

Universidad Nacional Autónoma de Nicaragua.

UNAN-LEÓN.

Facultad de Ciencias y Tecnología



**“SISTEMA TERMINAL DE PUNTO DE VENTAS, ORIENTADO A LA CADENA DE
SUPERMERCADOS SALMAN UBICADOS EN LEON – NICARAGUA.”**

Monografía para optar al título de Ingeniero en Sistemas de Información.

Autores:

- ✓ Br. Álvaro José Zapata Téllez.
- ✓ Br. David Leonardo Rivas Vargas.
- ✓ Br. José Francisco Sánchez Cano.

Tutor:

- Msc. Aldo René Martínez Delgadillo.

León, 10 de Noviembre del 2012.

1. INDICE

1.	INDICE	2
2.	AGRADECIMIENTOS	7
3.	DEDICATORIA	8
4.	INTRODUCCIÓN	9
5.	ANTECEDENTES	10
6.	JUSTIFICACIÓN	12
7.	OBJETIVOS	13
	OBJETIVO GENERAL:	13
	OBJETIVOS ESPECÍFICOS:	13
8.	DISEÑO METODOLÓGICO	14
8.1.	MODELO EN CASCADA	14
8.2.	ACTIVIDADES DEL CICLO DE VIDA A EN CASCADA.	14
8.3.	MATERIALES.....	17
9.	MARCO TEÓRICO.....	18
9.1.	SOFTWARE	18
9.1.1.	A medida.....	18
9.1.2.	Comerciales	18
9.1.3.	Específicos.....	19
9.2.	HARDWARE	19
9.2.1.	Compacto.....	19
9.2.2.	Modular.....	20
9.2.3.	Monitor.....	20
9.2.4.	Teclado	20
9.2.5.	Impresora de recibos	20
9.2.6.	Cajón de Dinero	21
9.2.7.	Lector de código de barras.	21
9.2.8.	Pantalla o visor electrónico del TPV.....	22
9.2.9.	Lector de banda magnética.	22
9.3.	LAS NUEVAS TENDENCIAS EN TPV:.....	22
9.4.	HERRAMIENTAS DE DESARROLLO.	23
9.4.1.	Visual Estudio 2008.	23
9.4.2.	Crystal Reports para VS 2008	23
9.4.3.	SQL Server 2005.....	24
10.	ESPECIFICACIÓN DE REQUISITOS SOFTWARE	26
10.1.	INTRODUCCIÓN.....	26
10.1.1.	Propósito.....	26
10.1.2.	Alcance	26
10.2.	DESCRIPCION GENERAL	27
10.2.1.	CARACTERÍSTICAS DEL USUARIO:	27
10.2.2.	REQUISITOS GENERALES:.....	27
10.2.3.	RESTRICCIONES GENERALES	28
10.2.4.	TIPOS DE USUARIOS.....	28

11.	RELACIÓN CON OTRO PROYECTOS	29
12.	REQUISITOS ESPECIFICOS	31
12.1.	INICIO DE SESIÓN.....	31
12.1.1.	Introducción:.....	31
12.1.2.	Entradas:	31
12.1.3.	Procesos:.....	31
12.1.4.	Salida:	31
12.1.5.	Interfaz Hardware	31
12.1.6.	Interfaz Software	31
12.2.	AGREGAR PRODUCTOS A UNA FACTURA:.....	32
12.2.1.	Introducción:.....	32
12.2.2.	Entrada:.....	32
12.2.3.	Procesos:.....	32
12.2.4.	Salida:	32
12.3.	FILTRAR PRODUCTO:	32
12.3.1.	Introducción:.....	32
12.3.2.	Entrada:.....	32
12.3.3.	Proceso:	33
12.3.4.	Salida:	33
12.4.	PESAR PRODUCTO:	33
12.4.1.	Introducción:.....	33
12.4.2.	Proceso:	33
12.4.3.	Salida:	33
12.5.	MODIFICAR LÍNEA DE LA FACTURA:	33
12.5.1.	Introducción:.....	33
12.5.2.	Entrada:.....	33
12.5.3.	Proceso:	34
12.5.4.	Salida:	34
12.6.	ELIMINAR LÍNEA DE LA FACTURA:	34
12.6.1.	Introducción:.....	34
12.6.2.	Proceso:	34
12.6.3.	Interfaz externa:	34
a.	Interfaz de usuario:	34
12.7.	REIMPRIMIR ULTIMA FACTURA:	35
12.7.1.	Introducción:.....	35
12.7.2.	Proceso:	35
12.8.	FILTRAR CLIENTE:	35
12.8.1.	Introducción:.....	35
12.8.2.	Entrada:.....	35
12.8.3.	Proceso:	35
12.8.4.	Salida:	35
12.9.	AGREGAR FORMAS DE PAGO:.....	35
12.9.1.	Introducción:.....	35
12.9.2.	Entrada:.....	35
12.9.3.	Proceso:	36
12.9.4.	Salida:	36

12.10.	CANCELAR FACTURA.	36
12.10.1.	Introducción:.....	36
12.10.2.	Proceso:.....	36
12.10.3.	Salida:.....	36
12.11.	ACCESO AL MENÚ DE SUPERVISOR.	37
12.11.1.	Introducción.....	37
12.11.2.	Entradas.....	37
12.11.3.	Procesos:.....	37
12.11.4.	Salida:.....	37
12.12.	REIMPRIMIR FACTURA.....	37
12.12.1.	Introducción:.....	37
12.12.2.	Entrada:.....	37
12.12.3.	Proceso:.....	38
12.12.4.	Salida:.....	38
12.13.	ANULAR FACTURA.....	38
12.13.1.	Introducción:.....	38
12.13.2.	Entrada:.....	38
12.13.3.	Proceso:.....	38
12.13.4.	Salida:.....	38
12.14.	ESTABLECER DESCUENTO EXTRA.	39
12.14.1.	Introducción:.....	39
12.14.2.	Entrada:.....	39
12.14.3.	Salida:.....	39
12.15.	PRE LECTURA.	39
12.15.1.	Introducción:.....	39
12.15.2.	Proceso:.....	39
12.15.3.	Salida:.....	39
12.16.	CIERRE.....	40
12.16.1.	Introducción:.....	40
12.16.2.	Entrada:.....	40
12.17.	EL SUPERVISOR.....	40
12.17.1.	Proceso:.....	40
12.17.2.	Salida:.....	41
12.18.	ABRIR CAJON DE DINERO.	41
12.18.1.	Introducción:.....	41
12.18.2.	Proceso:.....	41
12.18.3.	Salida:.....	41
13.	DIAGRAMAS DE SECUENCIA.....	47
14.	DIAGRAMA RELACIONAL.....	52
15.	DISEÑO DE INTERFAZ.....	53
16.	CONCLUSIONES.....	63
17.	RECOMENDACIONES.....	64
18.	BIBLIOGRAFÍA.....	65
19.	ANEXOS.....	66
1.	DICCIONARIO DE DATOS.....	66

20.	CÓDIGO FUENTE	70
20.1.	MANEJO DE TECLADO TOUCH	72
20.2.	VALIDAR QUE EL USUARIO SEA DE TIPO CAJERO PARA ABRIR LA CAJA.....	76
20.3.	ABRIR CAJA SI EL TIPO DE USUARIO ES CAJERO	77
20.4.	AGREGAR LÍNEAS A UNA FACTURA	79
20.5.	ELIMINA UNA LÍNEA DE UNA FACTURA.....	80
20.6.	ABRIR CAJA DE DINERO.....	80
20.7.	IMPRIMIR FACTURA	80
20.8.	ANULAR UNA FACTURA	81
20.9.	IMPRIMIR REPORTE DE PRE LECTURA.....	82
20.10.	CERRAR CAJA	84
20.11.	IMPRIMIR REPORTE DE CIERRE DE CAJA	84
21.	PROCEDIMIENTOS ALMACENADOS EN LA BASE DE DATOS	87
21.1.	ABRIR UNA CAJA.....	87
21.2.	CREAR PRE-LECTURA DE UNA CAJA	88
21.3.	FUNCIÓN PARA OBTENER EL PRECIO FINAL DE UN PRODUCTO.	91

INDICE DE FIGURAS

FIGURA 1. GRÁFICAMENTE EL MODELO DE CICLO DE VIDA CLÁSICO SE REPRESENTA DE LA SIGUIENTE MANERA:	16
FIGURA 2. GRÁFICA QUE DETALLA LAS SINCRONIZACIONES.	30
FIGURA 3. DIAGRAMA CASOS DE USOS	42
FIGURA 4. DIAGRAMA DE FLUJO DE DATOS.....	43
FIGURA 5. ROLES DE LOS USUARIOS.	44
FIGURA 6. DIAGRAMA DE CLASES PARTE 1.	45
FIGURA 7. DIAGRAMA DE CLASE PARTE 2.....	46
FIGURA 8. DIAGRAMA DE INICIO DE SESIÓN.....	47
FIGURA 9. DIAGRAMA DE AGREGACIÓN DE PRODUCTOS A UNA FACTURA.	47
FIGURA 10. DIAGRAMA DE FILTRADO DE PRODUCTO.....	48
FIGURA 11. DIAGRAMA DE EDITAR LÍNEA DE LA FACTURA.	48
FIGURA 12. DIAGRAMA DE ELIMINACIÓN LÍNEA DE LA FACTURA.	49
FIGURA 13. DIAGRAMA DE FILTRAR CLIENTE.	49
FIGURA 14. DIAGRAMA DE ANULAR FACTURA.	50
FIGURA 15. DIAGRAMA ESTABLECER DESCUENTO EXTRA.....	50
FIGURA 16. DIAGRAMA DE PRE LECTURA DE UNA FACTURA.	51
FIGURA 17. DIAGRAMA DE CIERRE DE CAJA.....	51
FIGURA 18. DIAGRAMA E-R	52
FIGURA 19. INTERFAZ GRÁFICO DE INICIO DE SESIÓN.....	53
FIGURA 20. INTERFAZ GRÁFICO PANTALLA PRINCIPAL.	54
FIGURA 21. INTERFAZ GRAFICO DE COBRO DE FACTURA	55
FIGURA 22. INTERFAZ GRAFICO DE FILTRO DE PRODUCTOS.....	56
FIGURA 23. INTERFAZ GRAFICO DE FILTRO DE CLIENTE.....	56
FIGURA 24. INTERFAZ GRAFICO DE EDICIÓN DE UNA LÍNEA DE LA FACTURA.	57
FIGURA 25. INTERFAZ GRAFICO DE APLICAR DESCUENTO EXTRA.....	57
FIGURA 26. INTERFAZ GRAFICO DEL MENÚ PRINCIPAL DEL SUPERVISOR.....	58
FIGURA 27. INTERFAZ GRAFICO DETALLANDO EL CIERRE DE LA CAJA.	58
FIGURA 28. INTERFAZ GRAFICA DETALLANDO EL CIERRE DE LA CAJA.....	59
FIGURA 29. REPORTE DONDE SE MUESTRA IMPRESO LA APERTURA DE LA CAJA.....	60
FIGURA 30. REPORTE DONDE SE MUESTRA IMPRESO LA FACTURA DE CRÉDITO.....	60
FIGURA 31. REPORTE DONDE SE MUESTRA IMPRESA LA FACTURA CON PAGO MIXTO.	61
FIGURA 32. REPORTE DONDE SE MUESTRA IMPRESA LA PRE LECTURA DE LA CAJA.	61
FIGURA 33. REPORTE DONDE SE MUESTRA IMPRESO LOS DATOS DE CIERRE DE UNA CAJA.	62

2. AGRADECIMIENTOS

Le agradecemos primeramente a *Dios*; por darnos la vida y por permitirnos llegar a culminar nuestra carrera.

A nuestros padres, que tuvieron que limitarse muchas veces, para que nosotros lográramos lo que ahora le estamos agradeciendo con todo el corazón, ya que sus sueños y nuestra meta se están cumpliendo después de haber concluido un largo tiempo de esfuerzo, empeño y dedicación. Sin ellos no podríamos haber llegado hasta aquí.

A nuestro tutor Msc. Aldo René Martínez Delgadillo, por su esfuerzo, dedicación.

Sus conocimientos, sus orientaciones, su manera de trabajar, su persistencia, su paciencia y su motivación han sido fundamentales. Le agradecemos por habernos ayudado a culminar nuestra carrera.

A nuestros Profesores ya que nos brindaron sus conocimientos, ayuda y paciencia en lo largo de este camino universitario.

A nuestros familiares y amigos por su apoyo y motivación para que saliéramos adelante. Y a todos aquellos que de una u otra manera nos desearon éxitos en nuestra tesis.

3. DEDICATORIA

Dedicamos este trabajo a Dios, Nuestro Padre Creador quien nos dio la vida, las oportunidades sabiduría y fuerza de culminar nuestros estudios.

A nuestros padres que nos han apoyado a lo largo de nuestras vidas.

A las personas que con su apoyo hicieron posible la culminación de nuestra carrera y nuestra realización como profesionales.

4. INTRODUCCIÓN

TPV es el acrónimo de **terminal punto de venta** (en inglés "*POS terminal*" o "*Point of sale terminal*"). Hace referencia al dispositivo y tecnologías que ayudan en las tareas de gestión de un establecimiento comercial de venta al público.

Estos locales pueden contar con sistemas informáticos especializados que ayudan en las tareas de gestión del punto de venta mediante una interfaz accesible para los vendedores que se denomina **terminal de punto de venta** o TPV.

Los TPV permiten la creación e impresión del ticket de venta mediante las referencias de productos, realizan diversas operaciones durante todo el proceso de venta, así como cambios en el inventario.

También generan diversos reportes que ayudan en la gestión del negocio.

Los TPV se componen de una parte hardware (dispositivos físicos) y otra software (sistema operativo y programa de gestión).

5. ANTECEDENTES

Hasta hace algunos años las cajas registradoras son y han sido el medio por el cual muchos supermercados, tiendas, distribuidoras, etc. que se dedican a la venta al por menor, han utilizado para realizar las diferentes operaciones de ventas, operaciones que se ven limitadas por estos equipos, que aunque estos hayan evolucionado con el tiempo, desde los más sencillos que poseen dos totalizadores, uno para el cálculo y registro de cada venta y otro para la suma de las ventas efectuadas durante la jornada, así como un dispositivo impresor, hasta los modelos más perfeccionados que permiten una clasificación más o menos detallada de las ventas por sección y, dentro de estas, por vendedor, Estos no permiten obtener información detallada de los artículos que permita la toma de decisiones ni que interactúen con el inventario de una tienda permitiendo establecer mejores controles sobre el stock, es por ello que surgieron los TVP o terminal de punto de venta, estas cajas se parecen más a un ordenador que a una caja registradora ya que cuenta con una serie de periféricos que facilitan la realización de las operaciones de ventas, este tipo de cajas pueden llegar a tener toda la información necesaria sobre los distintos productos que tiene el vendedor a su disposición en la tienda pero para ello es necesario la realización de un sistema que gestione cada una de las operaciones que el TVP debe realizar.

Supermercado salman ya contaba con un sistema administrativo para la gestión de información el cual se llama Exactus ERP (Enterprise Resource Planning) este sistema trabaja de forma modular y permite manejar la ***producción, logística, distribución, inventario, envíos, facturas y contabilidad*** de la compañía, sin embargo supermercado salman solo contaba con los módulos básicos que la empresa sofland (Desarrolladores de exactus), entrega por la compra de este sistema, los cuales son: **Administración de sistema, Inventario, Contabilidad general, Control bancario y cuentas por Cobrar**, además de un módulo que se adquirió tiempo después el cual es **Compras**.

Aprovechando que **sofland** ya había desarrollado el módulo de facturación, se decidió adquirirlo para integrarlo con los demás módulos que estaban funcionando en salman.

A inicio de julio del 2009 se implementó la facturación de **Exactus** el cual permite realizar y emitir ticket de ventas, pero esta facturación era demasiada compleja y tardía al momento de crear una factura de venta por lo que no agilizaba la atención al cliente, por lo que los clientes se quejaban de la atención, en este momento se requería agilizar este proceso y se estableció una comunicación con sofland para buscar una solución, por lo que enviaron a un asesor que tratara de ayudar a resolver este problema y a través de él se adquirió un punto de venta que no era parte de exactus, pero se integraba a sus bases de datos y trabajaba mucho más ágil, pero tardaba cierto tiempo en imprimir una ticket de venta además de que se presentaron otros problemas como era que el sistema quedaba bloqueado al momento que se estaba ejecutando una transacción de inventario desde el módulo de inventario de exactus pero era necesario aplicar los documentos de inventario para mantener el sistema actualizado, era imprescindible que la facturación nunca se viera bloqueada por ninguna razón por lo que se estudió bien el caso en busca de la mejor solución y es aquí donde se requirió que se trabajara en un punto de venta que permitiera aplicar los documentos de inventario y que se facturara sin ninguna interrupción.

6. JUSTIFICACIÓN

Supermercado salman por la necesidad de resolver los problemas de bloqueo en los procesos de transacciones a nivel de base de datos y la complejidad en la realización de las facturas e impresión de las mismas, se ve en la necesidad de desarrollar su propio sistema de punto de venta, el cual tenga la capacidad de permitir facturar sin interrupción y también de crear las facturas de una forma ágil y fácil para cualquier usuario.

Tomando la experiencia obtenida con TPV anteriores, y las desventajas del módulo de facturación del sistema Exactus ERP se hicieron mejoras significativas de cara a los usuarios y separando las bases de datos de Exactus y el TPV permitiendo que ambos trabajen de forma aislada y a la vez sincronizada. Aplicando las innovaciones tecnológicas que en la actualidad están disponibles y que se usan con frecuencia en la industria de la comercialización como son: Pantallas Táctiles, Scanner de barras, Impresoras Térmicas tipo Ticket, cajones de dinero, base de datos, lenguajes de programación, permite dar una mejor atención a los clientes al momento de facturar sus productos.

7. OBJETIVOS

Objetivo General:

Desarrollar un sistema Terminal Punto de Venta, orientado a la cadena de supermercados salman ubicados en león – Nicaragua.

Objetivos Específicos:

1. Gestionar las ventas registrando y emitiendo facturas de una forma más ágil y fiable, a través de una interfaz fácil e intuitiva.
2. Permitir tener un mayor control sobre los inventarios dentro de la tienda.
3. Obtener un mejor control en las cuentas por cobrar al cliente de crédito.
4. Generar información comercial para la toma de decisiones.

8. DISEÑO METODOLÓGICO

Para llevar a cabo el desarrollo de este trabajo se implementó la metodología del ciclo de vida clásico o en cascada.

Entiéndase por ciclo de vida todas aquellas etapas que asocian una serie de tareas que se deberán realizar incluyendo los documentos que cada una de estas generarán, y que serán las que servirán de entrada a las siguientes fases; por las cuales pasa el software, desde que el proyecto es concebido hasta que éste es dejado de usarse.

8.1. Modelo en Cascada

Descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas la entrada requerida por la siguiente. En éste modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

8.2. Actividades del Ciclo de Vida a en Cascada.

8.2.1. Análisis: Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada a SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

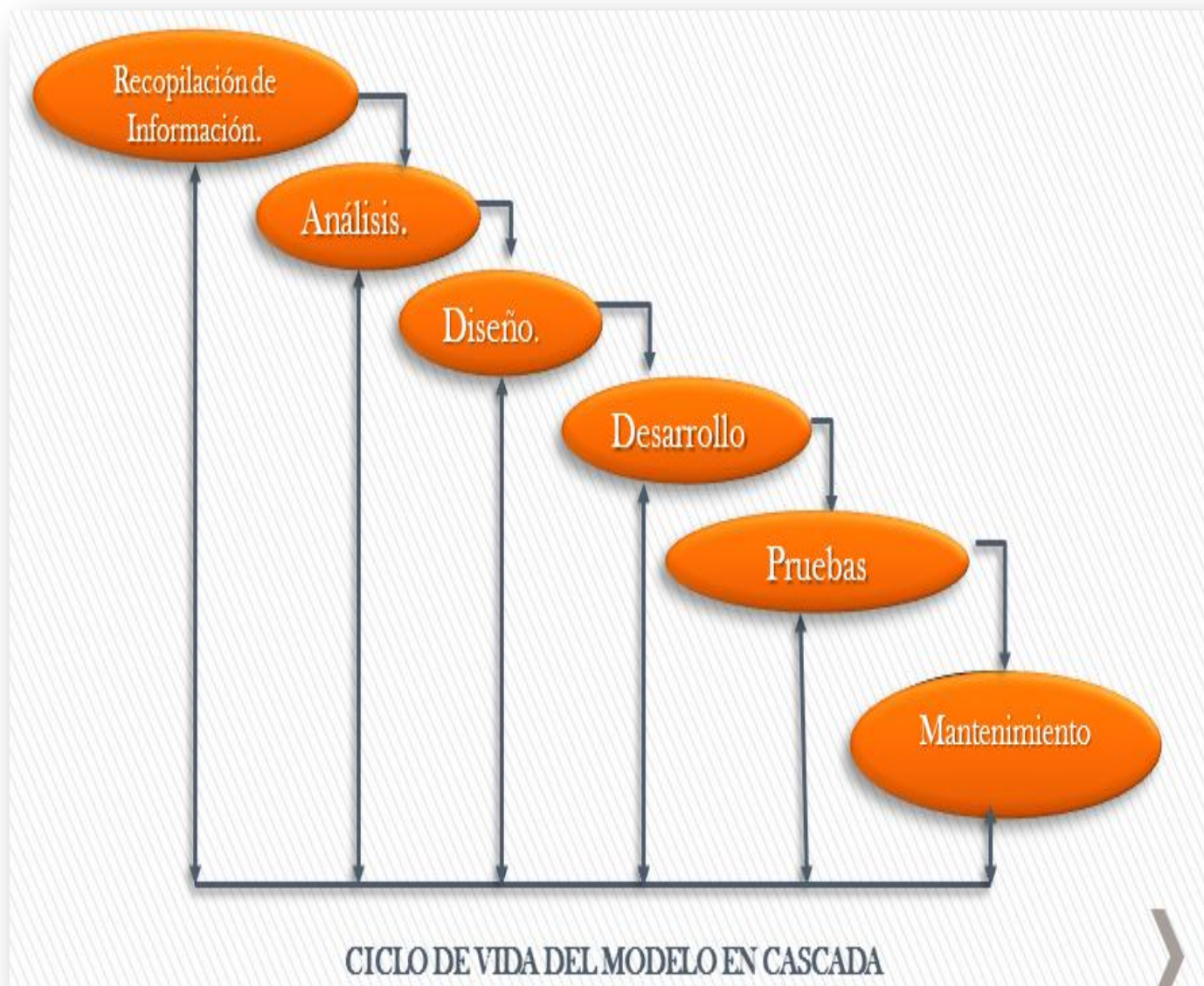
Es importante señalar que en esta etapa se deben reunir toda la información que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

8.2.2. Diseño: Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el Documento de Diseño estructural relacional global o Datos, que contiene la descripción de la estructura del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras. Otros documentos que salen de esta etapa son el diagrama de clase, diagrama relacional y diseño arquitectónico.

