo  donde se van apilando los diferentes permisos estos se agrupan en PRIVILEGIOS ASIGNADOS si se desea quitar el permiso se presiona 

PRIVILEGIOS ASIGNADOS			PRIVILEGIOS EXISTENTES	
OP	NOMBRE	DETALLE	OP	DETALLE
	CONFIGURAR	CONFIGURACION		USUARIOS
	EMPRESA	DATOS GENERALES DE LA EMPRESA		ADMINISTRACION DE USUARIOS
	CTA-COBRAR	CUNETAS X COBRAR		
	CTA-PAGAR	CUNETAS X PAGAR		
	ANULAR FACTURAS	REGISTRO DE DEVOLUCION VENTAS		
	CLIENTES	ADMINISTRACION DE CLIENTES		
	FACTURACION	EMISION DE FACTURAS		
	PROVEEDORES	ADMINISTRACION DE PROVEEDORES		
	PRODUCTOS	ADMINISTRACION DE PRODUCTOS		
	CATEGORIAS	ADMINISTRACION DE CATEGORIAS		
	REP-VT-PRODUCTO	REPORTE DE VENTAS TOTALES POR PRODUCTO		
	REP-VT-FACTURAS	REPORTE DE VENTAS TOTALES POR FACTURA		
	REP-VENTAS-USUARIO	REPORTE DE VENTAS POR USUARIO		
	REP-VENTAS-PRODUCTO	REPORTE DE VENTAS POR PRODUCTO		
	REP-VENTAS-CLIENTE	REPORTE DE VENTAS POR CLIENTE		
	REP-VENTAS	REPORTE VENTAS		
	REP-PROVEEDORES	REPORTE PROVEEDORES		
	REP-PRODUCTOS	REPORTES-PRODUCTOS		
	REP-FACTURAS	REPORTE FACTURAS		
	REP-CXC	REPORTE DE CUENTAS POR COBRAR		

## 2. MÓDULO DE INVENTARIOS.

En este módulo se despliegan las opciones de **CATEGORÍAS, PRODUCTOS Y PROVEEDORES.**





En la opción **CATEGORIAS** se muestra una pantalla principal con el listado de las categorías existentes.



**ADMINISTRACION DE CATEGORÍAS**

Busqueda:   

	CODIGO	NOMBRE	DESCRIPCION	ESTADO
  	1	PAPEL	TODA CLASE DE PAPELES	A
  	2	CUADERNOS	CLASES DE CUADERNOS	A
  	3	CINTAS	CLASES DE CINTAS	A
  	4	CARPETAS	CLASES DE CARPETAS	A
  	5	LAMINAS ILUSTRATIVAS	LAMINAS ILUSTRATIVAS	A
  	6	LIBROS EDUCATIVOS	LIBROS EDUCATIVOS	A
  	7	MARCADORES	MARCADORES	A
  	8	ESFEROS	TIPOS DE ESFEROS	A
  	9	CARTULINAS	TIPOS DE CARTULINAS	A
  	10	FOMIX	CLASES DE FOMIX	A

Aquí se muestra la **ADMINISTRACIÓN DE CATEGORÍAS** con las categorías existentes en el sistema. Podemos **ACTIVAR**  o **inactivar**  una categoría. En el espacio de búsqueda se encuentra una categoría específica, si

se desea agregar una nueva categoría presionamos el botón de **NUEVA CATEGORIA**.

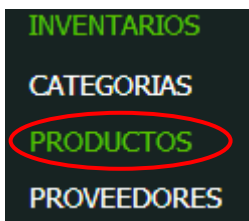
---

Busqueda:   

Al agregar una nueva categoría debemos llenar los campos requeridos, una vez ingresado todos los campos presionamos **GUARDAR**.

		<b>GUARDAR</b>	<b>CANCELAR</b>
<b>CODIGO:</b>		22	
<b>NOMBRE:</b>		<input type="text"/>	
<b>DESCRIPCION:</b>		<input type="text"/>	





En la opción **PRODUCTOS** se muestra una pantalla principal con el listado de los productos existentes.



