

Universidad de Ciencias Comerciales



Proyecto de Graduación

PARA OPTAR AL TÍTULO DE:

INGENIERIA DE SISTEMAS DE INFORMACION

Tesina:

**Sistema de Venta-Compra y Control de Inventario
(SIVECOIN), y Sitio Web**

Empresa que afectara el sistema:



Sáenz Bloques Adoquines y Mas

Integrantes:

Ennio Sáenz M.	002857-04-01
Yessy Bodden Mejía	000544-04-01
Karime Hurtado	000532-04-01
Sharity Brown Gordon	000394-04-01

INDICE

I. Introducción.....	1
II. Objetivos.....	2
III. Antecedentes.....	3
IV. Justificación.....	4
V. Marco Metodológico.....	5
1. Introducción del Marco metodológico.....	5
2. Fase de Definición.....	7
2.1 Planificación Estratégica de la información (PEI).....	7
2.2 Análisis del Área de Negocio (AAN).....	10
2.3 Diseño del Área de Negocio (DAN).....	12
VI. Planificación del Proyecto.....	17
1. Calculo de la Viabilidad Técnico Operativa.....	18
1.1 Viabilidad Técnica.....	18
1.2 Viabilidad Económica.....	20
1.3 Viabilidad Operativa.....	22
1.4 Viabilidad Legal.....	23
2. Gestión del Proyecto.....	24
2.1 Selección del personal.....	24
2.2 Funciones del personal.....	24
2.3 Problema.....	24
2.4 Proceso (Etapas).....	24
2.5 Selección del modelo de Proceso.....	24
3. Ámbito.....	25
4. Estimación de costo del proyecto.....	25
5. Análisis de Riesgo.....	27
6. Planificación Temporal.....	28
7. Gestión de la configuración del proyecto.....	28
VII. Fase de Desarrollo.....	29
1. Diseño Conceptual.....	29

1.1 Descripción dl Proyecto.....	29
1.2 Diagrama de Caso de Uso General.....	30
1.3 Diagrama de Componentes.....	32
2. Diseño Lógico.....	34
2.1 Caso de uso particulares y sus especificaciones.....	34
2.1.1 Aplicación Windows.....	34
2.1.2 Aplicación Web.....	66
2.2 Diagramas de Actividad.....	76
2.2.1 Aplicación Windows.....	76
2.2.2 Aplicación Web.....	92
2.3 Diagramas de Secuencia.....	97
2.3.1 Aplicación Windows.....	97
2.3.2 Aplicación Web.....	126
2.4 Diagramas de Colaboración.....	131
2.4.1 Aplicación Windows.....	131
2.4.2 Aplicación Web.....	147
2.5 Diagramas de Clases.....	152
2.5.1 Aplicación Windows.....	152
2.5.2 Aplicación Web.....	168
2.6 Diagrama de Despliegue / Distribución.....	173
2.7 Tablas de la base de datos.....	174
VIII. Diseño Físico y Despliegue.....	175
1. Pantalla de captación de datos y Codificación.....	175
1.1 Aplicación Windows.....	175
1.2 Aplicación Web.....	344
2. Pruebas.....	371
IX. Conclusiones.....	372
X. Recomendaciones.....	373
XI. Bibliografía.....	374
XII. Anexos.....	375

Dedicatoria

A DIOS Padre, por guiar el sendero de mi vida hacia la luz espiritual y elevar mis valores morales al quehacer profesional de la sociedad desprotegida y proveerme de sabiduría y paciencia.

A mi esposa María Antonieta, que por sus cualidades, sabiduría y gran amor ha logrado mantener en mí ser una fuente de inspiración y amor .

A mi pequeño hijo, y gran fuente de inspiración Ennio Gabriel, quien vive en mi corazón como una eternidad, con este ejemplo le enseñé el camino para alcanzar la misión que un día fue un sueño que ahora se cristaliza, y que todo es posible en la vida

A mis Padres, fuente de admiración y sabiduría que gracias a sus consejos, e intersección ante Dios por cada uno de sus hijos para que seamos conducidos en la luz del camino del bien y su apoyo incondicional he logrado este triunfo de mi vida.

A mis queridos Hermanos, por el ánimo y amor que siempre me brindaron

Especialmente agradezco a mi Maestra Belkys Iglesias su apoyo incondicional, desinteresado y oportuno en el transcurso de estos largos años ya que sin ello esta tarea habría sido más difícil.

Y finalmente agradezco a todas aquellas personas que de alguna manera se vieron involucradas en mis quehaceres de estudiante y me brindaron de manera incondicional y desinteresada su valioso apoyo.

A todos Ustedes muchísimas **gracias** y que **DIOS** los bendiga

Ennio Javier Sáenz Martínez

Dedicatoria

A Dios por darme la fortaleza y sabiduría para lograr la realización de este trabajo

A mi madre Estela Vallejos que me brindo todo su apoyo moral espiritual y económico necesario para lograr llegar a culminar mi carrera de ingeniería de sistemas.

A mi padre y hermano que en todo el transcurso de mi vida me han apoyado incondicionalmente y a todas aquellas personas como maestros y amigos que de alguna forma me instaron y apoyaron para culminar este trabajo.

Karime Hurtado Vallejos

Dedicatoria

A dios que me ha bendecido y dado la fuerza de voluntad para seguir adelante en cada instante de mi vida durante todo el proceso de aprendizaje por el cual logré la dicha de culminar uno de mis sueños.

A mi madre Betty Bodden Bell quien con tanto esfuerzo sacrificio y amor me apoyó para finalizar esta ardua tarea.

A mi padre y hermana que aún estando lejos siempre estuvieron pendiente brindándome su apoyo incondicional.

A mis profesores forjadores de la enseñanza en especial a mi tutor Ingeniero Fausto Quiñones Varela quien con su sabiduría dedicación y esmero me brindó el conocimiento necesario para enfrentarme sin temor en el campo laboral. A mis compañeros en especial a Ennio Sáenz quien fue un gran líder en todo el transcurso del proyecto.

Yessy Bodden Mejia

Dedicatoria

A Dios fuente e inspiración que me permitió culminar una de mis grandes metas.

A mi querida madre Cecilia Garth, guía y compañera fiel en este largo trayecto de mi vida, sin ti nada hubiera sido posible me has dado todo el apoyo incondicional has sido madre, padre y maestra gracias por confiar en mí y darme la dicha de graduarme y ser toda una profesional.

A mis queridos hermanos quienes me han acompañado en este recorrido y me han animado a seguir adelante y llegar a donde estoy.

A mis grandes tutores quienes me han ayudado profesionalmente a mi formación a crecer como alumna y ser hoy la muestra de todo lo me han enseñado con mucha dedicación, paciencia, esmero y sabiduría Ing. Fausto Quiñones e Ing. Belkys Iglesias.

A todos ustedes les expreso mi más sincero agradecimiento.

Sharity Rossel Brown Gordon

Agradecimiento

Agradecemos especialmente a nuestra Universidad y a nuestros profesores su valioso apoyo desinteresado a lo largo de nuestra carrera, ya que gracias a sus cualidades humildes fue posible que visualizáramos el salto cualitativo e importante que daríamos a nuestras vidas cristalizando este gran sueño

¡Gracias Maestros que Dios los bendiga!

I. Introducción

La empresa **Sáenz Bloques adoquines y Mas...** ubicada en el Km. 10 Carretera Vieja a León 1 km. Al Sur, en la Comarca Nejapa Sur, inició sus operaciones producción en el año 2005.

Esta empresa tiene como principal función la fabricación artesanal de bloques y adoquines de primera calidad a precios competitivos al mercado de la construcción tanto horizontal como vertical. Su principal segmento de mercado son los gobiernos municipales y empresas constructoras privadas y en menor escala el público en general por demanda espontánea.

Los bloques y adoquines están constituidos en base a materia prima de cemento, material cero, piedra pómez y arena. Los productos finales están sometidos por la empresa Nicasolum para verificación del control de calidad de forma aleatoria, los cuales deben cumplir con los requisitos emitidos por el MTI entidad rectora de la construcción en Nicaragua.

Actualmente esta empresa la constituye un gerente propietario una cajera, un vendedor, un Jefe de planta, un Responsable de Bodega, un Contador, un vigilante un Conductor de camión y siete trabajadores directos los cuales devengan un salario de acuerdo a la producción semanal, el personal administrativo tiene definido un salario mensual fijo.

La contabilidad es responsabilidad del Contador quien le asigna un fondo de tiempo de 40 horas Llevando los libros de rigor y haciendo las correspondientes gestiones de pago de impuestos de acuerdo a la ley, así como el correspondiente análisis contable.

La empresa cuenta con tres máquinas, una de hacer bloques y otras dos de adoquines así como también una mezcladota las cuales cada una produce un promedio de 2000 unidades de bloques o adoquines por día. El promedio de

ingreso producto de la venta asciende a un monto de 50,000 córdobas semanales, actualmente se cuenta con un inventario de 180,000 adoquines y 60,000 bloques.

Al contar la empresa con este “Sistema de Venta-Compra y control de Inventario” favorecería una mejor planificación de las ventas de producto final, así como compra de materia prima, y un mejor control en el inventario de materia prima y producto final, la calidad del almacenamiento de la misma y mejoraría la gestión en la oferta de los productos terminados

II. Objetivos

1. Objetivo General

- 1.1 Diseñar un Sistema de Control de Inventario, que facilite la gestión en la adquisición de materia prima y venta de productos finales de la empresa Sáenz Bloque Adoquines y Más.
- 1.2 Diseñar un sitio web, que facilite a clientes cautivos de la empresa Sáenz Bloque Adoquines y Más las compras de los productos finales a ofertarse.

2. Objetivos Específicos

- 2.1 Mantener un control de inventario tanto de materia prima entrante como del producto final que sale a la venta.
- 2.2 Mantener un Control de las facturas de compra y de venta
- 2.3 Incrementar las ventas a través de un sitio web

III. Antecedentes

La empresa Sáenz Bloque Adoquines y mas desde el inicio de sus operaciones no ha contado con equipo de cómputo, todas las entradas y salidas del inventario es registrado en un cuaderno de apuntes, el cual lo lleva el jefe de producción. Al final del día el jefe de producción le entrega al gerente una copia de todos los movimientos del día.

Las facturas se llenan manualmente, las cuales se van almacenando en una caja, al igual que las facturas de compras y los recibos de pago

IV. Justificación

En la actualidad la empresa **Sáenz Bloques, Adoquines y Más...** lleva sus controles de producción, compra de materia prima, controles de inventario, ventas, pago a empleados, de manera manual.

Es decir no cuenta con un sistema automatizado, No cuenta con ningún sistema de controles que le permitan manejar de manera más eficiente los ingresos, egresos, almacenamiento y salida tanto de materia prima como de productos terminados.

Con el montaje de un “Sistema de Venta-Compra y control de Inventario” le permitirá tener una hoja de controles sobre su producción, con el fin de evitar el desperdicio y la fuga de materia prima productos terminados y un control sobre las facturas de ventas y facturas de compras

V. Marco Metodológico

1. Introducción del Marco Metodológico

El método implementado para la recopilación de la información fue una entrevista realizada al gerente de la empresa **Sáenz Bloques Adoquines y más...**, donde se le hicieron una serie de preguntas relacionadas al funcionamiento de la empresa, al sistema y al sitio Web.

Existen varios entes relacionados para el desarrollo del sistema y sitio Web, entre ellos se encuentran los siguientes:

- ✓ **Gestor del proyecto:** el cual se encarga de ayudar con la codificación, administrar y evaluar el desarrollo del proyecto, también estará a cargo de administrar el proyecto instruyendo a los profesionales o expertos en sus labores correspondientes, tratando de tener un control, planificando y organizando cada tarea.
- ✓ **Profesionales:** Este proyecto consta de tres profesionales quienes estarán a cargo de la codificación y desarrollo del proyecto, que se desarrollara en una plataforma de Visual Basic Net 2005, con un gestor de datos en SQL.
- ✓ **Gerente de la Empresa:** El gestor del proyecto estará en contacto con el gerente para presentarle las etapas finalizadas del proyecto para su evaluación y verificación del cumplimiento de las expectativas.

Para el funcionamiento del sistema y sitio Web en la empresa es necesario la adquisición de 5 máquinas de cómputo las cuales trabajaran en diferentes áreas de la empresa, de las cinco maquinas una desempeñara el papel de servidor el cual trabajara sobre una plataforma de Windows Server 2003, y las otras cuatro servirán como clientes, las cuales tendrán instalada la plataforma de Windows XP

2003, dentro del servidor se encontrara alojado el sitio Web de la empresa, la cual se podrá visualizar a través del un explorador Web

Esto traerá beneficios a la empresa como:

- Agilizar los procesos internos
- Llevar un mejor control de las entradas y salidas del Inventario (Materia Prima y Producto Final)
- Llevar un mejor control de las facturas de Ventas y de compras
- Posibilidad de Capturar más Clientes vía Internet
- Aumentar las ventas por medio del Internet
- Llevar un control sobre las entradas y salidas de materia prima y productos

Las PC clientes estará ubicado en las diferentes aéreas de la empresa (Contabilidad, Venta, Producción y Producción-Bodega), montadas bajo una plataforma de Windows XP, en la gerencia estará el servidor el cual estará montado en una plataforma Windows Server 2003 donde estará localizada la base de datos (SBDBAM.mdf).Cada máquina tendrá una dirección IP estática clase C y una máscara de red de 255.255.255.0

El sitio Web estará localizado en el Servidor de la empresa el cual está ubicada en gerencia, bajo el Sistema Operativo Windows Server 2003, el cual se podrá visualizar vía Internet introduciendo el DNS en el campo de búsqueda del explorador de Internet, usará la misma base de datos del proyecto Windows (SIVECOIN) localizada en el servidor.

