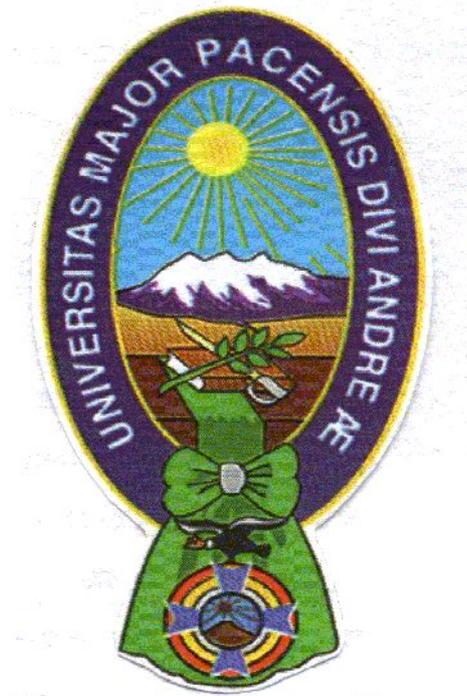


**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**“SISTEMA DE INFORMACIÓN PARA EL CONTROL Y  
SEGUIMIENTO DE VENTAS ON-LINE DE PRODUCTOS  
FARMACÉUTICOS DISTRIBUIDORA PHARMICA”**

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

**POSTULANTE:** MICHAEL FLORES GONZALES  
**TUTORA METODOLOGICA:** Dra. FÁTIMA CONSUELO DOLZ DE MORENO M. Sc.  
**ASESOR:** Lic. JUAN GONZALO CONTRERAS CANDIA

**LA PAZ – BOLIVIA**

2016



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



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

**LICENCIA DE USO**

El usuario está autorizado a:

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

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

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

## **DEDICATORIA**

*Con mucho Cariño A mi familia por su apoyo y confianza incondicional, en especial a mi hijo Alejandro Mijael quien me dio fuerzas para poder culminar el presente proyecto.*

## **AGRADECIMIENTOS**

Agradecer a mi madre, mi esposa, mi hijo y mi sobrino por su aliento, comprensión y cariño que me brindaron, por sus enseñanzas y valores por todo su apoyo a lo largo de mi vida gracias de todo corazón.

Agradecer a Dios que siempre está acompañándome cada día.

Agradecer a la Dra. Fátima Consuelo Dolz De Moreno por su paciencia, consejos, colaboración en la realización del presente Proyecto de Grado, de igual manera agradezco a mi Revisor Lic. Juan Gonzalo Contreras Candia por su apoyo, sugerencias, acompañamiento y observaciones que ayudaron a alcanzar los objetivos propuestos.

Expresar mi sincera gratitud al Sr. Sergio Gonzales Llanos por su comprensión, apoyo y por la acogida que me dieron en la Distribuidora PHARMICA.

Agradecer a cada uno de los docentes y admirativos con los que compartí mi vida universitaria pues todos ellos aportaron con sus conocimientos y apoyo en mi desarrollo profesional.

Muchas Gracias...

## RESUMEN

Los sistemas informáticos en la actualidad están siendo incorporados en una gran variedad de instituciones públicas y privadas con la tendencia a incrementarse su implementación hacia todos los campos de la sociedad a medida que transcurre el tiempo.

Además de lo mencionado, existe una presión constante por parte de la sociedad actual, que por distintos motivos necesita información cada vez más rápida y a la vez confiable, en las empresas que se dedican al comercio se tiene la necesidad de que la información respecto a los diferentes tipos de transacciones que realiza sea oportuna y constantemente actualizada, así como también responder los requerimientos de los usuarios, facilitándoles el trabajo al momento gestionar la información.

Tomando en cuenta estas consideraciones, es necesario implantar un “Sistema de Información para el Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora PHARMICA”. Este reto implica: ofrecer un servicio de información sobre los pedidos, ventas y el seguimiento de los mismos que sea confiable, pertinente y además que la información acerca del comportamiento de la empresa PHARMICA esté almacenada en una base de datos, segura en un servidor y protegida por un acceso restringido para su posterior utilización.

Para la implementación del sistema se utilizaron los lenguajes de Programación: Php, Html, JavaScript, Css, los frameworks: CodeIgniter, JQuery, una base de datos creada en Mysql y el servidor apache los cuales se adecuan a las necesidades y requerimientos de la Institución.

La metodología utilizada para el desarrollo del sistema es Proceso Unificado de Racional (RUP) para el análisis y diseño en sus diferentes fases, el modelado del sistema se realizó con el Lenguaje de Modelado Unificado (UML) y la herramienta de modelado Enterprise Architect.

## ÍNDICE

<b>Capítulo I: Introducción .....</b>	<b>10</b>
1.1. Introducción .....	10
1.2. Antecedentes .....	11
1.3. Planteamiento del Problema.....	12
1.3.1. Problema General .....	12
1.3.2. Problemas Específicos.....	12
1.4. Objetivo General .....	13
1.4.1. Objetivos Específicos .....	13
1.5. Justificación.....	14
1.5.1. Justificación Social.....	14
1.5.2. Justificación Técnica .....	14
1.5.3. Justificación Económica.....	15
1.6. Alcances .....	16
1.6.1. Alcance Temporal.....	16
1.6.2. Alcance Temático .....	16
1.6.3. Alcance Geográfico .....	16
<b>Capítulo II: Marco Teórico .....</b>	<b>18</b>
2.1. Introducción .....	18
2.2. Comercio Electrónico.....	18
2.2.1. Tipos de Comercio Electrónico.....	19
2.2.2. Disposiciones Legales del Comercio Electrónico en Bolivia. ....	19
2.2.3. Ventajas y Desventajas del Comercio Electrónico. ....	20
2.3. Estadística.....	21
2.3.1. Población.....	22
2.3.2. Datos Estadísticos. ....	22

2.3.3. Graficas Estadísticas.....	24
2.4. Sistema.....	26
2.5. Información.....	27
2.6. Sistema de Información.....	28
2.7. Aplicaciones Web “WebApps”.....	28
2.7.1. Protocolos de Internet.....	30
2.8. Metodología para el Desarrollo del Sistema.....	32
2.8.1. Proceso Unificado De Rational “RUP”.....	32
2.8.2. Fases y Descripción Del Proceso.....	35
2.9. El Lenguaje Unificado de Modelado (UML).....	39
2.9.1. Clases.....	39
2.9.2. Objetos.....	40
2.9.3. Asociaciones.....	41
2.9.4. Diagrama de Casos De Uso.....	42
2.9.5. Modelo Conceptual.....	43
2.9.6. Diagrama de Secuencia.....	44
2.9.7. Implementación y Pruebas.....	45
2.10. Tecnologías de Software.....	45
2.10.1. Tecnologías para el Sistema de Gestión de Bases de Datos (SGBD).....	45
2.10.2. Lenguajes de Programación.....	47
2.10.3. Frameworks.....	53
2.11. Herramienta de Modelado.....	58
2.11.1. Enterprice Architech.....	58
2.12. Metodología de Evaluación de La Calidad de un Sitio Web.....	59
2.12.1. Funcionalidad.....	60
2.12.2. Fiabilidad.....	61
2.12.3. Usabilidad.....	61
2.12.4. Eficiencia.....	62
2.12.5. Portabilidad.....	62
2.13. Modelo De Calidad En Uso.....	63

2.14. Modelo de Constructivo de Costo: COCOMO .....	64
2.15. Conclusión .....	65
<b>Capitulo III: Marco Aplicativo .....</b>	<b>66</b>
3.1. Introducción .....	66
3.2. Fase de Inicio .....	66
3.2.1. Modelado del Negocio .....	66
3.2.2. Modelado de Casos de Uso .....	67
3.2.3. Descripción de Casos de Uso .....	69
3.2.4. Descripción de los Actores .....	70
3.2.5. Análisis de Requerimientos .....	72
3.3. Fase de Elaboración .....	74
3.3.1. Identificación de Actores del Nuevo Sistema .....	74
3.3.2. Modelo de Casos de Uso del Nuevo Sistema .....	75
3.4. Análisis y Diseño .....	78
3.4.1. Especificación de Casos De Uso .....	78
3.5. Fase de Construcción .....	82
3.5.1. Diagramas de Secuencia para Diseño .....	82
3.5.2. Diagramas de Colaboración .....	85
3.5.3. Diagramas de Estados .....	88
3.5.4. Diagramas de Actividades .....	90
3.5.5. Diagramas de Clases .....	91
3.5.6. Definición Del Modelo Físico de la base de Datos .....	92
3.5.7. Diccionario de Datos .....	93
3.6. Fase de Transición .....	98
3.6.1. Diseño de Interfaz .....	98
3.7. Seguridad .....	102
3.7.1. Control de Acceso .....	102
3.7.2. Seguridad de Acceso .....	102
3.7.3. Seguridad de la Información .....	103

3.7.4. Arquitectura del Sistema .....	103
<b>CAPITULO IV: Calidad de Software .....</b>	<b>104</b>
4.1 Introducción .....	104
4.2 Confiabilidad.....	104
4.3 Usabilidad.....	104
4.4 Funcionalidad .....	105
4.4.1 Punto Función.....	105
4.5 Métricas de Mantenimiento.....	108
4.5.1 Métricas de Portabilidad.....	110
4.5.2 Eficiencia .....	110
4.5.3 Transportabilidad.....	110
4.6 Estimación del Costo del Proyecto.....	111
4.6.1 Métricas Orientadas Al Tamaño.....	111
4.7 Conclusión.....	113
<b>CAPITULO V:Conclusiones y Recomendaciones .....</b>	<b>114</b>
4.7 Conclusiones .....	114
5.2 Recomendaciones.....	114
6. Bibliografía.....	115
7. Anexos.....	118
8. Documentación.....	122

## ÍNDICE DE TABLAS

<b>Tabla 1.1:</b> Costo estimado de la implementación del Sistema .....	15
<b>Tabla 1.2:</b> Usuarios y alcances geográficos del Sistema.....	17
<b>Tabla 3.1:</b> Descripción de Casos de Uso Registra Cobros Pendientes en Planillas .....	69
<b>Tabla 3.2:</b> Descripción de Casos de Uso Registra Entrega de Pedido en planilla.....	69
<b>Tabla 3.3:</b> Descripción de Casos de Uso Centraliza Pedidos de forma manual en una planilla.....	69
<b>Tabla 3.4:</b> Descripción de Casos de Uso Estandariza la información de manera sistémica .....	70
<b>Tabla 3.5:</b> Descripción de Casos de Uso Elaborar Gráficas Estadísticas.....	70
<b>Tabla 3.6:</b> Descripción de actores del Nuevo Sistema .....	74
<b>Tabla 3.7:</b> Especificación Caso de Uso: Login .....	78
<b>Tabla 3.8:</b> Especificación Caso de Uso: Nuevo Usuario.....	79
<b>Tabla 3.9:</b> Especificación Caso de Uso: Modificar Usuarios.....	79
<b>Tabla 3.9:</b> Especificación Caso de Uso: Listar Usuario .....	80
<b>Tabla 3.10:</b> Especificación Caso de Uso: Logout .....	80
<b>Tabla 3.11:</b> Especificación Caso de Uso: Registrar Producto Nuevo .....	81
<b>Tabla 3.12:</b> Especificación Caso de Uso: Clasificar y Ordenar Pedidos.....	81
<b>Tabla 4.1:</b> Cuestionario para Calcular la Usabilidad.....	104
<b>Tabla 4.2:</b> Tabla De Ponderación .....	105
<b>Tabla 4.3:</b> Ajuste de Complejidad del punto función.....	106
<b>Tabla 4.4:</b> Escala de Funcionalidad.....	107

<b>Tabla 4.5:</b> Información requerida por el IMS.....	109
<b>Tabla 4.6:</b> Tabla de Estimación Cocomo .....	112
<b>Tabla 4.7:</b> Tabla de Categorías Modelo de Estimación Cocomo .....	113

## ÍNDICE DE FIGURAS

<i>Figura 2.1:</i> Datos Estadísticos.....	23
<i>Figura 2.2:</i> Gráfico de Barras .....	24
<i>Figura 2.3:</i> Índice de Precios al Consumo .....	25
<i>Figura 2.4:</i> Alojamientos Turísticos .....	26
<i>Figura: 2.5:</i> Proceso de la Información .....	27
<i>Figura: 2.6:</i> Sistema de Información .....	28
<i>Figura 2.7:</i> Incremento de la Productividad en Equipo .....	33
<i>Figura 2.8:</i> Desarrollo Iterativo .....	35
<i>Figura 2.9:</i> Fases del RUP .....	35
<i>Figura 2.10:</i> Metas del RUP .....	37
<i>Figura 2.11:</i> Iteraciones .....	37
<i>Figura 2.12:</i> Ejemplo de un Diagrama de Clases .....	40
<i>Figura 2.13:</i> Ejemplo de Estructura de una Clases .....	40
<i>Figura 2.14:</i> Ejemplo de Estructura de un Objeto .....	40
<i>Figura 2.15:</i> Ejemplo de multiplicidad en asociaciones. ....	41
<i>Figura 2.16:</i> Ejemplo de Agregación.....	41
<i>Figura 2.17:</i> Ejemplo de Herencia .....	42
<i>Figura 2.18:</i> Ejemplo de Caso de Uso .....	43
<i>Figura 2.19:</i> Ejemplo de Modelo Conceptual Parcial.....	44
<i>Figura 2.20:</i> Ejemplo de Diagrama de Secuencia.....	44
<i>Figura 2.21:</i> Esquema del funcionamiento de las páginas PHP. ....	49

<i>Figura 2.22:</i> Flujo de trabajo de CodeIgniter.....	57
<i>Figura 2.23:</i> Calidad Interna y Externa.....	60
<i>Figura: 2.24.</i> Modelo de Calidad .....	63
<i>Figura 2.25:</i> Mercado de Software Actual y Futuro .....	64
<i>Figura 3.1:</i> Modelado del Negocio Mediante Paquetes.....	67
<i>Figura 3.2:</i> Diagrama de Casos de Uso del Negocio .....	68
<i>Figura 3.3:</i> Casos de Uso del Sistema .....	76
<i>Figura 3.4:</i> Casos de Uso Administrar Usuarios .....	77
<i>Figura 3.5:</i> Casos de Uso Administrar Usuarios .....	77
<i>Figura 3.6:</i> Diagrama de Secuencia Listar Usuarios .....	82
<i>Figura 3.7:</i> Diagrama de Secuencia Login .....	83
<i>Figura 3.8:</i> Diagrama de Secuencia Crear Usuario .....	83
<i>Figura 3.9:</i> Diagrama de Secuencia Logout .....	84
<i>Figura 3.10:</i> Diagrama de Secuencia Modificar Usuario .....	84
<i>Figura 3.11:</i> Diagrama de Secuencia Listar Productos.....	85
<i>Figura 3.12:</i> Diagrama de Colaboración Listar Usuarios .....	85
<i>Figura 3.13:</i> Diagrama de Colaboración Login .....	86
<i>Figura 3.14:</i> Diagrama de Colaboración Modificar Usuario .....	86
<i>Figura 3.15:</i> Diagrama de Colaboración Crear Usuario .....	87
<i>Figura 3.16:</i> Diagrama de Colaboración Logout .....	87
<i>Figura 3.17:</i> Diagrama de Colaboración Listar Productos .....	88
<i>Figura 3.18:</i> Diagrama de Estados Login .....	88

<i>Figura 3.19:</i> Diagrama de Estados Registrar Nuevo Usuario.....	89
<i>Figura 3.20:</i> Diagrama de Estados Generar Informe Diario de Ingresos y Egresos de Productos.....	89
<i>Figura 3.21:</i> Diagrama de Estados Generar Nota de Venta.....	89
<i>Figura 3.22:</i> Diagrama de Actividad Registrar Nuevo Usuario .....	90
<i>Figura 3.23:</i> Diagrama de Actividad Eliminar Usuario.....	91
<i>Figura 3.24:</i> Diagrama de Clases.....	92
<i>Figura 3.25:</i> Modelo Físico de la Base de Datos .....	93
<i>Figura 3.26:</i> Interfaz de inicio .....	98
<i>Figura 3.27:</i> Interfaz de autenticación.....	99
<i>Figura 3.28:</i> Interfaz del Administrador .....	99
<i>Figura 3.29:</i> Interfaz de gestión de Productos .....	100
<i>Figura 3.30:</i> Interfaz adicionar usuario.....	100
<i>Figura 3.31:</i> Interfaz de agregar pedido.....	101
<i>Figura 3.32:</i> Arquitectura del sistema.....	103

## Capítulo I: Introducción

### 1.1. Introducción

En la actualidad el mundo está interconectado gracias al internet, el cual se utiliza en las diferentes áreas que existen (comunicación, comercio, salud, etc.), para ello se utilizan sistemas de información que deben tener las siguientes características: deben ser oportunos, confiables y que aseguren cumplir con las exigencias de toda institución, debido a que constituyen una parte esencial de las operaciones de toda institución. Los sistemas informáticos son parte esencial de las instituciones públicas y privadas, con la tendencia a incrementar su implementación hacia todas las áreas de las actividades, los sistemas de información que se aplican a las actividades comerciales que tiene lugar por medio de computadoras conectadas a través de una red tienen como denominación e-commerce o comercio electrónico, hoy en día cambia un poco la definición ya que en los tiempos actuales se pueden realizar actividades comerciales desde un Smartphone gracias a la tecnología que estos dispositivos poseen y con las aplicaciones web o webapp's se puede llegar a diferentes tipos de terminales como ser Tablet's, Smartphones y Computadoras de manera eficiente a través de la internet.

Algunas instituciones o empresas desconocen los beneficios que pueden traer las Tecnologías de la Información y Comunicación en especial en el área del comercio, Por lo tanto están obligadas a mejorar su situación mediante la implementación de un sistema informático que permita automatizar los procedimientos de compra, venta, reservación y seguimiento de los productos farmacéuticos.

Tomando en cuenta los puntos anteriores se considera la necesidad de desarrollar un Sistema de Información para el Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos

Distribuidora PHARMICA. Esta distribuidora tiene como área de trabajo los departamentos de La Paz, Cochabamba, Oruro, Santa Cruz, solo existe una sucursal en La Paz la cual envía los productos farmacéuticos vía terrestre a los demás departamentos, todo el registro de compra y venta se realiza de manera manual lo cual dificulta y genera deficiencias en la elaboración de reportes, informes, en el control de los productos farmacéuticos. Todo es proceso anterior debe ser sistematizado a través de un sistema informático con una base de datos generando resultados confiable y oportunos sobre la información generada.

## **1.2. Antecedentes**

La empresa distribuidora PHARMICA fue inscrita en el registro de comercio de Bolivia en fecha 30 de diciembre de 2013 con la matrícula de comercio N° 00278469 con número de identificación tributaria 3360384015, se encuentra ubicada en la ciudad de El Alto zona 1° de mayo manzano M número 254 se dedica a la comercialización de productos farmacéuticos de diferentes gamas los cuales tienen gran demanda, por tal efecto la empresa tiene un mercado que va creciendo continuamente tanto en población como geográficamente.

La empresa PHARMICA se dedica a la comercialización de los diferentes tipos de productos farmacéuticos:

- Accesorios médicos.
- Instrumentos.
- Materiales quirúrgicos.

La empresa PHARMICA realiza los siguientes procesos:

- Venta, envío, recepción, entrega de productos farmacéuticos.

- El personal de ventas oferta los productos constantemente a las instituciones consumidoras (farmacias, clínicas, centros de salud y otros).
- El acceso y verificación a los registros de venta se realiza constantemente en una hoja de cálculo en Excel.
- Se realiza una verificación física del stock de productos.
- El seguimiento de los Productos farmacéuticos se realiza en base a documentos físicos.

### **1.3. Planteamiento del Problema**

Actualmente la distribuidora PHARMICA administra las ventas, pedidos, reservas y otros procesos de manera personal y manual las cuales son registradas en formularios impresos, el proceso de registro de ventas se centraliza de en una hoja de cálculo electrónica, para dichas tareas se utiliza recursos humanos, material de escritorio, tiempo, costo, el cual es una desventaja para la empresa.

#### **1.3.1. Problema General**

Debido al proceso manual de información generada por la venta y adquisición de productos farmacéuticos, recolectados de cada una de las diferentes empresas abastecen y clientes a los que se distribuye los productos, existe mucha demora en la elaboración de reportes, informes, ocasionando que la asignación, control y seguimiento de los productos farmacéuticos por parte de la empresa no sea adecuada.

#### **1.3.2. Problemas Específicos**

Los problemas actuales que atraviesa la distribuidora PHARMICA en relación al manejo de información de los productos farmacéuticos son descritos a continuación:

- El registro y verificación de existencia de los productos farmacéuticos es manual se emplea demasiado tiempo en su seguimiento lo que provoca que la actualización periódica de los productos farmacéuticos registros sea manual y deficiente.
- El registro de adquisición y venta de productos farmacéuticos es de forma manual lo que dificulta Control del manejo de la información que estos registros generan.
- Se emplea mucho tiempo para conocer las característica y funciones de un producto farmacéutico lo cual provoca pérdida de tiempo en la ubicación de las especificaciones, proveedores de productos farmacéuticos e información de los clientes.
- La elaboración de informes de ventas y sobre existencia de productos farmacéuticos no se proporciona a tiempo, por tal motivo la empresa no cuenta con información oportuna para una buena toma de decisiones.
- En la empresa no existe un control adecuado del stock de productos farmacéuticos, por lo cual el abastecimiento de productos farmacéuticos no se lo pronostica de manera oportuna.

#### **1.4. Objetivo General**

Desarrollar e implementar un “Sistema de Información para el Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora PHARMICA” que cuente con información eficiente y oportuna sobre el registro control y seguimiento de productos farmacéuticos que la empresa PHARMICA distribuye.

##### **1.4.1. Objetivos Específicos**

Los objetivos planteados en el presente proyecto son:

- Crear una base de datos relacional para almacenar la información de los registros generados en la empresa PHARMICA.

- Realizar el proceso de registro y seguimiento de adquisición y venta de productos farmacéuticos.
- Se tendrá un control eficiente y oportuno sobre la compra y venta de productos farmacéuticos.
- Realizar el proceso de consultas a la base de datos para la buena toma de decisiones.
- Tratamiento de la información por medio de internet.
- Aplicar modelos estadísticos para la proyección de datos a través de graficas estadísticas.
- Realizar reportes diarios y semanales sobre el stock en productos.
- Realizar el proceso de acceso para usuarios que realicen el pedido vía internet.
- Realizar la venta o requerimiento de pedidos vía web.
- Administración de usuarios con diferentes roles para restringir y brindar seguridad a la información almacenada en la base de datos.

## **1.5. Justificación**

### **1.5.1. Justificación Social**

Con la implementación del “Sistema de Información de Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora PHARMICA”, el cual generara informes, reportes, información estadística, graficas de manera comprensible permitiendo la toma de decisiones oportunas, como campañas a favor de la población.

También serán beneficiados el personal y organización puesto que el sistema reducirá el tiempo y el trabajo manual con la búsqueda, clasificación, actualización, creación, modificación y ordenamiento automatizado de la información generada en los diferentes procesos de la empresa.

### **1.5.2. Justificación Técnica**

La distribuidora PHARMICA cuenta con equipos de computación y dispositivos móviles que puede ser utilizado a plenitud y de manera óptima, aprovechando la tecnología informática

mediante la implementación de un sistema en línea, permitiendo así una adecuada explotación de la información. El Sistema Informático beneficiará a el conjunto de clientes (farmacias, clínicas, centros de salud) y clientes potenciales dentro de los departamentos de La Paz, Cochabamba, Santa Cruz, Oruro y Potosí. Se tiene la tecnología disponible en la institución, que hace posible la instalación de programas requeridos para el desarrollo del sistema.

### 1.5.3. Justificación Económica

Mediante la automatización de los procesos que son realizados manualmente se minimiza el esfuerzo laboral, y también se reducen los errores cometidos ahorrando así mano de obra, tiempo y material, por otro lado, se evitan pérdidas económicas en cuanto a los ingresos de datos ya que se da eficiencia y eficacia, como también al despliegue continuo de la información .

El presente proyecto es realizable ya que se cuenta con todo el respaldo de la institución, y es factible cubrir todos los gastos necesarios, ya que es importante tanto como para la “Distribuidora PHARMICA” y la sociedad tener acceso a información.

**Tabla 1.1:** Costo estimado de la implementación del Sistema

COSTO	MONTO \$	TOTAL
<b>Costo de Construir el Sistema</b> Analista de Sistemas	150	150
<b>Costo de Implementar el Sistema</b> Capacitación Realización de Base de Datos Soporte	50 50 50	150
<b>Costos Operacionales</b> Hardware y Software Mantenimiento Hosting Web	0 50 150	200
<b>Total de Costos</b>		\$500

Fuente: Elaboración Propia

## **1.6. Alcances**

### **1.6.1. Alcance Temporal**

Las diferentes fases para la elaboración del “Sistema de Información de Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora PHARMICA” que comprenden en general el análisis, diseño e implementación, de acuerdo a la complejidad del sistema de información el proyecto se terminara de implementar en un lapso de ocho meses desde el inicio del proyecto.