Busqueda:

ID	BARRAS	NOMBRE	DESCRIPCION	COSTO	STOCK	MIN.	MAX.	UNIDAD	ESTADO
1	0045	PAPEL A5 BRILLANTE X 10	FUNDITA COLORES	0.35	7	10	100	DOCENA	A
2	0046	PAPEL A5 BRILLANTE X 12	FUNDITA DE COLORES	0.19	1044	100	200	UNIDAD	A
3	0047	CARPETA CARTÓN NACIONAL	COLORES VARIOS	1.27	952	10	20	UNIDAD	A
4	0048	CARPETA CARTÓN MANILA	COLORES VARIOS	0.22	211	10	20	UNIDAD	A
5	0049	CARPETA FOLDER TAPA TRANSPA A4	COLORES VARIOS	0.56	2	10	20	UNIDAD	A
6	0050	CARPETA PLASTICA A4 CARIOCA	COLORES VARIOS	0.21	318	10	20	CINCUENTA	A
7	7512442132223	BORRADOR TINTA ARTESCO PEQ	PEQUEÑO BICOLOR	0.03	167	100	200	UNIDAD	A
8	0051	CARPETA TIPO SOBRE MOTIVOS	IMPORTADAS	0.19	100	10	100	UNIDAD	A
9	0052	CARPETA TIPO SOBRE PEQUEÑA	MOTIVOS VARIOS	1	50	10	100	UNIDAD	A
10	0053	CARPETA TIPO SOBRE PLASTICA	CON BROCHE	0.12	60	10	100	DECENA	A

Aquí se muestra la **ADMINISTRACIÓN DE PRODUCTOS** con los productos existentes en el sistema pudiendo realizar las siguientes opciones.

- Con la opción  podemos **EDITAR** la información ingresada de un producto excepto el código de barras ya que ese código es único de cada producto.
- Podemos **ACTIVAR**  o **inactivar**  un producto.
- En el Icono de dólar  asignamos un **PORCENTAJE DE GANANCIA** a un producto ingresado.

GUARDAR

CANCELAR

**CODIGO BARRAS:**



**CODIGO BARRAS:**

7703064447509

**NOMBRE:**

BORRADOR PELIKAN PZ60

**DESCRIPCION:**



PZ60

**(%) DE GANANCIA:**

40

*El registro de los valores de precios se debe realizarlo con porcentajes (%)*

- d. En el botón de **KARDEX** analizamos los movimientos de compra y venta de los productos en un lapso de tiempo establecido con en el parámetro **FECHA DE INICIO Y FECHA FINAL** una vez establecidas las fechas aceptamos con el botón **CARGAR**.

KARDEX GENERADO POR PRODUCTO											
CODIGO BARRAS:			NOMBRE: BORRADOR TINTA ARTESCO PEQ								
			STOCK ACTUAL: 167								
			COSTO KARDEX: 0.03								
FECHA INICIO:		2013/01/01	FECHA FINAL:		2013/01/18						
FECHA	MOVIMIENTO	DOCUMENTO	COMPRAS			VENTAS			EXISTENCIA		
			CANT.	V.U.	V.T.	CANT.	V.U.	V.T.	CANT.	V.U.	V.T.
2013-01-04	VENTAS	FAC-VTA-13	0	0	0	2	0.03	0.06	167	0.03	5.01
2013-01-04	VENTAS	FAC-VTA-12	0	0	0	30	0.03	0.9	169	0.03	5.07
2013-01-03	COMPRAS	FAC-COMP-ESTADO INICIAL	100	0.11	11	0	0	0	199	0.03	6.62
2013-01-03	COMPRAS	FAC-COMP-ESTADO INICIAL	50	0.1	5	0	0	0	99	-0.04	-4.38
2013-01-03	COMPRAS	FAC-COMP-	100	0.1	10	0	0	0	49	-0.19	-9.38

- e. En el espacio de búsqueda se encuentra un producto específico, si se desea agregar un nuevo producto presionamos el botón de **NUEVO PRODUCTO**.

Busqueda:   

Al agregar un nuevo producto debemos llenar los campos requeridos. En el espacio de código de barras registramos los códigos que por fabrica vienen en los productos en caso de que el producto no tenga código de barras asignamos un código propio para cada producto. Una vez ingresado todos los campos presionamos **GUARDAR**.

GUARDAR

CANCELAR

PROVEEDOR:

CATEGORIA:

CODIGO BARRAS:

IMG CODIGO BARRAS:



NOMBRE:

DESCRIPCION:

UNIDAD DE MEDIDA:

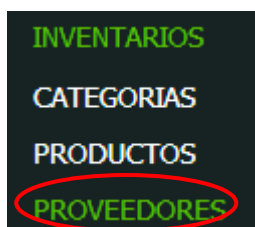
CONVERTIBILIDAD:

STOCK MINIMO:

STOCK MAXIMO:

APLICA IVA:

En la opción **PROVEEDORES** se muestra una pantalla principal con el listado de los proveedores existentes.







