

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UNA APLICACIÓN PARA AUTOMATIZAR
LOS MÓDULOS DE COMPRA Y VENTA DE UN SISTEMA
ADMINISTRATIVO PARA UNA EMPRESA BAJO LICENCIA DE
SOFTWARE LIBRE”**

Realizado Por:

PACHECO RAMNARINE, ROMINA HELEN

**Trabajo de Grado Presentado ante la Universidad de Oriente como Requisito
Parcial para Optar al Título de:**

INGENIERO EN COMPUTACIÓN

Puerto La Cruz, Julio de 2008

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UNA APLICACIÓN PARA AUTOMATIZAR
LOS MÓDULOS DE COMPRA Y VENTA DE UN SISTEMA
ADMINISTRATIVO PARA UNA EMPRESA BAJO LICENCIA DE
SOFTWARE LIBRE”**

Asesor Académico:

Prof. Víctor Mújica

**Trabajo de Grado Presentado ante la Universidad de Oriente como Requisito
Parcial para Optar al Título de:**

INGENIERO EN COMPUTACIÓN

Puerto La Cruz, Julio de 2008

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UNA APLICACIÓN PARA AUTOMATIZAR
LOS MÓDULOS DE COMPRA Y VENTA DE UN SISTEMA
ADMINISTRATIVO PARA UNA EMPRESA BAJO LICENCIA DE
SOFTWARE LIBRE”**

Jurado Calificador:

Prof. Víctor Mújica

Asesor Académico

Prof. José Bastardo

Jurado Principal

Prof. Claudio Cortínez

Jurado Principal

Puerto La Cruz, Julio de 2008

RESOLUCIÓN

De acuerdo al Artículo N° 44 del Reglamento de Trabajos de Grado:

“Los trabajos de grado son de exclusiva propiedad de la Universidad De Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo Universitario”.

DEDICATORIA

En primer lugar a Dios, por darme fuerzas, salud y guiarme por el camino que recorro. Por darme la oportunidad de vivir y una familia maravillosa.

A mi madre, Ingrid Dianne, por su gran amor, apoyo e infinita paciencia, te amo mami, gracias por creer en mí; y a mi padre, Hernán Rafael, quien ha sido un pilar fundamental en mi vida, sin ustedes no soy nadie, los adoro!!.

A mis hermanos, Ronald, Rohanna y Rosmery, quienes son mi mayor motivación e inspiración, que aunque hemos pasado momentos difíciles siempre han estado a mi lado apoyándome y brindándome todo su amor.

AGRADECIMIENTOS

A mis padres por enseñarme que no hay límites, que lo que me proponga lo puedo lograr y que solo depende de mí.

A mis hermanos, por haberme acompañado durante todo el desarrollo de esta etapa de mi vida, y a mi cuarta hermana, Shanty, por todo su apoyo, cariño y fidelidad.

A mis amigos los que han pasado y los que han quedado, porque todos ustedes han sido parte de este recorrido, lo han marcado de alguna forma y me han abierto los ojos al mundo; gracias Jaisfel, máxima definición de un amigo; Gaby, por enseñarme a sonreírle a la vida; Héctor Javier, alma libre, siempre al pendiente; José Elías, corazón transparente, me contagias la felicidad de tu alma; Fernando, actitud fuerte ante situaciones complicadas, los adoro..!!

A mi asesor académico Víctor Mújica, por su apoyo y orientación durante el desarrollo de este proyecto, Gracias!

A todos mis profesores y profesoras, que me enseñaron más que números y letras.

A la Universidad de Oriente, por ser la casa donde desarrollé uno de mis mayores logros.

CONTENIDO

RESOLUCIÓN.....	V
DEDICATORIA.....	VI
AGRADECIMIENTOS.....	VII
CONTENIDO.....	VIII
ÍNDICE DE TABLAS.....	XVI
ÍNDICE DE FIGURAS.....	XVII
RESUMEN	XX
INTRODUCCIÓN	1
CAPÍTULO I.....	5
EL PROBLEMA	5
1.1 PLANTEAMIENTO DEL PROBLEMA.....	5
1.2 FORMULACIÓN DEL PROBLEMA	8
1.3 OBJETIVOS DE LA INVESTIGACIÓN	9
1.3.1 <i>Objetivo General</i>	9
1.3.2 <i>Objetivos Específicos</i>	9
1.4 JUSTIFICACIÓN.....	9
1.5 ALCANCE.....	10
1.6 VENTAJAS.....	11
1.7 LIMITACIONES.....	11
CAPÍTULO II	13
MARCO TEÓRICO	13
2.1 ANTECEDENTES DE LA INVESTIGACIÓN	13
2.2 BASES TEÓRICAS	16
2.2.1 <i>Organización</i>	16
2.2.2 <i>Recursos organizacionales</i>	17
2.2.3 <i>Importancia de la Gestión de Compras en una Empresa</i>	18

2.2.4 Innovación Tecnológica en los Procesos de Gestión de Inventarios (Compra-Venta)	20
2.3 TECNOLOGÍA WEB	20
2.3.1 Ventajas	21
2.4 PHP	21
2.4.1 Ventajas de PHP	22
2.5 EL PROCESO RACIONAL UNIFICADO Ó RUP (RATIONAL UNIFIED PROCESS)	22
2.5.1 Historia sobre Rup	22
2.5.2 Características	23
2.5.3 Fases Del Proceso Unificado	24
2.5.3.1 INICIO	24
2.5.3.2 ELABORACIÓN	25
2.5.3.3 CONSTRUCCIÓN	25
2.5.3.4 TRANSICIÓN	25
2.5.4 Disciplinas De Proceso Unificado	26
2.6 LA INGENIERÍA DE SOFTWARE	26
2.6.1 Pasos Del Proceso	27
2.6.1.1 ANÁLISIS DE REQUISITOS	28
2.6.1.2 ESPECIFICACIÓN	28
2.6.1.3 DISEÑO Y ARQUITECTURA	28
2.6.1.4 PROGRAMACIÓN	28
2.6.1.5 PRUEBA	29
2.6.1.6 DOCUMENTACIÓN	29
2.6.1.7 MANTENIMIENTO	29
2.7 GENERALIDADES DE BASES DE DATOS	29
2.7.1 Base De Datos	29
2.7.2 Elementos De Almacenamiento De Una Base Datos	30
2.7.3 Tipos De Base De Datos	30
2.8 APLICACIÓN WEB	31
2.8.1 Historia de la Aplicación Web	31
2.8.2 Lenguajes de Programación Para Aplicaciones Web	32
2.9 UML	33
2.9.1 Diagramas de UML	33
CAPÍTULO III	35
FASE DE INICIO	35

3.1 INTRODUCCIÓN.....	35
3.2 PLANIFICACIÓN DE LA FASE.....	36
3.3 ALCANCE	36
3.4 DESCRIPCIÓN DE LA APLICACIÓN – S2A.	37
3.5 METAS PLANTEADAS	37
3.6 MODELO DE DOMINIO DEL SISTEMA.....	38
3.7 REQUISITOS FUNCIONALES DEL SISTEMA.....	39
3.7.1 Identificación de los Riesgos del Sistema.....	40
3.7.2 Identificación De Actores.....	41
3.7.3 Diagrama de Caso de Uso General ‘S2A’	42
REALIZAR VENTAS	43
CASO DE USO.....	43
VISUALIZAR REPORTES VENTAS	43
3.7.4 Diagrama de Caso de Uso Detallados de ‘S2A’ (Sistema de Administración de Almacén).....	44
3.7.4.1 Descripción del flujo de Sucesos del Caso de Uso ‘Administrar Almacén’	44
3.7.4.2 Descripción del flujo de Sucesos del Caso de Uso ‘Visualizar Resumen de Ventas’	52
3.7.4.3 Descripción del flujo de Sucesos del Caso de Uso ‘Realizar Ventas’	53
3.7.4.4 Descripción del flujo de Sucesos del Caso de Uso ‘Visualizar Reportes Venta’	59
3.8 REQUISITOS NO FUNCIONALES DEL SISTEMA.....	60
3.9 DIAGRAMA DE CLASE DE ANÁLISIS DEL SISTEMA.	60
3.9.1 DESCRIPCIÓN DE LA CLASE DE ANÁLISIS DEL SISTEMA.....	61
3.10 DIAGRAMA DE CLASES DE ANÁLISIS GENERAL DE ‘S2A’ (SISTEMA DE ADMINISTRACIÓN DE ALMACÉN).	63
3.10.1 Casos de Uso ‘Administrar Almacén’ de ‘S2A’ (Sistema de Administración de Almacén).....	64
3.10.2 Casos de Uso ‘Visualizar Reportes Ventas’ de ‘S2A’ (Sistema de Administración de Almacén).....	65
3.10.3 Casos de Uso ‘Realizar Venta’ de ‘S2A’ (Sistema de Administración de Almacén).....	67
3.10.4 Casos de Uso ‘Visualizar Resumen Ventas’ de ‘S2A’ (Sistema de Administración de Almacén).....	68
3.11 DIAGRAMA DE COLABORACIÓN GENERAL DE ‘S2A’ (SISTEMA DE ADMINISTRACIÓN DE ALMACÉN).	69

3.11.1 Diagrama de Colaboración General de ‘S2A’ (Sistema de Administración de Almacén).....	71
3.11.3 Diagrama de Colaboración ‘Visualizar Reportes Ventas’ de ‘S2A’ (Sistema de Administración de Almacén).....	72
3.11.4 Diagrama de Colaboración ‘Administrar Almacén’ de ‘S2A’ (Sistema de Administración de Almacén).....	73
3.11.5 Diagrama de Colaboración ‘Realizar Venta’ de ‘S2A’ (Sistema de Administración de Almacén).....	74
3.11.6 Diagrama de Colaboración ‘Visualizar Resumen Ventas’ de ‘S2A’ (Sistema de Administración de Almacén).....	75
3.12 PAQUETES DE ANÁLISIS.....	75
3.12.1 Identificación de los paquetes de análisis	75
3.13 CONCLUSIÓN DE LA FASE.....	77
CAPÍTULO IV	78
FASE DE ELABORACIÓN.....	78
4.1 INTRODUCCIÓN.....	78
4.2 PLANIFICACIÓN DE LA FASE.....	79
4.3 REQUISITOS DEL SISTEMA	79
4.4 IDENTIFICACIÓN DE LOS CASOS DE USO.....	81
4.5 DIAGRAMA DE CASO DE USO DETALLADOS DE ‘S2A’ (SISTEMA DE ADMINISTRACIÓN DE ALMACÉN)	82
4.5.1 Descripción del flujo de Sucesos del Caso de Uso ‘Registrar Nuevo Pedido’	82
4.5.2 Descripción del flujo de Sucesos del Caso de Uso ‘ Realizar Venta’	86
4.6 DIAGRAMA DE CLASES DE ANÁLISIS DEL SISTEMA.....	93
4.6.1 Casos de Uso ‘Registrar Nuevo Pedido’ de ‘S2A’ (Sistema de Administración de Almacén).....	94
4.6.2 Casos de Uso ‘Realizar Venta’ de ‘S2A’ (Sistema de Administración de Almacén).....	96
4.6.3 Casos de Uso ‘Seleccionar Forma de Pago’ de ‘S2A’ (Sistema de Administración de Almacén).....	98
4.7 DIAGRAMA DE COLABORACIÓN.....	99

4.7.1 Diagrama de Colaboración “Registrar Nuevo Pedido” de “S2A” (Sistema de Administración de Almacén).....	100
4.7.2 Diagrama de Colaboración “Realizar Ventas” de “S2A” (Sistema de Administración de Almacén).....	101
4.7.3 Diagrama de Colaboración “Seleccionar Forma de Pago” de “S2A” (Sistema de Administración de Almacén).....	102
4.8 IDENTIFICACIÓN DE LOS PAQUETES.....	103
4.9 DIAGRAMA DE CAPAS.....	104
4.9.1 Diagrama de Capas de la Aplicación.....	105
4.10 DIAGRAMA DE DESPLIEGUE.....	106
4.11 ANÁLISIS.....	106
4.11.1 Diagrama de clases de diseño.....	107
4.11.2 Diagrama de clases de Diseño “Administrar Almacén”.....	109
4.11.3 Diagrama de clases de Diseño “Realizar Venta”.....	110
4.11.4 Diagrama de clases de Diseño “Visualizar Reportes Venta”.....	111
4.12 DIAGRAMA DE SECUENCIA.....	112
4.12.1 Diagrama de Secuencia de “Modificar Producto”.....	113
4.12.2 Diagrama de Secuencia de “Realizar Venta”.....	115
4.13 DISEÑO.....	116
4.13.1 Diseño de la Base de Datos.....	116
4.13.2 Tablas de Base de datos.....	117
4.13.2.1 Tabla personal.....	117
Personal.....	117
Atributo.....	117
Dominio.....	117
Descripción.....	117
4.13.2.2 Tabla Ventas.....	118
Ventas.....	118
Atributo.....	118
Dominio.....	118
Descripción.....	118
4.13.2.3 Tabla Stock.....	118
Stock.....	119
Atributo.....	119
Dominio.....	119
Descripción.....	119

4.13.2.4	Tabla Grupos.....	119
	Grupos.....	119
	Atributo.....	119
	Dominio.....	119
	Descripción.....	119
4.13.2.5	Tabla Productos.....	119
	Productos.....	120
	Atributo.....	120
	Dominio.....	120
	Descripción.....	120
4.13.2.6	Tabla Lotes.....	120
	Lotes.....	120
	Atributo.....	120
	Dominio.....	120
	Descripción.....	120
4.13.2.7	Tabla niveles.....	120
	Niveles.....	121
	Atributo.....	121
	Dominio.....	121
	Descripción.....	121
4.13.2.8	Tabla Compras.....	121
	Compras.....	121
	Atributo.....	121
	Dominio.....	121
	Descripción.....	121
4.14	CONCLUSIÓN DE LA FASE.....	121
CAPÍTULO V		123
FASE DE CONSTRUCCIÓN.....		123
5.1	INTRODUCCIÓN.....	123
5.2	PLANIFICACIÓN DE LA FASE.....	123
5.3	DISEÑO.....	125
5.4	DISEÑO DE LA INTERFAZ DE USUARIO.....	125
	5.4.1 Registrar Nuevo Producto.....	126
	5.4.1.1 Codificación Registrar Producto.....	127
	5.4.2 Realizar Venta.....	131
	5.4.2.1 Codificación Registrar Producto.....	131

5.5	IMPLEMENTACIÓN	139
5.5.1	Lenguajes de Programación.....	139
5.5.2	Características del lenguaje PHP	140
5.5.3	Características de Html	142
5.5.4	Características de JavaScript.....	142
5.5.5	Hojas de estilo CCS.....	144
5.5.6	Servidor HTTP Apache.....	145
5.5.7	MySQL.....	145
5.6	CODIFICACIÓN DE LOS COMPONENTES DE S2A	147
5.6.1	Diagrama de componentes de S2A.....	147
5.7	FLUJO DE TRABAJO DE PRUEBAS.....	148
5.7.1	Partición Equivalente.....	149
5.7.2	Identificación de las clases de equivalencia.....	149
5.7.3	Grupo de tipos de entrada de Datos.....	149
5.7.4	Aplicación de Casos de Prueba.....	150
5.7.5	Grupo de Tipo de entradas de Datos Modificar Productos	150
5.7.6	Aplicación de Casos de Prueba.....	151
5.8	PRUEBAS	152
5.8.1	Pruebas de Unidad.....	152
5.8.1.1	Administrar Almacén.....	152
5.8.1.2	Realizar Venta.....	152
5.9	CONCLUSIÓN DE LA FASE	154
	CONCLUSIONES.....	155
	RECOMENDACIONES.....	157
	BIBLIOGRAFIA	158
	ANEXOS..... ;ERROR! MARCADOR NO DEFINIDO.	
	ANEXO A: MANUAL DE USUARIO ;ERROR! MARCADOR NO DEFINIDO.	
	METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:	160

ÍNDICE DE TABLAS

TABLA 3. 1 - REQUISITOS FUNCIONALES DEL SISTEMA	39
TABLA 3. 2 - RIESGO DE NO DESARROLLAR UNA ESTRUCTURA ADECUADA.	40
TABLA 3. 3 - RIESGO DE DESARROLLAR INTERFAZ NO ADECUADA.	40
TABLA 3. 4 - RIESGO DE NO IMPLANTAR EL SISTEMA.	41
TABLA 3. 5 - ACTORES IDENTIFICADOS DEL SISTEMA.	41
TABLA 3. 6 - CASO DE USO ADMINISTRAR ALMACÉN.	43
TABLA 3. 7 - CASO DE USO VISUALIZAR RESUMEN DE VENTAS.	43
TABLA 3. 8 - CASO DE USO REALIZAR VENTAS.	43
TABLA 3. 9 - CASO DE USO VISUALIZAR REPORTES VENTAS.	43
TABLA 3. 10 - REQUISITOS NO FUNCIONALES DEL SISTEMA	61
TABLA 4. 1 - DESCRIPCIÓN DE LOS CASOS DE USO.....	81
TABLA 4. 2 - “PERSONAL”	117
TABLA 4. 3 - “VENTAS”	118
TABLA 4. 4 - “STOCK”	118
TABLA 4. 5 - “GRUPOS”	119
TABLA 4. 6 - “PRODUCTOS”	120
TABLA 4. 7 - “LOTES”	120
TABLA 4. 8 - “NIVELES”	121
TABLA 4. 9 - “COMPRAS”	121
TABLA 5. 1 - APLICACIÓN DE LOS CASOS DE PRUEBA.	150
TABLA 5. 2 - APLICACIÓN DE LOS CASOS DE PRUEBA.	151

ÍNDICE DE FIGURAS

FIGURA 2. 1 - FASES Y FLUJOS DE TRABAJO FUNDAMENTALES EN EL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE (FUENTE: JACOBSON, 2000).....	24
FIGURA 3. 1 - DIAGRAMA DE LOS FLUJOS DE TRABAJO SOBRE CADA UNA DE LAS FASES (RESALTANDO LA FASE DE INICIO).....	35
FIGURA 3. 2 - MODELO DE DOMINIO DEL SISTEMA	39
FIGURA 3. 3 - CASO DE USO GENERAL DE LA APLICACIÓN S2A	42
FIGURA 3. 4 - CASO DE USO ADMINISTRAR ALMACÉN.....	44
FIGURA 3. 5 - CASO DE USO “REGISTRAR NUEVO PRODUCTO”	47
FIGURA 3. 6 - CASO DE USO “MODIFICAR PRODUCTO”	49
FIGURA 3. 7 - CASO DE USO “ELIMINAR PRODUCTO”	51
FIGURA 3. 8 - CASO DE USO “VISUALIZAR RESUMEN VENTAS”	52
FIGURA 3. 9 - CASO DE USO “REALIZAR VENTAS”	53
FIGURA 3. 10 - CASO DE USO “VISUALIZAR REPORTES VENTA”	59
FIGURA 3. 11 - CLASE DE ANÁLISIS GENERAL	63
FIGURA 3. 12 - DIAGRAMA DE CLASE DE ANÁLISIS “ADMINISTRAR ALMACÉN”	64
FIGURA 3. 13 - DIAGRAMA DE CLASE DE ANÁLISIS “VISUALIZAR REPORTES VENTAS”	66
FIGURA 3. 14 - DIAGRAMA DE CLASE DE ANÁLISIS “REALIZAR VENTA”	67
FIGURA 3. 15 - DIAGRAMA DE CLASE DE ANÁLISIS “VISUALIZAR RESUMEN VENTAS”	69
FIGURA 3. 16 - DIAGRAMA DE COLABORACIÓN GENERAL DEL SISTEMA.	71
FIGURA 3. 17 - DIAGRAMA DE COLABORACIÓN “VISUALIZAR REPORTES VENTAS”	72
FIGURA 3. 18 - DIAGRAMA DE COLABORACIÓN “ADMINISTRACIÓN DE ALMACÉN”	73
FIGURA 3. 19 - DIAGRAMA DE COLABORACIÓN “REALIZAR VENTA”	74
FIGURA 3. 20 - DIAGRAMA DE COLABORACIÓN “VISUALIZAR RESUMEN VENTAS”	75
FIGURA 3. 21 - DIAGRAMA DE PAQUETE DE ANÁLISIS GENERAL DEL SISTEMA.	76
FIGURA 3. 22 - PAQUETE DE ANÁLISIS ADMINISTRAR ALMACÉN.....	76
FIGURA 3. 23 - PAQUETE DE ANÁLISIS VISUALIZAR REPORTES VENTAS.....	76
FIGURA 3. 24 - PAQUETE DE ANÁLISIS REALIZAR VENTA	77
FIGURA 3. 25 - PAQUETE DE ANÁLISIS VISUALIZAR RESUMEN VENTAS	77

FIGURA 4. 1 - DIAGRAMA DE LOS FLUJOS DE TRABAJO SOBRE CADA UNA DE LAS FASES (RESALTANDO LA FASE DE ELABORACIÓN)	78
FIGURA 4. 2 - CASO DE USO “GENERAL DE S2A”	80
FIGURA 4. 3 - CASO DE USO “REGISTRAR NUEVO PEDIDO”	82
FIGURA 4. 4 - CASO DE USO “REALIZAR VENTA”	86
FIGURA 4. 5 - CASO DE USO “SELECCIONAR FORMA DE PAGO”	91
FIGURA 4. 6 - DIAGRAMA DE CLASE DE ANÁLISIS “REGISTRAR NUEVO PEDIDO”	95
FIGURA 4. 7 - DIAGRAMA DE CLASE DE ANÁLISIS “REALIZAR VENTA”	97
FIGURA 4. 8 - DIAGRAMA DE CLASE DE ANÁLISIS “SELECCIONAR FORMAS DE PAGO”	98
FIGURA 4. 9 - DIAGRAMA DE COLABORACIÓN “REGISTRAR NUEVO PEDIDO”	100
FIGURA 4. 10 - DIAGRAMA DE COLABORACIÓN “REALIZA VENTA”	101
FIGURA 4. 11 - DIAGRAMA DE COLABORACIÓN “SELECCIONAR FORMA DE PAGO”	102
FIGURA 4. 12 - PAQUETE DE ANÁLISIS “REGISTRAR NUEVO PEDIDO”	103
FIGURA 4. 13 - PAQUETE DE ANÁLISIS “SELECCIONAR FORMA DE PAGO”	103
FIGURA 4. 14 - DIAGRAMA DE CAPAS DE LA APLICACIÓN	105
FIGURA 4. 15 - DIAGRAMA DE DESPLIEGUE DEL SISTEMA “S2A”	106
FIGURA 4. 16 - DIAGRAMA DE CLASES DE DISEÑO GENERAL	107
FIGURA 4. 17 - DIAGRAMA DE CLASES DE DISEÑO “ADMINISTRAR ALMACÉN”	109
FIGURA 4. 18 - DIAGRAMA DE CLASES DE DISEÑO “REALIZAR VENTA”	110
FIGURA 4. 19 - DIAGRAMA DE CLASES DE DISEÑO “VISUALIZAR REPORTES VENTA”	111
FIGURA 4. 20 - DIAGRAMA DE SECUENCIA “MODIFICAR PRODUCTO”	114
FIGURA 4. 21 - DIAGRAMA DE SECUENCIA “REALIZAR VENTA”	115
FIGURA 5. 1 - DIAGRAMA DE LOS FLUJOS DE TRABAJO SOBRE CADA UNA DE LAS FASES (RESALTANDO LA FASE DE CONSTRUCCIÓN)	124
FIGURA 5. 2 - REGISTRAR PRODUCTO	126
FIGURA 5. 3 - INTERFAZ REALIZAR VENTA	131
FIGURA 5. 4 - DIAGRAMA DE COMPONENTES DE LA APLICACIÓN	148
FIGURA 5. 5 - PRUEBAS ADMINISTRAR ALMACÉN.....	153
FIGURA 5. 6 - PRUEBAS REALIZAR VENTA	153
FIGURA A. 1 - INTERFAZ DE AUTENTIFICACIÓN	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 2 - MENSAJE DE ERROR DE ENTRADA AL MENÚ.....	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 3 - MENÚ DEL SISTEMA	¡ERROR! MARCADOR NO DEFINIDO.

FIGURA A. 4 - ADMINISTRAR ALMACÉN	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 5 - VER COMPRAS.	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 6 - REGISTRAR PRODUCTO	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 7 - MODIFICAR PRODUCTO	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 8 - ELIMINAR PRODUCTO.....	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 9 - REGISTRAR NUEVO PEDIDO	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 10 - INTERFAZ SELECCIONAR CÓDIGO Y CANTIDAD DE PRODUCTO. ;	¡ERROR! MARCADOR NO DEFINIDO.
DEFINIDO.	
FIGURA A. 11 - INTERFAZ MENSAJE DE ERROR DE CANTIDAD INSUFICIENTE	¡ERROR! MARCADOR NO DEFINIDO.
DEFINIDO.	
FIGURA A. 12 - INTERFAZ AGREGAR O CONFIRMAR VENTA	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 13 - INTERFAZ FACTURAR VENTA	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 14 - INTERFAZ INGRESAR NÚMERO DE CUOTAS.....	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 15 - INTERFAZ FACTURA CON CUOTAS	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA A. 16 - INTERFAZ REPORTES DE VENTAS.....	¡ERROR! MARCADOR NO DEFINIDO.

RESUMEN

La presente investigación es el producto de un estudio minucioso para desarrollar una aplicación informática para automatizar los módulos de compra y venta de un sistema administrativo para una empresa, bajo licencia de software libre. Los resultados de la investigación determinan que los mecanismos de transmisión de datos entre el sistema administrativo y los módulos de compra y venta propuestos requieren de un interfaz adicional para la interacción de los datos y la consolidación contable. Por ende, la investigación se redireccionó a presentar como propuesta de desarrollo de módulos de compra y venta integrada parcialmente y no interactiva con el módulo contable. Esto solo limita la disponibilidad en línea de la información desde otros módulos del sistema, mas puede ser, mediante un mecanismo de actualización y consolidación de datos, en una frecuencia diaria, el cruce de los datos con el módulo contable. Es de hacer notar que el sistema si accede al inventario pues los datos del mismo se encuentran en una base de datos de acceso simple mediante el lenguaje de base de datos SQL y en este caso las transacciones de compra y venta pueden realizarse en línea con la actualización automatizada del inventario.

Descriptores: compra, venta, transacción, software libre. Sistema, módulo.

INTRODUCCIÓN

El uso de las tecnologías de la información, comunicación, electrónica y las telecomunicaciones (TICET), ha permitido a la humanidad mejorar y acelerar las operaciones administrativas y operacionales en las empresas y de todo tipo de instituciones. Los Sistemas de Información (SI) y las Tecnologías de Información (TI) han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas o reducir la ventaja de los rivales.

Las Tecnologías de la Información han sido conceptualizadas como la integración y convergencia de la computación, las telecomunicaciones y la técnica para el procesamiento de datos, donde sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.

Desde hace algunos años las organizaciones han reconocido la importancia de mantener controladas y administradas las transacciones de compra y venta de insumos y productos, ya que en la medida que se conoce con mayor exactitud el flujo comercial de la empresa se pueden tomar decisiones oportunas que minimicen las pérdidas financieras y mejoren la capacidad de respuesta y solución ante la demanda de los clientes. Adicionalmente ofrece mayores garantías de control en cuanto a las compras y/o ventas a crédito, facilitando el flujo de caja auditable en cualquier momento. La información se ha colocado en un buen lugar como uno de los principales recursos que poseen las empresas actualmente. Los entes que se encargan



de las tomas de decisiones han comenzado a comprender que la información no es sólo un subproducto de la conducción empresarial, sino que a la vez alimenta a los negocios y puede ser uno de los tantos factores críticos para la determinación del éxito o fracaso de éstos.

El costo de desarrollo de los sistemas de información es uno de los factores que durante décadas han impedido la masificación de sistemas hechos a medida, y adaptado exactamente a una empresa en específico, esta limitante conllevó a los desarrolladores de software a estandarizar los sistemas de información administrativos mediante módulos orientados a un tipo de gestión específica en el ciclo administrativo y productivo de las empresas, de forma tal que las organizaciones, mediante ajustes menores se adaptasen a los sistemas y no viceversa, lo que conllevó a la administración a transformarse en una ciencia estandarizada.

La proliferación de estos sistemas modulares pudo reducir los altos costos de desarrollo, mediante la filosofía de licenciamiento de uso, que no es más que, por una fracción del costo total de desarrollo, las organizaciones y/o personas, podían disponer de los sistemas y aplicaciones, por un tiempo determinado, o condiciones particulares. Durante un tiempo esto funcionó bien, hasta que los costos de licenciamiento comenzaron a subir, y las empresas desarrolladoras de software recuperaban su inversión exponencialmente, ya que la cantidad de consumidores de sus productos excedía en mucho la cantidad necesaria para recuperar la inversión. Esto llevó a una crisis de consumo en la industria del software, y condujo a las iniciativas de desarrollo de software libre, que no es más que el desarrollo de productos de software sin licenciamiento, bajo el principio de que el conocimiento es libre y las premisas de un el tipo de software que le da al usuario la libertad de usarlo, estudiarlo, modificarlo, mejorarlo, adaptarlo y redistribuirlo, con la única restricción de no agregar ninguna restricción adicional al software modificado, mejorado, adaptado o redistribuido. Vale aclarar que debe permitir el acceso al código fuente,



debido a que ello es una condición imprescindible para ejercer las libertades de estudiarlo, modificarlo, mejorarlo y adaptarlo.

Software libre no implica software gratis, ya que hay costos de asesoramiento, de documentación, de soporte técnico y otros derivados de las normativas de propiedad intelectual.

El presente trabajo de investigación pretende diseñar y desarrollar un módulo para compra y venta de insumos y/o productos que se integre a un sistema administrativo, utilizando la filosofía del software libre. Lo que permite adaptar la herramienta a necesidades particulares y a mejorar la aplicación mediante la práctica organizacional, para adecuar en el tiempo las aplicaciones a las nuevas necesidades y requerimientos que demanden las empresas, en la continua supervivencia del mercado, hoy cada vez mas globalizado.

El Capítulo I, presenta la problemática que da origen a la investigación, aquí se muestran los objetivos que se persiguen alcanzar en la investigación, así como también la justificación del trabajo, así como las limitaciones y el alcance del mismo.

El Capítulo II, muestra la base de sustanciación documental de la presente investigación, las investigaciones precedentes en el entorno de los sistemas administrativos y los módulos de compra/venta, y seguidamente las teorías, preceptos y conceptos relevantes para entender la terminología de este estudio.

En el Capítulo III, se presenta la primera fase del proceso unificado (RUP) Inicio (Inception), donde se hace un plan de fases, se identifican los principales casos de uso y riesgos del sistema a desarrollar, permitiendo esto establecer los requisitos que debe cumplir la aplicación y elaborar el conjuntos de modelos que describirán el comportamiento de la misma.



El Capítulo IV, muestra la segunda fase del proceso unificado de desarrollo de software, conocida como elaboración, el objetivo de esta fase es la definición de la estructura interna de la aplicación el cual proporcionará unas buenas bases para las siguientes fases de diseño e implementación. La definición de la arquitectura debe tener en cuenta los requerimientos obtenidos durante la fase anterior de inicio o concepción.

El Capítulo V, muestra la fase de construcción del RUP, en la que se observan los componentes de software, la integración y las pruebas realizadas luego de la misma.



CAPÍTULO I

EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA.

En los diferentes sectores de la economía, las empresas desempeñan un rol fundamental para la productividad y el desarrollo económico y social. Según su tipo esta participación puede darse en determinado nivel o alcance, es decir, que de acuerdo con sus características, las empresas pueden tener un determinado crecimiento y desempeño o verse condicionadas por factores que limitan su permanencia y consolidación en los mercados locales, regionales, nacionales e internacionales.

En observación diagnóstica realizada por un investigador en distintas empresas, se identificaron una serie de limitaciones en los mecanismos y herramientas tecnológicas que no les permite contar con la información actual y confiable sobre los procesos administrativos, financieros y contables para la toma de decisiones acertadas y en el momento oportuno, evidenciándose la falta de innovación e implementación de herramientas tecnológicas como por ejemplo el uso de programas o software de gestión administrativa.

Existen empresas que por su naturaleza requieren de un control efectivo en las compras y ventas o inventarios de mercancías, equipos, maquinarias o herramientas, resulta indispensable adoptar un sistema de administración apropiado de acuerdo con las nuevas tecnologías informáticas que no sólo cumpla con la función básica de un programa automatizado para la gestión administrativa, sino que además realice funciones de manera integrada, optimizando procesos sin incrementar los costos operativos y de fácil manejo para el personal, es decir, que sea técnicamente viable y



económicamente sostenible, características presentes en la denominada licencia de software libre, de las cuales se conocen varios tipos, sin embargo una de las más empleadas es la compatible con la Licencia Pública General, mejor conocida como licencia de software libre, de acuerdo a sus siglas en inglés, GNU (GNU General Public License) y GPL (GNU Public License).

Una licencia es la autorización formal con carácter contractual que un autor de software otorga a un interesado para ejercer actos de explotación legales. Desde el punto de vista del software libre, existen variantes del concepto de licencia, con el de la GNU donde las libertades están dadas en términos donde el usuario puede utilizar el programa con cualquier propósito, estudiar cómo funciona y adaptarlo a sus necesidades, distribuir copias y mejorar el programa, todas estas libertades implican además que el usuario no tiene que pedir o pagar permisos y además tiene acceso al código fuente, en relación a este último aspecto, es importante señalar que el código o programa fuente o de origen, es el conjunto completo de instrucciones y archivos digitales originales creados o modificados, más todos los archivos digitales de soporte, como tablas de datos, imágenes, especificaciones, documentación, y todo otro elemento que sea necesario para producir el programa ejecutable a partir de ellos. Como excepción, podrán excluirse de este conjunto aquellas herramientas y programas que sean habitualmente distribuidos como software libre por otros medios como, entre otros, compiladores, sistemas operativos y librerías.

De acuerdo con lo antes expuesto, se pueden distinguir las ventajas en el uso de sistemas administrativos bajo licencia de software libre, de modo que también es posible señalar las consecuencias negativas que tendría para la empresa en cuestión no adoptar esta herramienta tecnológica innovadora para el control de compras y ventas, en resumen el posible impacto negativo se caracteriza por la falta de disponibilidad de información actualizada sobre el inventario, datos para la administración financiera y contable, información sobre la facturación para el cálculo



y control de las obligaciones tributarias, limitaciones en la disposición de la información contable completa para realizar auditorias y datos sobre los clientes para mejorar el servicio y garantizar la satisfacción de los mismos, lo cual es un indicador de competitividad.