En el sistema se podrá generar:

- **Reporte de Venta:** Se visualizarán e imprimirán todas las facturas de venta de la siguiente manera:
 - ✓ Fecha
 - ✓ Tipo de compra (Crédito, Contado)

- ✓ Usuario
 - ✓ Estado de la factura (Anulada, Vigente)
 - ✓ Lugar Facturado (Web, Local)
 - ✓ Todas las facturas de venta
-
- **Impresión y visualización de factura de venta:** Aquí es cuando se imprime la factura de venta al cliente ya sea de contado o crédito

 - **Reporte de los abonos de créditos** Esta impresión se hace cada vez que el cliente hace un abono por el monto a deber en la factura de crédito

 - **Reporte de Compra:** Se visualizarán e imprimirán todas las facturas de compra de la siguiente manera:
 - ✓ Fecha
 - ✓ Tipo de compra (Crédito, Contado)
 - ✓ Usuario
 - ✓ Estado de la factura (Canceladas, Vigente)
 - ✓ Todas las facturas de compra

 - **Reporte del Producto Final:** Aquí se visualizaran e imprimirán las existencias de producto

 - **Reporte de Materia Prima:** En este reporte se visualizaran e imprimirán las existencias de materia prima

 - **Reportes de entradas y salidas de Materia prima:** Este tipo de reporte se imprimirá diariamente para poder llevar un mejor control de las entradas y salidas de la materia prima y evitar fugas

- **Reporte de entradas y salidas de Producto final:** Este tipo de reporte se imprimirá diariamente para poder llevar un mejor control de las entradas y salidas del producto final y evitar pérdidas.
- **Reporte de los usuarios, proveedores y clientes:** Este reporte se utilizara para poder llevar una lista de los Usuarios, Clientes y proveedores

2. Fase de definición

2.1 Planificación Estratégica de la Información (PEI)

El PEI tiene por objeto crear un modelo de datos a nivel de negocio que define los objetos de datos clave y sus relaciones entre ellos y con otras áreas del negocio.

2.1.1 Estructura Organizacional del negocio.

Actualmente el negocio está Estructurado de la siguiente manera:

a. Gerente General:

- Se encarga de alinear el cumplimiento de los procesos y procedimientos de la empresa, Así mismo desarrolla políticas y planes estratégicos y operativos.
- Se encarga de desarrollar las gestiones pertinentes al cumplimiento de los planes productivos de cara a la satisfacción del cliente.
- Monitorea constantemente la calidad de los procesos de producción para la satisfacción de la demanda
- Desarrolla políticas y planes estratégicos de la oferta de los servicios para satisfacción de la clientela.

- Elabora análisis del comportamiento de producción y mercadeo para la toma de decisiones oportunas y objetivas.
- Vela por el buen funcionamiento de la empresa en general

b. Limpieza:

Es la que se encarga de hacer la limpieza dentro de la empresa

c. Vigilancia:

El vigilante se encarga de cuidar el local en horas no laborales.

d. Venta:

En esta área existen dos Vendedores un vendedor ambulante el cual sale fuera de la empresa a ofrecer los servicios de la empresa con el objetivo de captar más clientes, y el otro estático que es el que permanece todo el día dentro de la empresa atendiendo a los clientes que llegan a las instalaciones.

e. Contabilidad:

En esta área el cajero es encargado de hacer facturas por cada Venta, recibir el monto de la factura y pagar a los proveedores el monto facturado por compras realizadas.

f. Producción:

El Responsable de esta área, tiene varias funciones:

Tiene a su cargo a ocho trabajadores que se encargan de fabricar los productos, llevarlos al área correspondiente de secado y luego al área de fraguado, y también...

- Solicita al Gerente los pedidos de materia prima para que este solicite a los proveedores,

- Lleva un cuaderno donde apunta el control de las entradas y salidas de materia, prima a las instalaciones de la bloquera.
- Lleva el control de materia prima que se utilizará en el día y del total de la producción diaria,
- Mantiene relación directa con el Responsable de Bodega y es quien le solicita la existencia de materia prima, y las cantidades a las que debe dar salida para la producción diaria.

El Jefe de Producción tiene como mayor responsabilidad la supervisión y verificación de la mezcla de la materia prima y de que esta cumpla con los requerimientos estipulados para la fabricación tanto de los bloques como los adoquines garantizando de esta manera la calidad de los mismos.

- También controla y garantiza el fraguado de la producción. Este proceso consta de tres etapas:
 - ✓ Fabricación del producto
 - ✓ Acarreo a la zona de secado
 - ✓ Acarreo a la zona de fraguado

En la zona de secado el producto terminado permanece de durante unas horas para luego ser trasladado al área de fraguado. En esta área los bloques y adoquines deberán fraguarse de tres a cinco veces al día hasta completar los 28 días de fabricación para cumplir con los requisitos del MTI (Ministerio de Infraestructura y Transporte) este proceso es ejecutado por los trabajadores directos de la producción durante el día y por el vigilante por la noche y parte de la madrugada para . Luego de cumplidos

los 28 días pasaran al área de almacenaje o bodega y de ahí saldrán a la venta.

g. Bodega:

El producto terminado se almacena en bodega. En la bodega también se almacena La materia prima en un área techada. El responsable de esta área, se encarga de llevar el control de las entradas y salidas tanto del producto terminado como de la materia prima a utilizar diariamente, llevando un control en un cuaderno.

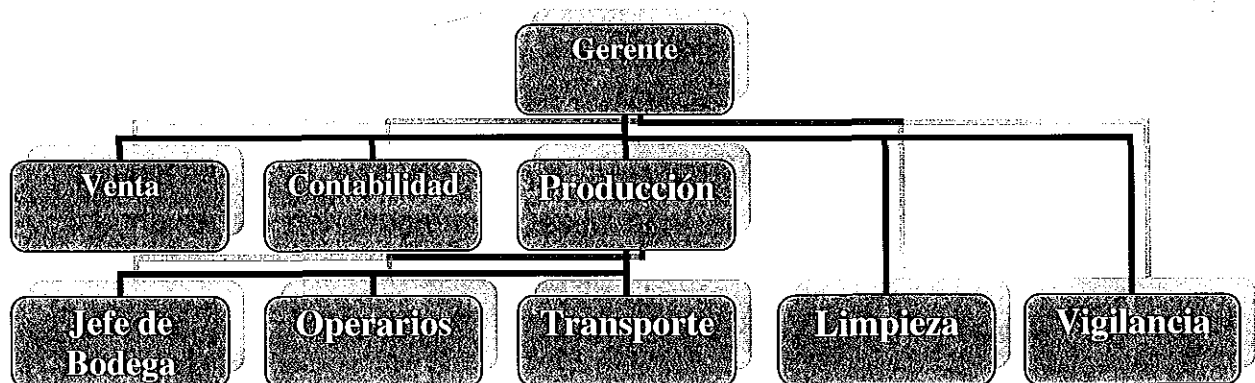
h. Operarios:

Son los trabajadores que se encargan de operar las maquinas (bloquera, adoquinera y mezcladora) de mezclar y trasladar la diferente materia prima a la maquina correspondiente para elaborar los productos finales: bloques y adoquines.

i. Transporte:

El Conductor es quien se encarga trasladar desde el proveedor la materia prima así como de la entrega de los pedidos a su destino final cuando el cliente así lo requiere.

2.1.2 Organigrama de la empresa Sáenz Bloques Adoquines y mas...



2.1.3 Funcionarios y Empleados

a. Funcionario

- Gerente General

b. Empleados

- Una Cajera
- Dos Vendedores
- Un Jefe de Producción
- Un Encargado de Bodega
- Tres Operarios
- Ocho Trabajadores de producción
- Un Conductor
- Una de Limpieza
- Uno de Vigilancia

2.2 Análisis del Área del Negocio (AAN)

2.2.1 Áreas Afectadas con el diseño e implementación del Software / aplicación

Las áreas afectadas con la implementación de la aplicación son:

a. El área del Gerente:

En esta Área se encontrará instalado el Servidor el cual tendrá la plataforma de Windows Server 2003 donde estará alojado el sitio Web y una Copia del Programa. Desde aquí el Gerente Pueda Visualizar los reporte y los Registros tanto del sitio Web como los del sistema.

b. El área de Venta:

En esta área estará instalado un equipo de cómputo (Cliente), con una plataforma de Windows XP 2003, ya que desde aquí se realizarán las ventas y orden de entrega del producto final, de la cual se imprimirá una copia en esta área para el cliente y otra copia en bodega para poder agilizar la entrega, desde aquí se ingresaran los registros del Cliente, registros de venta y registros de orden de salida.

c. El área de contabilidad:

Aquí se instalará un equipo de computo (Cliente), con el sistema operativo de Windows XP 2003, desde aquí se harán búsqueda de facturas elaboradas por el vendedor a la hora de introducir los datos de la venta, para ser impresa después de recibir el pago por dicho pedido.

Desde aquí también se podrán registrar las compras hechas a los proveedores y se podrá llevar un registro de los datos de las empresas proveedoras, y se podrán imprimir facturas elaboradas desde el Internet

d. El área de producción:

En esta área estará instalado un equipo de computo, desde aquí se elaboraran las ordenes de entrega de materia prima la cual se imprimirá en el área de bodega para agilizar la entrega y la producción.

e. El área de bodega:

Aquí Se instalara un equipo de cómputo para poder visualizar e imprimir las cantidades en existencia, para poder llevar un mejor control y garantizar un stock de productos y o materia prima.

2.2.2 Procesos de Trabajo en estas áreas

Cuando un cliente hace un pedido, Primero el vendedor pregunta al jefe de producción cuánto en existencia hay de producto final disponibles para vender, si no hay, el jefe de Producción le avisa al vendedor cuando habrá disponibilidad, en esta bodega el bodeguero se encarga de mantener siempre en el inventario un stock tanto de materia prima como de producto final. Cuando el encargado de bodega observa que se están agotando la materia prima el encargado de bodega avisa al jefe de producción que se necesita hacer pedidos de compra de materia prima, el jefe de producción informa al gerente para que este haga los pedidos a los proveedores, el jefe de producción también se encarga de mantener la calidad requerida por el MTI y evitar los desperdicios de materia prima; también el entrega con la ayuda de los trabajadores jornaleros al encargado de bodega el producto final de la producción y pedido de materia prima.

Después de verificar con el jefe de producción la disponibilidad, el vendedor le informa al cliente si existe en inventario la cantidad solicitada por el mismo, luego el vendedor le hace una orden de entrega, después el cliente con su orden pasa a caja para poder hacerle la factura

El cliente puede retirar su compra en ese mismo momento o indicarnos que día podremos ir a entregar su compra ya que tenemos servicio a domicilio, claro, dependiendo de la cantidad Comprada por el cliente, si la cantidad supera las 50 unidades y dependiendo de la distancia que se va a recorrer para llegar al destino de la entrega.

El cliente tiene que llegar con una orden de compra en mano con la cantidad a entregar y el producto para que el encargado de bodega

pueda entregar el producto al cliente, en el caso de mandárselo a dejar el encargado de bodega le avisa al jefe de producción que hay una entrega este verifica en caja, si todo está en orden el jefe de producción le da la orden de salida de los productos al encargado de bodega. Después el bodeguero entrega el pedido al conductor del camión para que este haga la entrega. El conductor debe de llevarse una orden de entrega para que el cliente la firme.

Todo se lleva por escrito, existe un archivero donde se guarda los recibos de pago, facturas de compra y ventas, y los cuadernos donde están las entradas y salidas de materia prima y producto final

2.3 Diseño del Sistema del Negocio (DSN)

2.3.1 Clases del Sistema

Para la elaboración de las clases es necesario recurrir al informe que se realizó a partir de la entrevista, donde se encuentran las necesidades de la empresa.

Las clases están formadas por tablas que contienen variables, estas variables están en dependencia de las necesidades de la empresa, ya que a partir de aquí se elabora la base de datos del sistema. Aquí mismo se encuentran las funciones que se realizarán para poder Mostrar o Ingresar datos.

Para elaborar las clases se utilizó el software de Rational Rose, aquí se realizó un diagrama de las clases que se implementarán al sistema (*ver diagrama en la fase de desarrollo en el diseño lógico*)

2.3.2 Flujo de los Procesos

a. Aplicación Windows

SIVECOIN (Sistema de venta, compra y control de inventario) está diseñado para seguir los procesos de ventas compras y control de inventario de la empresa Sáenz Bloques Adoquines y Mas..., Primeramente antes que nada en la empresa se deberá de crear un área de informática, en donde se encargaran de administrar tanto el sitio Web como el Sistema Local (SIVECOIN), o contratar a un informático para que se lo administre, esta decisión depende de la empresa; Estos procesos funcionan de la siguiente manera:

Primero se deben de crear las cuentas de los usuarios donde se registraran sus datos personales, nombre de usuario, contraseña y el cargo de este. Se le pide registrar el cargo ya que cada usuario estará restringido de las funciones de los demás cargos dentro del sistema, solo se limitara a realizar las acciones que el sistema le permita, por defecto el Sistema trae creada una cuenta para el administrador con un nombre de usuario ("admin") y una contraseña ("admin"), el administrador tiene control sobre todo el sistema. Ya creadas las cuentas de usuario, al iniciar el sistema, este le pedirá el nombre de usuario y contraseña de usuario por medio de un formulario (Login Form). Al introducir el nombre y contraseña el sistema identifica automáticamente el cargo de este, habilitando solamente sus formularios o acciones que puede realizar.

Dentro del sistema existen cinco tipos de cargos, los cuales son:

- Administrador
- Cajero
- Vendedor
- Bodeguero

Al entrar al sistema se visualiza un menú principal, el cual tiene una barra de menú, donde aparecen menú con nombre de las diferentes áreas de la empresa.

Para poder realizar los procesos con eficiencia y eficacia dentro del sistema a la hora de hacer una venta:

Cuando un cliente llega a la empresa y pregunta por uno de los productos, el vendedor visualiza la existencia en el sistema, donde aparecerá la cantidad exacta en existencia, si no existe la cantidad demandada por el cliente, el vendedor informa al jefe de producción de que la cantidad existente no es suficiente para cubrir la demanda solicitada, luego el vendedor le informa al cliente del inconveniente; pero si existe la cantidad solicitada por el cliente, el vendedor registra los datos de la venta en el sistema. Al hacer esto el vendedor crea una factura y una orden de entrega del producto final por medio de un formulario del sistema, se imprime una copia de la orden de entrega para el cliente y otra se imprime en bodega para agilizar la entrega, la factura se guarda en la base de datos, al ingresar las cantidades de los productos, en la base de datos del sistema la existencia disminuye en la base de datos.

Al momento de efectuar la factura, automáticamente el sistema hace los cálculos para poder sacar el total de cada detalle, el sub total, el IVA y el gran total de la factura.

