

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS

**“DESARROLLO DE UNA APLICACIÓN DE PUNTOS
DE VENTA EN AMBIENTE INTRANET E INTERNET”**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

**SR. DAVID ISRAEL BARRIGAS JÁCOME
SR. PABLO ANDRÉS BARRIGA TORRES**

Sangolquí, 28 de Septiembre del 2006

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los Sres. Pablo Andrés Barriga Torres Y David Israel Barrigas Jácome como requerimiento parcial a la obtención del título de Ingenieros en Sistemas e Informática.

11 de diciembre de 2010

Ing. Luis A. Escobar

DEDICATORIA

Dedico esta tesis a las personas que me acompañaron en las buenas y en las malas durante todo el periodo universitario, mis padres y mis compañeros, que más que ser compañeros de universidad llegaron a ser verdaderos amigos. Empiezo por mi compañero de tesis, Pablo Barriga que ha sido como un hermano. Mis compañeros de proyectos, Fernando Morales y Oswaldo Araujo, amigos siempre prestos para ayudar. El mejor grupo de amigos, compañeros y sanos competidores de estudio, Edwin Alcívar, Patricio Ávila, José Ayala, Carlos López, Juan Carlos Ruiz y Jorge Salinas.

Dedico esta tesis al mejor amigo que una persona pudo tener, un ser lleno de virtudes y que lamentablemente tuvo que partir para no volver, mi amigo Samuel.

Una dedicatoria especial a los padres de Oswaldo Araujo, que nos acogieron en su hogar tantas noches de proyectos, siempre amables y dispuestos a dar un buen consejo.

David Israel Barrigas Jácome

DEDICATORIA

Dedico esta tesis a las personas que siempre estuvieron a mi lado con sus consejos y apoyo, a toda mi familia con su amor, a mi hermana que estuvo dándome ánimos todo el tiempo, a Anita Raza que sin su persistencia y cariño no lo hubiera logrado, a mi gran amigo y hermano David Barrigas que logra posible lo imposible, a mis grandes amigos de la universidad Fernando Morales, Oswaldo Araujo, Juan Carlos Ruiz, Patricio Ávila, Carlos López, Jorge Salinas y José Ayala. A mis amigos de residencia Patricio Salazar y Juan Cristóbal Amores por su positivismo. A Paolo Checa y Daniel Acuña grandes amigos de toda la vida en Ambato.

Pablo Andrés Barriga Torres

AGRADECIMIENTOS

A Dios, por bendecirme con la inteligencia, perseverancia, paciencia y constancia que me ayudaron a terminar la tesis.

A mis padres, por darme la oportunidad y el apoyo, económico y anímico, durante todos mis periodos de aprendizaje y en toda mi vida.

A Elena Garcés, por haber estado siempre pendiente del avance de mi tesis y darme ánimos para seguir.

A Carolina Barriga, siempre dispuesta a darme una sonrisa de ánimos cuando no salía alguna parte del sistema.

A todas las personas que en un momento u otro compartieron su conocimiento y sabiduría.

A la Escuela Politécnica del Ejército, por haberme acogido en sus aulas para formarme académica y profesionalmente.

David Israel Barrigas Jácome

AGRADECIMIENTOS

A mis padres por apoyarme con todo su esfuerzo y afecto, a mis hermanas que siempre me acompañan en el cumplimiento de mis sueños.

A toda la familia de David Barrigas que me acogió en su hogar con su amabilidad y paciencia.

A mis abuelitas que siempre estuvieron a la distancia apoyándome con sus oraciones y bendiciones.

A Anita Raza que me motivó siempre a terminar este proyecto lleno de dificultades y retos.

A la Escuela Politécnica del Ejército que me brindó las herramientas para desarrollarme en mi vida profesional.

Pablo Andrés Barriga Torres

ÍNDICE DE CONTENIDOS

RESUMEN	xiv
CAPÍTULO I	xv
1 INTRODUCCIÓN	xv
1.1 JUSTIFICACIÓN	xv
1.2 OBJETIVOS	xvi
1.3 ALCANCE	xvi
CAPÍTULO II	xviii
2 FUNDAMENTOS TEÓRICOS	xviii
2.1 VENTURE COMO ORGANIZACIÓN	xviii
2.2 VENTURE COMO ERP	xxi
2.2.1 Particularidades del ERP Venture	xxiv
2.2.2 Módulos implementados del ERP Venture	xxvii
2.2.3 Módulo de Puntos de venta	xxviii
2.3 METODOLOGÍA	xxix
2.3.1 Metodología orientada a objetos OMT (Object Modeling Technique) con UML (Unified Modeling Language)	xxix
2.3.2 Metodología de la especificación de requerimientos	lvi
2.3.3 Metodología POS (Point of Sale)	lxii
2.4 HERRAMIENTAS DE ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	lxxxiv
2.4.1 Power Designer 9.5	lxxxiv
2.4.2 Power Builder 8.0	lxxxvii
2.4.3 Sun Application Server 8	lxxxix
2.4.4 NetBeans	xcii
2.5 PARADIGMA DE DESARROLLO	xciii
2.5.1 Paradigma Incremental (Punto de Venta tradicional)	xciii
2.5.2 Paradigma en Espiral (Punto de Venta virtual)	xciv
2.6 DEFINICIÓN DE ESTÁNDARES	xcvi
2.6.1 Estándares de diseño y programación	xcvi
2.6.2 Estándares de codificación	cii
2.6.3 Diseño de interfaces de usuario	cvi
CAPÍTULO III	cix
3 REQUERIMIENTOS	cix
3.1 PROCESOS, PROCEDIMIENTOS Y ACTIVIDADES	cix
3.1.1 Configuración del Módulo	cx
3.1.2 Venta por Intranet e Internet	cxii
3.2 ESPECIFICACIÓN DE REQUERIMIENTOS INFORMÁTICOS	cxii
3.2.1 Introducción	cxiii
3.2.2 Objetivos	cxiii
3.2.3 Alcance	cxiii
3.2.4 Definiciones de acrónimos y abreviaturas	cxiv
3.2.5 Descripción general	cxv

3.2.6	Perspectiva del producto	cxvi
3.2.7	Características del usuario	cxix
3.2.8	Limitaciones generales	cxx
3.2.9	Supuestos y dependencias	cxx
3.2.10	Requisitos funcionales	cxx
3.2.11	Requerimientos de Rendimiento	cxxviii
3.2.12	Restricciones de diseño	cxxviii
CAPÍTULO IV		cxxxí
4	DISEÑO	cxxxí
4.1	MODELO LÓGICO DE DATOS (Tamaño real anexo F)	cxxxí
4.2	DISEÑO DE DIAGRAMAS	cxxxii
4.2.1	Diagrama de procesos del negocio	cxxxii
4.2.2	Diagrama de casos de uso	cxxxiii
4.2.3	Diagrama de clases (intranet)	cxlviii
4.2.4	Diagrama de objetos (visuales, no visuales) para intranet	cxlix
4.2.5	Diagrama de secuencia (intranet)	clii
4.2.6	Diagrama de actividades (intranet)	clxi
4.2.7	Diagrama de casos de uso (Internet)	clxxiii
4.2.8	Diagrama de clases (Internet)	clxxiii
4.2.9	Diagrama de objetos (visuales, no visuales) para Internet	clxxiv
4.2.10	Diagrama de secuencia (Internet)	clxxv
4.2.11	Diagrama de actividades (Internet)	clxxvi
4.3	MODELO FÍSICO DE DATOS (Tamaño real anexo F)	clxxvii
CAPÍTULO V		clxxviii
5	IMPLEMENTACIÓN DEL MÓDULO	clxxviii
5.1	DESARROLLO DEL CÓDIGO	clxxviii
5.1.1	Descripción del código para intranet (Power Builder)	clxxviii
1.1.1	Descripción del código para Internet (Java)	clxxxvi
5.2	INTEGRACIÓN CON LOS DISPOSITIVOS POS	cxcvi
5.2.1	Lector de código de barras	cxcvi
5.2.2	Impresora POS	cxcvi
5.2.3	Pantalla táctil	cxcvi
5.2.4	Lector de tarjetas de crédito	cxcvii
5.2.5	Otros dispositivos	cxcix
5.3	INTEGRACIÓN FUNCIONAL CON EL ERP	cxcix
5.3.1	Integración en la base de datos	cc
5.3.2	Integración funcional	cc
5.3.3	Jerarquía modular	cciv
5.4	PLAN DE PRUEBAS	ccvi
5.4.1	Introducción	ccvi
5.4.2	Definición general de las pruebas	ccvii
5.4.3	Recursos	ccviii
5.4.4	Procedimiento para escenarios de prueba	ccx
5.4.5	Criterios de Aceptación	ccxii
5.4.6	Apéndices	ccxii
CAPÍTULO VI		ccxxxii

6	CONCLUSIONES Y RECOMENDACIONES	ccxxxii
6.1	INTRODUCCIÓN	ccxxxii
6.2	CONCLUSIONES	ccxxxii
6.2.1	Conclusiones generales	ccxxxii
6.2.2	Cumplimiento de objetivos	ccxxxiii
6.3	RECOMENDACIONES	ccxxxiv
	BIBLIOGRAFÍA	ccxxxviii
	WEBBIBLIOGRAFÍA	ccxxxviii

ÍNDICE DE ANEXOS

A. PROCESO DE VENTA PUNTO DE VENTA	ccxli
B. INTEGRACIÓN DEL PUNTO DE VENTA INTRANET FUNCIONALMENTE CON EL ERP EN USO	ccxlv
C. PAYPAL	ccxlix
D. MANUAL DE INSTALACIÓN DEL ERP VENTURE	ccliv
E. MANUAL DE USUARIO DEL PUNTO DE VENTA EN INTRANET	cclviii
F. MANUAL DE USUARIO DEL PUNTO DE VENTA EN INTERNET	cclxvi
G. ESPECIFICACIÓN DE LOS DIAGRAMAS	cclxxiii
H. GLOSARIO DE SIGLAS	cclxxxvi
I. DOCUMENTO DE APROBACIÓN DEL AUSPICIANTE	cclxxxix

ÍNDICE DE TABLAS

Tabla 2.1 Diagramas UML en OMT	lvi
Tabla 2.2 Relación de fases POS con OMT	lxvi
Tabla 2.3 Empresas Beneficiadas con un punto venta móvil	lxix
Tabla 2.4 Software de Punto de Venta existente en el mercado [De Remate]	lxxii
Tabla 2.5 ERP Microsoft GPE	lxxii
Tabla 2.6 ERP MAS 500	lxxiii
Tabla 2.7 ERP mySAP	lxxiii
Tabla 2.8 ERP Epicor	lxxiii
Tabla 3.1 Requerimiento ventana vendedores	cxxi
Tabla 3.2 Requerimiento ventana entidades de pago	cxxi
Tabla 3.3 Requerimiento ventana agencias de distribución	cxxi
Tabla 3.4 Requerimiento ventana jerarquía de clientes	cxxii
Tabla 3.5 Requerimiento ventana adicionales	cxxii
Tabla 3.6 Requerimiento ventana estado de cotización	cxxii
Tabla 3.7 Requerimiento ventana estado de pedido	cxxii
Tabla 3.8 Requerimiento ventana estado de factura	cxxiii
Tabla 3.9 Requerimiento ventana monto de venta	cxxiii
Tabla 3.10 Requerimiento ventana temporada	cxxiii
Tabla 3.11 Requerimiento ventana cajas de venta	cxxiii
Tabla 3.12 Requerimiento ventana descuentos	cxxiv
Tabla 3.13 Requerimiento ventana precios	cxxiv
Tabla 3.14 Requerimiento ventana estado cliente	cxxiv
Tabla 3.15 Requerimiento ventana tipo de cliente	cxxv
Tabla 3.16 Requerimiento ventana clientes	cxxv
Tabla 3.17 Requerimiento ventana grupo de clientes	cxxv
Tabla 3.18 Requerimiento ventana rangos de plazo	cxxvi
Tabla 3.19 Requerimiento ventana formas de pago	cxxvi
Tabla 3.20 Requerimiento ventana servicio para tarjetas	cxxvi
Tabla 3.21 Requerimiento ventana forma de pago tarjetas	cxxvi
Tabla 3.22 Requerimiento Punto de venta	cxxvii
Tabla 3.23 Requerimiento ventana volumen de venta	cxxvii
Tabla 3.24 Requerimiento ventana vencimientos	cxxvii
Tabla 3.25 Requerimiento ventana bodegaje	cxxviii
Tabla 3.26 Requerimiento ventana tipo de venta	cxxviii
Tabla 5.1 Descripción general de la librería punto_ve	clxxix
Tabla 5.2 Dataobject's en la librería punto_ve	clxxxii
Tabla 5.3 Estructuras en la librería punto_ve	clxxxii
Tabla 5.4 Objetos del usuario en la librería punto_ve	clxxxiv
Tabla 5.5 Ventanas en la librería punto_ve	clxxxvi
Tabla 5.6 Descripción general de la página w_qpunto_ve_fs	clxxxvii
Tabla 5.7 Descripción general de la página w_ing_cliente_fs	clxxxviii
Tabla 5.8 Descripción general de las páginas de Paypal	clxxxviii

Tabla 5.9 Descripción funciones clase c_QPunto_ve	ccxiii
Tabla 5.10 Descripción funciones clase c_ing_cliente	cxcv
Tabla 5.11 Descripción variables de sesión del punto de venta Internet	cxcv
Tabla 5.12 Descripción variables de sesión del punto de venta Internet	cxcv
Tabla 5.13 Variables globales utilizadas por el módulo de puntos de venta.	cciii
Tabla 5.14 Tipos de pruebas	ccviii
Tabla 5.15 Personal de pruebas y responsabilidades	ccix
Tabla 5.16 Requerimientos de recursos	ccx
Tabla 5.17 Descripción de la gravedad de los errores	ccxii
Tabla 5.18 Secuencia de ejecución de escenarios	ccxiii
Tabla 5.19 Escenario ingreso de usuarios	ccxiii
Tabla 5.20 Escenario ventanas de mantenimiento	ccxiv
Tabla 5.21 Escenario páginas de mantenimiento	ccxiv
Tabla 5.22 Escenario página y ventana de puntos de venta	ccxv
Tabla 5.23 Escenario ventanas de mantenimiento, página y ventana de puntos de venta	ccxv
Tabla 5.24 Resumen de ejecución de las pruebas	ccxvii

ÍNDICE DE CUADROS

Cuadro 5.1 Resultados del Escenario 1	ccxxiii
Cuadro 5.2 Resultados del Escenario 2	ccxxv
Cuadro 5.3 Resultados del Escenario 3	ccxxvii
Cuadro 5.4 Resultados del Escenario 4	ccxxix
Cuadro 5.5 Resultados del Escenario 5	ccxxxi

ÍNDICE DE ILUSTRACIONES

Fig. 2.1 Oficinas de Venture	xix
Fig. 2.2 Organigrama de Venture	xxi
Fig. 2.3 Interfaz del ERP Venture	xxiv
Fig. 2.4 Ventana de las variables a configurar	xxvi
Fig. 2.5 Ventana de permisos	xxvii
Fig. 2.6 Historia de UML	xxx
Fig. 2.7 Diagrama de casos de uso	xxxv
Fig. 2.8 Diagrama de contexto	xxxv
Fig. 2.9 Diagrama de casos de uso	xxxvi
Fig. 2.10 Diagrama de clases	xxxvii
Fig. 2.11 Diagrama de objetos	xxxvii
Fig. 2.12 Diagrama de componentes	xxxviii
Fig. 2.13 Diagrama de despliegue	xxxix
Fig. 2.14 Diagrama de secuencia	xl
Fig. 2.15 Diagrama de colaboración	xli
Fig. 2.16 Diagrama de actividades	xli
Fig. 2.17 Diagrama de estados	xlvi
Fig. 2.18 OMT modelos Hossian, A., Rossi, B. y García Martínez, R.	xlvi
Fig. 2.19 Especificaciones (Precisión Vs. Entendimiento)	lx
Fig. 2.20 Proceso de validación de requerimientos	lxi
Fig. 2.21 HP POS rp5000	lxxv
Fig. 2.22 Capas OPOS	lxxix
Fig. 2.23 Arquitectura JavaPos	lxxx
Fig. 2.24 Capas UPOS	lxxxii
Fig. 2.25 Pantalla de Power Designer 9.5	lxxxv
Fig. 2.26 Pantalla de Power Builder 8.0	lxxxviii
Fig. 2.27 Pantalla de Sun Application Server 8	lxxxix

Fig. 2.28 Pantalla de NetBeans	xciii
Fig. 2.29 Ciclo de vida paradigma incremental	xciii
Fig. 2.30 Ciclo de vida paradigma espiral	xcv
Fig. 2.31 Diseño del reporte	cvii
Fig. 3.1 Flujo de información	cix
Fig. 3.2 Procesos de negocio (intranet)	cxvi
Fig. 3.3 Procesos de negocio (Internet)	cxvii
Fig. 4.1 Diagrama lógico de datos	cxxxix
Fig. 4.2 Diagrama de procesos del negocio para intranet	cxxxix
Fig. 4.3 Diagrama de mantenimiento de adicionales	cxxxix
Fig. 4.4 Diagrama de mantenimiento de bodegas	cxxxix
Fig. 4.5 Diagrama de mantenimiento de cajas	cxxxix
Fig. 4.6 Diagrama de mantenimiento de clientes	cxxxix
Fig. 4.7 Diagrama de mantenimiento de descuentos	cxxxix
Fig. 4.8 Diagrama de mantenimiento de entidades de pago	cxxxix
Fig. 4.9 Diagrama de mantenimiento de estado de cotización	cxxxix
Fig. 4.10 Diagrama de mantenimiento de estado de factura	cxxxix
Fig. 4.11 Diagrama de mantenimiento de estado de pedido	cxxxix
Fig. 4.12 Diagrama de mantenimiento de estado de cliente	cxxxix
Fig. 4.13 Diagrama de mantenimiento de forma envío	cxxxix
Fig. 4.14 Diagrama de mantenimiento de forma pago tarjeta	cxxxix
Fig. 4.15 Diagrama de mantenimiento de formas pago	cxxxix
Fig. 4.16 Diagrama de mantenimiento de grupo clientes	cxxxix
Fig. 4.17 Diagrama de mantenimiento de monto venta	cxxxix
Fig. 4.18 Diagrama de mantenimiento de periodos de venta	cxxxix
Fig. 4.19 Diagrama de mantenimiento de precios	cxxxix
Fig. 4.20 Diagrama de mantenimiento de rango plazos	cxxxix
Fig. 4.21 Diagrama de mantenimiento de servicio para tarjeta	cxxxix
Fig. 4.22 Diagrama de mantenimiento de temporada	cxxxix
Fig. 4.23 Diagrama de mantenimiento de tipo de cupo	cxl
Fig. 4.24 Diagrama de mantenimiento de tipo de venta	cxl
Fig. 4.25 Diagrama de mantenimiento de tipos de clientes	cxl
Fig. 4.26 Diagrama de mantenimiento de transporte	cxli
Fig. 4.27 Diagrama de mantenimiento de vencimiento de clientes	cxli
Fig. 4.28 Diagrama de mantenimiento de vendedor	cxli
Fig. 4.29 Diagrama de mantenimiento de volumen de venta	cxlii
Fig. 4.30 Diagrama de mantenimiento de agencia de distribución	cxlii
Fig. 4.31 Diagrama de movimiento de cotización	cxlili
Fig. 4.32 Diagrama de movimiento de facturación	cxliv
Fig. 4.33 Diagrama de movimiento de pedidos	cxliv
Fig. 4.34 Diagrama de movimiento de puntos de venta	cxlvi
Fig. 4.35 Diagrama unificado de casos de uso (intranet)	cxlvii
Fig. 4.36 Diagrama de clases	cxlviii
Fig. 4.37 Diagrama de objetos (ventana de clientes)	cxlix
Fig. 4.38 Diagrama de objetos (ventana de descuento)	cxlix
Fig. 4.39 Diagrama de objetos (ventana forma de envío)	cxlix
Fig. 4.40 Diagrama de objetos (ventana periodo de venta)	cl
Fig. 4.41 Diagrama de objetos (ventana de precios)	cl
Fig. 4.42 Diagrama de objetos (ventana de vencimiento del cliente)	cl
Fig. 4.43 Diagrama de objetos (ventana del vendedor)	cl
Fig. 4.44 Diagrama de objetos (ventana padre de ventas)	cli
Fig. 4.45 Diagrama de objetos (ventana de cotizaciones)	cli
Fig. 4.46 Diagrama de objetos (ventana de pedidos)	cli
Fig. 4.47 Diagrama de objetos (ventana de facturas)	clii
Fig. 4.48 Diagrama de objetos (ventana de puntos de venta)	clii
Fig. 4.49 Diagrama de secuencia (mantenimiento genérico)	cliii
Fig. 4.50 Diagrama de secuencia (mantenimiento del cliente)	cliii
Fig. 4.51 Diagrama de secuencia (mantenimiento del descuento)	cliv
Fig. 4.52 Diagrama de secuencia (mantenimiento de forma de envío)	cliv
Fig. 4.53 Diagrama de secuencia (mantenimiento de periodo de venta)	clv

Fig. 4.54 Diagrama de secuencia (mantenimiento de precio)	clv
Fig. 4.55 Diagrama de secuencia (mantenimiento de vencimiento de clientes)	clvi
Fig. 4.56 Diagrama de secuencia (mantenimiento del vendedor)	clvi
Fig. 4.57 Diagrama de secuencia (movimiento de cotizaciones)	clvii
Fig. 4.58 Diagrama de secuencia (movimiento de pedidos)	clviii
Fig. 4.59 Diagrama de secuencia (movimiento de facturas)	clix
Fig. 4.60 Diagrama de secuencia (movimiento de puntos de venta)	clx
Fig. 4.61 Diagrama de actividades (mantenimiento general)	clxi
Fig. 4.62 Diagrama de actividades (mantenimiento de cliente)	clxii
Fig. 4.63 Diagrama de actividades (mantenimiento de descuento)	clxiii
Fig. 4.64 Diagrama de actividades (mantenimiento de forma de envío)	clxiv
Fig. 4.65 Diagrama de actividades (mantenimiento de periodo de venta)	clxv
Fig. 4.66 Diagrama de actividades (mantenimiento del precio)	clxvi
Fig. 4.67 Diagrama de actividades (mantenimiento del vencimiento de clientes)	clxvii
Fig. 4.68 Diagrama de actividades (mantenimiento del vendedor)	clxviii
Fig. 4.69 Diagrama de actividades (transacción de cotizaciones)	clxix
Fig. 4.70 Diagrama de actividades (transacción de pedidos)	clxx
Fig. 4.71 Diagrama de actividades (transacción de facturas)	clxxi
Fig. 4.72 Diagrama de actividades (transacción de puntos de venta)	clxxii
Fig. 4.73 Diagrama de clases (Punto de Venta Internet)	clxxiii
Fig. 4.74 Diagrama de objetos (página w_punto_ve)	clxxiv
Fig. 4.75 Diagrama de objetos (página DirectPaymentReceipt)	clxxiv
Fig. 4.76 Diagrama de objetos (página mantenimiento)	clxxv
Fig. 4.77 Diagrama de secuencia (página de compras)	clxxv
Fig. 4.78 Diagrama de actividades (transacción de compras)	clxxvi
Fig. 5.1 Módulos del sistema Venture (Internet)	clxxxvi
Fig. 5.2 Ingreso de un nuevo cajero (intranet)	ccxvii
Fig. 5.3 Ingreso al sistema (intranet)	ccxviii
Fig. 5.4 Ingreso al sistema (intranet)	ccxviii
Fig. 5.5 Ventana de mantenimiento genérico Adicionales	ccxviii
Fig. 5.6 Ventana de transacciones Cotización	ccxix
Fig. 5.7 Ventana de mantenimiento de Adicionales	ccxix
Fig. 5.8 (VMWare) ingresando al módulo de puntos de venta en Internet	ccxx
Fig. 5.9 (VMWare) ingresando al módulo de puntos de venta en intranet	ccxx
Fig. 5.10 Ventana Punto de venta (intranet)	ccxxi
Fig. 5.11 Ventana Punto de venta (Internet)	ccxxi
Fig. 5.12 Ventana Punto de venta (Vista completa Internet)	ccxxii

RESUMEN

El presente proyecto trata sobre el desarrollo del módulo de puntos de venta integrado al ERP Venture, esta aplicación se desenvuelve tanto en ambiente Intranet con en Internet.

El proyecto inicia con el reconocimiento de Venture a nivel de empresa y del ERP que desarrolla. Después de tener una visión de lo que se desea hacer se procede a investigar la metodología OMT, que es la que se aplica en el desarrollo del proyecto.

La Metodología OMT trabaja con los diagramas UML, y se relaciona con la metodología POS, la cual establece un marco de trabajo amigable para el usuario y la utilización de dispositivos POS como el lector de código de barra, la impresora de notas de venta, lector de tarjetas de crédito y la pantalla de táctil.

Posteriormente, utilizando la norma IEEE 830, se realizó la documentación para la recolección de requerimientos.

En base a los requerimientos obtenidos, y siguiendo la metodología OMT, se procedió al desarrollo del módulo de puntos de venta, en esta parte se dividió el proyecto en dos módulos, uno que trabajaba dentro del ERP para Intranet, desarrollado en Power Builder 8.03 y otro que trabajaba dentro del ERP para Internet desarrollado en Java.

Finalmente, después de haber realizado pruebas de caja negra y caja blanca y habiendo superado los errores resultantes, los módulos desarrollados se implantaron al sistema en producción de Venture. Se capacitó a los empleados en la programación u utilización de los módulos.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 JUSTIFICACIÓN

Venture Venti es una empresa de desarrollo de software, fundada en el año 2000 y desde sus inicios se ha orientado a la elaboración de un ERP (Enterprise Resources Planning o Planificador de Recursos de la Empresa).

En la actualidad el ERP Venture trabaja en dos capas, cliente servidor; su servicio se desarrolla solo en intranet aunque también tiene algunas aplicaciones para Palm. Existen módulos como: inventario y ventas, donde parte de su trabajo es la simulación de una venta, estos módulos no cubren ni un veinte por ciento de lo que es un punto de ventas, ya que su función es llevar un registro de los artículos que han salido de inventario y un registro de los artículos vendidos con la característica del precio con que se vendieron.

Los clientes de Venture Venti trabajan a nivel nacional e internacional con varias sucursales en el área de producción y también en ventas, por lo que el ERP Venture es multi empresa, multi sucursal y dependiendo de cada empresa se especializa en las áreas necesarias.

El ERP Venture carece de un área especializada en puntos de venta, tema que es fundamental para varias empresas que realizan una venta directa. Esta insuficiencia le quita un gran espacio de mercado además de no cerrar completamente el círculo formado por los procesos que manejan el paso de los artículos, es decir desde su adquisición hasta la contabilidad de la empresa.

El módulo de puntos de venta a realizarse tiene que cubrir varios requerimientos, entre los cuales están el uso de políticas de precios, descuentos y créditos; el manejo de periféricos especializados como el lector de código de barras, monitores táctiles y lectores de tarjetas de crédito. Por último, y como

parte fundamental de este módulo es el acoplamiento, asociación y funcionamiento con el ERP Venture.

1.2 OBJETIVOS

General

Desarrollar una aplicación de punto de venta virtual y tradicional que se integre con el ERP Venture.

Específicos

Realizar un análisis de la funcionalidad del ERP Venture y como puede funcionar dentro de este un módulo de puntos de venta.

Implementar al módulo de puntos de venta los dispositivos tales como la pantalla táctil, el lector de código de barras y el lector de tarjetas de crédito.

Desarrollar dos aplicaciones n capas capaces de adaptarse a distintas reglas de negocio.

1.3 ALCANCE

La aplicación de puntos de venta debe abarcar distintos tipos de negocio por lo que tendrá una orientación genérica. La aplicación será capaz de adaptarse a distintas reglas de negocio.

Se entregará la aplicación completa y funcionando tanto en el ambiente de intranet como en el de Internet; Venture será el encargado de su comercialización.

La aplicación deberá ser compatible con las bases de datos usadas por Venture, estas son: Anywhere, SQL Server, Oracle e Informix.

Para el desarrollo del punto de venta virtual se utilizarán tecnologías como JSP, EJB's, HTML y XML.

La tesis culmina con la entrega de los módulos de punto de venta tradicional y virtual, integrados y funcionando con el ERP Venture, sabiendo que estos módulos quedarán en el paso previo a la implantación y comercialización de los mismos.

CAPÍTULO II

2 FUNDAMENTOS TEÓRICOS

2.1 VENTURE COMO ORGANIZACIÓN

Venture es una empresa de desarrollo de software establecida a inicios del año 2000 como un negocio familiar; se desarrolla en una oficina del edificio Presidente y su alcance es a nivel nacional.

La empresa nace como una necesidad de presentar al país un ERP hecho en Ecuador, que preste servicios profesionales a varias empresas nacionales e internacionales que funcionen en el país.

Se puede ver que existen varias empresas nacionales de software pero ninguna llega a sistematizar todos los procesos como lo hace Venture, como ERP, ni llegan a cubrir todas las necesidades del negocio como empresa.

Venture presenta servicios anexos al ERP, así como son: capacitación, análisis de mejoras en las empresas clientes, mantenimiento diario, actualizaciones del sistema, mejoras y correcciones inmediatas dependiendo de la necesidad del cliente, asesoramiento tecnológico en hardware, software y redes además ofrece computadoras (servidores) y accesorios a precios de distribuidor.

La selección del personal se hace en la propia universidad, ya que el gerente técnico es profesor de los niveles superiores de la ESPE, lo que le permite hacer una clasificación minuciosa de los mejores estudiantes; esto garantiza en cierta parte el funcionamiento de Venture como empresa y como ERP.

Actualmente ofrecen los siguientes servicios:

- Desarrollo de sistemas a medida con el menor tiempo de implementación en el mercado.

- Tecnología de punta.
- Selección de la tecnología por parte del cliente de acuerdo a sus recursos.
- Solución integral de requerimientos partiendo desde el análisis de procesos.
- Disponibilidad de código fuente.
- Disponibilidad de paquetes informáticos específicos para sus clientes (empresas de servicios, comerciales, industriales, floricultores entre otras) en dos versiones:
 - Venture.- Para empresas pequeñas.
 - Venture Plus.- Para empresas grandes.

Tecnología Utilizada

- Modelo de 3 capas distribuido o cliente servidor 2 capas.
- Front end: Power builder, HTML, Java.
- Servidor de componentes: MTS (Microsoft Transaction Sever), Jaguar CTS (Component Transaction Server).
- Bases de datos: Oracle, SQL Server, Anywhere e Informix.
- Programación totalmente orientada a objetos.
- Programación genérica.
- Desarrollo de aplicaciones por módulos.

Espacio físico en que se desenvuelve la empresa

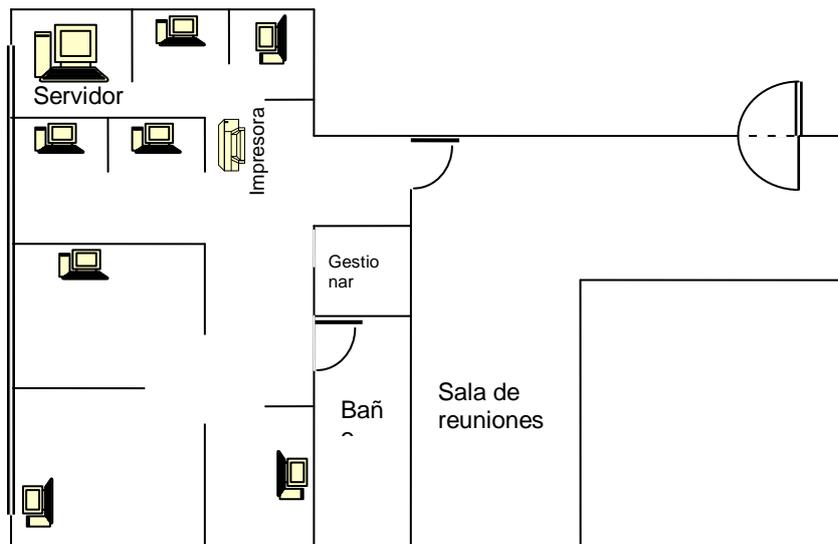


Fig. 2.1 Oficinas de Venture

Misión

“Ofrecer servicio a nuestros clientes manteniendo un balance entre: calidad de las soluciones, bienestar del recurso humano y utilidad. Colaborando de esta manera con el adelanto socio económico de quienes formamos Venture Venti, nuestros clientes y del País”.

Visión

“Somos una compañía de reconocida trayectoria a nivel nacional y con varios servicios ofrecidos fuera del país. Las empresas que se han beneficiado de nuestras soluciones poseen una importante ventaja competitiva ante sus similares, gracias a la eficiencia y eficacia en sus procesos, a la disponibilidad oportuna de información para la toma de decisiones y negociación en Internet”.

Valores

- La práctica de la ética profesional ha facilitado la mejor relación con clientes, recurso humano, proveedores y el estado
- El impartir el término honradez en nuestros colaboradores nos ha permitido ofrecer un servicio de calidad a un precio justo
- Nuestra responsabilidad establece la diferencia en el ámbito del servicio como la compañía que posee la más alta credibilidad
- El conocimiento nos permite ofrecer las mejores alternativas de solución para nuestros clientes

Organigrama

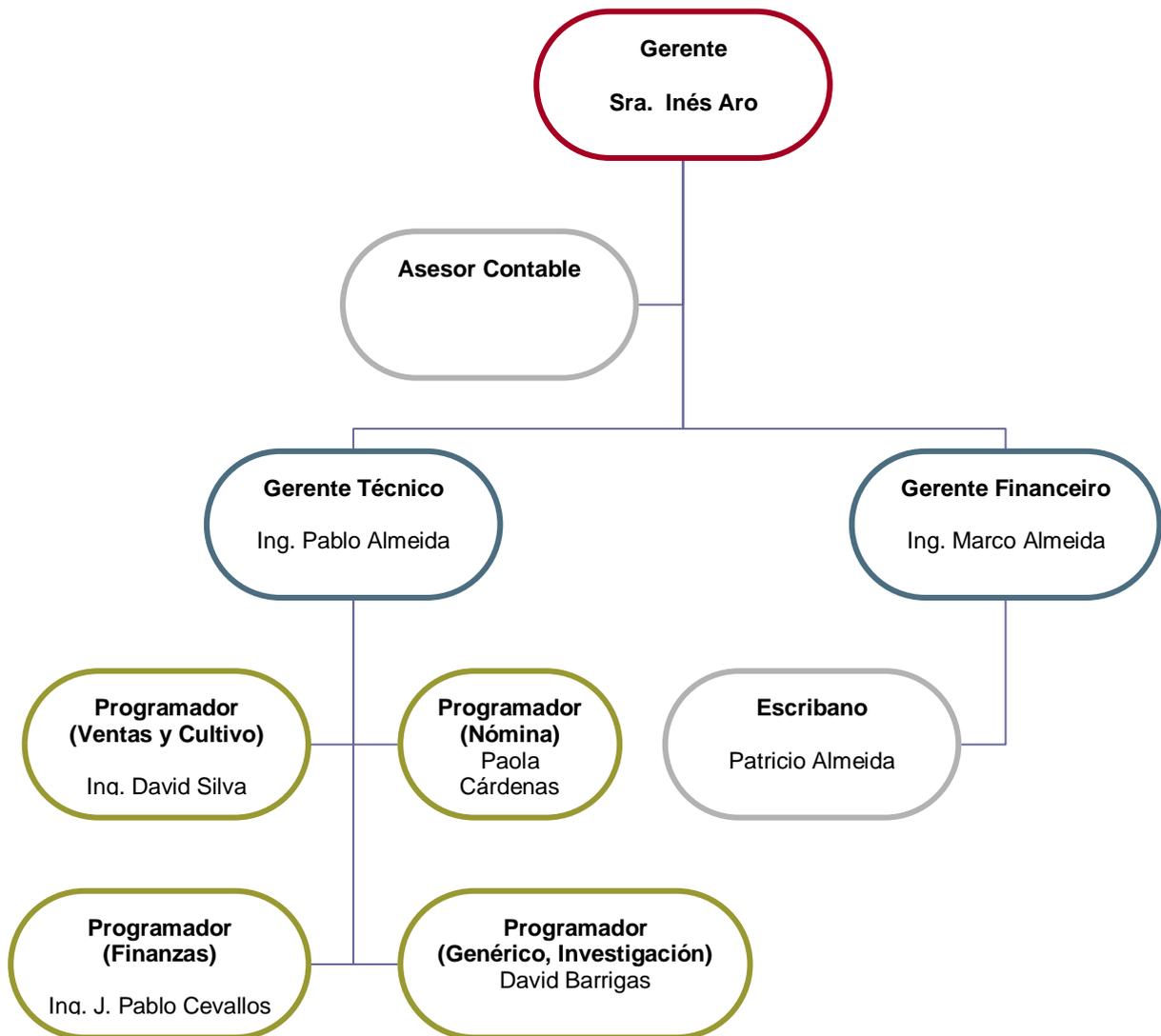


Fig. 2.2 Organigrama de Venture

2.2 VENTURE COMO ERP

Enterprise Resources Planning (ERP) o Planificador de Recursos de la Empresa, consiste en una aplicación informática que gestiona todos los aspectos de una empresa, desde la gestión de los empleados hasta las actividades de fidelización de clientes, pasando por el control de producción o por la gestión de los servicios que la compañía preste. Existen tantos ERP como empresas, ya que cada empresa dispone de sus propios departamentos y procesos de gestión. La principal característica de un sistema ERP es que integra la información de todos los aspectos de la compañía en una sola aplicación, desde la cual podemos

obtener informes para conocer en cualquier momento la situación real de la empresa y tomar así las mejores decisiones que traerán mayor rentabilidad. Ventas, existencias de almacén, vacaciones de los empleados, material de producción, transporte y logística de la empresa, es decir todas las formas necesarias para presentar la información y todo en un mismo sistema.

Cada situación del sector del software necesita obtener certificados de calidad de forma diferente según el tipo de empresa u organización en la que se esté. De esta forma las grandes empresas al disponer de departamentos de TI importantes, muchas veces optan por modificar y mejorar sus aplicaciones para soportar estas nuevas necesidades. En otros casos prefieren comprar un software ERP como solución rápida y radical.

Existen varios parámetros importantes que califican un software ERP y son los siguientes:

1. Un ERP puede estar compuesto por un determinado número de **módulos funcionales**. Estos se pueden catalogar principalmente en tres tipos:
 - **Módulos básicos:** suelen ser módulos obligatorios a adquirir, como el de contabilidad, alrededor de los cuales se añaden otros módulos.
 - **Módulos opcionales:** son los módulos no obligatorios y de extensión a añadir a los módulos básicos. Incorporan nuevas funcionalidades al paquete ERP.
 - **Módulos verticales:** se trata de módulos opcionales diseñados específicamente para resolver las funcionalidades de un sector específico como la administración pública, los hospitales y la banca.
2. La mayoría del software ERP existente en el mercado, se trata de aplicaciones para ser utilizadas en entornos cliente/servidor. El empleo de la tecnología cliente/servidor hace que sea escalable y ofrece configuraciones óptimas en hardware, gestión de base de datos y sistemas abiertos. Este sistema de arquitectura abierta facilita a las compañías el elegir los equipos informáticos y los sistemas operativos de tal manera que se pueda aprovechar al máximo los avances en la tecnología.

3. Ligado con el punto anterior, todas las áreas de la empresa comparten una **única base de datos**. Entre las bases de datos más utilizadas se encuentran Oracle, Anywhere, Informix, Microsoft SQL Server y DB2. Cabe destacar, que hay algunos fabricantes de software ERP que ya incorporan su propio Gestor de Base de Datos. Es recomendable utilizar una Base de Datos abierta, pues esta ofrece la oportunidad de ser también utilizada por otras aplicaciones.
4. Todas las aplicaciones ofrecen una **misma interfaz** de usuario.
5. Ofrecer **alta adaptabilidad** a la situación particular de cada empresa. Algunos ERP's ofrecen disponer al usuario de su código fuente, permitiendo de esta manera poder realizar un alto nivel de modificaciones.
6. Alta capacidad de facilidad de comunicación inter departamental.

Después de esta pequeña introducción se puede entender de mejor manera lo que viene a continuación, es decir Venture como ERP:

El ERP Venture es un sistema hecho en Power Builder 8 por su funcionalidad basada en objetos y todos sus principios así como son la herencia, la sobrecarga de herramientas y en especial la posibilidad de realizar ventanas genéricas. Otra de las ventajas que brinda Power Builder 8 es que el trabajo se lo puede hacer por módulo, ya que al momento de trabajar con este, se comportan totalmente independientes de los demás módulos.

Una ventaja al trabajar con Power Builder es el uso del datawindow, que es la parte fundamental del programa, porque esta herramienta es la que interactúa con la base de datos y sabiendo que un datawindow es totalmente genérico y dinámico, aún en el momento de ejecución, podemos decir que es la base del sistema. Venture no solo que utiliza el datawindow, sino que lo ha sobrecargado para tener una herramienta aún más potente.

El sistema está formado en su parte central por un módulo, que es el que administra a todo el sistema, además de ser el gestor del resto de módulos. La pregunta es cómo un módulo del sistema puede ser gestor de otros módulos; la

respuesta es que en este módulo se encuentran todas las variables generales, funciones, herramientas sobrecargadas, ventanas y políticas del sistema.

El ERP Venture es escalable y portable, lo de escalable es por su propia arquitectura, la cual está diseñada hasta para trabajar con distintas empresas en el mismo sistema; y lo de portable es por funcionar sin hacer cambios en el código con las bases de datos Anywhere, SQL Server, Informix y Oracle; pero no solo con las bases de datos, ya que Sybase, empresa creadora de Power Builder, está sacando nuevas versiones que permiten la migración automática a java dando la posibilidad de funcionar en cualquier plataforma.

Las ventanas genéricas, localizadas en el módulo central son utilizadas en todo el sistema o son heredadas hacia otros módulos para ponerles funcionalidades especiales dependiendo de las necesidades. La interfaz de usuario es amigable como se observa en la Fig. 2.3, permite una rápida manipulación de los distintos módulos del sistema.

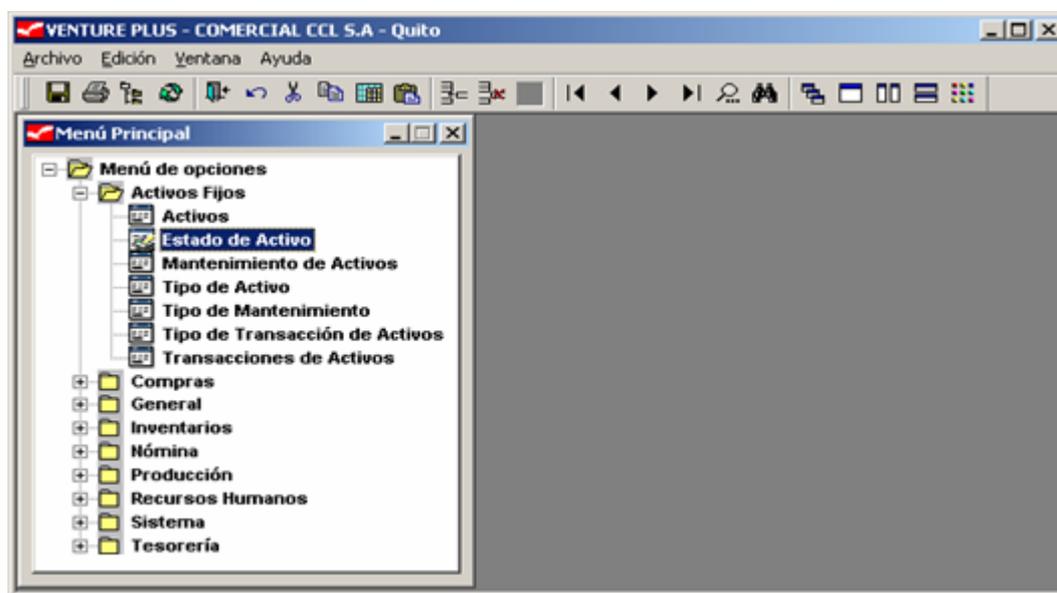


Fig. 2.3 Interfaz del ERP Venture

2.2.1 Particularidades del ERP Venture

Este sistema se distingue de los demás ERP's por dos factores:

- Políticas de programación
- Políticas de clientes

Aunque la documentación de estas dos políticas no se especifica a detalle, siendo muchas de estas reglamentaciones solo habladas y solo se tengan unos documentos con pocas especificaciones, la empresa se maneja en su totalidad bajo estas políticas y se las hace cumplir.

a) Políticas de programación

En general las políticas de programación son una mejora de las reglas que aconsejan seguir los manuales de Power Builder 8 de Sybase; la política también presenta formatos, en especial de reportes y ventanas, pero lo que le hace distinto de los demás ERP's es que todo lo que se programe debe ser genérico, además que debe ser probada la misma función, ventana o datawindow en todas las bases de datos que maneja Venture.

Una parte fundamental de las políticas de programación de Venture consiste en la protección del módulo padre del ERP, esto se lo hace mediante el encargo del mismo a un solo programador, el cual debe ser experimentado y trabajar en pro de todos sus compañeros programadores.

Otra de las políticas fundamentales del sistema es que nada en lo posible debe ser quemado en el código, esto quiere decir que para realizar cambios de estándares, datos, fechas, formatos, etc. No debe ser necesario entrar al código si no que en la ventana de configuración de variables (Fig. 2.4) se podrán hacerse todos estos cambios.

Código	Módulo	Nombre	Descripción	Valor
2060	Activos	is_Act_usa_costos	Indica si usa o no costos al depreciar, 0=	1
0055	Ventas Flores	is_activa_stock_cajas	Activa 1, Desactiva 0: el uso de stock caja	0
0170	Activos	is_activo_enlinea	Bandera que indica si hace Asientos por	1
1879	Postcosecha	is_ajuste_stock_mas	Código del motivo de transacción ingreso	001012
1880	Postcosecha	is_ajuste_stock_meno	Código del motivo de transacción egreso	001013
0067	Users	is_alstro	Código que identifica al tipo de variedad	:001002
0154	Users	is_alstro_extra_select	Código que identifica al tipo de variedad	:0010
0156	Users	is_alstro_fancy	Código que identifica al tipo de variedad	:0010
0151	Users	is_alstro_mix	Código que identifica al tipo de variedad	:0010
0152	Users	is_alstro_select	Código que identifica al tipo de variedad	:0010
0153	Users	is_alstro_sselect	Código que identifica al tipo de variedad	:0010
0268	Inventarios	is_asiento_enlinea_inv	1=realiza asiento, 0=no asiento ni registr	1

Fig. 2.4 Ventana de las variables a configurar

b) Políticas de clientes

Como cualquier empresa que ofrece un sistema posee políticas hacia el cliente y aunque la mayoría de las políticas de Venture sean igual a las de las otras empresas, existen artículos que difieren, así como:

- Dar en forma gratuita y en cualquier parte del Ecuador una presentación del sistema.
- Tomar los requerimientos adicionales del cliente y en poco tiempo, aproximadamente dos semanas, tener listo un prototipo de la base de datos y ventanas requeridas por el cliente potencial.
- En la fase de implementación, más o menos unos ocho meses, se recogen todos los requerimientos de la empresa cliente, se capacitará al personal y se les presta ayuda por teléfono o en forma presencial para el manejo del sistema, seis días a la semana y ocho horas en el día.
- Si el ERP presenta algún problema se presta ayuda gratuita e inmediata al cliente.
- La base para trabajar con varias empresas, utilizando en mismo sistema es la parte de los permisos como se muestra en la figura 2.5,

que dependiendo de cada cliente y de los módulos a su disposición, da o quita permisos a módulos completos y hasta botones específicos.

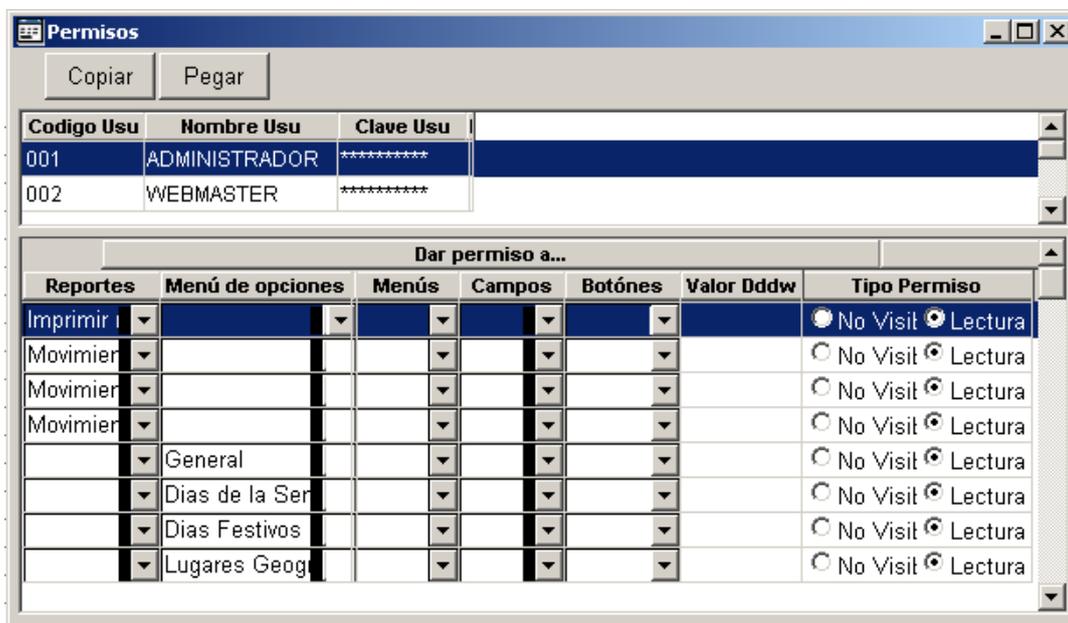


Fig. 2.5 Ventana de permisos

2.2.2 Módulos implementados del ERP Venture

El ERP Venture presenta una variedad de módulos, algunos generales y otros especializados. A continuación se presentan los módulos de mayor importancia con una pequeña síntesis de su funcionalidad:

- a. Requisiciones.- Permite realizar pedidos personales por parte de los empleados, estos pedidos va a Ordenes de compra o directamente a Inventarios.
- b. Compras.- Maneja todo lo que es proveedores, artículos, cotizaciones y órdenes de compra.
- c. Inventario.- Maneja todos los movimientos de la bodega, además de ser la entrada de la información de formas de pago e impuestos.
- d. Cuentas por pagar.- Este módulo se encarga de almacenar y elaborar los documentos para pagar.
- e. Activos fijos.- Como su nombre lo indica maneja la parte de activos, desde su adquisición hasta su depreciación.

- f. Contabilidad.- Este es el módulo central del sistema, y es porque interactúa con casi todos los demás módulos.
- g. Tesorería y Bancos.- Maneja los movimientos de las cuentas bancarias que posea la empresa, desde aquí también se realizan los pagos.
- h. Recursos humanos.- Módulo fundamental para el manejo del personal.
- i. Cuentas por cobrar.- Este módulo se encarga de almacenar y elaborar los documentos para cobrar.
- j. Ventas.- Registra todas las ventas que se realizan, en el caso de venture se especializa en ventas de flores.
- k. Nómina.- Maneja salarios de acuerdo a la ley ecuatoriana.

2.2.3 Módulo de Puntos de venta

El módulo de puntos de venta nace como una necesidad de las empresas clientes de Venture, ya que el sistema en si solo registra las ventas pero no interviene en la transacción misma y como algunas empresas realizan todo el ciclo del producto, es decir, desde su producción hasta la venta al consumidor final, se decidió crear un módulo especializado en puntos de venta.

2.2.3.1 Perspectivas del punto de venta

Este nuevo módulo a crearse, se divide en dos partes que innovarán al ERP Venture, en primer lugar por el modo de diseño gráfico-funcional, ya que tanto la parte para intranet como la parte para Internet tienen que ser totalmente agradables al usuario además de tener un funcionamiento rápido y seguro en el momento de la transacción. Por otra parte este nuevo módulo se integrará con varios dispositivos que agilitan la venta, así como el lector de código de barras, lector de tarjetas de crédito, impresora de notas de venta, monitor con pantalla táctil entre otros. Pero lo fundamental es la salida hacia la Web por medio de páginas JSP que conectadas al servidor de la base de datos de Venture pueden ser utilizadas en cualquier parte del mundo, rompiendo la barrera de la distancia en el momento de realizar una venta.

Una de las bases para la programación es mantener un carácter genérico y que con solo configurar variables creadas en el sistema se puedan cambiar las reglas del negocio para que no se especialice solo en una empresa.

2.2.3.2 Integración del Punto de Venta con el ERP Venture

Para la realización del este módulo se crearán nuevas tablas para el funcionamiento en la Intranet y en el Internet, estas tablas tienen que relacionarse con las demás tablas ya existentes en el ERP, además que se utilizará el módulo genérico hecho en Power Builder para trabajar en la programación para Intranet y el genérico hecho en Java para trabajar en la programación para Internet.

2.3 METODOLOGÍA

2.3.1 Metodología orientada a objetos OMT (Object Modeling Technique) con UML (Unified Modeling Language)

UML (lenguaje de modelado unificado) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema orientado a objetos.

Esta notación ha sido ampliamente aceptada debido a la facilidad de su uso y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y Object Oriented Software Engineering). UML ha puesto fin a las llamadas “guerras de modelado” que se han mantenido a lo largo de los 90, en las que las principales metodologías sacaban nuevas versiones que incorporaban las técnicas de los demás.

Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros de software que trabajan en el desarrollo orientado a objetos. La versión más actual es UML 2.0 que salió en el año 2004.

En la fig. 2.6 se puede apreciar la evolución de UML.

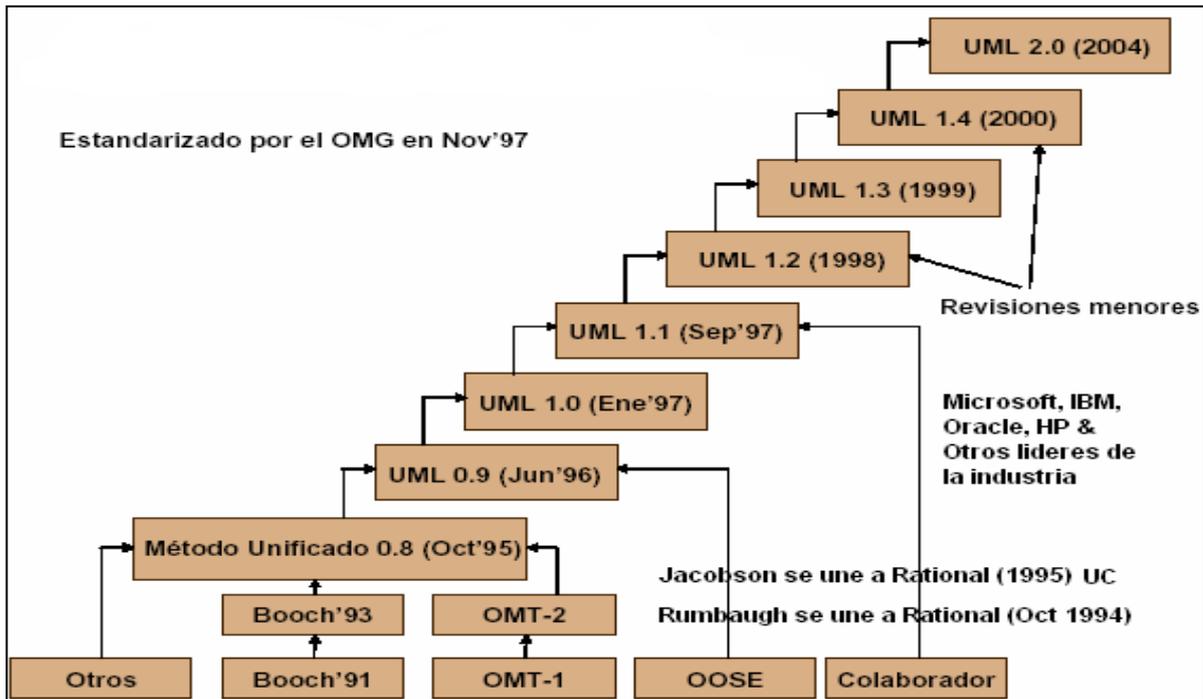


Fig. 2.6 Historia de UML

Los objetivos de la unificación fueron: el mantenerlo simple, quitar elementos de los lenguajes de Booch, OMT y OOSE que no funcionan en la práctica, añadir elementos de otros métodos que fueran más efectivos y el inventar nuevas construcciones solamente cuando la solución existente no estuviera disponible.

Hay que tener en cuenta que el estándar UML no es una metodología para el desarrollo de sistemas, tan solo se trata de una notación del proceso de desarrollo. En esta Tesis se sigue la notación UML que nos permite reducir la complejidad al momento de modelar y se la puede utilizar con cualquier metodología de análisis y diseño orientado a objetos. UML no está atado a ningún ciclo de vida de desarrollo concreto.

2.3.1.1 Diagramación en UML

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece varias vistas y diagramas en los cuales modelar sistemas.

Entre las vistas están las siguientes:

- Vista casos de uso: Se forma con los diagramas de casos de uso, colaboración, estados y actividades.

- Vista de diseño: Se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- Vista de procesos: Se forma con los diagramas de la vista de diseño. Recalcando las clases y objetos referentes a procesos.
- Vista de implementación: Se forma con los diagramas de componentes, colaboración, estados y actividades.
- Vista de despliegue: Se forma con los diagramas de despliegue, interacción, estados y actividades.

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

Los diagramas estáticos son:

- **Diagrama de clases:** muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- **Diagrama de objetos:** Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- **Diagrama de componentes:** Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- **Diagrama de despliegue:** Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.
- **Diagrama de casos de uso:** Muestran los casos de uso, actores y sus relaciones. Muestra quien puede hacer que y relaciones existen entre acciones (casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Lo diagramas dinámicos son:

- **Diagramas de interacción (secuencia y colaboración):** Muestran a los diferentes objetos, las relaciones que pueden tener y los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de

uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema.

- **Diagrama de estados:** muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.
- **Diagrama de actividades:** Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control.

2.3.1.1.1 Utilización de los diagramas UML

Los diagramas a representar dependerán del sistema a desarrollar, para ello se efectúan las siguientes recomendaciones dependiendo del sistema. Estas recomendaciones se deberán adaptar a las características de cada desarrollo, y seguramente será la práctica la que nos diga las cosas que echamos en falta o los diagramas que parecen ser menos necesarios.

- Para aplicación monopuesto:
 - Diagrama de casos de uso.
 - Diagrama de clases.
 - Diagrama de interacción.
 - Diagrama de secuencia
 - Diagrama de colaboración
- Para aplicación monopuesto, con entrada de eventos:
 - Diagrama de casos de uso.
 - Diagrama de clases.
 - Diagrama de interacción.
 - Diagrama de secuencia
 - Diagrama de colaboración
 - Diagrama de estados.
- Para aplicación cliente servidor:
 - Diagrama de casos de uso.
 - Diagrama de clases.

- Diagrama de interacción.
 - Diagrama de secuencia
 - Diagrama de colaboración
 - Diagrama de estados.
 - Diagrama de implementación
 - Diagrama de componentes, dependiendo de la complejidad.
 - Diagrama de despliegue
- Aplicación compleja distribuida
 - Diagrama de casos de uso.
 - Diagrama de clases.
 - Diagrama de objetos
 - Diagrama de actividades
 - Diagrama de interacción.
 - Diagrama de secuencia
 - Diagrama de colaboración
 - Diagrama de estados.
 - Diagrama de implementación
 - Diagrama de componentes, dependiendo de la complejidad.
 - Diagrama de despliegue

Así tenemos que para una aplicación sencilla debemos realizar entre tres y seis tipos de diagramas, y para una aplicación compleja hasta nueve. El tiempo dedicado a la realización de los diagramas es proporcional al tamaño del producto a realizar. Para la mayoría de los sistemas es suficiente con tres o cuatro diagramas.

2.3.1.1.2 Descripción breve de los diagramas

En la descripción de los diagramas se toma en cuenta los principales y son los siguientes:

- Diagrama de casos de uso.

Se emplean para visualizar el comportamiento del sistema, una parte de él o de una sola clase. De forma que se pueda conocer como responde esa parte del sistema. El diagrama también puede ser utilizado para que los expertos de dominio se comuniquen con los informáticos sin llegar a niveles de complejidad. Un caso de uso especifica un requerimiento funcional, es decir indica esta parte debe hacer esto cuando pase esto.

En el diagrama nos encontramos con diferentes figuras que pueden mantener diversas relaciones entre ellas:

Simbología

Include: Representado por una flecha, en el diagrama de ejemplo podemos ver como un caso de uso, el de totalizar el coste incluye a dos casos de uso.

Extends: Una relación de un caso de uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A.

Generalización: Es la típica relación de herencia.

- Actores: se representan por un muñeco y sus relaciones son:
 - Comunica (communicates): Comunica a un actor con un caso de uso, o con otro actor.
 - Usa (uses): Es la relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. Frecuentemente no hay actor asociado con el caso de uso común.
 - Extiende (extends): Relación entre dos casos de uso, se denota cuando un caso de uso es una especialización de otro. Se usa cuando se describe una variación sobre el normal comportamiento.

Ejemplo gráfico

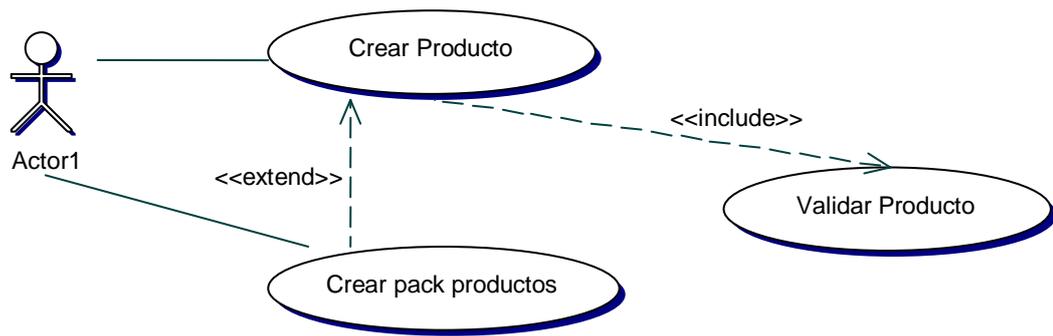


Fig. 2.7 Diagrama de casos de uso

Modelado

- Modelado del Contexto

Se debe modelar la relación del sistema con los elementos externos, ya que son estos elementos los que forman el contexto del sistema.

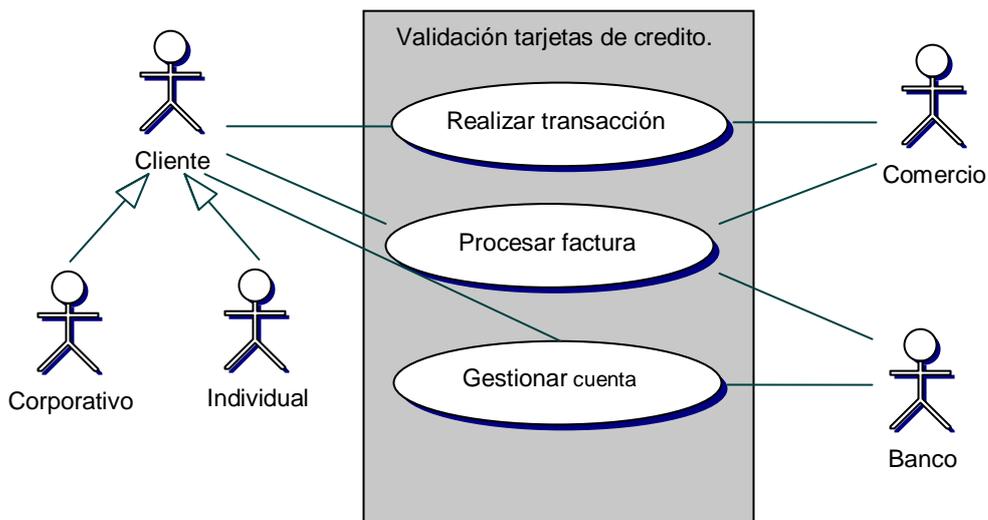


Fig. 2.8 Diagrama de contexto

- Modelado de Requisitos

La función principal, o la más conocida del diagrama de casos de uso es documentar los requisitos del sistema, o de una parte de el.

Ejemplo gráfico

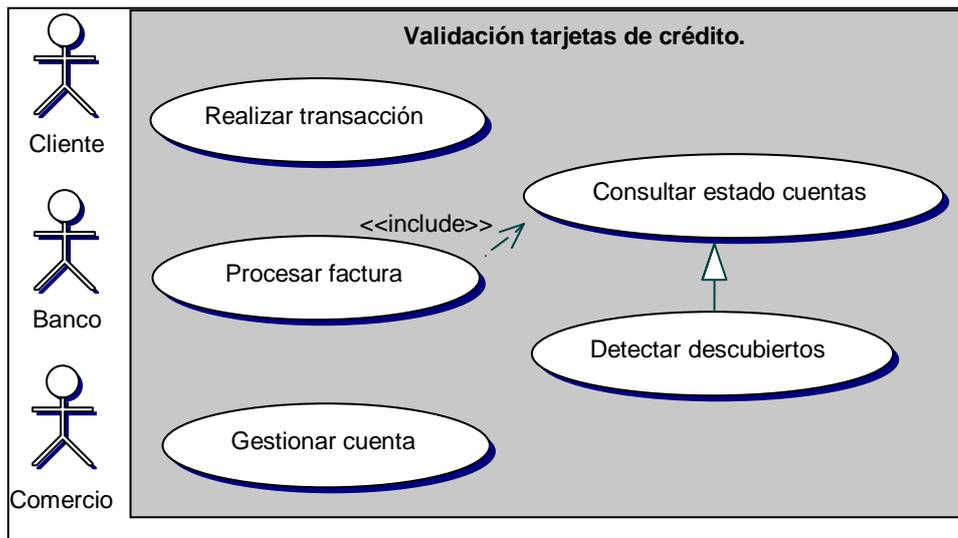


Fig. 2.9 Diagrama de casos de uso

- Diagrama de clases

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

Simbología

Clase: Una clase esta representada por un rectángulo que dispone de tres apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos.

Relaciones entre clases: Existen tres relaciones diferentes entre clases, dependencias, generalización y asociación: En las relaciones se habla de una clase destino y de una clase origen. El origen es desde la que se realiza la acción de relacionar. Es decir desde la que parte la flecha, el destino es la que recibe la flecha. Las relaciones se pueden modificar con estereotipos o con restricciones.

- Dependencias: Es una relación de uso, es decir una clase usa a otra, que la necesita para su cometido. Se representa con una flecha discontinua va desde la clase utilizadora a la clase utilizada.
- Generalización: Es la típica relación de herencia.
- Asociación: Especifica que los objetos de una clase están relacionados con los elementos de otra clase.

Ejemplo gráfico

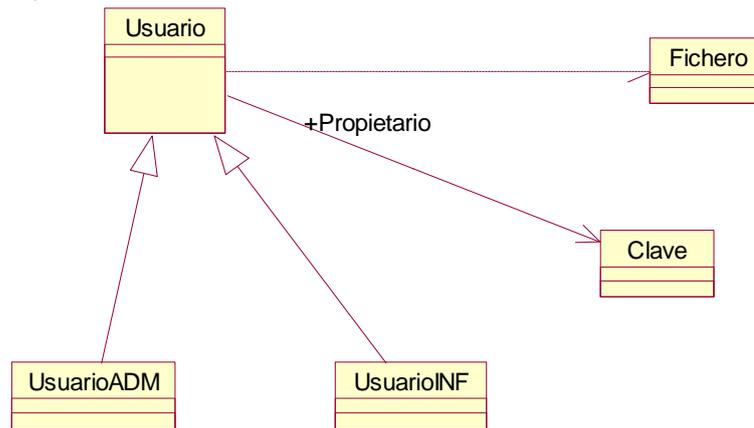


Fig. 2.10 Diagrama de clases

- Diagrama de objetos

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases. Muestra a los objetos y sus relaciones, pero en un momento concreto del sistema.

Ejemplo gráfico

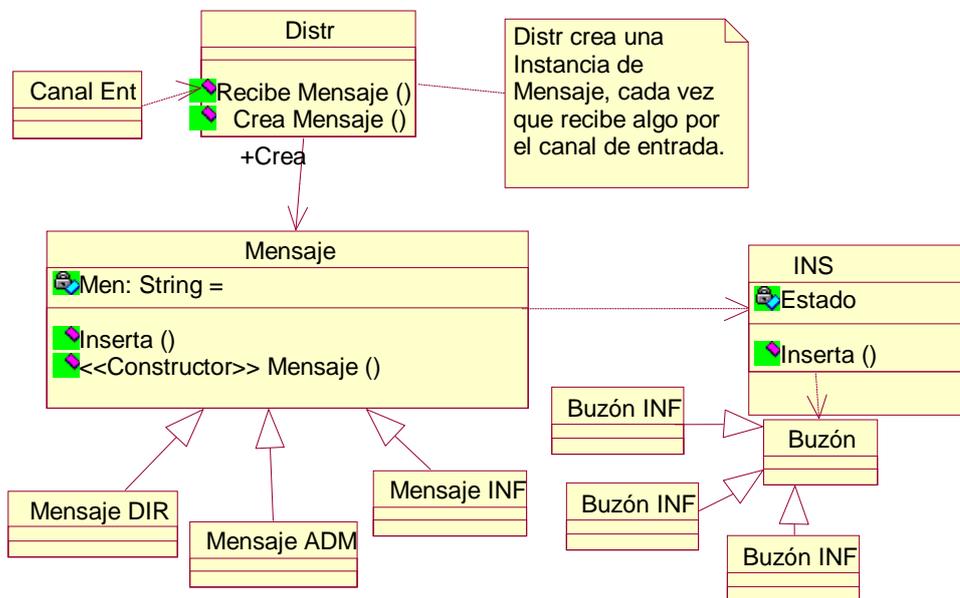


Fig. 2.11 Diagrama de objetos

- Diagrama de componentes

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

Los estándares que define UML son:

- Ejecutables
- Librerías
- Tablas
- Archivos
- Documentos

Cuando se desea modelar un proyecto para Internet donde nuestros componentes son JSP, HTML, y Scripts, se utiliza un estereotipo. Existen ya unos definidos por la WAE (Web Applications Extensión) y son:

- Ejecutables y bibliotecas.
- Tablas.
- API
- Código fuente.
- Hojas HTML.