Busqueda:   

1 - 10 de 10

	CODIGO RUC	RAZON SOCIAL	REPRESENTANTE	DIRECCION	TELEFONOS	ESTADO
  	1091737236001	SARI PAPELERÍA POPULAR S.A	SARA RODRIGUEZ	AV MARIANO ACOSTA 12-51	062606287	A
  	1790819515001	DILIPA	COMPAÑIA LIMITADA	OBISPO MOSQUERA 5-53	062601686   022418640	A
  	1306790708001	LLANZA	LEDIS LOOR	DIEGO DE GUEVARA OE 2-06	022660830   0997443617	A
  	1001718848001	DISTRIBUIDORA DM	ANGÉLICA MORALES	CHORLAVI 1-85 Y FRAY VACAS	062641792	A
  	1001449956001	PROVESUM S CIA	CARLOS MERA AGUAS	CHICA NARVAEZ 7-24	062959727	A
  	1001218427001	MEGA MONSERRATH	SILVIO HARO	OTAVALO	062920681	I
  	1706650312001	COMERCIAL SAN REMO	PATRICIO DOMINGUEZ	MACHACHI	022314310	A
  	10000000000	ANDES PAPELERÍA	DAVID ANDRADE	OTAVALO ECUADOR	062922621	A
  	1000000000001	COMERCIAL LANDAZURI	LUIS LANDAZURI	MIRES N 6-44	022584829   0997441007	A
  	1001767506001	FANTASÍAS ESTEFANY	EDGAR ALMEIDA	BARTOLOMÉ GARCÍA 1-90	062951006	A

Aquí se muestra la **ADMINISTRACIÓN DE PROVEEDORES** con los proveedores registrados en el sistema. Podemos **ACTIVAR**  o **inactivar**  un proveedor. En el espacio de búsqueda se encuentra un proveedor específico, si se desea agregar un nuevo proveedor presionamos el botón de **NUEVO PROVEEDOR**.

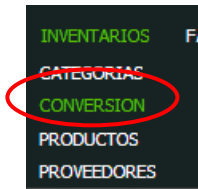
Busqueda:   

Al agregar una nuevo proveedor debemos llenar los campos requeridos, una vez ingresado todos los campos presionamos **GUARDAR**.

GUARDAR
CANCELAR

<b>RUC:</b>		<input style="width: 90%;" type="text"/>
<b>RAZON SOCIAL:</b>		<input style="width: 90%;" type="text"/>
<b>REPRESENTANTE:</b>		<input style="width: 90%;" type="text"/>
<b>TELF LOCAL:</b>		<input style="width: 90%;" type="text"/>
<b>TELF MOVIL:</b>		<input style="width: 90%;" type="text"/>
<b>DIRECCION:</b>		<input style="width: 90%;" type="text"/>
<b>CORREO:</b>		<input style="width: 90%;" type="text"/>

- f. En la opción **CONVERSIÓN** podemos transformar un producto que ha sido registrado como cajas a unidades.



Aquí debemos seleccionar el producto que se va a vender por caja y en la siguiente opción el producto que vamos a agregar las unidades.

CONVERSION DE PAQUETE A UNIDADES CONVERTIR VOLVER

PACKS						UNIDADES			
PRODUCTO PACK: CAJA						PRODUCTO UNIDAD: MARCADOR ESCOLA			
SELECCION	CODIGO BARRAS	NOMBRE	STOCK	CONVER	DIVISIBLE	SELECCION	CODIGO BARRAS	NOMBRE	STOCK
✓	8060606060	CAJA DE MARCADOR CARIOCA	1	12	SI	✓	7861129511180	MARCADOR ESCOLAR CARIOCA X12	152
						✓	7861129511173	MARCADOR ESCOLAR CARIOCA X6	64
						✓	7703064732605	MARCADOR ESCOLAR CAROSELLO X6	79

Una vez seleccionado el origen y el destino presionamos **“convertir”**

Y nos aparece una pantalla con la información del stock los productos

GENERACION DE CALCULOS DE CONVERSION

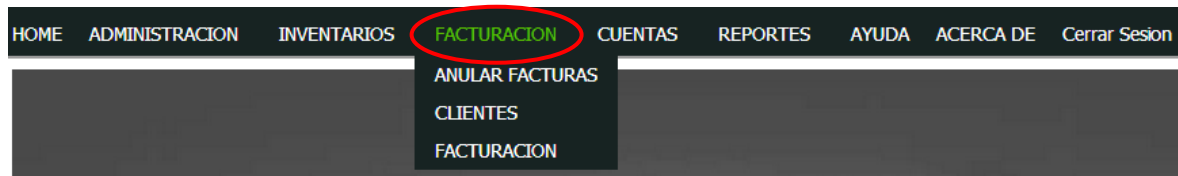
CANTIDAD	CONVERTIBILIDAD	TOTAL CONVERSION
	12	0
STOCK ORIGEN		STOCK DESTINO
		152

**GUARDAR**

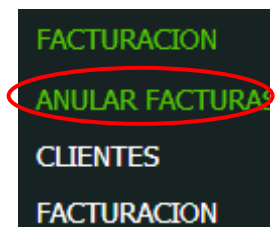
Cuando presionamos **“Guardar”** se pasa el stock de mercadería seleccionado de un producto al otro.

