

**UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIA PURAS Y NATURALES
CARRERA DE INFORMATICA**



PROYECTO DE GRADO

**SISTEMA CLIENTE – SERVIDOR DE INVENTARIO
DE PRODUCTOS DE POLIURETANO**

**(INDUMAR S.R.L. INDUSTRIA NACIONAL AL SERVICIO DE LA MINERIA Y LA
INDUSTRIA EN GENERAL)**

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

**POSTULANTE : MAIK LAURO CHAMBI MAMANI
TUTOR : LIC. GERMAN HUANCA TICONA
REVISOR : LIC. JAVIER REYES PACHECO**

**LA PAZ – BOLIVIA
2011**

DEDICATORIA

Dedico este trabajo con mucho cariño a mis padres Edgar Chambi, Hilda Mamani de Chambi y a mi hermanita Rossio Chambi Mamani quienes con su amor, dedicación y paciencia me dieron fuerza para seguir adelante.

AGRADECIMIENTOS

Agradezco:

A Dios por guiar mi camino.

A mi familia que sin su apoyo en momentos críticos quizás no hubiese salido adelante, y que gracias a los consejos de mi padre y de mi madre, a ellos principalmente les doy las gracias.

De gran manera al Lic. German Condori Ticona por su paciencia y colaboración en el transcurso de la elaboración del Proyecto de Grado, de igual manera agradezco a mi revisor Lic. Javier Reyes Pacheco por su apoyo, sugerencias y observaciones que me ayudaron a superar y alcanzar los objetivos trazados.

A la empresa INDUMAR S.R.L. específicamente al departamento de contabilidad a cargo del señor Raul Mamani agradecerle por darme la oportunidad y la información necesaria.

Muchas Gracias...

RESUMEN

INDUMAR S.R.L., se encuentra al servicio del país desde 1987, pioneros en la fabricación de repuestos en POLIURETANO, brindando eficiencia, calidad y puntualidad en la entrega de productos, con la capacidad instalada para cualquier tipo de pedido, y ofreciendo garantías de calidad.

A la fecha se tiene más de 20 años de funcionamiento fabricando variedad de repuestos y productos en poliuretano destinados a la industria minera, así mismo, INDUMAR S.R.L., se vio obligada a diversificar sus productos para atender los requerimientos de empresas dedicadas a diferentes rubros como ser empresas gráficas, mecánica e industria en general.

En el Departamentos de Contabilidad y Mercado – Ventas se observaron diversos problemas en el manejo de información, ya que estas estaban realizadas de forma manual sin contar con un sistema computarizado capaz de informar el movimiento de inventario de materia prima y productos terminados, por lo que no existía información oportuna y precisa en cuanto a emisión de reportes, comunicación de sucursal – central, y otros.

El presente Proyecto de Grado tuvo la finalidad de diseñar un sistema para dar solución a los distintos problemas y satisfacer los requerimientos de la empresa INDUMAR S.R.L. como mejorar el control de la entrada y salida de mercadería desde y hasta almacenes en base a esta y a otras necesidades se desarrollo un sistema que cumple con las expectativas de la empresa, donde el usuario puede realizar consultas via web para tener un control del movimeinto de mercadería. Para el desarrollo del sistema se tomo la metodología OOHDM de la mano de UML, el lenguaje de programación para la aplicación es PHP/MySQL.

Con este sistema se puede ayudar de gran manera al personal, facilitando la toma de desiciones para diferentes operaciones que se efectua diariamente ya que ahora se puede obtener una infomación rápida.

INDICE

CAPITULO I	MARCO REFERENCIAL	PÁGINA
1.1.	INTRODUCCION.	1
1.2.	ANTECEDENTES.	2
1.3.	DEFINICION DEL PROBLEMA.	4
1.3.1.	ANALISIS DE LA SITUACION.	4
1.3.2.	IDENTIFICACION DE PROBLEMAS.	4
1.3.3.	FORMULACION DEL PROBLEMA.	5
1.4.	JUSTIFICACION.	5
1.4.1.	JUSTIFICACION SOCIAL.	5
1.4.2.	JUSTIFICACION TECNICA.	6
1.4.3.	JUSTIFICACION ECONOMICA.	6
1.5.	PLANTEAMIENTO DE OBJETIVOS.	6
1.5.1.	OBJETIVO GENERAL.	6
1.5.2.	OBJETIVOS ESPECIFICOS.	7
1.6.	ALCANCES.	7
1.7.	APORTES.	8
1.8.	MARCO DE REFERENCIA.	8
1.8.1.	MARCO TEORICO.	8
1.8.2.	MARCO CONCEPTUAL.	9
1.9.	ASPECTOS METODOLÓGICOS.	9
1.9.1.	TIPO DE ESTUDIO O INVESTIGACIÓN.	9
1.9.2.	METODOLOGIA EMPLEADA.	11
1.9.2.1.	DIAGRAMA DE CONTEXTO.	12
1.10.	PRESUPUESTO ESTIMADO.	14

2. MARCO TEÓRICO.	15
2.1. CONCEPTO GENERAL DE INVENTARIO.	15
2.1.1. FUNCION DE LOS INVENTARIOS.	16
2.1.2. TIPOS DE MATERIAL O PRODUCTOS EN INVENTARIO.	16
2.1.3. INVENTARIO PERMANENTE.	16
2.1.3.1. KARDEX.	17
2.2. METODOLOGIA.	17
2.2.1. OOHDM.	17
2.2.1.1. OBTENCIÓN DE REQUERIMIENTOS.	18
2.2.1.2. MODELOS CONCEPTUAL.	21
2.2.1.3. DISEÑO NAVEGACIONAL.	22
2.2.1.4. DISEÑO DE LA INTERFAZ ABSTRACTA.	27
2.2.1.5. IMPLEMENTACIÓN.	29
2.2.1.6. COMPARATIVA.	30
2.2.1.7. REQUISITOS TRATADOS.	30
2.2.1.8. GRADO DE DETALLE EN LAS DESCRIPCIONES.	31
2.3. HERRAMIENTAS DE DESARROLLO.	32
2.3.1. UML.	32
2.3.2. DIAGRAMAS DE UML.	33
2.3.3. CASOS DE USO.	35
2.3.3.1. ACTORES.	35
2.3.3.2. FORMATO DE CASOS DE USO.	36
2.3.3.2.1. CASOS DE USO DE ALTO NIVEL.	36
2.3.3.2.2. CASOS DE USO CON FORMATO EXPANDIDO.	36
2.3.3.3. DIAGRAMAS DE LOS CASOS DE USO.	37
2.3.3.4. ELEMENTOS ESTRUCTURALES.	38
2.3.4. TRANSFORMACIÓN DE UN DIAGRAMA DE CLASES A UN MODELO ENTIDAD RELACION COMO EXTENSIÓN DE UML.	38

2.3.5. DIAGRAMA DE CLASE.	40
2.3.5.1. RELACION ENTRE CLASES.	40
2.3.6. DIAGRAMAS DE SECUENCIA.	42
2.4. TECNOLOGIAS.	43
2.4.1. WORLD WIDE WEB.	43
2.4.2. INTRANET.	44
2.4.3. JAVASCRIPT.	44
2.4.4. BASE DE DATOS.	45
2.4.5. GESTOR DE BASE DE DATOS.	45
2.4.6. SQLSERVER.	46
2.4.7. FLASH.	46
2.4.8. SERVIDOR APACHE.	47
2.4.8.1. CARACTERÍSTICAS DE APACHE.	47
2.4.9. AJAX.	47
2.5. TECNOLOGÍA CLIENTE SERVIDOR.	48
2.5.1. DESDE UN PUNTO DE VISTA CONCEPTUAL.	48
2.5.2. EN TERMINOS DE ARQUITECTURA.	48
2.5.3. DEFINICION DE IBM.	48
2.5.4. ¿QUE ES UN ARQUITECTURA?.	48
2.5.5. CLIENTE.	49
2.5.6. SERVIDOR.	49
2.5.7. PROCESO DISTRIBUIDO.	50
2.5.8. ELEMENTOS DE LA ARQUITECTURA C/S.	50
2.5.9. CARACTERISTICAS DEL MODELO C/S.	52
2.5.10. VENTAJAS DEL MODELO C/S.	53
2.5.11. SERVICIOS BASADOS EN C/S.	54
2.6. CALIDAD DE SOFTWARE.	55
2.6.1. MODELO DE CALIDAD ESTABLECIDO POR EL ESTÁNDAR ISO 9126.	56
2.6.2. CARACTERÍSTICAS PROPUESTAS POR LA ISO 9126.	57

2.6.2.1. FUNCIONALIDAD.	57
2.6.2.2. CONFIABILIDAD.	57
2.6.2.3. USABILIDAD.	58
2.6.2.4. MANTENIBILIDAD.	58
2.6.2.5. PORTABILIDAD.	59

CAPITULO III MARCO APLICATIVO

3. MARCO APLICATIVO.	60
3.1. DEFINICION DEL SISTEMA.	60
3.2. FUNCION DEL SISTEMA.	60
3.3. DESARROLLO DEL PROYECTO.	60
3.3.1. ESTRUCTURA ORGÁNICA.	60
3.3.2. DEPARTAMENTOS DE CONTABILIDAD Y MERCADO – VENTAS.	61
3.3.3. OBTENCION DE REQUERIMIENTOS.	62
3.3.4. OBTENCIÓN DE REQUISITOS.	62
3.3.5. IDENTIFICACIÓN DE ACTORES Y TAREAS.	63
3.3.6. ESPECIFICACIÓN DE ESCENARIOS.	65
3.3.7. ESPECIFICACIÓN DE CASOS DE USO.	68
3.3.8. CASOS DE USO EXTENDIDO.	69
3.3.9. DIAGRAMAS DE SECUENCIA.	76
3.3.10. DIAGRAMAS DE COLABORACIÓN.	79
3.3.11. DIAGRAMA DE DESPLIEGUE DEL SISTEMA.	82
3.3.12. ESPECIFICACIÓN DE UID's.	83
3.4. MODELO CONCEPTUAL.	85
3.4.1. DIAGRAMA DE CLASES.	85
3.5. DISEÑO NAVEGACIONAL.	86
3.6. DISEÑO DE INTERFAZ ABSTRACTA.	87
3.7. IMPLEMENTACIÓN.	90

CAPITULO IV

METRICAS DE CALIDAD

4. METRICAS DE CALIDAD.	93
4.1. FUNCIONALIDAD.	93
4.1.1. MÉTRICAS BASADAS EN EL PUNTO FUNCIÓN.	93
4.2. CONFIABILIDAD.	98
4.3. FACILIDAD DE MANTENIMIENTO.	99
4.4. PORTABILIDAD.	101

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5. CONCLUSIONES Y RECOMENDACIONES.	103
5.1. CONCLUSIONES.	103
5.2. RECOMENDACIONES.	103
REFENCIAS BIBLIOGRAFICAS.	105
REFERENCIAS WEB.	106
ANEXOS.	107
DOCUMENTOS.	111

CAPITULO I

1.1. INTRODUCCION.

Las empresas de nuestro país que producen bienes y servicios utilizan métodos para minimizar costos, basados en experiencias o en procesos que ellos mismos definen como lógicos, que hay resultados a veces inesperados para los propietarios, es decir, que sus métodos no son los más adecuados para el manejo de información ni tampoco para acrecentar sus producción y disminuir sus costos.

De la misma manera muchas de las empresas de nuestro país manejan la información manualmente por lo mismo el trabajo es lento y susceptible a errores además que la información no se la tiene en el momento preciso provocando demora en la toma de decisiones por lo mismo ocasiona pérdida de clientes.

Con el fin de registrar y controlar los inventarios, las empresas adoptan los sistemas pertinentes para valuar sus existencias de mercancías con el fin de fijar su posible volumen de producción y ventas.

Lo que se pretende realizar es un sistema de inventario permanente por medio de este sistema la empresa conoce en valor de la mercancía en existencia en cualquier momento, sin necesidad de realizar un conteo físico, porque los movimientos de compra y venta de mercancías se registra directamente en el momento de realizar las transacción a su precio de costo.

Las empresas que adoptan este tipo de sistema deben llevar un auxiliar de mercancías denominado kardex en el cual se registra cada artículo que se compre o que se venda. La suma y la resta de todas las operaciones en un periodo como resultado el saldo final de mercancía.

1.2. ANTECEDENTES.

Inventarios son bienes tangibles que se tienen para la venta en el curso ordinario del negocio o para ser consumidos en la producción de bienes o servicios para su posterior comercialización. Los inventarios comprenden, además de las materias primas, productos en proceso y productos terminados o mercancías para la venta, los materiales repuestos y accesorios para ser consumidos en la producción de bienes fabricados para la venta o en la prestación de servicios; empaques y envases y los inventarios en tránsito.

INDUMAR una empresa industrial que con aporte de capitales bolivianos se dedica al desarrollo, fabricación y comercialización de productos de Poliuretano orientados principalmente al sector minero e industrial.

La visión de la empresa es ser líder y en constante innovación en la fabricación y comercialización de productos de poliuretano para que el sector minero nacional y con gran proyección hacia el mercado exterior desarrollando nuevos productos y procesos que nos permitan incrementar nuestro nivel de ventas.

La empresa está en expansión y por lo mismo la comunicación de la central a sucursales tiene que ser fácil y rápida por lo que se implementa CLIENTE SERVIDOR.

Algunos puntos resaltantes son:

- ✓ La empresa fabrica una serie de productos.
- ✓ El stock no se dedica solo a la fabricación de un producto
- ✓ La empresa puede fabricar desde una tasa hasta un forro de máquina de succión que son máquina de minería
- ✓ El sistema tiene que tener la capacidad de poder crear ITEMS variados para productos fabricados.
- ✓ Cada producto tiene una hoja de ruta donde en cada sección anotan que cosas se gastan (costos directos e indirectos).

- ✓ El sistema además actuara de la forma CLIENTE SERVIDOR donde las diferentes sucursales podrán consultar desde un punto al servidor información actualizada de costos e inventarios.
- ✓ CLIENTE SERVIDOR permitirá crear múltiples usuarios, con el que se podrá designar a cada uno su trabajo y la administración correspondiente, pero por sobre todo consultar desde cualquier sucursal.
- ✓ Además el sistema nos ayudara a tener un control de calidad desde la adquisición de materias primas hasta los productos terminados, las materias primas son importadas desde EEUU garantizando la calidad de productos.

Los trabajos y estudios realizados no se enmarcan en la realidad nacional porque se necesitan sistemas de información que sean aplicados a una determinada empresa es decir se debe realizar un estudio y análisis de la empresa y elaborar un sistema que sea de su medida y cumpla sus requerimientos y expectativas como algunos sistemas elaborados en Carrera de Informática:

- Calculo de costos y gestión de información “El Dorado” (T713)
- Sistema de información de control de inventarios “Corporación Boliviana de bebidas” (T331)

Los proyectos mencionados no pueden ser implementados en la empresa INDUMAR porque no actúan como Cliente-Servidor lo que ocasiona es que la información de central a sucursal no sea exacta ni actualizada por lo que los antecedentes mencionados anteriormente motivan a que el presente trabajo sea desarrollado, por ser de gran utilidad para la empresa INDUMAR S.R.L.

1.3. DEFINICION DEL PROBLEMA

1.3.1. ANALISIS DE LA SITUACION.

La empresa INDUMAR S.R.L. maneja la información de manera manual y el volumen de información es considerable, no pudiendo así obtener resultados en el tiempo deseado.

La forma en que se maneja los datos no es la más adecuada ya que se debe registrar, clasificar y recuperar los datos de formas más eficiente y oportuna, tanto para actualizar los datos en el proceso productivo como el control de información de inventarios en insumos, materia prima, producto terminado; así mismo la comunicación de central – sucursal de la empresa.

1.3.2. IDENTIFICACION DE PROBLEMAS.

En la empresa INDUMAR se observo que no se cuenta con un sistema de inventario y costo de productos de Poliuretano es por lo cual se analizan las distintas deficiencias:

- El registro de inventarios en almacenes es lento, tedioso y predispuesto a cometerse errores, ya que es un proceso manual, esto provoca la falta de información oportuna y retrasa el abastecimiento de material.
- La clasificación de inventarios y actualización de precios es dificultoso ya que se cuenta con stock voluminoso, además de existir pérdida y deterioro de material en alguna ocasión.
- Ausencia de una Base de Datos que contenga esta información (Inventario).

- Existe una mala comunicación entre sucursal y central para la consulta de información actualizada, este problema provoca demora en pedidos e incluso en entregas a los clientes.
- No se cuenta con reportes actualizados de todo lo que la empresa fabrica.

1.3.3. FORMULACION DEL PROBLEMA.

Sobre la base de un estudio de información recopilada en base a entrevistas, cuestionarios dentro de la institución se pudo formular la siguiente pregunta en forma general.

¿Con un sistema informático capaz de optimizar, informar, almacenar el manejo de existencias en almacén de cada producto terminado o materia prima se mejorara la administración del registro de producción y almacenes de la empresa INDUMAR S.R.L.?

1.4. JUSTIFICACION

1.4.1. JUSTIFICACION SOCIAL

El presente trabajo contribuye a mejorar la atención al personal de la empresa INDUMAR S.R.L., público en general y a otras empresas dedicadas a la industria al servicio de la minería agilizando el servicio para que este ya no realice las esperas de respuesta, en caso de las personas ajenas a la empresa, a recibir información oportuna y fidedigna de forma rápida y digitalizada.

1.4.2. JUSTIFICACION TECNICA

Considerando que en la actualidad se encuentra la tecnología disponible, la empresa INDUMAR S.R.L., cuenta con los recursos técnicos necesarios que soportan la inserción del sistema de información que se propone en el presente proyecto de grado donde se especifica en la siguiente tabla:

Materiales	Descripción	Estado Actual
Una PC	Para uso del servidor	Existe
Instalación de la red interna	Cableado Estructurado listo para la conexión en intranet	Existe
PC's	Para la instalación del nuevo sistema.	Existe
Impresoras	Para proceder a la impresión de la documentación.	Existe
Sistema Operativo.	Equipos configurados con Sistema Operativo.	Existente
Acceso a Internet	Para implementar Cliente - Servidor	Existe

1.4.3. JUSTIFICACION ECONOMICA

El sistema cliente – servidor de inventario de productos de poliuretano permitirá contar con información oportuna y valida, incluyendo a la comunicación de sucursal y central. Así reduciendo el costo económico y tiempo que significa actualmente el proceso de la información en cuanto se refiere al registro, consulta y control de producción y almacenes.

1.5. PLANTEAMIENTO DE OBJETIVOS

1.5.1. OBJETIVO GENERAL

“Desarrollo de un Sistema Cliente – Servidor de inventario de productos de poliuretano que permita optimizar los procesos de organización, registro y control de los productos de poliuretano en la empresa INDUMAR S.R.L. (industria nacional al servicio de la minería)”

1.5.2. OBJETIVOS ESPECIFICOS

- ✓ Fijar el modelo de control de inventarios que proporcione información permanente sobre la existencia de stock en almacén.
- ✓ Diseñar un sistema de control de producción diaria a través de planillas que permita proporcionar información oportuna.
- ✓ Analizar y Evaluar la situación actual del registro y control de los productos de poliuretano.
- ✓ Desarrollar una base de datos que permita el registro, consulta, control de los productos de poliuretano.
- ✓ Registrar los productos de poliuretano bajo la idea de resguardar y tener a disponibilidad inmediata reportes de almacenes.

1.6. ALCANCES

El sistema brindara servicios en el área de registro de producción y almacenes, agilizando ciertos procesos manuales como ser el registro de materia prima, productos terminados, las búsquedas y obtención de datos oportunos y confiables además de tener un modulo de reportes utilizados para la toma de decisiones.

El sistema de información abarcara los siguientes puntos:

- El sistema manejara toda la información pertinente al ingreso y salida de materia prima e insumos de almacén, al mismo tiempo que registraran los productos terminados para ser llevados a almacén.
- Se registrara diariamente la producción de cada productos terminado.
- El control de producto terminado en almacenes facilitara al personal en sus tareas cotidianas como ser: actualización de precios de producto, control automatizado de stock mínimo, evitar faltantes de material.
- El sistema registrara los pedidos de diferentes productos terminados y la orden de compra de materia prima e insumos faltantes para la producción.

1.7. APORTES

El proyecto de grado beneficiara a la empresa INDUMAR S.R.L. en el acceso y administración de la información, en la manipulación de datos reduciendo tiempo y esfuerzo en la búsqueda de los mismos. Además de ser un aporte para la Industria Nacional dedicada a la fabricación de productos de Poliuretano.

Los informes y reportes que emitirá el sistema, con información actualizada ayudaran a que el personal de INDUMAR tome decisiones de mejor manera.

1.8. MARCO DE REFERENCIA

1.8.1. MARCO TEORICO

Es necesario registrar y almacenar bienes tangibles de una empresa, en nuestro caso mercadería (materia prima y producto terminado), para realizar tal tarea se manejan inventarios de productos.

Con el fin de registrar y controlar los inventarios las empresas adoptan los sistemas pertinentes para valuar sus existencias de mercancías con el fin de fijar su posible volumen de producción y ventas.

Se pretende manejar un tipo de inventario que es denominado Permanente porque por medio de este sistema la empresa conoce el valor de la mercancía en existencia en cualquier momento, sin necesidad de realizar conteo físico, porque los movimientos de compra y venta de mercancía se registran directamente en el momento de realizar las transacciones a su precio de costo. Junto a este tipo de inventario se maneja un auxiliar denominado Kardex en el cual se registra cada artículo que se compre o se venda, así solo mostrando un saldo final de mercancía.

1.8.2. MARCO CONCEPTUAL

Los términos que nos ayudaran a entender de una manera mejor el presente proyecto son:

Inventario. Asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden y precisión.

Permanente. Estar en algún sitio durante cierto tiempo.

Sistema. Programa que tiene capacidad para dar respuestas semejantes a la que daría un experto en el tema.

Tangible. Que se puede percibir de manera precisa.

Cliente. Persona que utiliza con frecuencia servicios de un profesional o empresa.

ITEM. Cada uno de los elementos que forman parte de un dato, para hacer una distinción.

Poliuretano. Resina sintética obtenida por condensación de poliésteres y caracterizada por su baja densidad (para fabricación de materia prima).

1.9. ASPECTOS METODOLÓGICOS

1.9.1. TIPO DE ESTUDIO O INVESTIGACIÓN

El presente proyecto de grado está orientado a constituirse en un estudio descriptivo porque me ayudara a analizar cómo se manifiesta un fenómeno y sus componentes.

El propósito de este tipo de estudio es la delimitación de los hechos que conforman el problema de investigación, como:

- 1) Establecer las características de los diferentes procesos.
- 2) Identificar el comportamiento de los diferentes procesos.
- 3) Descubrir y comprobar la posible asociación de procesos y usuarios.

INPUT (TODO LO QUE INGRESA AL SI)

A	B
ATRIBUTOS <ul style="list-style-type: none"> • CODIGO • NOMBRE • FECHA • GRUPO • SUB-GRUPO • ETC. 	GERENTE GENERAL - - AUXILIAR - - ALMACENERO - -

OUTPUT (TODO LO QUE SALE DEL SI)

A	B
REPORTES <ul style="list-style-type: none"> • CONSULTAS • ACTUALIZACIONES • ETC 	GERENTE GENERAL - REPORTE GENERAL AUXILIAR - REPORTES LIMITADOS ALMACENERO - REPORTE ALMACEN

PROCESOS (TODOS LOS PROCESOS QUE HACE EL SI)	USUARIOS
CALCULAR REGISTRAR ELIMINAR EDITAR ACTUALIZAR CONSULTAR ETC.	✓ ALMACENERO ✓ AYUDANTE ✓ JEFE ADMINISTRATIVO ✓ ADMINISTRADOR DE SISTEMAS ✓ OTROS

1.9.2. METODOLOGIA EMPLEADA.

Todo método para el desarrollo de sistemas nos indica cómo construir técnicamente el sistema, los métodos para el desarrollo de sistemas abarcan una gran gama de tareas que incluyen análisis, diseño, construcción del programa, pruebas y mantenimiento.

El sistema de información correspondiente al presente proyecto se desarrollará de acuerdo a los requerimientos de la empresa INDUMAR S.R.L., para tal efecto se aplicará la metodología **OOHDM** o método de diseño orientado al objeto de hipermedia, donde se tiene un conjunto de herramientas y técnicas que ayudan al análisis del diseño del sistema (diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación), dentro este conjunto de herramientas se optó por utilizar el Lenguaje de Modelado Unificado (**UML**), el cual nos ayudará a visualizar, construir y documentar nuestro proyecto.

Cuando un proceso implica la construcción de algún producto nos referimos al proceso como ciclo de vida, entonces “el proceso de desarrollo de software suele denominarse ciclo de vida del software porque describe la vida de un producto de software desde su concepción hasta su implementación, entrega y mantenimiento”.

UML es modelado de datos con sus diferentes tipos de diagramas que ayudara a la diagramación de los datos, entre los más importantes:

Diagrama de clases = base de datos

Casos de uso = procesos que harás en el sistema

Diagrama de paquetes = el envolvimiento de los datos

Diagrama de secuencias = como ira la estructura y porque pasos pasaran los distintos procesos del sistema.

OOHDM pinta escenarios en el que se tomara en cuenta todo un entorno, se realizara una tarea minuciosa de averiguar cómo funcionara el sistema; OOHDM es una metodología de investigación que en la mayoría de los casos siempre tiene que ir junto con UML porque a medida que se va con la investigación se va elaborando el sistema y a su vez se irá usando también UML, se empezara con los casos de uso y así sucesivamente.

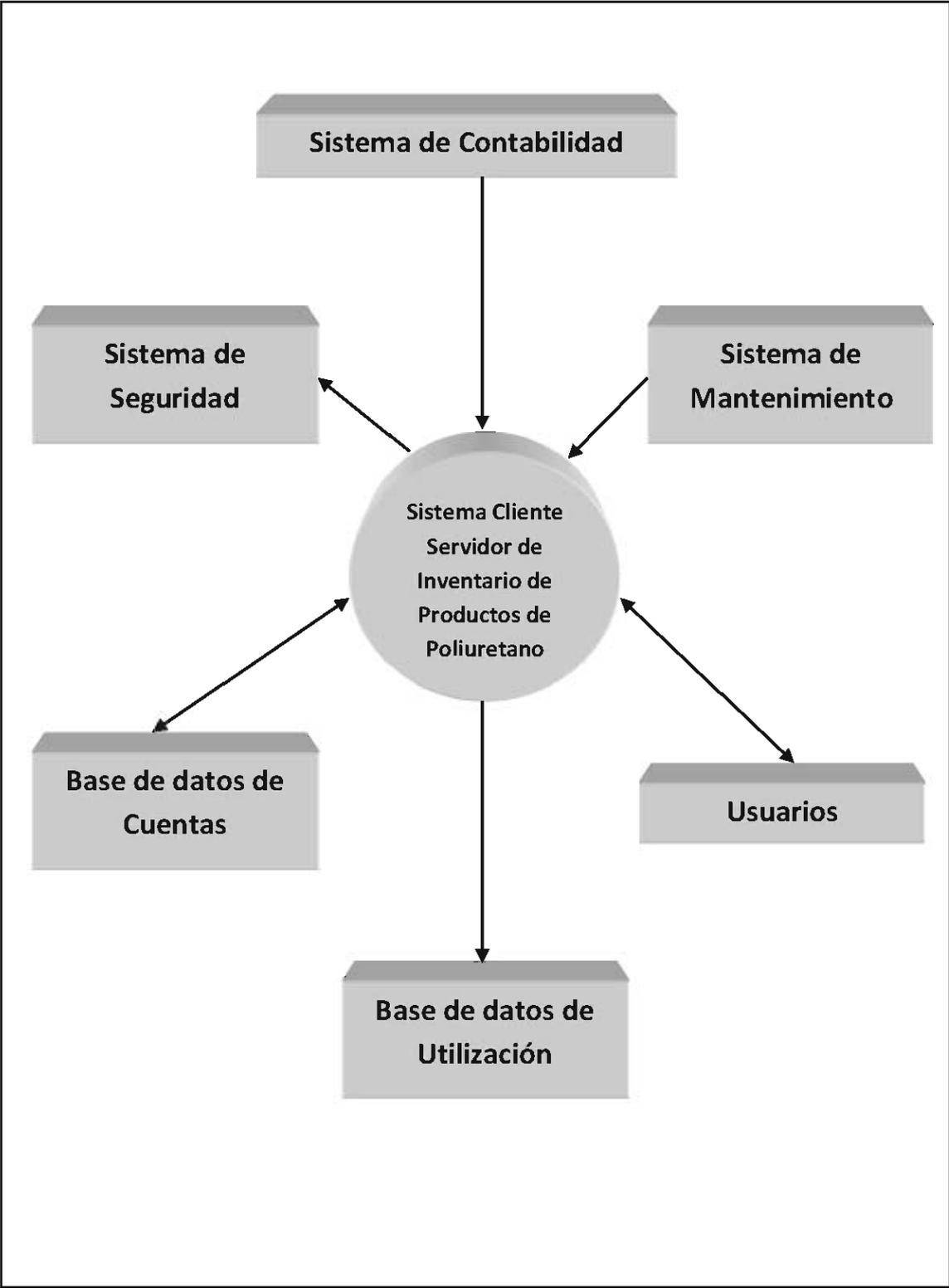
Se utilizará también herramientas para calcular la cantidad de inventario requerido en determinado tiempo, que nos permitirán controlar y planear de manera efectiva la producción de los bienes (productos terminados).