Ejemplo gráfico

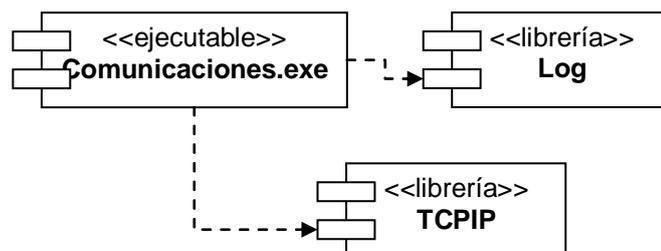


Fig. 2.12 Diagrama de componentes

- Diagramas de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo.

Un nodo se representa como un cubo, un nodo es un elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes.

Ejemplo gráfico

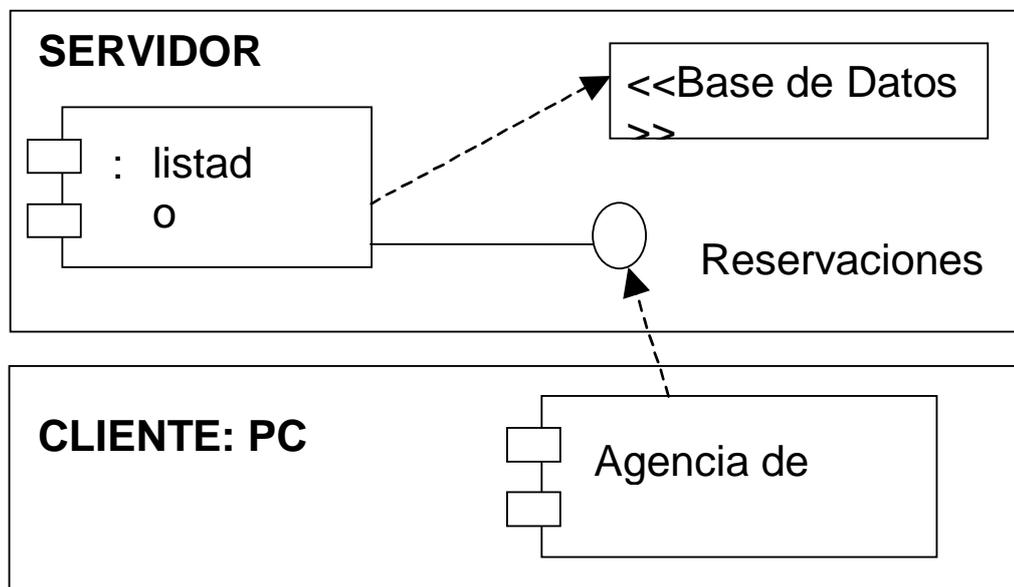


Fig. 2.13 Diagrama de despliegue

- Diagramas de secuencia

El diagrama de secuencia forma parte del modelado dinámico del sistema. Se modelan las llamadas entre clases desde un punto concreto del sistema. Es útil para observar la vida de los objetos en el sistema, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema.

Ejemplo gráfico

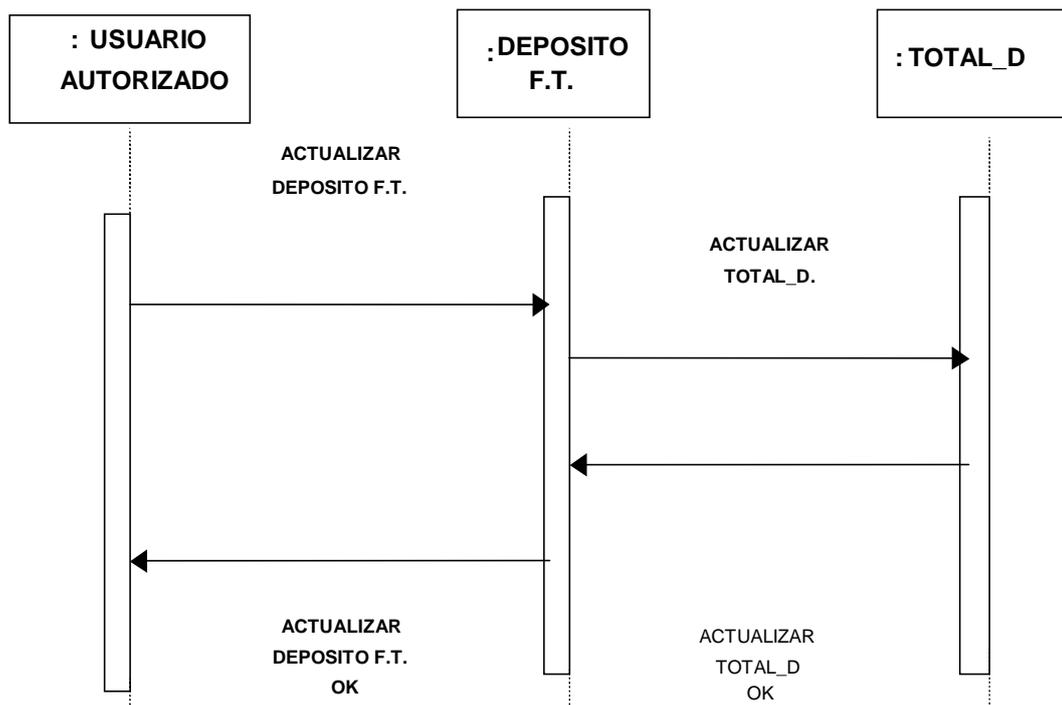


Fig. 2.14 Diagrama de secuencia

- Diagramas de colaboración

El diagrama de colaboración muestra la interacción de los objetos, a diferencia del diagrama de secuencia este diagrama se centra en el espacio, puede describir la ejecución de un caso de uso o simplemente de un escenario.

En este diagrama se representan los objetos, las relaciones entre objetos y finalmente los mensajes numerados secuencialmente. Se muestra las relaciones con todos los objetos, es utilizado por lo general en la fase de diseño.

Ejemplo gráfico

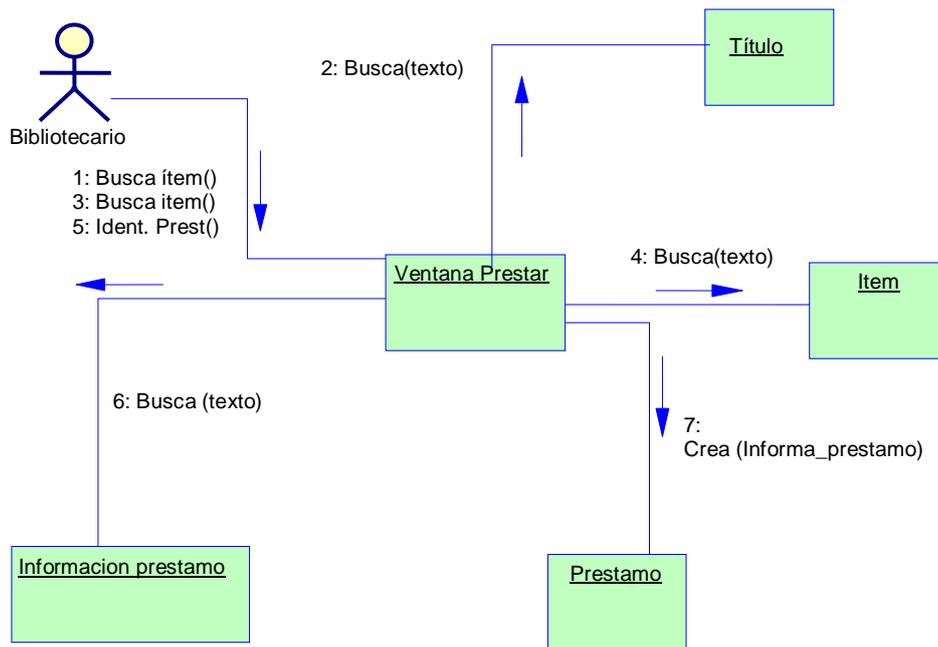


Fig. 2.15 Diagrama de colaboración

- Diagramas de actividades

Este diagrama es una variación del diagrama de estados, con este diagrama podemos modelar el comportamiento del sistema, se los utiliza para modelar un caso de uso, una clase o un método complicado.

Cada actividad se representa con un rectángulo con las esquinas redondeadas, una flecha representa la transición a otra actividad, también se muestra un inicio y un final. Se puede representar las decisiones.

Ejemplo gráfico

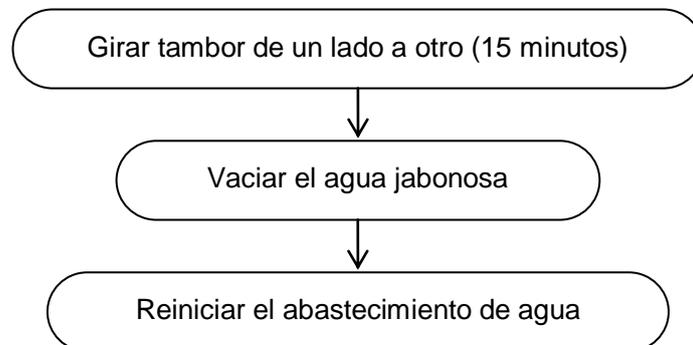


Fig. 2.16 Diagrama de actividades

- Diagramas de estado

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación junto con los cambios que permiten pasar de un estado a otro. Está representado principalmente por los siguientes elementos:

- Estado: Identifica un período de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos.
- Eventos: Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:
 - Condición que toma valores de verdadero o falso.
 - Recepción de una señal de otro objeto en el modelo.
 - Recepción de un mensaje.
 - Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular.
- Transición: Es una relación de tres o más estados de múltiples fuentes o múltiples destinos.

Ejemplo gráfico

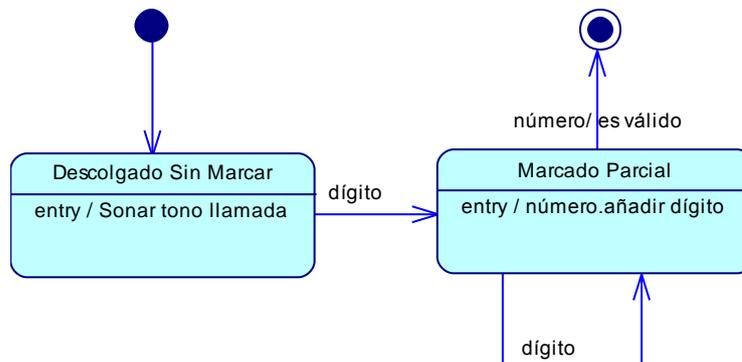


Fig. 2.17 Diagrama de estados

2.3.1.2 OMT (Técnica de modelado de objetos)

2.3.1.2.1 Origen de OMT

OMT nació para ayudar al desarrollo de software, James Rumbaugh el creador de esta metodología la estructura en el año 1991 en los laboratorios General Electric, la visión del autor era la definición de una metodología completamente

orientada a objetos capaz de representar de mejor forma a los objetos de la naturaleza y de una forma intuitiva.

Esta metodología se convirtió en una de las más utilizadas en el mundo por la facilidad de modelar los objetos y por su madurez. La metodología no es propietaria por lo cual muchas personas tienen acceso a toda la información para aplicarla en sus proyectos.

Rumbaugh nos indica que el propósito del analizar y diseñar el sistema se basa en 4 características fundamentales:

- Realizar simulaciones con entidades antes de construirlas
- Comunicación con el cliente
- Visualizar la solución
- Reducir la complejidad

Con el uso de OMT se puede representar todo el sistema de una manera fácil y entendible para el desarrollador y el cliente, la simulación es la base del éxito del análisis de sistemas permitiendo reducir el riesgo.

En la actualidad existen muchas metodologías orientadas a objetos, algunas propietarias y otras de libre distribución, muchas de ellas se las cataloga como metodologías orientadas a objetos porque manejan conceptos básicos de este paradigma. Algunos métodos de análisis y diseño los más conocidos son las siguientes:

- Object-Oriented Design (OOD), Booch.
- Object Oriented Analysis (OOA), Coad/Yourdon.
- Hierarchical Object Oriented Design (HOOD), ESA.
- Object Oriented Structured Design (OOSD), Wasserman.
- Object Oriented Systems Analysis (OOSA), Shaler y Mellor.
- Responsibility Driven Design (RDD), Wirfs-Brock

Todas las metodologías ayudaron a construir lo que actualmente es UML para el modelado del sistema y se lo utiliza en conjunto con OMT o con otras metodologías. La última versión de OMT es OMT 2.0.

2.3.1.2.2 Características Generales de OMT

Las características principales de OMT son las siguientes:

- Es una metodología que se basa principalmente en el modelado de los objetos como representarlos para identificar los requerimientos del sistema y la funcionalidad del mismo.
- OMT es de arquitectura abierta y por lo tanto está en constante evolución por la colaboración de varios grupos y personas naturales.
- La metodología se centra en el análisis y diseño más no en la implementación.
- Describir correctamente a los datos del sistema para luego definir las funciones de los mismos.

OMT utiliza tres tipos de modelos para describir a todo el sistema:

- 1. Modelo de objetos:** Este modelo define la estructura estática de los objetos del sistema como atributos, relaciones y operaciones. Es la base para el desarrollo de los otros dos modelos. Con el modelo se logra describir a detalle todas las características generales del sistema. Los diagramas de objetos ayudan a estructurar el sistema.
- 2. Modelo dinámico:** El modelo ayuda a representar los estados o transición de los eventos del sistema. En base a esto se puede modelar los cambios de estado del sistema, la secuencia de eventos y representar todo el comportamiento del sistema en base a eventos. Se utiliza el diagrama de estado para representar este modelo.
- 3. Modelo funcional:** Describe la forma en que los datos se transforman, se representa en el modelo a las funciones, actores, datos, flujos entre otros. Muestra lo que está haciendo el sistema ya sea si lo realiza o no, el diagrama de flujo de datos es el más importante en este modelo.

Con estos tres modelos la representación del sistema está completa desde cualquier aspecto, se debe tener mucho cuidado en modelar correctamente debido a que OMT es secuencial y un error demoraría el proceso de desarrollo del sistema.

Entre los tres modelos se describe todo el sistema como se muestra en la siguiente figura:

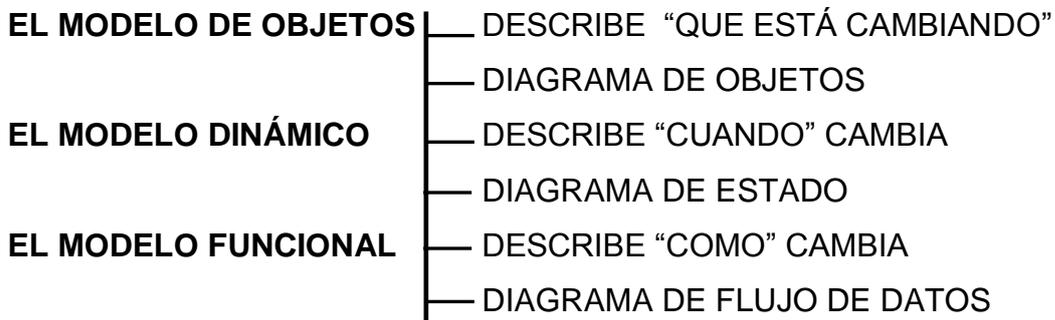


Fig. 2.18 OMT modelos Hossian, A., Rossi, B. y García Martínez, R.
Fuente: Revista del Instituto Tecnológico de Buenos Aires. Vol.23. Pág. 150-168. 2000.

2.3.1.2.3 Modelos en OMT

Modelo de Objetos

Para la estructuración de este modelo se aplican todos los conceptos básicos de la teoría orientada objetos. Se deben seguir los siguientes pasos para desarrollar este modelo:

- Determinar el problema
- Identificar y depurar cada uno de los objetos
- Desarrollar diccionario de datos
- Identificar y depurar las asociaciones entre los objetos
- Identificar y depurar los atributos de los objetos
- Utilizar herencia para reutilizar recursos
- Agrupar los objetos por módulos

El modelo se basa en los conceptos de clase y objeto, comprender estos dos aspectos es la base para diseñar el sistema.

Un objeto como es de conocimiento general en el paradigma OO es la representación de cualquier cosa del mundo real, nos ayuda a describirle el mundo que nos rodea de una forma simple y comprensible, los objetos pueden ser desde cosas tangibles como un esfero, avión, teléfono hasta cosas como facturas digitales, fotos digitales y muchos más.

El objetivo del modelo de objetos no es solo determinar todos los objetos que participan en el sistema sino integrarlos, esto se logra a través de las relaciones entre ellos, y en base a estas relaciones se descubre el comportamiento de cada objeto en el sistema.

El desarrollador debe tener la capacidad de modelar la realidad y aplicarla al sistema por medio de los objetos para resolver un problema, mientras más simple será de mejor comprensión tanto para el desarrollador y para realizar mantenimiento en el sistema a futuro.

Cuando el desarrollador determina los objetos es hora de transformar estos objetos a clases para manejar la implementación de los mismos en el sistema, las clases son las representación de los objetos, en las clases se describen todos los atributos del objeto, un ejemplo práctico entre una clase y un objeto, la casa equivale al objeto y el plano del casa a la clase.

Cada uno de los objetos definidos para el sistema presenta características propias las cuales se las conoce como atributos, el comportamiento de un objeto está definido por las operaciones o métodos las cuales se aplican sobre ellos u otros objetos del sistema. Las operaciones son las encargadas de cambiar los valores de los atributos del objeto o de otros, el método es la forma de describir las operaciones en las clases.

Los objetos interactúan constantemente en un sistema con otros elementos, esto se lo representa por medio de enlaces o asociaciones que son las relaciones que existen entre los objetos. Los enlaces es la forma en que se comunican los objetos en el sistema entre si.

La forma en que se representan las relaciones en OMT es las puede dividir en algunas categorías como:

Relaciones: Se identifica por medio de enlaces las conexiones con otros objetos, una relación identifica objetos comunes que trabajan entre si, una relación o asociación en la programación es vista como un puntero desde el objeto a otro.

Multiplicidad: Significa que un objeto se puede relacionar varias veces con uno o varios objetos.

Clasificación: Es la asociación de varios objetos que tienen características comunes, esto se convierte en una restricción ya que todos los objetos deben cumplir con estas condiciones.

Nombre de rol: Las asociaciones se las puede distinguir y agrupar para los objetos relacionados, el nombre del rol permite colocar restricciones para que algunos objetos se relacionen. Un nombre de rol se considera como un atributo de la clase cuyo valor es un conjunto de objetos relacionados.

Cualificación: Cuando se tienen relaciones de varios a varios o de uno a muchos se utilizan los calificadores, esto permite reducir el número de objetos.

Agregación: Esta relación indica que un objeto es parte de otro objeto, por ejemplo se tiene el objeto ciudad y el objeto provincia existirá una agregación entre ambas clases ya que una provincia contiene a varias ciudades.

La herencia es uno de los conceptos primordiales a la hora de diseñar el sistema, por medio de la herencia un objeto puede obtener características de uno o varios objetos logrando la reutilización de métodos y atributos para adaptarlos a nuevas necesidades. Cuando se modela los objetos y se crean las herencias entre objetos se convierte el modelo en una estructura jerárquica en la cual el grado de complejidad aumenta.

De allí OMT implementa la herencia por medio del siguiente concepto:

Generalización: Esto es la relación que existe entre la clase padre y la clase hija, por lo cual se van creando clases especializadas a partir de la clase padre original. Cada clase hija o subclase reutiliza las características de su predecesor y mejora e implementa nuevos atributos o métodos.

Todos los elementos que conforman el modelo de objetos se los puede agrupar para diseñar el sistema completo. Se busca que las clases se las agrupe

por módulos. En cada módulo se debe definir clases y atributos con nombres únicos y se puede utilizar una clase para describirla en distintos módulos.

Como conceptos avanzados en el desarrollo del modelo se pueden utilizar las clases abstractas, estas clases definen propiedades y operaciones las cuales serán redefinidas por las subclases. Esta clase abstracta es la base para la creación de las distintas subclases.

También la herencia múltiple genera controversias ya que no es una práctica muy común y muchas veces no es compatible con algunos sistemas OO, al realizar la herencia múltiple el sistema tiende a aumentar su grado de complejidad.

Como se muestra en este pequeño resumen de los elementos que utiliza OMT para definir el modelo de objetos es muy importante identificar los objetos correctamente, esto es la tarea esencial en este modelo ya que en base a estas descripciones el sistema tomará forma. Para mostrar los objetos se las representa gráficamente mediante los diagramas de clases es donde se muestran todas las clases que componen el sistema con sus atributos y relaciones, y existe el diagrama de casos concretos o de objetos donde se muestran las instancias como se relacionan y se crean en el sistema.

Modelo Dinámico

El modelo dinámico define el comportamiento de los elementos del sistema y sus cambios con respecto al tiempo.

Las relaciones dinámicas son complicadas de comprender, por lo que se pretende describir los siguientes aspectos del sistema:

- Eventos entre los objetos del sistema y sistemas externos
- Los estados de los objetos

La base para el modelado dinámico son los sucesos y estados, un suceso es una acción que cambia el valor de los objetos, y el estado es el valor del objeto en un espacio de tiempo. Para representar esta interactividad se los describe por medio del diagrama de estados,

En un sistema los objetos cambian sus valores constantemente por lo tanto es importante describir en que momento y quien activa estos cambios. El diagrama de estados muestra cuales fueron los sucesos que originaron estos cambios para que un objeto cambie de un estado a otro. Este diagrama será descrito para cada objeto que tenga un comportamiento dinámico en el sistema, al analizar este modelo es importante tomar los objetos críticos del sistema y mostrar todas las interacciones.

El diagrama permite analizar distintos escenarios del sistema, un escenario es conjunto de sucesos mostrados en secuencia que produce el sistema. El escenario puede ser dividido en varios o crear uno general, la tarea principal en la construcción de los escenarios es mostrar todos los objetos que interactúan como emisores o receptores según un suceso.

Como el diagrama está relacionado con el tiempo se puede formar ciclos finitos o continuos, los ciclos finitos tendrán un estado inicial y final, mientras que en los bucles se forma un círculo de sucesos o subrutinas que el sistema tiene que realizar constantemente en un intervalo de tiempo.

Para crear los cambios de estado el modelo se basa en las condiciones, que son funciones booleanas lógicas que determinar si cambia o no de estado en base a un suceso. La respuesta al cambio de estado es una operación. Las operaciones van atadas a un estado.

Para estructurar este modelo se pueden desarrollar los siguientes pasos:

- Describir los posibles escenarios
- Indicar los sucesos entre objetos
- Desarrollar el diagrama de estados para los objetos críticos del sistema

Modelo Funcional

Una vez creados los otros dos modelos, el modelo funcional muestra que sucede en el sistema con el comportamiento de los objetos.

Se describe la forma en que cambian los atributos de un objeto al interactuar con otros objetos del sistema. Para graficar la secuencia se utiliza el diagrama de actividades, en este se muestran el inicio o fin de una operación caso de uso o clases del sistema.

Por medio de los diagramas de actividades se describir los procedimientos lógicos, procesos del negocio y el flujo de los datos en el sistema. Este diagrama es muy parecido al diagrama de flujo de datos, la principal diferencia es que el diagrama de actividad soporta comportamiento en paralelo de los objetos.

Los diagramas de actividad permiten modelar el orden en que un proceso en particular realiza una acción con otros objetos. En otras palabras muestra la secuencia de reglas a seguir para iniciar y finalizar un proceso

Este diagrama describe el comportamiento de una clase en base a respuestas de procesos internos y no externos como en el diagrama de estados. Este diagrama contiene acciones, transiciones, objetos. Al realizar este diagrama se puede determinar las reglas de negocio que se utilizarán en el sistema.

La base para construir el modelo funcional es:

- Identificar las clases participantes en una actividad específica.
- Describir las reglas de negocio de tal actividad
- Identificar los estados de las acciones
- Describir las relaciones de las acciones con los objetos

2.3.1.2.4 Fases de OMT

El ciclo de desarrollo del sistema está representado por 4 fases en las cuales se crean varios modelos para estructurar todos los elementos necesarios para la creación del software. A continuación se describe cada una de las fases.

1. Análisis: Comienza en la descripción del problema el analista construye un modelo de la situación real que muestra las propiedades más importantes del sistema.

El modelo de análisis es una abstracción resumida y precisa de lo *que* debe hacer el sistema deseado y no de la *forma* en que se hará. Los objetos del modelo deberán ser conceptos del dominio de la aplicación y no

conceptos de implementación tales como clases. Un buen modelo podrá ser comprendido y criticado por expertos de la aplicación que no sean programadores. El modelo de análisis no deberá contener ninguna decisión de implementación, los objetos se describirán en términos de atributos y operaciones que son visibles para el usuario.

Su objetivo es desarrollar un modelo de lo que va a hacer el sistema. El modelo se expresa en términos de objetos y de relaciones entre ellos, flujo dinámico de control y las transformaciones funcionales.

Los pasos a seguir en el análisis son:

- Se escribe u obtiene una descripción inicial del problema.
 - Se concentra en entender el problema y modelar en el dominio del problema.
 - Se inicia con un texto que describe el problema.

- Se construye el modelo de objetos y sus relaciones.
 - Su objetivo es describir la estructura estática del software.
 - Se abstraen los conceptos de los datos que son más importantes para la aplicación.
 - Se describen gráficamente por los diagramas de objetos que definen las clases y sus relaciones

El documento que genera el modelo de objetos es el siguiente:

Modelo de objetos = diagrama de clases + diccionario de datos

- Se desarrolla el modelo dinámico.
 - Modela el comportamiento esencial del sistema mostrando los diferentes actores
 - Se captura el aspecto concerniente a la secuencia de las operaciones en el tiempo.

El documento que genera el modelo funcional es:

Modelo funcional = diagramas de casos de uso + diagramas de secuencia

- Se construye el modelo funcional.
 - Especifica el significado de las operaciones o métodos en el modelo de objetos y de las acciones en el modelo dinámico.
 - Muestra como se calculan los valores sin importar la secuencia, las decisiones ni la estructura de los objetos.
 - Se utilizan diagramas de actividades para mostrar las dependencias funcionales.

El documento que genera el modelo funcional es:

Modelo funcional = diagramas de actividades

Finalmente se verifican, iteran y refinan los tres modelos.

El documento que genera el análisis contiene:

Documento de análisis = definición del problema + modelo de objetos + modelo dinámico + modelo funcional.

2. Diseño del sistema: En base a los modelos anteriores se detalla la arquitectura del sistema. El sistema se lo divide en subsistemas según el problema a solucionar para organizar las funciones de cada subsistema. Se describe las posibles soluciones al problema y cuales son los propósitos de cada subsistema.

Los pasos que se llevan acabo son:

- Se organiza el sistema en subsistemas.
 - Cada subsistema comparte alguna propiedad en común. Las relaciones entre los subsistemas pueden ser: cliente – servidor o punto a punto.
 - La descomposición se puede organizar por capas horizontales o particiones verticales (cada uno proporciona un servicio).
- Se identifica la concurrencia inherente en el problema.
- El modelo dinámico nos permite identificar la concurrencia en el sistema.

- Se asignan los subsistemas a procesadores y a tareas.
- Cada subsistema se asigna a un procesador buscando satisfacer necesidades de rendimiento y minimizando la comunicación de los procesadores.
- Se selecciona la estrategia para implementar y administrar los almacenes de datos (archivos o bases de datos).
- Selección de la implementación del control del software.
- Existen dos tipos de control:
 - El control interno. Esta dado por el flujo de control en el programa o proceso.
 - El control externo. Esta dado por sucesos externos, los cuales pueden ser:
 - Control por procedimientos.
 - Control por sucesos.
 - Concurrentes
- Se consideran las condiciones de contorno.
 - Se trata de considerar como se hace la iniciación, terminación y como responderá a las fallas.
- Se establecen prioridades de compensación.

El documento que nos proporciona el diseño del sistema es:

Documento de diseño del sistema <=> estructura de la arquitectura básica del sistema (diagrama de procesos del negocio) + decisiones estratégicas de alto nivel

- 3. Diseño de objetos:** Se construyen los modelos de diseño en donde se detallan todos los elementos del sistema a nivel de implementación. Se describen los objetos y los procesos a desarrollar en el sistema, según el lenguaje de implementación se estructuran los modelos.

Los pasos que se realizan en el diseño de objetos son los siguientes:

- Se refinan las operaciones para el modelo de objetos a partir de los demás modelos:
 - Se busca una operación para cada proceso del modelo funcional.
 - Se define una operación para cada suceso del modelo dinámico.

- Se diseñan algoritmos para implementar las operaciones y las estructuras de datos

- Se optimizan las vías de acceso a los datos.

- Se implementa el control del software completando la aproximación propuesta en el diseño del sistema. Existen tres estrategias básicas para implementar el control:
 - Construir un sistema controlado por procedimientos.
 - Crear un motor de máquina de estados que responde a una tabla de transiciones y acciones.
 - Establecer un control como tareas concurrentes.

- Se ajusta la estructura de clases incrementando la herencia.

- Se diseña la implementación de las asociaciones. Las asociaciones conforman el pegamento en el modelo de objetos y proporcionan las vías de acceso entre los objetos. La implementación se hace dependiendo del tipo de asociación:
 - Asociaciones unarias. Estas asociaciones se establecen solamente en una sola dirección.
 - Asociaciones bidireccionales. Este tipo de asociaciones se pueden implementar de diferentes maneras:
 - Atributos en una dirección.
 - Como atributos en ambas direcciones.
 - Implementar como un objeto separado por medio de diccionarios.

- Se determina la representación exacta de los atributos de los objetos.
- Se empaquetan las clases y las asociaciones en módulos. El empaquetamiento implica:
 - Ocultar la información interna a los ojos externos (construir cajas negras con interfaces claras).
 - Determinar la coherencia de entidades, es decir, que cada clase o módulo debe de hacer una cosa y bien.
 - Construcción de los módulos. Cada módulo debe de tener una cohesión funcional, esto es, un propósito bien definido.

El documento que se genera el diseño de objetos es:

Documento de diseño de objetos = modelo de objetos detallado (diagrama de clases+diagrama de objetos visuales y no visuales) + modelo dinámico detallado (diagrama de casos de uso reales + diagrama de secuencia reales) + modelo funcional detallado (diagrama de actividades).

- 4. Implementación:** Los modelos creados en las anteriores fases se traducen en código, se tiene presente siempre los estándares de codificación, diseño de interfaces y usabilidad. Se busca que la implementación aplique todos los conceptos de POO (Programación Orientada a Objetos) para tener componentes reutilizables y facilitar el mantenimiento del sistema.

Rumbaugh explica que los modelos creados están basados en la creatividad del desarrollador, el punto de vista de un desarrollador puede discrepar con la de otros al diseñar los modelos, es importante discutir con todas las partes relacionadas usuarios, desarrolladores, analistas a fin de integrar ideas y construir modelos comprensibles para todos logrando satisfacer las necesidades del cliente.

En base a la metodología OMT se crearán los siguientes diagramas UML en el proyecto:

OMT	UML
Análisis	Diagrama de Procesos de Negocio
Diseño del sistema	Modelo Lógico Diagramas de Casos de Uso
Diseño de objetos	Diagramas de Secuencia Diagrama de Actividades Diagrama de Clases Diagrama de Objetos Modelo Físico

Tabla 2.1 Diagramas UML en OMT

En la fase de implementación se describirán las funciones principales del sistema, integración y pruebas de caja blanca y negra.

2.3.2 Metodología de la especificación de requerimientos

Esta fase permite conocer las expectativas del usuario. Para ello, se identifican los grupos de usuarios reales y posibles con sus áreas de aplicación, se revisa la documentación existente, se analiza el entorno operativo y sus requerimientos de procesado y se realizan entrevistas o cuestionarios a los usuarios.

Para todo este proceso existen técnicas formalizadas de especificación de requerimientos que más o menos concuerdan con las siguientes:

Se identifican las entradas del problema, los resultados deseados o salidas y cualquier requerimiento o restricción adicional en la solución.

- **Obtener información acerca de lo que los usuarios desean**

Los requerimientos son el punto en que el cliente y el proyecto de desarrollo de software se unen, esta unión es necesaria para poder construir un software que satisfaga las necesidades del cliente.

Si los requerimientos se enfocan a describir las necesidades del cliente, entonces es lógico que la obtención de esta información sea de primera mano.

Esto es, mediante entrevistas con el cliente o buscando documentación que describa la manera que el cliente desea que funcione el sistema de software.

Las necesidades y/o requerimientos del cliente evolucionan con el tiempo y cada cambio involucra un costo. Por eso es necesario tener archivada una copia de la documentación original del cliente, así como cada revisión o cambio que se haga a esta documentación

Como cada necesidad del cliente es tratada de diferente forma, es necesario clasificar estas necesidades para saber cuales de ellas serán satisfechas por el software y cuales por algún otro producto del sistema.

- **Clasificar y estructurar requerimientos**

El clasificar requerimientos es una forma de organizarlos, ya que hay requerimientos que por sus características no pueden ser tratados igual que a otros. Por ejemplo, los requerimientos de entrenamiento de personal no son tratados de la misma manera que los requerimientos de una conexión a Internet.

La siguiente es una recomendación de como pueden ser clasificados los requerimientos aunque cada proyecto de software pueda usar sus propias clasificaciones.

- Requerimientos del "entorno"

El entorno es todo lo que rodea al sistema. Aunque no podemos cambiar el entorno, existen ciertos tipos de requerimientos que se clasifican en esta categoría, ya que el sistema usa el entorno y lo necesita como una fuente de servicios necesarios para su funcionamiento. Ejemplos del entorno podemos mencionar: sistemas operativos, sistema de archivos, bases de datos.

- Requerimientos "ergonómicos"

El más conocido de los requerimientos ergonómicos es la interfaz con el usuario o GUI (Graphic User Interface). En otras palabras, los requerimientos ergonómicos son la forma en que el ser humano interactúa con el sistema.

- Requerimientos de Interfaz

La interfaz es como interactúa el sistema con el ser humano o con otros sistemas (el enfoque es prácticamente el opuesto a los requerimientos ergonómicos), La interfaz es la especificación formal de los datos que el sistema recibe o manda al exterior. Usualmente se especifica el protocolo, el tipo de información, el medio para comunicarse y el formato de los datos que se van a comunicar.

- Requerimientos funcionales

Estos son los que describen lo que el sistema debe hacer. Es importante que se describa el ¿Que? Y no el ¿Como? Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.

- Requerimientos de desempeño

Estos requerimientos nos informan las características de desempeño que debe tener el sistema. ¿Que tan rápido?, ¿Que tan seguido?, ¿Cuántos recursos?, ¿Cuántas transacciones?

Este tipo de requerimientos es de especial importancia en los sistemas de tiempo real en donde el desempeño de un sistema es tan crítico como su funcionamiento.

- Disponibilidad (en un determinado periodo de tiempo)

Este tipo de requerimientos se refiere a la durabilidad, degradación, portabilidad, flexibilidad, contabilidad y capacidad de actualización. Este tipo de requerimiento es también muy importante en sistemas de tiempo real puesto que estos sistemas manejan aplicaciones críticas que no deben estar fuera del servicio por periodos prolongados de tiempo.

- Entrenamiento

Este tipo de requerimientos se enfoca a las personas que van usar el sistema. ¿Que tipo de usuarios son?, ¿Que tipo de operadores?, ¿Que manuales se entregarán y en que idioma?

Este tipo de requerimientos, aunque muchas veces no termina con código dentro del sistema, son muy importantes en el proceso de diseño ya que facilitan la introducción y aceptación del sistema en donde será implementado.

- Restricciones de diseño

Muchas veces las soluciones de un sistema de software son normadas por leyes o estándares, este tipo de normas caen como "restricciones de diseño".

- Materiales

Aquí se especifica en que medio se entregara el sistema y como esta empaquetado. Es importante para definir los costos de industrialización del sistema.

- **Identificar los niveles jerárquicos del sistema y clasificar los requerimientos según estos niveles.**

Un sistema tiene diferentes niveles de jerarquía en sus componentes. Los componentes más pequeños se agrupan en módulos y estos a su vez se pueden agrupar en subsistemas. Si el sistema computacional se aloja en una red de telecomunicaciones entonces es probable que nuestro sistema entero pueda ser agrupado junto con otros componentes en un nodo. Entonces varios nodos pueden ser agrupados en una red. Si vemos a la red como un todo y nos concentramos en las interacciones de ésta y su entorno entonces la red y sus usuarios forman el Nivel cero de esta red.

Para poder alojar un requerimiento en el lugar correcto, hay que tomar en cuenta esta jerarquía. Es de hacer notar que entre más alto se aloje un requerimiento en la jerarquía, más costoso será su implementación.

- **Especificar formalmente los requerimientos de acuerdo al nivel de audiencia que se desea.**

Una vez que ya han sido recopilados, clasificados y alojados en el nivel de jerarquía que les corresponde, hay que poner por escrito los requerimientos del sistema. Sin embargo, ¿Hasta donde llegará esta especificación?

La habilidad de comunicarse con una audiencia debe de ser balanceada con la necesidad de especificar los requerimientos precisamente y sin ambigüedad alguna.

Existen ya algunos lenguajes abstractos y herramientas para el manejo de especificaciones de requerimientos. Tales como el ASN.1 para interfaces de telecomunicación. Sin embargo, el uso de estos lenguajes implica que la audiencia deba tener un conocimiento general de este tema.

Por el contrario, si usamos el lenguaje natural, podemos correr el riesgo que nuestra audiencia confunda los términos del requerimiento.

El UML incorpora en sus estructuras los casos de uso (Use cases) propuestos por Ivar Jacobsson. Esta Estructura maneja especificaciones formales y abstractas pero se acompaña de una descripción textual y estructurada del requerimiento del sistema. En este contexto entraría dentro de las características de las metodologías.

Hay que encontrar un punto medio entre precisión y entendimiento como se muestra en la Fig. 2.19

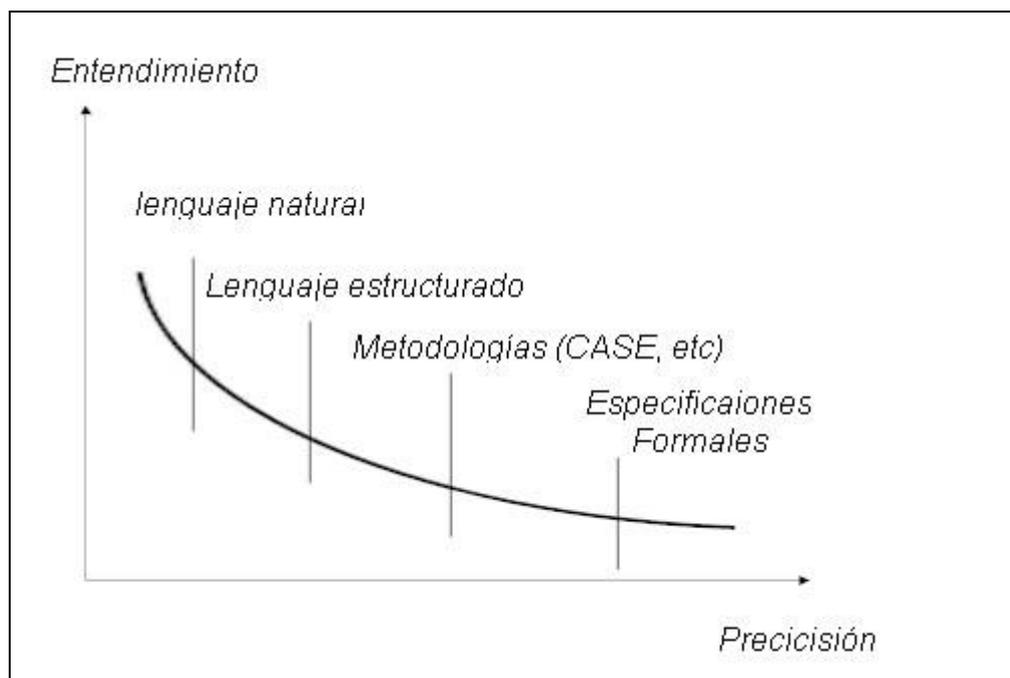


Fig. 2.19 Especificaciones (Precisión Vs. Entendimiento)

Una vez que el requerimiento ha sido plasmado, deberá de tener una identificación única. Esto es con el fin de facilitar el manejo de requerimientos. Tema que explicamos a continuación.

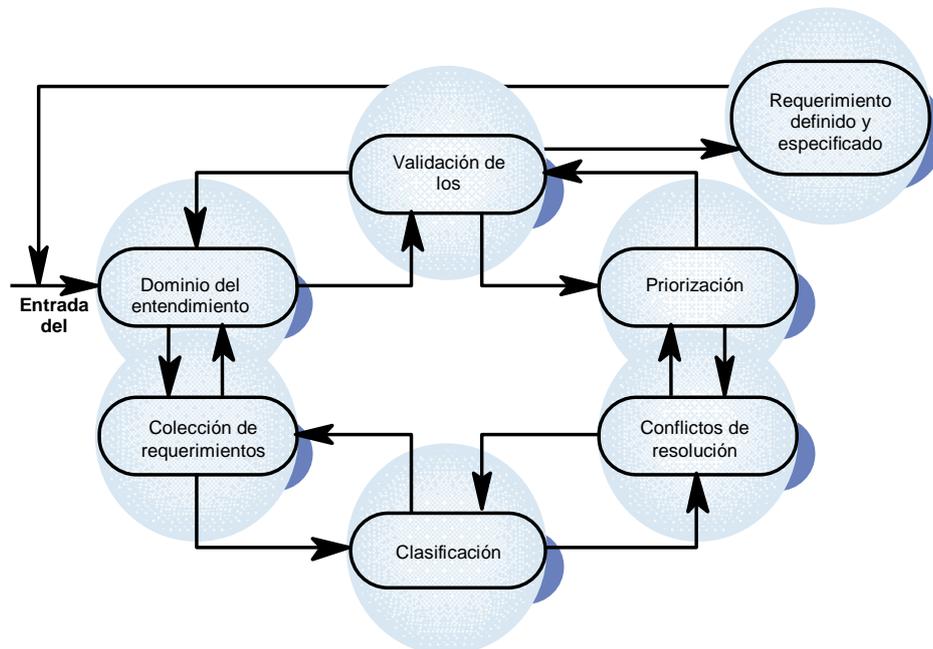


Fig. 2.20 Proceso de validación de requerimientos

2.3.2.1 Manejo de requerimientos

De acuerdo con el "Capability Maturity Model" (CMM) el manejo de requerimientos involucra:

“Establecer y mantener un acuerdo con el cliente sobre los requerimientos del proyecto de software. Este acuerdo son los requerimientos del sistema alojados al software.”

“Este acuerdo cubre requerimientos técnicos y no técnicos (como fechas de entrega). El acuerdo forma las bases para estimar, planear, ejecutar y monitorear el proyecto de desarrollo de software a través de todo su ciclo de vida.”

“Bajo las restricciones del proyecto, el grupo de manejo de requerimientos toma las medidas necesarias para que los requerimientos que están bajo su responsabilidad estén documentados y controlados”

"Para lograr el control de los requerimientos, el grupo de requerimientos revisan los requerimientos antes de que estos sean incorporados al proyecto de

software y cada vez que los requerimientos cambian los planes, productos, y actividades son ajustadas para quedar en línea con los nuevos requerimientos de software".

En otras palabras, para obtener el nivel que requiere el CMM en manejo de requerimientos se debe tomar en cuenta dos cosas.

Que los requerimientos deben de ser revisados (y aprobados) por el grupo de requerimientos y en ninguna forma deben ser impuestos en su totalidad por presiones externas al proyecto.

El requerimiento técnico podrá ser impuesto por el mercado o presiones de la competencia, pero entonces los requerimientos no técnicos (Calidad, Costo y Tiempo de entrega) deberán estar especificados de común acuerdo con el grupo de requerimientos del proyecto de software.

2.3.3 Metodología POS (Point of Sale)

2.3.3.1 Punto de Venta

Existen algunos conceptos interesantes de puntos de venta propuestos por varias empresas, entre estos los más significativos han sido los siguientes:

“Los Puntos de Venta han sido elegidos en función de su capacidad de aproximación al mercado residencial. Son empresas que ofrecen al gran público las mejores soluciones de telecomunicaciones, asesorándole sobre los equipos y servicios más adecuados para sus necesidades concretas.” Telefónica

“POS, es un espacio físico donde se encuentran productos para ser vendidos a los consumidores” Grupo WebFinance.

Resumiendo estos dos conceptos un punto de venta es el establecimiento donde el consumidor puede adquirir el producto. Es lógico suponer que si existe un punto de venta que es entonces un punto de compra. Un punto de compra es simplemente un lugar físico donde el producto esta en exhibición a la vista y alcance del cliente. El detalle del proceso de venta se encuentra en el anexo A.

Los puntos de venta nacieron principalmente por las siguientes causas:

- Satisfacer las necesidades de sus clientes.
- Diferenciarse de la competencia.
- Mejorar el rendimiento de sus categorías.

Cuando se habla del software de puntos de venta el concepto cambia y aumenta su funcionalidad, llegando a ser un sistema que se instala en una o varias computadoras ubicadas en el área donde se cierra la transacción compra-venta; teniendo como propósito fundamental registrar todos los movimientos y operaciones relacionadas con la venta de los productos en forma ágil y segura.

El concepto de puntos de venta es aplicado desde hace algún tiempo en los negocios, pero la orientación de este concepto hacia el software fue originalmente propuesto por empresas como IBM y Microsoft para buscar nuevos nichos de mercado, brindando soluciones en software y hardware. El impacto de estas empresas generó nuevas soluciones para puntos de venta en el mundo.

Estas soluciones ayudaron en la reducción de costos y la tecnología utilizada permitió una mayor eficiencia en sus transacciones comerciales. A medida que las tecnologías siguieron creciendo se aumentaron nuevas características a estos puntos de venta y por consiguiente los requerimientos para el software, el uso de periféricos permitió que se capturen rápidamente datos y agilicen las transacciones.

Se implementaron dispositivos como cajas y balanzas electrónicas. La evolución de estos dispositivos ha sido gigantesca, los dispositivos de los 80s eran grandes y complejos en su uso, mientras que en los 90s se reducen los tamaños y se crean nuevas funcionalidades, en el 2000 todos los dispositivos ya son digitales y poseen características inalámbricas que facilitan una transmisión de datos remota.

El punto de venta físico no nace de la noche a la mañana, con estudios de mercado se crea un punto de venta basándose en las necesidades de un segmento de mercado en particular.

Varias técnicas nuevas de venta al cliente son implementadas en el punto de venta, como el merchandising y la venta en autoservicio. “Merchandising es un conjunto de técnicas, destinadas a obtener decisiones favorables del consumidor en el punto de venta, respecto de los productos de nuestro interés, mediante la administración de la exhibición para resaltar sus atributos y producir atracción.”
Mercadeo.com

Normalmente todo comerciante desarrolla técnicas de merchandising, ya que debe presentar su establecimiento de manera atractiva frente a la competencia y para que las personas que ingresan al mismo comprendan los productos en oferta. Es el denominado "merchandising de punto de venta".

El mercadeo se ha desarrollado mucho en los últimos años, cambiando las funciones básicas del consumidor, brindándole una mayor independencia y flexibilidad como muestran los puntos a continuación:

- El comprador tiene la libertad de escoger su producto, transportarlo hasta la caja y ponerlo en la bolsa; actividades que antes las realizaba el vendedor.
- Libertad de observar todos los productos que desee.
- Los productos son los actores de toda la acción comercial para la decisión final de compra.
- El uso de dispositivos de información como escáner o pesas digitales permiten al consumidor aproximarse al total de sus compras y ajustarse rápidamente a su presupuesto.

Al utilizar mercadeo y merchandising mejoramos todas las características del punto de venta. Pero la forma en que nace un punto de venta se basa en un estudio de mercado, el cual se presenta en tres fases:

Análisis: Se realiza un levantamiento de la información, determinando necesidades y clientes potenciales.

Los datos esenciales para definir un punto de venta son:

- Producto: Características.

- Clientes: Datos personales
- Tipo de Venta: Por peso, cantidad. o volumen
- Transacciones comerciales: Compra, venta, devoluciones.
- Formas de Pago
- Manejo de precios
- Emisión de documentos contables
- Control de la contabilidad
- Control de Inventario
- Tecnología : Hardware , software y comunicaciones
- Seguridad: Física y lógica

Planeación: Se diseñan los productos, y las estrategias para atacar el mercado potencial.

En base a los datos del análisis se catalogan los productos, tipos de cliente, formas de pago e integración con otras áreas de la empresa.

Se determinan los recursos tecnológicos a implementarse:

- Hardware: Periféricos POS
- Software: Sistema POS
- Comunicaciones : Internet o intranet

Para atacar los mercados potenciales se deben aplicar las reglas de negocios y estas se ven afectadas por factores externos, estrategias de la empresa, gerencia y tecnología.

Se define el diagrama de actividades, con asignación de recursos y tiempos de implementación del Punto de Venta.

Ejecución: Se lanza el producto y el punto de venta.

Se realiza la implementación del punto de venta y la distribución de los productos, el personal es capacitado en los procesos del punto de venta.

En el punto de venta se expone el producto al cliente y se verifica que todo el proceso de venta funcione correctamente,

Finalmente se evalúa la eficiencia en las transacciones, la integridad de los datos y reducción de costos.

Los puntos de venta son necesarios en la actualidad tanto para las PYMES como para las grandes empresas, el marketing es una herramienta fundamental para la creación de los mismos.

Todos los negocios que deseen ofertar un producto deben tener un punto de venta sea cual sea el producto, por ejemplo:

- Productos deportivos
- Computadoras y accesorios electrónicos
- Música y videos
- Instrumentos Musicales
- Ropa, zapatos.
- Herramientas de trabajo

La aplicación de un punto de venta solo esta restringida por las necesidades que tiene la empresa en su utilización.

POS	OMT
Análisis	Análisis Diseño del sistema Diseño de objetos
Planeación	Diseño del sistema Diseño de objetos Implementación
Ejecución	Pruebas

Tabla 2.2 Relación de fases POS con OMT

2.3.3.2 Tipos de Puntos de Venta

Por temporada

Un punto de venta el cual funciona según una temporada específica del año y adopta su fachada y productos según sea la ocasión.

Este tipo de puntos de venta tiene las siguientes características:

- Puede contener exhibiciones impactantes, divertidas y diferentes, maximizando la compra por impulso.
- Cada temporada se renueva el concepto de exhibición, generando alegría, color y magia en la compra.
- Novedosos sistema de exhibición con una excelente capacidad y versatilidad en el manejo del producto.

Por su propiedad

Puntos de venta autorizados

Son puntos de venta de propiedad de una empresa en particular o no. Tiene la capacidad de promocionar el nombre de una marca patentada como parte de su línea de negocio. Al ser un punto de venta autorizado tendrá ventaja competitiva sobre otros puntos de venta que no lo sean.

Puntos de venta especializados

Son puntos de ventas que tienen productos de varias empresas, especializándose en líneas como electrodomésticos, computadoras, software.

Puntos de venta como parte de una franquicia

El punto de venta ha sido adquirido como parte de una franquicia.

Por su localización

Punto de venta local

Un punto de venta que se encuentra cerca de la casa matriz.

Punto de venta remoto

Punto de venta que se encuentra a distancias extremas de la casa matriz, estos puntos de venta suelen ser administrados desde la casa matriz.

Por el alcance

Punto de venta público

Este punto de venta está al alcance de todos los consumidores, orientado a un mercado masivo.

Punto de venta privado

Solo tiene acceso personas socias de la empresa, orientado a segmentos de mercado especializados.

2.3.3.3 Movilidad en puntos de venta

Hoy en día un negocio no puede darse el lujo de mantenerse estático frente a las nuevas tecnologías, los negocios deben moverse y entrar a nuevos mercados, la movilidad aumenta la cobertura y satisface a más clientes, para esto la empresa debe contar con una sólida infraestructura tecnológica y organizacional.

El Internet es el punto de partida para la creación de puntos de venta móviles, y el uso de tecnologías inalámbricas como wi-fi logran mover físicamente un punto de venta, muchas empresas en el mundo ya lo están utilizando pero en nuestro país la inversión para tecnologías de comunicación inalámbricas no es una práctica muy común.

Los puntos de venta online han proliferado en los últimos años. Y se crean nuevos conceptos como el “e-merchandising” como herramienta de comunicación corporativa de empresas virtuales y E-Loyalty para el manejo de la lealtad del cliente.

No cabe duda que el Internet y el comercio electrónico estén provocando cambios en muchas vertientes del comportamiento del consumidor ya que amplía y facilita información para la toma de decisiones y maximiza la capacidad de elección del cliente.

A medida que el formato on-line es usado por un mayor número de personas se incrementa su utilidad, los individuos pueden encontrar lo que realmente buscan, y construir lo que Moynagh y Worsley (2001) denominan un mundo a

medida. La capacidad de desarrollar el e-merchandising en las tiendas virtuales es un camino por explorar, con enormes posibilidades de desarrollo e importantes contribuciones a la configuración de la imagen corporativa on-line.

Los puntos de venta móviles son un nicho de mercado completamente desatendido por la industria. Es aquí donde se necesitan soluciones innovadoras con sistema de gestión de puntos de venta móviles dirigido a aquellas empresas cuyo modelo de negocio demanda una solución a sus problemas de movilidad o de dimensiones reducidas.

A que empresas pueden beneficiar los puntos móviles:

Empresas de reducidas dimensiones

Aquellas empresas que, por su modelo de negocio, dispongan de puntos de venta en los que no resulta viable la instalación de un TPV (Terminad de Punto de Venta) convencional (micro tiendas, corners, stands)

Empresas móviles

La solución perfecta para pequeños stands, corners y todo tipo de pequeños puntos de venta que no puedan depender de conexiones fijas para sus comunicaciones con la central y que, además, requieran una solución de reducido tamaño y gran funcionalidad. Tanto si su punto de venta cambia de ubicación por toda la ciudad como si sólo se desplaza dentro de un centro comercial.

Tabla 2.3 Empresas Beneficiadas con un punto venta móvil

En estudios recientes se menciona que muchas empresas se están enfocando en cambiar su esquema de puntos de venta de productos por puntos de venta de servicios y estos servicios podrían ser publicados en el Internet utilizando tecnología como los Web Services, permitiendo que cualquier empresa acceda a ellos.

2.3.3.4 Software de puntos de venta

Al ERP Venture le falta una implementación completa sobre los puntos de venta, tiene un módulo de ventas pero no abarca todas las características que debería tener un punto de venta.

Primeramente se analizan los paquetes existentes en el mercado para puntos de venta en español, presentados en el siguiente cuadro:

Software	Descripción
Microsip Software Administrativo y de Punto de Venta	Con este poderoso sistema controle desde un pequeño negocio con una sola caja, hasta un gran consorcio de sucursales multicajas.
Atrex Inventory Control	Atrex Inventory Control es un programa de punto de venta e inventario que nos permitirá un control completo sobre cada uno de nuestros artículos y sus niveles de existencia en nuestro almacén.
Billpro	Billpro es un programa de punto de venta diseñado para su uso en bares, restaurantes y hoteles.
TPV Comercios	La solución TPV para cualquier tipo de comercio. Si buscas un programa para la gestión completa y rápida de tu negocio, esta es una buena opción.
SimplyGest	SimplyGest Profesional es un programa para la gestión comercial de una empresa con el que podrás mantener el control del las compras, ventas, de las existencias en almacén, y exportar estos datos a programas de contabilidad, entre otras funciones.

IM PROJECTS	<p>IM PROJECTS ha pensado en el COMERCIO - tiendas, comercio minorista, cadenas de tiendas y franquicias comerciales - y le presenta una solución software innovadora, de carácter ágil e intuitivo, para la gestión integral de su negocio.</p>
Ventas 2000 PyME	<p>Obtendrá reportes como cortes de caja, entradas y salidas de mercancía, traspasos, remisiones, inventarios.</p> <p>Llevará el control de los movimientos financieros realizados en su negocio, así como el control de sus gastos y los saldos de bancos, proveedores y clientes.</p>
AdmiTienda Light	<p>AdmiTienda light es un software que le ayudara en su establecimiento mercantil, con el llevará cuenta de qué productos tiene, cuando se incorporaron a la base de datos, quien reabasteció la información, quien cuando y cuanto vende su personal , personalizando la información que se imprime, puede usar sus notas o impresoras de tickets, con una fácil maniobra puede auxiliarse de un lector de código de barras, el diseño del programa permite usar exclusivamente el teclado en las principales secciones, por ejemplo al momento de capturar una venta, un producto o reabastecer la información en sus productos, todos los resúmenes como el inventario y las ventas son exportables a Excel e imprimibles</p>

CTA Punto de Venta	<p>CTA Punto de Venta es un sistema creado para cajeros y clientes. La pantalla de caja proporciona información tanto al cajero como al cliente.</p> <p>Además CTA Punto de Venta soporta la mayoría del hardware estándar de punto de venta: lectores de código de barras, impresoras de ticket y cajones de dinero.</p>
--------------------	---

Tabla 2.4 Software de Punto de Venta existente en el mercado [De Remate]

El ERP Venture busca consolidar puntos de venta con todos sus módulos para competir a nivel internacional, pero además desea realizar una versión del módulo de puntos de ventas para Internet lo que le llevaría a la cabeza entre los demás programas ya que estos no son para Web y tampoco se integran aun sistema completo para la empresa como es un ERP. Los ERP más utilizados en el mundo son:

Microsoft Great Plains Enterprise
Tecnología
Escrito en C++, HTML, XML, Visual Basic, Dexterity. Con la base de datos MS-SQL.
Módulos
<ul style="list-style-type: none"> - Inventarios , procesamiento de ordenes de compra - Manejo de Proyectos - CRM - Manejo de Recursos humanos - Finanzas - E-commerce

Tabla 2.5 ERP Microsoft GPE

MAS 500
Tecnología
Visual Basic y Visual C++. Base de datos MS-SQL Server.
Módulos
<ul style="list-style-type: none"> - Contabilidad - Cuentas por cobrar - Nominas - Ordenes de Venta - Activos fijos - Planeación de recursos de manufactura - Cuentas por pagar - Manejo de Caja - Inventario - Recurso humanos

Tabla 2.6 ERP MAS 500

mySAP All-in-One
Tecnología
Hecho con tecnología de componentes de SAP.
Módulos
<ul style="list-style-type: none"> - Finanzas - Operaciones - Manejo del capital humano - Servicios Corporativos

Tabla 2.7 ERP mySAP

Epicor
Tecnología
Escrito en C++, Visual Basic. Con la base de datos MS-SQL.
Módulos
<ul style="list-style-type: none"> - CRM - Servicios empresariales - E-commerce - SCM - Cuentas por cobrar y pagar - SRM - Activos Fijos - E-Portales - E-intelligence

Tabla 2.8 ERP Epicor

Estos líderes en el mercado de los ERP's tiene módulos esenciales para administrar la empresa y mejorar la productividad, y además proveen de una interfaz Web que permite mantener al negocio en línea en todo momento.

El ERP Venture es un sistema que se adapta fácilmente a los requerimientos de las empresas y tiene una flexibilidad capaz de crecer en proporción al crecimiento de la empresa, para que el ERP de Venture pueda competir con otros ERP's debe integrar un nuevo módulo de punto de venta que sea capaz de acoplarse a cualquier estrategia de negocio y que tenga acceso remoto por Internet.

El módulo de punto de venta primeramente debe satisfacer el mercado nacional con características robustas como:

- Aumentar el volumen de compra de los clientes
- Disminuir el riesgo al evitar el manejo de dinero en efectivo
- Rapidez de transacciones
- Movilidad del Punto de venta
- Configuración de políticas de descuentos, créditos.
- Desenvolverse en ambientes Internet e Intranet.

Con estas características cada empresa podrá disponer de sus puntos de venta, logrando una mejor cobertura de sus productos y satisfaciendo en todo momento a sus usuarios con el módulo de puntos de venta y en ambiente al Internet se puede constituir una empresa que oferte sus productos con un horario 24/7.

2.3.3.5 Hardware de Puntos de Venta

En el mercado actual las grandes empresas de venta de computadores personales han optado por la venta de sistemas completos de Puntos de Venta como es el caso de HP, IBM entre otros.

Hewlett Packard cuenta con sistemas como el HP POS rp5000 que viene con el software incorporado POS y una máquina capaz de adaptarse a las necesidades del negocio.



Fig. 2.21 HP POS rp5000

Las características de este sistema son similares a una computadora personal, pero se adjunta los periféricos que constituyen el punto de venta como la caja registradora, el lector de código de barras y la impresora de recibos.

Este sistema cuenta con estándares abiertos los cuales facilitan un desarrollo personalizado para satisfacer los requerimientos de nuevos mercados y adaptarse a las políticas de cada país. Este estándar se lo conoce como Open Point Of Sale.

Otra compañía que ha tenido grandes avances en los sistemas POS es IBM, que tiene una línea muy completa; desde sistemas completos hasta los periféricos de uso diario, siendo líderes en el mercado en soluciones POS compiten con HP para dominar el mercado mundial.

2.3.3.6 Estándares POS de desarrollo

Actualmente el desarrollo de un sistema POS se ha facilitado notablemente gracias a los estándares, un desarrollador puede utilizar periféricos de distintas compañías e integrarlas al sistema de forma transparente, las empresas han trabajado conjuntamente con los desarrolladores para crear interfaces lógicas, funciones y librerías que ayuden a generar soluciones rápidas, escalables y altamente compatibles con la mayoría de dispositivos del mercado.

HP Open Point-of -Service

Open POS es una solución propuesta por HP para mejorar el desarrollo e implementación de sistemas POS, la constante demanda del mercado exige que los sistemas POS sean flexibles y altamente escalables.

La mayoría de las empresas tenía que comprar nuevo software o hardware para mejorar el rendimiento de su sistema POS, el estándar Open POS permite adaptar su sistema a nuevas necesidades y brindándole varios beneficios como son:

- Reducir los costos y mejorando los ingresos.
- Transacciones sencillas y rápidas mejorando la experiencia para el cliente.
- Permite al sistema POS seguir creciendo según los nuevos requerimientos.

El mercado actual cambia constantemente debido al constante crecimiento de la tecnología, la demanda de tiempos bajos para las transacciones comerciales se han convertido en una característica fundamental para los sistemas POS los consumidores exigen rapidez y eficiencia mientras que la empresa exigen confiabilidad y integridad en el sistema. Los sistemas actuales POS se manejan por un cajero, en el futuro se espera que por medio de audio o video se aumente la rapidez en las transacciones y mejore la atención al cliente.

El estándar Open POS ofrece 5 características claves para integrar este estándar en los sistemas a implementar:

- **Control del hardware, software y los costos laborales:** Ahorrar dinero es la propuesta de este estándar evitar desechar los sistemas POS

implementados en la empresa por otros nuevos, con la utilización de este estándar se pueden aplicar actualizaciones tanto de software como de hardware para acoplarse a nuevas necesidades y ayudar a los empleados a recibir una apropiada capacitación en los nuevos dispositivos o funciones del sistema. El estándar busca proteger la inversión de la empresa aplicando este estándar para que el sistema crezca.

- **Mejora la agilidad del negocio:** Mantener el negocio siempre evolucionado a medida que la tecnología cambia es un verdadero reto para las empresas en la actualidad, por medio del estándar se mejoran las operaciones transaccionales, especificaciones para cobrar por medio de tarjetas de crédito mejoran notablemente los tiempos, se puede utilizar el Internet para comunicarse con otras sucursales.
- **Mejora la eficiencia en las operaciones:** La planeación oportuna para la adquisición de nueva mercadería, inventario detallado son funcionalidades que pueden ser implementadas fácilmente. Mejorar los procesos de compra se los puede obtener integrando el sistema POS a sistemas como un ERP o un SCM.
- **Acceso a la información:** La información oportuna permitirá tomar decisiones a los gerentes para evitar pérdidas o pensar en nuevas oportunidades en el mercado. Se puede sacar estadísticas por cliente o por productos para analizar el mercado actual. Con toda esta información se pueden realizar benchmarkings con otras empresas o por años y diagnosticar como está la empresa posicionada en el mercado.
- **Mejorar la atención al cliente:** En la actualidad la clave para que un negocio tenga éxito es la atención al cliente, la experiencia de la compra debe ser lo más fácil posible para el cliente y que el vendedor tenga la información apropiada para responder a cualquier inquietud del cliente. El estándar propone mejorar la disponibilidad de la información para el vendedor y mejorar los tiempos de atención al cliente logrando un cliente satisfecho.

OPOS

OPOS significa Ole for Point of Sale, en pocas palabras un estándar de componentes originalmente creado por Microsoft, Epson y Fujitsu. Este fue el

primero en ser adoptado mundialmente para el desarrollo de aplicaciones POS en particular para la plataforma Windows.

OPOS tuvo su primera versión en 1996 y su última versión salió en el 2001, OPOS utiliza la tecnología COM, Active X y principalmente componentes OLE. El estándar tiene como objetivo crear componentes capaces de controlar los periféricos usando la tecnología OLE con el fin de permitir el desarrollo de software de propósitos generales.

Lo que se consigue con este estándar es aumentar la productividad del desarrollo de aplicaciones POS, ya que las tareas de comunicación con los periféricos ya han sido implementadas o documentadas facilitando la creación de software de calidad satisfaciendo las necesidades del usuario de mejor manera.

El programador puede dedicar más tiempo a la funcionalidad del sistema y como transformar esos datos recibidos o enviados a los periféricos en información, esta información para un negocio es vital ya que indica índices de rentabilidad, pérdida, gustos de sus clientes entre otras. Con la tecnología OLE puede dividir al software en varias partes para luego ser reutilizado, la reusabilidad reduce los costos de desarrollo y ayuda a enfocar al programador en las nuevas necesidades del usuario.

La interacción del sistema con los periféricos se la realiza a través de métodos o propiedades de estos componentes OLE, lo cual es totalmente transparente para el programador, el software OPOS esta construido principalmente en dos capas:

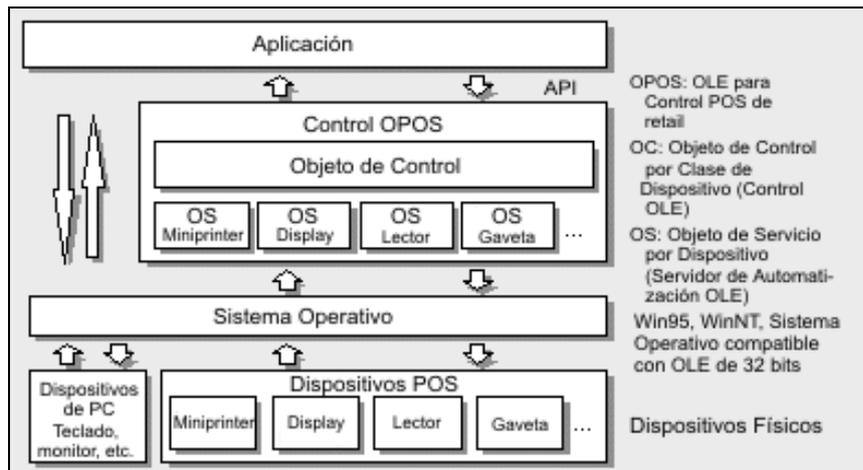


Fig. 2.22 Capas OPOS

La aplicación y los controles crean las dos capas del estándar OPOS, la aplicación invoca a las funciones y los controles son los encargados de comunicarse con los dispositivos.

OPOS es principalmente respaldado por Microsoft, por esta causa este estándar es altamente compatible con la mayoría de versiones de Windows desde Windows 95 hasta Windows XP, en el 2005 se lanzará oficialmente una versión del OPOS para .Net actualmente se encuentra en pruebas esta versión.

Java Pos

Java Pos son varias API's que permiten reducir los tiempos de desarrollo para comunicarse con los periféricos utilizados en los sistemas POS, por medio de estas librerías se puede construir un sistema POS compatible con varias compañías que han creado dispositivos que se acoplen con este estándar.

Las compañías que han colaborado para la elaboración de este estándar son muchas entre las más conocidas tenemos a Sun Microsystems, IBM, NCR, Epson, Fujitsu, Ultimate Technology, Star, Wincor Nixdorf, Home Depot, JC Penney, Kmart, Sears, PCMS Datafit, RCS, Triversity, BearingPoint y 360Commerce. Estas empresas son desde los creadores de hardware, software hasta las grandes cadenas de supermercados que buscan un estándar que genere soluciones escalables y altamente compatibles con todos sus productos y necesidades.

El estándar tiene poco tiempo en el mercado nace gracias a las empresas Sun, IBM, y NCR en el año 1997 se presenta los primeros documentos para que se revise este estándar, en el año 1998 se lanza oficialmente JavaPOS v2.3.2 Standard y luego JavaPOS v1.3 Standard, actualmente se encuentra en la versión JavaPOS v1.8, el crecimiento de este estándar ha sido rápido y se puede encontrar incluso en otros idiomas como el japonés.

Los objetivos de este estándar son:

- Definir una arquitectura para dispositivos POS que sean compatibles con las aplicaciones en Java.
- Definir un conjunto de librerías para los dispositivos POS multiplataforma.

La arquitectura la define en un gráfico mostrando en tres capas, la aplicación POS, las librerías JavaPos y los dispositivos físicos.