Dada la importancia del sistema administrativo para algunas empresas industriales en la gestión de compras y ventas y las ventajas que proporciona el uso del software libre en estos programas, se realiza la presente investigación, enmarcada en un tipo y diseño documental, descriptivo, transaccional y proyectivo, con el propósito de desarrollar una aplicación para automatizar los módulos de compra y venta de un sistema administrativo para una empresa bajo la licencia de Software libre. La propuesta se basa en una licencia de software libre compatible con GLP GNU de tipo Copyleft con herramientas de oficina office (open office) y servicios administrativos, con un sistema de control administrativo provisto de un módulo de control administrativo y contable que permita registrar y controlar el presupuesto de gastos.

El problema que se le presenta básicamente es la necesidad que tenga una empresa de controlar el flujo de sus compras y ventas de insumos y/o productos de forma tal que le permita hacer las previsiones necesarias para poder cubrir la demanda de sus clientes minimizando los costos e incrementando la rentabilidad. Además controlar la gestión de sus vendedores, comisiones y proyecciones, para facilitar la toma de decisiones, por medio de una herramienta automatizada que acelera los procesos de la gestión comercial y puede dar resultados de información de manera casi instantánea.



1.2 FORMULACIÓN DEL PROBLEMA

De acuerdo a la situación antes mencionada, incrementa la problemática cuando emergen las siguientes interrogantes:

- ¿Cómo automatizar la gestión de compras y de ventas de una empresa, mediante un módulo de sistema de información de tipo informático, que no impacte considerablemente en los costos de la empresa y pueda integrarse parcial o totalmente a las demás gestiones administrativas vinculantes del sistema de información que da servicios a la empresa?
- ¿Cuál es el riesgo de NO controlar la gestión de compras de manera automatizada?
- ¿Cuál es la capacidad de la empresa para satisfacer oportunamente la demanda de sus clientes?
- ¿Cuál es la rentabilidad en la relación de compra de insumos frente a la venta de productos en la empresa?
- ¿Cuál es el índice de venta por período?
- ¿Cada cuanto tiempo es necesario abastecer el almacén de insumos y productos?
- ¿Podría un módulo de sistema automatizado de compra/venta mejorar el índice de rentabilidad de la empresa?
- ¿Es factible integrar la información de un módulo de compra/venta al sistema de gestión administrativa automatizado existente?



1.3 OBJETIVOS DE LA INVESTIGACIÓN

1.3.1 Objetivo General.

Desarrollar una aplicación para automatizar los módulos de compra y venta de un sistema administrativo para una empresa bajo licencia de software Libre

1.3.2 Objetivos Específicos.

1. Identificar los requisitos necesarios para la elaboración del proyecto.
2. Establecer la arquitectura final necesaria en la elaboración del software en la empresa.
3. Diseñar la base de datos y la interfaz necesaria que permita mostrar una información clara al usuario.
4. Realizar la codificación de los diferentes módulos existentes en el proyecto.
5. Integración de los módulos para depurar y corregir las fallas que puedan ser encontradas en el proceso.
6. Elaborar el manual de usuario que sirva de apoyo para el manejo de la aplicación dentro de la empresa.

1.4 JUSTIFICACIÓN

Es significativo el desarrollo de un módulo de compra/venta para ser integrado a un sistema de información administrativo de una empresa para optimizar la eficiencia de la gestión de compra de insumos y productos, así como la eficacia en la venta controlada de productos en la gestión comercial de la empresa, ya que las buenas prácticas mundiales, durante años, han demostrado una considerable reducción de los costos operacionales en la gestión de compra y aumento sustancial en la disponibilidad de productos en base a la demanda por temporadas. Igualmente,



se facilita la toma de decisiones coyuntural para mejorar los niveles de servicio y plusvalía empresarial.

Por otra parte el uso de software libre, permite a los desarrolladores adaptar la aplicación a la filosofía y cultura de la empresa, reduciendo los impactos en el factor humano, en cuanto a las formas de trabajar, y la productividad. Es importante señalar que este sistema administrativo es más efectivo en la medida que todos y cada uno de los módulos de información están disponibles, ya que el flujo de la información puede viabilizar la disposición de los indicadores de gestión que evalúen el rendimiento y situacionalidad de la empresa de manera oportuna, facilitando la toma de decisiones y conduciendo a la organización a la continua mejora en sus procesos del negocio.

Finalmente, es importante señalar que como investigadora tendré como beneficio el cumplimiento de un requisito trascendental para optar al título de Ingeniero en Computación, en la Universidad de Oriente, Núcleo Anzoátegui.

1.5 ALCANCE

La no disponibilidad de la información contable de forma integrada, limita la capacidad del módulo de compra/venta para consolidar sus datos en todas las gestiones empresariales que la organización requiere. Por esto el alcance de este módulo es sólo contemplar las gestiones de compra y venta de insumos y productos, no contempla facturación, pero provee y recibe la información de la gestión de facturación, no contempla conciliación bancaria ni emisión de cheques, no cruza los datos con los módulos de contabilidad, ni maneja de manera directa el inventario de productos e insumos, sino que le provee y recibe información de éste. Todos los módulos se integran por un interfaz de usuario que cruza los datos por la intervención humana, solo la gestión de compra/venta está totalmente automatizada en el módulo.



El desarrollo del sistema se ejecuta en una empresa en específico, tomando en cuenta elementos de estandarización, para poder replicar la experiencia en cualquier empresa de gestión comercial sobre productos tangibles.

1.6 VENTAJAS

El módulo de compra/venta para un sistema de información automatizado de tipo administrativo tendrá la capacidad de manejar grandes volúmenes de información, solventando uno de los más grandes problemas a los que se enfrenta el personal administrativo, durante la gestión de compras y ventas, además la falta de este tipo de información y acceso oportuno a ella puede afectar negativamente la toma de decisiones acertada. La gran velocidad de procesamiento de un computador usando un sistema automatizado para la compra/venta de insumos y productos permite mejorar los procesos comerciales de la organización y controlar efectivamente el flujo de la información necesaria para satisfacer la demanda de productos, mantener los inventarios nivelados y mejorar la rentabilidad de la empresa.

Las estadísticas emanadas del módulo de compra/venta pueden facilitar la prospectiva de las ventas y prever las compras necesarias, con descuentos considerables por adquisiciones de lotes, que reduzcan la escasez de mercancía durante los períodos picos, y/o cuando los proveedores cierran sus puertas, por períodos vacacionales.

1.7 LIMITACIONES

El acceso a la información de los especialistas de compra y venta en las empresas es limitado, debido a la dedicación que estos tienen en su gestión comercial. El dinamismo de la compra y venta de insumos y productos, prácticamente reduce su tiempo para proveer la información requerida por la analista que desarrolla el sistema,



a su mínima expresión, lo que merma en el tiempo de recopilación de datos, e impacta directamente en la entrega oportuna del módulo.

Por otra parte la disponibilidad de equipos para desarrollo y prueba de sistema es limitante, y es un factor que debe considerarse en el cronograma de ejecución, y en los costos de producción de la herramienta.

Finalmente, la escogencia de una empresa para desarrollar el módulo, depende en principio que ésta tenga en operación un sistema administrativo, que su personal esté dispuesto y disponible para las labores de recopilación de datos, pruebas y puesta en producción. Igualmente, la empresa debe contemplar que, aunque se trata de un módulo en software libre, hay costos asociados y requerimientos de espacio que deben ser cubiertos por ésta. Ya que no se dispone de recursos económicos ni tecnológicos suficientes para llevar a cabo el proyecto.

El tiempo de ejecución del proyecto está limitado por las condiciones académicas de tiempo para la entrega de trabajos de investigación de grado y por el sentido de oportunidad que la empresa tiene de satisfacer su necesidad de incorporar un módulo de compra/venta en su gestión administrativa.

CAPÍTULO II

MARCO TEÓRICO

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Domínguez (2005) llevó a cabo un estudio, titulado: “Software libre”, la investigación tuvo como propósito el análisis de los aspectos relacionados con el software libre, en el cual se describió el significado de un software libre, historia, libertades del software libre, tipos de Licencias, comparación con el software Open Source, culminando con la significación política. Entre las conclusiones más resaltantes presentadas por el autor, se destacan las siguientes:

- El software libre es un término que una vez obtenido puede ser copiado, estudiado y modificado y redistribuido libremente. Este está disponible gratuitamente en Internet entre los años 60 y 70 era considerado un añadido en vez de un producto para los grandes vendedores de computadoras. Dicho proyecto fue trabajado por Richard Stallmon en 1984, el cual llevaría por nombre GNU.
- El software cuenta con tres libertades las cuales son: libertad "0", "1", "2" y "3". Se puede decir que el software libre viene a ser de gran ayuda a las personas para el estudio ya que está accesible en Internet sin ninguna restricción.

Ríos, Alcocer y Oller (2005) realizaron un estudio, titulado “Desarrollo de una plataforma de VoIP basada en Software Libre”, la investigación proporciona una serie de indicaciones tanto para el desarrollo de una plataforma de VoIP, mediante el uso de herramientas de software libre, como para la implementación de servicios de valor añadido en un ambiente avanzado de telefonía y servicios multimedia en general. La plataforma de VoIP que se propone es abierta, flexible, escalable y está



basada en librerías abiertas. Además sigue el principio de que todo servicio multimedia puede ser accesible mediante el protocolo SIP, sea cual sea la naturaleza del servicio, como por ejemplo videoconferencia o streaming.

El principal objetivo de la investigación fue presentar una solución que pueda servir para mejorar la infraestructura telefónica de un usuario en un entorno residencial o de una PYME, proporcionando nuevos servicios, como lo son servicios de centralita como la captura de llamada, transferencia de llamada, etc., con un coste bajo, sin la necesidad de contratar ese servicio con el operador de telecomunicaciones, ni realizar ninguna inversión importante (por ejemplo la compra de una centralita). Los mismos principios se pueden aplicar a operadores de telecomunicaciones o terceras partes que implementen y ofrezcan servicios de valor añadido que serán proporcionados al usuario final.

Los autores concluyeron que La Voz sobre IP (Voice over IP, VoIP) es una tecnología usada para transmitir conversaciones de voz sobre una red de datos usando el protocolo IP (Internet Protocol). Las redes de datos pueden ser: Internet, una Intranet Corporativa, una red manejada por un Operador Local o de Larga Distancia o un Proveedor de Servicio de Internet (ISP).

Sillitti y Succi (2006), realizaron una investigación, la cual se tituló: “Agilidad y desarrollo de Software Libre”. En el desarrollo del estudio los autores plantearon que las metodologías de desarrollo ágiles y el Software Libre son enfoques muy conocidos para el desarrollo de software. Aunque son muy diferentes, presentan muchas concordancias como, por ejemplo, los principios y valores básicos. En particular, hay muchas analogías entre el desarrollo de Software Libre y la programación extrema (enfoque al código e inclusión de cambios).



Este artículo presenta estos principios y valores básicos e identifica las concordancias entre ambas metodologías. Se concluyó que:

- Se observó un nivel considerablemente alto de solapamiento entre los valores adoptados por las AMs (XP en particular) y los de desarrollo de código abierto según Raymond. La comunicación, la retroalimentación y la sencillez son apoyadas completamente por ambos enfoques.
- En cuanto a las prácticas, la situación es claramente diferente. Las prácticas relacionadas con el proceso, el entendimiento compartido y el bienestar del programador son un tanto similares en los dos casos. Las prácticas relacionadas con una cuidadosa retroalimentación no están tan extensamente presentes en la descripción de Raymond.

Velasco y Gotilla (2006) desarrollaron una investigación titulada: “Diseño de un sistema de información bajo licencia de software libre para el manejo de las actividades administrativas de los módulos compra y venta en una pequeña y mediana empresa. El objetivo general fue diseñar un sistema de información bajo licencia de software libre para el manejo de las actividades administrativas de los módulos compra y venta en una pequeña y/o mediana empresa. La investigación fue documental descriptiva, se emplearon técnicas de recolección de datos de fuentes secundarias.

El autor concluyó que: el software libre es un software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Asimismo expone que la licencia GNU GPL posibilita la modificación, predistribución del software, pero únicamente bajo esa misma licencia.



2.2 BASES TEÓRICAS

2.2.1 Organización

La organización incluye reunir y disponer con lógica las personas, equipo y otros recursos necesarios para cumplir un trabajo. Esta íntimamente relacionada con la planeación. Un supervisor que desarrolla la actividad de organización trabaja con los empleados para dividir y ordenar las tareas con el fin de cumplir los objetivos con eficiencia.

Dentro de un enfoque más amplio se entiende que las organizaciones son unidades sociales (o agrupaciones humanas) construidas intencionalmente y reconstruidas para alcanzar objetivos específicos. Esto significa que las organizaciones se proponen y construyen con planeación y se elaboran para conseguir determinados objetivos; así mismo, se reconstruyen y replantean a medida que los objetivos se alcanzan o se descubren medios mejores para alcanzarlos a menor costo y esfuerzo. La organización no es una unidad inmodificable, sino un organismo social vivo sujeto a cambios.

El funcionamiento de las organizaciones debe orientarse hacia la búsqueda de la más alta productividad y eficiencia en el trabajo, a fin de que este último redunde en beneficio de la organización, de cada uno de sus miembros, y del mejoramiento de la calidad de vida del colectivo.

Las empresas son importantes como unidades de producción de bienes y servicios, debido a que desarrollan un menor volumen de actividad, las organizaciones poseen mayor flexibilidad para adaptarse a los cambios del mercado y emprender proyectos innovadores que resultaran una buena fuente generadora de empleo, sobre todo profesionales y demás personal calificado. En la casi totalidad de



las economías de mercado las empresas pequeñas y medianas, incluidos los micro-empresarios, constituyen una parte sustancial de la economía.

2.2.2 Recursos organizacionales

Los recursos son todos aquellos medios con que cuenta la organización para realizar tareas y alcanzar sus metas. En un sentido amplio los recursos son necesarios para elaborar el producto final o el servicio prestado por la organización. Los recursos organizacionales pueden clasificarse en: recursos físicos y materiales, recursos financieros, recursos humanos, recursos mercado-lógicos y recursos administrativos.

- Recursos físicos o materiales: Necesarios para llevar a cabo las operaciones básicas de la empresa; el espacio físico, los terrenos y los edificios, los predios, el proceso productivo, la tecnología que la orienta, los métodos y procesos de trabajo dirigidos a la producción de los bienes y servicios producidos por la organización conforman los recursos materiales.
- Recursos financieros: Están constituidos por el dinero, en forma de capital, el flujo de caja (cuentas por cobrar y cuentas por pagar), los empréstitos, la financiación y los créditos que están a la disposición de manera inmediata o mediata para cumplir con todos aquellos compromisos que adquiere la organización para su funcionamiento.
- Recursos humanos: Constituido por las personas que ingresan, permanecen y participan en la organización, ya sea como parte del nivel jerárquico o ejerciendo una tarea determinada.



- Recursos mercado-lógicos (marketing): Están conformados por todos aquellos medios que las empresas emplean para localizar, contactar e influir en los clientes o usuarios.
- Recursos administrativos: Necesarios para planificar, organizar, dirigir y controlar las actividades empresariales. Estos recursos incluyen los procesos de toma de decisiones, así como los canales de distribución de la información requerida en el funcionamiento y coordinación de la empresa.

2.2.3 Importancia de la Gestión de Compras en una Empresa

Toda empresa lleva a cabo diferentes procesos básicos para su operación. Lo que varía con el tamaño es la complejidad de los procesos y la intensidad de la división de trabajo. La compra de insumo constituye uno de esos procesos, por lo que es importante para la empresa tener claro quien, cómo y según que premisas se compra para evitar “abultar” los inventarios, pues esto puede tener consecuencias inmediatas y visibles.

La necesidad de emplear dinero en la adquisición de inventarios, es una fuente de interacción de la empresa con el ambiente que, eventualmente, conduce a la aparición de unidades o personas encargadas directamente de las compras y, en términos generales, del manejo de los inventarios de materias primas, repuesto o productos terminados. Cuando la calidad y la disponibilidad de la materia prima afectan sensiblemente la eficiencia de las operaciones productivas, puede que sea necesario crear y desarrollar una unidad dedicada a la gestión de compras y a la administración de los inventarios. Esto, por lo general es típico de los procesos de flujo continuo en los que, por el tamaño de la planta y los largos tiempo de puestas a



punto o arranque de las operaciones, es muy costosa una parada de planta por falta de materia prima adecuada, en cantidad y/o en calidad.

Cuando se trata de compras de materias primas estandarizadas y de fácil obtención en el mercado, como láminas de hierro, tuberías, plásticos, cartones y telas, la gestión de compras es bastante sencilla: se limita a disponer de un buen registro de proveedores y establecer convenios sobre las condiciones de pago de los despachos.

Una manera de manejar este tema en las empresas del sector de la construcción y la industria en general donde se ofrecen sus servicios que operan como talleres o por lotes de productos elaborados según previos pedidos, casos éstos en lo que no es aconsejable la posesión física previa de inventarios de materias primas, pero si la ubicación y la relación permanente con fuentes de suministro. La gestión de compra de insumos estandarizados se puede delegar a una persona del área administrativa, siempre y cuando se le adiestre sobre las especificaciones y se le informe oportunamente sobre las necesidades de compra.

En algunos casos, la dependencia de la materia prima es de tal naturaleza que obliga a la propia gerencia de las empresas pequeñas a gestionar directamente las compras y a relacionarse con sus proveedores, particularmente cuando éstos son grandes. En otras ocasiones, el desbalance en el poder negociador con los proveedores aconseja la conformación de cooperativas de consumidores que permitan “agregar” los volúmenes bajos de varios compradores y satisfacer los requerimientos de compras mínimas de las empresas grandes.

Esto es sumamente útil para las pequeñas empresas metalmecánicas que no pueden acudir directamente a las fábricas para conseguir sus pedidos. Igual situación en el caso de las pequeñas empresas de confección en su relación con la industria textil, proveedora de su principal insumo.



2.2.4 Innovación Tecnológica en los Procesos de Gestión de Inventarios (Compra-Venta)

La aplicación de los denominados sistemas administrativos con la ayuda del computador y del software, disponibles a bajos costos, puede ser de utilidad para las empresas que fabriquen una amplia gama de productos, cada uno de los cuales tiene una determinada estructura de materiales.

Si la empresa trabaja según previos pedidos, los inventarios de materias primas y productos semi-determinados deben ser mínimos, no requiriéndose mayores recursos para su manejo (almacenista, por ejemplo). Cuando el proceso productivo es un ensamblaje de piezas, algunas de las cuales se elaboran previamente en el taller y otras se compran, la gestión de compras y almacenamiento pasan a ser crítica, por cuanto puede producir cualquiera de los dos defectos negativos de una mala política de inventarios: o bien no hay disponibilidad de materia prima para lanzar una orden de producción, o bien quedan sin rotar algunos ítems sobre-inventariados. Un buen sistema de pronóstico de ventas y una estrecha coordinación entre los procesos de ventas, producción y compras ayudan a mitigar las consecuencias de las variaciones a las que está expuesta toda empresa.

2.3 TECNOLOGÍA WEB

Se puede considerar el Web como una serie de ficheros de texto y multimedia y otros servicios conectados entre sí por medio de un sistema de documentos hipertexto. Http (Hypertext Transfer Protocol) fue creado en 1990, en el CERN, el European Particle Physics Laboratory, en Ginebra, como medio para compartir los datos científicos a nivel mundial, de manera inmediata y barata. Con hipertexto, una palabra o frase puede contener un enlace con otro texto. Para lograr esto, el CERN



desarrolló un lenguaje de programación llamado HTML, que permite enlazar fácilmente con otras páginas o servicios en el WEB.

Para poder utilizar el servicio Web se necesitan dos partes. Por un lado, la empresa o institución que quiere facilitar su información tiene que crear páginas WWW, siguiendo el estándar definido por el lenguaje HTML, y ponerlas a disposición del público en Internet, en lo que se llama un servidor WWW. Por otro lado, el usuario que quiere acceder a dichas páginas tiene que utilizar un programa (cliente WWW) que lea las páginas WWW e interprete su significado. Estos programas navegadores o clientes WWW son los que permiten a la computadora del usuario interpretar el lenguaje HTML.

2.3.1 Ventajas

Las principales ventajas del servicio WWW son tres. Primera, que puede combinar diferentes tipos de representaciones de la información: texto, audio, video, etc. Segunda, que los hiperenlaces o ligas permiten cargar páginas de cualquier otro servidor conectado a Internet, da igual que esté localizado en cualquier lugar del mundo. Y, tercera, que la creación de páginas WWW es bastante sencilla mediante el lenguaje HTML.

2.4 PHP

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.



2.4.1 Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y Posgresql.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Es libre, por lo que se presenta como alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.

2.5 EL PROCESO RACIONAL UNIFICADO Ó RUP (RATIONAL UNIFIED PROCESS)

El RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

2.5.1 Historia sobre Rup

Sus orígenes se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software es comprada por una compañía sueca



llamada Objectory AB. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory, proceso desarrollado por el fundador de Objectory Ivan Jacobson. El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten.

2.5.2 Características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante. (Ver figura 2.1).

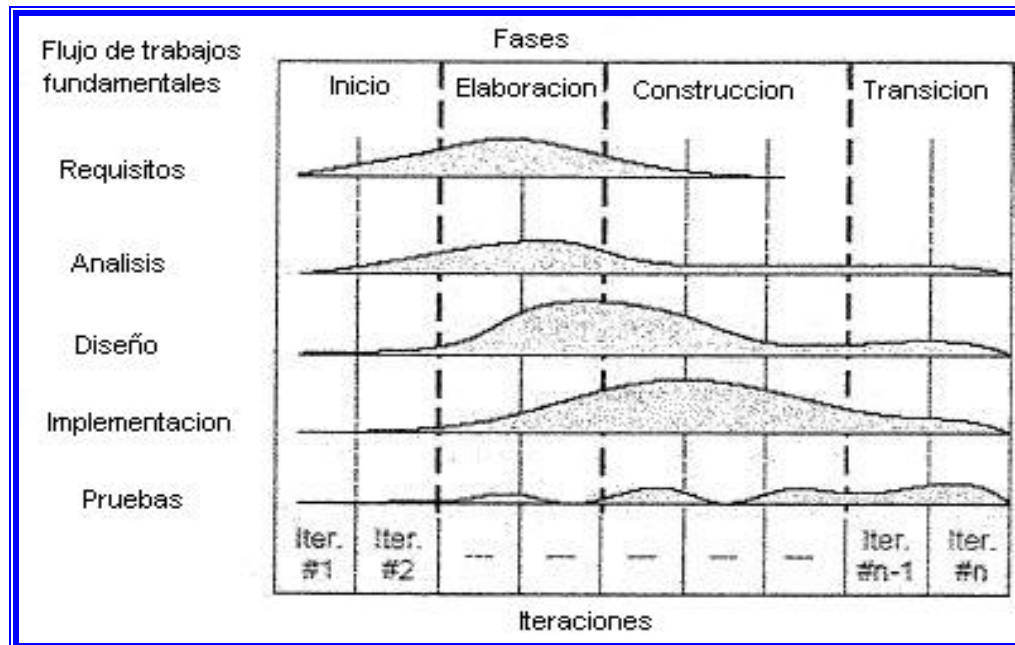


Figura 2. 1 - Fases y flujos de trabajo fundamentales en el ciclo de vida del desarrollo de software (Fuente: Jacobson, 2000)

2.5.3 Fases Del Proceso Unificado

Las cuatro fases del Proceso Unificado son:

- Inicio.
- Elaboración.
- Construcción y
- Transición.

2.5.3.1 INICIO

El objetivo general de esta fase es establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Esta fase es significativamente primaria para el desarrollo del nuevo software, ya que se asegura de identificar los riesgos relacionados con el negocio y requerimientos. Para proyectos de mejora de



software existente esta fase es más breve y se centra en asegurar que vale la pena y es posible, desarrollar el proyecto.

2.5.3.2 ELABORACIÓN

El objetivo en esta fase es establecer la arquitectura base del sistema para proveer las bases estables para el esfuerzo de diseño e implementación en la siguiente fase. La arquitectura debe abarcar todas las consideraciones de mayor importancia de los requerimientos y una evaluación del riesgo.

2.5.3.3 CONSTRUCCIÓN

El objetivo de la fase de construcción es clarificar los requerimientos faltantes y completar el desarrollo del sistema basados en la arquitectura base. Vista de cierta forma esta fase es un proceso de manufactura, en el cual el énfasis se torna hacia la administración de recursos y control de las operaciones para optimizar costos, tiempo y calidad.

2.5.3.4 TRANSICIÓN

Esta fase se enfoca en asegurar que el software esté disponible para sus usuarios, se pueden subdividir en varias iteraciones, además incluye pruebas del producto para poder hacer el entregable del mismo, así como realizar ajuste menores propuestos por el usuario. En este punto, la retroalimentación de los usuarios se centra en depurar el producto, configuraciones, instalación y aspectos sobre utilización.



2.5.4 Disciplinas De Proceso Unificado

Una disciplina es una colección de actividades relacionadas con un área de atención dentro de todo el proyecto. El grupo de actividades que se encuentran dentro de una disciplina principalmente son una ayuda para entender el proyecto desde la perspectiva clásica de cascada.

Las disciplinas son:

- Modelado de Negocios.
- Requerimientos.
- Análisis y Diseño.
- Implementación.
- Pruebas, Transición.
- Configuración y Administración del Cambio.
- Administración de Proyectos y Ambiente.

2.6 LA INGENIERÍA DE SOFTWARE

Es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

El software es el conjunto de instrucciones que permite al hardware de la computadora desempeñar trabajo útil. En las últimas décadas del siglo XX, las reducciones de costo en hardware llevaron a que el software fuera un componente ubicuo de los dispositivos usados por las sociedades industrializadas.



La ingeniería del software es multidisciplinaria. Utiliza las matemáticas para analizar y certificar algoritmos; la ingeniería para estimar costo; la administración para definir requerimientos, evaluar riesgos, administrar personal, etc.

La ingeniería de software se inicia con la necesidad para solucionar las inhabilidades de las organizaciones para predecir el tiempo, esfuerzo y costo para el desarrollo de software y la pobre calidad del software que se producía, además de esto hay otros factores que forzaron el surgimiento de la ingeniería de software como una disciplina.

Los factores que contribuyeron al surgimiento de la ingeniería de software y continúan influenciando en el desarrollo y refinamiento de la ingeniería de software son:

- Cambio en la relación de costos de hardware y software.
- Incremento importante del rol de mantenimiento.
- Avances en Hardware.
- Avances en técnicas de software.
- Incremento de la demanda de Software.
- La demanda de sistemas de software más largos y complejos.

Las metas de la ingeniería de software son: Producir un sistema de software de calidad, a tiempo y a bajo costo.

2.6.1 Pasos Del Proceso

La ingeniería de software requiere llevar a cabo muchas tareas, sobre todo las siguientes:



2.6.1.1 ANÁLISIS DE REQUISITOS

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios.

2.6.1.2 ESPECIFICACIÓN

Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.

2.6.1.3 DISEÑO Y ARQUITECTURA

Se refiere a determinar cómo funcionará de forma general sin entrar en detalles. Yourdon dice que consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc.

2.6.1.4 PROGRAMACIÓN

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga.



2.6.1.5 PRUEBA

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral.

2.6.1.6 DOCUMENTACIÓN

Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema.

2.6.1.7 MANTENIMIENTO

Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o bugs. La mayor parte consiste en extender el sistema para hacer nuevas cosas. De manera similar, alrededor de 2/3 de toda la ingeniería civil, arquitectura y trabajo de construcción es dar mantenimiento.

2.7 GENERALIDADES DE BASES DE DATOS

2.7.1 Base De Datos

De forma sencilla puede indicar que una base de datos no es más que un conjunto de información relacionada que se encuentra agrupada o estructurada.



2.7.2 Elementos De Almacenamiento De Una Base Datos.

Campo: Es la unidad básica de una base de datos.

Registro: Es el conjunto de información referida a una misma persona u objeto.

2.7.3 Tipos De Base De Datos

Las bases de datos se pueden dividir en cuatro tipos básicos:

- Bases de datos de fichero plano (o ficheros por bloques).
- Bases de datos relacionales.
- Bases de datos orientadas a objetos.
- Bases de datos híbridas.

Base de datos de fichero. Las bases de datos de fichero plano consisten en ficheros de texto divididos en filas y columnas. Estas bases de datos son las más primitivas y quizás ni tan siquiera merezcan considerarse como tales. Pueden ser útiles para aplicaciones muy simples, pero no para aplicaciones medianas o complejas, debido a sus grandes limitaciones.

Base de datos relacional. Las bases de datos relacionales son las más populares actualmente. Su nombre proviene de su gran ventaja sobre las bases de datos de fichero plano: la posibilidad de relacionar varias tablas de datos entre sí, compartiendo información y evitando la duplicidad y los problemas que ello conlleva (espacio de almacenamiento y redundancia). Existen numerosas bases de datos relacionales para distintas plataformas (Access, Paradox, Oracle, Sybase,) y son ampliamente utilizadas. Sin embargo, tienen un punto débil: la mayoría de ellas no admite la incorporación de objetos multimedia tales como sonidos, imágenes o animaciones.



Base de datos orientada a objeto. Las bases de datos orientadas a objetos incorporan el paradigma de la Orientación a Objetos (OO) a las bases de datos. La base de datos está constituida por objetos, que pueden ser de muy diversos tipos, y sobre los cuales se encuentran definidas unas operaciones. Las bases de datos orientadas a objetos pueden manejar información binaria (como objetos multimedia) de una forma eficiente. Su limitación suele residir en su especialización, ya que suelen estar diseñadas para un tipo particular de objetos (por ejemplo, una base de datos para un programa de CAD).

Base de datos híbrida. Las bases de datos híbridas combinan características de las bases de datos relacionales y las bases de datos orientadas a objetos. Manejan datos textuales y datos binarios, a los cuales se extienden las posibilidades de consulta. Es una tecnología reciente y aún existen pocas en el mercado.

2.8 APLICACIÓN WEB

Es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

2.8.1 Historia de la Aplicación Web

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes



instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

En contraste, las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores Web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, php para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva

2.8.2 Lenguajes de Programación Para Aplicaciones Web

Existen numerosos lenguajes de programación utilizados para el desarrollo de Aplicaciones Web, entre los que destacan:

- PHP
- ASP/ASP.NET
- JSP
- Perl
- Ruby
- Python

Aunque ciertamente ASP no es un lenguaje de programación, sino una arquitectura de desarrollo Web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET, o VBScript/JScript para ASP).



2.9 UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Racional) pero no especifica en sí mismo qué metodología o proceso usar.

2.9.1 Diagramas de UML

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

En UML hay 13 tipos de diagramas. Para comprenderlos, a veces es útil categorizarlos jerárquicamente,



Diagramas de estructura enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos
- Diagrama de estructura compuesta
- Diagrama de despliegue
- Diagrama de paquetes

Diagramas de comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

Diagramas de Interacción, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación
- Diagrama de tiempos

CAPÍTULO III

FASE DE INICIO

3.1 INTRODUCCIÓN

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema y cada ciclo concluye con una versión del producto para los clientes. Cada ciclo consta de 4 fases: inicio, elaboración, construcción y transición. Cada fase se subdivide en iteraciones y estas a su vez constan de cinco flujos de trabajo fundamentales que son: captura de requisitos, análisis, diseño, implementación y pruebas. (Ver figura 3.1).

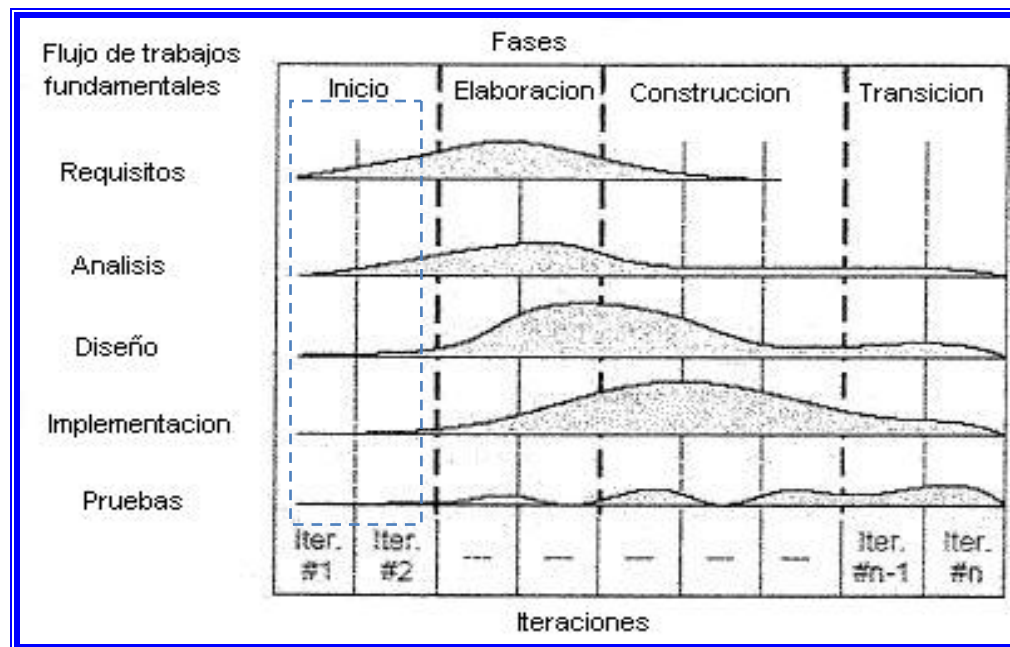


Figura 3. 1 - Diagrama de los flujos de trabajo sobre cada una de las fases (Resaltando la Fase de Inicio)



Durante la fase de inicio, se desarrolla una descripción de cuales sin las principales funciones del sistema para sus usuarios, se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

En la figura 3.1 se puede observar que la fase de Inicio pasa principalmente por los tres primeros flujos de trabajo.

El objetivo principal de esta fase es definir el ámbito del sistema y demostrar la factibilidad del proyecto mediante la consecución de un esbozo de la arquitectura a partir de las especificaciones funcionales que se obtenga de la identificación de los requisitos. Para esto es necesario llevar a cabo los flujos de Requisitos, Análisis y Diseño, al menos en un 50% el primero y en un 10% los 2 últimos.