### 3. MÓDULO DE FACTURACIÓN.

En este módulo se despliegan las opciones de **ANULACIÓN DE FACTURAS, CLIENTES FACTURACIÓN.**



En la opción **ANULAR FACTURAS** se despliega el menú **GENERAR DEVOLUCIÓN EN VENTAS**



A screenshot of the 'GENERAR DEVOLUCION EN VENTAS' form. The form has a pink header with the title 'GENERAR DEVOLUCION EN VENTAS'. Below the header, there are several input fields: 'NRO FACTURA:' with a search icon, 'CEDULA|RUC', 'CORREO ELECTRONICO', 'FECHA REGISTRO', 'NOMBRES', and 'TELEFONOS'. At the bottom, there is a table with columns for 'CODIGO', 'NOMBRES', 'VALOR UNITARIO', 'CANTIDAD', and 'VALOR TOTAL'. To the right of the table, there are summary rows: 'SubTotal', 'SubTotalNoIva', 'Descuento(%)', 'Valor descuento', 'Iva(12%)', and 'Total', each with a corresponding input field.

- a. En la opción **NRO FACTURA** ingresamos el número de una factura existente, automáticamente el sistema carga la información del cliente y el detalle de la factura.

HOME ADMINISTRACION INVENTARIOS FACTURACION CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

**GENERAR DEVOLUCION EN VENTAS**

NRO FACTURA: 2

CEDULA/RUC: 0400345786      NOMBRES: ALBERTO MARTINEZ

CORREO ELECTRONICO: albmart@hotmail.com      TELEFONOS: 0993456787 |

FECHA REGISTRO: 2012-12-27      NRO FACTURA: 2

**ANULAR**

APLICA NOTA DE CREDITO  
SI:  NO:

CODIGO	NOMBRES	VALOR UNITARIO	CANTIDAD	VALOR TOTAL
1	PAPEL BRILLANTE	1.89	8	15.12
2	PAPEL BOND A4	0.20	50	10.00
3	PAPEL PERIODICO A4	0.10	100	10.00
4	CUADERNO DE 200 HOJAS ACADEMIC	3.49	8	27.92

SubTotal 27.92

SubTotalNoIva 35.12

Descuento(%) 1.80

Valor descuento 0.15

Iva(12%) 3.35

Total 59.89

- b. Una vez cargada la factura escogemos si deseamos **APLICAR NOTA DE CRÉDITO O ANULAR FACTURA** al seleccionar la opción **NO** y el botón **ANULAR** la mercadería vuelve a los inventarios y el número de factura queda anulada.

**ANULAR**

APLICA NOTA DE CREDITO  
SI:  NO:

HOME ADMINISTRACION INVENTARIOS FACTURACION CUENTAS

LA FACTURA FUE ANULADA CON ÉXITO.

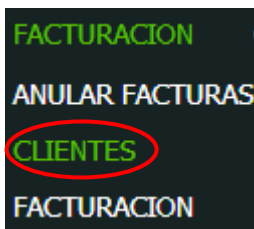
- c. En la opción **APLICAR NOTA DE CRÉDITO** se genera una nota de crédito autorizada por el SRI se imprime los detalles de la nota de crédito y la mercadería vuelve al inventario.

**ANULAR**

**APLICA NOTA DE CREDITO**

**SI:**  **NO:**

- En la opción **CLIENTES** se muestra una pantalla principal con el listado de los clientes existentes.



**CLIENTES**

Busqueda:

1	2	3	4	1 - 10 de 40		1	lr
CEDULA/RUC	NOMBRES	TELEFONOS	DIRECCION	MAIL	ESTADO		
	0400345786	ALBERTO MARTINEZ	0993456787 - QUITO 3-56 Y AMBATO	albmart@hotmail.com	INACTIVO		
	1002562369001	FAUSTO LOPEZ	0987654567 - 0990967677 AZAYA	cpga@hotmail.com	ACTIVO		
	1001001001011	PAUL ORDOÑEZ	098712345 - 067897897 YACUCALLE	paul@jhkgasdkj.com	ACTIVO		
	1001768934001	FERRERIA FERROLEG	062609194 - COTACACHI		ACTIVO		
	0400019216	XAVIER BENAVIDEZ	2609447 - LITA		ACTIVO		
	1007867879	RICARDO ROSALES	2609134 - IBARRA		ACTIVO		
	1002567865001	ROSA UBIDIA CERÓN	2609876 - COTACACHI		ACTIVO		
	1006789788	FELIPE ARAGÓN MARCILLO	26093484 - IBARRA		ACTIVO		
	1003597554	ARACELY PABÓN	- ABELARDO MONCAYO		ACTIVO		
	1060011540001	CEB 5 DE JUNIO	2683171 - PABLO ARENAS URCUQUI		ACTIVO		