1.9.2.1. DIAGRAMA DE CONTEXTO

En el centro se ubica el nombre del sistema y se rodea de tres tipos de entidades:

- ✓ Los demás sistemas con los que interactúa.
- ✓ Los usuarios del sistema
- ✓ Los centros de información a los cuales necesita acceder para realizar su función correctamente.

El sistema se conecta a estas entidades por medio de flechas (uni o bidireccionales). Las unidireccionales significan que el sistema toma o entrega información a las entidades. Las bidireccionales muestran intercambios de información entre el sistema y su entorno.



1.10. PRESUPUESTO ESTIMADO.

El análisis de costes/beneficios determina los costes para el desarrollo del proyecto y los pondera con los objetos tangibles del sistema.

El desarrollo del sistema involucra costos en diferentes aspectos que a continuación los clasificaremos en:

- Costos de Sistema
- Costos de Personal
- Costo de Operación
 - Elaboración
 - Impresión
 - Trámites de administrativos
- Costo de Equipamiento

Los costos expresados en valores monetarios son:

<u>COSTOS</u>	<u>VALOR MONETARIO EN Bs.-</u>
SISTEMA Y PERSONAL	NO DETERMINADO
OPERACIÓN	2500.00.-
EQUIPAMIENTO	(Será otorgado por la institución)
<u>TOTAL</u>	2500.00.-

CAPITULO II

MARCO TEÓRICO.

El propósito de este capítulo es el de establecer una visión teórica de los procesos involucrados en el desarrollo del software. Este marco de referencia se convertirá en un instrumento de apoyo para una mejor comprensión en la aplicación posterior.

Se expondrá un conjunto de conceptos y definiciones que se tomara en cuenta para lograr los objetivos propuestos para el desarrollo del proyecto. En el presente proyecto utilizará la metodología de Modelado y Desarrollo Orientado a aplicaciones hipertexto (OOHDM), para el desarrollo del software.

2.1. CONCEPTO GENERAL DE INVENTARIO

Inventario es un registro documental de los bienes y demás cosas pertenecientes a la empresa de una manera ordena y precisa.

Con el fin de registrar y controlar los inventarios, las empresas adoptan los sistemas pertinentes para valuar sus existencias de mercancías con el fin de fijar su posible volumen de producción y ventas

Comprender el concepto, características y los fundamentos de los sistemas de valoración de inventarios puede ser de gran utilidad para la empresa, ya que son estos lo que realmente fijan el punto de producción que se pueda tener en un periodo. El administrador financiero debe tener la información pertinente que le permita tomar decisiones sobre el manejo que se le debe dar a este rubro del activo organizacional. En el campo de la gestión empresarial, el inventario registra el conjunto de todos los bienes propios y disponibles para la venta a los clientes, considerados como activo corriente. Los bienes de una entidad empresarial que son objeto de inventario son las existencias que se destinan a la venta directa o aquellas destinadas internamente al proceso productivo como materias primas, productos inacabados, materiales de embalaje o envasado y piezas de recambio para mantenimiento que se consuman en el ciclo de operaciones.

2.1.1. FUNCION DE LOS INVENTARIOS

- ✓ Ayudar a la independencia de operaciones - Continuidad de las variaciones de demanda
- ✓ Determinar condiciones económicas de aprovisionamiento
- ✓ Determinar la óptima secuencia de operaciones
- ✓ Uso óptimo de la capacidad productiva

2.1.2. TIPOS DE MATERIAL O PRODUCTOS EN INVENTARIO

Los tipos de materiales que pueden ser registrados en inventario son los siguientes:

- Materias primas y partes compradas.
- Bienes parcialmente terminados, llamados: inventario en proceso.
- Inventario de bienes terminados (empresas de manufactura, comercializadoras).
- Partes de reemplazo, herramientas y consumibles.
- Bienes en tránsito a empresas o clientes.

2.1.3. INVENTARIO PERMANENTE

El sistema de inventarios permanente, o también llamado perpetuo, permite un control constante de los inventarios, al llevar el registro de cada unidad que ingresa y sale del inventario. Este control se lleva mediante **tarjetas llamada Kardex**, en donde se lleva el registro de cada unidad, su valor de compra, la fecha de adquisición, el valor de la salida de cada unidad y la fecha en que se retira del inventario. De esta forma, en todo momento se puede conocer el saldo exacto de los inventarios y el valor del costo de venta.

Además del control permanente de los inventarios, este sistema permite la determinación del costo al momento de hacer la venta, debido a que en cada salida de un producto se registra su cantidad y costo.

2.1.3.1. KARDEX

Es un auxiliar que nos ayuda a realizar un seguimiento de los movimientos de los inventarios y de los costos de mercancías en los almacenes. Estos costos se calculan mediante los métodos Promedio continuo, LIFO continuo y FIFO continuo utilizando la medida definida para el artículo.

2.2. METODOLOGIA.

2.2.1. OOHDM.

La Metodología OOHDM (Método de Diseño Hipermedia Orientado a Objeto, de sus siglas en inglés *Object Oriented Hypermedia Design Method*) propuesta por D. Schwabe y G. Rossi es una metodología de desarrollo de aplicaciones Web. En el dominio de la hipermedia hay requerimientos contradictorios que deben ser satisfechos en una estructura unificada.

Los principios básicos del método de OOHDM son:

- ❖ Obtención de Requerimientos. Es una de las etapas más importantes, en esta se identifican roles, y funciones en determinados procesos en los que se tendrá una relación con el sistema.
- ❖ Diseño conceptual. Viene a ser equivalente al diseño del modelo entidad - relación. En OOHDM se especifica mediante la notación UML.
- ❖ Diseño navegacional. OOHDM considera una aplicación web como vistas navegacionales sobre un modelo de objetos (el modelo desarrollado previamente en el diseño conceptual).
- ❖ Diseño de la interfaz abstracta. diseño de la interfaz de usuario atendiendo al diseño navegacional. Se expresa mediante ADV's la estructura que tendrán las distintas clases navegacionales.
- ❖ Implementación. se especifica el código de la aplicación a partir del diseño.

OOHDM es una mezcla de estilos de desarrollo basado en prototipos, en desarrollo interactivo y de desarrollo incremental. En cada fase se elabora un modelo orientado a objetos conceptual que recoge las características a resaltar en la misma incrementando los resultados de la fase o fases anteriores.

En la Siguiete Figura se grafican las cinco etapas de OOHDM. [SCHWABE, 1998]



Figura 2. 1. : Las cinco etapas de la Metodología OOHDM

Fuente: [SCHWABE, 1998]

2.2.1.1. OBTENCIÓN DE REQUERIMIENTOS

Como en todo proyecto informático la obtención de requerimientos es una de las etapas más importantes, la mayoría de los estudios entregan resultados claros, que los errores más caros son los que se cometen en esta etapa.

Para enfrentar esta dificultad OOHDM propone dividir esta etapa en cinco sub etapas: Identificación de roles y tareas, Especificación de escenarios, Especificación de Casos de Uso, Especificación de UIDs (Diseño de Interacción de Usuarios) y Validación de casos de uso y UIDs. [Souza, 2003]

▪ IDENTIFICACIÓN DE ROLES Y TAREAS

En esta sub etapa el analista deberá introducirse cuidadosamente en el dominio del sistema, ahora su principal labor será identificar los diferentes roles que podrían cumplir cada uno de los potenciales usuarios de la aplicación.

Los usuarios juegan roles importantes en cada intercambio de información con el sistema. Como por ejemplo: Alumno, Potencial Alumno, Profesor, Agente de Ventas, Secretaria, Coordinador, etc.

Para efectos de validación de los casos de uso es muy importante tener identificado el rol de cada usuario, ya que serán ellos los que entregaran su conformidad con respecto al caso de uso en el que participan.

Luego para cada rol el analista deberá identificar las tareas que deberá soportar la aplicación, como por ejemplo para el rol estudiante: Buscar Información acerca de un curso, Buscar información acerca de un profesor u Obtener el material para el curso.

▪ **ESPECIFICACIÓN DE ESCENARIOS.**

Los Escenarios son descripciones narrativas de como la aplicación será utilizada. En esta sub etapa, cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea. A continuación a manera de ejemplo, en la figura se grafican dos escenarios. [Ivilain&Schwabe, 2002].

Especificación de un escenario			
Escenario	Cn	Usuario:	
Contexto:			
Objetivo:			
Acciones:			

Figura 2. 2.: Especificación de escenario
Fuente: [IVilain&Schwabe, 2002]

▪ **ESPECIFICACIÓN DE CASOS DE USO.**

Un caso de uso es una forma de utilizar la aplicación. Específicamente representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes. Es muy importante que el analista identifique cual es la información relevante en cada uno de ellos, para luego generar un caso de uso coherente. En la siguiente figura se grafica el caso de uso “Buscando un curso dado el tema”.

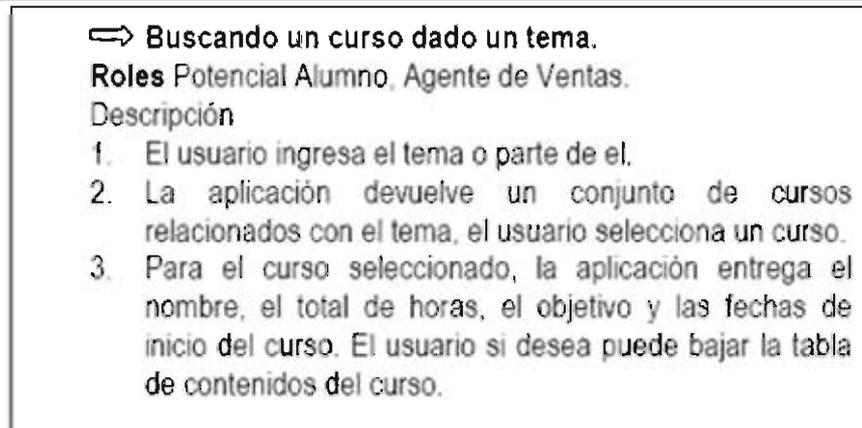


Figura 2. 2. : Caso de Uso “Buscando un curso dado un tema”
Fuente: [Soto, 2000]

▪ ESPECIFICACIÓN DE UIDS (DISEÑO DE INTERACCIÓN DE USUARIOS)

De acuerdo a UML, los diagramas de secuencia, de colaboración y de estado son capaces de representar un caso de uso. Sin embargo, la especificación de casos de usos usando estas técnicas es un amplio trabajo y puede anticiparse inesperadamente a tomar algunas decisiones de diseño. Para evitar esto OOHDM propone la utilización de una herramienta, llamada UID, que permite representar en forma rápida y sencilla los casos de uso generados en la etapa anterior.

Para obtener un UIDs desde un caso de uso, la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones. Identificar la información de intercambio es crucial ya que es la base para la definición de los UIDs.



Figura 2. 3. : UID correspondiente al caso de uso “Buscando un curso dado un tema”
Fuente: [Soto, 2000]

2.2.1.2. MODELOS CONCEPTUAL

Durante esta actividad, se construye un esquema conceptual que representa objetos, sus relaciones y colaboraciones que existen en el dominio designado. En aplicaciones de hipertexto convencionales, es decir, aquellos en los que los componentes de la hipertexto no serán modificados durante su ejecución, se podría usar un modelo semántico estructural, sin embargo, cuando la base de información puede cambiar dinámicamente o si se piensa realizar cálculos complejos o consultas en los objetos o el esquema, se necesita una abundante conducta del modelo orientado a objetos.

En OOADM, el esquema conceptual es construido en las clases, relaciones y sub-sistemas. Las clases son descritas como de costumbre en el modelo orientado a objetos, sin embargo, pueden multi-digitar atributos representando perspectivas diferentes de la misma entidad del mundo. Se usa una notación que es similar a UML, la Clase y Tarjetas de las relaciones, similar a las tarjetas de CRC son usadas como una ayuda de la documentación, ayudando remontar decisiones de diseño enviados y al revés.

En OOADM el esquema de la clase consiste en una colección de clases conectado por relaciones. Los objetos son instancias de clases, y de esa manera, cuando una relación se mantiene entre las clases. Las Clases se usarán después, durante el Diseño de navegación para derivar Nodos, y las relaciones que serán usadas para derivar Enlaces (Links).

Usando la conducta del modelo orientado a objeto para describir aspectos diferentes de una aplicación de hipertexto permite expresar una gran variedad de actividades de la informática como preguntas dinámicas a una base objeto, las modificaciones del objeto on-line, búsquedas basadas en heurísticas, etc. El tipo de funcionamiento requerido en el modelo conceptual depende en los aspectos deseados de la aplicación.

Cuando la aplicación se ejecute en un ambiente de soporte (distribuido) de objetos, las clases en el modelo conceptual serán implementadas directamente como están definidos, que especifican perspectivas múltiples como atributos separados. Si no, ellos servirán como especificaciones de diseño para el de navegación y actividades de diseño de interfaz

En esta etapa se genera las clases, relaciones y cardinalidades que se definen de acuerdo a reglas aplicadas sobre los UIDs. Cabe destacar que gran parte de ellas provienen de las técnicas de normalización.

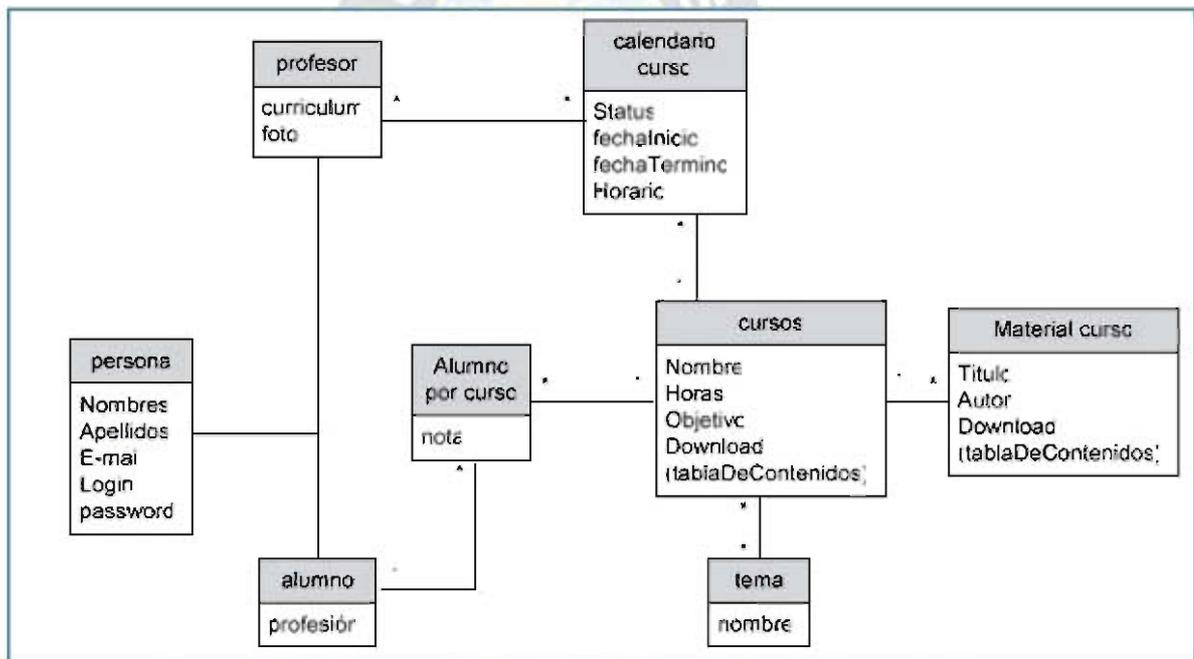


Figura 2.5.: Esquema conceptual resultante de los anteriores pasos.
Fuente: [Soto, 2000]

2.2.1.3. DISEÑO NAVEGACIONAL.

En OOHD, la navegación es considerada un paso crítico en el diseño de una aplicación de hipertexto. Un Modelo de navegación se construye como una vista más de un modelo conceptual y permite la construcción de modelos diferentes según los perfiles diferentes de los usuarios. Cada modelo de navegación proporciona una vista "Subjetiva" del modelo conceptual.

Mientras se diseña la estructura de navegación de una aplicación Web, se tiene en cuenta varios aspectos como: ¿Que objetos serán navegados, que atributos poseen, y que son las relaciones entre estos objetos y los mismos definidos en el esquema conceptual? Se hará esto definiendo nodos y enlaces (Links) como vistas orientadas a objetos de objetos conceptuales y relaciones. ¿Qué tipo de estructuras de composición existe entre los objetos de navegación y cómo son relacionados? ¿Cuál es la estructura fundamental de navegación? ¿En qué contexto el usuario navegará?.

Se introducirá el concepto de contextos de navegación, una arquitectura primitiva para organizar el espacio de la navegación. Se necesita decidir los objetos navegados que pueden parecer diferentes según el contexto en el que ellos son visitados, y se debe especificar esas diferencias claramente. ¿Cuales conexiones y estructuras de acceso existen entre objetos que serán navegados (enlaces, trayecto de búsqueda, camino o trayecto, índices, etc.)? ¿Cómo procede la navegación cuando el usuario salta "Jump" de un objeto a otro, es decir, lo que es el efecto de navegación en la fuente "source" y en el destino "target object" y posiblemente en otro objeto relacionado también?

El diseño de navegación se expresa en dos esquemas, el esquema de la Clase de navegación, y el Esquema del Contexto de navegación. En OOHDM, hay un juego de tipos pre-definidos de clases de navegación: nodos, links o enlaces, y estructuras de acceso.

- ❖ **Nodos:** Los nodos son contenedores básicos de información de las aplicaciones hipermedia. Se definen como vistas orientadas a objeto de las clases definidas durante el diseño conceptual usando un lenguaje basado en query, permitiendo así que un nodo sea definido mediante la combinación de atributos de clases diferentes relacionadas en el modelo de diseño conceptual. Los nodos contendrán tanto atributos de tipos básicos (donde se pueden encontrar tipos como imágenes o sonidos) y enlaces.

- ❖ **Enlaces:** Los enlaces reflejan la relación de navegación que puede explorar el usuario. Ya se sabe que para un mismo esquema conceptual puede haber diferentes esquemas navegacionales y los enlaces van a ser imprescindibles para poder crear esas vistas diferentes. Las clases enlaces sirven para especificar los atributos de enlaces y estos a su vez para representar enlaces entre clases nodos o incluso entre otros enlaces. En cualquier caso, el enlace puede actuar como un objeto intermedio en un proceso de navegación o como un puente de conexión entre dos nodos.
- ❖ **Estructuras de Acceso:** Las estructuras de acceso actúan como índices o diccionarios que permiten al usuario encontrar de forma rápida y eficiente la información deseada. Los menús, los índices o las guías de ruta son ejemplos de estas estructuras. Las estructuras de acceso también se modelan como clases, compuestas por un conjunto de referencias a objetos que son accesibles desde ella y una serie de criterios de clasificación de las mismas.
- ❖ **Contexto Navegacional:** Para diseñar bien una aplicación hipertexto, hay que prever los caminos que el usuario puede seguir, así es como únicamente se podrá evitar información redundante o que el usuario se pierda en la navegación. En OOHD un contexto navegacional está compuesto por un conjunto de nodos, de enlaces de clases de contexto y de otros contextos navegacionales. Estos son introducidos desde clases de navegación (enlaces, nodos o estructuras de acceso), pudiendo ser definidas por extensión o de forma implícita.
- ❖ **Clase de Contexto:** Es otra clase especial que sirve para complementar la definición de una clase de navegación. Por ejemplo, sirve para indicar qué información está accesible desde un enlace y desde dónde se puede llegar a él.

En OOHDM, la estructuración principal primitiva del espacio de navegación es el concepto de Contexto de Navegación. Un contexto de navegación es un conjunto de nodos, enlaces, contexto, clases y otros contextos de navegación (anidados).

1. **Clase basados en Objetos** - en este tipo de contexto pertenecen a la misma clase C y son seleccionados por dar una propiedad P, por el que debe satisfacerse a una propiedad todos los elementos: Contexto = {e | P (e), e ∈ C}. Un caso común es cuando incluye todas las instancias de una clase (P es idénticamente verdad).
2. **Clase basado en grupos** - Es un juego de contextos cada uno de los cuales son una clase simple basado en contextos. Es especificado para dar una propiedad del parámetro y permitiendo que el parámetro asuma todo los posibles valores (en un e numerable dominio finito).
3. **Enlaces basados en Objetos** - en este tipo de contexto son de la misma clase y son seleccionados cuando ellos pertenecen a la relación de 1 a n.

Note que un caso particular de este tipo es el contexto formado por todos los elementos que son parte de un objeto compuesto.

1. **Enlaces basados en grupo** - Es un juego de contextos cada uno de los cuales son un enlace basado en contexto. Es especificado dando una relación de 1 a n y formando el enlace basado en contextos para cada posible valor de la fuente de la relación.
2. **Enumerar** - En este tipo de contexto, se enumeran elementos explícitamente, y puede pertenecer a las clases diferentes. Además de sus elementos, hay otra dimensión a lo largo de contexto los cuales serán definidos, relativo a una sesión de la navegación. Si los elementos de un contexto pueden variar como una consecuencia de la navegación

por el usuario, se dice que el contexto es dinámico.

Los contextos navegacionales juegan un papel similar como colecciones y han sido inspirados por el concepto de contextos anidados. Los contextos navegacionales organiza el espacio de navegación en conjuntos consistentes que pueden ser atravesados siguiendo un orden particular, ellos deben ser definidos de la misma manera en lo que se refiere a la ayuda del usuario para realizar su tarea deseada.

Para cada UID se crearan diagramas de contexto y tarjetas de especificación que detallan la información contenida en el diagrama. En la siguiente figura se grafica el diagrama de contexto correspondiente al UID del caso de uso “Buscando un curso dado un tema”.

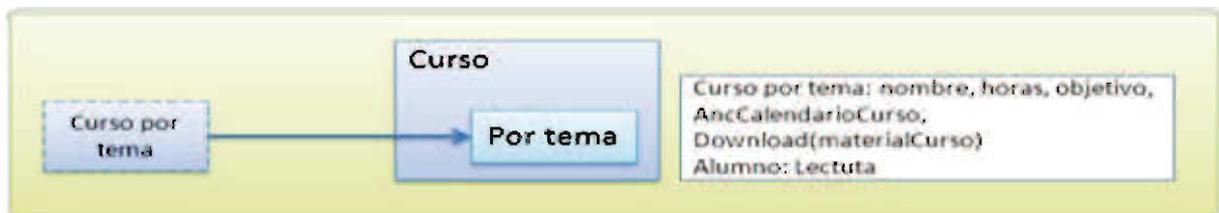


Figura 2.6.: Diagrama de contexto correspondiente al UID del caso de uso “Buscando un curso dado un tema”.

Fuente: [SCHWABE, 1998]

Aplicación del Diseño Navegacional.

Una vez que ya se han diseñado todos los diagramas de contexto, uno para cada caso de uso con sus respectivas tarjetas de especificación, es necesario realizar la unión de todos los diagramas para formar uno solo. El diagrama resultante corresponderá al diagrama de contexto de toda la aplicación. La figura siguiente ilustra el diagrama resultante de la unión de todos los diagramas de contexto obtenidos.

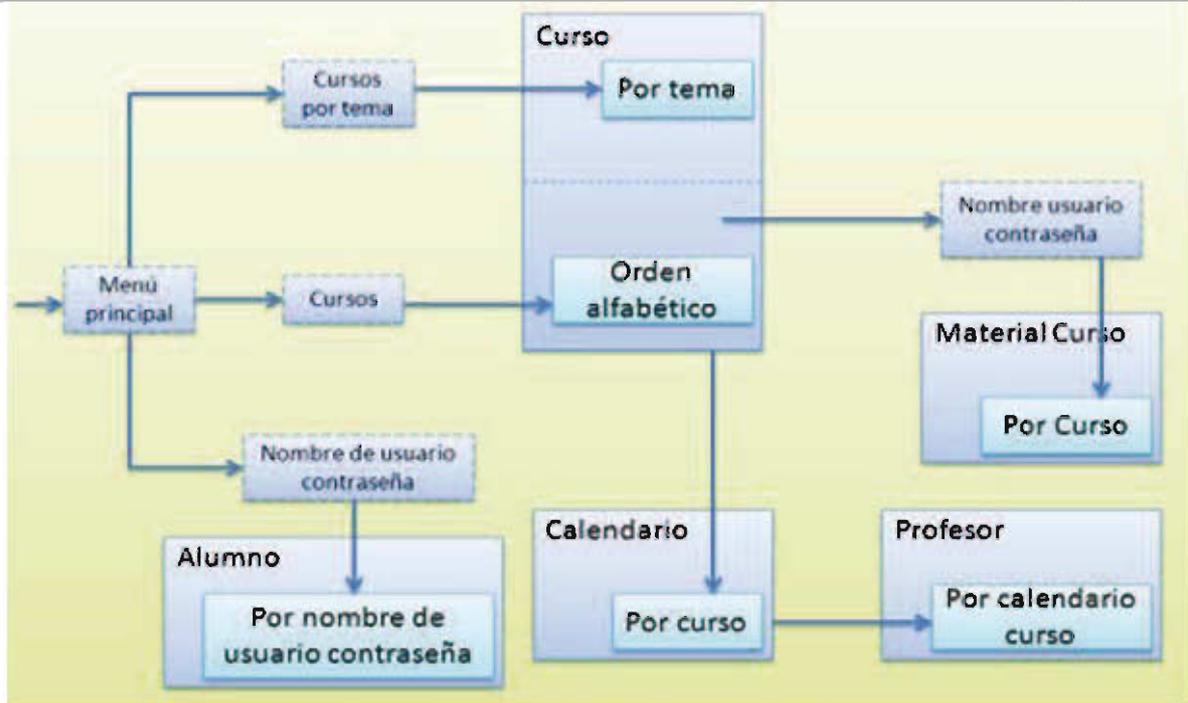


Figura 2.7.: Diagrama de contexto final.

Fuente: [SCHWABE, 1998]

Esquema de Clase Navegacionales.

El diseño navegacional en OOHDH corresponde a un conjunto de modelos que se van desarrollando paso a paso, ya se ha desarrollado el diagrama de contexto con sus respectivas tarjetas de especificación. En la siguiente tarea corresponde desarrollar el esquema de clases navegacionales, este modelo corresponde a una combinación entre el modelo conceptual y el diagrama de contexto, donde las clases navegacionales son llamadas nodos, las relaciones navegacionales se llaman vínculos y los atributos de los nodos que activan navegaciones son llamados anclas.