3.2 PLANIFICACIÓN DE LA FASE

En esta fase se ajustará el proceso unificado al proyecto y se llevará a cabo el estudio necesario para entender el contexto del sistema mediante la elaboración del modelo de dominio y se identificarán los riesgos técnicos, los cuales serán colocados en una lista para su control y posterior mitigación.

Se identificarán actores y casos de uso para la construcción de casos de usos más significativos. Además, se analizarán cada uno de los casos mediante los diagramas de colaboración del modelo de análisis.

3.3 ALCANCE

La aplicación de esta fase permitirá la integración de toda la información generada en las actividades de compra y venta de la empresa, específicamente la parte



de administración de almacén de la misma, desde su inicio hasta su culminación bajo una misma plataforma con una base de datos única empleada para los eventos de las operaciones en una aplicación cliente - servidor.

3.4 DESCRIPCIÓN DE LA APLICACIÓN – S2A.

Esta aplicación es un sistema que permite de forma simple, llevar el control del stock de un negocio, mediante el control de compra y venta de mercancía, resúmenes de venta, elaboración de facturas.

Se sustenta en un sistema de base de datos MySQL el cual debe contar con un administrador encargado de asignar contraseñas, remover contraseñas, etc. El esquema de la base de datos será explicado posteriormente. La interfaz gráfica de la aplicación se basa en un entorno web desarrollado para ser usado de forma óptima bajo Mozilla Firefox, aún cuando es totalmente compatible con Opera y Explorer.

Debido a que se usa conexión con base de datos, es decir se usan páginas dinámicas, se emplea un servidor web, en este caso Apache el cual se ha escogido por ser software de distribución libre al igual que MySQL.

3.5 METAS PLANTEADAS

Desarrollar una aplicación Web de acceso restringido, que satisfaga los requerimientos de los módulos de compra y venta de un sistema administrativo a través del control de almacén de productos permitiendo, sólo al personal autorizado, editar y actualizar registros de la gestión.

La aplicación debe ser:



- Sencilla de usar para todo tipo de usuarios, pero a su vez confiable, segura, rígida y con buen tiempo de respuesta.
- De uso general o multiusuario.
- Agilizar el proceso de compra y venta de productos.
- Lograr rapidez en el análisis y evitar errores en el procesamiento de la información.

3.6 MODELO DE DOMINIO DEL SISTEMA

El sistema será una aplicación Web que proporcionará a los usuarios las herramientas necesarias para registrar las gestiones de compra y venta de la empresa. Se apoyará en la tecnología de multimedia, facilitando el control y almacenamiento de productos de un Stock.

Con el diagrama de dominio se puede establecer y visualizar el alcance de la aplicación y la interacción de las entidades que intervienen en el contexto del desarrollo del sistema. Entre las entidades encontramos:

- Empresa, representa la organización general, en ella encontramos al personal, que son aquellos usuarios encargados de realizar las operaciones y actividades de la empresa dependiendo del nivel que tenga, se presentan las compras y ventas, que son las distintas operaciones que realiza la empresa las cuales son llevadas a cabo por el personal que trabaja en ella, la entidad stock, que lleva el control y registros de nuevos productos al almacén de la empresa, los grupos, que se encargan de la clasificación de los productos según su tipo, y los lotes, que contienen los nuevos pedidos realizados por el personal de la organización. (Ver figura 3.2).

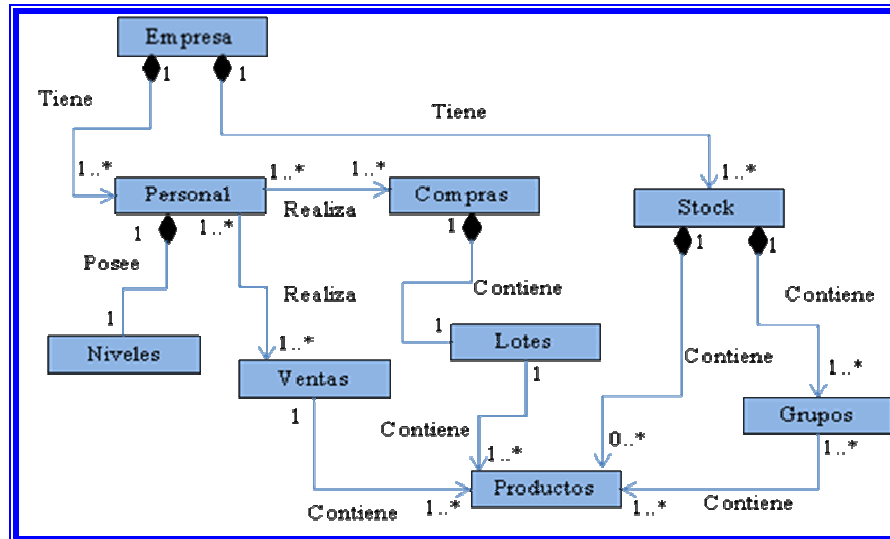


Figura 3. 2 - Modelo de Dominio del Sistema

3.7 REQUISITOS FUNCIONALES DEL SISTEMA

Es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. (Ver tabla 3.1).

Tabla 3. 1- Requisitos Funcionales del sistema

Descripción	Impacto	Acción
Falla de dominio	Software	Insistir en el dominio del contexto
Aplicación de herramientas insuficientes para desarrollar la aplicación	Software	Verificar las capacidades de las herramientas a emplear
Problemas al momento de ingresar al sistema	Software	Estudiar la compatibilidad del sistema con las plataformas usadas, software y hardware
Fallas al acceder a la Base de Datos	Consultas	Realizar un buen diseño de la BD
Interfaz inadecuada a las necesidades del sistema	Software	Profundizar el análisis del flujo de actividades con el apoyo de los futuros usuarios



3.7.1 Identificación de los Riesgos del Sistema

Para que la aplicación de este sistema sea eficiente y confiable deben ser identificados los riesgos, los cuales serán manejados de distintas formas. Siendo las opciones más adecuadas las siguientes: evitarlo, aislarlo, controlarlo y reducir su alcance. Algunos de estos riesgos identificados podrían evitarse tal vez haciendo un replanteamiento, otros aislándolos para que solo afecten una parte. Lo único que se puede hacer para controlarlos es observar mediante pruebas si aparece y en dado caso de que ocurra, se debe tener una ruta alterna. (Ver tablas 3.2, 3.3, 3.4).

En este proyecto fueron identificados los siguientes riesgos:

Tabla 3. 2 - Riesgo de no desarrollar una estructura adecuada.

Riesgo	No desarrollar estructura adecuada
Descripción:	Es necesario desarrollar una arquitectura adecuada que cumpla con los requerimientos funcionales. De no ser así, no se estaría cumpliendo con uno de los objetivos específicos del proyecto.

Tabla 3. 3 - Riesgo de desarrollar interfaz no adecuada.

Riesgo	Desarrollar interfaz no adecuada
Descripción:	El principal objetivo de la aplicación consiste en ayudar a la automatización de la gestión por lo tanto el sistema debería interactuar con el usuario de una forma correcta y amigable. De no ser así, no se estaría cumpliendo con el tercer objetivo y se le podría dar un mal uso a la aplicación no siendo aprovechada al máximo.

**Tabla 3. 4 - Riesgo de no implantar el sistema.**

Riesgo	No implantar el sistema
Descripción:	Se desea que el usuario interactúe con la aplicación, es posible que por el uso de nuevas tecnologías se dificulte la interacción por parte del usuario, esto traería como consecuencia directa que no sea usada.

3.7.2 Identificación De Actores

Aquí se muestra una tabla con los actores principales del sistema identificados:

Tabla 3. 5 - Actores Identificados del Sistema.

ACTOR	DESCRIPCION
Usuario Administrador	Es el responsable de la administración del sistema: Poseerá todos los permisos del sistema. El administrador tiene privilegios para; manejar la información contenida, estar pendiente del buen funcionamiento de la aplicación y modificar datos de ser requerido.
SuperUsuario	Es parte del usuario final de la aplicación. Es el actor principal de la aplicación y se refiere o representa a aquellos usuarios que poseen acceso completo al sistema. El usuario poseerá permisos para agregar, borrar y modificar datos de los registros del sistema.
Usuario	Es el usuario final de la aplicación. Representa al usuario común y solo poseerá permisos para la realización de una de las operaciones del sistema.



3.7.3 Diagrama de Caso de Uso General ‘S2A’

Un diagrama de casos de uso es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un diagrama de caso de uso, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

El siguiente diagrama en la figura 3.3, representa el caso de uso general, desde el punto de vista del usuario de la aplicación.

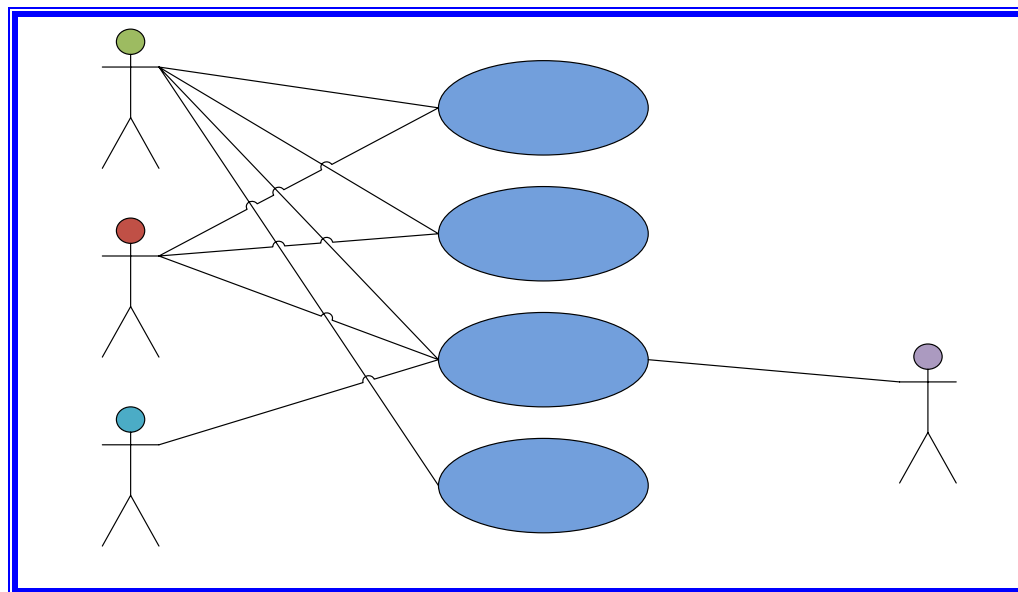


Figura 3. 3 - Caso de uso general de la aplicación S2A



Identificación De Los Casos De Uso:

Tabla 3. 6 - Caso de Uso Administrar Almacén.

CASO DE USO	ADMINISTRAR ALMACÉN
Descripción	Permite a el Súper Usuario declarar nuevos productos en el stock, así como eliminarlos, modificarlos, ingresar nuevos pedidos, entre otros
Actores Implicados	Súper Usuario

Tabla 3. 7 - Caso de Uso Visualizar Resumen de Ventas.

CASO DE USO	VISUALIZAR RESUMEN DE VENTAS
Descripción	Permite visualizar las ventas efectuadas, así como el estado de la venta, si ya fue cancelada o falta parte por cancelar
Actores Implicados	Súper Usuario

Tabla 3. 8 - Caso de Uso Realizar Ventas.

CASO DE USO	REALIZAR VENTAS
Descripción	Permite al usuario y al súper usuario realizar una venta la cual quedará debidamente registrada en la base de datos para posteriores revisiones.
Actores Implicados	Súper Usuario Usuario

Tabla 3. 9 - Caso de Uso Visualizar Reportes Ventas.

CASO DE USO	VISUALIZAR REPORTES VENTAS
Descripción	Permite al súper usuario visualizar a todos los trabajadores de la empresa y las estadísticas de las ventas de cada uno de los usuarios.
Actores Implicados	Súper Usuario



3.7.4 Diagrama de Caso de Uso Detallados de 'S2A' (Sistema de Administración de Almacén)

La representación general por medio de un caso de uso muestra el contexto de trabajo de la aplicación, el tipo de información que se maneja, los actores que intervienen en las operaciones, las interacciones entre los procesos y con los actores.

Para la visualización más detallada de los procesos representados de forma general en el diagrama de contexto (Figura 3.3), se presenta a continuación de forma más detallada la estructura interna de cada proceso a través de su representación en casos de uso:

3.7.4.1 Descripción del flujo de Sucesos del Caso de Uso 'Administrar Almacén'

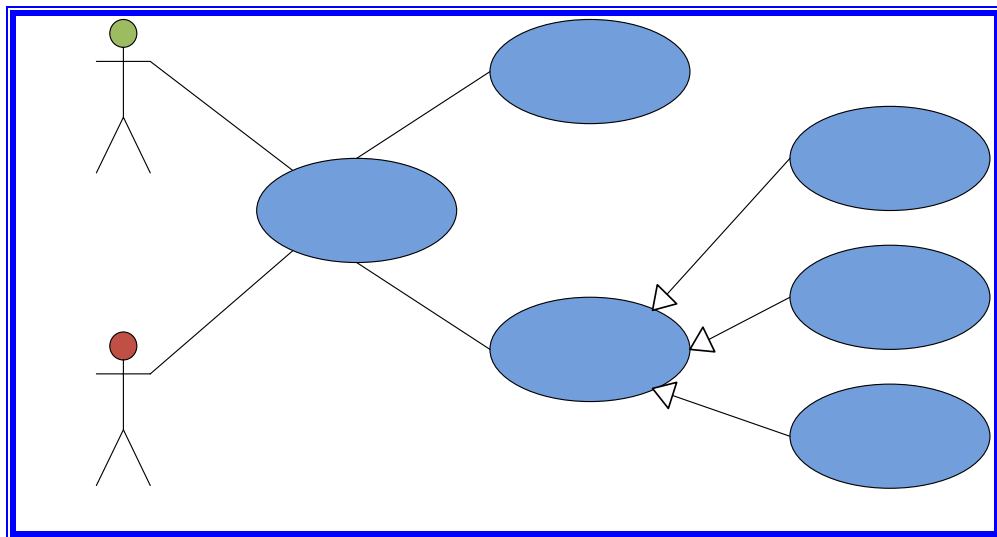


Figura 3. 4 - Caso de Uso Administrar Almacén



Nombre del Caso: Administrar Almacén

Actores: Súper Usuario, Usuario Administrador

Descripción: Permite a el Súper Usuario y al Usuario Administrador declarar nuevos productos en el stock, así como eliminarlos, modificarlos, ingresar nuevos pedidos, entre otros.

Pre-Condición: Este proceso es invocado por el usuario desde el proceso “Menú”, al pulsar la opción o enlace indicado para la ficha

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra la ficha “Administrar Almacén”, en la que se muestran las distintas opciones del proceso, como por ejemplo registrar pedidos, modificar, eliminar, entre otros.
3. El usuario visualiza la información y selecciona la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de seleccionar otra operación o puede salir del sistema.

Nombre del Caso: “Registrar Nuevo Pedido”.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar nuevos pedidos en el stock.

Pre-Condición: El usuario realiza la selección desde el proceso “Administrar Almacén”

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.



2. Se llenan los formularios de registro de producto con la información del mismo, para llevar a cabo el registro.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

Nombre del Caso: "Registrar Nuevo Producto".

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar nuevos productos con sus características específicas, en la base de datos del sistema.

Pre-Condición: El usuario realiza la selección desde el proceso "Administrar Almacén".

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario tendrá que suministrar los datos requeridos para la ejecución del proceso.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

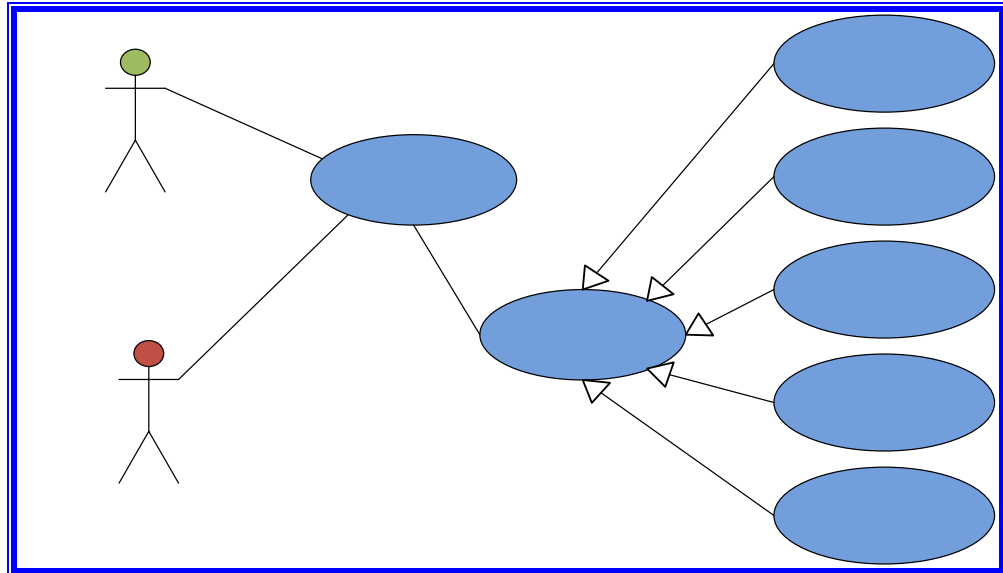


Figura 3. 5 - Caso de Uso "Registrar Nuevo Producto"

Registrar
Producto

Nombre del Caso: "Registrar Nuevo Producto" **Usuario Administrador**

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar nuevos productos con sus características específicas, en la base de datos del sistema.

Pre-Condición: El usuario realiza la selección desde el proceso "Administrar Almacén".

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se muestra un recuadro en el cual el usuario tendrá que suministrar los datos requeridos para la ejecución del proceso. **Súper Usuario**
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alterno:



El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: ''Declarar Nuevo Producto'.

Actores: Súper Usuario, Usuario Administrador, Base de datos.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar nuevos productos, en la base de datos del sistema.

Pre-Condición: El usuario realiza la selección desde el proceso ''Registrar Nuevo Producto''.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario tendrá que suministrar los datos requeridos para la ejecución del proceso, código, grupo de producto, nombre, descripción, entre otros.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

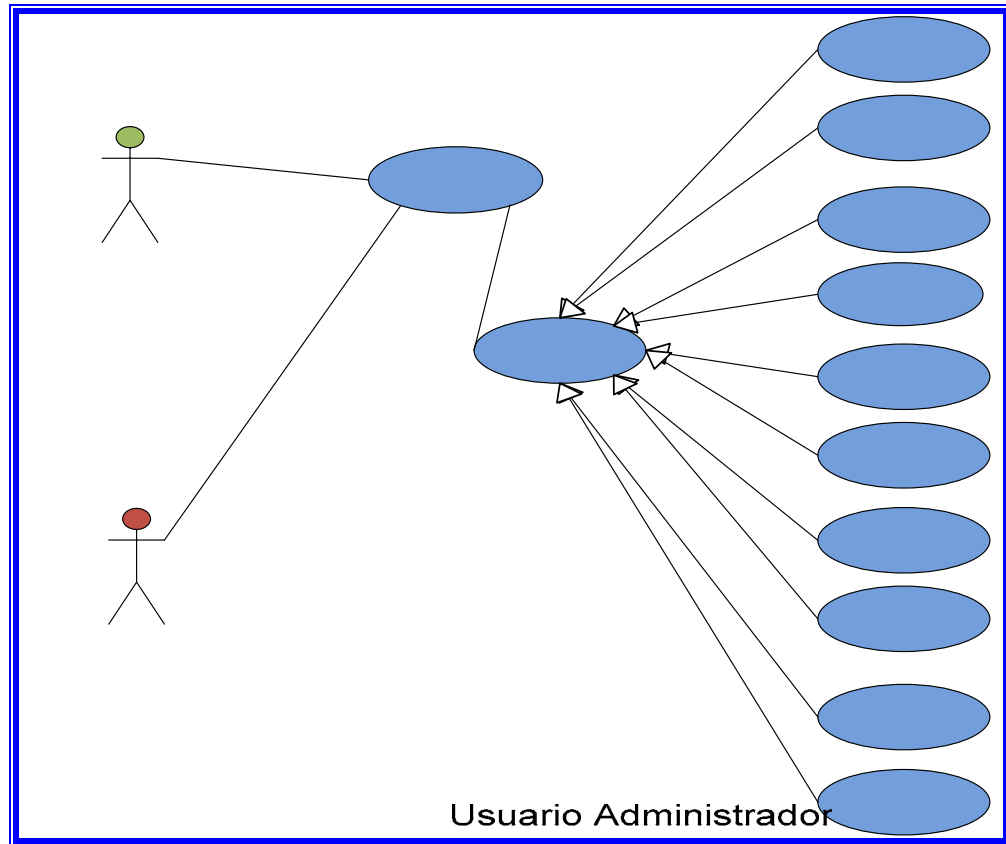


Figura 3. 6 - Caso de Uso 'Modificar Producto'

Nombre del Caso: 'Modificar Producto'.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador modificar productos de la base de datos del sistema.

Pre-Condición: El usuario realiza la selección desde el proceso 'Administrar Almacén'.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.

Súper Usuario

Modific



2. Se muestra una lista con los productos pertenecientes a la base de datos, en la cual el usuario podrá seleccionar uno de ellos, para la ejecución del proceso.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: ''Seleccionar Producto'.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador elegir que producto y que parámetro del producto, desea modificar del inventario.

Pre-Condición: El usuario realiza la selección desde el proceso ''Modificar Producto''.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario podrá seleccionar que parámetro desea cambiar.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

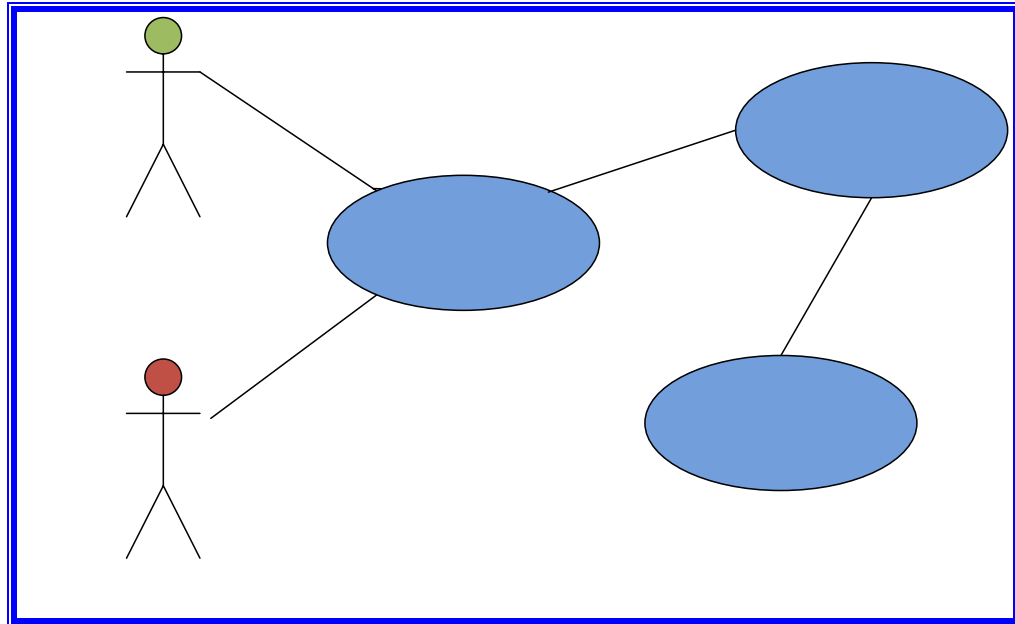


Figura 3. 7 - Caso de Uso ‘Eliminar Producto’

Nombre del Caso: ‘Eliminar Producto’

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador eliminar productos de la base de datos del sistema.

Pre-Condición: El usuario realiza la selección desde el proceso ‘Administrar Almacén’.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra una lista con los productos pertenecientes a la base de datos, en la cual el usuario tendrá que seleccionar uno o varios de ellos, para la ejecución del proceso.
3. El usuario visualiza la información y escoge la operación que desea realizar.

Usuario Administrador

Elimina

Súper Usuario



4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

3.7.4.2 Descripción del flujo de Sucesos del Caso de Uso “Visualizar Resumen de Ventas”

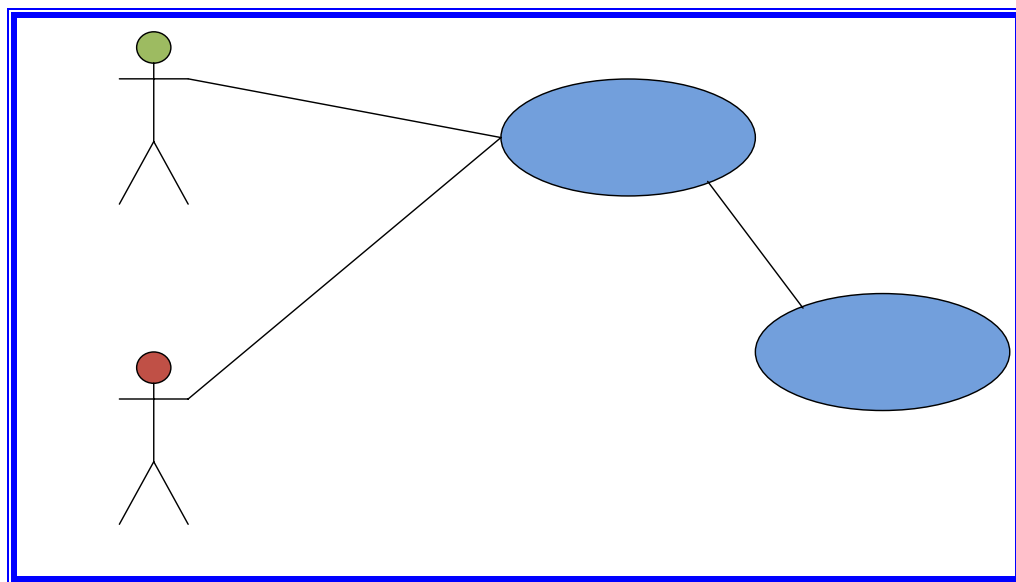


Figura 3. 8 - Caso de Uso “Visualizar Resumen Ventas”

Nombre del Caso: “Visualizar Resumen de Ventas”.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador mostrar el resumen de ventas realizadas por los usuarios de la aplicación.

Pre-Condición: El usuario realiza la selección desde la opción “Menú”

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.

Usuario Administrador



2. Se muestra un recuadro en el cual el usuario podrá visualizar las ventas realizadas por los usuarios.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

3.7.4.3 Descripción del flujo de Sucesos del Caso de Uso “Realizar Ventas”

En el siguiente diagrama se mostrará los casos de uso detallado del caso de uso Realizar Ventas perteneciente al sistema.

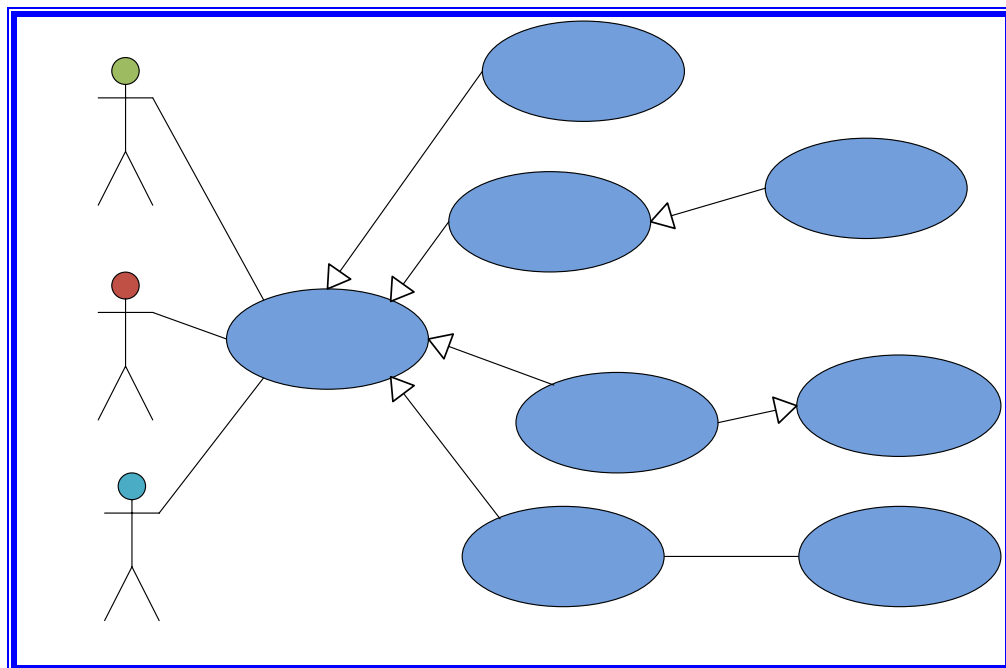


Figura 3.9 - Caso de Uso “Realizar Ventas”



Nombre del Caso: ''Realizar Ventas'.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite al Usuario, al Súper Usuario y al Usuario Administrador realizar alguna de venta de productos disponibles en el stock.

Pre-Condición: El usuario realiza la selección desde la opción ''Menú''

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario podrá seleccionar los productos que desea y la cantidad correspondiente de cada uno.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

Nombre del Caso: ''Seleccionar código de producto'.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite que cualquiera de los usuarios seleccione el código del producto que se va a vender.

Pre-Condición: El usuario realiza la selección desde el proceso ''Realizar Venta''

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario podrá seleccionar o escribir el ó los códigos de los productos involucrados en la venta.
3. El usuario visualiza la información y escoge la operación que desea realizar.



4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

Nombre del Caso: "Seleccionar cantidad de producto".

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite que cualquiera de los usuarios seleccione la cantidad de productos que se van a vender.

Pre-Condición: El usuario realiza la selección desde el proceso "Realizar Venta"

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario podrá marcar la cantidad del ó los productos involucrados en la venta.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Consultar Disponibilidad".

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso es utilizado para consultar la existencia en stock del producto que se va a vender.

Pre-Condición: El usuario realiza la selección desde el proceso "Seleccionar Cantidad Producto"

Flujo de Eventos:

Flujo Principal:



1. El usuario invoca el caso de uso.
2. Es un proceso no visible por el usuario ya que consiste en una búsqueda en la base de datos del sistema, luego de ejecutado se visualiza la información de los productos.
3. El Usuario visualiza información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Agregar productos".

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite que los usuarios agreguen productos a la venta antes de ser facturados para ejecutar la venta ó que confirmen los productos ya seleccionados.

Pre-Condición: El usuario realiza la selección desde el proceso "Realizar Venta"

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario podrá marcar la cantidad del ó los productos involucrados en la venta.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Actualizar Inventario".

Actores: Súper Usuario, Usuario Administrador, Usuario.



Descripción: Este proceso realiza la actualización de los productos restantes en stock una vez finalizada la venta.

Pre-Condición: El usuario realiza la selección desde el proceso “Agregar Productos”

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Es un proceso no visible por el usuario y consiste en la actualización de la base de datos del sistema. luego de ejecutado se visualiza la información correspondiente.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: “ Facturar Venta”.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso realiza la facturación de los productos incluidos en la venta.

Pre-Condición: El usuario realiza la selección desde el proceso “Realizar Venta”

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro en el cual el usuario podrá marcar la opción de facturar si así lo desea.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.



Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Generar Reporte Venta."

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso genera automáticamente según las ventas, las estadísticas mensuales de las mismas.

Pre-Condición: El usuario realiza la selección desde el proceso "Realizar Venta"

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Es un proceso no visible por el usuario, se generan los reportes de ventas según los valores actualizados en la base de datos del sistema.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Generar Reporte Venta."

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso genera automáticamente según las ventas, las estadísticas mensuales de las mismas.

Pre-Condición: El usuario realiza la selección desde el proceso "Realizar Venta"

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.



2. Es un proceso no visible por el usuario, se generan los reportes de ventas según los valores actualizados en la base de datos del sistema.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

3.7.4.4 Descripción del flujo de Sucesos del Caso de Uso “Visualizar Reportes Venta”

En el siguiente diagrama se mostrará los casos de uso detallado del caso de uso Realizar Ventas perteneciente al sistema.

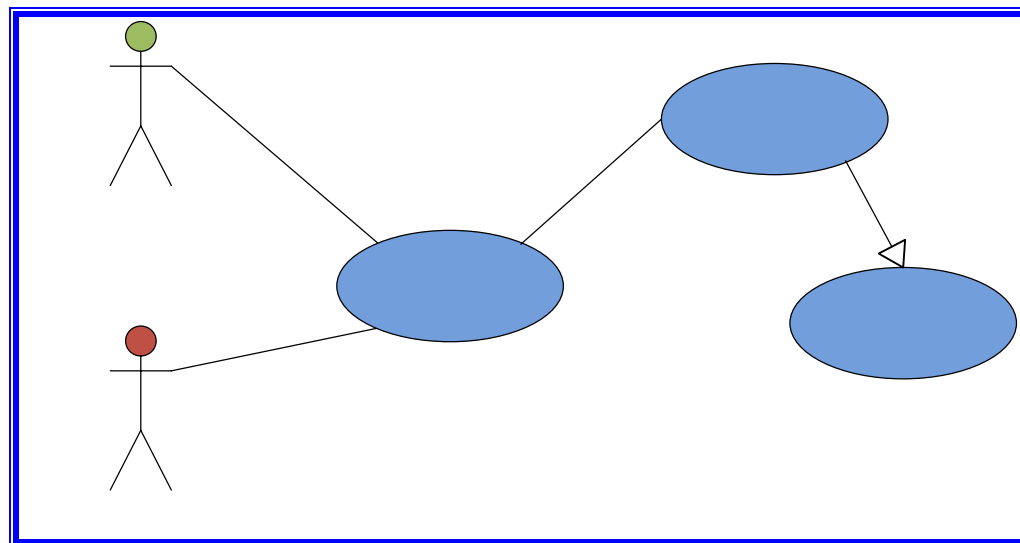


Figura 3. 10 - Caso de Uso “Visualizar Reportes Venta”

Nombre del Caso: “Visualizar Reportes Venta”.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso muestra gráficamente las estadísticas mensuales de las ventas realizadas.