Aquí se muestra la **ADMINISTRACIÓN DE CLIENTES** con el listado de los clientes ingresados anteriormente. Podemos **ACTIVAR** o **inactivar** un cliente para que no parezca en la búsqueda. En el espacio de búsqueda se encuentra una categoría específica, si se desea agregar una nueva cliente presionamos el botón de **NUEVO**.

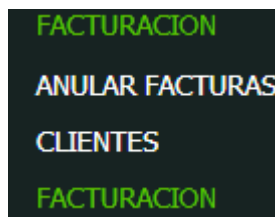
Busqueda:   

Al agregar un nuevo cliente debemos llenar los campos requeridos, una vez ingresado todos los campos presionamos **GUARDAR**.

## NUEVO CLIENTE

CEDULA / RUC:	<input type="text" value="0401593421"/>
NOMBRES:	<input type="text" value="CARLOS JULIO"/>
APELLIDOS:	<input type="text" value="LANDÁZURI ORTIZ"/>
DIRECCION 1:	<input type="text" value="JUAN HERNANDEZ"/>
DIRECCION 2:	<input type="text" value="ESPEJO Y COLON"/>
CORREO:	<input type="text" value="cjllandazuri@hotmail.com"/>
TELF 1:	<input type="text" value="2643210"/>
TELF 2:	<input type="text" value="2609193"/>

- En la opción **FACTURACIÓN** podemos realizar una transacción comercial es aquí dónde nos aparece la pantalla principal para **FACTURAR PRODUCTOS**



HOME ADMINISTRACION INVENTARIOS FACTURACION CUENTAS REPORTES AYUDA ACERCA DE Cerrar Sesión

**FACTURAR PRODUCTOS**

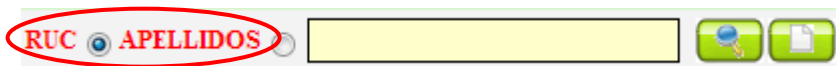
RUC  

...SELECCIONE UN CLIENTE POR FAVOR...

PRODUCTO:  

CODIGO	NOMBRE	% APLICADO	STOCK	PRECIO	CANTIDAD	TOTAL
CODIGO	NOMBRES		VALOR UNITARIO		CANTIDAD	VALOR TOTAL
					SubTotal	0
					SubTotal(0%)	0
					Descuento(%)	0
					Valor descuento	0
					Iva(12%)	0
					Total	0

- a. Aquí encontramos un botón de búsqueda de Usuarios pudiendo escoger si buscar por **RUC** o cédula por **APELLIDOS** en caso de un cliente no registrado nos vamos al ícono de **NUEVO** y nos conectamos con el módulo de **CLIENTES** y agregamos un nuevo cliente.



- b. Una vez se carguen o se ingresen los datos del cliente en el espacio de búsqueda de producto buscamos entre los ingresados y que tengan stock pudiendo ver un listado y consultar su valor unitario antes de agregarlo a la factura.

**FACTURAR PRODUCTOS**

GUARDAR CANCELAR

RUC  APELLIDOS

...SELECCIONE UN CLIENTE POR FAVOR...

PRODUCTO

CODIGO	NOMBRE	STOCK	PRECIO	CANTIDAD	TOTAL	
0005	ESFERO BIC PUNTA FINA AZUL	440	0.3		0	+
0006	ESFERO BIC PUNTA FINA NEGRO	363	0.3		0	+
0001	ESFERO BIC PUNTA MEDIA AZUL	650	0.25		0	+
0003	ESFERO BIC PUNTA MEDIA NEGRO	642	0.25		0	+
0002	ESFERO BIC PUNTA MEDIA ROJO	386	0.25		0	+

CODIGO      NOMBRES      VALOR UNITARIO      CANTIDAD      VALOR TOTAL

- Con el botón agregamos el producto seleccionado a la nueva factura siempre verificando que el campo **CANTIDAD** sea el correcto.
- Escogemos el **DESCUENTO** que se va aplicar a ese cliente si no se aplica descuento por defecto es el 0 %.

Una vez seleccionados todos los productos de la lista ponemos **GUARDAR**

- c. A continuación en esta nueva pantalla nos muestra el resumen de la factura antes de imprimirse.