### **1.6.2. Alcance Temático**

El “Sistema de Información para el Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora PHARMICA” en su primera etapa tendrá como prioridad la implementación de los módulos pertinentes como ser:

- Módulo de Marketing para difundir los productos ofrecidos por la distribuidora.
- Módulo de Registro de Clientes Nuevos.
- Módulo de ventas y pedidos para clientes registrados.
- Módulo de estadísticas sobre ventas.
- Módulo de consultas para el seguimiento de procesos.
- Módulo de compras o adquisiciones de productos farmacéuticos.
- Módulo de control y seguimiento de procesos.

### **1.6.3. Alcance Geográfico**

La Distribuidora PHARMICA es un empresa en expansión, y tomando en cuenta que el “Sistema de Información para el Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora PHARMICA” tendrá módulos para los diferentes tipos de usuarios (clientes, personal de la empresa, clientes potenciales) y se encontrara alojado en un servidor

contratado para su acceso mediante internet definimos los siguientes alcances según los tipos de usuarios:

**Tabla 1.2:** Usuarios y alcances geográficos del Sistema.

Tipo de Usuario	Alcance
Personal de la Empresa	La Paz, Cochabamba, Oruro, Potosí y Santa Cruz
Clientes	La Paz, Cochabamba, Oruro, Potosí y Santa Cruz
Clientes potenciales	Mundial

*Fuente: [elaboración propia]*



## Capítulo II: Marco Teórico

### 2.1. Introducción

En este capítulo haremos referencia de algunos conceptos sobre comercio electrónico, ventajas y desventajas concerniente a la realización de la venta de los productos en la web además de las transacciones en línea, protocolos de internet que nos ayudarán en la documentación de este proyecto.

También plasmar un pequeño informe de los métodos, herramientas que se pueden utilizar para el desarrollo del sistema en cuanto al modelado, metodologías para la evaluación de la calidad de sitios web, tecnologías para la base de datos.

### 2.2. Comercio Electrónico

Según Electronic Commerce 2012 (Edición Global) el comercio electrónico es una modalidad que irrumpió en el mundo a mediados de los 90's de la mano del auge de Internet... entre las facilidades que permite es el poder estar presente en todos los mercados del mundo 24hs al día, estar actualizado y seguirle la pisada a la competencia.

También indica que el comercio electrónico es un arma de venta que nace a partir de las necesidades de empresas que desean llegar a posicionarse en un lugar importante dentro del mercado, proyectándose a nivel local, nacional e internacional y permite optimizar operaciones, realizar ventas en línea y brindar un mejor servicio a sus clientes.

En el Diccionario de Internet y Redes de Microsoft se define al Comercio Electrónico como “Actividad comercial que tiene lugar por medio de computadoras conectadas a través de una red” (Mc Graw Hill, 2003, Madrid – España).

## **2.2.1. Tipos de Comercio Electrónico.**

### **2.2.1.1. B2B (*Business to Business*).**

“Business to Business (B2B) Son los negocios de Comercio realizados entre Empresas. Contratos EDI (Entre empresas, contratando modelos económicos, contratos Asp). Por Ejemplo, yo le entrego la contabilidad a otra empresa, una empresa realiza pedidos de materia prima a sus proveedores por internet.” [MELENDEZ C. s/f]

### **2.2.1.2. B2C (*Business to Consumer*).**

“Este tipo de comercio se produce entre una empresa y un consumidor final que consume el bien o servicio, esto es, no lo transforma en algo distinto para venderlo o traspasarlo a otro actor del ciclo de negocios” [NUÑEZ, S/F].

## **2.2.2. Disposiciones Legales del Comercio Electrónico en Bolivia.**

Según la Ley General De Telecomunicaciones, Tecnologías De Información Y Comunicación Sobre Desarrollo Contenidos Y Aplicaciones De Tecnologías De Información Y Comunicación (Ley N° 164) en el Capítulo 14 decreta: que régimen técnico del comercio electrónico se desarrollado por la ATT, también que la oferta electrónica de bienes y servicios debe contener:

- ✓ Las condiciones generales del contrato y la descripción precisa de los procedimientos para su celebración, su conservación y accesibilidad, en caso de ser necesario.

- ✓ Los medios técnicos para identificar y corregir los errores de introducción de datos antes de efectuar el pedido.
- ✓ Los procedimientos para que el adquirente reciba el comprobante de la operación o factura en su caso.

Además el ofertante de bienes y servicios deberá proporcionar información precisa y suficiente sobre las características de los bienes o servicios que ofrece y toda la información relativa a los productos ofrecidos, incluyendo especialmente cualquier advertencia sobre peligros o limitaciones relativos a su uso o consumo, asimismo, proporcionará información precisa y suficiente sobre el precio de los bienes o servicios que ofrece, incluyendo todos los costos adicionales.

### **2.2.3. Ventajas y Desventajas del Comercio Electrónico.**

*Ventajas para el Usuario: Según Loyola L. (2010)*

- ✓ Comodidad: evita desplazamientos y horarios.
- ✓ Tener acceso a más diversidad e información.
- ✓ Facilita y acelera el proceso de comparación y selección.
- ✓ Mercado más competitivo y por lo tanto más barato.
- ✓ Evita la acción directa del comerciante en la toma de decisiones
- ✓ Presentación de mejores ofertas y descuentos
- ✓ El uso empresarial del Web reduce errores, tiempo y sobrecostos en el tratamiento de la información.

- ✓ Un sitio Web se encuentra disponible las 24 horas del día bajo demanda de los clientes.

***Ventajas para la empresa: Según Loyola L. (2010)***

- ✓ Acceso al mayor número de clientes potenciales.
- ✓ Máxima disponibilidad al menor coste.
- ✓ Evita la necesidad de los costes físicos.
- ✓ Facilidad de extensión del negocio y entrada en nuevos mercados.
- ✓ Artículos digitales con coste de distribución cero.
- ✓ Contacto directo con el cliente, evita intermediarios.
- ✓ Mayor eficiencia en las transacciones.
- ✓ Facilita el marketing y el soporte al cliente.
- ✓ Mercado accesible a las pequeñas empresas en igualdad de condiciones.

### **2.3. Estadística**

La estadística es comúnmente considerada como una colección de hechos numéricos expresados en términos de una relación sumisa, y que han sido recopilados a partir de otros datos numéricos.

La estadística es una ciencia con base matemática referente a la recolección, análisis e interpretación de datos, que busca explicar condiciones regulares en fenómenos de tipo aleatorio.

- La estadística descriptiva, que se dedica a los métodos de recolección, descripción, visualización y resumen de datos originados a partir de los fenómenos en estudio.

Los datos pueden ser resumidos numérica o gráficamente. Ejemplos básicos de parámetros estadísticos son: la media y la desviación estándar. Algunos ejemplos gráficos son: histograma, pirámide poblacional, clústers, etc.

- La inferencia estadística, que se dedica a la generación de los modelos, inferencias y predicciones asociadas a los fenómenos en cuestión teniendo en cuenta la aleatoriedad de las observaciones. Se usa para modelar patrones en los datos y extraer inferencias acerca de la población bajo estudio [MURRAY, 1998].

### **2.3.1. Población.**

El concepto de población en estadística va más allá de lo que comúnmente se conoce como tal. Una población se precisa como un conjunto finito o infinito de personas u objetos que presentan características comunes [AZUNA J. 1991].

"Una población es un conjunto de todos los elementos que estamos estudiando, acerca de los cuales intentamos sacar conclusiones"[AZORIN F. 1972].

### **2.3.2. Datos Estadísticos.**

Los datos estadísticos no son otra cosa que el producto de las observaciones efectuadas en las personas y objetos en los cuales se produce el fenómeno que queremos estudiar. Dicho en otras palabras, son los antecedentes (en cifras) necesarios para llegar al conocimiento de un hecho o para reducir las consecuencias de este.

Los datos estadísticos se pueden encontrar de forma no ordenada, por lo que es muy difícil en general, obtener conclusiones de los datos presentados de esta manera. Para poder obtener una

precisa y rápida información con propósitos de descripción o análisis, estos deben organizarse de una manera sistemática; es decir, se requiere que los datos sean clasificados. Esta clasificación u organización puede muy bien hacerse antes de la recopilación de los datos. [MURRAY, 1998].



*Figura 2.1: Datos Estadísticos  
Fuente: Elaboración propia*

La información sobre tipos de datos y Gráficas estadísticas fue recabada de una publicación de internet en la siguiente dirección (<http://www.ine.es/explica/docs/>) en un documento en formato PDF titulado “pasos\_tipos\_graficos.pdf”

### **2.3.2.1. Tipos de Datos.**

**Cualitativos:** se refieren a cualidades o modalidades que no pueden expresarse numéricamente. Pueden ser:

- ⇒ **Ordinales:** si siguen un orden o secuencia (ejemplo: el abecedario, los meses del año).
- ⇒ **Catagóricos:** si no siguen ningún orden (ejemplo el estado civil de las personas: solteros, casados, viudos, divorciados y separados).

**Cuantitativos:** se refieren a cantidades o valores numéricos. Pueden ser:

- ⇒ **Discretos:** si toman valores enteros (0, 1, 2, 3...). Ejemplo: el número de hijos, el número de alumnos de una clase.

- ⇒ **Continuos:** si pueden tomar cualquier valor dentro de un intervalo (Ejemplo: la estatura o el peso de las personas).

### 2.3.3. Graficas Estadísticas.

Un gráfico estadístico es una representación visual de una serie de datos estadísticos. Es una herramienta muy eficaz, ya que un buen gráfico:

- ⇒ Capta la atención del lector
- ⇒ Presenta la información de forma sencilla, clara y precisa
- ⇒ No induce a error
- ⇒ Facilita la comparación de datos y destaca las tendencias y las diferencias
- ⇒ Ilustra el mensaje, tema o trama del texto al que acompaña.

#### 2.3.3.1. Gráfico de Barras.

Un gráfico de barras es una representación gráfica en un eje cartesiano de las frecuencias de una variable cualitativa o discreta. En uno de los ejes se posicionan las distintas categorías o modalidades de la variable cualitativa o discreta y en el otro el valor o frecuencia de cada categoría en una determinada escala.



Figura 2.2: Gráfico de Barras  
Fuente: Organización de la Naciones Unidas

Existen diferentes tipos de gráficos de barras y son los siguientes:

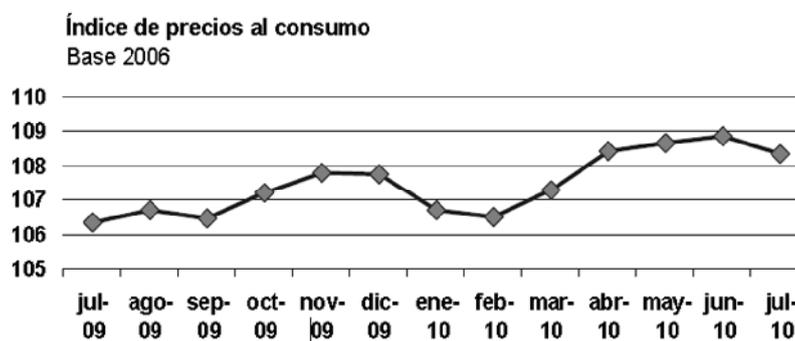
- ⇒ Sencillo
- ⇒ Agrupado
- ⇒ Apilado
- ⇒ Bi-direccionales

### 2.3.3.2. Gráfico de Líneas

Un gráfico de líneas es una representación gráfica en un eje cartesiano de la relación que existe entre dos variables reflejando con claridad los cambios producidos. En cada eje se representa cada una de las variables cuya relación se quiere observar, en la mayoría de los casos se utilizan para representar tendencias temporales.

#### IPC 2009 - 2010 Base 2006

	IPC
jul-09	106,3
ago-09	106,7
sep-09	106,4
oct-09	107,2
nov-09	107,8
dic-09	107,8
ene-10	106,7
feb-10	106,5
mar-10	107,3
abr-10	108,4
may-10	108,7
jun-10	108,9
jul-10	108,4



Fuente: IPC. INE

Figura 2.3: Índice de Precios al Consumo  
Fuente: IPC.INE

### 2.3.3.3. Gráfico de Sectores.

Un gráfico de sectores es una representación circular de las frecuencias relativas de una variable cualitativa o discreta que permite, de una manera sencilla y rápida, su comparación.

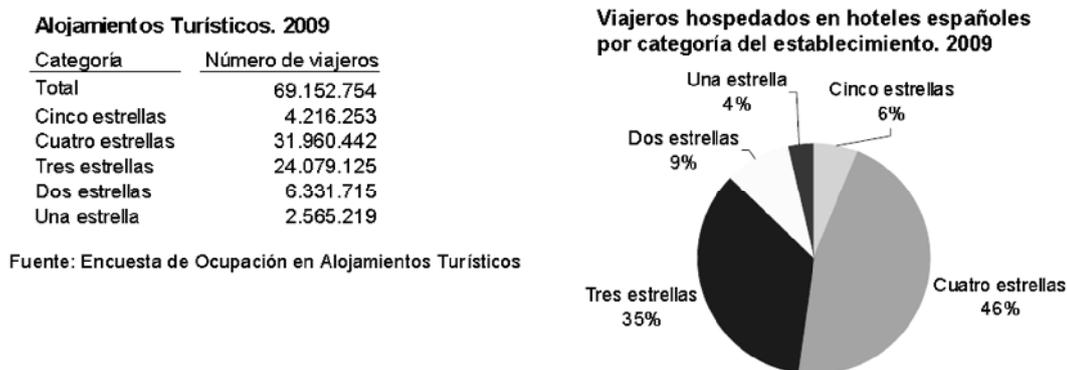


Figura 2.4: Alojamientos Turísticos  
Fuente: Encuesta de Ocupación en alojamientos Turísticos INE

El círculo representa la totalidad que se quiere observar y cada porción, llamadas sectores, representan la proporción de cada categoría que es variable respecto el total. Suele expresarse en porcentajes.

## 2.4. Sistema

Sistema es un conjunto de elementos relacionados entre sí y que interactúan de modo que forman una unidad lógica. En informática la palabra sistema se utiliza precisamente para indicar varios componentes, tanto de hardware como de software que en su conjunto constituyen un sistema de procesamiento de datos. [PRESSMAN, 2002].

## 2.5. Información

La información es un fenómeno que proporciona significado o sentido a las cosas, e indica mediante códigos y conjuntos de datos, los modelos del pensamiento humano. La información por tanto, procesa y genera el conocimiento humano. Aunque muchos seres vivos se comunican transmitiendo información para su supervivencia, la diferencia de los seres humanos radica en su capacidad de generar y perfeccionar tanto códigos como símbolos con significados que conformaron lenguajes comunes útiles para la convivencia en sociedad, a partir del establecimiento de sistemas de señales y lenguajes para la comunicación.

En sentido general, la información es un conjunto organizado de datos procesados, que constituyen un mensaje sobre un determinado ente o fenómeno. De esta manera, si por ejemplo organizamos datos sobre un país, tales como: número de habitantes, densidad de población, nombre del presidente, etc. y escribimos por ejemplo, el capítulo de un libro, podemos decir que ese capítulo constituye información sobre ese país. Cuando tenemos que resolver un determinado problema o tenemos que tomar una decisión, empleamos diversas fuentes de información, y construimos lo que en general se denomina conocimiento o información organizada que permite la resolución de problemas o la toma de decisiones para la mejora de los mismos.

Existe una relación indisoluble entre los datos, la información, el conocimiento, el pensamiento y el lenguaje, por lo que una mejor comprensión de los conceptos sobre información redundará en un aumento del conocimiento. [ALCALDE, 2007]



*Figura: 2.5: Proceso de la Información*  
*Fuente: [ALCALDE, 2007]*

## 2.6. Sistema de Información

Un sistema de información es el conjunto de procedimientos ordenados que, al ser ejecutados, proporcionan información para apoyar a la toma de decisiones y al control de la organización.

Un sistema de información (SI) es un conjunto organizado de elementos, los cuales estarán formados por:

- ⇒ Personas.
- ⇒ Datos.
- ⇒ Actividades o técnicas de trabajo.
- ⇒ Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente parte de alguna de las siguientes categorías) [PRESSMAN, 2002].

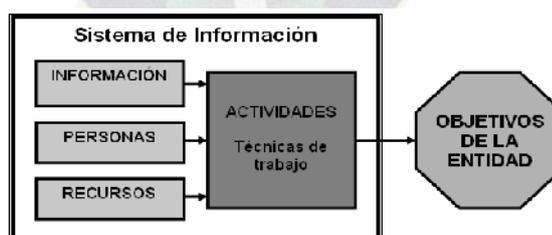


Figura: 2.6: Sistema de Información

Fuente: [PRESSMAN, 2002].

## 2.7. Aplicaciones Web “WebApps”

Son sistemas de información que se encuentran accesibles mediante el internet. En la actualidad las webapps están evolucionando hacia ambientes de cómputo sofisticados que no sólo

proveen características aisladas, funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios.

Una webapp presenta los siguientes atributos:

**Uso intensivo de redes.** Una webapp reside en una red y debe atender las necesidades de una comunidad diversa de clientes. La red permite acceso y comunicación mundiales (por ejemplo, Facebook) o tiene acceso y comunicación limitados (por ejemplo, una intranet corporativa).

**Concurrencia.** A la webapp pueden acceder un gran número de usuarios a la vez.

**Carga impredecible.** La cantidad de usuarios que acceden a la webapp varía en magnitud de un día a otro.

**Rendimiento.** Si un usuario de la webapp debe esperar demasiado (para entrar, para el procesamiento por parte del servidor, para el formado y despliegue del lado del cliente), él o ella quizá decidan irse a otra parte.

**Disponibilidad.** Es frecuente que los usuarios de webapps populares demanden acceso las 24 horas de los 365 días del año (Ejemplo: Las redes sociales).

**Orientadas a los datos.** La función principal de muchas webapp es el uso de hipermedios para presentar al usuario final contenido en forma de texto, gráficas, audio y video.

Además, las webapps se utilizan en forma común para acceder a información que existe en bases de datos que no son parte integral del ambiente basado en web (por ejemplo, comercio electrónico o aplicaciones financieras).

**Contenido sensible.** La calidad y naturaleza estética del contenido constituye un rasgo importante de la calidad de una webapp.

**Evolución continua.** A diferencia del software de aplicación convencional que evoluciona a lo largo de una serie de etapas planeadas y separadas cronológicamente, las aplicaciones web evolucionan en forma continua. No es raro que ciertas webapp (específicamente su contenido) se actualicen minuto a minuto o que su contenido se calcule en cada solicitud.

**Inmediatez.** Es una característica en muchos dominios de aplicación, es frecuente que las webapps tengan plazos de algunos días o semanas para llegar al mercado.

**Seguridad.** Debido a que las webapps se encuentran disponibles con el acceso a una red, es difícil o imposible limitar la población de usuarios finales que pueden acceder a la aplicación.

Con el fin de proteger el contenido sensible y brindar modos seguros de transmisión de los datos, deben implementarse medidas estrictas de seguridad a través de la infraestructura de apoyo de una webapp y dentro de la aplicación misma.

**Estética.** Parte innegable del atractivo de una webapp es su apariencia y percepción, la estética tiene tanto que ver con el éxito como el diseño técnico [PRESSMAN, R. 2010].

## 2.7.1. Protocolos de Internet

### 2.7.1.1. HTTP.

Según la W3C el protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C y la IETF, colaboración que culminó en 1999 con la publicación de una serie de RFC, siendo el más importante de ellos el RFC 2616, que especifica la versión 1.1.

HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura Web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a

transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

### **2.7.1.2. FTP.**

La W3C define a FTP (File Transfer Protocol) como un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El Servicio FTP es ofrecido por la capa de Aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos.

## **2.8. Metodología para el Desarrollo del Sistema**

El análisis y diseño estará basado con los métodos orientados a objetos. El enfoque orientado a objetos no es nuevo, han transcurrido muchos años para una plena aplicación de este enfoque.

Dado el acercamiento con el mundo real, es decir observar y analizar las cosas como son “Objetos” nos ayudara a ver de una perspectiva mejor las cosas.

Las diferentes etapas o ciclos serán basados en el método interactivo o incremental (RUP), porque las diferentes etapas son aplicadas respectivamente.

Para la representación de la arquitectura se utilizará el Lenguaje Unificado De Modelado UML, asimismo es una notación estándar aprobada por el Object Management Group (OMG).

Por tanto la metodología a ser utilizada en el desarrollo, está enfocado en el Proceso Unificado Racional RUP, que utiliza técnicas de desarrollo a través del lenguaje Unificado de Modelado (Unified Modeling Lenguaje UML). [RUP, 2007]

### **2.8.1. Proceso Unificado De Rational “RUP”.**

El proceso Unified de Rational, es un proceso que provee un enfoque ordenado para planificar y asignar tareas dentro de un proyecto informático, captura algunas de las mejores prácticas de desarrollo, de forma que es adaptable a un amplio rango de organizaciones. La empresa que desarrollo este proceso es Rational Software y una de las cualidades de esta empresa es que los desarrolladores trabajan de manera muy estrecha con los clientes.

El RUP tiene grandes ventajas en su utilización porque permite: documentar todas las etapas del ciclo de vida, utiliza de manera extensa al Lenguaje de Modelado Unificado (UML),

ordena las tareas y brinda herramientas de software para su aplicación. Este proceso enfatiza en la creación y el mantenimiento de modelos antes que la generación de grandes documentos sobre papel, este proceso puede aplicarse a una pequeña empresa como a una grande.

Rational Unified Process (RUP) es un proceso de Ingeniería de Software planteado por Kruchten (1996) cuyo objetivo es producir software de alta calidad. es decir:

- ⇒ Cumple con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos.
- ⇒ Proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo de software.
- ⇒ Utiliza Unified Modeling Language (UML), para especificar, visualizar, construir y documentar componentes de sistemas de software, así también permite especificar modelos de negocio y otros sistemas.
- ⇒ Es un proceso controlado de iteraciones con fuerte focalización hacia la arquitectura.

El proceso intenta cubrir 6 de las mejores prácticas en el desarrollo de software, que son las siguientes:



*Figura 2.7: Incremento de la Productividad en Equipo  
Fuente: [JACOBSON, 2000]*

**a. Desarrollar software interactivamente**

Actualmente ya no es posible desarrollar software de manera lineal hasta la entrega del producto, sino que pueden realizarse entregas parciales llamadas interacciones, lo cual trae una serie de beneficios.

**b. Administrar requerimientos**

El proceso describe como relevar, organizar y documentar la funcionalidad requerida además de documentar las alternativas y decisiones y comunicar los requerimientos del negocio. Para este propósito se utilizarán los casos de uso y los escenarios que han demostrado ser un excelente medio para capturar los requerimientos funcionales que permiten los diseños, implementación y testeo del software.

**c. Usar arquitecturas basadas en componentes**

Describe como diseñar una arquitectura elástica que es flexible, el proceso soporta el desarrollo de software basado en componentes, entendiéndose a los componentes como módulos no triviales o subsistemas que cumplen con una función clara.

**d. Modelar software visualmente**

Guía a una modelización visual capturando la estructura y componentes escondiendo los detalles. Este nivel de abstracción ayuda a comunicar diferentes aspectos del software, ver como los elementos del software se adaptan unos a otros, mantienen la consistencia entre el diseño y su implementación.

**e. Verificar la calidad del software**

La calidad del software debe ser revisada respecto a los requerimientos de confiabilidad, funcionalidad de las aplicaciones y del sistema.

## f. Controlar los cambios al software

En un ambiente en donde los cambios son inevitables, la posibilidad de buscar y documentar los cambios, es esencial. El RUP describe como controlar, buscar y monitorear los cambios, para un desarrollo exitoso. También provee la guía para controlar los cambios en modelos, ejemplos documentos, código, etc.

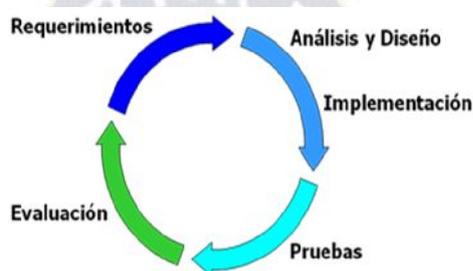


Figura 2.8: Desarrollo Iterativo  
Fuente: [JACOBSON, 2000]

### 2.8.2. Fases y Descripción Del Proceso.

El proceso puede ser descrito a lo largo de 2 ejes: el eje vertical representa los aspectos estáticos del proceso en cuanto al eje horizontal representa el tiempo el cual es expresados en términos de ciclos, fases, iteraciones y punto de control como se muestra en el grafico siguiente.