Pre-Condición: El usuario realiza la selección desde el proceso ‘‘Menú’’

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Es un recuadro donde se puede seleccionar el nombre de algún personal de la empresa, donde se mostrará gráficamente las ventas del mes.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

3.8 REQUISITOS NO FUNCIONALES DEL SISTEMA

Especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad, fiabilidad. La fiabilidad hace referencias a características como la disponibilidad, exactitud, tiempo medio entre fallo, defectos por miles de líneas de código, y defectos por clase. Un requisito de rendimiento impone condiciones sobre los requisitos funcionales como la velocidad, rendimiento, tiempo de respuesta, y uso de memoria. La mayoría de los requisitos de rendimientos afectan solo a ciertos casos de uso.

3.9 DIAGRAMA DE CLASE DE ANÁLISIS DEL SISTEMA.

Una vez definidos los casos de uso del sistema, se procedió a la realización de un estudio más detallado de los mismos empleando diagramas de clases de análisis. De esta forma se representó la estructura de la aplicación, luego de un estudio independiente de cada caso de uso.

**Tabla 3. 10 - Requisitos No Funcionales del Sistema**

Requisito	Descripción
Extensible	El software debe permitir la incorporación de nuevas funcionalidades en su estructura
Mantenible	El software debe permitir el mantenimiento sin ninguna complicación
Amigable	El diseño de la interfaz del usuario debe poseer una estructura amigable para que resulte de fácil entendimiento para los usuarios

3.9.1 DESCRIPCIÓN DE LA CLASE DE ANÁLISIS DEL SISTEMA.

La ejecución de los casos de uso ya definidos se inicia con la entrada a la aplicación luego de un “logueo” o validación como trabajador de la empresa, esta entrada se lleva a cabo por medio de una clase de interfaz así como también la entrada a los módulos siguientes a la validación.

Las clases de control son las encargadas de la ejecución y cumplimiento de las solicitudes hechas por los usuarios, tomando la información que se necesite en cada caso de las clases entidad; cuya tarea es modelar la información que se debe almacenar y proporcionar a las clases de control.

El diagrama general de clases de análisis de la aplicación “S2A” muestra una visión general de las interfaces, gestores y entidades que intervienen en el trabajo de la aplicación.

Las interfaces con los trabajadores de la empresa, los gestores de validación como trabajador, los de autenticación como un tipo determinado de usuario y los



encargados de operaciones determinadas propias de “S2A” y que trabajan sobre las entidades de información para la misma.

Entre los gestores se encuentran, el gestor de autenticación, el de administración de almacén, de visualización de venta y reportes y el gestor de realización de ventas.

Se muestran los caminos entre procesos que debe poder seguir el usuario en su interacción con la aplicación y los gestores, entidades e interfaces que intervienen. Luego de la interacción *actor-interfaz*, actúa el gestor de autenticación para validar al usuario y desplegar las opciones posibles a seguir.



3.10 DIAGRAMA DE CLASES DE ANÁLISIS GENERAL DE "S2A" (SISTEMA DE ADMINISTRACIÓN DE ALMACÉN).

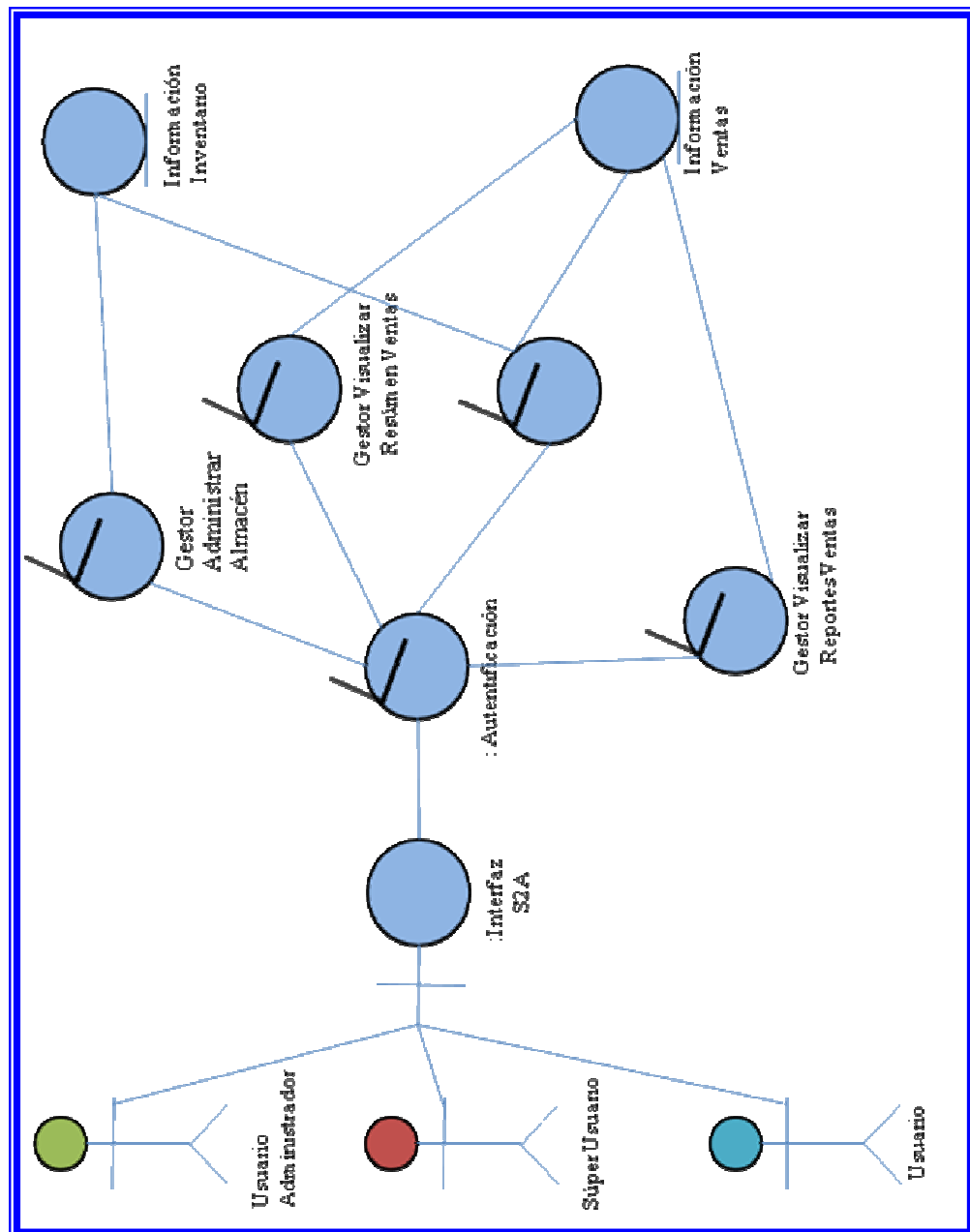


Figura 3. 11 - Clase de Análisis General



3.10.1 Casos de Uso ‘Administrar Almacén’ de ‘S2A’ (Sistema de Administración de Almacén).

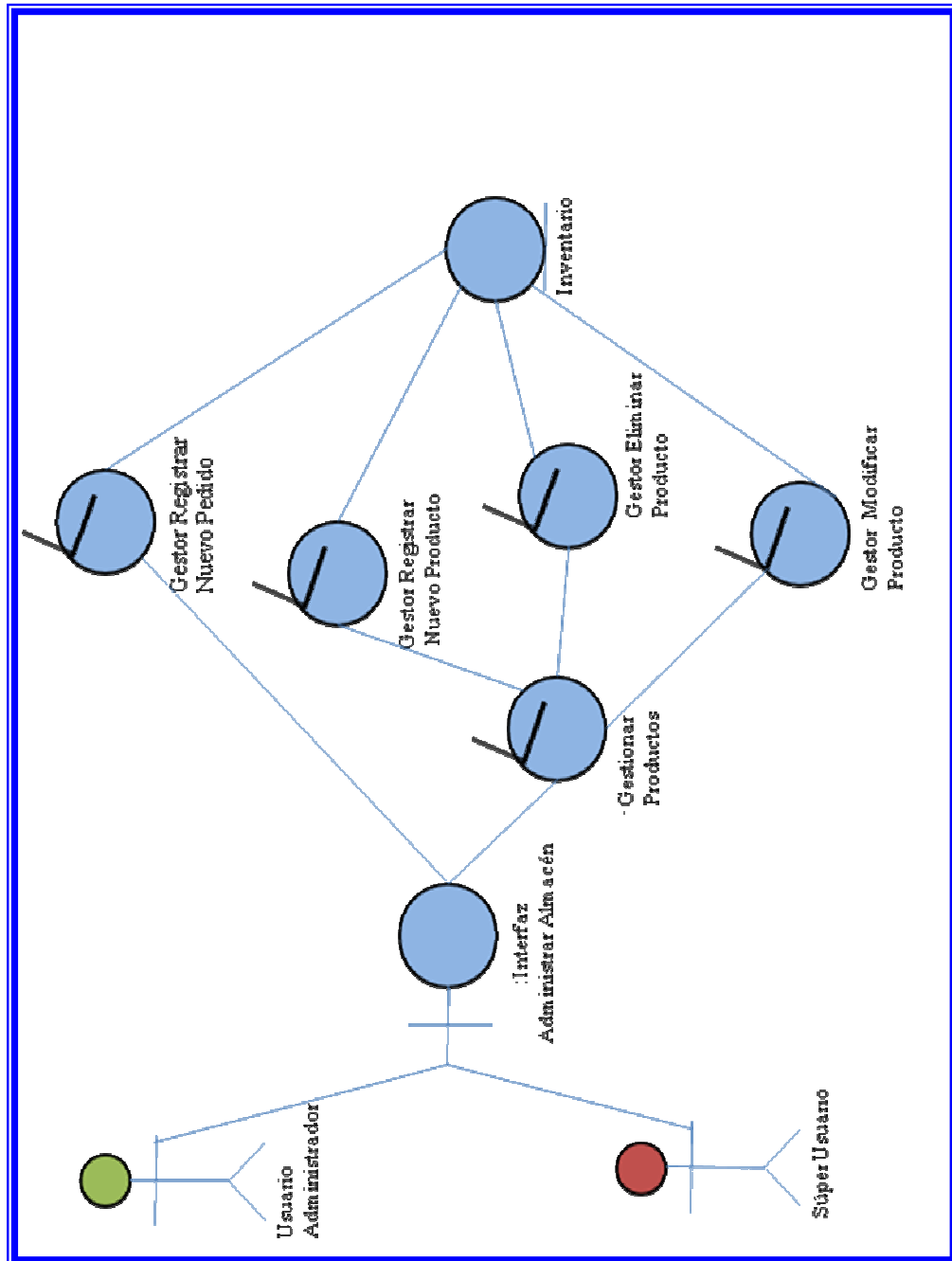


Figura 3. 12 - Diagrama de Clase de Análisis ‘Administrar Almacén’



Este diagrama de clases de análisis (Figura 3.12), se muestra de forma más detallada cómo interactúan las interfaces, los gestores y las entidades en el módulo de administración de almacén. Esta página muestra las opciones correspondientes al módulo.

La interfaz de entrada a la administración del almacén inicia el proceso de validación de los trabajadores pidiendo un *id de usuario*, un *nombre de usuario* y un *password*.

El gestor de registrar nuevo pedido, permite el ingreso de datos al sistema, el gestor de gestiones de productos se encarga del manejo de los gestores de registro de nuevos productos, del gestor de eliminar y modificar productos.

Este diagrama muestra la administración del stock de la empresa a través de los procesos necesarios.

3.10.2 Casos de Uso “Visualizar Reportes Ventas” de “S2A” (Sistema de Administración de Almacén)

Este diagrama de clases de análisis muestra de forma más detallada cómo interactúan las interfaces, los gestores y las entidades en el módulo de reportes de ventas.

La interfaz de entrada a la visualización de reportes de ventas, se inicia con el proceso de validación de los trabajadores pidiendo un *id de usuario*, un *nombre de usuario* y un *password*.

El gestor de Seleccionar empleado, permite extraer de la entidad Personal, la información solicitada por el usuario correspondiente al trabajador y de la entidad



ventas, los datos almacenados acerca de las ventas mensuales realizadas por ese empleado, para así poder realizar la visualización de los reportes.

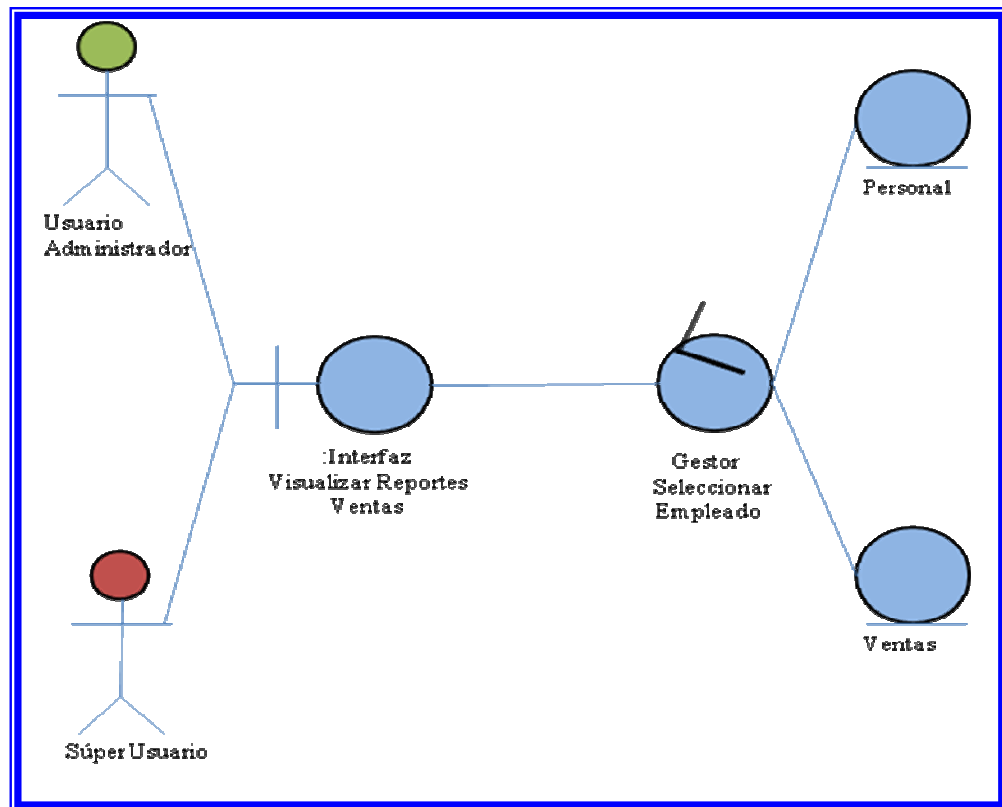


Figura 3. 13 - Diagrama de Clase de Análisis " Visualizar Reportes Ventas".

El caso de uso Realizar venta es iniciado por una instancia del actor Súper Usuario, o por el mismo Súper Usuario, mediante la activación de la Interfaz de entrada "Menú", lo cual conlleva a que el actor pueda seleccionar las operaciones correspondientes al proceso.



3.10.3 Casos de Uso "Realizar Venta" de "S2A" (Sistema de Administración de Almacén).

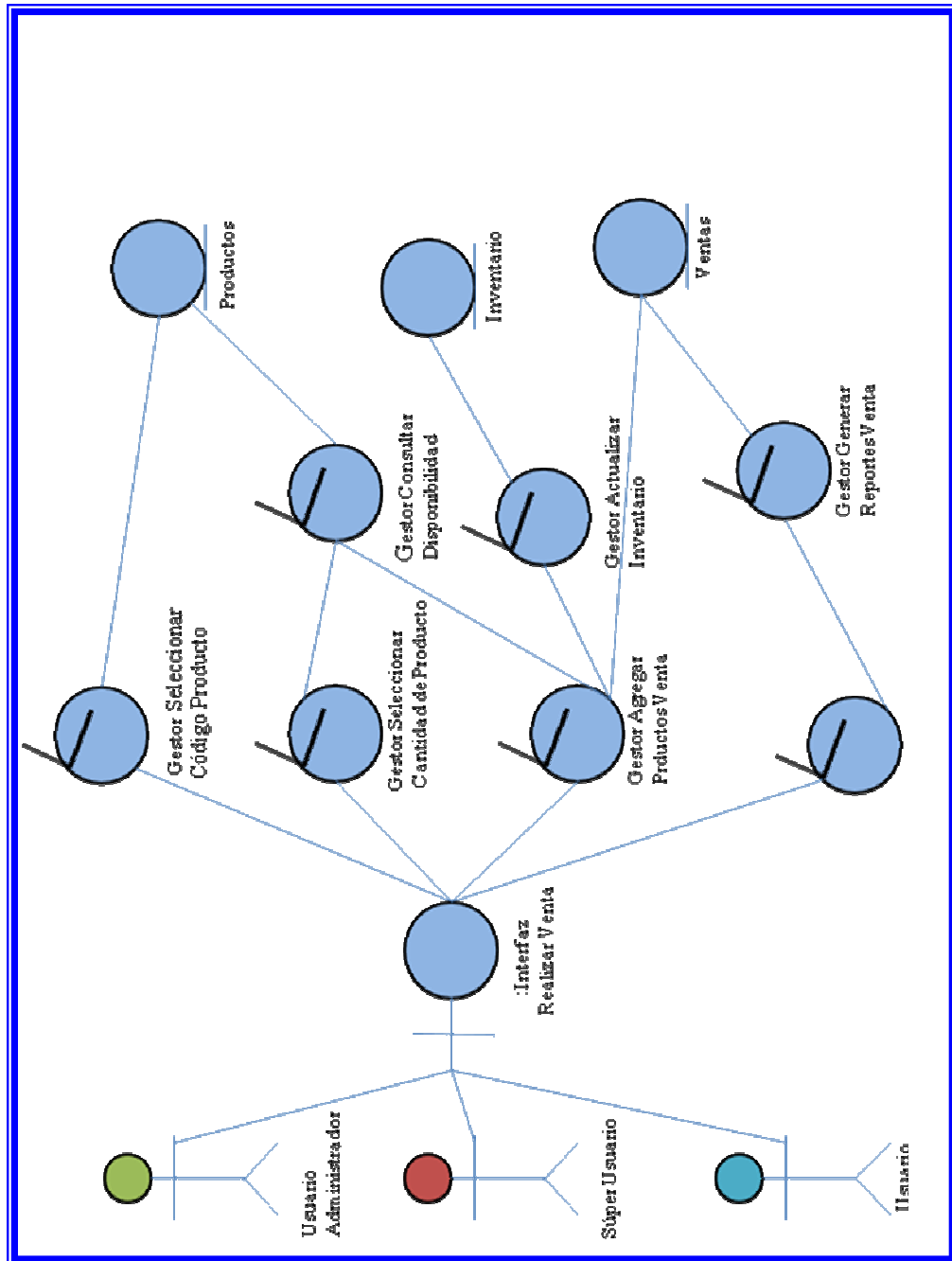


Figura 3. 14 - Diagrama de Clase de Análisis " Realizar Venta".



Los gestores son los encargados de controlar las operaciones realizadas por los usuarios, el gestor de selección de código de producto al igual que el gestor de selección de cantidad de producto, se encarga de elegir la información solicitada por el usuario de la entidad Productos. Luego interviene el gestor de consultar disponibilidad, que es el encargado de manejar las búsquedas en la base de datos, solicitada por los gestores anteriores

El gestor agregar productos se encarga de controlar la agregación de productos a la venta y el gestor de actualizar inventario, interviene luego de la finalización del gestor agregar productos, actualizando los valores en la entidad inventario del sistema.

El gestor de facturar venta, se encarga de controlar las operaciones correspondientes a la facturación de la venta y el gestor de generar reporte de venta, se encarga de la realización de los reportes a nivel de sistema, tomando los datos correspondientes de la entidad ventas.

3.10.4 Casos de Uso “Visualizar Resumen Ventas” de “S2A” (Sistema de Administración de Almacén).

Este diagrama de clases de análisis muestra de forma más detallada cómo interactúan las interfaces, los gestores y las entidades en el módulo de resumen de ventas.

La interfaz de entrada a la visualización de resumen de ventas, se inicia con el proceso de validación de los trabajadores pidiendo un *id de usuario*, un *nombre de usuario* y un *password*.



El gestor de Mostrar información, es el encargado de permitir la visualización de la información detallada, correspondiente a las ventas realizadas por los empleados de la empresa.

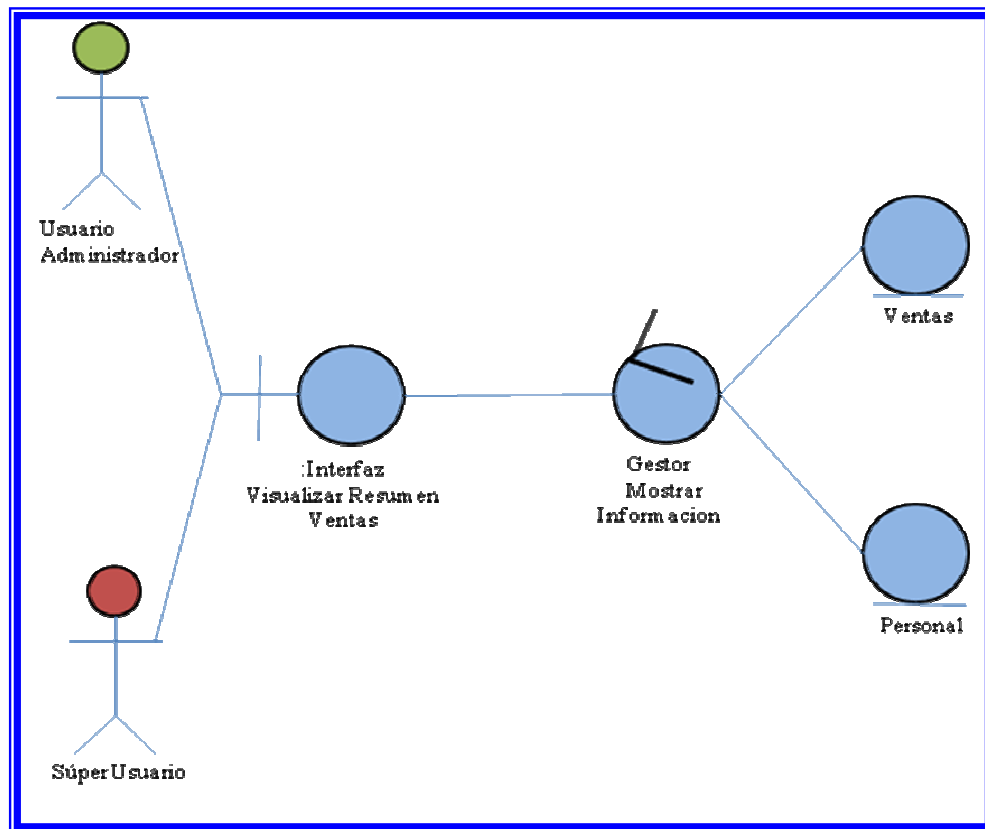


Figura 3. 15 - Diagrama de Clase de Análisis " Visualizar Resumen Ventas".

3.11 DIAGRAMA DE COLABORACIÓN GENERAL DE "S2A" (SISTEMA DE ADMINISTRACIÓN DE ALMACÉN).

El diagrama de colaboración muestra los diferentes objetos, las relaciones e interacciones que pueden darse entre ellos, permitiendo expresar el contexto de un grupo de objetos, a través de enlaces, y la interacción entre estos objetos, a través de mensajes.



El diagrama de colaboración se centra en estudiar todos los efectos de un objeto dado durante un escenario. Los objetos se conectan por medio de enlaces, cada enlace representa una instancia de una asociación entre las clases implicadas.

A continuación se muestran los diagramas de colaboración de “S2A”, el diagrama general y el de los módulos de “*Administrar Almacén*”, “*Visualizar Reportes Ventas,*” de “*Realizar Venta*” y el de “*Visualizar Resumen Ventas*”, cuyo trabajo en conjunto es fundamental para que la aplicación cumpla con los requerimientos, por esto son los módulos sometidos a la mayor cantidad de pruebas.



3.11.1 Diagrama de Colaboración General de “S2A” (Sistema de Administración de Almacén).

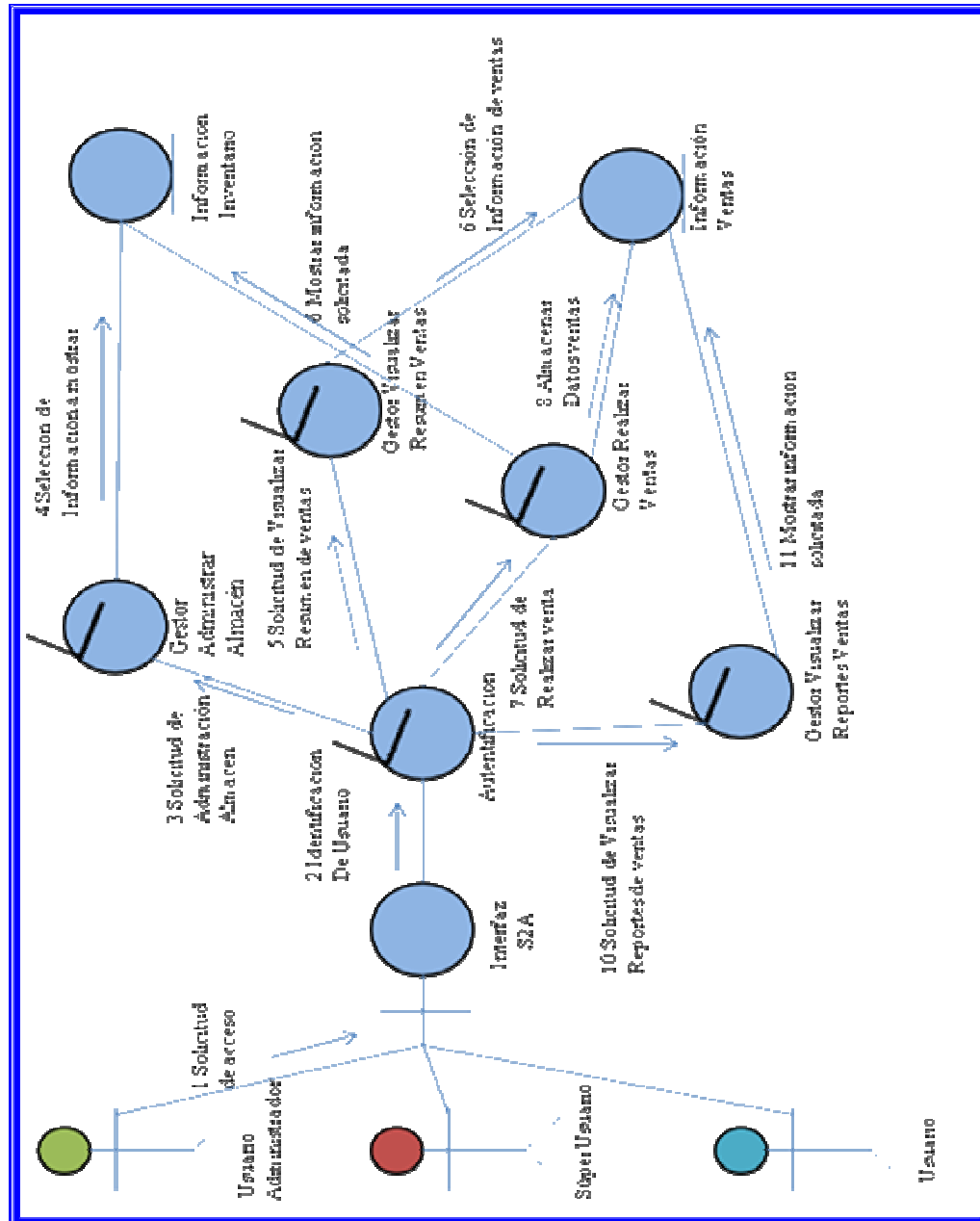


Figura 3. 16 - Diagrama de Colaboración General del Sistema.



3.11.3 Diagrama de Colaboración “Visualizar Reportes Ventas” de “S2A” (Sistema de Administración de Almacén).

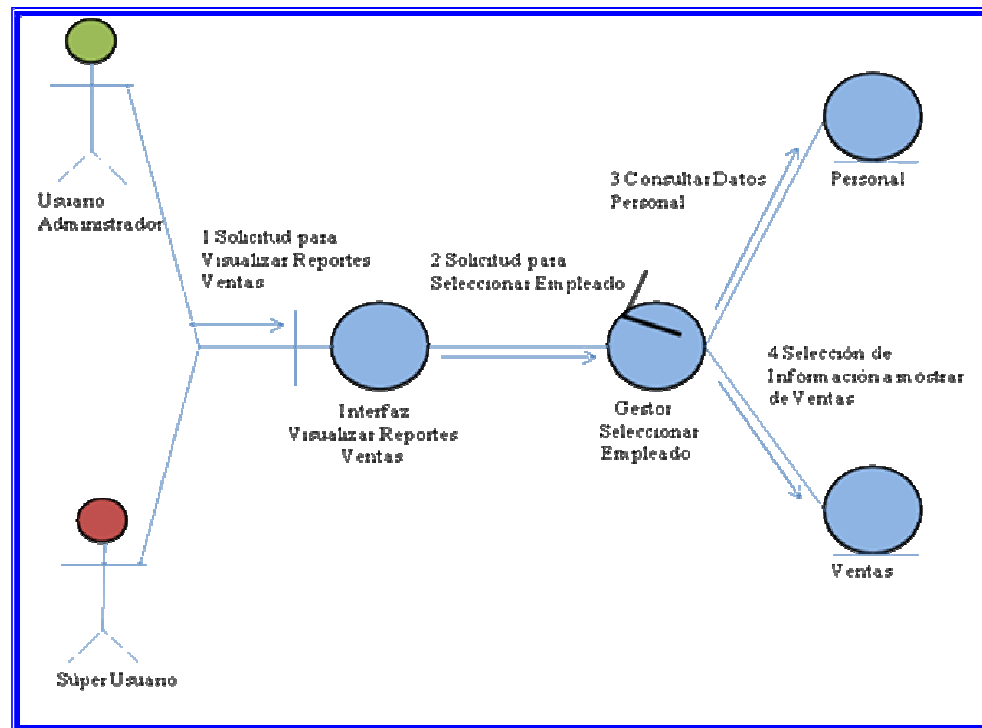


Figura 3. 17 - Diagrama de Colaboración “Visualizar Reportes Ventas”.



3.11.4 Diagrama de Colaboración ‘Administrar Almacén’ de ‘S2A’ (Sistema de Administración de Almacén).

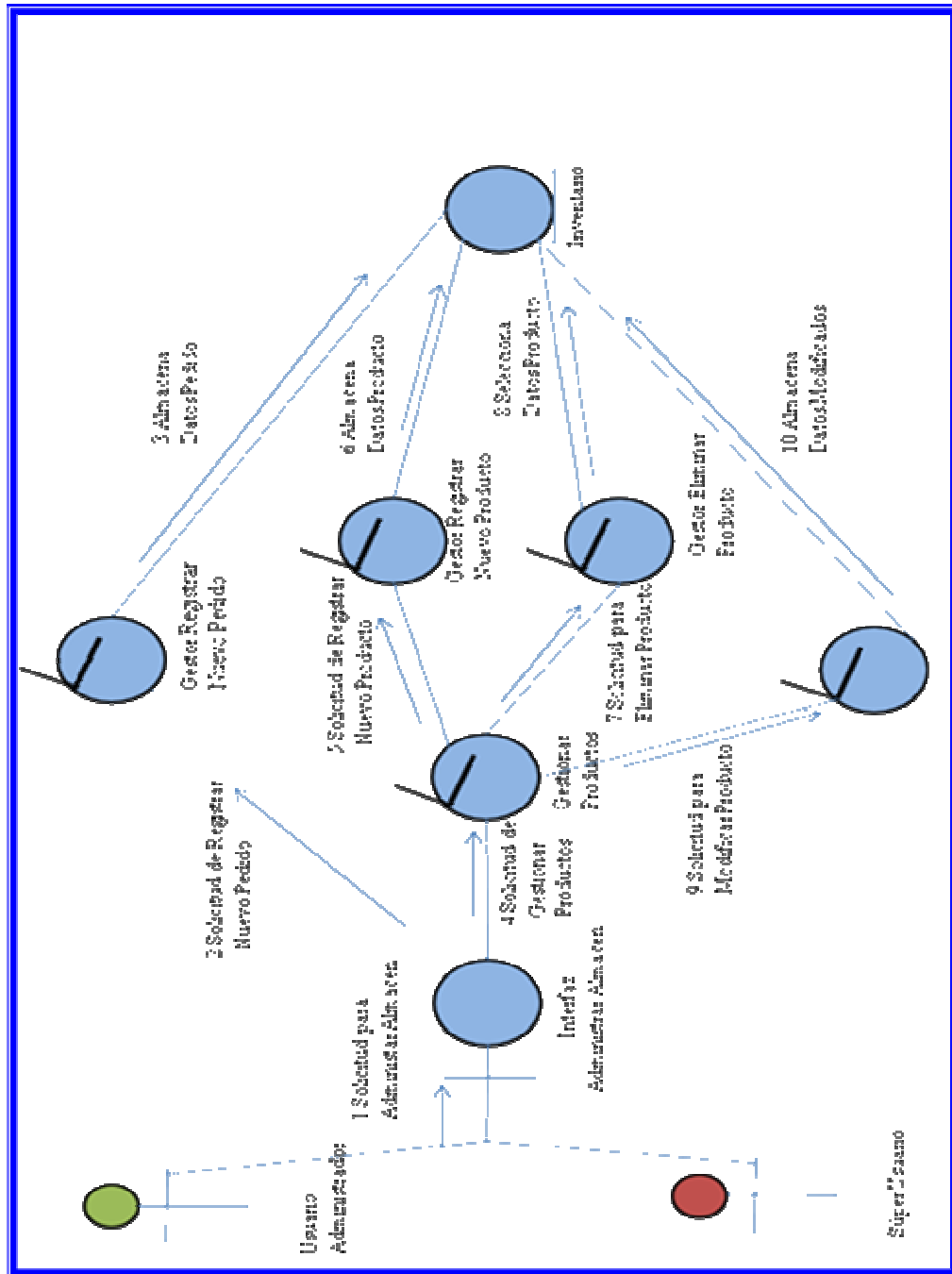


Figura 3. 18 - Diagrama de Colaboración ‘Administración de Almacén’.