2.2.1.4. DISEÑO DE LA INTERFAZ ABSTRACTA.

En OOHDH, se usa un acercamiento del Diseño de Datos de Vista Abstractos (ADVs), para describir la interfaz del usuario de una aplicación de hipertexto. Los ADVs son objetos en los que tienen un estado y una interfaz, donde la interfaz

puede ser ejercido a través de mensajes (en particular, eventos externos generados por el usuario). Las ADVs son abstractas en el sentido de que ellos sólo representan la interfaz y el estado, y no la aplicación. Las ADVs han sido usados para representar interfaces entre dos medios de comunicación diferentes como un usuario, una red o un dispositivo (un cronómetro, por ejemplo) o como una interfaz entre dos u mas Objetos de Datos Abstractos (ADOs).

Un ADV usado en el diseño de aplicaciones Web puede verse como un objeto de interfaz. Comprende un conjunto de atributos (y objetos de interfaz anidado) que define sus propiedades de percepción, y el conjunto de eventos que puede manejar, como eventos generados por el usuario. En general, los ADVs especifican la organización y el comportamiento de la interfaz, pero la apariencia física real o de los atributos, y el diseño de la ADV en la pantalla real se hace en la fase de la implementación.

En el contexto de OOHDM, los objetos de navegación como nodos, e índices actuarán como ADOs, y su ADVs asociados se usará para especificar su apariencia al usuario. A continuación se usará el término ADV para referirse a clases de interfaz y objetos.

La manera en que se estructuran objetos de la interfaz usando agregación y generalización o especialización como mecanismos de abstracción. ADVs expresan la estructura del esquema estático que lleva a cabo la metáfora de la interfaz. Las ADVs permiten definir la apariencia de la interfaz de objetos de navegación y otros objetos de la interfaz útiles (como barras del menú, botones y menús).

Para lograr esto se utilizaran ADVs, es necesario aclarar que las ADVs representan estados o interfaces y no la implementación propiamente tal. En la siguiente figura se visualiza la ADV de “Curso por tema”.

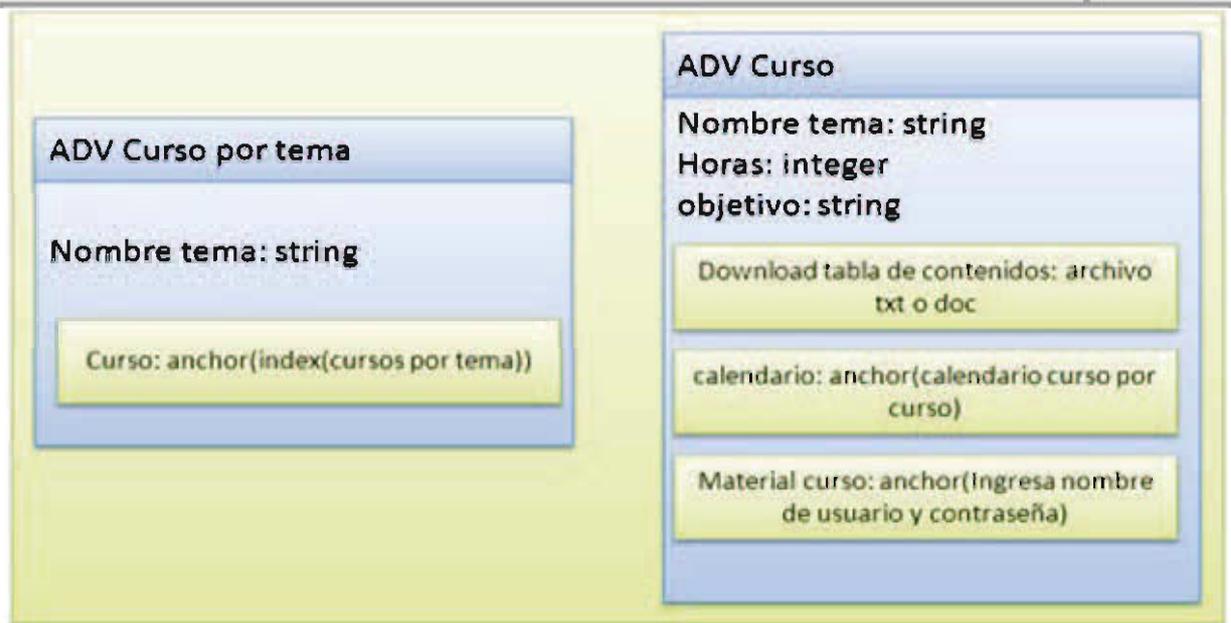


Figura 2.4.: ADVs relacionadas con el caso de uso "Buscando un curso dado un tema"
Fuente: [SCHWABE, 1998]

2.2.1.5. IMPLEMENTACIÓN.

En esta fase, el diseñador realmente implementará el diseño. Hasta ahora, todos los modelos fueron deliberadamente contruidos de semejante manera en lo que se refiere a ser independiente de la plataforma de implementación; en esta fase el ambiente particular de (tiempo de ejecución) runtime se toma el derecho de acceso a un servidor o a la red Internet. A continuación se fijará cómo los diseños de OOHDM pueden ser implementados en el WWW. Cuando la fase de implementación se alcanza, el diseñador ya tiene definido los artículos de información que son parte del dominio del problema. También tiene identificado cómo estos artículos deben ser organizados según el perfil del Usuario y asignaciones; ya que se ha decidido lo que la interfaz se parecerá, y cómo se comportará. En orden para implementar todos esto en el ambiente de WWW y aplicaciones de multimedia, el diseñador tiene que decidir cómo los artículos de información (ambos conceptual y objeto de navegación) será almacenada. También debe decidir cómo se comprenderán la apariencia de la interfaz y el comportamiento serán realizados usando HTML y posiblemente use algunas extensiones.

2.2.1.6. COMPARATIVA

Una vez enunciadas las propuestas, se van a presentar una serie de estudios comparativos de las mismas. Los estudios que se presentan se basan en tres aspectos. El primero analiza qué requisitos son cubiertos en cada metodología. El segundo estudio presenta las fases dentro del proceso de tratamiento de requisitos que cada una afronta y las técnicas que para ello proponen. El último estudio, evalúa como de detallada es cada propuesta a la hora de definir su proceso de desarrollo, las técnicas que aplica y los resultados que propone obtener. Por último, se presentan algunos otros aspectos que han surgido durante la realización de las comparativas.

2.2.1.7. REQUISITOS TRATADOS

En base a la clasificación de requisitos, la primera comparativa que se va a realizar de las propuestas estudiadas consiste en ver qué tipos de requisitos contempla cada propuesta. En la tabla 1 se presentan los diferentes requisitos y se indica cuáles de ellos son tratados en cada metodología.

Esta idea de separación de conceptos se observó desde el principio en las propuestas para la web pero solo se planteaba en fases avanzadas del proceso de desarrollo, principalmente en diseño. Se puede observar la tendencia actual de partir de esta separación de concepto ya en la fase de especificación de requisitos.

	Req. datos	Req. interfaz al usuario	Req. navegacionales	Req. personalización	Req. transaccionales	Req. no funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
UWA	✓	✓	✓	✓	✓	
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

Tabla 2.1. Requisitos tratados
 Fuente: [M.J. Escalona y N. Koch, 2001, Pág. 9]

2.2.1.8. GRADO DE DETALLE EN LAS DESCRIPCIONES

El siguiente estudio que se presenta se basa en la forma en que definen las metodologías el tratamiento de requisitos.

Para ello fueron comparadas las descripciones de las propuestas. La definición de algunas propuestas se basan principalmente en describir de forma concreta el proceso de desarrollo, sin llegar a concretar qué técnicas aplicar o la estructura de los resultados que se deben obtener. De esta forma, se van a evaluar las propuestas analizando:

- ✓ Si son *orientadas al proceso* o lo que es lo mismo, es si describen un proceso claro y concreto a seguir para realizar la definición, captura y validación de requisitos. Si son *orientadas a la técnica* o dicho de otro modo si describen técnicas concretas a aplicar durante el proceso.
- ✓ Si son *orientadas al resultado*, es decir, si da una descripción concreta del producto que se debe obtener tras el proceso de desarrollo. Evaluando la definición de las propuestas, se han catalogado cada una de ellas en base a lo detalladas que son a la hora de describir el proceso de definición de requisitos, las técnicas a aplicar o los resultados a obtener. Los resultados de esta evaluación se pueden ver en la tabla 1. Hay que tener en cuenta que la valoración se hace solo con respecto a la fase de ingeniería de los requisitos.

El peso dado a cada técnica se corresponde con la siguiente evaluación:

- ✓ Orientación al proceso: describe claramente los pasos a seguir (+), describe el proceso sin detallarlo (o), no describe ningún proceso (-).
- ✓ Orientación a la técnica: describe claramente las técnicas y la forma de aplicarlas (+), enumera las técnicas a aplicar (o), no propone ninguna técnica concreta o referencia a técnicas generales (-).

- ✓ Orientación al producto: describe claramente la estructura del producto a obtener (+), describe el contenido del producto sin entrar en detalle de su estructura (o), no comenta nada sobre el producto resultante (-).

	Orientación al proceso	Orientación a la técnica	Orientación al resultado
WSDM	o	-	-
SOHDM	-	+	-
RNA	÷	-	-
HFPM	+	o	+
OOHDM	o	+	-
UWE	÷	o	o
W2000	o	o	-
UWA	+	o	-
NDT	o	+	+
DDDP	÷	o	-

Tabla 2.2. Orientación de las propuestas
Fuente: [M.J. Escalona y N. Koch, 2001, Pág. 10].

2.3. HERRAMIENTAS DE DESARROLLO

2.3.1. UML

“UML” son las siglas de Unified Modeling Language (Lenguaje Unificado de Construcción de Modelos). Esta herramienta nos permitirá que el sistema propuesto genere diseños que capturen las ideas en una forma convencional y fácil de comprender.

UML se define como un “lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software” [Larman, 1999, Pág. 4]. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientado a objetos.

El UML es un estándar para construir modelos orientados a objetos. Nació en 1994 por iniciativa de Grady Booch y Jim Rumbaugh para combinar sus dos famosos métodos: el de Booch y el OMT (Técnica de Modelado de Objetos). Más tarde se les unió Ivar Jacobson creador del método OOSE (Ingeniería de

Software Orientada a Objetos). En respuesta a una petición de OMG para definir un lenguaje y una notación estándar del lenguaje de construcción de modelos, en 1997 se propone el UML.

2.3.2. DIAGRAMAS DE UML

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La explicación de la arquitectura se basa en los diagramas, en lugar de en vistas o anotaciones, ya que son estos la esencia de UML.

Las vistas existentes en UML son:

- Vista casos de uso: Se forma con los diagramas de casos de uso, colaboración, estado y actividades.
- Vista de diseño: se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- Vista de procesos: Se forma con los diagramas de la vista de diseño recalcando las clases y objetos referentes al proceso.
- Vista de implementación: Se forma con los diagramas de componentes, colaboración de estados y actividades.
- Vista de despliegue: se forma con los diagramas de despliegue, interacción, estados y actividades.

Se dispone de de dos tipos de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica.

- Diagramas de clase: Describe los tipos de objeto que hay en un sistema y las diversas clases de relaciones estáticas que existen entre ellos. [Fowler, 2000, Pág. 5]

- Diagramas de objeto: Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas se da una visión de casos reales.
- Diagramas de componentes: Muestran la organización de los componentes del sistema, un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- Diagrama de despliegue: muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema sirve como resumen o índice.
- Diagrama de casos de uso: Una iteración típica entre un usuario y un sistema de cómputo, muestra los actores y sus relaciones existentes entre acciones, importantes para modelar y organizar el comportamiento del sistema.

Los diagramas dinámicos son:

- Diagramas de secuencia: diagrama de colaboración, muestra a los referentes objetos y las relaciones que pueden tener entre ellos los mensajes que se envíen entre ellos. Son dos diagramas diferentes que se pueden pasar de uno a otro sin pérdida de información, pero que dan puntos de vista diferentes del sistema.
- Diagrama de estados: Describen el comportamiento de un sistema, describiendo todos los estados posibles en los que se puede entrar un objeto particular y la manera en que cambia el estado del objeto, como resultado de los eventos que llegan a él.
- Diagrama de actividades: muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre ellos.

2.3.3. CASOS DE USO

Es una estructura que ayudara a los analistas a trabajar con los usuarios para determinar la forma en que se usara un sistema. Con una colección de casos de uso se puede hacer el bosquejo de un sistema en términos de lo que los usuarios intenten hacer con él.

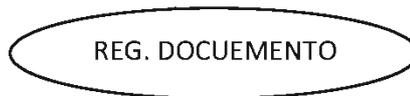


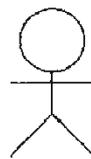
Figura 2.9. Icono del UML representa un Caso de uso
Fuente: [LARMAN, 1999, Pág. 49]

Un método con se identifican los casos de uso se basa en los actores:

- Se identifican los actores relacionados con un sistema.
- Cada actor. Se identifican los procesos que inician o en que participan

2.3.3.1. ACTORES

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana (ver figura 2.2.) sirve tanto para actores que son personas como para otro tipo de actores.



Cliente

Figura 2.10. Representación UML de Actor
Fuente: [Larman, 1999, Pág. 52]

2.3.3.2. FORMATO DE CASOS DE USO

Los casos de uso poden ser expresados en dos formatos:

2.3.3.2.1. CASOS DE USO DE ALTO NIVEL.

Hace referencia a los principales procesos globales. LARMAN sugiere la siguiente estructura.

Caso de Uso	Nombre del caso de uso
Actores:	Lista de actores (agentes externos), en la cual se indica quien inicia el caso de uso
Tipo:	Primario, secundario u opcional
Descripción:	Resumen del proceso concerniente al caso de uso

Tabla 2.3. Estructura de caso de uso de alto nivel
Fuente: [Larman, 1999, Pág. 65]

2.3.3.2.2. CASOS DE USO CON FORMATO EXPANDIDO.

Este tipo de casos permite alcanzar un conocimiento más profundo del proceso y requerimientos que se está tratando, Larman propone la siguiente estructura:

Casos de Uso	Nombre del caso de uso
Actores:	Lista de actores (agentes externos), en la cual se indica quien inicia el caso de uso
Propósito:	Se describe la intención del caso de uso
Resumen:	Repetición del caso de uso de alto nivel o alguna síntesis.
Tipo:	1. Primario, secundario u opcional
Referencia cruzada:	Casos relacionados de uso y funciones también relacionadas del sistema

Tabla 2.4. Caso de uso formato expandido.
Fuente: [Larman, 1999, Pág. 67]

2.3.3.3. DIAGRAMAS DE LOS CASOS DE USO.

Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre estos y los casos de uso. Estos últimos se muestran en óvalos y los actores son figuras utilizadas.

En el diagrama se dispone diferentes figuras que pueden mantener diversas relaciones entre ellas:

Casos de uso: representado por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad los casos de uso pueden tener relación con otros casos de uso sus relaciones son:

- Include: Representado por una flecha en el diagrama de la Figura 2.3 podemos ver como un caso de uso1, incluye al caso de uso 2.
- Extends: Una relación de un caso de uso2 hacia el caso de uso3

Generalización: es un tipo de relación de herencia.

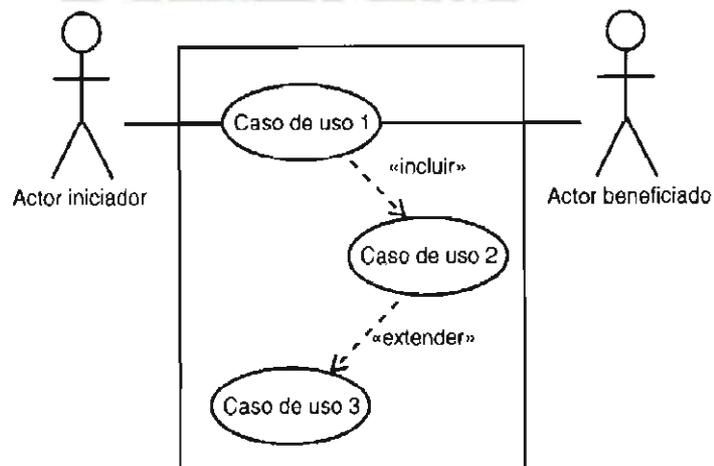


Figura 2.11. Diagrama de caso de uso
Fuente: [Schmuller, 2003, Pág. 67]

2.3.3.4. ELEMENTOS ESTRUCTURALES.

Las clases, objetos actores y casos de uso son cinco de los electos estructurales en el UML aunque tiene dientes diferencias son similares en el sentido de que representan partes ya sea físicas o conceptuales de un modelo. Veamos otros elementos estructurales:

Relaciones: la asociación generalización y dependencia y realización, son las relaciones en UML, conectan a tales elementos y de ese modo conectan los modelos con la realidad.

Agrupamiento: permite organizar los elementos estructurales de un modelo. Un paquete puede contener cualquier tipo de electo estructural.

Anotación: permite adjuntar restricciones, comentarios requerimientos y gráficos a sus modelos.

2.3.4. TRANSFORMACIÓN DE UN DIAGRAMA DE CLASES A UN MODELO ENTIDAD RELACION COMO EXTENSIÓN DE UML.

El diagrama de clases presenta un mecanismo de implementación neutral para modelar los aspectos de almacenado de datos del sistema. Las clases, sus atributos y relaciones pueden ser implementados directamente en una base de datos orientada a objetos. Aun así, en el entorno de desarrollo para el presente, será el método que se utiliza para el almacenamiento de datos.

El diagrama de clase se puede usar para modelar la estructura lógica de la base de datos.

Independientemente de si es orientada a objetos o relacional, con clases representando tablas, y atributos de la clase, representando columnas.

Si una base de datos relacional es el método de implementación escogido,

entonces el diagrama de clase puede ser una forma de un diagrama de entidad relación de entidad lógica.

Las clases y sus atributos (Diagrama de Clases) hacen referencia directamente a las entidades y sus atributos (Diagrama Entidad Relación).

Al implementar el diseño de las clases la estrategia está encaminada a hacer referencia al diagrama entidad relación, y es así que los creadores de UML plantean una comparación para lograr una transformación de un modelo de objetos en diagramas entidad relación. [LARMAN, 1999] [Joel, 2006].

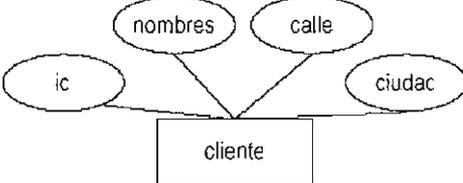
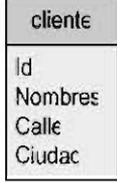
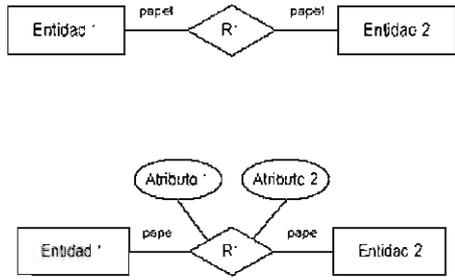
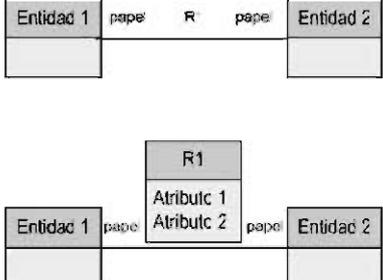
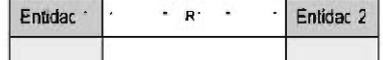
Elemento	Entidad Relación	Clases
Entidades y Atributos		
Relaciones		
Cardinalidad entre entidades		

Figura 2.12.: Objetos en diagramas entidad relación.
 Fuente: [LARMAN, 1999]

2.3.5. DIAGRAMA DE CLASE.

Forma parte de la vista estática del sistema. En el diagrama de clases se definen las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización.

La Clase.

Una clase está representada por un rectángulo que dispone de tres apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos. Cada clase debe tener un número único que las diferencia de las otras.

Atributo.

Representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Los atributos pueden representarse solo mostrando su nombre, y su tipo e incluso su valor por defecto.

2.3.5.1. RELACION ENTRE CLASES.

Existen tres relaciones diferentes entre clases, dependencias, generalización y asociación. En las relaciones se habla de una clase destino y de una clase origen.

Dependencias

Es una relación de uso, es decir una clase usa a otra que la necesita para su cometido. Se representa con una flecha discontinua va desde la clase utilizadora a la clase utilizada. Aunque las dependencias se pueden crear tal cual, es decir sin ningún estereotipo UML permite dar más significado a las dependencias. Es decir concretar más, mediante el uso de estereotipos.

Generalización.

Pues la herencia, donde se lleve una o varias clases padre, también llamada superclase o madre, y una clase hija o subclase. UML soporta tanto herencia simple como herencia múltiple.

Asociación.

Especifica que los objetos de una clase están relacionados con los elementos de otra clase. Se representa mediante una línea continua, que una las dos clases. Se pueden indicar el nombre, multiplicidad en los extremos, su rol y agregación.

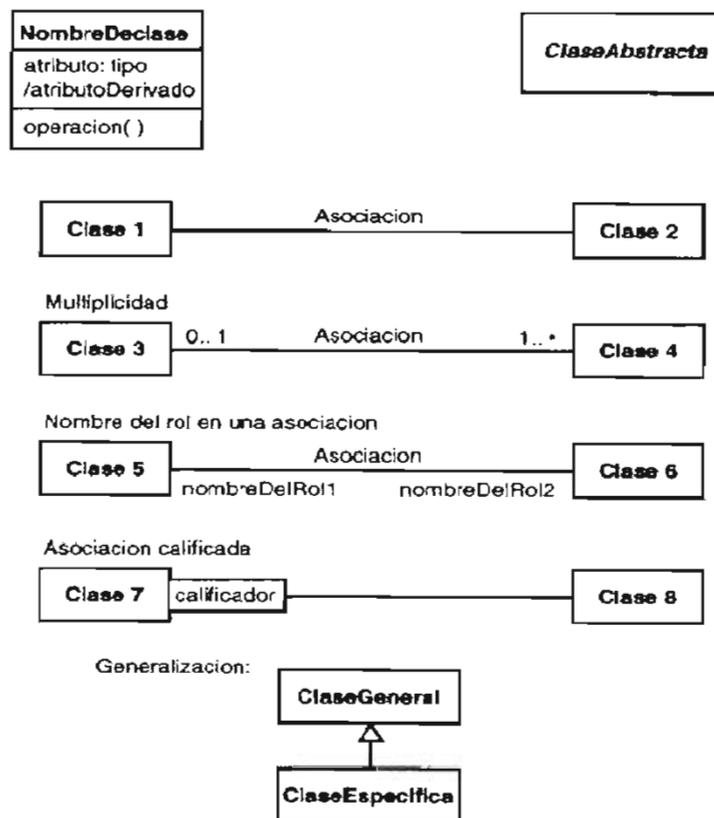


Figura 2.13. Diagrama de clase.
Fuente: [Schmuller, 2003, Pág. 75]

2.3.6. DIAGRAMAS DE SECUENCIA.

El diagrama de secuencias consta de objetos que se representan de un modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Objeto.

Se colocan cerca de la parte superior del diagrama de izquierda a derecha de manera que simplifiquen al diagrama. La extensión que este debajo será una línea discontinua conocida como línea de vida de un objeto junto a ella se encuentra el rectángulo de activación.

Mensaje.

Un mensaje vade un objeto a otro pasa por la línea de vida. Un objeto puede enviarse un mensaje así mismo, un mensaje puede ser simple, sincrónico, o asincrónico. Un mensaje simple es la transferencia de control de un objeto a otro, sincrónico espera la respuesta a tal mensaje antes de continuar con su trabajo, asincrónico no espera la respuesta antes de continuar.

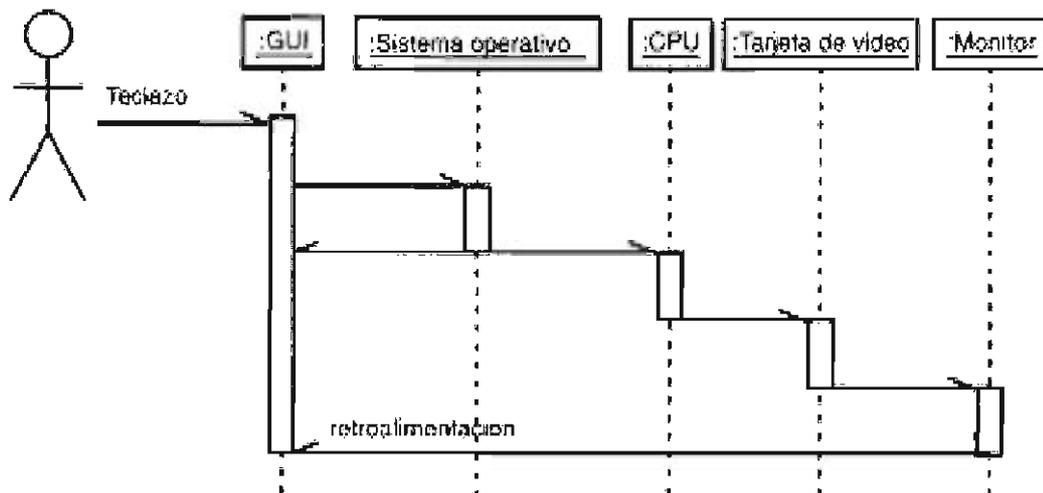


Figura 2.14. Diagrama de secuencia. Fuente:
[Shmuller, 2003, Pág. 103]

2.4. TECNOLOGIAS

2.4.1. WORLD WIDE WEB

La World Wide Web o simplemente WWW o Web es uno de los métodos más importantes de comunicación que existe en Internet. Consiste en un sistema de información basado en Hipertexto (texto que contiene enlaces a otras secciones del documento o a otros documentos). La información reside en forma de páginas Web en ordenadores que se denominan servidores Web y que forman los nodos de esta telaraña. Se denomina páginas Web a documentos que contienen elementos multimedia (imágenes, texto, audio, vídeo, etc.) además de enlaces de hipertexto.

http (*hypertext transfer protocol*): El protocolo http es el protocolo utilizado para la transferencia de páginas Web. Éste es el protocolo con el que se entienden cliente y servidor y por medio de él puede recibirse información de todo tipo (texto, gráficos, etc.),

html (*HyperText Markup Language*): Es un lenguaje de definición de páginas con extensiones hipertextuales portable a cualquier tipo de plataforma gráfica. Este lenguaje tiene la ventaja de ser muy sencillo y potente, pues permite combinar imágenes, textos y enlaces. La Web es en realidad una red de servidores dentro de Internet que ofrecen páginas hipertextuales en formato html.

URL (*Uniform Resource Locator*) : Es el mecanismo con el cual la Web asigna una dirección única a cada uno de los recursos de información localizado en cualquier lugar de Internet. Consiste en un código que contiene la identificación del servicio, la dirección del servidor (se puede especificar hasta el número de puerto), y, si es necesario, el directorio donde se encuentran los ficheros dentro del servidor. De esta forma se direccionan los recursos accesibles a través de un navegador Web. El formato general de una URL es el siguiente:

Protocolo o servicio: especifica el protocolo a utilizar para acceder al objeto. Puede ser uno de los siguientes:

telnet:// acceso remoto usando protocolo telnet;

tn3270:// acceso remoto usando protocolo telnet con emulación 3270;

ftp:// transferencia de ficheros;

gopher:// información en modo texto basado en menús;

news: grupos de interés;

file:// acceso a ficheros.

Cientes Web: Son los programas que se utilizan para 'navegar' por las páginas Web distribuidas por Internet. Los más conocidos son: Netscape, Internet Explorer y Mosaic. Son de muy fácil manejo y, además, suelen integrar programas que acceden a otros servicios como el correo electrónico o las news.

2.4.2. INTRANET.

Es una red privada dentro de una compañía u organización que utiliza el mismo tipo de software usado en el Internet público, pero que es sólo para uso interno, muchas veces este término se usa para referirse al componente más visible y popular, que es el sitio web interno. Las intranets se construyen sobre las mismas bases, protocolos y tecnologías sobre las que está construida Internet.