8.2.3. Prueba: El software debe ser probado para descubrir los defectos que puedan existir en la función, en la lógica y en la implementación.

8.2.4. Mantenimiento: La fase de mantenimiento se centra en el cambio. Esta fase aplica los pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existe. Durante la fase de mantenimiento se centran tres tipos de cambios: ***corrección, adaptación y mejora.***

Figura 1. Gráficamente el modelo de ciclo de vida clásico se representa de la siguiente manera:



8.3. Materiales

8.3.1. Hardware: En nuestro proyecto a nivel de hardware emplearemos las siguientes herramientas:

3 PC con las siguientes características:

- Memoria RAM de 1 GB
- 160 GB en Disco Duro
- Procesador Pentium 4 de 2.66 GHz.
- Pantalla Táctil.
- Impresora Térmica Epson TMU-IV.
- Cajón de Dinero.
- Scanner de Códigos de Barras.

8.3.2. Software: Las herramientas software que emplearemos en nuestra aplicación son:

- Microsoft Office Word 2010.
- Microsoft Office Visio 2010.
- Windows XP (Sistema Operativo)
- SQL Server 2005.
- Visual Studio 2008.
- Cristal Report XI.

9. MARCO TEÓRICO

Los primero TPV son implementados de las 1879, cuando se inventó la primera caja registradora de dinero, totalmente mecánica (modelo Ritty).

En la mayoría de los casos el cajón para el dinero solo puede abrirse luego de una venta, excepto cuando se usa una llave especial. Al mismo tiempo, la apertura del cajón genera un sonido de aviso. Ambas características han tenido el objetivo de mejorar las posibilidades de control por parte del dueño o encargado del comercio, evitando los hurtos.

Las primeras cajas registradoras eran enteramente mecánicas y no entregaban recibo. Con el paso de los años y el perfeccionamiento de los sistemas fiscales, los países han implementado la facturación obligatoria de las ventas, que se perfecciona con el registro y entrega de un ticket o factura, por lo general con el fin tributario de recaudar el Impuesto al valor agregado u otros impuestos que graven las ventas.

Los Terminales puntos de Ventas están Compuesto principalmente de **SOFTWARE y HARDWARE.**

9.1. SOFTWARE

El T.P.V tiene su programa de gestión o software que puede ser:

9.1.1. A medida.

Contienen software específico para una única empresa. Suelen ser mucho más caros y las modificaciones o actualizaciones van siempre ligadas a la disponibilidad de la empresa que desarrolla ese software.

9.1.2. Comerciales

Dentro de este grupo pueden estar predefinidos para tiendas de ropa, hostelería, ferretería, farmacia, videoclubes o TPV de carácter general. Suelen estar diseñados para un establecimiento tipo del sector al que va dirigido y no admite cambios específicos. Suelen ser mucho más económicos.

9.1.3. Específicos

Algunas empresas fabricantes de TPV desarrollan un software específico para un tipo de negocio en concreto. Este software puede ir junto con un sistema operativo propio del fabricante embebido en la propia memoria del terminal (no tienen discos duros ni trabajan con sistemas operativos como los PC) o bien instalados en el disco duro del terminal, como cualquier otro PC. Este tipo software cuenta con múltiples opciones de configuración en función de las necesidades de un cliente concreto, siempre que trabaje dentro de este sector en concreto.

9.2. HARDWARE

Los tipos de TPV actual son:

9.2.1. Compacto

Se trata de equipos pensados específicamente para trabajar como punto de venta. Estos equipos pueden integrar los periféricos necesarios como impresoras de recibos, pantallas para el cliente, y otros, o no. En todo caso, siempre poseen un conjunto rico de conexiones y puertos (en algunos casos dichos puertos incluyen un tipo de alimentación especial) para poder utilizar los periféricos más comunes, necesarios en un punto de venta. Es decir, integran la CPU, la impresora, la pantalla y el teclado en una sola máquina. Suelen equipar pantallas táctiles, aunque permiten la conexión de otras interfaces de usuario y periféricos, como teclados, cajón portamonedas etc. Pueden parecer un PC normal, pero hay varias características que los diferencian de un PC común: Utilizan placas de PC industriales o diseñadas a medida, de manera que muchos de ellos no precisan ventilación (aumento de fiabilidad, disminución de mantenimiento), pantallas que pueden resistir el vertido de líquidos, cuya resistencia es muy superior a un teclado común, discos de estado sólido SSD, que no contienen partes mecánicas (inmunes a vibraciones), consiguiendo más fiabilidad, menor consumo del terminal, y un mantenimiento más bajo. Además estos equipos reducen casi totalmente las averías provocadas por la desconexión de los diversos cables, que en los TPV modulares conectan los diferentes subconjuntos. Habitualmente los TPV incluyen versiones industriales de sistema operativo (licencias full y similar), las cuales permiten un inicio más rápido y eliminan características innecesarias, lo cual también incrementa la estabilidad.

9.2.2. Modular

Suelen ser equipos basados en un PC normal con un software instalado sobre un sistema operativo convencional. Todos los componentes necesarios se deben cablear y conectar a la CPU a través de sus diferentes puertos e interfaces. Si bien pueden ejecutar la mayoría de aplicaciones pensadas para punto de venta, resultan más voluminosos, menos fiables, y requieren más mantenimiento. No hay que olvidar que los PC no están pensados para estar en un punto de venta, si no en una oficina o el hogar. No obstante pueden ser una solución más económica.

Los elementos que habitualmente componen este tipo tan extendido de TPV Modular, son los que se describen a continuación:

9.2.3. Monitor

Puede ser un monitor normal de PC o uno que incorpore un sistema táctil (el pulsar con el dedo o puntero emula el "Crick" del ratón) que evita la utilización de ratón y agiliza las labores de gestión de cobro al permitir al usuario manejar más fácilmente los menús. Existen muchos fabricantes de pantallas táctiles y diferencias importantes en el precio, la calidad, la tecnología usada y la vida media de esta interfaz.

9.2.4. Teclado

Puede ser un teclado de PC normal o uno de reducidas dimensiones para ahorrar espacio. Aunque lo ideal es disponer de teclados específicos para TPV configurables (programables) con accesos directos y posibilidad de incorporar imágenes o símbolos a dichas teclas.

9.2.5. Impresora de recibos

Sirve para expedir el recibo o resguardo de compra al cliente. Estas pequeñas impresoras pueden ser matriciales, térmicas y de tinta (poco usuales). Las matriciales son las más lentas y más económicas; utilizan una cinta de tinta.

No obstante permiten obtener una copia del resguardo mediante calco ya que la impresión se realiza por impacto. Las térmicas son más rápidas y versátiles, utilizar un rollo de papel térmico y permiten la impresión de gráficos (logotipos). El ticket se puede deteriorar con el paso del tiempo si se encuentra expuesto a

temperaturas elevadas, siempre en función de la calidad del papel. Existen múltiples tipos de impresoras de ticket/factura, grandes factura (slip-printer), con doble imprenta, o incluso impresoras que contemplan todas estas funciones a la vez. Para el sector de la hostelería existen impresoras específicas para el envío de comandos a la cocina. En cumplimiento de normativas de sanidad algunos fabricantes utilizan carcasas de acero inoxidable. Estas impresoras de cocina por lo general cuentan muchas veces con avisadores acústicos que indican la entrada de una comando al personal de la cocina. También existen impresoras inteligentes que pueden trabajar de diferente forma en función de las necesidades de un negocio en especial.

9.2.6. Cajón de Dinero

Los cajones portamonedas más usuales se conectan a un puerto de tipo RJ11, que incorpora la propia impresora de tickets; el cajón se abre automáticamente (sin necesidad de llave) en el momento del cobro. Otro modelo de cajón es el conectado a través de un puerto serie directamente a la CPU y se abre al recibir un impulso por dicha conexión o manualmente con la llave.

Existen cajones portamonedas de seguridad, más próximos a una caja fuerte. Tienen elevados espesores en sus carcasas de acero y cuentan con gavetas extraíbles para realizar entradas y salidas de cambio de efectivo. Cajones verticales para espacios reducidos o incluso terminales para control de efectivo donde el cajero no toca nunca el dinero siendo el cliente el que realiza el pago directamente en el terminal y este quien realiza el computo del entregado y la devolución del cambio.

9.2.7. Lector de código de barras.

Dispositivo que interpreta los símbolos del código de barras (usualmente EAN, UPC, Codebar, Code 39 o GS1-128) que el fabricante imprime en la etiqueta de los productos. El código de barras suele ser una serie de dígitos o caracteres que representan unívocamente un producto. La única función del lector es transcribir dicho código como si fuera tecleado por el cajero, evitar posibles errores al teclearlo además de reducir significativamente el tiempo empleado por el vendedor. Además, el sistema dispone de una base de datos que identifica dicho código con el artículo, su descripción, su precio y otras características relevantes.

Existen códigos de barras variables que además de identificar el producto incorporan información adicional. Por ejemplo el peso que registra una balanza. Una parte del código sirve para la identificar al producto y la otra indica la cantidad. Es muy habitual encontrar estos códigos de barras variables en grandes superficies donde los productos vendidos a granel son pesados en balanzas diseñadas para el autoservicio.

9.2.8. Pantalla o visor electrónico del TPV.

Pantalla de visualización de datos donde el cliente puede ver el resultado de la operación de venta u otra información adicional antes de imprimir el ticket. Suelen tener dos filas de 20 caracteres y suelen ser de tipo: LCD, VDF (retro iluminado) o gráficos (mediante puntos y no caracteres). En la actualidad algunos fabricantes utilizan pequeñas pantallas TFT como visores para el cliente, donde además de la información para el usuario en el momento de la venta, se pueden mostrar gráficos con ofertas, sugerencias de platos, etc.

9.2.9. Lector de banda magnética.

Dispositivo que es capaz de transcribir la información contenida en la banda magnética de una tarjeta plástica (normalmente las tarjetas de crédito o débito de los bancos) para realizar una transacción bancaria en la venta o para identificar una persona, ya sea para acceso a determinadas zonas o para fidelización de clientes, por ejemplo, en videoclubes o gimnasios. Aunque se trata de una tecnología cada vez más en desuso, muchos TPV incorporan este tipo de dispositivo. En la mayoría de casos se limita a la utilización para la identificación del usuario (cajero) o para sistemas de puntos de cliente o fidelización. Solo en grandes superficies podemos encontrar TPV conectados a un servidor que realiza las transacciones periódicamente a las entidades bancarias bajo sus protocolos y autorización. Los diferentes protocolos de comunicación utilizados por los bancos y sistemas de tarjeta bancaria, dificultan el uso de este lector para el pequeño comercio, obligando a este tipo de negocio a recurrir a los clásicos datafonos (Equipos que realizar transacciones electrónicas con las tarjetas de crédito o débito de los bancos).

9.3. Las nuevas tendencias en TPV:

Es disminuir costos y hacer accesible la tecnología para todos los negocios ya sea pequeño o grande, además de que la información siempre esté disponible en cualquier momento o en cualquier dispositivo utilizando conceptos como el de computación en la nube.

9.4. Herramientas de desarrollo.

9.4.1. Visual Estudio 2008.

Microsoft Visual C # 2008, es una plataforma de programación diseñada para la creación de una amplia gama de aplicaciones que se ejecutan en. NET Framework. C # es un lenguaje de programación simple, eficaz, con seguridad, y orientado a objeto. Con sus numerosas innovaciones, C # permite el desarrollo rápido de aplicaciones sin perder la expresividad y elegancia de los lenguajes C-estilo. Visual Studio es compatible con Visual C# con todas las funciones del editor de código, plantillas de proyecto, diseñadores, asistentes de código, un potente y fácil de depurador de uso, y otras herramientas. NET Framework proporciona acceso a una amplia gama de servicios del sistema operativo y otras clases útiles y bien diseñadas que aceleran el ciclo de desarrollo de manera significativa.

9.4.2. Crystal Reports para VS 2008

Crystal Reports es un popular generador de informes basado en Windows (programa de generación de informes) que permite a un programador crear informes a partir de una variedad de orígenes de datos con la mínima escritura de código. Desarrollado por Seagate Software, Crystal Reports puede acceder a los gestores de bases de datos más ampliamente utilizados y se puede integrar datos de múltiples bases de datos dentro de un informe con Open Database Connectivity(ODBC). Crystal Reports utiliza un control ActiveX llamado Crystal Report para establecer una conexión con otro programa. Un programador puede establecer las propiedades del control Crystal Report durante el tiempo de diseño o en tiempo de ejecución. El programador puede utilizar herramientas de automatización llamados expertos para guiarse a través de tareas comunes, como la vinculación e incrustación de informes. Crystal Reports trata a todos los textos, gráficos, y los campos de base de datos como objeto que un programador puede

colocar, organizar y formato en los formularios. El programa también genera un objeto de registros y el código necesario para realizar tareas de programación, tales como lazo de `do` o de cálculos matemáticos.

9.4.3. SQL Server 2005.

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

a. Características de Microsoft SQL Server.

- ✓ Soporte de transacciones.
- ✓ Escalabilidad, estabilidad y seguridad.
- ✓ Soporta procedimientos almacenados.
- ✓ Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- ✓ Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- ✓ Además permite administrar información de otros servidores de datos.

b. T-SQL

T-SQL (Transact-SQL) es el principal medio de programación y administración de SQL Server. Expone las palabras claves para las operaciones que pueden realizarse en SQL Server, incluyendo creación y modificación de esquemas de la base de datos, introducir y editar datos en la base de datos, así como supervisión y gestión del propio servidor. Las aplicaciones cliente, ya sean que consuman datos o administren el servidor, aprovechan la funcionalidad de SQL Server mediante el envío de consultas de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente. SQL Server permite que sean administrados mediante T-SQL. Para esto, expone tablas de sólo lectura con estadísticas del servidor.

La funcionalidad para la administración se expone a través de procedimientos almacenados definidos por el sistema que se pueden invocar desde las consultas de T-SQL para realizar la operación de administración. También es posible crear servidores vinculados (Linked Servers) mediante T-SQL. Los servidores vinculados permiten el funcionamiento entre múltiples servidores con una consulta.

c. *Cliente Nativo de SQL*

Cliente Nativo de SQL es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa nativamente soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos. Cliente Nativo de SQL se utiliza como extensión de SQL Server plugins para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos genéricos.

10. Especificación de Requisitos Software

10.1. Introducción

10.1.1. Propósito

Describir el conjunto de funcionalidades, restricciones y requisitos que debe cumplir la aplicación.

10.1.2. Alcance

El nombre de la aplicación será “POS Salman”, el cual realizará las siguientes funciones:

- a) Buscar artículo a vender ya sea por medio del código de barras impreso en su etiqueta o con el código asignado, con el cual fue registrado en la base de datos.
- b) Filtrar artículos por su descripción en los casos en que no se hayan encontrado a través de sus códigos asociados.
- c) Registrar artículo a vender.
- d) Establecer cantidad de artículos a vender.
- e) Editar cantidad a vender del artículo.
- f) Editar precio de venta del artículo esto con permiso de supervisor.
- g) Borrar artículo anteriormente seleccionado.
- h) Obtener peso del artículo, esto si se dispone de un scanner balanza (Opción configurable).
- i) Permite distintas forma de pago con la que se cancelará la factura (Efectivo, Tarjeta, Crédito, Promociones o Mixto).
- j) Establecer el cliente con el que se generara la factura, de contado o clientes de crédito.
- k) Filtrar el cliente a través del nombre.
- l) Generar factura.
- m) Imprimir factura o ticket de venta.
- n) Reimprimir ultima factura generada.
- o) Acceder a nivel de supervisor para:
 - ✓ Reimprimir factura, filtrándola por número consecutivo.
 - ✓ Anular factura, filtrándola por número consecutivo.

- ✓ Establecer porcentaje de descuento ya sea por factura actual o por artículo seleccionado.
- ✓ Abrir caja de dinero.
- ✓ Crear pre lectura, imprimiendo reporte de ventas totalizadas por forma de pago.
- ✓ Realizar cierre de ventas.

10.2. DESCRIPCION GENERAL

10.2.1. CARACTERÍSTICAS DEL USUARIO:

Los usuarios del sistema tendrán que tener conocimientos básicos de computación y caja además de conocimientos de arqueos los cuales serán hechos por el personal de contabilidad, todos los usuarios serán entrenados en el uso y manejo del sistema.

10.2.2. REQUISITOS GENERALES:

Para la instalación del sistema de necesita:

- a) **Net Framework 3.5 SP1:** Es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este framework.
- b) **Windows Installer 4.5:** Es un motor para la instalación, mantenimiento y eliminación de programas en plataformas Microsoft Windows.
- c) **Pantalla táctil:** es una pantalla que mediante un toque directo sobre su superficie permite la entrada y salida de datos.
- d) **Impresora Térmica Epson TMU-IV.**
- e) **Scanner de Códigos de Barras.**
- f) **Cajón de dinero.**

10.2.3. RESTRICCIONES GENERALES

El sistema operativo a usarse será Microsoft Windows XP o superiores. Únicamente podrán acceder al sistema usuarios registrados. Una vez dentro del sistema las funciones a las que se tendrán acceso dependerán del usuario que acceda.

10.2.4. TIPOS DE USUARIOS

- a) **Cajeros:** Estos son los que crean las facturas, producto por producto, para posteriormente cobrarla.
- b) **Supervisores:** Tienen más privilegios que los cajeros ya pueden gestionar las facturas ya creadas, así como los cierres de cajas.

11. Relación con otros proyectos

Supermercado salman cuenta con un sistema llamado EXACTUS ERP, el cual se encarga de las gestiones administrativas tales como:

1. Control de inventario.
2. Compras.
3. Cuentas por pagar.
4. Cuentas por cobras.
5. Contabilidad general.
6. Facturación.

Todo el personal administrativo realiza sus tareas diarias dentro de este sistema, eso significa que desde este sistema se alimenta el inventario para cada bodega, se calculan los precios, se establecen las promociones de los artículos, se asocian sus impuestos, etc.