3.11.5 Diagrama de Colaboración ‘Realizar Venta’ de ‘S2A’ (Sistema de Administración de Almacén).

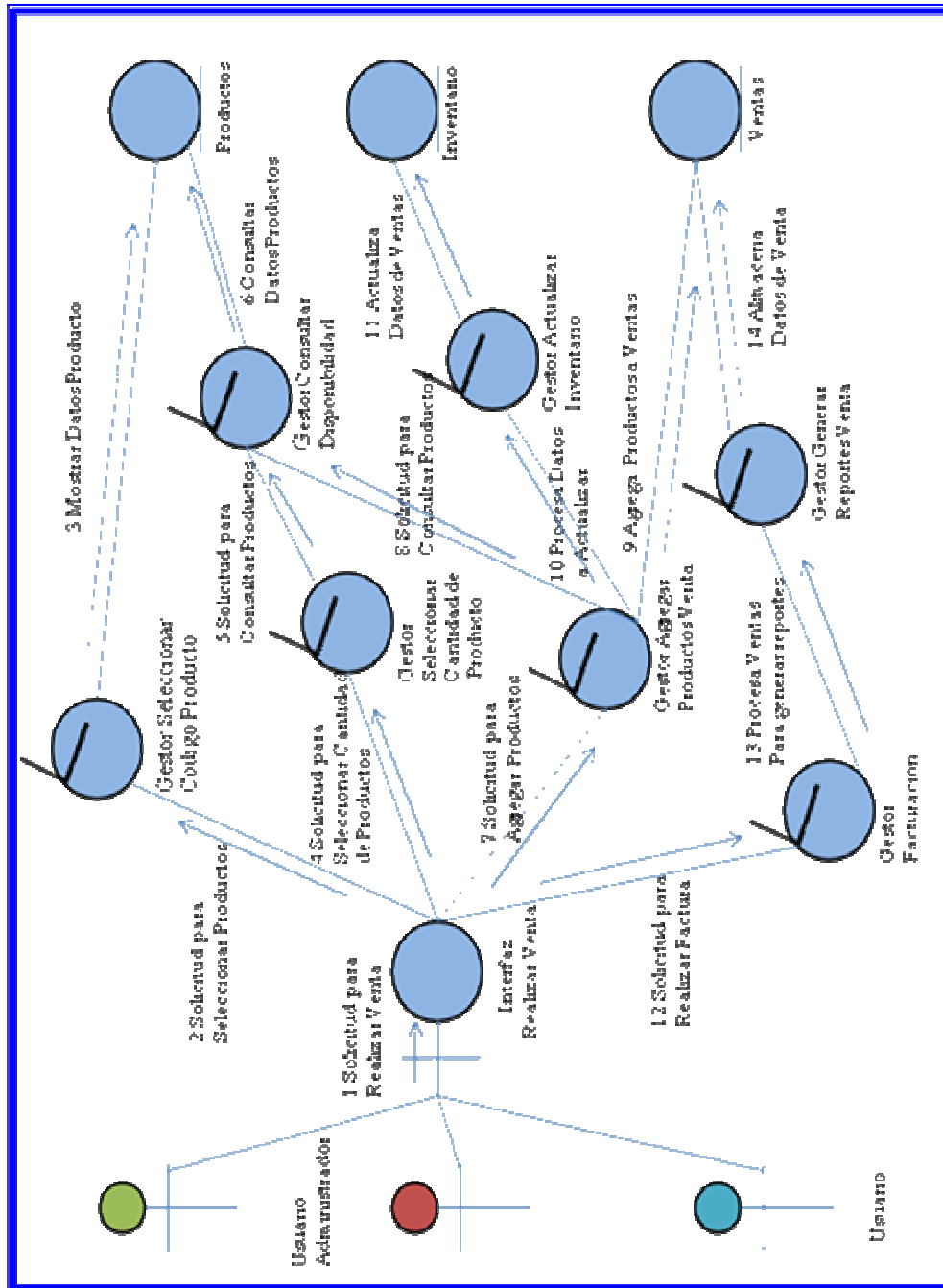


Figura 3. 19 - Diagrama de Colaboración ‘Realizar Venta’



3.11.6 Diagrama de Colaboración “Visualizar Resumen Ventas” de “S2A” (Sistema de Administración de Almacén).

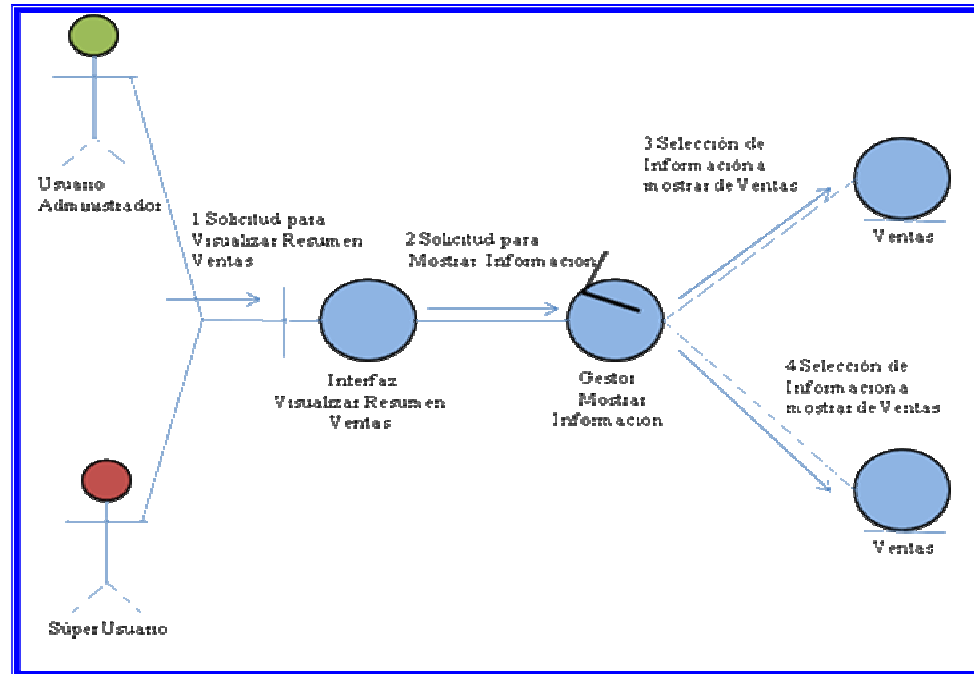


Figura 3. 20 - Diagrama de Colaboración “Visualizar Resumen Ventas”.

3.12 PAQUETES DE ANÁLISIS.

3.12.1 Identificación de los paquetes de análisis

Representa un medio de organizar el modelo de análisis en piezas más pequeñas basándose en requisitos funcionales y dominio del sistema.

Define un espacio de nombre de modo que dos elementos diferentes, contenidos en dos paquetes diferentes, pueden tener el mismo nombre, pueda contener otros paquetes, sin límite de nivel de anidamiento. Un nivel dado puede contener una mezcla de paquetes y de otros elementos de modelado, de la misma manera que un



directorio puede contener directorios y archivos. Puede constar de clases de análisis, de casos de uso y de otros paquetes. Poseen trazas con subsistemas en el diseño.

- **Paquete de análisis general**

Contiene los paquetes específicos del sistema.

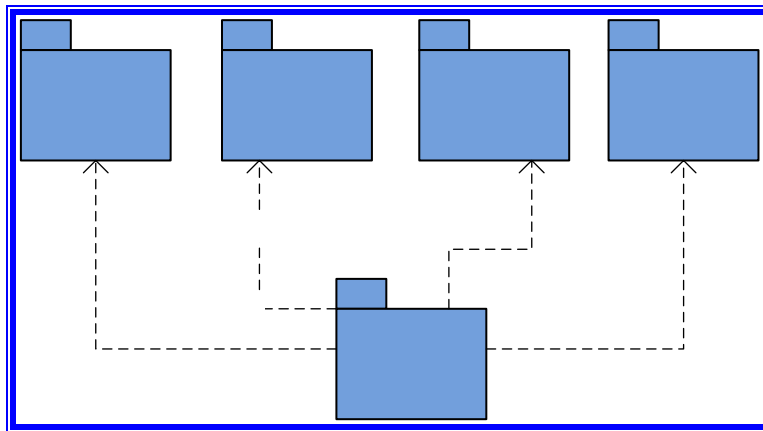


Figura 3. 21 - Diagrama de Paquete de Análisis General del Sistema.

- **Paquete de Análisis Administrar Almacén**

El paquete de análisis esta asociado al caso de uso Administrar Almacén.

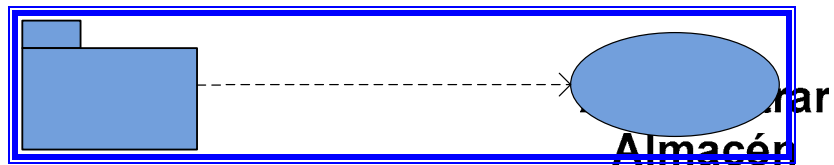


Figura 3. 22 - Paquete de Análisis Administrar Almacén

- **Paquete de Análisis Visualizar Reportes Ventas**

El paquete de análisis está asociado al caso de uso Visualizar Reportes Ventas

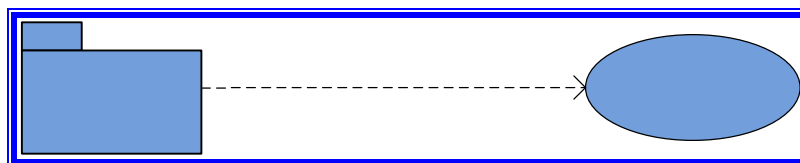


Figura 3. 23 - Paquete de Análisis Visualizar Reportes Ventas

Visualizar
Resumen
Ventas

«traces»

«traces»



- **Paquete de Análisis Realizar Venta**

El paquete de análisis está asociado al caso de uso Realizar Venta

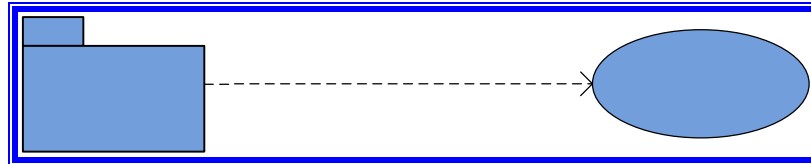


Figura 3. 24 - Paquete de Análisis Realizar Venta

- **Paquete de Análisis Visualizar Resumen Ventas**

El paquete de análisis está asociado al caso de uso Visualizar Resumen Ventas.

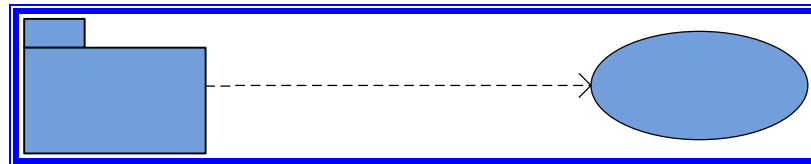


Figura 3. 25 - Paquete de Análisis Visualizar Resumen Ventas

Realizar Venta

3.13 CONCLUSIÓN DE LA FASE

Los objetivos principales de la fase de inicio son: establecer un análisis del sistema propuesto, describir el contexto del sistema, capturar los requerimientos funcionales, identificar los riesgos críticos que pondrían en peligro el desarrollo del proyecto y proponer una arquitectura candidata factible. Dichos objetivos se cumplieron de una forma satisfactoria, ya que se obtuvo una primera versión del modelo que describe el contexto del sistema, una lista inicial de riesgos y un esbozo de los modelos que representan una primera versión el modelo de casos de uso y de modelo de análisis, los cuales describen una arquitectura candidata factible. En esta fase se obtuvo una buena comprensión del proyecto y la factibilidad de culminarlo. Los resultados alcanzados en esta fase se refinarán en la fase de elaboración.

**Visualizar
Resumen
Ventas**

CAPÍTULO IV

FASE DE ELABORACIÓN

4.1 INTRODUCCIÓN

En esta fase se trabajará con la segunda fase del proceso unificado de desarrollo de software, conocida como elaboración el objetivo de esta fase es la definición de la estructura interna de la aplicación el cual proporcionará unas buenas bases para las siguientes fases de diseño e implementación. La definición de la arquitectura debe de tener en cuenta los requerimientos obtenidos durante la fase anterior de inicio o concepción.

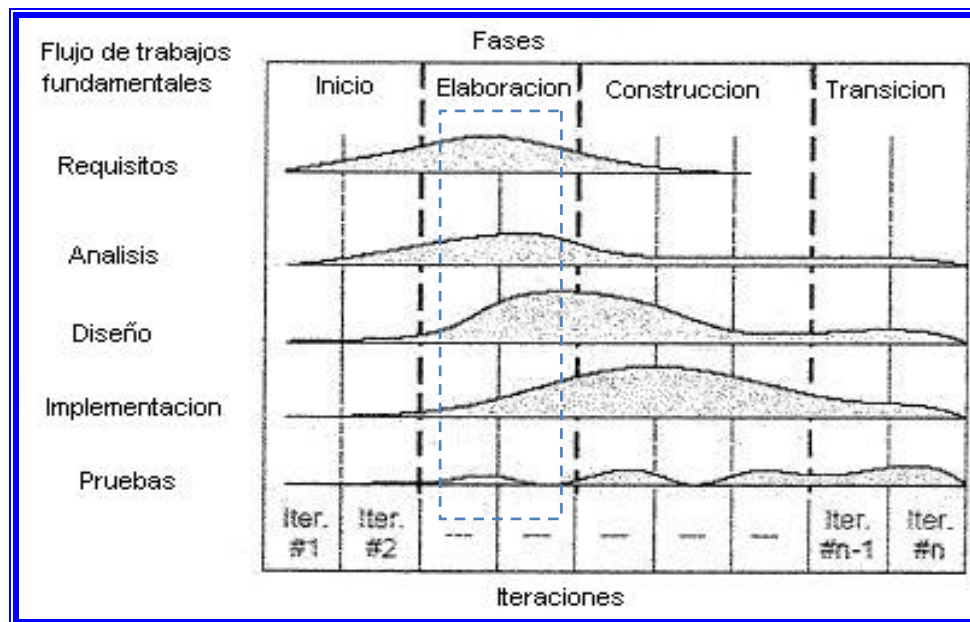


Figura 4. 1 - Diagrama de los flujos de trabajo sobre cada una de las fases (Resaltando la Fase de Elaboración)

En esta fase se hará énfasis en analizar y refinar todo lo recabado en la fase de inicio para ir consiguiendo soluciones. Se establecerá aquí la arquitectura de la



aplicación, así como también el diseño de la Base de Datos y la interfaz de usuario. Durante esta fase se especifican la mayoría de los casos de usos, finalizando con las distintas vistas de los diferentes modelos del sistema.

4.2 PLANIFICACIÓN DE LA FASE

En esta fase se logra recopilar la mayor parte de los requerimientos que aún quedan pendientes o casi todos los requisitos; establecer una arquitectura sólida para guiar el trabajo de las fases; continuar la observación y control de los riesgos críticos que aun queden y completar los detalles del plan de proyecto.

Para ello se identificarán los nuevos requisitos y de ser necesario se representarán como casos de usos, se analizarán haciendo uso de los diagramas de clase de análisis y de colaboración. Luego se reflejan como se encapsulará ese nuevo caso de uso en los subsistemas e interfaces, usando los paquetes del análisis, el diagrama de capas y el diagrama de clases de diseño del sistema.

4.3 REQUISITOS DEL SISTEMA

En este flujo de trabajo se hizo una nueva evaluación de los casos de uso del sistema, con el fin de analizar los nuevos casos de uso que aparecen en esta fase, esto a razón de que se profundizó e indagó un poco más en el análisis del sistema, lo cual dio como resultado una nueva iteración ya que se detecto la necesidad de agregar unos nuevos casos de uso que en la fase anterior no se desarrollaron. Es por ello que ahora en la fase de elaboración se analizarán y se detallarán los nuevos casos de uso presentes.



Se puede observar que los casos de uso que sufrieron cambios siguiendo el esquema planteado en la fase de inicio, fueron los casos de uso “Administrar Almacén” y “Realizar Venta”.

Dentro de las nuevas iteraciones agregadas a estos casos de uso, tenemos la expansión del caso de uso “Registrar Nuevo pedido” que se encuentra contenido dentro del caso de uso “administrar almacén” y el caso de uso “selección forma de pago” y su respectiva expansión, que es un nuevo caso de uso agregado al proceso de facturación que se encuentra vinculado al caso de uso “realizar venta”. En la figura 4.1 se resaltan los casos de uso alterados con respecto a la fase de inicio.

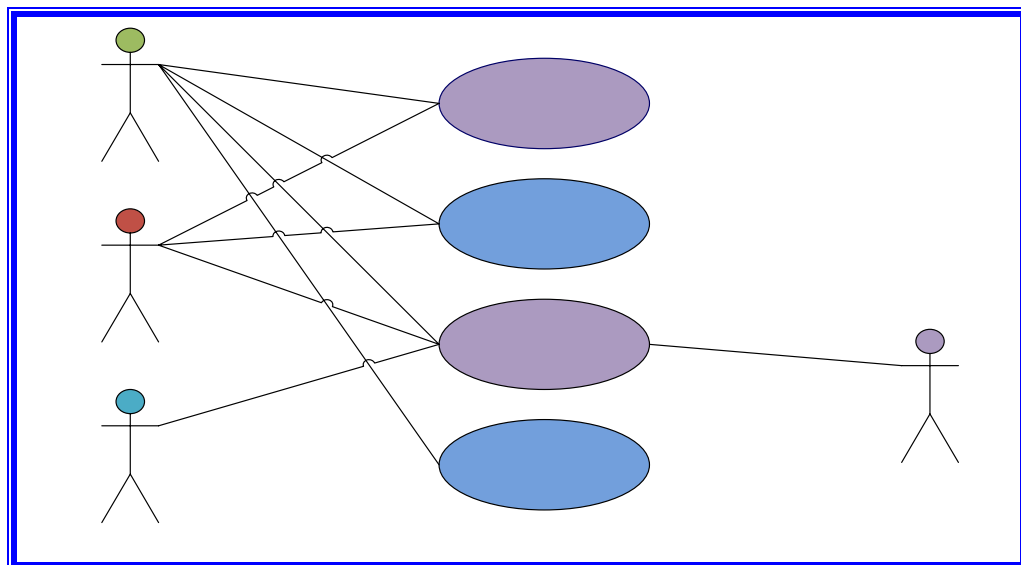


Figura 4.2 - Caso de Uso “General de S2A”



4.4 IDENTIFICACIÓN DE LOS CASOS DE USO.

A continuación se describirán los casos de uso presentes en la fase de elaboración.

Tabla 4. 1 - Descripción de los Casos de Uso.

Caso de Uso	Descripción
Registrar Nuevo Pedido	Mediante este caso de uso el usuario puede hacer la solicitud de nuevos pedidos de productos para el almacén
Registrar Datos	Este caso de uso permite el registro de los datos de los nuevos pedidos de productos, así como el código, el precio y la cantidad
Registrar Nuevo Producto	Mediante este caso de uso el usuario puede hacer registrar nuevos productos en el almacén
Declarar Nuevo producto	Este caso de uso se utiliza para ingresar los datos de producto, código, precio, cantidad, descripción, entre otros
Realizar Venta	Se utiliza para la realización de ventas de productos, consultando la disponibilidad del producto y las formas de pago.
Configurar Usuario	Este caso de uso permite crear cuentas de usuarios, se realiza a nivel de código de sistema, por el usuario administrador
Seleccionar Forma de pago	Este caso de uso permite a los usuarios establecer la forma de pago de la venta de los productos
Seleccionar Número de Cuotas	Mediante este caso de uso se señalan la cantidad de cuotas en las cuales será cancelado el monto de la venta



4.5 DIAGRAMA DE CASO DE USO DETALLADOS DE 'S2A' (SISTEMA DE ADMINISTRACIÓN DE ALMACÉN)

Para la visualización de los procesos representados de forma general en el diagrama de contexto (Figura 4.3), se presenta a continuación de forma más detallada la estructura interna de cada proceso a través de su representación en casos de uso:

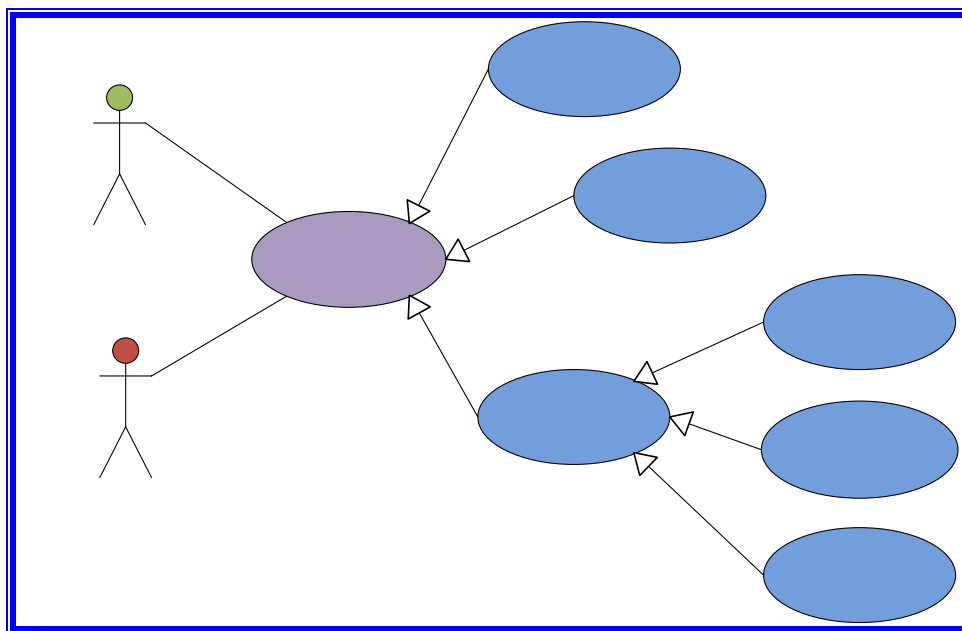


Figura 4.3 - Caso de Uso "Registrar Nuevo Pedido"

4.5.1 Descripción del flujo de Sucesos del Caso de Uso "Registrar Nuevo Pedido"

Nombre del Caso: "Registrar Nuevo Pedido".

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar nuevos pedidos en el stock.

Usuario Administrador

Registrar Nuevo
Pedido



Pre-Condición: El usuario realiza la selección desde el proceso “Administrar Almacén”

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se llenan los formularios de registro de producto con la información del mismo, para llevar a cabo el registro.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

Nombre del Caso: “Ingresar Número Pedido”.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador ingresar el número de pedido de productos.

Pre-Condición: El usuario realiza la selección desde el proceso “Registrar Nuevo Pedido”.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se proporciona la información requerida para la ejecución del proceso.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema



Nombre del Caso: ‘‘Ingresar cantidad de productos’’.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador ingresar la cantidad de productos del de pedido.

Pre-Condición: El usuario realiza la selección desde el proceso ‘‘Registrar Nuevo Pedido’’.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se proporciona la información requerida para la ejecución del proceso.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: ‘‘Registrar Datos’’.

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar en el sistema la declaración de los productos, así como el código, el precio unitario y la cantidad.

Pre-Condición: El usuario realiza la selección desde el proceso ‘‘Registrar Nuevo Pedido’’.

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se proporciona la información requerida para la ejecución del proceso.
3. El usuario visualiza la información y escoge la operación que desea realizar.



4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Gestionar Productos".

Actores: Súper Usuario, Usuario Administrador.

Descripción: Este proceso permite al Súper Usuario y al Usuario Administrador registrar nuevos productos en la base de datos, eliminarlos o modificarlos

Pre-Condición: El usuario realiza la selección desde el proceso "Administrar Almacén".

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. No es un proceso visible para el usuario, pero si sus resultados o salidas, que serán la información visible para él.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

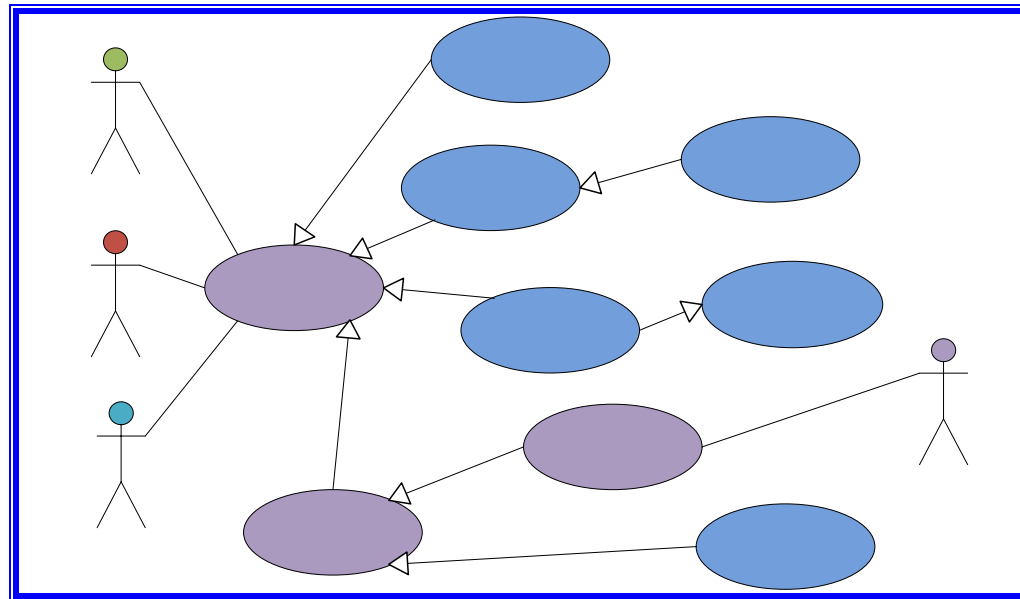


Figura 4.4 - Caso de Uso "Realizar Venta"

Usuario Administrador

4.5.2 Descripción del flujo de Sucesos del Caso de Uso " Realizar Venta"

Nombre del Caso: "Realizar Ventas".

Realizar Venta

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite al Usuario, al Súper Usuario y al Usuario Administrador realizar alguna de venta de productos disponibles en el stock.

Pre-Condición: El usuario realiza la selección desde la opción "Menú" «extends»

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se muestra un recuadro en el cual el usuario podrá seleccionar los productos que desea y la cantidad correspondiente de cada uno. Facturar Venta
7. El usuario visualiza la información y escoge la operación que desea realizar. Usuario
8. Finaliza el caso de uso.



Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

Nombre del Caso: ''Seleccionar código de producto'.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite que cualquiera de los usuarios seleccione el código del producto que se va a vender.

Pre-Condición: El usuario realiza la selección desde el proceso ''Realizar Venta''

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se muestra un recuadro en el cual el usuario podrá seleccionar o escribir el ó los códigos de los productos involucrados en la venta.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

Nombre del Caso: ''Seleccionar cantidad de producto'.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite que cualquiera de los usuarios seleccione la cantidad de productos que se van a vender.

Pre-Condición: El usuario realiza la selección desde el proceso ''Realizar Venta''

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.



6. Se muestra un recuadro en el cual el usuario podrá marcar la cantidad del ó los productos involucrados en la venta.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Consultar Disponibilidad".

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso es utilizado para consultar la existencia en stock del producto que se va a vender.

Pre-Condición: El usuario realiza la selección desde el proceso "Seleccionar Cantidad Producto"

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Es un proceso no visible por el usuario ya que consiste en una búsqueda en la base de datos del sistema, luego de ejecutado se visualiza la información de los productos.
7. El Usuario visualiza información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Agregar productos".

Actores: Súper Usuario, Usuario Administrador, Usuario.



Descripción: Este proceso permite que los usuarios agreguen productos a la venta antes de ser facturados para ejecutar la venta ó que confirmen los productos ya seleccionados.

Pre-Condición: El usuario realiza la selección desde el proceso “Realizar Venta”

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se muestra un recuadro en el cual el usuario podrá marcar la cantidad del ó los productos involucrados en la venta.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alterno:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: “Actualizar Inventario”.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso realiza la actualización de los productos restantes en stock una vez finalizada la venta.

Pre-Condición: El usuario realiza la selección desde el proceso “Agregar Productos”

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Es un proceso no visible por el usuario y consiste en la actualización de la base de datos del sistema. luego de ejecutado se visualiza la información correspondiente.



7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Facturar Venta".

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso realiza la facturación de los productos incluidos en la venta.

Pre-Condición: El usuario realiza la selección desde el proceso "Realizar Venta"

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se muestra un recuadro en el cual el usuario podrá marcar la opción de facturar si así lo desea.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Generar Reporte Venta.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso genera automáticamente según las ventas, las estadísticas mensuales de las mismas.



Pre-Condición: El usuario realiza la selección desde el proceso “Realizar Venta”

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Es un proceso no visible por el usuario, se generan los reportes de ventas según los valores actualizados en la base de datos del sistema.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

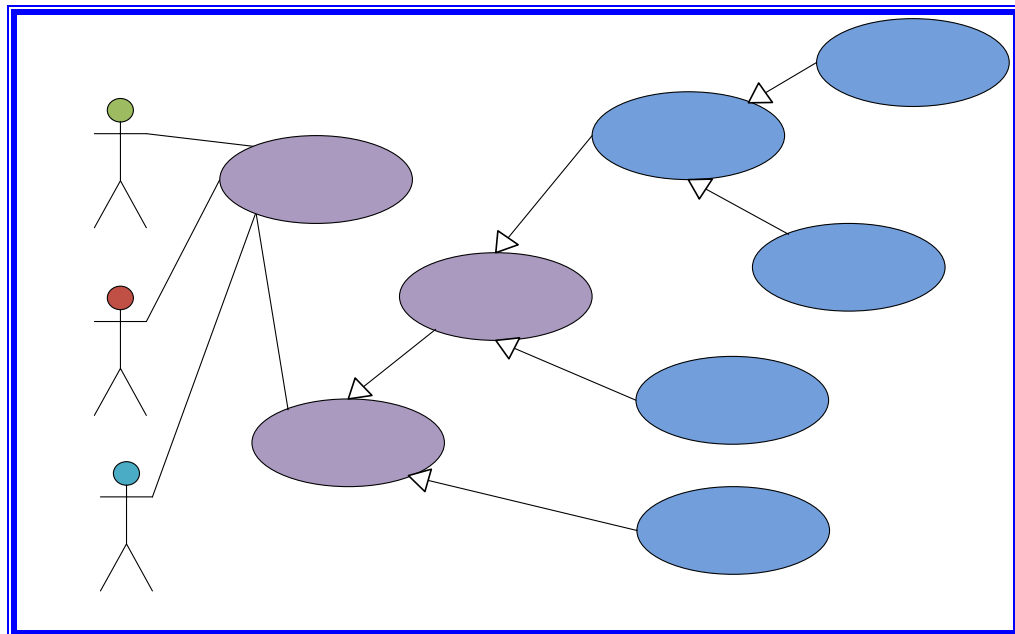


Figura 4. 5 - Caso de Uso “Seleccionar Forma de Pago”



Nombre del Caso: ''Seleccionar forma de Pago''.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite la selección de la forma de pago de la venta, bien sea de contado o en cuotas.

Pre-Condición: El usuario realiza la selección desde el proceso ''Realizar Venta''

Flujo de Eventos:

Flujo Principal:

5. El usuario invoca el caso de uso.
6. Se muestra un recuadro donde el usuario podrá marcar la forma de pago de la venta.
7. El usuario visualiza la información y escoge la operación que desea realizar.
8. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: ''Seleccionar Cantidad de Cuotas''.

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite realizar el pago de la venta de los productos en una cantidad de cuotas señaladas por el usuario.

Pre-Condición: El usuario realiza la selección desde el proceso ''Seleccionar Forma de Pago''

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro donde el usuario podrá señalar la cantidad de cuotas en el momento de la venta.
3. El usuario visualiza la información y escoge la operación que desea realizar.



4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema

Nombre del Caso: "Seleccionar Cuotas a Pagar".

Actores: Súper Usuario, Usuario Administrador, Usuario.

Descripción: Este proceso permite que el usuario especifique cuantas cuotas de la venta pagará en ese momento.

Pre-Condición: El usuario realiza la selección desde el proceso "Seleccionar Forma de Pago"

Flujo de Eventos:

Flujo Principal:

1. El usuario invoca el caso de uso.
2. Se muestra un recuadro donde el usuario podrá señalar la cantidad de cuotas que pagará en el momento de la venta.
3. El usuario visualiza la información y escoge la operación que desea realizar.
4. Finaliza el caso de uso.

Flujo Alternativo:

El usuario tiene la opción de realizar otra operación o puede salir del sistema.

4.6 DIAGRAMA DE CLASES DE ANÁLISIS DEL SISTEMA.

Se realiza una evaluación muy general de los casos de usos agregados al sistema, sin llegar a detalles muy profundos de estos, obteniendo como resultado el Modelo de Análisis (diagramas de colaboración y clases de análisis), el cual permite tener una especificación un poco más precisa de los requisitos obtenidos y recogidos, logrando de esta manera una guía para estructurar el sistema.



4.6.1 Casos de Uso “Registrar Nuevo Pedido” de “S2A” (Sistema de Administración de Almacén).

Este diagrama de clases de análisis se muestra de forma más detallada como interactúan las interfaces, los gestores y las entidades en el módulo de registro de nuevo pedido. Esta página muestra las opciones correspondientes a al módulo.

La interfaz de entrada al registro de nuevos pedidos, se inicia mediante la solicitud del proceso administrar almacén.

El gestor de registrar nuevo pedido, permite el ingreso de datos al sistema mediante el gestor de número de pedido, por el gestor de cantidad de productos, que indica cuantos productos tendrá el pedido y por el gestor ingresar datos, que se encarga de declarar los productos del pedido.

Luego, para culminar con el ingreso de los datos del pedido, intervienen el gestor de ingresar cantidad de productos, el gestor de ingresar precio unitario y el gestor de cantidad de producto.