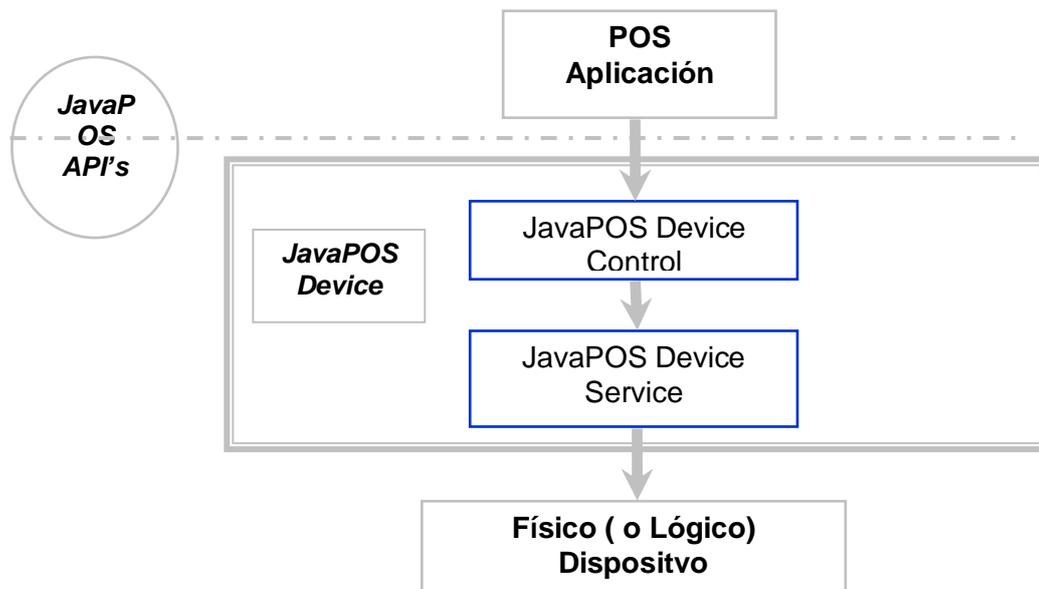


Fig. 2.23 Arquitectura JavaPos

JPOS tiene algunas ventajas significativas que han permitido que este estándar crezca rápidamente:

- Permite reducir los costos de integración a través de múltiples plataformas.
- Un periférico puede ser fácilmente integrado a una aplicación que utilice Java Virtual Machine.

- Interactuar con varios dispositivos POS de distintos fabricantes en sus sistemas.
- Alta capacidad de crear aplicaciones distribuidas para mejorar el soporte y mantenimiento de la aplicación.
- Mejora la confiabilidad del sistema a través de métodos internos que manejan adecuadamente punteros, desbordamiento de memoria y posibles errores de la aplicación.
- La seguridad en JPOS permite comunicación transparente y confidencial en ambientes Intranet o Internet.
- Ayuda a un crecimiento de la aplicación a medida que nuevos dispositivos se integran al estándar.
- Compatibilidad con Linux reduciendo los costos de las terminales a un sistema operativo no propietario.

UnifiedPOS

UnifiedPOS es un estándar encargado de crear las especificaciones de arquitectura para las interfaces de los distintos dispositivos POS. UPOS nace por la NRF National Retail Federation, el estándar es independiente de cualquier sistema operativo y de lenguaje de programación. Tanto JavaPOS como OPOS utilizan como base UPOS para crear sus librerías.

La primera versión de UPOS es la 1.5 y en la actualidad llega a la 1.8; UPOS define las siguientes especificaciones:

- Una arquitectura de aplicación para la utilización de las interfaces de los distintos dispositivos POS.
- Un conjunto de especificaciones para dispositivos POS para la creación y soporte de aplicaciones.
- Documentación describiendo las distintas funciones del dispositivo.
- Terminología UML y diagramas para cada dispositivo, en la cual describen las relaciones entre las clases, interfaces, y objetos del sistema.
- Las bases para crear C++, Java, IDL u otra terminología orientada a objetos para la implementación del diseño de la arquitectura de los dispositivos en UML.
- Características operacionales de los dispositivos.

Este estándar es la base del desarrollo de los sistemas POS pero UPOS no es la encargada de especificar las API's y componentes de software que sean desarrollados por terceros.

Las metas de UPOS son:

- Arquitectura común para todos los dispositivos POS a nivel internacional y sin importar el fabricante, plataforma.
- Estándares de aplicación para las interfaces de los dispositivos en cualquier sistema operativo y lenguaje de programación.
- Reducir los costos de implementación y mejorar el soporte a múltiples plataformas de desarrollo logrando que compartan la misma arquitectura

UPOS busca que los fabricantes innoven en sus productos y que busquen la competencia a nivel de implementación y no a nivel de estándares.

UPOS define una arquitectura multicapa en la cual la aplicación POS interactúa con la parte física o lógica del dispositivo POS a través de la capa de control.

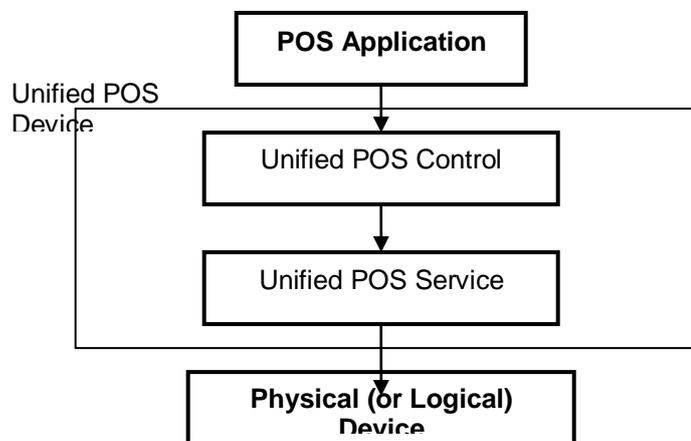


Fig. 2.24 Capas UPOS

La primera capa es la de aplicación, consiste en el sistema que utiliza los distintos dispositivos POS. La siguiente capa es la de control encargándose de proveer de una interfase entre la aplicación y el dispositivo, esta interfase no es gráfica y no es totalmente transparente durante la ejecución del programa.

La capa de control ha sido diseñada de tal forma que la implementación de un dispositivo pueda ser desarrollada por diferentes empresas.

UPOS utiliza la terminología UML para diagramar las especificaciones de cada dispositivo. Se encuentran como describir los paquetes, los tipos de datos, métodos y propiedades de los dispositivos POS.

IXRetail

La asociación ARTS creó IXRetail para implementar el uso de XML para el diseño de esquemas XML para implementar el modelado de datos y lograr una comunicación transparente entre varios sistemas.

El objetivo principal de este estándar es la integración de sistemas a través de datos en XML, esto permite acoplar varios sistemas rápidamente formando un sistema altamente compatible y escalable.

Al describir los datos en XML estos pueden ser integrados a sistemas de e-commerce y lograr comunicación constante con otras empresas y mejorar la cadena de compra, venta y distribución de sus productos.

Lograr sistemas que cooperen entre sí es disminuir los tiempos de ejecución de varios procesos de la empresa, al usar un formato estándar de datos permite a más sistemas unirse y colaborar para construir información valiosa para la empresa mejorando su productividad y reduciendo costos.

El describir los datos en forma de XML ayuda a los desarrolladores a crear modelos en UML permitiendo una documentación apropiada para el diseño y desarrollo.

POSlog

POSlog es un estándar para crear mensajes compatibles con otras aplicaciones, al usar IXRetail se tienen los datos en XML pero estos pueden ser transformados para enviar estos datos a otros sistemas, estos mensajes servirán para integrar estos datos a un sistema ERP, sistemas de análisis de fraudes, sistemas KM (Knowledge manager) y sistemas a los cuales se desee publicar o suscribirse.

Los datos obtenidos de una venta son los más importantes en un sistema POS, estos datos se reflejan en todos los módulos que tenga un sistema corporativo como en finanzas, inventarios y es la base para la toma de decisiones gerenciales.

La venta es el origen de todos los datos del sistema POS, la única forma de mantener el equilibrio en la empresa es que el sistema POS envíe estos datos a los demás sistemas consiguiendo que los productos estén disponibles para los clientes y que no se tenga desperdicio en la mercadería.

POSlog es una excelente solución para integrar a las aplicaciones y mantener comunicación constante entre ellas facilitando la integración sin poner en riesgo la disponibilidad del sistema y la integridad de los datos.

Los beneficios de POSlog son los siguientes:

- Permitir la integración de varios sistemas relacionados con el sistema POS.
- Reducir el impacto de migrar sistemas a nuevas tecnologías.
- Elimina la necesidad de comprar software que transforme los datos.

2.4 HERRAMIENTAS DE ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

2.4.1 Power Designer 9.5

2.4.1.1 Descripción

Power Designer es una herramienta CASE que permite diseñar Bases de datos cliente / servidor y los objetos de una aplicación. La arquitectura de datos permite a los ingenieros diseñar la base de datos desde la parte conceptual y pasar su diseño a una forma física lista para ser utilizada.

El modelo de datos permite realizar ingeniería inversa, mantenimiento de las tablas y documentación con soporte para más de 30 DBMS.

Power Designer modela los procesos con la capacidad de mostrar los flujos de información como se generan.

La herramienta permite realizar trabajos en equipo con el uso de MetaWorks, asigna a cada grupo una tarea específica en el diseño. La pantalla de Power Designer permite un manejo rápido de todos los objetos de los modelos como se muestra en la Fig. 2.25

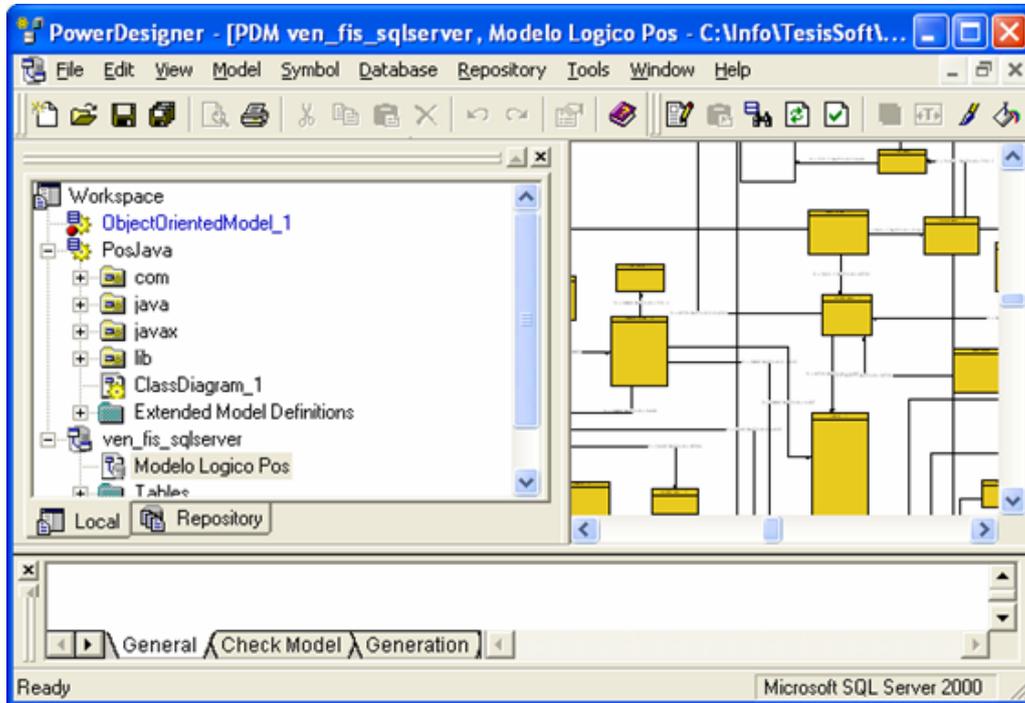


Fig. 2.25 Pantalla de Power Designer 9.5

Power Designer tiene la capacidad de crear prototipos basados en los modelos de datos, estos prototipos pueden generar código en algunos lenguajes conocidos como Power Builder o Visual Basic

2.4.1.2 Características

El programa tiene una interfaz gráfica tipo Windows en donde se trabaja en un entorno de trabajo (Workspace) donde se añaden los distintos diagramas del sistema.

Power Designer tiene los siguientes modelos

- **Modelo Conceptual (CDM)** : Es el modelo lógico de la base de datos, la representación se realiza basándose en el diagrama entidad relación , permite realizar una validación del modelo y

convertir el modelo conceptual en modelo físico , de objetos u otro modelo conceptual.

- **Modelo Físico (PDM):** Permite diseñar la estructura física de una base de datos, el modelo físico trabaja con una base de datos específica logrando diseñar todas las estructuras necesarias para poner en producción el modelo.
- **Modelo de Objetos (ODM):** Describe el sistema utilizando objetos, este modelo utiliza el estándar de modelado UML y cuenta con los siguientes diagramas :
 - Diagrama de clases
 - Diagrama de objetos
 - Diagrama de casos de uso
 - Diagrama de colaboración
 - Diagrama de secuencia
 - Diagrama de estados
 - Diagrama de actividades
 - Diagrama de componentes
 - Diagrama de implementación
- **Modelo de Procesos (BPM):** Es un modelo conceptual que describe la lógica y reglas del negocio, describe por medio de un diagrama los flujos, procesos, relaciones e interacción con otros sistemas.
- **Modelo libre (FEM):** Permite describir a detalle una característica del sistema. Este gráfico ayuda a explicar la arquitectura del sistema.

Power Designer tiene algunas características importantes:

- Compartir objetos entre modelos
- Comparar modelos
- Editar e imprimir reportes en distintos lenguajes
- Automatizar funciones de Power Designer utilizando VBScript

2.4.2 Power Builder 8.0

2.4.2.1 Descripción

Power Builder es un entorno de desarrollo gráfico orientado a objetos. Por medio de Power Builder se puede generar componentes y crear aplicaciones robustas y reutilizables. Power Builder proporciona todas las herramientas necesarias para construir sistemas empresariales como ERP, CRM.

En Power Builder se puede trabajar con las aplicaciones clásicas cliente servidor en la cual se interactúa con una base de datos como backend y el frontend es la aplicación del usuario. El marco de trabajo permite generar ventanas ricas en componentes gráficos como botones, combos, checkbox, listas y crear componentes visuales personalizados.

Se puede generar aplicaciones N capas con esta herramienta. En aplicaciones multicapa por lo general existe una aplicación cliente que realiza llamadas a los distintos componentes publicados para obtener o procesar datos. Un ejemplo práctico es una aplicación cliente que realice peticiones a un componente configurado en EAServer o MTS. El componente del servidor solicita a menudo servicios de una base de datos o de otros componentes del servidor

Los usos de Power Builder pueden también ser orientados a la Web. Se puede crear una nueva aplicación Web para la Intranet o el Internet o migrar una aplicación existente. Se maneja por entornos de trabajo para trabajar con varias aplicaciones al mismo tiempo como se muestra en la figura 2.26

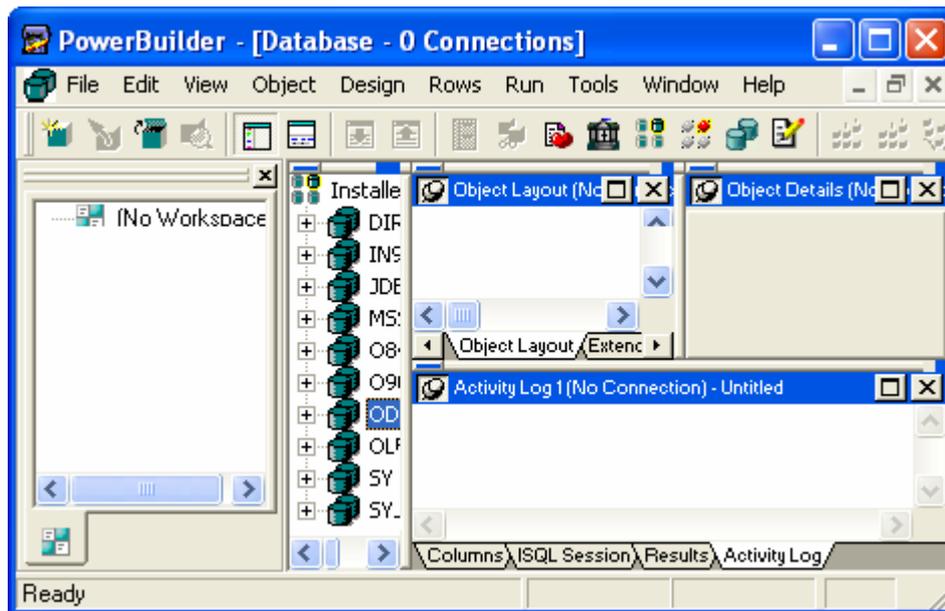


Fig. 2.26 Pantalla de Power Builder 8.0

2.4.2.2 Características

Entre las mejoras de esta versión está el rediseño de la interfaz gráfica, permite crear varias áreas de trabajo logrando trabajar con varias aplicaciones al mismo tiempo. Mejora el manejo de targets ya sean de PowerScript o Web.

Tiene varios asistentes que ayudan al programador a actualizar una aplicación de una versión anterior o a crear una aplicación nueva. Estos asistentes por lo general aparecen al iniciar el Power Builder.

Se han implementados nuevos códigos de ejemplo de ciertas tecnologías de Power Builder para que el programador los utilice en sus aplicaciones.

Tiene facilidades para el desarrollo de aplicaciones Web, permite la creación de los Datawindows para la Web, generación del código html y javascripts para realizar validaciones a nivel de cliente.

Integración rápida con el EAServer, permite la creación de perfiles para conectarse con el servidor de componentes, EJB de cliente, conexiones SSL.

Contienen mejoras en la configuración de perfiles para base de datos, tiene compatibilidad con las bases de datos más utilizadas del mercado utilizando sus propios conectores. Soporta la creación de conexiones Proxy de bases de datos

en el EAServer, esto permite compartir pools de conexiones o accesos a bases de datos remotas reduciendo el tiempo de conexión y manipulación de los datos.

2.4.3 Sun Application Server 8

2.4.3.1 Descripción

Esta plataforma fue diseñada para mejorar la productividad al desarrollar con herramientas que ayuden a un desarrollo rápido de aplicaciones. Es licencia libre para el desarrollo, despliegue, y redistribución.

Sun Java™ System Application Server Platform Edition 8 simplifica las tarea de creación y administración aplicaciones Web. El servidor de aplicaciones proporciona una guía rápida de desarrollo para servicios Web que simplifica el proceso de desarrollo mientras que proporciona oportunidades flexibles de crecimiento. Tiene nuevas características en ejecución especificadas por la plataforma de J2EE 1.4. La interfase Web permite una configuración rápida de las opciones del servidor como se muestra en la Fig. 2.27.

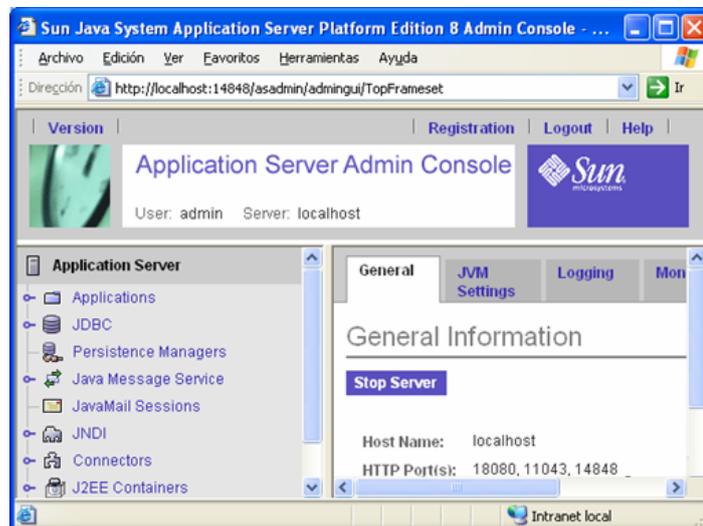


Fig. 2.27 Pantalla de Sun Application Server 8

2.4.3.2 Características

2.4.3.2.1 Soporte J2EE 1.4

La plataforma más completa para los servicios Web en el mercado de hoy. La plataforma J2EE 1.4 soporta WS-I Basic Profile 1.0 y permite servicios Web a ínter operar en base a Web Service Description.

Tiene soporte para el lenguaje (WSDL) y el protocolo (SOAP). Donde sus características incluyen:

- Exposición de aplicaciones J2EE como servicios Web que se basan en el protocolo SOAP y en el HTTP.
- Integración con servicios Web e implementación de Web-service con los servlets de Java y la tecnología de Enterprise Java Beans (EJB).
- Nuevos API's para los servicios Web, el desarrollo, y el manejo de la plataforma JSEE.
- EL API actualizado, incluyendo EJB, el servlet de Java, JavaServer Pages (JSP), y el conector API's de J2EE

2.4.3.2.2 Experiencia, utilidad y productividad

Sun Java System Application Server 8 Platform Edition proporciona herramientas relevantes como las siguientes:

- **Consola de administración:** para administrar y configurar Application Server Platform Edition 8 la interfaz de usuario. Específicamente, se puede instalar la siguiente consola de administración:
 - o Java Database Connectivity (JDBC), JMS, y recursos de conector
 - o Campos de usuario y seguridad
 - o Configuración y afinación de parámetros para contenedores y servicios
 - o Registro y niveles de monitoreo de subsistemas
- **Despliegue de herramientas:** para montar componentes de J2EE en aplicaciones y desplegar esas aplicaciones a la Sun Java System Application Server Platform Edition 8.
- **Verificador de aplicación de J2EE:** Para tareas de verificación de sintaxis de los paquetes

- **Aumento de herramienta:** para la migración y configuración de archivos de la edición Application Server 7 Platform.
- **Herramientas de migración:** para la migración de aplicaciones de BEA, IBM, y de JBoss.
- **Apache Ant 1.5:** para construir aplicaciones J2EE. Application Server Platform Edition 8 también incluye un número de tareas custom Ant que simplifican el despliegue y la administración de archivos Ant build.

Sun Java System Application Server 8 Platform Edition es el primer producto middleware que soporta JavaServer Faces technology. JavaServer Faces technology consiste en un sistema del server-side API's que represente los componentes de la interfaz del usuario que manejan su estado, evento, manipulación, y validación de entrada. El API también define la navegación de la página y apoya la internacionalización y la accesibilidad

Application Server PE 8 tiene disponibles tres maneras en las cuales se despliega aplicaciones:

- **De la consola de administración basada en Web:** se puede desplegar aplicaciones J2EE o ficheros de archivo J2EE.
- **De la interfase de línea de comando (CLI):** El comando asadmin permite el despliegue de aplicaciones de la línea de comando. Para automatizar el despliegue y para construir tareas, se puede utilizar las capacidades integradas Ant.

Ambos, la consola de administración y los CLI soportan la administración remota y la administración de múltiples dominios, cada uno con su propia configuración, almacenaje de aplicaciones, y registros.

- **Con Autodeploy:** Se coloca el archivo en la carpeta de deploy, en donde automáticamente el servidor sube la aplicación.

2.4.4 NetBeans

2.4.4.1 Descripción

El Entorno de Desarrollo Integrado (IDE) NetBeans es un entorno de programación para varios lenguajes, incluyendo a Java y C++. Este desarrollo es de fuente abierto, es decir, se proporciona el código fuente del entorno para que se pueda modificar de acuerdo a ciertos parámetros de licencia.

NetBeans es también una plataforma de ejecución de aplicaciones, es decir, facilita la escritura de aplicaciones Java, proporcionando una serie de servicios comunes, que a su vez están disponibles a través del IDE.

Sun Microsystems fundó el proyecto open source NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Dos proyectos están a disposición: el IDE NetBeans y la Plataforma NetBeans.

El IDE NetBeans es un entorno de desarrollo, una herramienta para programadores para escribir, compilar, corregir errores y para ejecutar programas. Está escrito en Java, pero puede servir de soporte a cualquier otro lenguaje de programación. Existe también un número enorme de módulos para extender el NetBeans IDE. El NetBeans IDE es un producto libre y gratuito sin restricciones de utilización.

También disponible está la plataforma de NetBeans; una fundación modular y extensible usada como una estructura de base para crear aplicaciones de escritorio. Sociedades de desarrollo especializadas proporcionan plug-ins inestimables que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son open source y gratuitos para el uso comercial y no comercial. El código fuente está disponible para la reutilización de acuerdo con la Licencia Pública de Sun (SPL). NetBeans maneja de forma no visual todo el código que se genera en la herramienta como se observa en la Fig. 2.28. Permitiendo una manipulación rápida del código fuente.

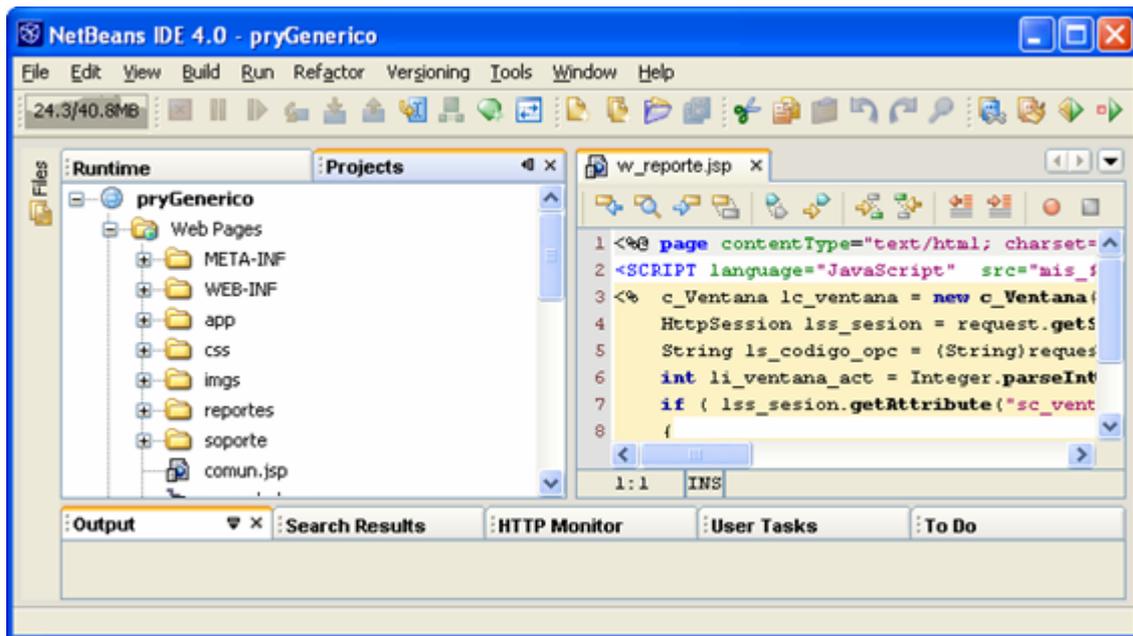


Fig. 2.28 Pantalla de NetBeans

2.5 PARADIGMA DE DESARROLLO

2.5.1 Paradigma Incremental (Punto de Venta tradicional)

El modelo Incremental es parte de los modelos evolutivos, este modelo es una evolución del modelo lineal secuencial e implanta elementos para la construcción de prototipos. El éxito de este paradigma es el dialogo constante con el cliente en cada fase del modelo.

El paradigma se divide en 4 fases: Análisis, Diseño, Código y Prueba.

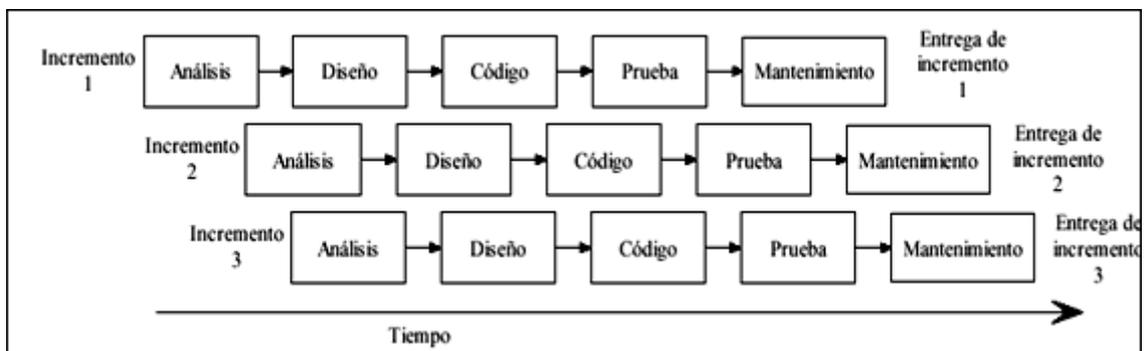


Fig. 2.29 Ciclo de vida paradigma incremental

El desarrollo del software es de manera secuencial logrando que el cliente interactúe en cada una de las fases.

En el primer incremento se determinan los requerimientos esenciales del sistema logrando identificar el núcleo del sistema, en cada incremento se aumentan nuevas funcionalidades según las necesidades del cliente. En cada incremento el producto es software completamente operacional.

El modelo es utilizado cuando se cuenta con un grupo de trabajo reducido. En cada incremento se puede aumentar el personal para desarrollar el proyecto.

Entre las desventajas del modelo se tiene:

- No se puede evaluar el coste total del proyecto,
- No se puede aplicar fácilmente en sistemas transaccionales
- Un error en una fase del sistema es detectado al final del incremento.

Tanto la metodología OMT como la notación UML no dependen del ciclo de desarrollo del software, OMT puede ser aplicado en cada incremento y utilizando UML para el modelamiento de las nuevas características del sistema.

2.5.2 Paradigma en Espiral (Punto de Venta virtual)

El modelo espiral a diferencia de sus predecesores aumenta una fase de análisis de riesgo, el modelo mejora el modelo de cascada priorizando el desarrollo de prototipos esto genera un proceso iterativo en que la participación del cliente es fundamental para reducir el riesgo y cumplir con los objetivos del sistema.

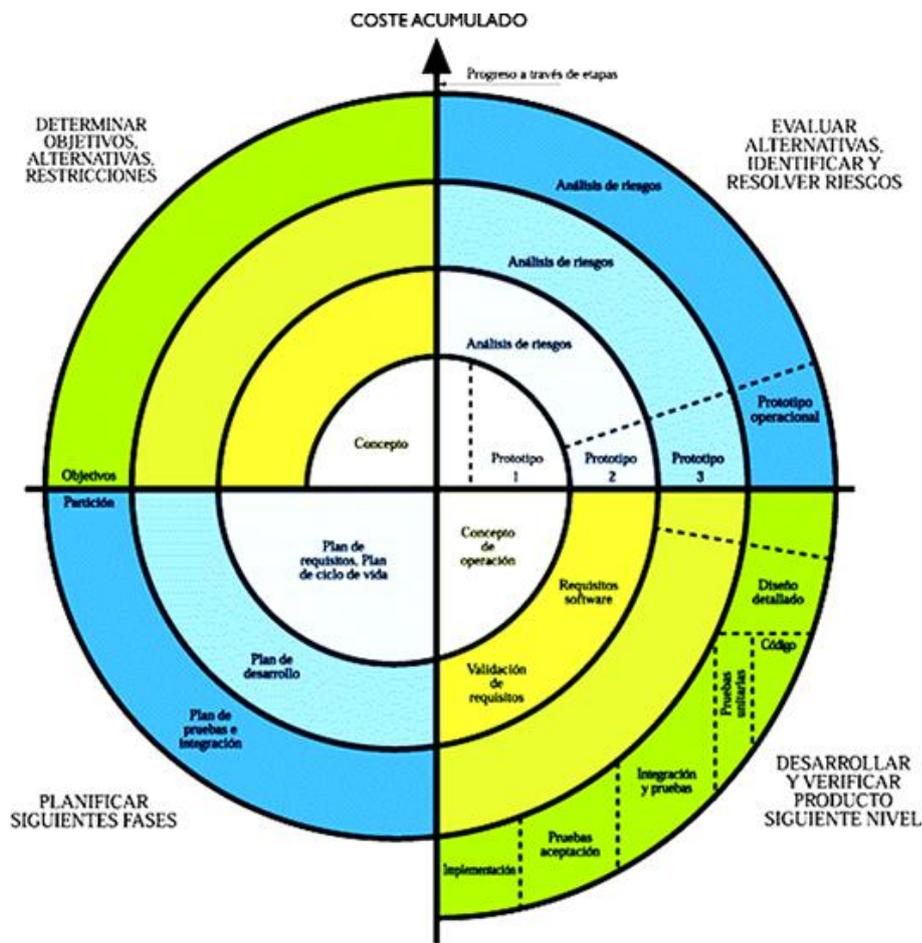


Fig. 2.30 Ciclo de vida paradigma espiral

Las cuatro fases que tiene el modelo son:

1. **Planificación:** Se describen los objetivos de cada fase, alternativas y restricciones. Obtención de los requerimientos del sistema y se planifican las fases.
2. **Análisis de riesgo:** Evaluación de las alternativas, se determinan los riesgos y las actividades para reducir el riesgo. Según los riesgos obtenidos se determina si se debe continuar o no el desarrollo del sistema.
3. **Ingeniería:** Se determinan el modelo de desarrollo. Se genera el prototipo inicial de software. Se utiliza la metodología OMT que soporta ciclos completos de desarrollo de software permitiendo la reutilización

de los objetos. Los objetos serán modelados utilizando el estándar de notación UML.

4. Evaluación del cliente: Evaluación general del proyecto y se determina las acciones para la siguiente fase.

La primera versión del sistema saldrá de la primera vuelta alrededor de la espiral, en esta primera vuelta se obtienen los objetivos, alternativas y restricciones del sistema. En base a estos elementos se identifica el riesgo. En la fase de ingeniería el desarrollo de modelos está orientado no solo extraer los requerimientos del cliente sino también a reducir el riesgo.

En cada ciclo de la espiral la fase de análisis de riesgo determina la factibilidad del proyecto y los riesgos que implica este nuevo ciclo.

El modelo espiral se acerca a la realidad del desarrollo de software, en base a este modelo se puede construir sistemas grandes en donde el riesgo es factor determinante de éxito. La meta es reaccionar de forma oportuna al riesgo y desarrollar un sistema confiable y escalable.

Entre las desventajas del modelo se encuentra:

- El modelo exige que se consideren todos los riesgos técnicos en todas las etapas del proyecto sino se determina correctamente puede generar grandes problemas en todo el sistema.
- Se necesita de personal experto en el manejo del riesgo.

2.6 DEFINICIÓN DE ESTÁNDARES

2.6.1 Estándares de diseño y programación

La empresa Venture maneja algunas normas de programación las cuales sirven de base para realizar un desarrollo exitoso y en beneficio de todos.

- Se utilizará solo programación orientada a objetos
- Siempre se usará el concepto de programación genérica
- Tener en mente la frase “Programar una sola vez bien y luego reutilizar”
- Tener en mente la frase: “No pensar. Solo por esta vez no lo hago genérico por que no tengo tiempo suficiente”
- Tener en mente la frase: “No al uso de variables globales, siempre debo pasar parámetros”
- Tener en mente la frase: “No quemar datos en el código, en su lugar debo usar tablas”
- No al desorden en el código
- Siempre usar comentarios
- No dejar nada abierto y confiar en que el usuario lo digitará bien. Siempre validar
- Respalda diariamente el trabajo
- Correr antivirus 1 vez por semana
- Toda duda importante preguntar al jefe de proyecto
- No desesperar porque algo no sale
- Si hay algún control en Internet o de otra fuente que funciona bien (no inventar el agua tibia, úselo)
- Toda experiencia importante será compartida con el grupo
- Todo código debe ser lo más eficiente posible
- En caso de campos calculados preguntar al jefe de proyecto
- En caso de requerir utilizar una función genérica, avisar al Jefe de Proyecto
- Si necesita programar una función que puede ser genérica, avisar al Jefe de Proyecto
- Centralizar los algoritmos.
- Todos los programas deben estar en un solo sitio (servidor). Solo debe haber copias en las estaciones cuando se desea sacar un respaldo o se desea trabajar en la estación hasta que el programa esté completo y actualizarlo en el servidor

- Centralizar los algoritmos en el programa. Todos los programas deben estar en un solo sitio (clases de negocio). No debe haber copias a lo largo del programa.
- Cuando se desarrolla una función nueva esta primero deberá ser probada en todos los casos especiales, antes de pasar definitivamente a que afecte a todo el código
- En cuanto al código genérico, no se pueden realizar cambios sin la autorización del jefe de proyecto
- Los nombres de las funciones siempre serán en infinitivo
- Cualquier objeto específico que se desarrolle se deberá guardar en la librería que corresponda.
- No usar tipos de letra u otras características que no sean estándares al sistema operativo
- Siempre se trabajará sobre una base de datos real con un importante volumen de datos
- Cuando se realice un cambio al código siempre se deberá revisar la cadena del proceso. Ej.: si se desea cambiar algo en el proceso de facturación. Se revisará desde el ingreso del producto a bodega hasta la presentación de reportes, facturas, estadísticas, involucrada en dicha cadena
- Todos los mensajes el sistema, deberán ser expresados en el título como mensaje (no como error)
- Cada vez que se desarrolle un nuevo reporte, revisar si suma correctamente en la totalidad, por grupos y comparar siempre con otros reportes existentes. Si no existen otros reportes con los cuales comparar se deberá hacer comparaciones manualmente

Los estándares de diseño se basan en tres modelos

- Modelo de procesos
- Modelo de objetos
- Modelo físico

Los modelos además deben seguir los estándares de la empresa que son:

- Cualquier nombre de archivo grabado en disco deberá poseer un máximo de 8 caracteres como longitud del nombre
- Toda palabra será declarada en singular
- Toda palabra será declarada en minúsculas.
- Separación entre palabras (guión bajo) Ej.: nomb_emp (nombre empleado)
- Todo será declarado en español
- No existirán palabras en inglés excepción de las propias del lenguaje de programación

Se debe tener las siguientes consideraciones en el modelo lógico de la base de datos:

Nombre de tablas

- Total : 15 caracteres máximo
- Primer carácter identifica el módulo de origen de la tabla (módulo dueño de la tabla) Ej.: r_empleado (rrhh empleado)
- Si se quiere representar varias palabras se seguirá utilizando (guión bajo) En.: n_asiste_emplea (nomina_asistencia_empleado)
- Tratar de completar los 15 caracteres (con palabras completas de ser necesario o balancear en # de caracteres en cada palabra. Si hay desbalance, la(s) última palabra tendrá(n) mayor cantidad de dígitos con excepción de que la última palabra por naturaleza sea corta
- Para el caso de tablas similares en distintos módulos (Ej.: unidad de productos de bodegas, unidades de productos en poscosecha) siempre se usarán mínimo 2 palabras> la última palabra tendrá relación con el módulo (Ej. unidad postsospecha -> e_unidad_postco)

Campos de tablas

- Total: 15 caracteres máximo (estándar antiguo), 16 caracteres máximo (estándar nuevo)

- 11 primeros caracteres son el nombre del campo + guión bajo + 3 caracteres de código de tabla.
 - Ej.: (codigo_aem) codigo_asistenciaempleado (estándar antiguo)
 - Ej.: (codigo_naem) codigo_asistenciaempleado (estándar nuevo)
 - Ej.: razon_socia_emr (razon_social_empresa) (estándar antiguo)
 - Ej.: razon_socia_semr (razon_social_empresa) (estándar nuevo)
- Código de tabla (si son 3 palabras el primer carácter de cada una, si son 2 palabras primer carácter de la primera palabra y 2 de la segunda, si es una los 3 primeros caracteres, si por alguna razón se repite se usará el siguiente carácter que no se repita de la primera palabra).
- En el estándar nuevo se siguen las mismas reglas pero se antecede al código de la tabla la letra del módulo correspondiente
- Todo campo que represente clave primaria se llamará código y poseerá la palabra código + guión bajo + código de tabla
 - Ej.: codigo_emp (codigo_empleado) estándar antiguo
 - Ej.: codigo_remp (codigo_empleado) estándar nuevo
- Tratar de completar los 15 caracteres (estándar antiguo), 16 caracteres (estándar nuevo) con palabras completas de ser necesario o balancear el # de caracteres en cada palabra
- Si se requieren usar números, empezar desde el 1: Ej. Telefono1, telefono2

Estándar de longitud de campos

Campos código

Longitud mínima para los códigos 2 (sin sucursal)

Campos nombre de personas (Apellidos + Nombre)

60 Caracteres

Campos nombre de personas (Apellidos y nombre separados)

30 Caracteres

Campos nombres estándar (Normal)

A (30 caracteres)

Campos nombres estándar (largos)

A (40 caracteres)

Direcciones

A (100 caracteres)

Teléfonos

A (15 caracteres)

Email

A (60 caracteres)

Web

A (60 caracteres)

Moneda (2 decimales)

Numérico 12,2

Moneda (alta precisión)

Numérico 22,15

Observaciones (normal)

A (100 caracteres)

Observaciones (grandes)

A (250 caracteres)

Fecha

Solo date

Porcentajes

N5,2

Árboles (NORMAL)

Código 28

Árboles (GRANDE)

Código y padre 53

Cualquier otro tipo de longitud que pueda darse, de acuerdo al tipo de negocio cliente

Claves (Índices)

Utilizar el estándar de Power Designer

- pk_seguido del nombre de la tabla (clave primaria)
- fk_seguido del nombre de la tabla (clave foránea 1)
- fk_seguido del nombre de la tabla (clave foránea 2)

Orden de campos en las tablas

- 1 Campo primario
- 2 Clave foránea
- 3 Demás campos

2.6.2 Estándares de codificación

2.6.2.1 Normas de programación específicas

Los botones de las ventanas que las contengan deben tener su respectiva descripción.

Todos los dropdown o dataobject deben tener codigo_com (aun cuando no se requiera)

Las variables del sistema deberán ser configuradas con el criterio de (1 activa, 0 desactiva), (1 precio promedio, 2 fifo, 3 lifo). A excepción de los casos en los que se requiera valores constantes donde se colocará el valor exacto (Ej.: 01 tipo bunch sólido). Siempre se deberá incluir una descripción con los valores posibles en cada caso

El orden de recuperación de argumentos es el siguiente: Blob, string, number, date, time, dentro de cada tipo se considerarán primero el tipo estándar y segundo el array.

Todas las fechas desplegadas al usuario tanto en reportes como en pantalla, serán día mes año. En el caso de usar código que concatene fechas utilizar la función de casting: String (fecha, gs_formato_fecha).

No utilizar iconos que no sean propios de la herramienta

No utilizar este tipo de sintaxis if (a=b) then a=1

En su lugar utilizar if (a=b) then

A=1

end if

Cuando no se quiere heredar no solo se debe quitar el ancestro script del menú, sino también un comentario al inicio que indique //no hereda

Cuando en el código se requiera definir un valor constante Ej.: el campo codigo_cpr posee un ancho de 9 caracteres. En ocasiones es necesario hacer relación a dicho valor (9). No se puede quemar el 9 en código, en su lugar se deberá utilizar la función del datawindow padre, datastore padre, u otra que calcule el ancho respectivo

Cuando se desee tomar datos de una columna que proviene de una tabla, a través de datastore, datawindow u otro medio. La variable temporal donde se almacenará, será el estándar de power builder + guión bajo + nombre de la columna en la tabla

2.6.2.2 Comentarios

Para que el código pueda ser modificado por otro programador existen varios aspectos de la codificación que debe ser comentado con algunas especificaciones, el comentario en Power Builder se lo utiliza con los símbolos `/* COMENTARIOS */`, durante la programación se tendrán presentes los siguientes estándares:

Funciones

Ej.:

```
/*Actualizado: 2002/11/11
```

Descripción: Recupera estado de cuenta por factura para el cliente Florecal, para lo cual recupera solo las facturas con saldo pendiente y las retorna a través de un datastore

Parámetros: rango de fechas, cliente (solo uno ya que es estado de cuenta individual)

Situaciones especiales: No usar esta función en el caso de otros clientes ya que la forma de cálculo es diferente en relación al manejo de créditos*/

Datawindows

Todo Datawindow (reportes, procesos) poseerá en propiedades lo siguiente:

Ej.:

Descripción: Recupera facturas solo con saldos pendientes para un cliente

Parámetros: rango de fechas, cliente (uno solo ya que no usa in en el select)

Situaciones especiales: La instrucción SQL es de tipo anidada y para desplegar los datos se requiere que se encuentre oculta la banda del detalle

Clases de negocio

Toda clase de negocio poseerá en propiedades lo siguiente:

Descripción: posee la funcionalidad para el manejo de facturas como: cálculos de facturas, reportes acerca de facturas, impresión de facturas

Situaciones especiales: No se la debe usar en el caso de otros reportes que provienen de pedidos

Grupos de código Importantes

Ej.:

/* Las siguientes líneas recorren un datastore realizando una actualización de la columna estado en 1 para fecha mayores a la variable Id_fecha*/

Otras consideraciones son las siguientes:

- Uso de sangrías en forma ordenada y clara (tabulador 3 espacios)
- Utilización de líneas en blanco para separar grupos de líneas de código que cumplen un objetivo específico
- Debe existir una tabla de configuración que considere el código quemado en el programa (s_variab_sistem)
- No depender de una marca de base de datos. El sistema funciona en Sybase, SQL Server, Informix y Oracle. Cualquier sentencia especial que se desee utilizar se la deberá probar en todas las bases de datos

2.6.2.3 Variables globales, instancia y locales (privadas o públicas)

Estándar de power builder + guión bajo + nombre del campo en la tabla

Ej.: li_i contador de un lazo for

Ej.: li_total_empleado

Las variables deben ser declaradas en orden alfabético tanto vertical (por tipo de dato), como horizontal (nombre de la variable) Ej:

Blob b_1

integer li_1,li_2

Sring ls_1,ls_2

El nombre de la variable tendrá como máximo 20 caracteres adicionales del estándar de PB

Si no tiene relación con un campo de la tabla: Estándar de Power builder+guión bajo+ estándar de campos de tablas de acuerdo al nombre que se desea colocar Ej.: li_total_fila_empleado

Siempre utilizar nombres que tengan relación al objetivo para el cual se creó la variable.

2.6.2.4 Nombres de controles

Nombres de controles por defecto: dw_1, dw_2, etc.

Para otros casos: Estándar de Power builder+guión bajo+estándar de campos de tablas (sin código de tabla). Ej.:

d_rol_pago (datawindow para reporte u otra actividad)

d_dddw_departamento(dropdown)

r_departamento (dropdown genérico en la librería dtemp coincide con los nombres de las tablas).

2.6.3 Diseño de interfaces de usuario

2.6.3.1 Especificaciones generales

- De acuerdo programa genérico
- Colores por defecto del panel de control y de Power Builder
- Siempre utilizar grids para pantallas
- Siempre utilizar grids en reportes, excepto en los reportes donde sea imposible hacerlo. Ej.: documentos preimpresos

2.6.3.2 Tipos de letra

Siempre utilizar Arial, a excepción donde se autorice la utilización de otro tipo. Ej.: códigos de barra

2.6.3.3 Reportes

- Las ventanas del tipo w_reporte solo pueden contener if simples (no usar else if), y todas declaraciones de variables se realizaran al inicio, donde se deben respetar las sangrías
- Los nombres de los reportes cuando el usuario los selecciona deberán ser palabras genéricas no relacionadas al cliente. A excepción de documentos preimpresos.
Ej.: retención CCI

Ej.: Esquina superior izquierda logotipo y a continuación (abajo) datos de la empresa (dirección, teléfono, fax, email, página web, ruc en documentos)

- Para el genérico se debe aumentar un StaticText con nombre t_dato_empresa Width 869, Height 216 , Arial de 7.
- Siempre utilizar el tipo de datawindow grid para los reportes ya que ayuda al usuario a: mover, cambiar tamaños de columnas, exportar datos. A excepción de los casos donde sea imposible utilizar Ej.: Facturas preimpresas, retenciones preimpresas
- Título del reporte al centro superior del reporte tamaño 14 bold
- Subtítulo del reporte 12 Bold.
- Criterios 10 Bold Itálico
- Números de página alineación superior derecha tamaño 10
- Grupos y subgrupos bold 10 con sangrías
- Detalle 10
- Totales y subtotales tamaño 10 bold
- fecha y hora bajo los datos de la empresa tamaño 10
- Nombre del reporte tamaño 8 caracteres

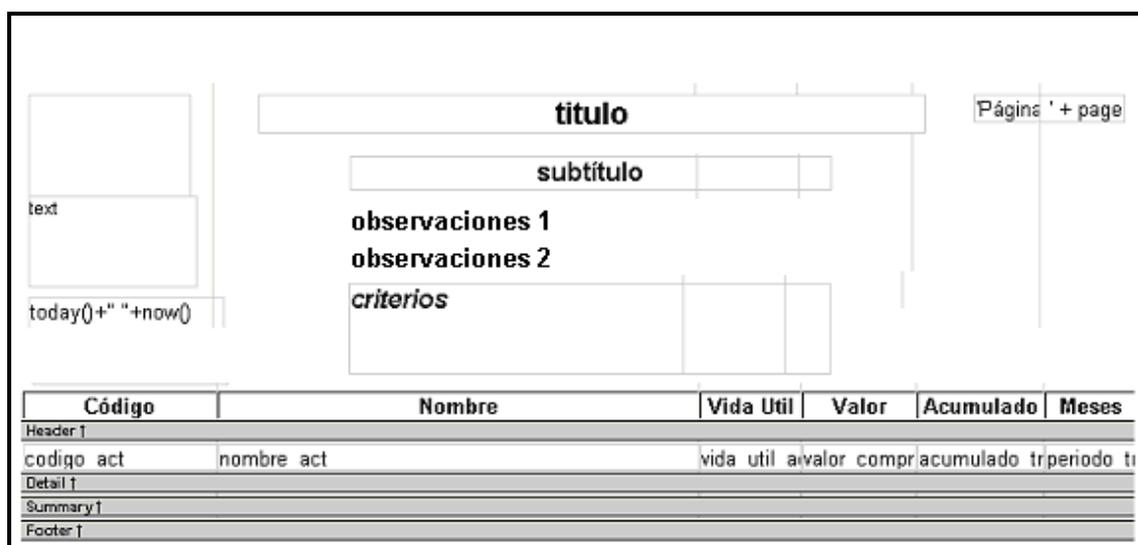


Fig. 2.31 Diseño del reporte

2.6.3.4 Menú de opciones

Las opciones de los menús se colocarán bajo su padre (nombre del módulo) sin ninguna topología en particular.

2.6.3.5 Estándares para desarrollo de ayudas

El archivo del proyecto posee el nombre manual.hpp

3 zonas a describir

Elementos

Descripción de elementos en el siguiente orden:

- Datawindows (ventanas)
- Otros controles importantes (botones, check box, etc.). De izquierda a derecha y de arriba hacia abajo de acuerdo a la posición en pantalla. Sin imágenes

Procesos

Procesos importantes que se realicen en dicha pantalla. Algunas pantallas no poseen este tipo de procesos como las pantallas de parámetros

Formato de importación

Vinculo hacia el archivo en Excel respectivo para que el usuario pueda grabar en su disco y modificar. Los archivos susceptibles de importación son:

activo.xls

articulo.xls

cen_cost.xls

empleado.xls

pla_cuen.xls

proveedo.xls

CAPÍTULO III

3 REQUERIMIENTOS

Para el desarrollo de este capítulo se va aplicar el estándar de requerimientos IEEE 830, previo a la recolección de requerimientos se diagrama y describe los procesos básicos que cumplirá el módulo en sus dos interfaces.

3.1 PROCESOS, PROCEDIMIENTOS Y ACTIVIDADES

Para determinar los procesos que cumple cualquier actividad se debe saber como se mueve la información y como va cambiado, y esto se inicia con el diagrama de flujo de información.

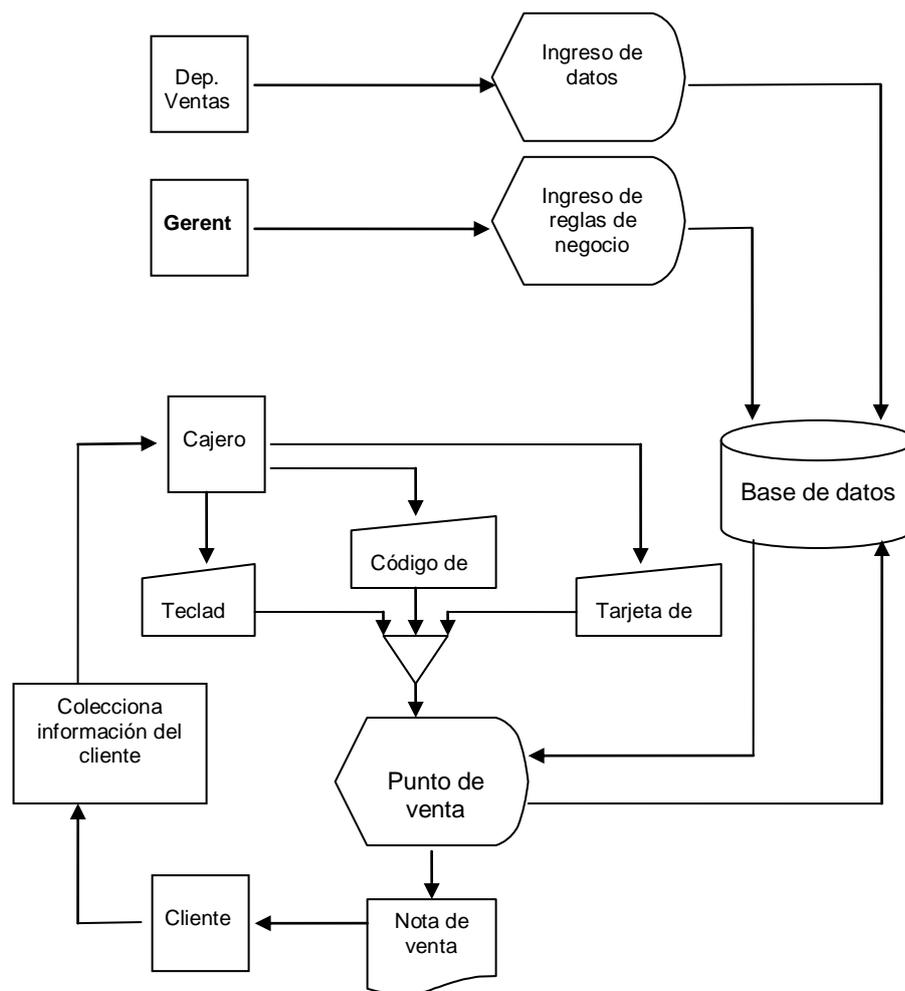


Fig. 3.1 Flujo de información

3.1.1 Configuración del Módulo

La configuración generalmente consta de pantallas simples ya sea a nivel de Intranet como Internet.

3.1.1.1 Proceso: Ingreso de datos (intranet)

- **Requiere:** Datos necesarios para cada ventana.
- **Procedimiento:** Del árbol de opciones se selecciona la que se desee configurar haciendo clic con el mouse, la ventana se abrirá y desplegará la información contenida en datawindows, la forma de ingreso es genérica, es decir que el procedimiento es el mismo para todas y es el siguiente: Para insertar un registro se puede ir al menú principal o a la barra de herramientas o al menú flotante con el botón derecho del mouse o con el comando control+ins. Para editar solo se debe colocar con el mouse en el ítem que se desea editar, aunque algunos pueden estar protegidos o pueden tener dropdowns para facilitar la edición. Para eliminar se debe posicionar con el mouse sobre el registro que se desea borrar y seleccionar eliminar en el menú principal o en la barra de herramientas o en el menú flotante o con el comando control+supr. Al salir de la ventana si ha realizado cambios se le preguntará si desea guardar los cambios, pero también se puede guardar antes con la opción de guardar localizada en los lugares del insertar y eliminar.

3.1.1.2 Proceso: Ingreso de datos (Internet)

- **Requiere:** Datos necesarios para cada ventana.
- **Procedimiento:** Del árbol de opciones se selecciona la que se desee configurar haciendo clic con el mouse, se desplegará la información contenida en tablas, la forma de ingreso es genérica, es decir que el procedimiento es el mismo para todas y es el siguiente: Para insertar un registro se edita la última fila que es la que se va a insertar al hace clic en el botón insertar. Para editar solo se debe hacer clic en la parte derecha de cada registro, en el símbolo de edición, esta acción cambia la pantalla dejando solo los registros a editar habilitados. Para eliminar, se debe seleccionar los registros que se deseen borrar haciendo clic en un check box que se encuentra en la parte derecha de cada registro, para finalmente

hacer clic en el botón guardar para que se apliquen las acciones de inserción, eliminación y edición.

3.1.2 Venta por Intranet e Internet

Estos procesos nos permiten realizar una venta a través de una caja que funcione con el sistema, específicamente con el módulo de puntos de venta. La diferencia entre la parte en internet es que el punto de venta funcionará a través de una página Web y el pago con tarjetas de crédito se realizará a través de PayPal.

3.1.2.1 Proceso: Ingreso de cliente

- **Requiere:** Datos del cliente.
- **Procedimiento:** Para llevar un registro de clientes o reconocer clientes especiales, es necesario que se apliquen algunas reglas de negocio, se ingresa el nombre del cliente, si el cliente es nuevo, el cajero tiene la decisión de ingresarlo o no en la base de datos. Si ya está en el sistema, el cajero lo seleccionará de una lista para que se ejecuten las reglas de negocio que dependen del cliente.

3.1.2.2 Proceso: Ingreso de artículos que el cliente desea comprar

- **Requiere:** El código de barras del artículo, cantidad o peso (casos especiales) y nombre del artículo (casos especiales).
- **Procedimiento:** El proceso empieza cuando llega el cliente con los artículos que desea comprar. El cajero toma el artículo y lo va ingresando en el sistema mediante el código de barras. Existen algunas variantes así como cuando no puede leer el código de barras o simplemente el artículo no lo tiene, para esto el sistema desplegará la lista de artículos para que el cajero al digitar el nombre del artículo, lo encuentre en la lista y lo seleccione. Otro caso especial es cuando existen varios artículos iguales, para agilizar la venta puede contar cuantos hay e ingresar el número y multiplicar por la cantidad del producto.

3.1.2.3 Proceso: Selección de forma de pago

- **Requiere:** Dedicación del cliente.
- **Procedimiento:** El cajero presenta al cliente todas las formas de pago que puede realizar, entonces dependiendo de su decisión el procedimiento cambiará. Si paga en efectivo se ingresará el dinero dado por el cliente, entonces el sistema presenta la resta para poder dar el cambio. En el caso de que sea pago directo con tarjeta de crédito, hay que ingresar la tarjeta en el lector, mandar la información de la compra y forma de pago, la entidad que preste el servicio de tarjetas de crédito retornará la aprobación de la compra en conjunto con otros datos, la compra continuará dependiendo de la aprobación. Si es un crédito propio, entonces el cliente tiene que elegir los plazos disponibles para los pagos.

3.1.2.4 Proceso: Ejecución de políticas de venta

- **Requiere:** Ingreso de todos los artículos, cliente y forma de pago.
- **Procedimiento:** Las reglas de negocio se ejecutan al tener estas tres variantes, modificando el precio final o por artículo; también se pueden presentar descuentos o promociones.

3.1.2.5 Proceso: Comprobante de venta

- **Requiere:** Que la compra sea finalizada
- **Procedimiento:** En el momento que se cierra la venta, se ha realizado el pago o se a puesto en el sistema como se va a pagar, entonces para finalizar la venta se emite un comprobante, que en la mayoría de casos será la nota de venta, aunque en ciertos casos esta será reemplazada por una factura o por un especie de boucher.

3.2 ESPECIFICACIÓN DE REQUERIMIENTOS INFORMÁTICOS

Después de una teoría rápida sobre los requerimientos de un software se puede empezar con los requerimientos para este sistema, siguiendo el orden que sugiere el estándar IEEE 830.

3.2.1 Introducción

Los requerimientos de este software no solo se basan en la empresa Venture, ya que los usuarios reales del programa van a ser los clientes de Venture así que la programación debe estar de acuerdo a las necesidades de Venture como filtro de las necesidades de sus clientes y eso será lo que se especificará a continuación.

3.2.2 Objetivos

El objetivo principal de esta especificación de requerimientos es conocer las necesidades de Venture como empresa, Venture como ERP y en especial las necesidades de los clientes de venture que son los que van a utilizar este módulo.

Otro de los objetivos es mantener los estándares con los que Venture se rige para programar y presentar su sistema ante los usuarios.

3.2.3 Alcance

Este módulo se desenvuelve en dos áreas bien marcadas, el que trabaja en Intranet que estará en un punto de venta físico, es decir en el momento mismo y siendo parte del proceso de vender. Por otra parte el módulo desarrollado para Internet se convertirá en el mostrador, vendedor y cajero para el cliente que desee realizar una compra por Internet.

La funcionalidad del software a realizar debe basarse en los siguientes puntos:

- Adaptación a las reglas de negocio que tenga cada empresa en la parte de puntos de venta.
- Uso de periféricos tales como lector de código de barras y en especial del lector de tarjetas de crédito.
- Funcionalidad comprobada con las bases utilizadas por Venture.
- Su interfaz será la fusión entre los estándares de interfaz que exige Venture y las necesidades del usuario final.

Este software esta limitado en los siguientes aspectos:

- Solo es un prototipo por lo que solo se relacionará con una parte pequeña del ERP Venture.
- Para que la parte de Internet funcione perfectamente, la base de datos debe actualizarse en forma constante para tener un stock más real cada vez que se realice una compra.

3.2.4 Definiciones de acrónimos y abreviaturas

3.2.4.1 Acrónimos

ERP: Planificación de recursos empresariales (Enterprise Resource Planning). Es un software de gestión empresarial diseñada para cubrir todas las áreas funcionales de una empresa. Se trata de un conjunto o paquete de aplicaciones, cada uno diseñado para cubrir las exigencias de cada área funcional de una empresa, de tal manera que crea un flujo de trabajo (Workflow) entre los distintos usuarios.

Un ERP puede estar compuesto por un determinado número de módulos funcionales. Estos se pueden catalogar principalmente en tres tipos:

- Módulos básicos: suelen ser módulos obligatorios a adquirir
- Módulos opcionales: módulos no obligatorios. Incorporan nuevas funcionalidades al paquete ERP.
- Módulos verticales: módulos opcionales diseñados específicamente para resolver las funcionalidades de un sector específico

3.2.4.2 Abreviaturas

- **CTS:** Component Transaction Server
- **ERP:** Enterprise Resources Planning.
- **Kbps:** Kilo bits por segundo.
- **MB:** Mega byte.
- **Mbps:** Mega bits por segundo.

- **MDI:** Interfase con múltiples documentos.
- **MTS:** Microsoft Transaction Sever.
- **OMT:** Object Modeling Technique.
- **POS:** Point Of Sale.
- **UML:** Unified Modeling Language.

3.2.5 Descripción general

Este documento será una guía de principio a fin sobre el desarrollo y funcionamiento del módulo de ventas, guía que ayudará a un desarrollo planificado y ordenado manteniendo los objetivos claros a igual que las limitaciones, pero esta guía no solo es para personas que tengan conocimientos técnicos, ya que tiene una explicación comprensible de todos los términos técnicos o no muy comunes, llevándoles a una comprensión global de cómo fue concebido y las necesidades que orientaron el desarrollo de este módulo.

3.2.6 Perspectiva del producto

3.2.6.1 Diagramas de procesos de negocio

Después de saber todo el movimiento de la información y los cambios que ha ido sufriendo, se puede realizar un diagrama, determinando entradas, salidas y los procesos en sí. A continuación se presenta el diagrama de procesos de negocio para la parte de intranet.

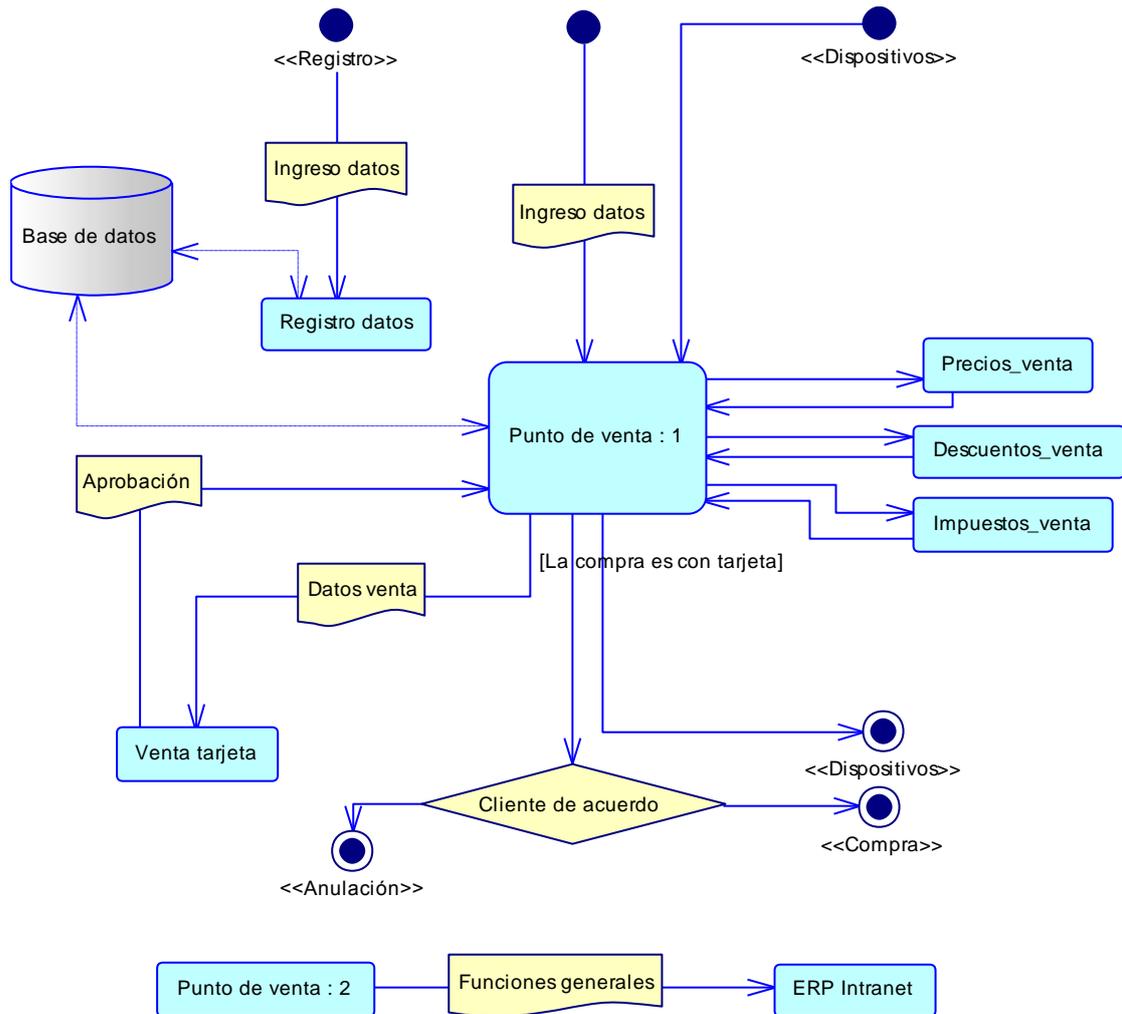


Fig. 3.2 Procesos de negocio (intranet)

Los procesos de negocio que se cumplen a nivel intranet e Internet varían en ciertos aspectos por el mismo hecho de ser interfaces totalmente diferentes, por esta razón a continuación se presenta un diagrama exclusivo para la modalidad Internet.

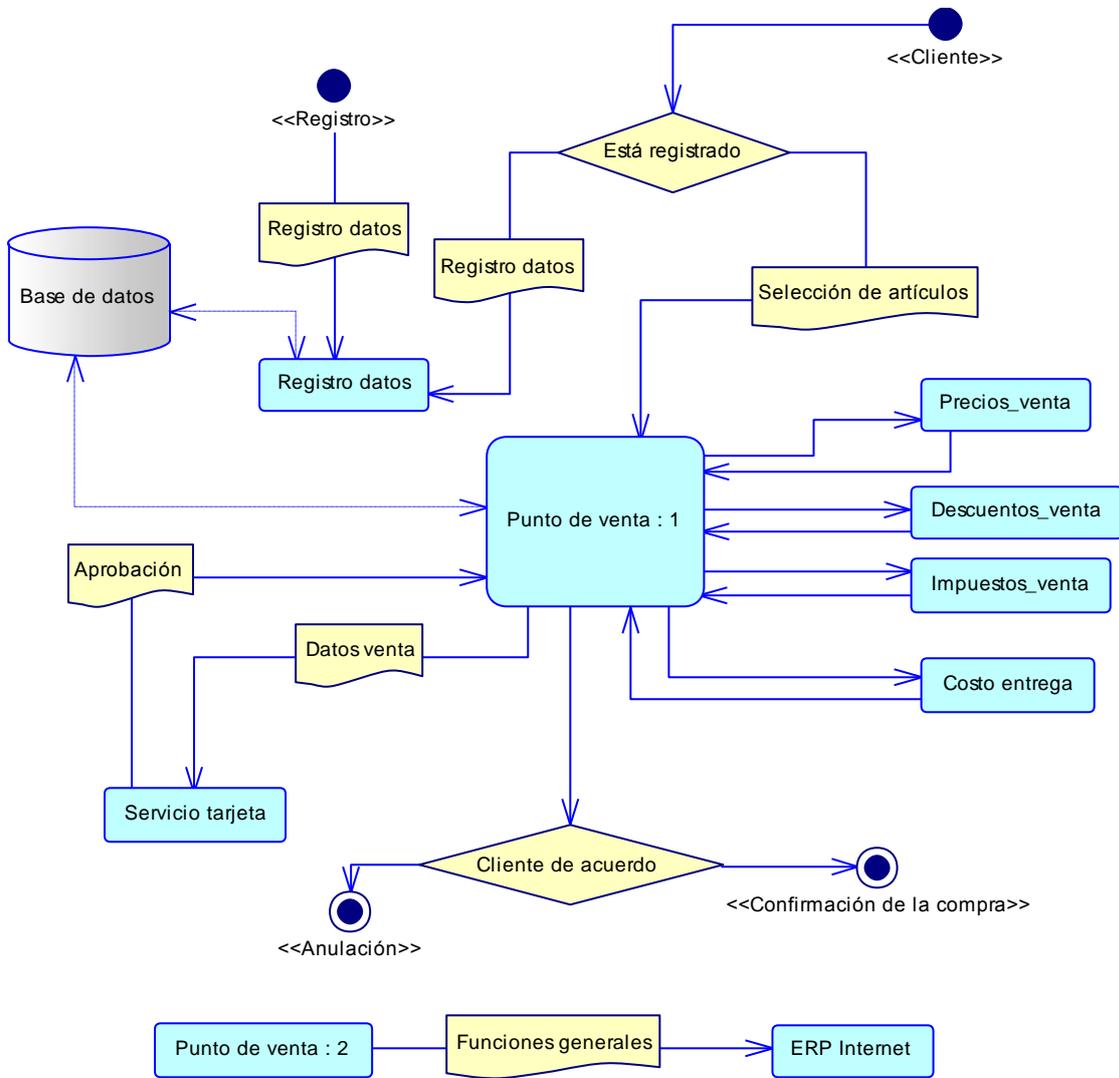


Fig. 3.3 Procesos de negocio (Internet)

3.2.6.2 Entidades externas

Departamento de ventas

Es el personal que se encarga de ingresar los datos para que pueda funcionar el punto de venta, su aporte principal son las reglas de negocio para dar los precios y descuentos para los artículos.