PAPELANDIA		FACTURA			
LOS CEIBOS		150			
TU MEJOR PROVEEDOR EN PAPELES					
DATOS DEL CLIENTE					
CEDULA/RUC		NOMBRES			
1003934310		WILLMER ERAZO			
DIRECCION		TELEFONOS			
IBARRA		0939971543			
DETALLE DE COMPRA					
CODIGO	NOMBRES	VALOR UNITARIO	CANTIDAD	TOTAL	
0005	ESFERO BIC PUNTA FINA AZUL	0.3	2	0.6	
0048	CARPETA CARTÓN MANILA	0.31	2	0.62	
7861084214164	CUADERNO COSIDO 4LINEAS ESCRIB	1.26	1	1.26	
0099	BORRADOR EN LÁPIZ	0.7	3	2.1	
8690826360208	LAPIZ ADEL NEGRO	0.5	1	0.5	
0079	BORRADOR ARTESCO PEQ	0.1	1	0.1	
0057	BORRADOR ESTAEDLER TRENCITO	0.35	2	0.7	
0093	FORRO PLASTICO ACADEMICO SENCI	0.41	2	0.82	
7750082007093	CRAYÓN ARTESCO ESTUCHE	1.75	1	1.75	
7702163140793	PLASTILINA PARCHECITOS	1.15	1	1.15	
0052	CARPETA TIPO SOBRE PEQUEÑA	1.4	1	1.4	
0075	GOMA BIOPLAST 260 GR	0.76	1	0.76	
				SUBTOTAL	11.76
				SUBTOTAL NO IVA	0
				DESCUENTO (%)	0
				VALOR DESCUENTO	0
				IVA 12%	1.41
				TOTAL	13.17
				RETENCION	0
				NOTA DE CREDITO	0
				TOTAL A COBRAR	13.17
				APLICA RETENCION:	<input type="checkbox"/>
				APLICA NOTA DE CREDITO:	<input type="checkbox"/>

Aquí podemos verificar el número de factura que debe coincidir con el número de la factura física. Una vez comprobado el número de factura ponemos **GUARDAR** para imprimir la factura.



FACTURA  
Aut. S.R.L: 1111669671  
Landázuri Ortiz Carlos Julio  
RUC: 0401593421001  
Dir.: Eugenio Espejo y  
Cristóbal Colón Telf: 062 609 193  
Ibarra - Ecuador

001 - 001  
Nº 00000150

CLIENTE:  
Cédula/RUC: \_\_\_\_\_ Teléfono: \_\_\_\_\_  
Dirección: \_\_\_\_\_ Fecha: \_\_\_\_\_

Cantidad	Detalle	P. Unitario	P. Total

- d. Si el cliente va a pagar con una **NOTA DE CRÉDITO** o debemos seleccionar la opción que se encuentra en la parte inferior de nuestra pantalla.

APLICA RETENCION:  **APLICA NOTA DE CREDITO:**

Aquí se muestra una pantalla dónde en el botón debemos ingresar el numero de nota de crédito para q coincida con los emitidos en el local y así el documento quede anulado.

Caso contrario se emite la nota de crédito y se devuelve el dinero.

**APLICA NOTA DE CREDITO**

CEDULA/RUC: 1003934310

NOMBRES: WILLMER ERAZO

TOTAL FACTURA: 13.17

NRO. NOTA DE CREDITO:

Eliminar	Nro. Nota Crédito	Valor
--	--	--
TOTAL		0

- e. Si el cliente va a pagar con una retención se selecciona la opción de **APLICAR RETENCIÓN.**

**APLICA RETENCION:**  **APLICA NOTA DE CREDITO:**

Aquí se muestra los datos del comprador con el total de su factura y el porcentaje que se va a retener. Ingresamos el número de la retención física y guardamos. Automáticamente en la factura se resta el porcentaje retenido.

CEDULA/RUC	1003934310
NOMBRES	WILLMER ERAZO
TOTAL FACTURA	13.17
IVA FACTURA	1.41
NRO. RETENCION	
RETENCION (%)	1
VALOR RETENCION	0.01

Cualquiera sea la opción escogida c,d,e se mostrará la factura lista para imprimir .