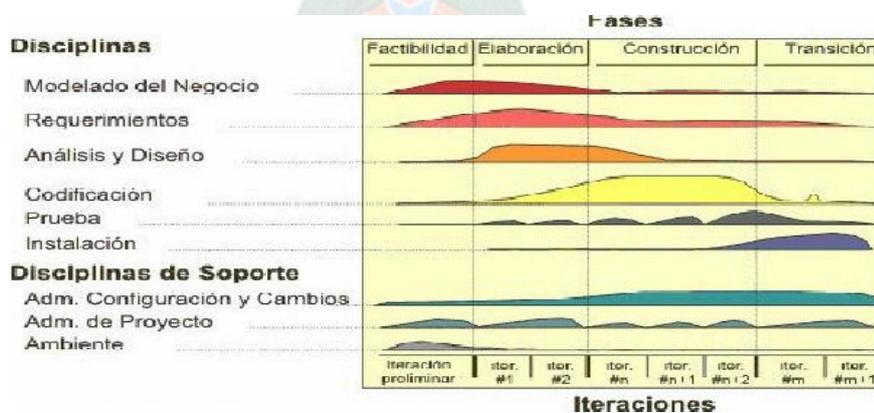


Figura 2.9: Fases del RUP  
Fuente: Fuente: [JACOBSON, 2000]

### **2.8.2.1. Fase De Iniciación: Establecer la Viabilidad**

Establece la planificación del negocio y se limita el alcance, se identifican las entidades externas. Se incluye la identificación de todos los casos de uso, criterios de éxito, riesgo, análisis y estimación de los recursos necesario.

### **2.8.2.2. Fase de Elaboración: Factibilidad**

Analiza el dominio del problema, establece una base arquitectónica sólida, desarrolla un plan del proyecto y elimina los elementos de mayor riesgo. Se construye un prototipo de una o más iteraciones dependiendo de la magnitud del proyecto. Los casos de uso son realizados hasta un 80%, la lista de riesgo revisada y se especifica el proceso de desarrollo a utilizar.

### **2.8.2.3. Fase de Construcción: Desarrollo**

Es donde todos los componentes son desarrollados e integrados al producto y toda la funcionalidad es completamente testeada. En esta fase se decide el lugar donde se instalara y si los usuarios están preparados para empezar a ser funcionar el software.

### **2.8.2.4. Fase de Transacción: En el entorno del Usuario**

Se transfiere el sistema a los usuarios donde habitualmente surgen requerimientos que justifican nuevas versiones, corregir algunos problemas no detectados o conclusión de procedimientos que había sido propuestos. Es aquí donde se establece si el usuario está satisfecho con el proyecto desarrollado [JACOBSON, 2000].

### **2.8.2.5. Ciclo de Vida del RUP.**

El proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, cada ciclo constituye con una versión del prototipo para los clientes. Así mismo, cada ciclo consta de cuatro fases: como se ha mencionado anteriormente [RATIONAL, 2000].

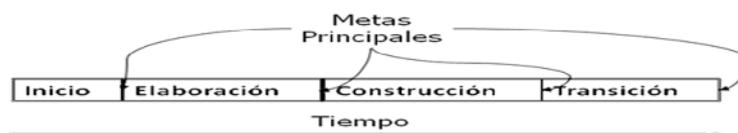


Figura 2.10: Metas del RUP  
Fuente: [RUP, 2007]

### 2.8.2.6. Flujo de Trabajo.

El eje vertical representa el aspecto estático del proceso; como está descrito en términos de actividades, artefactos, trabajadores y flujos de trabajo. [RUP, 2007].

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

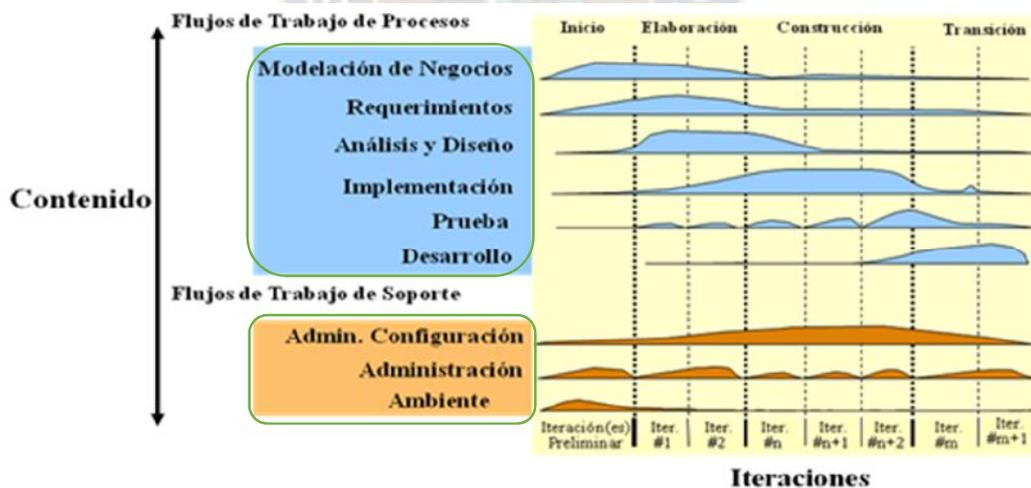


Figura 2.11: Iteraciones  
Fuente: I. Jacobson, G. Booch, J. Rumbauch

- ⇒ **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- ⇒ **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen en el desarrollo del software.
- ⇒ **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ⇒ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ⇒ **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- ⇒ **Instalación:** Produce realce del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- ⇒ **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ⇒ **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ⇒ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

## 2.9. El Lenguaje Unificado de Modelado (UML)

El lenguaje de modelado (UML), “Unified Modeling Language” es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo del Software. UML proporciona una forma de modelar cosas conceptuales, tales como procesos de negocio y funciones del sistema además de cosas concretas como lo son escribir clases en un lenguaje determinar, esquemas de bases de datos y componentes software. UML es un lenguaje muy expresivo.

Los modelos de UML con los que vamos a trabajar son los siguientes:

- ⇨ Diagrama de Estructura Estática.
- ⇨ Diagrama de Casos de Uso.
- ⇨ Diagrama de Secuencia.
- ⇨ Diagrama de Colaboración.
- ⇨ Diagrama de Estados.
- ⇨ Diagrama de Paquetes.

### 2.9.1. Clases.

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase. En la figura se ve

cómo una misma clase puede representarse a distinto nivel de detalle según interese, y según la fase en la que se esté.

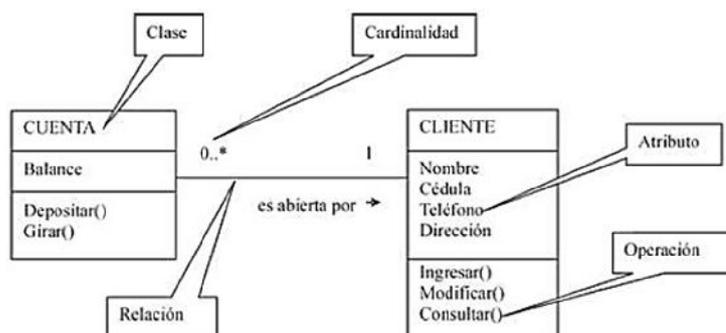


Figura 2.12: Ejemplo de un Diagrama de Clases

Fuente: Elaboración Propia

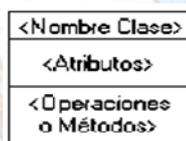


Figura 2.13: Ejemplo de Estructura de una Clase

Fuente: Elaboración Propia

## 2.9.2. Objetos.

Un objeto se representa de la misma forma que una clase. En el compartimento superior aparece el nombre del objeto junto con el nombre de la clase subrayados.

Puede representarse un objeto sin un nombre específico, entonces sólo aparece el nombre de la clase.

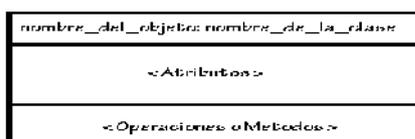


Figura 2.14: Ejemplo de Estructura de un Objeto

Fuente: Elaboración Propia

### 2.9.3. Asociaciones.

Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación se verán los más importantes de entre dichos elementos gráficos.

#### 2.9.3.1. Multiplicidad.

La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase. Puede expresarse de las siguientes formas:

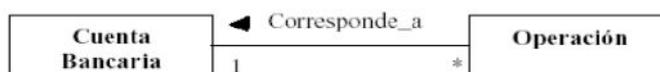


Figura 2.15: Ejemplo de multiplicidad en asociaciones.

Fuente: I. Jacobson, G. Booch, J. Rumbauch

#### 2.9.3.2. Agregación.

El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”.

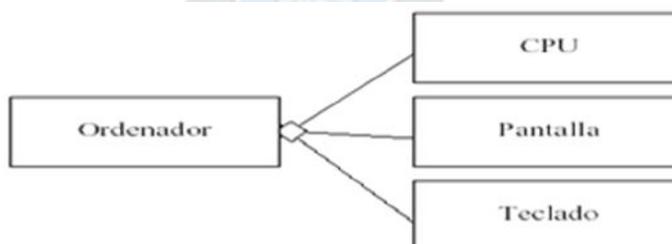


Figura 2.16: Ejemplo de Agregación

Fuente: I. Jacobson, G. Booch, J. Rumbauch

### 2.9.3.3. Herencia.

La relación de herencia se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase “padre”.

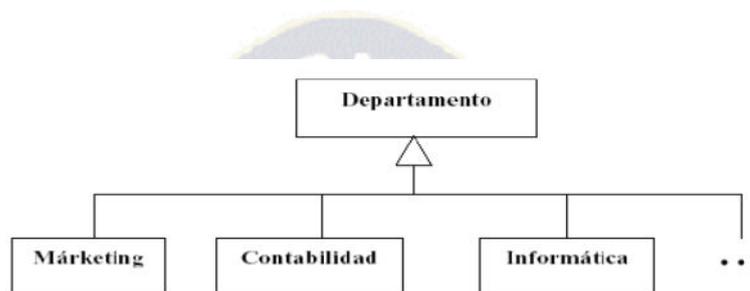


Figura 2.17: Ejemplo de Herencia  
Fuente: I. Jacobson, G. Booch, J. Rumbauch

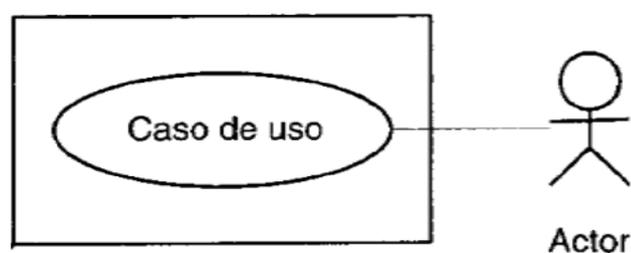
### 2.9.4. Diagrama de Casos De Uso.

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

- ⇒ ELEMENTOS: Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: Actores, casos de uso y relaciones entre casos de uso.
- ⇒ ACTORES: Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.). [ADDISON, 1999].
- ⇒ RELACIONES ENTRE CASOS DE USO: Entre dos casos de uso puede haber las siguientes relaciones:

- Extiende: Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- Usa: Cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior.



*Figura 2.18: Ejemplo de Caso de Uso  
Fuente: UML y Patrones [Craig Larman]*

### **2.9.5. Modelo Conceptual.**

Una parte de la investigación sobre el dominio del problema consiste en identificar los conceptos que lo conforman. Para representar estos conceptos se va a usar un Diagrama de Estructura Estática de UML, al que se va a llamar Modelo Conceptual.

El objetivo de la creación de un Modelo Conceptual es aumentar la comprensión del problema. Por tanto, a la hora de incluir conceptos en el modelo, es mejor crear un modelo con muchos conceptos que quedarse corto y olvidar algún concepto importante.

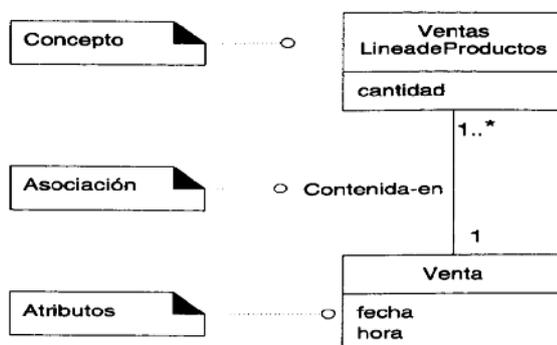


Figura 2.19: Ejemplo de Modelo Conceptual Parcial  
Fuente: UML y Patrones [Craig Larman]

### 2.9.6. Diagrama de Secuencia.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. [ADDISON, 1999].

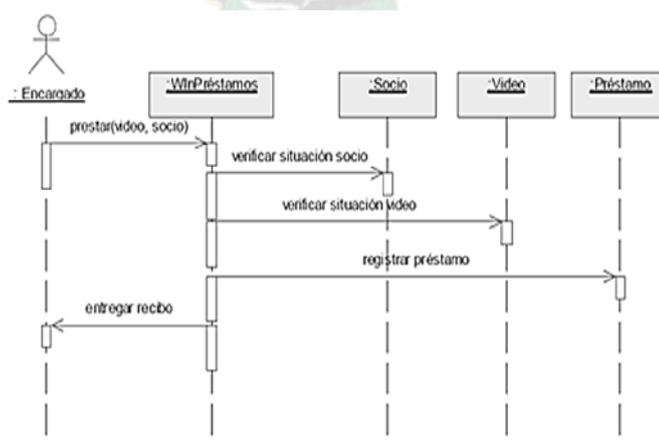


Figura 2.20: Ejemplo de Diagrama de Secuencia  
Fuente: UML y Patrones [Craig Larman]

### **2.9.7. Implementación y Pruebas.**

Una vez se tiene completo el Diagrama de Clases de Diseño, se pasa a la implementación en el lenguaje de programación elegido.

El programa obtenido se depura y prueba, y ya se tiene una parte del sistema funcionando que se puede probar con los futuros usuarios, e incluso poner en producción si se ha planeado una instalación gradual.

## **2.10. Tecnologías de Software**

### **2.10.1. Tecnologías para el Sistema de Gestión de Bases de Datos (SGBD).**

Son un conjunto de programas (en realidad, un sistema software) de propósito general que facilita el proceso de definición, construcción y manipulación de bases de datos para usos diversos.

- Definición. Para especificar tipos de datos, estructuras de datos y restricciones de los datos.
- Construcción. Para guardar los datos en un dispositivo de almacenamiento controlado por el SGBD.
- Manipulación. Para poder consultar y actualizar la información.

Un Sistema De Gestión De Base De Datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. Los sistemas de base de datos se diseñan para gestionar grandes cantidades de información. El modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones de los datos, la semántica y las ligaduras de consistencia. Los diferentes modelos de datos se clasifican en tres grupos diferentes:

### ➤ Modelos lógicos basados en objetos

Se usan para describir datos en los niveles lógicos y de vista, se caracteriza por proporcionar capacidad estructural muy flexible. Varios de los más aplicables conocidos son:

- ⇨ El modelo de Entidad Relacional
- ⇨ El modelo Orientado A Objetos
- ⇨ El modelo de datos semánticos
- ⇨ El modelo de datos funcional

#### **2.10.1.1. *MySql.***

MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. Y así ocurre. MySQL ha pasado de ser una "pequeña" base de datos a una completa herramienta y ha conseguido superar a una gran cantidad de bases de datos comerciales (lo que ha asustado a la mayor parte de los proveedores comerciales de bases de datos).

MySQL fue creada por la empresa sueca MySQL LAB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQLAB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.[THOMS,03].

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

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

### **2.10.2. Lenguajes de Programación**

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis. También se puede decir que un programa es un conjunto de órdenes o instrucciones que resuelven un problema específico basado en un Lenguaje de Programación.

- **Lenguajes de Alto Nivel:** Para acelerar aún más el proceso de programación se desarrollaron los lenguajes de alto nivel, en los que se puede escribir un sólo enunciado para realizar tareas sustanciales. Los lenguajes de alto nivel permiten a los programadores escribir instrucciones que asemejan al inglés cotidiano y contiene notaciones matemáticas de uso común. Estos lenguajes en la actualidad son de una gran variedad y tienen diferentes utilidades (Php, Asp, Java, SQL, HTML, etc.).

Ejemplo: echo (“Hola”);  
Print (“Hasta Luego”);  
If (A>B)

### **2.10.2.1.Php.**

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

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de correo electrónico, subida de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales. [SUEHRING S., CONVERSE T., PARK J., 2009]



Figura 2.21: Esquema del funcionamiento de las páginas PHP.

Fuente: DesarrolloWeb

Para colocar código Php en cualquier página HTML debe tener las siguientes etiquetas:

`<?Php` o `<?` Estas etiquetas indican el comienzo de un fragmento de código escrito en Php.

`?>` Esta etiqueta indica el término de un fragmento de código escrito en Php.

### 2.10.2.2. HTML.

HTML significa HyperText Markup Language. Es el lenguaje en que se escriben los millones de documentos que hoy existen en el World Wide Web (Internet). Cuando accedemos a uno de estos documentos, el cliente (Netscape, IE, Mozilla, Opera, etc.) los interpreta y los

despliega. Crear una buena página tiene dos aspectos; por un lado el conocimiento técnico para crear código HTML correcto, por otro lado el claro diseño para presentar la información.

Una página web es un archivo de texto, se puede crear con cualquier editor de texto como el bloc de notas de Windows. El principio esencial del lenguaje HTML (HyperText Markup Language) es el uso de las etiquetas (tags). Funcionan de la siguiente manera:

*<XXX> Este es el inicio de una etiqueta.*

*</XXX> Este es el cierre de una etiqueta.*

Las letras de la etiqueta pueden estar en mayúsculas o minúsculas, indiferentemente.

Todo el documento HTML debe estar entre las etiquetas `<HTML>` y `</HTML>`:

*<HTML> [Todo el documento] </HTML>*

Una página web la vemos en nuestro navegador, o cliente web, y parece una sola entidad, pero no es así, está compuesta por multitud de diferentes ficheros, como son las imágenes, los posibles vídeos y lo más importante: el código fuente. El código de las páginas está escrito en un lenguaje llamado HTML, que indica básicamente donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.

### **2.10.2.3. Java Script.**

JavaScript es un lenguaje de script, interpretado, orientado a objetos. Aunque tiene menos capacidad que un lenguaje totalmente orientado a objetos como C++ y Java, JavaScript es lo suficientemente potente para los propósitos que requerimos.

JavaScript no es una versión limitada de otro lenguaje, y no es una simplificación de cualquier otro. Es por lo tanto, limitado. No se pueden escribir aplicaciones autónomas, por ejemplo, y tiene una capacidad muy limitada para leer y escribir ficheros. Es más, los script (o guiones) de JavaScript no se pueden ejecutar sin la presencia de un intérprete, bien sea en un servidor Web o en un navegador (o browser).

JavaScript es un lenguaje sin tipo de datos. Es decir, no necesitas declarar el tipo de datos de las variables explícitamente. Es más, en muchos casos JavaScript realiza la conversión automáticamente cuando sea necesario. Por ejemplo, si quieres añadir un número a un campo o ítem que es una cadena de caracteres (o string), el número se convierte a texto. [ANTONIO DE ÁNGEL, 1998].

El programa en JavaScript debe ir encerrado entre la marca script e inicializada la propiedad lenguaje con la cadena javascript:

```
<script language="javascript">
```

*Aquí va el código JavaScript*

```
</script>
```

#### **2.10.2.4. CSS.**

CSS, es una tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores... CSS son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada.

Las hojas de estilo es una tecnología que nos permite controlar la apariencia de una página web.

CSS describe como los elementos dispuestos en la página son presentados al usuario. CSS es un gran avance que complementa el HTML y la Web en general.

Con CSS podemos especificar estilos como el tamaño, fuentes, color, espaciado entre textos y recuadros así como el lugar donde disponer texto e imágenes en la página.

#### **2.10.2.5. Tecnologías Basadas en Lenguajes.**

- ✓ Ajax: Son las siglas de Asynchronous JavaScript And XML. No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que nos permiten hacer páginas de internet más interactivas. La característica fundamental de AJAX es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar podemos enviar información al servidor.

La complejidad se encuentra en que debemos dominar varias tecnologías:

- HTML o XHTML
- CSS
- JavaScript
- DHTML Básicamente debemos dominar todos los objetos que proporciona el DOM.
- XML Para el envío y recepción de los datos entre el cliente y el servidor.
- PHP o algún otro lenguaje que se ejecute en el servidor (ASP.Net/JSP).

- ✓ JSON: Douglas Crockford, un experimentado ingeniero software, propuso un nuevo formato de datos construido sobre JavaScript llamado JSON, JavaScript Object Notation (notación de objetos JavaScript).

JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XML.

### **2.10.3. Frameworks.**

En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Los frameworks tienen como objetivo principal ofrecer una funcionalidad definida, auto contenido, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta. Esta funcionalidad, está constituida por objetos llamados fríos, que sufren poco o ningún cambio en la vida del framework, permitiendo la portabilidad entre distintos sistemas. Algunas características notables que se pueden observar:

- **Básicos:** No es más que una base de programación que atiende a sus descendientes (manejado de una forma estructural y/o en cascada), posibilitando cualquier respuesta ante las necesidades de sus miembros, o en secciones de una aplicación (web), satisfaciendo así las necesidades más comunes del programador.
- **Arquitectura:** Dentro de este aspecto, podemos basarnos en el modelo–vista–controlador o MVC (Controlador => Modelo => Vista), ya que debemos fragmentar nuestra programación. Tenemos que contemplar estos aspectos básicos en cuanto a la implementación de nuestro sistema:
  - ✓ **Modelo:** Este miembro del controlador maneja las operaciones lógicas, y de manejo de información para resultar de una forma explicable y sin titubeos. Cada miembro debe ser meticulosamente llamado, con su correcto nombre y en principio, con su verdadera naturaleza: el manejo de información, su complementación directa.
  - ✓ **Vista:** Al final, a este miembro de la familia le corresponde dibujar, o expresar la última forma de los datos: la interfaz gráfica que interactúa con el usuario final del programa (GUI). Después de todo, a este miembro le toca evidenciar la información obtenida hasta hacerla llegar al controlador. Solo (e inicialmente), nos espera demostrar la información.
  - ✓ **Controlador:** Con este apartado podemos controlar el acceso (incluso todo) a nuestra aplicación, y esto puede incluir: archivos, scripts, y/o programas; cualquier tipo de información que permita la interfaz. Así, podremos diversificar nuestro contenido de forma dinámica, y estática (a la vez); pues, solo debemos controlar ciertos aspectos.

**Una base (MVC):**

- ✓ Controlador: este debe ser capaz de manejar rutas, archivos, clases, métodos y funciones.
- ✓ Modelo: es como un script habitual en el servidor, solo que agrupado bajo un 'modelo' reutilizable.
- ✓ Vista: como incluyendo cualquier archivo en nuestra ejecución, muy simple.

**2.10.3.1. CodeIgniter.**

CodeIgniter es un framework para desarrollo de aplicaciones - un conjunto de herramientas - para gente que construye sitios web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como y una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. CodeIgniter le permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesaria para una tarea dada.

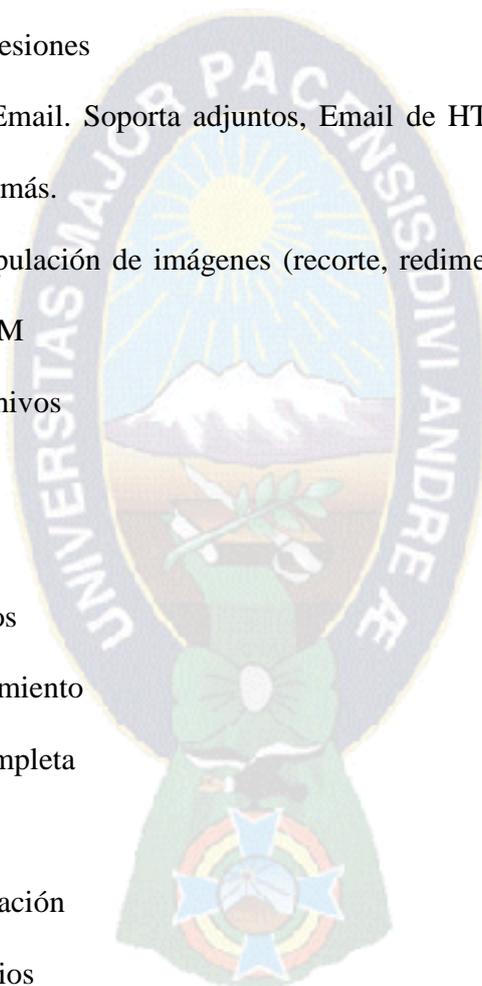
CodeIgniter está liberado bajo licencias open source del estilo Apache/BSD, así que se puede usar en cualquier lugar.

CodeIgniter usa el enfoque Modelo-Vista-Controlador, que permite una gran separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo.

Funcionalidades de CodeIgniter.