Vendedor

Es la persona que se encuentra en la caja para la parte de Intranet o frente a un computador en la parte de Internet, para el primer caso se registra la compra y para el segundo caso se aprueba la compra registrada por el cliente.

Cliente

El cliente se comporta de distinta manera según la modalidad, en Intranet simplemente es un espectador del sistema, ya que el registro de la compra o cotización es hecha por el vendedor; para la modalidad sobre Internet el cliente trabaja directamente con el sistema, haciendo las veces de cliente y vendedor en el momento de registrar la compra, la cotización o el pedido.

Entidades que manejan las tarjetas de crédito

Son entidades que a través de una red de comunicaciones y de dispositivos lectores de tarjetas de crédito, permiten el uso de las mismas en un punto de venta.

ERP (intranet / Internet)

Se lo puede tomar como una entidad externa que contiene todas las funciones generales, las cuales son heredadas al módulo de Puntos de venta.

3.2.6.3 Descripción general del hardware

El software desarrollado para Intranet será parte del ERP, por lo que los requerimientos del módulo de puntos de venta serán los mismos que los de todo el sistema. Generalmente se utiliza un servidor para la base de datos y como puesto de trabajo se requiere como base una computadora Pentium 3 de 1HGz, 500MB de memoria RAM y una capacidad de 100MB en disco. Además de un

computador se usarán dispositivos como el lector de código de barras y una impresora de notas de venta.

Por otra parte el software desarrollado para Internet se tiene que instalar en un host con cualidades de un servidor de aplicaciones y con una conexión a Internet mayor a 128 kbps.

Funciones del módulo

La principal función de este módulo es automatizar el proceso que cumple un punto de venta realizando las siguientes funciones:

- Tomando la información ingresada por el teclado y el lector de código de barras; recalculando automáticamente el precio, descuento y adicionales de los artículos; registrando cualquier forma de pago con los datos necesarios para finalmente imprimir la nota de venta, el boucher de ser una venta con tarjeta de crédito y registrando todo el movimiento en la base de datos.

En el caso del módulo para Internet, este se presentará a través de varias páginas Web que cumplirán con las siguientes funciones:

- Autenticación del cliente
- Registro del cliente
- Presentación de los artículos
- Registro de los artículos seleccionados por el cliente
- Calculando del precio, descuento, adicionales y precio de envío de los artículos.
- Paso de la venta a la entidad con la que se maneje la tarjeta de crédito
- Impresión del comprobante de la venta.

3.2.7 Características del usuario

El usuario no necesariamente tiene que conocer de computación, simplemente es la persona que registra las compras en un local de venta directa, que tiene conocimientos del trabajo en una caja o como en el caso de la modalidad de

Internet se requiere que sepa navegar en Internet y que posea una tarjeta de crédito válida para estas transacciones.

3.2.8 Limitaciones generales

En la parte de Intranet tiene las siguientes limitaciones:

- Solo funciona en conjunto con el sistema.
- Tiene que estar conectada a la red para poder hacer transacciones y consultas.
- La utilización del servicio para el uso de las tarjetas de crédito presenta varias limitaciones con respecto a la conexión, el servicio prestado por la entidad y los permisos por parte de las entidades bancarias o financieras que emiten las tarjetas de crédito.

En la parte de Internet tiene las siguientes limitaciones:

- Depende de la alta disponibilidad de ancho de banda del cliente para que la aplicación funcione correctamente y en forma ágil.
- Para las transacciones electrónicas depende del servicio prestado por Paypal.

3.2.9 Supuestos y dependencias

- El sistema depende de la comunicación con el servidor de la base de datos, por esto hay que revisar bien las protecciones o firewalls que tenga el servidor o el cliente.
- Para la parte de Intranet e Internet dependerán de su genérico respectivamente, es decir, que cualquier cambio que se haga en este afectará también a este módulo de puntos de venta.
- Este software trabajará como parte del ERP Venture, por lo que el diseño, funcionalidad e interfaces dependen mucho del ERP.

3.2.10 Requisitos funcionales

Los requisitos funcionales se muestran a través de las actividades que va cumplir de este módulo.

A continua se presentan cuadros con las distintas funciones y una descripción tanto en los datos que manejan como la funcionalidad que cumplen.

Función: Registro de vendedores		
Descripción	Datos:	El árbol describe los grupos de vendedores y los vendedores. El datawindow derecho superior presenta y permite editar los datos de los vendedores. El datawindow derecho inferior despliega y permite editar los cupos de venta de cada vendedor.
	Función:	Esta ventana permite hacer transacciones con la tabla de los vendedores además de organizarlos por grupos o padres.

Tabla 3.1 Requerimiento ventana vendedores

Función: Registro de entidades de pago		
Descripción	Datos:	El datawindow permite modificar la tabla de entidades de pago que son el intermediario para los pagos.
	Función:	Realizar transacciones con la tabla de entidades de pago y se puede asignar un cliente a una entidad.

Tabla 3.2 Requerimiento ventana entidades de pago

Función: Registro de agencias de distribución		
Descripción	Datos:	El datawindow permite modificar la tabla de las agencias, que son las entidades que se encargan de transportar los productos vendidos hacia los clientes.
	Función:	Realizar transacciones con la tabla de agencias de distribución.

Tabla 3.3 Requerimiento ventana agencias de distribución

Función: Registro de jerarquía de clientes		
Descripción	Datos:	Con los clientes ya creados esta ventana juega con el valor del padre, dándoles a los clientes jerarquías.
	Función:	Cambia el valor del padre del registro del cliente, lo que permite generar una jerarquía entre los clientes.

Tabla 3.4 Requerimiento ventana jerarquía de clientes

Función: Registro de adicionales		
Descripción	Datos:	Esta ventana maneja las transacciones de la tabla de adicionales, que son los valores adicionales al valor total, pudiendo ser positivos o negativos.
	Función:	Realizar transacciones con la tabla de adicionales.

Tabla 3.5 Requerimiento ventana adicionales

Función: Registro de estado de cotización		
Descripción	Datos:	Maneja la tabla deL estado de cotización, que es la que indica si una cotización está anulada o en función.
	Función:	Realizar transacciones con la tabla de estado de cotización.

Tabla 3.6 Requerimiento ventana estado de cotización

Función: Registro de estado de pedido		
Descripción	Datos:	Maneja la tabla de estado del pedido, que es la que indica si un pedido está anulado, en proceso o facturado.
	Función:	Realizar transacciones con la tabla de estado pedido.

Tabla 3.7 Requerimiento ventana estado de pedido

Función: Registro de estado de factura		
Descripción	Datos:	Maneja la tabla de estado de cotización, indica si una factura está anulada o normal.
	Función:	Realizar transacciones con la tabla de estado de factura.

Tabla 3.8 Requerimiento ventana estado de factura

Función: Registro de monto de venta		
Descripción	Datos:	Maneja la tabla de monto de venta, que son rangos de valores numéricos que representan el monto de dinero, a los que se les asigna un nombre para utilizarlos en otras transacciones.
	Función:	Realizar transacciones con la tabla monto de venta.

Tabla 3.9 Requerimiento ventana monto de venta

Función: Registro de temporada		
Descripción	Datos:	Maneja dos tablas relacionadas. La primera describe la temporada y su rango de fechas. La segunda presenta rangos de días en la semana para la temporada.
	Función:	Permite modificar las temporadas con sus respectivos períodos de días en la semana.

Tabla 3.10 Requerimiento ventana temporada

Función: Registro de cajas ventas		
Descripción	Datos:	Maneja la tabla de cajas de venta, la cual registra mediante nombres a cada una de las cajas registradoras en función.
	Función:	Realizar transacciones con la tabla de cajas de venta.

Tabla 3.11 Requerimiento ventana cajas de venta

Función: Registro de descuentos		
Descripción	Datos:	Maneja la tabla de descuentos y basándose en los demás datos ingresados como temporada, lugar geográfico, montos, entre otros, se puede ingresar un porcentaje de descuento que se efectúa de acuerdo al orden que le da un campo numérico dentro de esta tabla.
	Función:	Realizar transacciones con la tabla de descuentos.

Tabla 3.12 Requerimiento ventana descuentos

Función: Registro de precios		
Descripción	Datos:	Maneja la tabla de precios y basándose en los demás datos ingresados como temporada, lugar geográfico, montos, entre otros, se puede ingresar un porcentaje de descuento que se efectúa de acuerdo al orden que le da un campo numérico dentro de esta tabla.
	Función:	Realizar transacciones con la tabla de precios.

Tabla 3.13 Requerimiento ventana precios

Función: Registro de estado cliente		
Descripción	Datos:	Maneja la tabla de estado deL cliente, que es la que indica si un cliente está activo o inactivo.
	Función:	Realizar transacciones con la tabla de estado deL cliente.

Tabla 3.14 Requerimiento ventana estado cliente

Función: Registro de tipo de cliente		
Descripción	Datos:	Maneja la tabla deL tipo de cliente, que es la que agrupa a los clientes por el tipo de consumo que realiza.
	Función:	Realizar transacciones con la tabla deL tipo de cliente.

Tabla 3.15 Requerimiento ventana tipo de cliente

Función: Registro de clientes		
Descripción	Datos:	El árbol describe los grupos de clientes y los clientes. El datawindow derecho superior presenta y permite editar los datos de los clientes. El datawindow derecho inferior despliega y permite editar la forma de pago del cliente para cada sucursal.
	Función:	Esta ventana permite realizar transacciones con la tabla de clientes además de organizarlos por grupos o padres. También permite asignar a cada cliente una forma de pago por sucursal.

Tabla 3.16 Requerimiento ventana clientes

Función: Descripción grupos de clientes		
Descripción	Datos:	Maneja la tabla de grupos de clientes, que es la que agrupa a los clientes por la entidad a la que pertenecen.
	Función:	Realizar transacciones con la tabla de grupos de clientes.

Tabla 3.17 Requerimiento ventana grupo de clientes

Función: Descripción rangos de plazo		
Descripción	Datos:	Maneja la tabla de rangos de plazo, que son rangos de valores en días, a los que se les asigna un nombre para utilizarlos cuando se asigna un plazo de pago a un cliente.
	Función:	Realizar transacciones con la tabla rango de plazo.

Tabla 3.18 Requerimiento ventana rangos de plazo

Función: Descripción formas pago		
Descripción	Datos:	Maneja dos tablas relacionadas. La primera nombra los tipos de pago. La segunda describe cada forma de pago.
	Función:	Permite modificar las temporadas con sus respectivos períodos de días en la semana.

Tabla 3.19 Requerimiento ventana formas de pago

Función: Descripción servicio para tarjetas		
Descripción	Datos:	Maneja la tabla de servicio para manejar las tarjetas de crédito, que son los nombres de las empresas que prestan este servicio.
	Función:	Realizar transacciones con la tabla j_servic_tarjet.

Tabla 3.20 Requerimiento ventana servicio para tarjetas

Función: Descripción forma de pago tarjetas		
Descripción	Datos:	Esta maneja las formas de pago que se pueden hacer con una determinada tarjeta de crédito y el porcentaje de recargo por la transacción.
	Función:	Realizar transacciones con la tabla j_for_pag_tarje.

Tabla 3.21 Requerimiento ventana forma de pago tarjetas

Función: Punto de venta (pedidos, cotización, facturación)		
Descripción	Datos:	Trabaja con todas la tablas mencionadas a en las funciones de los registros, además de una cabecera, detalle, forma de pago, detalle de forma de pago y una tabla para adicionales.
	Función:	Trabaja con todas las tablas que intervienen en el pedido y sus tablas semejantes para cotización y facturación.

Tabla 3.22 Requerimiento Punto de venta

Función: Descripción volumen de venta		
Descripción	Datos:	Maneja la tabla de volumen de venta, que son rangos de valores numéricos que representan la cantidad vendida, a los que se les asigna un nombre para utilizarlos en otras transacciones.
	Función:	Realizar transacciones con la tabla volumen de venta.

Tabla 3.23 Requerimiento ventana volumen de venta

Función: Descripción vencimientos		
Descripción	Datos:	Dependiendo del tipo y grupo de cliente se puede seleccionar un plazo para los créditos propios.
	Función:	Realizar transacciones con la tabla vencimiento.

Tabla 3.24 Requerimiento ventana vencimientos

Función: Descripción bodegaje		
Descripción	Datos:	Maneja la tabla de bodegaje, que es la que describe las bodegas donde se guardan los artículos ya comprados.
	Función:	Realizar transacciones con la tabla de bodegaje.

Tabla 3.25 Requerimiento ventana bodegaje

Función: Descripción tipo de venta		
Descripción	Datos:	Permite registrar los tipos de venta, sabiendo que puede ser una venta directa o con crédito.
	Función:	Realizar transacciones con la tabla de tipos de venta.

Tabla 3.26 Requerimiento ventana tipo de venta

3.2.11 Requerimientos de Rendimiento

Como se sabe para una venta en un punto de venta, se necesita rapidez y agilidad tanto en el cajero como en el sistema que se está utilizando para registrar la venta, por esto el tiempo de respuesta tiene que ser tomado muy en cuenta en el momento de realizar las pruebas.

Es necesario, tanto para la parte de Intranet como para la parte de Internet tener todos los datos en memoria al momento de empezar la pantalla de puntos de venta, esto es para que no pierda tiempo tomando datos de la base de datos a cada momento.

3.2.12 Restricciones de diseño

Para la parte Intranet se utilizarán las ventanas genéricas del ERP Venture y ventanas heredadas con funciones especializadas. Las ventanas genéricas son ventanas hijas de una ventana MDI, la cual llevará internamente según sus necesidades uno o varios datawindows en un orden predispuesto por el genérico, separados entre si por una barra movable, que da flexibilidad de tamaño a los datawindows y los botones van en la parte superior. En el caso de la ventana para

puntos de venta será una ventana heredada, su diseño será nuevo pero llevando una similitud con las ventanas genéricas.

Para la parte de Internet, las ventanas se generarán con el genérico que tiene Venture para Internet, el cual tiene la misma funcionalidad que el genérico para Intranet hecho en power builder. Para la ventana que se utilizará para hacer la venta tendrá un formato especial, con un diseño propio el cual será aprobado por el gerente técnico de Venture y será una nueva clase que se llamará desde el menú de opciones del sistema.

3.2.12.1 Estándares a cumplir

Estos estándares son dados por la empresa Venture porque el módulo de puntos de venta tanto para intranet como para Internet van dentro del ERP entonces se tiene que seguir estos parámetros. Los estándares a cumplir se describen en el capítulo 2 en las definiciones de estándares.

3.2.12.2 Limitaciones de hardware

Este sistema en su primera etapa, Intranet, se desarrollará en un sistema Operativo Windows para la parte de ejecución y para la base de datos aunque se prefiere un servidor Windows, el cliente tiene la decisión de utilizar otro sistema operativo, teniendo en cuenta que en este debe correr la base de datos que haya elegido de entre las que maneja el ERP Venture y tiene que tener una perfecta conectividad con los puestos de trabajo.

En su segunda etapa, Internet, para la parte del programa se puede elegir cualquier sistema operativo que le permita ser un servidor Web, levantar el servidor Tomcat y salir al Internet, en el caso de la base de datos sería igual que para la parte de Intranet y de la misma forma debe haber una perfecta conexión entre la base de datos y el programa.

3.2.12.3 Requisitos de interfaces externas

Se requiere utilizar para la comunicación entre el servidor de la base de datos y los puestos de trabajo una red 10/100 o mayor con cableado y dispositivos que soporten esta velocidad.

Es necesario que el usuario de Internet tenga un ancho de banda mayor o igual a 46 Kbps.

En el caso de las replicaciones se necesita una conexión rápida, de preferencia que sea a Internet con una IP fija o una conexión dedicada para que los intervalos de tiempo en que se replique la base sean muy cortos aunque lo indispensable es que la conexión sea constante.

3.2.12.4 Otros

Uno de los requerimientos de Venture para el desarrollo del código es mantener la seguridad de las fuentes, por lo que la programación en la parte Intranet se hará en la empresa.

En la última reunión con la empresa auspiciante, Venture Venti surgieron nuevos requerimientos, los cuales están fuera del planteamiento inicial de la tesis, pero al ver la necesidad de la empresa se consintió cubrir los cambios que se presentan a continuación:

- Para el ambiente intranet el punto de venta tiene que interactuar con los demás módulos, es decir inventario, cuentas por cobrar y contabilidad. Como estos cambios no modifican el diseño establecido, se presentará la documentación de cómo se hizo esta implementación en el anexo B.

- Para el ambiente Internet se requiere realizar una página que realice las funciones básicas de la ventana de punto de venta en intranet, para esto el diseño no cambia a excepción del diagrama de clases ya que este cambio representa nuevas clases, el desarrollo de esta página se describe en el Capítulo V.

CAPÍTULO IV

4 DISEÑO

Después de conocer los requerimientos se empieza el diseño del módulo de puntos de venta, este diseño se basa en la creación de varios diagramas que serán la guía en el momento de desarrollar el software.

4.1 MODELO LÓGICO DE DATOS (Tamaño real anexo G)

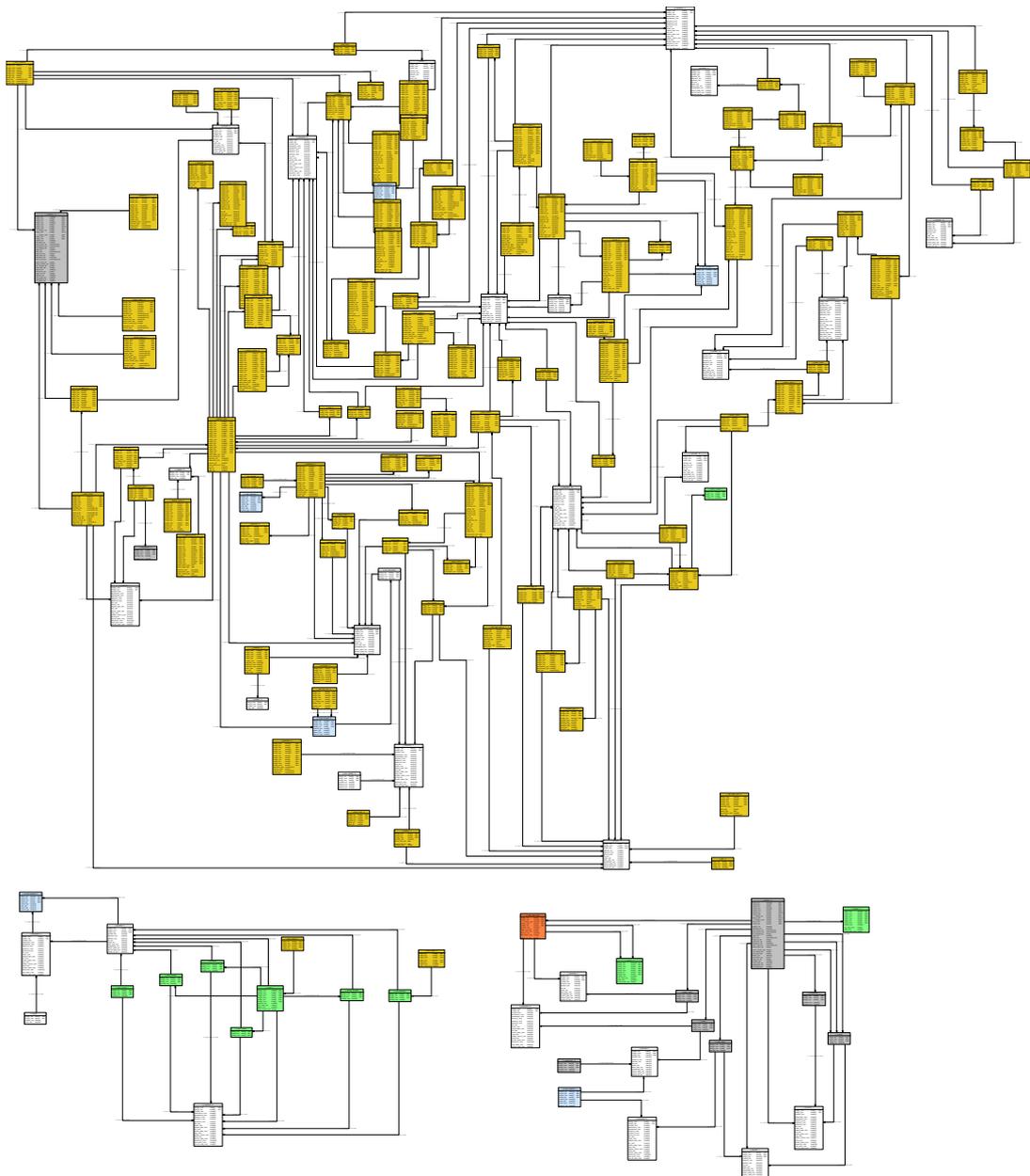


Fig. 4.1 Diagrama lógico de datos

4.2 DISEÑO DE DIAGRAMAS

4.2.1 Diagrama de procesos del negocio

Este diagrama define un contexto aislado en el cual la integridad de elementos pueda ser comprobada al ver los procesos que cumple.

4.2.1.1 Procesos del negocio (intranet)

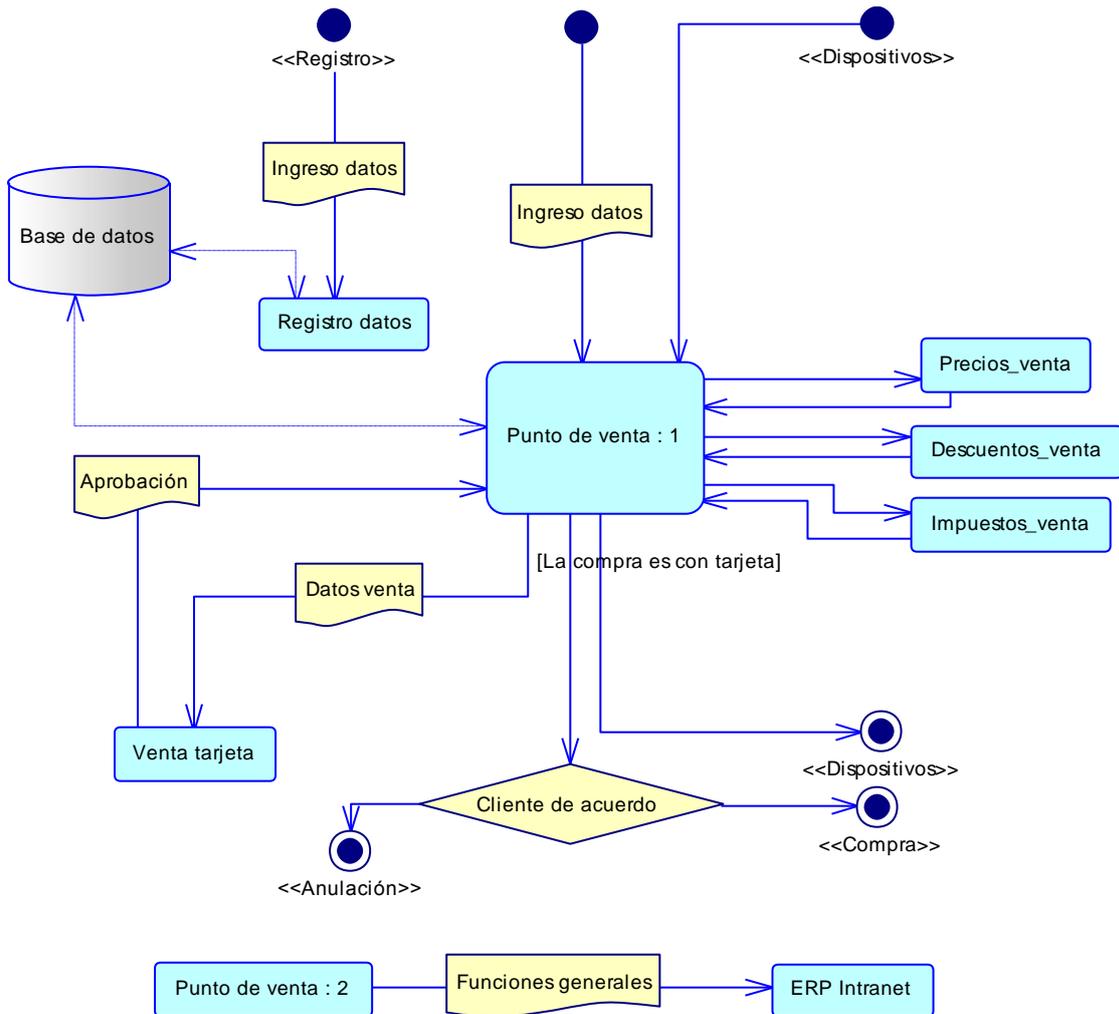


Fig. 4.2 Diagrama de procesos del negocio para intranet

4.2.2 Diagrama de casos de uso

Estos diagramas permiten definir la estructura fundamental del módulo que se está diseñando.

4.2.2.1 Diagramas individuales de los casos de uso

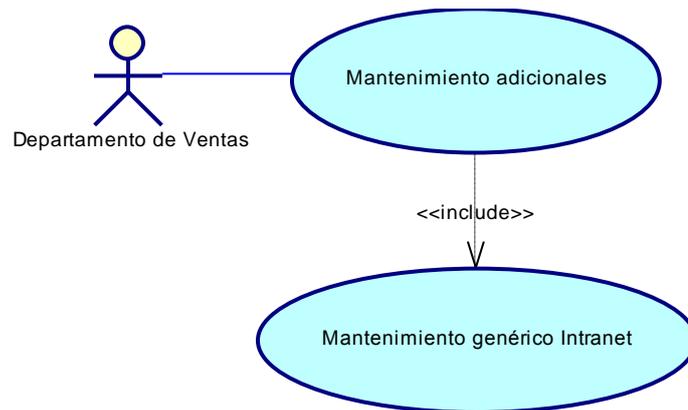


Fig. 4.3 Diagrama de mantenimiento de adicionales

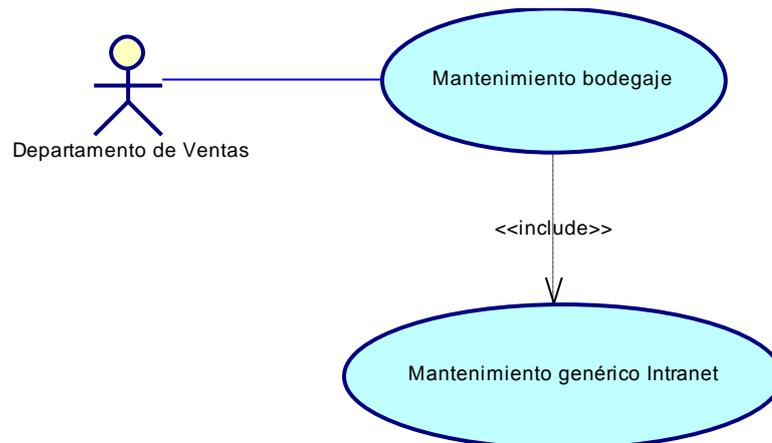


Fig. 4.4 Diagrama de mantenimiento de bodegas

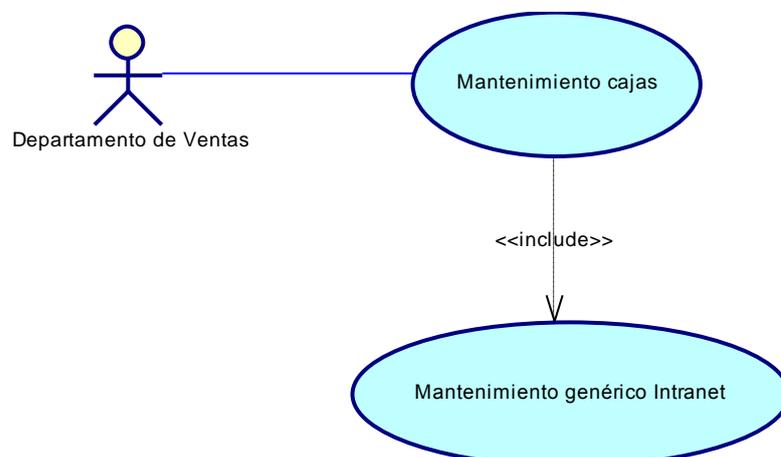


Fig. 4.5 Diagrama de mantenimiento de cajas

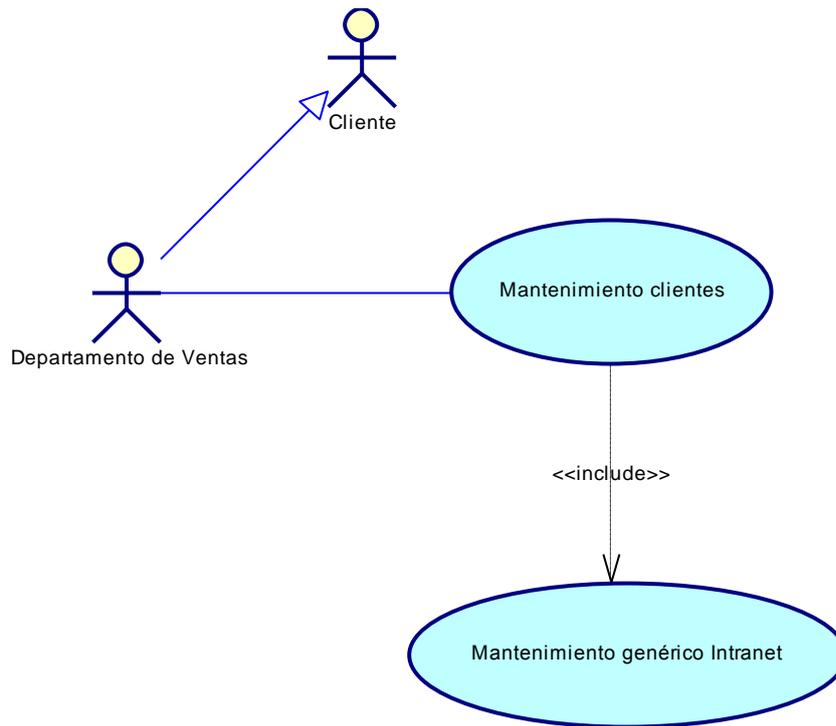


Fig. 4.6 Diagrama de mantenimiento de clientes

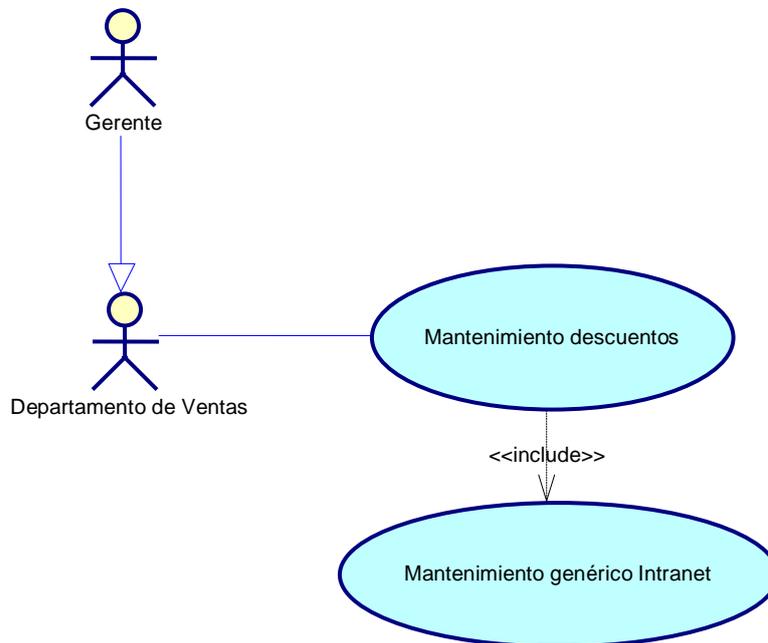


Fig. 4.7 Diagrama de mantenimiento de descuentos

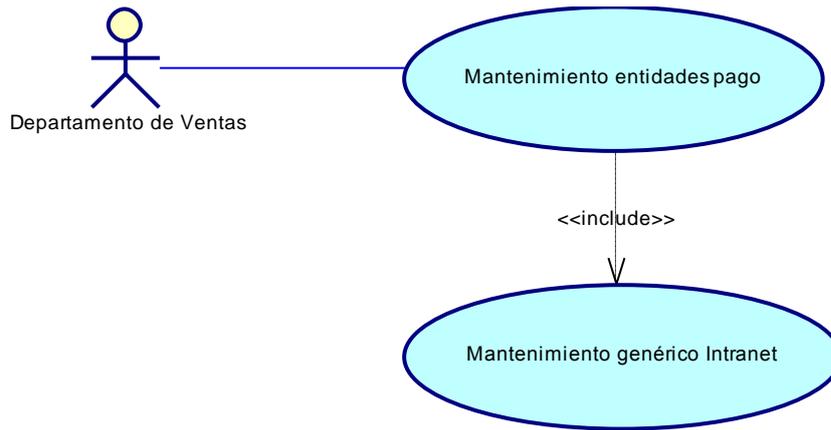


Fig. 4.8 Diagrama de mantenimiento de entidades de pago

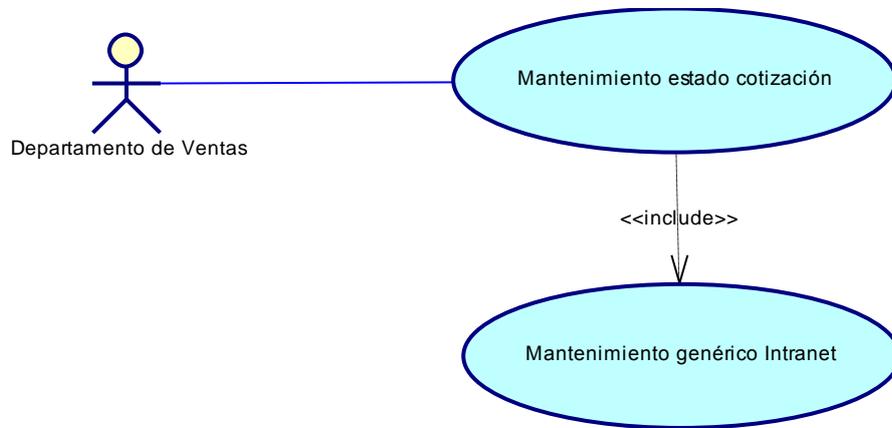


Fig. 4.9 Diagrama de mantenimiento de estado de cotización

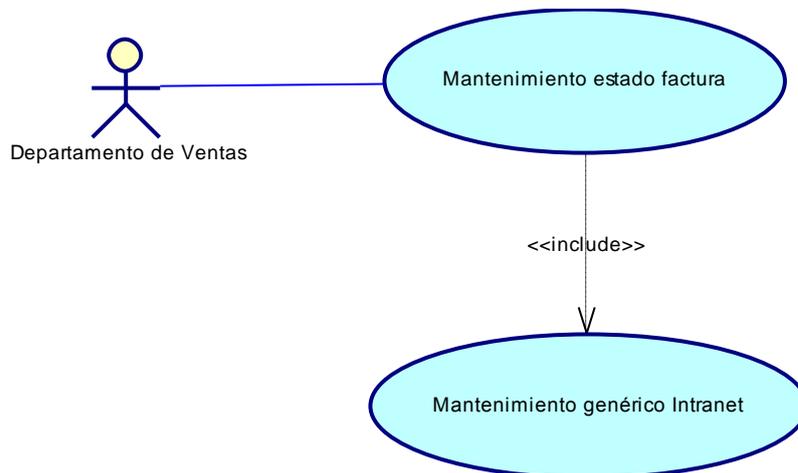


Fig. 4.10 Diagrama de mantenimiento de estado de factura

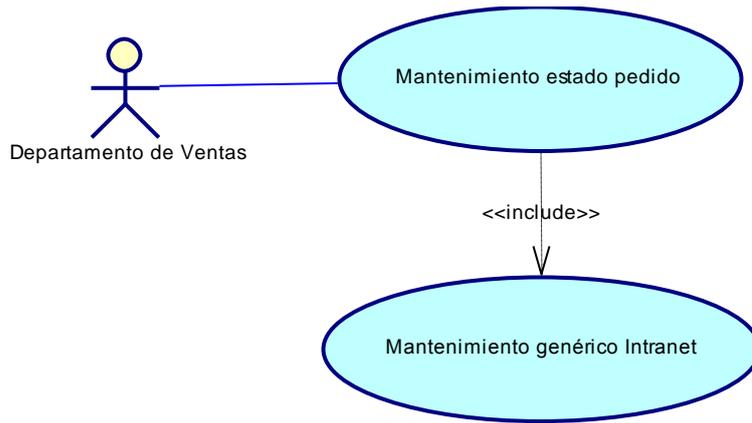


Fig. 4.11 Diagrama de mantenimiento de estado de pedido

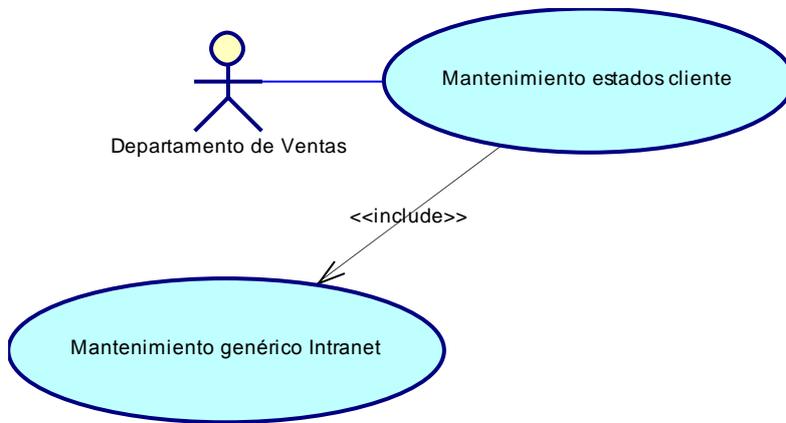


Fig. 4.12 Diagrama de mantenimiento de estado de cliente

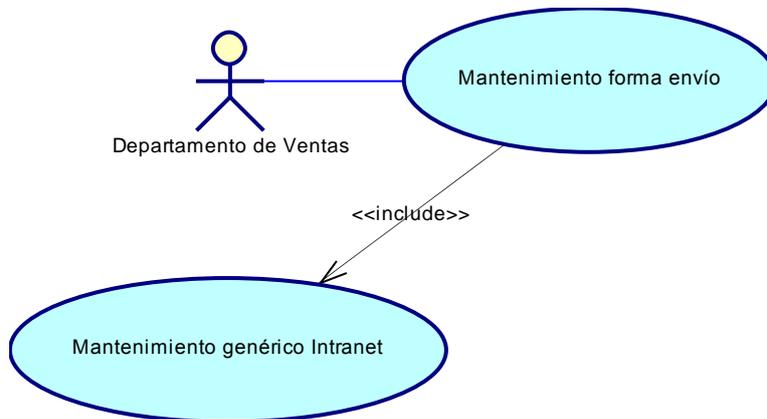


Fig. 4.13 Diagrama de mantenimiento de forma envío

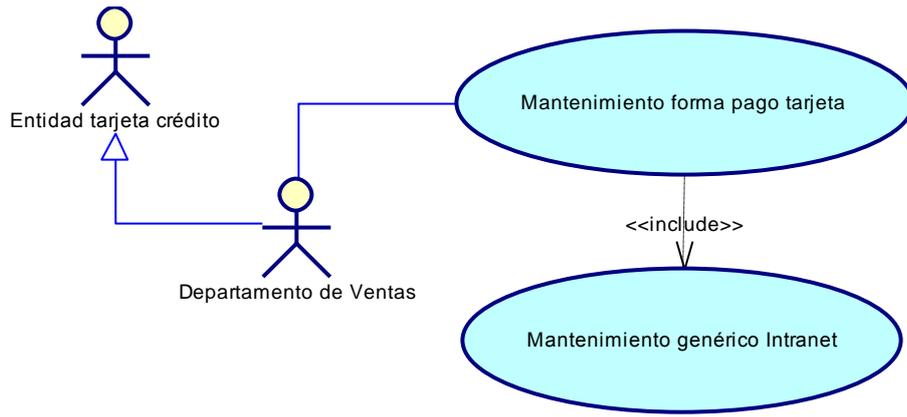


Fig. 4.14 Diagrama de mantenimiento de forma pago tarjeta

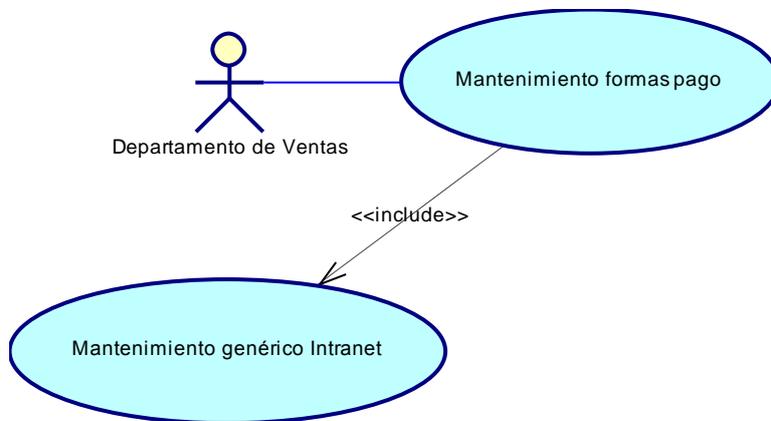


Fig. 4.15 Diagrama de mantenimiento de formas pago

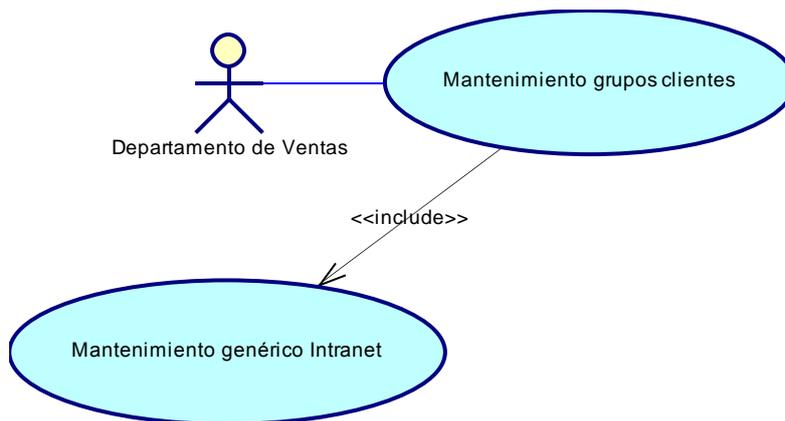


Fig. 4.16 Diagrama de mantenimiento de grupo clientes

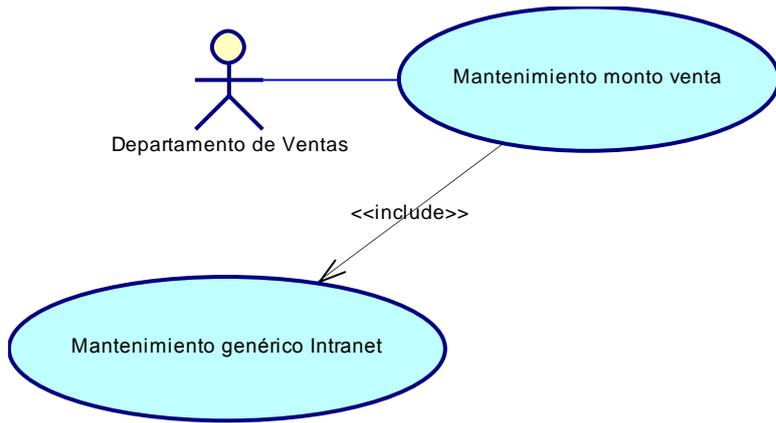


Fig. 4.17 Diagrama de mantenimiento de monto venta

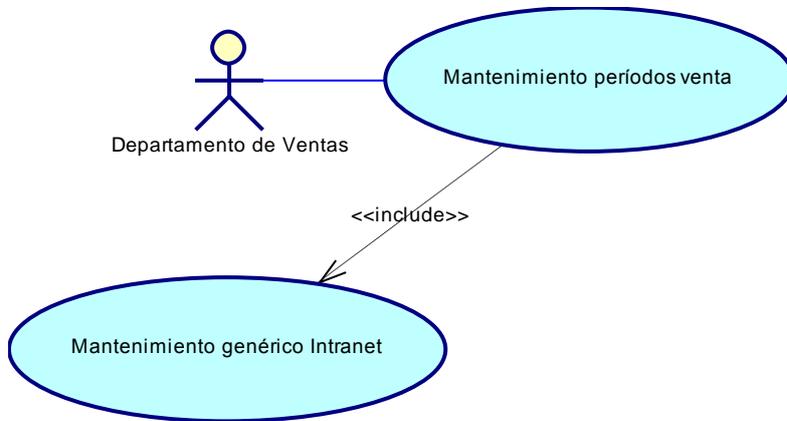


Fig. 4.18 Diagrama de mantenimiento de periodos de venta

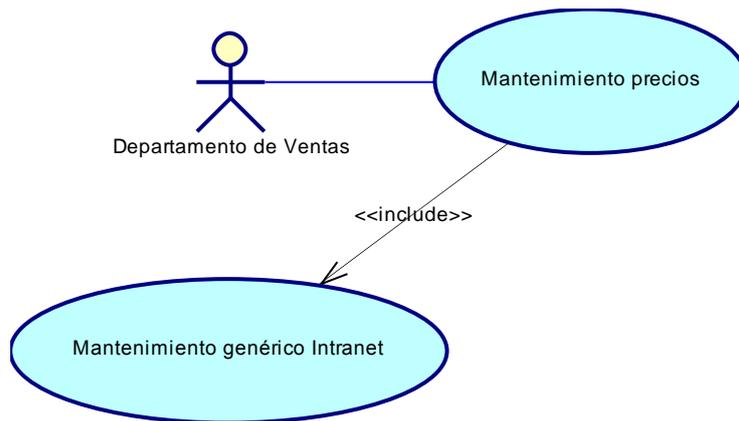


Fig. 4.19 Diagrama de mantenimiento de precios

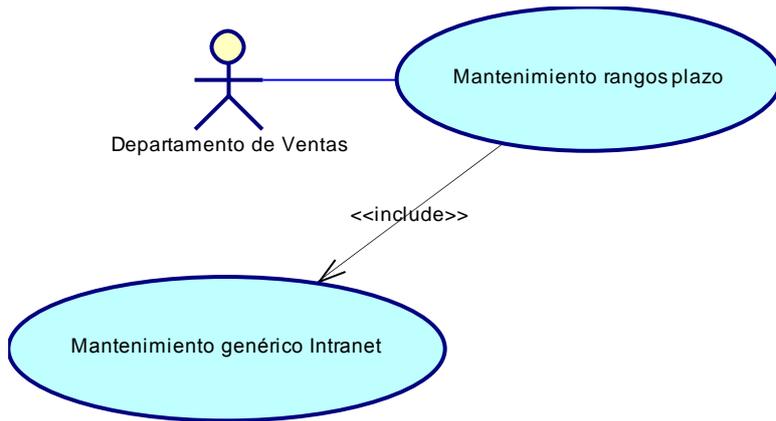


Fig. 4.20 Diagrama de mantenimiento de rango plazos

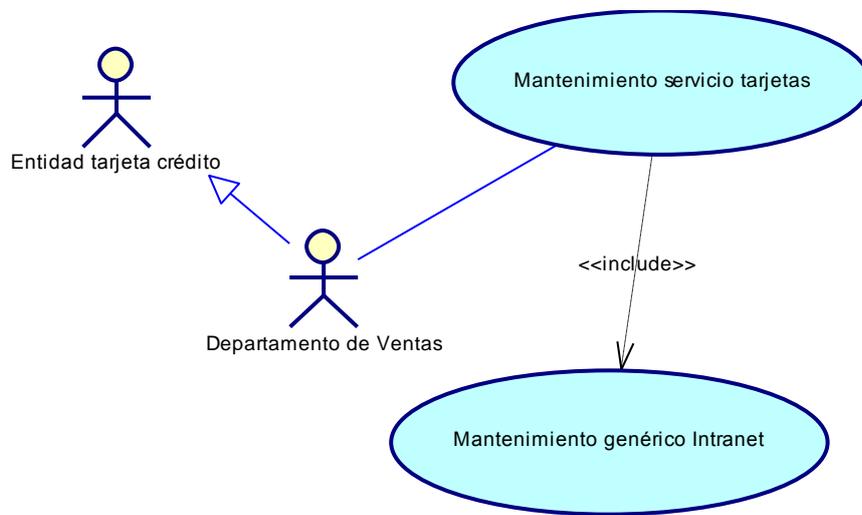


Fig. 4.21 Diagrama de mantenimiento de servicio para tarjeta

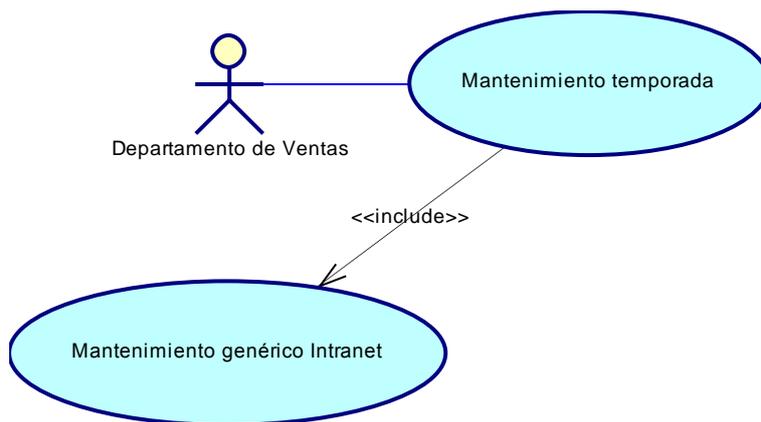


Fig. 4.22 Diagrama de mantenimiento de temporada

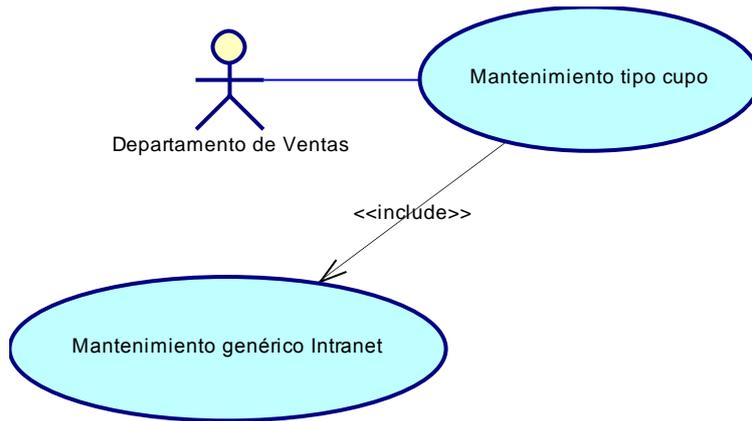


Fig. 4.23 Diagrama de mantenimiento de tipo de cupo

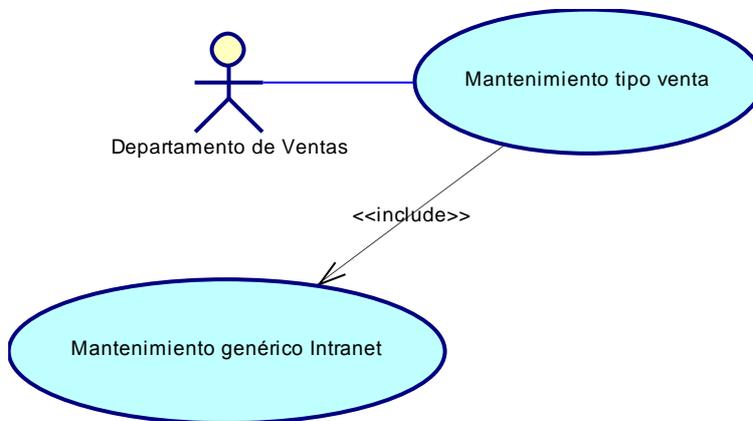


Fig. 4.24 Diagrama de mantenimiento de tipo de venta

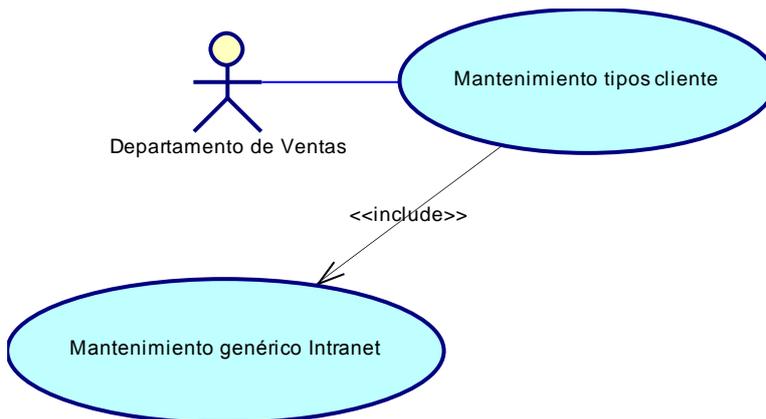


Fig. 4.25 Diagrama de mantenimiento de tipos de clientes

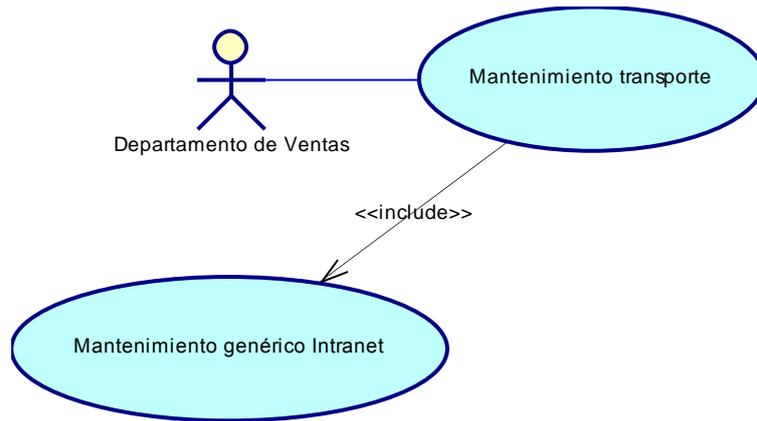


Fig. 4.26 Diagrama de mantenimiento de transporte

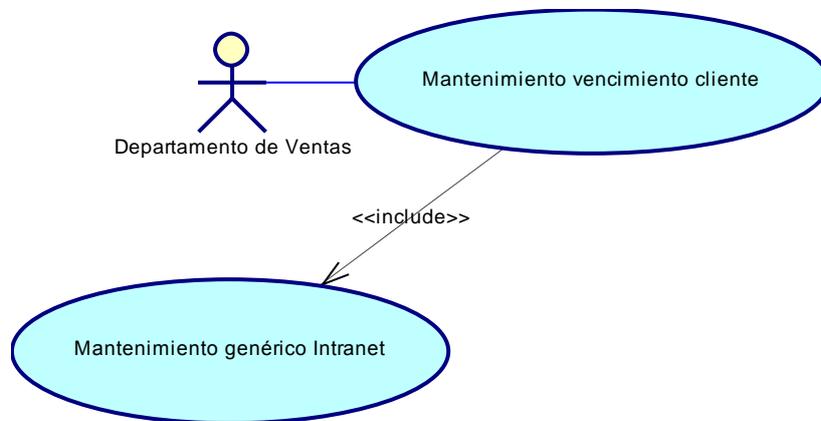


Fig. 4.27 Diagrama de mantenimiento de vencimiento de clientes

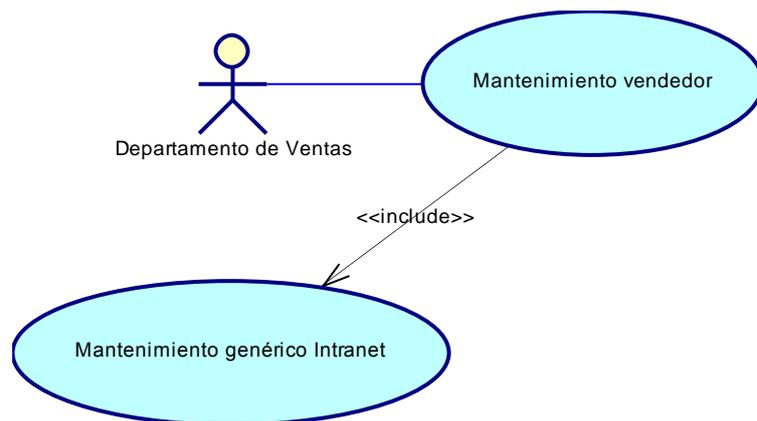


Fig. 4.28 Diagrama de mantenimiento de vendedor

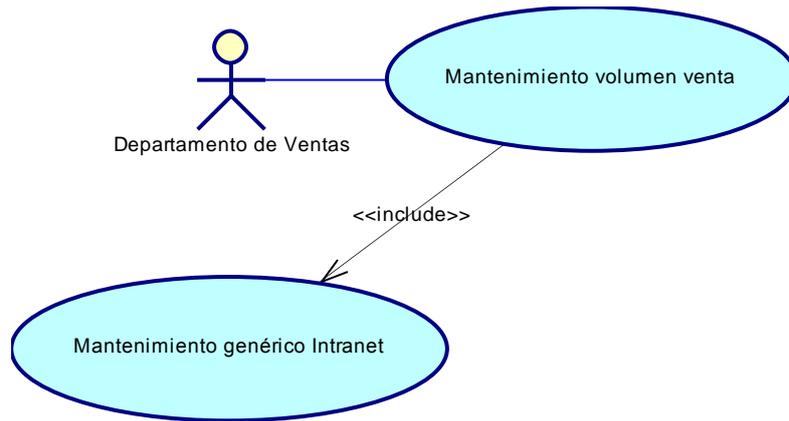


Fig. 4.29 Diagrama de mantenimiento de volumen de venta

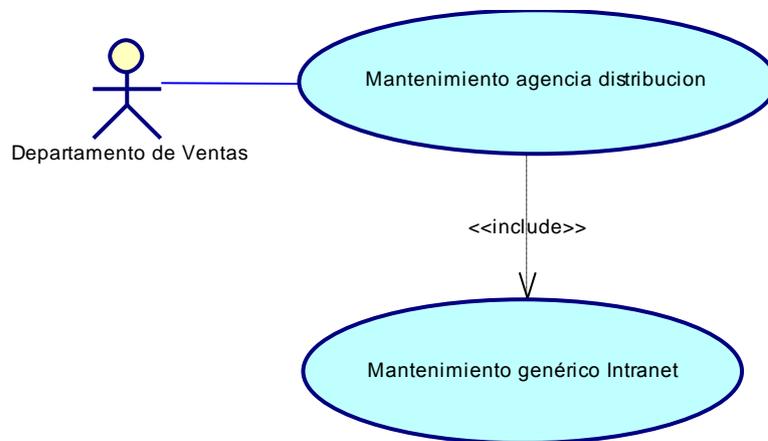


Fig. 4.30 Diagrama de mantenimiento de agencia de distribución

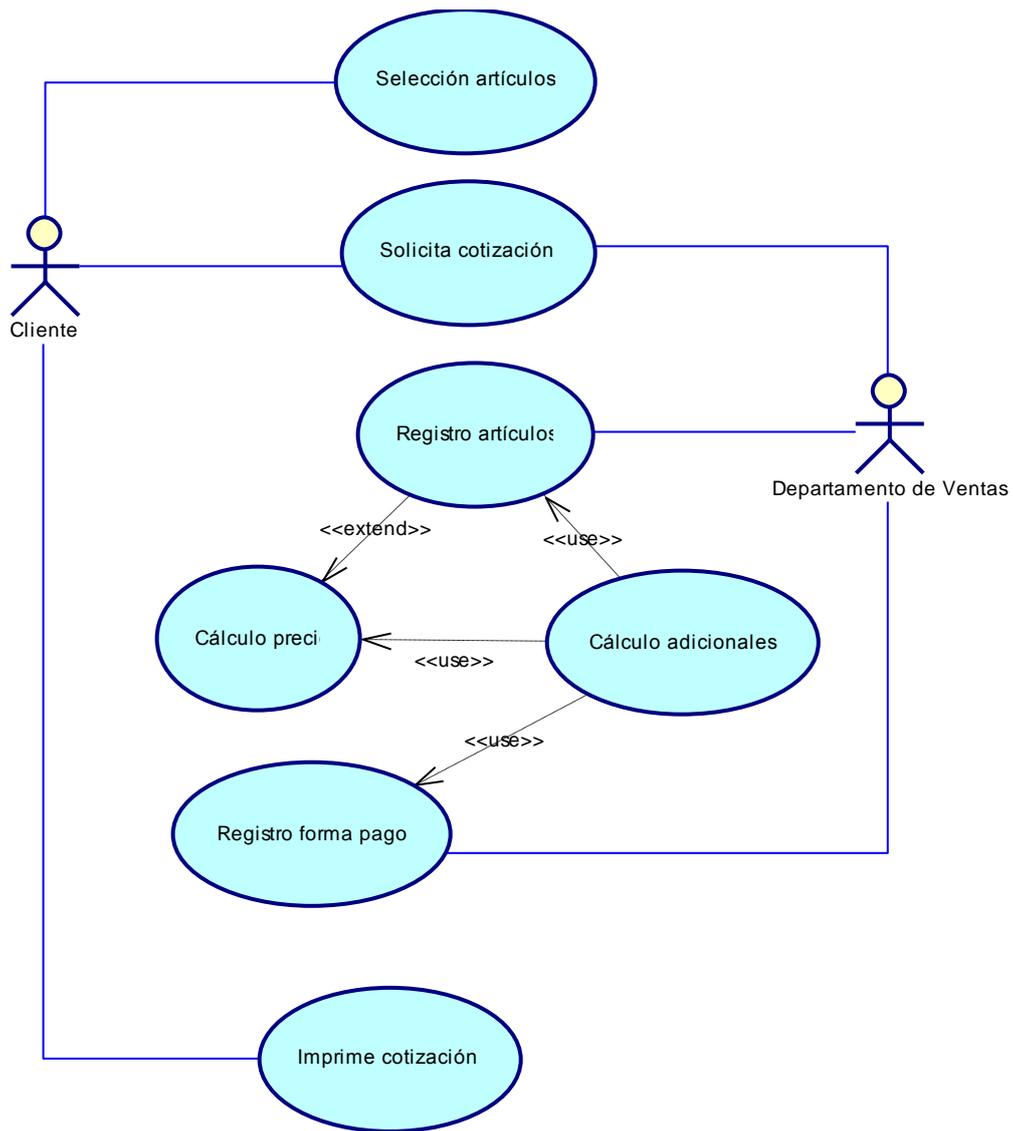


Fig. 4.31 Diagrama de movimiento de cotización

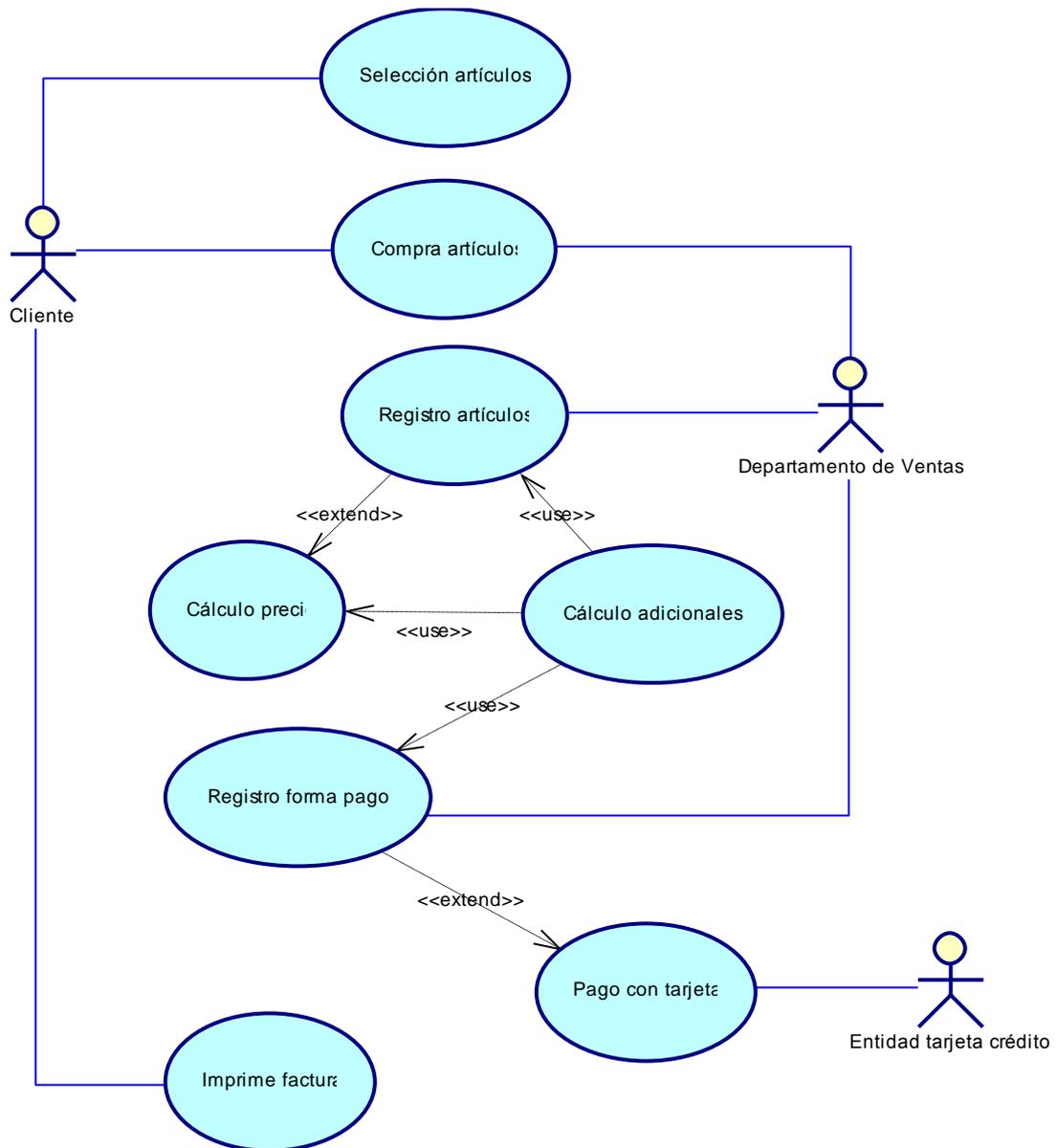


Fig. 4.32 Diagrama de movimiento de facturación

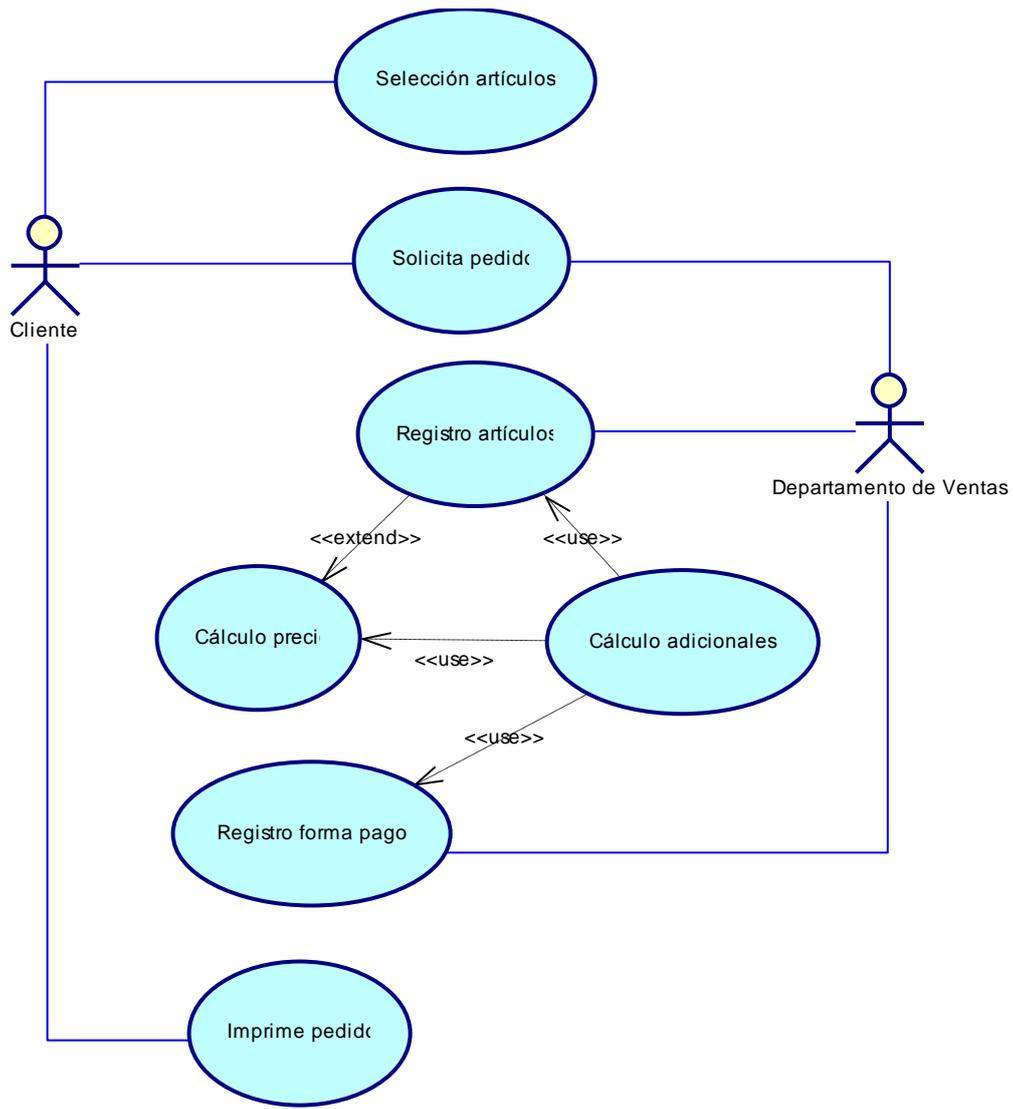


Fig. 4.33 Diagrama de movimiento de pedidos

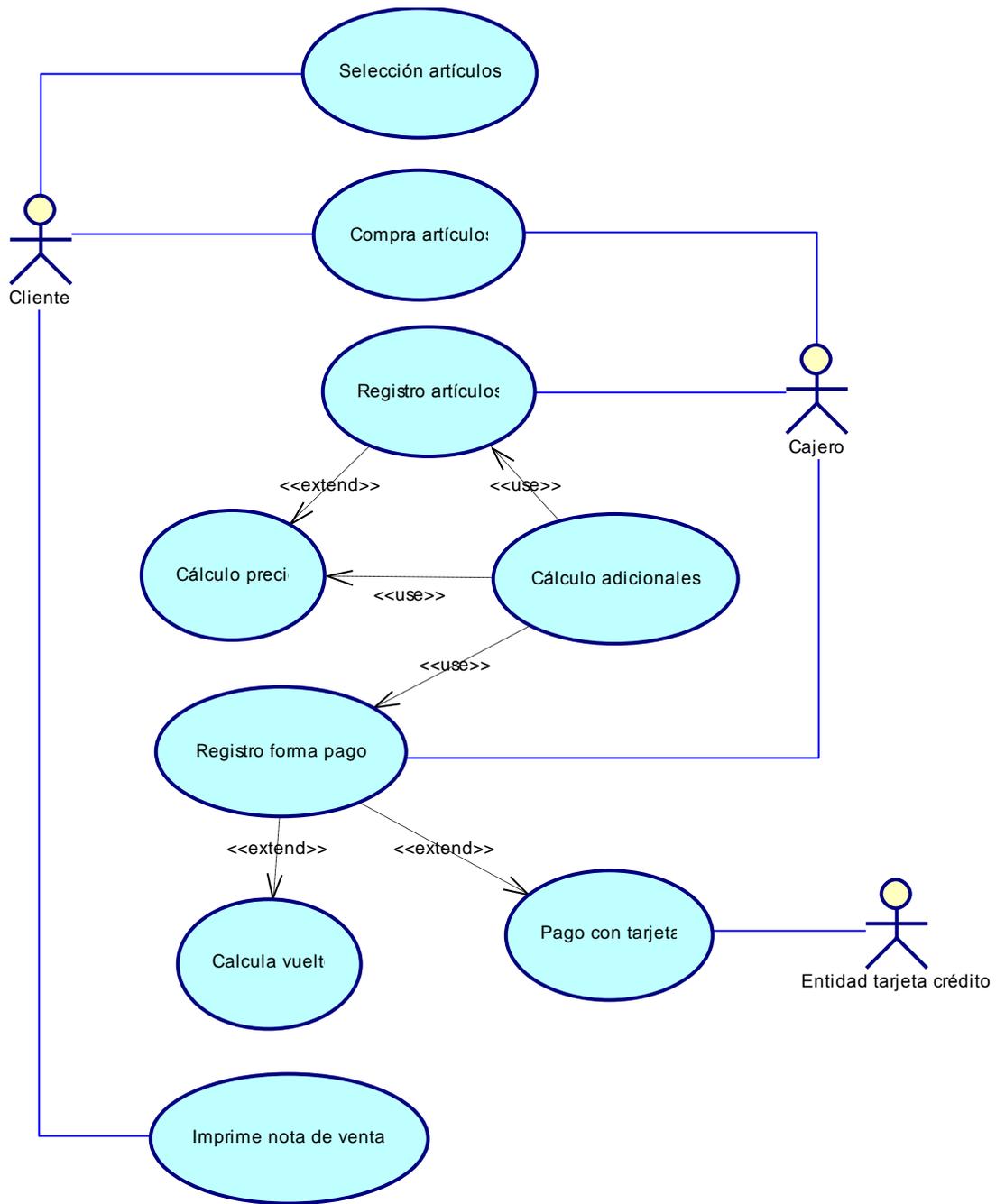


Fig. 4.34 Diagrama de movimiento de puntos de venta

4.2.2.2 Diagrama de Casos de Uso Unificado (intranet)

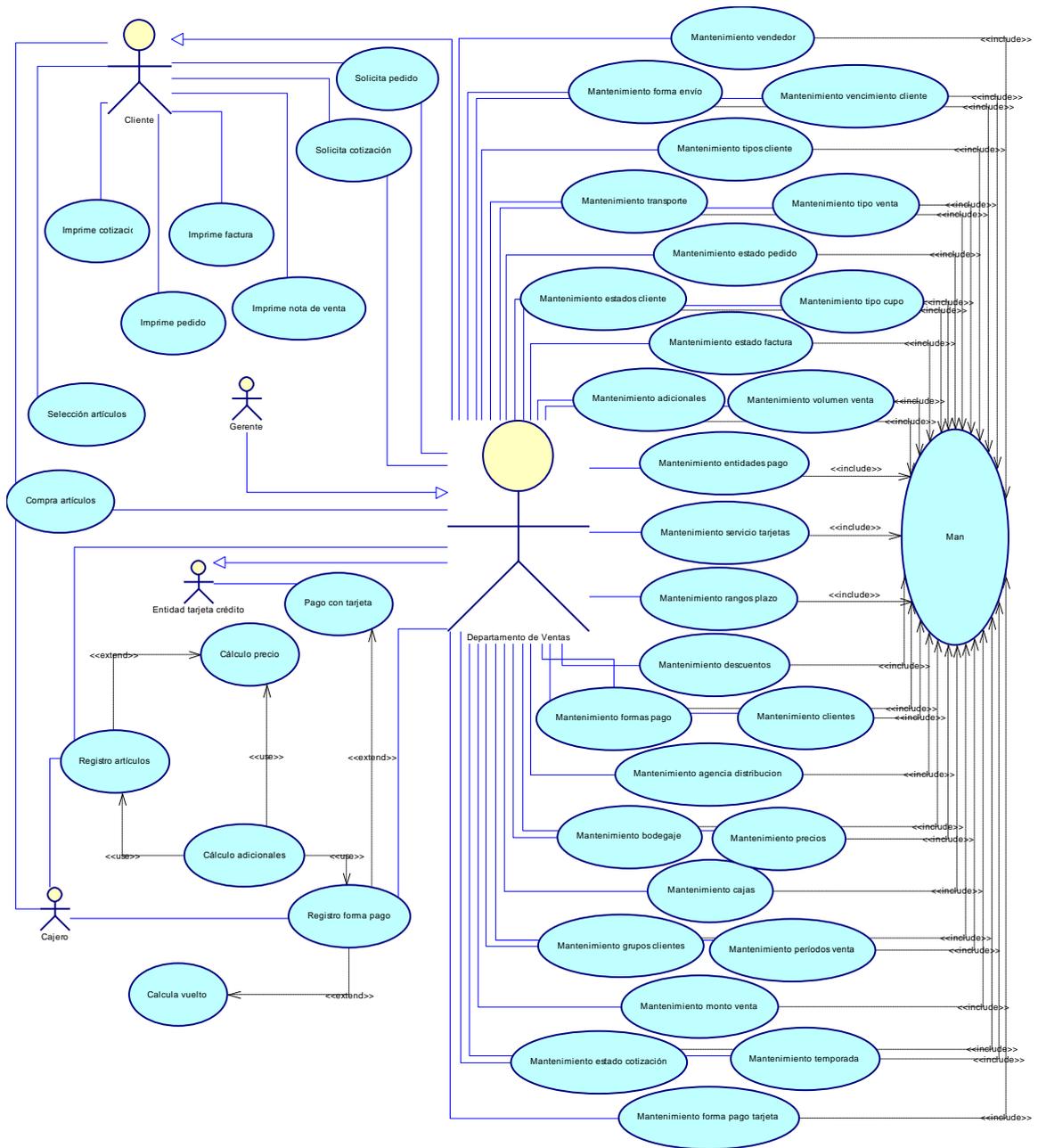


Fig. 4.35 Diagrama unificado de casos de uso (intranet)

4.2.4 Diagrama de objetos (visuales, no visuales) para intranet

Es un conjunto de diagramas que presentan las instancias de las clases del diagrama de clases con valores que van a tomar en el momento de ejecución. Estos diagramas de objetos visuales y no visuales se presentan por cada inicio de algún proceso como es el caso de algún mantenimiento o transacción.