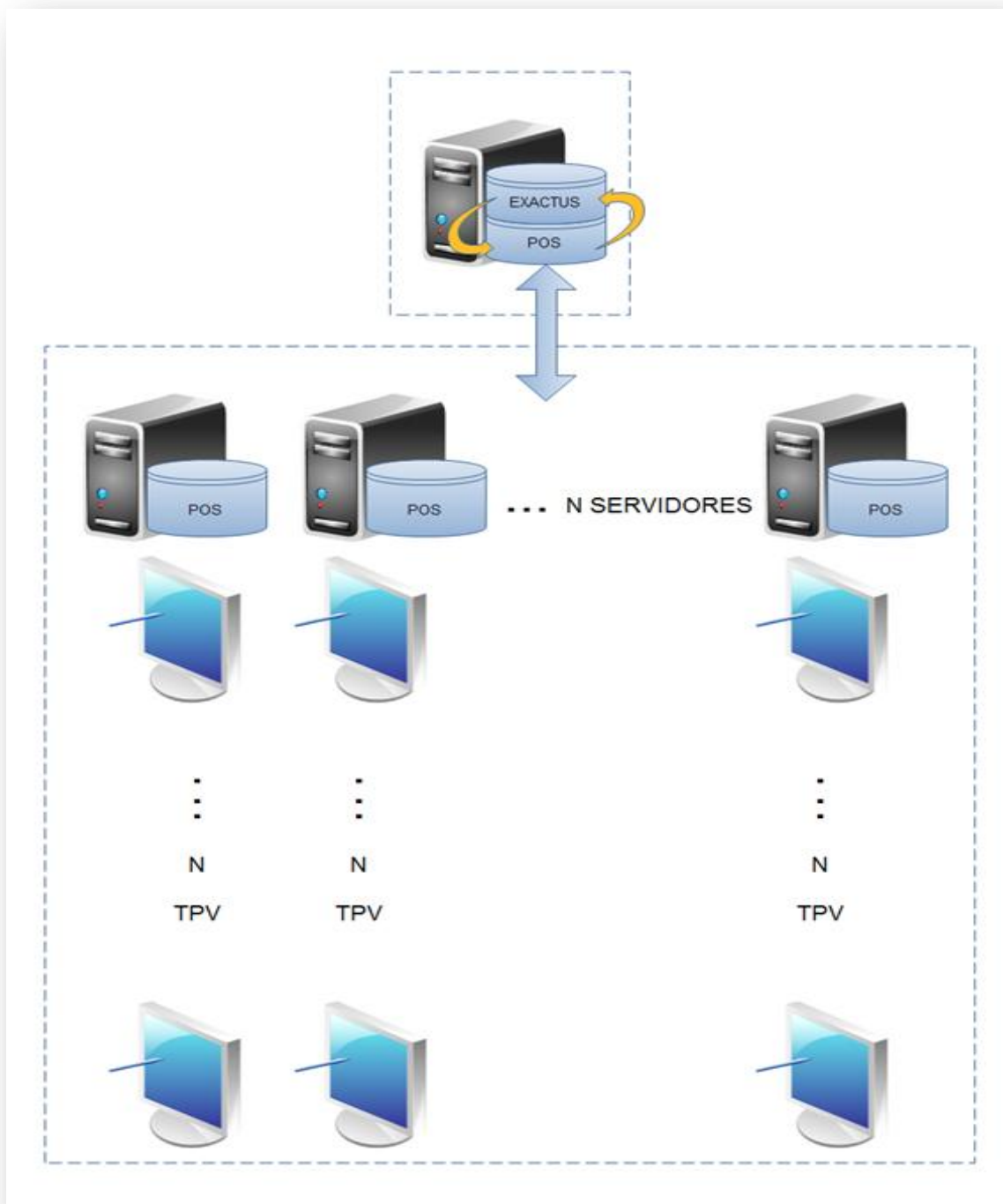
Cada vez que se aplica un documento en inventario el sistema Exactus realiza un bloqueo transaccional lo que repercute en el TPV al momento de estar realizando o creando una factura, por lo que se decidió crear una base de dato que contenga la información que el sistema TPV necesita para poder crear una factura.

La sincronización se trabaja en dos niveles una para mover los datos que los usuarios están trabajando en Exactus hacia una base de datos que se encuentra en el mismo servidor llamada POS, y otra que se encarga de estar sincronizando los datos desde la base de datos POS hacia las bases de datos que se encuentra en otro servidor y es donde se conectan los TPV.

El primer nivel se realiza con triggers en las tablas de Exactus, e insertando, actualizando y eliminando el la base de datos POS y los datos del POS hacia Exactus se mueven a través de procedimientos almacenados al final del día para que todo lo que se realice en el POS se vea reflejado y se creen las transacciones en Exactus que es quien maneja el inventario.

El segundo nivel es una replicación a nivel de SQL server quien es el encargado de sincronizar los datos entre las base de datos que se encuentran en los diferentes servidores y utilizamos la replicación de mezcla SQL server 2005.

Figura 2. Gráfica que detalla las sincronizaciones.



12. REQUISITOS ESPECIFICOS

12.1. Inicio de sesión

Especificación.

12.1.1. Introducción:

El Cajero debe ingresar su nombre de usuario y contraseña (previamente registrado) para poder ingresar al sistema.

12.1.2. Entradas:

Datos proporcionados por el cajero

- ✓ Usuario
- ✓ Contraseña

12.1.3. Procesos:

Se validan los datos introducidos y si el usuario y contraseña están registrados en la BD y si el usuario no tiene abierta otra caja, se da acceso al sistema.

12.1.4. Salida:

Se presentara la interface principal del sistema.

12.1.5. Interfaz Hardware

Se utilizara cualquier computador que tenga el sistema instalado y tenga acceso a la red LAN del servidor.

12.1.6. Interfaz Software

El proceso interactúa con la BD en el cual se encuentra almacenada toda la información

Nota aclaratoria: Las interfaces de Hardware, Software y de comunicaciones son las mismas en todas las siguientes referencias, así que se omitirán para evitar repeticiones.

12.2. Agregar Productos a una Factura:

12.2.1. Introducción:

El usuario agrega productos para crear una factura.

12.2.2. Entrada:

Datos proporcionados por el cajero.

- ✓ Escaneando un producto
- ✓ código de barras
- ✓ código interno
- ✓ Cantidad

12.2.3. Procesos:

Con el dato proporcionado por el cajero el sistema busque la información de este, verificando que exista en la base de datos y consultando si tiene descuentos asociados e impuestos, para establecer el precio final y agregar el producto a la factura actual.

12.2.4. Salida:

Se presenta en pantalla la descripción, cantidad y el precio total de la línea.

12.3. Filtrar producto:

12.3.1. Introducción:

El usuario podrá listar el universo de los productos almacenados en la base de datos con el objetivo de realizar una búsqueda por medio de la descripción.

12.3.2. Entrada:

El usuario proporcionara una breve descripción del producto haciendo uso del comodín *.

12.3.3. Proceso:

Se realiza un filtro de los productos registrados, dada la descripción proporcionada por el usuario.

12.3.4. Salida:

El sistema muestra las posibles coincidencias de la descripción proporcionada.

12.4. Pesar producto:

12.4.1. Introducción:

Usuario puede verificar y pesar un producto al momento de agregar una línea a la factura.

12.4.2. Proceso:

Cuando es un producto que debe ser pesado el usuario puede pesarlo presionando sobre el botón de peso.

12.4.3. Salida:

Se muestra el peso en pantalla.

12.5. Modificar línea de la factura:

12.5.1. Introducción:

- ✓ El usuario podrá modificar la cantidad y el precio de la factura.

12.5.2. Entrada:

El usuario introduce la cantidad o el precio.

- ✓ Cantidad
- ✓ Precio

12.5.3. Proceso:

El usuario selecciona la línea de la factura que desea modificar, luego introduce la nueva cantidad y el precio, si se pretende cambiar el precio, el sistema le pedirá que se autentifique como supervisor de no hacerlo el precio no podrá ser actualizado.

12.5.4. Salida:

Se presentara en pantalla la nueva cantidad o el nuevo precio total de la línea en base al precio introducido por el usuario.

12.6. Eliminar línea de la factura:

12.6.1. Introducción:

El usuario tiene la posibilidad de eliminar una línea de la factura que se está realizando.

12.6.2. Proceso:

El usuario elimina un producto dando doble clic sobre el o seleccionándolo y luego presionando el botón eliminar.

12.6.3. Interfaz externa:

a. Interfaz de usuario:

Al momento de eliminar un producto se muestra un mensaje de texto, solicitando confirmación de esta acción.

12.7. Reimprimir ultima factura:

12.7.1. Introducción:

El usuario puede reimprimir la última factura cobrada.

12.7.2. Proceso:

El usuario pulsa sobre el botón reimprimir en la pantalla principal.

12.8. Filtrar cliente:

12.8.1. Introducción:

El usuario podrá listar el universo de los clientes almacenados en la base de datos con el objetivo de realizar una búsqueda por medio del nombre del cliente.

12.8.2. Entrada:

El usuario proporcionara una breve descripción del nombre del cliente haciendo uso del comodín *.

12.8.3. Proceso:

Se realiza un filtro de los clientes registrados, dada la entrada proporcionada por el usuario.

12.8.4. Salida:

El sistema muestra las posibles coincidencias de la descripción del nombre proporcionado.

12.9. Agregar formas de pago:

12.9.1. Introducción:

El usuario debe registrar la o las formas de pago con la que cancelara la factura.

12.9.2. Entrada:

El usuario introduce el monto de la forma de pago elegida:

✓ Efectivo

- ✓ Tarjeta.
- ✓ Crédito
- ✓ Cheque
- ✓ Promociones

12.9.3. Proceso:

El usuario selecciona una de las formas de pago, y le asigna un monto a dicha forma. Y si ese monto es menor al total de la factura tiene que elegir otra forma de pago hasta lograr cancelar el total de la factura.

Si la forma de pago es crédito se verifica que el saldo del cliente sea menor o igual al monto asignado a la forma de pago, también se verifica que el cliente al que le está cobrando la factura sea un cliente diferente al cliente de contado.

12.9.4. Salida:

Se presenta en pantalla una lista de las formas de pago y sus respectivos montos.

12.10. Cancelar factura.

12.10.1. Introducción:

El usuario cancela la factura realizada.

12.10.2. Proceso:

Después de haber seleccionado el cliente, de registrar las formas de pago y de verificar que no quede saldo pendiente, el usuario debe cancelar la factura.

12.10.3. Salida:

El sistema muestra un mensaje con el monto del vuelto, en el caso de no haber vuelto el monto reflejado es cero. De igual manera se imprime un ticket con una lista de todos los productos que pertenecen a la factura, indicando la descripción, cantidad, montos de cada producto. También el subtotal de la factura antes de impuesto, el total del impuesto y el total general de la factura. En el encabezado de la factura se muestra el logo del salman, el número de ruc, la dirección, el número de factura, la fecha y hora de la factura, la bodega donde se realizó la factura, el número de caja y el cajero que la realizó.

12.11. Acceso al menú de supervisor.

12.11.1. Introducción.

El supervisor debe ingresar su nombre de usuario y contraseña (previamente registrado) para poder ingresar al menú supervisor.

12.11.2. Entradas.

Datos proporcionados por el supervisor

- ✓ Usuario
- ✓ Contraseña

12.11.3. Procesos:

Se validan los datos introducidos y si el usuario y contraseña están registrados en la BD y si el usuario tiene privilegios, se da acceso al menú supervisor.

12.11.4. Salida:

Se presentara la interface con el menú de supervisor.

12.12. Reimprimir factura

12.12.1. Introducción:

El supervisor podrá listar el universo de las facturas almacenados en la base de datos con el objetivo de realizar una búsqueda por medio del consecutivo de la factura.

12.12.2. Entrada:

El usuario proporcionara una breve descripción del consecutivo de factura haciendo uso del comodín *.

12.12.3. Proceso:

Se realiza un filtro de las facturas registrados, dada la entrada proporcionada por el usuario.

12.12.4. Salida:

El sistema muestra las posibles coincidencias de la descripción del número de factura proporcionado.

12.13. Anular factura.

12.13.1. Introducción:

El supervisor podrá listar el universo de las facturas almacenados en la base de datos con el objetivo de realizar una búsqueda por medio del número de factura y poder elegir una para posteriormente anularla.

12.13.2. Entrada:

- ✓ El supervisor proporcionara una breve descripción del número de factura haciendo uso del comodín *.
- ✓ El supervisor escribe el motivo por el cual se anula esta factura.

12.13.3. Proceso:

Se realiza un filtro de las facturas registradas, dada la entrada proporcionada por el usuario luego se selecciona la factura a anular y se pulsa dos veces sobre esta o sobre “Enter”, luego de esto se solicita que se escriba el motivo de la anulación.

12.13.4. Salida:

Se imprime un ticket con una lista de todos los productos que pertenecen a la factura, indicando la descripción, cantidad, montos de cada producto. También el subtotal de la factura antes de impuesto, el total del impuesto y el total general de la factura. En el encabezado de la factura se muestra el logo del salman, un rotulo indicando que es una factura anulada, el número de ruc, la dirección, el número de factura, la fecha y hora de la factura, la bodega donde se realizó la factura, el número de caja y el cajero que la realizó.

12.14. Establecer descuento extra.

12.14.1. Introducción:

12.14.2. Entrada:

- ✓ Se establecer un monto representando el porcentaje de descuento que se desea aplicar.
- ✓ Se tiene que elegir a quien se aplicara el descuento si:
 - Al artículo seleccionado
 - A la factura que se está realizando
- ✓ Seleccionar si se quiere:
 - Agregar descuento
 - Remplazar descuento

12.14.3. Salida:

Se presenta el nuevo precio del producto (si se aplicó el descuento a un producto seleccionado) o el nuevo total de la factura, calculados en base al descuento agregado.

12.15. Pre Lectura.

12.15.1. Introducción:

El supervisor puede imprimir un reporte detallado por formas de pago, de los datos de las ventas realizadas por la caja, desde su apertura hasta el momento de esta acción.

12.15.2. Proceso:

Se pulsa sobre el botón pre-lectura dentro del menú de supervisor, esta acción calcula los totales detallados por formas de pago.

12.15.3. Salida:

Se imprime un reporte detallado por forma de pago, de los datos de ventas realizadas por la caja, desde su apertura hasta el momento de esta acción.

12.16. Cierre.

12.16.1. Introducción:

El supervisor da acceso al cajero para que este detalle los datos de ventas totales conforme el efectivo, facturas de crédito, facturas cobradas con cheque, promociones y los Boucher de tarjetas que tiene físicamente. Luego de esto el supervisor puede realizar el cierre de esta caja.

12.16.2. Entrada:

El cajero detalla:

- ✓ El efectivo por denominación de moneda o billete, córdobas y dólares.
- ✓ La suma total de todas las factura de crédito.
- ✓ El monto total de ventas cobradas con tarjetas en el pos de Banpro.
- ✓ El monto total de ventas cobradas con tarjetas en el pos de Credomatic.
- ✓ La suma total de todas las facturas cobradas con cheque.
- ✓ La suma total de todas las factura con forma cobradas con la forma de pago promoción.

12.17. El supervisor.

- ✓ Registrar cualquier observación adicional.

12.17.1. Proceso:

El supervisor da acceso a que el cajero detalle el número de billetes o monedas tanto córdobas y dólares en base a este detalle el sistema calcula el total de efectivo que el cajero tiene físicamente, también registra el total de crédito, tarjetas, cheques y promociones, todo en base al físico que se tiene como soporte de cada forma de pago distinta al efectivo. Luego que el cajero realiza este detalle, el supervisor agrega cualquier observación adicional, si este tiene alguna, y procede a verificar estos datos con la pre lectura antes mencionada, de no presentar anomalías realiza el cierre de la caja, de lo contrario se realiza una verificación del físico que el cajero detallo.

12.17.2. Salida:

Se imprime un reporte totalizado de las formas de pago que el sistema tiene registrado y se compara con los datos que el cajero registro especificando donde hay alguna diferencia, tambien se muestra el monto total de ventas, el detalle de las facturas de credito (este detalle es por las empresas a las que pertenecen los clientes), el detalle de las denominaciones registradas por el cajero, el detalle de las facturas anuladas con sus motivos de anulacion, las observaciones que el supervisor registro (si hubo alguna), el espacio donde firma el cajero y el supervisor, en el encabezado se muestra el número de apertura de la caja, el numero de caja, el cajero que abrio la caja, la fecha, hora de apertura y la hora que se realizo el cierre.

12.18. Abrir cajon de dinero.

12.18.1. Introducción:

El supervisor puede abrir la caja de dinero en el momento que sea necesario.

12.18.2. Proceso:

El supervisor pulsa sobre el boton abrir caja de dinero para que desde el programa se mande un comando a la impresora quien es la que se encarga de gestionar esta apertura.

12.18.3. Salida:

Se abre la caja de dinero.

Figura 3. DIAGRAMA CASOS DE USOS

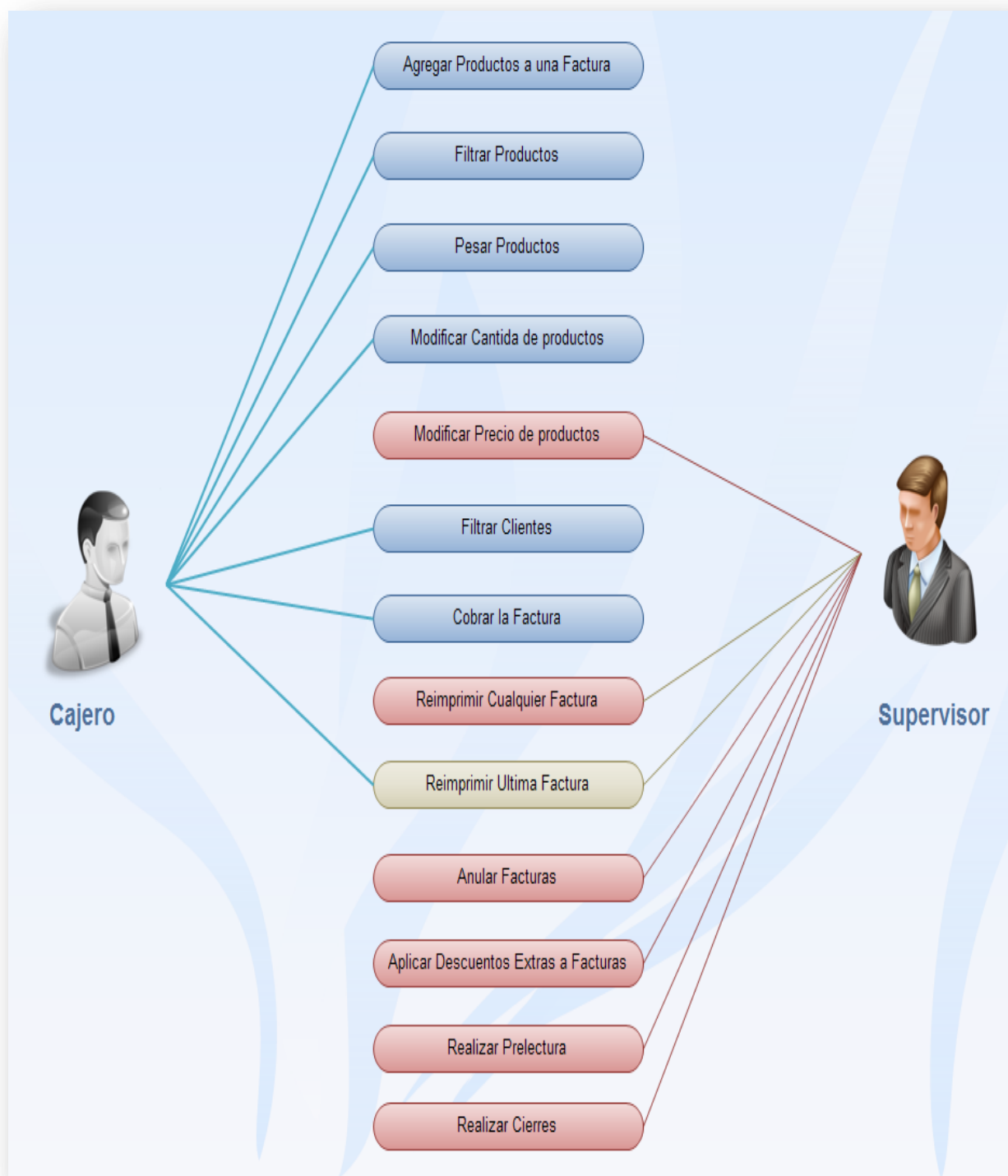


Figura 4. DIAGRAMA DE FLUJO DE DATOS

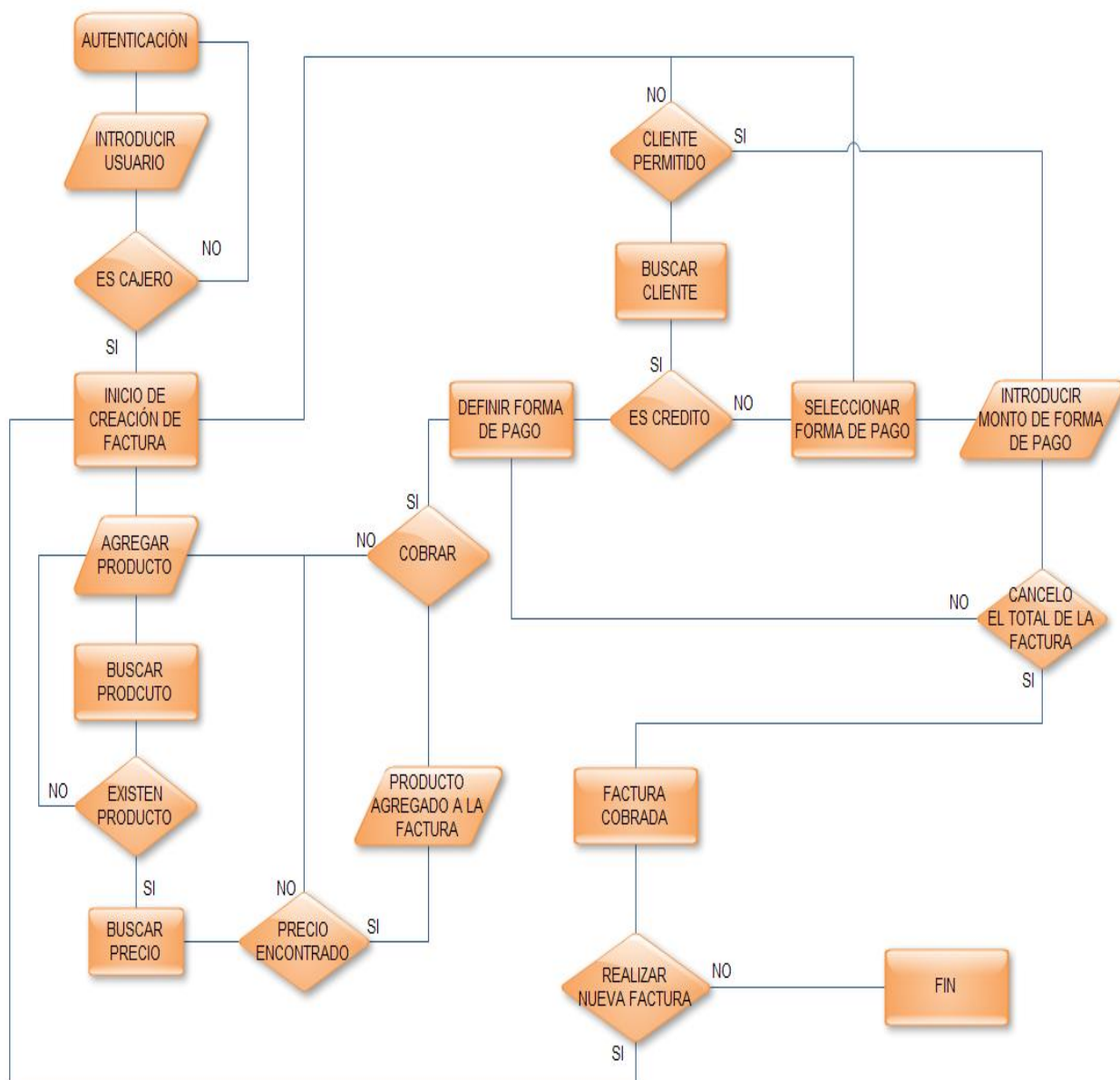


Figura 5. ROLES DE LOS USUARIOS.