- Sistema basado en Modelo-Vista-Controlador
- Extremadamente liviano

- Clases de base de datos con soporte para varias plataformas
- Soporte para base de datos con Active Record
- Validación de datos y formularios
- Seguridad y filtrado XSS
- Administración de sesiones
- Clase para enviar Email. Soporta adjuntos, Email de HTML/Texto, varios protocolos (sendmail, SMTP, y Mail) y más.
- Biblioteca de manipulación de imágenes (recorte, redimensión, rotación, etc). Soporta GD, ImageMagick, y NetPBM
- Clase para subir archivos
- Clase para FTP
- Paginación
- Encriptación de datos
- Evaluación de rendimiento
- Caché de página completa
- Historial de errores
- Perfilado de la aplicación
- Clase para Calendarios
- Clase para Agente del Usuario
- Clase para codificación Zip
- Clase de motor de plantillas
- Biblioteca XML-RPC



- Clase para pruebas de unidad
- URLs amigables para los motores de búsqueda
- Ruteo URI flexible
- Soporte para hooks y extensiones de clase

El siguiente gráfico ilustra como los datos fluyen a través del sistema:

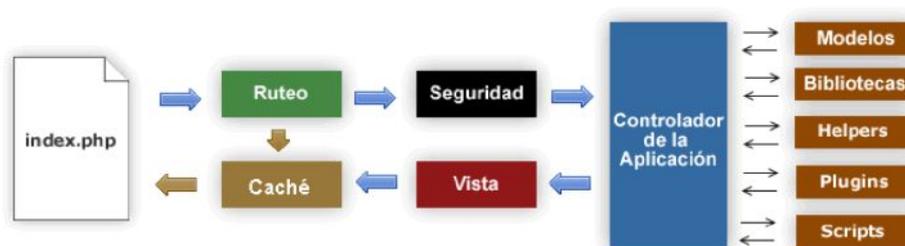


Figura 2.22: Flujo de trabajo de CodeIgniter  
Fuente: Guía del Usuario de CodeIgniter Versión 2.0.2

### 2.10.3.2. JQuery.

JQuery es un framework Javascript. Pues es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores lo utilizan para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio JQuery ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar.

jQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. [SAWYER D.2011]

- ✓ **JQGrid:** Es una librería dependiente de JQuery la cual nos permite elaborar de manera rápida y sencilla tablas de acceso a datos, esta librería usa JSON, AJAX, PHP, etc. para el acceso a la información de una base de datos. Las tablas creadas con esta librería son de fácil configuración, además cuenta con diferentes opciones para la forma de manipular los datos.

## 2.11. Herramienta de Modelado

### 2.11.1. Enterprise Architect.

Es una herramienta para realizar el modelado y diseño visual basado en la herramienta de OMG UML. La plataforma soporta: el diseño y construcción de sistemas de software; modelado de procesos de negocio; y la industria del modelaje dominios basados. Es utilizado por las empresas y organizaciones, no sólo para modelar la arquitectura de sus sistemas, pero para procesar la aplicación de estos modelos a través del completo ciclo de vida de desarrollo de aplicaciones.

Características Generales:

- ✓ La gestión de requisitos
- ✓ Modelado de negocios y análisis
- ✓ Simulación
- ✓ El desarrollo del sistema
- ✓ Gestión de pruebas
- ✓ El análisis visual de ejecución
- ✓ La ingeniería de sistemas
- ✓ El modelado de datos
- ✓ La gestión de proyectos
- ✓ La gestión del cambio

## 2.12. Metodología de Evaluación de La Calidad de un Sitio Web

El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente. La gestión del cambio independiente para cada contexto de uso a la hora de la entrega y del uso por parte de los usuarios. Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requisitos).

En la actualidad no están claras las ideas acerca de Calidad Web, existen varias alternativas para la valoración de un sitio Web. Entre las alternativas que presentan para el control de calidad de aplicaciones Web, está la propuesta por Luis Olsina, “Metodología Cuantitativa para la Evaluación y Comparación de la calidad de Sitios Web” [Olsina 1999]. Este, está basada en un modelo jerárquico de requerimientos de calidad, que toma las características presentadas en el modelo ISO – 9126(15), Esto permite que a partir de las características del modelo ISO- 9126, se divide en Sub-características y a partir de éstas, siguiendo un proceso de descomposición jerárquico se definen atributos. [POLO, 2004].

En 1992 fue publicado un estándar internacional para la evaluación de la calidad de productos de software, llamado “Tecnologías de Información –Calidad de los Productos Software”, también conocido como ISO 9126 (o ISO / IEC 1926). El mismo que tiene sus orígenes, en el modelo de calidad de producto propuesto por McCall.

El estándar ISO 9126 describe a la calidad de software a partir de seis características generales de los cuales Olsina refiere de la siguiente manera “Esas características proveen un alineamiento base para posteriores refinamientos y descripciones de la calidad de software”. En la Figura 2.14 se presenta una descripción sintética de las características y Sub-características del estándar

ISO por medio de la formulación de una pregunta central. [CASTRILLÓN A., DANIEL & CASTRO Q., GILDARDO. 2000].

- ✓ Parte 1: Modelo de Calidad
- ✓ Parte 2: Métricas Externas
- ✓ Parte 3: Métricas Internas
- ✓ Parte 4: Métricas de Calidad en Uso

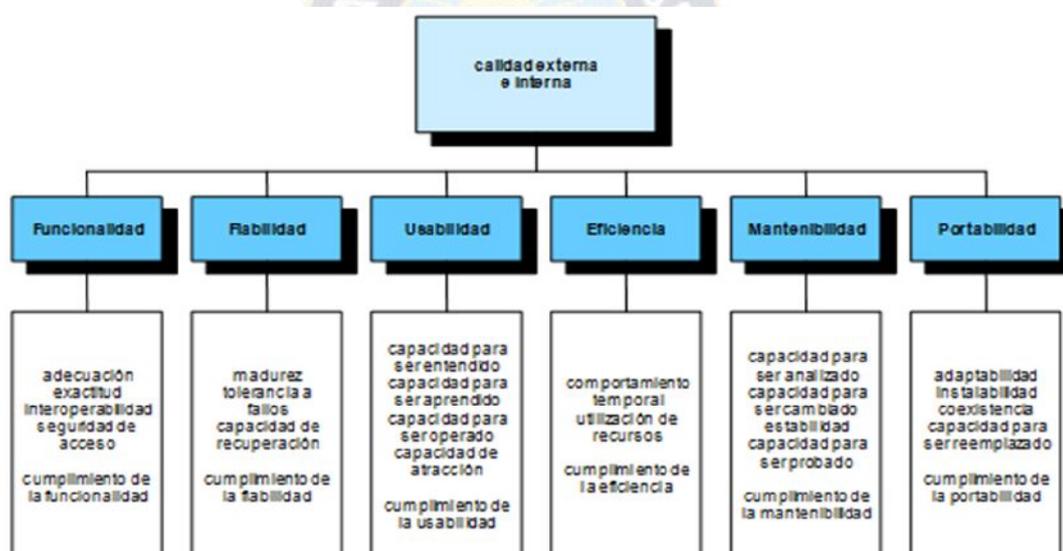


Figura 2.23: Calidad Interna y Externa  
 Fuente: [CASTRILLÓN A., DANIEL & CASTRO Q., GILDARDO. 2000].

### 2.12.1. Funcionalidad.

**Adecuación:** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

**Exactitud:** Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.

**Interoperabilidad:** Capacidad del producto software para interactuar con uno o más sistemas especificados.

**Seguridad de Acceso:** Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados

**Cumplimiento Funcional:** Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.

### 2.12.2. Fiabilidad.

**Madurez:** Capacidad del producto software para evitar fallar como resultado de fallos en el software.

**Tolerancia a Fallos:** Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.

**Capacidad de Recuperación:** Capacidad del producto software para reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.

**Cumplimiento de la Fiabilidad:** Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con a la fiabilidad.

### 2.12.3. Usabilidad.

**Capacidad para ser Entendido:** Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.

**Capacidad para ser Aprendido:** Capacidad del producto software que permite al usuario aprender sobre su aplicación.

**Capacidad para ser Operado:** Capacidad del producto software que permite al usuario operarlo y controlarlo.

**Capacidad de Atracción:** Capacidad del producto software para ser atractivo al usuario.

**Cumplimiento de la Usabilidad:** Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

#### 2.12.4. Eficiencia.

**Comportamiento Temporal:** Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.

**Utilización de Recursos:** Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

**Cumplimiento de la Eficiencia:** Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.

#### 2.12.5. Portabilidad.

**Adaptabilidad:** Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.

**Instalabilidad:** Capacidad del producto software para ser instalado en un entorno especificado.

**Coexistencia:** Capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes.

**Capacidad para Reemplazar:** Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.

**Cumplimiento de la Portabilidad:** Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad. [CASTRILLÓN A., DANIEL & CASTRO Q., GILDARDO. 2000].

### 2.13. Modelo De Calidad En Uso



*Figura: 2.24. Modelo de Calidad*  
*Fuente: [CASTRILLÓN A., DANIEL & CASTRO Q., GILDARDO. 2000].*

**Efectividad:** Capacidad del producto software para permitir a los usuarios alcanzar objetivos especificados con exactitud y completitud, en un contexto de uso especificado.

**Productividad:** Capacidad del producto software para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada, en un contexto de uso especificado.

**Seguridad física:** Capacidad del producto software para alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente en un contexto de uso especificado.

**Satisfacción:** Capacidad del producto software para satisfacer a los usuarios en un contexto de uso especificado. [SILVA, 2000].

## 2.14. Modelo de Constructivo de Costo: COCOMO

Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica las economías y deseconomías de escala producidas a medida que un proyecto de software incrementa su tamaño.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiendo detalladamente en su libro “Software Engineering Economics”

<b>Aplicaciones desarrolladas por usuarios finales</b>		
<b>Generadores de Aplicaciones</b>	<b>Aplicaciones con Componentes</b>	<b>Sistemas Integrados</b>
<b>Infraestructura</b>		

*Figura 2.25: Mercado de Software Actual y Futuro  
Fuente: [Boehm 1995]*

Este modelo pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en líneas de código principalmente.

## 2.15. Conclusión

En este capítulo se realizó una revisión conceptual de algunos lenguajes, metodología y herramientas que existen para el desarrollo de un sistema, conociendo esto aplicaremos la metodología Proceso Unificado de Rational (RUP) para así permitirnos documentar todas las etapas del ciclo de vida, que utiliza de manera extensa al Lenguaje Unificado de Modelado UML para modelar cosas conceptuales del software, así también para el modelado de los datos tenemos a Enterprise Architect una de las herramientas poderosa que trabaja junto a UML.



## Capítulo III: Marco Aplicativo

### 3.1. Introducción

En el presente capítulo se describe todo el proceso de análisis, diseño e implementación del “Sistema de Información para el Control y Seguimiento de Ventas On-Line de Productos Farmacéuticos Distribuidora Pharmica” siguiendo las fases de factibilidad, elaboración, construcción y transición aplicadas a la metodología RUP (Proceso Unificado Racional), utilizando este modelo para la evaluación del trabajo en sus diferentes flujos de trabajo.

El análisis consiste en obtener una perspectiva, antes de que los requisitos puedan ser analizados, modelados o especificados, deben ser recogidos a través de un proceso de obtención de requisitos como el despliegue de función de calidad.

### 3.2. Fase de Inicio

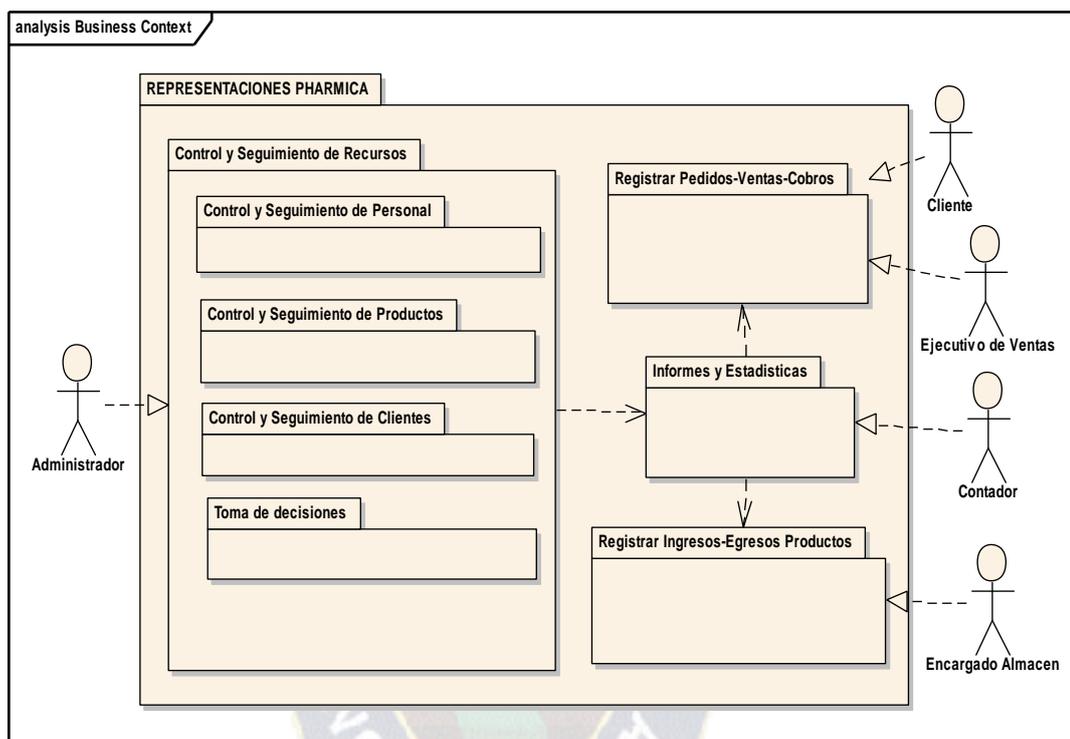
#### 3.2.1. Modelado del Negocio

El Modelado de Negocio se utiliza para describir las organizaciones de negocios en el mundo real y para tener una mejor comprensión de las situaciones reales, más que para la complementación del software.

Para tener una mejor comprensión se realiza su abstracción mediante diagramas de paquetes.

En el paquete núcleo se describen las principales estructuras estáticas de UML. Entre ellas se encuentran sus clasificadores, sus contenidos (atributos, operaciones, métodos y parámetros) y sus relaciones.

En la lógica de Negocios se definen los componentes que contienen las definiciones de las operaciones que son necesarias para que nuestro sistema cumpla con las funciones establecidas.



*Figura 3.1: Modelado del Negocio Mediante Paquetes  
Fuente: Elaboración Propia*

### 3.2.2. Modelado de Casos de Uso

El modelado de casos de uso del negocio describe a la empresa o institución en su forma de funcionamiento, en la figura 3.2 Funcionamiento Actual Representaciones Pharmica, muestra a los actores, el conjunto de casos de uso y sus relaciones. Este modelado cubre la vista estática de los casos de uso y son especialmente importantes para el modelado de la organización y su funcionamiento.

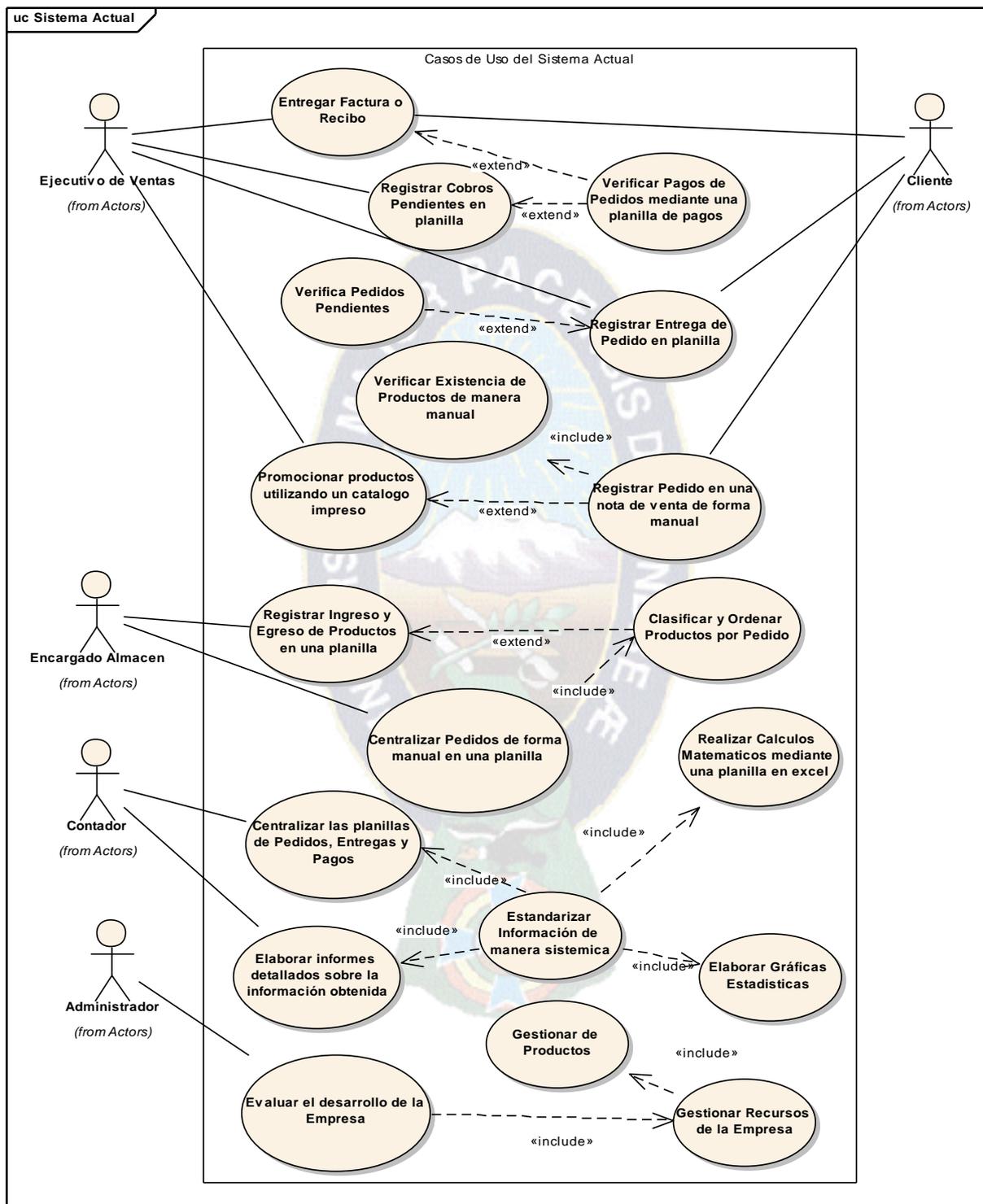


Figura 3.2: Diagrama de Casos de Uso del Negocio  
 Fuente: Elaboración Propia

### 3.2.3. Descripción de Casos de Uso

**Tabla 3.1:** Descripción de Casos de Uso Registra Cobros Pendientes en Planillas

<b>Caso de Uso:</b>	Registra Cobros Pendientes en Planillas	
<b>Actores:</b>	Ejecutivo de Ventas, Cliente	
<b>Propósito:</b>	Registrar los abonos en efectivo cancelados por el Cliente	
<b>Resumen:</b>	El Ejecutivo de Ventas realiza el cobro de saldos pendientes por parte del Cliente y lo registra en la planilla de cobros.	
<b>Referencia Cruzadas</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema Actual</b>	
1. El Ejecutivo de Ventas Verifica si el Cliente tiene saldos pendientes de los pedidos anteriores. 3. a El Ejecutivo hace conocer al Cliente sus saldos pendientes. 3.b El Cliente no tiene deudas pendientes 4.a El Cliente solicita realizar la cancelación parcial o total de su saldo 5.a El Ejecutivo de Ventas Verifica el monto a ser cancelado.  7.a El Ejecutivo de Ventas entrega un recibo que constata la cancelación parcial del pedido o en caso de cancelación total se entrega la factura.	2. Verifica de forma manual la planilla de pagos antes de realizar un cobro       6.a Registra en la planilla de pagos el monto, la fecha, y el código de pedido al que corresponde el pago.	

*Fuente: Elaboración propia*

**Tabla 3.2:** Descripción de Casos de Uso Registra Entrega de Pedido en planilla

<b>Caso de Uso:</b>	Registra Entrega de Pedido en planilla	
<b>Actores:</b>	Ejecutivo de Ventas y Cliente	
<b>Propósito:</b>	Entrega de Productos Solicitados por el Cliente	
<b>Resumen:</b>	El Ejecutivo de Ventas realiza el registro de las entrega de los productos solicitados a los Clientes.	
<b>Referencia Cruzadas</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema Actual</b>	
1. El Ejecutivo de Ventas verifica si existen pedidos de productos pendientes a entregar al Cliente. 3.a El Cliente tiene registrado un pedido pendiente 3.b El Cliente no tiene pedidos pendientes  5.a El Ejecutivo de Ventas sella como entregado la nota de venta.	2. La verificación se realiza mediante las órdenes de Pedido o notas de venta.  4.a Registra la entrega del pedido al Cliente y el tipo de pago en planilla.	

*Fuente: Elaboración propia*

**Tabla 3.3:** Descripción de Casos de Uso Centraliza Pedidos de forma manual en una planilla

<b>Caso de Uso:</b>	Centraliza Pedidos de forma manual en una planilla	
<b>Actores:</b>	Encargado Almacén	
<b>Propósito:</b>	Tener todos los pedidos centralizados para su preparación	
<b>Resumen:</b>	El Encargado del Almacén debe centralizar los pedidos de productos registrados por los Ejecutivos de Ventas.	
<b>Referencia Cruzadas</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema Actual</b>	

<ol style="list-style-type: none"> <li>1. El Encargado de Almacén recaba una copia de todos los pedidos registrados.</li> <li>2. El Encargado de Almacén Organiza los pedidos en orden de registro.</li> </ol>	<ol style="list-style-type: none"> <li>3. Realiza una sumatoria por producto.</li> <li>4. Elabora informe de manera manual sobre la cantidad de productos solicitados por día.</li> </ol>
--	---

*Fuente: Elaboración propia*

**Tabla 3.4:** Descripción de Casos de Uso Estandariza la información de manera sistémica

<b>Caso de Uso:</b>	Estandariza la información de manera sistémica
<b>Actores:</b>	Contador
<b>Propósito:</b>	facilitar el análisis de la información
<b>Resumen:</b>	La información generada es clasificada por sus propiedades y valores cuantitativos de las distintas fuentes de información (cantidad de productos vendidos, solicitados, en existencia, también la cantidad monetaria de ventas, pedidos , etc)
<b>Referencia Cruzadas</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema Actual</b>
<ol style="list-style-type: none"> <li>1. El Contador solicita copias de las diferentes planillas (ventas, pedidos, pagos, etc).</li> <li>2. El contador centraliza la información adquirida.</li> </ol>	<ol style="list-style-type: none"> <li>2. El personal de Pharmica debe presentar informes con fotocopias de las planillas que les corresponden.</li> <li>4. Se llena la información adquirida en hojas de cálculo elaboradas en Excel de manera manual.</li> </ol>

*Fuente: Elaboración propia*

**Tabla 3.5:** Descripción de Casos de Uso Elaborar Gráficas Estadísticas

<b>Caso de Uso:</b>	Elaborar Gráficas Estadísticas
<b>Actores:</b>	Contador
<b>Propósito:</b>	Mostrar el análisis de la información de manera sencilla y clara
<b>Resumen:</b>	El contador elabora un conjunto de gráficas estadísticas de diferentes tipos para que el administrador analice la información tome decisiones
<b>Referencia Cruzadas</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema Actual</b>
<ol style="list-style-type: none"> <li>1. El Contador utiliza la información estandarizada.</li> <li>3. El contador recaba los cálculos realizados con la información estandarizada.</li> <li>5. El contador toma los valores cuantitativos, las diferentes variables del contexto y selecciona un tipo de gráfica para elaborar la gráfica estadística.</li> </ol>	<ol style="list-style-type: none"> <li>2. La información estandarizada se encuentra en hojas de cálculo en Excel.</li> <li>4. Los cálculos de la información estandarizada se encuentran en hojas de cálculo en Excel.</li> <li>6. Se tiene gráficas de análisis anteriores que se usan como modelos de referencia para las gráficas.</li> </ol>

*Fuente: Elaboración propia*

### 3.2.4. Descripción de los Actores

La identificación de actores en términos generales son usuarios o personal que tienen acceso al sistema actual los cuales interactúan, aportan y reciben información del sistema para coadyuvar a sus tareas cotidianas o necesidades demandadas dentro la labor que realizan en la institución.

A continuación se da una lista de los actores o usuarios identificados.

**Cliente:** Es la persona que inicia la actividades relacionadas con la venta de productos. Sus funciones son:

- Solicita productos.
- Realiza el pago de los pedidos.
- Cancela pedidos.

**Ejecutivo de Ventas:** Es la persona que registra la mayoría de la información generada. Sus funciones son:

- Registra pedidos.
- Entrega Pedidos.
- Registra Pagos.
- Registra Clientes.
- Realiza la promoción de los productos.