En el momento de efectuar el pago la cajera le pide al cliente el nombre de quien se hizo la factura para poder hacer una búsqueda de la factura en la base de datos y poder verificar los datos ingresados por el vendedor, si la compra es de contado o

con tarjeta de crédito, la cajera imprime la factura en la impresora localizada en su área, si la compra es al crédito el sistema le pide que introduzca la cantidad del primer abono, mostrándole a través de un data grid los abonos realizados anteriormente si la compra es antigua o el abono realizado si es el primer abono, el total de la deuda y el total que se debe. En la factura impresa de Crédito se reflejarán todos estos datos mencionados anteriormente.

Cuando el encargado de bodega ve la orden de entrega del producto, inmediatamente solicita ayuda para poder alistar el pedido con las cantidades y los productos según la orden de entrega del producto final; cuando el cliente ya pasó por caja su próxima estación será el área de bodega, donde mostrará la factura al encargado de bodega para que este compare los datos de la factura mostrada por el cliente con los datos de la orden impresa en la impresora localizada en bodega por el vendedor, para luego proceder a la entrega de los productos mostrados en la orden de entrega al cliente.

Para poder tener un inventario de productos finales es necesario tener materia prima en inventario, para poder tener materia prima en inventario es necesario hacer compras de la misma a empresas proveedoras; para lograr esto en el sistema antes que nada primero debemos de registrar a los proveedores a través de un formulario del sistema, donde el proveedor nos facilitará sus datos en el cual necesitamos los datos personales del contacto y los datos de la empresa

b. Aplicación Web

Esta empresa tendrá su propio sitio Web el cual estará alojado en el servidor de la empresa, desde aquí se podrán hacer compras vía Internet, el proceso a seguir para poder hacer compra online es el siguiente:

Primero es necesario que el cliente introduzca en su navegador Web la dirección IP o el nombre de dominio de la empresa en el browser de su explorador el cual dirige al explorador al sitio Web que estará alojado en el servidor de la empresa, si el cliente quiere ver los productos tendrá que hacer Clic sobre el botón de catalogo del menú del sitio Web, aquí aparecerán imágenes de los productos con sus características y precios actualizados.

Para que el cliente pueda realizar la compra, primero tendrá que registrarse, por medio de un formulario en donde le pedirá sus datos personales, un Login y una contraseña. Al registrarse, toda la información ingresada por el cliente será almacenada en nuestra base de datos localizada en el servidor de la empresa. Una vez registrado el cliente, el siguiente paso será escoger los productos de una lista que le aparecerá en la página de compras y un campo vacío donde introducirá la cantidad del producto, cada vez que el cliente escoge un producto nuevo e introduce la cantidad solicitada, deberá de dar Click en botón agregar, la función de este botón es agregar el producto seleccionado y la cantidad a una tabla de detalle de la base de datos, cada vez que agregue un producto, en la parte inferior de la página aparecerá un data grid en donde podrá visualizar el producto, la cantidad, el sub total, el IVA y el total de la compra; Cuando el cliente haya terminado de escoger los productos y las cantidades, tendrá que

pulsar el botón "btnListo", este botón es un link a la siguiente pagina, en la cual se hace el pago mediante tarjeta de crédito o debito. El cliente tendrá que introducir los datos de la tarjeta de crédito o debito para poder finalizar con la compra, para poder finalizar con la compra, el cliente deberá de hacer click en el botón "btnFinalizar", para poder guardar los datos de la tarjeta en nuestra base de datos, y a la vez dando Click en este botón se abre una nueva página Web, donde se visualizara la factura del cliente, con la cual puede ir a retirar en las instalaciones de la empresa el pedido.

Diariamente la cajera o el administrador tendrá que revisaran las ventas mediante un data grid en el Sistema Local, donde se visualizara los datos de la factura y detalles de cada venta hecha desde el Internet, teniendo la opción de imprimir la factura para tener una copia de esta. Así mismo también se llevaran reportes los cuales se podrán imprimir en el sistema Local (SIVECOIN)

2.3.3 Servicios que ofrecerá la red

Para la implementación de la red, es necesario un Switch de 8 Pts., 300mt de cable UTP cat-5e, 12 conectores RJ-45, esta red estará conectada a 5 equipos de computo, uno es un servidor y 4 son clientes.

Esta red ofrecerá los siguientes servicios dentro de la empresa:

Comunicación entre nodos

Podrán compartir archivos

Se agilizaran los procesos

Compartirán una sola base de datos la cual estará alojada en el servidor

2.3.4 Diseño de la arquitectura del software

a. Aplicación Windows

Esta aplicación está diseñada en una plataforma con un lenguaje de programación de desarrollo de software Visual Basic .Net (VB.NET), y un gestor de Base de Datos en SQL Server Express 2005, dicha aplicación se instalara en cuatro Computadora con un Sistema Operativo Windows XP 2003 y un servidor con un sistema operativo de Windows Server 2003, la aplicación tendrá una interfase de Cliente-Servidor.

Se necesitaran cinco equipos de Cómputo, uno en cada área de la empresa (Gerencia, Producción, Venta, Contabilidad, Bodega), las cuales estarán enlazadas por una red de ordenadores LAN, así mismo un servidor que estará instalado en el área del Gerente, donde estará instalada la base de datos, el gestor de datos SQL Server Express, visual estudio net 2005 y una copia del sistema para cada estación.

La aplicación consta de formularios, en los cuales se pueden ingresar o visualizar los datos que se encuentran dentro de la base de datos (DBSBAM.MDF).

Para la conexión de datos con los formularios se usaron consultas de SQL Express dentro de cada formulario, relacionando cada campo de texto, combobox, listbox, Label etc. con los campos de las tablas de la base de datos correspondiente a cada formulario, de la misma manera se relaciona cada Reporte y DataGrid del sistema pero con la diferencia de que todos los reportes y

DataGrid hacen referencia a un DataSet que al mismo tiempo hace referencia a la base de datos del sistema.

Los Reportes del sistema están Diseñados en Crystal Report 8.5

b. Aplicación Web

la aplicación Web está diseñada bajo un lenguaje de programación de Visual Basic.net, HTML y ASPX, y un gestor de datos de SQL Server Express 2005, dicha aplicación estará alojada en un servidor, el cual estará instalado en el área de gerencia de la empresa. Esta aplicación podrá ser visualizada por los clientes vía Internet

El sitio Web tendrá un Menú Principal dentro de un "MasterPage", el menú contiene enlaces en forma de botones.

El sitio Web esta diseñado de la misma forma que la aplicación Windows, donde se hacen consultas SQL para hacer conexión con la base de datos. Las consultas conectan a las tablas de la base de datos según la tabla que se necesite usar pudiendo usar condiciones y relaciones entre tablas o campos de las tablas

Del mismo modo que la aplicación Web, se utilizan formularios con campos de texto, combobox, label, listbox, dataGrid, Reporte, para poder ingresar, visualizar, e imprimir los datos de la base de datos

Dentro del sitio Web se creo un reporte de Cristal Report 8.5, en el cual se visualizara la factura del cliente.

VI. Planificación del Proyecto

Elementos necesarios para implementar el Proyecto Windows

Los elementos necesarios para la implementación del proyecto son los siguientes:

- 5 equipos de cómputo:
 - 4 equipos los cuales se usaran como clientes
 - 1 equipo con características para ser un servidor
- 1 Sistema operativo Windows XP
- 1 Sistema Operativo Windows Server 2003
- Visual Basic.Net 2005
- SQL Express 2005
- 3 impresoras láser
- 1 impresora matricial
- 5 Usuarios

Implementación de la Red y el Sitio Web

Para la implementación de la red local es necesario:

- 1 Switch 8 Pts.
- 1 Caja de cable UTP Cat-5e de 308mts
- 12 conectores RJ-45
- 100mts de Tubos Conduit
- 4 Canaletas de 1" de ancho
- 1 Canaleta de 2" de ancho

Para la implementación del sitio Web es necesario:

- Un Servidor
- Un Nombre de dominio DNS (Domain Name Service)
- Un Proveedor de Internet ISP (Internet Service Provider)

1. Cálculo de la Viabilidad Técnico Operativo

1.1 Viabilidad Técnica

La empresa Sáenz Bloques Adoquines y Mas..., actualmente no cuenta con equipo de cómputo, es necesario que la empresa adquiera equipos en los cuales el sistema necesita para funcionar.

En esta viabilidad vamos a implementar los recursos necesarios para el funcionamiento del proyecto

Equipo para utilizarlo como Servidor		
Alternativa	Recursos	Características
1	Hardware	Computadora Clon Procesador Intel Core 2 Duo 2.0Ghz socket 775 Case ATX 450W Memoria Ram DDR2 1GB PC 533 Tarjeta Madre Asrock 775 Disco Duro 160GB RPSM 7200 Rev. CD ROM 52x Interno Floppy Drive 1.44MB Teclado PS/2 Mouse Óptico PS/2 Monitor 17" CRT SVGA Parlantes 240W Pad Mouse
1	Software	Visual Basic.Net 2005 SQL Express 2005 Windows Server 2003 SP2

Equipos para utilizarlos como Clientes		
Alternativa	Recursos	Características
4	Hardware	Computadora Clon Procesador Intel Celeron 3.0Ghz socket 775 Case ATX 450W Memoria Ram DDR2 512MB PC 533 Tarjeta madre Asrock 775 Disco Duro 80GB RPSM 7200 Rev. CD Rom 52x Floppy Drive 1.44MB Teclado PS/2 Mouse Óptico PS/2 Monitor 17" CRT SVGA Parlantes 240W Pad Mouse
1	Software	Visual Basic.Net 2005 SQL Express 2005 Windows XP SP2
Accesorios Para Impresión de documento		
Alternativa	Recursos	Características
2	Periféricos	Impresora LasserJet HP 1018
1		Impresora Matricial LX 300+II EPSON
Accesorios Para la Seguridad de los Equipos		
Alternativa	Recursos	Características
5	Periféricos	UPS 500VA Forza Con Regulador de Voltaje

Equipos y Herramientas para el Cableado Estructurado	
Cantidad	Características
1	Caja de Cable UTP Cat-5e 305Mts
18	Conectores RJ 45
1	Switch 18 Puertos 10/100
50Mts	Tubo Convit
3	Curvas Tuvo conduit
10Mts	Canaleta de 1" ½
4Mts	Canaletas 2" ½
Equipo Necesario para el buen funcionamiento y desempeño de las Computadoras	
Cantidad	Características
1	Aire Acondicionado Split 46000 BTU
1	Aire Acondicionado de Ventana 46000 BTU
Servicios Necesarios para Internet y Sitio Web	
Cantidad	Servicios
1	Proveedor de Servicios de Internet (ISP)
1	Servicio de Nombre de Dominio (DNS)

1.2 Viabilidad Económica

Aquí se valorará el costo económico del Sistema (SIVECOIN) para la ejecución del proyecto

Inversión fija para el funcionamiento del Sistema (SIVECOIN) y Sitio Web			
	Cantidad	Costo Unitario	Costo Total
Equipos			
Equipo de Computo Servidor	1	\$578	\$578.00
Equipo de Computo Cliente	4	\$417.50	\$1670.00
Impresora Matricial	1	\$245	\$245.00
Impresora Láser	3	\$202.5	\$337.05
UPS	5	\$50	\$250.00
Personal			
Encargado del Soporte Informático	1	\$150	\$150.00
Capacitación del Personal	5	\$30	\$150.00
Muebles y Equipos de Oficina			
Aire acondicionado de Ventana	1	\$542.8	\$542.80
Aire Acondicionado Split	1	\$832.6	\$832.60
Instalación de Aire Acondicionado	2	\$529	\$529.00
Rema de Papel Bond	2	\$2.60	\$5.98
Caja de lapiceros	1	\$2.60	\$2.60
Caja de fastener	1	\$1.80	\$1.80
Caja de fólder	1	\$6,50	\$6,50
Perforadora	1	\$8,05	\$8,05
Sello de Hule (Cancelado y Entregado)	2	\$5,75	\$11.50
Engrapadora	3	\$7.5	\$7.50
Saca grapa	1	\$3.4	\$3.40
Caja de Grapas	3	\$1.00	\$1.00
Software del Sistema			
Programas			
Windows Server 2003	1	\$2.5	\$2.50
Windows XP 2003	1	\$2.5	\$2.50
SQL Express 2005	1	\$2.5	\$2.50
Visual Studio Ne 2005	1	\$2.5	\$2.50
Cds. Media	1	\$1.5	\$1.50
Estimación del Software	1	\$3858.55	\$4131.3
Total			\$9475.58

	Cantidad	Costo Unitario	Costo Total
Presupuesto para la Red Local e Internet			
Red Local			
Hardware			
Equipo de Computo Servidor	1		
Software			
Windows Server 2003	1		
Internet Explorer	1		
Materiales			
Switch 8 pts.	1	\$25	\$25
Caja Cable UTP	1	\$98	\$98
Conectores RJ-45	18	\$0.28	\$5.18
Tuvo Conduit	50Mts	\$0.63	\$31.62
Curvas Conduit	3	\$0.28	\$0.84
Internet			
Cuota Mensual 256K	1	\$34.49	\$34.49
Costo de Suscripción Internet	1	\$57.49	\$57.49
Nombre de Dominio (DNS)		\$57.50	\$57.50
Total			\$310.12

Inversiones Fijas + Presupuesto para la red = Costo total del proyecto

\$9475.58 + \$310.12 = \$9785,7

1.3 Viabilidad Operativa

Actualmente la Empresa Sáenz Bloques Adoquines y Mas..., no tiene un sistema de computo, todo se lleva por escrito en un cuaderno, el motivo de implementar un sistema y un sitio web en la empresa, es para agilizar las operaciones internas y para poder captar más clientes a través del Internet.

El Sistema SISCOIN Va a Operar en el área de Gerencia como servidor, y en el área de Caja, bodega, Producción y venta como cliente, para que los responsables de cada área introduzcan datos para el funcionamiento adecuado de la empresa. El personal será capacitado para que puedan interactuar con el sistema.

Cada Usuario tendrá un nombre de usuario y una contraseña, la cual le permitirá acceder al sistema, dependiendo del cargo del mismo, el sistema le restringirá acceso a funciones que no les son permitidas.

Este sistema ayudará a tener un mejor control del inventario tanto de materia prima como de producto final, también se podrán imprimir reportes de las entradas y salidas de los productos y materia prima. Este Sistema llevara un control de las ventas y compras realizadas dentro y fuera del local (On Line), también se podrán imprimir reportes de todas las facturas de ventas y compras.