4.2.4.1 Ventana de clientes

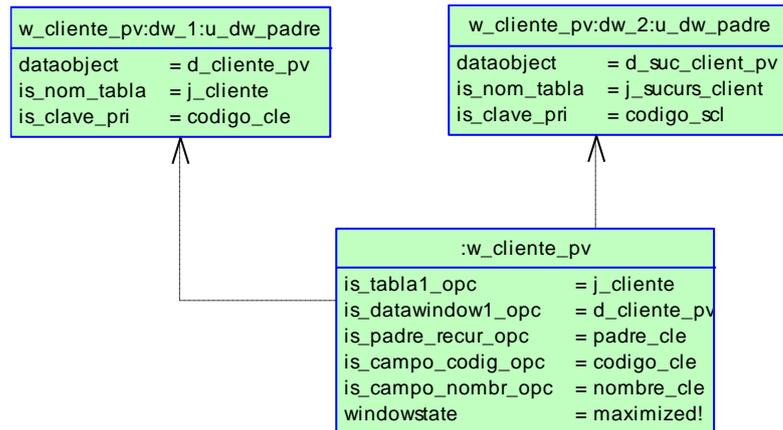


Fig. 4.37 Diagrama de objetos (ventana de clientes)

4.2.4.2 Ventana de descuentos

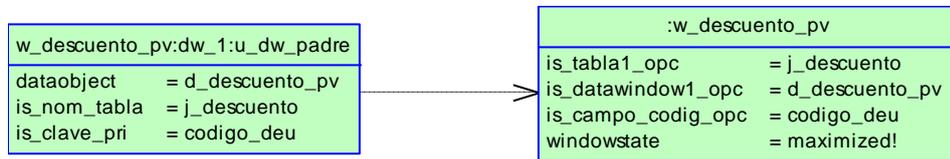


Fig. 4.38 Diagrama de objetos (ventana de descuento)

4.2.4.3 Ventana forma de envío

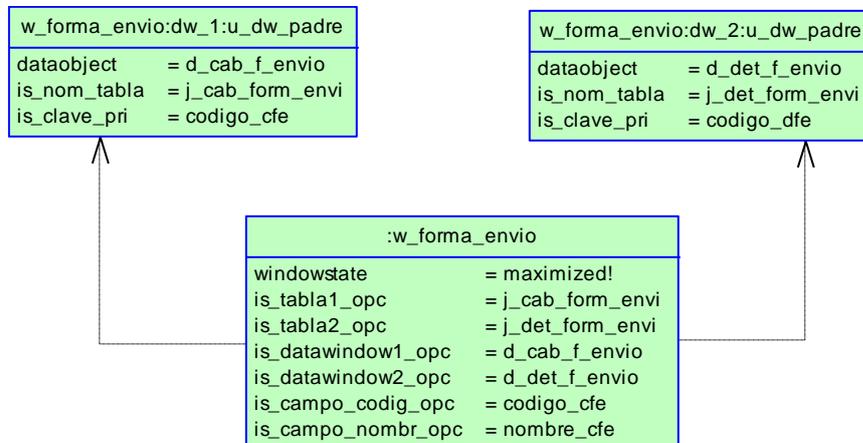


Fig. 4.39 Diagrama de objetos (ventana forma de envío)

4.2.4.4 Ventana de periodos de venta

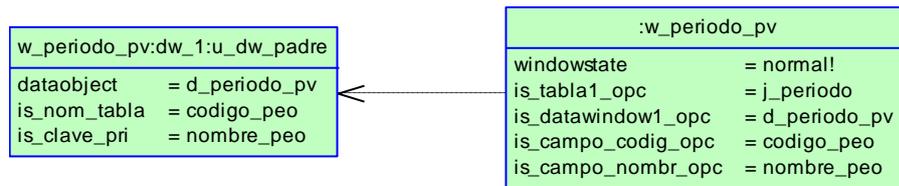


Fig. 4.40 Diagrama de objetos (ventana periodo de venta)

4.2.4.5 Ventana de precios



Fig. 4.41 Diagrama de objetos (ventana de precios)

4.2.4.6 Ventana de vencimiento del cliente

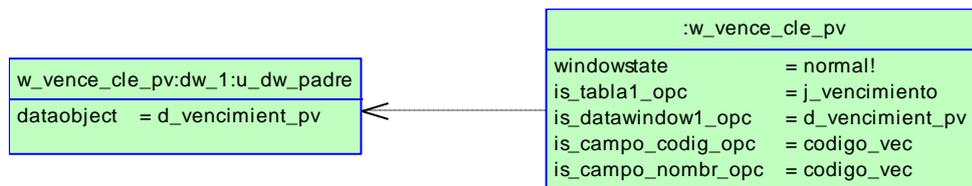


Fig. 4.42 Diagrama de objetos (ventana de vencimiento del cliente)

4.2.4.7 Ventana del vendedor

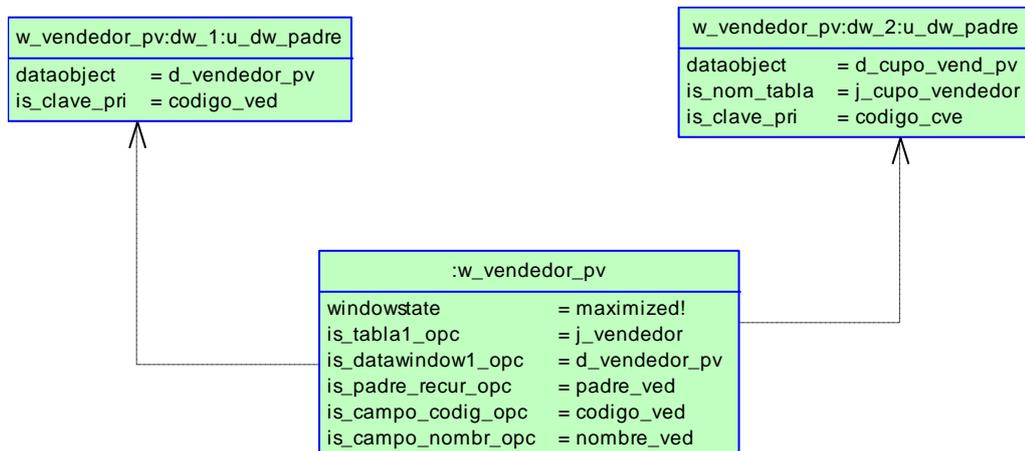


Fig. 4.43 Diagrama de objetos (ventana del vendedor)

4.2.4.8 Ventana padre de ventas

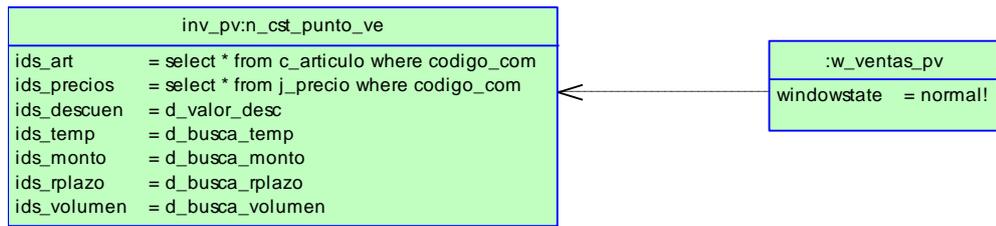


Fig. 4.44 Diagrama de objetos (ventana padre de ventas)

4.2.4.9 Ventana de cotizaciones

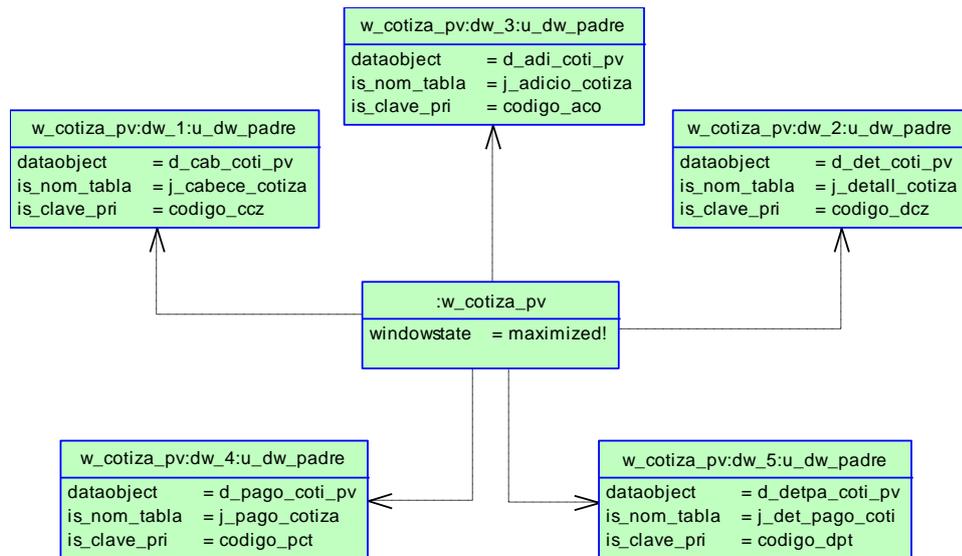


Fig. 4.45 Diagrama de objetos (ventana de cotizaciones)

4.2.4.10 Ventana de pedidos de venta

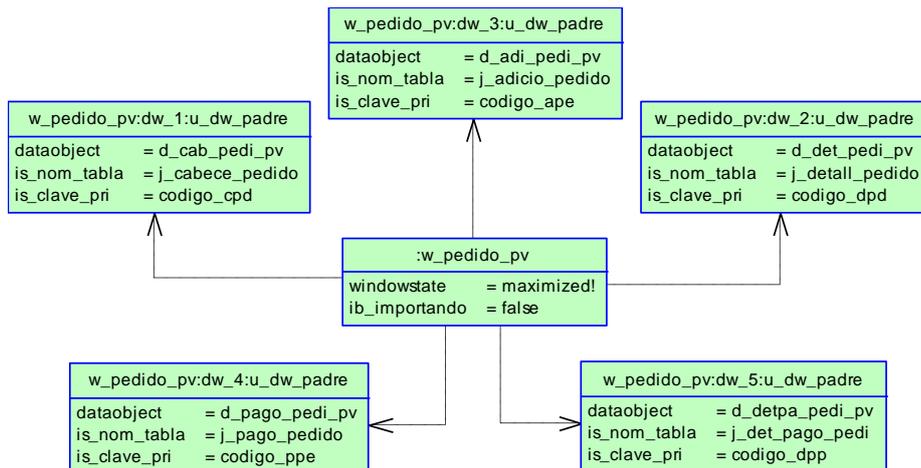


Fig. 4.46 Diagrama de objetos (ventana de pedidos)

4.2.4.11 Ventana de facturas

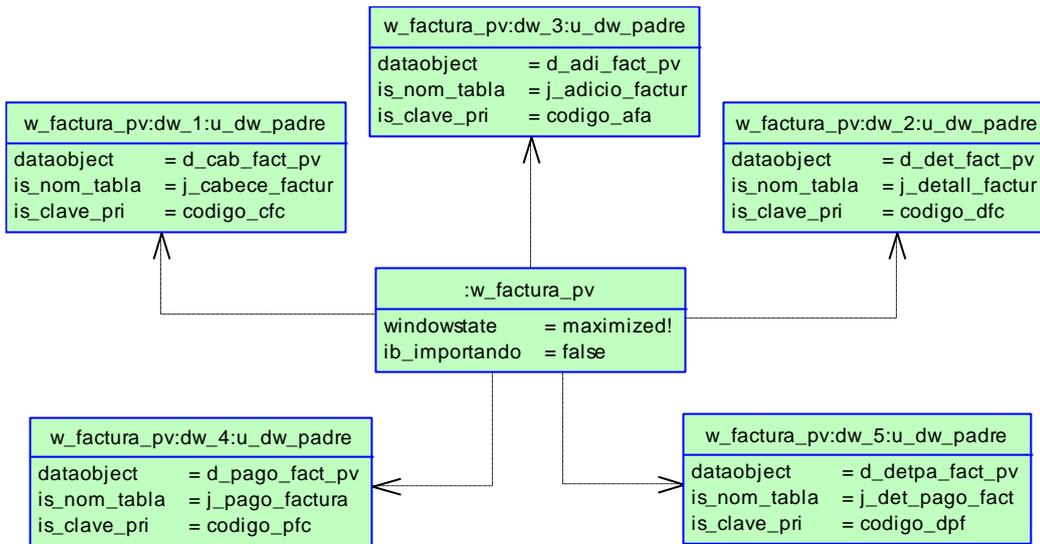


Fig. 4.47 Diagrama de objetos (ventana de facturas)

4.2.4.12 Ventana de puntos de venta

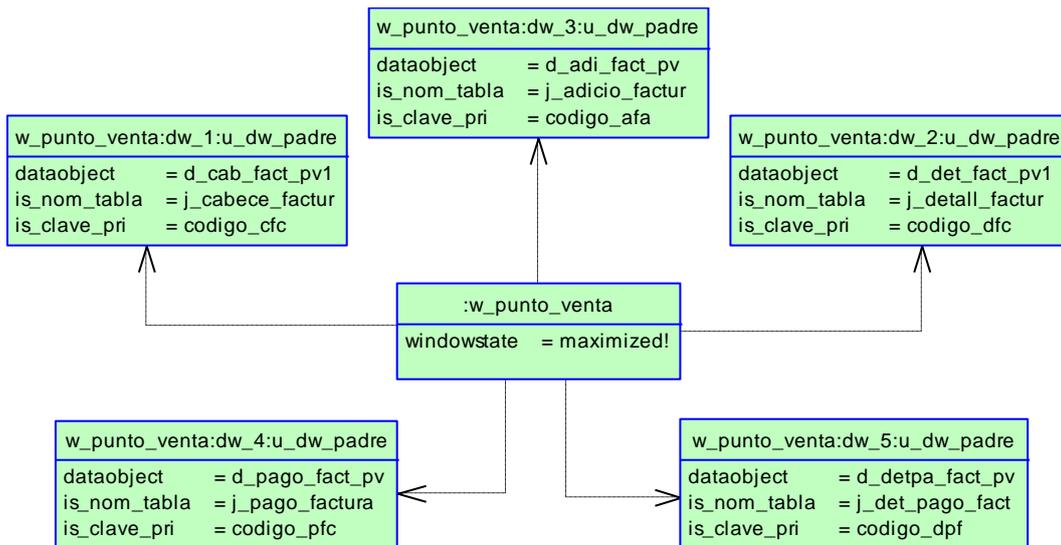


Fig. 4.48 Diagrama de objetos (ventana de puntos de venta)

4.2.5 Diagrama de secuencia (intranet)

Estos diagramas especifican el camino por donde la información pasa en el momento de realizar algún proceso.

4.2.5.1 Mantenimiento genérico

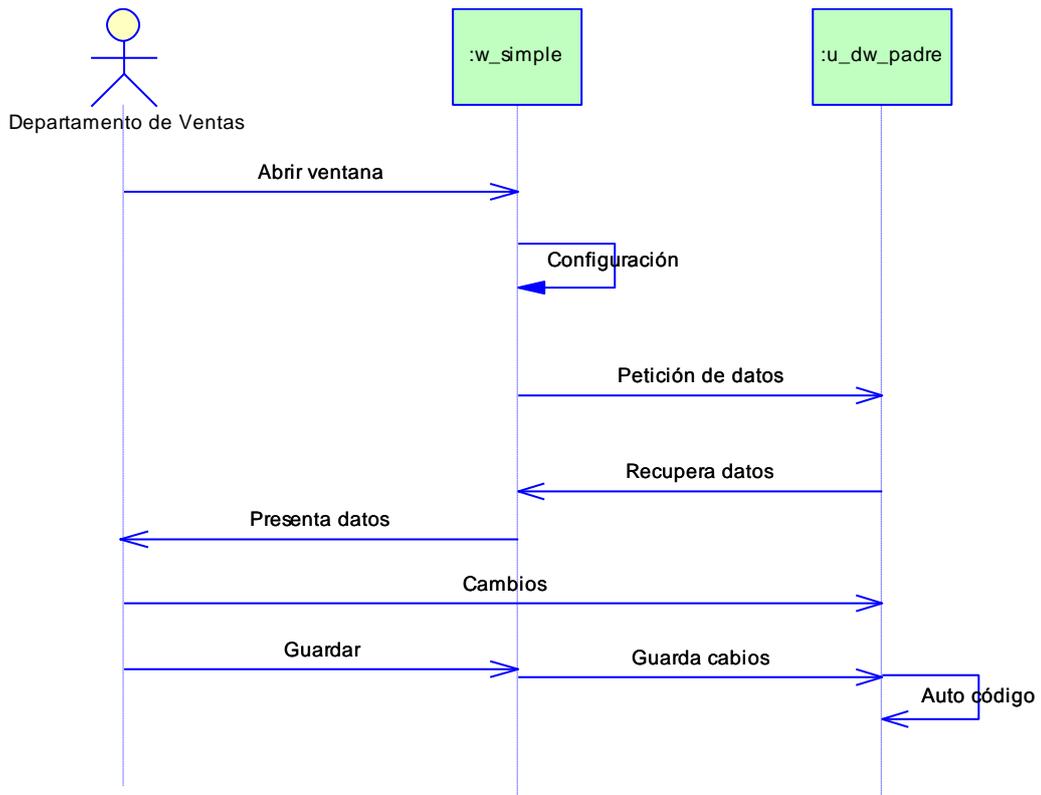


Fig. 4.49 Diagrama de secuencia (mantenimiento genérico)

4.2.5.2 Mantenimiento del cliente

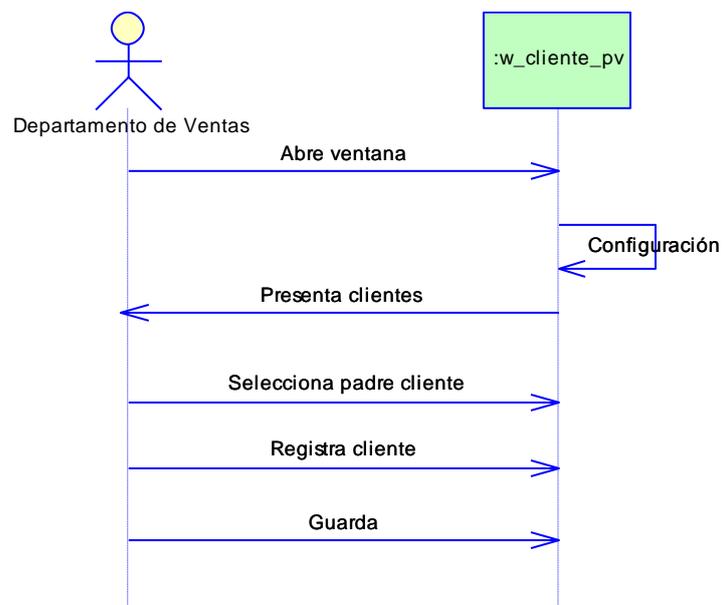


Fig. 4.50 Diagrama de secuencia (mantenimiento del cliente)

4.2.5.3 Mantenimiento del descuento

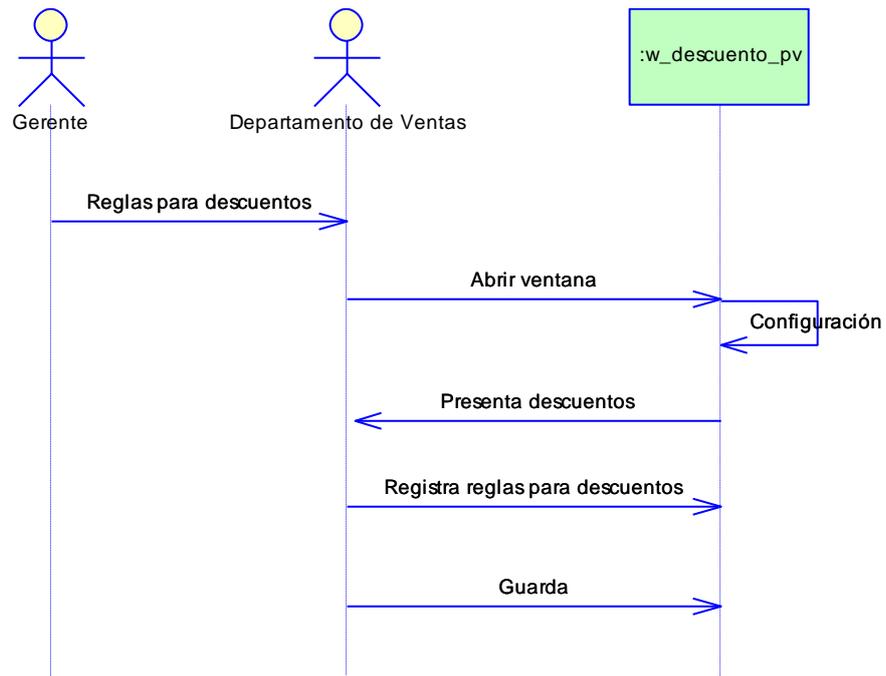


Fig. 4.51 Diagrama de secuencia (mantenimiento del descuento)

4.2.5.4 Mantenimiento de la forma de envío

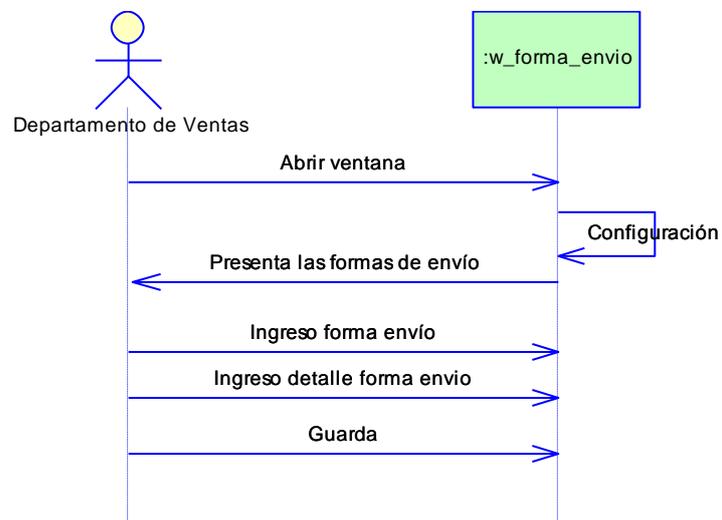


Fig. 4.52 Diagrama de secuencia (mantenimiento de forma de envío)

4.2.5.5 Mantenimiento del periodo de venta

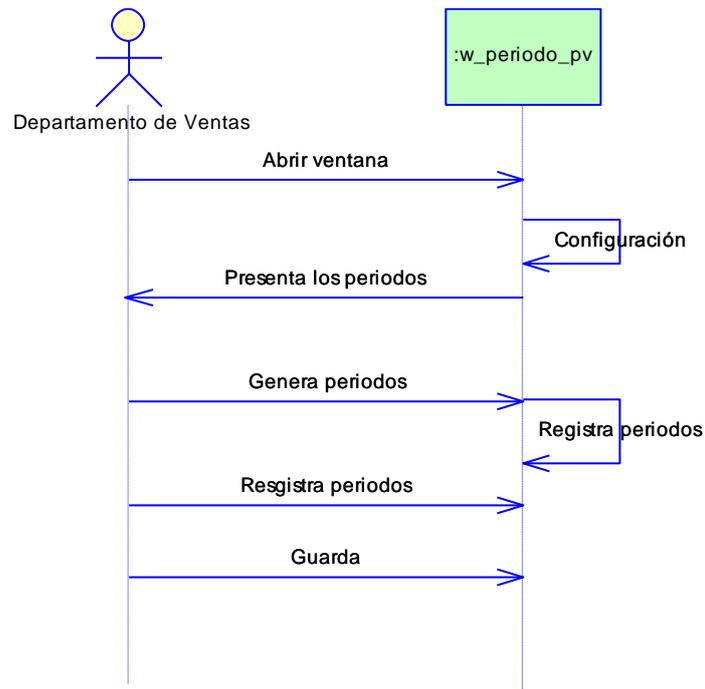


Fig. 4.53 Diagrama de secuencia (mantenimiento de periodo de venta)

4.2.5.6 Mantenimiento del precio



Fig. 4.54 Diagrama de secuencia (mantenimiento de precio)

4.2.5.7 Mantenimiento del vencimiento de clientes

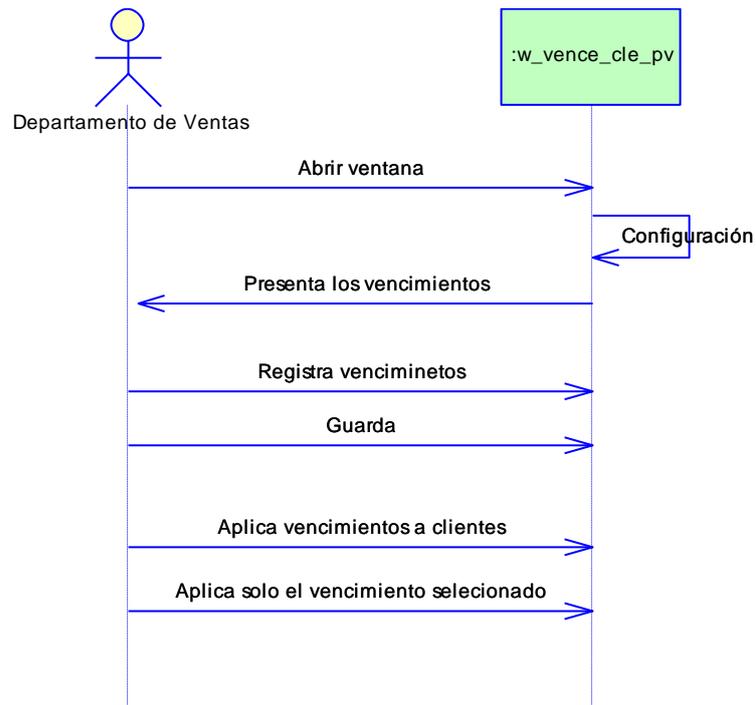


Fig. 4.55 Diagrama de secuencia (mantenimiento de vencimiento de clientes)

4.2.5.8 Mantenimiento del vendedor

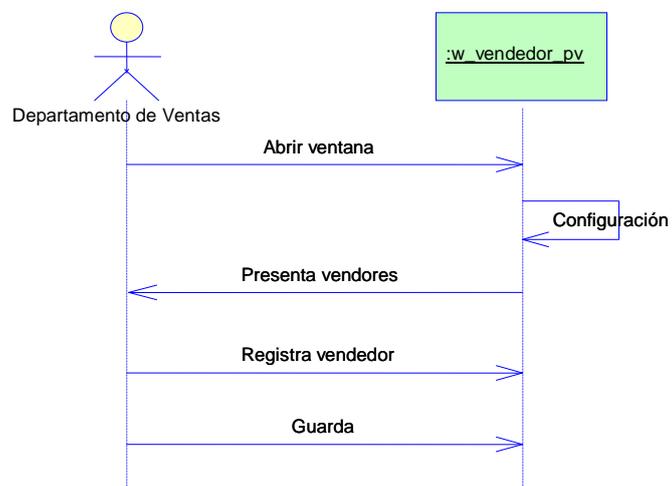


Fig. 4.56 Diagrama de secuencia (mantenimiento del vendedor)

4.2.5.9 Movimiento de cotizaciones

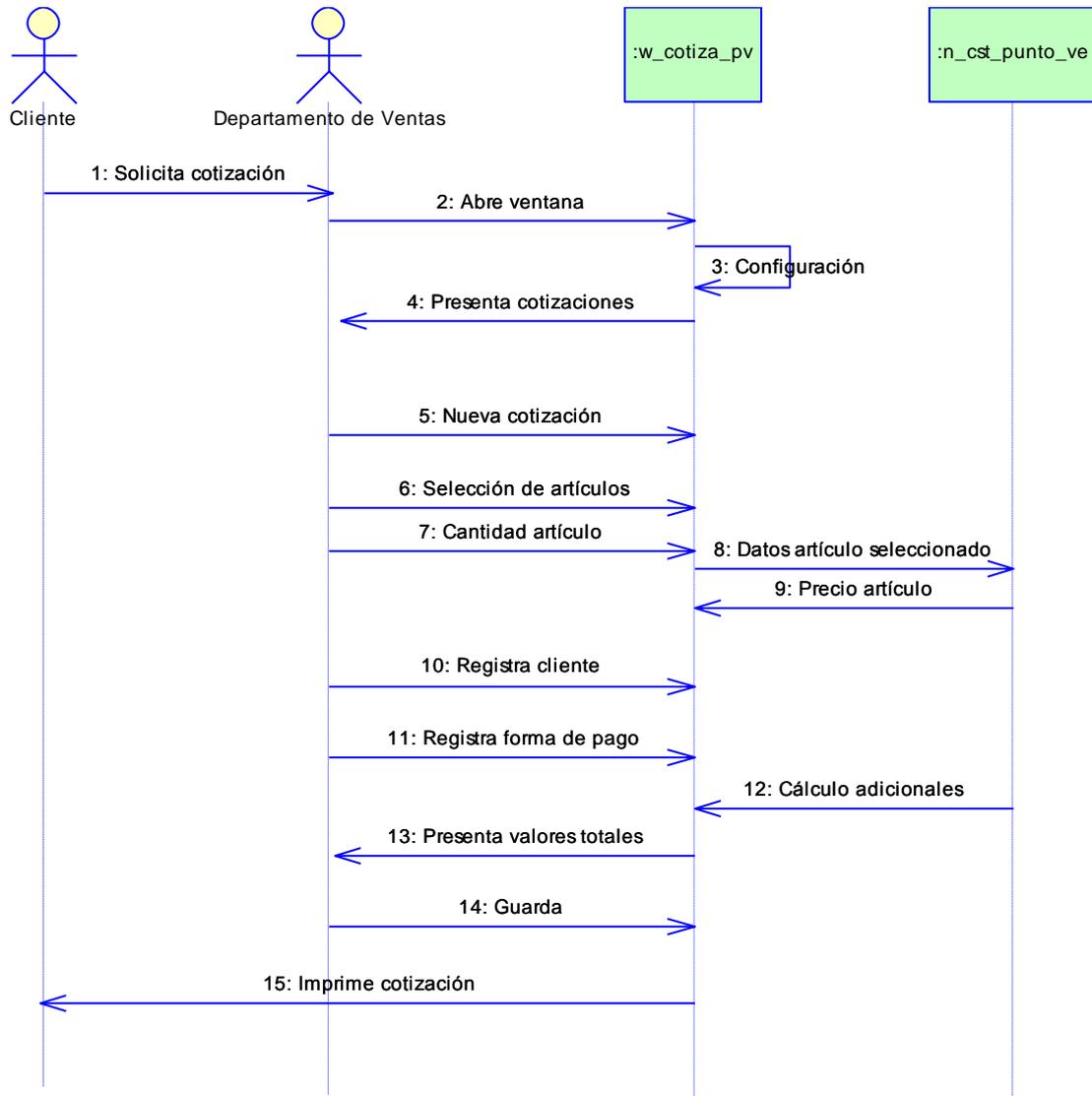


Fig. 4.57 Diagrama de secuencia (movimiento de cotizaciones)

4.2.5.10 Movimiento de pedidos

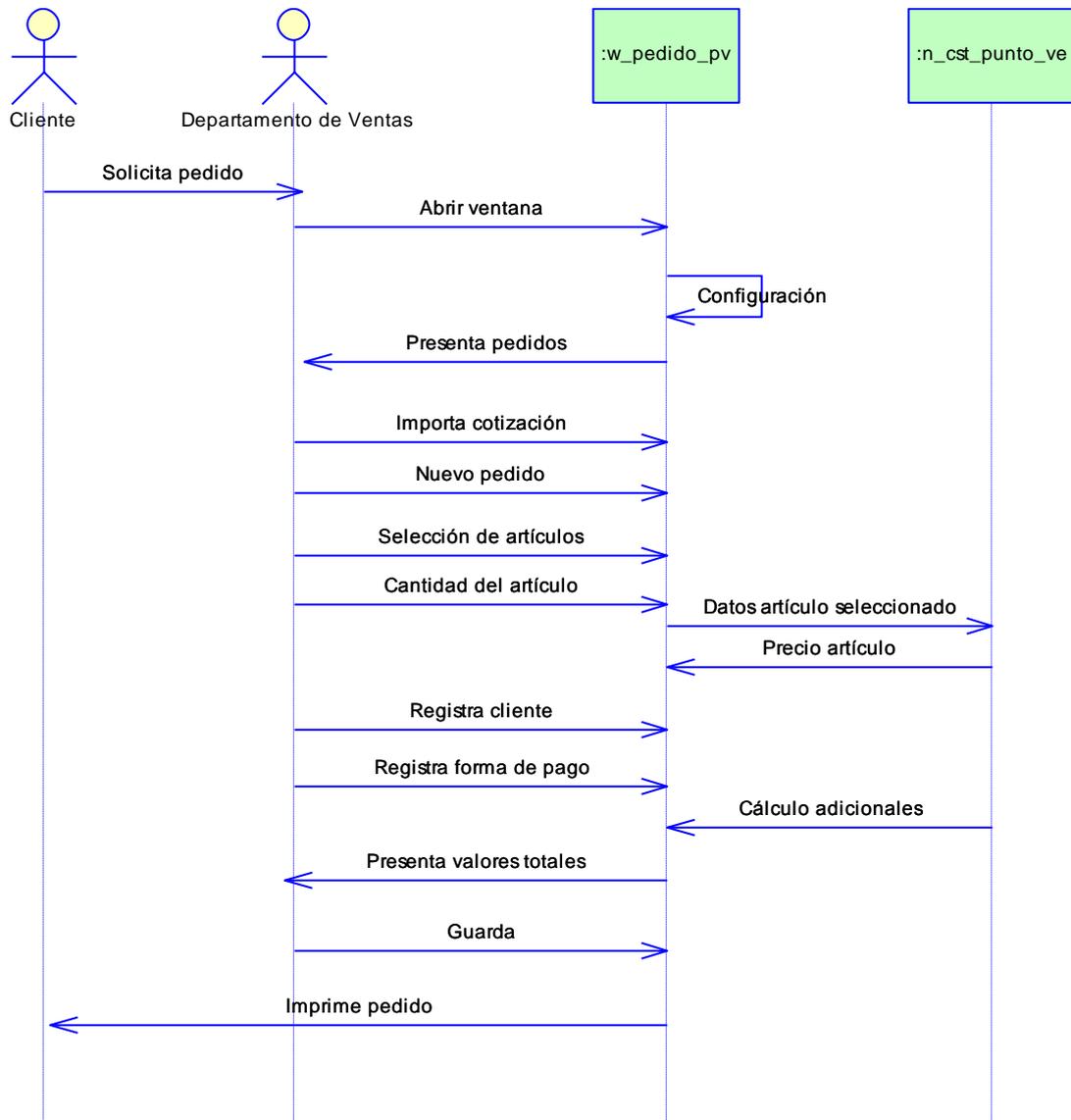


Fig. 4.58 Diagrama de secuencia (movimiento de pedidos)

4.2.5.11 Movimiento de facturas

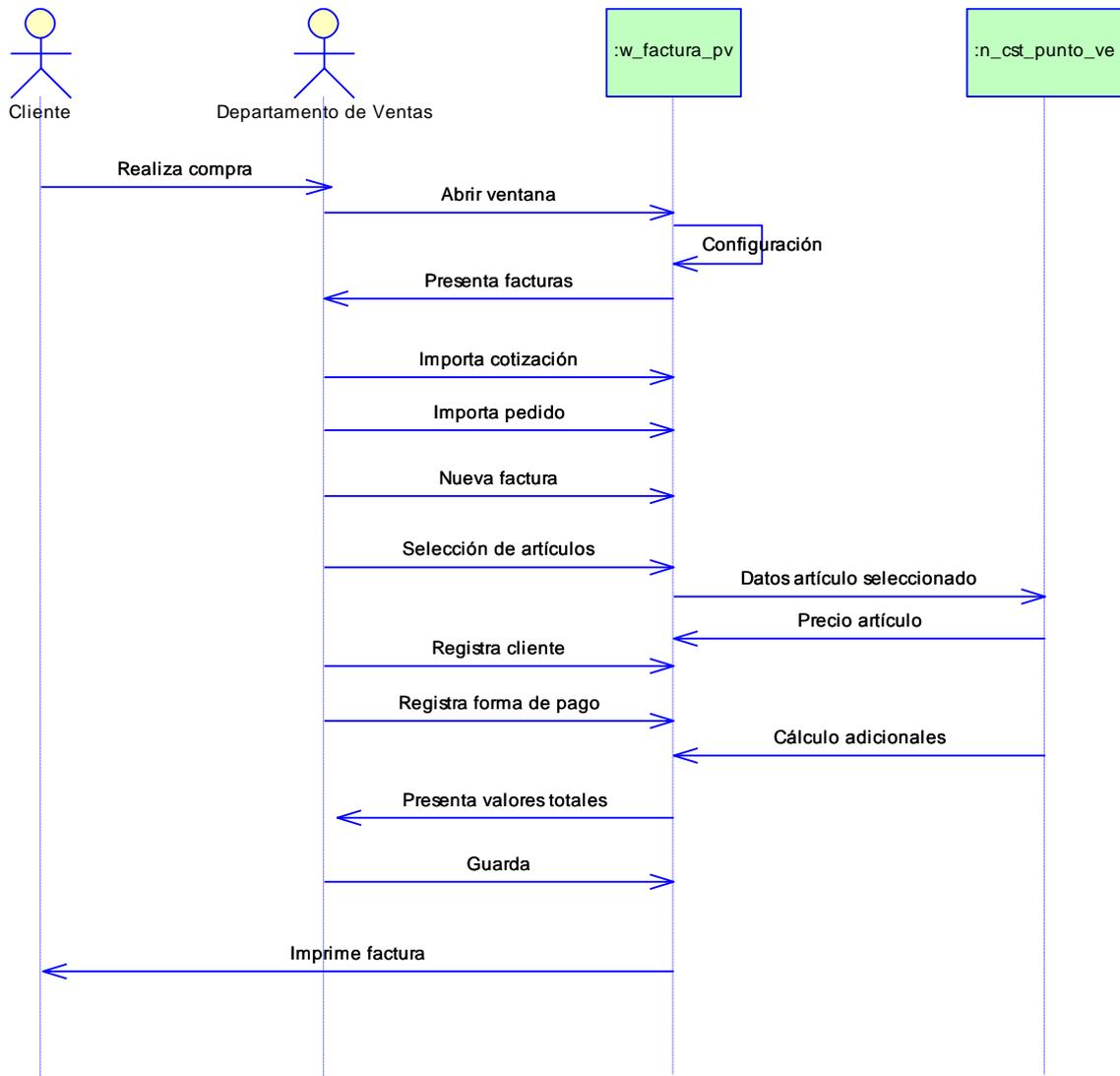


Fig. 4.59 Diagrama de secuencia (movimiento de facturas)

4.2.5.12 Movimiento de Puntos de venta

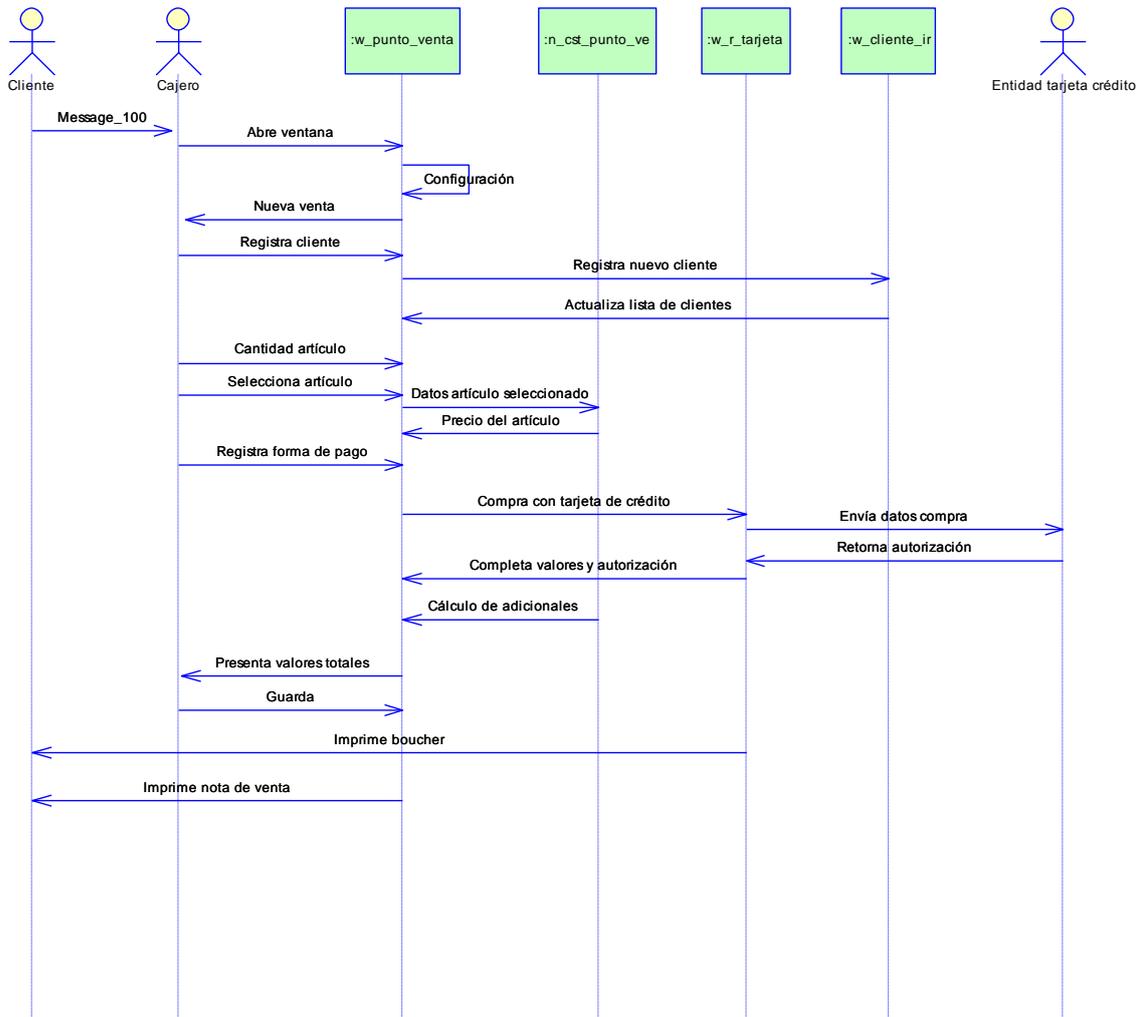


Fig. 4.60 Diagrama de secuencia (movimiento de puntos de venta)

4.2.6 Diagrama de actividades (intranet)

Muestra el flujo entre los objetos. Presenta el funcionamiento del sistema y el flujo de control en las actividades que cumplirá el módulo de puntos de venta.

4.2.6.1 Mantenimiento genérico

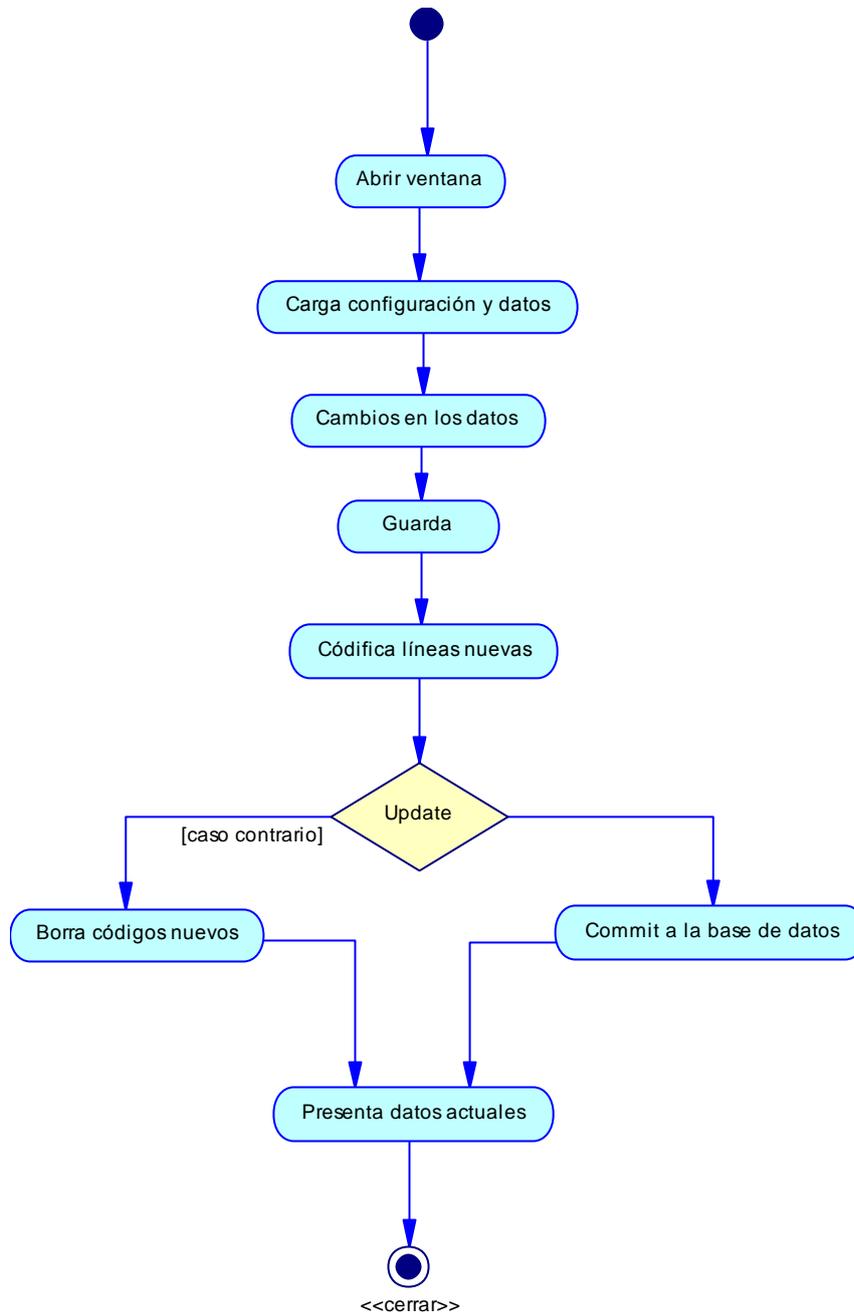


Fig. 4.61 Diagrama de actividades (mantenimiento general)

4.2.6.2 Mantenimiento del cliente

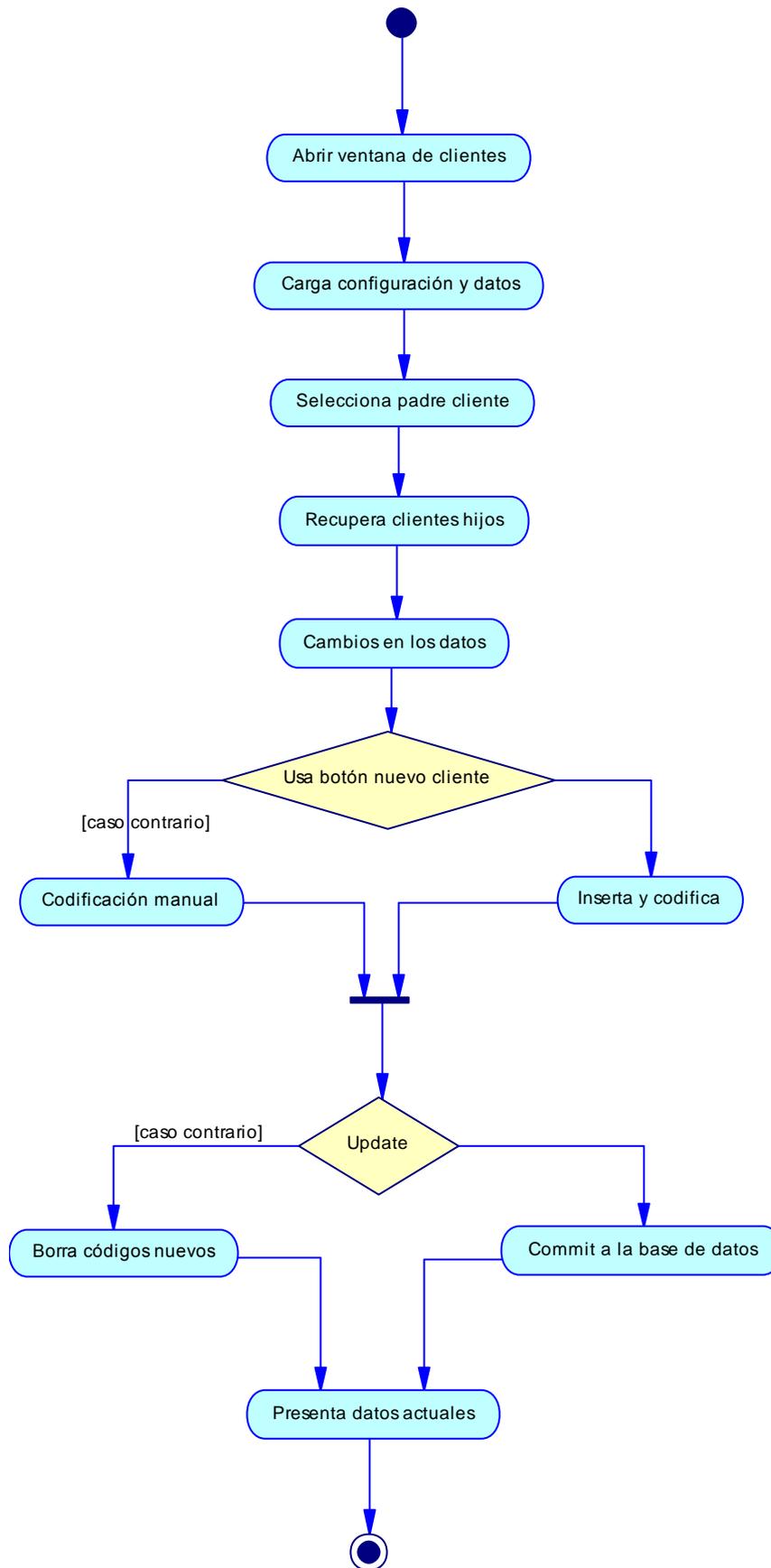


Fig. 4.62 Diagrama de actividades (mantenimiento de cliente)

4.2.6.3 Mantenimiento del descuento

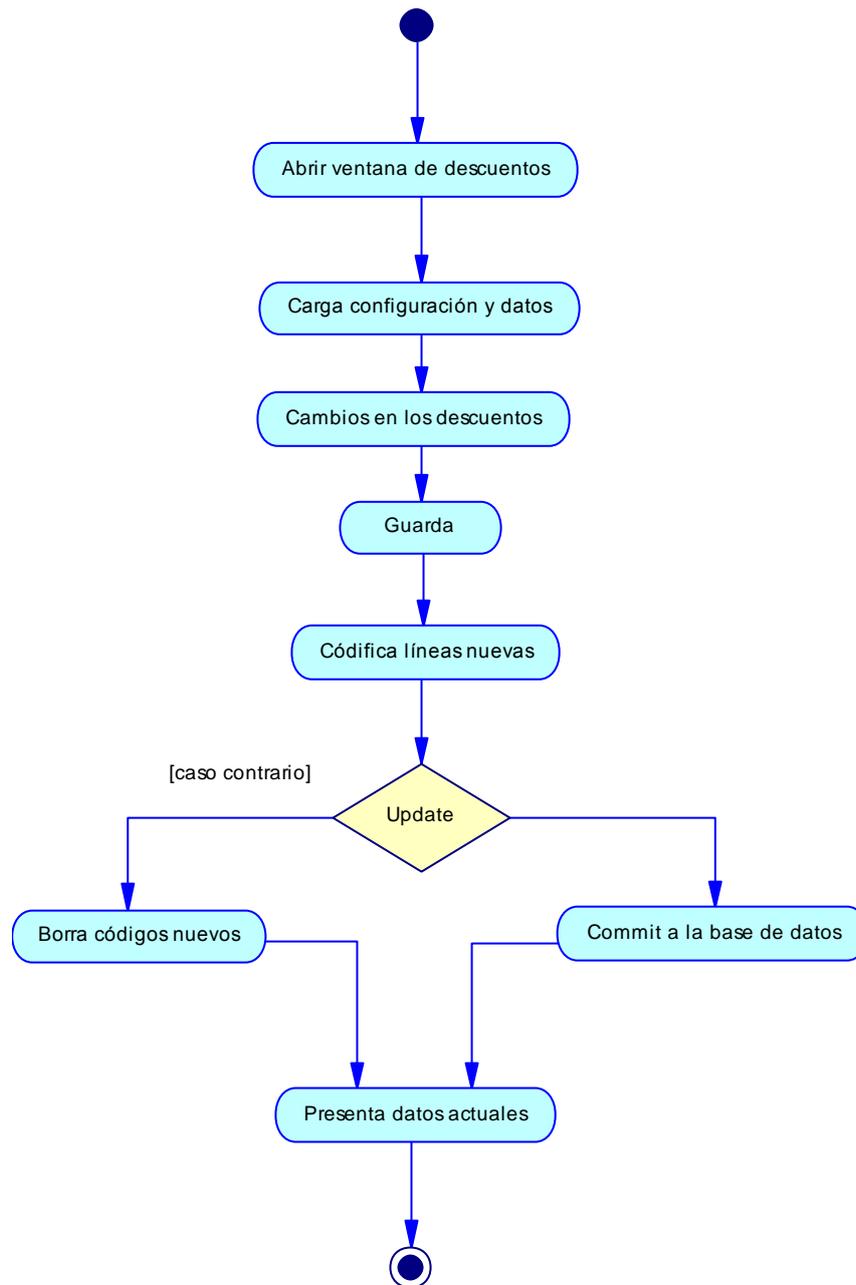


Fig. 4.63 Diagrama de actividades (mantenimiento de descuento)

4.2.6.4 Mantenimiento de la forma de envío

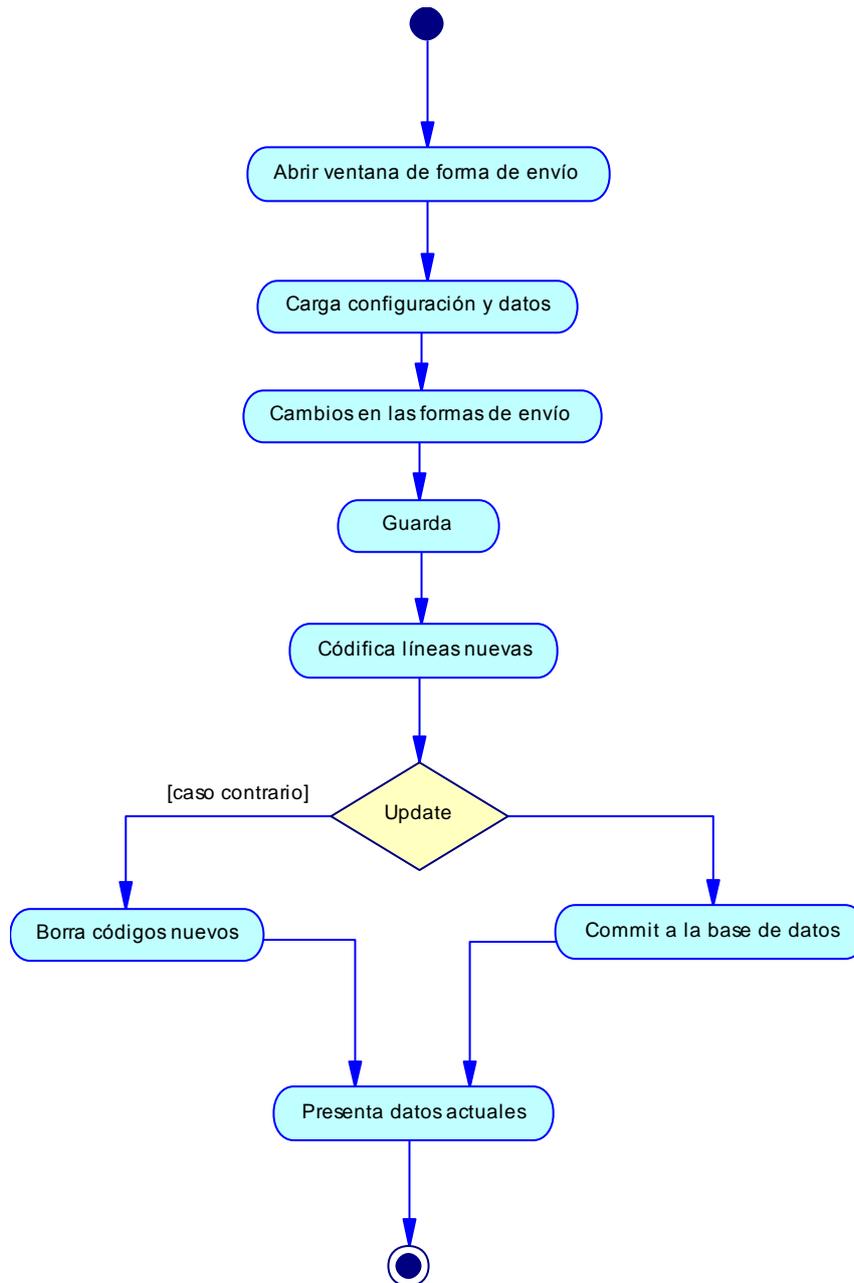


Fig. 4.64 Diagrama de actividades (mantenimiento de forma de envío)

4.2.6.5 Mantenimiento del periodo de venta

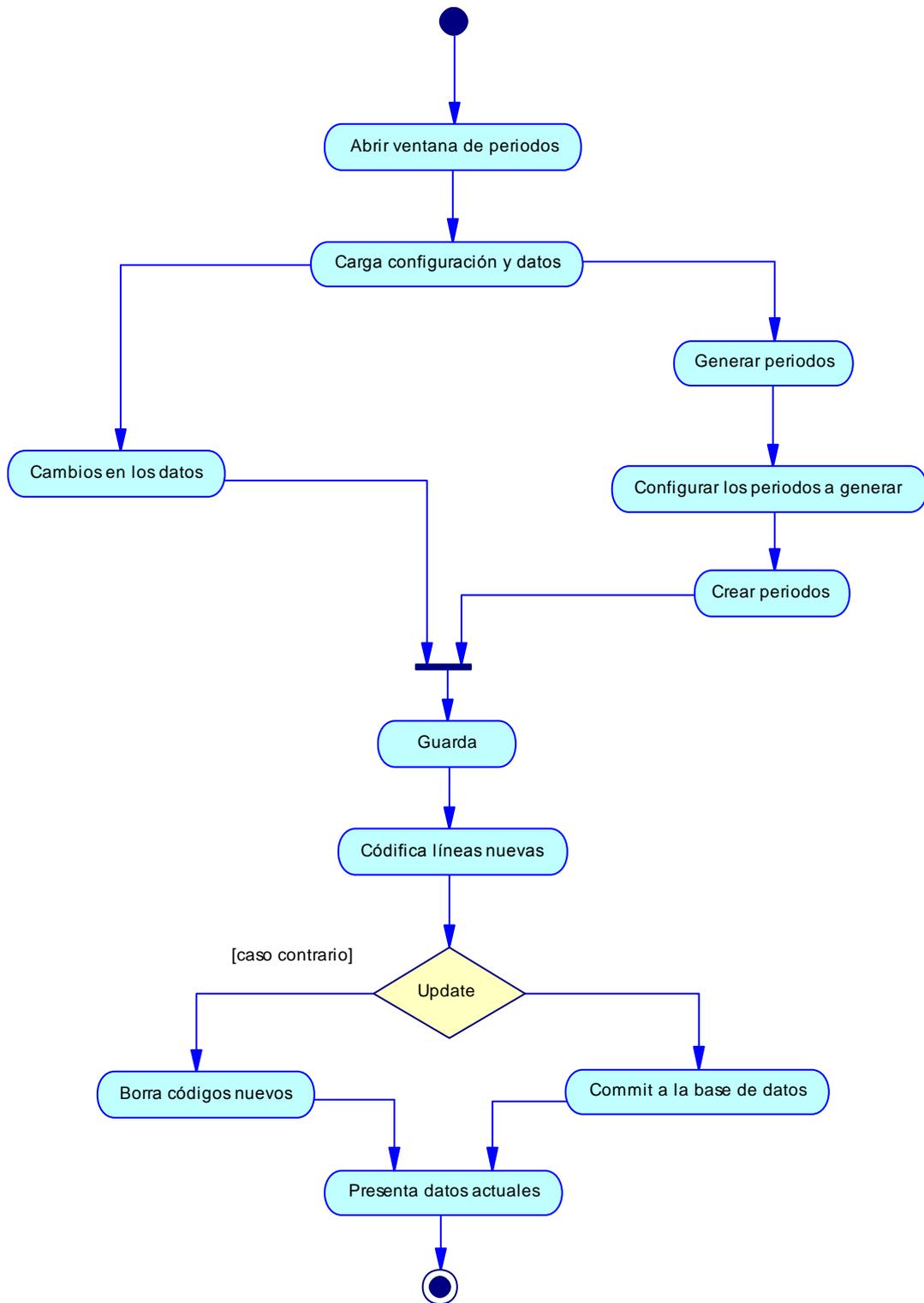


Fig. 4.65 Diagrama de actividades (mantenimiento de periodo de venta)