	Factura	Cobrar	Descuentos	Cierres
Cajero	✓ Agregar Productos	✓ Filtrar Clientes		
	✓ Filtrar Productos	✓ Agregar Formas de Pagos		
	✓ Pesar Productos	✓ Generar la Factura.		
	✓ Modificar Cantidades	✓ Reimprimir Factura		
	✓ Modificar Cantidades			
Supervisor			✓ Aplicar descuentos Extras	✓ Realizar Prelectura
			✓ Modificar Descuentos	✓ Realizar cierre de Caja.

Figura 6. DIAGRAMA DE CLASES PARTE 1.

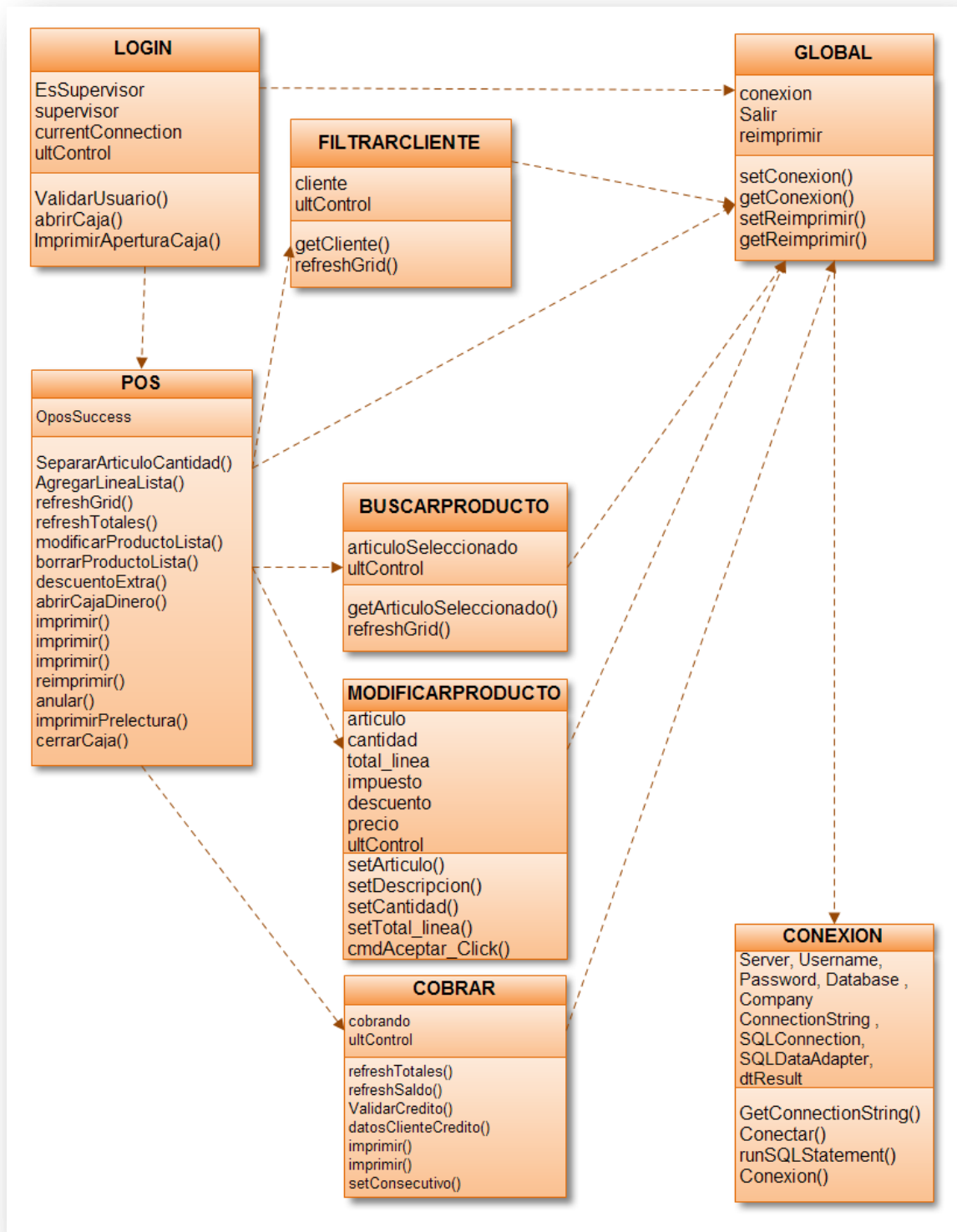
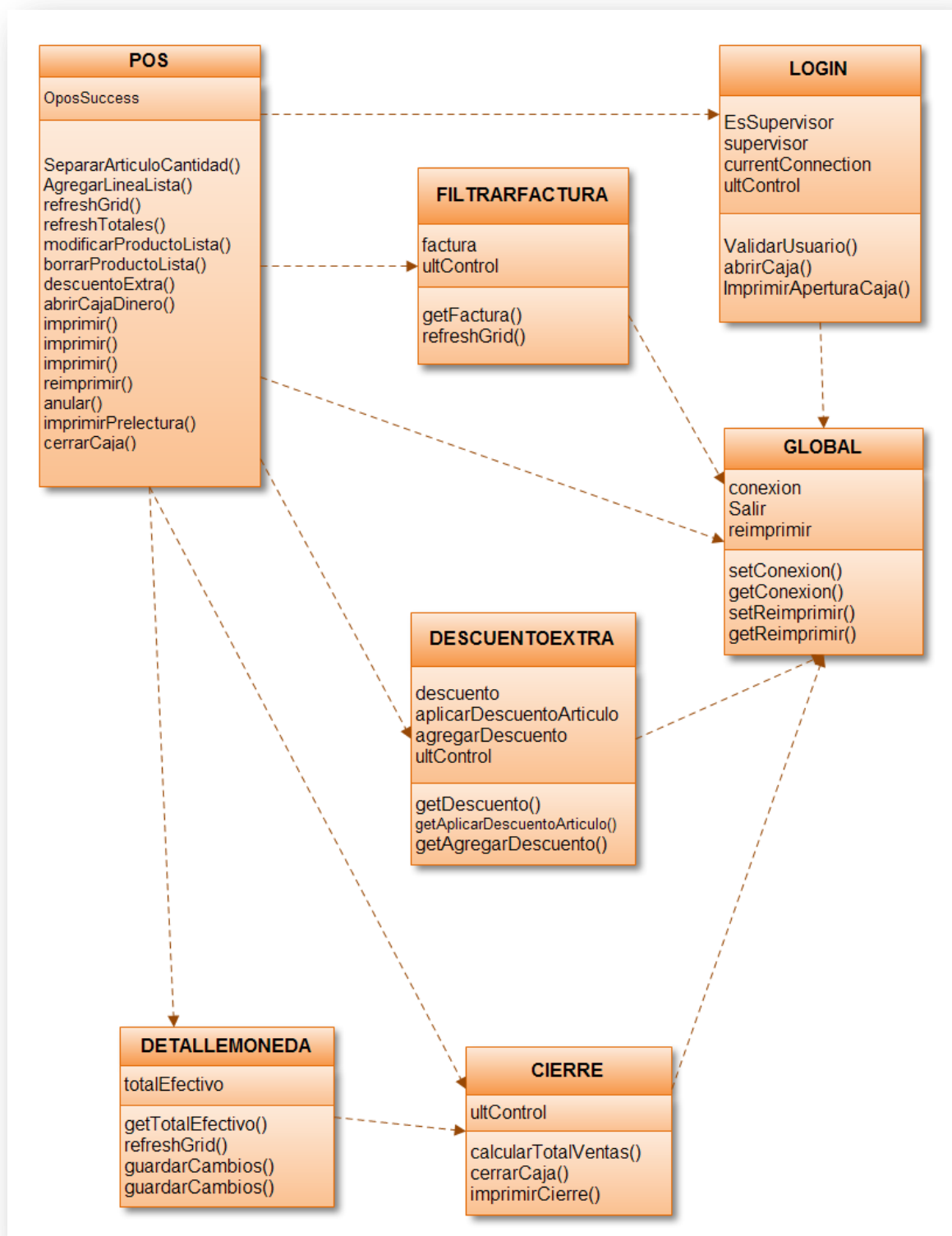


Figura 7.DIAGRAMA DE CLASES PARTE 2.



13. DIAGRAMAS DE SECUENCIA.

En los siguientes diagramas se muestra cómo se realizan cada una de las funciones de la aplicación, las cuales puedes realizar el Cajero y/o el Supervisor.

Figura 8. Diagrama de Inicio de sesión.

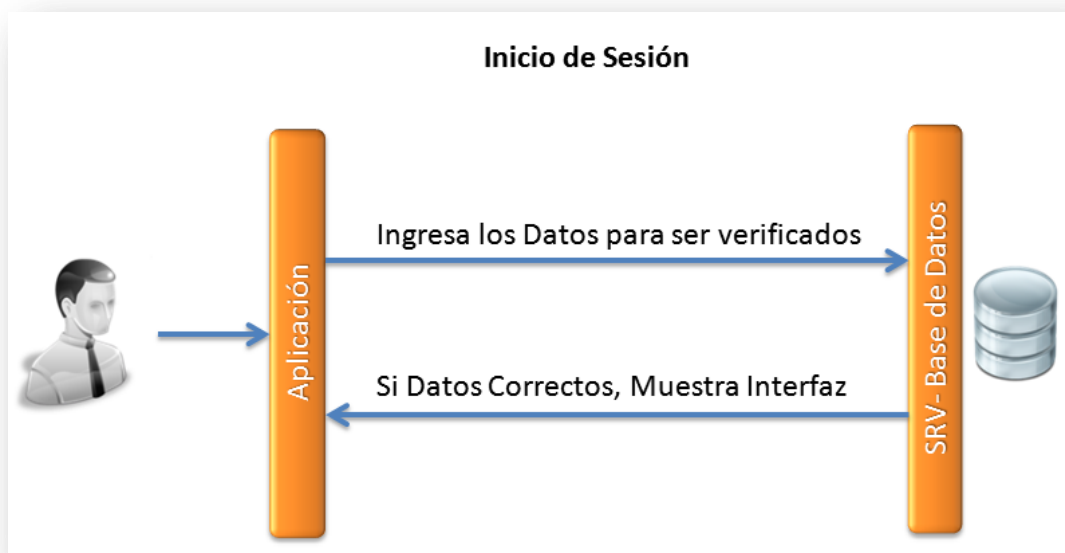


Figura 9. Diagrama de Agregación de productos a una factura.

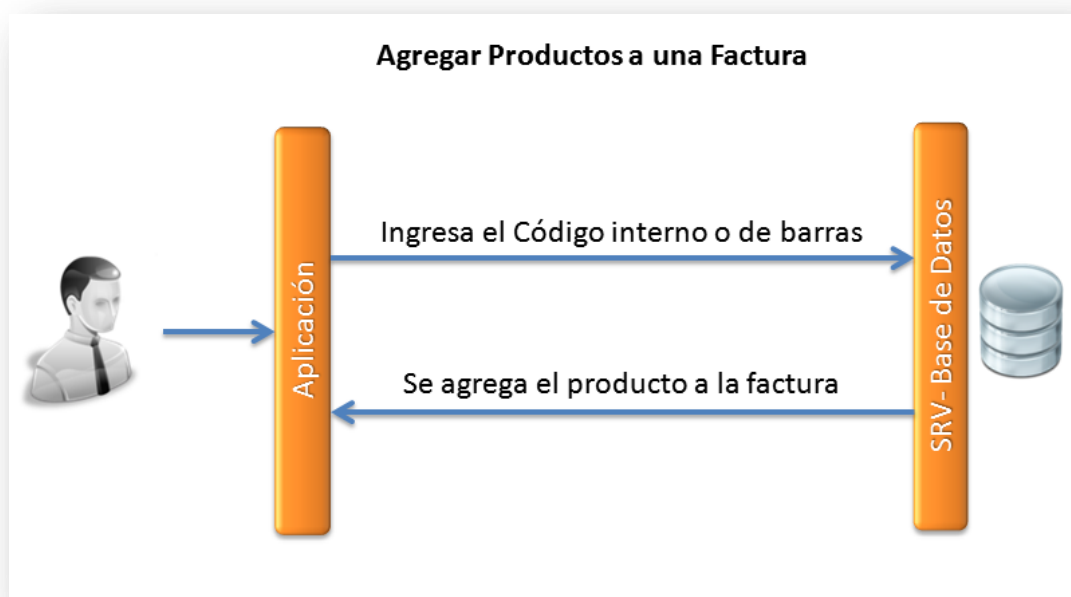


Figura 10. Diagrama de filtrado de producto.



Figura 11. Diagrama de editar línea de la factura.

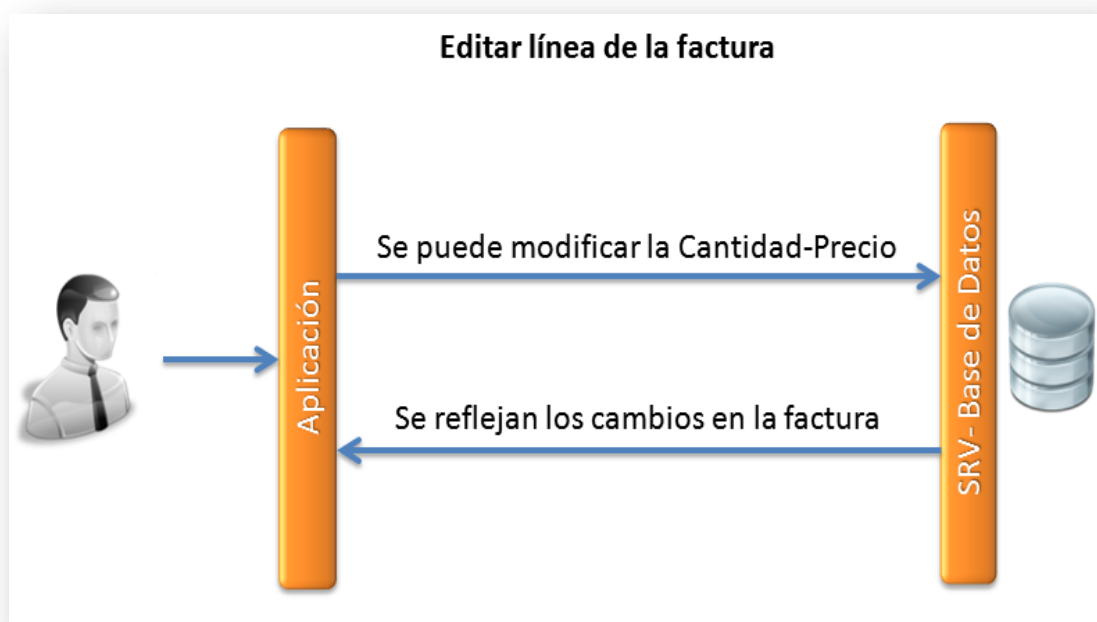


Figura 12. Diagrama de eliminación línea de la factura.

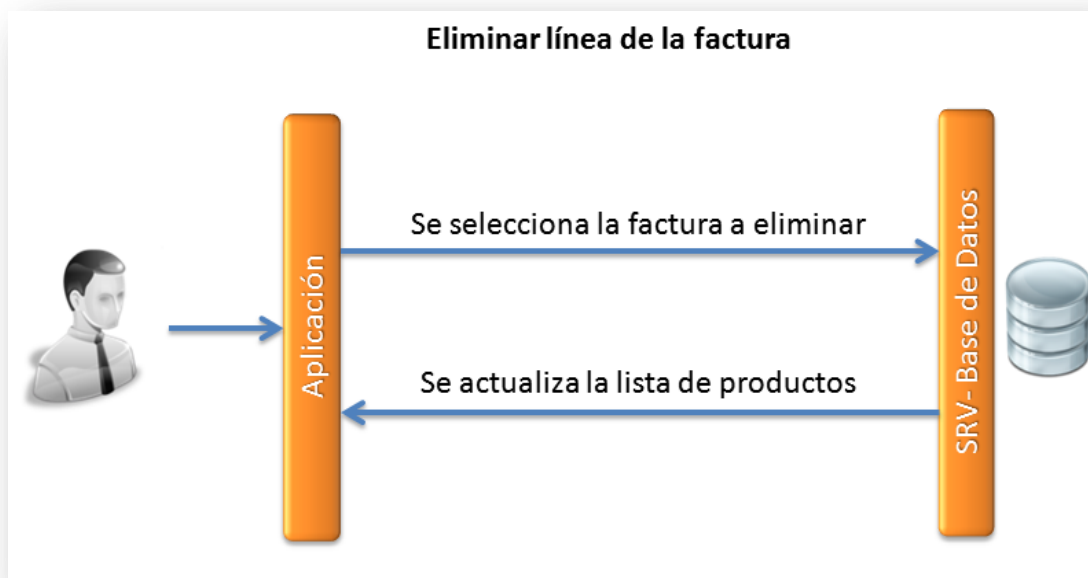


Figura 13. Diagrama de filtrar cliente.



Figura 14. Diagrama de anular factura.

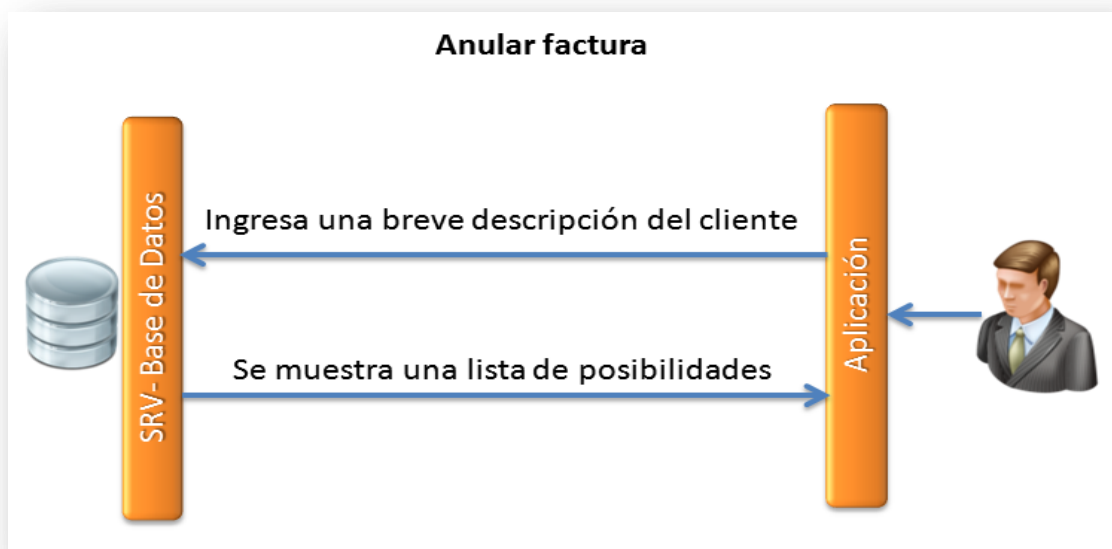


Figura 15. Diagrama establecer descuento extra.

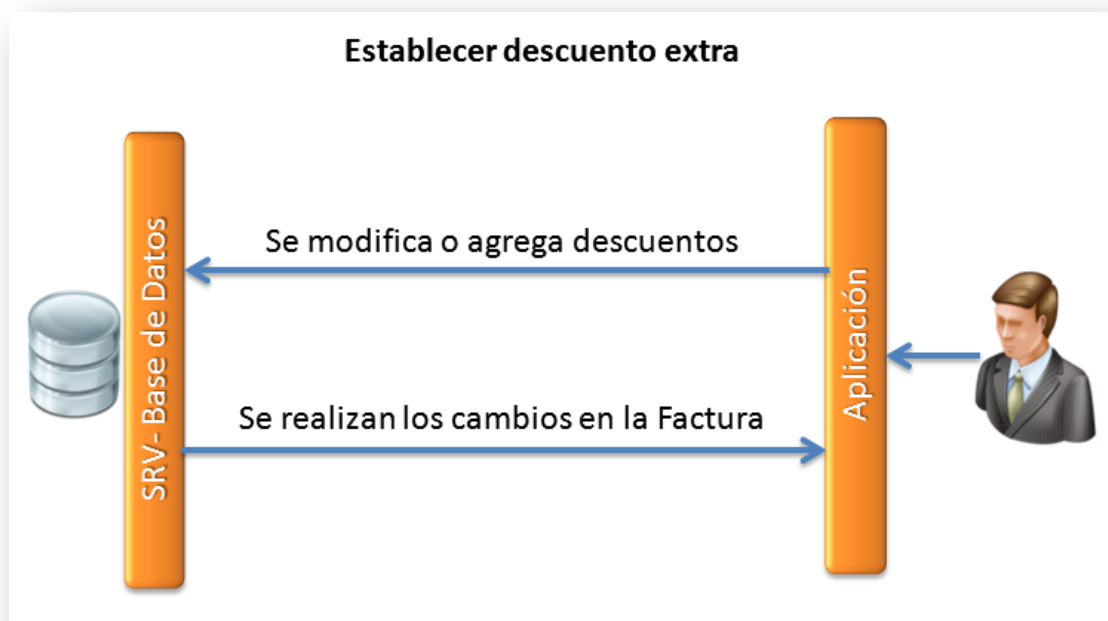


Figura 16. Diagrama de Pre Lectura de una factura.