1.4 Viabilidad Legal

Al finalizar con el proyecto, elaboraremos un contrato legal donde tanto el cliente como el gestor del proyecto firmaran, en el cual se establecerán normas para la viabilidad Legal de este.

Contrato Legal

- El lapso de tiempo para la finalización del software según la planificación temporal es de 6 meses como mínimo,
- Se debe de tomar en cuenta que la planificación temporal solo es un estimado del tiempo que se pueda llevar para la finalización del proyecto, no es 100% seguro, ya que pueden existir factores que afecten a la planificación temporal como por ejemplo, recursos monetarios, falta de tiempo, atraso con el pago, atraso en la revisión de las diferentes etapas de diseño del proyecto.
- Las exenciones de impuestos correrán a cuenta del cliente
- El software y código fuente, serán propiedad de los diseñadores del mismo, no serán permitidas las copias ya que se estará infligiendo la ley de derechos del autor.
- Se le facilitara una copia del software a la empresa
- La forma de pago será de tres cuotas, los cuales se dividirán de la siguiente manera:
 - Primera cuota de 35% será al inicio del diseño del software
 - Segunda cuota será de 35% a la mitad del tiempo estipulado de la entrega del software
 - Tercera cuota será de 30% al concluir el proyecto.
- Todos los ajustes exigidos por el cliente dentro del tiempo estipulado para la realización del sistema y sitio web serán aceptados, si se piden ajustes del sistema y sitio web después del tiempo estipulado, estos ajustes tendrán un valor extra.
- La garantía del software y sitio web no será mayor de 3 meses después de haber sido instalado en cada ordenador, si se da alguna falla en ese lapso de tiempo se cubrirá garantía sin ningún costo.
- Si el software y sitio web se daña por falta de mantenimiento de su equipo, en el lapso de tiempo de garantía, esta no se le cubrirá.

- El sistema se instalará en un equipo de cómputo sobre la plataforma de Windows XP como cliente y como servidor en Windows server 2003, el sitio web en el servidor, la cual trabajara sin licencia alguna.
- Si Microsoft el Dueño de la plataforma Windows XP y Windows Server2003, al mismo tiempo servirá como host del sitio web de la empresa, si Microsoft Dueño de estas plataformas demanda a la empresa **Sáenz Bloques Adoquines y Mas...**, por la adquisición de su software ilegalmente, la empresa **Sáenz Bloques Adoquines y Mas...**, nos libera de toda responsabilidad. Ya que fue su decisión haber adquirido este software de manera ilegal.
- A este sistema se le podrán hacer actualizaciones esto tendrá un costo .extra

Gestor del Proyecto

Cliente

2. Gestión del Proyecto

2.1 Selección del Personal

El personal del proyecto está formado por 4 egresados de la carrera de Ingeniería de Sistemas de la UCC.

2.2 Funciones del Personal

En este proyecto, todos desempeñan la mismas funciones, aunque existe un coordinador o Gestor, quien se encarga de que todo esté bien en la programación del sistema, de acuerdo a la información que nos facilitó el cliente por medio de una entrevista y los ajustes solicitados por el mismo en el momento de mostrarle los progresos del desarrollo del sistema.

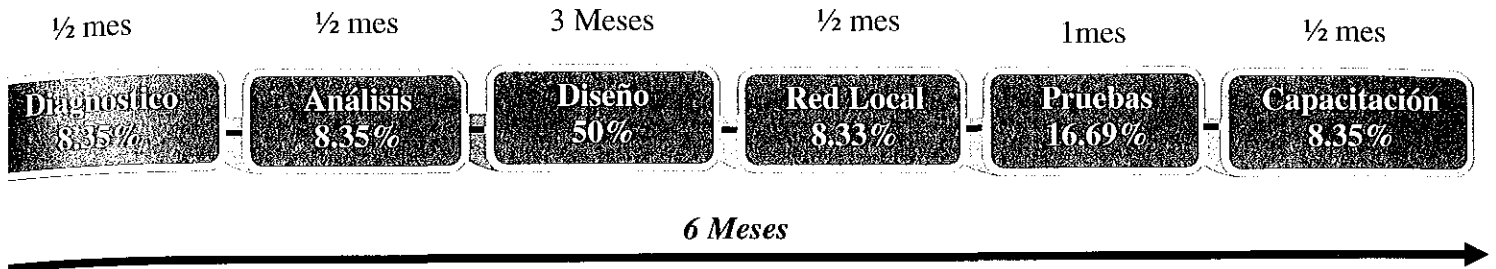
La función principal del personal, es la codificación tanto del sistema como del Sitio Web de la empresa.

2.3 Problema

En la actualidad la empresa Sáenz Bloques Adoquines y Más., no cuenta con un sistema de cómputo, todo los controles de producción se llevan en un cuaderno donde se apuntan las entradas y salidas de la materia prima y producto final, y los formatos de facturas se llenan manualmente, esto genera pérdida de tiempo para la empresa y el personal, lo que se traduce a perdida de dinero, ya que este tiempo puede ser utilizado para hacer producir más dinero a la empresa

2.4 Proceso (Etapas)

Distribución de la utilización de recursos en el ciclo de vida



2.5 Selección del Modelo de Proceso

Nuestro Método es el “**Basado en la experiencia**”. En este método utilizamos la **Distribución de la utilización de recursos en el ciclo de vida**, aquí se distribuyen las etapas para el desarrollo del proyecto en masa porcentual dependiendo del tiempo que se estipulo para finalizar el proyecto, en este caso es de 6 meses sin prorroga.

3. Ámbito

En la empresa se instalará un sistema de Ventas-Compras y Control de inventario, del cual harán uso el gerente de la empresa, la cajera, el vendedor, el encargado de bodega y el jefe de producción. El sistema le servirá como una herramienta para facilitarle el trabajo en sus funciones diarias, ya no se llevara el control en un cuaderno, y tampoco se llenaran los formatos de facturas manualmente, ya que el sistema, tendrá una base de datos donde se almacenaran los registros, tanto de la compras como de las ventas, y registros de entradas y salidas de materia prima y producto final, se mostraran los datos mediante reportes ya sean visuales o impresos en papel.

Esta empresa también se instalara un sitio web, donde los clientes podrán ver el catalogo de los productos así mismo se visualizaran los precios de cada uno de los productos.

Se tiene programado que desde el sitio web se puedan realizar compras con tarjeta de crédito o debito, pero por falta de tiempo esto se realizara mas adelante como una segunda etapa del sitio web.

4. Estimaciones de Costos del Proyecto

4.1 Estimación del costo del Sistema y Sitio Web

Para poder estimar el valor del Sistema (SIVICOIN) y sitio Web, es necesario calcular el valor de la mano de obra de los desarrolladores del Sistema y sitio Web, y los Gastos Operativos.

Un salario regular para un profesional recién graduado, es de aproximadamente \$300 mensuales, entonces al día su salario seria de \$1.5; Para poder estimar el valor de la mano de obra de los desarrolladores, se debe de hacer lo siguiente:

Se multiplica las horas diarias trabajadas en el proyecto, por los días trabajados en la semana por las semanas que hay en un mes, y por ultimo, por los meses dedicados a la realización de este, luego esto se multiplica por el valor que un profesional recién graduado gana al día y luego se multiplica por el numero de desarrolladores del sistema, por último se se suman los gastos operativos que se realizaron para poder realizar el proyecto.

Para la realización del Sistema SIVCOIN y Sitio Web, le dedicamos 5 horas diarias 6 días a la semana por 6 meses

Calculo de la mano de Obra

$((5\text{Horas} \times 5 \text{ Días}) \times 4 \text{ Semanas} \times 6 \text{ Meses}) = 600 \text{ Horas}$

$600 \text{ Horas} \times \$1.5 = \$900 \text{ Salario Individual}$

$\$900 \times 4 \text{ Desarrolladores} = \3600

Mano de Obra = \$3600

Gastos Operativos				
	Días	5 días a la Semana	1 mes	6 meses (Total)
Transporte	\$1.7	\$8.5	\$34	\$204
Servicios Básicos (aquí se incluye el precio básico de cada factura de los servicios)				
Agua			\$3,5	\$21
Luz			\$5.35	\$32.1
Internet			\$35	\$210
Teléfono			\$10.7	\$64.2
Total				\$531.3

Costo del Software = Mano de Obra + Gastos Operativos

$\$3600 + \$531.3 = \$4131.3$

Costo del Software = \$4131.3

4.2 Estimación del costo total del Proyecto

Para poder estimar el costo total del proyecto se tuvo que solicitar varias cotizaciones de los precios de los recursos, accesorios y herramientas necesarias en diferentes tiendas de la ciudad de Managua, para poder implementar el sistema en la empresa **Sáenz Bloques Adoquines y Mas...**

Para poder realizar la estimación, se tuvieron que escoger las cotizaciones con precios accesibles.

En la viabilidad económica que se realizó refleja que el costo del proyecto es de: **\$9785,7**

Costo total del Proyecto = \$9785,7

5. Análisis de riesgo

No es posible evitar todos los riesgos, por eso recurrimos a un plan de contingencia para que nos permita tomar decisiones de una manera eficaz y controlada que eviten los riesgos

Análisis de Riesgo	
Descripción del Riesgo	Plan de Contingencia
Fallas de energía (voltajes inestables), corto circuito, corte de energía	Utilizar equipos de respaldos tales como: baterías, estabilizadores y plantas eléctricas
Daños o fallos en el equipo de Hardware	Los miembros de equipo podrán reparar el equipo o sustituir piezas dañadas inmediatamente
Infección por virus informático	Utilizar programas antivirus actualizado
Daños al Software ya sea por virus, problemas de disco duro o errores del sistema operativo	Realizar copias actualizadas de respaldo del software a medida que se va desarrollando y tomar medidas de seguridad
Nuevos requerimientos del Sistema	Modificaciones después del tiempo acordado tendrá un costo extra, Modificaciones dentro del lapso del tiempo el costo siempre será el mismo acordado, planificación y presentar nueva propuesta
Ausencia de los miembros del equipo por enfermedad	El equipo debe de asumir la responsabilidad de seguir con el Proyecto
Retrazo en el desarrollo del sistema	Crear un plan de trabajo para lograr que todas las tareas se logren en sus fechas previstas
Mala interpretación de los requerimientos del sistema	Realizar explicaciones concretas y detalladas al equipo
Incendios	Uso de extinguidotes y mantener siempre un respaldo en casa de habitación de cada integrante del equipo desarrollador
Terremotos e Inundaciones	Crear una copia de seguridad y guardarla en un servidor Web facilitado por ANSAP.ORG y actualizarlo a medida que se desarrolle el sistema

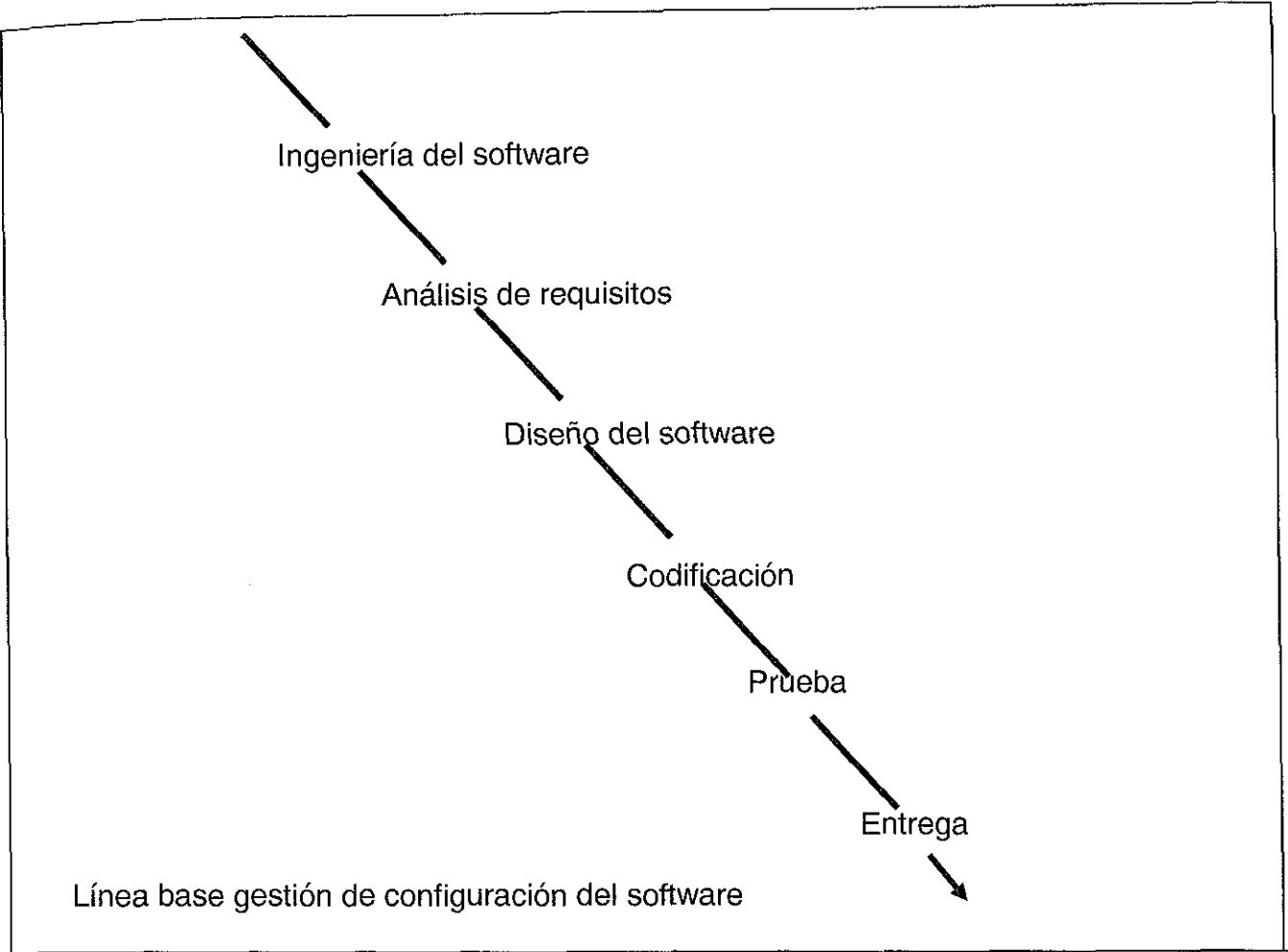
6. Planificación Temporal

Para la planificación temporal, se utilizó el diagrama de Gannt, en el cual se delimita el tiempo en que se realizaron las tareas de la aplicación (ver en anexo *Planificación temporal*)

7. Gestión de la Configuración del proyecto

Es un conjunto de actividades desarrolladas para gestionar los cambios a lo largo del ciclo de vida del software, es una actividad de garantía de calidad del software que se aplica en todas las fases del proceso de ingeniería del software.