4.2.6.6 Mantenimiento del precio

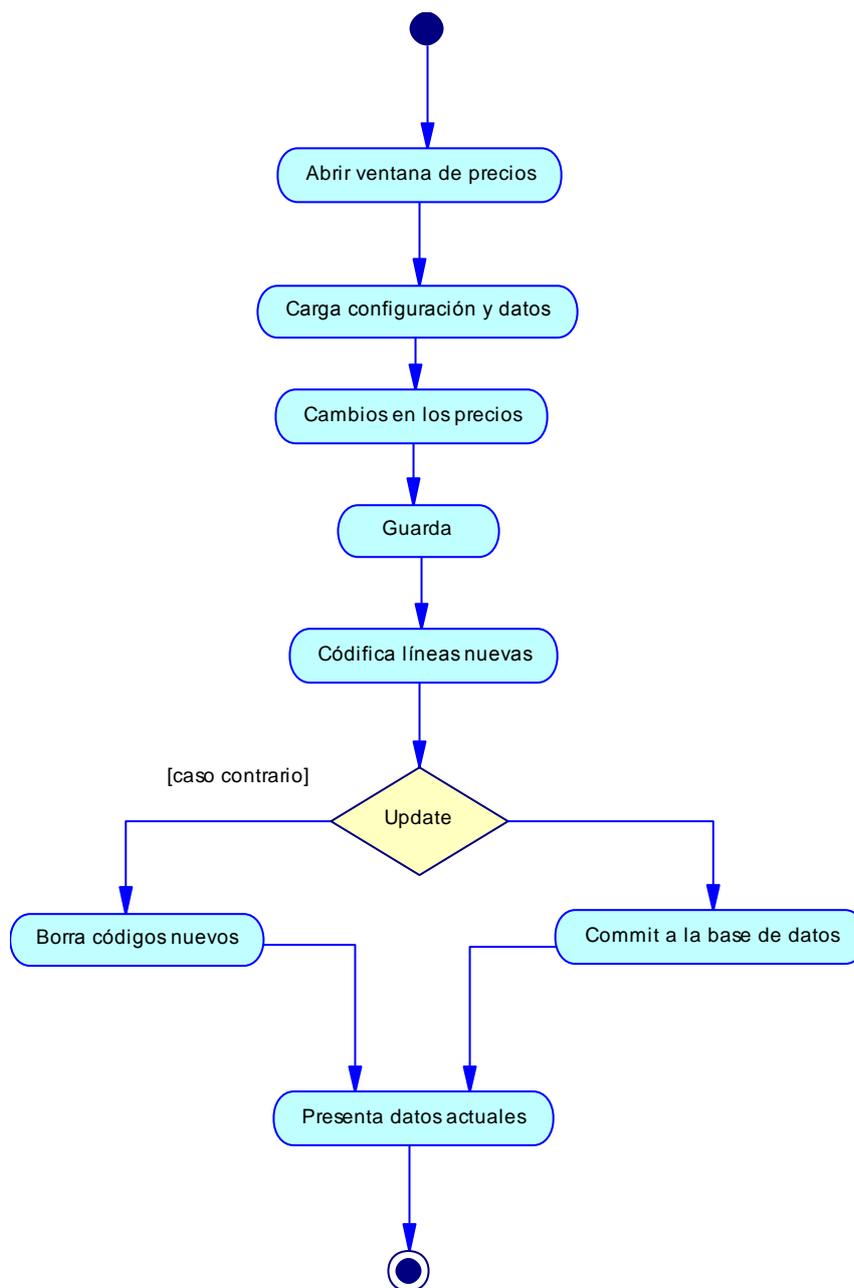


Fig. 4.66 Diagrama de actividades (mantenimiento del precio)

4.2.6.7 Mantenimiento del vencimiento de clientes

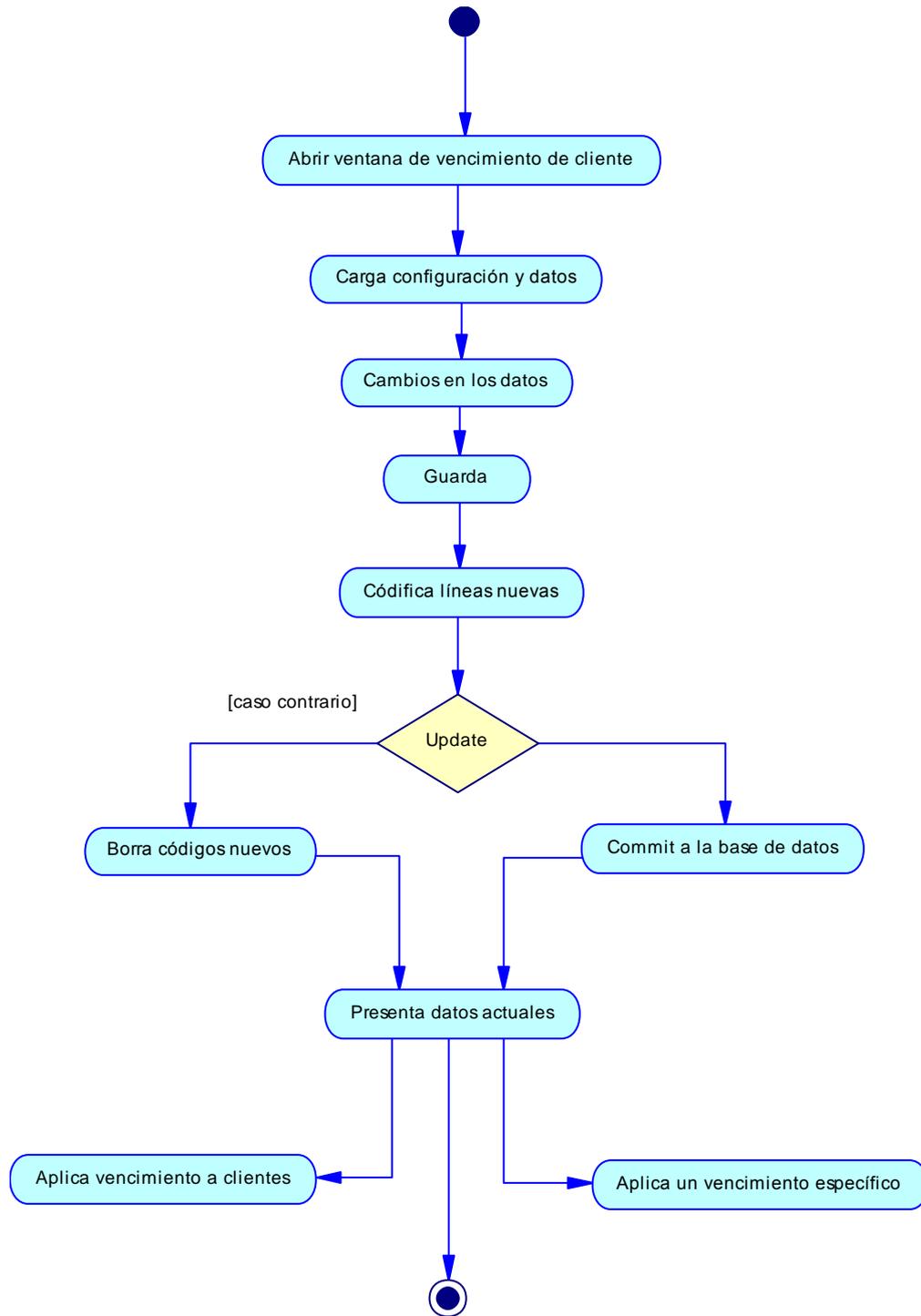


Fig. 4.67 Diagrama de actividades (mantenimiento del vencimiento de clientes)

4.2.6.8 Mantenimiento del vendedor

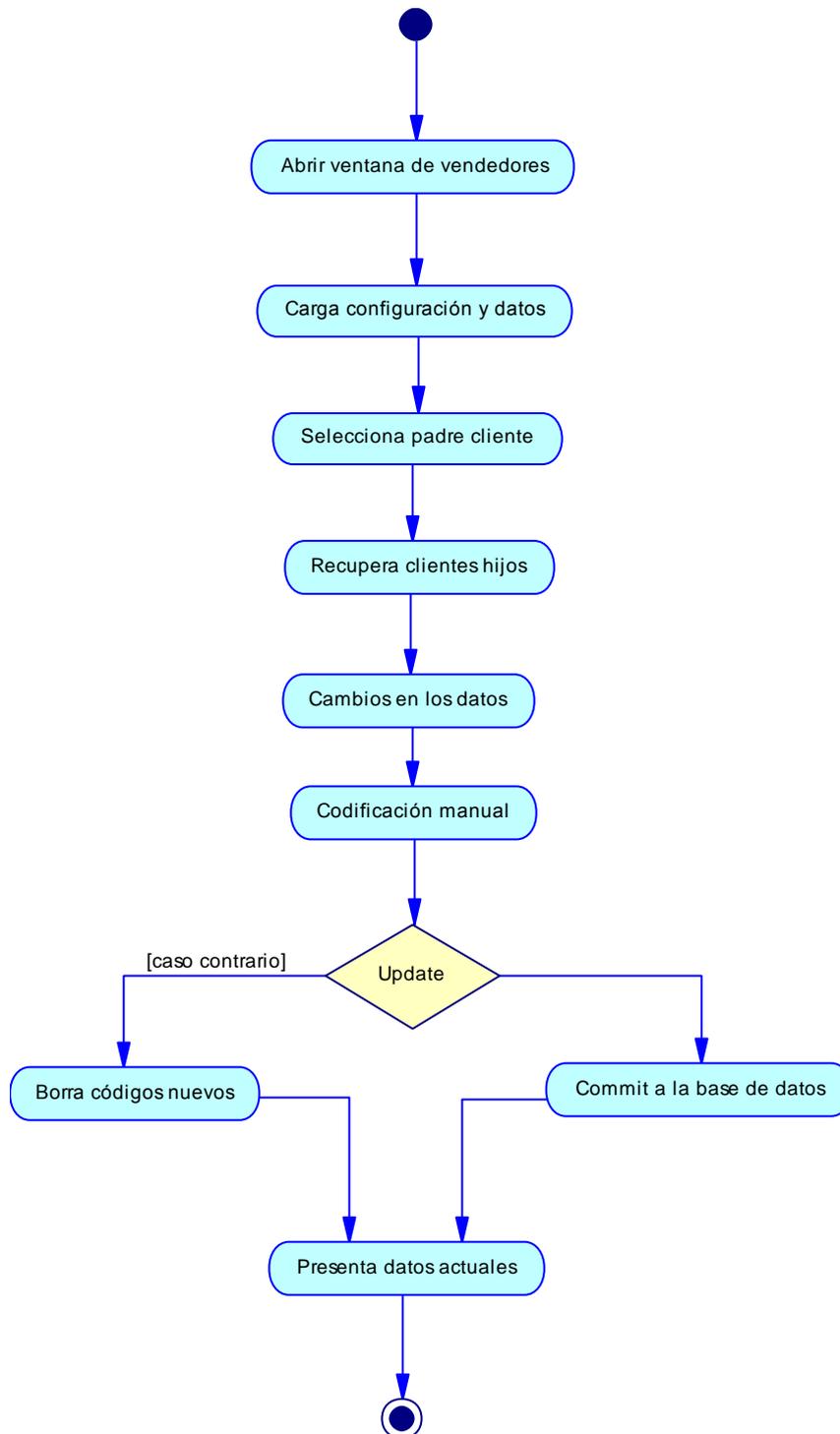


Fig. 4.68 Diagrama de actividades (mantenimiento del vendedor)

4.2.6.9 Movimiento de cotizaciones

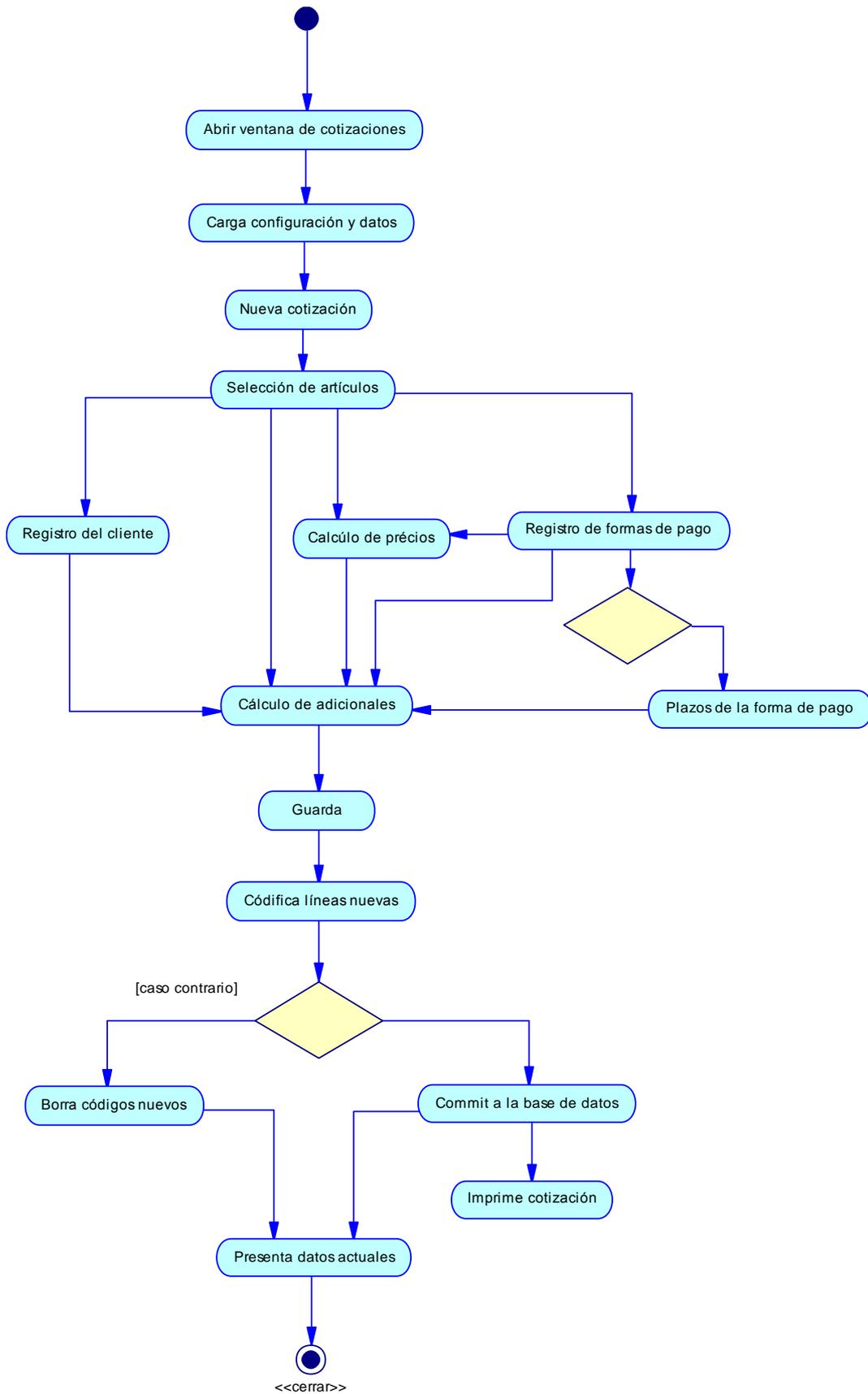


Fig. 4.69 Diagrama de actividades (transacción de cotizaciones)

4.2.6.10 Movimiento de pedidos

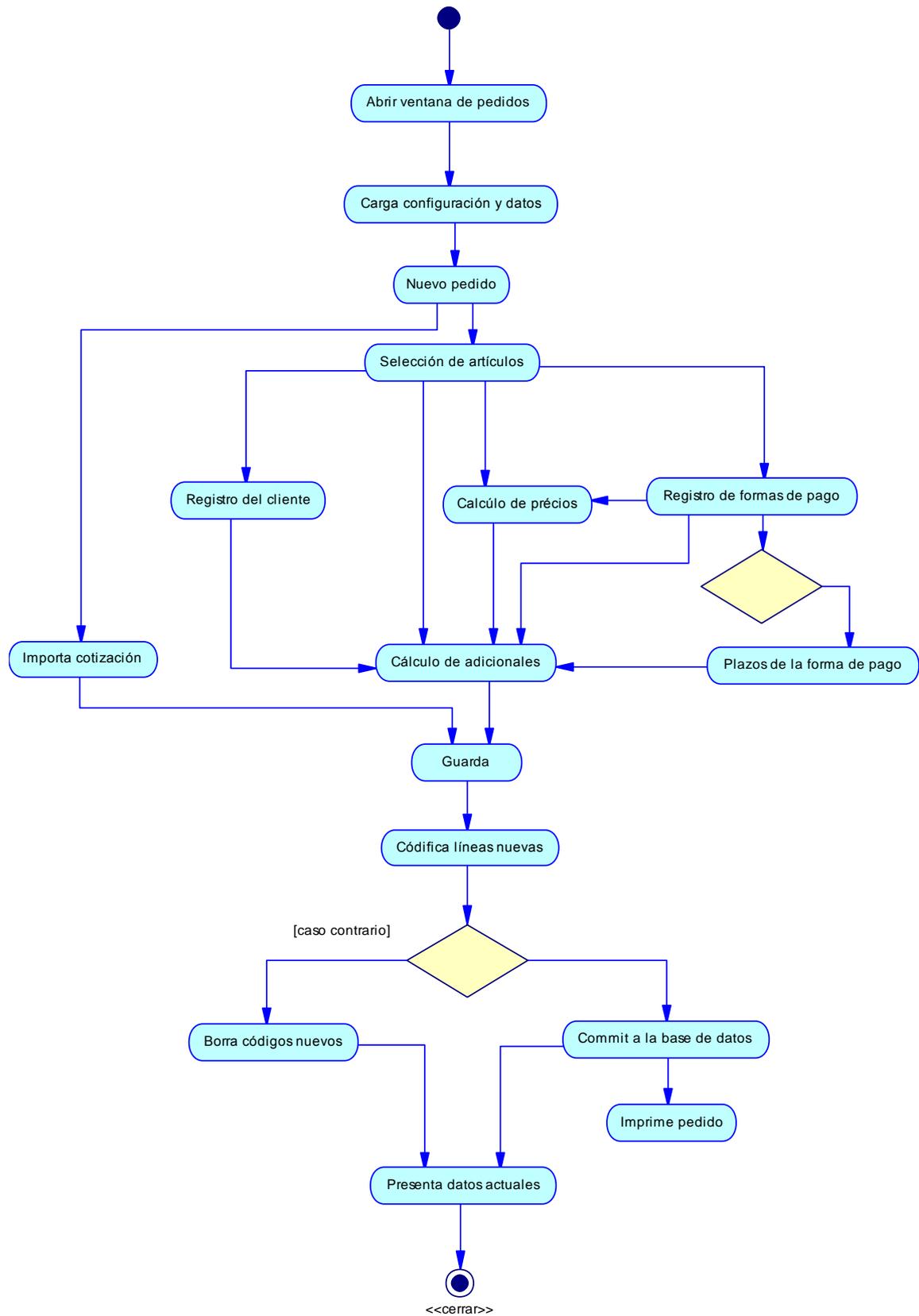


Fig. 4.70 Diagrama de actividades (transacción de pedidos)

4.2.6.11 Movimiento de facturas

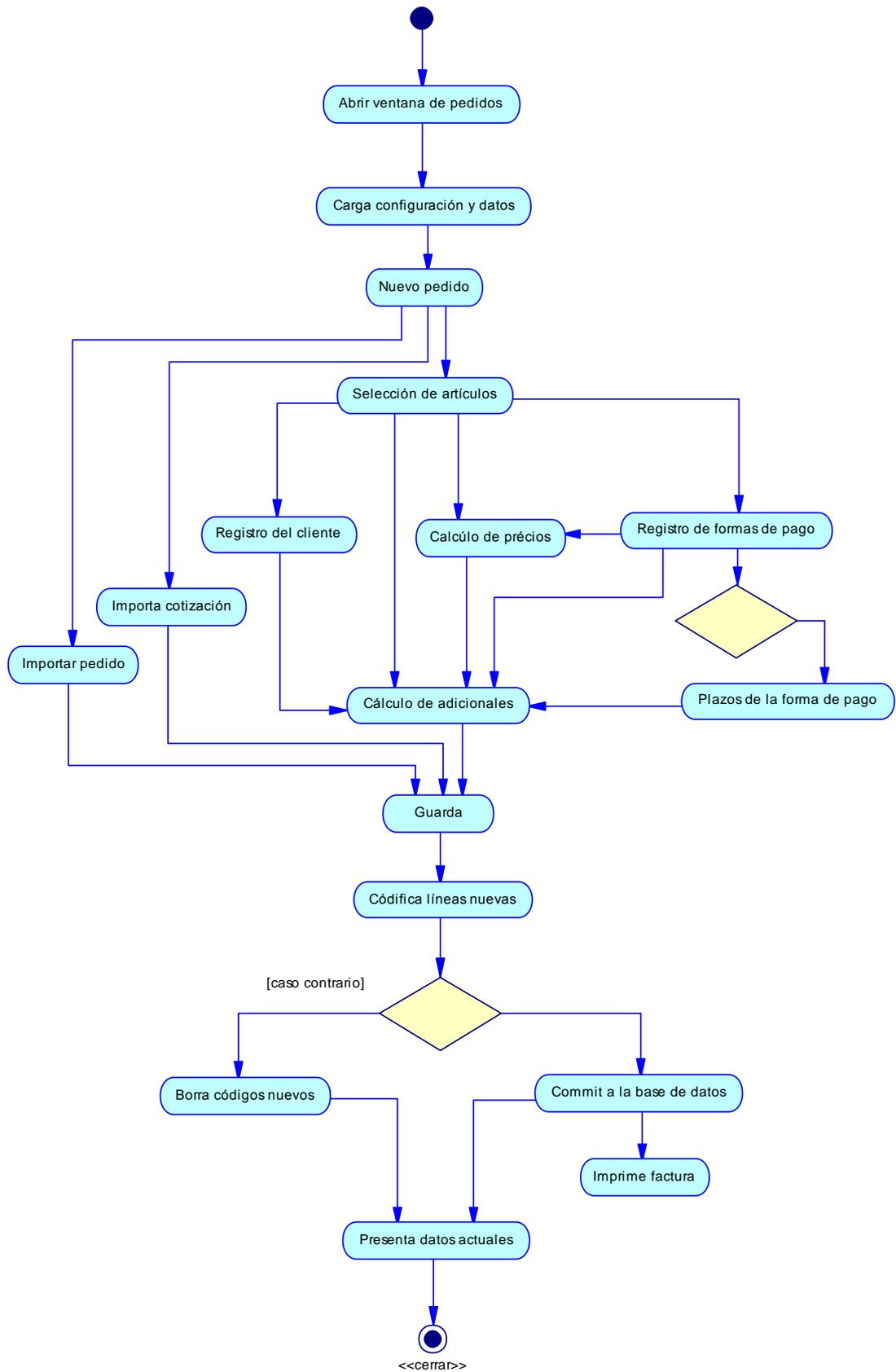


Fig. 4.71 Diagrama de actividades (transacción de facturas)

4.2.6.12 Movimiento de Puntos de venta

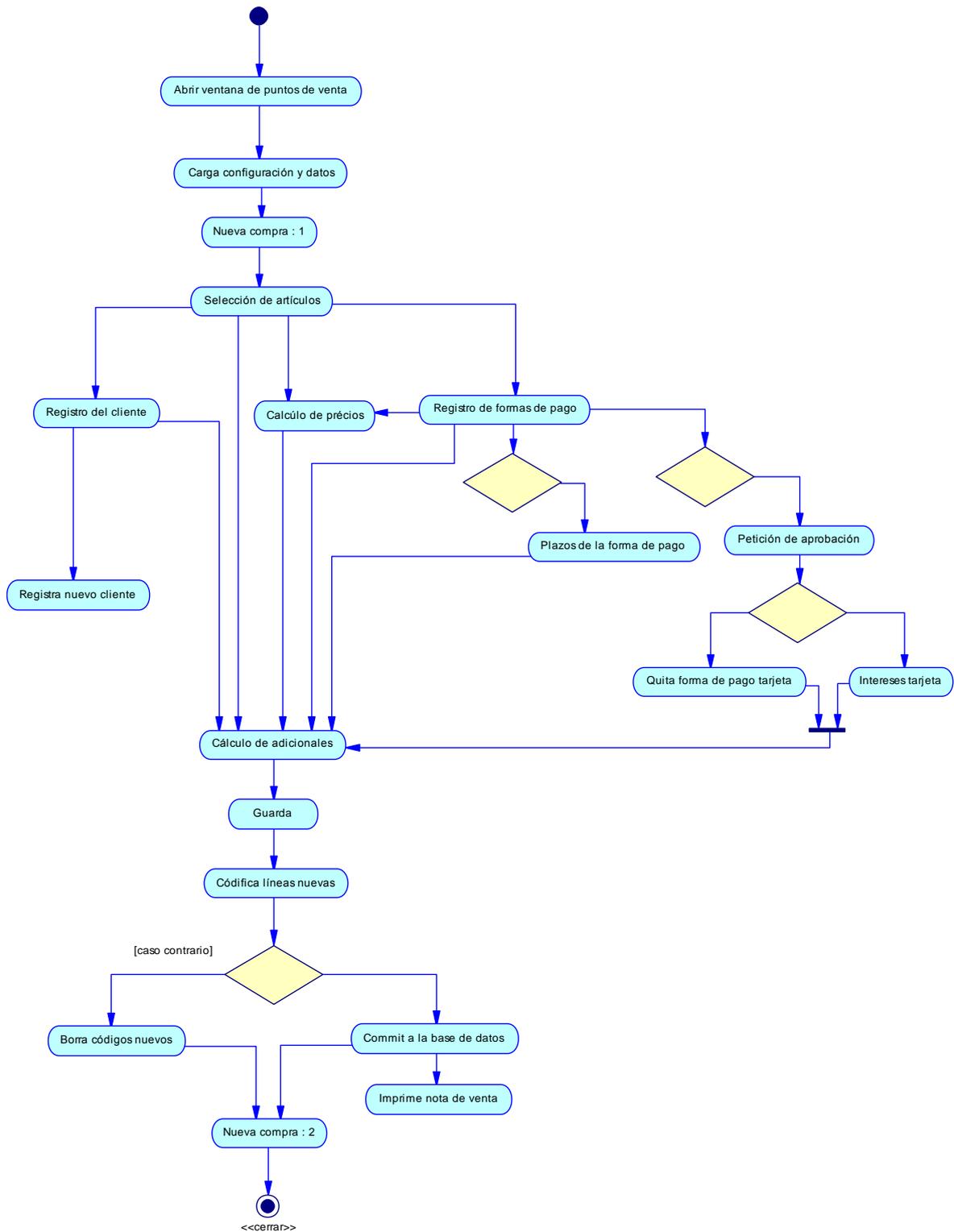


Fig. 4.72 Diagrama de actividades (transacción de puntos de venta)

4.2.7 Diagrama de casos de uso (Internet)

Los diagramas de casos de uso para la parte de Internet en su mayoría son iguales a los de intranet

4.2.8 Diagrama de clases (Internet)

El diagrama de clases para Internet presenta todas las clases creadas en Java para el manejo del punto de venta.

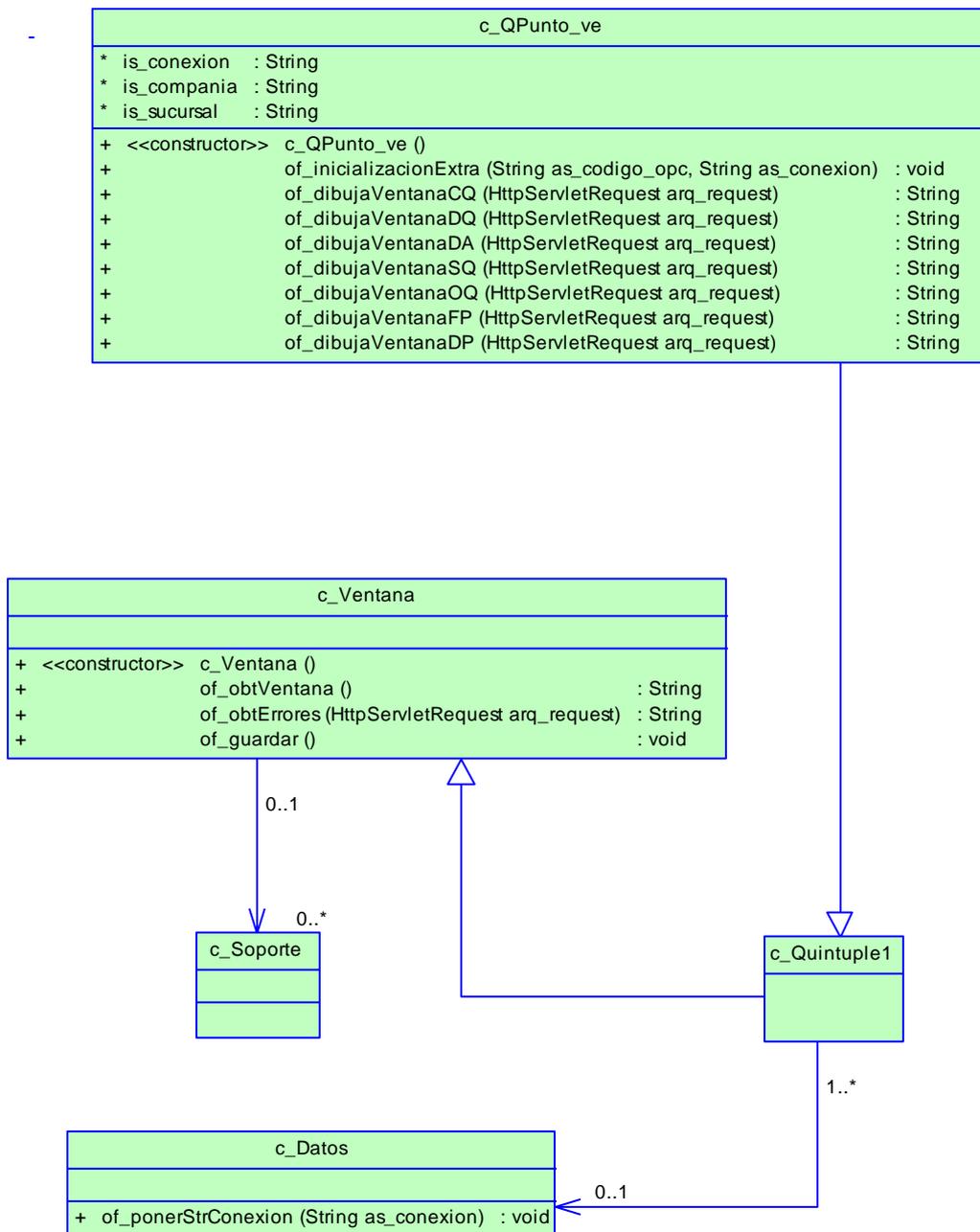


Fig. 4.73 Diagrama de clases (Punto de Venta Internet)

4.2.9 Diagrama de objetos (visuales, no visuales) para Internet

Este diagrama presenta los objetos no genéricos que serán los encargados de presentar las páginas para las transacciones.

4.2.9.1 Página w_qpunto_ve

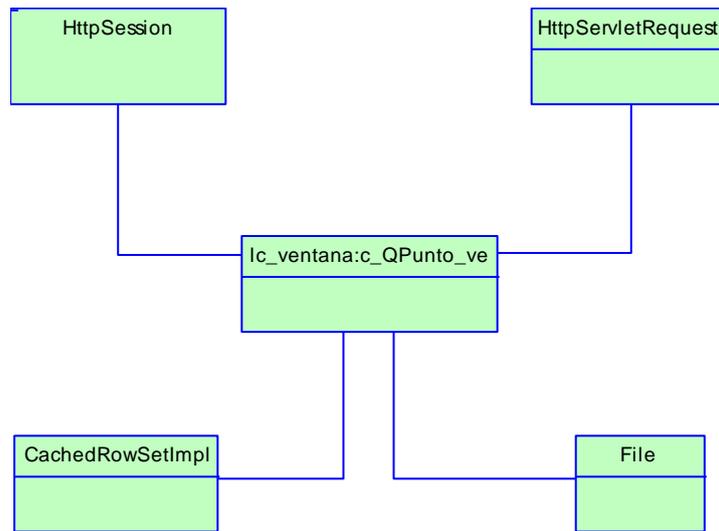


Fig. 4.74 Diagrama de objetos (página w_punto_ve)

4.2.9.2 Página DirectPaymentReceipt

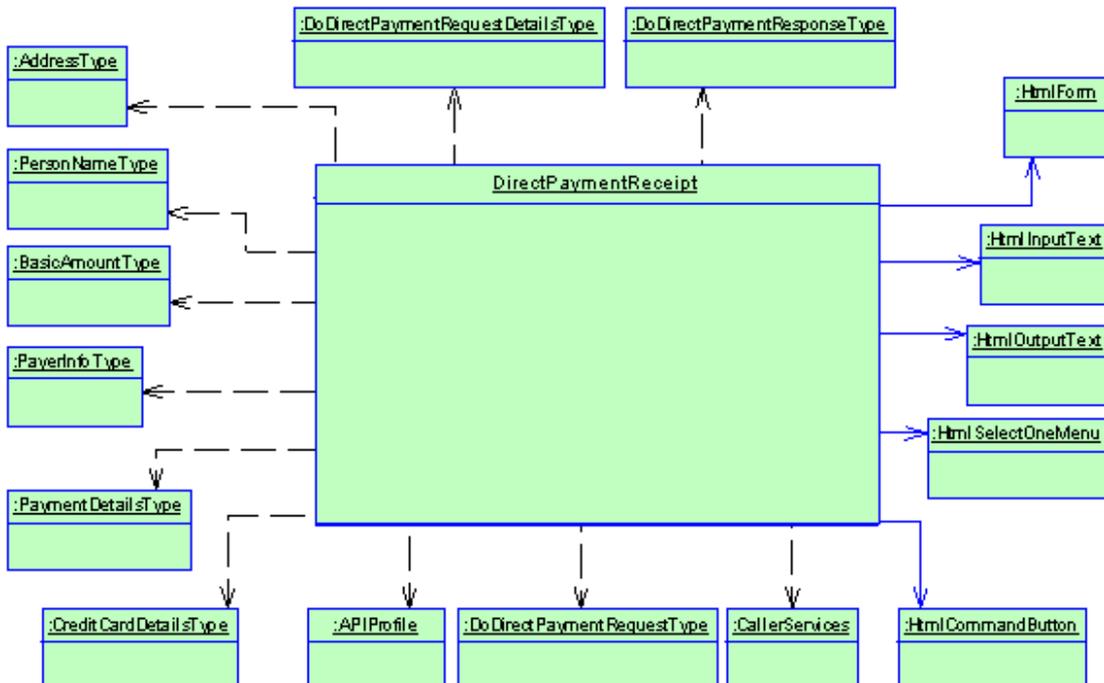


Fig. 4.75 Diagrama de objetos (página DirectPaymentReceipt)

4.2.9.3 Página Mantenimiento

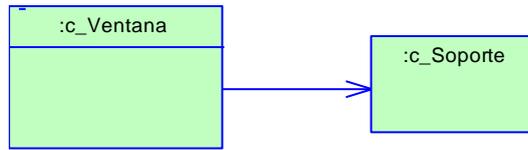


Fig. 4.76 Diagrama de objetos (página mantenimiento)

4.2.10 Diagrama de secuencia (Internet)

Los diagramas de secuencia para Internet son iguales que los diagramas de intranet, a excepción del diagrama en la página de compras.

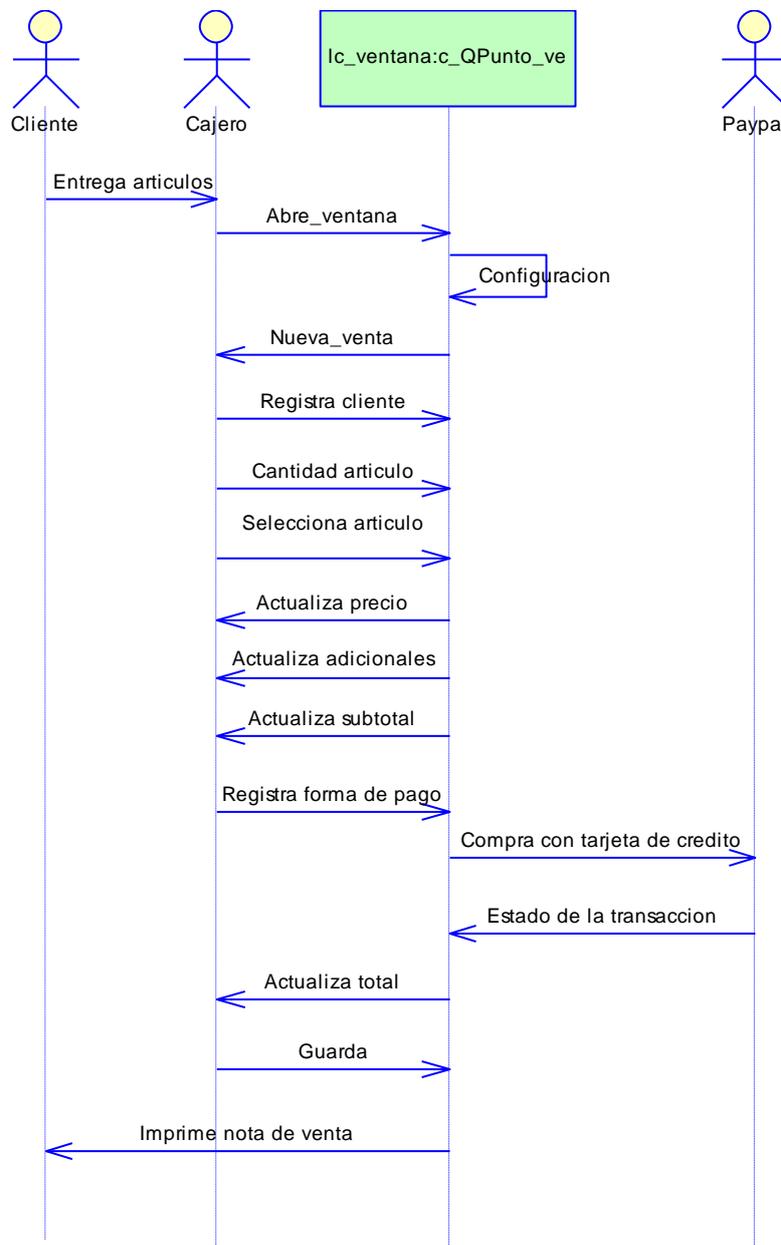


Fig. 4.77 Diagrama de secuencia (página de compras)

4.3 MODELO FÍSICO DE DATOS (Diccionario de datos anexo G)

El modelo físico de datos presenta el modelo lógico con tipos de datos, relaciones y claves foráneas hechas para una base de datos específica, en este caso hecha para Sybase Anywhere.

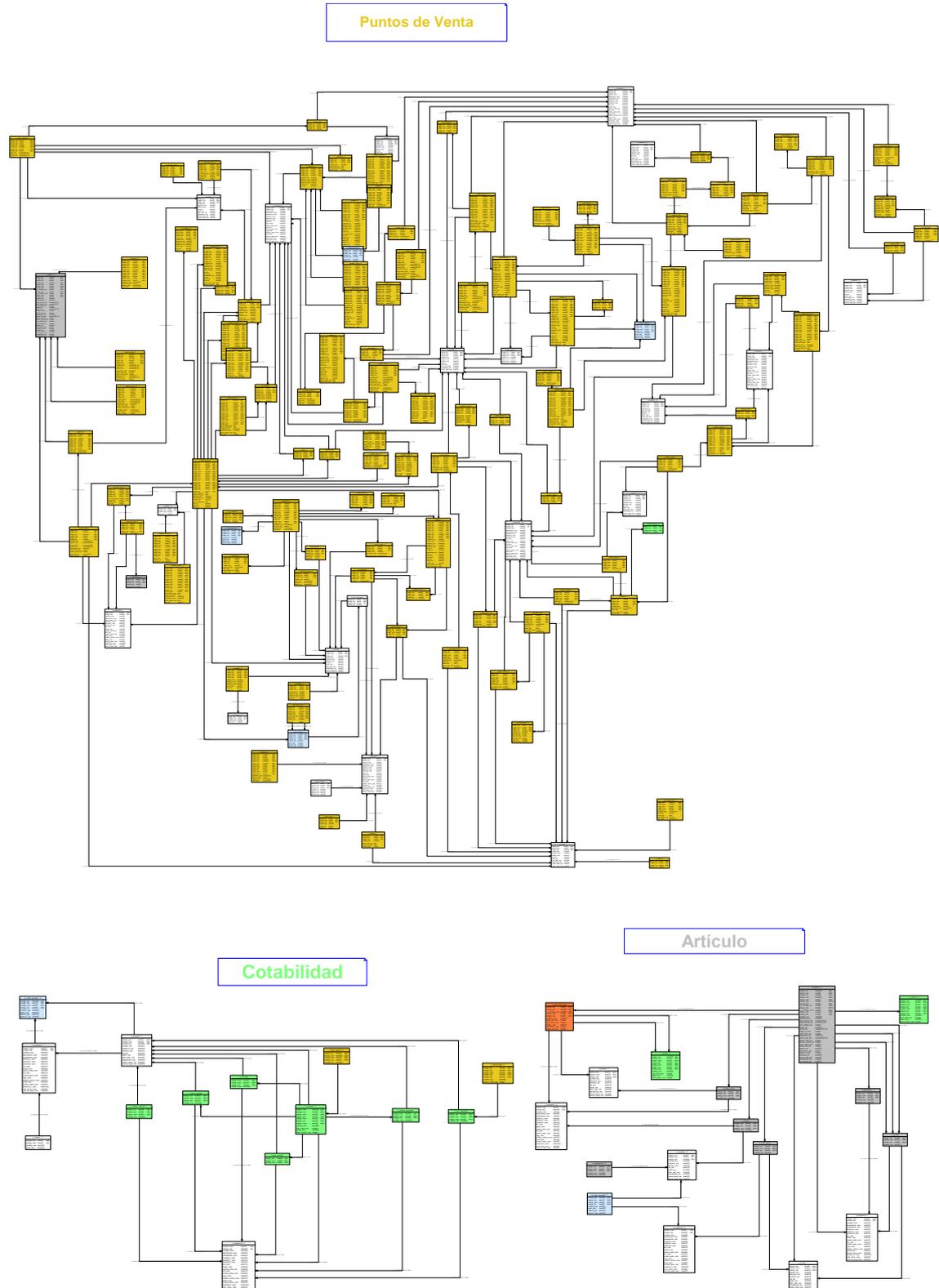


Fig. 4.1 Diagrama físico de datos

CAPÍTULO V

5 IMPLEMENTACIÓN DEL MÓDULO

Este capítulo presenta la programación que se realizó, detallando el contenido del nuevo módulo además de los cambios que se realizaron, siempre tomando como base el diseño establecido en el capítulo anterior. Para finalizar la implementación este capítulo presenta la documentación de las pruebas realizadas, los resultados de dichas pruebas y los cambios que se realizaron en base a los resultados.

5.1 DESARROLLO DEL CÓDIGO

El código se desarrolla para Internet y para intranet, por lo que siguiendo con los estándares de Venture para la parte de intranet se utiliza el programa Power Builder 8.0.3 y para la parte de Internet se utiliza Java. Para cada uno de estos caminos encontramos un programa genérico el cual debe acoger al nuevo módulo "Puntos de venta". El nuevo módulo a desarrollarse depende de todas las funciones básicas del genérico. Es así que son las funciones especiales las que se desarrollarán, tanto en el objeto de funciones hecho para este módulo como en las mismas ventanas específicas.

5.1.1 Descripción del código para intranet (Power Builder)

A continuación se describe todo el módulo de puntos de venta hecho en Power Builder, este se desarrolla dentro de la PBL llamada punto_ve.

PBL punto_ve: Descripción General

Tamaño	1108 Kb
Ruta	\CODIGO\punto_ve.pbl
Comentarios	Esta librería de Power Builder contiene toda la programación desarrollada para el módulo Puntos de venta.

Tabla 5.1 Descripción general de la librería punto_ve

PBL punto_ve: Dataobject

Nombre	Comentario
d_adi_coti_pv	Presenta la información y el formato necesario para mostrar y modificar los valores adicionales en una cotización. Este dataobject es usado en la ventana w_cotiza_pv.
d_adi_fact_pv	Presenta la información y el formato necesario para mostrar y modificar los valores adicionales en una factura. Este dataobject es usado en la ventana w_factura_pv.
d_adi_pedi_pv	Presenta la información y el formato necesario para mostrar y modificar los valores adicionales en un pedido. Este dataobject es usado en la ventana w_pedido_pv.
d_adicional_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_adicional.
d_busca_monto	Trae la información necesaria para especificar el código del monto de venta que aplica.
d_busca_rplazo	Trae la información necesaria para especificar el código del plazo de pago que aplica.
d_busca_temp	Trae la información necesaria para especificar el código de la temporada que aplica.
d_busca_volumen	Trae la información necesaria para especificar el código del volumen de venta que aplica.
d_cab_coti_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la cabecera en una cotización. Este dataobject es usado en la ventana w_cotiza_pv.
d_cab_fact_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la cabecera de una factura. Este dataobject es usado en la ventana

w_factura_pv.

d_cab_pedi_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la cabecera de un pedido. Este dataobject es usado en la ventana w_pedido_pv.
d_cliente_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_cliente.
d_control_plazo	Es un dataobject externo, que almacena el código y los días plazo de todas las formas de pagos que ocupen plazos para efectuar el pago.
d_cupo_vend_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_cupo_vendedor.
d_dddw_cliente_pv	Drop down específico utilizado en los dataobject que tienen el campo codigo_cle.
d_dddw_mventa_pv	Drop down específico utilizado en los dataobject que tienen el campo codigo_mve.
d_dddw_tempor_pv	Drop down específico utilizado en los dataobject que tienen el campo codigo_tem.
d_descuento_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_descuento.
d_det_coti_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del detalle en una cotización. Este dataobject es usado en la ventana w_cotiza_pv.
d_det_fact_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del detalle de una factura. Este dataobject es usado en la ventana w_factura_pv.
d_det_pedi_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del detalle de un pedido. Este dataobject es usado en la ventana w_pedido_pv.

d_detpa_coti_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del detalle del pago en una cotización. Este dataobject es usado en la ventana w_cotiza_pv.
d_detpa_fact_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del detalle del pago de una factura. Este dataobject es usado en la ventana w_factura_pv.
d_detpa_pedi_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del detalle del pago de un pedido. Este dataobject es usado en la ventana w_pedido_pv.
d_entidad_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_entidad_pago.
d_fo_pa_tarj_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_for_pag_tarje.
d_pago_coti_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del pago en una cotización. Este dataobject es usado en la ventana w_cotiza_pv.
d_pago_fact_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del pago de una factura. Este dataobject es usado en la ventana w_factura_pv.
d_pago_pedi_pv	Presenta la información y el formato necesario para mostrar y modificar los valores del pago de un pedido. Este dataobject es usado en la ventana w_pedido_pv.
d_periodo_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_periodo.
d_precio_pv	Presenta la información y el formato necesario para

mostrar y modificar los valores de la tabla j_precio.

d_sel_cab_cot	Recupera y presenta la información y el formato necesario para mostrar y seleccionar una cotización específica.
d_suc_client_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_sucurs_client.
d_temporada_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_temporada.
d_valor_desc	Recupera la información necesaria para a través de filtros llegar a la regla de descuento deseada.
d_vencimient_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_vencimiento.
d_vendedor_pv	Presenta la información y el formato necesario para mostrar y modificar los valores de la tabla j_vendedor.

Tabla 5.2 Dataobject's en la librería punto_ve

PBL punto_ve: Estructuras

Name	Comment
s_ret_adicio	Permite retornar el valor de cada adicional que cumple con los filtros para un artículo determinado.
s_ret_precio	Permite retornar el precio de un artículo después de que se ejecuta la función para recuperar el precio del artículo.

Tabla 5.3 Estructuras en la librería punto_ve

PBL punto_ve: Objetos del usuario

Name	Comment
n_cst_punto_ve	Objeto no visible donde se guardan las funciones especiales que sirven para todo el módulo.

Funciones

of_adicio_art

Calcula los valores adicionales por artículo, devuelve el valor del descuento.

Parámetros:

s_ret_adicio: astr_adicio

Dato por referencia para devolver los resultados de la función.

string: as_art

Dato por valor que contiene el código del artículo a analizar.

string: as_cle

Dato por valor que contiene el código del cliente que realiza la transacción.

string: aas_fpo[]

Dato por valor, arreglo, que contiene los códigos de las formas de pago.

long: al_plazo

Dato por valor, que contiene el día máximo de todas las formas de pago.

decimal: adc_cantidad

Dato por valor, que contiene la cantidad que se pide del artículo.

decimal: adc_precio2

Dato por valor, que contiene el valor después del descuento individual.

decimal: adc_monto

Dato por valor, que contiene el valor monetario que alcanzó el total de los artículos.

boolean: ab_per_desc

Dato por valor, que contiene el valor que indica si se permiten descuentos.

boolean: ab_acumula

Dato por valor, que contiene el valor

of_precio_art que indica si los descuentos que entraron en el filtro se acumulan.
 Calcula el precio y el posible descuento individual por cada artículo.

Parámetros:

s_ret_precio: astr_ret
 Dato por referencia para devolver los resultados de la función.
 string: as_codigo_cle
 Dato por valor que contiene el código del cliente que realiza la transacción.
 string: as_codigo_art
 Dato por valor que contiene el código del artículo a analizar.
 decimal: adc_cantidad
 Dato por valor, que contiene la cantidad que se pide del artículo.

Tabla 5.4 Objetos del usuario en la librería punto_ve

PBL punto_ve: Ventanas

Name	Comment
w_cliente_ir	Ventana que permite ingresar un cliente en forma rápida desde la ventana de puntos de venta.
W_cliente_pv	Ventana que permite visualizar y modificar los datos del cliente.
W_cotiza_pv	Ventana heredada de w_ventas_pv, que permite realizar cotizaciones.
W_datafast	Ventana que imita el funcionamiento del lector de tarjetas de crédito, tanto para el servicio de datafast como para el de mediante.
w_descuento_pv	Ventana que permite visualizar y modificar las reglas para descuentos.
w_factura_pv	Ventana heredada de w_ventas_pv, que permite realizar ventas.

W_forma_envio	Ventana que permite establecer que tipos de envío tiene la empresa y las características del artículo que evalúa para determinar el precio del envío.
W_identifica	Ventana que permite la búsqueda de un cliente por identificación en la ventana w_punto_venta
w_pedido_pv	Ventana heredada de w_ventas_pv, que permite realizar pedidos.
w_periodo_pv	Ventana que permite visualizar y modificar los periodos.
w_precio_pv	Ventana que permite visualizar y modificar las reglas para los precios.
w_punto_venta	Ventana heredada de w_ventas_pv, utilizada para el punto de venta.
w_punto_venta_t	Ventana heredada de w_punto_venta, utilizada para el punto de venta pero con un teclado numérico para el manejo de una pantalla táctil.
w_r_tarjeta	Ventana child! que realiza la comunicación, envío y toma de la información con el datafast.
w_rep_p_venta	Ventana donde se despliegan los reportes del punto de venta.
w_vence_cle_pv	Ventana que permite visualizar y modificar los rangos de días para dar de baja un cliente.
w_vendedor_pv	Ventana que permite visualizar y modificar los datos de los vendedores.
w_ventas_pv	Ventana raíz de la que parten las ventanas: w_cotiza_pv, w_factura_pv, w_pedido_pv y la de puntos de venta propiamente dicha.

Funciones

of_act_monto	Actualiza el total de las ventanas hijas
of_igualar_total	Actualiza el subtotal de las ventanas hijas
of_act_adic_dst	Actualiza el valor de los adicionales después del subtotal en las ventanas hijas
of_act_dif_pago	Actualiza el total de los pagos en relación al monto en las ventanas hijas

of_act_pagos	Actualiza el valor en el detalle de los pagos en las ventanas hijas
of_act_plazos	Actualiza el valor máximo de días de los plazos.
of_pago_tarjeta	Realiza el proceso para pago con tarjeta.
of_set_adicionales	Realiza el cálculo de los adicionales para las ventanas hijas.

Tabla 5.5 Ventanas en la librería punto_ve

1.1.1 Descripción del código para Internet (Java)

A continuación se describe todo el módulo de puntos de venta Internet hecho en Java.

Toda la parte gráfica del sistema se encuentra en el módulo web_venture-war simbolizada con un círculo azul como en la figura 5.1 y las clases se encuentran en web_venture_ejb con el símbolo de un bean figura 5.1. El módulo web_venture una aplicación empresarial con el símbolo de un triángulo figura 5.1, es la encargada de integrar la parte gráfica y los objetos del sistema.

La parte gráfica del módulo están constituidos por páginas en jsp que instanciarán los objetos que manipula la pantalla de punto de venta. Las nuevas páginas se desarrollan dentro la carpeta App del sistema genérico en el módulo web_venture-war.

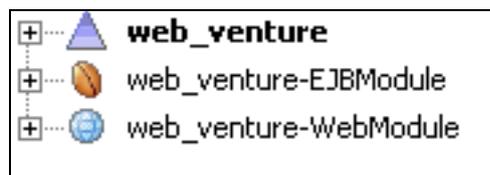


Fig. 5.1 Módulos del sistema Venture (Internet)

La página principal es w_qpunto_ve.jsp que contiene el frontend para el punto de venta de Internet, la página w_qpunto_ve_fs.jsp contiene la declaración de todos los frames utilizados para las distintas funciones del punto de venta.

A continuación se describe los frames utilizados para la construcción de la pantalla principal.

Página w_qpunto_ve_fs.jsp: Descripción General

w_qpunto_ve.jsp	Detalle de la factura, muestra los artículos ingresados por el cajero, visualiza el total y subtotal.
w_qpunto_ve_det1.jsp	Cabecera de la factura, contiene la fecha, nota de venta, cliente, caja, usuario, observaciones. Además contiene la marquesina donde se muestra el último artículo ingresado y el total.
w_qpunto_ve_det2.jsp	Permite seleccionar el artículo por su nombre o código de barras.
w_qpunto_ve_det3.jsp	Contiene las opciones siguientes: <ul style="list-style-type: none">- Ingresar la cantidad de un artículo- Permite devolver uno o varios productos- Ingresar el pago en efectivo- Cancelar toda la venta- Cerrar la venta
w_qpunto_ve_det4.jsp	Ingresar y visualizar los datos para la forma de pago.
w_qpunto_ve_det5.jsp	Permite manipular el detalle de la forma de pago cuando el pago es del tipo crédito propio.
w_qpunto_ve_det6.jsp	Visualiza el detalle de los adicionales, se muestra el IVA, descuentos, recargos y el total de los adicionales.

Tabla 5.6 Descripción general de la página w_qpunto_ve_fs

Al momento de ingresar un nuevo cliente se invoca una pantalla simple w_ing_cliente, la página que contiene los frames es w_ing_cliente_fs

Página w_ing_cliente_fs: Descripción General

w_ing_cliente.jsp	Pantalla que permite el ingreso de la información de un nuevo cliente, nombre, tipo de cliente, grupo, tipo ID, id, teléfono, dirección, e-mail.
--------------------------	--

Tabla 5.7 Descripción general de la página w_ing_cliente_fs

El pago con tarjeta de crédito se utilizaron las librerías y páginas de Paypal:

Páginas Paypal: Descripción General

Calls.html	<p>Página que contiene las opciones de pago y consulta de transacciones:</p> <ul style="list-style-type: none"> - Pago directo - Detalle de transacciones - Búsqueda de transacciones - Reembolso de transacciones
DoDirectPayment.jsp	Es la página en donde se ingresan los valores para la transacción con la tarjeta de crédito.
DirectPaymentReceipt.jsp	Página que envía la información a Paypal, y muestra el estado de la misma.
GetTransactionDetails.html	Realiza consultas a Paypal con el código de transacción para obtener un detalle de la misma.
TransactionSearch.jsp	Búsqueda por fecha de inicio y fin de transacciones realizadas en el sistema, también busca con el código de la transacción.
RefundTransaction.jsp	Permite reembolsar el valor de una transferencia con tarjeta de crédito, se necesita el código de la transacción.

Tabla 5.8 Descripción general de las páginas de Paypal

Todas las funciones del sistema están encapsuladas dentro de la clase principal c_QPunto_ve, la clase c_QPunto_ve es instanciada en la pantalla w_qpunto_ve_fs.jsp la cual es guardada en una variable de sesión y esta es recuperada en cada frame.

Las funciones principales de la clase c_QPunto_ve son:

Funciones	c_QPunto_ve
of_act_adicionales	Actualiza los adicionales de la factura

Parámetros

String as_cle : Código del cliente
String as_fp: Código de la forma de pago
int ai_dpl : Dias de pago
float af_monto: Monto de venta

of_calendario

Crea un calendario para seleccionar la fecha del detalle de la forma de pago

Retorna

String

of_crearCombo

Construye un combobox en html para añadirlo a cualquier parte de la página.

Parámetros

String as_nombrecombo: Nombre del combobox para el id en html
String as_tabla: Tabla de donde extrae los valores para el combo
String as_etiqueta: Campo del combo que será visualizado en la página
String as_valor: Valor del campo seleccionado del combo.
String as_condicion: Condiciones para cargar los datos en el combo.
String as_valdefecto: Valor por defecto del combo.
int ai_valnull: Permite que exista un valor nulo en el combo.
String as_compania: Código de la compañía
String as_sucursal: Código de la sucursal

Retorna

String

of_dibujaVentanaCQ

Función que construye la cabecera de

la factura, maneja eventos en el lado del cliente y servidor. La función es invocada desde la página w_qpunto_ve_det1.jsp

Parámetro

HttpServletRequest arq_request :
Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

Retorna

String

of_dibujaVentanaDA

Muestra los adicionales actualizados cada vez que se realiza la actualización del detalle de la venta, esta función es invocada desde la página w_qpunto_ve_det6.jsp.

Parámetro

HttpServletRequest arq_request :
Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

Retorna

String

of_dibujaVentanaDP

Visualiza el detalle de la forma de pago y permite manipular los datos, la llamada a esta función se la realiza desde la página w_qpunto_ve_det5.jsp.

Parámetro

HttpServletRequest arq_request :
Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

Retorna

String

of_dibujaVentanaDQ

Muestra el detalle de los artículos

ingresados por el cajero o por el lector de código de barras, desde la página w_qpunto_ve.jsp se ejecuta esta función. Permite seleccionar artículos para posteriormente ser eliminados.

Parámetro

HttpServletRequest arq_request :
Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

Retorna

of_dibujaVentanaFP

String
Visualiza la forma de pago, permite insertar varias formas de pago, se ejecuta desde la página w_qpunto_ve_det4.jsp.

Parámetro

HttpServletRequest arq_request :
Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

Retorna

of_dibujaVentanaOQ

String
Crea los botones para ingresar la cantidad del artículo a seleccionar, además permite devolver uno o varios productos, ingresa el pago en efectivo, cancela toda la venta, cierra la venta. La función es invocada desde la página w_qpunto_ve_det3.jsp.

Parámetro

HttpServletRequest arq_request :
Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

	<i>Retorna</i>	String
of_dibujaVentanaSQ		Permite seleccionar el artículo por nombre o por código de barras, valida la cantidad del artículo. Se ejecuta desde w_qpunto_ve_det2.jsp.
	<i>Parámetro</i>	HttpServletRequest arq_request : Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.
	<i>Retorna</i>	String
of_guardar_nv		Guarda la cabecera de la forma de pago para luego guardar el detalle de la forma de pago.
	<i>Retorna</i>	Int
of_precio_art		Actualiza el precio de un artículo basado en las políticas de precio.
	<i>Parámetros</i>	String as_art : Código del artículo float af_cant: Cantidad del artículo String as_cle . Código del cliente
of_recuperainfo		Inicializa los valores de los cachedrowset que son utilizados para manipular los datos de las tablas del punto de venta.
	<i>Parámetro</i>	HttpServletRequest arq_request : Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.
of_set_datos		Configura las variables del sistema

		genérico para guardar los datos de la venta.
	<i>Parámetro</i>	
of_trae_dato_archivo		boolean ab_recupera: Variable que permite recuperar los datos. Función que busca los valores del archivo de configuración del sistema Venture
	<i>Parámetro</i>	
		String as_archivo : Nombre del archivo String as_seccion: Sección a buscar en el archivo. String as_clave : Busca la variable de la sección String as_val_defecto: Si no se obtiene ningún resultado, se devuelve el valor por defecto.
	<i>Retorna</i>	String
of_val_num		Función que escribe código javascript que busca la tecla presionada por el usuario.
	<i>Retorna</i>	String

Tabla 5.9 Descripción funciones clase c_QPunto_ve

Otra clase personalizada es el ingreso de clientes nuevos, la clase que maneja estas operaciones es c_ing_cliente.

Funciones	c_ing_cliente
of_codigo_sig	Obtiene el código para el nuevo cliente
	<i>Parámetros</i>
	String ls_tabla : Nombre de la tabla

String ls_codigo : Nombre del campo de la clave primaria.

Retorna

String

of_crearCombo

Construye un combobox en html para añadirlo a cualquier parte de la página.

Parámetros

String as_nombrecombo: Nombre del combobox para el ID en html

String as_tabla: Tabla de donde extrae los valores para el combo

String as_etiqueta: Campo del combo que será visualizado en la página

String as_valor: Valor del campo seleccionado del combo.

String as_condicion: Condiciones para cargar los datos en el combo.

String as_valdefecto: Valor por defecto del combo.

int ai_valnull: Permite que exista un valor nulo en el combo.

String as_compania: Código de la compañía

String as_sucursal: Código de la sucursal

Retorna

String

of_dibujaVentanaIngCliente

Construye la página de ingreso del nuevo cliente , esta función es ejecutada desde w_ing_cliente.jsp

Parámetro

HttpServletRequest arq_request :

Tiene el objeto request de la página para manipular los eventos del cliente

of_recuperalInfo

Retorna String y los valores de los objetos.
 Inicializa los valores de los cachedrowset que son utilizados para ingresar el cliente.

Parámetro HttpServletRequest arq_request :
 Tiene el objeto request de la página para manipular los eventos del cliente y los valores de los objetos.

Tabla 5.10 Descripción funciones clase c_ing_cliente

Las variables locales que utilizan las clases son las siguientes

Variables	c_QPunto_ve
String is_conexion	Cadena de conexión de la base de datos
String is_compania	Código de la compañía
String is_sucursal	Código de la sucursal
String is_clienten	Código del cliente nuevo
String is_codigo_lge	Código del lugar geográfico
String is_mensaje	Mensaje que despliega la cabecera del detalle
String is_pagos	Códigos de la forma de pago
float if_iva	Porcentaje del iva
float if_subt	Subtotal de la venta

Tabla 5.11 Descripción variables de sesión del punto de venta Internet

Variables	c_ing_cliente
String is_conexion	Cadena de conexión de la base de datos
String is_compania	Código de la compañía
String is_sucursal	Código de la sucursal

Tabla 5.12 Descripción variables de sesión del punto de venta Internet

5.2 INTEGRACIÓN CON LOS DISPOSITIVOS POS

Los dispositivos POS que se desea integrar al punto de venta son:

- Lector de código de barras.
- Impresora POS (para notas de venta).
- Pantalla táctil.
- Lector de tarjetas de crédito.

A continuación se describe el funcionamiento y forma de integrar cada dispositivo mencionado con el módulo de puntos de venta.

5.2.1 Lector de código de barras

Este dispositivo funciona enviando una luz láser o infrarroja hacia el código de barras, recoge la información numérica y la envía a través del puerto del teclado.

Para la integración de este dispositivo fue necesario hacer que el foco de la pantalla de puntos de venta esté siempre en el ítem donde se leería el próximo código de barras del artículo.

5.2.2 Impresora POS

Una impresora pos presta la facilidad de trabajar con un papel continuo de un ancho pequeño, además que el uso de pines para imprimir permite tener una copia instantánea con papel químico.

Para integrar a cualquier programa POS solo es necesario un driver genérico solo texto, para integrarlo al módulo de puntos de venta fue necesario encontrar los driver exactos de la impresora, ya que para imprimir los bouchers se necesitan varios tipos de formatos. El driver de la impresora viene con sus propios tipos de letras y solo pueden ser utilizados por esta impresora.

5.2.3 Pantalla táctil

La particularidad de la pantalla táctil es que cuando se presiona la pantalla tiene el mismo efecto que se hiciera un clic con el mouse.

Para integrarla al sistema pos se realizó una pantalla de puntos de venta que pueda ser manejada en casi todo con el mouse, tomando en cuenta que el puntero será manejado por un dedo, lo que exige que los controles tengan un tamaño superior al normal.

5.2.4 Lector de tarjetas de crédito

Después de varios intentos por conseguir utilizar un lector de tarjetas de crédito no se lo logró por las siguientes razones:

- La confidencialidad de las empresas que prestan el servicio con los lectores de tarjetas de crédito.
- Son pocas las compañías que trabajan con un sistema que interactúe con el lector.
- Para tener uno de estos lectores es necesario solicitar a las entidades financieras una autorización de uso de las tarjetas a las que representan, con esta autorización se puede solicitar un lector a Datafast o Medianet que son las entidades que prestan el servicio, pagando el valor del dispositivo y del servicio.

En vista a estos impedimentos se llegó a un acuerdo con la empresa auspiciante de hacer una simulación de la máquina lectora después investigar su funcionamiento en un sistema utilizado por “Magda” que trabaja con puntos de venta integrando estos lectores.

5.2.4.1 Resultados de la investigación

Las empresas que prestan este servicio son Datafast y Medianet. Aunque es el mismo servicio los procesos que cumplen son distintos:

a. Datafast

Funciona con la red interna teniendo un servidor que es donde se registra cada máquina que va a utilizar un lector de tarjetas de crédito, además de tener una conexión telefónica con la empresa Datafast.

Para la comunicación entre el dispositivo y el computador por hardware se utiliza el puerto serial y en la parte de software un componente que envía información encriptada y toma información a través de un archivo de texto.

Datos que envía:

- El valor base.
- El valor del IVA.
- Plan de pago (meses).
- Interés cobrado por la tarjeta.

Datos que recibe:

- Nombre tarjeta
- Número tarjeta
- Vencimiento tarjeta
- Número de referencia=000042
- Número de aprobación
- Número de transacción
- Nombre del comercio
- Número del comercio
- Nombre del lote
- Número del lote
- Teléfono
- Fecha
- Hora

b. Medianet

Funciona con una red inalámbrica para comunicarse con Medianet, en este caso es la misma máquina lectora la que da a escoger al cajero el plan de pago para la venta realizada.

Para la comunicación entre el dispositivo y el computador por hardware se utiliza el puerto de red y en la parte de software un componente que envía información encriptada y toma información a través de un archivo de texto.

Datos que se envía:

- El valor base.
- El valor del IVA.

Datos que recibe:

- Nombre tarjeta
- Número tarjeta
- Vencimiento tarjeta
- Número de referencia=000042
- Número de aprobación
- Número de transacción
- Nombre del comercio
- Número del comercio
- Nombre del lote
- Número del lote
- Teléfono
- Fecha
- Hora
- Forma de pago
- Intereses

5.2.5 Otros dispositivos

En la parte de Internet para poder manejar tarjetas de crédito se optó por utilizar los servicios de Paypal, para esto se contrató el servicio y viene con este una clase que se la integró al módulo de puntos de venta para Internet. Las especificaciones para el uso del Paypal se describen en el Anexo C.

5.3 INTEGRACIÓN FUNCIONAL CON EL ERP

La integración con el ERP Venture se divide en dos partes que no tienen que ver con la diferencia que se marca con el aspecto intranet/Internet. Esta integración se da en la parte de la base de datos y en el funcionamiento de este módulo en conjunto con el resto del sistema, más que en su funcionamiento como interfaz ante el cliente, se habla del funcionamiento en la parte de programación ya que se han usado todos los objetos que ofrece el genérico además de ahora ser parte de las distintas funciones generales que tiene el sistema como tal.

5.3.1 Integración en la base de datos

La integración en la base de datos no se basa mucho en la asociación de este nuevo módulo con los demás, ya que de esto se encargarán los programadores que estén a cargo de los módulos existentes que deseen asociarse a este.

La integración es más una creación de relaciones y campos que permiten el trabajo del módulo puntos de venta con el sistema y con todas sus reglas de negocio. Además se realizaron cambios en la parte de los datos del sistema propiamente dicho para que este se adapte al genérico hecho para la Web, es decir no se que no se hizo un conjunto de tablas distinto para intranet y otro para Internet, si no que después de un análisis de los todos requerimientos que se tenían se logró unificar todo en una sola base y hacer pequeños cambios en las tablas existentes para que se adapten a las necesidades de este nuevo módulo en sus dos fases.

Para esta integración se utilizaron las actualizaciones que posee el ERP Venture, que se encuentra en la PBL venturep, en el objeto no visible n_cst_actualizacion, en la función of_actualizacion_configura.