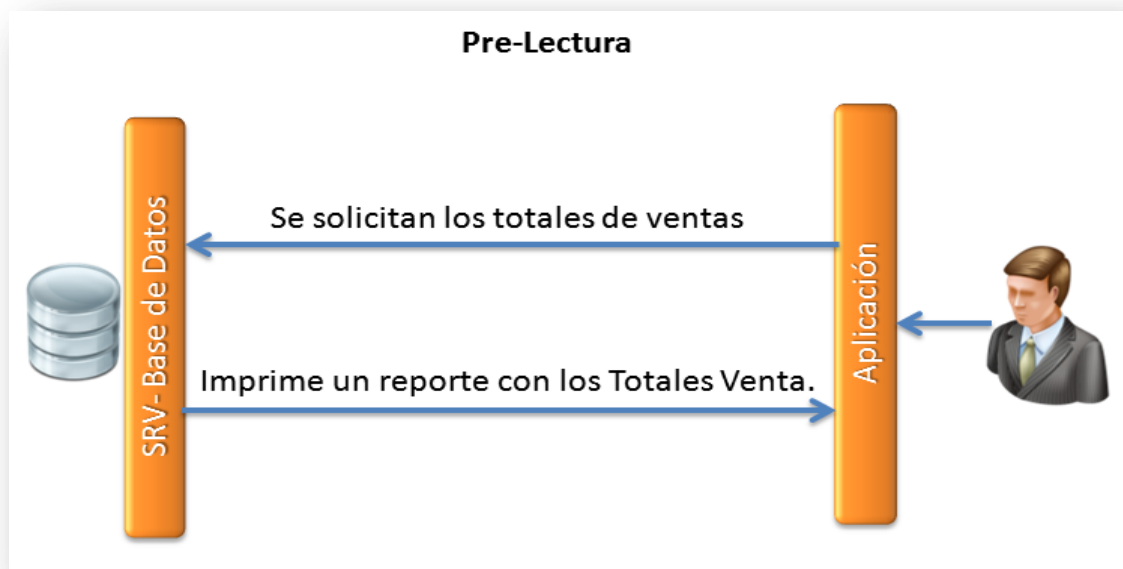
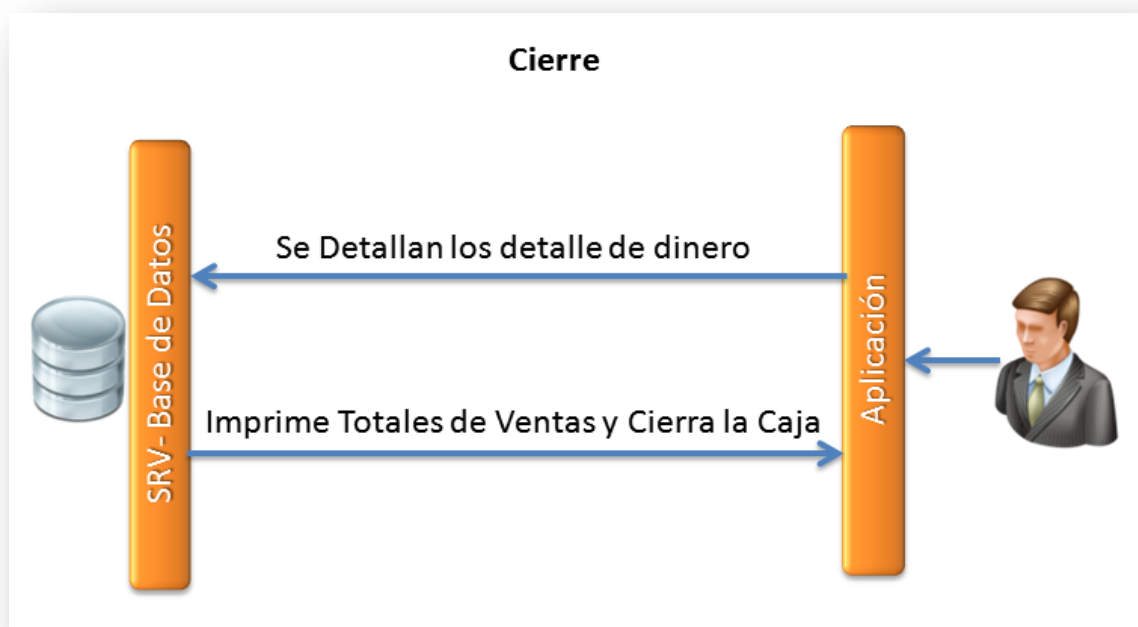


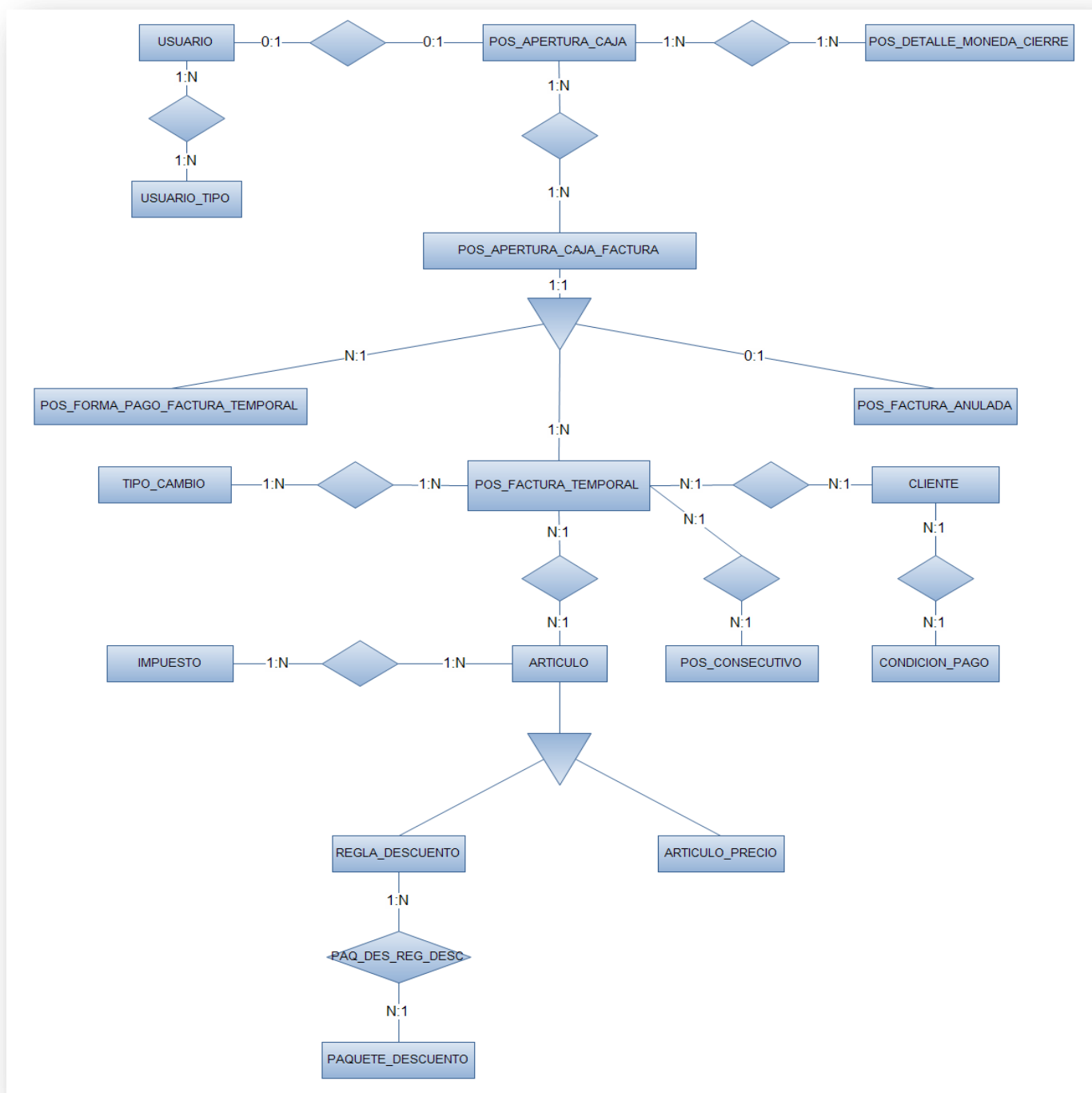
Figura 17. Diagrama de cierre de caja.



14. DIAGRAMA RELACIONAL

Figura 18. DIAGRAMA E-R

BASE DE DATOS: POS



15. DISEÑO DE INTERFAZ

Figura 19. Interfaz gráfico de Inicio de Sesión.

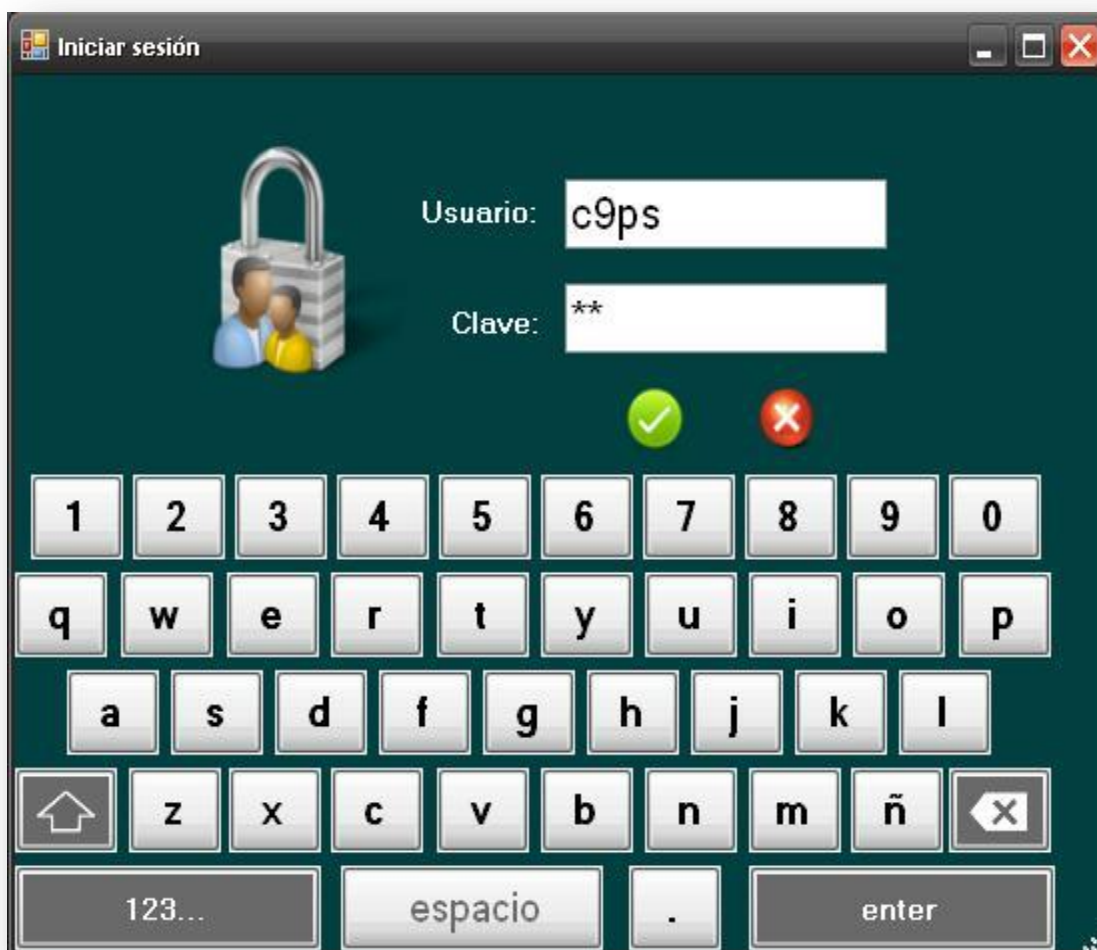
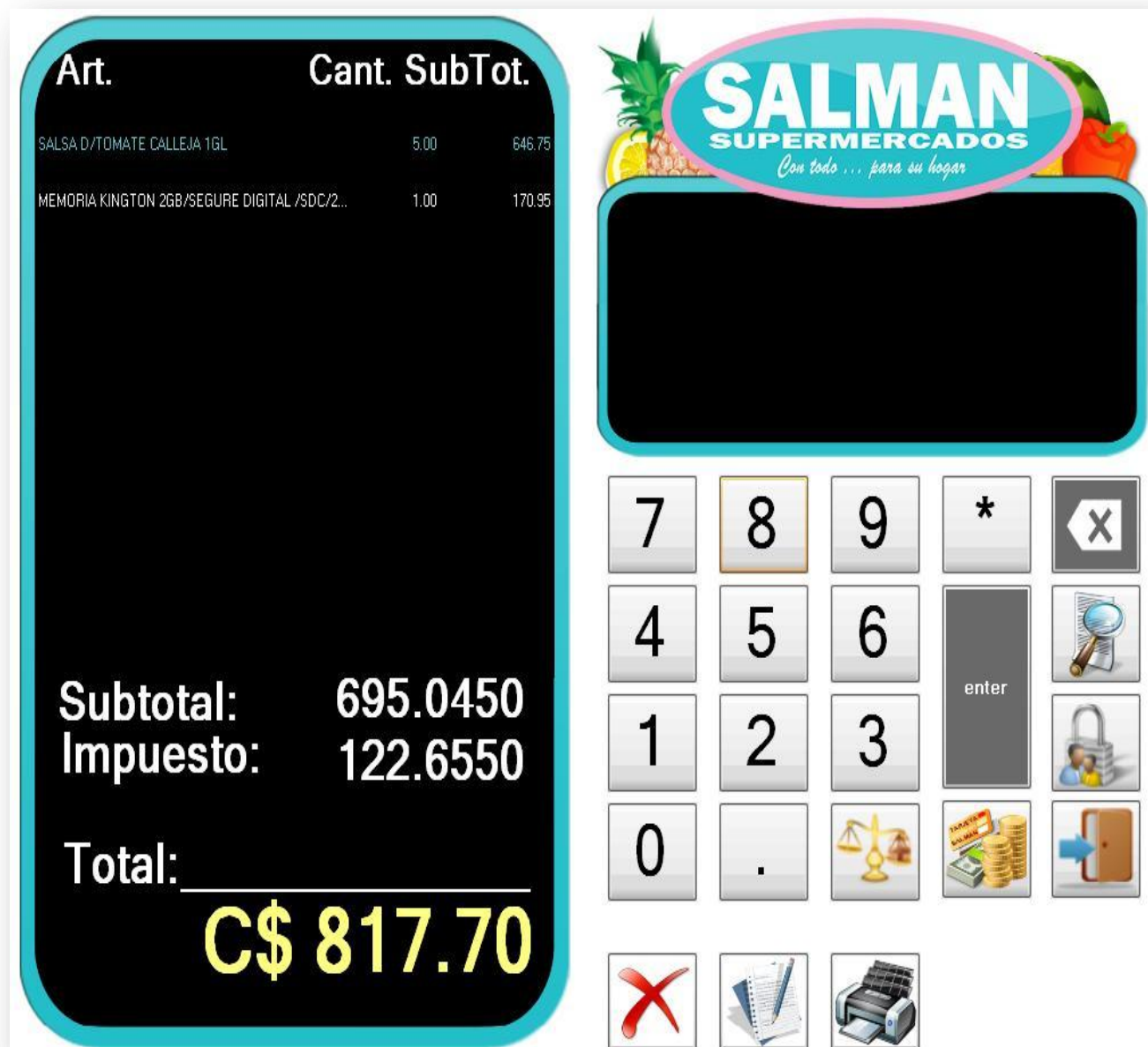


Figura 20. Interfaz gráfico Pantalla Principal.



Art.	Cant.	SubTot.
SALSA D/TOMATE CALLEJA 16L	5.00	646.75
MEMORIA KINGTON 2GB/SEQUIRE DIGITAL /SDC/2...	1.00	170.95

Subtotal:	695.0450
Impuesto:	122.6550
Total:	C\$ 817.70

SALMAN
SUPERMERCADOS
Con todo ... para su hogar

7 8 9 * [X]

4 5 6 [enter] [Search]

1 2 3 [Lock]

0 . [Scales] [Coins] [Exit]

[Cancel] [Receipt] [Printer]

Figura 21. Interfaz gráfica de cobro de factura

Subtotal: 695.0450

Impuesto: 122.6550

Total:
C\$ 817.70

C C000 CLIENTE DE CONTADO

CRÉDITO

CHEQUE

PROMOCIÓN

Forma de Pago	Monto
▶ EFECTIVO	500
TARJETA	317.70

Cerrar

Saldo: 0.00

7

8

9

⬅ x

4

5

6

enter

1

2

3

0

.

Figura 22. Interfaz gráfico de Filtro de Productos.

Descripción:

Top:

DESCRIPCION	PRECIO
LECHE E/POLVO PROGRESS GOLD 3 400GR 1/12	217.60
LECHE E/POLVO PROMIL GOLD 2 400GR 1/12	237.55
LECHE E/POLVO ANCHOR BOLSA 120 GR 1/48	30.05
LECHE E/POLVO ANCHOR BOLSA 360 GR 1/24	87.20
LECHE E/POLVO ANCHOR POTE 1600 GR 1/6	389.85
LECHE E/POLVO ANCHOR POTE 2200 GR 1/6	515.15
LECHE E/POLVO ANCHOR POTE 800 GR 1/12	201.45
LECHE E/POLVO DELUSOYA CREMA Y GALLETTAS 360 GR 1/42	60.85
LECHE E/POLVO DELUSOYA FRESA 360 GR 1/42	60.85
LECHE E/POLVO DELUSOYA LITRO 120 GR 1/84	19.15
LECHE E/POLVO DELUSOYA NATURAL 1/12 800GR	138.60
LECHE E/POLVO DELUSOYA NATURAL 360 GR BOLSA 1/42	70.80

Figura 23. Interfaz grafico de Filtro de Cliente.

Nombre cliente o beneficiario:

EMPRESA	CUENTE	NOMBRE_CLIENTE	BENEFICIARIO	ACTIVO
SUPERMERCA...	0066	KARLA PATRICIA MORALES MARADIAGA	DAVID ISAAC MORALES MARADIAGA	S
UNAN-LEON	1023	SERGIO DAVID ESPINOZA BALDELOMAR	SERGIO DAVID ESPINOZA BALDELOMAR	S
TRANSACTEL	1355	MARIO DAVID BONILLA NARVAEZ	MARIO DAVID BONILLA NARVAEZ	S
TRANSACTEL	1547	DYLAN DAVID OTERO AVENDAÑO (INACTIVADO)	DYLAN DAVID OTERO AVENDAÑO	N
COLEGIO BAUT...	1739	JOSE DAVID CORONADO	JOSE DAVID CORONADO	N
PARTICULARES	2824	DAVID ANTONIO ALVAREZ Q.	DAVID ANTONIO ALVAREZ Q.	N
UNAN-LEON	3502	ALISTAIR DAVID THIRKETTLE	ALISTAIR DAVID THIRKETTLE	N
I.N.S.S. - LEON	3863	ELVIA TELLEZ CENTENO	BIARNI DAVID SALINAS TELLEZ	S
UNAN-LEON	4092	OSCAR DAVID REAL QUINTANA	OSCAR DAVID REAL QUINTANA	N
TRIB. DE APEL....	4340	JOSE DAVID GONZALEZ	JOSE DAVID GONZALEZ	N
UNAN-LEON	5220	LESTER DAVID RIVERA PEREZ	LESTER DAVID RIVERA PEREZ	S
SUPERMERCA...	5913	LUIS DAVID AREAS LOPEZ	LUIS DAVID AREAS LOPEZ	N
SUPERMERCA...	5971	DAVID ALEJANDRO FLORES CHAVARRIA	KAREN DE LOS ANGELES LOPEZ PINEDA	N
SUPERMERCA...	6067	DAVID RAFAEL SANCHEZ PENAS	DAVID RAFAEL SANCHEZ PENAS	N

Figura 24. Interfaz gráfico de Edición de una línea de la Factura.

The dialog box has a title bar 'Modificar cantidad / precio'. The main content area displays 'SALSA D/TOMATE CALLEJA 1GL'. Below this, there are two input fields: 'Cantidad:' with the value '5.00' and 'Precio Total:' with the value '646.75'. To the right of these fields is a numeric keypad with buttons for digits 0-9, a decimal point, and a backspace button (arrow with 'X'). Below the keypad are two buttons: 'Aceptar' and 'Cancelar'.