2.4.3. JAVASCRIPT

JavaScript es un lenguaje de programación utilizado para crear pequeños programitas. Encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. [Herrera, Javascript, 2006].

2.4.4. BASE DE DATOS.

Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna organización dada, esta sea independiente de tipo comercial, técnico, científico u otro. Por persistentes queremos decir, de manera intuitiva, que el tipo de datos de la base de datos difiere de otros datos más efímeros, como los datos de entrada, los datos de salida, las instrucciones de control, colas de trabajo, los bloques de control de software, los resultados intermedios y de manera más general cualquier dato que sea de naturaleza transitoria.

En forma más precisa decimos que los datos de la base de datos “persisten” debido a que en primer lugar una vez aceptados por el DBMS para entrar en la base de datos, en lo sucesivo solo pueden ser removidos de la base de datos por alguna solicitud explícita al DBMS, no como mero efecto lateral de algún programa que determina su ejecución. Por ello tanto es esta noción de persistencia a la que nos permitió dar una definición más precisa del término “base de datos”. [Date, C. J.; 2001].

2.4.5. GESTOR DE BASE DE DATOS.

Los Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la Base de datos y el usuario, las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan

este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

2.4.6. SQLSERVER.

SQL Server es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, y éstas a su vez están compuestas de filas y columnas. En el centro de SQL Server está el motor de SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente de conexiones y de sentencias SQL.

SQL Server incluye herramientas para la administración de los recursos que el ordenador nos proporciona y los gestiona para un mejor rendimiento de la base de datos.

Transact-SQL es el lenguaje que utiliza **SQL Server** para poder enviar peticiones tanto de consultas, inserciones, modificaciones, y de borrado a las tablas, así como otras peticiones que el usuario necesite sobre los datos. En definitiva, es un lenguaje que utiliza SQL Server para poder gestionar los datos que contienen las tablas.

El lenguaje estándar **SQL** (*Structured Query Language*) se emplea para los sistemas de bases de datos relacionales **RDBMS** (*Relational Database Management System*), es el estándar ANSI (*American National Standards Institute*). También es utilizado por otros sistemas como: Oracle, Access, Sybase, etc..

2.4.7. FLASH.

Flash es una tecnología para crear animaciones gráficas vectoriales independientes del navegador y que necesitan poco ancho de banda para

mostrarse en los sitios web. La animación en Flash se ve exactamente igual en todos los navegadores, un navegador sólo necesitan un plug-in para mostrar animaciones en Flash.

Con Flash los usuarios pueden dibujar sus propias animaciones o importar otras imágenes vectoriales. [Herrera, Flash, 2006]

2.4.8. SERVIDOR APACHE.

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para Plataformas Unix, Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. [Bakken, 2001].

2.4.8.1. CARACTERÍSTICAS DE APACHE.

- ✓ Módulos de autenticación: mod_access, mod_auth y mod_digest.
- ✓ Soporte para SSL y TLS.
- ✓ Permite la configuración de mensajes de errores personalizados y negociación de Contenido.
- ✓ Soporte para los lenguajes perl, python, tcl y PHP.
- ✓ Permite autenticación de base de datos basada en SGBD.

Apache es principalmente usado para servir páginas Web estáticas y dinámicas en la WWW. Apache es el servidor Web del popular sistema AppServer, junto con MySQL y los lenguajes de programación PHP/Perl/Python. [Bakken, 2001].

2.4.9. AJAX.

AJAX en realidad no es una tecnología en sí mismo. En realidad se trata de la reunión de varias tecnologías que se desarrollaran de forma autónoma y que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- ✓ XHTML y CSS, para crear una presentación basada en estándares.
- ✓ DOM para la interacción y Manipulación dinámica de la presentación.
- ✓ XML, XSLT y JSON, para el intercambio asíncrono de la información.
- ✓ JAVASCRIPT, para unir todas las demás tecnologías. [Perez, 2010].

2.5. TECNOLOGÍA CLIENTE SERVIDOR

2.5.1. DESDE UN PUNTO DE VISTA CONCEPTUAL

“Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información”

2.5.2. EN TERMINOS DE ARQUITECTURA

“Los distintos aspectos que caracterizan a una aplicación (proceso, almacenamiento, control y operaciones de entrada y salida de datos) en el sentido más amplio, están situados en más de un computador, los cuales se encuentran interconectados mediante una red de comunicaciones”.

2.5.3. DEFINICION DE IBM

“Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores”

2.5.4. ¿QUE ES UN ARQUITECTURA?

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos,

permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

Debemos señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento y, que la arquitectura Cliente/Servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran.

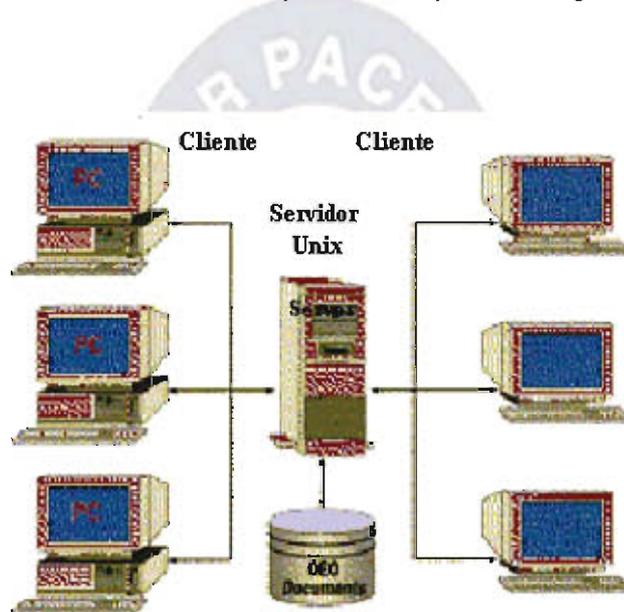


Figura 2.15. Arquitectura Cliente/Servidor.
Fuente: [<http://www.inei.gob.pe>]

2.5.5. CLIENTE.

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

2.5.6. SERVIDOR.

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax,

procesamiento de imágenes, etc.

2.5.7. PROCESO DISTRIBUIDO.

Es un modelo de sistemas y/o de aplicaciones, en el cual las funciones y los datos pueden estar distribuidos a través de múltiples recursos de cómputo, conectados en un ambiente de redes LAN o WAN.

2.5.8. ELEMENTOS DE LA ARQUITECTURA C/S

En esta aproximación, y con el objetivo de definir y delimitar el modelo de referencia de una arquitectura Cliente/Servidor, debemos identificar los componentes que permitan articular dicha arquitectura, considerando que toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

- Presentación/Captación de Información
- Procesos
- Almacenamiento de la Información

Los cuales se suelen distribuir tal como se presenta en la figura :

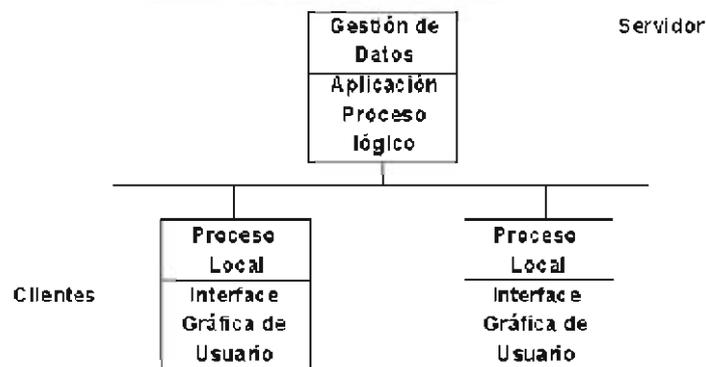


Figura 2.16. Aplicaciones Cliente/Servidor
Fuente: [<http://www.inei.gob.pe>]

Y se integran en una arquitectura Cliente/Servidor en base a los elementos que caracterizan dicha arquitectura, es decir:

- Puestos de Trabajo
- Comunicaciones
- Servidores

De estos elementos debemos destacar:

- El Puesto de Trabajo o Cliente “Una Estación de trabajo o microcomputador (PC: Computador Personal) conectado a una red, que le permite acceder y gestionar una serie de recursos” el cual se perfila como un puesto de trabajo universal. Nos referimos a un microcomputador conectado al sistema de información y en el que se realiza una parte mayoritaria de los procesos.

Se trata de un fenómeno en el sector informático. Aquellos responsables informáticos que se oponen a la utilización de los terminales no programables, acaban siendo marginados por la presión de los usuarios.

Debemos destacar que el puesto de trabajo basado en un microcomputador conectado a una red, favorece la flexibilidad y el dinamismo en las organizaciones. Entre otras razones, porque permite modificar la ubicación de los puestos de trabajo, dadas las ventajas de la red.

- Los Servidores o Back-end. “Una máquina que suministra una serie de servicios como Bases de Datos, Archivos, Comunicaciones,...”.

Los Servidores, según la especialización y los requerimientos de los servicios que debe suministrar pueden ser:

- Mainframes
- Miniordenadores
- Especializados (Dispositivos de Red, Imagen, etc.)

- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.
También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

2.5.10. VENTAJAS DEL MODELO C/S

- Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, podemos integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que

utilizar el mismo sistema operacional.

- Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos bajo este esquema tienen mayor interacción más intuitiva con el usuario. El uso de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets).
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

2.5.11. SERVICIOS BASADOS EN C/S

Dentro del modelo Cliente/Servidor, IBM ha identificado los siguientes servicios hacia los cuales está orientado sus esfuerzos de desarrollo de productos para satisfacerlo:

a) Servicios de Datos e Impresión:

Servicios que permiten compartir archivos, bases de datos, impresoras y plotters. Administración de las colas de impresión en diferentes dispositivos.

b) Servicios de Comunicaciones:

Aseguran que cada componente físico de la red sea capaz de comunicarse exitosamente con otros componentes, tales como LAN a LAN y LAN a WAN. El sistema puede incluir dispositivos de comunicaciones que manejen diferentes tipos de protocolos para conectar sistemas heterogéneos.

c) Servicio de Administración:

Administración de Sistemas involucra administración de cambios, de problemas, operaciones, configuración y rendimiento.

2.6. CALIDAD DE SOFTWARE

Todas las metodologías y herramientas tienen un único fin producir software de gran calidad: “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” [PRESSMAN, 2002]

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad.

El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

2.6.1. MODELO DE CALIDAD ESTABLECIDO POR EL ESTÁNDAR ISO 9126

La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de "Information technology –Software product evaluation: Quality characteristics and guidelines for their use", en el cual se establecen las características de calidad para productos de software. El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. La tabla 1 muestra la pregunta central que atiende cada una de estas características.

Características	Pregunta central
Funcionalidad	¿Las funciones y propiedades satisfacen las necesidades explícitas; esto es, el que...?
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
Usabilidad	¿El software es fácil de usar y de aprender?
Eficiencia	¿Es rápido y minimalista en cuanto al uso de
Mantenibilidad	¿Es fácil de modificar y verificar?
Portabilidad	¿Es fácil de transferir de un ambiente a otro?

Tabla 2. 5.: Características de ISO-9126 y aspecto que atiende cada una.
Fuente: [PRESSMAN, 2002]

2.6.2. CARACTERÍSTICAS PROPUESTAS POR LA ISO 9126

2.6.2.1. FUNCIONALIDAD

En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

Adecuación. Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.

Exactitud. Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.

Interoperabilidad. Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.

Conformidad. Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.

Seguridad. Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

2.6.2.2. CONFIABILIDAD

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

Las sub-características que el estándar sugiere son:

Nivel de Madurez. Permite medir la frecuencia de falla por errores en el software.

Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.

Recuperación. Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

2.6.2.3. USABILIDAD.

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

Comprensibilidad. Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.

Facilidad de Aprender. Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.

Operabilidad. Agrupa los conceptos que evalúan la operación y el control del sistema.

2.6.2.4. MANTENIBILIDAD

Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:

Capacidad de análisis. Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.

Capacidad de modificación. Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.

Estabilidad. Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.

Facilidad de Prueba. Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

2.6.2.5. PORTABILIDAD

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

Adaptabilidad. Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.

Facilidad de Instalación. Es el esfuerzo necesario para instalar el software en un ambiente determinado.

Conformidad. Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.

Capacidad de reemplazo. Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.



CAPITULO III

3. MARCO APLICATIVO

En el presente capítulo se realizara el análisis y diseño del sistema donde se aplicara métodos y herramientas mencionadas en el capítulo anterior.

3.1. DEFINICION DEL SISTEMA

El software Sistema Cliente Servidor De Inventario De Productos De Poliuretano realizará la organización, información, registro de entrada y salidade mercadería.

3.2. FUNCION DEL SISTEMA

El sistema tendrá operaciones básicas de adición, eliminación y actualización de registros de movimiento de mercadería con la aplicación de atención cliente – servidor para sucursales.

3.3. DESARROLLO DEL PROYECTO

La empresa INDUMAR S.R.L. cuenta con diferentes niveles jerárquicos donde varios intervienen para el desarrollo y movimiento de mercadería.

3.3.1. ESTRUCTURA ORGÁNICA

Para el entendimiento del presente proyecto de grado se muestra la estructura organizacional de la empresa **INDUMAR S.R.L.** presentada en la figura 3.1, y los recuadros resaltados nos permiten observar los departamentos para los cuales se desarrollara el proyecto.

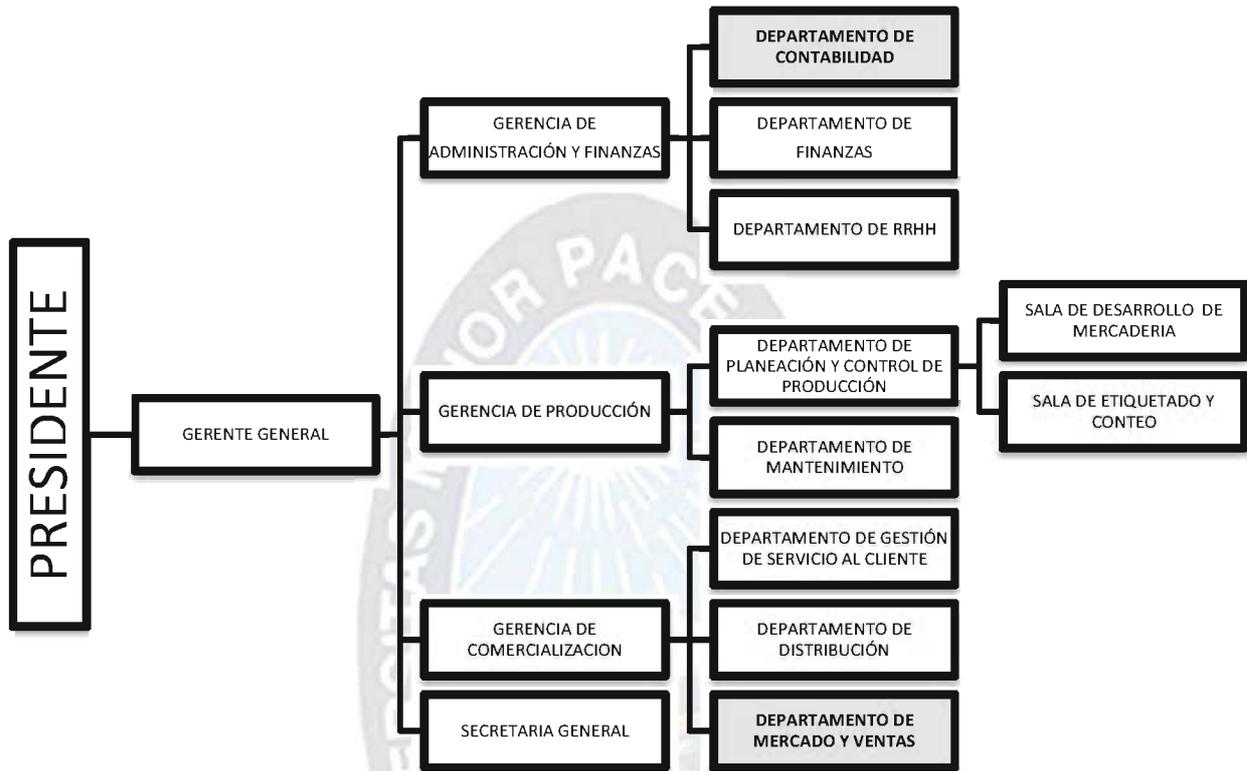


Figura 3.5. : Estructura Orgánica
Fuente:[Elaboración Propia]

3.3.2. DEPARTAMENTOS DE CONTABILIDAD Y MERCADO - VENTAS

La función de estos departamentos es planear, ejecutar y controlar las actividades de ventas, interactuando con los distribuidores y especialmente están obligados a poner especial cuidado con los costos. Además trabaja a través de otros departamentos como finanzas, personal, etc. Encargados de realizar el pedido de los productos terminados de acuerdo a la cantidad de ventas proyectadas en un tiempo estimado.

3.3.3. OBTENCION DE REQUERIMIENTOS

La tarea de Ingeniería de Requisitos es fundamental para que un sistema sea exitoso, en este sentido para la realización del presente proyecto se realizarón 3 actividades como se indica en la tabla 3.1, ahora se describiran cada una de ellas.

3.3.4. OBTENCIÓN DE REQUISITOS

Las tareas que se realizaron en esta fase son 3 las cuales se puede observar:

Entrevista	Entrevistas frecuentes con jefe del departamento de contabilidad.
Observación	Se observaron todos los procesos que realizan los diferentes miembros que están directamente relacionados con el ingreso y salida de mercadería.
Documentación	Fue posible tener copias del registro de ingreso y salida de mercadería.

Tabla 3.1.: Tareas en la Obtención de Requisitos
Fuente: [Elaboración Propia]

Podemos presentar los siguientes requerimientos que son fruto de lo anterior:

IDENTIFICADOR	REQUERIMIENTO
R1	Contar con un mecanismo de seguridad de acceso al sistema.
R2	Registro y actualización de usuarios.
R3	Registro y actualización de proveedores.
R4	Registrar las entradas de productos.
R5	Registrar las salidas de productos.
R6	Actualización del inventario de productos.
R7	Verificación rápida de los productos en almacén.

R8	Generar información actualizada sobre el stock de productos.
R9	Realizar reportes del movimiento en almacenes (entradas y salidas).
R10	Generar reportes (usuarios y proveedores)

Tabla 3.2.: Listado de requisitos
 Fuente: [Elaboración Propia]

3.3.5. IDENTIFICACIÓN DE ACTORES Y TAREAS

En esta etapa especificaremos a todos los Actores y tareas desempeñadas de los potenciales usuarios. Estableciendo las tareas específicas que desempeñara el sistema para cada actor presentado.

USUARIO:	Responsable de Sistemas
TAREA:	<ul style="list-style-type: none"> - Gestión de Usuarios - Administración Del Sistema

Tabla 3.3.: Identificador de tareas Responsable de Sistemas
 Fuente: [Elaboración Propia]

USUARIO:	Jefe del Departamento de Contabilidad
TAREA:	<ul style="list-style-type: none"> - Registra entrada de mercadería - Registra salida de mercadería - Registra solicitudes de productos - Control de inventario. - Registro de clientes. - Registro de proveedores. - Emisión de reportes

Tabla 3.4.: Identificador de tareas Jefe del Departamento de Contabilidad
 Fuente: [Elaboración Propia]

USUARIO:	Jefe del Almacenes
TAREA:	<ul style="list-style-type: none"> - Registra entrada de mercadería - Registra salida de mercadería - Emisión de reportes

Tabla 3.5.: Identificador de tareas Jefe de almacenes
 Fuente: [Elaboración Propia]

USUARIO:	Unidad Solicitante
TAREA:	<ul style="list-style-type: none">- Verifica estado de la mercadería- Registro de solicitud de productos

Tabla 3.6.: Identificador de tareas Unidad Solicitante
Fuente: [Elaboración Propia]

USUARIO:	Recepcionista de datos
TAREA:	<ul style="list-style-type: none">- Registrar a los clientes- Registrar a los proveedores- Asignar información básica sobre los productos.

Tabla 3.7.: Identificador de tareas Recepcionista de Datos
Fuente: [Elaboración Propia]

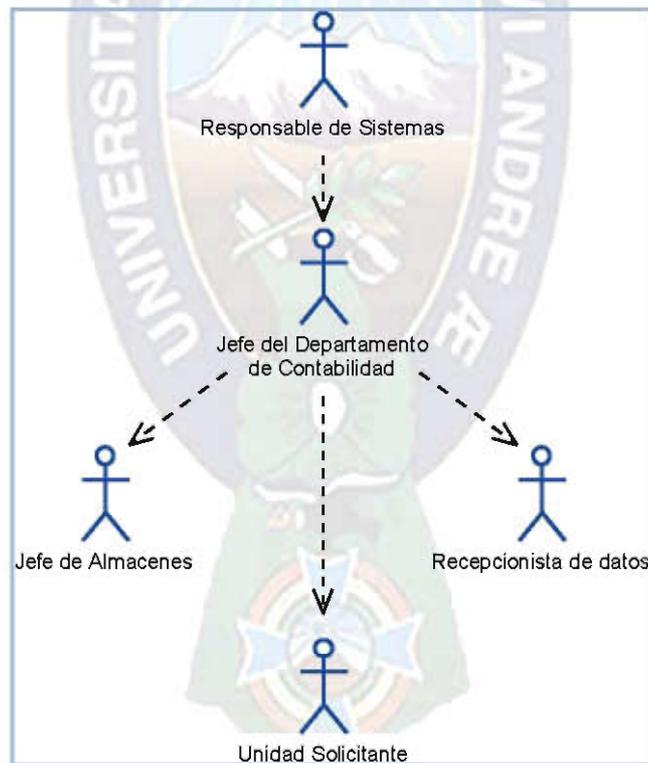


Figura 3.2.: Principales Actores
Fuente: [Elaboración Propia]

3.3.6. ESPECIFICACIÓN DE ESCENARIOS

Una vez que se han identificado a los actores principales, lo siguiente es especificar cómo será utilizada la aplicación por los actores los cuales utilizarán el Sistema. Los datos obtenidos se hicieron en base a entrevistas y conversaciones con cada uno de los usuarios. A continuación se muestran algunos de estos escenarios:

Escenario	E1	Administrar Usuarios	Usuario: Responsable de Sistemas
Contexto		Es el encargado de revisar, registrar a los usuarios del Sistema, de manera que en este, los usuarios estén siempre con los privilegios adecuados, ya que estos pueden ir cambiando de acuerdo a las necesidades de la capacitación.	
Objetivo		Mantener actualizado el registro y privilegios de los usuarios del Sistema	
Acciones		Cuando el administrador del sistema requiere modificar permisos de los usuarios el sistema le muestra un menú de la gestión de usuarios, le presenta una lista de todos los usuarios y su respectivo grupo, dando las opciones de Actualización de perfil, Eliminación o Agregación de un usuario a un grupo determinado	

Tabla 3.8.: Escenario Administrar Usuario
Fuente: [Elaboración Propia]

Escenario	E2	Registro de movimiento de mercadería.	Usuario: Jefe del Departamento de Contabilidad
Contexto	Es el encargado del registro de entradas y salidas de mercadería y también del registro de clientes y proveedores.		
Objetivo	Registrar de una manera correcta el movimiento de mercadería (entrada y salida).		
Acciones	Cuando el encargado del registro de entrada y salida de mercadería requiera registrar el movimiento de la mercadería el sistema le mostrará un menú para el registro de dicho movimiento.		

Tabla 3.9.: Escenario Jefe del Departamento de Contabilidad
 Fuente: [Elaboración Propia]

Escenario	E3	Registro de mercadería	Usuario: Jefe de Almacenes
Contexto	Es el encargado de verificar el stock de productos en almacén para poder solicitar la cantidad de productos faltantes para ello debe llenar correctamente datos del producto.		
Objetivo	Registrar de una manera correcta la mercadería y verificar el stock de la misma.		
Acciones	Cuando el encargado de Almacén requiera registrar la mercadería y verificar el stock de la misma, el sistema le mostrara un menú donde se identificará para poder elegir la opción que le corresponde para poder registrar de una manera correcta la mercadería y a su vez controlar el stock de la misma.		

Tabla 3.10.: Escenario Jefe de Almacenes
 Fuente: [Elaboración Propia]

Escenario	E4	Salida de mercadería	Usuario: Unidad Solicitante
Contexto	Es el encargado de verificar el estado de la mercadería y todas las solicitudes que existan para la salida de la misma.		
Objetivo	Controlar el estado de la mercadería y todas las solicitudes de la misma.		
Acciones	Cuando la unidad solicitante necesite un informe sobre el estado de la mercadería y su vez las solicitudes de la misma el encargado realizará los reportes correspondientes.		

Tabla 3.11.: Escenario Unidad Solicitante
 Fuente: [Elaboración Propia]

Escenario	E5	Registro de clientes y proveedores	Usuario: Recepcionista de Datos
Contexto	Es el encargado de registrar a todos los clientes e interesados en adquirir mercadería, así mismo registrar a los proveedores o actualizar la información de los mismos.		
Objetivo	Registrar o actualizar de una manera correcta los datos de clientes y proveedores nuevos y antiguos.		
Acciones	Cuando el recepcionista de datos requiera registrar a clientes y proveedores el sistema le mostrará un menú donde se identificará para elegir la opción que le corresponde para poder registrar y actualizar de una manera correcta a clientes y a proveedores.		

Tabla 3.12.: Escenario Recepcionista de Datos
 Fuente: [Elaboración Propia]

3.3.7. ESPECIFICACIÓN DE CASOS DE USO

En la figura 3.3 se presenta el diagrama de casos de uso de la aplicación, que muestra las funciones que desarrollan los usuarios, además de la información visible para cada uno de ellos.

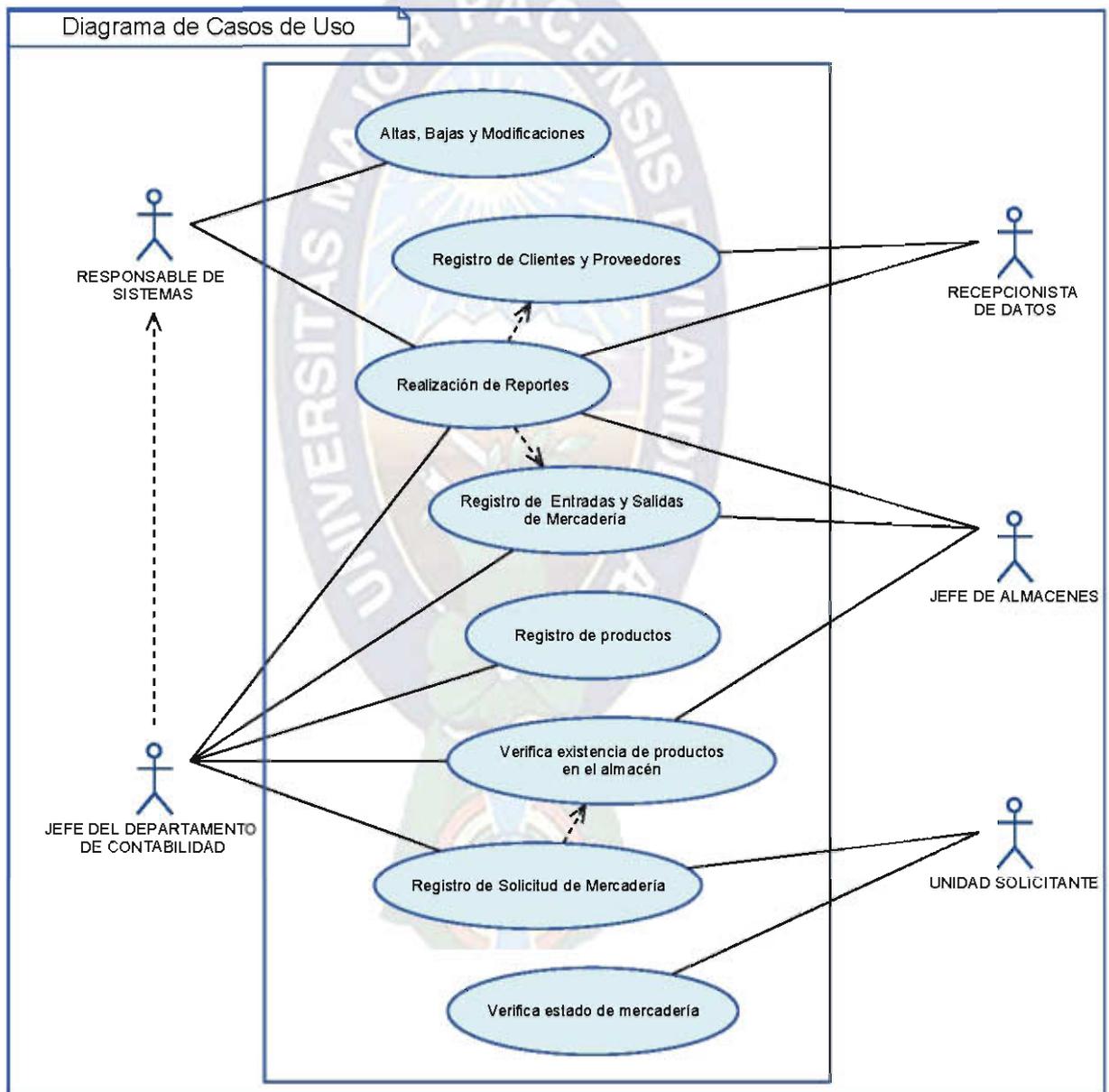


Figura 3.3.: Diagrama de Casos de Uso Principal
Fuente: [Elaboración Propia]

3.3.8. CASOS DE USO EXTENDIDO

Con la especificación de escenario, se analiza los casos de uso, y se comienza con la especificación de casos de uso extendidos como se realiza en las tablas (3.13), (3.14), (3.15), (3.16), (3.17).