**Encargado Almacén:** Es la persona que controla el ingreso y egreso de productos de los almacenes de la empresa. Sus funciones son:

Centraliza los pedidos para que sean preparados.

- Registra el ingreso de productos al almacén.
- Registra el egreso de productos del almacén.
- Elabora informes diarios de Ingreso y Egreso de productos.

**Contador:** Es la persona que necesita tener acceso amplio a las diferentes planillas que se utilizan en la empresa. Sus funciones son:

- Centraliza las planillas de pedidos, pagos, ingreso de productos, egreso de productos, productos y precios.
- Elabora informes detallados sobre los movimientos de los recursos de la empresa.

**Administrador:** Es la persona que necesita tener toda la información de los movimientos de la empresa y los recursos con los que cuenta. Sus funciones son:

- Gestionar productos.
- Gestionar personal.
- Toma de Decisiones.

### 3.2.5. Análisis de Requerimientos

En el proceso de desarrollo se identifican dos tipos de requerimientos que son los requerimientos funcionales y los requerimientos no funcionales.

#### A). Requerimientos Funcionales

La descripción de requerimientos funcionales del sistema se presenta mediante casos de uso y actores, en esta oportunidad los expresaremos de forma narrativa.

Casos de Uso que se han identificado son los siguientes:

- ⇒ Caso de Uso Centralizar pedidos, productos, pagos: al realizar el centralizado de esta información tiene una demora significativa lo que evita que se puedan tomar decisiones oportunas.
- ⇒ Casos de Uso Realizar Cálculos Matemáticos mediante una planilla en Excel: Para elaborar las gráficas estadísticas se necesita que la información centralizada deba estar procesada mediante sumatorias, diferencias, productos y otras operaciones matemáticas ya que este método realice el seguimiento correspondiente a la información recogida.

- Caso de Uso Estandarizar Información de manera sistémica: Es necesario definir las variables y englobarlas para poder obtener tablas totales o parciales de los pedidos, pagos, entregas, ingreso y egreso.

## **B). Requerimientos No Funcionales**

Los requerimientos no funcionales especifican atributos, cualidades o requerimientos del sistema, como son los requerimientos de Hardware, Software y Plataforma.

### Requerimientos de Software

Los requerimientos de software para el desarrollo del sistema son:

- Sistema Operativo Windows 8 o Ubuntu 12.10.
- Xampp Server o Lampp Server
- Mysql Server 5.5
- Php 5

### Requerimiento de Hardware

El requerimiento de Hardware

- 1 equipo computacional con las siguientes características (En caso de que el sistema funcione de manera local).
  - Procesador Dual Core (mínimo)
  - Disco duro de 320 Gb(mínimo)
  - Memoria RAM de 4Gb(mínimo)
  - Tarjeta de red
- Para el servidor se evalúa los servicios de hosting y dominio de pago existentes en Bolivia que cumplan con los requerimientos del sistema.
- 4 equipos computacionales con las siguientes características (Clientes).

- Procesador Pentium 4 (mínimo)
- Disco duro de 40 Gb(mínimo)
- Memoria RAM de 512 Mb(mínimo)
- Tarjeta de red
- Acceso a Internet

### 3.3. Fase de Elaboración

Para iniciar esta fase se tiene conocimiento del negocio y su funcionamiento, que se va ampliando con cada iteración, esta información significativa para el desarrollo de la fase de elaboración, seguidamente puntualizara en la mayoría de los casos de uso y se diseñara la arquitectura del sistema las cuales se expresaran en vistas arquitectónicas del modelo de casos de uso, del modelo de análisis y del modelo del diseño.

#### 3.3.1. Identificación de Actores del Nuevo Sistema

Durante el análisis de requerimientos se identifican a los actores que intervendrán en el funcionamiento del nuevo sistema.

**Tabla 3.6:** Descripción de actores del Nuevo Sistema

Actor	Descripción
Administrador del Sistema	Está facultado para hacer modificaciones a la interfaz del Sistema, como un Web máster, Crear, Ver, Modificar, Eliminar, los usuarios del Sistema, modificar los roles existentes.
Administrador	Está facultado para acceder en modo de solo lectura a toda la información registradas en la base de datos y los módulos de consulta y estadística existentes en el sistema.

Contador	Está facultado para acceder en modo de solo lectura a la información registrada en la base de datos referente a pedidos, pagos, entregas, ingresos-egresos de productos y los módulos de consulta y estadística existentes en el sistema.
Encargado de Almacén	Está facultado para acceder en modo de solo lectura a la información registrada en la base de datos referente a pedidos y tiene privilegios de modificación en la información ingreso de productos, egreso de productos y productos.
Ejecutivo de Ventas	Está facultado para añadir, eliminar y modificar pedidos, también para añadir pagos, añadir entregas.
Cliente	Está facultado para añadir, eliminar y modificar pedidos.

*Fuente: Elaboración Propia*

### **3.3.2. Modelo de Casos de Uso del Nuevo Sistema**

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan del sistema y como se relacionan con su entorno (usuarios).

El modelo de casos de uso nos permite representar a los diferentes actores, los casos de uso y las interacciones que soporta el software, por tanto diseñaremos estos casos de uso de cada usuario con su respectiva documentación.

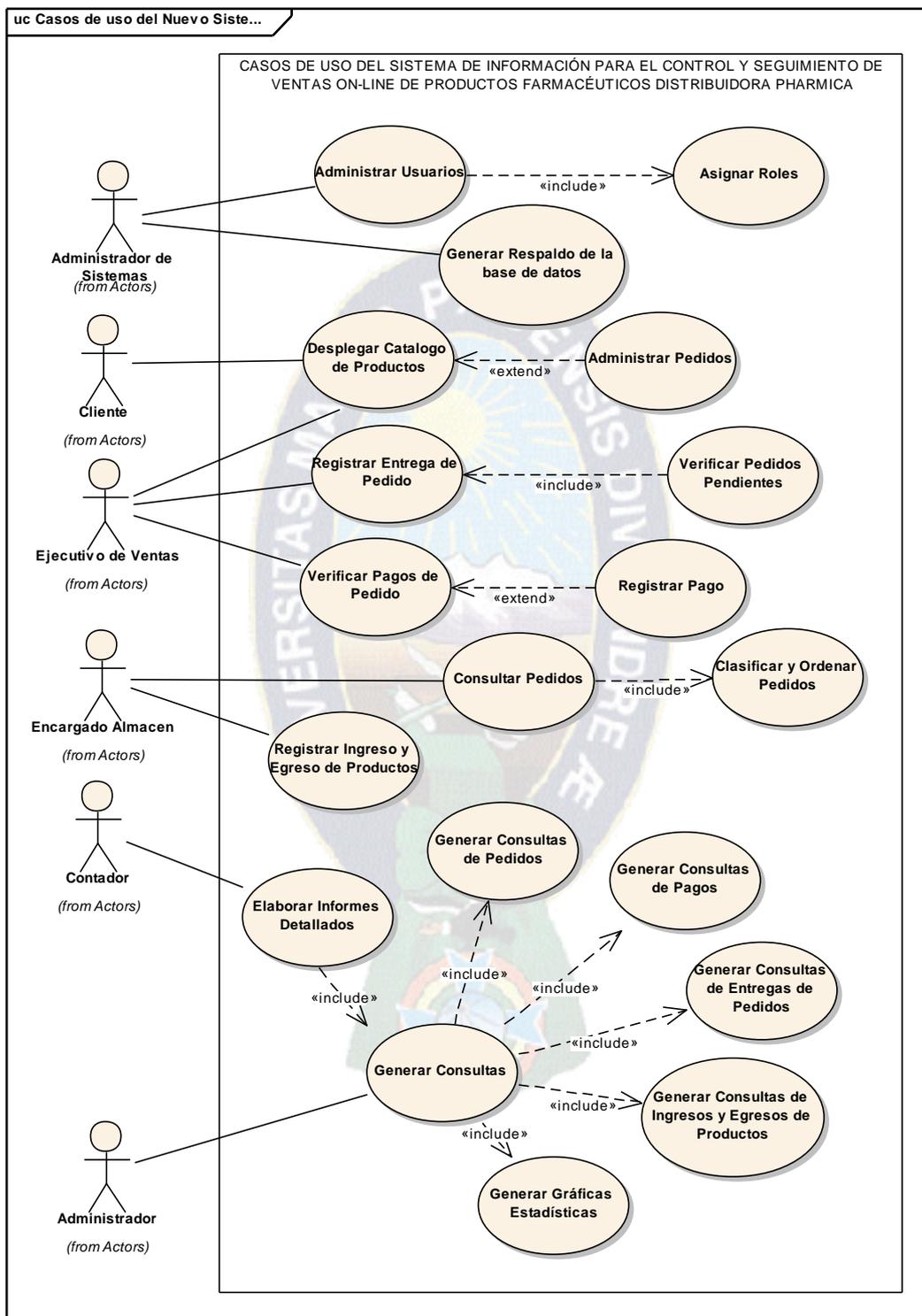


Figura 3.3: Casos de Uso del Sistema  
Fuente: Elaboración Propia

a) Diagrama de Caso de Uso Administrar Usuarios

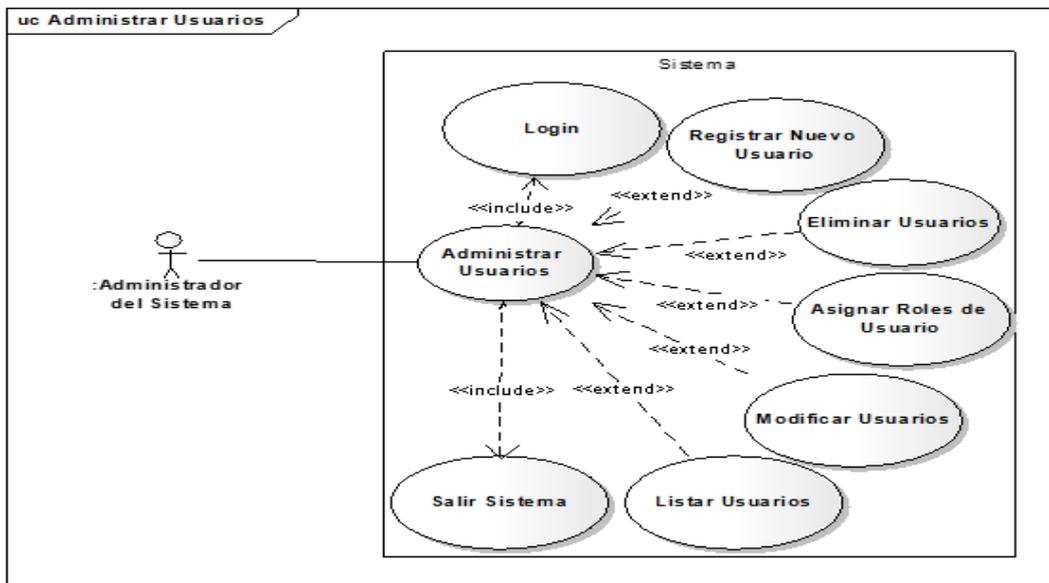


Figura 3.4: Casos de Uso Administrar Usuarios  
Fuente: Elaboración propia

b) Diagrama de Caso de Uso Administrar Productos

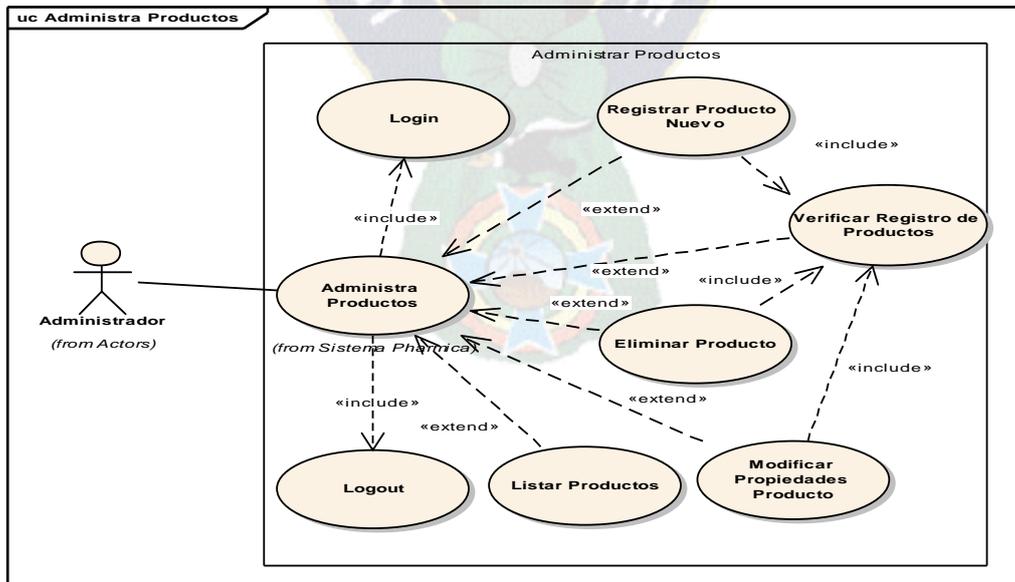


Figura 3.5: Casos de Uso Administrar Usuarios  
Fuente: Elaboración propia

### 3.4. Análisis y Diseño

#### 3.4.1. Especificación de Casos De Uso

A continuación se muestran las tablas referentes a la especificación del caso de uso, cada tabla muestra la descripción por cada caso de uso a fin de entender rápidamente el grado de complejidad y funcionalidad del sistema.

##### a) Caso de Uso: Login

**Tabla 3.7:** Especificación Caso de Uso: Login

<b>Nombre:</b> Login		
<b>Estado:</b> Análisis		<b>Tipo:</b> Primario
<b>Actores:</b> Administrador del Sistema, Administrador, Contador, Encargado Almacén, Cliente, Ejecutivo de Ventas.		
<b>Descripción:</b> Constata el acceso de un usuario al sistema. Otorga el acceso al menú de opciones del módulo correspondiente. Si el usuario que intenta acceder no está autorizado, se exhibe el correspondiente mensaje de error.		
<b>Precondición:</b> El usuario debe identificarse y autenticarse.		
<b>Post-condición:</b> El usuario tiene acceso a la información y opciones del módulo correspondiente.		
<b>Actor</b>		<b>Sistema</b>
1		Muestra una ventana para que se ingrese el nombre de usuario y la contraseña.
2	Ingresar nombre de usuario y contraseña.	
3		Comprueba los datos ingresados.
4		Almacena los datos y envía un mensaje de confirmación.
5	Visualiza la interfaz principal del módulo correspondiente.	
<b>Caminos Alternos</b>		
3.a		Los datos ingresados por el usuario son incorrectos.
5.a	Visualiza un mensaje de error "Datos incorrectos. Por favor corrija o complete los datos".	
<b>Frecuencia esperada:</b> Utilizado permanentemente		
<b>Importancia:</b> Es de vital importancia para el funcionamiento de la aplicación.		
<b>Comentarios:</b> Ninguno		
<b>Rendimiento Esperado:</b> Optimo		

*Fuente: Elaboración propia*

## b) Caso de Uso: Registrar Nuevo Usuario

**Tabla 3.8:** Especificación Caso de Uso: Nuevo Usuario

<b>Nombre:</b> Registrar Nuevo Usuario		
<b>Estado:</b> Análisis		<b>Tipo:</b> Primario
<b>Actores:</b> Administrador del Sistema		
<b>Descripción:</b> Integra nuevos usuarios en la base de datos del sistema, incorporando la información pertinente del nuevo usuario.		
<b>Precondición:</b> Administrador del Sistema debe identificarse y autenticarse.		
<b>Post condición:</b> El administrador incorpora un nuevo usuario del sistema.		
<b>Actor</b>		<b>Sistema</b>
1		Exhibe la ventana para el ingreso de usuarios.
2	Ingresa la información para la incorporación de un nuevo usuario.	
3		Comprueba los datos ingresados.
4		Almacena los datos y envía un mensaje de confirmación.
5	Visualiza mensaje de confirmación de nuevo usuario creado.	
<b>Caminos Alternos</b>		
3		Los datos ingresados por el usuario son incorrectos.
5	Visualiza un mensaje de error "Datos incorrectos. Por favor corrija o complete los datos".	
<b>Frecuencia esperada:</b> Utilizado permanentemente		
<b>Importancia:</b> Es de vital importancia para el funcionamiento de la aplicación.		
<b>Comentarios:</b> Ninguno		
<b>Rendimiento Esperado:</b> Optimo		

*Fuente: Elaboración propia*

## c) Caso de Uso: Modificar Usuarios

**Tabla 3.9:** Especificación Caso de Uso: Modificar Usuarios

<b>Nombre:</b> Modificar Usuarios		
<b>Estado:</b> Análisis		<b>Tipo:</b> Primario
<b>Actores:</b> Administrador del Sistema		
<b>Descripción:</b> Modifica la información de un usuario vinculado a la base de datos del sistema.		
<b>Precondición:</b> Administrador del Sistema debe identificarse y autenticarse.		
<b>Post condición:</b> El administrador modifica los datos de un usuario del sistema.		
<b>Actor</b>		<b>Sistema</b>
1		Muestra la ventana para la modificación de la información de los usuarios.
2	Elige el usuario que desea modificar y efectúa los cambios.	
3		Comprueba los datos ingresados.
4		Envía un mensaje de confirmación y guarda los cambios realizados.

5	Visualiza mensaje de confirmación de usuario modificado.	
<b>Caminos Alternos</b>		
3.a		Error. El usuario no ha seleccionado ningún usuario a modificar.
5.a	Visualiza mensaje de error “No se ha seleccionado ningún registro”	
3.b		Los datos están incompletos o incorrectos.
5.b	Visualiza un mensaje de error “Datos incorrectos. Por favor corrija o complete los datos”.	
<b>Frecuencia esperada:</b> Utilizado permanentemente		
<b>Importancia:</b> Es de vital importancia para el funcionamiento de la aplicación.		
<b>Comentarios:</b> Ninguno		
<b>Rendimiento Esperado:</b> Optimo		

*Fuente: Elaboración propia*

d) Caso de Uso: Listar Usuario

**Tabla 3.9:** Especificación Caso de Uso: Listar Usuario

<b>Nombre:</b> Listar Usuario		
<b>Estado:</b> Análisis		<b>Tipo:</b> Primario
<b>Actores:</b> Administrador del Sistema		
<b>Descripción:</b> Crea una lista detallada de los usuarios que contiene la base de datos del sistema.		
<b>Precondición:</b> El Administrador del Sistema debe identificarse y autenticarse.		
<b>Post condición:</b> El Administrador del Sistema no podrá efectuar ningún cambio.		
<b>Actor</b>		<b>Sistema</b>
1		Muestra la interfaz con toda la información acerca de los usuarios del sistema.
<b>Frecuencia esperada:</b> Utilizado Ocasionalmente		
<b>Importancia:</b> Es de vital importancia, para el funcionamiento de la aplicación.		
<b>Comentarios:</b> Ninguno		
<b>Rendimiento Esperado:</b> Optimo		

*Fuente: Elaboración propia*

e) Caso de Uso Logout

**Tabla 3.10:** Especificación Caso de Uso: Logout

<b>Nombre:</b> Logout		
<b>Estado:</b> Análisis		<b>Tipo:</b> Primario
<b>Actores:</b> Administrador del Sistema, Administrador, Contador, Encargado Almacén, Cliente y Ejecutivo de Ventas.		
<b>Descripción:</b> Permite al Administrador del Sistema y demás usuarios salir del módulo correspondiente.		
<b>Precondición:</b> El usuario debe identificarse y autenticarse.		
<b>Post condición:</b> El usuario sale del módulo.		
<b>Actor</b>		<b>Sistema</b>
1	Elige la opción “Salir del sistema”.	
2		Finaliza la sesión del Usuario

<b>Frecuencia esperada:</b> Utilizado Ocasionalmente
<b>Importancia:</b> Es de vital importancia, para el funcionamiento de la aplicación.
<b>Comentarios:</b> Ninguno
<b>Rendimiento Esperado:</b> Optimo

*Fuente: Elaboración propia*

f) Caso de Uso Registrar Producto Nuevo

**Tabla 3.11:** Especificación Caso de Uso: Registrar Producto Nuevo

<b>Nombre:</b> Registrar Producto Nuevo	
<b>Estado:</b> Análisis	<b>Tipo:</b> Primario
<b>Actores:</b> Administrador.	
<b>Descripción:</b> Permite al administrador ingresar productos nuevo y definir sus propiedades.	
<b>Precondición:</b> El Administrador debe identificarse y autenticarse.	
<b>Post condición:</b> El Administrador tiene habilitadas las diferentes opciones para registrar un nuevo producto.	
<b>Actor</b>	<b>Sistema</b>
1	Muestra la interfaz correspondiente al registro de un nuevo producto
2	Introduce los valores de las propiedades del nuevo producto.
3	Presiona el botón Guardar
4	Verifica que la información introducida
5	Muestra la información del nuevo producto registrado.
<b>Caminos Alternos</b>	
5.a	Muestra Mensaje: "El Producto que quiere registrar no cuenta con información completa"
<b>Frecuencia esperada:</b> Utilizado Ocasionalmente	
<b>Importancia:</b> Es de vital importancia, para el funcionamiento de la aplicación.	
<b>Comentarios:</b> Ninguno	
<b>Rendimiento Esperado:</b> Optimo	

*Fuente: Elaboración propia*

g) Caso de Uso Clasificar y Ordenar Pedidos

**Tabla 3.12:** Especificación Caso de Uso: Clasificar y Ordenar Pedidos

<b>Nombre:</b> Clasificar y Ordenar Pedidos	
<b>Estado:</b> Análisis	<b>Tipo:</b> Primario
<b>Actores:</b> Encargado Almacén	
<b>Descripción:</b> Permite definir las variables de clasificación y orden de los pedidos en su conjunto.	
<b>Precondición:</b> El Encargado Almacén debe identificarse y autenticarse.	
<b>Post condición:</b> El Encargado Almacén tiene habilitadas las diferentes opciones para clasificar y ordenar el conjunto de pedidos.	
<b>Actor</b>	<b>Sistema</b>
1	Muestra la interfaz correspondiente la lista de los pedidos registrados existentes

2	Introduce los valores de los criterios o propiedades con las que se clasificara u ordenara.	
3		Muestra la información clasificada y ordenada de acuerdo a los criterios definidos
<b>Frecuencia esperada:</b> Utilizado Frecuentemente		
<b>Importancia:</b> Es de vital importancia, para el funcionamiento de la aplicación.		
<b>Comentarios:</b> Ninguno		
<b>Rendimiento Esperado:</b> Optimo		

*Fuente: Elaboración propia*

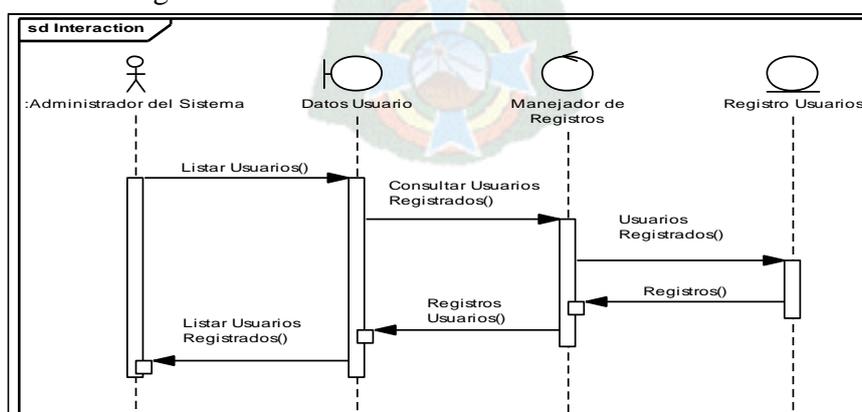
### 3.5. Fase de Construcción

En la fase de construcción se tiene más énfasis en el diseño y la implementación que es el flujo fundamental en donde modelamos el sistema y encontramos su forma para que soporte todos los requerimientos, y desarrollo del proyecto tomando como una abstracción para la implementación del sistema.