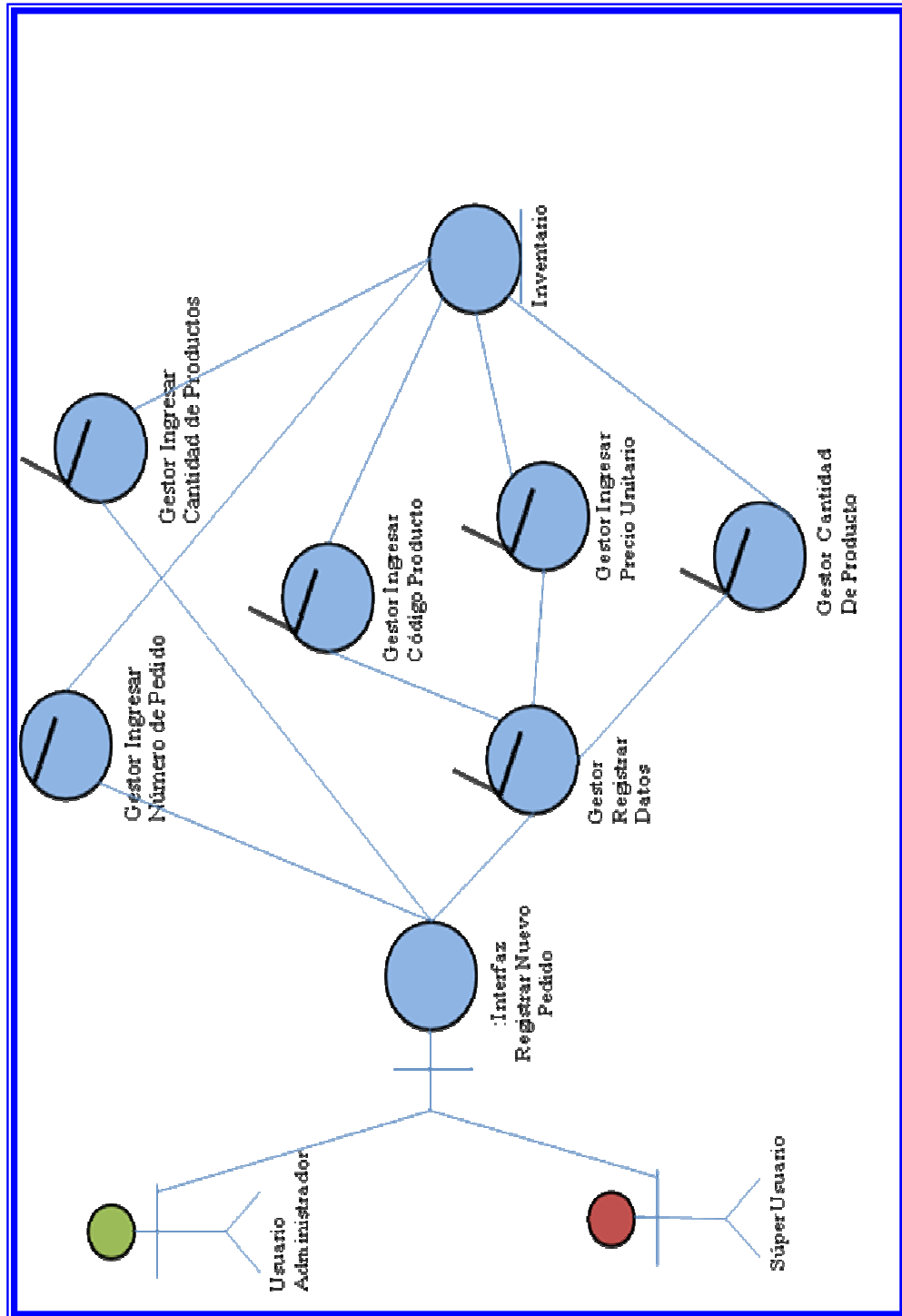


Figura 4. 6 - Diagrama de Clase de Análisis ‘Registrar Nuevo Pedido’



4.6.2 Casos de Uso “Realizar Venta” de “S2A” (Sistema de Administración de Almacén).

El caso de uso Realizar venta es iniciado por una instancia del actor Súper Usuario, o por el mismo Súper Usuario, mediante la activación de la Interfaz de entrada “Menú”, lo cual conlleva a que el actor pueda seleccionar las operaciones correspondientes al proceso.

Los gestores son los encargados de controlar las operaciones realizadas por los usuarios, el gestor de selección de código de producto al igual que el gestor de selección de cantidad de producto, se encarga de elegir la información solicitada por el usuario de la entidad Productos. Luego interviene el gestor de consultar disponibilidad, que es el encargado de manejar las búsquedas en la base de datos, solicitada por los gestores anteriores

El gestor agregar productos se encarga de controlar la agregación de productos a la venta y el gestor de actualizar inventario, interviene luego de la finalización del gestor agregar productos, actualizando los valores en la entidad inventario del sistema.

El gestor de facturar venta, se encarga de controlar las operaciones correspondientes a la facturación de la venta y el gestor de generar reporte de venta, se encarga de la realización de los reportes a nivel de sistema, tomando los datos correspondientes de la entidad ventas.

El gestor seleccionar forma de pago, se encarga de especificar de que manera se efectuara el pago de la venta.

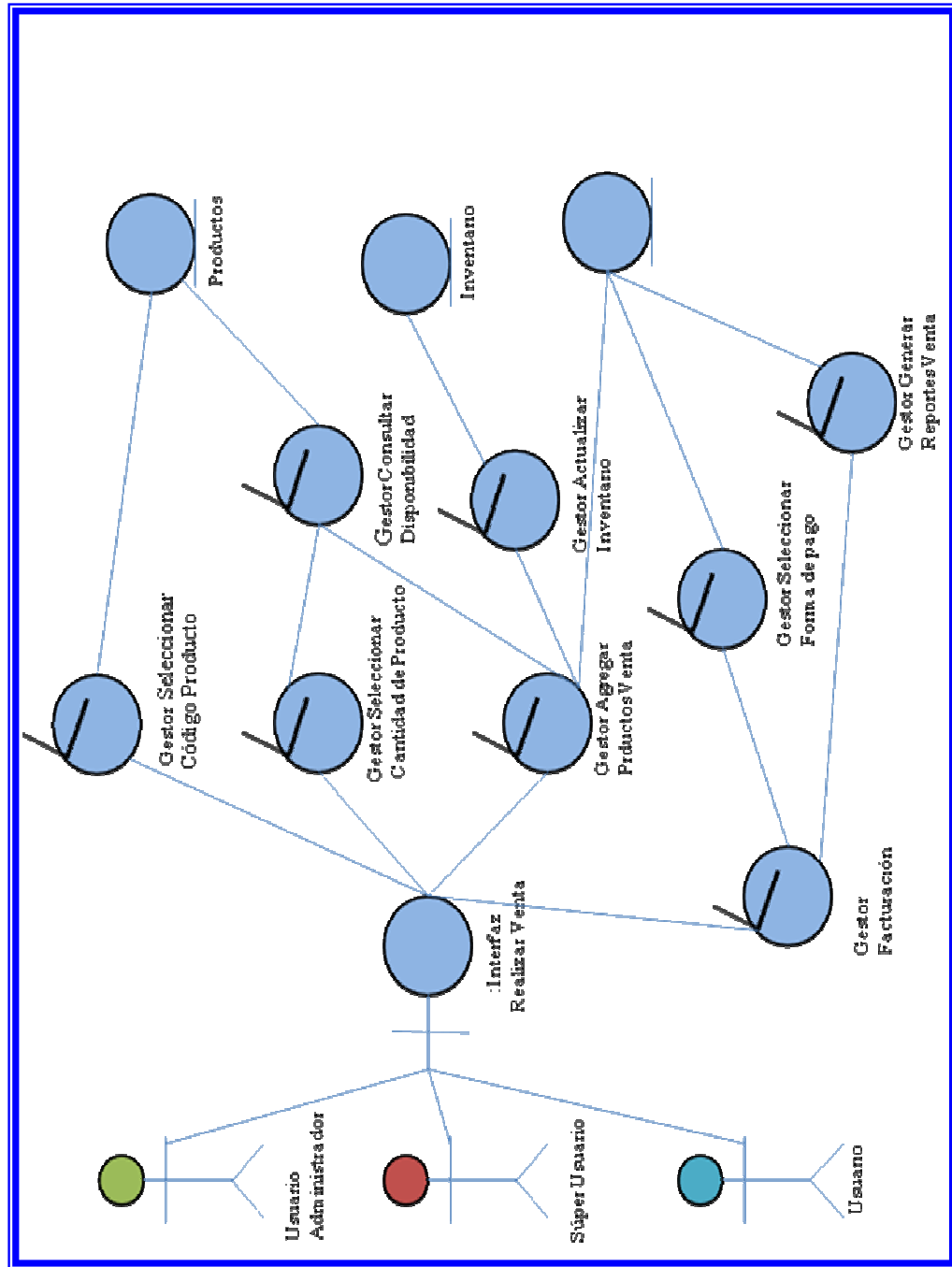


Figura 4. 7 - Diagrama de Clase de Análisis ‘Realizar Venta’



4.6.3 Casos de Uso ‘Seleccionar Forma de Pago’ de ‘S2A’ (Sistema de Administración de Almacén).

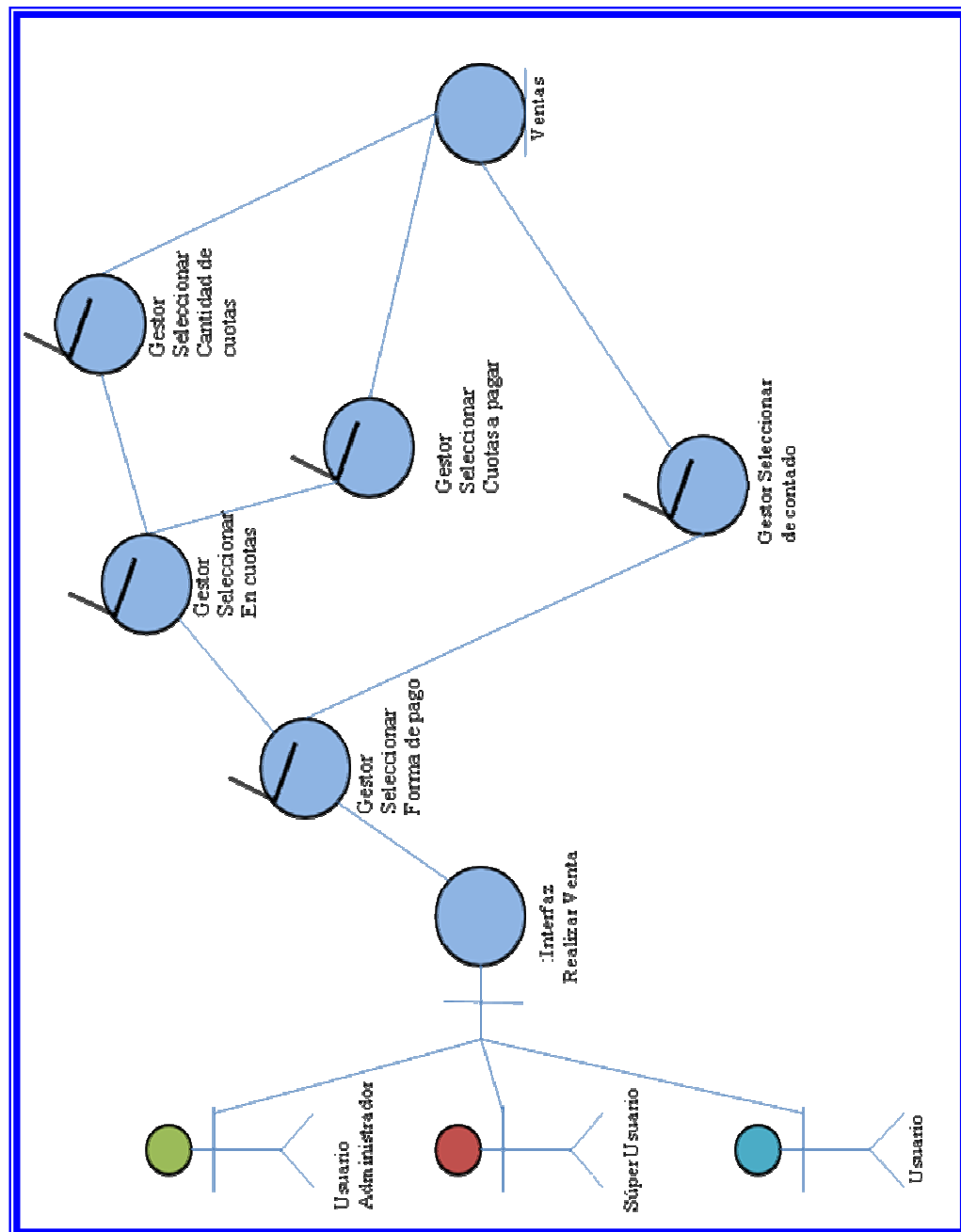


Figura 4.8 - Diagrama de Clase de Análisis ‘Seleccionar Formas de Pago’



El caso de uso Realizar venta es iniciado por una instancia del actor Súper Usuario, o por el mismo Súper Usuario, mediante la activación de la Interfaz de entrada “Menú”, lo cual conlleva a que el actor pueda seleccionar las operaciones correspondientes al proceso.

Los gestores son los encargados de controlar las operaciones realizadas por los usuarios, el gestor de seleccionar forma de pago, se encarga de especificar de que manera el usuario pagará por la venta realizada, si lo hará de contado o lo realizará en cuotas. Si selecciona la opción en cuotas, intervienen los gestores de selección número de cuotas que se encarga de establecer en cuantas cuotas se pagará la venta y el gestor seleccionar cuotas a pagar, que señala la cantidad de cuotas que cancelará el usuario al momento de la venta

Finalmente, el gestor seleccionar de contado, se encarga de facturar el monto total de la venta, en un solo pago,

4.7 DIAGRAMA DE COLABORACIÓN

A continuación se muestran los diagramas de colaboración de “S2A”, de los módulos de “*Registrar Nuevo Pedido*”, el de “*Realizar Venta*” y el de “*Seleccionar Forma de Pago*”, cuyo trabajo en conjunto es fundamental para que la aplicación cumpla con los requerimientos, por esto son los módulos sometidos a la mayor cantidad de pruebas.



4.7.2 Diagrama de Colaboración “Realizar Ventas” de “S2A” (Sistema de Administración de Almacén).

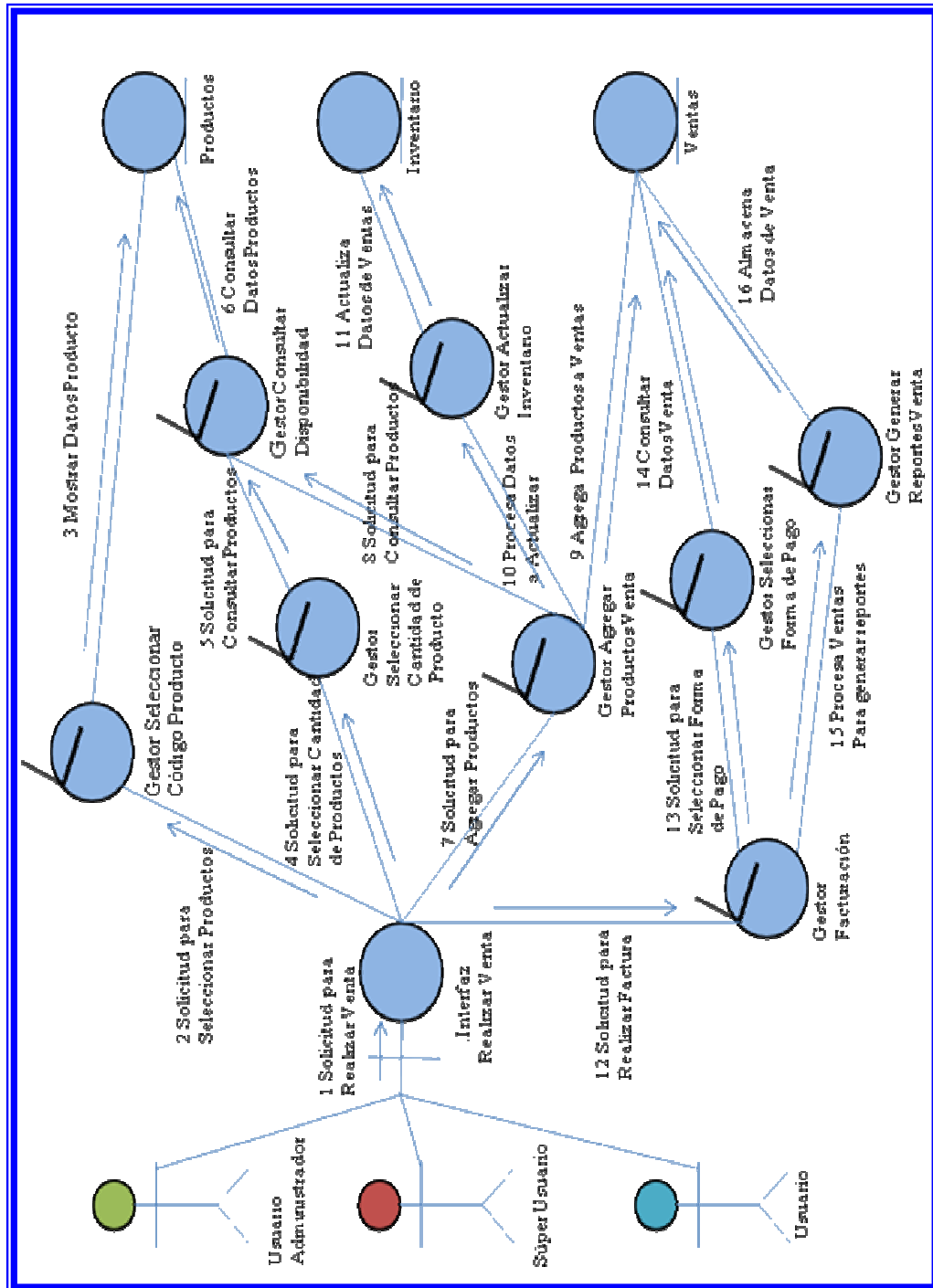


Figura 4. 10 - Diagrama de Colaboración “Realiza Venta”



4.7.3 Diagrama de Colaboración “Seleccionar Forma de Pago” de “S2A” (Sistema de Administración de Almacén).

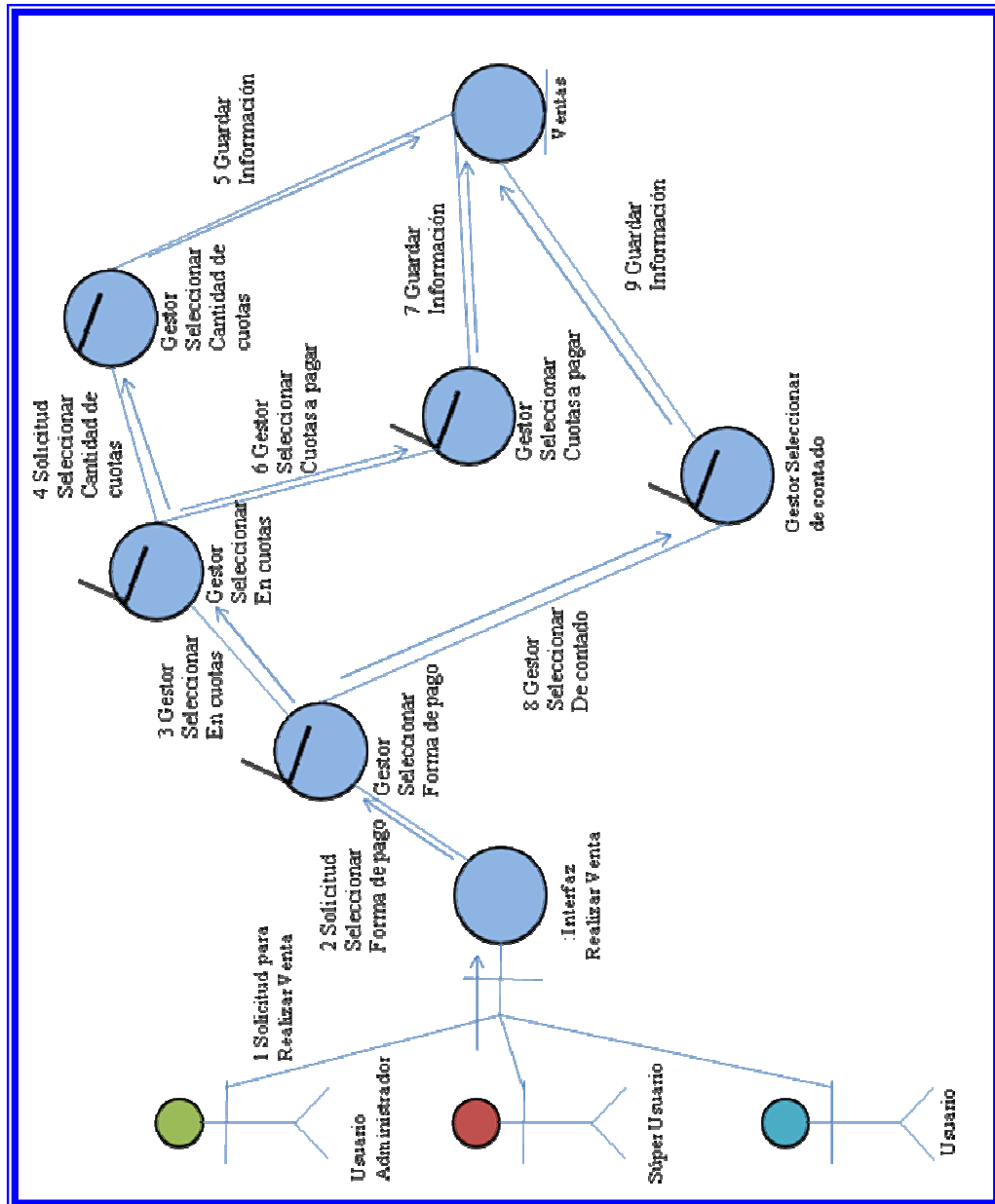


Figura 4. 11 - Diagrama de Colaboración “Seleccionar Forma de Pago”



4.8 IDENTIFICACIÓN DE LOS PAQUETES.

Los paquetes iniciales permanecen intactos, solo se hará el estudio, a los nuevos casos de uso que se encuentran en el sistema.

- **Paquete de análisis Registrar Nuevo Pedido**

El paquete de análisis está asociado al caso de uso Registrar Nuevo Pedido.

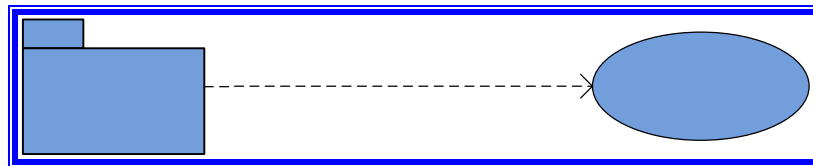


Figura 4. 12 - Paquete de Análisis "Registrar Nuevo Pedido"

- **Paquete de análisis Seleccionar Forma de Pago**

El paquete de análisis está asociado al caso de uso Seleccionar Forma de Pago.

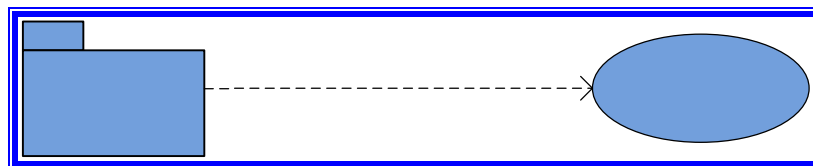


Figura 4. 13 - Paquete de Análisis "Seleccionar Forma de Pago"

**Registrar
Nuevo Pedido**



4.9 DIAGRAMA DE CAPAS

El diseño de la arquitectura del sistema involucra cuatro capas: la capa de aplicación, la capa general de la aplicación, la capa intermedia y finalmente la capa de software del sistema.

El diagrama de capas mostrado en la figura 4.7 presenta todas las capas del software en las que se distribuye la funcionalidad del sistema que se requieren para llevar a cabo la aplicación y en un futuro ser implementado.

Como se puede visualizar en la capa específica de la aplicación se presentan un conjunto de paquetes que encapsulan las diferentes clases del sistema.

En la capa general se representa el sistema en su totalidad y la interacción de cada uno de los paquetes necesarios para la realización de la gestión del sistema (Administrar almacén, resumen de ventas, realizar ventas, reportes de ventas).

De igual forma se presentan las capas intermedias de software del sistema.

Las relaciones entre los elementos de cada capa se representan mediante trazas.



4.9.1 Diagrama de Capas de la Aplicación

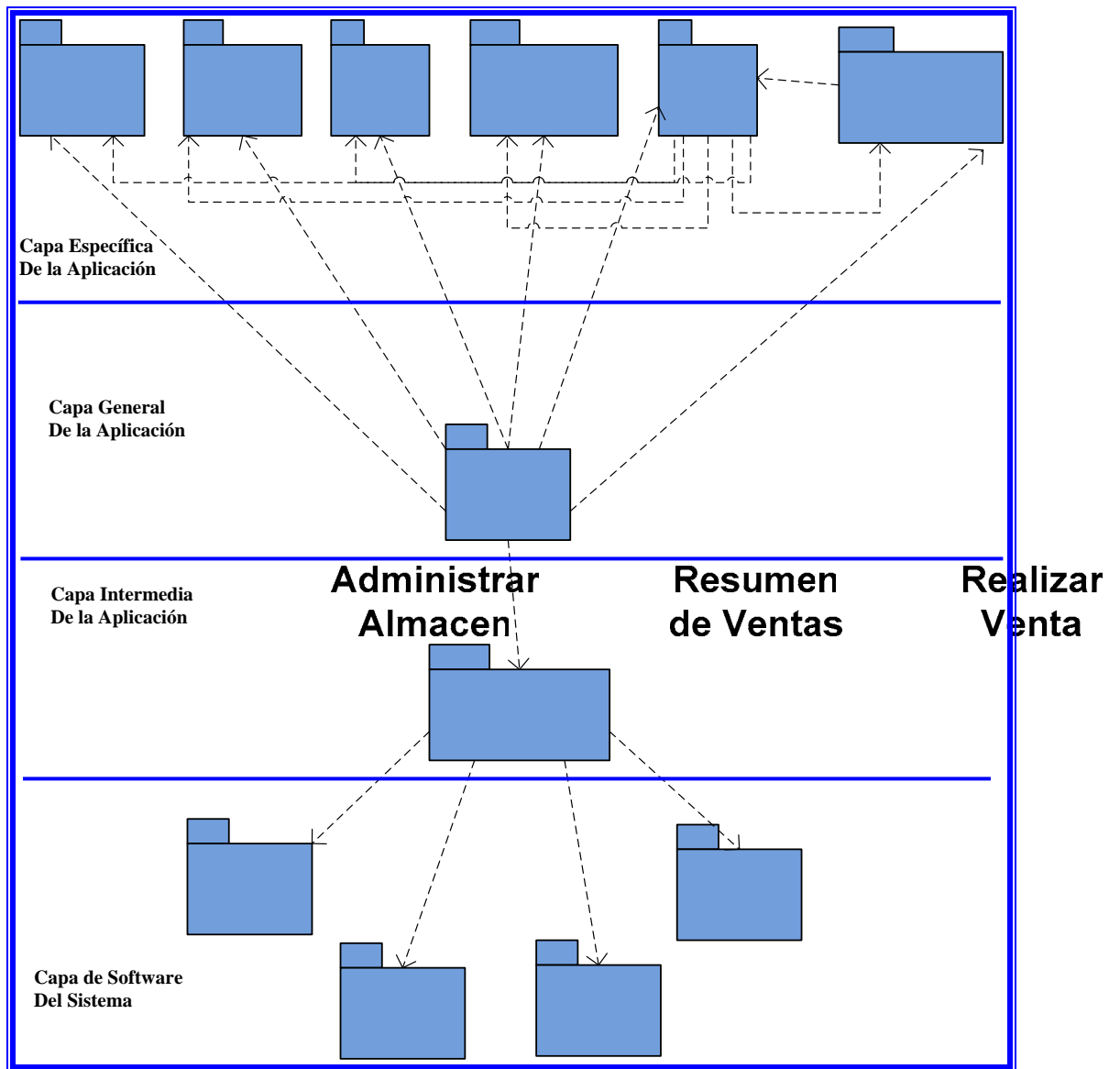


Figura 4. 14 - Diagrama de Capas de la aplicación



4.10 DIAGRAMA DE DESPLIEGUE.

En este diagrama se describe la arquitectura física del sistema durante la ejecución de la aplicación, su topología: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos.

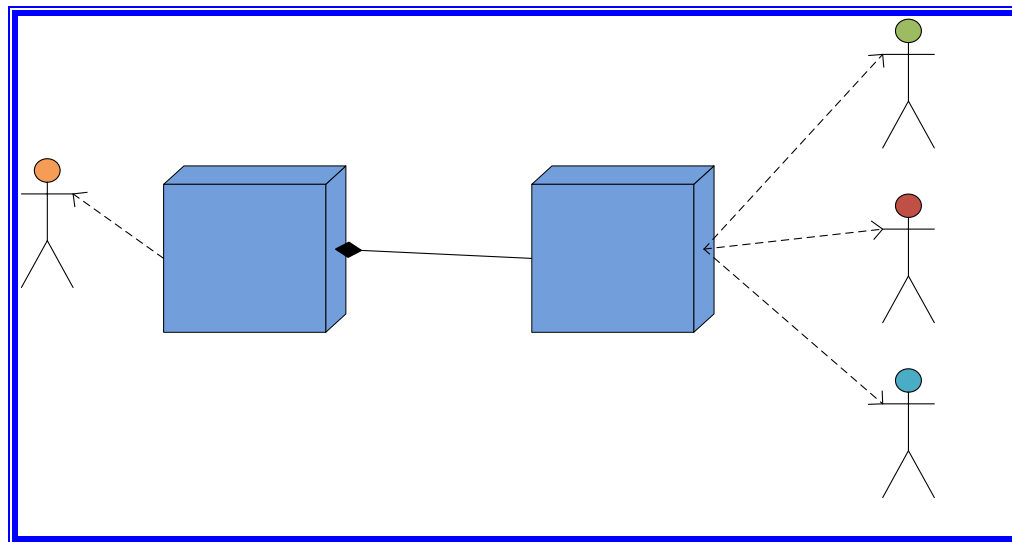


Figura 4. 15 - Diagrama de Despliegue del Sistema "S2A"

4.11 ANÁLISIS

En la fase de elaboración el objetivo del diseño es la obtención de la vista de la arquitectura del modelo de diseño, el cual está formado por los subsistemas e interfaces, las clases de diseño de la arquitectura, los diagramas de secuencia, el diseño de la base de datos, tomando como punto de partida los modelos previamente diseñados en la fase de inicio.

**Servidor
Aplicaciones
Web**

-TCP/IP

1

BD



4.11.1 Diagrama de clases de diseño

Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, los componentes que se encargarán del funcionamiento y la relación entre uno y otro. Este diagrama muestra la estructura estática del sistema reflejando las relaciones entre las clases, lo que permitirá visualizar lo que el sistema puede hacer, además de cómo puede ser construido.

Para la realización del diagrama de clases de diseño mostrado en la figura 4.13 se tomo como base las clases de análisis, y las nuevas clases que no se habían considerado al representar completamente la estructura de la aplicación.

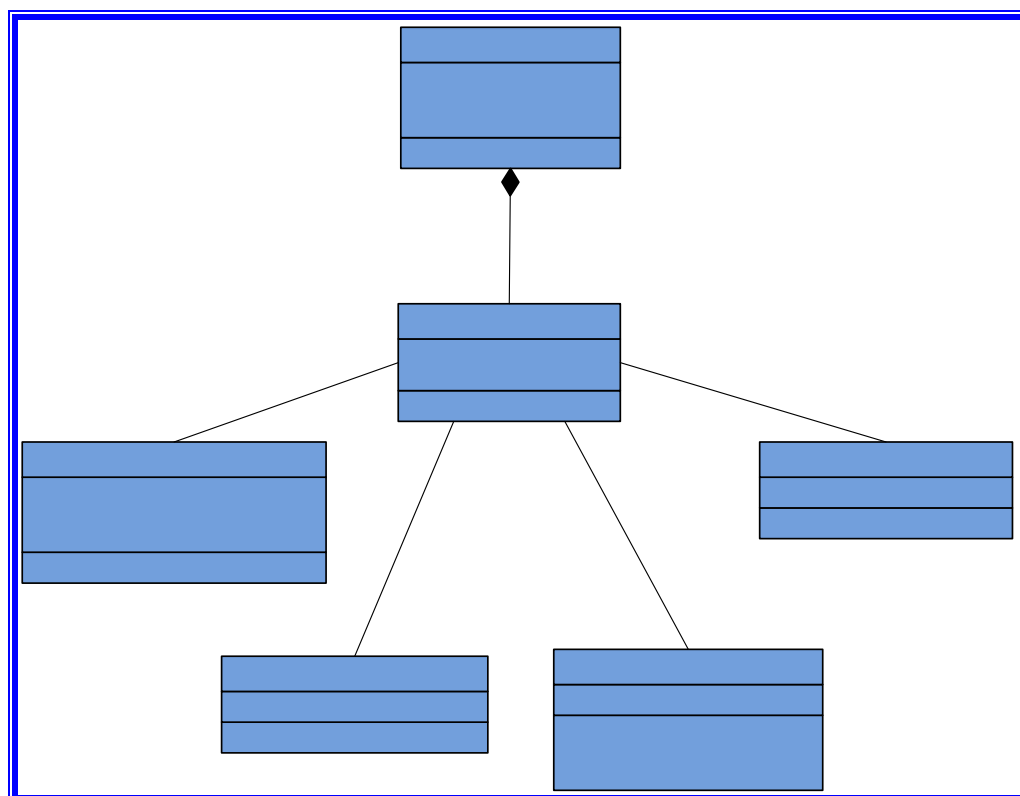


Figura 4. 16 - Diagrama de Clases de Diseño General



En el diagrama de clases de diseño general se puede observar como esta compuesta la aplicación. En este se muestra la clase “Autenticación” que representa la pantalla de entrada a la aplicación e identifica y valida a los usuarios que intenten entrar a la misma.

Luego de la identificación y validación del usuario actúa la clase “Menú” que es la pantalla de inicio de la aplicación, en la que se muestra tanto la información determinada por los *Usuarios Administradores*, como la barra de navegación por medio de la cual se activan las demás clases que intervienen en la ejecución de la misma.

La clase “AdministrarAlmacén” en esta clase se realizan diversas operaciones en las cuales se interactúa con el usuario y el sistema, en esta clase se agregan, modifican y eliminan productos de la base de datos.

La clase “ResumenVentas” es de gran importancia ya que representa el status de las ventas efectuadas y el nombre de los usuarios que la realizo.