5.3.2 Integración funcional

5.3.2.1 Integración funcional (intranet)

Esta integración se da en el momento de la programación, ya que no se utilizan objetos , visibles y no visibles, que no sean parte del genérico, además que la programación se la realiza siguiendo los estándares marcados en el documento de estándares en capítulo 2, además se encuentra un lazo que une estrechamente a este módulo con el sistema, que es la utilización de las variables del sistema, que pueden ser utilizadas en cualquier módulo y se configuran de acuerdo a las necesidades del cliente a igual que el resto de módulos, estas se encuentran en la PBL venturep, en el objeto no visible n_cst_variable, en las variables de instancia de este objeto y se configuran en la función n_cst_variable. Las variables utilizadas son las siguientes:

Variable	Descripción
is_PV_formato_cliente	Es el formato con el que se despliega el código del cliente en el momento de ingreso de clientes.
is_PV_formato_vendedor	Es el formato con el que se despliega el código del vendedor en el momento de ingreso de vendedores.
is_PV_dias_defecto	Indica los días atrás de la fecha actual filtra la información de las ventanas de transacción.
is_PV_dias_validez_cot	Indica los días que por defecto una cotización tiene validez.
is_PV_fpago_efectivo	Identifica el código que se utiliza para la forma de pago en efectivo.
is_PV_fpago_cheque	Identifica el código que se utiliza para la forma de pago en cheque.
is_PV_fpago_tarjeta	Identifica el código que se utiliza para la forma de pago con tarjeta de crédito.
is_PV_fpago_credprop	Identifica el código que se utiliza para la forma de pago con un crédito propio.
is_PV_fpago_chequepos	Identifica el código que se utiliza para la forma de pago con cheque posfechado.
is_PV_fpago_imp_iva	Identifica el código que se utiliza para la forma de pago con impuestos IVA.
is_PV_fpago_imp_renta	Identifica el código que se utiliza para la forma de pago con impuestos a la renta.
is_PV_fpago_web	Códigos de las formas de pago para la web.
is_PV_fpago_Paypal	Identifica el código que se utiliza para la forma de pago con Paypal.
is_PV_cambia_precio	Indica si el usuario del sistema puede cambiar los precios que el sistema le pone.
is_PV_politica_precio	Indica si se va a utilizar la tabla de precios

	para poner en el momento que se haga la venta un precio de acuerdo a las condiciones actuales.
is_PV_politica_descue	Indica si se va a utilizar la tabla de descuentos para poner en el momento que se haga la venta un descuento de acuerdo a las condiciones actuales.
is_PV_fpago_defecto	Identifica la forma de pago que utiliza la empresa por defecto.
is_PV_adic_iva	Identifica el código que se utiliza para el adicional IVA.
is_PV_adic_desc	Identifica el código que se utiliza para el adicional Descuento.
is_PV_adic_valtar	Identifica el código que se utiliza para el adicional Valor por uso de tarjeta.
is_PV_srvt_datafast	Identifica el código que tiene el servicio de Datafast para la lectura de tarjetas.
is_PV_srvt_medianet	Identifica el código que tiene el servicio de Medianet para la lectura de tarjetas.
is_PV_est_normal_cot	Código que identifica el estado normal de una cotización.
is_PV_est_normal_ped	Código que identifica el estado normal de un pedido.
is_PV_est_normal_fac	Código que identifica el estado normal de una factura.
is_PV_ningun_vend	Código que identifica al vendedor 'Ninguno'.
is_PV_fpago_usa_det	Conjunto de códigos de las formas de pago que necesitan detalle.
is_PV_fpago_nec_cliente	Conjunto de códigos de las formas de pago que necesitan cliente
is_PV_fpago_usa_pventa	Conjunto de códigos de las formas de pago que usará el punto de venta.

is_PV_dir_arch_med	Dirección y archivo de los datos para el datafast.
is_PV_dir_arch_dat	Dirección y archivo de los datos para el medianet.
is_PV_est_anulado_fac	Código que identifica el estado anulado de una factura.
is_PV_est_activo_cli	Código que identifica el estado activo de un cliente.
is_PV_peso_art	Código que identifica la característica peso.
is_PV_volumen_art	Código que identifica la característica volumen.
is_PV_fragilidad_art	Código que identifica la característica fragilidad.
is_PV_padre_clientes_nuevos	Código que identifica el padre de los clientes nuevos.
is_PV_tipo_cli_def	Código que identifica el grupo de cliente por defecto.
is_PV_grupo_cli_def	Código que identifica el tipo de cliente por defecto.
is_PV_cliente_defecto	Cliente por defecto que saldrá en el punto venta automáticamente cada vez que se realice una nueva venta.

Tabla 5.13 Variables globales utilizadas por el módulo de puntos de venta.

5.3.2.2 Integración funcional (Internet)

La integración del módulo Web del punto de venta es a nivel de base de datos y funcional, la base de datos que utilizan ambos módulos es la misma, las variables del sistema para el punto de venta Internet son recuperadas utilizando las funciones genéricas del sistema Venture Web. El EJB del sistema genérico es el encargado de las operaciones con la base de datos, estas operaciones son recuperar, guardar, actualizar y eliminación de registros

El módulo de punto de Venta Web recibe del genérico 5 variables de sesión. Las variables de sesión contienen información sobre : el código del usuario del sistema (ts_codigo_usu), código de la compañía (ts_codigo_com) , código de la sucursal (ts_código_suc), código del cliente (ts_código_cle), y finalmente la cadena de conexión a la base de datos (ts_base_datos).

La página del punto de venta Web tiene que ser declarada en la tabla de opciones del sistema Venture con el fin de administrar sus permisos. En la tabla de opciones estará declarada la ventana como app/w_qpunto_ve, la misma que estará relacionada con las tablas j_cabece_factur (cabecera de la factura), j_detall_factur (detalle de la factura), j_adicio_factur (adicionales de la factura), j_pago_factura (forma de pago de la factura), j_det_pago_fact (detalle de la forma de pago).

La clase genérica que utiliza el módulo Web es parte del paquete app, app es un paquete que contiene las clases heredadas de otras ventanas de mantenimiento, se creó la clase c_QPunto_ve que se obtiene de la herencia de una ventana de mantenimiento quintuple.

5.3.3 Jerarquía modular

A continuación se presenta en forma textual y jerárquica como se manejará el módulo “Puntos de venta” en conjunto con los módulos propios del ERP necesarios para el funcionamiento del mismo.

Sistema

- Módulos
- Compañías
- Sucursales
- Opciones
- Reportes del sistema
- Permisos
- Variables del sistema
- Utilitarios

General

Lugares geográficos
Tipos de identificación

Compras

Característica de Artículo
Nivel del Artículo
Marca del Artículo
Estado del Artículo
Unidades
Artículos

Contabilidad

Identificadores de cuenta
Tipo de depósito

Puntos de venta

Mantenimiento
Adicionales
Agencia de Distribución
Bodegaje
Forma de envío
Cajas (punto de venta)
Períodos de Venta
Estado de Cotización
Estado de Factura
Estado de Pedido
Estados de Cliente
Grupos de Clientes
Monto de Venta
Rangos de Plazo
Temporada
Tipo de Cupo
Tipo de Venta
Tipos de Cliente
Transporte
Servicio para tarjetas
Forma de pago con tarjeta
Entidades de Pago
Formas de Pago

Clientes PV
Vencimiento cliente
Volumen venta
Descuentos
Precios
Vendedor
Movimiento
Cotización
Pedidos
Facturación
Punto de venta

5.4 PLAN DE PRUEBAS

5.4.1 Introducción

Este documento contiene las especificaciones para cumplir las pruebas de funcionalidad, manejo e ingreso de datos del módulo desarrollado.

5.4.1.1 Objetivos y Alcance

Las Especificaciones de Prueba de navegación e ingreso de datos para el proyecto verificarán que la funcionalidad del módulo de puntos de venta satisfaga los requerimientos.

5.4.1.2 Estrategia

El encargado de realizar las pruebas preparará las especificaciones y ejecutará las pruebas. La especificación de requerimientos será la base para iniciar la evaluación del módulo, combinando con el escenario lógico que corresponde a la secuencia de los procesos y finalizando con la interfaz.

5.4.1.3 Organización del Documento

Este documento define el plan para conducir las pruebas de acuerdo con las especificaciones de la Prueba en las áreas siguientes:

- **Objetivos:** identifica las categorías de las pruebas que van a ser incluidas o excluidas desde la Especificación de Prueba del <<prueba>>.

- Cronograma de Pruebas: una lista de tareas y actividades de alto nivel indicando las fechas de inicio y terminación.
- Las responsabilidades: identifica los recursos disponibles y sus responsabilidades.
- Los recursos: identifica los recursos y los requisitos de software y hardware.
- Los procedimientos: describe los procedimientos a seguir en la preparación, desarrollo y verificación de los resultados de la prueba.
- Supuestos: documenta los supuestos hechos en la preparación de la especificación de la prueba.
- Los apéndices: contienen una secuencia de los escenarios de Prueba, y una muestra de los documentos y entregables que resultan de las pruebas definidas.

5.4.2 Definición general de las pruebas

Las especificaciones para las pruebas de funcionalidad, ingreso de datos e interfaz proveerán una metodología formal para las pruebas del módulo. Al identificar los tipos de prueba necesarios para esta aplicación, colocando un X en Si o No, si se va a ejecutar o no ese Tipo de prueba (esto si el Tipo de prueba aplica para el proyecto), o N.A. si la prueba no aplica para el proyecto.

Probar			Tipo de prueba	Descripción del Prueba	Dura ción
Si	No	NA			
X			Procesos e Interfase de Usuario	Procesamiento Lógico en el sistema: actualización de datos, cálculos y que el sistema satisfaga lo descrito en el documento de especificaciones funcionales y la ayuda en línea.	
	X		Interfase con otros sistemas.	Módulos de interfaces o flujo de información requerido.	

X			Volumen	Simulación de los volúmenes de datos esperados en ambiente de producción.	
X			Concurrencia	Usuarios simultáneos accediendo a la aplicación.	
	X		Recuperación	Procedimientos de Backup y recuperación.	
X			Seguridad	Especificaciones de Seguridad de acuerdo con las especificaciones requeridas	
		X	Conversión	Conversión de los datos actuales.	
X			Documentación	Concordancia de la documentación, incluida la ayuda, con respecto a la aplicación.	
		X	Procedimientos Administrativos	Pruebas de Formas y procedimientos requeridos en el ambiente de producción.	

Tabla 5.14 Tipos de pruebas

5.4.3 Recursos

La siguiente sección define los recursos necesarios, personas, hardware o software.

5.4.3.1 Miembros y responsabilidades del equipo de pruebas

El equipo de pruebas es el siguiente:

Nombre	Tipo de prueba	Responsabilidad
David Barrigas Pablo Barriga	Procesos e Interfase de Usuario	Hacer uso del sistema, probando así todas las funciones del software. Hacer comentarios sobre diseño gráfico y presentación de la aplicación WEB.

Nombre	Tipo de prueba	Responsabilidad
David Barrigas	Volumen	Hacer el ingreso del mayor número de información posible, con el fin de probar la capacidad de la base de datos.
Pablo Barriga	Concurrencia	Acceder a la página y ventana principal de la aplicación, desde varios equipos terminales con el fin de verificar que no se bloqueen los datos.
David Barrigas Pablo Barriga	Seguridad	Tratar de acceder a páginas validadas y ventanas con nombre de usuario y contraseña, de todas las formas posibles con el fin de asegurar la seguridad a los usuarios.
David Barrigas	Documentación	Verificar que todas las funciones y relaciones de datos estén de acuerdo con los documentos presentados previamente.

Tabla 5.15 Personal de pruebas y responsabilidades

5.4.3.2 Requerimientos de Recursos

Las personas, el hardware y el software requerido es el siguiente:

Tipo de prueba	Cantidad	Nombre del Recurso
Procesos e Interfase de Usuario	1	PC Pentium IV → 512Mb RAM
	1	Windows Me o superior Internet Explorer 5.x o superior Firefox 1.5.0.6
Volumen	1	PC Pentium IV → 512Mb RAM
	1	Windows Me o superior Internet Explorer 5.x o superior

Tipo de prueba	Cantidad	Nombre del Recurso
		Firefox 1.5.0.6
Concurrencia	2	PC Pentium IV → 512Mb RAM
	1	Windows Me o superior VMWare 5 Internet Explorer 5.x o superior Firefox 1.5.0.6
Seguridad	2	PC Pentium IV → 512Mb RAM
	1	Windows Me o superior Internet Explorer 5.x o superior Firefox 1.5.0.6
Documentación	1	PC Pentium IV → 512Mb RAM
	1	Windows Me o superior Internet Explorer 5.x o superior Firefox 1.5.0.6 Base de datos Sybase Anywhere 8 Base de datos SQL Server 2000 Base de datos Oracle 9i Base de datos Informix

Tabla 5.16 Requerimientos de recursos

5.4.4 Procedimiento para escenarios de prueba

El siguiente es el procedimiento para la preparación de la ejecución de los escenarios de las pruebas.

5.4.4.1 Preparación de la Prueba

Múltiples casos deben ser preparados, uno por cada proceso definido en las especificaciones funcionales.

Los casos serán combinados en los escenarios, cada escenario:

- Corresponde a una o más tareas dentro del proceso.
- Secuencias Lógicas que pueden ser repetidas.
- Excepciones del proceso.

5.4.4.2 Ambiente de prueba

Para las pruebas con el módulo en intranet se realizarán en un terminal real y la simulación de varios terminales con el programa VMWare 5, estos terminales tendrán conexión a las bases de datos e instalado el ERP. Para Internet las pruebas serán realizadas desde un cliente remoto al servidor, simulando así el uso como un asistente más desde el Internet. La máquina de pruebas estará conectada a la red y configurada en el mismo dominio/grupo de trabajo que el servidor. Desde el explorador WEB, el equipo de pruebas usará la dirección IP del servidor como dirección URL.

5.4.4.3 Ejecución y evaluación de las pruebas

Los escenarios de prueba serán efectuados en la secuencia enumerados en el Apéndice A.

El encargado de ejecutar un escenario de prueba, evaluará y escribirá los resultados de la prueba. La documentación de apoyo (pantallas e informes) debe conservarse para cada corrida de prueba, al igual que los registros de las pruebas efectuadas y de los problemas encontrados. Una copia del formato para el Registro y Control de estas fallas se encuentra en el Apéndice C.

Las fallas presentadas deberán ser archivadas en el Fólder del Proyecto e informadas en la reunión de control semanal.

Cada falla debe calificarse de acuerdo con su gravedad y determinar si afecta la secuencia de las pruebas programadas, de acuerdo con la siguiente tabla:

Gravedad	Descripción
1	Error grave que causa la suspensión del trabajo, es un error crítico y su solución debe ser de inmediato.
2	Error medio es posible continuar con otras pruebas, pero el error corresponde a una funcionalidad esencial. La solución a estas situaciones debe tener prioridad alta.

3	Errores leves de presentación que no afectan la operación de la aplicación.
---	---

Tabla 5.17 Descripción de la gravedad de los errores

Cuando la falla se soluciona la prueba debe volverse a correr y todos los escenarios relacionados.

5.4.4.4 Supuestos

El Cronograma de pruebas esta basado en los siguientes supuestos:

- El ambiente para la prueba va a estar disponible en la fecha de inicio de acuerdo con el cronograma de pruebas.
- Las ventanas y páginas WEB que integran la aplicación estarán disponibles y serán probadas.
- Los documentos de referencia fueron la base para la elaboración de los casos de prueba y estarán disponibles durante la prueba.

Todos los escenarios tendrán un mecanismo de aprobación que asegure que el caso a probar cumple con los requerimientos, y de no ser así será modificado hasta que cumpla con los requisitos.

5.4.5 Criterios de Aceptación

La Aplicación será aceptada cuando todas las pruebas especificadas en el Apéndice B sean ejecutadas satisfactoriamente y:

- a. Los Resultados esperados estén de acuerdo con las especificaciones funcionales.
- b. Todos los problemas hayan sido corregidos y los escenarios asociados a estas fallas se hayan vuelto a correr satisfactoriamente.

5.4.6 Apéndices

5.4.6.1 Apéndice A: Secuencia de ejecución de los escenarios

Orden / Sec	Escr No.	Nombre Escenario	Escenarios Previos Requeridos
1	1	Ingreso de usuarios	

2	2	Ventanas de mantenimiento.	Ingreso de Usuarios
3	3	Páginas de mantenimiento.	Ingreso de Usuarios
4	4	Página y ventana de puntos de venta.	
5	5	Ventanas de mantenimiento, página y ventana de puntos de venta.	

Tabla 5.18 Secuencia de ejecución de escenarios

5.4.6.2 Apéndice B: Escenarios de prueba

Escenario de Prueba		
Proyecto:	Módulo de puntos de venta	
Escenario:	Ingreso de usuarios	No. 1
Módulo:	Sistema ERP	
Caso de prueba:	Ingreso nuevo usuario o cliente.	
Tipo de prueba:	Seguridad	Pág. 1
Definido por:	David Barrigas Pablo Barriga	Fecha Creación: 2006-01-03
Participantes:	David Barrigas Pablo Barriga	
Descripción de la prueba:	Ingreso al sistema tanto al módulo de Internet como al de intranet con un usuario nuevo y en el caso específico de Internet como un cliente nuevo.	

Tabla 5.19 Escenario ingreso de usuarios

Escenario de Prueba		
Proyecto:	Módulo de puntos de venta	
Escenario:	Ventanas de mantenimiento	No. 2
Módulo:	Puntos de venta, genérico venture	
Caso de prueba:	Ingreso y cambios en las pantallas de mantenimiento y transacciones	
Tipo de prueba:	Procesos e interfase de usuario	Pág. 1
Definido por:	David Barrigas	Fecha Creación: 2006-01-04

Participantes:	Clara Rojas
Descripción de la prueba:	Ingreso a las ventanas de mantenimiento, modificación de datos y comprobando la relación entre las pantallas.

Tabla 5.20 Escenario ventanas de mantenimiento

Escenario de Prueba		
Proyecto:	Módulo de puntos de venta	
Escenario:	Páginas de mantenimiento	No. 3
Módulo:	Genérico venture Web	
Caso de prueba:	Ingreso y cambios en las pantallas de mantenimiento y transacciones	
Tipo de prueba:	Procesos e interfase de usuario	Pág. 1
Definido por:	Pablo Barriga	Fecha Creación: 2006-01-06
Participantes:	Celio Barrigas	
Descripción de la prueba:	Ingreso a las ventanas de mantenimiento, modificación de datos y comprobando la relación entre las pantallas.	

Tabla 5.21 Escenario páginas de mantenimiento

Escenario de Prueba		
Proyecto:	Módulo de puntos de venta	
Escenario:	Página y ventana de puntos de venta.	No. 4
Módulo:	Punto de venta	
Caso de prueba:	Realizar simultáneamente varias compras.	
Tipo de prueba:	Concurrencia	Pág. 1
Definido por:	Pablo Barriga	Fecha Creación: 2006-01-08
Participantes:	David Barrigas Pablo Barriga	

Descripción de la prueba:	Ingreso a varios puntos de venta, en intranet e Internet y realizar ventas en el caso de la ventana de puntos de venta y compras en la página de compras.
----------------------------------	---

Tabla 5.22 Escenario página y ventana de puntos de venta

Escenario de Prueba		
Proyecto:	Módulo de puntos de venta	
Escenario:	Ventanas de mantenimiento, página y ventana de puntos de venta.	No. 5
Módulo:	Punto de venta	
Caso de prueba:	Ingreso de 200 artículos, 50 reglas de descuento y 50 reglas de precios	
Tipo de prueba:	Volumen	Pág. 1
Definido por:	David Barrigas	Fecha Creación: 2006-01-14
Participantes:	David Barrigas Pablo Barriga Clara Rojas	
Descripción de la prueba:	Realizar ventas tanto desde intranet como Internet con una base que posea unos 200 artículos, 50 reglas de descuento y 50 reglas de precios.	

Tabla 5.23 Escenario ventanas de mantenimiento, página y ventana de puntos de venta

5.4.6.3 Apéndice C: Resumen de la ejecución de las pruebas

No.	Esctr	Usuarios	Resultado	Fecha	Grav
1	1	David Barrigas Pablo Barriga	La prueba de ingreso de un nuevo usuario, brindo resultados satisfactorios en cuanto a validación y registro del nuevo usuario.	2006-01-03	Ninguna

2	2	David Barrigas	La prueba de modificar las ventanas de mantenimiento fue exitosa, en las ventanas de transacciones se tuvo que mejorar el refrescamiento del datawindow de adicionales. La persona que realizó el manejo se adaptó rápidamente a la forma de trabajo del ERP.	2006-01-04	Leve
3	3	Pablo Barriga	La prueba de modificar las páginas de mantenimiento fue exitosa. Además se demostró que el módulo Web es amigable con el usuario.	2006-01-06	Ninguna
4	4	Pablo Barriga	Al utilizar máquinas reales y virtuales se pudo realizar una concurrencia considerable, aunque se detectaron errores de refrescamiento en la página de compras en el momento de seleccionar los artículos y en la ventana de puntos de venta en el momento de ingresar mal el número de la tarjeta de crédito.	2006-01-08	Grave

5	5	David Barrigas Pablo Barriga	Esta prueba no solo se la hizo para ver el funcionamiento con el volumen de información, también fue una prueba para probar la amigabilidad, funcionamiento y operabilidad de la ventana de punto de venta y la página de compras. En la parte Web como en la parte de intranet trabajó perfectamente con el volumen de información incrementado aunque se encontró un error la asignación del interés de las tarjetas de crédito y la forma de guardar los detalles de pagos..	2006-01-14	Grave
---	---	-------------------------------------	---	------------	-------

Tabla 5.24 Resumen de ejecución de las pruebas

5.4.6.4 Apéndice D: Muestra gráfica de los escenarios de pruebas

Escenario 1

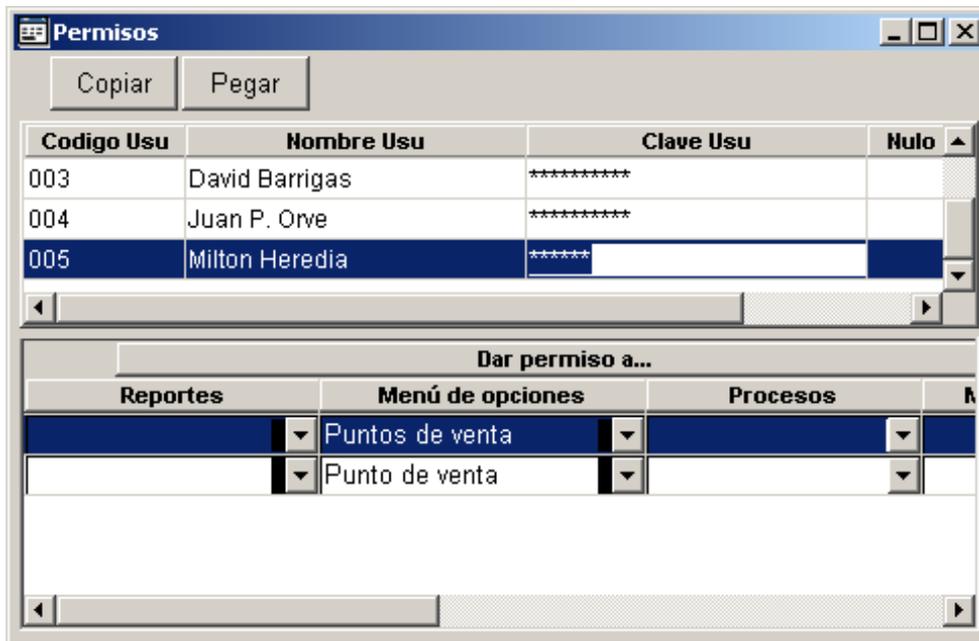


Fig. 5.2 Ingreso de un nuevo cajero (intranet)

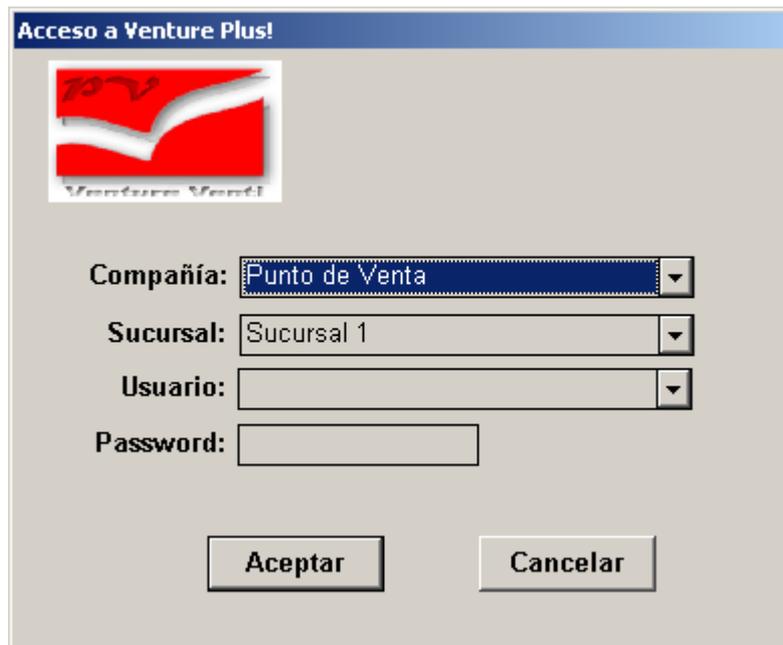


Fig. 5.3 Ingreso al sistema (intranet)

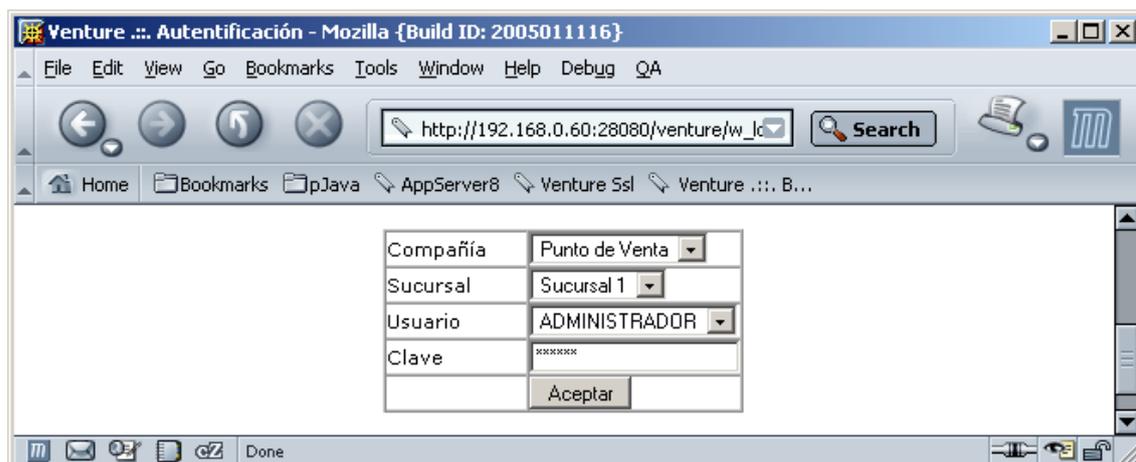


Fig. 5.4 Ingreso al sistema (intranet)

Escenario 2

Código	Nombre	Signo	Sumar al Subtotal	Parametro
001001	IVA	<input checked="" type="radio"/> + <input type="radio"/> -	<input type="checkbox"/>	12.00
001002	Recargo tarjeta	<input checked="" type="radio"/> + <input type="radio"/> -	<input checked="" type="checkbox"/>	8.00
001003	Descuento	<input type="radio"/> + <input checked="" type="radio"/> -	<input checked="" type="checkbox"/>	
001004	Interés tarjeta	<input checked="" type="radio"/> + <input type="radio"/> -	<input type="checkbox"/>	
001005	Recargo crédito	<input checked="" type="radio"/> + <input type="radio"/> -	<input checked="" type="checkbox"/>	

Fig. 5.5 Ventana de mantenimiento genérico Adicionales



Fig. 5.6 Ventana de transacciones Cotización

Escenario 3

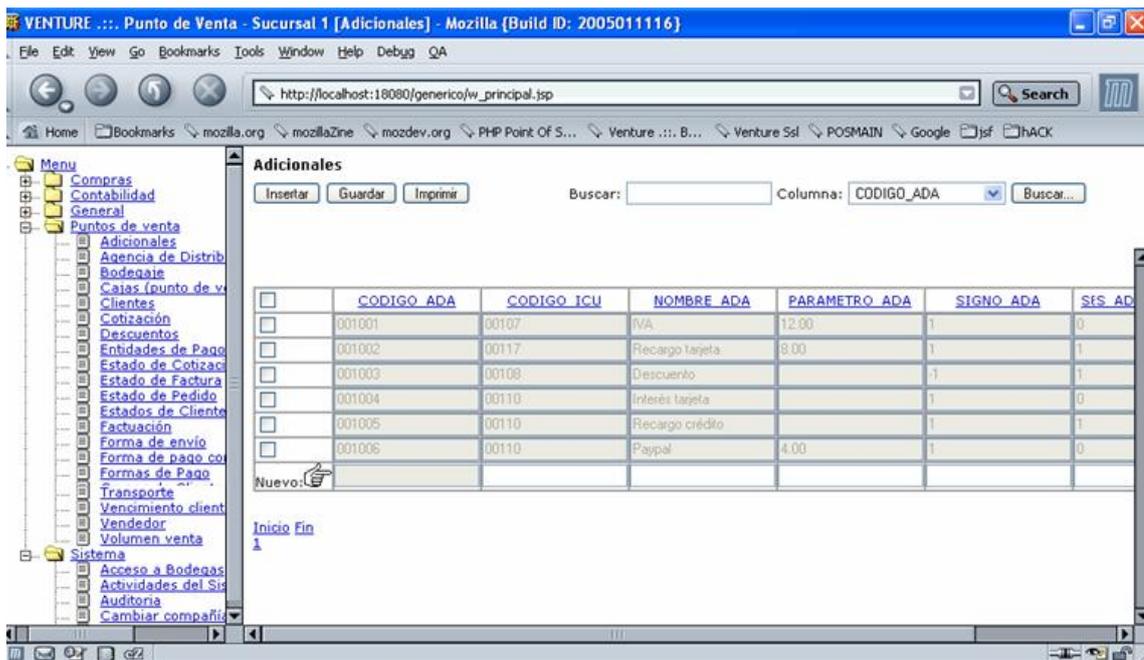


Fig. 5.7 Ventana de mantenimiento de Adicionales

Escenario 4

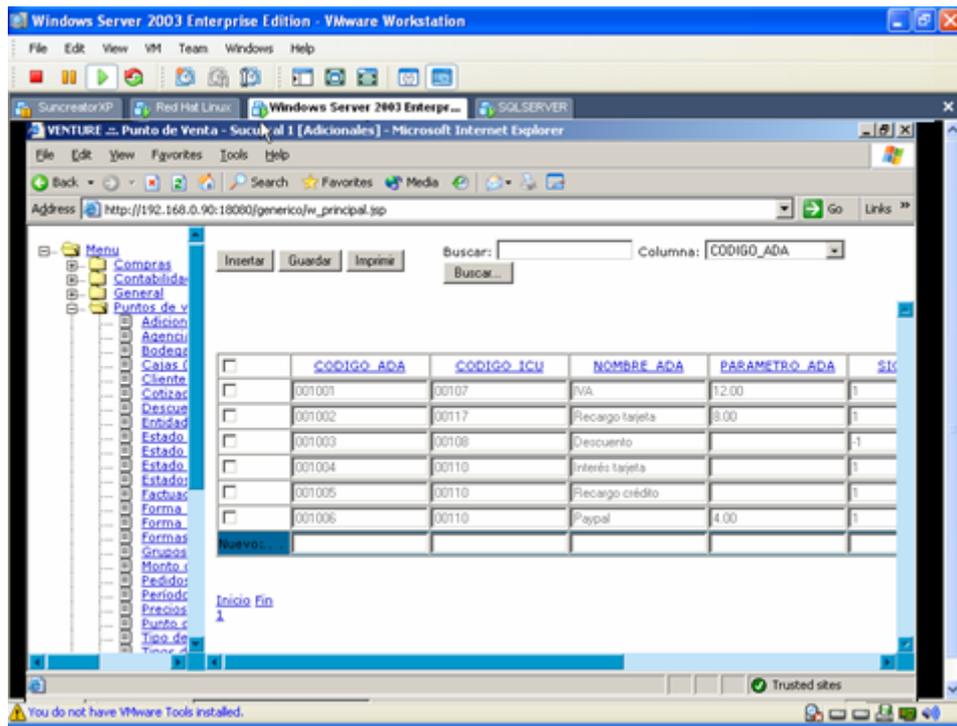


Fig. 5.8 (VMWare) ingresando al módulo de puntos de venta en Internet

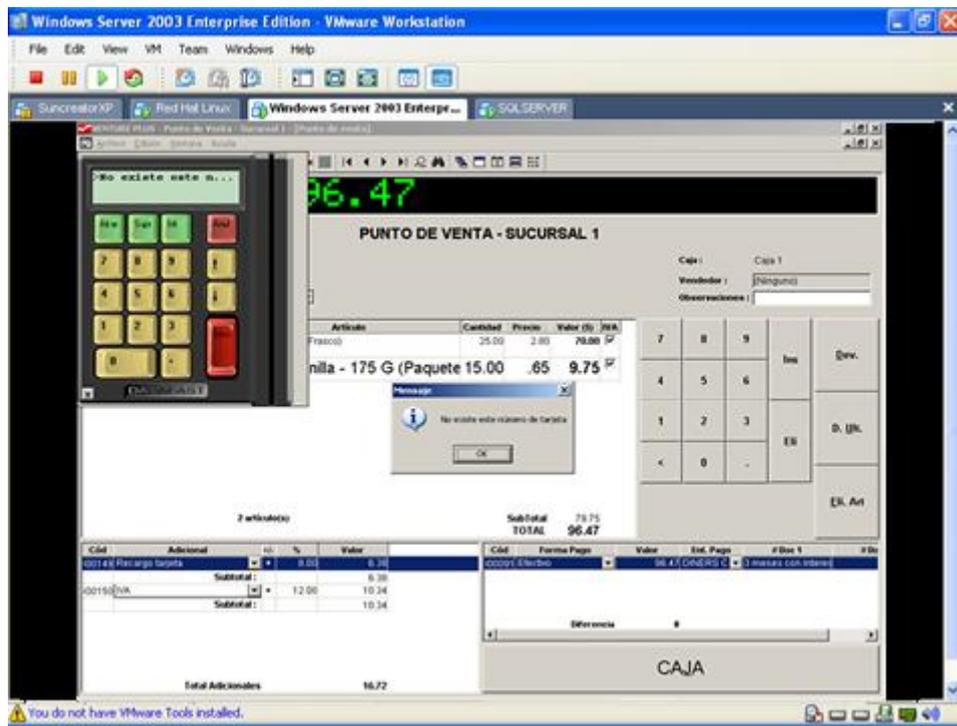


Fig. 5.9 (VMWare) ingresando al módulo de puntos de venta en intranet

Escenario 5

Total = \$31.54

PUNTO DE VENTA - SUCURSAL 1

Fecha : 2006-02-03 Caja : Caja 1

Nota de Venta : Vendedor : Milton de Heredia

Cliete : Observaciones :

Item	Código de barras	Artículo	Cantidad	Precio	Valor (\$)	IVA
1	754105002840	Rinze - hierbazid - 200cm3 (Uni)	15.00	1.73	25.95	<input checked="" type="checkbox"/>
2	7702001044504	Avena Alpina 250g (Uni)	2.00	.56	1.12	<input type="checkbox"/>
3	7861018587609	Papas fritas - crema cebolla - 100 G (Funda)	2.00	.75	1.50	<input checked="" type="checkbox"/>
4			1.00	.00	0.00	<input type="checkbox"/>

SubTotal: 28.57
TOTAL: 31.54

4 artículo(s)

Cód	Adicional	+/-	%	Valor
	Descuento	-	1.00	.29
	Subtotal:			-0.29
	IVA	+	12.00	3.26
	Subtotal:			3.26
	Total Adicionales			2.97

Cód	Forma Pago	Valor	Ent. Pago	# Doc 1	# Doc
	Efectivo	31.54			
	Diferencia	0			

CAJA

Fig. 5.10 Ventana Punto de venta (intranet)

VENTURE ... Punto de Venta - Sucursal 1 - Mozilla {Build ID: 2005011116}

http://192.168.0.60:28080/venture/w_principal.jsp

Total: \$5.0

Nota de venta: Usuario: WEBMASTER

Cliete: Nuevo Observaciones:

Búsqueda:

Cantidad:

No.	Artículo	Cant.	Precio	Valor	IVA	Dev
1	Tampico - citrus punch - 500ml	2.0	0.45	0.9	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Desodorante lady - Aloe - 45g	1.0	2.56	2.56	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Salsa para tallerin - facundo - 42g	1.0	1.0	1.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Total: 5.0 Subtotal: 4.46

Adicional	+/-	%	Valor
IVA	+	12.0	0.54
Suma:			0.54

Forma Pago	Valor
Crédito propio	5

Días	%	Fecha	Valor	Ent.
0	70	2006-08-07	3.5	
0	30	2006-08-07	1.5	

Fig. 5.11 Ventana Punto de venta (Internet)

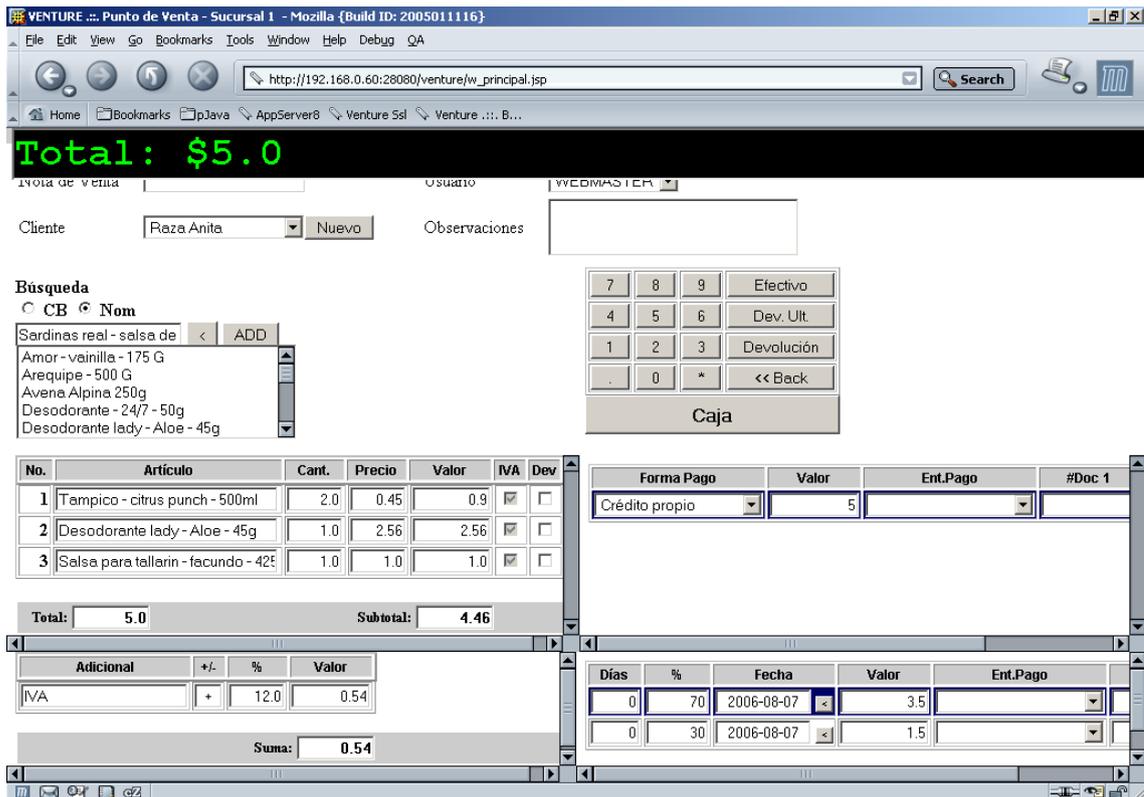


Fig. 5.12 Ventana Punto de venta (Vista completa Internet)

5.4.6.5 Apéndice E: Resultados de las Pruebas

Escenario 1 (Caja negra)

Requisitos		
Procedimiento	Descripción	ok
Pruebas Previas	Ninguna	Ok
Requeridas:		
Requisitos Funcionales:	Usuarios: ADMINISTRADOR	Ok
Ambiente Técnico Previo	PC Pentium IV → 512Mb RAM	Ok
Requerido:	Windows Me o superior Internet Explorer 5.x o superior Firefox 1.5.0.6	
Comentarios: Las pruebas serán realizadas desde un computador con el programa para intranet y desde un computador remoto que ingresará a través del explorador. La máquina de pruebas estará conectada a la red y configurada en el mismo dominio/grupo de trabajo que el servidor. Desde el explorador WEB, el		

equipo de pruebas usará la dirección IP del servidor como dirección http://192.168.0.90:28080/venture		
Secuencia de la Prueba		
Procedimientos	Resultados Esperados	ok
Ingresar el sistema como administrador.	Ingresar con todos los permisos y verificar que se abran las opciones.	Ok
Crear un nuevo usuario y asignarle permisos de opciones y de reportes.	En la pantalla de permisos me permite ingresar un nuevo usuario y los permisos para este.	Ok
Ingresar como el nuevo usuario y verificar que se aplique la configuración de permisos que se le dio.	Permitir ingresar como el nuevo usuario y poder utilizar todo a lo que se le dio permiso y no poder usar a lo que se le restringió.	Ok
Fallas Encontradas	Descripción	Grav edad
Comentarios de la prueba :		
La prueba de ingreso de un nuevo usuario, brindo resultados satisfactorios en cuanto a validación y registro del nuevo usuario.		
Nombre Ejecutor de la Prueba:		
David Barrigas		Firma Ejecutor de la Prueba
Pablo Barriga		

Cuadro 5.1 Resultados del Escenario 1

Escenario 2 (Caja negra)

Requisitos		
Procedimiento	Descripción	ok
Pruebas Previas Requeridas:	Ingreso de usuarios	Ok
Requisitos Funcionales:	Usuarios: Personal que trabaja en un punto de venta.	Ok
Ambiente Técnico Previo Requerido:	PC Pentium IV → 512Mb RAM Windows Me o superior	Ok
Comentarios: Las pruebas serán realizadas desde un computador con el programa para intranet de puntos de venta.		
Secuencia de la Prueba		
Procedimientos	Resultados Esperados	ok
Ingresar al sistema como administrador.	Ingresar con todos los permisos y verificar que se abran las opciones.	Ok
Ingresar a cada una de las ventanas verificado que retorne bien los datos.	Todas las opciones (ventanas) se abren y despliegan los datos correctamente.	Ok
Dentro de las pantallas realizar cambios y verificar que se guarden.	Las ventanas permiten realizar cambios y guardarlos.	Ok
Verificar que la interfaz funcione correctamente.	Las pantallas se presentan con los valores calculados correctos.	No
Fallas Encontradas	Descripción	Gravidad
No se refresca correctamente el datawindow de adicionales en las pantallas de movimiento.	El momento de recalcular los adicionales el datawindow no se refresca correctamente.	Leve

Comentarios de la prueba :

La prueba demostró que .un usuario que cumple funciones de cajero en un punto de venta se adaptó fácilmente con el sistema, además de que se encontró una falla.

Nombre Ejecutor de la Prueba:

David Barrigas

**Firma Ejecutor de la
Prueba**

Cuadro 5.2 Resultados del Escenario 2

Escenario 3 (Caja negra)

Requisitos		
Procedimiento	Descripción	ok
Pruebas Previas Requeridas:	Ingreso de usuarios	Ok
Requisitos Funcionales:	Usuarios: Persona que hace compras por Internet.	Ok
Ambiente Técnico Previo Requerido:	PC Pentium IV → 512Mb RAM Internet Explorer 5.x o superior Firefox 1.5.0.6	Ok
Comentarios: Las pruebas serán realizadas desde un computador remoto que ingresará a través del explorador.		
Secuencia de la Prueba		
Procedimientos	Resultados Esperados	ok
Ingresar al sistema como administrador.	Ingresar con todos los permisos y verificar que se abran las opciones.	Ok
Ingresar a cada una de las páginas verificado que retorne bien los datos y el tiempo de respuesta.	Todas las opciones (páginas) se abren y despliegan los datos correctamente.	Ok
Dentro de las páginas realizar cambios y verificar que se guarden.	Las ventanas permiten realizar cambios y guardarlos.	Ok
Verificar que la interfaz funcione correctamente.	Las pantallas se presentan con los valores calculados correctos.	Ok
Fallas Encontradas	Descripción	Grav edad

Comentarios de la prueba :

La prueba demostró que .un usuario que realiza compras por Internet navega en forma natural por las páginas mostradas por el módulo de puntos de venta para Internet.

Nombre Ejecutor de la Prueba:

Pablo Barriga

**Firma Ejecutor de la
Prueba**

Cuadro 5.3 Resultados del Escenario 3

Escenario 4 (Caja blanca)

Requisitos		
Procedimiento	Descripción	ok
Pruebas Previas Requeridas:	Ingreso de usuarios	Ok
Requisitos Funcionales:	Usuarios: Personal que trabaja en un punto de venta y una persona que hace compras por Internet.	Ok
Ambiente Técnico Previo Requerido:	PC Pentium IV → 512Mb RAM Windows Me o superior Internet Explorer 5.x o superior Firefox 1.5.0.6	Ok
Comentarios: Las pruebas serán realizadas desde computadores virtuales que representarán movimientos normales de operación tanto para la parte de intranet como para la de Internet.		
Secuencia de la Prueba		
Procedimientos	Resultados Esperados	ok
Ingreso a la pantalla Punto de venta para intranet y la página de compras para Internet.	Al ingresar se deben inicializar todas las variables y objetos gráficos correctamente.	Ok
Realizar una compra.	Todos los adicionales tienen que calcularse en el módulo de intranet.	Ok
Seleccionar el pago con tarjeta de crédito.	Las ventanas permiten realizar cambios y guardarlos.	Ok
Se recalculan los adicionales	Muestra nuevos adicionales.	Ok
Se despliega la tarjeta y el servicio que se va a utilizar	Muestra las opciones a escoger.	Ok
Se ingresa la tarjeta en una máquina lectora de tarjetas virtual	Registra el número de tarjeta y si no es correcto termina el proceso	No
Se imprime el boucher y termina la compra.	Retorna los intereses y se imprime el boucher.	Ok

Fallas Encontradas	Descripción	Grav edad
Entra en un ciclo infinito si el número la tarjeta es incorrecto.	El timer sigue leyendo constantemente el error de número de tarjeta.	Grave
Comentarios de la prueba :		
La prueba demostró que el sistema funciona con el uso concurrente y se detectó que no se validaba un ingreso erróneo de la tarjeta de crédito.		
Nombre Ejecutor de la Prueba: David <div style="text-align: right; margin-top: 20px;"> _____ Firma Ejecutor de la Prueba </div>		

Cuadro 5.4 Resultados del Escenario 4

Escenario 5 (Caja blanca)

Requisitos		
Procedimiento	Descripción	ok
Pruebas Previas Requeridas:	Ingreso de usuarios	Ok
Requisitos Funcionales:	Usuarios: Personal que trabaja en un punto de venta	Ok
Ambiente Técnico Previo Requerido:	PC Pentium IV → 512Mb RAM Windows Me o superior Internet Explorer 5.x o superior Firefox 1.5.0.6	Ok
Comentarios: Precio esta prueba se tuvo que ingresar 200 artículos y 100 reglas de descuento y precios.		
Secuencia de la Prueba		
Procedimientos	Resultados Esperados	ok
Ingreso a la pantalla Punto de venta para intranet y la página de compras para Internet.	Al ingresar se deben inicializar todas las variables y objetos gráficos correctamente.	Ok
Realizar una compra por Internet.	Los artículos se siguen incorporando a la lista de lo que se va a comprar.	Ok
Ejecutar la compra.	Calcula precios y adicionales de todos los artículos.	Ok
Se calcula el costo de envío según la forma de envío elegida por el cliente	Indica como un adicional más el costo del envío.	No
Fallas Encontradas	Descripción	Grav edad
El recalcule en las formas de pago	El frame no se estaba actualizando correctamente con los nuevos totales.	Grave

Comentarios de la prueba :

La prueba demostró que los tiempos de respuesta eran pequeños a pesar del volumen de artículos y reglas. Aplicando correctamente las funciones en javascript en los frames la actualización de los datos en todos los frames fue exitosa.

Nombre Ejecutor de la Prueba:

David Barrigas

Pablo Barriga

**Firma Ejecutor de la
Prueba**

Cuadro 5.5 Resultados del Escenario 5

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 INTRODUCCIÓN

Concluido el desarrollo del módulo de puntos de venta en ambiente Internet e intranet, basados en el marco teórico y los requerimientos del sistema se construyeron los modelos de análisis y se desarrolló el código.

Al unir la teoría con la práctica los objetivos fueron cumplidos exitosamente, tomando como punto de partida los requerimientos del usuario se hallaron soluciones efectivas, la investigación de código innovador produjo que los módulos tengan una usabilidad simple, rápida y eficiente.

El presente documento contiene el análisis final, basado en los objetivos de la tesis y las recomendaciones para cumplirlos.

6.2 CONCLUSIONES

El objetivo general indica que se debe desarrollar un punto de venta para intranet y otro para Internet, desde este punto de vista los objetivos se duplican, el desarrollo de un mismo objetivo cambia dependiendo del ambiente en el que se quiera aplicar.

6.2.1 Conclusiones generales

La teoría es punto de partida esencial para el desarrollo de la tesis, pero en el momento de la práctica no tiene validez universal, ya que se debe adaptar esa teoría al ambiente en el cual se esté trabajando.

El uso de varias bases de datos restringe las consultas SQL y esta es la razón de que muchas consultas sean más elaboradas y con un alto grado de complejidad.

La programación para Internet basada en el genérico para Web ayudó en las páginas simples, pero para necesidades específicas como una pantalla con objetos visuales personalizados dificultó el trabajo.

La validación en la interfase del cliente con javascript mejoró el rendimiento de la aplicación y minimizó los posibles errores en el procesamiento de los datos enviados a la capa de datos al servidor.

6.2.2 Cumplimiento de objetivos

6.2.2.1 Integración

El ERP Venture se adapta fácilmente a cualquier módulo, para el desarrollo fue necesario crear opciones y permisos para las ventanas del punto de venta adicionalmente se crearon algunas variables de sistema.

6.2.2.2 Dispositivos

El lector de código de barras tiene una entrada compartida con el teclado por lo que su utilización es sencilla, envía el número del código de barras leído como si se lo ingresara por el teclado y finaliza con un salto de línea (Intro).

Para la utilización de la pantalla táctil es necesario tener una interfaz que permita trabajar solo con el mouse, pensando que los dedos son un poco más grandes que un puntero de mouse normal.

Las impresoras POS funcionan con el driver genérico de texto y el programador es el encargado de colocar el tamaño de papel y los márgenes para que salga bien la impresión.

6.2.2.3 Reglas del negocio

La mejor forma de hacer que un programa se adapte a múltiples reglas de negocio es creando tantas tablas como variables que influyan en las reglas de

negocio y relacionarlas con una tabla que valore la combinación de estas variables.

6.2.2.4 Metodología de desarrollo

OMT con UML mostró ser una metodología madura y fácil de integrar en todo el proceso de desarrollo, los diagramas generados guiaron la programación orientada a objetos tanto en Power Builder como en Java.

6.2.2.5 Tecnologías de desarrollo

Power Builder se desarrolla en un ambiente totalmente orientado a objetos, además de esto contiene una clase Datawindow que permite al programador tener un control completo y genérico sobre la base de datos.

Java tiene una gran cantidad de recursos para el desarrollo compatibles con múltiples plataformas y servidores de aplicaciones.

6.3 RECOMENDACIONES

Realizar una investigación de campo detallada para saber la factibilidad en el uso de algunas tecnologías y requerimientos específicos del sistema

Se debe recopilar una base de consultas SQL que funcionen en todas las bases de datos con las que se va a trabajar, cuando un sistema maneje varios motores de datos.

Tomar en cuenta como están desarrollados los módulos, y la comunicación entre estos, cuando se trabaja dentro de un sistema modular.

Para mejorar la usabilidad del sistema es necesario dar al usuario varios métodos de búsqueda y tener valores por defecto.

Trabajar siempre con los drivers propios de cada dispositivo ya que ofrecen una garantía de funcionamiento.

Probar las funciones de javascript a utilizarse por lo menos en Firefox e Internet Explorer.

Utilizar solo los diagramas necesarios con la metodología OMT con UML ya que se puede detallar el desarrollo del proyecto tanto como se desee, pero se llenaría de documentación poco aprovechable. Así, un diagrama de clases bien hecho ayuda a crear el modelo lógico mientras que el diagrama de estados se utilizará solo con los objetos que se desee especificar los cambios que sufre en algún proceso o evento; mientras que el diagrama de secuencia solo se tiene que hacer con los procesos donde existe bastante flujo de información y pueda existir algún error.

ÍNDICE DE DE CONTENIDOS ALFABÉTICO

CONCLUSIONES	233
Conclusiones generales	233
Cumplimiento de objetivos	234
DEFINICIÓN DE ESTÁNDARES	96
Estándares de diseño y programación	96
Estándares de codificación	102
Diseño de interfaces de usuario	106
DESARROLLO DEL CÓDIGO	179
Descripción del código para intranet (Power Builder)	179
Descripción del código para Internet (Java)	187
DISEÑO DE DIAGRAMAS	132
Diagrama de procesos del negocio	132
Diagrama de casos de uso	133
Diagrama de clases (intranet)	148
Diagrama de objetos (visuales, no visuales) para intranet	149
Diagrama de secuencia (intranet)	152
Diagrama de actividades (intranet)	161
Diagrama de casos de uso (Internet)	174
Diagrama de clases (Internet)	174
Diagrama de objetos (visuales, no visuales) para Internet	175
Diagrama de secuencia (Internet)	176
Diagrama de actividades (Internet)	177
ESPECIFICACIÓN DE REQUERIMIENTOS INFORMÁTICOS	112
Introducción	113
Objetivos	113
Alcance	113
Definiciones de acrónimos y abreviaturas	114
Descripción general	115
Perspectiva del producto	116
Características del usuario	119
Limitaciones generales	120
Supuestos y dependencias	120
Requisitos funcionales	120
Requerimientos de Rendimiento	128
Restricciones de diseño	128
HERRAMIENTAS DE ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	84
Power Designer 9.5	84
Power Builder 8.0	87
Sun Application Server 8	89
NetBeans	92
INTEGRACIÓN CON LOS DISPOSITIVOS POS	197
Lector de código de barras	197
Impresora POS	197
Pantalla táctil	197
Lector de tarjetas de crédito	198
Otros dispositivos	200
INTEGRACIÓN FUNCIONAL CON EL ERP	200
Integración en la base de datos	201
Integración funcional	201
Jerarquía modular	205

METODOLOGÍA	29
Metodología orientada a objetos OMT con UML	29
Metodología de la especificación de requerimientos	56
Metodología POS (Point of Sale)	62
MODELO FÍSICO DE DATOS	178
MODELO LÓGICO DE DATOS	131
PARADIGMA DE DESARROLLO	93
Paradigma Incremental (Punto de Venta tradicional)	93
Paradigma en Espiral (Punto de Venta virtual)	94
PLAN DE PRUEBAS	207
Introducción	207
Definición general de las pruebas	208
Recursos	209
Procedimiento para escenarios de prueba	211
Criterios de Aceptación	213
Apéndices	213
PROCESOS, PROCEDIMIENTOS Y ACTIVIDADES	109
Configuración del Módulo	110
Venta por Intranet e Internet	111
RECOMENDACIONES	235
VENTURE COMO ERP	21
Particularidades del ERP Venture	24
Módulos implementados del ERP Venture	27
Módulo de Puntos de venta	28
VENTURE COMO ORGANIZACIÓN	18

BIBLIOGRAFÍA

HIEBER, Chetney. Advanced Powerbuilder 8.0. USA. Envision Software Systems. Septiembre 1, 2001.

ARMSTRONG, Bruce; BROWN, Millard. PowerBuilder 9: Advanced Client/Server Development. USA. SAMS; 1ra edición. Julio 25, 2003

KURNIAWAN, Budi. Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions. USA. SAMS; Bk&CD-Rom. Abril 12, 2002

JAMES RUMBAUGH, OMT Insights: Perspective on Modeling from the Journal of Object-Oriented Programming (SIGS Reference Library). 2003.

JON HOLT. UML for Systems Engineering: Watching the Wheels. Institution of Electrical Engineers. 2001.

STEPHEN T. ALBIN. The Art of Software Architecture: Design Methods and Techniques, John Wiley & Sons, 2003.

MURRAY R. CANTOR. Object-Oriented Project Management with UML. John Wiley & Sons. 1998.

JOAN PECKHAM AND SCOTT J. LLOYD. Practicing Software Engineering in the 21st Century. Idea Group Publishing. 2003.

WEBBIBLIOGRAFÍA

Usable Web **Internet:** <http://www.usableweb.com/> **Último Acceso:** 15/06/2004

WebWorld Usability Weblob **Internet:** <http://www.usabilitynews.com/> **Último Acceso:** 15/06/2004

Human-Computer-Interaction on the Internet
Internet: <http://www.degraaff.org/hci/index.html/> **Último Acceso:** 15/06/2004

Human-Computer Interaction Resources **Internet:** <http://www.hcibib.org> **Último Acceso:** 15/06/2004

Empresa Calitrol **Internet:** <http://www.controlgram.com/> **Último Acceso:** 21/07/2004

Clarkson Laboratory & Supply Inc **Internet:** <http://www.clarksonlab.com/v0s.htm>
Último Acceso: 21/07/2004

EAN Internacional **Internet:** <http://www.ean-int.org/> **Último Acceso:** 21/07/2004

Uniform Code Council **Internet:** <http://www.uc-council.org/> **Último Acceso:** 21/07/2004

3M Worldwide **Internet:**

<http://www.3m.com/3mtouchsystems/Products/Capacitive/NFI.jhtml> **Último Acceso:** 21/07/2004

ANEXO A

A. PROCESO DE VENTA PUNTO DE VENTA

Descripción del Punto de Venta

A partir del siglo XIX con la aparición de los grandes almacenes, se produjo un importante cambio en las condiciones de venta:

- El producto se pone al alcance del consumidor eliminando el mostrador y el dependiente.
- El vendedor actúa de mero consultor por lo que su participación no es imprescindible.
- Se generaliza la libre circulación de mercancías y las grandes superficies.

Esta revolución en las condiciones de venta es lo que originó el punto de venta, mundialmente utilizado en empresas grandes y pequeñas.

El punto de venta conocido como POS es utilizado por empresas que buscan vender sus productos o servicios permitiendo que el cliente interactúe en todo el proceso de venta de manera rápida. La tecnología que se utilizan en los puntos de venta permitirá que la venta para el cliente sea rápida, ágil y segura.

En un POS se busca atraer al cliente por medio de técnicas de marketing y merchandising, se maneja todo el tiempo políticas de descuento basados en temporadas y promociones que desee aplicar la empresa.

En un punto de venta encontraremos el producto o servicio de forma atractiva para el cliente, algunas características son comunes para todos los puntos de venta:

- Ubicación preferente de producto. Se trata de situar el producto en lugares donde aumentan las posibilidades de ser adquirido por el consumidor.
- Pilas y exposiciones masivas de producto.

- Carteles de promociones
- Señalizadores. Indicadores como flechas o líneas en el suelo que sirven para dirigir el tráfico y la atención de los consumidores.
- Publicidad en punto de venta. Expositores o presentadores de producto de carácter permanente o temporal.
- Demostraciones y degustaciones. Suelen tener mucha aceptación en los establecimientos y provocar importantes incrementos de venta.
- Animación en punto de venta. Conjunto de acciones promocionales que se celebran en un establecimiento durante un tiempo determinado con motivo de un acontecimiento particular. Por ejemplo: San Valentín, Navidad, etc.

En un punto de venta de una empresa se encontrará una caja registradora que tendrá un sistema de Punto de Venta, el proceso tradicional es que el cliente escoge sus productos o servicios basado en sus preferencias o necesidades y el cajero o vendedor le ayudará con el pago. Las tiendas en centros comerciales utilizan este proceso de venta, por ejemplo Supermaxi (supermercado de alimentos), Juguetón (tienda de juguetes), Tower Records (tienda de música).

La tecnología utilizada en un POS permite que este proceso se agilite notablemente, en el más simple punto de venta muchas empresas adquieren una caja registradora que solo permite mantener un control de las ventas diarias y una lista de productos limitada.

En empresas con mayores recursos, adquieren una gran variedad de dispositivos POS como pantallas táctiles, lectores de código de barras, pesas electrónicas, dispositivos de pago de tarjetas de crédito, etc. Todos estos dispositivos facilitan notablemente el proceso de venta. También invierten en sistemas robustos que pueden ser distribuidos o centralizados, o ser parte de un ERP que integre todos los procesos de la empresa para mantener un control efectivo de la venta de un producto.

La meta de un punto de venta es que el cliente sea capaz de interactuar en todo el proceso inclusive en el pago, varias tecnologías están siendo desarrolladas, como etiquetas inteligentes que estarán en los artículos de un supermercado que permitirán que por medio de un sensor determine el monto total a pagar sin la necesidad de que un cajero ingrese estos ítems al sistema. Incluso existen empresas que desarrollan puntos de venta similares a un cajero automático para adquirir algún servicio o producto.

ANEXO B

B. INTEGRACIÓN DEL PUNTO DE VENTA INTRANET FUNCIONALMENTE CON EL ERP EN USO

El ERP en uso es el sistema que se está comercializando, es decir que al integrar este módulo las empresas que actualmente son clientes de Venture ya dispondrían de este nuevo servicio.

Este anexo comprende los pasos para la integración y los cambios que se hicieron en el código, tanto en el ERP como en el módulo de puntos de venta.

Pasos para la integración

Para detallar los pasos a seguir para la integración, a estos se los a dividido en tres grupos, a nivel de base de datos, a nivel del ERP y finalmente a nivel del punto de venta.

1 Cambios en la base de datos

Como el ERP presenta un actualizador automático solo se realizo una copia de las actualizaciones al actualizador del ERP adicionando las relaciones entre la cabecera de la factura con la cabecera de las cuentas por cobrar y del detalle de las facturas con el detalle del inventario.

Todas las actualizaciones se pusieron en una sola con el nombre “Campos y tablas necesarias para el Punto de venta” la cual se corre desde la ventana de utilitarios en la pestaña de Actualizaciones.

2 Cambios en el ERP

Para la integración con el ERP además de cambios a la base de datos es necesario hacer unos cambios en código del ERP, estos cambios se presentan a continuación:

1. Creación de variables

Por decisión del gerente técnico se tuvieron que cambiar el nombre de las variables y crear nuevas para la relación con los demás módulos.

is_v_adic_iva	is_v_en_linea_ventas_a_contab
is_v_cliente_defecto_pv	is_v_en_linea_ventas_a_cxc_ge
is_v_cod_dddw_agencia_distribu	is_v_en_linea_ventas_a_invent
is_v_cod_estado_anulado_factur	is_v_formato_cliente
is_v_cod_estado_cliente_activo	is_v_formato_vendedor
is_v_cod_estado_normal_factura	is_v_fpago_nec_cliente
is_v_cod_forma_pago_efectivo	is_v_fpago_usa_det
is_v_cod_forma_pago_propio	is_v_fpago_usa_venta
is_v_cod_forma_pago_tarje_cred	is_v_politica_desc_subtotal
is_v_cod_vendedor_ninguno	is_v_politica_precios
is_v_dias_recuperar_documento	is_v_srvt_datafast
is_v_dir_arch_dat	is_v_srvt_medianet
is_v_dir_arch_med	
is_v_en_linea_ventas_a_bancos	

2. Crear los dataobject necesarios en la librería dtemp

Estos dataobject son los que se crean con el nombre de la tabla que representan y sirven para los Dropdowns.

j_adicional
j_agrupa_client
j_bodegaje
j_caja
j_cliente
j_entidad_pago
j_forma_pago
j_tipo_cliente
j_tipo_venta
j_transporte

3. Pasar el módulo de puntos de venta con el nombre `ventas_ge` por lo que ya existe un módulo para ventas flores que se llama `ventas`.

Esto se hace creando una librería copiando físicamente la librería en la carpeta de las librerías, cambiarle el nombre a `ventas_ge` y haciendo una migración en la aplicación del ERP.

3 Cambios en el Módulo de puntos de venta

1. Como se cambió el nombre de las variables se pasa la librería a código fuente y se busca las variables anteriores para cambiarlas por las nuevas.
2. Se incluyó en las ventanas el control del inventario llamando a la clase `n_cst_requisicion`, la que tiene la función `of_saldo(string as_articulo)` la cual me devuelve el saldo del artículo, saldo que es controlado si está activada la variable `is_V_en_linea_ventas_a_invent`.
3. Se creó una nueva clase `n_cst_ventasg_procesos` que también es llamada desde las ventanas de puntos de venta y de facturación para que en momento de grabar mermita mandar los datos de la venta a inventario, cuentas por cobrar, contabilidad y bancos, cada uno de estos procesos se inician si la variable correspondiente está activada.

La forma de realizar el paso de datos es llamando a las funciones propias de cada módulo y solo poniendo los datos que necesita cada función. La función que realiza todo este proceso es `of_facturar(string as_cod_cfc)` a esta función se manda el código de la factura.

Siguiendo estos pasos el paso del módulo de puntos de venta funcionalmente al ERP y sus módulos es sencillo.

ANEXO C

C. PAYPAL

1. Descripción de PayPal

PayPal es una empresa internacional del grupo eBay que brinda soluciones para pagos en Internet, cuenta con millones de cuentas de usuarios en todo el mundo. Esta solución de PayPal está disponible para 45 países y 6 divisas: euros, dólares estadounidenses, libras esterlinas, yenes japoneses, dólares australianos y dólares canadienses.

El pago que se realiza a través de PayPal es gratis, tiene un fácil manejo y una alta seguridad, permite vincular una dirección de correo electrónico y número de la tarjeta de débito o crédito.

En la siguiente imagen se puede visualizar el funcionamiento de PayPal:

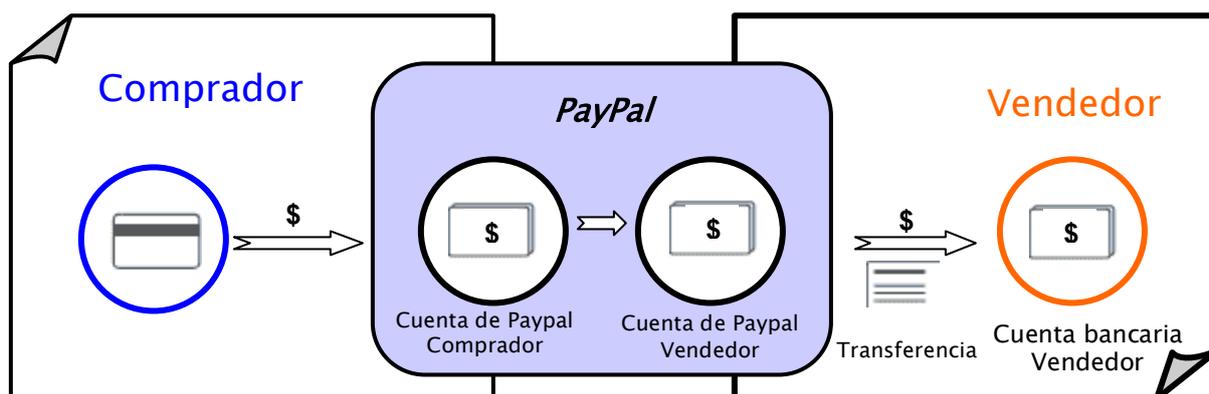


Fig. C.1 Proceso de pago Paypal

2. Herramientas de desarrollo Paypal

Para la implementación de las funciones de Paypal en el módulo de Punto de Venta se utilizó un kit de desarrollo (SDK) para java que contiene todas las librerías necesarias para una comunicación con Paypal segura y fácil.

Las librerías de Paypal contienen un Webservice que se comunica con Paypal, esta comunicación es totalmente transparente para el desarrollador, ya que solo tiene que inicializar la información de la empresa y luego ingresar los datos del cliente para la transacción.

Para empezar con la integración de PayPal se prosigue con los siguientes pasos:

- Primero debemos obtener una cuenta de desarrollador en <https://developer.paypal.com/>, la cual nos permitirá crear las cuentas de prueba de Paypal. Para crear cuentas se debe ingresar al portal de desarrolladores e ir a la opción Sandbox, allí se debe crear una cuenta de negocios que será la que reciba todos los pagos, también se debe generar varias cuentas personales para crear tarjetas de crédito a estos perfiles que nos permitan enviar pagos a la cuenta de negocios.
- Es necesario descargar las últimas versiones del SDK, este se encuentra en el portal de desarrolladores de Paypal La última versión para java es Paypal SDK 3.2. Si se necesita trabajar con otro lenguaje de programación Paypal tiene varios SDKs en varios lenguajes como asp, php, aspx.
- Paypal exige que todas las transacciones a su página sean encriptadas y seguras. En el portal de desarrolladores se tiene que descargar el certificado digital de la cuenta de negocios. El certificado viene en un formato de texto (cert_key_pem.txt:) el cual debe ser procesado con la herramienta OpenSSL para que sea válido, OpenSSL es parte del SDK mencionado anteriormente, el comando que crear el certificado es:
 - o `OpenSSL pkcs12 -export -in cert_key_pem.txt -inkey cert_key_pem.txt -out paypal_cert.p12.`
- Para verificar que nuestro certificado digital funcione correctamente se debe implementar los ejemplos de Paypal, el ejemplo está empaquetado con el formato WAR, este paquete se lo implementa utilizando la interfaz gráfica del servidor de aplicaciones o se lo coloca en la carpeta de autodeploy del servidor de aplicaciones.

- La aplicación de ejemplo Figura 1.2 se muestra una cuenta por defecto y también para ingresar nuestra cuenta de prueba, los campos que debemos ingresar son:
 - o **API usuario:** Es la cuenta que nos asigna Paypal de desarrollador para una cuenta de negocios, no es la misma cuenta con la que nos registramos en el portal de desarrollador.
 - o **API clave:** Password configurado para la cuenta
 - o **Certificado digital:** El certificado digital generado con OpenSSL
 - o **Clave privada del certificado digital:** La clave con que se generó el certificado digital.
 - o **Entorno:** Para las pruebas se utiliza el entorno Sandbox

- Las cinco variables mencionadas se tendrían que configurar para cada empresa que desee realizar transacciones con Paypal.