Se define una línea como una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios



VII. Fase de Desarrollo

Para poder modelar El diseño Conceptual y el diseño lógico se uso una herramienta para la modelación. La herramienta es un programa para elaborar diagramas UML (Lenguaje Unificado de Modelado)

UML es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.

1. Diseño Conceptual

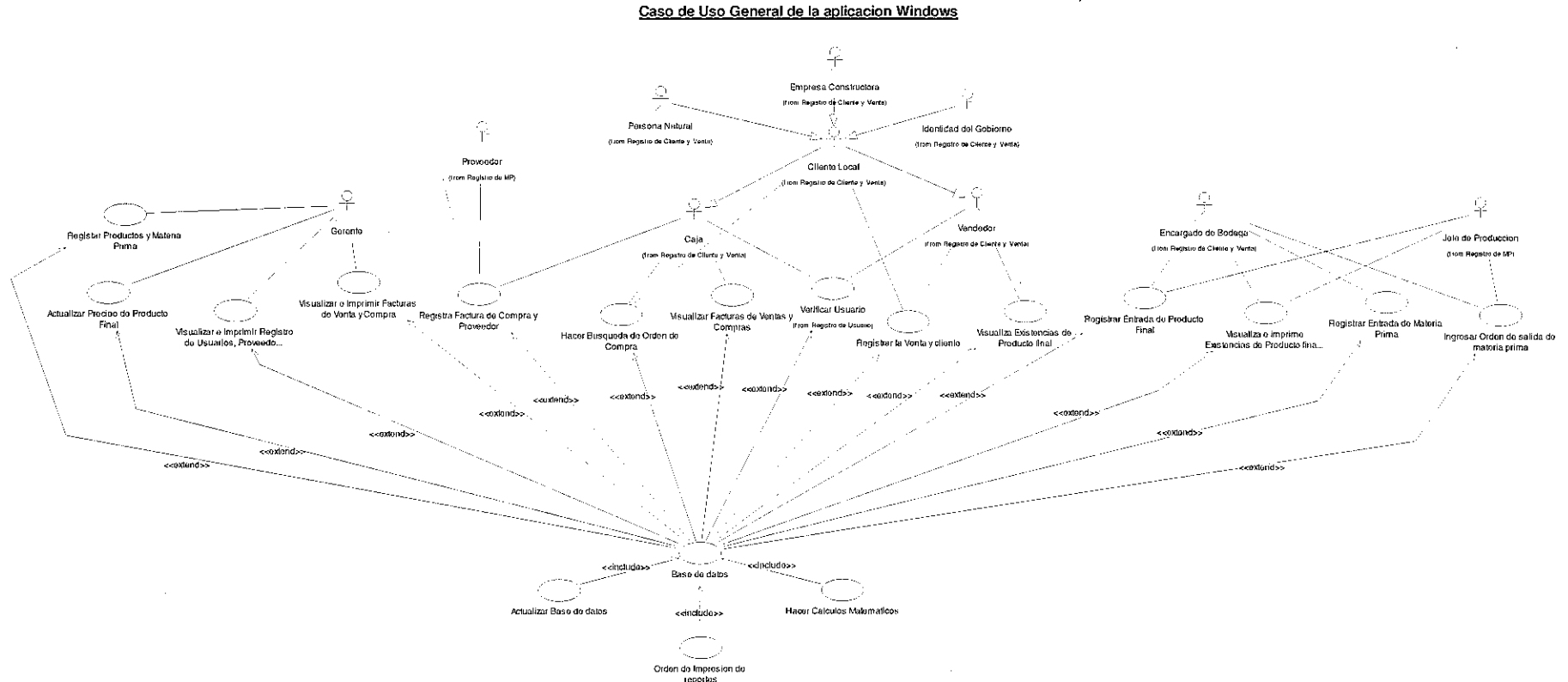
1.1 Descripción del proyecto

Dentro del proyecto se diseñará una aplicación Windows para el control de las facturas de venta y compra, y un control del inventario de materia prima y producto final. Esta aplicación se instalara en 5 equipos de cómputo de los cuales uno llevara la función de servidor de la red. Este equipo tendrá instalado Windows Server 2003, SQL Express y Visual Studio.Net 2005, en este equipo estará instalada la base de datos del sistema .los otros cuatro equipos de cómputo jugarán el papel de Clientes. Estos equipos tendrán instalado Windows XP 2003, SQL Express y Visual Studio.Net 2005.

También se diseñara una Aplicación Web, donde se podrán mostrar los diferentes productos y sus precios, y se podrán hacer compras a través del Internet. El sitio Web estará alojado en el servidor de la empresa y estará conectada a la misma base de datos del Sistema

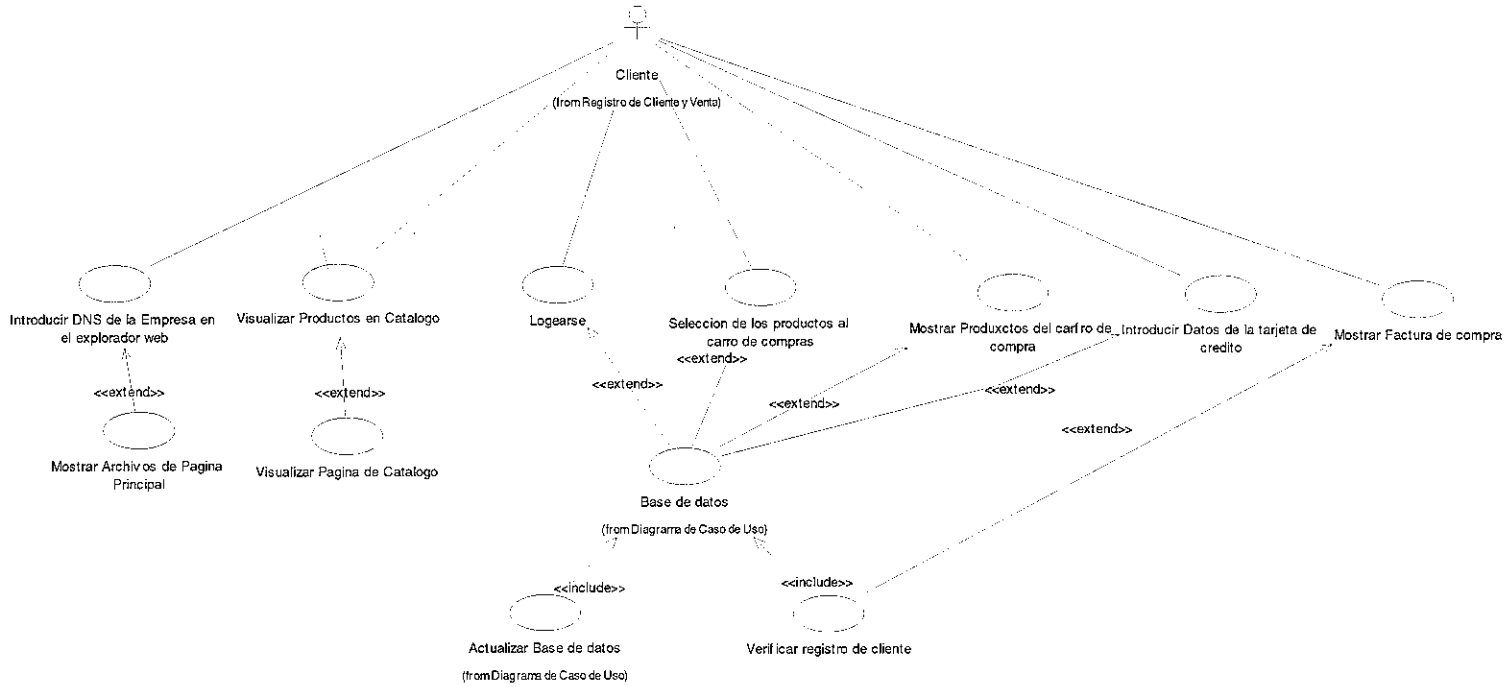
1.2 Diagrama de caso de uso general

1.2.1 Aplicación Windows



1.2.2 Aplicación Web

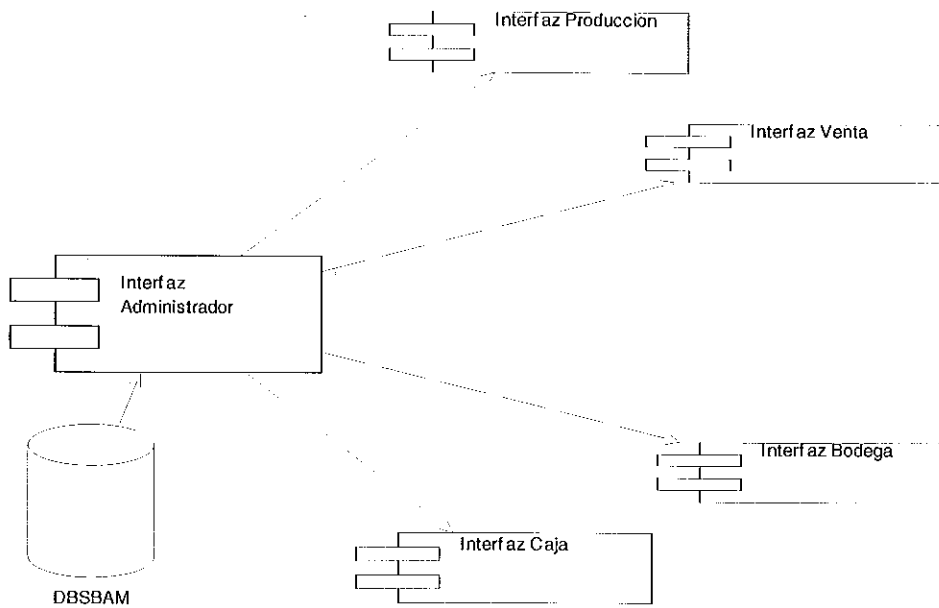
Caso de Uso General de la aplicacion Web



1.3 Diagrama de componentes

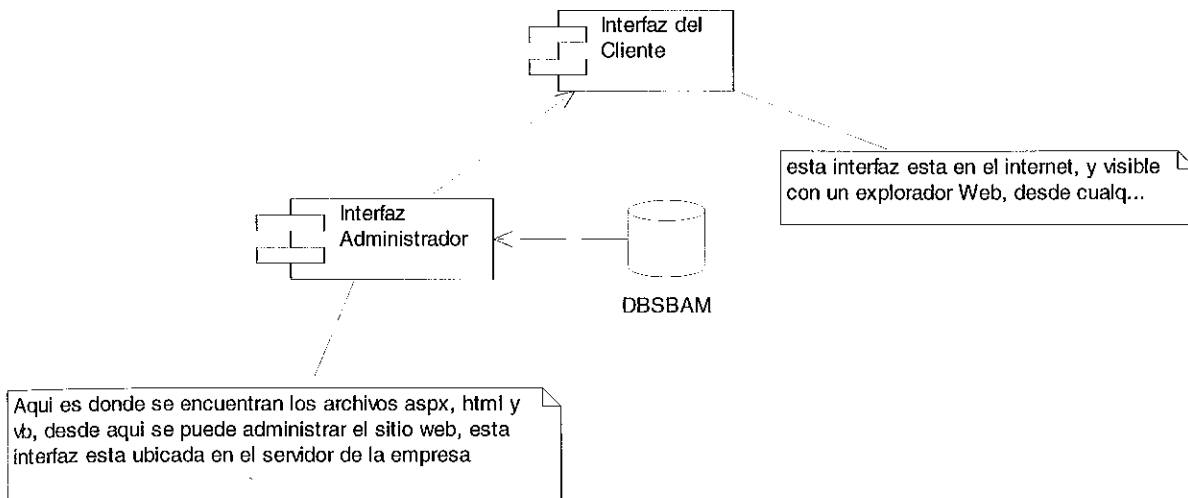
1.3.1 Aplicación Windows

Diagrama de Componente de la aplicacion Windows



1.3.2 Aplicación Web

Diagrama de Componente del Sitio Web

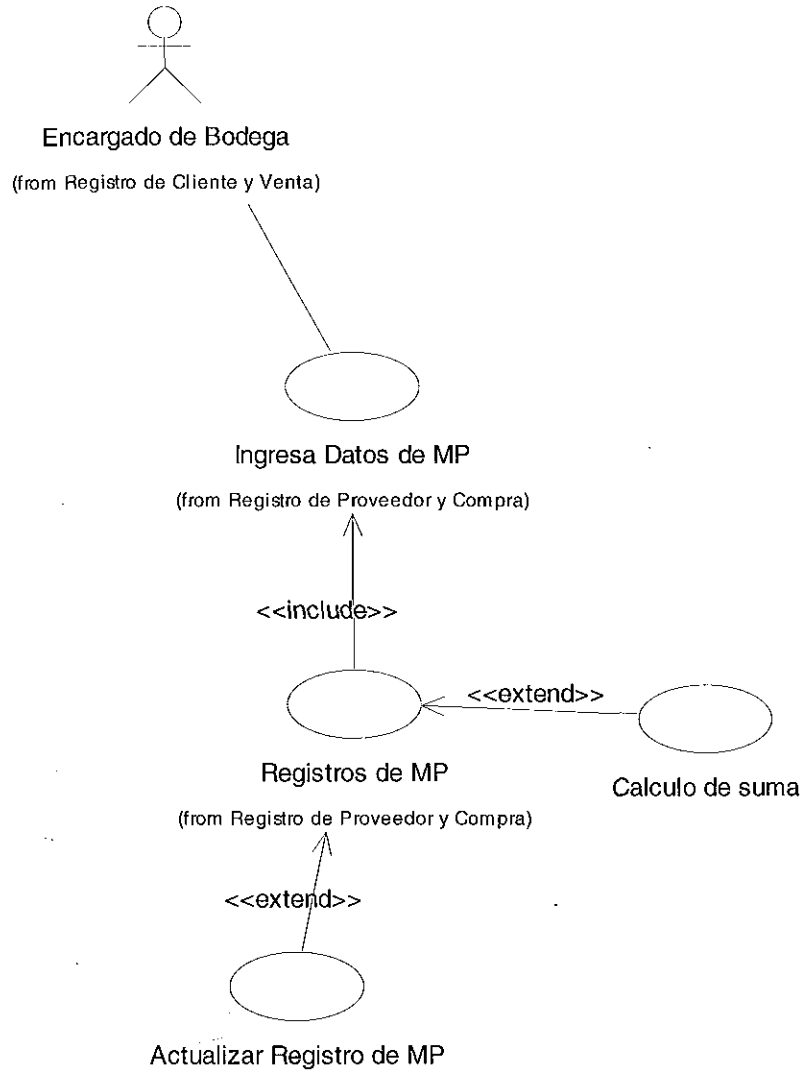


2. Diseño lógico

2.1 Caso de uso particulares y sus especificaciones

2.1.1 Aplicación Windows

Registro de Materia Prima



Caso de Uso: Registro de Materia Prima

Objetivo:

Registrar toda la materia Prima en una Tabla

Llevar Un Control del registro de toda la materia Prima

Actor:

- Encargado de Bodega

Precondiciones: El Jefe de área debe de haber hecho el pedido al proveedor

El proveedor debe de entregar la materia prima al jefe de área, para ser almacenada en bodega

Pos condiciones: Debe quedar registrada la materia prima en la tabla de Materia Prima

de la base de datos del Sistema

Hay que darle salida a la Materia prima para la producción de Producto Final

Flujo Básico:

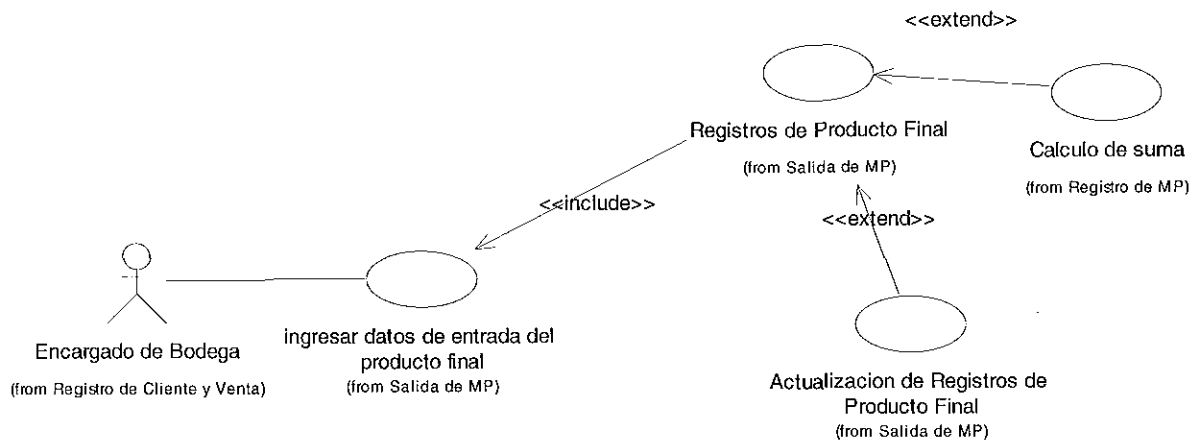
<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
<i>1. El Encargado de Bodega introduce datos de Materia Prima</i>	<i>2. Registrar materia Prima en registros de Materia Prima</i>
	<i>3. Actualizar registros de Materia Prima de la base de datos del Sistema</i>

Flujo Alternativo: No Hay

Requisitos Especiales: No Hay.

Frecuencia: Cada Vez Que el bodeguero reciba Materia Prima del Jefe de Producción de Materia Prima

Registro de Producto Final



Caso de Uso: Registro de producto final

Objetivo: Llevar el control de Entrada del Producto Final para Su Venta

Actor Principal:

- Encargado de bodega

Precondiciones: Debe de haber existencia de materia prima para la salida de este al área de producción para la elaboración del Producto final.

Pos condiciones: El Producto Final Debe de Ser Puesto a la Venta

Flujo Básico:

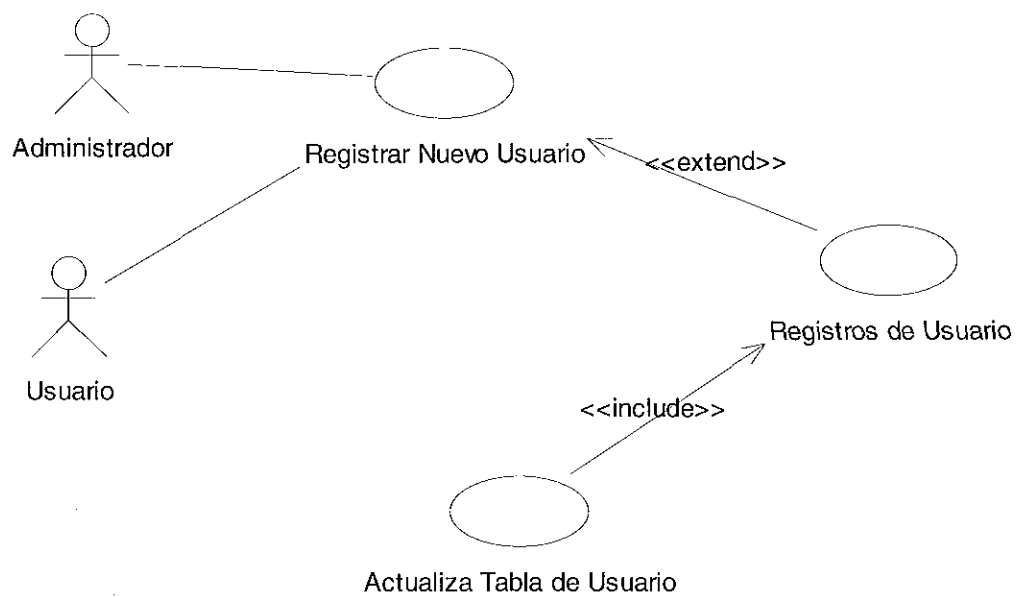
<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1. Encargada de Bodega Ingresar datos de Entrada Del Producto Final	2.Registrar el Producto Final en la tabla de Producto Final
	3. Actualizar Registros del Producto Final

Flujo Alternativo: No Hay

Requisitos Especiales: No Hay.

Frecuencia: Siempre que exista una nueva producción de Producto Final

Registrar Usuario



Caso de Uso: Registrar Usuario

Objetivo: Llevar un Registro de Todos Los Usuarios del Sistema

Actor Principal:

- Administrador

Personal Involucrado e Intereses:

- Usuario

Precondiciones: El usuario Nuevo no debe de estar registrado en la tabla de Usuario de la Base de datos del Sistema

Post condiciones: El Nuevo usuario debe de quedar registrado en la tabla de Usuario de la Base de datos del Sistema

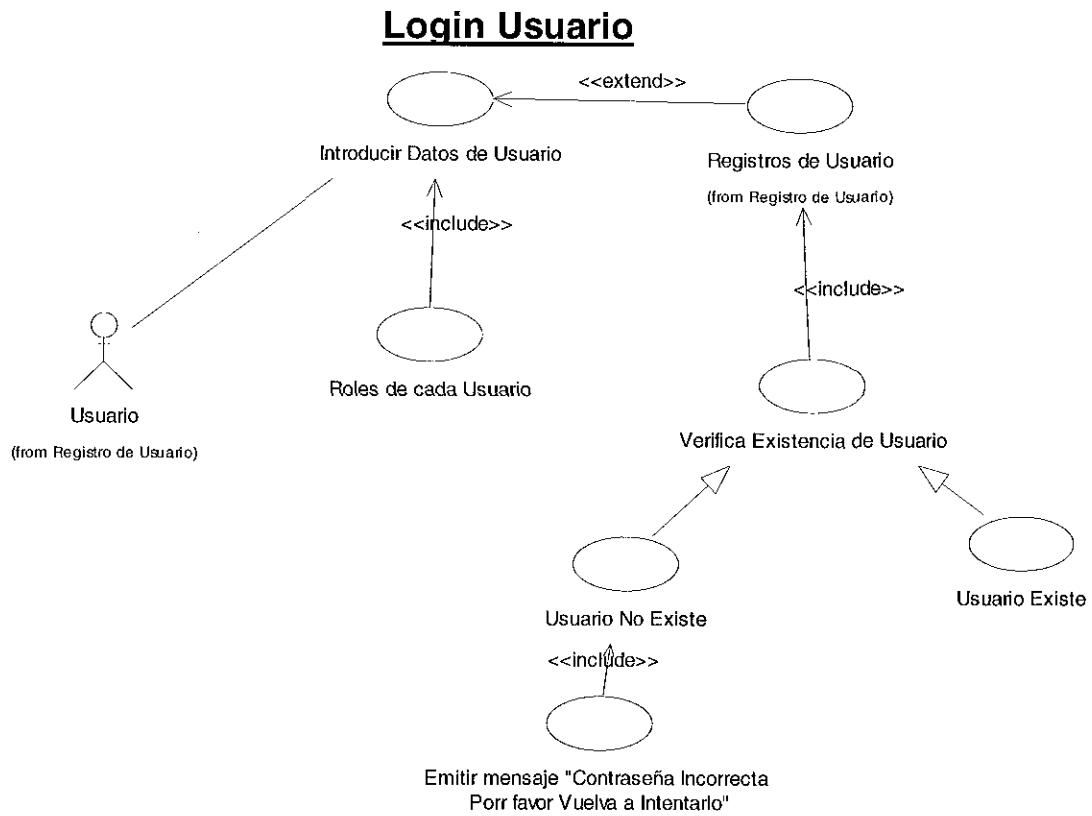
Flujo Básico:

<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sistema</i></u>
1. El Administrador Introduce datos del nuevo Usuario	2. Introducir datos en el registro de Usuarios
	3. Actualizar Registro de Usuario

Flujo Alternativo: No Hay

Requisitos Especiales: No Hay

Frecuencia: Cada vez que se quiera registrar a un nuevo usuario



Caso de Uso: Login Usuario

Objetivo: No permitir la entrada al sistema a Personas no autorizadas

Actor Principal:

- Usuario

Precondiciones: El usuario debe de estar registrado en la tabla de la base de datos de Usuario para poder entrar al sistema

Post condiciones: El usuario debe de poder hacer operaciones permitidas en el Sistema

Flujo Básico:

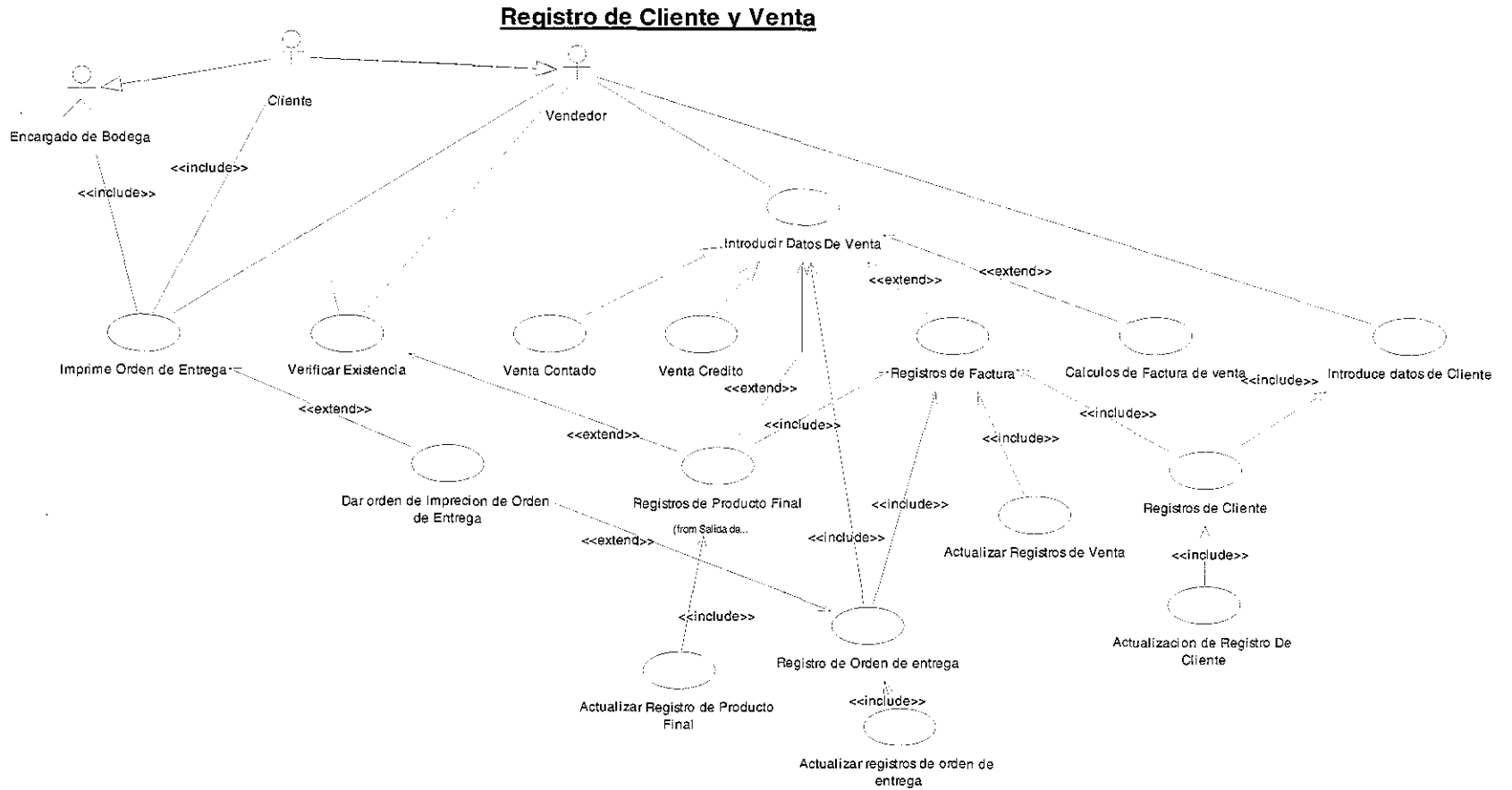
<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1.El Usuario Introduce Usuario y Contraseña	2. Verificar Si el usuario esta En registros del Sistema 3.Dar pase al usuario de poder trabajar en el sistema

Flujo Alternativo: 2. el sistema Verifica Si el Usuario Esta registrado en la tabla de Usuario de La base de datos del sistema:

Si el usuario existe da pase de entrada al sistema al usuario, si el usuario no existe el sistema manda un mensaje de "Usuario o Contraseña Incorrecta por favor introduzca bien su Usuario o Contraseña", a la tercera vez de introducir mal los campos el sistema automáticamente se cierra.