La clase que lleva el nombre “RealizarVenta” es principal, ya que en ella se basa la aplicación. Aquí se manejan las ventas y las actualizaciones de la base de datos. La clase “ReportesVenta”, aquí se reflejan las estadísticas de las ventas por mes, se actualiza conjuntamente con la actualización del inventario al efectuar una venta.



4.11.2 Diagrama de clases de Diseño ‘Administrar Almacén’.

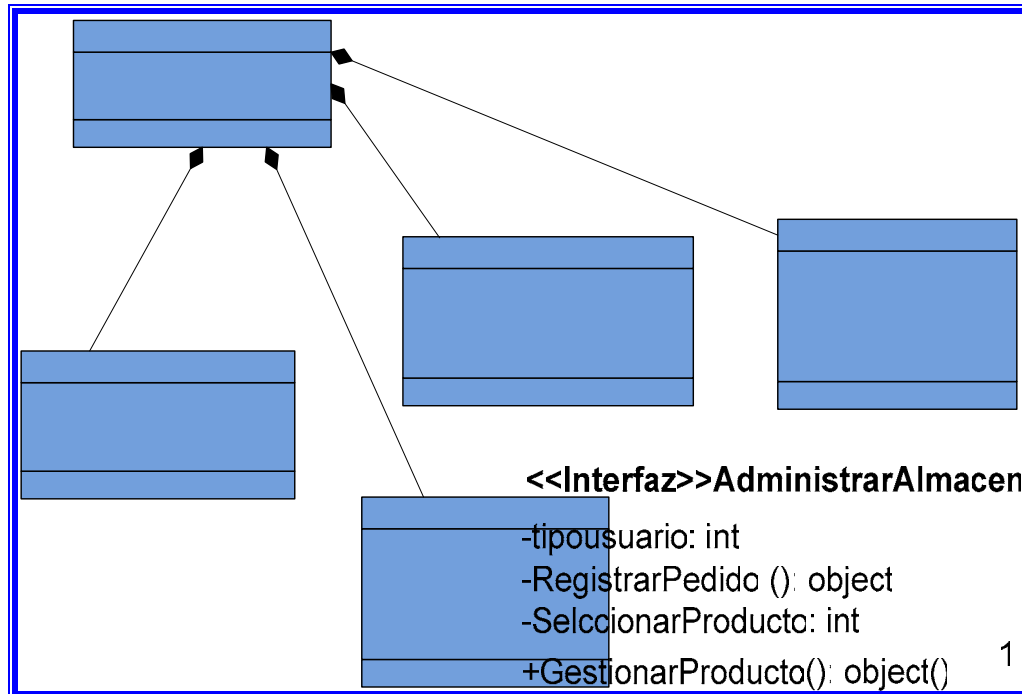


Figura 4. 17 - Diagrama de clases de Diseño ‘Administrar Almacén’

Este diagrama representa el modulo de administración de almacén de la aplicación.

La clase ‘RegistrarNuevoPedido’ representa la interfaz de ingreso de datos de los pedidos a realizar por la empresa.

La clase ‘RegistrarNuevoProducto’ representa la interfaz de ingreso de datos de los nuevos productos que se incluirán en el stock de la empresa y en la base de datos de la aplicación.

<<Interfaz>>RegistrarNuevoPedido

-NumPedido: int
 -cantidad: int
 -codigo: string
 -precio: int
 +GuardarPedido(): bool()

0..*

<<Int

-codig
 -grupo
 -nombr
 -desc
 -preci
 +Gua

0..*

<<Interfaz>

-codigo: stri
 -cantidad: in
 -precio: int
 -grupo: int
 -nombre: st
 -descripcio



La clase "ModificarProducto" representa la interfaz de ingreso de la lista de productos contenidos en la base de datos, a los cuales se les puede modificar cualquier campo seleccionado.

La clase "EliminarProducto" representa la interfaz de selección del o de los productos que se deseen eliminar de la base de datos del sistema.

4.11.3 Diagrama de clases de Diseño "Realizar Venta".

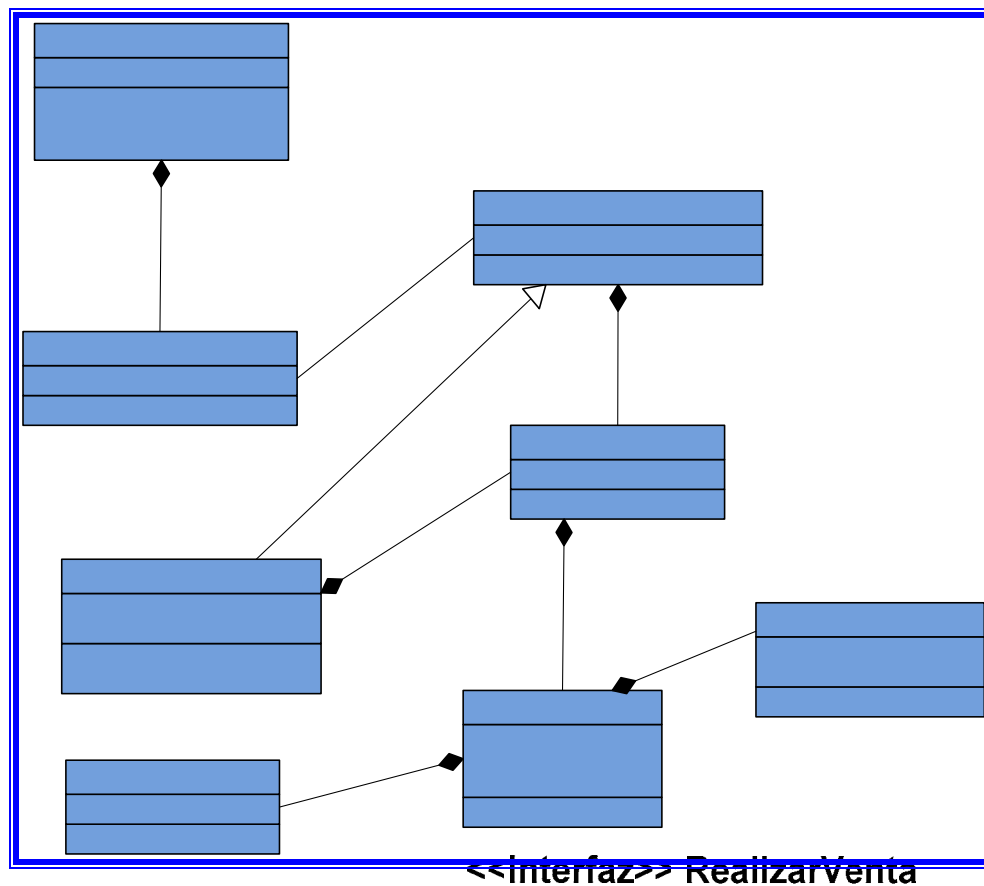


Figura 4. 18 - Diagrama de clases de Diseño "Realizar Venta"

-tipousuario: int

+SeleccionarPrd(): object()

+ConsultarInventario(): object()

+GenerarReportes(): void()

Este diagrama representa el modulo de ventas de productos.



La clase “Seleccionar Código” representa la interfaz de selección de dato (código) que fue solicitado.

La clase “Seleccionar Cantidad” al igual que la anterior, permite seleccionar la cantidad de productos solicitados por el usuario.

La clase “Consultar Disponibilidad” realiza la búsqueda de los productos seleccionados en la base de datos de la aplicación, retorna la información solicitada.

La clase “Facturar” representa la interfaz de creación de factura de los productos añadidos a la venta, incluye, la cantidad del producto, el precio, y el código del mismo.

La clase “Actualizar Inventario” y “Generar Reportes”, son operaciones que se realizan a nivel de sistema, no visibles, se realizan conjuntamente una vez ejecutado el módulo de facturación, actualizando la base de datos de la aplicación.

4.11.4 Diagrama de clases de Diseño “Visualizar Reportes Venta”.

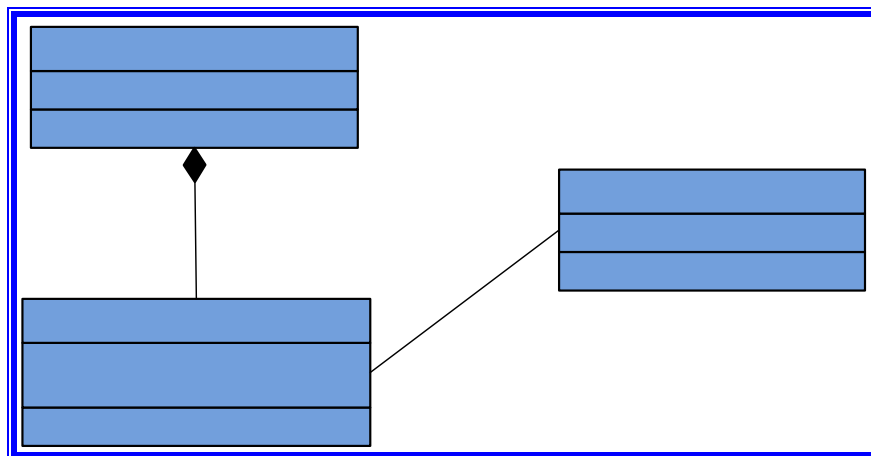


Figura 4. 19 - Diagrama de clases de Diseño “Visualizar Reportes Venta”



Este diagrama representa al módulo de muestra de información de las ventas realizados por los usuarios de la aplicación.

La clase “Usuario” representa la interfaz que muestra el personal que forma parte de la empresa, los cuales contienen estadísticas de las ventas mensuales de la empresa.

La clase restante no es visible, ejecuta operaciones que el usuario no puede ver. La clase “Reportes” representa la búsqueda y selección de la información que solicita el usuario de acuerdo a su selección.

4.12 DIAGRAMA DE SECUENCIA

Los diagramas de secuencia permiten especificar el sentido o recorrido del usuario dentro de la aplicación, es decir, muestran los procesos que se van ejecutando a medida que el mismo se mueve dentro de ella.

En estos se representan los mensajes que son enviados durante el camino de un módulo a otro a través de líneas con punta de flecha y la duración de cada uno de los módulos por los que pasa la ejecución, por medio de rectángulos cuya extensión vertical representa esa duración.

Del estudio de los requerimientos para el desarrollo de la aplicación, se derivaron los casos de uso ya especificados y entre ellos los de mayor importancia y cuya operatividad en conjunto determina o permite la realización de las operaciones de mayor valor en el manejo de eventos. Estos casos de uso son:

- Administrar Almacén
- Realizar Ventas



El trabajo en conjunto de estos módulos son determinantes en lo que respecta al cumplimiento de los requerimientos de la aplicación y deben ser los sometidos a las *Pruebas Por Unidad* y a las de *Integración*. A continuación se muestran los diagramas de secuencia de los módulos de Administrar Almacén.

4.12.1 Diagrama de Secuencia de “Modificar Producto”

Este diagrama muestra la secuencia de pasos y los módulos que se ejecutan durante la utilización de la aplicación por parte del usuario, para la modificación de un producto dentro del módulo de administración del almacén.

También se muestran los mensajes descriptivos del tipo de información que va transmitiendo de un módulo a otro.

Se da inicio al proceso de modificación de un producto cuando luego de ser validado por la aplicación para su entrada a la misma y luego al módulo de “*Administrar Almacén*”, el *Usuario* activa la ejecución del sub.-proceso “*Modificar Producto*”. Luego se continúa con la solicitud de información sobre el código, que tendrá el producto; el grupo, nombre, la descripción, precio y para culminar se guardará la información ingresada en cada solicitud.

Con los resultados exitosos de estas operaciones, se desplegará un mensaje afirmando esto y se volverá a la parte de Administración de Almacén; en la que se podrá verificar la modificación del producto.

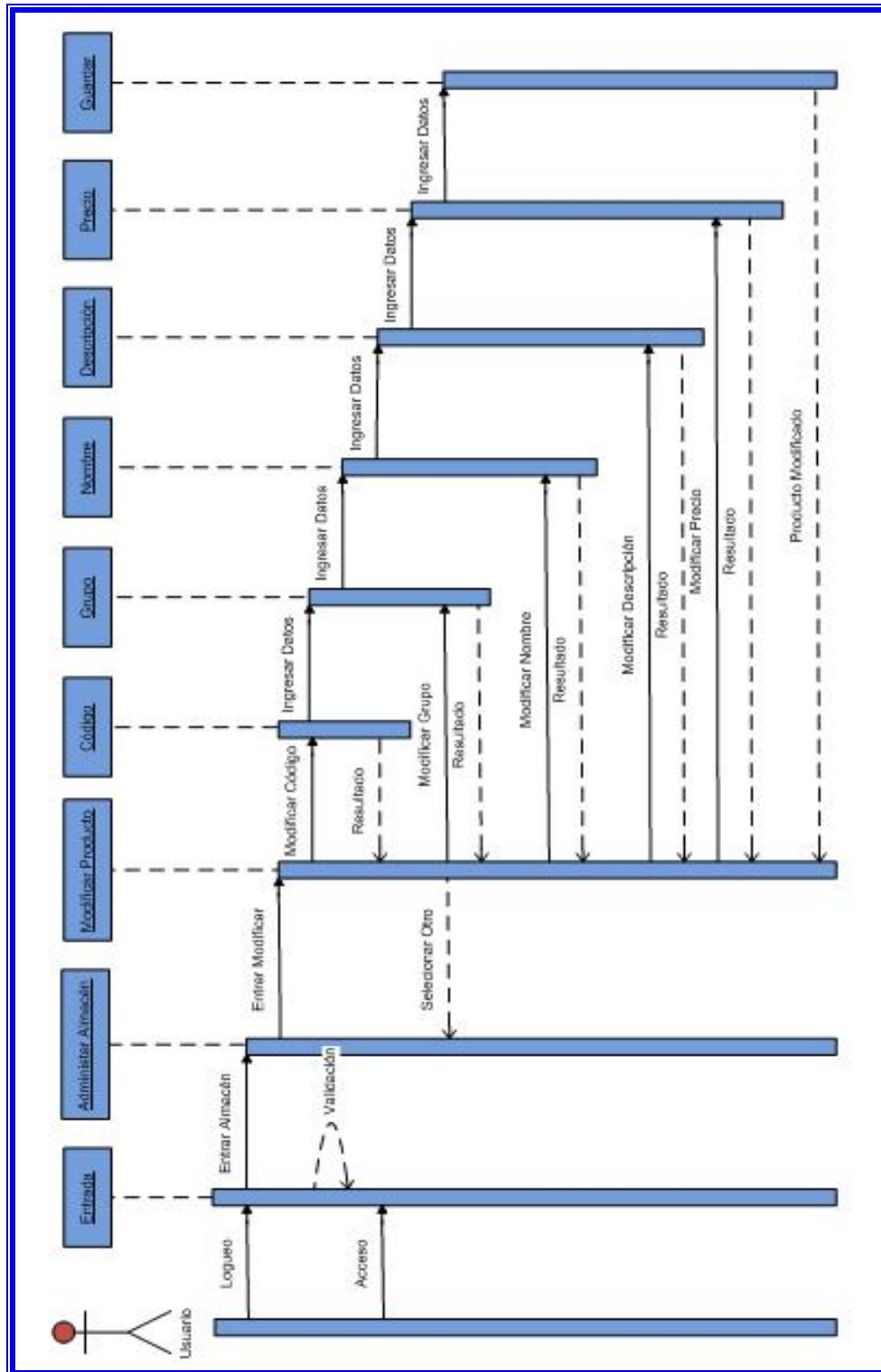


Figura 4. 20 - Diagrama de Secuencia ‘Modificar Producto’



4.12.2 Diagrama de Secuencia de "Realizar Venta"

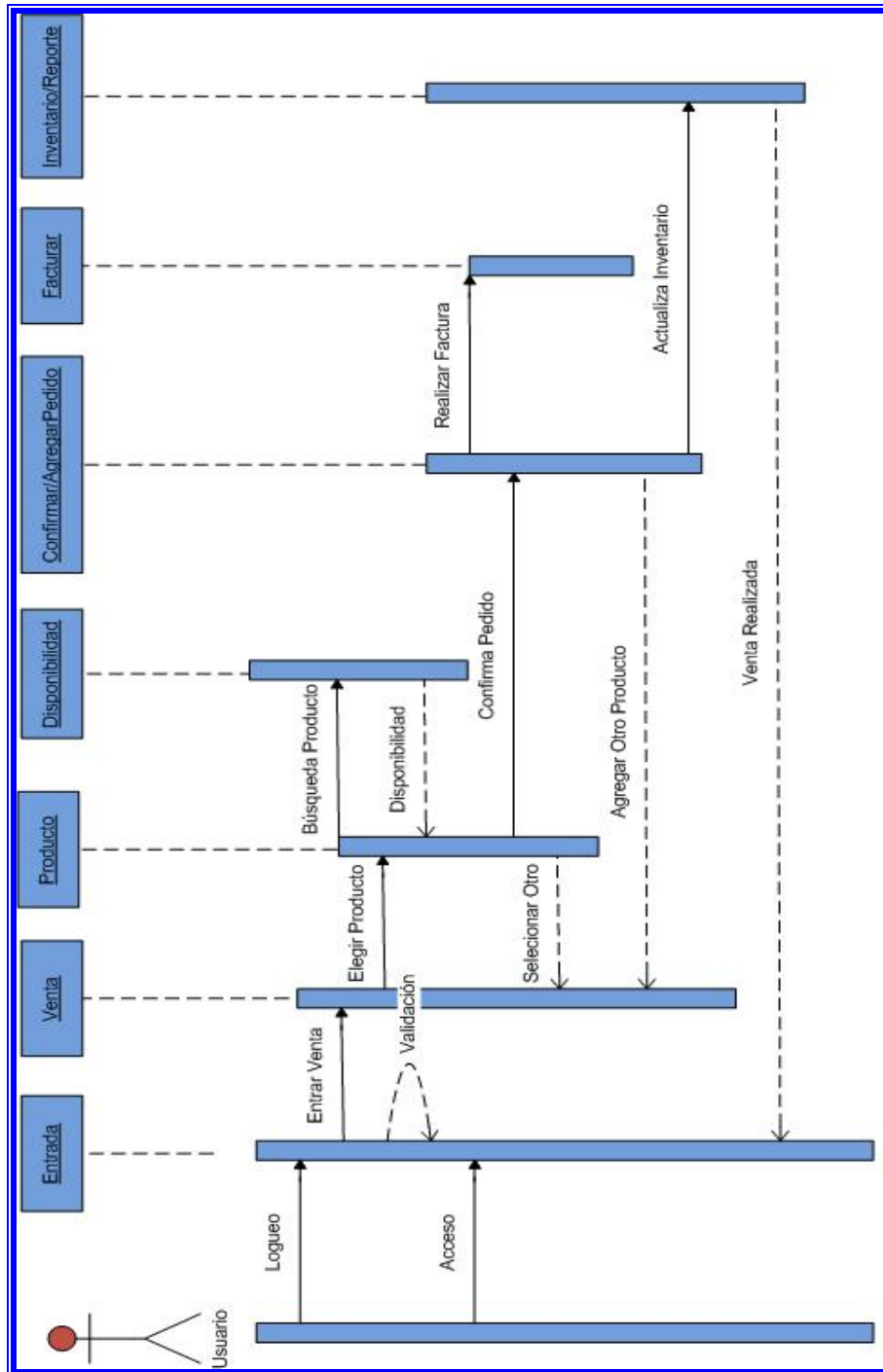


Figura 4. 21 - Diagrama de Secuencia "Realizar Venta"



Aquí se representa la secuencia de operaciones ejecutadas durante la realización de una venta.

El proceso se activa cuando luego de ser validado por la aplicación para su entrada a la misma, accede al módulo de *Menú* de la aplicación donde luego se selecciona la opción de “Realizar Venta”. La primera operación que se lleva a cabo es la seleccionar el producto a vender. Seguidamente una vez de señalar la cantidad del mismo, se prosigue a consultar la disponibilidad del producto en el stock.

Si el producto existe, la aplicación realizará la búsqueda de la información referente al mismo, y se mostrará la información solicitada. Si el producto no existe o no es suficiente la cantidad, luego de la búsqueda de los datos, aparecerá un mensaje indicado que el producto consultado no esta disponible.

Una vez ingresada toda la información del producto y consultada, se confirma la venta y se genera la factura. Finalmente se actualiza el inventario, descontando los productos vendidos de la base de datos.

4.13 DISEÑO

4.13.1 Diseño de la Base de Datos

Como se observó en los diferentes diagramas elaborados durante las dos fases del desarrollo del sistema, muchas de las tareas que se realizarán de acuerdo a los casos de uso tendrán acceso a la base de datos. Como se menciona anteriormente, en esta fase del proyecto se pretende el desarrollo de la estructura interna de la aplicación, por lo que se hace indispensable el diseño de la base de datos, como una de las partes más importantes de la estructura, ya que esta es la que permite almacenar los datos de manera que puedan ser manejados de una forma óptima.



El modelo de datos que se empleó sigue con todos los lineamientos y pautas del modelo relacional, donde se presentan los datos y las relaciones entre estos mediante el uso de tablas. Siendo este el elegido por la robustez que ofrece a la hora de manejar los datos y por ser el de mayor uso actualmente.

4.13.2 Tablas de Base de datos

Se presentarán a continuación las tablas diseñadas siguiendo el modelo relacional, las cuales son la base de la aplicación, resaltando en cada una el campo que represente la clave principal con sus respectivas descripciones:

4.13.2.1 Tabla personal

La tabla personal permite almacenar la información asociada a los usuarios de la aplicación, así como nombre, tipo de usuario, clave, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.2.

Tabla 4. 2 - ‘Personal’

Personal		
Atributo	Dominio	Descripción
Usuario	Texto	Identificador de la localización a la cual se reconocerá.
Password	Número	Clave del usuario.
NombreReal	Texto	Nombre del usuario.
Nivel	Número	Identificador del nivel del usuario
id_usuario	Número	Identificador de usuario.



4.13.2.2 Tabla Ventas

La tabla ventas permite almacenar toda la información asociada a las ventas de los productos del stock, señalando la fecha de venta, el vendedor, el monto y los productos de a misma. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.3.

Tabla 4. 3 - “Ventas”

Ventas		
Atributo	Dominio	Descripción
Id_ventas	Número	Identificador de la venta
Fecha_venta	Fecha	Fecha de realización de la venta
Vendedor	Texto	Nombre del usuario.
Monto	Número	Identificador del monto de la venta
producto	Texto	Identificador de los productos de la venta.
id_usuario	Número	Identificador de usuario.

4.13.2.3 Tabla Stock

La tabla stock permite almacenar la información correspondiente a los productos que se encontraran en la base de datos de la aplicación, así como precio, cantidad, cantidad original, abonado, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.4.

Tabla 4. 4 - “Stock”



Stock		
Atributo	Dominio	Descripción
Tipo_prod	Texto	Identificador del tipo de producto
Preciounit	Número	Identificador del precio unitario del producto.
Cant	Texto	Cantidad de productos.
CantOri	Número	Identificador de la cantidad original del producto
Abonado	Número	Cantidad de abonado del precio de los productos.
id_usuario	Número	Identificador de usuario.

4.13.2.4 Tabla Grupos

La tabla grupos permite almacenar la información asociada a los grupos de los productos de la aplicación, así como nombre, descripción, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.5.

Tabla 4. 5 - "Grupos"

Grupos		
Atributo	Dominio	Descripción
groupName	Número	Nombre del grupo del producto.
descripción	Texto	Descripción del grupo del producto.
Tipo_product	Número	Identificador del tipo de productos del grupo
id_usuario	Número	Identificador de usuario.

4.13.2.5 Tabla Productos

La tabla productos permite almacenar la información asociada a los productos de la aplicación, así como nombre, precio, número de lote, descripción, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.6



Tabla 4. 6 - ‘Productos’

Productos		
Atributo	Dominio	Descripción
Nombre	Texto	Nombre del producto.
descripción	Texto	Descripción del producto.
Precio	Número	Precio del producto
numLote	Número	Identificador del numero de lote del producto.
Lote	Número	Identificador del lote del producto.
id_usuario	Número	Identificador de usuario.

4.13.2.6 Tabla Lotes

La tabla lotes permite almacenar la información asociada a los lotes de los productos de la aplicación, así como fecha, cantidad de productos, número de lote, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.7

Tabla 4. 7 - ‘Lotes’

Lotes		
Atributo	Dominio	Descripción
Fecha	Fecha	Fecha de llegada del lote.
NumLote	Número	Número de lote.
Cantidad	Número	Cantidad de producto
Id_lote	Número	Identificador del lote
id_usuario	Número	Identificador de usuario.

4.13.2.7 Tabla niveles



La tabla niveles permite almacenar la información asociada a los niveles de los usuarios de la aplicación, así como el nombre, el tipo de usuario, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.8

Tabla 4. 8 - “Niveles”

Niveles		
Atributo	Dominio	Descripción
NombreNivel	Número	Nombre del nivel.
tipo_usuario	Número	Identificador del tipo de usuario.

4.13.2.8 Tabla Compras

La tabla compras permite almacenar la información asociada a las compras de los productos que estarán en la base de datos de la aplicación, así como fecha, cantidad, grupo, etc. La descripción de los datos que forman parte de esta tabla se muestran a continuación en la tabla 4.9.

Tabla 4. 9 - “Compras”

Compras		
Atributo	Dominio	Descripción
Fecha	Fecha	Fecha de compra.
Monto	Número	Monto de la compra
cantidad	Número	Cantidad de producto
id_usuario	Número	Identificador de usuario.

4.14 CONCLUSIÓN DE LA FASE.

En el primer flujo de trabajo, el de requisitos, se identificaron los nuevos casos de uso, que hacían falta para el desarrollo del proyecto, y se desarrollaron sus



diagramas de análisis y colaboración. Siguiendo la cronología del flujo de trabajo para la fase de elaboración, se identificaron los paquetes de análisis de los nuevos casos de uso.

Seguidamente se entró en el diseño, donde se identificó la organización en capas, se identificaron los nodos y la configuración de la red. También se desarrolló un diagrama de clase de diseño general del sistema, aquí se reflejan las diferentes relaciones impuestas entre clases, tales como las de agregación y composición.

Para complementar lo anterior, se presentó el diseño de las clases más importantes del sistema, identificando sus atributos y métodos, luego se continuó con un diagrama de secuencia para el caso de uso Administrar Almacén y Realizar Venta, en este diagrama se mostraron las instancias de los objetos del diseño y las transmisiones de mensajes entre estos. Finalmente se elaboraron las tablas que van a conformar la base de datos del sistema, dando una breve explicación de ellas.

CAPÍTULO V

FASE DE CONSTRUCCIÓN

5.1 INTRODUCCIÓN

La fase de construcción está enfocada al engranaje e integración del sistema, por medio de la construcción de versiones de prueba del software final. Estas versiones deben ofrecer la calidad necesaria para su utilización.

En esta fase se completa la arquitectura y se llega a un punto estable del proyecto, para dar comienzo a la codificación y construcción de los subsistemas. Estos son sometidos a rigurosas pruebas para garantizar el buen funcionamiento del sistema.

Como se muestra en la figura 5.1, el flujo de trabajo se destina mayormente a la finalización del diseño e implementación absoluta y en menor medida a la búsqueda de requisitos, análisis y diseño, se inicia la ejecución de las primeras pruebas de los subsistemas.

5.2 PLANIFICACIÓN DE LA FASE

En esta fase se construirá el software, para ello se selecciona el lenguaje de programación adecuado a través de un análisis de los requerimientos funcionales del sistema. También se escogerán los programas complementarios que serán utilizados por el software para ejecutar funciones que el lenguaje no contiene o simplemente no conviene su utilización.



Se da comienzo con el diseño de las interfaces de usuario, los cuales deben ser entendible y de fácil utilización. El modelo de implementación se diseña y describe en esta fase, al igual que el diagrama de componentes.

Una vez obtenido todos los diagramas, se procede a codificar cada subsistema, luego se integran para formar el sistema.

Se realizará las pruebas específicas de cada proceso y de integración del sistema, de forma que se minimice o elimine si es posible la posibilidad de un error que impida un funcionamiento normal de S2A.

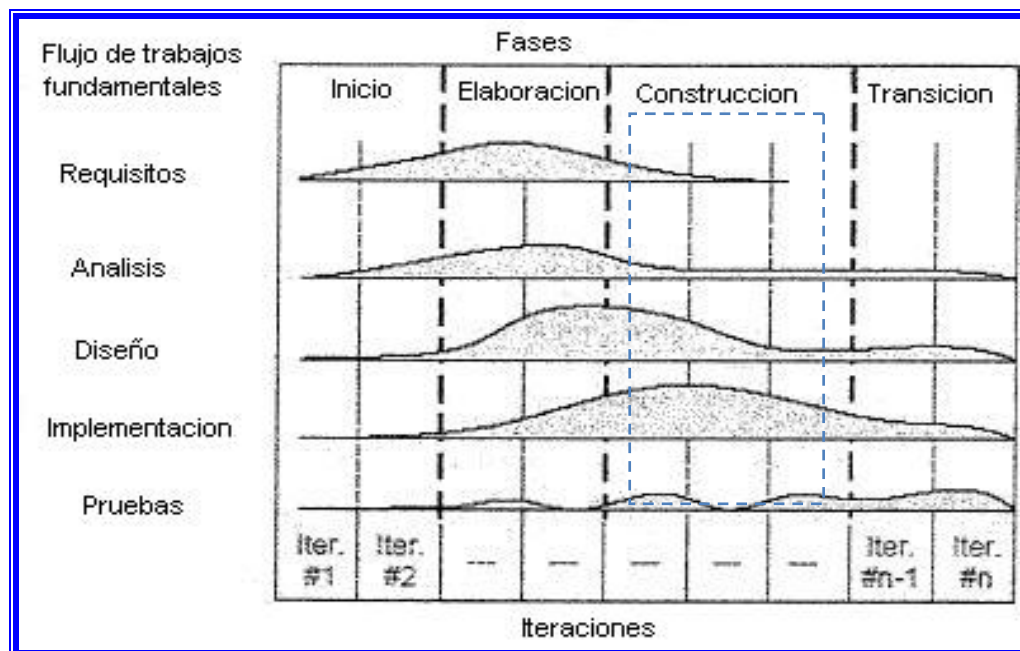


Figura 5. 1 - Diagrama de los flujos de trabajo sobre cada una de las fases (Resaltando la Fase de Construcción)



5.3 DISEÑO

Los casos de uso fueron identificados y desarrollados completamente en la fase de elaboración, por lo que ya no serán necesarios modificarlos ni identificar nuevos casos de uso en esta fase. De manera que, el diseño de esta fase se concentra en la construcción de las interfaces de los usuarios.

Se levantan los diseños preliminares de las interfaces de usuario, luego son mostradas a los usuarios finales del S2A, quienes probarán e emitirán sus criterios, con respecto a la interfaz del sistema.

5.4 DISEÑO DE LA INTERFAZ DE USUARIO

El diseño de interfaz de usuario permite al usuario interactuar con los casos de uso de manera eficiente y eficaz. Para ello son construidas de forma precisa tomando en consideración las configuraciones más importantes.

Una temprana revisión y evaluación de estas, puede prevenir muchos errores que serían más costosos en términos de esfuerzo corregir después.

Entre los aspectos más importantes de esta fase de construcción se encuentra el diseño de la ya mencionada interfaz de usuario, su importancia se basa en las siguientes características:

- Permitir que el usuario interactúe de forma adecuada.
- Proporcionar una apariencia tanto amigable como agradable y muy consistente.



La interfaz de usuario de S2A está conformada o constituida por un conjunto de ventanas o páginas que permiten seleccionar e interactuar con opciones específicas dándole dinamismo a la misma, y de esta forma facilitar el uso de la aplicación. Estas son presentadas como respuesta a eventos del usuario al interactuar con el teclado o mouse.

En las siguientes secciones serán descritas las diferentes interfaces usadas por el sistema S2A, así como también serán especificadas y detalladas las pantallas de la aplicación, mensajes de error, etc.

A continuación se presenta las interfaces más importantes de la aplicación con su codificación.

5.4.1 Registrar Nuevo Producto

En este módulo es usuario podrá ingresar los datos del nuevo producto, primero debe ingresar el código del producto y luego se continua con la declaración completa del mismo, nombre, grupo, descripción, precio.