#### 3.5.1. Diagramas de Secuencia para Diseño

El diagrama de secuencia le muestra la forma en que se comunican los objetos al transcurrir el tiempo en el orden de las llamadas/eventos del sistema

##### a. Diagrama de Secuencia: Listar Usuarios



*Figura 3.6: Diagrama de Secuencia Listar Usuarios*

*Fuente: Elaboración propia*

b. Diagrama de Secuencia: Login

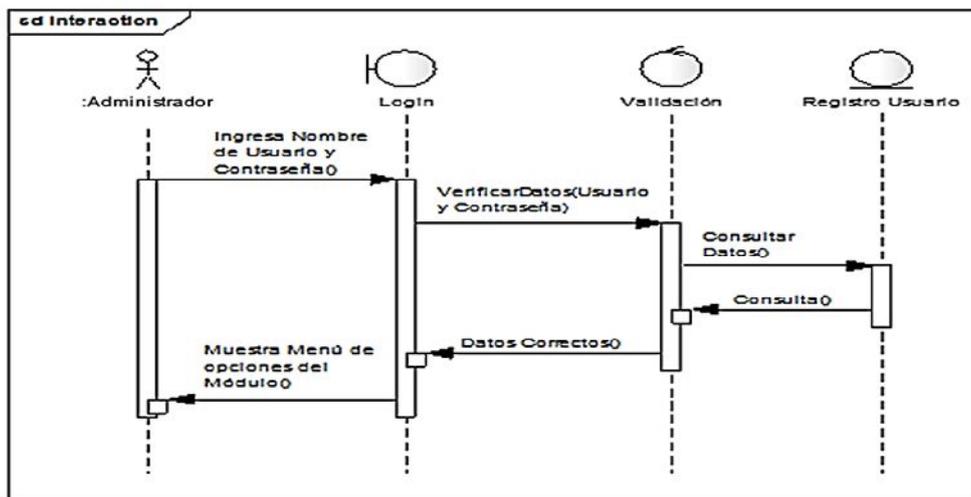


Figura 3.7: Diagrama de Secuencia Login  
 Fuente: Elaboración propia

c. Diagrama de Secuencia: Crear Usuario

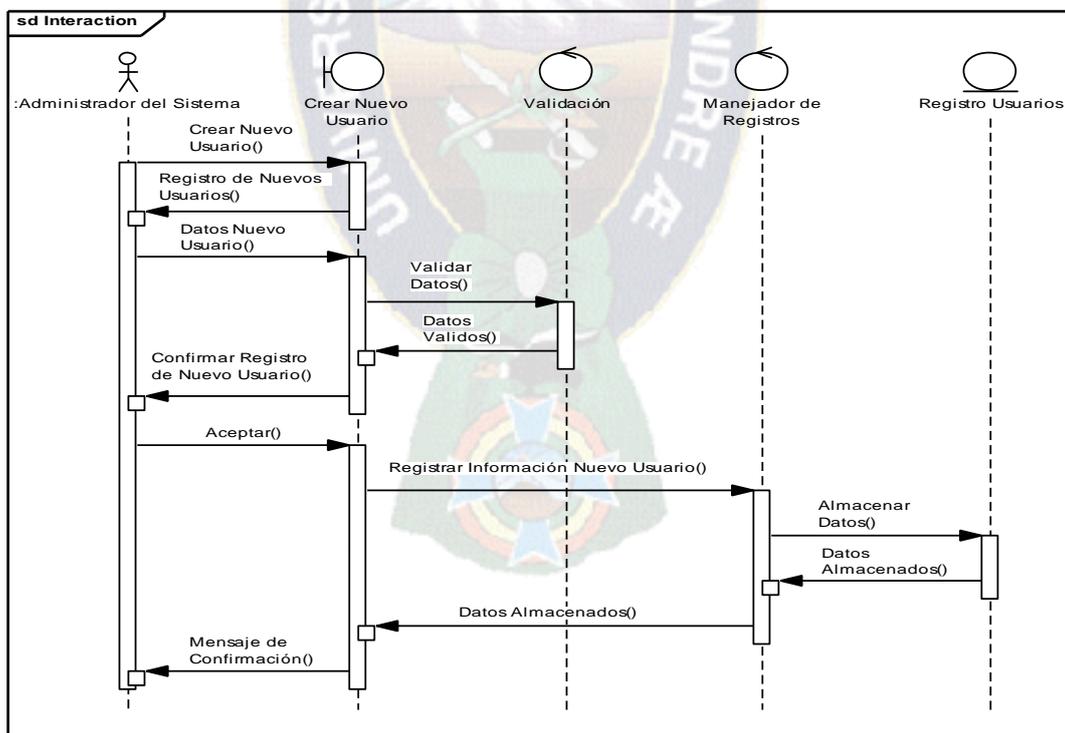


Figura 3.8: Diagrama de Secuencia Crear Usuario  
 Fuente: Elaboración propia

d. Diagrama de Secuencia: Logout

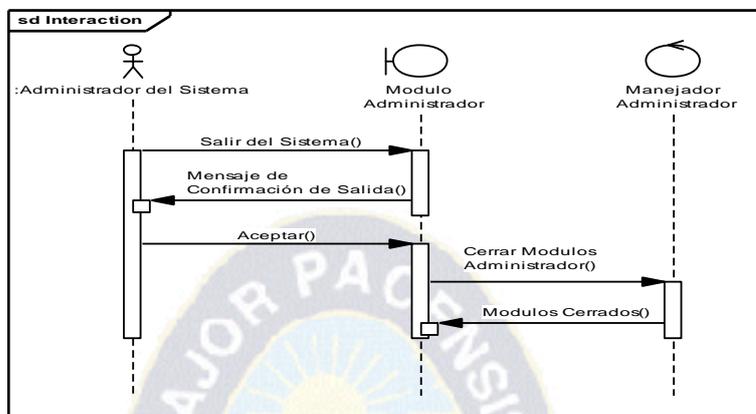


Figura 3.9: Diagrama de Secuencia Logout

Fuente: Elaboración propia

e. Diagrama de Secuencia: Modificar Usuario

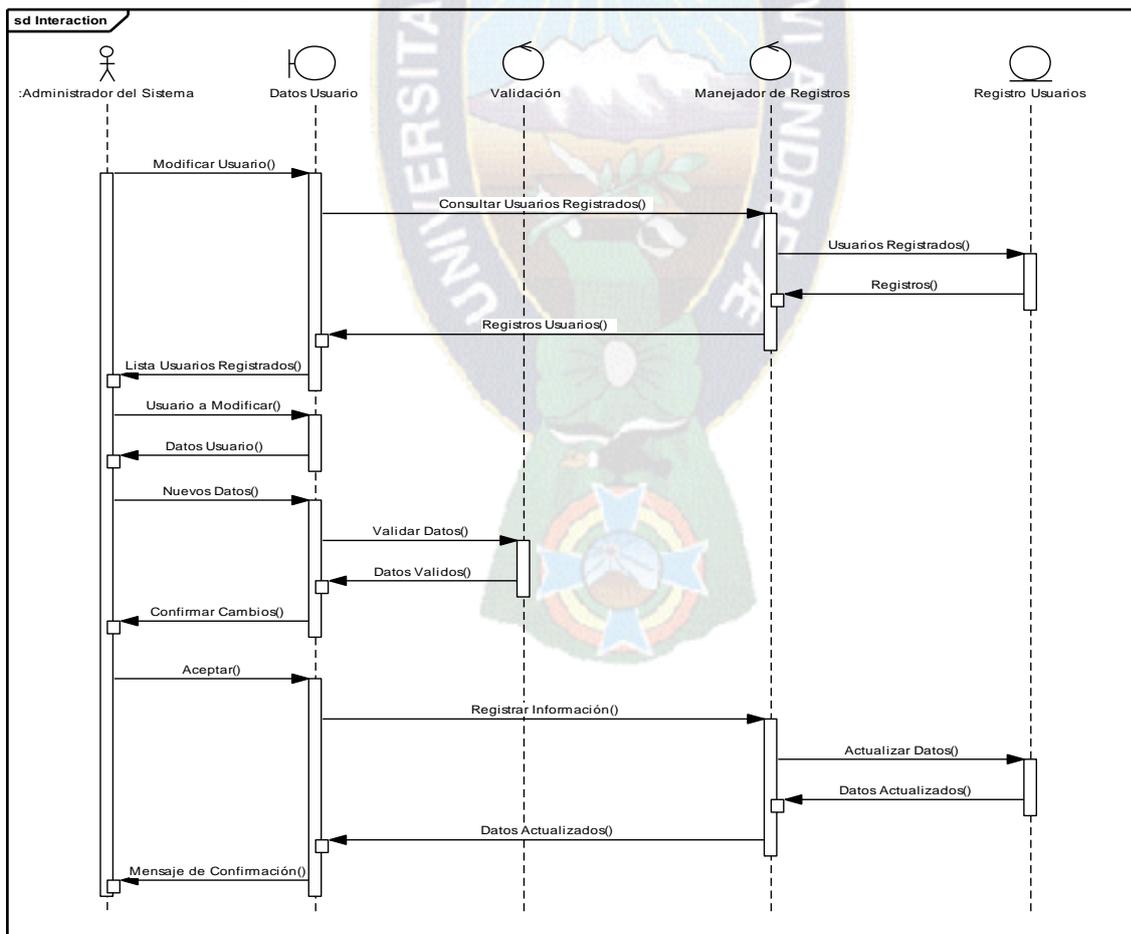


Figura 3.10: Diagrama de Secuencia Modificar Usuario

Fuente: Elaboración propia

f. Diagrama de Secuencia: Listar Productos

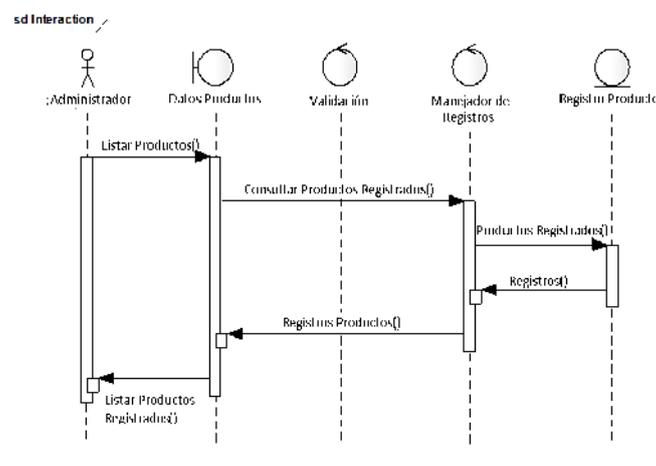


Figura 3.11: Diagrama de Secuencia Listar Productos  
Fuente: Elaboración propia

3.5.2. Diagramas de Colaboración

Los diagramas de colaboración muestra la forma en que los objetos colaboran entre sí, mostrando los mensajes que envían entre ellos, destacando el contexto y organización en general de los objetos que interactúan. Es la asignación de responsabilidades entre los objetos y mostrar su interacción. A continuación se muestra los diagramas de colaboración respecto al sistema.

a. Diagrama de Colaboración: Listar Usuarios

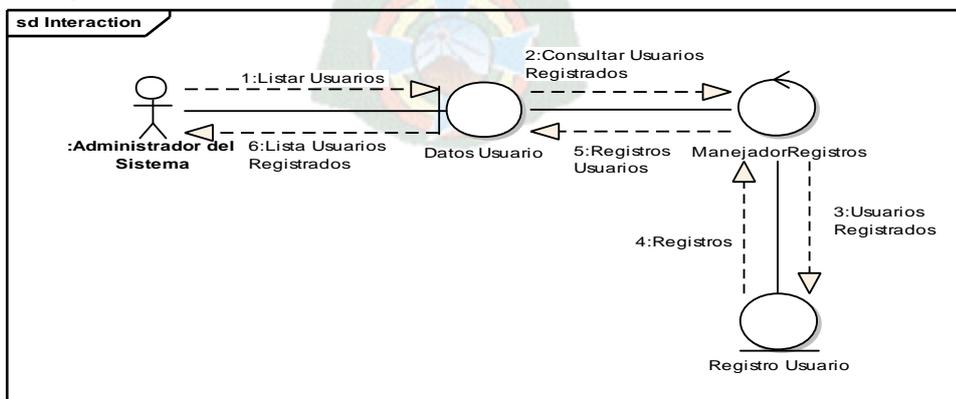


Figura 3.12: Diagrama de Colaboración Listar Usuarios  
Fuente: Elaboración propia

b. Diagrama de Colaboración: Login

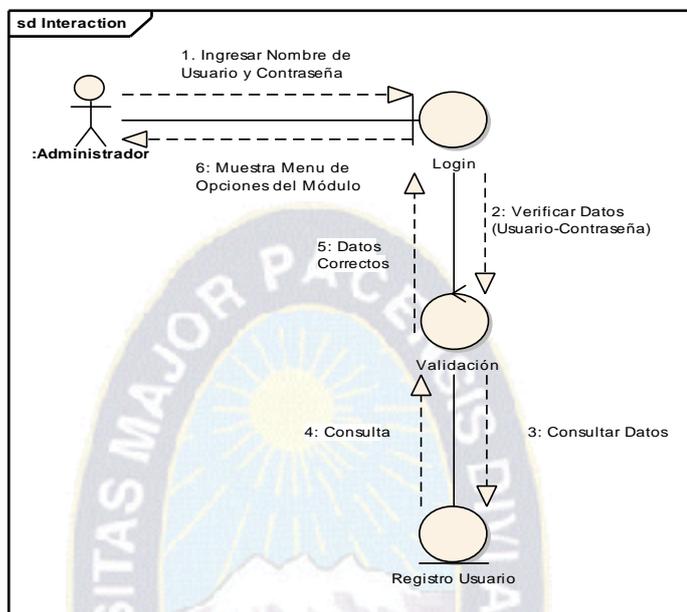


Figura 3.13: Diagrama de Colaboración Login

Fuente: Elaboración propia

c. Diagrama de Colaboración: Modificar Usuario

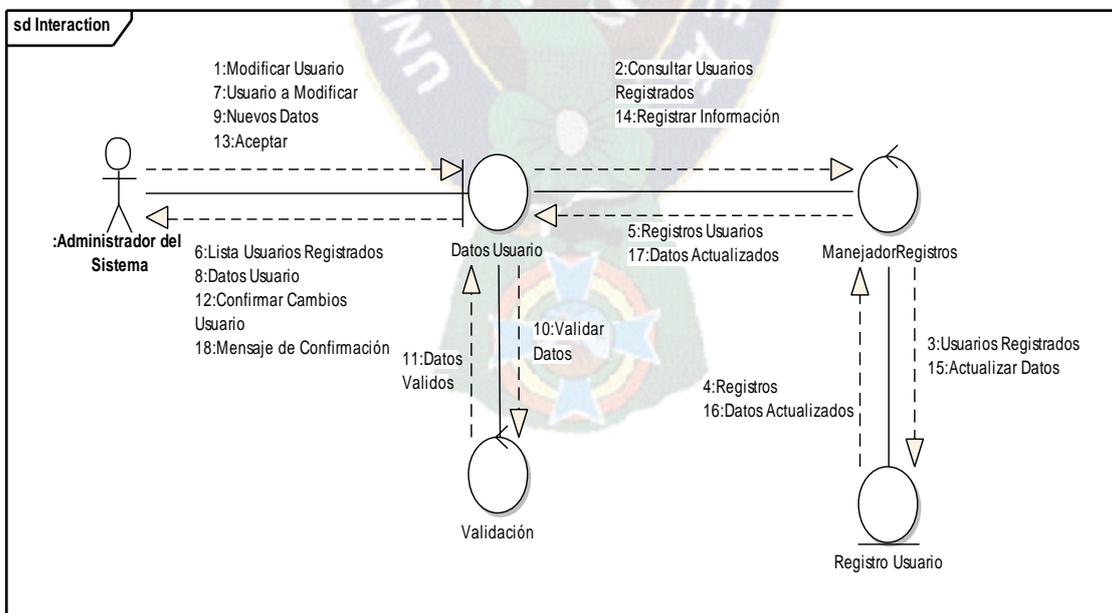


Figura 3.14: Diagrama de Colaboración Modificar Usuario

Fuente: Elaboración propia

d. Diagrama de Colaboración: Crear Usuario

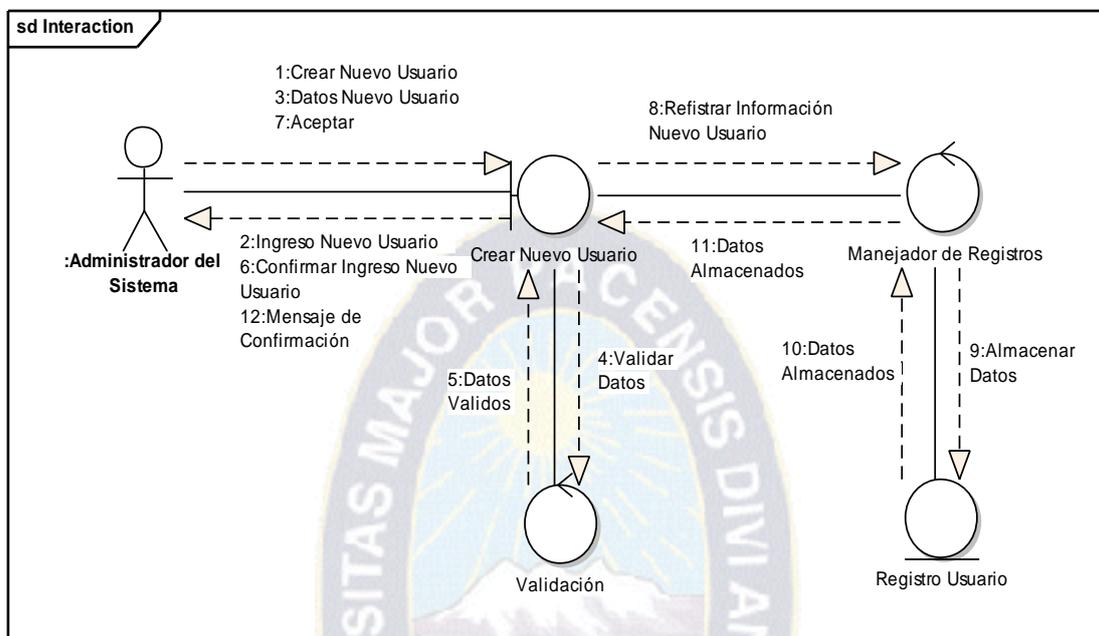


Figura 3.15: Diagrama de Colaboración Crear Usuario  
Fuente: Elaboración propia

e. Diagrama de Colaboración: Logout

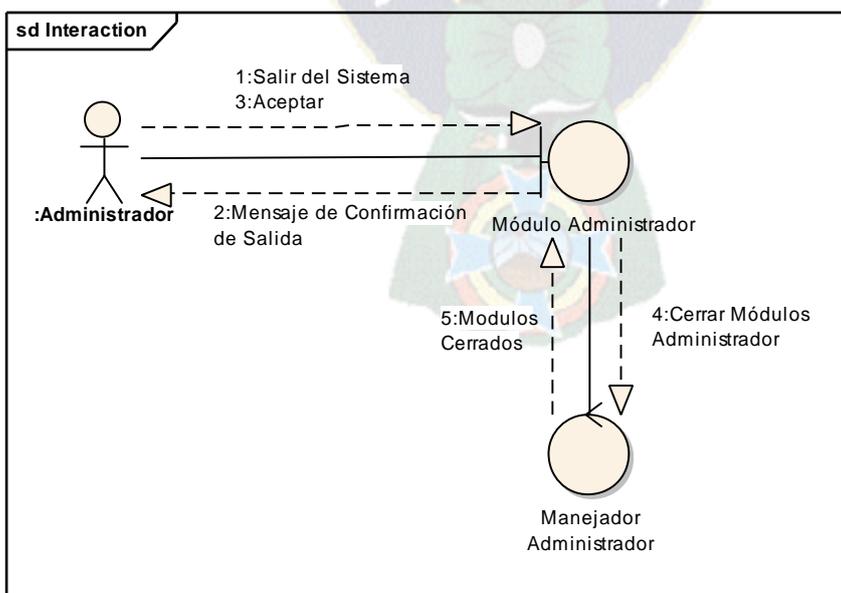


Figura 3.16: Diagrama de Colaboración Logout  
Fuente: Elaboración propia

## f. Diagrama de Colaboración: Listar Productos

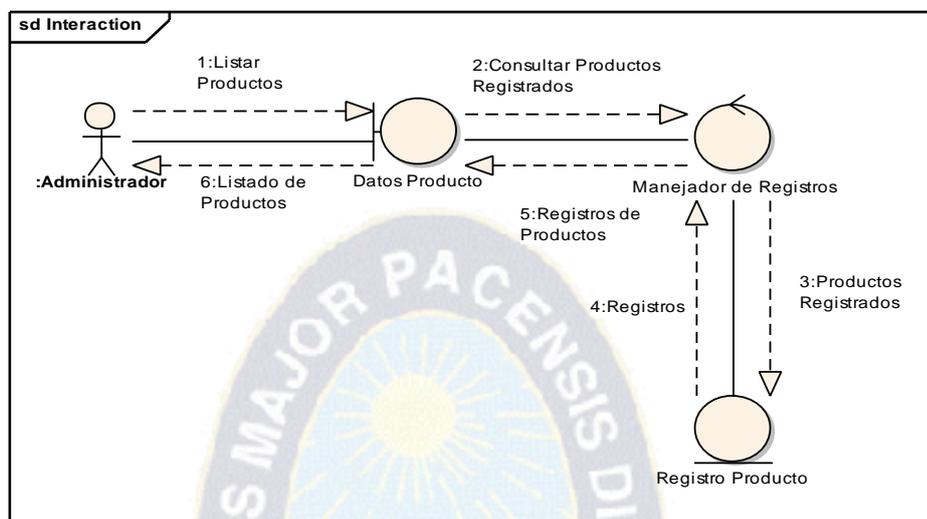


Figura 3.17: Diagrama de Colaboración Listar Productos

Fuente: Elaboración propia

## 3.5.3. Diagramas de Estados

Describe visualmente los estados y eventos más interesados de un objeto, así como su comportamiento ante un evento. Un diagrama de estado presenta el ciclo de vida de un objeto: los eventos que le ocurren, sus transiciones y los estados que media n entre sus eventos.

Los diagramas de estados correspondientes a los casos de uso son los siguientes:

## a. Diagrama de Estados: Login

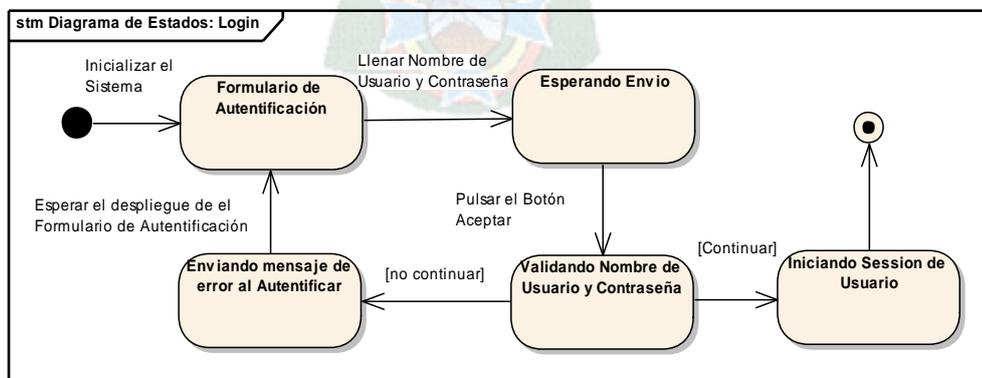


Figura 3.18: Diagrama de Estados Login

Fuente: Elaboración propia

b. Diagrama de Estados: Registrar Nuevo Usuario

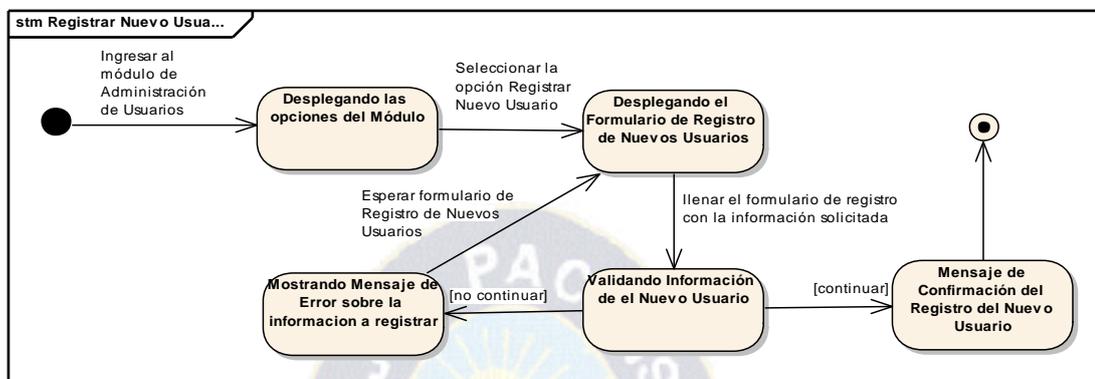


Figura 3.19: Diagrama de Estados Registrar Nuevo Usuario

Fuente: Elaboración propia

c. Diagrama de Estados: Generar Informe Diario de Ingresos y Egresos de Productos

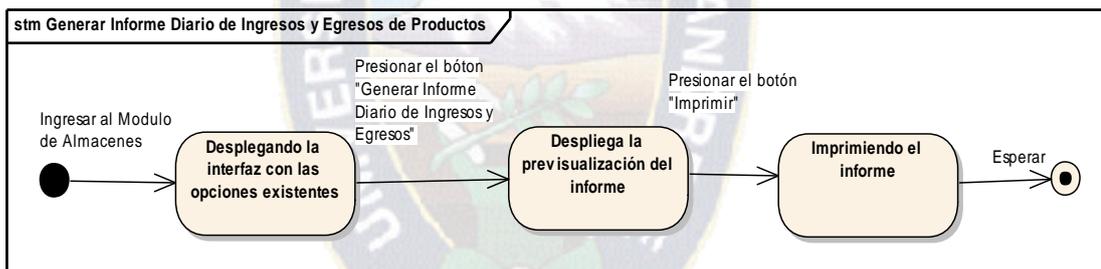


Figura 3.20: Diagrama de Estados Generar Informe Diario de Ingresos y Egresos de Productos

Fuente: Elaboración propia

d. Diagrama de Estados: Generar Nota de Venta

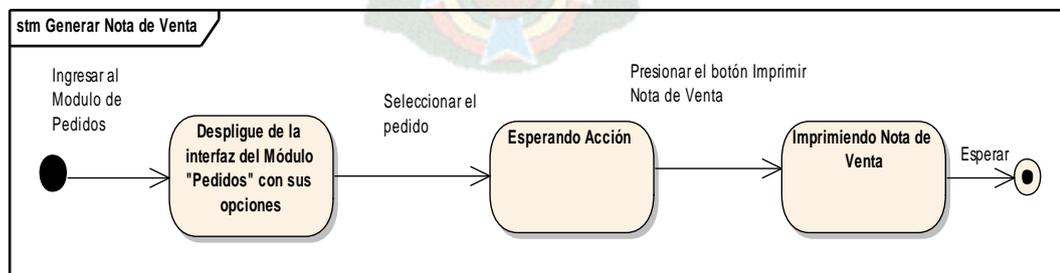


Figura 3.21: Diagrama de Estados Generar Nota de Venta

Fuente: Elaboración propia

### 3.5.4. Diagramas de Actividades

Los diagramas de actividades han sido diseñados para mostrar una visión simplificada de lo que ocurre durante una operación o proceso, son una extensión de los diagramas de estados.