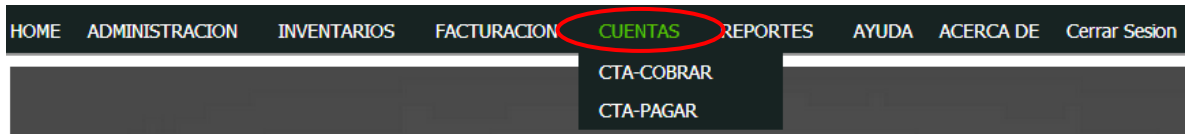
CLIENTE: WILLMER ERAZO	CLIENTE: WILLMER ERAZO
CEDULA/RUC: 1003934310	CEDULA/RUC: 1003934310
DIRECCION: IBARRA	DIRECCION: IBARRA
TELEFONO: 0939971543 /	TELEFONO: 0939971543 /
FECHA: 2012-12-26	FECHA: 2012-12-26

QTY	DESCRIPCION	UNIDAD	PRECIO UNITARIO	PRECIO TOTAL	QTY	DESCRIPCION	UNIDAD	PRECIO UNITARIO	PRECIO TOTAL
2	CARPETA CARTÓN MANILA		0.31	0.62	2	CARPETA CARTÓN MANILA		0.31	0.62
1	CARPETA TIPO SOBRE PEQUEÑA		1.40	1.40	1	CARPETA TIPO SOBRE PEQUEÑA		1.40	1.40
2	ESFERO BIC PUNTA FINA AZUL		0.30	0.60	2	ESFERO BIC PUNTA FINA AZUL		0.30	0.60
1	LAPIZ ADEL NEGRO		0.50	0.50	1	LAPIZ ADEL NEGRO		0.50	0.50
1	PLASTILINA PARCHECITOS		1.15	1.15	1	PLASTILINA PARCHECITOS		1.15	1.15
1	CRAYÓN ARTESCO ESTUCHE		1.75	1.75	1	CRAYÓN ARTESCO ESTUCHE		1.75	1.75
2	BORRADOR ESTADLER TRENCITO		0.35	0.70	2	BORRADOR ESTADLER TRENCITO		0.35	0.70
1	GOMA BIOPLAST 260 GR		0.76	0.76	1	GOMA BIOPLAST 260 GR		0.76	0.76
1	BORRADOR ARTESCO PEG		0.10	0.10	1	BORRADOR ARTESCO PEG		0.10	0.10
2	FORRO PLASTICO ACADEMICO SENCI		0.41	0.82	2	FORRO PLASTICO ACADEMICO SENCI		0.41	0.82
1	CUADERNO COSIDO ALINEAS ESCRIB		1.26	1.26	1	CUADERNO COSIDO ALINEAS ESCRIB		1.26	1.26
3	BORRADOR EN LAPIZ		0.70	2.10	3	BORRADOR EN LAPIZ		0.70	2.10

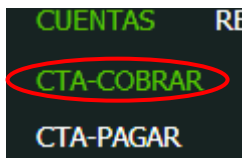
SUBTOTAL(0%)	0.00	SUBTOTAL(0%)	0.00
SUBTOTAL(12%)	11.76	SUBTOTAL(12%)	11.76
DESCUENTOS(%)	0.00	DESCUENTOS(%)	0.00
SUBTOTAL DSCT	0.00	SUBTOTAL DSCT	0.00
IVA(12%)	1.41	IVA(12%)	1.41
TOTAL	13.17	TOTAL	13.17

#### 4. MÓDULO DE CUENTAS.

En este módulo se despliegan las opciones de **CTA-COBRAR Y CTA-PAGAR**



- En la opción de **CUENTAS POR COBRAR** se registran las cuentas que se han dado a crédito estableciendo el número de cuotas y las fechas de pago.




En el espacio de búsqueda se ingresa el usuario para consultar su estado de cuenta.

A screenshot of the search results table for 'CUENTAS POR COBRAR'. The table has columns for 'OP', 'CEDULA', 'NOMBRES', and 'APELLIDOS'. There are five rows of data, each with a checked checkbox in the 'OP' column.

OP	CEDULA	NOMBRES	APELLIDOS
<input checked="" type="checkbox"/>	0400345786	ALBERTO	MARTINEZ
<input checked="" type="checkbox"/>	1002562369001	FAUSTO	LOPEZ
<input checked="" type="checkbox"/>	1001001001011	PAUL	ORDOÑEZ
<input checked="" type="checkbox"/>	1001768934001	FERRETERIA	FERROLEG
<input checked="" type="checkbox"/>	0400019216	XAVIER	BENAVIDEZ

Con la opción  seleccionamos el cliente.


Datos Cliente 			
Cedula	Cliente	Telefono	Direccion
1001001001011	PAUL ORDOÑEZ	098712345	YACUCALLE
<input type="button" value="Nueva Cuenta"/>			

- En el botón **NUEVA CUENTA** creamos una nueva cuenta para el cliente seleccionado.