Figura 25. Interfaz grafico de Aplicar Descuento Extra.

The dialog box has a title bar 'Descuento Extra'. It contains a 'Descuento(%)' input field. Below it is a section 'Aplicar a...' with two radio buttons: 'Artículo seleccionado' (selected) and 'Factura completa'. Below that is a section 'Modo descuento...' with two radio buttons: 'Agregar descuento' (selected) and 'Reemplazar descuento'. To the right of these sections is a numeric keypad with buttons for digits 0-9, a decimal point, a backspace button (arrow with 'X'), and an 'enter' button. At the bottom are two buttons: 'Aceptar' and 'Cancelar'.

Figura 26. Interfaz gráfico del Menú Principal del Supervisor.



Figura 27. Interfaz gráfico Detallando el Cierre de la Caja.

MONEDA	TIPO	DENOMINACION	CANTIDAD
COR	M	0.05	0
COR	M	0.10	0
COR	M	0.25	0
COR	M	0.50	0
COR	M	1.00	0
COR	M	5.00	0
COR	M	10.00	0
COR	B	10.00	0
COR	B	20.00	0
COR	B	50.00	0
COR	B	100.00	0
COR	B	200.00	0
COR	B	500.00	0
DOL	M	0.05	0
DOL	M	0.10	0
DOL	M	0.25	0
DOL	M	0.50	0
DOL	M	1.00	0
DOL	M	5.00	0
DOL	M	10.00	0
DOL	B	10.00	0
DOL	B	20.00	0
DOL	B	50.00	0
DOL	B	100.00	0
DOL	B	200.00	0
DOL	B	500.00	0

Córdobas:

Dólares:

Total:

7	8	9	*
4	5	6	←
1	2	3	enter
0	.		

Figura 28. Interfaz gráfica Detallando el Cierre de la Caja.

The interface displays the following data and controls:

Efectivo:	259202.6	...
Tarjetas BANPRO:	40000	Total Tarjetas:
Tarjetas CREDOMATIC:	0.00	40,000.00
Crédito:	500	
Cheque:	0.00	
Promoción:	0.00	
Total ventas:	299702.6	

Navigation and Input Controls:

- Home icon (house symbol)
- abc... button
- Numeric keypad: 7, 8, 9, *, 4, 5, 6, 1, 2, 3, 0, .
- Backspace icon (arrow and X)
- enter button

Summary and Action:

Esto es una ejemplo del reporte del Cierre...

Aceptar **Cancelar**

Figura 29. Reporte donde se muestra impreso la apertura de la caja.



APERTURA DE CAJA

BODEGA: 009
CAJA: CAJA1
USUARIO: clps
No. APERTURA: 875
FECHA HORA: 13/09/2012 09:49:16 P.M.

Figura 30. Reporte donde se muestra impreso la factura de crédito.



COSTADO NORTE COL. LA SALLE, LEÓN
RUC: J03100000565 TELÉF: (505) 2311-5027
AUT. DGE/ASFC/012/#1/07/2009

FACTURA FSAL130912-100000004
CONDICIÓN Credito a 15 días
CAJERO FRANCISCO
FECHA 21/10/2012 04:34:02 P.M.
CLIENTE 6096 DAVID LEONARDO RIVAS VARGAS
 E89 Farmacia La Esperanza(Personal)

77629	MEMORIA KINGTON 2GB/SEGUIRE	1.00	170.95
64940	PASTE P/LAVAR TRASTE	1.00	1.75
11111	SALSA D/TOMATE CALLEJA 1GL	1.00	129.35
00010	COLADO GERBER PURE PERA /FR	1.00	22.95
IMPUESTOS:			48.75
TOTAL C\$:			325.00
CREDITOS:			325.00
CAMBIO C\$:			0.00

Figura 31. Reporte donde se muestra impresa la Factura con pago Mixto.



COSTADO NORTE COL. LA SALLE, LEÓN
RUC: J03100000565 TELÉF: (505) 2311-5027
AUT. DGE/ASFC/012/#1/07/2009

FACTURA	FSAL130912-100000005
CONDICIÓN	CONTADO
CAJERO	FRANCISCO
FECHA	21/10/2012 04:40:34P.M.
CLIENTE	C000 CLIENTE DE CONTADO

90980 TELEVISOR SONY PANTALLA LCD	1.00	55,986.55
IMPUESTOS:		8,397.98
TOTAL CS:		55,986.55
TARJETAS:		35,986.55
EFFECTIVOS:		20,000.00
CAMBIO CS:		0.00

Figura 32. Reporte donde se muestra impresa la Pre Lectura de la caja.



PRELECTURA DE CAJA

BODEGA:	009
CAJA:	CAJA1
USUARIO:	CIPS
No. APERTURA:	875
FECHA HORA:	13/09/2012 09:49:16P.M.

CREDITO:	325.00
EFFECTIVO:	20,504.15
TARJETA:	35,986.55

DETALLE CREDITO

Farmacia La Esperanza(Personal)	325.00
Saldo Créditos:	325.00

Figura 33. Reporte donde se muestra impreso los datos de cierre de una caja.



CIERRE DE CAJA

BODEGA	009
CAJA	CAJA1
USUARIO	CIPS
No. APERTURA	875
FECHA HORA APERTURA	13/09/2012 09:49:16P.M.
FECHA HORA CIERRE	21/10/2012 05:04:18P.M.
No. CLIENTES	5
TIPO CAMBIO	23.40

	SISTEMA	CAJERO
EFFECTIVO	20,504.15	259,202.60
	DIF.	238,698.45
TARJETA	35,986.55	40,000.00
	DIF.	4,013.45
BANPRO: 40000 CREDOMATIC: 0.00		
CREDITO	325.00	500.00
	DIF.	175.00
CHEQUE	0.00	0.00
	DIF.	0.00
PROMOCION	0.00	0.00
	DIF.	0.00
DIFERENCIA 242,886.90		
TOTAL VENTA 56,815.70		
NOTA		
Esto es una ejemplo del reporte del Cierre...		
DETALLE CREDITO		
Farmacia La Esperanza(Personal)		325.00
Saldo Créditos:		325.00

16. CONCLUSIONES

Al finalizar nuestro trabajo monográfico, hemos podido lograr los objetivos que nos habíamos planteados inicialmente, realizando de esta manera un sistema tipo TPV innovador que cumple los requerimientos del Supermercado Salman.

Con la elaboración de este sistema se ayudará al Supermercado Salman a realizar un manejo ágil, fiable y con mayor control de la información de los clientes, las ventas e inventarios, generando información comercial en el momento que se considere necesario.

Al realizar nuestro sistema utilizamos Visual Studio .NET 2008 y llegamos a la conclusión que es una plataforma de mucho potencial y que incluye herramientas que permiten al programador crear aplicaciones de muy buena calidad trabajando en un entorno fácil, confiable y seguro para la elaboración de aplicaciones.

17. RECOMENDACIONES

1. Todo personal contratado para ejercer el cargo de cajero y supervisor deben ser previamente entrenados para el buen uso del sistema T.P.V.
2. Se debe asignar un nombre de usuario con su respectiva clave a cada cajero.
3. Los ordenadores donde se instalen el sistema T.P.V. deben cumplir con los requerimientos anteriormente expuestos.
4. Energía
 - a. Se debe instalar un sistema de respaldo energético que sea capaz de brindar energía estable en un promedio de entre 5 a 8 horas continuas. Recomendamos motores impulsados a gas con encendido y transfer automático.
 - b. El cuarto de comunicaciones debe de estar protegido con un sistema de respaldo online. Esto es importante para las fluctuaciones que se dan constantemente del sistema energético local.
 - c. Cada T.P.V. debe tener batería de 750 VA de buena calidad que permita al menos 10 minutos de energía de respaldo.
5. Respallos de las Bases de Datos:
 - a. Los servidores deben de contar con discos espejos que sirvan como respaldo inmediato de la base de datos.
 - b. Respallos Automáticos: Se debe programar en los servidores el respaldo de las bases de datos diario. Esta tarea deberá ejecutarse al finalizar la jornada laboral.
 - c. Se debe de guardar un respaldo en un dispositivo externo o portable el cual deberá de mantenerse en otra ubicación geográfica al cuarto de servidores.
6. Programar mantenimiento a las bases de datos por lo menos una vez por semana.
7. Programar el mantenimiento de los equipos por lo menos una vez al mes.
8. Mejoras del sistema T.P.V.:
 - a. Realizar un sistema de pedido para agilizar la atención al cliente que realizan compras en las tiendas.
 - b. Permitir que el sistema pueda soportar diferenciación de precios para los diferentes niveles de expresiones del supermercado salman.

18. BIBLIOGRAFÍA

- **Visual C#. Interfaces gráficas y aplicaciones, Francisco Javier Ceballos Sierra**

Referencias:

<http://es.wikipedia.org/wiki/TPV>

<http://www.iconfinder.com/>

[http://msdn.microsoft.com/es-es/library/1dbsh6t3\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/1dbsh6t3(v=vs.90).aspx)

<http://www.aulacltic.es/sqlserver/index.htm>

19. ANEXOS

1. Diccionario de Datos

SISTEMA: Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto.

TPV: Es un sistema que ayuda en las tareas de gestión de un negocio, es un punto de venta que normalmente cuentan con sistemas informáticos especializados y que facilita las tareas de venta, gestión de inventario, gestión de productos, etc.

Hardware: Son todos los dispositivos y componentes físicos que realizan las tareas de entrada y salida, también se conoce al hardware como la parte dura o física del computador. La mayoría de las computadoras están organizadas de la siguiente forma: Los dispositivos de entrada como teclados, Lectores de Tarjetas, Lápices Ópticos, Lectores de Códigos de Barra, Escáner, Mouse, etc.) y salida (Monitor, Impresoras, Plotters, Parlantes, etc.).

Cajas Registradoras: Es un aparato mecánico o electrónico que permite calcular y registrar transacciones comerciales, e incluye un cajón para guardar dinero. Por lo general, permite además imprimir un recibo o factura para el cliente, según las disposiciones legales de cada País.

Stock: bienes poseídos por una empresa para su venta en el curso ordinario de la explotación, o bien para su transformación o incorporación al proceso productivo.

Exactus: Es un sistema que trabaja de forma modular y permite manejar la producción, logística, distribución, inventario, envíos, facturas y contabilidad de la compañía.

Sofland: Empresa desarrolladora de aplicaciones robustas

Pantallas Táctiles: Es una pantalla que mediante un toque directo sobre su superficie permite la entrada de datos y órdenes al dispositivo, y a su vez muestra los resultados introducidos previamente; actuando como periférico de entrada y periférico de salida de datos.

Clientes: Es un equipo o proceso que consume recursos y servicios brindados por otro llamado servidor, generalmente de forma remota. Es quien accede a un producto o servicio por medio de una transacción financiera (dinero) u otro medio de pago. Quien compra, es el comprador, y quien consume el consumidor. Normalmente, cliente, comprador y consumidor son la misma persona.

Codificación: Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente.

Slip-printer: Impresoras de facturas o tickets de cobro.

Cajón de Dinero: Máquina que permite sacar o meter dinero, en cualquier momento mediante una tarjeta especial que tiene asignada una clave personal.

PuertoRJ11: Conector 45 registrado. Es un conector de forma especial con 8 terminales, que se utilizan para interconectar computadoras y generar redes de datos de área local (LAN - red de computadoras cercanas interconectadas entre sí).

Lector de código de barras: Dispositivo que interpreta los símbolos del código de barras que el fabricante imprime en la etiqueta de los productos.

LCD: Dispositivo o display de periférico de salida de datos de cristal líquido.

VDF: Es un dispositivo de ciertos aparatos electrónicos que permite mostrar información al usuario de manera visual.

TFT: Las siglas TFT significan ("**Thin Film Transistor**")ó transistor de película delgada. Se trata de una tecnología basada en transistores de efecto de campo, esto es, se coloca sobre una placa de cristal un electrodo (laminita que conduce electricidad), sobre la cual se colocan capas delgadas, y al activarse por medio del electrodo cada una, se van activando los colores, formándose de esta forma cada píxel. Este tipo de pantallas se utilizan de manera común las pantallas de computadoras portátiles (Laptop, Netbook), colectoras de datos, etc.

Visual Estudio: Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows.

C#:C# (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Framework: Es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos.

CrystalReports: Es una de Business Intelligence aplicación utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos.

ODBC:(Open DatabaseConnectivity) es un estándar de acceso a bases de datos, que permite mantener independencia entre los lenguajes de programación, los sistemas de bases de datos (las bases de datos y su software gestor), y los sistemas operativos.

Microsoft SQL Server: Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

T-SQL: Es una extensión al SQL de Microsoft y Sybase. SQL, que frecuentemente se dice ser un Lenguaje de Búsquedas Estructurado (por sus siglas en inglés), es un lenguaje de cómputo estandarizado, desarrollado originalmente por IBM para realizar búsquedas, alterar y definir bases de datos relacionales utilizando estatutos declarativos. T-SQL expande el estándar de SQL para incluir programación procedural, variables locales, varias funciones de soporte para procesamiento de strings, procesamiento de fechas, matemáticas, etc, y cambios a los estatutos DELETE y UPDATE. Estas características adicionales hacen de T-SQL un lenguaje

DDL: Un **lenguaje de definición de datos** (**Data Definition Language**, DDL por sus siglas en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

DML: Lenguaje artificial de cierta complejidad que permite el manejo y procesamiento del contenido de la base de datos.

Plugins: La palabra plugin (del inglés plug-in) hace referencia a pequeñas aplicaciones funcionales creadas para ser añadidas a programas más grandes con el propósito de ampliar su capacidad en alguna actividad. El concepto de plugin implica que el objeto se "enchufa" en un programa y puede ser utilizado.

POS Salman: aplicación de terminal de punto de venta.

Factura de venta: Las facturas de venta son un **instrumento** que sirve como constancia para el vendedor y para el comprador de la operación realizada. Describe en ella lo que se ha comprado y por ende vendido, y el precio pagado. En este caso el hecho descrito es la operación de compraventa.

Crédito: El crédito es una operación financiera en la que se pone a nuestra disposición una cantidad de dinero hasta un límite especificado y durante un período de tiempo determinado.

Windows Installer: Es un motor para la instalación, mantenimiento y eliminación de programas en plataformas Microsoft Windows.

Cajeros: Es una persona responsable de sumar la cantidad debida por una compra, cargar al consumidor esa cantidad y después, recoger el pago por las mercancías o servicios proporcionados.

Supervisor: Persona encargada de velar por el trabajo en una determinada empresa.

Tarjeta debito: Es una tarjeta bancaria de plástico de 8.5 × 5.3 cm con una banda magnética en el reverso (y actualmente también se incluye un chip electrónico) que guarda información sobre los datos de acceso, el nombre y número de cuenta del titular, usada para poder efectuar con ella operaciones financieras activas (incrementar el saldo), pasivas (disminuir el saldo) o neutrales (no incrementan ni disminuyen el saldo disponible).

Tarjeta Crédito: es un instrumento material de identificación del usuario, que puede ser una tarjeta plástica con una banda magnética, un microchip y un número en relieve. Es emitida por un banco o entidad financiera que autoriza a la persona a cuyo favor es emitida, utilizarla como medio de pago en los negocios adheridos al sistema, mediante su firma y la exhibición de la tarjeta. Es otra modalidad de financiación, por lo tanto, el usuario supone asumir la obligación de devolver el importe dispuesto y de pagar los intereses, comisiones bancarias y gastos pactados.

Productos: Se denomina producto a lo obtenido de la venta o alquiler cosa o servicios, es cualquier cosa que se puede ofrecer a un mercado para satisfacer un deseo o una necesidad.

Descuentos: Disminución o reducción que se hace en una cantidad o en un precio:

Pago Mixto: Es cuando una factura es cancelada de con más de una forma de pago.

20. CÓDIGO FUENTE

Clase para gestionar las conexiones, que permite la manipulación y obtención de datos almacenados en el servidor de Base de Datos.

```
public class Conexion
{
    private string Server;
    private string Username;
    private string Password;
    private string Database;
    private string Company;
    private string ConnectionString;

    private SqlConnection SQLConnection;
    private SqlDataAdapter SQLDataAdapter;
    public DataTable dtResult;

    public string GetConnectionString(){return ConnectionString;}

    public Conexion()
    {
        Server="";
        Username="";
        Password="";
        Database="";
        Company="";
        ConnectionString="";
    }

    public Conexion(string server,string user,string pass,string db,string company)
    {
        Server=server;
        Username=user;
        Password=pass;
        Database=db;
        Company=company;

        ConnectionString="Data Source="+Server+";";
        ConnectionString+="User ID="+Username+";";
        ConnectionString+="Password="+Password+";";
        ConnectionString+="Initial Catalog="+Database;
    }

    public Conexion(Conexion con)
    {
        Server=con.Server;
        Username=con.Username;
        Password=con.Password;
        Database=con.Database;
    }
}
```

```
Company=con.Company;
ConnectionString=con.ConnectionString;
}

public bool Conectar()
{
    try
    {
        SqlConnection=new SqlConnection();
        SqlConnection.ConnectionString=ConnectionString;
        SqlConnection.Open();
        return true;
    }
    catch (Exception)
    {
        // Try to close the connection
        if (SqlConnection != null)
            SqlConnection.Dispose();

        // Show error message (this = the parent Form object)
        frmConectandoConectando=new frmConectando();
        conectando.ShowDialog();

        if (conectando.getReintentar())
            Conectar();

        // Stop here
        return false;
    }
}

public void runSQLStatement(String SQLStatement)
{
    if (!Conectar()) return;

    // Create a SqlDataAdapter to get the results as DataTable
    SqlDataAdapter=new SqlDataAdapter(SQLStatement, SqlConnection);

    // Create a new DataTable
    dtResult=new DataTable();

    // Fill the DataTable with the result of the SQL statement
    SqlDataAdapter.Fill(dtResult);
}

~Conexion()
{
    try
    {
        SqlConnection.Close();
        SqlConnection.Dispose();
        SqlDataAdapter.Dispose();
    }
}
```

```
catch(Exception)
{
    Return;
}
}
```