#### a) Diagrama de Actividad: Registrar Nuevo Usuario

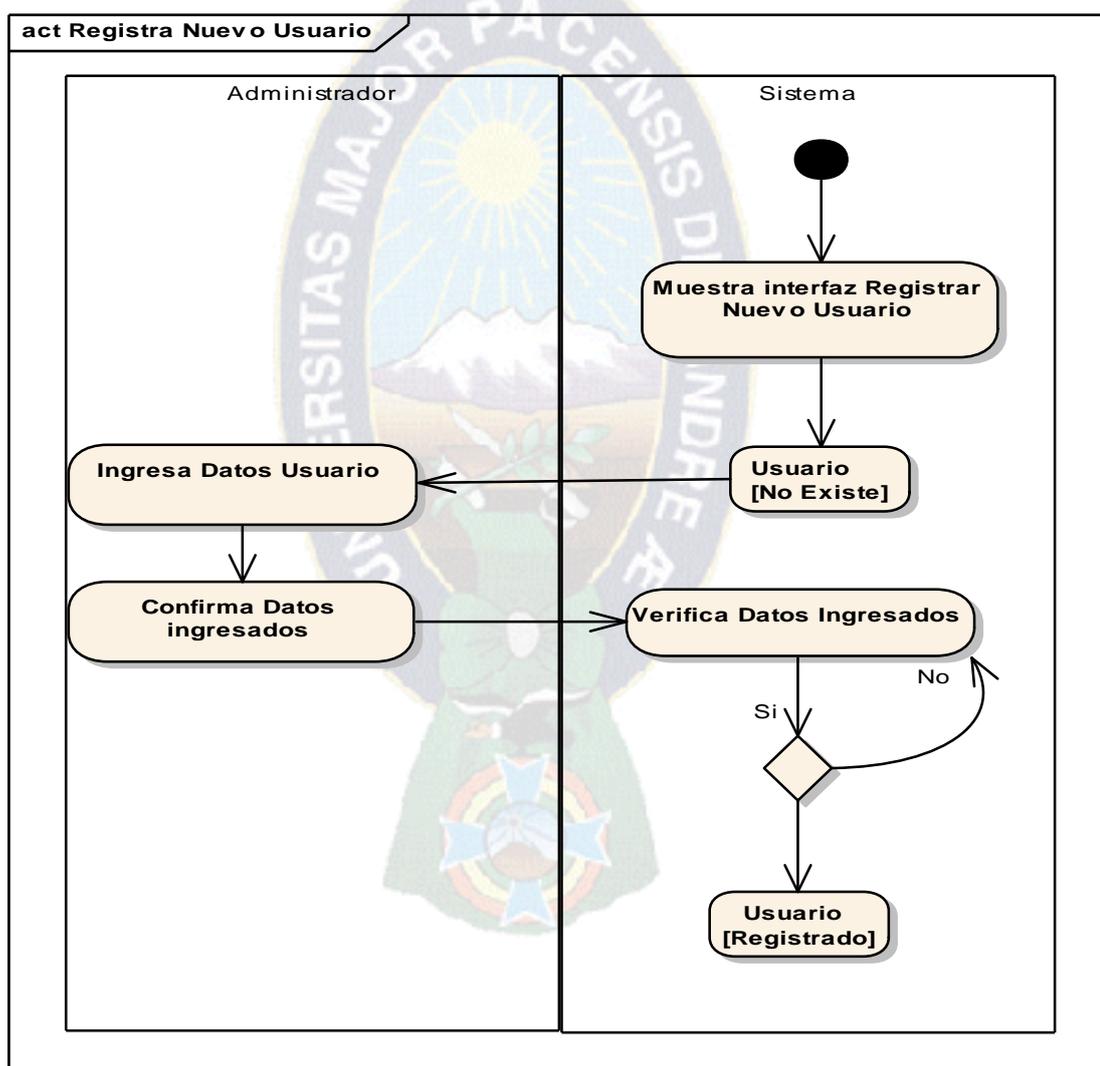


Figura 3.22: Diagrama de Actividad Registrar Nuevo Usuario

Fuente: Elaboración propia

## a) Diagrama de Actividad: Eliminar Usuario

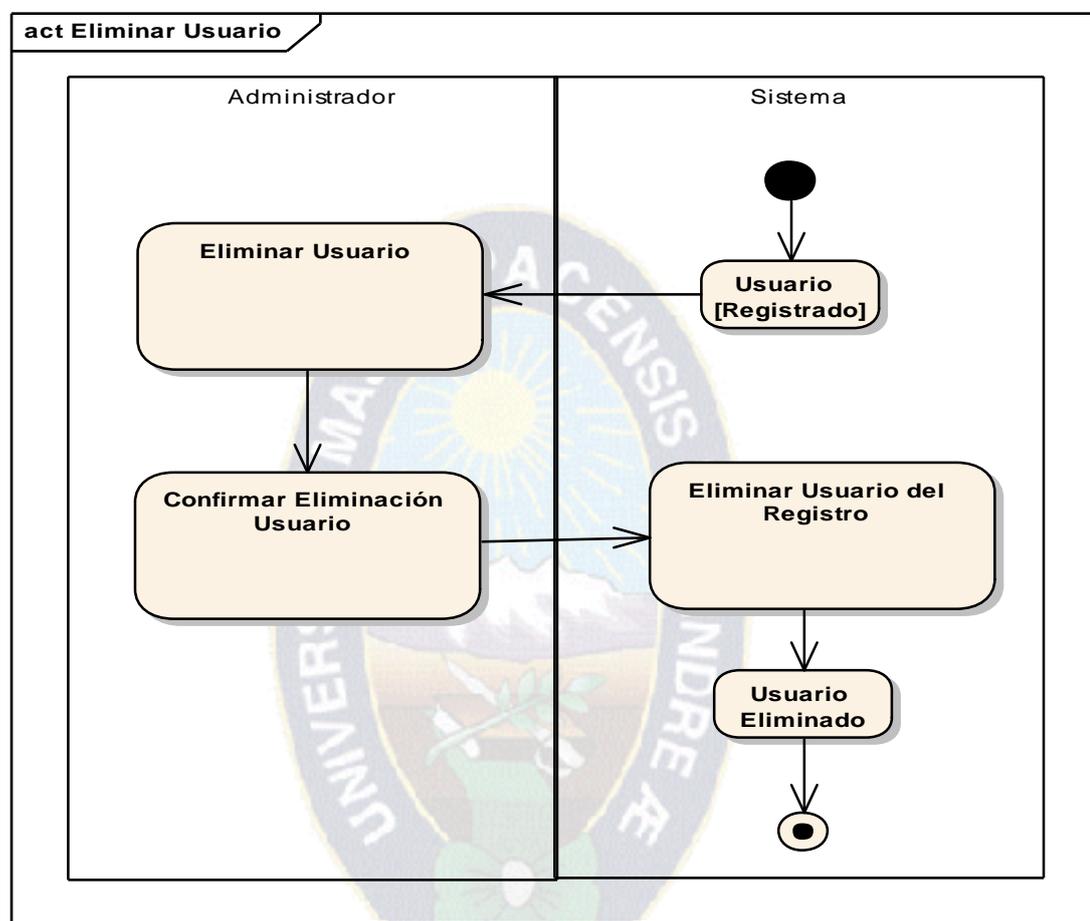


Figura 3.23: Diagrama de Actividad Eliminar Usuario

Fuente: Elaboración propia

### 3.5.5. Diagramas de Clases

Este diagrama describe gráficamente la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Se define a una clase como categoría o grupo de cosas que tiene atributos o acciones similares. A continuación se muestra el diagrama de clases del Sistema.

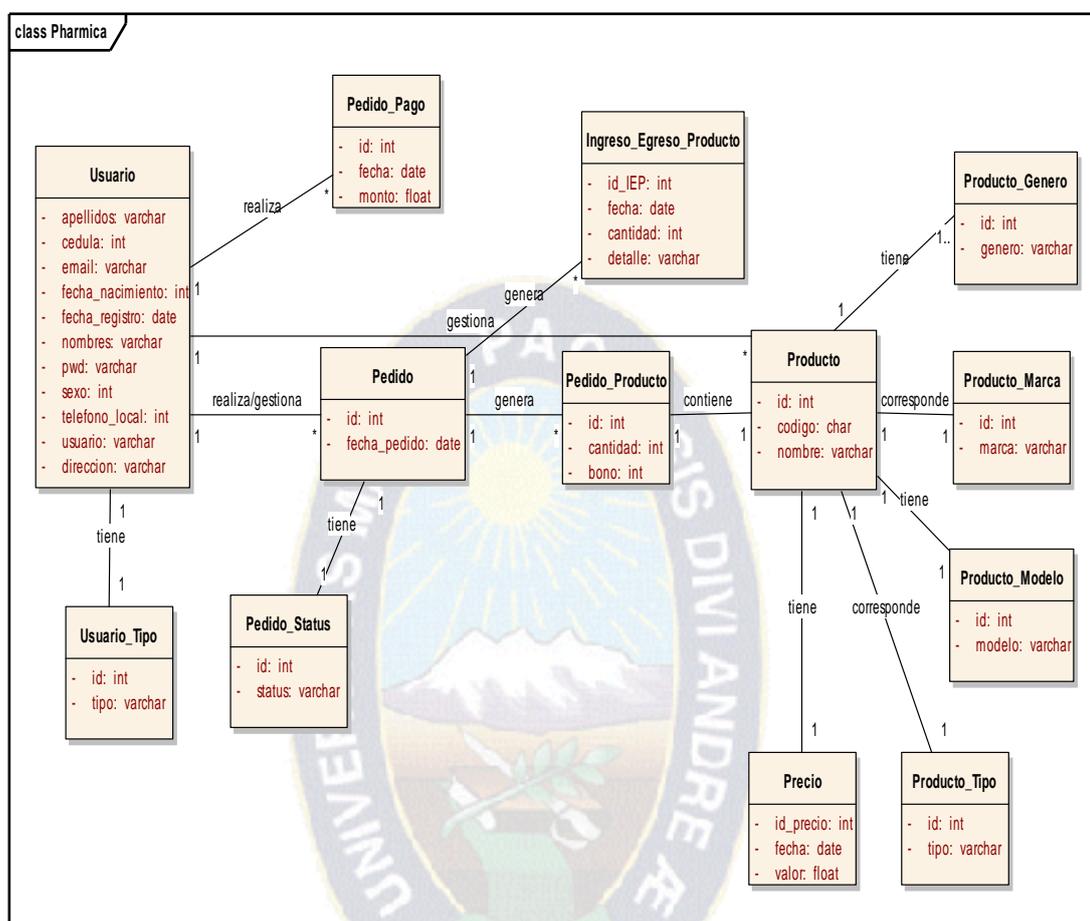


Figura 3.24: Diagrama de Clases

Fuente: Elaboración propia

### 3.5.6. Definición Del Modelo Físico de la base de Datos

A partir de las especificaciones que se tienen para transformar un diagrama de clases a un modelo físico en la base de datos.

A continuación se presenta el modelo que representa de manera física la base de datos del Sistema el cual se va a crear en el gestor de la base de datos Mysql.

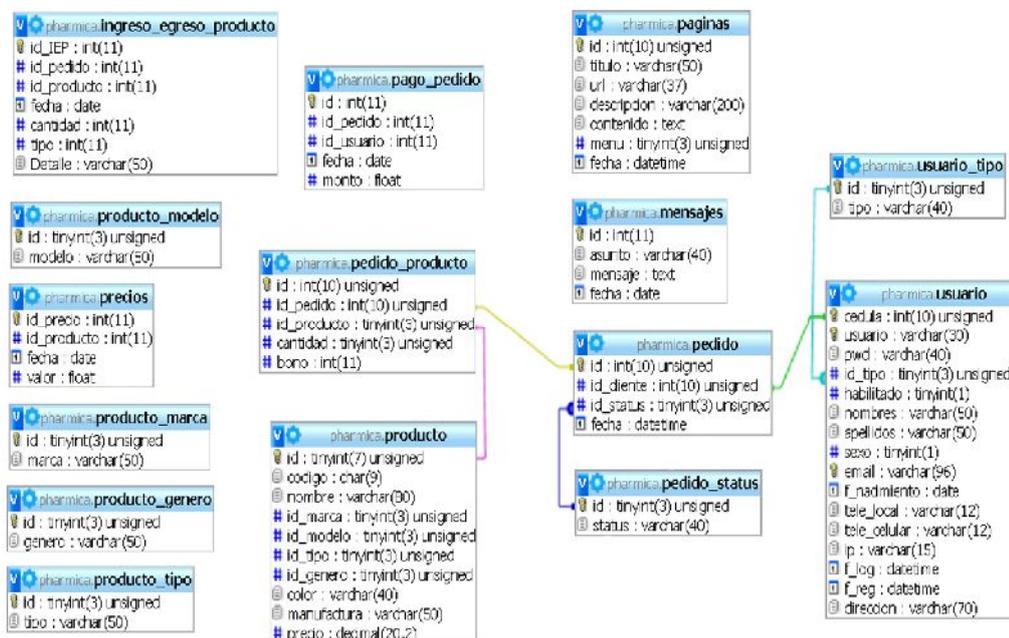


Figura 3.25: Modelo Físico de la Base de Datos

Fuente: Elaboración propia

### 3.5.7. Diccionario de Datos

A continuación se realiza un el detalle de los campos o columnas de las tablas existentes en la Base de Datos PHARMICA.

#### Estructura de tabla para la tabla pedido

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(10)	No	
id_cliente	int(10)	No	
id_status	tinyint(3)	No	1
fecha	datetime	No	

**Estructura de tabla para la tabla pedido\_producto**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	int(10)	No	
id_pedido	int(10)	No	
id_producto	tinyint(3)	No	
cantidad	tinyint(3)	No	
bono	int(11)	No	

**Estructura de tabla para la tabla pedido\_status**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(3)	No	
status	varchar(40)	No	

**Estructura de tabla para la tabla precios**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id_precio</i>	int(11)	No	
id_producto	int(11)	No	
fecha	date	No	
valor	float	No	

**Estructura de tabla para la tabla producto**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(7)	No	
codigo	char(9)	No	
nombre	varchar(80)	No	
id_marca	tinyint(3)	No	
id_modelo	tinyint(3)	No	
id_tipo	tinyint(3)	No	
id_genero	tinyint(3)	No	0
color	varchar(40)	No	
manufactura	varchar(50)	No	
precio	decimal(20,2)	No	0.00

**Estructura de tabla para la tabla producto**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(7)	No	
codigo	char(9)	No	
nombre	varchar(80)	No	
id_marca	tinyint(3)	No	
id_modelo	tinyint(3)	No	
id_tipo	tinyint(3)	No	
id_genero	tinyint(3)	No	0
color	varchar(40)	No	
manufactura	varchar(50)	No	
precio	decimal(20,2)	No	0.00

**Estructura de tabla para la tabla producto\_genero**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(3)	No	
genero	varchar(50)	No	

**Estructura de tabla para la tabla producto\_marca**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(3)	No	
marca	varchar(50)	No	

**Estructura de tabla para la tabla producto\_modelo**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(3)	No	
modelo	varchar(50)	No	

**Estructura de tabla para la tabla producto\_tipo**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>
<i>id</i>	tinyint(3)	No	

tipo	varchar(50)	No
------	-------------	----

#### Estructura de tabla para la tabla ingreso\_egreso\_producto

Columna	Tipo	Nulo	Predeterminado
<i>id_IEP</i>	int(11)	No	
id_pedido	int(11)	No	
id_producto	int(11)	No	
fecha	Date	No	
cantidad	int(11)	No	
tipo	int(11)	No	
Detalle	varchar(50)	No	
gfdsa	int(11)	No	

#### Estructura de tabla para la tabla mensajes

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
asunto	varchar(40)	No	
mensaje	Text	No	
fecha	Date	No	

#### Estructura de tabla para la tabla paginas

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(10)	No	
titulo	varchar(50)	No	
url	varchar(37)	No	
descripcion	varchar(200)	No	
contenido	Text	No	
menu	tinyint(3)	No	0
fecha	Datetime	No	

#### Estructura de tabla para la tabla pago\_pedido

Columna	Tipo	Nulo	Predeterminado
---------	------	------	----------------

<b>id</b>	int(11)	No
id_pedido	int(11)	No
id_usuario	int(11)	No
fecha	Date	No
monto	Float	No

### Estructura de tabla para la tabla usuario

Columna	Tipo	Nulo	Predeterminado
<b>Cedula</b>	int(10)	No	
<b>Usuario</b>	varchar(30)	No	
Pwd	varchar(40)	Sí	NULL
id_tipo	tinyint(3)	No	2
Habilitado	tinyint(1)	No	1
Nombres	varchar(50)	No	
Apellidos	varchar(50)	No	
Sexo	tinyint(1)	No	
<b>Email</b>	varchar(96)	No	
f_nacimiento	Date	Sí	NULL
tele_local	varchar(12)	No	
tele_celular	varchar(12)	No	
Ip	varchar(15)	No	
f_log	Datetime	No	
f_reg	Datetime	No	
Direccion	varchar(70)	Sí	NULL

### Estructura de tabla para la tabla usuario\_tipo

Columna	Tipo	Nulo	Predeterminado
<b>id</b>	tinyint(3)	No	
tipo	varchar(40)	No	

### 3.6. Fase de Transición

Dentro de esta fase desplegarán los entornos a los que accederán los usuarios para la manipulación del sistema.

#### 3.6.1. Diseño de Interfaz

A continuación se muestran las principales interfaces del sistema:

- Interfaz de inicio al “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“



Figura 3.26: Interfaz de inicio  
Fuente: Elaboración propia

- Interfaz de Autenticación al “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“



Figura 3.27: Interfaz de autenticación  
Fuente: Elaboración propia

- Interfaz de Administrador “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“

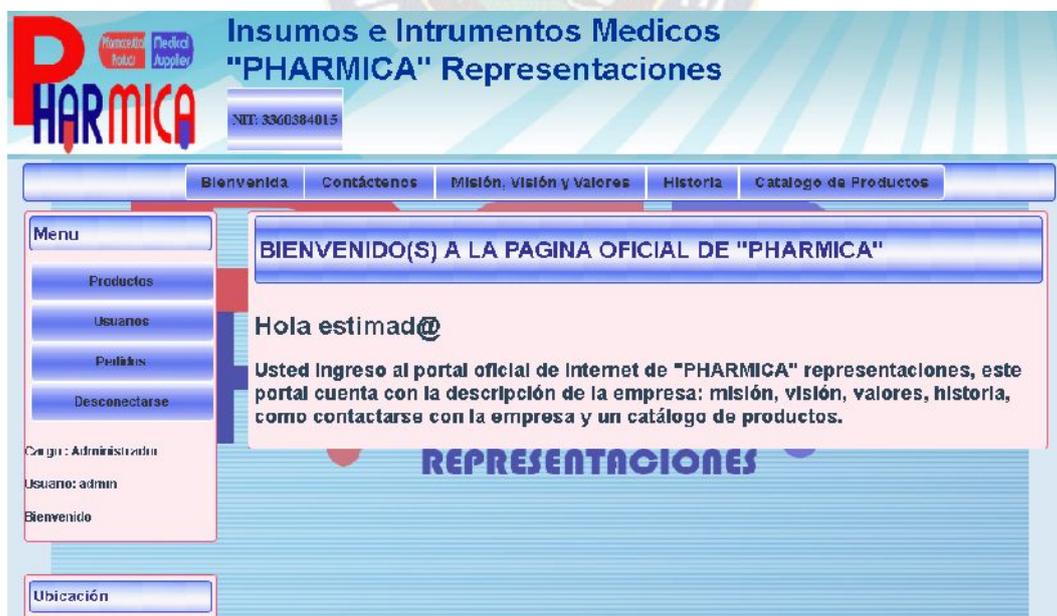


Figura 3.28: Interfaz del Administrador  
Fuente: Elaboración propia

- Interfaz de gestión de Productos “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“

Código	Producto	Precio	Marca	Modelo	Tipo	Genero	Color
DA001	DAQUJOS CALASTICO X 50 PC	10.00	SM	SM	INSUMOS, INSTR	INSUMOS	BLANCO
BI001	BIBERÓN 150 ml	23.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	CELESTE
LI002	LIULIÓN 150ml	20.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	ROSA
BI005	BIBERÓN 150ml ANTICÓLICO	25.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	AMARILLO
BI006	BIBERÓN 150ml ANTICÓLICO	25.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	VERDE
BI007	BIBERÓN 200ml ANTICÓLICO	27.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	AMARILLO
BI003	BIBERÓN 280ml	25.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	CELESTE
HI014	HIBERÓN 280ml	25.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	ROSA
BI008	BIBERÓN 280ml ANTICÓLICO	27.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	VERDE
HI011	HIBERÓN HOCA ANCHA 300ml	28.00	POTATO	SM	INSUMOS, INSTR	UTILNSILIOS	BLANCO
BM001	BISTURI CON MANGO X 10 UNIDADES	30.00	SM	SM	INSUMOS, INSTR	UTILNSILIOS	CROMADO
DO005	DOTIGUIN DELSIZALL X	70.00	SM	SM	INSUMOS, INSTR	UTILNSILIOS	BLANCO
BO003	BOTIQUIN GRANDE DE PRIMEROS AUXILIOS DE PLAS	50.00	SM	SM	INSUMOS, INSTR	UTILNSILIOS	BLANCO
DO002	DOTIGUIN MEDIANO DE PRIMEROS AUXILIOS DE PLAS	30.00	SM	SM	INSUMOS, INSTR	UTILNSILIOS	BLANCO
BO001	BOTIQUIN MINI DE PRIMEROS AUXILIOS DE PLAS	25.00	SM	SM	INSUMOS, INSTR	UTILNSILIOS	BLANCO

Figura 3.29: Interfaz de gestión de Productos

Fuente: Elaboración propia

- Interfaz agregar usuario “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“

Figura 3.30: Interfaz adicionar usuario

Fuente: Elaboración propia

- Interfaz agregar pedido “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“

**PHARMICA** Insumos e Instrumentos Medicos  
"PHARMICA" Representaciones  
NIT: 3360384015

Bienvenida | Contáctenos | Misión, Visión y Valores | Historia | **Catálogo de Productos**

[Volver...](#)

Responsable de Ventas: admin  
Codigo RV: 20674317  
NIT Cliente:

COD	Descripcion	Cantidad	Precio	Opc	Bono	SubT
BI005	BIBERÓN 150ml ANTICÓLICO AMARILLO	<input type="text" value="1"/>	25.00	✘	<input type="text" value="0"/>	25
BI007	BIBERÓN 280 ml ANTICÓLICO AMARILLO	<input type="text" value="1"/>	27.00	✘	<input type="text" value="0"/>	27
Total: 52						

**Formas de pago**

Cancelado-Entregado
  Cancelado-Pendiente
  Credito-Entregado
  Credito-Pendiente

[Comprar](#)

Codigo	Nombre	Color	Precio
BA001	BARBIJOS C/ELASTICO X 50 PC	BLANCO	18.00
BI001	BIBERÓN 150 ml	CELESTE	23.00
BI002	BIBERÓN 150ml	ROSADO	23.00
BI003	BIBERÓN 280ml	CELESTE	25.00
BI004	BIBERÓN 280ml	ROSADO	25.00
BI005	BIBERÓN 150ml ANTICÓLICO	AMARILLO	25.00
BI006	BIBERÓN 150ml ANTICÓLICO	VERDE	25.00
BI007	BIBERÓN 280 ml ANTICÓLICO	AMARILLO	27.00
BI008	BIBERÓN 280ml ANTICÓLICO	VERDE	27.00
BI009	BIBERÓN BOCA ANCHA 300ml	BLANCO	28.00
BM001	BISTURI CON MANGO X 10 UNIDADES	CROMADO	30.00
BO001	BOTIQUIN MINI DE PRIMEROS AUXILIOS DE PLASTICO	BLANCO	25.00
BO002	BOTIQUIN MEDIANO DE PRIMEROS AUXILIOS DE PLASTICO	BLANCO	30.00

Figura 3.31: Interfaz de agregar pedido

Fuente: Elaboración propia

### **3.7. Seguridad**

El sistema está diseñado para funcionar de manera local como de manera remota, también para funcionar en la web es decir en internet es por esas circunstancias se debe tomar muy en cuenta la seguridad.