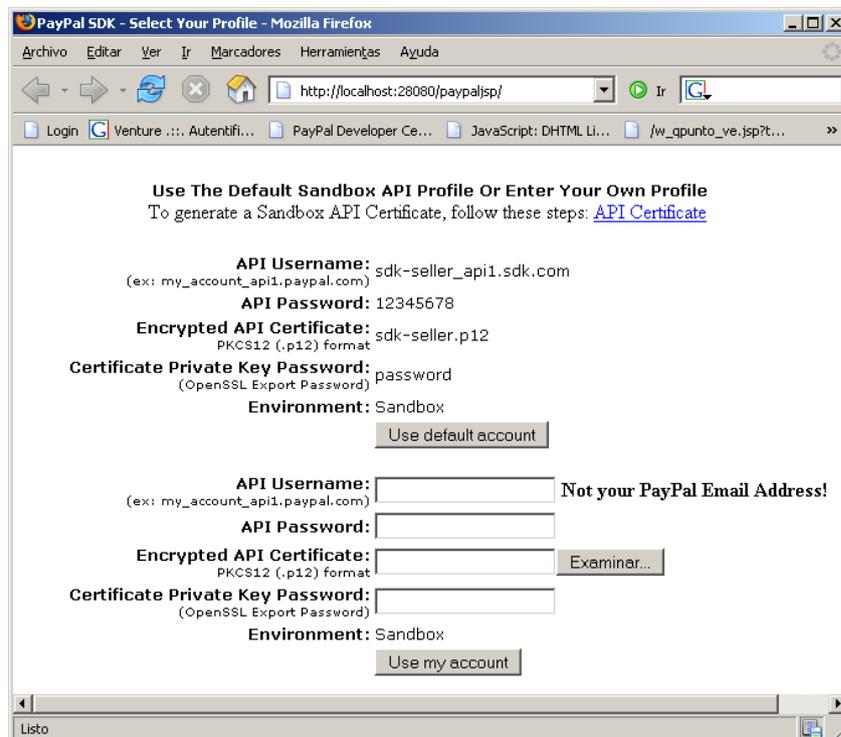


Fig. C.2 Página de ejemplos de Paypal

- Ingresados todos los datos mencionados, podemos realizar transacciones con nuestra cuenta a Paypal. Para observar si las

transacciones fueron exitosas podemos ingresar al portal de desarrolladores e ingresar en la cuenta y visualizar las transacciones realizadas.

3. Implementación

El entorno SandBox de Paypal es solamente para pruebas, ninguna de las transacciones realizadas allí son reales, para implementar transacciones reales se debe registrar una cuenta de negocios en el sitio oficial de Paypal www.paypal.com , se solicita una cuenta para transacciones, con la cual nos genera un usuario, clave y certificado digital. El entorno en producción es Paypal, configurando estos parámetros en el sistema las transacciones que realicemos con la tarjeta de crédito en el sistema se reflejará en el estado de cuenta del cliente y en el balance general de la cuenta de negocios en Paypal.

ANEXO D

D. MANUAL DE INSTALACIÓN DEL ERP VENTURE

Instalación del ERP Venture para intranet

Para la instalación del ERP Venture en su versión para intranet es necesario los siguientes requisitos:

- Un equipo servidor de base de datos. La capacidad del servidor depende el movimiento que vaya a tener la base pero con un CPU Pentium 4 con 1 GB en RAM funciona perfectamente.
- Un equipo cliente que como base tenga con un procesador Pentium 3 y 128 MB en RAM.
- El equipo cliente tiene que funcionar con una versión superior del sistema operativo de Windows 98.
- Una red 10/100 entre las computadoras clientes y el servidor de la base de datos.
- Configuración de los formatos utilizados por el sistema, esto se lo realiza en la configuración regional. (separación de decimales con punto '.', separación de miles con coma ',' y el formato de la fecha aaaa-mm-dd).
- Para la bases de datos Anywhere y para SQL Server configurar en el cliente un ODBC con el nombre base y base_sqlserver respectivamente.

Pasos para la instalación

La instalación del ERP Venture para Intranet consiste en los siguientes pasos:

Crear una carpeta llamada venture en unidad de disco duro D de ser posible, de no tenerlo, crearla en la unidad de disco duro C.

Dentro de la carpeta creada se procede a copiar las librerías y el ejecutable del ERP en conjunto con las librerías propias de Power Builder 8.03.

Se levanta la base de datos en el servidor previo la configuración de los datos iniciales, tales como la compañía, la sucursal, al menos un usuario con los sus respectivos permisos.

Antes de ejecutar el ERP existe un archivo que también es copiado en la carpeta creada y es "database.ini", el cual contiene los valores iniciales del sistema, entre estos tenemos la base de datos que va a utilizar, la compañía y sucursal por defecto, la caja que funcionará en esa máquina, entre otros.

Al ejecutarse el ERP pedirá las variables de sistema que no han sido configuradas previamente por el asistente de Venture.

Finalmente para cada pantalla se configurará la ventana de opciones, para cada reporte se configurará la pantalla de reportes y se dará permisos de las ventanas y los reportes a los usuarios que se desee.

Instalación del ERP Venture para intranet

Para la instalación del ERP Venture en su versión para internet son necesario los siguientes requisitos:

- Un equipo servidor de aplicaciones Web. Existen equipos Sun Solaris, IBM o HP que pueden ser adquiridos en el mercado.
- El equipo servidor debe tener instalado el servidor de aplicaciones Sun Application Server 8 o una versión superior.
- Un equipo servidor de base de datos de ser necesario.
- Un equipo cliente que como base tenga con un procesador Pentium 3 y 128 MB en RAM y un explorador.
- Tanto cliente como servidor deben tener una conexión a internet considerable, para el servidor superior a 128 Kbps y para el cliente como base 64 Kbps.

Pasos para la instalación

La instalación del ERP Venture para Internet consiste en los siguientes pasos:

Copiar la aplicación empresaria “web_venture.ear” en la carpeta donde se encuentran las aplicaciones empresariales del servidor de aplicaciones Web. Este archivo “web_venture.ear” es el conjunto de los archivos war y jar, que son en sí la aplicación.

Levantar la base de datos en el servidor. Este puede ser un servidor a parte del que contiene la aplicación. En algunos casos se utiliza el mismo.

Configurar en el servidor de aplicaciones la conexión JDBC a la base de datos (Ver anexo G).

ANEXO E

E. MANUAL DE USUARIO DEL PUNTO DE VENTA EN INTRANET

Para usar el punto de venta se debe tomar en cuenta dos procesos, uno previo a la venta y otro que sería el propio proceso de la venta.

Procesos previos al uso del punto de venta

Para usar el módulo de puntos de venta es necesario antes de usarlo seguir los siguientes pasos:

1. Correr todas las actualizaciones de los puntos de venta para poder tener las tablas y los cambios hechos en las tablas existentes.

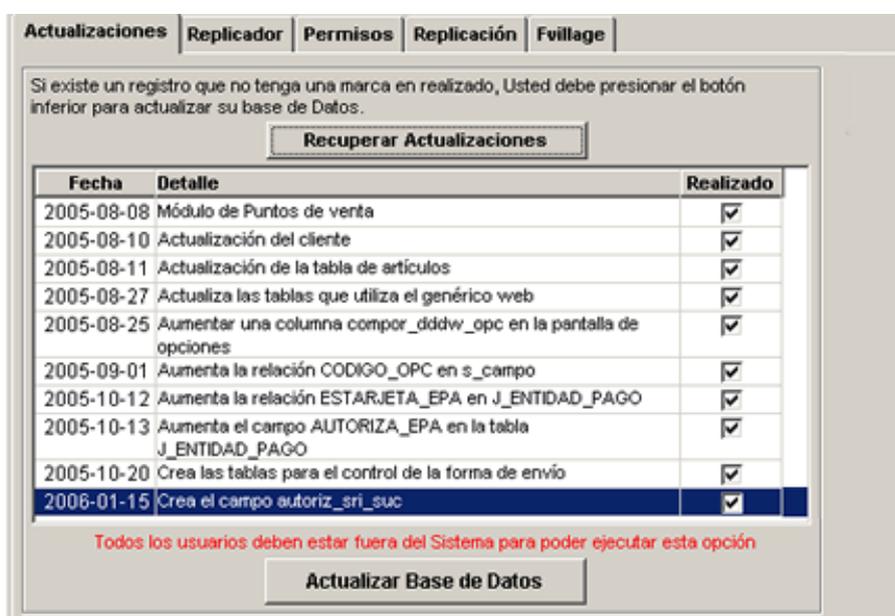


Fig. E.1 Ventana de actualizaciones del ERP Venture

2. Crear todas las opciones, reportes y sus respectivos permisos por cada usuario.

Nombre Opción	Ventana	Tabla 1	Tabla 2
Adicionales	w_simple	j_adicional	
Agencia de distribución	w_simple	j_agenci_distri	
Bodegaje	w_simple	j_bodegaje	
Cajas (punto de venta)	w_simple	j_caja	
Períodos de venta	w_periodo_pv	j_periodo	
Estado de cotización	w_simple	j_estado_cotiza	
Estado de factura	w_simple	j_estado_factur	

Estado de pedido	w_simple	j_estado_pedido	
Estado de cliente	w_simple	j_estado_client	
Grupos de clientes	w_simple	j_agrupa_client	
Monto de venta	w_simple	j_monto_venta	
Rangos de Plazo	w_simple	j_rango_plazo	
Temporada	w_simple	j_temporada	
Tipo de cupo	w_simple	j_tipo_cupo	
Tipo de venta	w_simple	j_tipo_venta	
Tipos de cliente	w_simple	j_tipo_cliente	
Transporte	w_simple	j_transporte	
Servicio para tarjetas	w_simple	j_servic_tarjet	
Forma de pago con tarjeta	w_simple	j_for_pag_tarje	
Entidades de pago	w_simple	j_entidad_pago	
Formas de pago	w_doble	j_tipo_pago	j_forma_pago
Clientes PV	w_cliente_pv	j_cliente	
Vencimiento cliente	w_vence_cle_pv	j_vencimiento	
Volumen venta	w_simple	j_volumen_venta	
Descuentos	w_descuento_pv	j_descuento	
Precios	w_precio_pv	j_precio	
Vendedor	w_vendedor_pv	j_vendedor	
Cotización	w_cotiza_pv		
Pedidos	w_pedido_pv		
Facturación	w_factura_pv		
Punto de venta	w_punto_venta		
Forma de envío	w_forma_envio	j_cab_form_envi	j_det_form_envi

Tabla. E.1 Opciones del módulo Puntos de venta

Reportes
Movimientos PV Fecha Vendedor
Movimientos PV Caja Fecha
Movimientos Caja
Imprimir nota de venta

Tabla. E.2 Reportes del módulo Puntos de venta

3. Ingresar los datos en las siguientes ventanas:
 - a. Adicionales.- se ingresan todos los adicionales que pueda tener una factura o nota de venta. Ej.
IVA, Recargo tarjeta, Descuento, Interés tarjeta y Recargo crédito
Con su respectivo signo si suma o resta el valor de la factura y si suma o no al subtotal.

- b. Artículos.- en el módulo de compras se tiene que ingresar todos los artículos que se van a vender y llenar las columnas de nombre, código de barras, precios y si carga IVA.
La ventana de artículos necesita a su vez otros datos como estado del artículo (activo, inactivo), marca, nivel (grupo, artículo) y las unidades con las que se comercia.
- c. Cajas.- Esta ventana contiene las cajas o estaciones de venta que se utilizarán como punto de venta.
- d. Clientes.- Esta ventana se ingresa en forma de árbol para formar grupos de clientes.
Para poder ingresar un cliente es necesario previamente ingresar estado cliente (activo, inactivo), tipo, grupo, lugar geográfico y el tipo de identificación (cédula, RUC).
- e. Estado de factura.- Esta ventana contiene los estados posibles de una factura, los básicos son normal y anulado.
- f. Entidades de pago.- Esta ventana contiene todas las entidades con las que se va a trabajar, incluyendo bancos y tarjetas.
- g. Forma de pago con tarjeta.- esta contiene los planes que puede utilizar cada tarjeta de crédito y el porcentaje de interés. Esta ventana se llenará dependiendo de los datos escritos que envíe Datafast, cada línea se relacionará con una tarjeta.
- h. Formas de pago.- esta ventana consta de dos datawindows, teniendo en el primero los tipos de pago (contado, crédito) y la segunda el pago en si por cada clase (contado: cheque, efectivo; crédito: tarjeta, crédito propio).

- i. Vendedor.- Esta ventana se llena en forma de árbol para tener grupo de vendedores.
 - j. Precios.- Esta ventana pertenece a las reglas de negocio, es en esta ventana donde por medio de arreglos de orden, artículo, cliente, temporada, lugar geográfico y volumen de venta se puede tener un precio distinto por artículo o por artículos.
 - k. Descuentos.- Esta es la segunda ventana que guarda las reglas de negocio, en este caso se ponen las reglas para aplicar los adicionales en porcentajes automáticamente basándose en el artículo, orden, temporada, lugar geográfico, monto de venta, cliente, forma de pago, plazo y volumen de venta.
4. Después de haber ingresado los datos en las tablas, se prosigue a llenar los valores de las variables del sistema para puntos de venta, estas variables serán pedidas por el sistema automáticamente al inicio. La explicación de cómo llena cada variable está dada en el momento de configurarlas.
5. Finalmente se tiene que configurar el archivo database.ini que es donde se pondrá la caja que representa cada computador. Ej.

[ventas]

caja= 0010001

Procesos para el manejo del punto de venta

Para el proceso mismo de la venta se utiliza la ventana de puntos de venta, en su forma clásica o para ser usada con una pantalla táctil, la funcionalidad es la misma, la única diferencia es un teclado numérico que permitirá ingresar cantidades u comando en lugar de utilizar el teclado, claro que se limita a la venta basada en códigos y no en letras y se puede ingresar nuevos clientes.

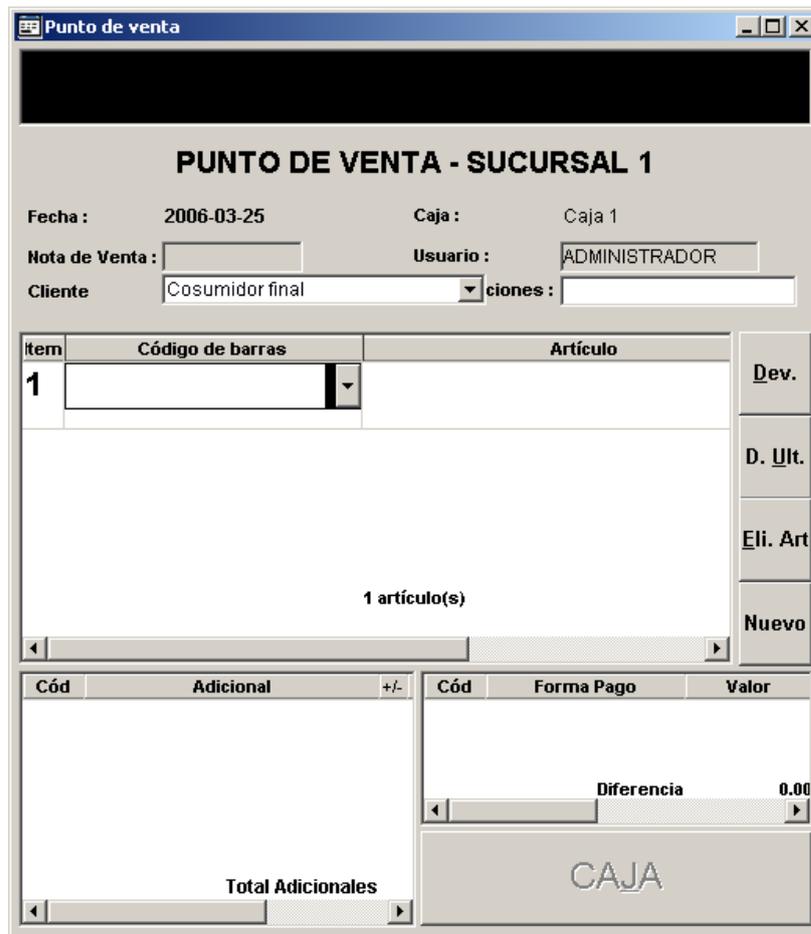


Fig. E.2 Ventana Punto de venta

Para ver correctamente el funcionamiento de esta ventana es necesario dividirla en tres partes: datos iniciales y registro de artículos, formas de pago y finalización de la venta.

a. Datos iniciales y registro de artículos

Existen datos que se ponen por defecto como la fecha, la caja, el usuario y el cliente dependiendo de la variable de sistema, si se tiene un cliente por defecto (recomendado). Los datos iniciales que pueden ser cambiados son el cliente y la observación.

Al hacer clic derecho se presenta una pantalla para ingresar un nuevo cliente mientras que si se hace doble clic se puede buscar un cliente por su identificación.

Los artículos tienen tres formas de ser ingresados: digitando el código de barras, buscando el artículo expandiendo el dropdown del código de barras o buscando con letras guías en el campo del artículo, para este último es necesario volver al campo de código de barras para que continúe el proceso.

El campo del código de barras sirve también para poner la cantidad del producto escribiendo el número y el signo '*' o también el dinero que se recibe para que de el vuelto si se va a pagar con efectivo poniendo la cantidad y el signo '\$'.

Cada vez que se hace cualquier cambio en el datawindow de los artículos el datawindow de las formas de pago queda vacío y se recalculan los adicionales.

También existen botones especiales como:

Dev.: Este permite seleccionar la cantidad y el artículo que se desea devolver en el momento de la compra.

D. Ult.: Este permite devolver el último artículo ingresado, con la cantidad que se ingresó.

Eli. Art.: Este permite eliminar totalmente un artículo de la lista, este se utiliza previo la selección del artículo en el datawindow.

Nuevo: Este botón se lo utiliza cuando se desea eliminar toda una venta y empezar de nuevo.

b. Formas de pago

Para iniciar la forma de pago se hace clic en el botón Caja, con esto se inserta automáticamente una línea con la forma de pago en efectivo y si se ingresó precio a esto el pago del cliente la pantalla superior indica cual es el vuelto.

Los procesos que cumplen las formas de pago dependen de la forma de pago a utilizar:

- Efectivo y cheque: El pago es simple y puede ser utilizado solo como una parte del pago.

- Crédito propio y cheque posfechado: para estas formas de pago se despliega un nuevo datawindow el cual contiene los plazos y las cantidades a pagar en cada plazo.
- Tarjeta de crédito: Para esta forma de pago se guarda la factura con estado anulado para poder enviar el código a la empresa que preste el servicio, en caso de que sea Datafast se presenta una ventana donde se elige el plan de pago con tarjeta, en el caso de Medianet es la misma máquina lectora la que da a seleccionar el plan de pago.

Tanto la ventana de pago como la de plazos pasa verificando que los valores a ingresar sean correctos. Una forma de ayuda para completar el pago cuando se realiza varias formas de pago es haciendo doble clic sobre la cantidad.

Después de hacer cualquier cambio en la forma de pago o en los plazos el datawindow de adicionales se actualiza.

c. Finalización de la venta

Al hacer un nuevo clic en el botón Caja cuando ya se ha definido correctamente las formas de pago, se graba la factura con estado Normal, se imprime la nota de venta y se prepara la ventana para una nueva venta.

ANEXO F

F. MANUAL DE USUARIO DEL PUNTO DE VENTA EN INTERNET

Procesos previos al uso del punto de venta Web

Para la instalación del módulo se deben seguir los siguientes pasos:

- En el equipo servidor de aplicaciones Web debe encontrarse instalado el servidor de aplicaciones Sun Application Server 8 o una versión superior, puede ser descargado de la página de Sun Microsystems.
- La base de datos debe estar configurada y con las tablas de datos del módulo de puntos de venta.
- Con el servidor de aplicaciones instalado se debe crear la conexión JDBC. Primero se crea un conjunto de conexiones el driver utilizado es `com.microsoft.jdbcx.sqlserver.SQLServerDataSource` , el nombre de la conexión debe ser `Base_sql` en el caso de utilizar la base SQL Server , si utiliza otro tipo de base de datos debe ser `Base_<base de datos>`, por ejemplo `Base_oracle` o `Base_sysbase`, los parámetros necesarios para el conjunto de conexiones son:

<input type="checkbox"/>	Embedded	false
<input type="checkbox"/>	SelectMethod	cursor
<input type="checkbox"/>	BatchPerformanceWork	false
<input type="checkbox"/>	NetAddress	000000000000
<input type="checkbox"/>	HostProcess	0
<input type="checkbox"/>	LoginTimeout	0
<input type="checkbox"/>	Password	amn110
<input type="checkbox"/>	SendStringParameters	true
<input type="checkbox"/>	User	sa
<input type="checkbox"/>	PortNumber	1433
<input type="checkbox"/>	ServerName	localhost
<input type="checkbox"/>	DatabaseName	base_pvd

Fig. F.1 Ventana de actualizaciones del ERP Venture

- Una vez creado se puede ingresar en el conjunto de conexiones y realizar una prueba para verificar que exista la conectividad con la base de datos, después se crea un recurso JDBC el cual recibe el mismo nombre del conjunto de conexiones y se lo relaciona con este.
- Para implementar el pago con tarjeta de crédito se debe instalar el Paypal Java SDK. Se debe verificar que en las variables de entorno del sistema se encuentre registrada la ruta de las librerías de Paypal, la variable de entorno es CLASSPATH.
- Se debe modificar los permisos de las políticas de seguridad en el servidor de aplicaciones en el archivo Server.policy, se adiciona la siguiente línea `grant {permission java.security.AllPermission;};`.
- Adicionalmente cada cliente debe modificar las políticas del java, añadiendo la siguiente línea a su java.policy `grant {permission java.io.FilePermission "<<ALL FILES>>", "read";};`.
- Se verificar que el equipo no tenga restricciones para el acceso de los cajeros a través de Internet o de una red privada.
- En el servidor de aplicaciones se implementa el archivo web_venture.ear, esto se realiza ingresando a la opción aplicaciones de empresa y presionando en el botón implementar, seleccionamos el archivo web_venture.ear y se procesa este archivo, luego verificamos que la aplicación este funcionando ingresando a la dirección <http://localhost:28080/venture> o en modo seguro <https://localhost:21043/venture>.

Procesos para el manejo del punto de venta Web

El cajero primeramente debe ingresar la IP asignada al servidor e ingresando en la carpeta venture <https://<IP SERVIDOR>:28080/venture>, al cajero le será asignado un usuario y clave para ingresar en el sistema, debe ir a la opción Punto de VentaG. Aparecerá una ventana similar al punto de venta generado en Powerbuilder como en la figura E.2.

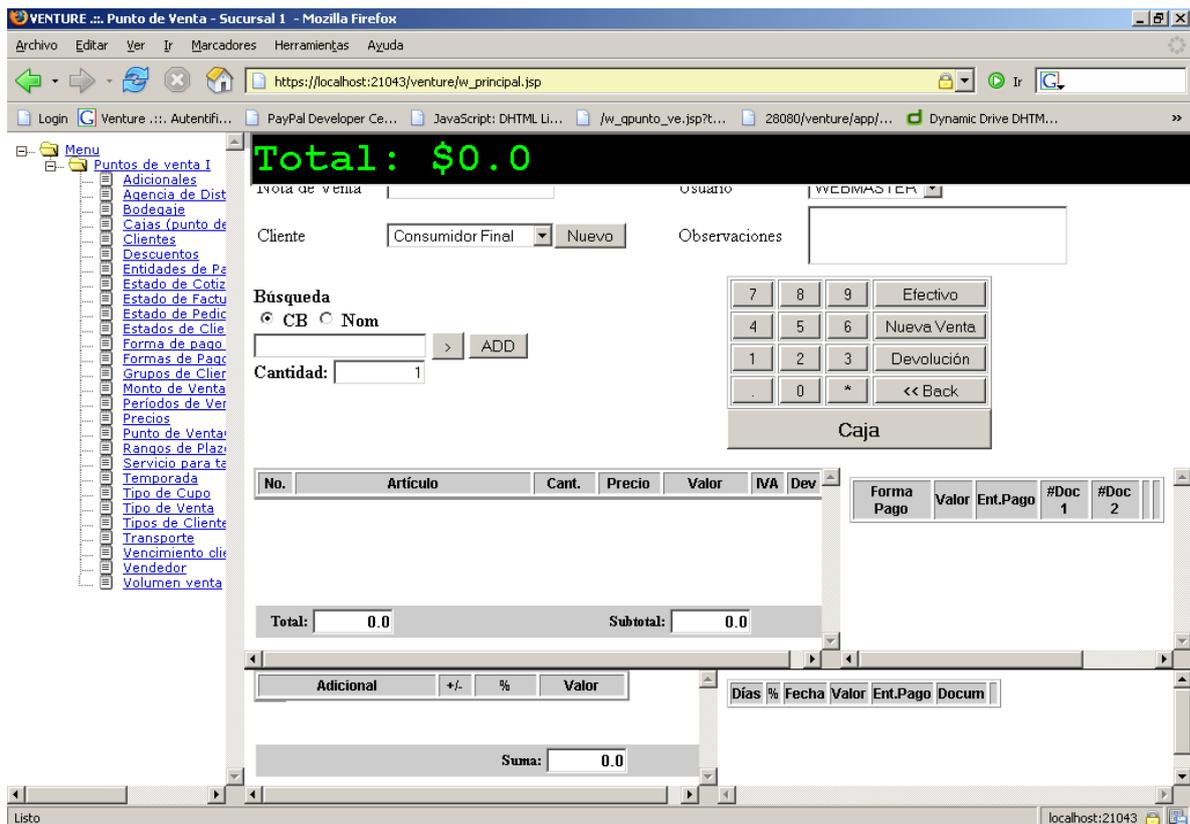


Fig. F.2 Pantalla del punto de venta

d. Datos iniciales y registro de artículos

Existen datos por defecto como la fecha, la caja, el usuario y el cliente dependiendo de la variable de sistema, si se tiene un cliente por defecto (recomendado). Los datos iniciales que pueden ser cambiados son el cliente y las observaciones.

Al presionar el botón nuevo a lado del combo de clientes se presenta una pantalla para ingresar un nuevo cliente. Los datos por defecto son tipo cliente,

grupo, tipo ID como se muestra en la figura E.3, los datos obligatorios son nombre, ID, teléfono, dirección y e-mail.

http://192.168.0.90:28080 - Mozilla Firefox

Nombre:

Tipo cliente:

Grupo:

Tipo ID:

ID:

Teléfono:

Dirección:

e - mail :

Listo

Fig. F.3 Pantalla de ingreso nuevo cliente

Los artículos tienen tres formas de ser ingresados: capturando el código de barras con el lector de códigos, buscando el artículo presionado el botón con el símbolo de mayor > y seleccionado el tipo de búsqueda, el cual permite una búsqueda por código de barras (CB) o el nombre del artículo (Nom).

Búsqueda

CB Nom

- Amor - vainilla - 175 G
- Arequipe - 500 G
- Avena Alpina 250g
- Desodorante - 24/7 - 50g
- Desodorante lady - Aloe - 45g

Fig. F.4 Pantalla de búsqueda artículo

El campo cantidad sirve para poner la cantidad del producto escribiendo el número en el teclado numérico de la parte derecha y el signo ‘*’ o también el dinero que se recibe para el vuelto si se va a pagar con efectivo poniendo la cantidad y el signo ‘-’.

Cada vez que se hace cualquier cambio en el detalle de los artículos o en las formas de pago se recalculan los adicionales.

También existen botones especiales como:

Efectivo.: Ingresa el pago en efectivo, para determinar el vuelto.

Nueva Venta.: Crea una nueva venta, eliminando los registros anteriores en la pantalla.

Dev.: Este permite seleccionar el artículo que se desea devolver en el momento de la compra

<<Back: Elimina por carácter valores ingresados en el campo de búsqueda.

Caja: Cierra el proceso de venta, validando artículos y formas de pago.

e. Formas de pago

Para iniciar la forma de pago se hace clic derecho en la cabecera de la forma de pago, se presiona insertar con esto se crea automáticamente una línea con la forma de pago en efectivo y si se ingresó precio a esto el pago del cliente la pantalla superior indica cual es el vuelto.

Forma Pago	Valor	Ent.Pago
Efectivo	0	

Insertar
Eliminar

Fig. F.5 Pantalla de cabecera de forma de pago

Los procesos que cumplen las formas de pago dependen de la forma de pago a utilizar:

- Efectivo y cheque: El pago es simple y puede ser utilizado solo como una parte del pago.
- Crédito propio y cheque posfechado: para estas formas de pago se ingresa en el detalle de la forma de pago el cual contiene los plazos y las cantidades a pagar en cada plazo.
- Paypal: Para esta forma de pago se presenta una pantalla de Paypal donde se realiza el pago con tarjeta de crédito y si la transacción fue correcta al campo de #Doc 1 envía el código de transacción.

Tanto la ventana de cabecera de forma de pago y la de detalle verifican que los valores ingresados sean correctos. Una forma de ayuda para completar el pago cuando se realiza varias formas de pago es haciendo doble clic sobre la cantidad.

Después de hacer cualquier cambio en la forma de pago o en los plazos los adicionales se actualiza.

f. Finalización de la venta

Al hacer un nuevo clic en el botón Caja cuando verifica las formas de pago, se graba la factura con estado Normal, se imprime la nota de venta y se prepara la ventana para una nueva venta.

ANEXO G

G. ESPECIFICACIÓN DE LOS DIAGRAMAS

MODELO LÓGICO DE DATOS

DICCIONARIO DEL MODELO FÍSICO DE DATOS

Nombre	Tabla	Primary Key	Foreign Key	Mandatory	Observaciones
ACUMU_MONTO_ART	c_articulo	NO	NO	NO	Tabla que permite el ingreso y edición de los artículos, es una tabla recursiva.
ALTERNATIVO_ART	c_articulo	NO	NO	NO	
ALTERNO_ART	c_articulo	NO	NO	NO	
C_U_CODIGO_UNI	c_articulo	NO	SI	NO	
C_U_CODIGO_UNI2	c_articulo	NO	SI	NO	
CARGA_ICE_ART	c_articulo	NO	NO	NO	
CARGA_IVA_ART	c_articulo	NO	NO	NO	
COD_BARRAS_ART	c_articulo	NO	NO	NO	
CODIGO_ART	c_articulo	SI	NO	SI	
CODIGO_CAC	c_articulo	NO	SI	NO	
CODIGO_COM	c_articulo	NO	SI	NO	
CODIGO_CUE	c_articulo	NO	SI	NO	
CODIGO_EAR	c_articulo	NO	SI	NO	
CODIGO_MAR	c_articulo	NO	SI	NO	
CODIGO_NAR	c_articulo	NO	SI	NO	
CODIGO_OCP	c_articulo	NO	SI	NO	
CODIGO_SUC	c_articulo	NO	SI	NO	
CODIGO_UNI	c_articulo	NO	SI	NO	
CREDI_TRIBU_ART	c_articulo	NO	NO	NO	
DEVOLUCION_ART	c_articulo	NO	NO	NO	
DIA_BODEGA_ART	c_articulo	NO	NO	NO	
FABRICANTE_ART	c_articulo	NO	NO	NO	
NANDINA_ART	c_articulo	NO	NO	NO	
NOMBRE_ART	c_articulo	NO	NO	SI	
NUMER_TARIF_ART	c_articulo	NO	NO	NO	
PADRE_ART	c_articulo	NO	NO	NO	
PERMIT_ADIC_ART	c_articulo	NO	NO	NO	
PRECIO_ART	c_articulo	NO	NO	NO	
PRECIO_OFER_ART	c_articulo	NO	NO	NO	
PRESUPUESTO_ART	c_articulo	NO	NO	SI	
STOCK_MAXIM_ART	c_articulo	NO	NO	NO	
STOCK_MINIM_ART	c_articulo	NO	NO	NO	
USA_INVENTA_ART	c_articulo	NO	NO	NO	
VENTA_WEB_ART	c_articulo	NO	NO	NO	
CODIGO_CAC	c_caracteristic	SI	NO	SI	Tabla que sirve para asignar características de cualquier índole a un artículo.
CODIGO_COM	c_caracteristic	NO	SI	NO	
CODIGO_SUC	c_caracteristic	NO	SI	NO	
MASCARA_CAC	c_caracteristic	NO	NO	NO	
NOMBRE_CAC	c_caracteristic	NO	NO	SI	
CODIGO_COM	c_estado_articu	NO	SI	NO	Tabla donde se ponen los estados del artículo.
CODIGO_EAR	c_estado_articu	SI	NO	SI	
CODIGO_SUC	c_estado_articu	NO	SI	NO	
NOMBRE_EAR	c_estado_articu	NO	NO	SI	Tabla que guarda las marcas con las que se trabaja.
CODIGO_COM	c_marca	NO	SI	NO	
CODIGO_MAR	c_marca	SI	NO	SI	
CODIGO_SUC	c_marca	NO	SI	NO	
NOMBRE_MAR	c_marca	NO	NO	SI	
CODIGO_COM	c_nivel_articul	NO	SI	NO	Como la tabla de artículos es recursiva se
CODIGO_NAR	c_nivel_articul	SI	NO	SI	

CODIGO_SUC	c_nivel_articul	NO	SI	NO	necesita una tabla que registre los niveles.
NOMBRE_NAR	c_nivel_articul	NO	NO	SI	
CODIGO_COM	c_unidad	NO	SI	NO	Tabla que registra las unidades con los que se trabaja.
CODIGO_SUC	c_unidad	NO	SI	NO	
CODIGO_UNI	c_unidad	SI	NO	SI	
NOMBRE_UNI	c_unidad	NO	NO	SI	
ALTERNO_LGE	g_lugar_geograf	NO	NO	NO	
ALTERNO1_LGE	g_lugar_geograf	NO	NO	NO	Tabla recursiva donde se ingresan los lugares geográficos con el nivel de precisión que se desee.
CODIGO_COM	g_lugar_geograf	NO	SI	NO	
CODIGO_LGE	g_lugar_geograf	SI	NO	SI	
CODIGO_SUC	g_lugar_geograf	NO	SI	NO	
CODIGO_TLG	g_lugar_geograf	NO	SI	NO	
NOMBRE_LGE	g_lugar_geograf	NO	NO	SI	
PADRE_LGE	g_lugar_geograf	NO	NO	NO	
CODIGO_COM	g_tip_luga_geog	NO	SI	NO	
CODIGO_SUC	g_tip_luga_geog	NO	SI	NO	
CODIGO_TLG	g_tip_luga_geog	SI	NO	SI	
NOMBRE_TLG	g_tip_luga_geog	NO	NO	SI	Tabla donde se poner el tipo de lugar geográfico.
ALTERNO1_TID	g_tipo_identifi	NO	NO	NO	Los tipos de identificación que se utilizarán en todo el sistema.
ALTERNO2_TID	g_tipo_identifi	NO	NO	NO	
ALTERNO3_TID	g_tipo_identifi	NO	NO	NO	
CODIGO_COM	g_tipo_identifi	NO	SI	NO	
CODIGO_SUC	g_tipo_identifi	NO	SI	NO	
CODIGO_TID	g_tipo_identifi	SI	NO	SI	
NOMBRE_TID	g_tipo_identifi	NO	NO	SI	
CANTIDAD_ACO	j_adicio_cotiza	NO	NO	SI	
CODIGO_ACO	j_adicio_cotiza	SI	NO	SI	
CODIGO_ADA	j_adicio_cotiza	NO	SI	NO	
CODIGO_CCZ	j_adicio_cotiza	NO	SI	NO	
CODIGO_COM	j_adicio_cotiza	NO	SI	NO	
CODIGO_SUC	j_adicio_cotiza	NO	SI	NO	
VALOR_ACO	j_adicio_cotiza	NO	NO	SI	
CANTIDAD_AFA	j_adicio_factur	NO	NO	SI	Tabla que registra los adicionales en una factura.
CODIGO_ADA	j_adicio_factur	NO	SI	NO	
CODIGO_AFA	j_adicio_factur	SI	NO	SI	
CODIGO_CFC	j_adicio_factur	NO	SI	NO	
CODIGO_COM	j_adicio_factur	NO	SI	NO	
CODIGO_SUC	j_adicio_factur	NO	SI	NO	
VALOR_AFA	j_adicio_factur	NO	NO	SI	
CANTIDAD_APE	j_adicio_pedido	NO	NO	SI	Tabla que registra los adicionales en un pedido.
CODIGO_ADA	j_adicio_pedido	NO	SI	NO	
CODIGO_APE	j_adicio_pedido	SI	NO	SI	
CODIGO_COM	j_adicio_pedido	NO	SI	NO	
CODIGO_CPD	j_adicio_pedido	NO	SI	NO	
CODIGO_SUC	j_adicio_pedido	NO	SI	NO	
VALOR_APE	j_adicio_pedido	NO	NO	SI	
ALTERNO_ADA	j_adicional	NO	NO	NO	Es donde se registra todos los tipos de adicionales que se utilizarán.
CODIGO_ADA	j_adicional	SI	NO	SI	
CODIGO_COM	j_adicional	NO	SI	NO	
CODIGO_ICU	j_adicional	NO	SI	SI	
CODIGO_SUC	j_adicional	NO	SI	NO	
NOMBRE_ADA	j_adicional	NO	NO	SI	

PARAMETRO_ADA	j_adicional	NO	NO	NO	Tabla que registra las agencias de distribución para el punto de venta.
SIGNO_ADA	j_adicional	NO	NO	SI	
SUMA_ANTES_ADA	j_adicional	NO	NO	SI	
CODIGO_ADS	j_agenci_distri	SI	NO	SI	
CODIGO_COM	j_agenci_distri	NO	SI	NO	
CODIGO_SUC	j_agenci_distri	NO	SI	NO	
COTACTO_ADS	j_agenci_distri	NO	NO	NO	
DIRECCION_ADS	j_agenci_distri	NO	NO	SI	
FAX_ADS	j_agenci_distri	NO	NO	NO	
MAIL_ADS	j_agenci_distri	NO	NO	NO	
NOMBRE_ADS	j_agenci_distri	NO	NO	SI	
TELEFONO_ADS	j_agenci_distri	NO	NO	NO	
TELEFONO1_ADS	j_agenci_distri	NO	NO	NO	
WEB_ADS	j_agenci_distri	NO	NO	NO	
CODIGO_ACL	j_agrupa_client	SI	NO	SI	Tabla que crea grupos de clientes.
CODIGO_COM	j_agrupa_client	NO	SI	NO	
CODIGO_SUC	j_agrupa_client	NO	SI	NO	
NOMBRE_ACL	j_agrupa_client	NO	NO	SI	
OBSERVACION_ACL	j_agrupa_client	NO	NO	NO	
CODIGO_BOE	j_bodegaje	SI	NO	SI	Tabla que registra las bodegas utilizadas por el punto de venta.
CODIGO_COM	j_bodegaje	NO	SI	NO	
CODIGO_SUC	j_bodegaje	NO	SI	NO	
DIRECCION_BOE	j_bodegaje	NO	NO	NO	
FAX_BOE	j_bodegaje	NO	NO	NO	
NOMBRE_BOE	j_bodegaje	NO	NO	SI	
TELEFONO1_BOE	j_bodegaje	NO	NO	NO	
TELEFONO2_BOE	j_bodegaje	NO	NO	NO	
CODIGO_BOE	j_cabece_cotiza	NO	SI	NO	Tabla cabecera en el momento de realizar una cotización.
CODIGO_CCZ	j_cabece_cotiza	SI	NO	SI	
CODIGO_CLE	j_cabece_cotiza	NO	SI	NO	
CODIGO_COM	j_cabece_cotiza	NO	SI	NO	
CODIGO_ECT	j_cabece_cotiza	NO	SI	NO	
CODIGO_LGE	j_cabece_cotiza	NO	SI	NO	
CODIGO_SCL	j_cabece_cotiza	NO	SI	NO	
CODIGO_SUC	j_cabece_cotiza	NO	SI	NO	
CODIGO_TRP	j_cabece_cotiza	NO	SI	NO	
CODIGO_TVN	j_cabece_cotiza	NO	SI	NO	
CODIGO_USU	j_cabece_cotiza	NO	SI	NO	
CODIGO_VED	j_cabece_cotiza	NO	SI	NO	
FECHA_CCZ	j_cabece_cotiza	NO	NO	SI	
OBSERVACION_CCZ	j_cabece_cotiza	NO	NO	NO	
VALIDEZ_CCZ	j_cabece_cotiza	NO	NO	SI	
AWB_CFC	j_cabece_factur	NO	NO	NO	Tabla cabecera en el momento de realizar una facturación.
AWB1_CFC	j_cabece_factur	NO	NO	NO	
CODIGO_ADS	j_cabece_factur	NO	SI	NO	
CODIGO_BOE	j_cabece_factur	NO	SI	NO	
CODIGO_CAJ	j_cabece_factur	NO	SI	NO	
CODIGO_CCZ	j_cabece_factur	NO	SI	NO	
CODIGO_CFC	j_cabece_factur	SI	NO	SI	
CODIGO_CFE	j_cabece_factur	NO	SI	NO	
CODIGO_CLE	j_cabece_factur	NO	SI	NO	
CODIGO_COM	j_cabece_factur	NO	SI	NO	

CODIGO_CPD	j_cabece_factur	NO	SI	NO	
CODIGO_EFA	j_cabece_factur	NO	SI	NO	
CODIGO_LGE	j_cabece_factur	NO	SI	NO	
CODIGO_SCL	j_cabece_factur	NO	SI	NO	
CODIGO_SUC	j_cabece_factur	NO	SI	NO	
CODIGO_TRP	j_cabece_factur	NO	SI	NO	
CODIGO_TVN	j_cabece_factur	NO	SI	NO	
CODIGO_USU	j_cabece_factur	NO	SI	NO	
CODIGO_VED	j_cabece_factur	NO	SI	NO	
DIRECCION_CFC	j_cabece_factur	NO	NO	NO	
FECHA_ELABO_CFC	j_cabece_factur	NO	NO	SI	
FECHA_EMBAR_CFC	j_cabece_factur	NO	NO	SI	
FECHA_ENTRE_CFC	j_cabece_factur	NO	NO	SI	
FUE_CFC	j_cabece_factur	NO	NO	NO	
NUMERO_CFC	j_cabece_factur	NO	NO	NO	
OBSERVACION_CFC	j_cabece_factur	NO	NO	NO	
CODIGO_ADS	j_cabece_pedido	NO	SI	NO	Tabla cabecera en el momento de realizar un pedido.
CODIGO_BOE	j_cabece_pedido	NO	SI	NO	
CODIGO_CCZ	j_cabece_pedido	NO	SI	NO	
CODIGO_CLE	j_cabece_pedido	NO	SI	NO	
CODIGO_COM	j_cabece_pedido	NO	SI	NO	
CODIGO_CPD	j_cabece_pedido	SI	NO	SI	
CODIGO_EPE	j_cabece_pedido	NO	SI	NO	
CODIGO_LGE	j_cabece_pedido	NO	SI	NO	
CODIGO_SCL	j_cabece_pedido	NO	SI	NO	
CODIGO_SUC	j_cabece_pedido	NO	SI	NO	
CODIGO_TRP	j_cabece_pedido	NO	SI	NO	
CODIGO_TVN	j_cabece_pedido	NO	SI	NO	
CODIGO_USU	j_cabece_pedido	NO	SI	NO	
CODIGO_VED	j_cabece_pedido	NO	SI	NO	
DIRECCION_CPD	j_cabece_pedido	NO	NO	NO	
FECHA_ELABO_CPD	j_cabece_pedido	NO	NO	SI	
FECHA_EMBAR_CPD	j_cabece_pedido	NO	NO	NO	
FECHA_ENTRE_CPD	j_cabece_pedido	NO	NO	SI	
FUE_CPD	j_cabece_pedido	NO	NO	NO	
OBSERVACION_CPD	j_cabece_pedido	NO	NO	NO	
CODIGO_CAJ	j_caja	SI	NO	SI	Tabla que registra las cajas que funcionan en cada sucursal.
CODIGO_COM	j_caja	NO	SI	NO	
CODIGO_SUC	j_caja	NO	SI	NO	
NOMBRE_CAJ	j_caja	NO	NO	NO	
CLAVE_CLE	j_cliente	NO	NO	NO	Tabla recursiva que registra los datos de los clientes.
CODIGO_ACL	j_cliente	NO	SI	NO	
CODIGO_CLE	j_cliente	SI	NO	SI	
CODIGO_COM	j_cliente	NO	SI	NO	
CODIGO_ECL	j_cliente	NO	SI	NO	
CODIGO_LGE	j_cliente	NO	SI	NO	
CODIGO_SUC	j_cliente	NO	SI	NO	
CODIGO_TCL	j_cliente	NO	SI	NO	
CODIGO_TID	j_cliente	NO	SI	NO	
CODIGO_VEC	j_cliente	NO	SI	NO	
CODIGO_VED	j_cliente	NO	SI	NO	
DIRECCION_CLE	j_cliente	NO	NO	NO	

EMAIL_CLE	j_cliente	NO	NO	NO		
FAX_CLE	j_cliente	NO	NO	NO		
FECHA_INGRE_CLE	j_cliente	NO	NO	SI		
FECHA_NACIM_CLE	j_cliente	NO	NO	NO		
IDENTIFICA_CLE	j_cliente	NO	NO	NO		
MOVIL_CLE	j_cliente	NO	NO	NO		
NOMBRE_CLE	j_cliente	NO	NO	SI		
PADRE_CLE	j_cliente	NO	NO	SI		
PADRE2_CLE	j_cliente	NO	NO	NO		
REPRESENTAN_CLE	j_cliente	NO	NO	NO		
RUC_CLE	j_cliente	NO	NO	NO		
TELEFONO1_CLE	j_cliente	NO	NO	SI		
TELEFONO2_CLE	j_cliente	NO	NO	NO		
WEB_CLE	j_cliente	NO	NO	NO		
CODIGO_ART	j_cupo_vendedor	NO	SI	NO	Tabla que registra el cupo de venta para cada vendedor.	
CODIGO_COM	j_cupo_vendedor	NO	SI	NO		
CODIGO_CVE	j_cupo_vendedor	SI	NO	SI		
CODIGO_PEO	j_cupo_vendedor	NO	SI	NO		
CODIGO_SUC	j_cupo_vendedor	NO	SI	NO		
CODIGO_TCP	j_cupo_vendedor	NO	SI	NO		
CODIGO_VED	j_cupo_vendedor	NO	SI	NO		
VALOR_CVE	j_cupo_vendedor	NO	NO	SI		
VALOR1_CVE	j_cupo_vendedor	NO	NO	SI		
VALOR2_CVE	j_cupo_vendedor	NO	NO	SI		
CODIGO_ADA	j_descuento	NO	SI	NO		Tabla donde se pondrán las reglas para los descuentos.
CODIGO_ART	j_descuento	NO	SI	NO		
CODIGO_CLE	j_descuento	NO	SI	NO		
CODIGO_COM	j_descuento	NO	SI	NO		
CODIGO_DEU	j_descuento	SI	NO	SI		
CODIGO_FPO	j_descuento	NO	SI	NO		
CODIGO_LGE	j_descuento	NO	SI	NO		
CODIGO_MVE	j_descuento	NO	SI	NO		
CODIGO_RPL	j_descuento	NO	SI	NO		
CODIGO_SUC	j_descuento	NO	SI	NO		
CODIGO_TEM	j_descuento	NO	SI	NO		
CODIGO_VVE	j_descuento	NO	SI	NO		
COLUMNA_DEU	j_descuento	NO	NO	NO		
DESCUENTO_DEU	j_descuento	NO	NO	SI		
ORDEN_DEU	j_descuento	NO	NO	NO		
RECARGO_DEU	j_descuento	NO	NO	NO		
CANTIDAD_DPT	j_det_pago_coti	NO	NO	SI	Tabla que registra el detalle del pago al realizar una cotización.	
CODIGO_COM	j_det_pago_coti	NO	SI	NO		
CODIGO_DPT	j_det_pago_coti	SI	NO	SI		
CODIGO_EPA	j_det_pago_coti	NO	SI	NO		
CODIGO_PCT	j_det_pago_coti	NO	SI	NO		
CODIGO_SUC	j_det_pago_coti	NO	SI	NO		
DIAS_DPT	j_det_pago_coti	NO	NO	SI		
DOCUMENTO_DPT	j_det_pago_coti	NO	NO	NO		
FECHA_DPT	j_det_pago_coti	NO	NO	SI		
VALOR_DPT	j_det_pago_coti	NO	NO	SI		
CANTIDAD_DPF	j_det_pago_fact	NO	NO	SI	Tabla que registra el detalle del pago al	
CODIGO_COM	j_det_pago_fact	NO	SI	NO		

CODIGO_DPF	j_det_pago_fact	SI	NO	SI	realizar una facturación.
CODIGO_EPA	j_det_pago_fact	NO	SI	NO	
CODIGO_PFC	j_det_pago_fact	NO	SI	NO	
CODIGO_SUC	j_det_pago_fact	NO	SI	NO	
DIAS_DPF	j_det_pago_fact	NO	NO	SI	
DOCUMENTO_DPF	j_det_pago_fact	NO	NO	NO	
FECHA_DPF	j_det_pago_fact	NO	NO	SI	
VALOR_DPF	j_det_pago_fact	NO	NO	SI	
CANTIDAD_DPP	j_det_pago_pedi	NO	NO	SI	
CODIGO_COM	j_det_pago_pedi	NO	SI	NO	
CODIGO_DPP	j_det_pago_pedi	SI	NO	SI	
CODIGO_EPA	j_det_pago_pedi	NO	SI	NO	
CODIGO_PPE	j_det_pago_pedi	NO	SI	NO	
CODIGO_SUC	j_det_pago_pedi	NO	SI	NO	
DIAS_DPP	j_det_pago_pedi	NO	NO	SI	
DOCUMENTO_DPP	j_det_pago_pedi	NO	NO	NO	
FECHA_DPP	j_det_pago_pedi	NO	NO	SI	
VALOR_DPP	j_det_pago_pedi	NO	NO	SI	
ACUMU_MONTO_DCZ	j_detall_cotiza	NO	NO	NO	Tabla que registra los artículos, precios y descuentos en una cotización.
CANTIDAD_DCZ	j_detall_cotiza	NO	NO	SI	
CARGA_IVA_DCZ	j_detall_cotiza	NO	NO	NO	
CODIGO_ART	j_detall_cotiza	NO	SI	NO	
CODIGO_CCZ	j_detall_cotiza	NO	SI	NO	
CODIGO_COM	j_detall_cotiza	NO	SI	NO	
CODIGO_DCZ	j_detall_cotiza	SI	NO	SI	
CODIGO_PRE	j_detall_cotiza	NO	SI	NO	
CODIGO_SUC	j_detall_cotiza	NO	SI	NO	
DESCUENTO_DCZ	j_detall_cotiza	NO	NO	NO	
OBSERVACION_DCZ	j_detall_cotiza	NO	NO	NO	
PERMIT_ADIC_DCZ	j_detall_cotiza	NO	NO	NO	
PRECIO_DCZ	j_detall_cotiza	NO	NO	SI	
PRECIO2_DCZ	j_detall_cotiza	NO	NO	SI	
PRECIO3_DCZ	j_detall_cotiza	NO	NO	NO	
ACUMU_MONTO_DFC	j_detall_factur	NO	NO	NO	
CANTIDAD_DFC	j_detall_factur	NO	NO	SI	
CARGA_IVA_DFC	j_detall_factur	NO	NO	NO	
CODIGO_ART	j_detall_factur	NO	SI	NO	
CODIGO_CFC	j_detall_factur	NO	SI	NO	
CODIGO_COM	j_detall_factur	NO	SI	NO	
CODIGO_DFC	j_detall_factur	SI	NO	SI	
CODIGO_PRE	j_detall_factur	NO	SI	NO	
CODIGO_SUC	j_detall_factur	NO	SI	NO	
DESCUENTO_DFC	j_detall_factur	NO	NO	NO	
OBSERVACION_DFC	j_detall_factur	NO	NO	NO	
PERMIT_ADIC_DFC	j_detall_factur	NO	NO	NO	
PRECIO1_DFC	j_detall_factur	NO	NO	SI	
PRECIO2_DFC	j_detall_factur	NO	NO	SI	
PRECIO3_DFC	j_detall_factur	NO	NO	NO	
ACUMU_MONTO_DPD	j_detall_pedido	NO	NO	NO	Tabla que registra los artículos, precios y descuentos en un pedido.
CANTIDAD_DPD	j_detall_pedido	NO	NO	SI	
CARGA_IVA_DPD	j_detall_pedido	NO	NO	NO	
CODIGO_ART	j_detall_pedido	NO	SI	NO	

CODIGO_COM	j_detall_pedido	NO	SI	NO		
CODIGO_CPD	j_detall_pedido	NO	SI	NO		
CODIGO_DPD	j_detall_pedido	SI	NO	SI		
CODIGO_PRE	j_detall_pedido	NO	SI	NO		
CODIGO_SUC	j_detall_pedido	NO	SI	NO		
DESCUENTO_DPD	j_detall_pedido	NO	NO	NO		
OBSERVACION_DPD	j_detall_pedido	NO	NO	NO		
PERMIT_ADIC_DPD	j_detall_pedido	NO	NO	NO		
PRECIO_DPD	j_detall_pedido	NO	NO	SI		
PRECIO2_DPD	j_detall_pedido	NO	NO	SI		
PRECIO3_DPD	j_detall_pedido	NO	NO	NO		
AUTORIZA_EPA	j_entidad_pago	NO	NO	NO		Registro de las entidades de pago que se van a utilizar en la venta.
CODIGO_CLE	j_entidad_pago	NO	SI	NO		
CODIGO_COM	j_entidad_pago	NO	SI	NO		
CODIGO_EPA	j_entidad_pago	SI	NO	SI		
CODIGO_SUC	j_entidad_pago	NO	SI	NO		
ESTARJETA_EPA	j_entidad_pago	NO	NO	NO		
LOGO_EPA	j_entidad_pago	NO	NO	NO		
NOMBRE_EPA	j_entidad_pago	NO	NO	NO		
USOLOCAL_EPA	j_entidad_pago	NO	NO	NO		
USOWEB_EPA	j_entidad_pago	NO	NO	NO		
CODIGO_COM	j_estado_client	NO	SI	NO	Tabla que registra el estado del cliente.	
CODIGO_ECL	j_estado_client	SI	NO	SI		
CODIGO_SUC	j_estado_client	NO	SI	NO		
NOMBRE_ECL	j_estado_client	NO	NO	SI		
CODIGO_COM	j_estado_cotiza	NO	SI	NO	Tabla que registra el estado de la cotización.	
CODIGO_ECT	j_estado_cotiza	SI	NO	SI		
CODIGO_SUC	j_estado_cotiza	NO	SI	NO		
NOMBRE_ECT	j_estado_cotiza	NO	NO	SI		
CODIGO_COM	j_estado_factur	NO	SI	NO	Tabla que registra el estado de la facturación.	
CODIGO_EFA	j_estado_factur	SI	NO	SI		
CODIGO_SUC	j_estado_factur	NO	SI	NO		
NOMBRE_EFA	j_estado_factur	NO	NO	SI		
CODIGO_COM	j_estado_pedido	NO	SI	NO	Tabla que registra el estado del pedido.	
CODIGO_EPE	j_estado_pedido	SI	NO	SI		
CODIGO_SUC	j_estado_pedido	NO	SI	NO		
NOMBRE_EPE	j_estado_pedido	NO	NO	SI		
CODIGO_COM	j_for_pag_tarje	NO	SI	NO	Tabla donde se guardan las formas posibles para pagos con tarjeta.	
CODIGO_EPA	j_for_pag_tarje	NO	SI	NO		
CODIGO_FPT	j_for_pag_tarje	SI	NO	SI		
CODIGO_SUC	j_for_pag_tarje	NO	SI	NO		
MESES_FPT	j_for_pag_tarje	NO	NO	NO		
NOMBRE_FPT	j_for_pag_tarje	NO	NO	SI		
POCENTAJE_FPT	j_for_pag_tarje	NO	NO	NO		
CODIGO_COM	j_forma_pago	NO	SI	NO	Registro de las formas de pago que se pueden presentar.	
CODIGO_FPO	j_forma_pago	SI	NO	SI		
CODIGO_SUC	j_forma_pago	NO	SI	NO		
CODIGO_TDE	j_forma_pago	NO	SI	NO		
CODIGO_TPG	j_forma_pago	NO	SI	NO		
NOMBRE_FPO	j_forma_pago	NO	NO	SI		
PRESEN_COMP_FPO	j_forma_pago	NO	NO	NO		
CODIGO_COM	j_monto_venta	NO	SI	NO	Rangos de valores	

CODIGO_MVE	j_monto_venta	SI	NO	SI	monetarios que se utilizarán en las reglas de negocio.
CODIGO_SUC	j_monto_venta	NO	SI	NO	
NOMBRE_MVE	j_monto_venta	NO	NO	SI	
VALOR_FINAL_MVE	j_monto_venta	NO	NO	SI	
VALOR_INICI_MVE	j_monto_venta	NO	NO	SI	
CODIGO_CCZ	j_pago_cotiza	NO	SI	NO	Tabla que registra el pago que se realizo en una cotización.
CODIGO_COM	j_pago_cotiza	NO	SI	NO	
CODIGO_EPA	j_pago_cotiza	NO	SI	NO	
CODIGO_FPO	j_pago_cotiza	NO	SI	NO	
CODIGO_PCT	j_pago_cotiza	SI	NO	SI	
CODIGO_SUC	j_pago_cotiza	NO	SI	NO	
DOCUMENTO1_PCT	j_pago_cotiza	NO	NO	NO	
DOCUMENTO2_PCT	j_pago_cotiza	NO	NO	NO	
VALOR_PCT	j_pago_cotiza	NO	NO	SI	
CODIGO_CFC	j_pago_factura	NO	SI	NO	
CODIGO_COM	j_pago_factura	NO	SI	NO	
CODIGO_EPA	j_pago_factura	NO	SI	NO	
CODIGO_FPO	j_pago_factura	NO	SI	NO	
CODIGO_PFC	j_pago_factura	SI	NO	SI	
CODIGO_SUC	j_pago_factura	NO	SI	NO	
DOCUMENTO1_PFC	j_pago_factura	NO	NO	NO	
DOCUMENTO2_PFC	j_pago_factura	NO	NO	NO	
VALOR_PFC	j_pago_factura	NO	NO	SI	
CODIGO_COM	j_pago_pedido	NO	SI	NO	Tabla que registra el pago que se realizo en un pedido.
CODIGO_CPD	j_pago_pedido	NO	SI	NO	
CODIGO_EPA	j_pago_pedido	NO	SI	NO	
CODIGO_FPO	j_pago_pedido	NO	SI	NO	
CODIGO_PPE	j_pago_pedido	SI	NO	SI	
CODIGO_SUC	j_pago_pedido	NO	SI	NO	
DOCUMENTO1_PPE	j_pago_pedido	NO	NO	NO	
DOCUMENTO2_PPE	j_pago_pedido	NO	NO	NO	
VALOR_PPE	j_pago_pedido	NO	NO	SI	
CODIGO_COM	j_periodo	NO	SI	NO	
CODIGO_PEO	j_periodo	SI	NO	SI	
CODIGO_SUC	j_periodo	NO	SI	NO	
FECHA_FINAL_PEO	j_periodo	NO	NO	SI	
FECHA_INICI_PEO	j_periodo	NO	NO	SI	
NOMBRE_PEO	j_periodo	NO	NO	SI	
NUMERO_PEO	j_periodo	NO	NO	SI	
CODIGO_ART	j_precio	NO	SI	NO	Tabla que registra las reglas de negocio para los precios.
CODIGO_CLE	j_precio	NO	SI	NO	
CODIGO_COM	j_precio	NO	SI	NO	
CODIGO_LGE	j_precio	NO	SI	NO	
CODIGO_MVE	j_precio	NO	SI	NO	
CODIGO_PRE	j_precio	SI	NO	SI	
CODIGO_SUC	j_precio	NO	SI	NO	
CODIGO_TEM	j_precio	NO	SI	NO	
CODIGO_VVE	j_precio	NO	SI	NO	
ORDEN_PRE	j_precio	NO	NO	NO	
PRECIO_PRE	j_precio	NO	NO	SI	
CODIGO_COM	j_rango_plazo	NO	SI	NO	Rangos de días plazo que se utilizarán en las
CODIGO_RPL	j_rango_plazo	SI	NO	SI	

CODIGO_SUC	j_rango_plazo	NO	SI	NO	reglas de negocio.	
DESDE_RPL	j_rango_plazo	NO	NO	NO		
HASTA_RPL	j_rango_plazo	NO	NO	NO		
NOMBRE_RPL	j_rango_plazo	NO	NO	NO	Tabla que registra los servicios existentes para tarjetas de crédito.	
CODIGO_COM	j_servic_tarjet	NO	SI	NO		
CODIGO_STA	j_servic_tarjet	SI	NO	SI		
CODIGO_SUC	j_servic_tarjet	NO	SI	NO	Tabla que relaciona un vendedor con la sucursal donde trabaja.	
NOMBRE_STA	j_servic_tarjet	NO	NO	SI		
CODIGO_CLE	j_sucurs_client	NO	SI	NO		
CODIGO_COM	j_sucurs_client	NO	SI	NO		
CODIGO_ECL	j_sucurs_client	NO	SI	NO		
CODIGO_LGE	j_sucurs_client	NO	SI	NO		
CODIGO_SCL	j_sucurs_client	SI	NO	SI		
CODIGO_SUC	j_sucurs_client	NO	SI	NO		
CODIGO_TID	j_sucurs_client	NO	SI	NO		
CODIGO_VED	j_sucurs_client	NO	SI	NO		
DIRECCION_SCL	j_sucurs_client	NO	NO	NO		
EMAIL_SCL	j_sucurs_client	NO	NO	NO		
FAX_SCL	j_sucurs_client	NO	NO	NO		
FECHA_INGRE_SCL	j_sucurs_client	NO	NO	NO		
FECHA_NACIM_SCL	j_sucurs_client	NO	NO	NO		
IDENTIFICA_SCL	j_sucurs_client	NO	NO	NO		
MOVIL_SCL	j_sucurs_client	NO	NO	NO		
NOMBRE_SCL	j_sucurs_client	NO	NO	SI		
REPRESENTAN_SCL	j_sucurs_client	NO	NO	NO		
RUC_SCL	j_sucurs_client	NO	NO	NO		
TELEFONO1_SCL	j_sucurs_client	NO	NO	NO		
TELEFONO2_SCL	j_sucurs_client	NO	NO	NO		
WEB_SCL	j_sucurs_client	NO	NO	NO		
CODIGO_COM	j_temporada	NO	SI	NO		Rangos de fechas que se utilizarán en las reglas de negocio.
CODIGO_SUC	j_temporada	NO	SI	NO		
CODIGO_TEM	j_temporada	SI	NO	SI		
FECHA_FINAL_TEM	j_temporada	NO	NO	SI		
FECHA_INICI_TEM	j_temporada	NO	NO	SI		
NOMBRE_TEM	j_temporada	NO	NO	SI		
CODIGO_COM	j_tipo_cliente	NO	SI	NO	Tabla que tipifica a los clientes.	
CODIGO_DEU	j_tipo_cliente	NO	SI	NO		
CODIGO_SUC	j_tipo_cliente	NO	SI	NO		
CODIGO_TCL	j_tipo_cliente	SI	NO	SI		
NOMBRE_TCL	j_tipo_cliente	NO	NO	SI		
CODIGO_COM	j_tipo_pago	NO	SI	NO	Tabla que registra el tipo de pago por cada forma de pago.	
CODIGO_SUC	j_tipo_pago	NO	SI	NO		
CODIGO_TPG	j_tipo_pago	SI	NO	SI		
NOMBRE_TPG	j_tipo_pago	NO	NO	SI		
ALTERNO_TRP	j_transporte	NO	NO	NO	Tabla que registra las entidades que prestan el servicio de transporte para la empresa.	
CODIGO_COM	j_transporte	NO	SI	NO		
CODIGO_SUC	j_transporte	NO	SI	NO		
CODIGO_TRP	j_transporte	SI	NO	SI		
DIRECCION_TRP	j_transporte	NO	NO	NO		
EMAIL_TRP	j_transporte	NO	NO	NO		
FAX_TRP	j_transporte	NO	NO	NO		
NOMBRE_TRP	j_transporte	NO	NO	SI		

PAGINA_WEB_TRP	j_transporte	NO	NO	NO	Tabla que registra las reglas para inhabilidad a un cliente.
TELEFONO1_TRP	j_transporte	NO	NO	NO	
TELEFONO2_TRP	j_transporte	NO	NO	NO	
CODIGO_ACL	j_vencimiento	NO	SI	NO	
CODIGO_COM	j_vencimiento	NO	SI	NO	
CODIGO_SUC	j_vencimiento	NO	SI	NO	
CODIGO_TCL	j_vencimiento	NO	SI	NO	
CODIGO_VEC	j_vencimiento	SI	NO	SI	
DIA_VEC	j_vencimiento	NO	NO	SI	
CODIGO_COM	j_vendedor	NO	SI	NO	Tabla que registra los datos de los vendedores.
CODIGO_LGE	j_vendedor	NO	SI	NO	
CODIGO_SUC	j_vendedor	NO	SI	NO	
CODIGO_USU	j_vendedor	NO	SI	NO	
CODIGO_VED	j_vendedor	SI	NO	SI	
DIRECCION_VED	j_vendedor	NO	NO	SI	
FAX_VED	j_vendedor	NO	NO	NO	
MOVIL_VED	j_vendedor	NO	NO	NO	
NOMBRE_VED	j_vendedor	NO	NO	SI	
PADRE_VED	j_vendedor	NO	NO	NO	
TELEFONO1_VED	j_vendedor	NO	NO	SI	
TELEFONO2_VED	j_vendedor	NO	NO	NO	
CODIGO_COM	j_volumen_venta	NO	SI	NO	Rangos de valores cuantitativos que se utilizarán en las reglas de negocio.
CODIGO_SUC	j_volumen_venta	NO	SI	NO	
CODIGO_VVE	j_volumen_venta	SI	NO	SI	
DESDE_VVE	j_volumen_venta	NO	NO	SI	
HASTA_VVE	j_volumen_venta	NO	NO	SI	
NIVEL_VVE	j_volumen_venta	NO	NO	SI	
NOMBRE_VVE	j_volumen_venta	NO	NO	SI	
ABREVIACION_COM	s_compania	NO	NO	SI	Tabla que registra los datos de la compañía.
ALIAS_COM	s_compania	NO	NO	NO	
COD_TIP_EMP_COM	s_compania	NO	NO	NO	
CODI_ACTIVI_COM	s_compania	NO	NO	NO	
CODIGO_COM	s_compania	SI	NO	SI	
CODIGO_CULTIVO_COM	s_compania	NO	NO	NO	
CODIGO_LGE	s_compania	NO	SI	NO	
DIRECCION_COM	s_compania	NO	NO	SI	
DIRECCION1_COM	s_compania	NO	NO	NO	
DOCUM_REPRE_COM	s_compania	NO	NO	NO	
EMAIL_COM	s_compania	NO	NO	NO	
FAX_COM	s_compania	NO	NO	NO	
LOGO_COM	s_compania	NO	NO	NO	
NOMBRE_COM	s_compania	NO	NO	SI	
NUMER_PATRO_COM	s_compania	NO	NO	SI	
PAGINA_WEB_COM	s_compania	NO	NO	NO	
REPRESENTAN_COM	s_compania	NO	NO	SI	
RUC_COM	s_compania	NO	NO	SI	
TELEFONO1_COM	s_compania	NO	NO	SI	
TELEFONO2_COM	s_compania	NO	NO	NO	
ARCHI_REPLI_SUC	s_sucursal	NO	NO	NO	Tabla que registra los datos de la sucursal.
CODIGO_COM	s_sucursal	NO	SI	NO	
CODIGO_LGE	s_sucursal	NO	SI	NO	
CODIGO_SUC	s_sucursal	SI	NO	SI	

DIRECCION_SUC	s_sucursal	NO	NO	NO	
DOCUM_REPRE_SUC	s_sucursal	NO	NO	NO	
EMAIL_SUC	s_sucursal	NO	NO	NO	
FAX_SUC	s_sucursal	NO	NO	NO	
NOMBRE_SUC	s_sucursal	NO	NO	SI	
NUMER_PATRO_SUC	s_sucursal	NO	NO	NO	
REPRESENTAN_SUC	s_sucursal	NO	NO	NO	
TELEFONO_SUC	s_sucursal	NO	NO	NO	
CLAVE_USU	s_usuario	NO	NO	NO	
CODIGO_COM	s_usuario	NO	SI	NO	
CODIGO_USU	s_usuario	SI	NO	SI	Tabla del sistema que registra los usuarios del ERP.
NOMBRE_USU	s_usuario	NO	NO	SI	

ANEXO H

H. GLOSARIO DE SIGLAS

APPLET: Componente de software que corre en el contexto de otro programa
ASN: Autonomous System Number
ASP: Active Server Page
ASPX: Active Server Page XML
CB: Código de Barras
CDM: Conceptual Data Model
CLI: Interface de línea de comando
COM: Component Object Model
CTS: Component Transaction Server
DBMS: Data Base Management System
EJB: Enterprise Java Beans
ERP: Enterprise Resource Planning
FEM: Free Enterprise Model
GPE: General Protection Exception
HOOD: Hierarchical Object Oriented Design
HTML: Hypertext Markup Language
IBM: International Business Machines Corp.
IDE: Entorno de Desarrollo Integrado
IP: Internet Protocol
JAR: Java Archive
JDBC: Java Database Connectivity
JSP: Java Server Pages.
KM: Knowledge manager
MTS: Microsoft Transaction Sever
NCR: Empresa especializada en terminales POS
ODM: Object Data Model
OOA: Object Oriented Analisis
OOD: Object Oriented Design
OOO: Object Oriented
OOP: Object Oriented Programation
OOSA: Object Oriented Systems Analysis
OOSD: Object Oriented Structured Design
PBL: Power Builder Library
PDM: Physical Data Model
POS: Point Of Sale
RDD: Responsibility Driven Design
SAP: System Applications and Products
SCM: Software Configuration Management
SCM: Supply Chain Management
SDK: Software Development Kit
SOAP: Simple Object Access Protocol
SPL: Licencia Pública de SUN
SQL: Structured Query Language
SRM: Standard Reference Material
SSL: Secure Sockets Layer
TPV: Terminal Punto de Venta virtual
WAE: Wireless Application Environment
WAR: Web Archive

WSDL: Web Services Description Language
XHTML: eXtensible Hyper Text Markup Language

ANEXO I

I. DOCUMENTO DE APROBACIÓN DEL AUSPICIANTE

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Sr. Barriga Torres Pablo Andrés

Sr. Barrigas Jácome David Israel

DIRECTOR DEL DEPARTAMENTO CIENCIAS DE LA COMPUTACIÓN

TCRN. DE E. M. ING. MARCO QUINTANA

Sangolquí, 28 de Septiembre del 2006