20.1. Manejo de teclado Touch

```
/****** CODIGO PARA TECLADO TOUCH *****/

private List<Button>keyCollection=new List<Button>();

private IntPtr ctrl;

private void cmdMayusculas_Click(object sender, EventArgs)
{
    if(cmdA.Text=="a")
    {
        cmdQ.Text="Q";
        cmdW.Text="W";
        cmdE.Text="E";
        cmdR.Text="R";
        cmdT.Text="T";
        cmdY.Text="Y";
        cmdU.Text="U";
        cmdI.Text="I";
        cmdO.Text="O";
        cmdP.Text="P";
        cmdA.Text="A";
        cmdS.Text="S";
        cmdD.Text="D";
        cmdF.Text="F";
        cmdG.Text="G";
        cmdH.Text="H";
        cmdJ.Text="J";
        cmdK.Text="K";
        cmdL.Text="L";
        cmdEnye.Text="Ñ";
        cmdZ.Text="Z";
        cmdX.Text="X";
        cmdC.Text="C";
        cmdV.Text="V";
        cmdB.Text="B";
        cmdN.Text="N";
        cmdM.Text="M";
        cmdQ.Tag="Q";
        cmdW.Tag="W";
        cmdE.Tag="E";
        cmdR.Tag="R";
        cmdT.Tag="T";
    }
}
```

```
cmdY.Tag="Y";
cmdU.Tag="U";
cmdI.Tag="I";
cmdO.Tag="O";
cmdP.Tag="P";
cmdA.Tag="A";
cmdS.Tag="S";
cmdD.Tag="D";
cmdF.Tag="F";
cmdG.Tag="G";
cmdH.Tag="H";
cmdJ.Tag="J";
cmdK.Tag="K";
cmdL.Tag="L";
cmdEnye.Tag="Ñ";
cmdZ.Tag="Z";
cmdX.Tag="X";
cmdC.Tag="C";
cmdV.Tag="V";
cmdB.Tag="B";
cmdN.Tag="N";
cmdM.Tag="M";
}
else
{
cmdQ.Text="q";
cmdW.Text="w";
cmdE.Text="e";
cmdR.Text="r";
cmdT.Text="t";
cmdY.Text="y";
cmdU.Text="u";
cmdI.Text="i";
cmdO.Text="o";
cmdP.Text="p";
cmdA.Text="a";
cmdS.Text="s";
cmdD.Text="d";
cmdF.Text="f";
cmdG.Text="g";
cmdH.Text="h";
cmdJ.Text="j";
cmdK.Text="k";
cmdL.Text="l";
cmdEnye.Text="ñ";
cmdZ.Text="z";
cmdX.Text="x";
cmdC.Text="c";
cmdV.Text="v";
cmdB.Text="b";
cmdN.Text="n";
cmdM.Text="m";
```

```
cmdQ.Tag="q";  
cmdW.Tag="w";  
cmdE.Tag="e";  
cmdR.Tag="r";  
cmdT.Tag="t";  
cmdY.Tag="y";  
cmdU.Tag="u";  
cmdI.Tag="i";  
cmdO.Tag="o";  
cmdP.Tag="p";  
cmdA.Tag="a";  
cmdS.Tag="s";  
cmdD.Tag="d";  
cmdF.Tag="f";  
cmdG.Tag="g";  
cmdH.Tag="h";  
cmdJ.Tag="j";  
cmdK.Tag="k";  
cmdL.Tag="l";  
cmdEnye.Tag="ñ";  
cmdZ.Tag="z";  
cmdX.Tag="x";  
cmdC.Tag="c";  
cmdV.Tag="v";  
cmdB.Tag="b";  
cmdN.Tag="n";  
cmdM.Tag="m";  
}  
}
```

```
private void cmdNumeros_Click(object sender, EventArgs e)  
{  
    if (cmd0.Visible)  
        cmdNumeros.Text="123...";  
    else  
        cmdNumeros.Text="abc...";  
        button1.Visible=!button1.Visible;  
        button2.Visible=!button2.Visible;  
        button3.Visible=!button3.Visible;  
        button4.Visible=!button4.Visible;  
        button5.Visible=!button5.Visible;  
        button6.Visible=!button6.Visible;  
        button7.Visible=!button7.Visible;  
        button8.Visible=!button8.Visible;  
        button9.Visible=!button9.Visible;  
        button10.Visible=!button10.Visible;  
        cmd0.Visible=!cmd0.Visible;  
        cmd1.Visible=!cmd1.Visible;  
        cmd2.Visible=!cmd2.Visible;  
        cmd3.Visible=!cmd3.Visible;  
        cmd4.Visible=!cmd4.Visible;  
        cmd5.Visible=!cmd5.Visible;  
        cmd6.Visible=!cmd6.Visible;
```

```
cmd7.Visible=!cmd7.Visible;
cmd8.Visible=!cmd8.Visible;
cmd9.Visible=!cmd9.Visible;
cmdAsterisco.Visible=!cmdAsterisco.Visible;

cmdQ.Visible=!cmdQ.Visible;
cmdW.Visible=!cmdW.Visible;
cmdE.Visible=!cmdE.Visible;
cmdR.Visible=!cmdR.Visible;
cmdT.Visible=!cmdT.Visible;
cmdY.Visible=!cmdY.Visible;
cmdU.Visible=!cmdU.Visible;
cmdI.Visible=!cmdI.Visible;
cmdO.Visible=!cmdO.Visible;
cmdP.Visible=!cmdP.Visible;
cmdA.Visible=!cmdA.Visible;
cmdS.Visible=!cmdS.Visible;
cmdD.Visible=!cmdD.Visible;
cmdF.Visible=!cmdF.Visible;
cmdG.Visible=!cmdG.Visible;
cmdH.Visible=!cmdH.Visible;
cmdJ.Visible=!cmdJ.Visible;
cmdK.Visible=!cmdK.Visible;
cmdL.Visible=!cmdL.Visible;
cmdEnye.Visible=!cmdEnye.Visible;
cmdZ.Visible=!cmdZ.Visible;
cmdX.Visible=!cmdX.Visible;
cmdC.Visible=!cmdC.Visible;
cmdV.Visible=!cmdV.Visible;
cmdB.Visible=!cmdB.Visible;
cmdN.Visible=!cmdN.Visible;
cmdM.Visible=!cmdM.Visible;

cmdEspacio.Visible=!cmdEspacio.Visible;
}

private void installClickEvents()
{
    foreach (Control contHijo in panelTeclado.Controls)
    {
        // add to list
        if (contHijo is Button)
        this.keyCollection.Add((Button)contHijo);
    }

    foreach (Button buttonElement in keyCollection)
    {
        // install click event
        buttonElement.Click += new EventHandler(buttonElement_Click);
    }
}

private void buttonElement_Click(object sender, EventArgs)
```

```
{
// create variable for holding string
StringsendString="";

try
{
// set sendstring to key
sendString=((Button)sender).Tag.ToString();

// if something to send
if(!String.IsNullOrEmpty(sendString))
{
// if sending a string
if(sendString.Length>1)
{
// add {}
sendString="{ "+sendString+"}";
}
Controlcontrol=Control.FromChildHandle(ultControl);
control.Focus();

// send key to simulate key press
System.Windows.Forms.SendKeys.SendWait(sendString);
}
}
catch(Exception)
{
Console.WriteLine("Could not send key press: {0}",sendString);
}
}
/***** FIN TECLADO TOUCH *****/
```

20.2. Validar que el usuario sea de tipo cajero para abrir la caja.

```
privatebool ValidarUsuario()
{
bool UsuarioInvalido=false;

Global.getConexion().runSQLStatement(" SELECT (CASE (TIPO) WHEN '0' THEN
'CAJERO' "+
" WHEN '1' THEN 'SUPERVISOR' END) TIPOUSUARIO "+
"
FROM
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".USU
ARIO_TIPO "+
" WHERE USUARIO='"+txtUsuario.Text+"'");

if(Global.getConexion().dtResult.Rows.Count>0)
{
if(Global.getConexion().dtResult.Rows[0][0].ToString()=="SUPERVISOR")
```

```
{
EsSupervisor=true;
supervisor=txtUsuario.Text;
}
elseif(Global.getConexion().dtResult.Rows[0][0].ToString()=="CAJERO")
EsSupervisor=false;
else UsuarioInvalido=true;
}
else
UsuarioInvalido=true;

if(UsuarioInvalido==true)
{
Global.mensaje.Mostrar("Este usuario no tiene privilegios para facturar","Error!!!");
if(EsInitLogin==true)
return false;
else
{
EsSupervisor=false;
Close();
}
}
else
{
if(EsInitLogin==true)
Global.aperturaActual.setTipoUsuario(EsSupervisor);
else Close();
}
return true;
}
```

20.3. Abrir Caja si el tipo de usuario es cajero

```
private bool abrirCaja()
{
    if(!ValidarUsuario()) return false;

    //Obtener el tipo de cambio
    Global.getConexion().runSQLStatement("SELECT
    "+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+
    ".POS_GET_TIPO_CAMBIO ('"+Properties.Settings.Default.TipoCambio+"' )");

    if(Global.getConexion().dtResult.Rows.Count>0)
    {Global.aperturaActual.setTipoCambio(Convert.ToDouble(Global.getConexion().dtResult.
    Rows[0][0].ToString()));
    }

    else
    {
    Global.mensaje.Mostrar("Error al leer el tipo de cambio","Error!!!");
    return false;
    }
}
```

//Verificar si la caja está abierta y si el cajero que la quiere abrir es el que la abrió
previamente o si el cajero tiene abierta otra caja

```
Global.getConexion().runSQLStatement("SELECT USUARIO, APERTURA FROM "+  
Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+  
".POS_APERTURA_CAJA WHERE BODEGA ='"+Properties.Settings.Default.Bodega+"'  
AND CAJA='"+  
Properties.Settings.Default.Caja+" AND ESTADO='A'");
```

```
if(Global.getConexion().dtResult.Rows.Count>0)  
{  
if(Global.getConexion().dtResult.Rows[0][0].ToString().ToUpper()!=txtUsuario.Text.ToUpp  
er())  
{  
Global.mensaje.Mostrar("El usuario  
"+Global.getConexion().dtResult.Rows[0][0].ToString()+" tiene abierta esta  
caja."+Environment.NewLine+"Solo el puede abrir esta caja","Verifique!!!");  
returnfalse;  
}  
else  
{Global.aperturaActual.setApertura(Convert.ToInt32(Global.getConexion().dtResult.Rows[  
0][1].ToString()));  
ImprimirAperturaCaja();  
returntrue;  
}  
}  
else  
{  
Global.getConexion().runSQLStatement("SELECT CAJA FROM "+  
Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+  
".POS_APERTURA_CAJA WHERE ESTADO='A' AND  
USUARIO='"+Properties.Settings.Default.UltimoUsuario+  
"");
```

```
if(Global.getConexion().dtResult.Rows.Count>0)  
{  
Global.mensaje.Mostrar("Este usuario ya tiene abierta otra caja ("+  
Global.getConexion().dtResult.Rows[0][0].ToString()+")"+Environment.NewLine+  
"No puede abrir dos cajas con el mismo usuario!!!","Verifique!!!");  
returnfalse;  
}
```

```
Global.getConexion().runSQLStatement("EXEC  
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+  
".POS_ABRIR_CAJA '"+Properties.Settings.Default.Bodega+"', '"+  
Properties.Settings.Default.Caja+"', '"+Properties.Settings.Default.TipoCambio+"', '"+  
Properties.Settings.Default.Consecutivo+"");
```

```
if(Global.getConexion().dtResult.Rows.Count>0)
```

```
{Global.aperturaActual.setApertura(Convert.ToInt32(Global.getConexion().dtResult.Rows[0][0].ToString()));
ImprimirAperturaCaja();
return true;
}
else
{
Global.mensaje.Mostrar("Error al generar la nueva apertura de caja","Error!!!");
return false;
}}
```

20.4. Agregar Líneas a una Factura

```
private void AgregarLineaLista()
{
Global.getConexion().runSQLStatement("SELECT ISNULL(MAX(LINEA), 0) + 1 FROM
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".POS
_FACTURA_TEMPORAL WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"'
AND FACTURA = 'X' AND CAJA = '"+Properties.Settings.Default.Caja+"' AND USUARIO
= '"+Properties.Settings.Default.UltimoUsuario+"'");

int linea=0;
if(Global.getConexion().dtResult.Rows.Count>0)
linea= Convert.ToInt32(Global.getConexion().dtResult.Rows[0][0].ToString());

Global.getConexion().runSQLStatement("INSERT INTO
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".POS
_FACTURA_TEMPORAL
(BODEGA,CAJA,USUARIO,APERTURA,CLIENTE,LINEA,ARTICULO,CANTIDAD,PRECIO,IMPUESTO_LINEA,DESCUENTO_LINEA,TOTAL_LINEA,COSTO,FECHA,TIPO_CAMBIO,NIVEL_PRECIO,MONEDA_NIVEL,VERSION_NIVEL,ESTADO) "+
"VALUES (" +Properties.Settings.Default.Bodega+",
"+Properties.Settings.Default.Caja+", "+Properties.Settings.Default.UltimoUsuario+",
"+Global.aperturaActual.getApertura().ToString()+",
"+Properties.Settings.Default.Cliente_contado+", "+linea+",
"+Global.articuloActual.getArticulo()+", "+Global.articuloActual.getCantidad()+",
"+Global.articuloActual.getPrecio()+", "+Global.articuloActual.getImpuesto()+",
"+Global.articuloActual.getDescuento()+", "+Global.articuloActual.getTotalLinea()+",
"+Global.articuloActual.getCosto()+", GETDATE(),
"+Global.aperturaActual.getTipoCambio()+", "+Properties.Settings.Default.Nivel_precio+",
"+Properties.Settings.Default.Moneda_nivel+",
"+Properties.Settings.Default.Version_nivel+", 'A')");

refreshGrid();
}
```

20.5. Elimina una Línea de una Factura

```
private void borrarProductoLista()
{
    if (dgvListaProducto.SelectedRows.Count <= 0) return;
    if (MessageBox.Show("Desea eliminar los  
productos seleccionados", "Eliminar?", MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.No)
    return;

    foreach (DataGridViewRow row in dgvListaProducto.SelectedRows)
    {
        Global.getConexion().runSQLStatement("DELETE  
"+ Properties.Settings.Default.BaseDatos + ". "+  
Properties.Settings.Default.Company + ". POS_FACTURA_TEMPORAL "+  
" WHERE BODEGA = '"+ Properties.Settings.Default.Bodega + "' AND CAJA =  
"+ Properties.Settings.Default.Caja +  
" AND USUARIO = '"+ Properties.Settings.Default.UltimoUsuario + "' AND APERTURA =  
"+ Global.aperturaActual.getApertura().ToString() +  
" AND FACTURA = 'X' AND LINEA = '"+ row.Cells["LINEA"].Value.ToString());
    }

    lblProducto.Text = "";
    lblPrecio.Text = "";
    refreshGrid();
}
```

20.6. Abrir Caja de dinero

```
private void abrirCajaDinero()
{
    //Abrir Caja....
    string openCash = "\x1B\x70\x00\x50\x255";
    PrintDocument documento = new PrintDocument();
    ImprimirAbrirCajaDeDinero.SendStringToPrinter(documento.PrinterSettings.PrinterName,
    openCash);
}
```

20.7. Imprimir Factura

```
public void imprimir(string factura, int copias, bool reimpresion)
{
    if (factura == "") return;
    bool anulada = false;
```

```
Reportes.rptFacturarptFac = new Reportes.rptFactura();
```

```
Global.getConexion().runSQLStatement(" SELECT DISTINCT LINEA,  
FORMA_PAGO_DERIVADA, MONTO FROM POS..POS_FACTURA_FORMAS_PAGO "+  
" WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"' "+" AND FACTURA =  
"'+factura+' " ORDER BY LINEA ");
```

```
if(Global.getConexion().dtResult.Rows.Count>0)  
rptFac.Subreports["rptFacturaFormaPago.rpt"].SetDataSource(Global.getConexion().dtRe  
sult);  
else  
{  
Global.getConexion().runSQLStatement(" SELECT '-' FORMA_PAGO_DERIVADA, 0.00  
MONTO ");  
rptFac.Subreports["rptFacturaFormaPago.rpt"].SetDataSource(Global.getConexion().dtRe  
sult);  
}
```

```
Global.getConexion().runSQLStatement(" SELECT FACTURA, DESC_COND_PAGO,  
CAJERO, FECHA, CLIENTE, NOMBRE_CLIENTE, EMPRESA, NOMBRE_EMPRESA,  
"+" F.ARTICULO, F.DESCRIPCION, CANTIDAD, PRECIO, PRECIO_TOTAL_LINEA,  
IMPUESTO_LINEA, DESCUENTO_LINEA, A.CLASIFICACION_6 "+" FROM  
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+" POS  
_FACTURA F "+  
" INNER JOIN  
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+" ARTI  
CULO A "+  
" ON F.ARTICULO = A.ARTICULO "+  
" WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"' "+" AND FACTURA =  
"'+factura+' " ORDER BY LINEA ");
```

```
rptFac.SetDataSource(Global.getConexion().dtResult);  
rptFac.SetParameterValue("Duplicado",reimpresion);  
Global.getConexion().runSQLStatement("SELECT FACTURA FROM  
"+Properties.Settings.Default.BaseDatos+"  
"."+Properties.Settings.Default.Company+" POS_FACTURA_ANULADA "+" WHERE  
BODEGA = '"+Properties.Settings.Default.Bodega+"' AND FACTURA = '"+factura+'";
```

```
if(Global.getConexion().dtResult.Rows.Count>0)  
anulada=true;  
else  
anulada=false;  
  
rptFac.SetParameterValue("Anulada", anulada);  
rptFac.PrintToPrinter(copias,false,0,0);  
}
```

20.8. Anular una Factura

```
private void anular(string factura, string supervisor)  
{  
if(factura == "") return;
```

```
frmMotivoAnulacion motivo =new frmMotivoAnulacion();
motivo.setFactura(factura);
motivo.ShowDialog();

if(motivo.getMotivo()!="")
{
Global.getConexion().runSQLStatement("INSERT INTO
"+Properties.Settings.Default.BaseDatos+"."+
Properties.Settings.Default.Company+".POS_FACTURA_ANULADA(BODEGA, CAJA,
USUARIO, APERTURA, FACTURA, MOTIVO, SUPERVISOR, FECHA_HORA,
ANULADA_EN_EXACTUS)+"
" VALUES('"+Properties.Settings.Default.Bodega+"', '"+Properties.Settings.Default.Caja+"',
 '"+Properties.Settings.Default.UltimoUsuario+"', '"+
Global.aperturaActual.getApertura()+"', '"+ factura +"' , '"+motivo.getMotivo()+"', '"+
supervisor +"' , GETDATE(), 'N' )");

imprimir(factura,1,false);
}
}
```

20.9. Imprimir Reporte de Pre lectura

```
private void imprimirPrelectura()
{
Reportes.rptPrelecturarptPrelect=new Reportes.rptPrelectura();

Global.getConexion().runSQLStatement(" SELECT CASE FORMA_PAGO_DERIVADA
WHEN 'CAMBIO' THEN 'EFECTIVO' ELSE FORMA_PAGO_DERIVADA END
FORMA_PAGO_DERIVADA, "+
" SUM(CASE FORMA_PAGO_DERIVADA WHEN 'CAMBIO' THEN MONTO * -1 ELSE
MONTO END) MONTO "+
" FROM "+
" ( "+
"SELECT DISTINCT FACTURA, FORMA_PAGO_DERIVADA, MONTO FROM
"+Properties.Settings.Default.BaseDatos+
"."+Properties.Settings.Default.Company+".POS_FACTURA_FORMAS_PAGO "+
" WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"' "+
" AND CAJA = '"+Properties.Settings.Default.Caja+"' "+
" AND APERTURA = "+Global.aperturaActual.getApertura().ToString()+
" AND FACTURA NOT IN (SELECT FACTURA FROM
"+Properties.Settings.Default.BaseDatos+"."+
Properties.Settings.Default.Company+".POS_FACTURA_ANULADA) "+
" )X"+
" GROUP BY CASE FORMA_PAGO_DERIVADA WHEN 'CAMBIO' THEN 'EFECTIVO'
ELSE FORMA_PAGO_DERIVADA END ");

if(Global.getConexion().dtResult.Rows.Count>0)
rptPrelect.Subreports["rptFacturaFormaPago.rpt"].SetDataSource(Global.getConexion().dt
Result);
else
```

```
{
Global.getConexion().runSQLStatement(" SELECT '-' FORMA_PAGO_DERIVADA, 0.00
MONTO ");
rptPrelect.Subreports["rptFacturaFormaPago.rpt"].SetDataSource(Global.getConexion().dt
Result);
}

Global.getConexion().runSQLStatement("      SELECT      E.NOMBRE      EMPRESA,
SUM(X.MONTO) MONTO "+
" FROM "+
" ( "+
" SELECT DISTINCT F.FACTURA, F.CLIENTE, FP.MONTO MONTO "+
" FROM "+Properties.Settings.Default.BaseDatos+" "+
Properties.Settings.Default.Company+".POS_FORMA_PAGO_FACTURA_TEMPORAL
FP "+
" INNER JOIN "+Properties.Settings.Default.BaseDatos+" "+
Properties.Settings.Default.Company+".POS_FACTURA_TEMPORAL F "+
" ON FP.BODEGA = F.BODEGA AND FP.FACTURA = F.FACTURA "+
" WHERE FORMA_PAGO_DERIVADA = 'CREDITO' "+
" AND F.BODEGA = '"+Properties.Settings.Default.Bodega+"' AND F.CAJA =
 '"+Properties.Settings.Default.Caja+"' "+
" AND F.APERTURA = "+Global.aperturaActual.getApertura().ToString()+
" AND F.FACTURA NOT IN (SELECT FACTURA FROM
 '"+Properties.Settings.Default.BaseDatos+" "+
Properties.Settings.Default.Company+".POS_FACTURA_ANULADA WHERE BODEGA =
 '"+Properties.Settings.Default.Bodega+"' "+
" AND CAJA = '"+Properties.Settings.Default.Caja+"' "+
" AND F.APERTURA = "+Global.aperturaActual.getApertura().ToString()+") "+
" )X "+
" INNER JOIN "+Properties.Settings.Default.BaseDatos+" "+
Properties.Settings.Default.Company+".CLIENTE C "+
" ON X.CLIENTE = C.CLIENTE "+
" LEFT JOIN "+Properties.Settings.Default.BaseDatos+" "+
Properties.Settings.Default.Company+".CLIENTE E "+
" ON C.CLI_CORPORAC_ASOC = E.CLIENTE "+
" GROUP BY E.NOMBRE ");

if(Global.getConexion().dtResult.Rows.Count>0)
rptPrelect.Subreports["rptDetalleCredito.rpt"].SetDataSource(Global.getConexion().dtResu
lt);
else
{
Global.getConexion().runSQLStatement(" SELECT '-' EMPRESA, 0.00 MONTO ");

rptPrelect.Subreports["rptDetalleCredito.rpt"].SetDataSource(Global.getConexion().dtResu
lt);
}

Global.getConexion().runSQLStatement(" SELECT BODEGA, CAJA, USUARIO,
APERTURA, FECHA_HORA_APERTURA "+
```

```
"
FROM
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".POS
_APERTURA_CAJA "+
" WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"'" +
" AND CAJA = '"+Properties.Settings.Default.Caja+"'" AND APERTURA =
"+Global.aperturaActual.getApertura().ToString());

rptPrelect.SetDataSource(Global.getConexion().dtResult);

rptPrelect.PrintToPrinter(1,false,0,0);
}
```

20.10. Cerrar Caja

```
private void cerrarCaja()
{
Global.getConexion().runSQLStatement("UPDATE
"+Properties.Settings.Default.BaseDatos+"."+
Properties.Settings.Default.Company+".POS _APERTURA_CAJA SET
FECHA_HORA_CIERRE = GETDATE(), ESTADO = 'C', "+ RECIBIDO_EFECTIVO = "+
Convert.ToDouble(txtEfectivo.Text).ToString()+"", "+ RECIBIDO_TARJETA =
"+Convert.ToDouble(txtTarjeta.Text).ToString()+"", "+ RECIBIDO_CREDITO =
"+Convert.ToDouble(txtCredito.Text).ToString()+"", "+ RECIBIDO_ABONO_CREDITO =
SALDO_ABONO_CREDITO, "+ RECIBIDO_CHEQUE =
"+Convert.ToDouble(txtCheque.Text).ToString()+"", "+ RECIBIDO_REGALIA =
"+Convert.ToDouble(txtPromocion.Text).ToString()+"", "+ RECIBIDO_RETIRO = 0, "+
TOTAL_VENTAS = SALDO_FINAL_EFECTIVO + SALDO_TARJETA +
SALDO_CREDITO + SALDO_CHEQUE, "+ NOTAS = '"+txtNota.Text.Replace("'", "''")+"'|
BANPRO: '"+txtBanpro.Text+"| CREDOMATIC: '"+txtCredomatic.Text+"'" +
"+Properties.Settings.Default.Bodega+"'" AND CAJA =
"+Properties.Settings.Default.Caja+"'" AND APERTURA = "+
Global.aperturaActual.getApertura().ToString());
}
```