Requisitos Especiales: E Usuario debe de estar registrado para poder ingresar al sistema

Frecuencia: Cada vez que el sistema se ejecuta



Caso de Uso: Registro de Cliente y Venta

Objetivo: Llevar un Registro de Todos los Clientes y Ventas Realizadas

Actor Principal:

- Vendedor

Personal Involucrado e Intereses:

- Cliente
- Encargado de bodega

Precondiciones:

- Debe de haber un nuevo cliente para poder ser registrado
- Debe de existir PF o MP para poder Hacer una Venta

Pos condiciones: Debe de Realizarse una Búsqueda de Cliente en caja para que el cliente proceda a pagar el producto

Flujo Básico:

<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sistema</i></u>
<i>1. El Vendedor Verifica Existencia</i>	<i>2. Mostrar los registros del producto final de la base de datos</i>
<i>3. El Vendedor Introduce datos De Cliente</i>	<i>4. Introducir datos de Cliente en registro de Clientes de la Base de datos del Sistema</i>
	<i>5. Actualizar registro de Clientes de la Base de Datos del Sistema</i>
<i>6.El Vendedor Introduce datos de la Venta</i>	<i>7. Introducir Datos en registro en la tabla de la Factura e Venta de la base de datos del Sistema</i>

- | | |
|---|--|
| | 8. Actualizar registro de la factura de venta de la Base de datos Del Sistema |
| 9. El vendedor visualiza e Imprime la Orden de Entrega d PF | 10. Orden de Impresión de la Tabla de Orden de entrega de la Base de datos del Sistema |
| 11. El Vendedor Recibe Orden de Entrega de PF Impresa | |
-

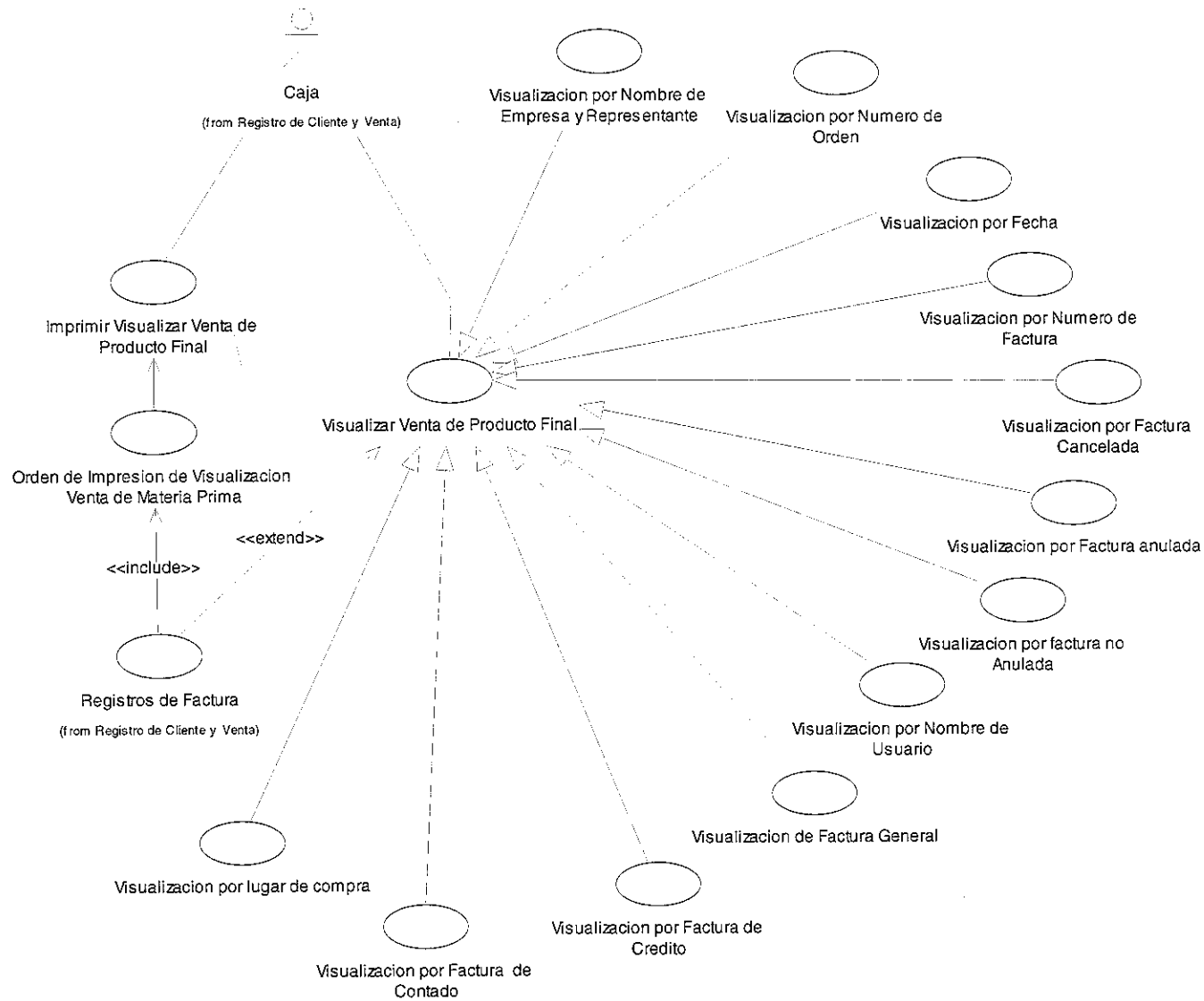
Flujo Alternativo: 3. El vendedor Introduce datos del Cliente:

Si el Cliente Existe en la tabla de Cliente, pasar a:

6. El Vendedor Introduce Datos de La Compra

Frecuencia: Cuando haya existencia de producto final

Visualizar e imprimir Reporte de Factura de Venta de Producto Final



Caso de Uso: Visualizar e Imprimir Reporte de Factura de Venta de PF

Objetivo: Visualizar e Imprimir reporte de Factura de Venta de Producto Final según la opción seleccionada

Actor Principal:

- Caja

Precondiciones:

Deben de Existir Registros de Compra

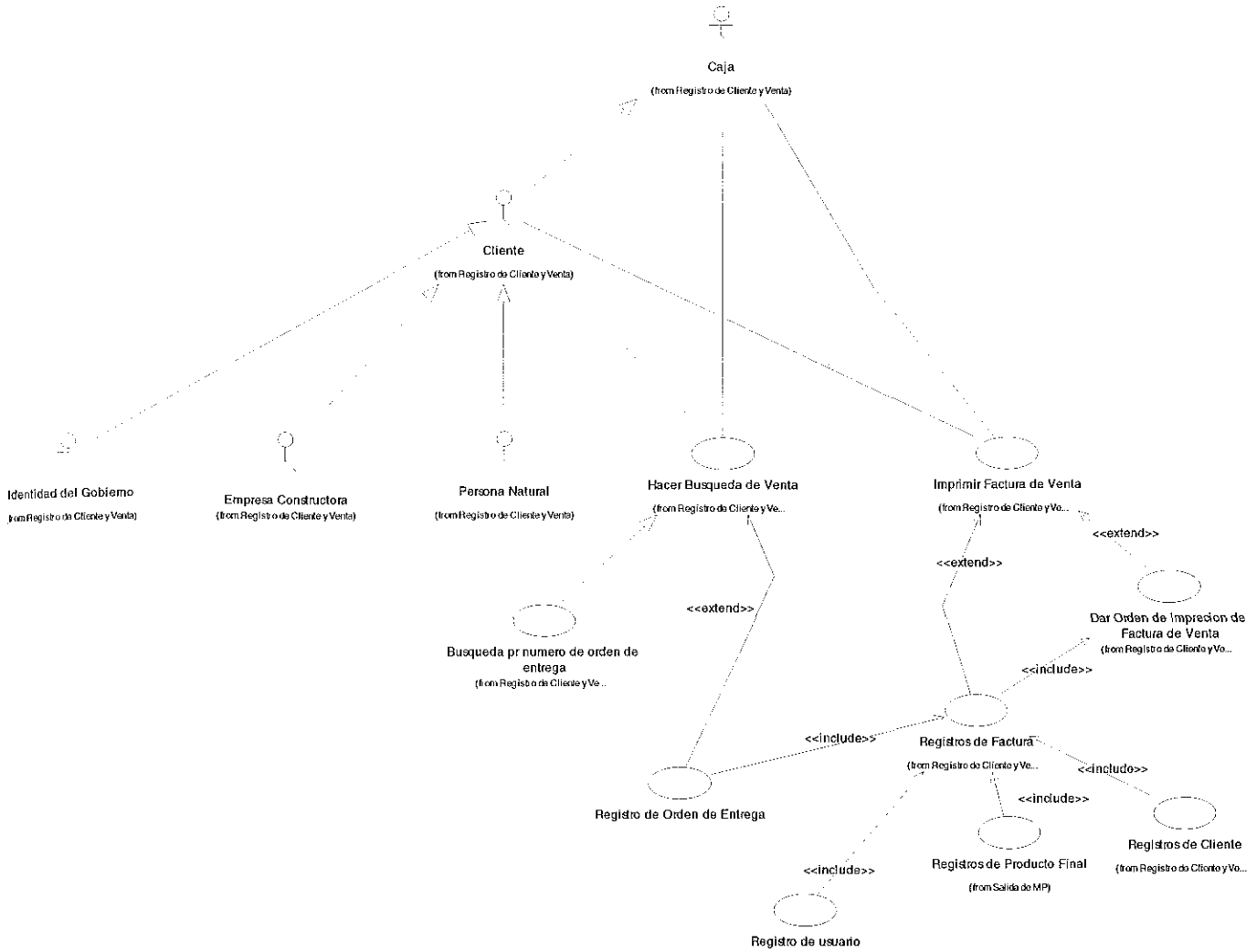
Pos condiciones: No Ha

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1. Caja Visualiza Reporte de Factura de Venta	2. Mostrar Registros de Tabla de Compra MP
3.Caja .Imprime reporte	4.Dar orden de impresión del reporte visualizado
5.Caja recibe impresión del reporte	

Flujo Alternativo: 2. El Sistema Muestra Registros de Tabla de Fatura de Venta, Según la condición seleccionada mediante un check box

Busqueda e Impresion de Factura de Venta del Producto Final



Caso de Uso: Búsqueda e Impresión de Factura de Venta

Objetivo: Hacer Búsqueda de la Venta e Imprimir Factura de Venta

Actor Principal:

- Caja

Personal Involucrado e intereses:

- Cliente

Precondiciones: Se debe de imprimir una orden de entrega para el cliente

Pos condiciones: Debe de Haber Producto Final en Bodega

Flujo Básico:

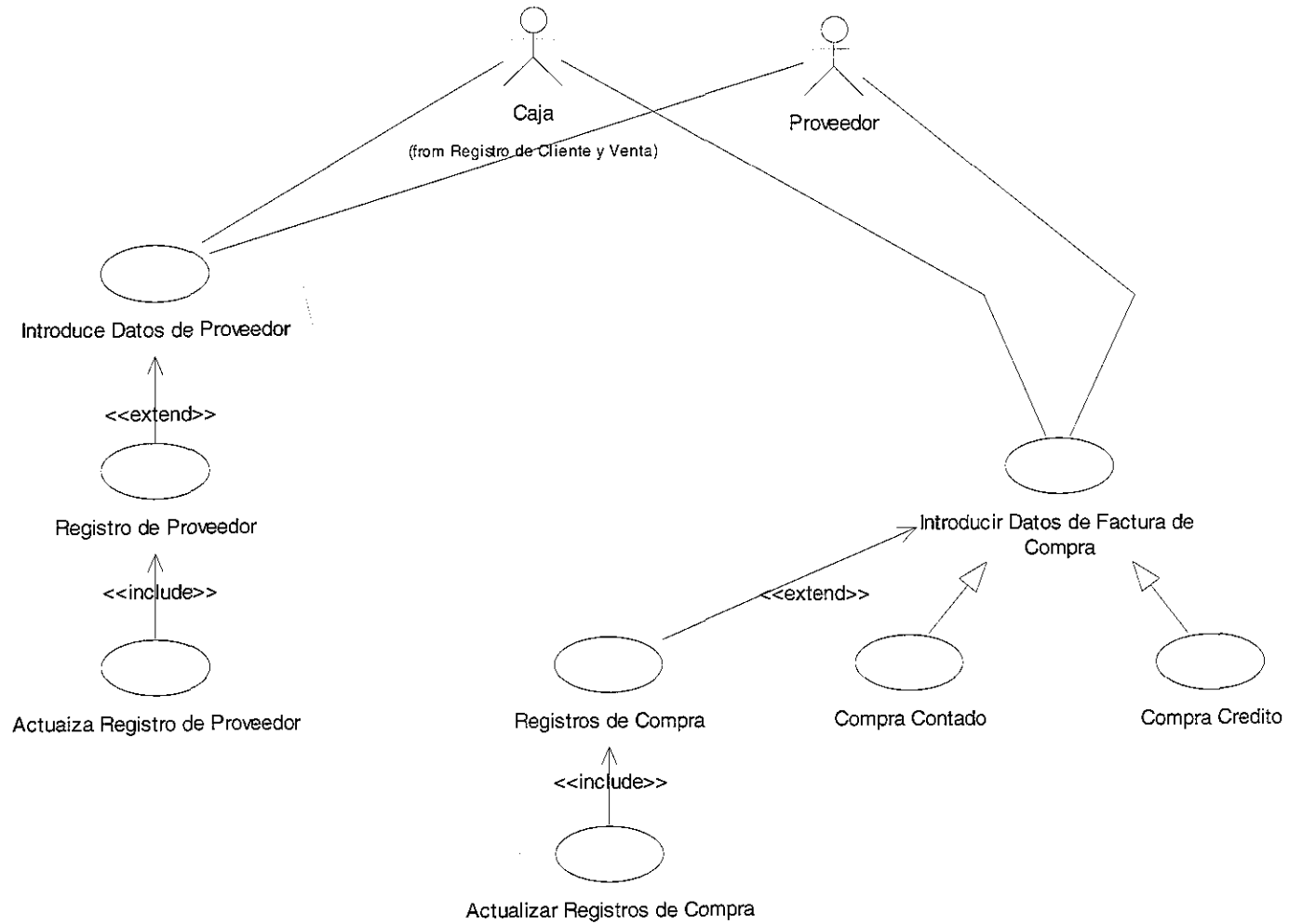
<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1. Caja Hace Búsqueda de Venta Por numero de orden de Entrega	2. Hacer búsqueda en el registro de Factura de venta de la base de datos del Sistema Por orden de entrega
	3. Mostrar Datos Relacionados l numero de orden de entrega
4. Caja Imprime Factura de Venta	5. Mandar Orden de Impresión de Registros de Factura de venta
6. caja recibe Factura de Venta Impresa	