▪ ALTAS, BAJAS Y MODIFICACIONES

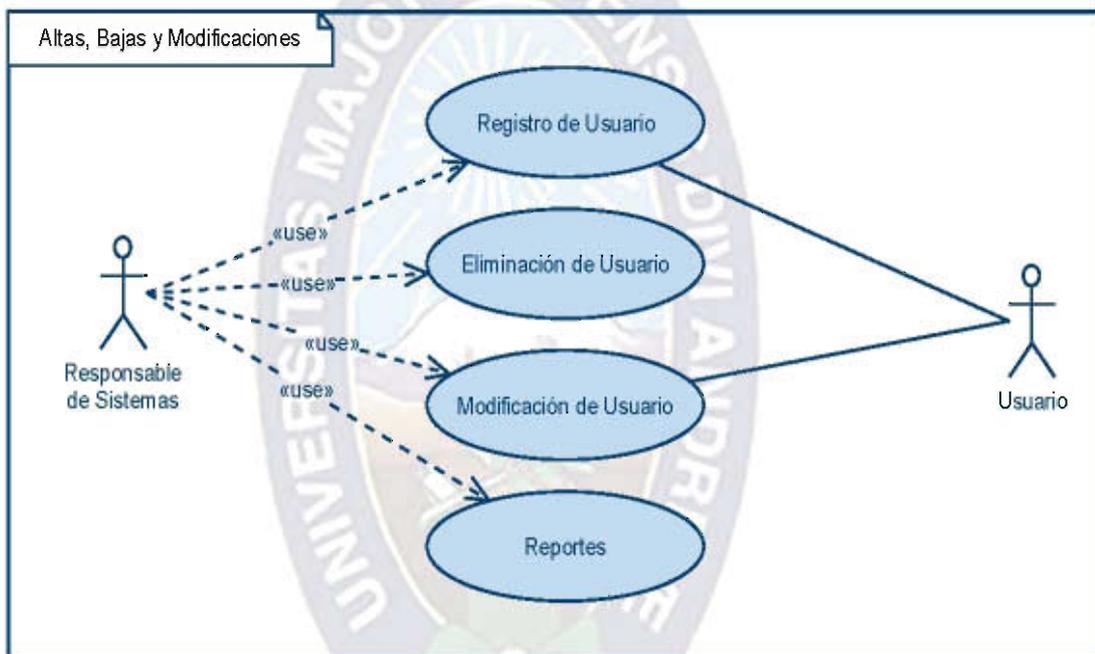


Figura 3.4.: Caso de Uso Altas, bajas y modificaciones
 Fuente: [Elaboración Propia]

Caso de Uso:	Altas, bajas y modificaciones
Actores:	Responsable de Sistemas, Usuario
Propósito:	Registrar usuarios Nuevos
Resumen:	Proceso por el cual es registrar al nuevo usuariodel sistema para que realice los registros delmovimiento de mercadería.
Tipo:	Primario
Curso Normal de Eventos	

Acción de los Actores	Respuesta del Sistema
<p>1.- Este caso de uso comienza cuando un nuevo usuario necesita acceder al sistema y verifica si es un nuevo usuario.</p> <p>2.- El Responsable de Sistemas atiende al usuario le pide sus datos para verificar y ejecutar la acción.</p> <p>4.- El Responsable de Sistemas procede a obtener datos del Usuario.</p> <p>5.-El Usuario procede a ir a su punto de trabajo para verificar y actuar en sus funciones.</p>	<p>3. Procesa la solicitud y efectúa el proceso y ve que no está registrado.</p> <p>6.-Procesa, efectúa y envía el mensaje de éxito de los registros nuevos y actualizados de los Usuarios a manejar el sistema.</p>

Tabla 3.13.: Altas, bajas y modificaciones
 Fuente: [Elaboración Propia]

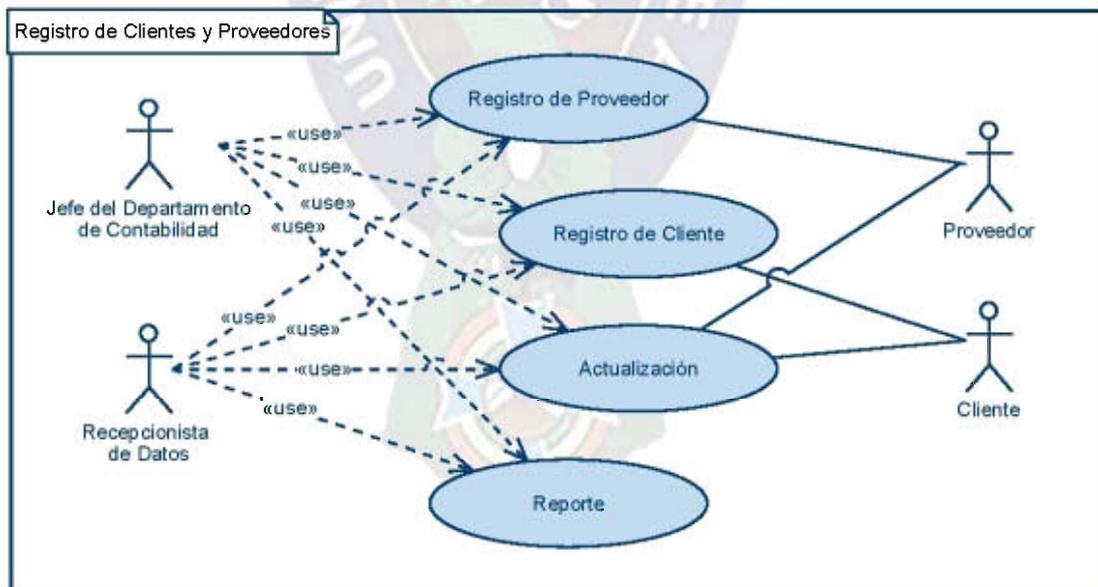


Figura 3.5.: Caso de Uso Registro de Clientes y Proveedores
 Fuente: [Elaboración Propia]

Caso de Uso:	Registro de Clientes y Proveedores
Actores:	Jefe del Departamento de Contabilidad, Recepcionista de Datos, Cliente, Proveedor.
Propósito:	Registrar y actualizar nuevos clientes y proveedores.
Resumen:	Proceso por el cual se registran y actualizan clientes y proveedores.
Tipo:	Primario
Curso Normal de Eventos	
Acción de los Actores	Respuesta del Sistema
<p>1.- Este caso de uso comienza cuando un nuevo cliente o proveedor necesita registrarse o actualizar su información para ser atendido en la empresa.</p> <p>2.- El Jefe del Departamento de Contabilidad o el Recepcionista de datos atiende al cliente o proveedor al cual le pide sus datos respectivos para el registro o actualización correspondiente y ejecuta la acción.</p> <p>4.- El Jefe del Departamento de Contabilidad o el Recepcionista de datos procede a la actualización de datos del cliente o proveedor.</p> <p>6.- El Jefe del Departamento de Contabilidad o el Recepcionista de datos procede a introducir todos los datos del nuevo cliente o proveedor.</p>	<p>3.- Procesa la solicitud y efectúa el proceso y verifica si está registrado o no, si lo está pasa al paso 4 y si no está pasa al paso 6.</p> <p>5.- El sistema dará toda la información del cliente o proveedor y actualizará la información.</p> <p>7.- El sistema actualizará la base de datos de clientes o proveedores y asignará un código de identificación.</p>

Tabla 3.14.: Registro de Clientes y Proveedores
 Fuente: [Elaboración Propia]

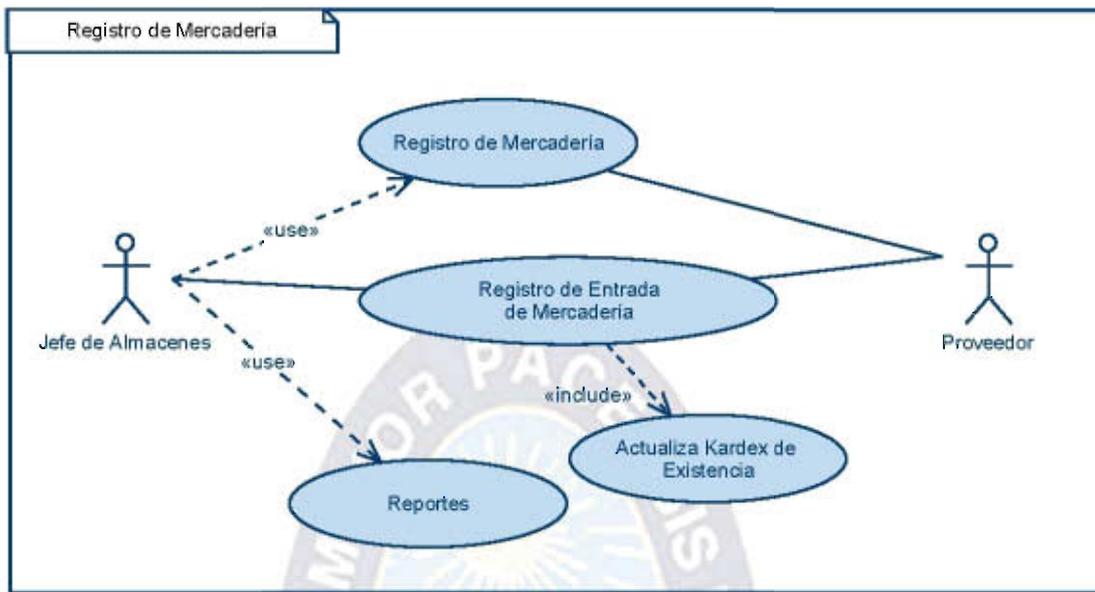


Figura 3.6.: Caso de Uso Registro de Mercadería
 Fuente: [Elaboración Propia]

Caso de Uso:	Registro de Mercadería	
Actores:	Jefe de Almacenes, Proveedor	
Propósito:	Registrar el ingreso de mercadería a los almacenes.	
Resumen:	Proceso por el cual se registra la mercadería que ingresa al almacén.	
Tipo:	Primario	
Curso Normal de Eventos		
Acción de los Actores	Respuesta del Sistema	
1.- Este caso de uso comienza cuando ingresa nueva mercadería a los almacenes. 2.- El Jefe de Almacenes atiende el ingreso de la mercadería pide sus datos para el respectivo registro de entrada y ejecuta la acción.		

<p>2.- El Jefe de Almacenes atiende el ingreso de la mercadería pide sus datos para el respectivo registro de entrada y ejecuta la acción.</p> <p>4.- El Jefe de Almacenes procede a la verificación del código de la mercadería si se encuentra registrado solo actualiza el Kardex.</p> <p>6.- El Jefe de Almacenes procede a introducir todos los datos de la mercadería.</p> <p>8.- El Jefe de Almacenes procederá a verificar la asignación de código.</p>	<p>3.- Procesa la solicitud y efectúa el proceso y verifica si está registrado o no, si lo está pasa al paso 4 y si no está pasa al paso 6.</p> <p>5.- El sistema verifica el código y actualiza el Kardex.</p> <p>7.- El sistema actualizara la base de datos de mercadería y asignará un código.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 3.15.: Registro de Mercadería
 Fuente: [Elaboración Propia]

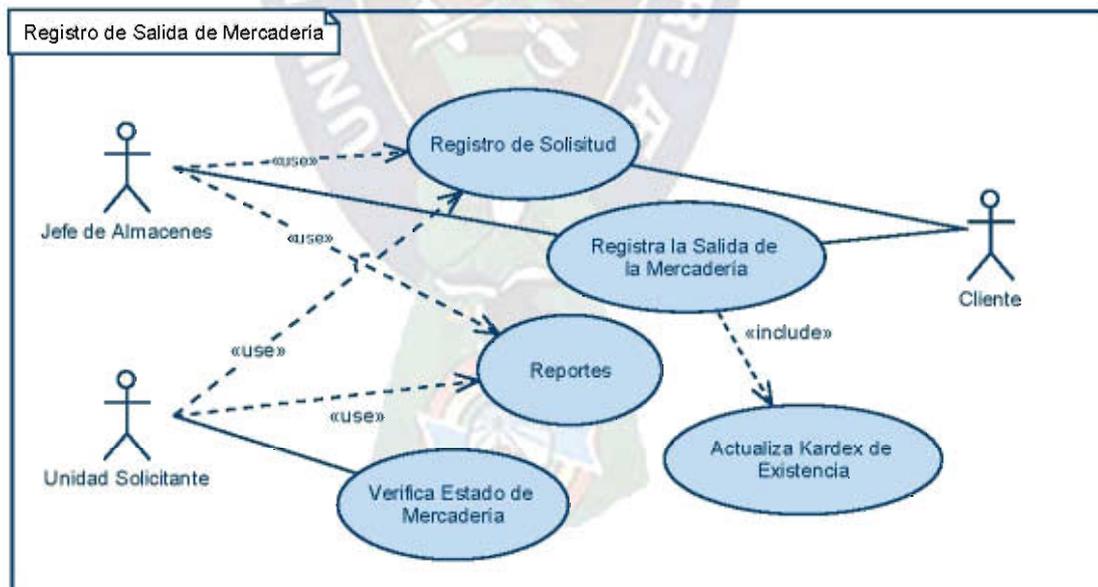


Figura 3.7.: Caso de Uso Registro de Salida de Mercadería
 Fuente: [Elaboración Propia]

Caso de Uso:	Registro de Salida de Mercadería
Actores:	Jefe de Almacenes, Unidad Solicitante, Cliente
Propósito:	Registrar la salida de mercadería.
Resumen:	Proceso por el cual se registra la salida de mercadería y verificar la estado de la misma.
Tipo:	Primario
Curso Normal de Eventos	
Acción de los Actores	Respuesta del Sistema
<p>1.- Este caso de uso comienza cuando existe un pedido de mercadería y la misma sale de almacenes.</p> <p>2.- El Jefe de Almacenes y Unidad Solicitante atiende la salida de mercadería solicita los respectivos datos para el registro y ejecuta la acción.</p> <p>4.- El Jefe de Almacenes y Unidad Solicitante procede a la verificación del pedido de la mercadería y procede al registro de la salida de almacenes.</p> <p>6.- El Jefe de Almacenes y Unidad Solicitante procede a informar la falta de mercadería.</p>	<p>3.- Procesa la solicitud y verifica si existe la cantidad suficiente de mercadería si existe pasa al paso 4 y si no existe pasa al paso 6.</p> <p>5.- El sistema verifica el código de mercadería saliente y actualiza el Kardex.</p> <p>7.- El sistema actualizara la base de datos de mercadería.</p>

Tabla 3.16.: Registro de Salida de Mercadería
 Fuente: [Elaboración Propia]

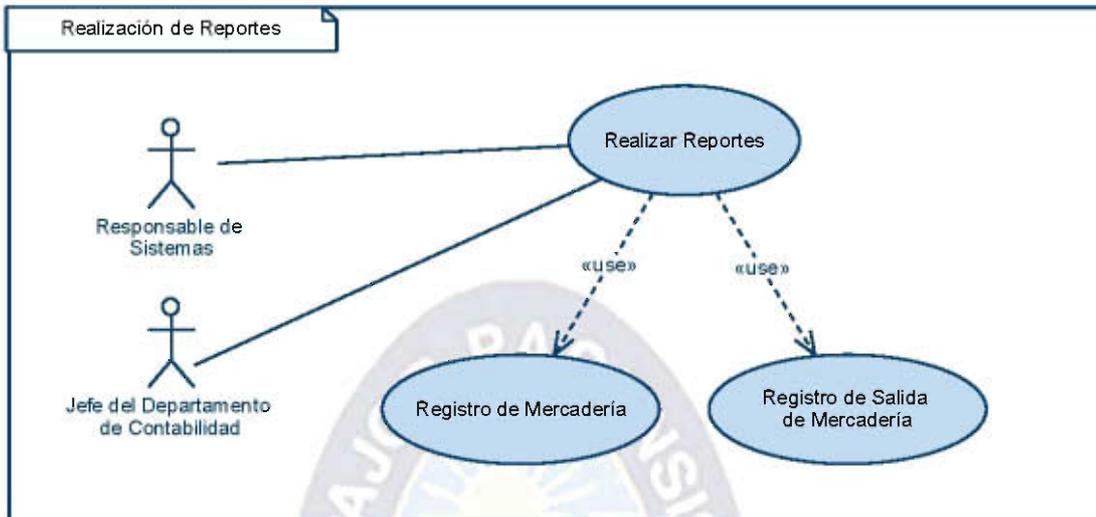


Figura 3.8.: Caso de Uso Realización de Reportes
 Fuente: [Elaboración Propia]

Caso de Uso:	Realización de Reportes	
Actores:	Responsable de Sistemas, Jefe del Departamento de Contabilidad	
Propósito:	Imprimir o visualizar el reporte de los ingresos y salidas de mercadería.	
Resumen:	Proceso por el cual se imprime o visualiza el reporte diario, semanal, mensual o anual.	
Tipo:	Primario	
Curso Normal de Eventos		
Acción de los Actores	Respuesta del Sistema	
1.- Este caso de uso comienza cuando se tiene que tomar algún tipo de decisión o se le pide simplemente un reporte de actividades. 2.- El Responsable de Sistemas o Jefe del Departamento de Contabilidad realizara el reporte con datos que introducirá al sistema como ser reporte diario, mensual, anual, dando parámetros de referencia.		

<p>4.- El Responsable de Sistemas o Jefe del Departamento de Contabilidad realiza su tarea asignada por su superior.</p>	<p>3.- Procesa la solicitud y efectúa el proceso mostrando en pantalla todo el reporte requerido.</p>
--------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------

Tabla 3.17.: Realización de Reportes
 Fuente: [Elaboración Propia]

3.3.9. DIAGRAMAS 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.

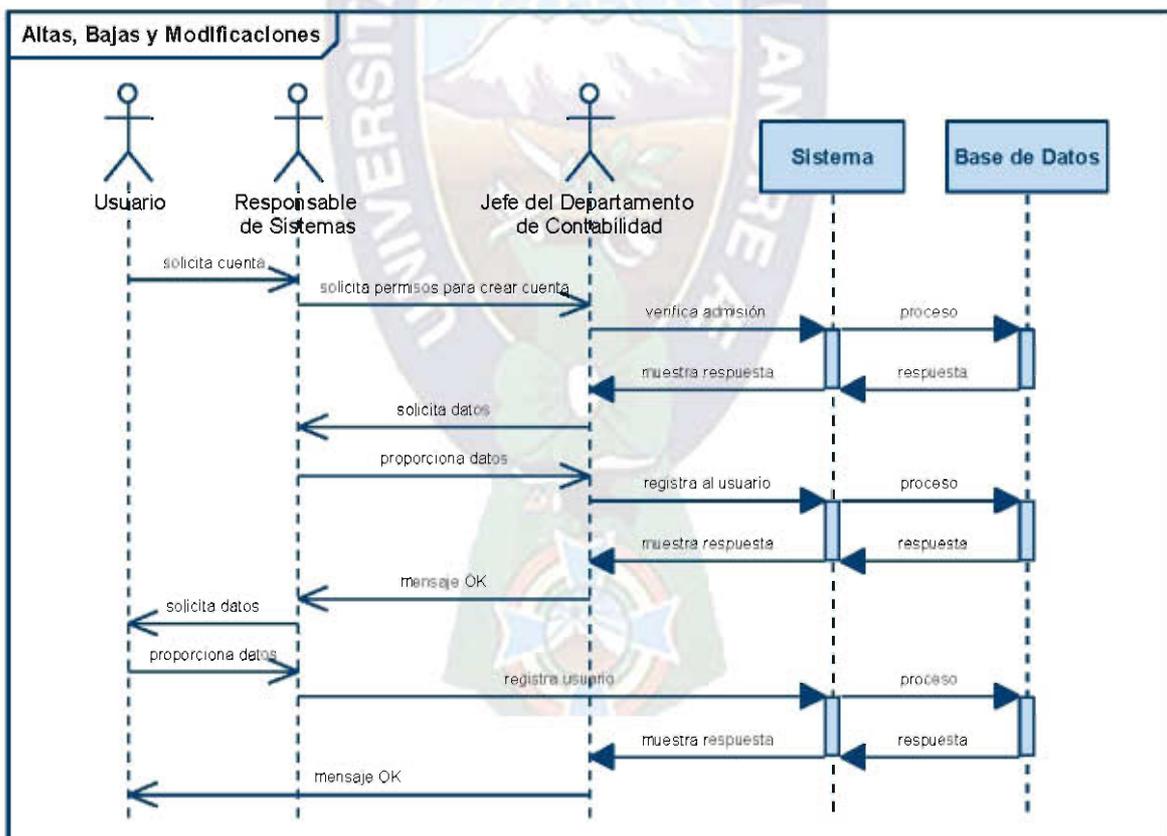


Figura 3.9.: Diagrama de Secuencia Altas, Bajas y Modificaciones
 Fuente: [Elaboración Propia]

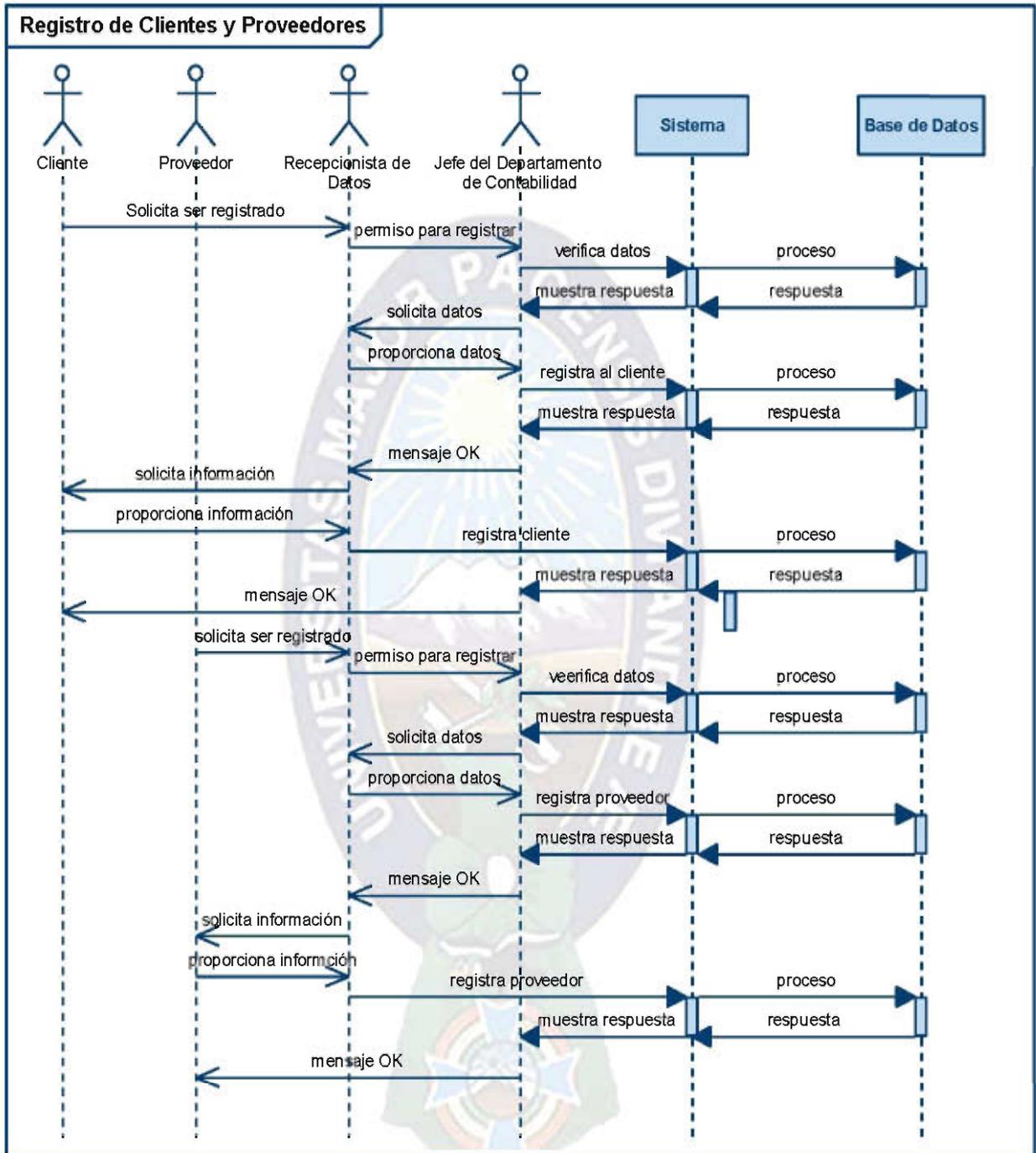


Figura 3.10.: Diagrama de Secuencia Registro de Clientes y Proveedores
 Fuente: [Elaboración Propia]

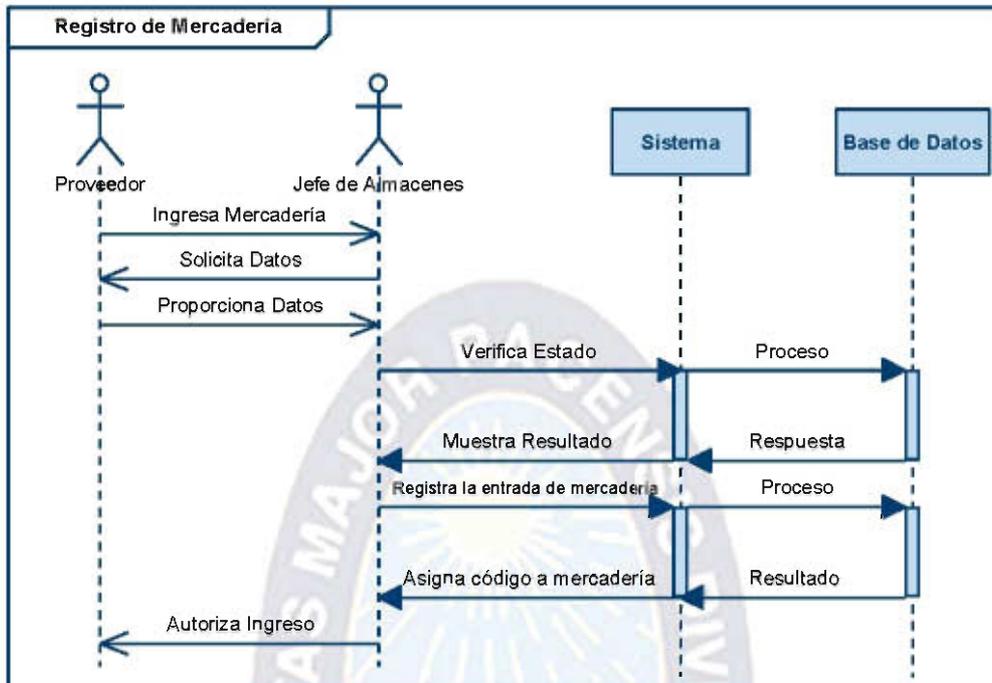


Figura 3.11.: Diagrama de Secuencia Registro de Mercadería
 Fuente: [Elaboración Propia]