20.11. Imprimir Reporte de Cierre de Caja

```
private void imprimirCierre()
{
Reportes.rptCierre cierre = new Reportes.rptCierre();

Global.getConexion().runSQLStatement("SELECT BODEGA, CAJA, USUARIO,
APERTURA, MONEDA, TIPO, DENOMINACION, CANTIDAD "+
"
FROM
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".POS
_DETALLE_MONEDA_CIERRE "+
```

```
" WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"' AND CAJA =  
"'+Properties.Settings.Default.Caja+" "+  
" AND APERTURA = "+Global.aperturaActual.getApertura().ToString()+  
" ORDER BY MONEDA ASC, TIPO DESC, DENOMINACION ASC");  
cierre.Subreports["rptDetalleEfectivo.rpt"].SetDataSource(Global.getConexion().dtResult);
```

```
Global.getConexion().runSQLStatement(" SELECT E.NOMBRE EMPRESA,  
SUM(X.MONTO) MONTO "+  
" FROM "+  
" ( "+  
" SELECT DISTINCT F.FACTURA, F.CLIENTE, FP.MONTO MONTO "+  
" FROM '"+Properties.Settings.Default.BaseDatos+".' "+  
Properties.Settings.Default.Company+".POS_FORMA_PAGO_FACTURA_TEMPORAL  
FP "+  
" INNER JOIN  
"'+Properties.Settings.Default.BaseDatos+".'"+Properties.Settings.Default.Company+".POS  
_FACTURA_TEMPORAL F "+ " ON FP.BODEGA = F.BODEGA AND FP.FACTURA =  
F.FACTURA "+  
" WHERE FORMA_PAGO_DERIVADA = 'CREDITO' "+  
" AND F.BODEGA = '"+Properties.Settings.Default.Bodega+"' AND F.CAJA =  
"'+Properties.Settings.Default.Caja+" "+ " AND F.APERTURA =  
"'+Global.aperturaActual.getApertura().ToString()+  
" AND F.FACTURA NOT IN (SELECT FACTURA FROM  
"'+Properties.Settings.Default.BaseDatos+".' "+  
Properties.Settings.Default.Company+".POS_FACTURA_ANULADA WHERE BODEGA =  
"'+Properties.Settings.Default.Bodega+" "+  
" AND CAJA = '"+Properties.Settings.Default.Caja+" "+  
" AND F.APERTURA = "+Global.aperturaActual.getApertura().ToString()+") "+  
" )X "+ " INNER JOIN  
"'+Properties.Settings.Default.BaseDatos+".'"+Properties.Settings.Default.Company+".CLIE  
NTE C "+  
" ON X.CLIENTE = C.CLIENTE "+ " LEFT JOIN  
"'+Properties.Settings.Default.BaseDatos+".' "+  
Properties.Settings.Default.Company+".CLIENTE E "+  
" ON C.CLI_CORPORAC_ASOC = E.CLIENTE "+  
" GROUP BY E.NOMBRE ");
```

```
if(Global.getConexion().dtResult.Rows.Count>0)  
cierre.Subreports["rptDetalleCredito.rpt"].SetDataSource(Global.getConexion().dtResult);  
else  
{  
Global.getConexion().runSQLStatement(" SELECT '-' EMPRESA, 0.00 MONTO ");  
cierre.Subreports["rptDetalleCredito.rpt"].SetDataSource(Global.getConexion().dtResult);  
}
```

```
Global.getConexion().runSQLStatement("SELECT FA.BODEGA, FA.CAJA, FA.USUARIO,  
FA.APERTURA, FA.FACTURA, FA.MOTIVO, FA.SUPERVISOR, SUM(T.TOTAL_LINEA)  
TOTAL_FACTURA "+  
" FROM  
"'+Properties.Settings.Default.BaseDatos+".'"+Properties.Settings.Default.Company+".POS  
_FACTURA_ANULADA FA "+
```

```
"                                INNER                                JOIN
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".POS
_FACTURA_TEMPORAL T "+
" ON FA.BODEGA = T.BODEGA AND FA.CAJA = T.CAJA AND FA.APERTURA =
T.APERTURA AND FA.FACTURA = T.FACTURA "+
" WHERE FA.BODEGA = '"+Properties.Settings.Default.Bodega+"' AND FA.CAJA =
 '"+Properties.Settings.Default.Caja+"' "+
" AND FA.APERTURA = '"+Global.aperturaActual.getApertura().ToString()+"
" GROUP BY FA.BODEGA, FA.CAJA, FA.USUARIO, FA.APERTURA, FA.FACTURA,
FA.MOTIVO, FA.SUPERVISOR ");

cierre.Subreports["rptDetalleFacturasAnuladas.rpt"].SetDataSource(Global.getConexion().
dtResult);

Global.getConexion().runSQLStatement("SELECT  BODEGA,  CAJA,  USUARIO,
APERTURA,  FECHA_HORA_APERTURA,  FECHA_HORA_CIERRE,  ESTADO,
SALDO_INICIAL_EFECTIVO,  "+"  SALDO_FINAL_EFECTIVO,  SALDO_TARJETA,
SALDO_CREDITO, SALDO_ABONO_CREDITO, SALDO_CHEQUE, SALDO_REGALIA,
SALDO_RETIRO,  "+" TIPO_CAMBIO, TOTAL_CLIENTES, RECIBIDO_EFECTIVO,
RECIBIDO_TARJETA,  RECIBIDO_CREDITO,  RECIBIDO_ABONO_CREDITO,  "+"
RECIBIDO_CHEQUE, RECIBIDO_REGALIA, RECIBIDO_RETIRO, TOTAL_VENTAS,
SUBSTRING(NOTAS, 0, CHARINDEX('|', NOTAS))  NOTAS, "+" SUBSTRING(NOTAS,
CHARINDEX('|', NOTAS)  +  1  ,  LEN(CAST(NOTAS  AS  VARCHAR(300))))
DETALLE_TARJETAS                                "+"                                FROM
"+Properties.Settings.Default.BaseDatos+"."+Properties.Settings.Default.Company+".POS
_APERTURA_CAJA "+" WHERE BODEGA = '"+Properties.Settings.Default.Bodega+"'
AND  CAJA  =  '"+Properties.Settings.Default.Caja+"'  "+" AND  APERTURA  =
 '"+Global.aperturaActual.getApertura().ToString()";

cierre.SetDataSource(Global.getConexion().dtResult);
cierre.PrintToPrinter(1,false,0,0);
}
```

21. PROCEDIMIENTOS ALMACENADOS EN LA BASE DE DATOS

21.1. Abrir una caja

```
USE[POS]
GO
/***** Object: StoredProcedure [dbo].[POS_ABRIR_CAJA]    Script Date: 10/21/2012
15:35:17 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTERPROCEDURE[dbo].[POS_ABRIR_CAJA](@BODEGA VARCHAR(4), @CAJA
VARCHAR(10), @TCNAME VARCHAR(10), @CONSECUTIVO VARCHAR(20))
ASBEGIN
DECLARE @TC DECIMAL(28,8)
DECLARE @APERTURA INT
DECLARE @SIG_CONSECUTIVO VARCHAR(20)
DECLARE @DIA INT, @MES INT,@ANYOINT

SET @TC = DBO.POS_GET_TIPO_CAMBIO(@TCNAME)
SET @APERTURA =(SELECTISNULL(MAX(APERTURA),0)+1FROM
POS_APERTURA_CAJA WHERE BODEGA=@BODEGA AND CAJA=@CAJA)

INSERTINTO POS_APERTURA_CAJA (BODEGA, CAJA, USUARIO, APERTURA,
FECHA_HORA_APERTURA, TIPO_CAMBIO,ESTADO)
VALUES(@BODEGA, @CAJA, SUSER_SNAME(), @APERTURA,GETDATE(), @TC,'A')

--REINICIAR EL CONSECUTIVO DE CAJA POR DIA
SELECT @DIA =SUBSTRING(ULTIMO_VALOR,5,2), @MES
=SUBSTRING(ULTIMO_VALOR,7,2),
@ANYO =SUBSTRING(ULTIMO_VALOR,9,2)
FROM POS_CONSECUTIVO WHERE CONSECUTIVO = @CONSECUTIVO

IF(@DIA <>DAY(GETDATE())OR @MES <>MONTH(GETDATE())OR @ANYO
<>RIGHT(YEAR(GETDATE()),2))
BEGIN
SET @SIG_CONSECUTIVO =(SELECT
CAST((SELECT VALOR_INICIAL FROM POS_CONSECUTIVO WHERE
CONSECUTIVO = @CONSECUTIVO)ASVARCHAR(4))
+(CASEWHENDAY(GETDATE())>9
THENCAST(DAY(GETDATE())ASVARCHAR)
ELSE'0'+CAST(DAY(GETDATE())ASVARCHAR)END)
+
(CASEWHENMONTH(GETDATE())>9
```

```
THENCASE(MONTH(GETDATE()))ASVARCHAR)
ELSE'0'+CAST(MONTH(GETDATE()))ASVARCHAR)END)
+
RIGHT(CAST(YEAR(GETDATE()))ASVARCHAR),2)
+'-'+LEFT(REPLACE(@CAJA,'CAJA','')+ '000000000',9))

UPDATE POS_CONSECUTIVO SET ULTIMO_VALOR = @SIG_CONSECUTIVO
WHERE CONSECUTIVO = @CONSECUTIVO
END

SELECT @APERTURA
END
```

21.2. Crear Pre-lectura de una caja

```
USE[POS]
GO
/***** Object: StoredProcedure [dbo].[POS_SEND_PRELECTURA_APERTURA_CAJA]
Script Date: 10/21/2012 15:23:59 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTERPROCEDURE[dbo].[POS_SEND_PRELECTURA_APERTURA_CAJA](@BODEGA
VARCHAR(4), @CAJA VARCHAR(10), @APERTURA INT, @DATOS_TEMPORAL
VARCHAR(1)='N')
ASBEGIN
DECLARE @EFECTIVO DECIMAL(28,8), @TARJETA DECIMAL(28,8), @CREDITO
DECIMAL(28,8), @ANTICIPO DECIMAL(28,8)
DECLARE @CHEQUE DECIMAL(28,8), @REGALIA DECIMAL(28,8), @RETIRO
DECIMAL(28,8)
DECLARE @TOTAL_CLIENTES INT

IF @DATOS_TEMPORAL ='N'
BEGIN
SELECT @EFECTIVO =ISNULL(SUM(CASE FPB.FORMA_PAGO_BASE
WHEN'EFECTIVO'THEN FC.MONTO WHEN'CAMBIO'THEN FC.MONTO *-
1ELSE0END),0),
@TARJETA =ISNULL(SUM(CASE FPB.FORMA_PAGO_BASE
WHEN'TARJETA'THEN FC.MONTO ELSE0END),0),
@CREDITO =ISNULL(SUM(CASE FPB.FORMA_PAGO_BASE
WHEN'CREDITO'THEN FC.MONTO ELSE0END),0),
@ANTICIPO =(SELECTISNULL(SUM(F.MONTO_ANTICIPO),0)
FROM SALMAN.FACTURA F
INNERJOIN SALMAN.POS_APERTURA_CAJA_FACTURA ACF
ON F.FACTURA = ACF.FACTURA
WHERE ACF.BODEGA = @BODEGA
AND ACF.CAJA = @CAJA
AND ACF.APERTURA = @APERTURA
```

```
AND F.ANULADA='N'
),
    @CHEQUE =ISNULL(SUM(CASE FPB.FORMA_PAGO_BASE
WHEN'CHEQUE'THEN FC.MONTO ELSE0END),0),
    @REGALIA =ISNULL(SUM(CASE FPB.FORMA_PAGO_BASE
WHEN'REGALIA'THEN FC.MONTO ELSE0END),0)
FROM SALMAN.FACTURA F
INNERJOIN SALMAN.FACTURA_CANCELA FC
ON F.FACTURA = FC.FACTURA
INNERJOIN SALMAN.POS_FORMA_PAGO_DERIVADA FPD
ON FPD.ABREVIATURA = FC.TIPO
INNERJOIN SALMAN.POS_FORMA_PAGO_BASE FPB
ON FPB.FORMA_PAGO_BASE = FPD.FORMA_PAGO_BASE
INNERJOIN SALMAN.POS_APERTURA_CAJA_FACTURA ACF
ON ACF.FACTURA = FC.FACTURA
WHERE ACF.BODEGA = @BODEGA AND ACF.CAJA = @CAJA AND ACF.APERTURA
= @APERTURA
AND F.ANULADA='N'

SET @RETIRO =(SELECTISNULL(SUM(MONTO),0)FROM
SALMAN.POS_RETIRO_EFECTIVO WHERE BODEGA = @BODEGA AND CAJA =
@CAJA AND APERTURA = @APERTURA)
SET @TOTAL_CLIENTES =(SELECTCOUNT(FACTURA)FROM
SALMAN.POS_APERTURA_CAJA_FACTURA WHERE CAJA=@CAJA AND APERTURA
= @APERTURA)
END
ELSEBEGIN
SELECT @EFECTIVO =ISNULL(SUM(X.EFECTIVO),0), @TARJETA
=ISNULL(SUM(X.TARJETA),0), @CREDITO =ISNULL(SUM(CREDITO),0), @CHEQUE
=ISNULL(SUM(CHEQUE),0), @REGALIA =ISNULL(SUM(REGALIA),0)
FROM
(
SELECTDISTINCT F.FACTURA,ISNULL((CASE FP.FORMA_PAGO_DERIVADA
WHEN'EFECTIVO'THEN FP.MONTO WHEN'CAMBIO'THEN FP.MONTO *-
1ELSE0END),0) EFECTIVO,
ISNULL((CASE FP.FORMA_PAGO_DERIVADA WHEN'TARJETA'THEN FP.MONTO
ELSE0END),0) TARJETA,
ISNULL((CASE FP.FORMA_PAGO_DERIVADA WHEN'CREDITO'THEN FP.MONTO
ELSE0END),0) CREDITO,
ISNULL((CASE FP.FORMA_PAGO_DERIVADA WHEN'CHEQUE'THEN FP.MONTO
ELSE0END),0) CHEQUE,
ISNULL((CASE FP.FORMA_PAGO_DERIVADA WHEN'PROMOCION'THEN FP.MONTO
ELSE0END),0) REGALIA
FROM POS_FACTURA_TEMPORAL F
INNERJOIN POS_FORMA_PAGO_FACTURA_TEMPORAL FP
ON F.FACTURA = FP.FACTURA AND F.BODEGA = FP.BODEGA
WHERE F.BODEGA = @BODEGA
AND F.CAJA = @CAJA
AND F.APERTURA = @APERTURA
AND F.FACTURA NOTIN(SELECT FACTURA FROM POS_FACTURA_ANULADA)
)X

SET @ANTICIPO =(SELECTISNULL(X.TOTAL - Y.MONTO,0)
```

```
FROM
(
SELECT ISNULL(SUM(F.TOTAL_LINEA),0) TOTAL
FROM POS_FACTURA_TEMPORAL F
INNER JOIN POS_FORMA_PAGO_FACTURA_TEMPORAL FP
ON F.FACTURA = FP.FACTURA AND F.BODEGA = FP.BODEGA
WHERE F.BODEGA = @BODEGA
AND FP.FORMA_PAGO_DERIVADA = 'CREDITO'
AND F.CAJA = @CAJA
AND F.APERTURA = @APERTURA
AND F.FACTURA NOT IN (SELECT FACTURA FROM POS_FACTURA_ANULADA)
)X

FULL JOIN

(
SELECT SUM(MONTO) MONTO
FROM
(
SELECT DISTINCT F.FACTURA, ISNULL(FP.MONTO,0) MONTO
FROM POS_FACTURA_TEMPORAL F
INNER JOIN POS_FORMA_PAGO_FACTURA_TEMPORAL FP
ON F.FACTURA = FP.FACTURA AND F.BODEGA = FP.BODEGA
WHERE F.BODEGA = @BODEGA
AND FP.FORMA_PAGO_DERIVADA = 'CREDITO'
AND F.CAJA = @CAJA
AND F.APERTURA = @APERTURA
AND F.FACTURA NOT IN (SELECT FACTURA FROM POS_FACTURA_ANULADA)
)X
)Y
ON 1=1)
SET @RETIRO = 0
SET @TOTAL_CLIENTES = (SELECT ISNULL(COUNT(FACTURA),0) FROM
POS_APERTURA_CAJA_FACTURA WHERE BODEGA = @BODEGA AND
CAJA=@CAJA AND APERTURA = @APERTURA)
END

UPDATE POS_APERTURA_CAJA SET SALDO_FINAL_EFECTIVO = @EFECTIVO -
@RETIRO, SALDO_TARJETA = @TARJETA,
SALDO_CREDITO = @CREDITO, SALDO_ABONO_CREDITO = @ANTICIPO,
SALDO_CHEQUE = @CHEQUE,
SALDO_REGALIA = @REGALIA, SALDO_RETIRO = @RETIRO,
TOTAL_CLIENTES = @TOTAL_CLIENTES,
TOTAL_VENTAS = @EFECTIVO + @TARJETA + @CREDITO + @CHEQUE -
@RETIRO
WHERE BODEGA = @BODEGA AND CAJA = @CAJA AND APERTURA =
@APERTURA
END
```

21.3. Función para Obtener el precio final de un producto.

```
USE[POS]
GO
/***** Object: UserDefinedFunction [dbo].[POS_GET_PRECIO]    Script Date: 10/21/2012
15:24:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTERFUNCTION[dbo].[POS_GET_PRECIO](@ARTICULO VARCHAR(20))
RETURNSDECIMAL(28,8)
ASBEGIN
DECLARE @PRECIO DECIMAL(28,8), @DESCUENTO DECIMAL(28,8), @IMPUESTO
DECIMAL(28,8),
    @DESCUENTO_FIJO DECIMAL(28,8), @COSTO DECIMAL(28,8),
    @COSTO_IVA_DESC DECIMAL(28,8), @MARGEN DECIMAL(28,8)
DECLARE @BONIFICACION INT, @BONIFICACION_POR_CADA INT
DECLARE @TIPO_DESCUENTO VARCHAR(2)
DECLARE @CANTIDAD_COMPRA DECIMAL(28,8), @CANTIDAD_FACTURAR
DECIMAL(28,8)
DECLARE @PRECIO_V DECIMAL(28,8)
DECLARE @DESCRIPCION VARCHAR(300)

SELECT @PRECIO =ISNULL(LEFT(AP.PRECIO,CHARINDEX('.', AP.PRECIO)+5),0)
FROM ARTICULO A
LEFTJOIN ARTICULO_PRECIO AP
ON A.ARTICULO = AP.ARTICULO
WHERE(A.ARTICULO = @ARTICULO OR A.CODIGO_BARRAS = @ARTICULO)
AND AP.NIVEL_PRECIO='ND-LOCAL'

SELECT @DESCUENTO = RD.DESCUENTO, @TIPO_DESCUENTO =
RD.TIPO_DESCUENTO,
    @CANTIDAD_COMPRA = RD.CANTIDAD_COMPRA, @CANTIDAD_FACTURAR
= RD.CANTIDAD_FACTURAR
FROM PAQUETE_DESCUENTO PD

INNERJOIN PAQ_DESC_REG_DESC PDRD
ON PD.PAQUETE_DESCUENTO = PDRD.PAQUETE_DESCUENTO
INNERJOIN REGLA_DESCUENTO RD
ON PDRD.REGLA_DESCUENTO = RD.REGLA_DESCUENTO
INNERJOIN ARTICULO A
ON A.ARTICULO = RD.ARTICULO

WHERE(RD.ARTICULO = @ARTICULO OR A.CODIGO_BARRAS = @ARTICULO)AND
PD.ACTIVO ='S'
ANDCONVERT(char(8),GETDATE(),112)>=CONVERT(char(8), PD.FECHA_RIGE,112)
```

```
ANDCONVERT(char(8),GETDATE(),112)<=CONVERT(char(8),
PD.FECHA_VENCE,112)
ANDCONVERT(char(8),GETDATE(),108)>=CONVERT(char(8),
PD.FECHA_HORA_INICIO,108)
ANDCONVERT(char(8),GETDATE(),108)<=CONVERT(char(8),
PD.FECHA_HORA_FIN,108)

SET @IMPUESTO =(select ISNULL(I.VALOR,0)
FROM ARTICULO A
INNERJOIN IMPUESTO I
ON A.IMPUESTO=I.IMPUESTO
WHERE(ARTICULO=@ARTICULO OR CODIGO_BARRAS =@ARTICULO))

SET @PRECIO_V = @PRECIO +(@PRECIO *ISNULL(@IMPUESTO,0)/100)

SET @PRECIO_V = POS.DBO.POS_ROUND(@PRECIO_V,0.05,'TRUE')

SET @PRECIO_V = @PRECIO_V -(@PRECIO_V *ISNULL(@DESCUENTO,0)/100)

RETURNISNULL(POS.DBO.POS_ROUND(@PRECIO_V,0.05,'TRUE'),0)
END
```