Se despliega una pantalla con la información de las cuentas por cobrar

DATOS CLIENTE			
NOMBRE:	PAUL	CEDULA:	1001001001011
APELLIDOS:	ORDOÑEZ	TELEFONO 1:	098712345
DIRECCION:	YACUCALLE 1-23	TELEFONO:	067897897

		<input type="button" value="GUARDAR"/>	<input type="button" value="CANCELAR"/>
Datos del Crédito de Compra			
Fecha Registro:	<input type="text" value="2013-01-21"/>		
Total Cuenta :	<input type="text"/>	<input type="button" value="...."/>	
Numero Cuotas:	<input type="text" value="0"/>		
Abono Inicial:	<input type="text" value="0"/>		
Valor cuotas:	<input type="text"/>		
Saldo:	<input type="text"/>		
usuario:	<input type="text" value="1002562369"/>		

En TOTAL CUENTA buscamos en los registros de cuentas si el cliente tiene un adeuda pendiente si es asi con el botón  se suma el valor a la nueva cuenta.

DATOS CLIENTE			
NOMBRE:	PAUL	CEDULA:	1001001001011
APELLIDOS:	ORDOÑEZ	TELEFONO 1:	098712345
DIRECCION:	YACUCALLE 1-23	TELEFONO:	067897897
<input type="button" value="CANCELAR"/>		<input type="button" value="GUARDAR"/>	
PAGOS CUENTAS POR COBRAR			
OP	NUM. FACUTRA	TOTAL FACTURA	TIPO FACTURA
	8	11.88	F

Luego de GUARDAR seleccionamos entre las facturas a nombre el cliente y



la cargamos así se crea una nueva cuenta

Datos Cliente							
Cedula	Cliente	Telefono	Direccion				
0400019216	XAVIER BENAVIDEZ	2609447	LITA				
Datos Cuenta							
OP	Cod. Cuenta	Abono Inicial	N. Cuotas	N Cuotas Pendientes	Saldo	Total Deuda	Fecha Registro
	4	16.63	2	0	0.00	36.63	2013-01-04
Datos Pagos							
Cod. Cuenta	Numero Cuotas	Numero Cuotas Pendientes	Total Deuda	Pago	Fecha Registro		
4	2	0	36.63	10.00	2013-01-04		
4	2	0	36.63	10.00	2013-01-04		

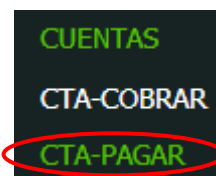
a. Para Consultar el valor de la deuda debe presionar




b. Si el usuario va a cancelar el total de la duda deberá presionar



- En la opción de **CUENTAS POR PAGAR** se registran las cuentas que se deben a los diferentes proveedores.






## CUENTAS POR PAGAR

BUSCAR PROVEEDOR:  

En el espacio de búsqueda se ingresa el proveedor para consultar las cuentas pendientes con él.

OP	CEDULA/RUC	RAZON SOCIAL	TELEFONO	DIRECCION
<input checked="" type="checkbox"/>	1091737236001	SARI PAPELERÍA POPULAR S.A	062606287/	AV MARIANO ACOSTA 12-51
<input checked="" type="checkbox"/>	1790819515001	DILIPA	062601686/022418640	OBISPO MOSQUERA 5-53
<input checked="" type="checkbox"/>	1306790708001	LLANZA	022660830/0997443617	DIEGO DE GUEVARA OE 2-06
<input checked="" type="checkbox"/>	1001718848001	DISTRIBUIDORA DM	062641792/	CHORLAVI 1-85 Y FRAY VACAS
<input checked="" type="checkbox"/>	1001449956001	PROVESUM S CIA	062959727/	CHICA NARVAEZ 7-24
<input checked="" type="checkbox"/>	1001218427001	MEGA MONSERRATH	062920681/	OTAVALO
<input checked="" type="checkbox"/>	1706650312001	COMERCIAL SAN REMO	022314310/	MACHACHI
<input checked="" type="checkbox"/>	10000000000	ANDES PAPELERÍA	062922621/	OTAVALO ECUADOR
<input checked="" type="checkbox"/>	10000000000001	COMERCIAL LANDAZURI	022584829/0997441007	MIRES N 6-44
<input checked="" type="checkbox"/>	1001767506001	FANTASÍAS ESTEFANY	062951006/	BARTOLOMÉ GARCIA 1-90

Con la opción  seleccionamos al proveedor.

CUENTAS X PAGAR AL PROVEEDOR: DILIPA 						
OP	FECHA CREACION	TOTAL CUENTA	NUM CUOTAS	SALDO PENDIENTE	CUOTAS PENDIENTES	ESTADO
	2012-12-27 20:13:04	125.00	1	0.00	0	CANCELADO
	2013-01-04 01:06:39	600.00	90909	-600.00	90908	ACTIVO

Nos aparece un listado con las facturas pendientes por pagar el total, el saldo pendiente, y la fecha de creación.

Cuando se cancela una cuanta se debe dar por terminada la deuda con el

botón 