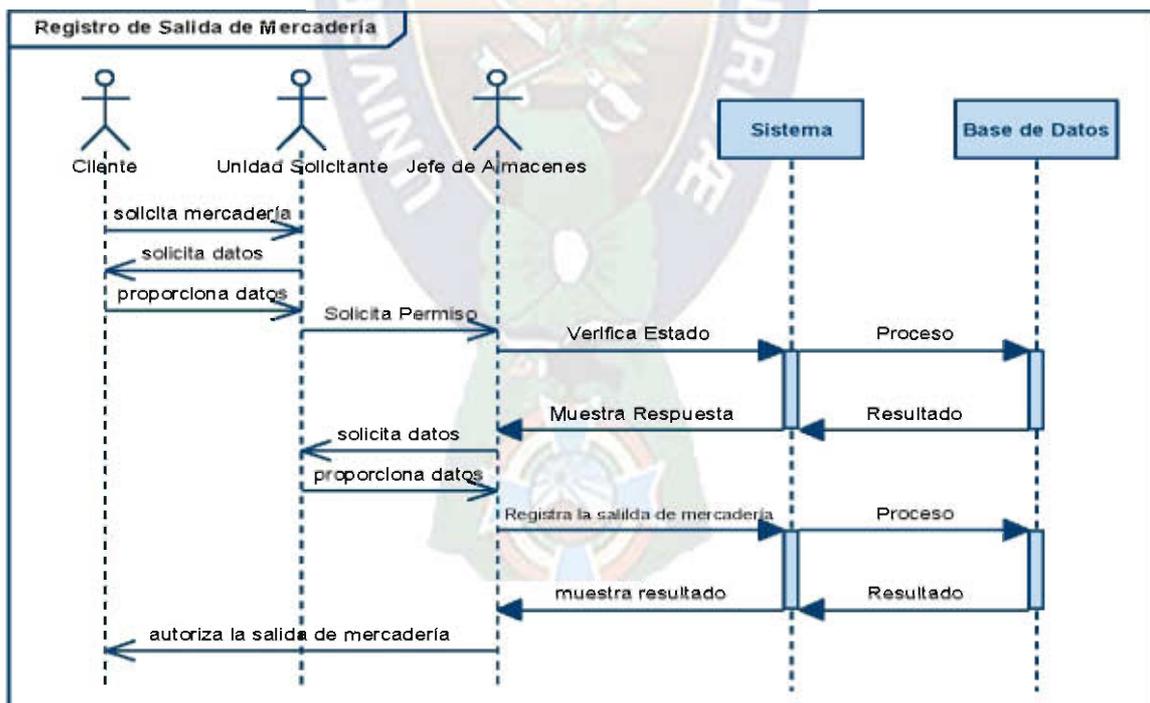


Figura 3.12.: Diagrama de Secuencia Registro de Salida de Mercadería
 Fuente: [Elaboración Propia]

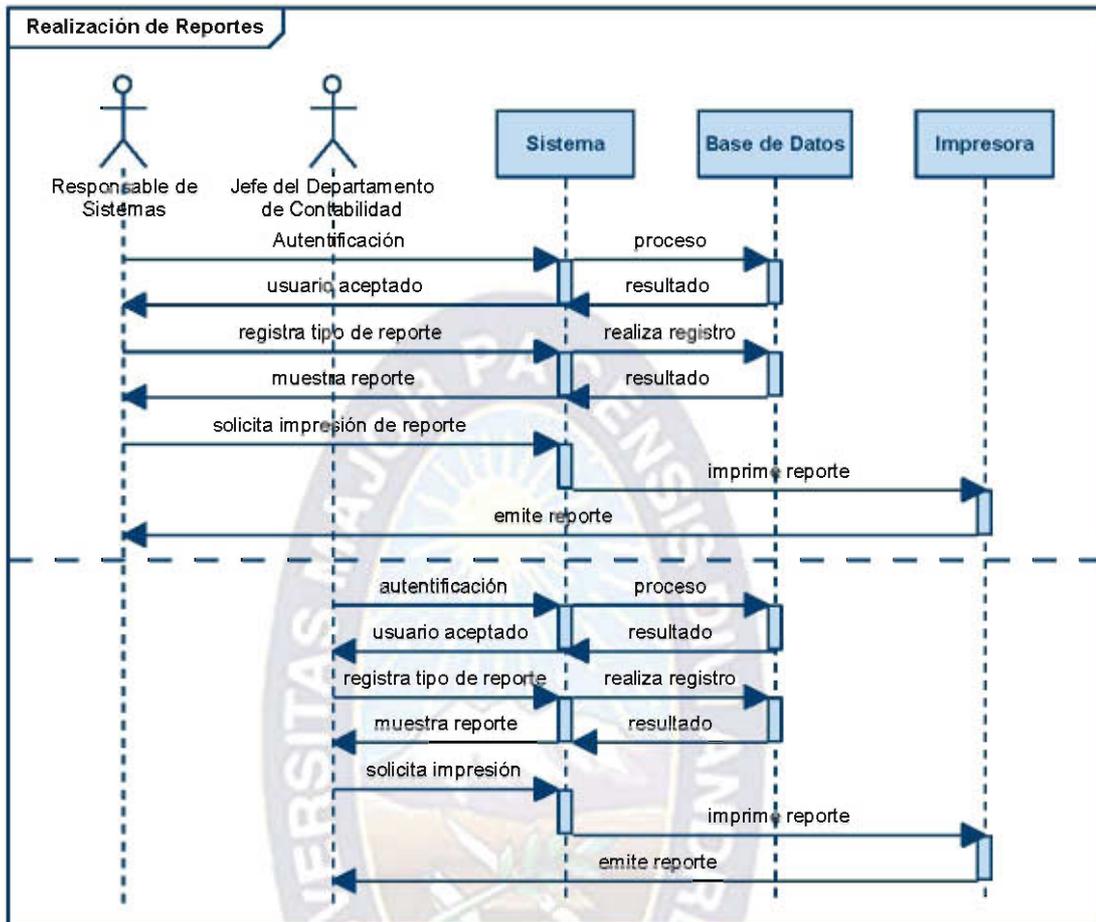


Figura 3.13.: Diagrama de Secuencia Realización de Reportes
Fuente: [Elaboración Propia]

3.3.10. DIAGRAMAS DE COLABORACIÓN

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos.

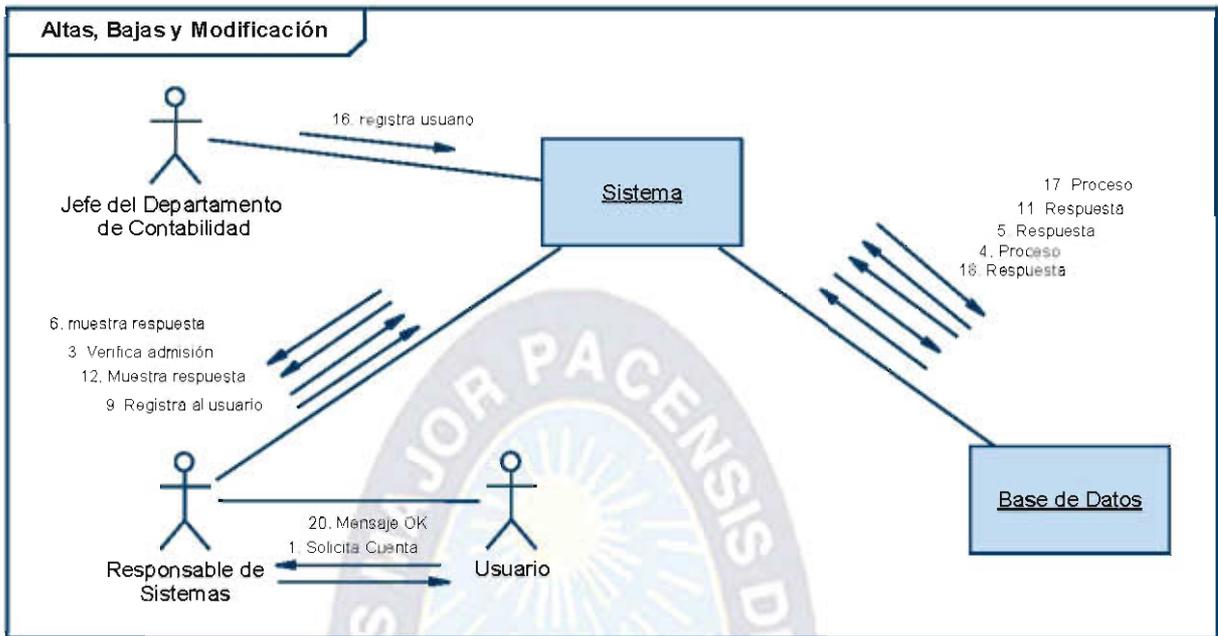


Figura 3.14.: Diagrama de Colaboración Altas, Bajas y Modificación
 Fuente: [Elaboración Propia]

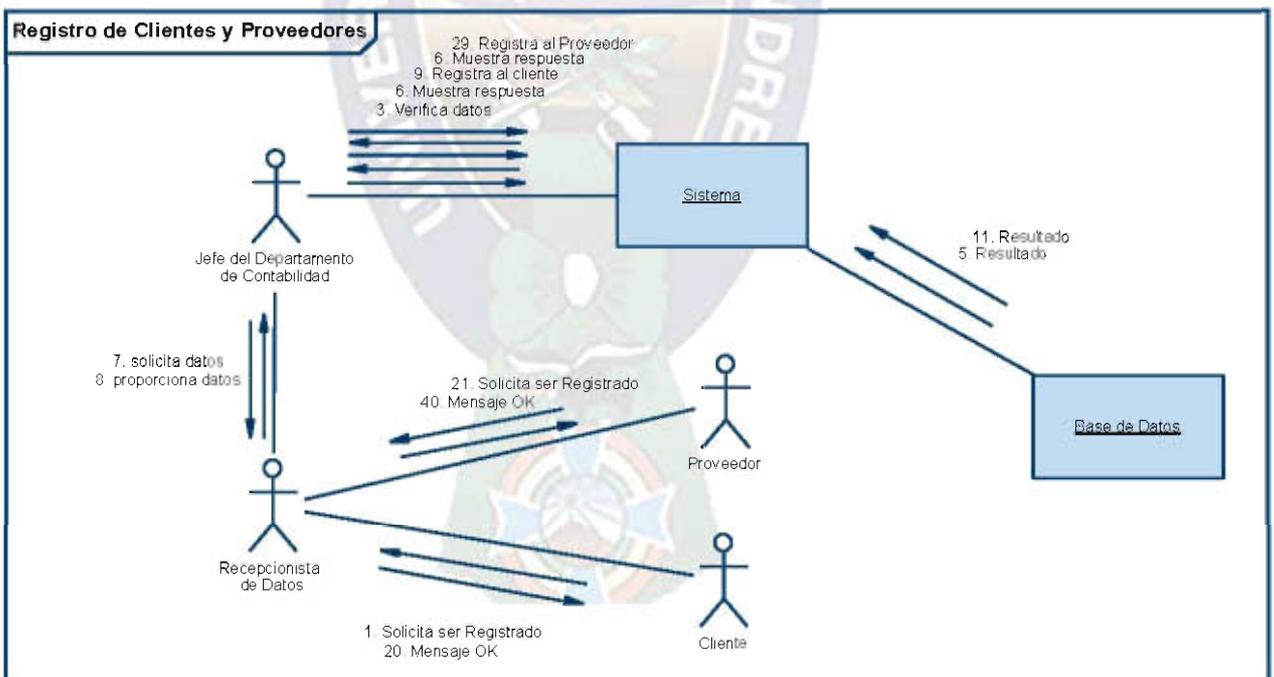


Figura 3.15.: Diagrama de Colaboración Registro de Clientes y Proveedores
 Fuente: [Elaboración Propia]

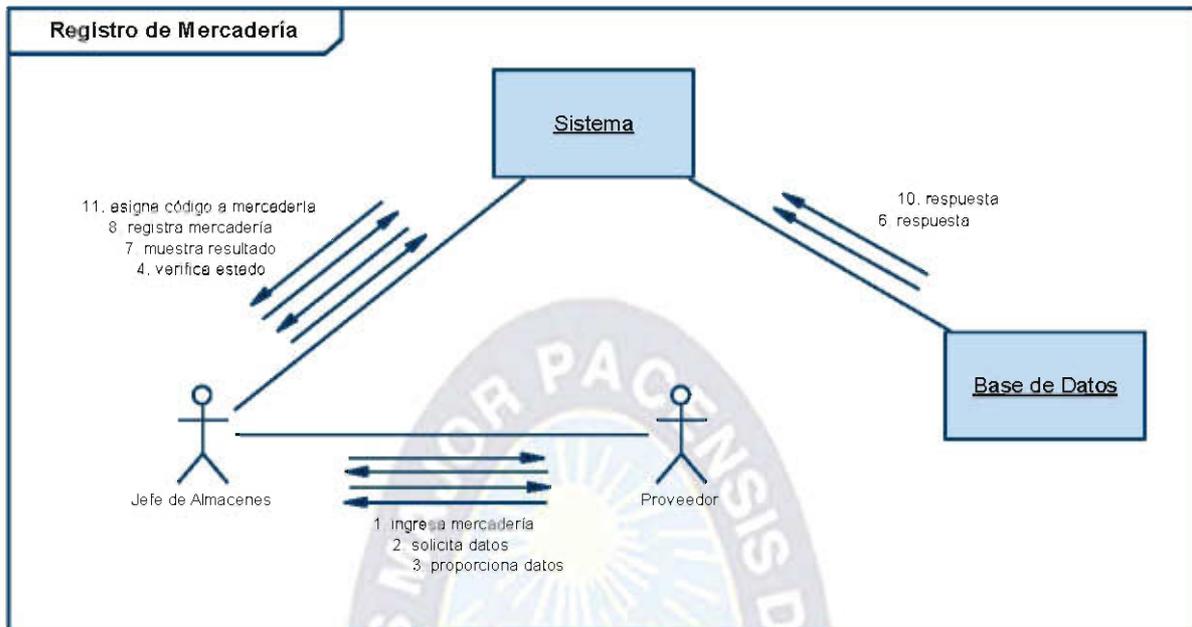


Figura 3.16.: Diagrama de Colaboración Registro de Mercadería
 Fuente: [Elaboración Propia]

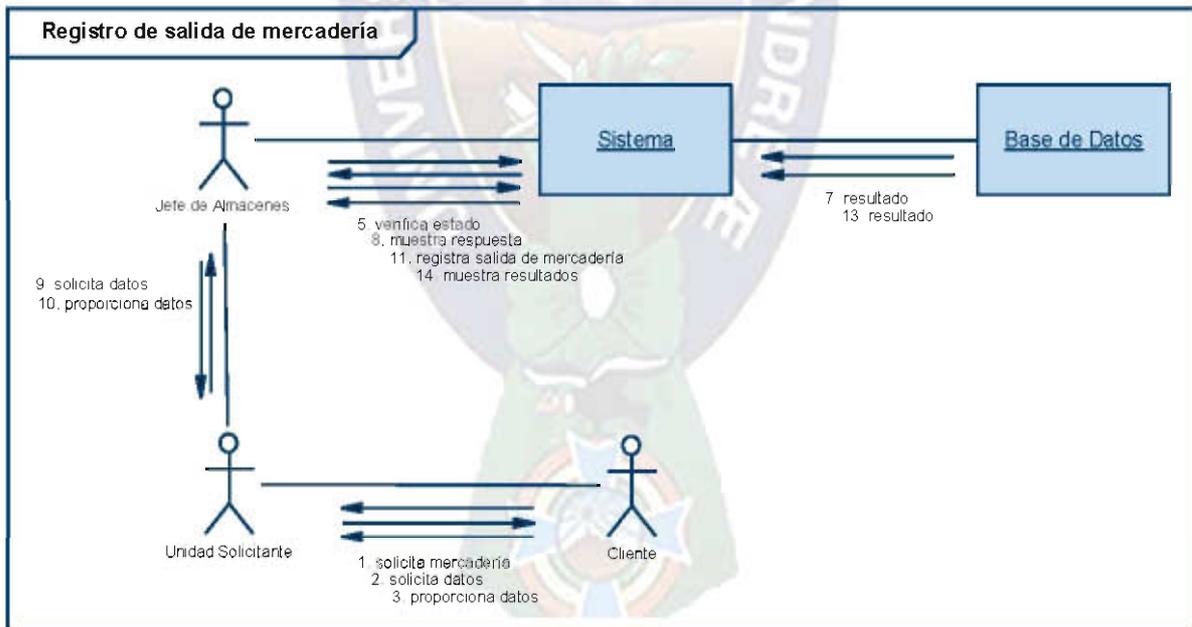


Figura 3.17.: Diagrama de Colaboración Registro de Salida de Mercadería
 Fuente: [Elaboración Propia]

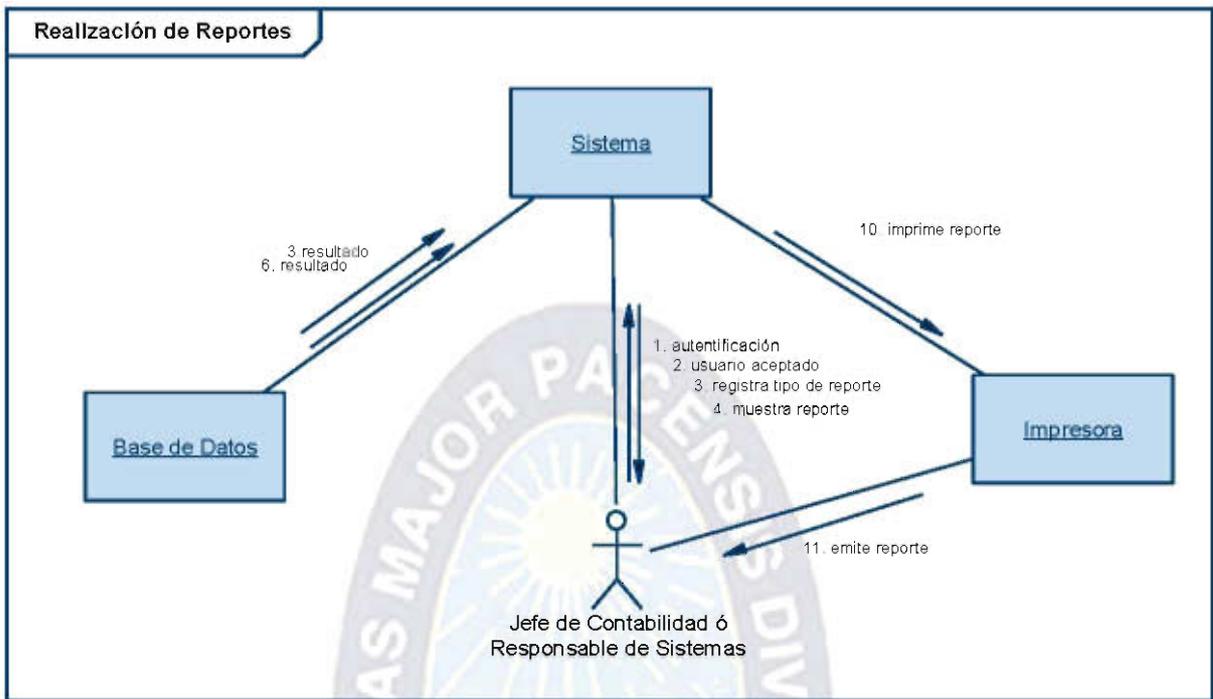


Figura 3.18.: Diagrama de Colaboración Realización de Reportes
 Fuente: [Elaboración Propia]

3.3.11. DIAGRAMA DE DESPLIEGUE DEL SISTEMA

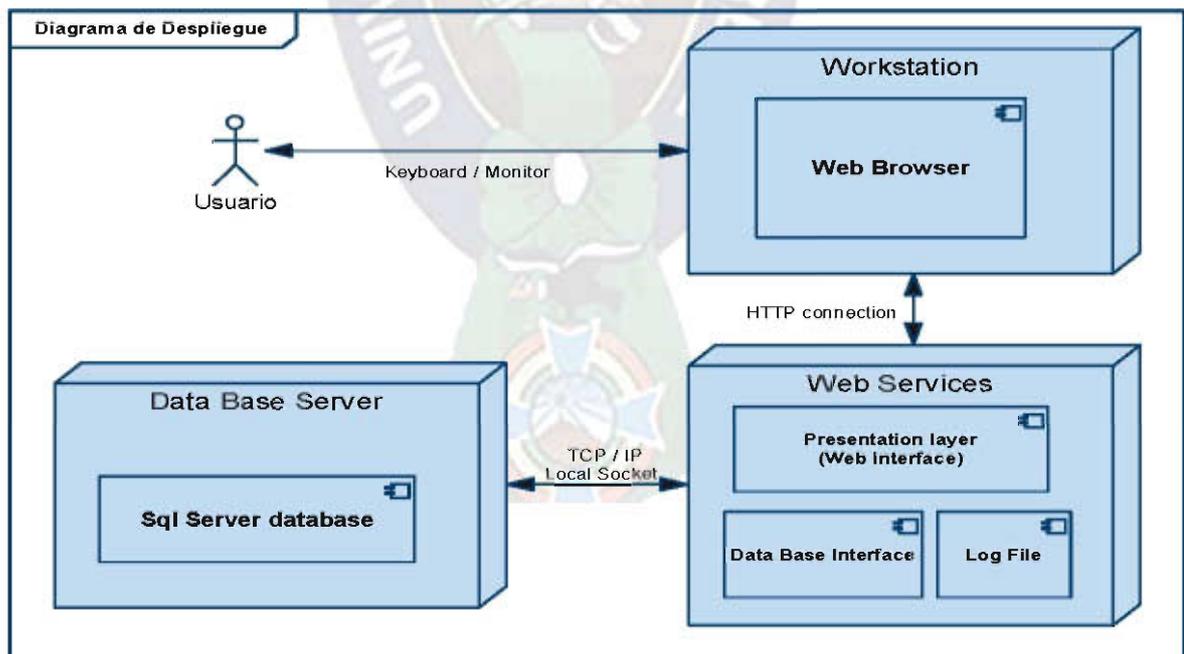


Figura 3.19.: Diagrama de Despliegue
 Fuente: [Elaboración Propia]

3.3.12. ESPECIFICACIÓN DE UID's

Los **Diagramas De Interacciones De Usuario** o UID nos permitirán representar de forma sencilla las historias de usuario de manera que logremos ver de una manera más simple el intercambio de información que existe entre el sistema y el usuario.

UID: Altas, Bajas y Modificaciones
Actor: Responsable de Sistemas o Jefe del Departamento de Contabilidad



Figura 3.20.: UID Altas Bajas y Modificaciones
Fuente: [Elaboración Propia]

UID: Registro de Clientes y Proveedores
Actor: Recepcionista de Datos o Jefe del Departamento de Contabilidad

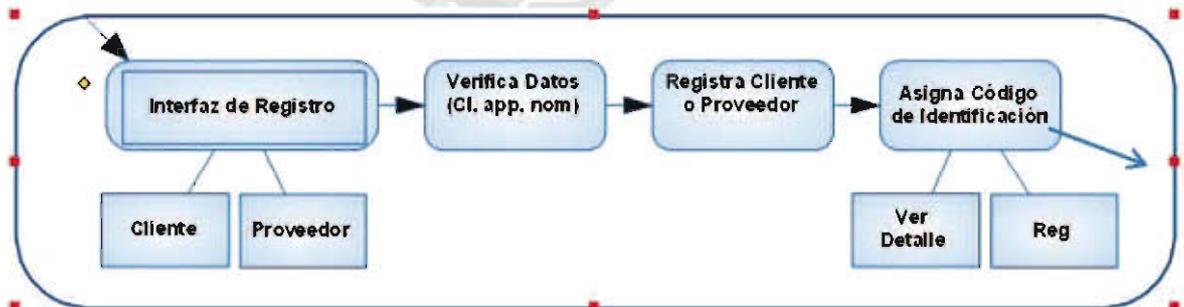


Figura 3.21.: UID Registro de Clientes y Proveedores
Fuente: [Elaboración Propia]

UID: Registro de Mercadería
Actor: Jefe de Almacenes



Figura 3.22.: UID Registro de Mercadería
Fuente: [Elaboración Propia]

UID: Registro de Salida de Mercadería
Actor: Jefe de Almacenes y Unidad Solicitante



Figura 3.23.: UID Registro de Salida de Mercadería
Fuente: [Elaboración Propia]

UID: Realización de Reportes
Actor: Responsable de Sistemas, Jefe del Departamento de Contabilidad



Figura 3.24.: UID Realización de Reportes
Fuente: [Elaboración Propia]

3.4. MODELO CONCEPTUAL

3.4.1. DIAGRAMA DE CLASES

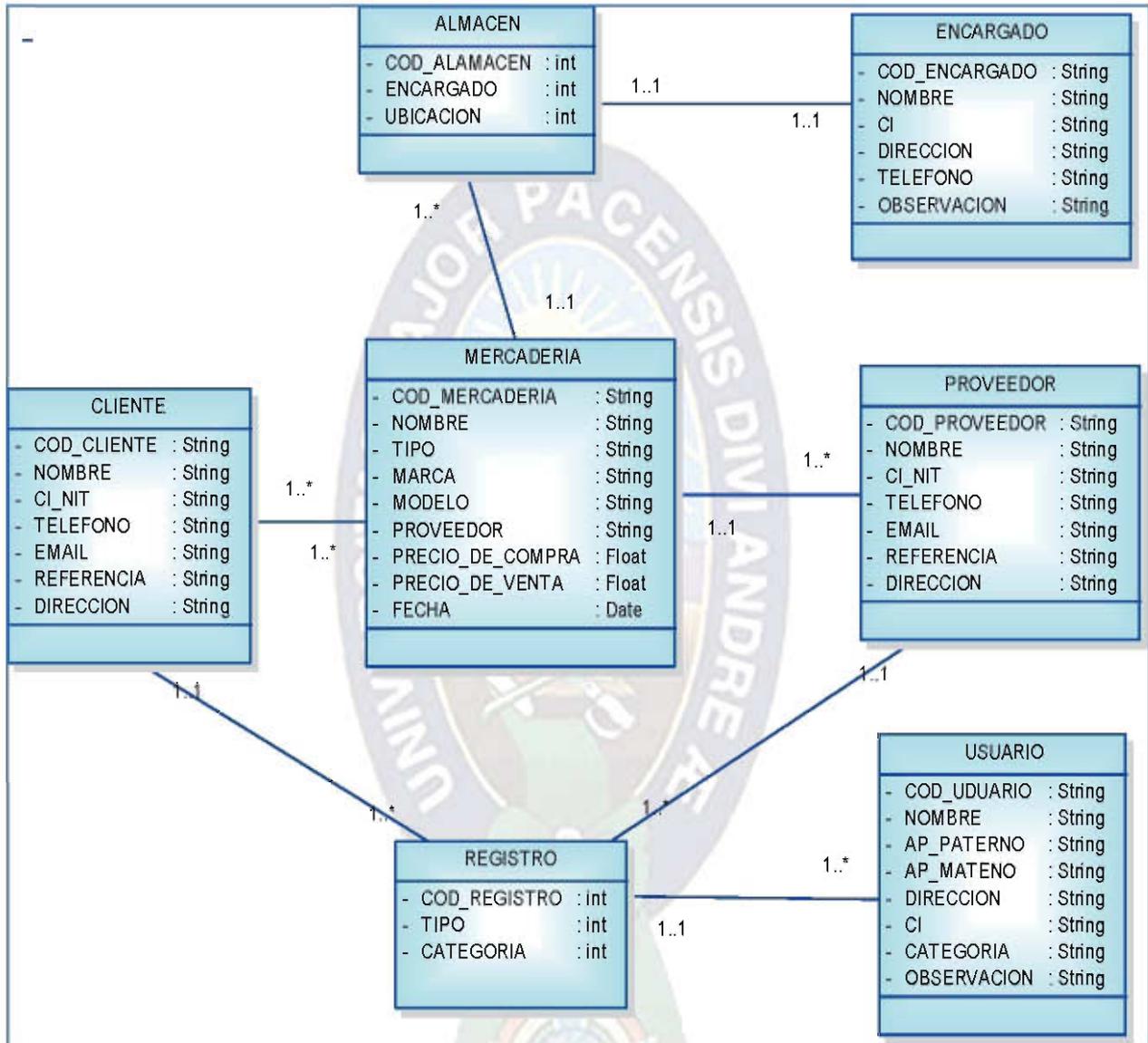


Figura 3.25 Diagrama de Clase
Fuente: [Elaboración Propia]