#### **3.7.1. Control de Acceso**

El sistema cuenta con acceso restringido a dos niveles los cuales son:

- **Autenticación:** Todos los usuarios son autenticados mediante un usuario y contraseña asignado de manera individual cuando ingresan al sistema.
- **Roles:** Los usuarios tiene diferentes niveles y contextos de acceso al sistema es decir que todos los usuarios no tienen los mismos privilegios, estos privilegios son asignados de acuerdo al rol que desempeñan los usuarios.

#### **3.7.2. Seguridad de Acceso**

El software, se identifica antes de acceder a la base de datos solicitada, con un identificador de usuario y contraseña para poder iniciar sesión en el sistema.

Las contraseñas están protegidas, mediante un código de 32 caracteres, generados por y algoritmo de encriptación (MD5).

El sistema se encuentra desarrollado bajo el patrón MVC (Modelo Vista Controlador) lo cual permite que el acceso al sistema está protegido por tres capas, las cuales permiten que no se pueda interactuar directamente con la base datos, la vista es la interfaz entre el sistema y el usuario

mediante la que realizan las peticiones, el modelo se encarga de obtener la información de la base de datos y el controlador está encargado de intercambiar información entre la vista y el modelo y viceversa. Se utiliza sesiones para autenticar a los usuarios activos, lo cual impide que se pueda acceso al sistema sin estar autenticado.

### 3.7.3. Seguridad de la Información

El administrador del sistema es el responsable de realizar respaldos de la información periódicamente. El sistema genera una gran cantidad de información la cual es constante mente añadida y actualizada lo cual implica que se necesitan respaldos de manera continua lo cual obliga al administrador del sistema a generar de manera diaria los respaldos y también deberán ser respaldados todos los archivos que fueron modificados o creados.

### 3.7.4. Arquitectura del Sistema

El Sistema presenta la siguiente arquitectura

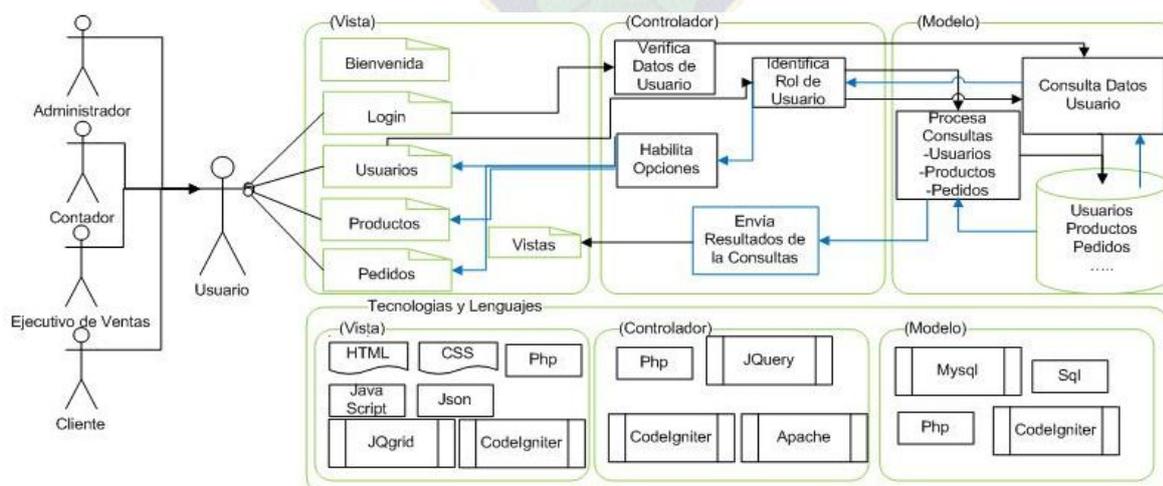


Figura 3.32: Arquitectura del sistema  
Fuente: Elaboración propia

## CAPITULO IV: Calidad de Software

### 4.1 Introducción

Tomando como base el estándar ISO 9126, se presenta la aplicación de la métrica punto función se desarrollaran a continuación.

### 4.2 Confiabilidad

La Confiabilidad es la propiedad del sistema en cumplir su función de manera eficiente.

Se tomó como referencia un intervalo de tiempo de 4 días y 8 ocho horas de uso por día teniendo como resultado un error 0.001%

Dada por la fórmula:  $R(t) = e^{-\lambda t}$

Dónde:  $\lambda$  es el error (0.001) y  $t$  es el intervalo de tiempo (4 días a 8 horas por día)

$$R(t) = e^{-\lambda t} = e^{-(0.001 \cdot 32)}$$

$$R(t) = 0.96 \cong 96\%$$

### 4.3 Usabilidad

La usabilidad no expresa la facilidad de uso e intuición de manejo de un determinado sistema.

**Tabla 4.1:** Cuestionario para Calcular la Usabilidad

Pregunta	Ponderación
¿La interfaz del usuario es amigable?	95%
¿El manejo del sistema es comprensible y sencillo?	95%
¿Son las operaciones y la navegación intuitivas desde la interfaz del usuario?	95%
¿El Sistema satisface todos los requerimientos?	90%
¿Los datos de salida son confiables?	95%
¿El tiempo de respuesta de la aplicación es óptimo?	95%

Fuente: Elaboración Propia

$$U = \frac{\sum xi}{n} = \frac{565}{6} = 94.1\% \text{ de Usabilidad}$$

## 4.4 Funcionalidad

### 4.4.1 Punto Función

La funcionalidad se valora evaluando el conjunto de características y generalidad de las funciones.

La funcionalidad se valora evaluando el conjunto de características y generalidad de las funciones entregadas, para se tiene la siguiente relación.

$$PF = CuentaTotal * [0.65 + (0.01 * \sum F_i)]$$

Donde:

$F_i$  : Valores De Complejidad

$PF$  : Medida De Funcionalidad Entregada a La Aplicación como valor de Normalización.

$Cuenta Total$  : La Suma de los siguientes datos

Se debe determinar cinco características de dominios de información las cuales son:

- Número de entradas de usuario.
- Número de salidas de usuario.
- Número de peticiones de usuario.
- Número de archivos.
- Número de interfaces externas.

A continuación se detalla los datos necesarios para ubicar los parámetros anteriormente mencionados; los datos se obtienen del sistema que se está desarrollando.

**Tabla 4.2:** *Tabla De Ponderación*

Parámetros de Medición	Cuenta	Factor de Ponderación			Total
		Simple	Medio	Complejo	
Número de entradas de usuario	7	3*0	4*7	6*0	28
Número de salidas de usuario	10	4*3	5*3	7*4	55
Número de peticiones de usuario	10	3*0	4*5	6*5	50
Número de archivos	10	7*0	10*5	15*5	125
Número de interfaces externas	4	5*0	7*4	10*0	28
<b>Cuenta Total</b>					286

*Fuente: Elaboración Propia*

El factor de ponderación de desarrolla de acuerdo a número de módulos encontrados en el sistema.

Para el cálculo del factor de ajustes de complejidad se debe responder a cada una de las preguntas que se formulan en la Tabla 4.2, según el valor de ajustes de complejidad.

**Tabla 4.3:** *Ajuste de Complejidad del punto función*

ESCALA	Sin Importancia	Incidental	Moderado	Medio	Significativo	Esencial
	0	1	2	3	4	5
Factor						
1. Requiere el sistema copias de seguridad y de recuperación fiable						X
2. Se requiere comunicación de datos						X
3. Existen funciones de procesos distribuidos.			X			
4. Es critico el rendimiento				X		

5. Será ejecutado el sistema en un S.O. existente.			X		
6. Requiere el sistema entrada interactiva.				X	
7. Requiere de entrada de datos interactiva sobre múltiples ventanas.				X	
8. Se actualizan los archivos maestros de forma interactiva.				X	
9. Son complejos las entradas, las salidas, los archivos a las peticiones.	X				
10. Es complejo el procesamiento interno.				X	
11. Se ha diseñado el código para ser reutilizable.				X	
12. Están incluidas en el diseño la conversión y la instalación.				X	
13. Se ha diseñado el sistema para soportar múltiples instalaciones.				X	
14. Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario.					X

*Fuente: Elaboración Propia*

Con los valores de ajustes de complejidad obtenidos en la tabla 4.2 se calcula la sumatoria de los  $F_i$ .

$$\sum F_i = 51$$

Luego se calcula el punto función del sistema:

$$PF = Cuenta\ Total * [R(t) + (0.01 * \sum F_i)]$$

$$PF = 420.42$$

Con  $PF$  igual a 572.45 concluimos según la escala de funcionalidad definida en la Tabla 4.3, que el sistema tiene una funcionalidad óptima.

**Tabla 4.4:** Escala de Funcionalidad

ESCALA	FUNCIONALIDAD
PF > 300	Óptimo
200 > PF > 300	Buena

100 > PF > 200	Suficiente
PF < 100	Deficiente

TABLA APLICATIVO

PF < 300	Aplicativo Pequeño
300-600	Aplicativo Mediano
PF > 600	Aplicativo Grande

Por lo que podemos decir que el software tiene:

- Tamaño Mediano
- Complejidad Baja
- Esfuerzo PF=420.42
- Tiempo 6 meses
- $\frac{PF}{PF_{Max}} = \frac{420.42}{468.12} = 0.90$
- 0.90 = 90% de Funcionalidad

#### 4.5 Métricas de Mantenimiento

El análisis de este tipo de mantenimiento se lo realiza mediante el método del índice de madurez propuesta por el estándar IEEE982.1-1988 que proporciona una indicación de la estabilidad de un producto de software (basada en los cambios que ocurren con cada versión del producto)

Para calcular la mantenibilidad con el estándar IEEE982.1-1988 se debe calcular la siguiente información.

$M_t$  = Numero de módulos en la versión actual.

$F_c$  = Numero de módulos en la versión actual que se han cambiado.

$F_a$  = Numero de módulos en la función actual que se han añadido.

$F_d$  = Numero de módulos en la versión anterior que se han borrado en la versión actual.

La relación matemática sugerida es la siguiente:

$$IMS = \frac{[M_t - (F_a + F_c + F_b)]}{M_t}$$

Luego si el valor de IMS se aproxima a 1, se dice que el software empieza a estabilizarse.

**Tabla 4.5:** Información requerida por el IMS

Información	Valores Obtenidos
$M_t$	8
$F_c$	1
$F_a$	0
$F_d$	0

*Fuente: Elaboración Propia*

Por lo que se obtiene el siguiente resultado:

$$IMS = \frac{8 - (1 + 0 + 0)}{8} = 0.875$$

88% de Mantenibilidad

Luego se puede concluir que el presente proyecto tiene una estabilidad del 88%

### 4.5.1 Métricas de Portabilidad

Un sistema es portable, si hablando de costos, el costo de transportar y adaptar a un nuevo ambiente es menor al costo de rediseñar el sistema para el mismo ambiente. La relación que nos permite determinar el grado de portabilidad es el siguiente:

$$\text{Grado de Portabilidad (GP)} = (1 - \text{Costo de Portabilidad}) / \text{Costo de Rediseño}$$

$$\text{Portabilidad Efectiva: } GP > 0 \quad ; \quad \text{Portabilidad Perfecta: } GP = 1$$

$$GP = 1 - \frac{ET}{ER}$$

$$GP = 1 - \frac{1}{5} = 0.8$$

80% de Eficiencia

### 4.5.2 Eficiencia

Este valor nos indica si cumple los objetivos propuestos y además ocupa el mínimo de recursos para su funcionamiento.

$$\text{Eficiencia}_{sist} = \frac{\text{disponibilidad} * \text{Confiabilidad} * \text{Mantenibilidad} * \text{Capacidad}}{\text{Costo Ciclo de Vida}}$$

$$\text{Eficiencia}_{sist} = \frac{0.9 * 0.96 * 0.9 * 0.95}{0.8} = 0.92$$

92% de Eficiencia

### 4.5.3 Transportabilidad

El presente proyecto, respecto al programa puede ser transportado a cualquier ambiente que tenga entorno Windows pero con respecto a la venta en línea si se tiene que realizar mediante internet y las características de Hardware mencionadas en el capítulo III y si se desea instalar en un ambiente cliente/ servidor, se necesita una red; respecto al gestor de base de datos.

## 4.6 Estimación del Costo del Proyecto

### 4.6.1 Métricas Orientadas Al Tamaño

Calcula el tamaño en función a la utilización de un lenguaje (Número de líneas de código Fuente o sentencias del lenguaje de Programación). Para poder medir el tamaño de un Software utilizamos el programa o Modelo COCOMO Orientado a líneas de código, mide el proyecto de Software su significado es Modelo Construido de Costo (CONstruch COst MOdel) propuesto por Barry Boehm.

#### 4.6.1.1 Modelo COCOMO

Como ya observamos anteriormente la jerarquía de COCOMO en el capítulo II Tabla C ahora aplicamos la fórmula para calcular el número de líneas de código del proyecto para así medir el esfuerzo que se necesita para desarrollar en un lenguaje.

$$E = a \cdot KLDC^b \text{ personas/mes}$$

Donde:

E: Esfuerzo aplicado

T: Tiempo Aplicado En Mes Cronológico

KLDC: Líneas De Código Expresado En Miles

NRDP: Número de personas para el desarrollo del proyecto

Tomamos los datos de la siguiente tabla y sabiendo que nuestro proyecto es sencillo de 50000

LDC (Líneas de Código) así remplazamos los datos Orgánicos:

**Tabla 4.6:** *Tabla de Estimación Cocomo*

MODO	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

*Fuente: Elaboración Propia*

$$E = (2.4) (8.200)^{1.05} = 21.86 \text{ personas/mes}$$

$$E = (2.5) (2186)^{0.38} = 2186 \text{ mes}$$

$$NPDP = \frac{E}{T} = \frac{21.86 \text{ personas/mes}}{8.07 \text{ mes}}$$

$$NPDP \approx 271 \text{ person}$$

Lo cual equivale al número de personas a desarrollar:

$$NPDP \approx 3 \text{ person}$$

Y observando la tabla vemos que nuestra categoría es Media con 2 a 5 programadores que tendrá una duración de 6 meses a 1 año con tres mil líneas de códigos.

**Tabla 4.7:** *Tabla de Categorías Modelo de Estimación Cocomo*

<b>Categoría</b>	<b>Programadores</b>	<b>Duración</b>	<b>Líneas de Código</b>
Trivial	1	0 – 4 semanas	<1 K
Pequeño	1	1 - 6 meses	1K-3K
Media	2-5	6 meses - 1 años	3K-50K
Grande	2-5	2 - 3 años	50K-100K
Muy Grande	2-5	4 -5 años	100K-1MB
Gigante	1000-5000	5 -10	>1k

*Fuente: Elaboración Propia*

#### **4.7 Conclusión**

En este capítulo se ha realizado la medición y valoración del sistema para ello se ha aplicado las métricas de calidad de software y se demostró que el sistema desarrollado cumple las métricas por tanto el software desarrollado es de calidad.

## CAPITULO V:Conclusiones y Recomendaciones

### 4.7 Conclusiones

El proyecto de grado titulado “Sistema de Información de Control y Seguimiento de Ventas On-Line De Productos Farmacéuticos Distribuidora PHARMICA“ cumple con los objetivos previstos puesto que produce y proporciona información la cual tiene seguimiento, dicha información puede ser expresada de manera cuantitativa para la toma de decisiones.

- Se aplicó la metodología de Proceso Unificado (RUP), para realizar el análisis y diseño del sistema, utilizando la notación del lenguaje de Modelado Unificado.
- Se logró crear una base de datos relacional, proporcionando información por medio de graficas estadísticas.
- El sistema de Información permite reducir el tiempo de proceso y entrega de información en las áreas de ventas, almacenes, productos, usuarios,

### 5.2 Recomendaciones

En base al estudio realizado y siguiendo con el objetivo de la propuesta sobre la modificación del sistema, se recomienda que para diseños posteriores de la misma:

Incorpore estrategias y tecnologías de transacciones monetarias.

Además podría desarrollarse un módulo de Marketing donde se pueda acceder a información sobre promociones, ofertas y otro tipo de promociones de los productos.

Aplicar tecnología responsiva al sistema para que sea totalmente compatible con los diferentes dispositivos existentes en la actualidad.

## 6. Bibliografía

- HERNÁNDEZ, R. (2005): Metodología de la Investigación. México, D.F.: Editorial McGraw-Hill Interamericana, 3ra Edición.
- ALCALDE, E. (1997): Informática básica. México: 2da. Edición, 1997.
- RUMBAUGH, J. (1999): El Lenguaje Unificado de Modelado. España: Editorial Addison Wesley Iberoamericana, 2da Edición.
- PRESSMAN, R. (2010): Ingeniería del Software un Enfoque Práctico. México, D.F.: Editorial McGRALL-HILL, 7ma Edición.
- JACOBSON, I. (2000): El proceso de unificado de Desarrollo de Software. Madrid: Editorial Person Educación S.A.
- MURRAY, R. (1998): La estadística. España: Editorial Schaum, 3era Edición, 2002.
- MONTERO, M. (2003): La Estadística Informática, España: Editorial Schaum, 1era. Edición.
- RUP, (2007): Departamento de Sistema de Información, Universidad Politécnica de Valencia.  
Disponible en: <http://pid.dsic.upv.es>
- SANTANDER, (2007): Metodología de la investigación.  
Disponible en: <http://www.aibarra.org/investig/tema0.htm>
- ATAJO, CIENCIA (2006): Análisis de stakeholder  
Disponible en: <http://www.elatajoparaejecutivos.com>
- SILVA (2000): Construyendo Aplicaciones Web con una metodología de Diseño Orientada a Objetos. Disponible en:  
[www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)
- ADDISON, W. (1999): Lo mejor de Fowler, recetas organizadas para optimizar la

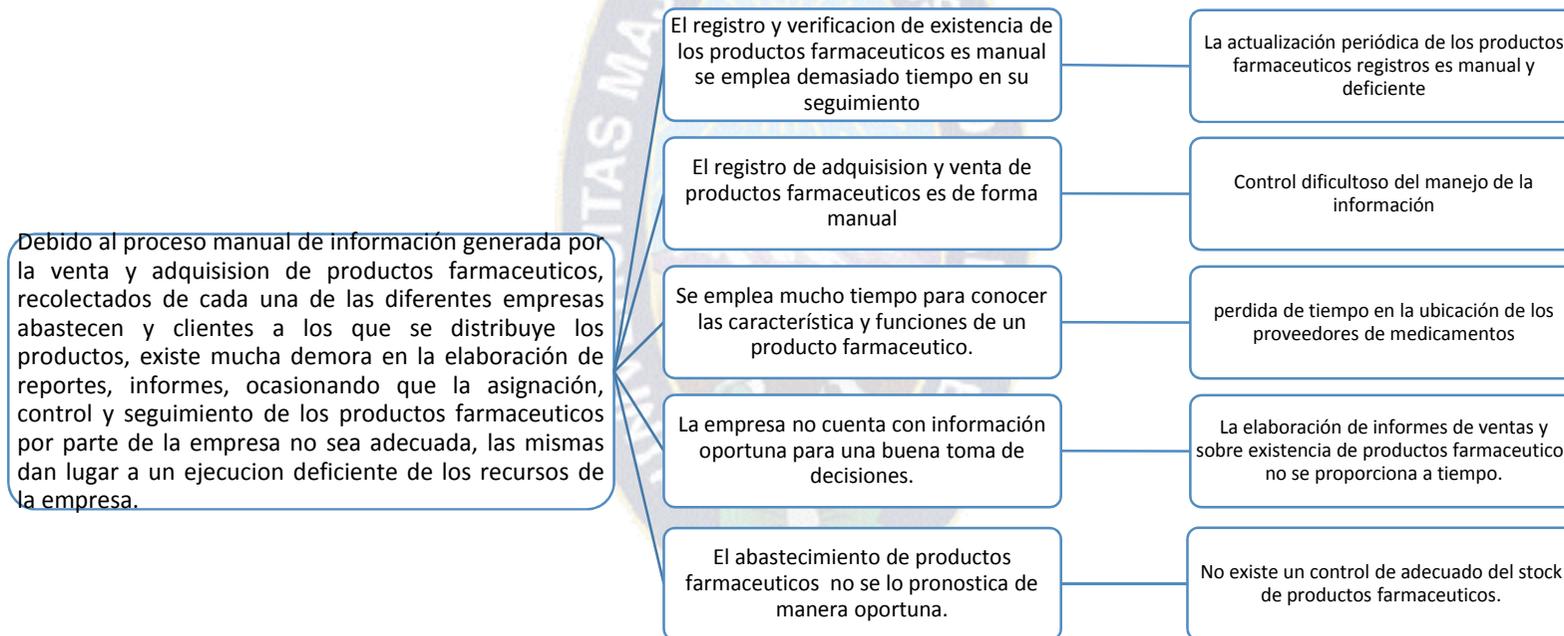
- Eficiencia del código Java. Todo presentado con notación UML. Para arquitectos y programadores. Massachusetts, 1999
- POLO, M. (2004): Calidad en Mantenimiento de Software. España: Capítulo 13.
- CASTRILLÓN A., DANIEL & CASTRO Q., GILDARDO. (2000): Metodología para el aseguramiento de la calidad del software. Tesis Universidad EAFIT, Medellín, 2000. Capítulo 4.
- Diccionario de Internet y Redes de Microsoft, Edit. Mc Graw Hill, 2003, Madrid – España.
- LARMAN C., Uml y Patrones, Editorial Pearson, 2º Edición, 2004
- GERMER J., NARAMORE E., OWENS M., WARDEN M. Professional Lamp, Wiley Publishing, Indiana, 2006.
- SAWYER D. JavaScript & jQuery the missing manual, Editorial O'REILLY, 2ª Edición, Estados Unidos, 2011.
- SUEHRING S., CONVERSE T., PARK J., Php 6 MySQL 6 Bible, Wiley Publishing, Indiana, 2009.
- BABIN L., Beginning Ajax with PHP, Editorial Apress, Estados Unidos, 2007.
- RATIONAL, 2000 Rational Unified Process. Version 2000. Rational Software Corporation. USA. 2000.
- LOZOYA L. (28 de 09 de 2010). Clases del Comercio Electrónico. Recuperado el 17 de 04 de 2013, de <http://suite101.net/article/clases-de-comercio-electronico-b2b-b2c-b2a-b2e-c2c-c2g-b2g-a26589#axzz2QIJ56GpO>
- MELÉNDEZ C. (s.f.). Ley de Comercio Electrónico y Delito Informático. Recuperado el 17 de 04 de 2013, de <http://es.slideshare.net/cmelenendez77/comercio-electronico-2155286>

NÚÑEZ, H. (s.f.). Módulo de Aprendizaje del Área Comercial de 3° Año Medio . Recuperado el 17 de 04 de 2013, de <http://www.isca.cl/modulos/tercerosPDF/unidad1.pdf>



**ANEXOS**

# ÁRBOL DE PROBLEMAS



# ÁRBOL DE OBJETIVOS



## CRONOGRAMA

Actividades	2016										
	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre		
Contacto con la institución	■	■									
Evaluación de la disposición de la Información		■									
Elaboración de Encuestas, Cuestionarios y Entrevistas		■									
Detalle de cada una de las Areas			■	■							
Selección de las Areas a Analizar				■							
Identificación del Problema				■							
Observación Directa				■							
Análisis de la Información				■	■						
Levantamiento de Requerimientos					■	■					
Entrega de perfil					■	■	■	■			
Diseño del Sistema						■	■	■	■		
Elaboración de Diagramas						■	■	■			
Elaboración de la parte documental				■	■	■	■	■	■	■	
Desarrollo de Código							■	■	■	■	■
Implementación y pruebas							■	■	■	■	■

DOCUMENTACIÓN