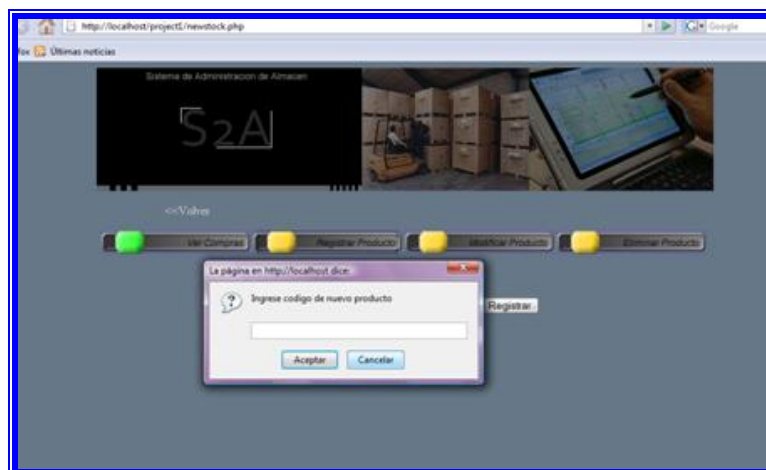


Figura 5. 2 - Registrar Producto



5.4.1.1 Codificación Registrar Producto

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>
<link rel="stylesheet" type="text/css" href="./estilos.css" />
<script language="javascript" type="text/javascript">
function aceptar()
{
    codigo=document.newP.code.value;
    grupo=document.newP.group.value;
    precio=document.newP.prize.value;
    nombre=document.newP.name.value;
    descripcion=document.newP.describe.value;
    insert="("+codigo+"",""+grupo+"",""+nombre+"",""+precio+"",""+descripcion+"")";
    window.open("./newproduct.php?insert="+insert,"_self");
}
function update()
{
    codigo=document.newP.code.value;
    grupo=document.newP.group.value;
    precio=document.newP.prize.value;
    nombre=document.newP.name.value;
    descripcion=document.newP.describe.value;
```



```
        insert="setprodcode='"+codigo+"',groupcode='"+grupo+"',nombre='"+nombre+
        "','precio='"+precio+"',descripcion='"+descripción+''''+"whereprodcode='"+codigo+"";
        window.open("./newproduct.php?update=2&update2="+insert,"_self");
    }
</script>
<body>
<?php
mysql_connect("localhost","root");
$code=$_GET['code'];
printf("<form name='newP'>");
printf("<table border='0'>");
printf("    <tr>");
printf("        <th colspan='2' align='center'>Declaracion de Producto</th>");
printf("    </tr>");
printf("    <tr>");
printf("        <td>Codigo:</td>");
printf("        <td><input type='text' name='code' value='$code'/></td>");
printf("    </tr>");
printf("    <tr>");
printf("        <td>Grupo de Producto:</td>");
printf("        <td><input type='text' name='group'/></td>");
printf("    </tr>");
printf("    <tr>");
printf("        <td>Nombre de Producto: </td>");
printf("        <td><input type='text' name='name'/></td>");
printf("    </tr>");
printf("    <tr>");
printf("        <td>Precio: </td>");
printf("        <td><input type='text' name='prize'/></td>");
```



```
printf("    </tr>");
printf("    <tr>");
printf("        <td>Descripcion: </td>");
printf("        <td><input type='text' name='describe' size='80'/></td>");
printf("    </tr>");
printf("    <tr>");

if($_GET['update']==1)
printf("    <td align='right'><input type='button' name='ok' value='actualizar'
onclick='update()'/></td>");
else
printf("    <td align='right'><input type='button' name='ok' value='aceptar'
onclick='aceptar()'/></td>");
printf("    <td align='right'><input type='button' name='ok' value='Cancelar'
onclick='window.close()'/></td>");
printf("    </tr>");
printf("</form>");

if($_GET['update']==1)
{
    $select="select ProdCode,GroupCode, Nombre, precio, Descripcion from
productos where prodcode like('".$code."')";
    $query=mysql_db_query("productos",$select);
    while($res=mysql_fetch_row($query))
    {
        printf("<script language='javascript'> ");
        printf("    document.newP.code.value='$res[0]';");
        printf("    document.newP.group.value='$res[1]';");
        printf("    document.newP.name.value='$res[2]';");
    }
}
```



```
        printf("    document.newP.prize.value='$res[3]';");
        printf("    document.newP.describe.value='$res[4]';");
        printf("</script>");
    }
}
if($_GET['update']==2)
{
    $update_x="update productos ".$_GET['update2'];
    $update=str_replace("\\" , "" , $update_x);
    //echo $update;
    mysql_db_query("productos", $update);
}
if($_GET['insert'])
{
    $valores_x=$_GET['insert'];
    $valores=str_replace("\\" , "" , $valores_x);
    //echo $valores;
    $query="insert into productos values ".$valores;
    //echo $query;
    mysql_db_query("productos", $query);
    printf("<script>
    alert('Listo');
    window.close();
    </script>");
}
?>
</body>
</html>
```



5.4.2 Realizar Venta

En este módulo el usuario realiza la solicitud de venta de un producto, donde seleccionara el código del producto, la cantidad que será consultada su disponibilidad en stock, y se confirmará la venta para facturar.

Marcar	Código	Lote	Cantidad	Nombre	Descripción
<input checked="" type="checkbox"/>	C023	DCO022	5	Arroz Primor	Arroz 1kg

Agregar

Pagar En Cuotas Pagar de Contado

Figura 5. 3 - Interfaz Realizar Venta

5.4.2.1 Codificación Registrar Producto

```
?php
session_start();
include_once("../funcionesphp.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```



```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    background-color: #667788;
}
-->
</style></head>
<script language="javascript" type="text/javascript" src="./funcionesjava.js"/>
</script>
<script language="javascript" type="text/javascript">
var car="";
function confirmar(cuotas)
{
    cantCuo=1;
    cancel=1;
    if(cuotas==1)
    {
        cantCuo=prompt("Ingrese la cantidad cuotas");
        cancel=prompt("Ingrese la cantidad cuotas a cancelar ahora:");
    }
    for (i=0;i<document.tabla.elements.length;i++)
    {
        if(document.tabla.elements[i].type=="checkbox")
        {
            addtocar(document.tabla.elements[i+2].value,document.tabla.elements[i+1].value,document.tabla.elements[i].checked,document.tabla.elements[i+3].value,0);
        }
    }
}
```



```
    }
    //alert(car);
    car2=car;
    window.open('./index.php','_self');
    window.open("./planilla.php?car2="+car2+"&cuotas="+cantCuo+"&cancel="+
cancel,"_new","width=900");
}
function volver()
{
    history.go(-1);
}
</script>
<bodsy>
<div align="center"></div>
<p>&nbsp;</p>
<p>
<?php
    $link=mysql_connect("localhost","root");
    $car_aux=$_SESSION['carrito'];
    //echo $car_aux;
    printf("<script language=\"javascript\" type=\"text/javascript\">
        car2=\"\$car_aux\";</script>");
    $token="|";
    $cant=split_car($car_aux,$token,$code_array,$cant_array,$lote_array);
    printf('<form name="tabla" id="tabla"><table width="800" border="1"
align="center" bordercolor="#000000" id="princ" hspace="1">
        <tr>
            <th width="60">Marcar</th>
```




```
<th width="60">C&oacute;digo</th>
<th width="100">Lote</th>
<th>Cantidad</th>
<th>Nombre</th>
<th>Descripci&oacute;n</th>
</tr>');
for($j=0;$j<=$cant;$j++)
{
$sentence="select prodcode,Numlote,nombre,cant, descripcion from stock join
productos using (ProdCode) where productos.prodcode like ('$code_array[$j]')
and NumLote like ('$lote_array[$j]') order by nombre";
//echo $sentence;
$query=mysql_db_query("productos",$sentence);
while($res=mysql_fetch_row($query))
{
printf("<tr>");
printf("<td><inputtype='checkbox'name='c_&#x27;$res[0]'/></td>");
printf("<td id='code_&#x27;$i'> $res[0] </td>");
printf("<td id='lote_&#x27;$i'> $res[1] </td>");
printf("<td><input type='text' name='text_&#x27;$i' style='text-align:center' size='4' value='&#x27;$cant_array[$j]'/> </td>");
printf("<td> $res[2] </td>");
printf("<td> $res[4] </td>");
printf("</tr>");
printf("<script language='javascript' type='text/javascript'>
document.tabla.c_&#x27;$res[0].checked=true;
</script>");
printf(' <input type="hidden" name="code_&#x27;$i"
value="%s">', $res[0]);
```



```
                printf('    <input        type="hidden"        name="lote_$i"
value="%s">', $res[1]);
            }
        }
        printf('</table>    <div align="center">    <br/><br/><input    type="button"
name="agregar" value="Agregar" onclick="volver()"/> </div></form>');

        printf('<div align="center"> <br/><br/><input type="button" name="aceptar1"
value="Pagar En Cuotas" onclick="confirmar(1)"/> &nbsp;');
        printf('<input    style="background-color:#555555;"        type="button"
name="aceptar2" value="Pagar de Contado" onclick="confirmar(0)"/> </div>');
?>
</p>
<p>&nbsp;</p>
</body>
</html>

//Llenado de formulario de la venta.
function salir()
{
    window.open('./index.php', '_parent');
    window.close();
}
</script>
<body>
<div    align="center"></div>
<br/>
<br/>
```



```
<?php
if($_GET['car2'])
    $_SESSION['venta']=$_GET['car2'];
$link=mysql_connect("localhost","root");
$cant=0;
if ($_SESSION['venta'])
{
    $car_aux=$_GET['car'].$_SESSION['venta'];
    $_SESSION['venta']=$car_aux;
    printf("<script language=\"javascript\" type=\"text/javascript\">
           car=\"\$car_aux\";
           </script>");
    $code=strtok($car_aux, "|");
    $i=0;
    while($code)
    {
        $code_array_aux_x[$i]=$code;
        $i++;
        $code=strtok("|");
    }
    for($j=0;$j<$i;$j++)
    {
        $code_array_x[$j]=strtok($code_array_aux_x[$j], "-");
        $cant_array_x[$j]=strtok("-");
        $lote_array_x[$j]=strtok("-");
        //          echo "CODIGO:".$code_array_x[$j]." CANT:".$cant_array_x[$j].
        "<br/>"; //descomentar para depurar
    }
    $cant=$i;
}
```



```
$k=0;
}
printf('<table width="400" border="0" align="center" bordercolor="#000000"
id="princ" hspace="1">
    <tr>
        <th width="80">Codigo</th>
        <th width="80">NumLote</th>
        <th width="150">Producto</th>
        <th width="80">Cant</th>
        <th width="15\80" align="center"> Precio Unitario
(Bs.)</th>
        <th width="100">Total (Bs.)</th>
    </tr>');
$total=0;
for($j=0; $j < $cant; $j++)
{
    $query="select  nombre,preciounit,preciounit*$cant_array_x[$j]  from
stock join productos using (ProdCode) where Prodcode like ('$code_array_x[$j]') and
NumLote like ('$lote_array_x[$j]')";
    //echo $query;
    $res=mysql_db_query("productos",$query);
    if($r=mysql_fetch_row($res))
    {
        $update="update";
        mysql_db_query("productos",$update);
        printf("
        <tr>
            <td align='center'>$code_array_x[$j]</td>
            <td align='center'>$lote_array_x[$j]</td>
```



```

        <td align='center'>$r[0]</td>
        <td align='center'>$cant_array_x[$j]</td>
        <td align='center'>$r[1]</td>
        <td align='center'>$r[2]</td>
    </tr>");
    $total+=$r[2];

    $update="update stock set cant=cant-$cant_array_x[$j] where Prodcod
like ('$code_array_x[$j]') and NumLote like ('$lote_array_x[$j]')";
    $res=mysql_db_query("productos",$update);
    // echo '<a>'.$insert.'</a>';
    }

}

$insert='insert into venta (fecha, vendedor, monto, productos, cantcuotas,
canceladas)
values(now(),''.$_SESSION['user'].''.$total.'''.$_SESSION['venta'].''.$_GET['cuotas'].''.$_GET['cancel'].')';
    $res=mysql_db_query("productos",$insert);
    //echo '<a>'.$insert.'</a>';

printf("    <tr>
        <td></td>
        <td></td>
        <td align='center'></td>
        <td align='center'>TOTAL</td>
        <td align='center'>Bs. $total</td>
        <td></td>
    </tr>");
```



```
printf("</table>");
$_SESSION['venta']="";
printf("<input type='button' value='Cerrar' onclick='salir()'>");
printf("<br/><br/><br/><br/><a>NOTA: Usted ha elegido pagar en %s cuotas
de %.2f</a> ",$_GET['cuotas'],$total/$_GET['cuotas']);
$query="select max(id) from venta";
//echo $query;
$res=mysql_db_query("productos",$query);
$r=mysql_fetch_row($res);
printf("<br/><a align='center'> Identificador de Venta: $r[0]</a>");
?>
</body>
</html>
```

5.5 IMPLEMENTACIÓN

En este flujo de trabajo es donde se implementa y se llevan a cabo las pruebas de unidad de todos los componentes del sistema. Este flujo de trabajo es decisivo para la culminación del desarrollo del sistema S2A, ya que es aquí donde se lleva a cabo la construcción de todos los que conforman la aplicación, obteniendo de esta manera una versión confiable del sistema.

5.5.1 Lenguajes de Programación

Para realizar la codificación de los componentes que conforman el sistema se utilizó el lenguaje de programación PHP. Para diseño de las paginas fue usando Html, JavaScript y hojas de estilo ccs, con Apache Server como servidor de aplicaciones y manejador de transacciones. Adicionalmente, se utilizó Mysql como manejador de Base de Datos. Todas estas herramientas software libre.



5.5.2 Características del lenguaje PHP

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.

Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.



El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Un sitio con páginas webs dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.

Seguridad. PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.

PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.



5.5.3 Características de Html

HTML es el acrónimo inglés de HyperText Markup Language que se traduce al español como Lenguaje de Marcas Hipertextuales. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para web.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit, Kwrite, el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim etc.

Existen además, otros programas para la realización de sitios Web o edición de código HTML, como por ejemplo Microsoft FrontPage. También existe el famoso software de Macromedia (que adquirió la empresa Adobe) llamado Dreamweaver, siendo uno de los más utilizados en el ámbito de diseño y programación Web. Estos programas se les conoce como editores WYSIWYG o What You See Is What You Get (en español: “lo que ves es lo que obtienes”). Esto significa que son editores en los cuales se ve el resultado de lo que se está editando en tiempo real a medida que se va desarrollando el documento.

5.5.4 Características de JavaScript

JavaScript es un lenguaje interpretado, posee una característica que lo hace especialmente idóneo para trabajar en Web, ya que son los navegadores que utilizamos para viajar por ella los que interpretan (y por tanto ejecutan) los programas



escritos en JavaScript. De esta forma, podemos enviar documentos a través de la Web que llevan incorporados el código fuente de programas, convirtiéndose de esta forma en documentos dinámicos y dejando de ser simples fuentes de información estáticas.

Los programas en JavaScript no son la primera forma que conoce la Web para transformar información, dado que el uso de CGIs está ampliamente difundido. La diferencia básica que existe entre un programa CGI y uno escrito en JavaScript es que el CGI se ejecuta en el servidor de páginas Web mientras que el programa en Javascript se ejecuta en el cliente (es decir, en el navegador). Por regla general, el CGI necesita unos datos de entrada (que normalmente se proporcionan mediante un formulario), los procesa y emite un resultado en forma de documento HTML. Esto implica tres transacciones en la red:

- Cargar la página del formulario
- Enviar los datos al servidor, y
- Recibir la nueva página que ha generado el CGI.

Por el contrario, los programas escritos en JavaScript se ejecutan en el navegador del cliente, sin necesidad de que intervenga el servidor. De esta forma, una sola transacción basta para cargar la página en la que se encuentra tanto el formulario, para los datos de entrada, como el programa en JavaScript que proporciona los resultados.

Las dos principales características de JavaScript son, por un lado que es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones), y por otro JavaScript es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza (Windows y sistemas X-Windows). Esto implica que gran parte de la programación en JavaScript se centra en describir objetos (con sus



variables de instancia y métodos de "clase") y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos.

Por último, es necesario resaltar que hay dos tipos de JavaScript. Por un lado está el que se ejecuta en el cliente. Este es el JavaScript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un JavaScript que se ejecuta en el servidor. Este, más reciente, se denomina LiveWire JavaScript.

5.5.5 Hojas de estilo CCS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un



sintetizador de voz. El documento HTML en sí mismo es más claro de entender y se puede reducir considerablemente su tamaño.

5.5.6 Servidor HTTP Apache

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

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

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 48% de los sitios web en el mundo y decreciendo su cuota de mercado.

5.5.7 MySQL

MySQL surgió alrededor de la década del 90, Michael Widenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a mSQL.



Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (**SQL**) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. Entre las características encontramos las siguientes:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas



5.6 CODIFICACIÓN DE LOS COMPONENTES DE S2A

Para la construcción del software es importante codificarlo en pasos manejables, de tal manera que se puedan hacer pasos pequeños y se pueda realizar la integración y prueba de una manera más fácil.

5.6.1 Diagrama de componentes de S2A.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

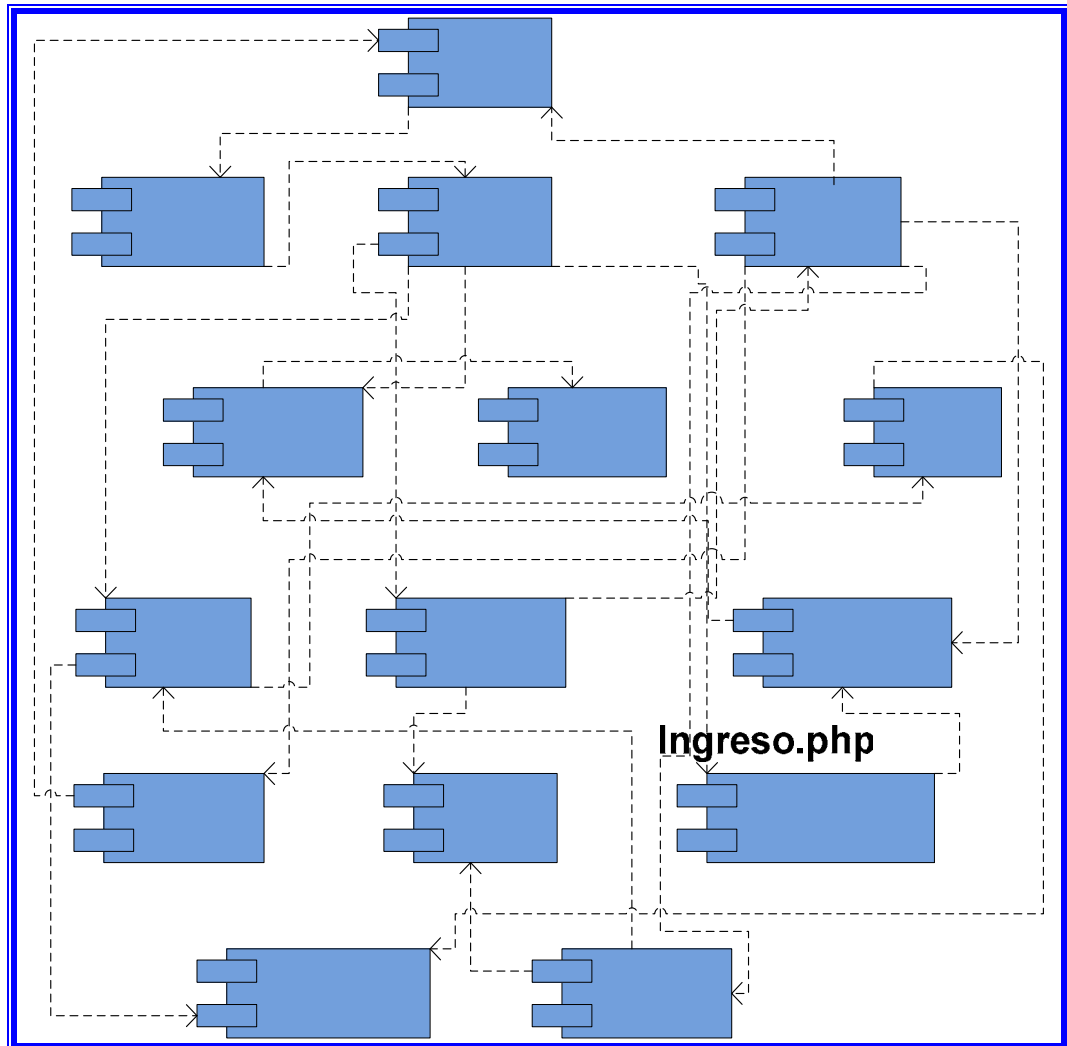


Figura 5. 4 - Diagrama de Componentes de la Aplicación.

Gráficas.php

5.7 FLUJO DE TRABAJO DE PRUEBAS

El objetivo principal de llevar a cabo un flujo de trabajo de pruebas es asegurarse de que todos los subsistemas de todos los niveles hasta las capas específicas de la aplicación funcionen y no presenten ningún problema.

Stock.php



5.7.1 Partición Equivalente

Una partición equivalente es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos. El diseño de casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

5.7.2 Identificación de las clases de equivalencia

1. Solo números.
2. Solo caracteres.
3. Caracteres y números
4. Ningún carácter

5.7.3 Grupo de tipos de entrada de Datos

1. Id_Ocup: Hace referencia al campo de entrada del nivel de usuario, solo recibe caracteres numéricos.
2. Nombre: Hace referencia al campo de ingreso del pseudónimo de autenticación del usuario dentro de la empresa, este debe ser alfanumérico.
3. Password: Hace referencia al campo de ingreso de la clave de acceso asociada al usuario, este debe ser numérico.



5.7.4 Aplicación de Casos de Prueba

Tabla 5. 1 - Aplicación de los casos de prueba.

Grupo	Casos De Prueba	Válida	No Válida	Clases De Equivalencia
1	553356		X	1
1	2	X		1
1	ABBdefg		X	2
1	45ggh46		X	3
1	“ ”		X	4
2	553356		X	1
2	ABBdefg	X		2
2	45ggh46		X	3
2	“ ”		X	4
3	553356	X		1
3	ABBdefg		X	2
3	45ggh46		X	3
3	“ ”		X	4

5.7.5 Grupo de Tipo de entradas de Datos Modificar Productos

1. Número de Pedido: Hace referencia al campo de entradas del identificador del pedido que se va a realizar, solo recibe caracteres numéricos.

2. Cantidad de Producto: Hace referencia a un campo numérico que captura la cantidad de productos que tendrá el pedido.

3. Código de Producto: Hace referencia al campo de ingreso del código de los productos del pedido, este debe ser alfanumérico.



4. Precio Unitario: Hace referencia al campo de ingreso del precio de cada producto del pedido, este debe ser numérico.

5.7.6 Aplicación de Casos de Prueba

Tabla 5. 2 - Aplicación de los casos de prueba.

Grupo	Casos De Prueba	Válida	No Válida	Clases De Equivalencia
1	553356	X		1
1	ABBdefg		X	2
1	45ggh46		X	3
1	“ ”		X	4
2	553356	X		1
2	ABBdefg		X	2
2	45ggh46		X	3
2	“ ”		X	4
3	553356		X	1
3	ABBdefg		X	2
3	45ggh46	X		3
3	“ ”		X	4
4	553356	X		1
4	ABBdefg		X	2
4	45ggh46		X	3
4	“ ”		X	4



5.8 Pruebas

El propósito esencial de realizar las pruebas a la aplicación es el resultado de la implementación mediante las pruebas de cada construcción, incluyendo tanto las construcciones internas como intermedias, así como la versión final del sistema, con la finalidad de descubrir posibles defectos o errores que pudiera presentar la aplicación durante su ejecución; Estas pruebas nos permiten verificar si el sistema satisface las especificaciones requeridas por medio de los procesos de ejecución y evaluación. En este flujo de trabajo se realizarán pruebas por unidad.

5.8.1 Pruebas de Unidad

Para las pruebas de unidad se tomaron en cuenta los módulos críticos o de mayor importancia dentro de la aplicación.

Entre los módulos que componen la aplicación se destacan como los más importantes por sus funciones, los siguientes:

5.8.1.1 Administrar Almacén.

En este módulo se configuran y modifican muchos factores propios de la administración del stock de productos. Como el ingreso de nuevos pedidos y productos, eliminación o bien sea la modificación de los mismos, para esto se requiere del acceso y modificación de la base de datos.

5.8.1.2 Realizar Venta.

Aquí se lleva a cabo la realización de venta de productos, se hace un interacción directa con la base de datos del sistema, se realizan consultas y actualizaciones a la misma, de manera simultánea con otras operaciones.

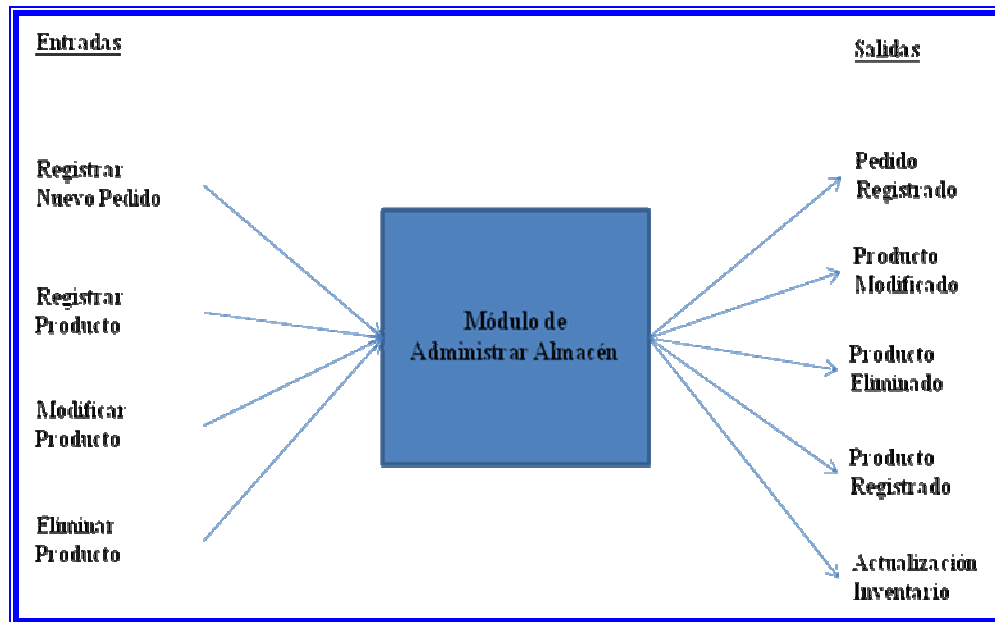


Figura 5. 5 - Pruebas Administrar Almacén

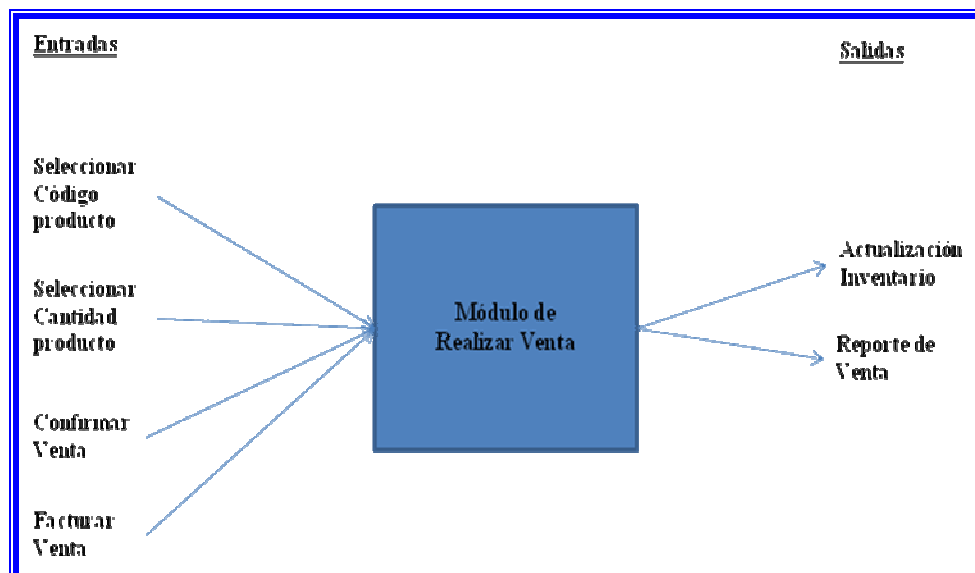


Figura 5. 6 - Pruebas Realizar Venta



5.9 CONCLUSIÓN DE LA FASE

Uno de los objetivos de la fase de construcción es desarrollar el software a partir de una línea base de la arquitectura ejecutable, hasta el punto que esté listo, para ser transmitido a la comunidad de usuarios. Aquí se detallan y analizan los requisitos restantes, la mayor parte de este flujo de trabajo fue realizada en las dos fases anteriores. El diseño juega un papel importante, y es en esta fase en la que tiene lugar la mayor parte del trabajo de los flujos de implementación y pruebas.

Dichos objetivos se cumplieron de una forma satisfactoria, ya que se acumuló la información necesaria para concluir esta fase. En esta fase se obtuvo información suficiente para realizar un diseño óptimo del Sistema S2A (Sistema de Administración de Almacén), el cual comenzó en las fases anteriores.

Al final de la fase de construcción se ha llevado el producto software a su versión operativa.

CONCLUSIONES

En primer lugar, es importante acotar que el módulo propuesto no es en ningún caso un sistema integral, pues se limita únicamente a manejar las transacciones de compra y de venta de una empresa, se integra a un sistema automatizado de gestión administrativa, sin embargo no transfiere de manera automática sus transacciones para alimentar la contabilidad del sistema y tampoco maneja la relación de compras con la conciliación bancaria del Sistema

Se desarrolla un módulo de compra/venta integrado entre las dos gestiones, con un interfaz de datos que permita intercambiar datos con el sistema administrativo, sin contemplar la gestión contable.

Utilizando el *Proceso Unificado Racional* fue posible la visualización, representación, diseño y modelado de la aplicación; los distintos diagramas considerados en *UML*, como herramienta del *proceso unificado racional* hicieron posible la representación de las diferentes fases de desarrollo. Los casos de uso para las operaciones que debía realizar la aplicación según las necesidades del usuario. Las clases de análisis para la identificación y diferenciación entre los procesos y entidades que intervienen.

En la fase de inicio fueron capturados y analizados los requisitos necesarios para dar origen al modelo general del sistema, fueron identificados los actores y casos de usos más importantes del sistema permitiendo una visión muy general del mismo.

En la fase de elaboración se logro establecer la base de la arquitectura del Sistema y recopilar la información necesaria para realizar un análisis del mismo.



En la fase de construcción se especificaron y analizaron los requisitos restantes, fueron diseñadas las diferentes interfaces del sistema, se realizó la implementación y las pruebas necesarias con el fin de dejar sentado un prototipo de alta fidelidad para ser usado por los usuarios finales.

El diseño de la base es robusto, se utilizaron bases de datos relacionales para garantizar la fiabilidad y consistencia de los datos.

La interfaz de usuario fue diseñada para evitar errores en el ingreso de información, logrando obtener confiabilidad en los datos que se registran. Este diseño está basado en el estilo propio de la empresa, fijado para mantener un estándar de interfaces en aplicaciones propias de la compañía.

El Sistema Manejador de Base de Datos utilizado fue MySQL ya que permite de una manera fácil y clara manipular los datos guardados en un servidor remoto al que acceden simultáneamente a la misma base de datos, distintos usuarios.



RECOMENDACIONES

- Se recomienda demostrar la garantía de los sistemas de automatizados ante situaciones reales, siempre y cuando la información provista sea confiable.
- La implementación de cualquier módulo de un sistema de automatizado que involucre datos financieros debe estar completamente avalada por los especialistas, sin que quede duda alguna de su efectividad.
- Se recomienda la implantación de este sistema utilizando herramientas bajo software libre, dando con esto cumplimiento al decreto presidencial 3390.
- Se recomienda realizar mantenimiento o mejoras del sistema periódicamente, a fin de garantizar resguardo de la información y el buen funcionamiento del mismo.
- Es importante ejecutar auditorías al sistema, para el evaluar la eficiencia del mismo y así alargar su vida útil. Esto se realiza mediante la agregación de nuevos procedimientos que adaptan el sistema a las necesidades y plataformas del momento.



BIBLIOGRAFIA

1. Domínguez, A. (2005). **Software Libre**. Mc GrawHill. Mexico
2. Ríos, A., Alcocer, J. y Oller, A. (2005). **Desarrollo de una plataforma de VoIP basada en Software Libre**. Prentice Hall
3. Sillitti, A. y Succi, G. (2006). **Agilidad y desarrollo de Software Libre**. Mc GrawHill. México
4. Velasco, M. y Gotilla, N. (2006). **Diseño de un sistema de información bajo licencia de software libre para el manejo de las actividades administrativas de los módulos compra y venta en una pequeña y mediana empresa**. Trabajo Especial de Grado, no publicado. Universidad de Oriente. Puerto la Cruz
5. Porta, M. (2003). **PyMes**. Urania. Venezuela
6. Rosales, R. (2002). **Estrategias gerenciales para la pequeña y mediana empresa**. Ediciones IESA. Caracas
7. Araujo, M. y Fernández, D. (2005). **Mitos y Realidades del Software Libre en la Administración Pública**. Mc GrawHill. Mexico
8. González, J. y Koch, S. (2006). **El software libre como objeto de estudio**. Mc GrawHill. Mexico
9. Kon, M. (2005). **El software libre**. Pearson. México



10. Abad, M. (2004). **Licencias**. Mc GrawHill. Mexico

11. Sabino, C. (2002). **El proceso de investigación**. Ensayo. Medellín

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

TÍTULO	“DESARROLLO DE UNA APLICACIÓN PARA AUTOMATIZAR LOS MÓDULOS DE COMPRA Y VENTA DE UN SISTEMA ADMINISTRATIVO PARA UNA EMPRESA BAJO LICENCIA DE SOFTWARE LIBRE”
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Pacheco R, Romina	CVLAC: 16.222.423 E MAIL: romishan03@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALABRAS O FRASES CLAVES:

Compra, Venta, Transacción, Software Libre, Sistema, Módulo

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
Ingeniería y Ciencias Aplicadas	Ingeniería en Computación

RESUMEN (ABSTRACT):

La presente investigación es el producto de un estudio minucioso para desarrollar una aplicación informática para automatizar los módulos de compra y venta de un sistema administrativo para una empresa, bajo licencia de software libre. Los resultados de la investigación determinan que los mecanismos de transmisión de datos entre el sistema administrativo y los módulos de compra y venta propuestos requieren de un interfaz adicional para la interacción de los datos y la consolidación contable. Por ende, la investigación se redireccionó a presentar como propuesta de desarrollo de módulos de compra y venta integrada parcialmente y no interactiva con el módulo contable. Esto solo limita la disponibilidad en línea de la información desde otros módulos del sistema, mas puede ser, mediante un mecanismo de actualización y consolidación de datos, en una frecuencia diaria, el cruce de los datos con el módulo contable. Es de hacer notar que el sistema si accede al inventario pues los datos del mismo se encuentran en una base de datos de acceso simple mediante el lenguaje de base de datos SQL y en este caso las transacciones de compra y venta pueden realizarse en línea con la actualización

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
	ROL	CA	AS X	TU	JU
Victor Mújica	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU X
Claudio Cortínez	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU X
José Luis Bastardo	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU

FECHA DE DISCUSIÓN Y APROBACIÓN:

2008	07	07
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
Automatización de Módulos Administrativos.doc	Application/MSWord

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J
K L M N O P Q R S T U V W X Y Z . a b c d e f g h i j k l m n o p q r s t u
v w x y z . 0 1 2 3 4 5 6 7 8 9 .

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ **Intemporal** _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente. Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

ARTÍCULO 44: "LOS TRABAJOS DE GRADO SON DE EXCLUSIVA PROPIEDAD DE LA UNIVERSIDAD DE ORIENTE Y SOLO PODRÁN SER UTILIZADOS A OTROS FINES CON EL CONSENTIMIENTO DEL CONSEJO DE NÚCLEO RESPECTIVO, QUIEN LO PARTICIPARÁ AL CONSEJO UNIVERSITARIO"

ROMINA HELEN PACHECO RAMNARINE

AUTOR

VICTOR MÚJICA

TUTOR

CLAUDIO CORTÍNEZ

JURADO

JOSÉ BASTARDO

JURADO

POR LA SUBCOMISIÓN DE TESIS