3.5. DISEÑO NAVEGACIONAL

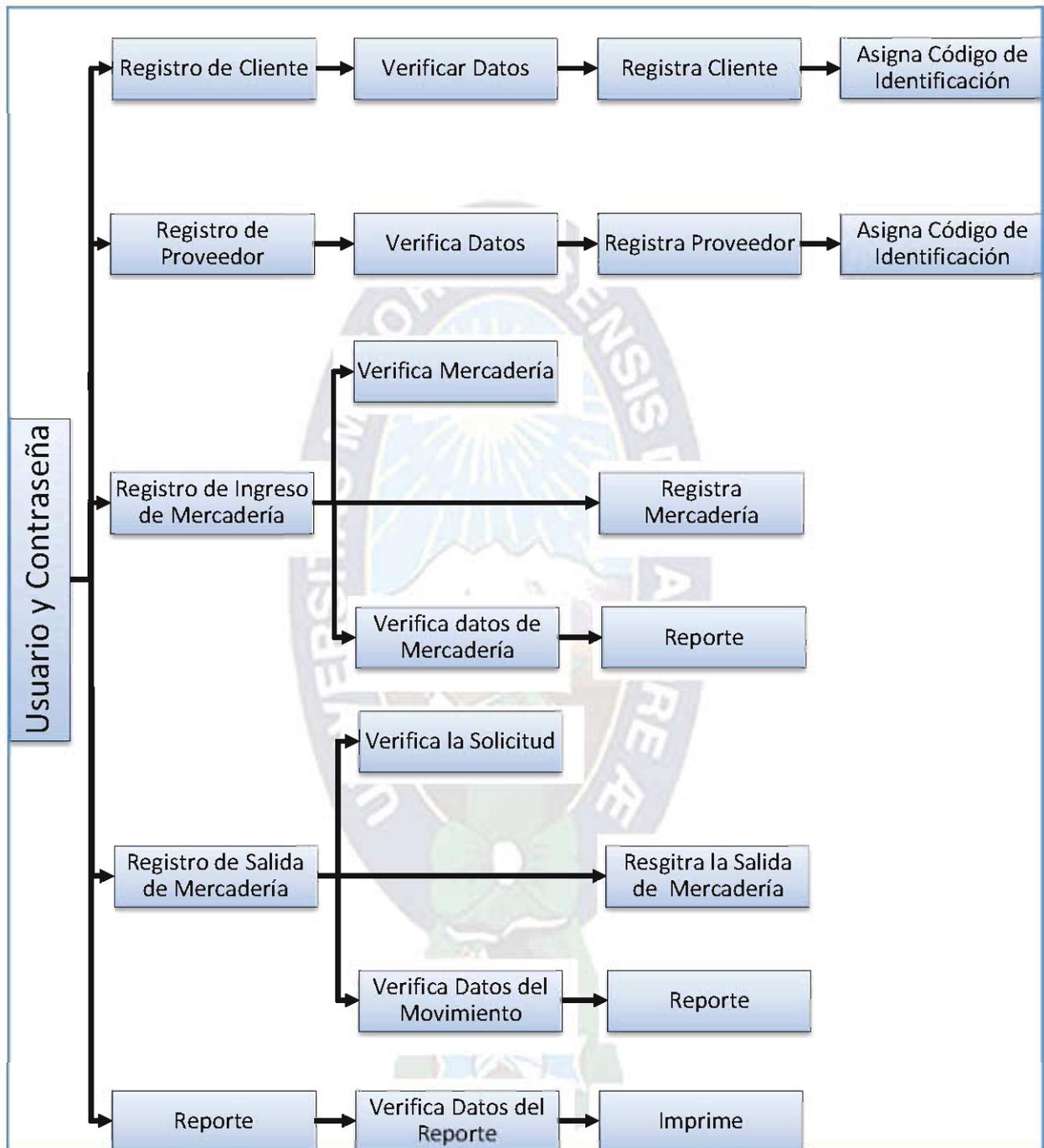


Figura 3.26 Diseño Navegacional
Fuente: [Elaboración Propia]

3.6. DISEÑO DE INTERFAZ ABSTRACTA

- ADV PRINCIPAL



Figura 3.27 ADV Principal
Fuente: [Elaboración Propia]

- ADV REGISTRO DE USUARIOS



Figura 3.28 ADV Registro de Usuario
Fuente: [Elaboración Propia]

▪ **ADV REGISTRO DE MERCADERÍA**

ADV PRINCIPAL BANNER

ADV PRINCIPAL [INGRESO DE MERCADERIA]

REGISTRO DE MERCADERÍA

CODIGO DE MERCADERÍA:	
TIPO	
MARCA	
MODELO	
COLOR	
FECHA DE INGRESO	
PROVEEDOR	
DESCRIPCIÓN	
CANTIDAD	

GUARDAR

HOME

- INGRESAR MERCADERIA
- ACTUALIZAR
- CREAR UBICACION
- REGISTRA ENCARGADO
- ACTUALIZAR USERS
- ACTUALIZAR ENCARGADOS
- REGISTRO USUARIOS
- REPORTES

Figura 3.29 ADV Registro de Mercadería
Fuente: [Elaboración Propia]

▪ **ADV REGISTRO DE SALIDA DE MERCADERÍA**

ADV PRINCIPAL BANNER

ADV PRINCIPAL [SALIDA DE MERCADERIA]

REGISTRO DE SALIDA DE MERCADERÍA

CODIGO DE MERCADERÍA:	
TIPO	
MARCA	
MODELO	
COLOR	
FECHA DE SALIDA	
CANTIDAD	

GUARDAR

HOME

- INGRESAR MERCADERIA
- ACTUALIZAR
- CREAR UBICACION
- REGISTRA ENCARGADO
- ACTUALIZAR USERS
- ACTUALIZAR ENCARGADOS
- REGISTRO USUARIOS
- REPORTES

Figura 3.30 ADV Registro de Salida de Mercadería
Fuente: [Elaboración Propia]

- **ADV REGISTRO DE CLIENTE O PROVEEDOR**

The screenshot shows a web interface for 'ADV PRINCIPAL BANNER'. Below the banner is a sub-header 'ADV PRINCIPAL [CLIENTE/PROVEEDOR]'. The main content area is divided into two sections. The left section, titled 'CLIENTE O PROVEEDOR', contains a large text input field at the top, followed by a table of input fields: 'CI:', 'NOMBRE DEL ENTE', 'TELEFONO', 'EMAIL', and 'REFERENCIA'. Below this table is an 'OBSERVACIÓN' text input field and a 'GUARDAR' button. The right section, titled 'HOME', contains a vertical list of menu items: 'INGRESAR MERCADERIA', 'ACTUALIZAR', 'CREAR UBICACION', 'REGISTRA ENCARGADO', 'ACTUALIZAR USERS', 'ACTUALIZAR ENCARGADOS', 'REGISTRO USUARIOS', and 'REPORTES'.

Figura 3.31 ADV Registro de Cliente o Proveedor
Fuente: [Elaboración Propia]

- **ADV REALIZACION DE REPORTE**

The screenshot shows a web interface for 'ADV PRINCIPAL BANNER'. Below the banner is a sub-header 'ADV PRINCIPAL [REPORTES]'. The main content area is divided into two sections. The left section, titled 'REPORTES', contains a large text input field at the top, followed by a table of input fields: 'POR MFC', 'POR DIA', and 'GENERAL'. Below this table is a 'DETALLE' text input field. The right section, titled 'HOME', contains a vertical list of menu items: 'INGRESAR MERCADERIA', 'ACTUALIZAR', 'CREAR UBICACION', 'REGISTRA ENCARGADO', 'ACTUALIZAR USERS', 'ACTUALIZAR ENCARGADOS', 'REGISTRO USUARIOS', and 'REPORTES'.

Figura 3.32 ADV Realización de Reportes
Fuente: [Elaboración Propia]

3.7. IMPLEMENTACIÓN

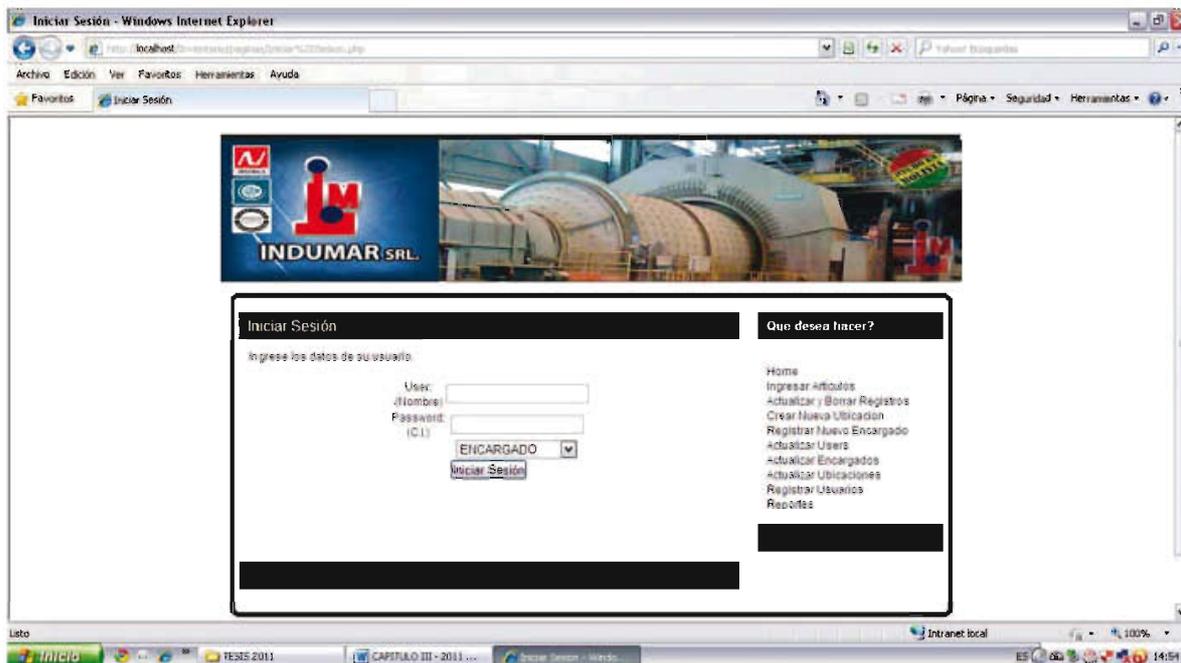


FIGURA 3.33: PANTALLA DE LOGIN
FUENTE: [ELABORACIÓN PROPIA]

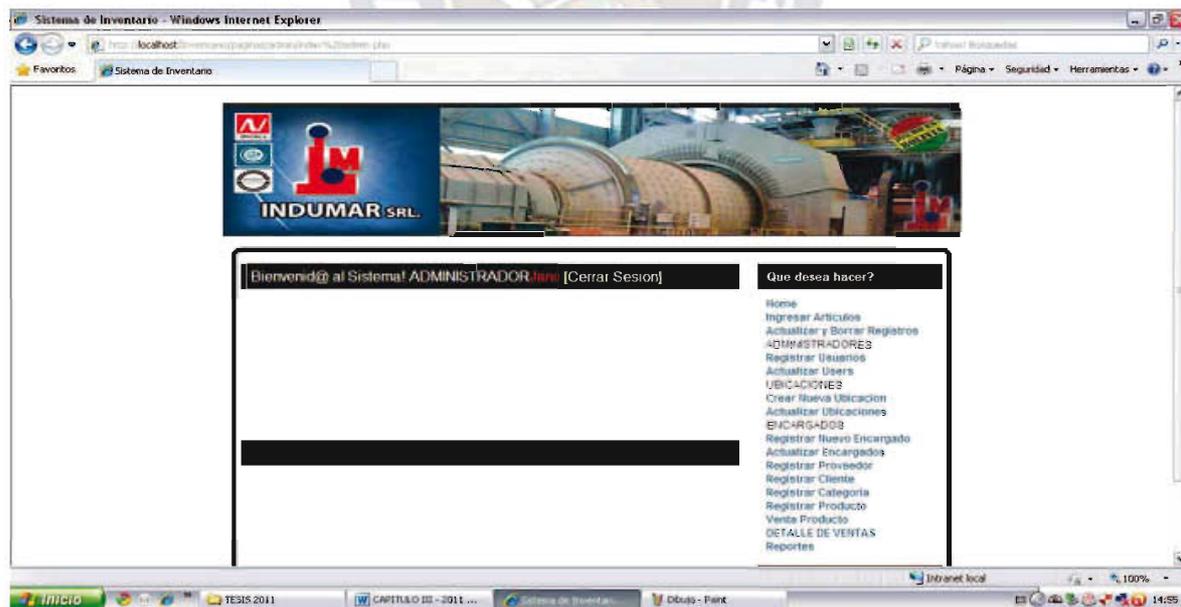


FIGURA 3.34: PANTALLA DE BIENVENIDA
FUENTE: [ELABORACIÓN PROPIA]

Sistema Cliente – Servidor de Inventario de Productos de Poliuretano
Caso: INDUMAR S.R.L.

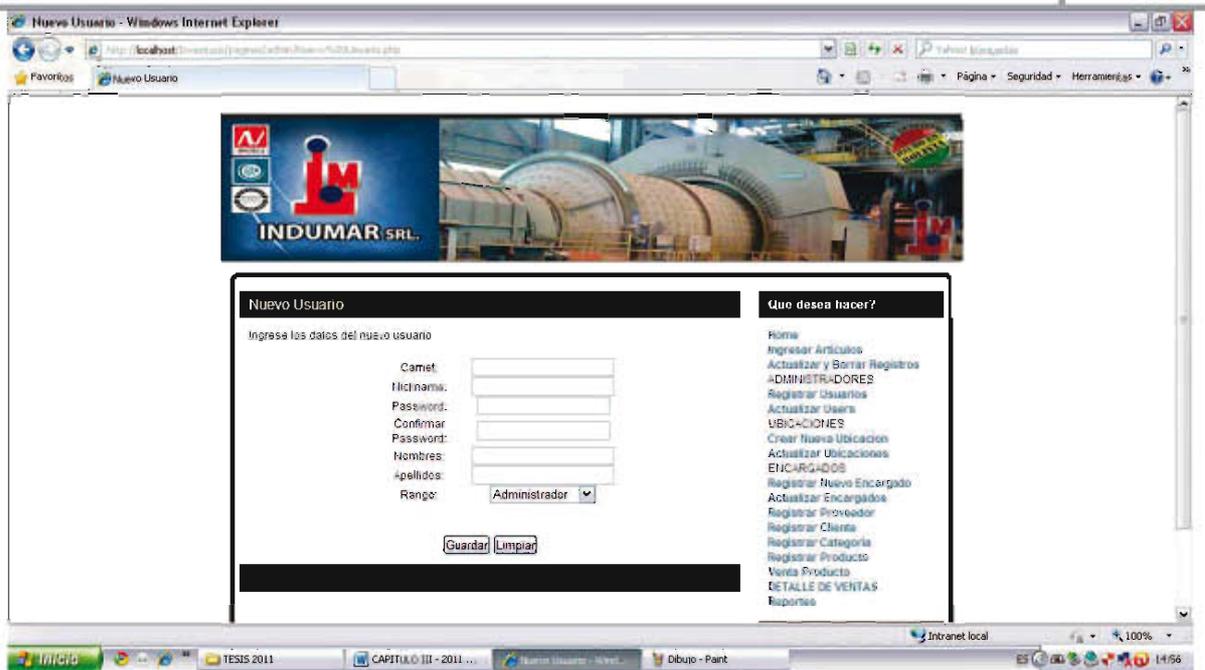


FIGURA 3.35: PANTALLA DE REGISTRO DE USUARIOS
FUENTE: [ELABORACIÓN PROPIA]

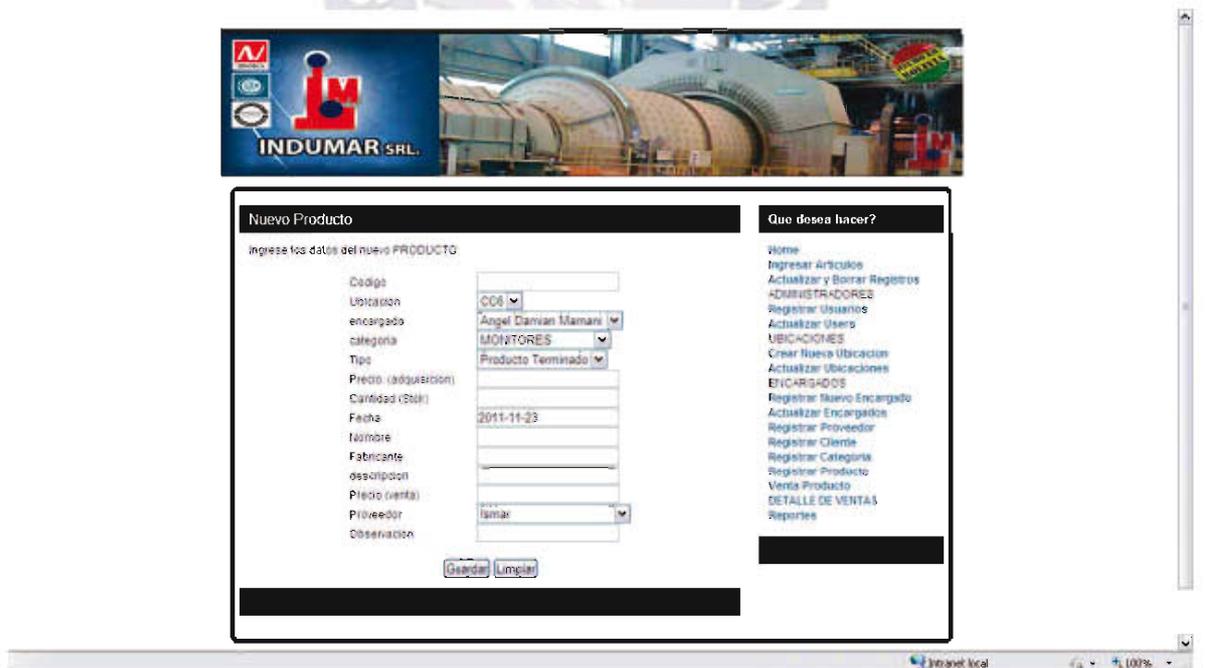


FIGURA 3.36: PANTALLA DE REGISTRO DE MERCADERÍA
FUENTE: [ELABORACIÓN PROPIA]

Sistema Cliente – Servidor de Inventario de Productos de Poliuretano
 Caso: INDUMAR S.R.L.

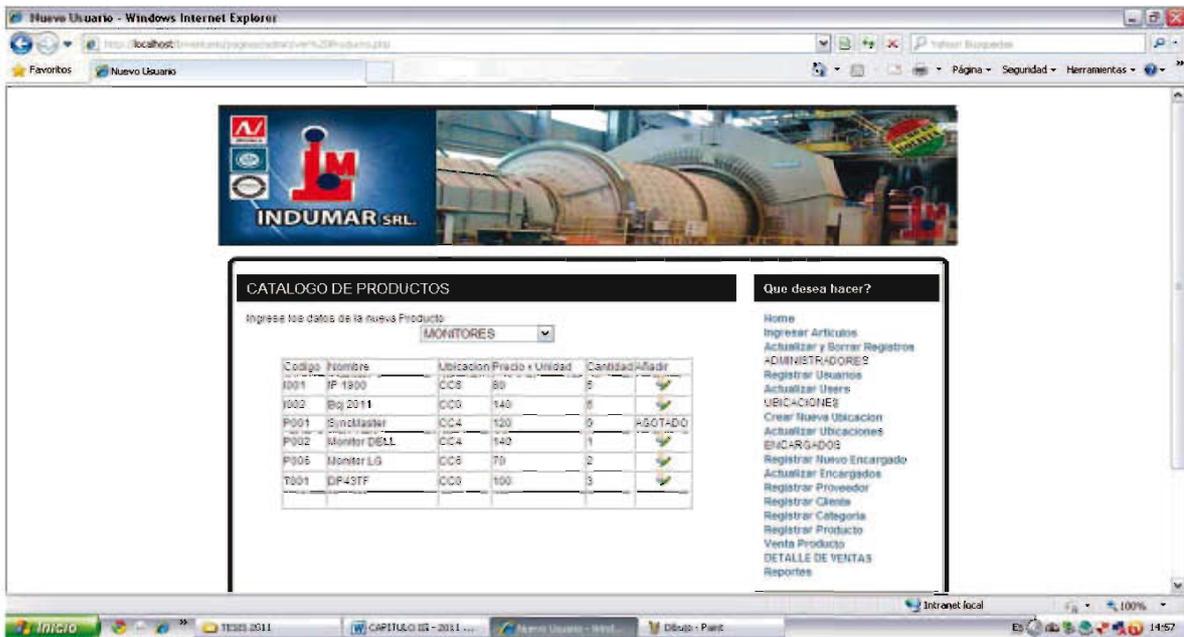


FIGURA 3.37: PANTALLA DE REGISTRO DE VENTA DE MERCADERÍA
 FUENTE: [ELABORACIÓN PROPIA]

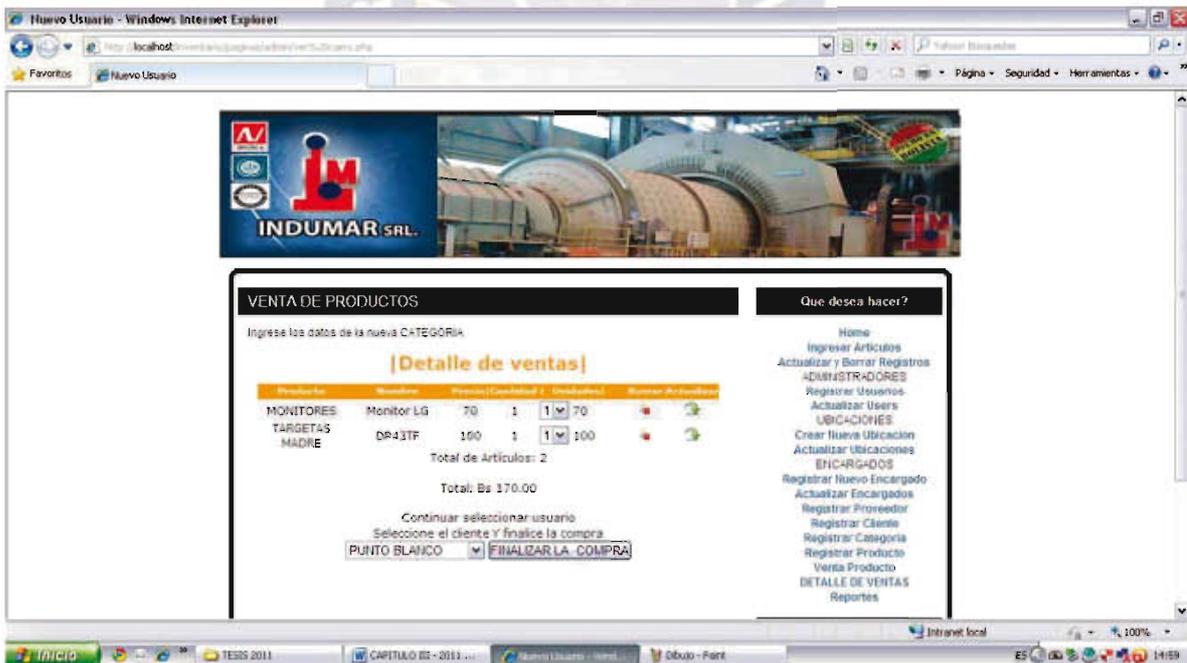


FIGURA 3.38: PANTALLA DEL DETALLE DE VENTAS
 FUENTE: [ELABORACIÓN PROPIA]

CAPITULO IV

4. METRICAS DE CALIDAD

Las métricas de calidad que se utilizan para la medición de un software, son importantes ya que mediante ellas se puede testear la calidad de un software [PRESSMAN, 2003]. Se puede mencionar que existen diferentes factores de calidad de software el factor a utilizar en el presente proyecto es el estándar ISO 9126 que identifica los atributos claves de calidad para el software los cuales son:

- Funcionalidad
- Confiabilidad
- Eficiencia
- Facilidad de mantenimiento
- Portabilidad

4.1. FUNCIONALIDAD

Se entiende por funcionalidad el grado que el software satisface las necesidades funcionales.

4.1.1. MÉTRICAS BASADAS EN EL PUNTO FUNCIÓN

La métrica punto función se usa como medio para predecir el tamaño del sistema que se va obtener por medio del análisis.

- Número de entradas de Usuario

Es el número de entradas que proporciona datos al sistema con el fin de poder realizar las distintas operaciones (Altas, Bajas, modificaciones).

- Número de peticiones de Usuario

Es una entrada interactiva que produce la generación de alguna respuesta de software en forma de salida interactiva.

- Numero de archivos.
 Es un grupo lógico de datos que pueden ser parte de una gran base de datos o de un archivo independiente
- Numero de Interfaces Externas
 Es el número de interfaces legibles por la máquina que se utiliza para transmitir información a otra máquina.
- Tamaño de sistema

Entrada de Usuario	<ul style="list-style-type: none"> • Registro de usuarios • Registro de Entrada de Mercadería • Registro de Salida de Mercadería • Registro de Clientes • Registro de Proveedores
Salidas de Usuario	<ul style="list-style-type: none"> • Reportes de usuario registrados • Reportes de Entrada de Mercadería • Reportes de Salida de Mercadería • Reportes por periodo • Reportes por encargado • Reportes por ubicación
Peticiones de Usuario	<ul style="list-style-type: none"> • Listado de clientes • Listado de proveedores • Listado de Mercadería • Listados de usuarios • Listado de ubicaciones • Listado de encargados
Número de Archivos	Tablas de la base de datos 7
Interfaces Externas	<ul style="list-style-type: none"> • Disco • Impresora • Backups • Conexión de red

Tabla 4.1.: Tamaño de Sistema
 Fuente: [Elaboración Propia]

Una vez recopilados los datos anteriores, a continuación se realiza el cálculo de punto función, asignando un factor de ponderación a los parámetros de medición ya descritos.

Parámetros de Medición	Cuenta	Factor de Ponderación	Total
Número de entradas de usuario	5	4	20
Número de salidas de usuario	6	5	30
Número de peticiones de usuario	6	4	24
Número de archivos	11	10	110
Número de interfaces externas	4	7	28
Cuenta Total			212

Tabla 4.2 Parámetros de Medición
 Fuente: [Elaboración Propia]

Para calcular el punto función (PF), se utiliza la siguiente relación:

$$PF = \text{Cuenta Total} * [0.65 + 0.01 * \sum F_i] \quad (1)$$

Dónde:

Cuenta Total: Es la suma de todas las entradas PF de la tabla 4.2

F_i (i=1 a 14): son valores de ajustes de complejidad.

0.65: Confiabilidad del sistema

0.01: Error de confiabilidad del sistema

Los valores de ajuste de complejidad, se evalúan en base a una escala de ajustes vista en la tabla 4.3

Descripción	Escala
No influencia	0
Incidencia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Tabla 4.3 Escala de Ajustes
 Fuente: [Elaboración Propia]

Los valores de ajuste de complejidad se obtienen respondiendo a las preguntas destacables que se muestran en la tabla 4.4 Aplicando los valores de la tabla de escala de ajustes tabla 4.3

Valores de Ajustes (Preguntas)	Valor
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?	5
2.- ¿Se requiere comunicación de datos?	5
3.- ¿Existen funciones procesos distribuidos?	3
4.- ¿Es crítico el rendimiento?	2
5.- ¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	5
6.- ¿Requiere el sistema la entrada de datos interactiva?	4
7.- ¿Requiere la entrada de datos interactiva que las transacciones se lleven a cabo sobre múltiples pantallas u operaciones?	2
8.- ¿Se actualiza los archivos maestros de forma interactiva?	4
9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	2
10.- ¿Es complejo el procesamiento interno?	3

11.- ¿Se ha diseñado el código para ser reutilizable?	4
12.- ¿Están incluidos en el diseño la conversión y la instalación?	4
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?	5
∑Fi = Total	53

Tabla 4.4 Valores de Ajustes de complejidad
 Fuente: [Elaboración Propia]

Una vez obtenido el valor de ajuste y reemplazando en la ecuación (1) se tiene:

$$PF = \text{Cuenta Total} * [0.65 + 0.01 * \sum Fi]$$

$$PF = 212 * [0.65 + 0.01 * 53] = 250.16$$

Ajuste a la curva normal:

$$\text{Funcionalidad} = PF / PF \text{ Mximo}$$

Dnde:

$$PF \text{ Mximo} = \text{Cuenta Total} * [0.65 + 0.01 * \sum Fi]; \text{ se toma el valor mximo de } \sum Fi [i=1 \text{ a } 14] [0 \leq \sum Fi \leq 70]$$

Reemplazando se tiene:

$$PF \text{ Mximo} = \text{Cuenta Total} * [0.65 + 0.01 * \sum Fi]$$

$$PF \text{ Mximo} = 212 * [0.65 + 0.01 * 70] = 286.2$$

Por tanto:

$$\begin{aligned} \text{Funcionalidad} &= (PF / PF \text{ Mximo}) * 100 = (250.16 / 286.2) * 100 \\ &= 87.40 \end{aligned}$$

La funcionalidad del sistema es de: 87%

4.2. CONFIABILIDAD

La confiabilidad es la cantidad de tiempo que el software está disponible para su uso, es decir, la cantidad de tiempo que el sistema se encuentra en funcionamiento dentro de la institución y esté libre de fallas; se puede considerar que a mayor número de fallas, menos confiabilidad, pero a menor número de fallas mayor será la confiabilidad.

- Para poder medir la confiabilidad del sistema, se aplicó la teoría estadística, el cual nos permitirá calcular el porcentaje de confiabilidad, para este se tomaron las siguientes consideraciones:
- Supongamos que el sistema comienza a trabajar en el instante $t = 0$ y se observa hasta que falle, lo cual ocurre en el instante t .
- Designamos una variable aleatoria T , que representa la duración del tiempo de trabajo del sistema sin fallas, entonces la probabilidad de que el sistema falle en un tiempo t será:
 $P [T \leq t]$ y la probabilidad de funcionamiento del sistema será:
 $P [T > t] = 1 - F (t)$
- Para calcular la confiabilidad $R (t)$ del sistema, se tomara en cuenta un periodo t que será el periodo de prueba hasta que se produzca la falla, para esto aplicaremos la función exponencial y considerando el valor del punto función calculado en la anterior sección.
 $F (t) = PF * e^{(-x/10)}$; con $x > 0$
- El margen de error de Lambda es de un décimo, significa que calculando el error durante 10 ejecuciones en un mes obtendremos un valor probable de fallas que puede tener el sistema

- Si calculamos para una gestión de un año es decir 12 meses aplicando el valor Punto Función $PF = 0.874$ obtenido en la sección anterior.

A continuación se realizara el cálculo de la probabilidad de que el sistema tenga fallas:

$$P [T \leq t] = F(t)$$

$$F(t) = PF * e^{(-x/10)} \quad ; \text{ con } x > 0$$

$$F(t) = 0.874 * (\text{Exp} (-1/10 * 12))$$

$$\text{Para } t=12$$

$$F(12) = 0.26 = 26\%$$

Calculo de probabilidad del sistema sin fallas:

$$P [T \leq t] = 1 - F(t)$$

$$F(t) = PF * e^{(-x/10)} \quad ; \text{ con } x > 0$$

$$F(t) = 1 - [0.874 * (\text{Exp} (-1/10 * 12))]$$

$$\text{Para } t = 12$$

$$F(12) = 1 - 0.26 = 0.73 = 73\%$$

Por tanto se puede decir que el sistema tiene un grado de confiabilidad del 73% y que seguirá funcionando en un año (12 meses).

4.3. FACILIDAD DE MANTENIMIENTO

La Facilidad de mantenimiento, es la facilidad con que una modificación puede ser realizada en el sistema. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del sistema a cambios en el entorno y especificaciones de requerimiento funcionales [PRESSMAN, 2003].

Capacidad de ser analizado: La capacidad del producto de software para atenerse a diagnósticos de deficiencias o causas de fallas en el sistema o la identificación de las partes a ser modificadas, por lo cual se fue analizando e identificando en el proceso de construcción del sistema las posibles fallas luego identificadas para poder ser modificadas.

Cambiabilidad: La capacidad del software para permitir que una determinada modificación sea implementada. En el proceso de desarrollo del sistema se fue realizando diferentes modificaciones, hasta que el sistema tome la madurez adecuada en cada etapa.

Estabilidad: La capacidad del producto de software para evitar efectos inesperados debido a modificaciones del software. Es necesario que el sistema tenga estabilidad por las modificaciones que se hicieron por esto fue necesario tomar en cuenta la estabilidad del sistema para que pueda adecuarse a los cambios evitando efectos inesperados.

Tomando en cuenta los anteriores conceptos, se observa que el software fue tomando la madures adecuada para tener un buen producto. En este sentido se tomo en cuenta el estándar IEEE 9824-1998 [IEEE94] sugiere un índice de madurez de software (IMS) que proporciona una indicación de estabilidad de un producto software (basado en los cambios que ocurren con cada versión del producto)[PRESSMAN, 2003].

La ecuación que nos ayudara a calcular el índice de madurez es la siguiente:

$$IMS = [M_T - (F_a + F_o + F_d)] / M_q$$

Dónde:

M_T = Numero de módulos de la versión actual.

F_a = Numero de módulos en la versión actual que sean cambiado.

F_o = Numero de módulos en la versión actual que sean añadido.

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

4.4. PORTABILIDAD

El esfuerzo de transportar o migrar un producto de una configuración hardware y/o software a otro está referido a las siguientes características.

Adaptabilidad:

La capacidad del producto de software para ser adaptado a diferentes entornos especificados sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerando. En cuando a la plataforma el sistema está desarrollado para el sistema operativo Microsoft Windows, como ser Windows Xp, Windows 7 con el Servi Pack 2. Tener instalado el navegador Explorer o Mozilla para poder ver el sistema a través de Internet.

Facilidad de Instalación:

La capacidad del producto de software para ser instalado en un ambiente especificado. En cuanto al sistema este se realizó en PHP, el cual permite que se tenga un acceso rápido a la aplicación, por cuanto no necesita de instalación pues solo se accede mediante Internet Explorer o una dirección (Servidos de Aplicaciones). En

cuanto a las restricciones, se cuenta con Internet Explorer 6.0 o versiones superiores para tener acceso a la red de área de la institución

Coexistencia:

La capacidad del producto de software para coexistir con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes. El sistema coexistente con otros sistemas dentro de un mismo entorno y comparte ciertos recursos como Base de datos de otro sistema.



CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Al terminar el presente proyecto, se observó que se realizaron todas las actividades propuestas al inicio y las conclusiones a las que se llegaron son las siguientes:

- ✓ A Partir de los requerimientos que se obtuvieron se pudieron identificar los problemas del Departamento de Contabilidad de los cuales se propusieron soluciones factibles que permitieron resolver los problemas planteados.
- ✓ Se logró modernizar las actividades más importantes en el almacén de la empresa como ser el caso de un buen control de inventario de mercadería.
- ✓ La información que se maneja es más detallada y rápida al realizar las diferentes consultas que se generan en los reportes del movimiento de mercadería.
- ✓ Acortar el tiempo de búsqueda, acceso y suministro de mercadería en almacén.
- ✓ Información precisa y confiable
- ✓ Control de stock mínimo de la mercadería en almacén.
- ✓ Interfaz amigable para el usuario con pantallas comprensibles y de fácil manejo.

5.2. RECOMENDACIONES

Con la finalización del presente proyecto se puede efectuar las siguientes recomendaciones:

- ✓ Realizar el mantenimiento del software del sistema de manera práctica periódica, para su buen funcionamiento.

- ✓ Para futuros trabajos desarrollar un módulo que integre el sistema actual con el área de control de personal lo cual beneficiara de manera positiva a la empresa.
- ✓ Emplear los manuales ante cualquier duda con respecto a la instalación y uso del sistema.
- ✓ Realizar copias de seguridad periódicas para evitar perdida de información frente a cualquier incidente.
- ✓ Prohibir el ingreso a personas ajenas al almacén
- ✓ Tener sumo cuidado respecto a las claves de acceso.



REFERENCIAS BIBLIOGRAFICAS

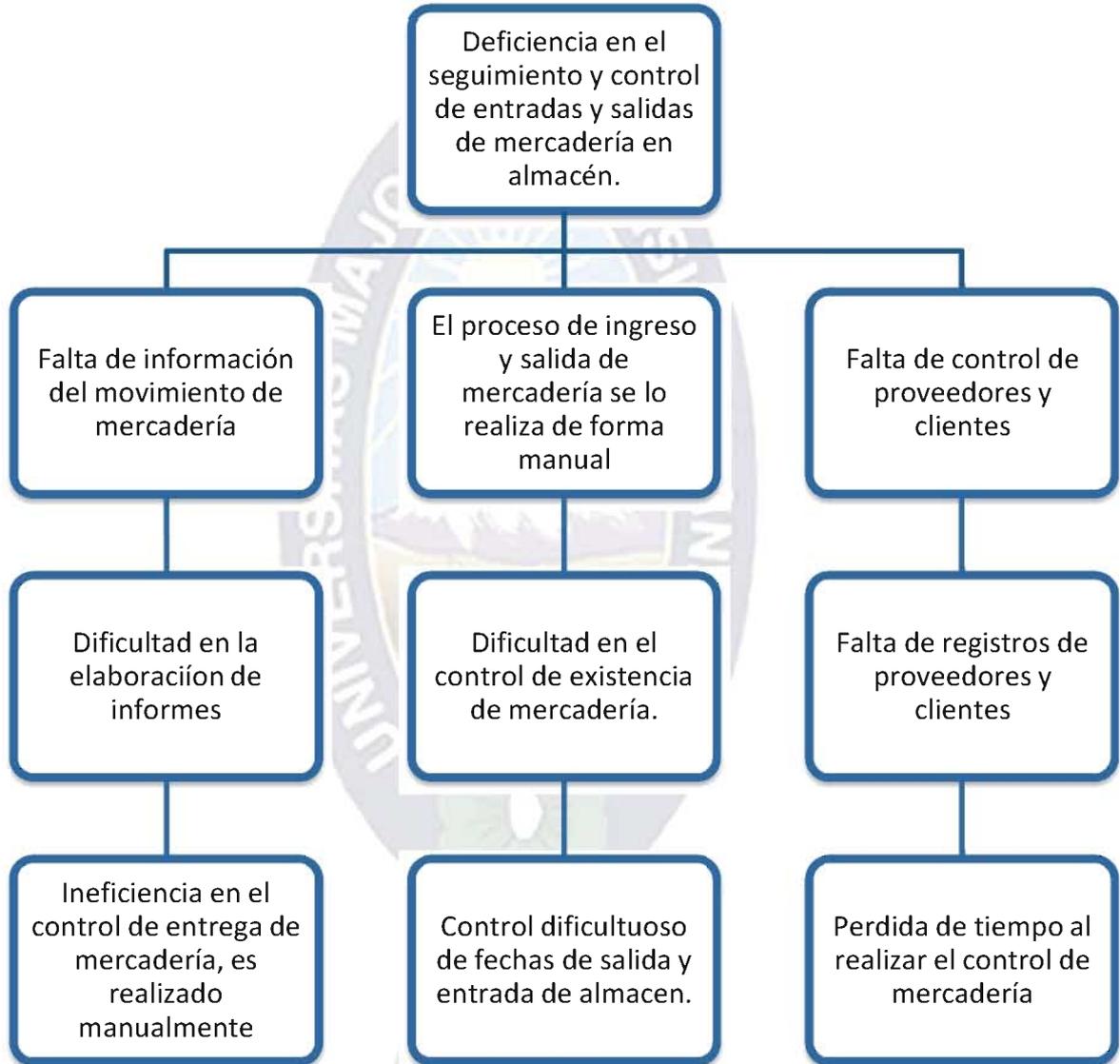
- [SCHUMULLER, 2001]** Josehp Schumuller, Aprendiendo UML en 24 Horas, Pag. 230, Primera Edición, Prentice Hall, 2001
- [LARMAN, 1999]** Larman, C., UML y patrones, Introducción al análisis de diseño orientado a objetos, Pag. 506, Prentice-Hall, Hispanoamérica 1999.
- [JACOB, 1999]** Jacobson/Booch/Rumbaugh, The Unified Software Development Process Pag. 210, Addison-Wesley 2000
- [PRESSMAN, 2005]** Pressman Rogers, 2003, Ingeniería de Software un enfoque Práctico Pag. 17, quinta edición MCGRAW-HILL Interamericana de España, Madrid España
- [FOWLER, 1999]** Martin Fowler, 1999, UML, gota a gota Pag. 22, tercera edición, Addison Wesley, Longman, de México S.A, DE C.V.
- [KEN, 1998]** Kendal & Kendal, 1998, Analisis y Diseño de Sistemas, Pag. 65, segunda Edición, Mexico
- [SCHWABE, 1998]** Schwabe D., "OOHDM-WEB: Rapid Prototyping of Hypermedia Applications in the WWW"
- [MAMANI, 2007]** Mamani Mónica: "Refinamiento Del Método De Diseño De Hipermedia Orientado A Objetos" , La Paz, Bolivia, 2007. Tesis de Grado (informática). Universidad Mayor de San Andrés, Facultad de Ciencias y Puras y Naturales, Carrera de Informática.

Referencias Web

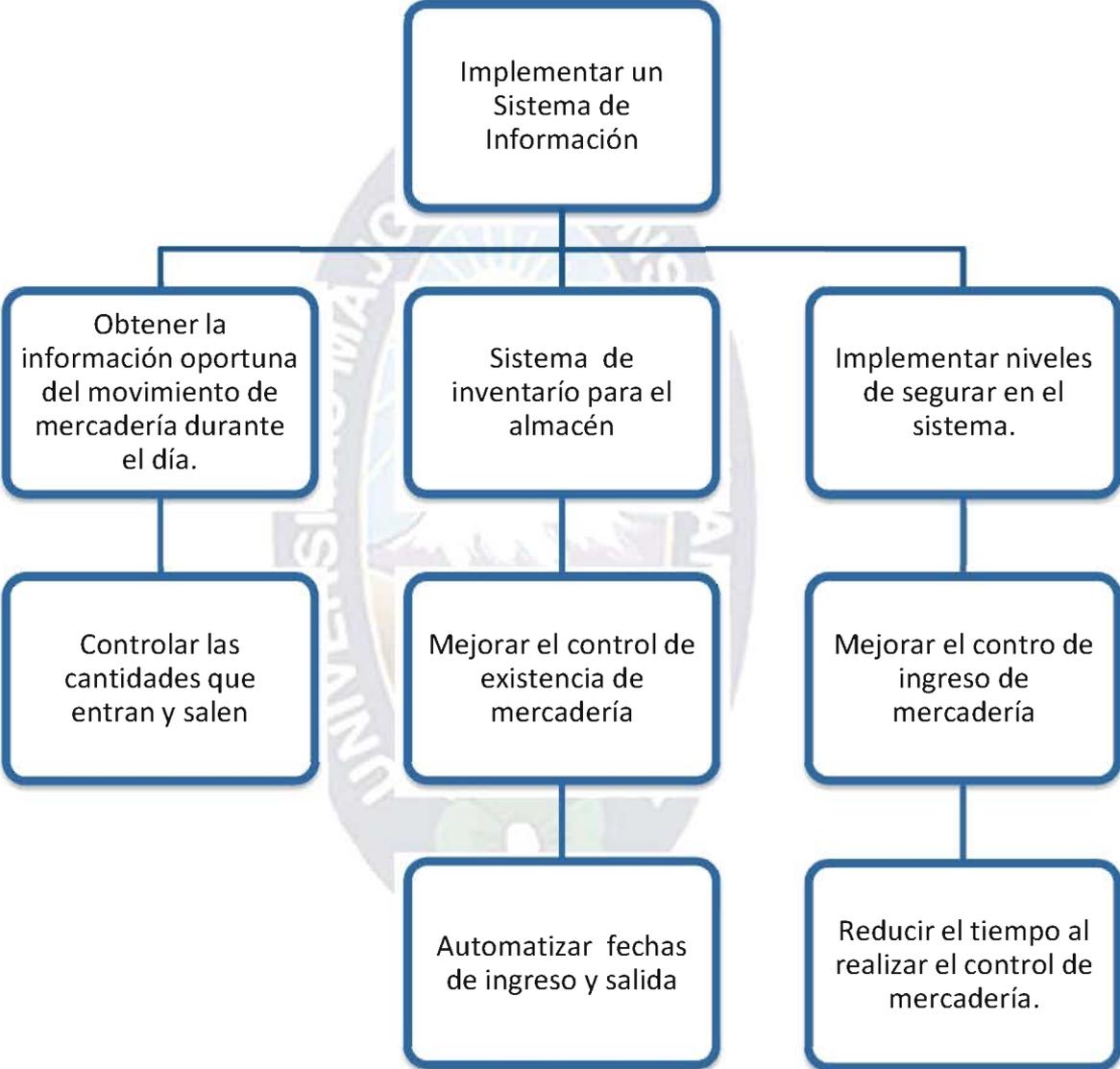
- [METD04]** Metodologías de Desarrollo de Software (2004), Extraído en Octubre 27.2007 Disponible en:
www.informatizate.net/metodologias_de_desarrollo_de_software
- [WIK04]** Introducción al lenguaje Modelado de Unificado ejemplos de UML (2005).
Extraído en Septiembre 15.2007 Disponible en:
http://es.wikipedia.org/wiki/Lenguaje_unificado_de_Modelo
- [iCalle, 2002]** Analisis sobre portales, 2002
http://viait.com.ar/docs/metodologias_de_desarrollo_web.pdf
- [iMartinez, 2002]** Martines Vidrio, Luis Manuel. “Modulo Administrativo de contenidos para portales interactivos web” Universidad de las Américas puebla
http://catarina.uplap.mx:9090/udla/tales/documentos/lis/martinez_vlm/capitulo1.pdf
- [iMegasoft, 2005]** Mega Sofá Server, “Que es un portal de información?”
http://www.megaserv.com/pdf/portal_server.pdf
- [imilemiun, 2007]** Informática Milenium, S.A. de C.V.
<http://www.informaticamilenium.com.mx/paginas/espa%F1ol/sitioweb.htm#dsitio>
- [iine, 2007]** Institutu Nacional, Tecnología Cliente Servidor
<http://www.inei.gob.pe>

ANEXOS

ANEXO A. ARBOL DE PROBLEMAS



ANEXO B. ARBOL DE SOLUCIONES

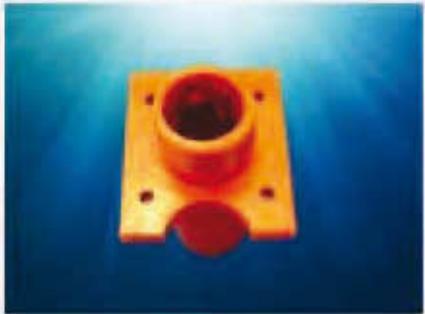


ANEXO C. ALGUNOS PRODUCTOS QUE MANEJA INDUMAR S.R.L.

CLASIFICADOR



ABRAZADERA DE REDUCCIÓN



ASIENTO FARHENWALD



DISPERSER



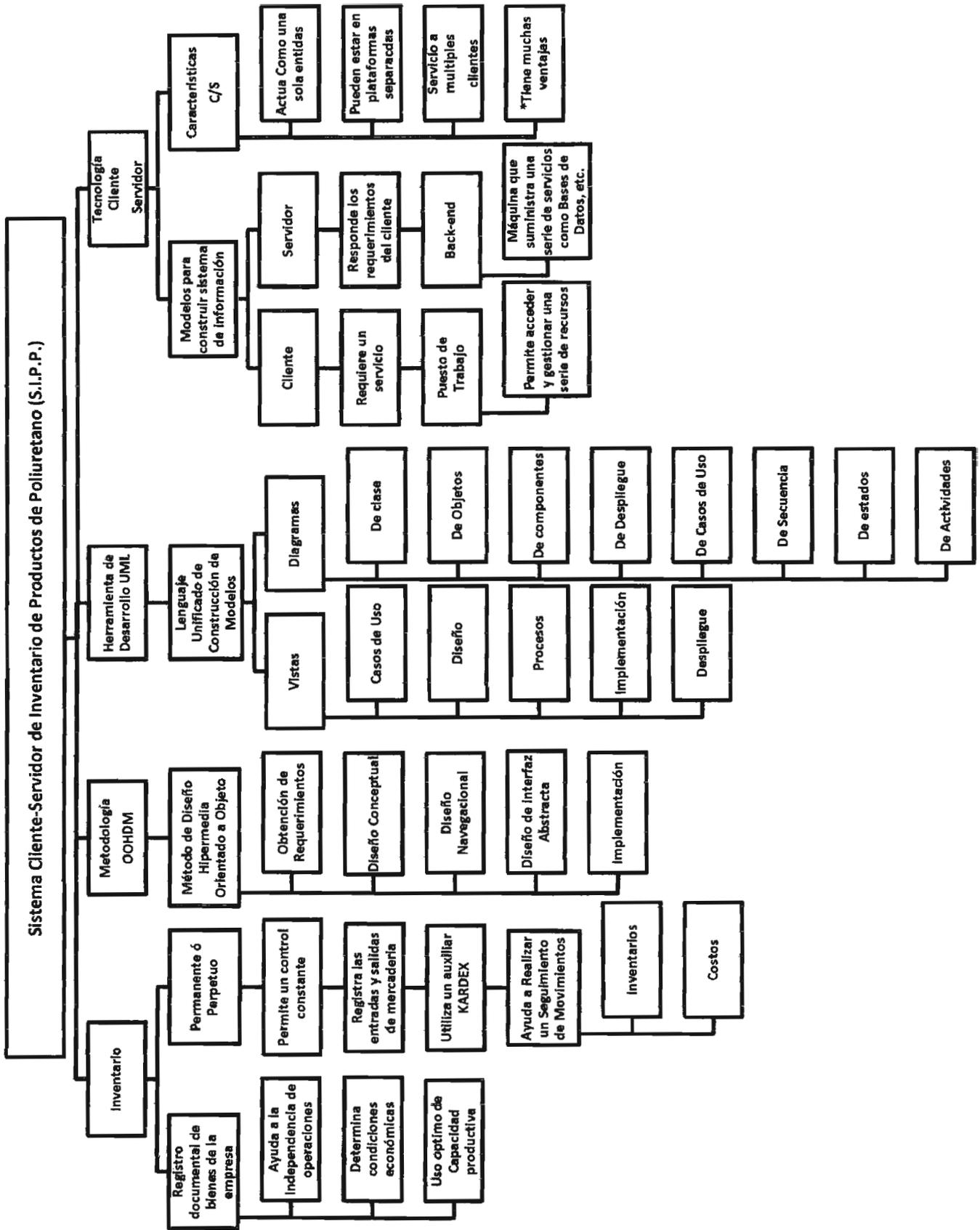
DISPERSER



BOMBA-FLYGT

RODILLOS





La Paz, 23 de Noviembre del 2011

Señora:
Líc. Menfy Morales R.
DIRECTORA DE LA CARRERA DE INFORMÁTICA
FACULTAD DE CIENCIAS PURAS Y NATURALES
UNIVERSIDAD MAYOR DE SAN ANDRÉS
Presente.-

Ref.: **AVAL DE CONCLUSIÓN DE PROYECTO DE GRADO**

Distínguida Licenciada:

Tengo a bien dirigirme a su persona para darle a conocer que luego del seguimiento a la estructura y contenido del Proyecto de Grado titulado "**SISTEMA CLIENTE - SERVIDOR DE INVENTARIO DE PRODUCTOS DE POLIURETANO (INDUMAR S.R.L. INDUSTRIA NACIONAL AL SERVICIO DE LA MINERIA Y LA INDUSTRIA EN GENERAL)**" elaborado por el universitario MAIK LAURÓ CHAMBI MAMANI con C.I. 6136489 L.P., y habiendo el postulante realizado las respectivas correcciones a mis observaciones y no existiendo otra observación más, me corresponde dar mi conformidad y Aval para su respectiva aprobación del presente proyecto de grado, para que el postulante pueda realizar la defensa de su Trabajo de Licenciatura a objeto de optar el Título de Licenciado en Informática.

Es cuanto certifico para fines consiguientes:

Atentamente,



Lic. Germán Huanca Ticona
DOCENTE TUTOR

La Paz, 16 de Noviembre del 2011

Señor:

Lic. German Huanca Ticona

DOCENTE DE LA CARRERA DE INFORMATICA

Presente.-

Ref.: AVAL DE CONCLUSIÓN DE PROYECTO DE GRADO

Mediante la presente tengo bien dirigirme a su persona a objeto de informar la conclusión del proyecto de grado "**SISTEMA CLIENTE – SERVIDOR DE INVENTARIO DE PRODUCTOS DE POLIURETANO INDUMAR S.R.L. (INDUSTRIA NACIONAL AL SERVICIO DE LA MINERIA Y LA INDUSTRIA EN GENERAL)**", desarrollado por el universitario MAIK LAURO CHAMBI MAMANI con C.I. 6136489 L.P., que después de realizar el seguimiento y las correcciones respectivas doy el aval correspondiente para que el estudiante prosiga los trámites correspondientes.

Sin otro particular me despido.

Atentamente,



Lic. Javier Reyes Pacheco

DOCENTE REVISOR

La Paz, 21 de Noviembre del 2011

Señora:
Lic. Menfy Morales
JEFE DE LA CARRERA DE INFORMATICA
FACULTAD DE CIENCIAS PURAS Y NATURALES
UNIVERSIDAD MAYOR DE SAN ANDRES
Presente:

Ref.: AVAL DE PROYECTO DE GRADO

Mediante la presente nos dirigimos a usted para hacerle conocer que la empresa **INDUMAR S.R.L. (INDUSTRIA NACIONAL AL SERVICIO DE LA MINERÍA Y LA INDUSTRIA EN GENERAL)** certifica que el universitario **MAIK LAURO CHAMBI MAMANI** con C.I. 6136489 L.P. ha cumplido con el desarrollo del proyecto de grado denominado **"SISTEMA CLIENTE-SERVIDOR DE INVENTARIO DE PRODUCTOS DE POLIURETANO"**, que actualmente está en funcionamiento con base a la información suministrada por el Departamento De Contabilidad.

Es cuanto se certifica para fines académicos que convengan al interesado.

Atentamente.



S. Raúl Mamani Dindones
JEFATURA CONTABILIDAD
INDUMAR S.R.L.



INDUMAR
PRODUCTOS DE POLIURETANO