Flujo Alternativo: No Hay

Requisitos Especiales: No Hay

Frecuencia: Cada vez que se imprima una Orden de Entrega

Registro de Proveedor y Compra



Caso de Uso: Registro de Proveedor y Compra

Objetivo: Llevar Un registro de todos los proveedores y compras que se realicen

Actor:

- Caja

Personal Involucrado e Intereses:

- Proveedor

Precondiciones: El Proveedor debe de haber entregado la Materia Prima solicitada al jefe de bodega

Pos condiciones: Debe quedar registrada todas las compras efectuadas y Debe quedar

Registrado los datos del proveedor

Flujo Básico:

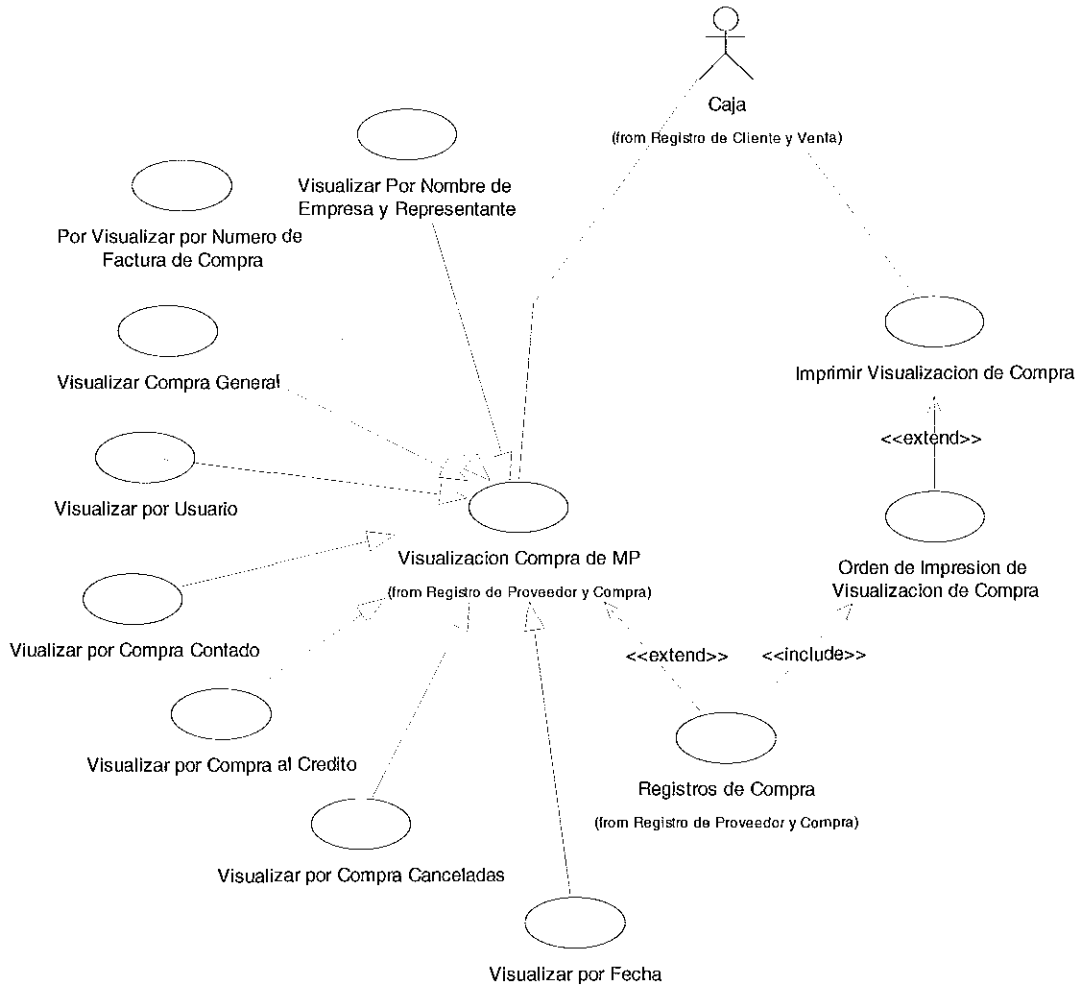
<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sistema</i></u>
1. Caja Introduce datos de Proveedor	2. Registrar al proveedor
	3. actualizar la Tabla del Proveedor
4. Caja Introduce Datos de Compra de MP	5. El Sistema Registrar Compra de Materia Prima
	6. Actualizar Registros de Compra de MP

Flujo Alternativo: 4. Caja introduce datos de Compra el cual puede ser Contado o al crédito (la tabla compra MP tendrá un campo con el nombre **Tipo** Donde se almacenara el tipo de compra MP ya sea crédito o Contado)

Requisitos Especiales: No Hay.

Frecuencia: Cada Vez Que se realice una entrega de Materia Prima

Visualizar e imprimir Reporte de Factura de Compra de Materia Prima



Caso de Uso: Visualizar e Imprimir Reporte de Factura de Materia Prima

Objetivo: Visualizar e Imprimir reporte de Factura de Compra de Materia Prima según la opción seleccionada

Actor Principal:

- Caja

Precondiciones:

Deben de Existir Registros de Compra

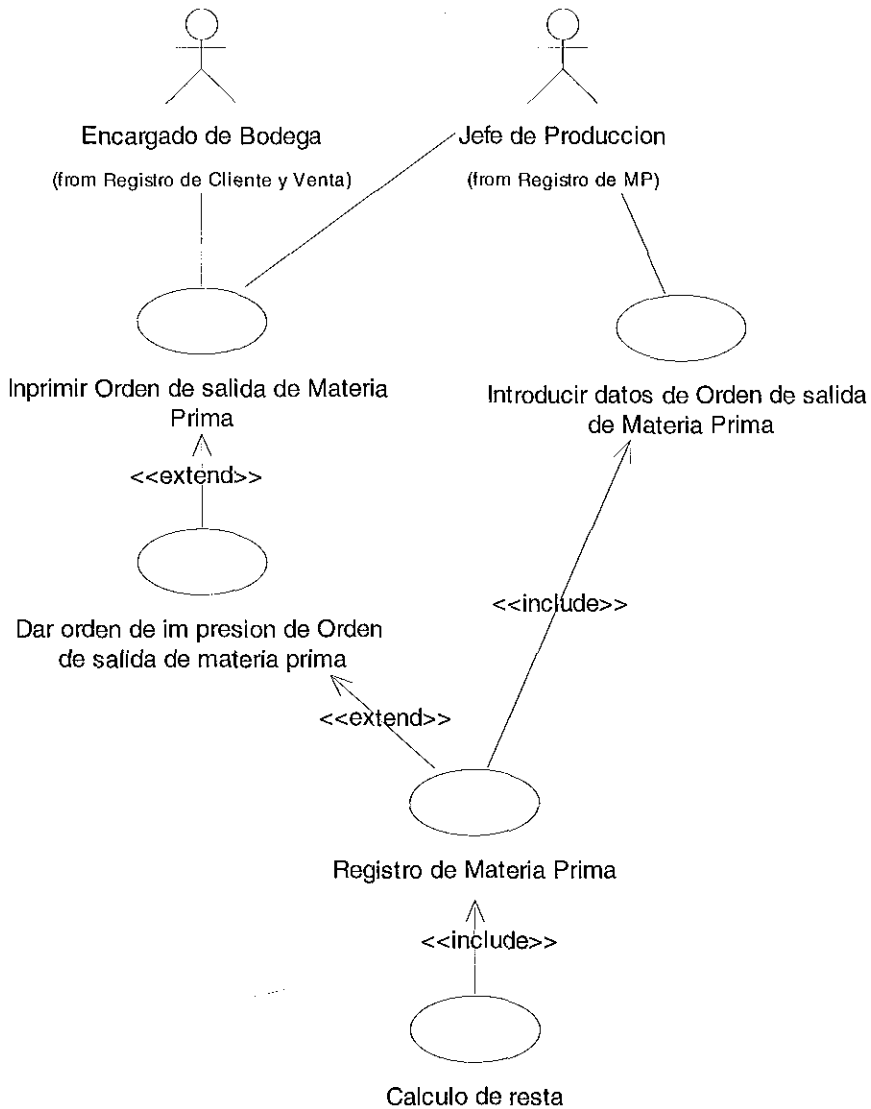
Pos condiciones: No Ha

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1. Caja Visualiza Reporte de Factura de Compra	2. Mostrar Registros de Tabla de Compra MP
3.Caja .Imprime reporte	4.Dar orden de impresión del reporte visualizado
5.Caja recibe impresión del reporte	

Flujo Alternativo: 2. El Sistema Muestra Registros de Tabla de Compra MP, Según la condición seleccionada mediante un check box

Registro e Impresion de Orden de Entrega de Materia Prima



Caso de Uso: Registro e Impresión de orden de entrega de Materia Prima

Objetivo: Ingresar datos para la salida de materia prima e impresión de orden de entrega de materia prima

Actor Principal:

Jefe de Producción

Personal Involucrado e Intereses:

Encargado de Bodega

Precondiciones:

Necesidad de Nueva producción

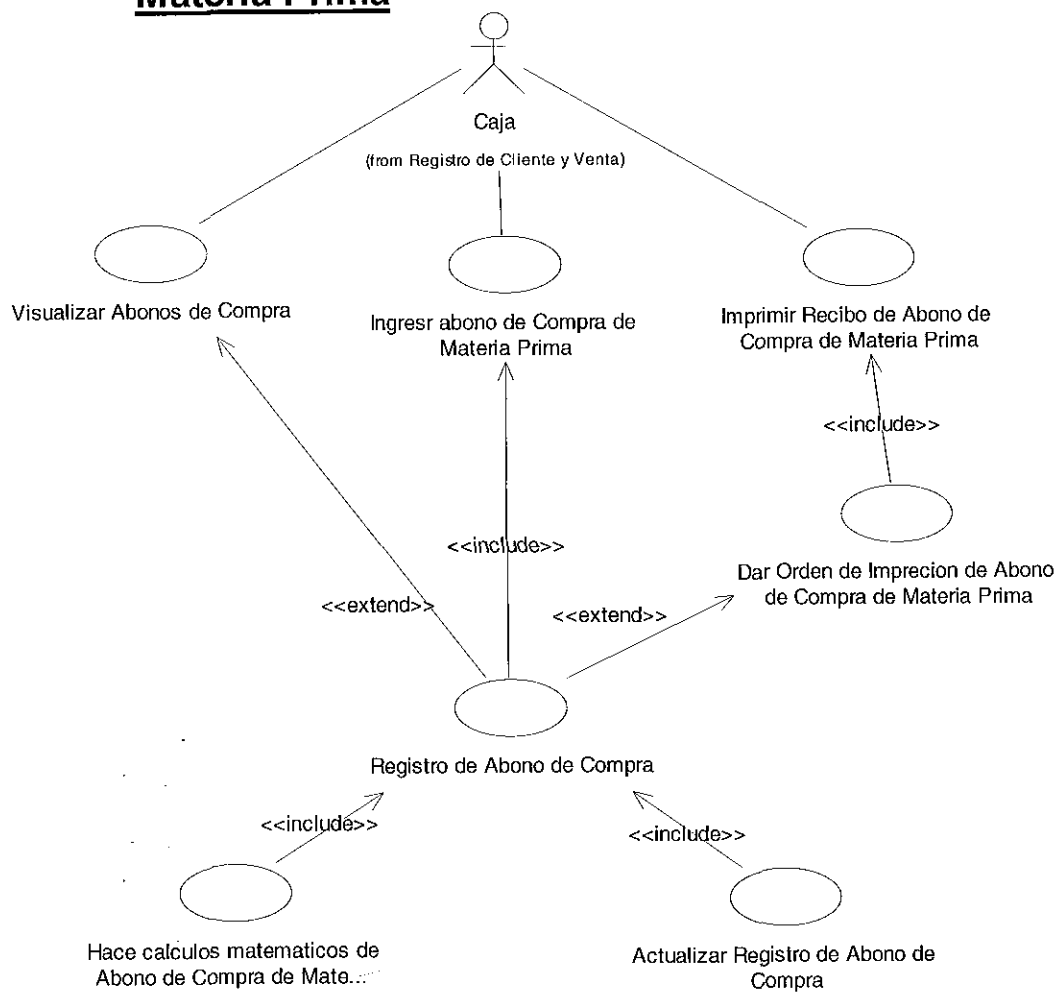
Deben de Existir Existencia de Materia Prima

Pos condiciones: No Ha

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1. <i>Jefe de Producción introduce datos de materia prima</i>	2. <i>Ingresar Datos a la tabla de Materia prima de la base de datos</i>
	3. <i>Hace calculo de resta de total de Materia prima existente</i>
4. <i>Jefe de Producción Imprime orden de Entrega de Materia Prima Encargado de bodega recibe impresión de orden de entrega de materia prima</i>	5. <i>Da orden de impresión del la orden de entrega en impresora de bodega</i>

Registrar e Imprimir Abono de Compra de Materia Prima



Caso de Uso: Registrar e Imprimir Abono de Compra de Materia Prima

Objetivo: Llevar Un registro de todos los abonos realizado por parte de la empresa a los proveedores

Actor:

- Caja

Precondiciones: Deben de existir compras al crédito

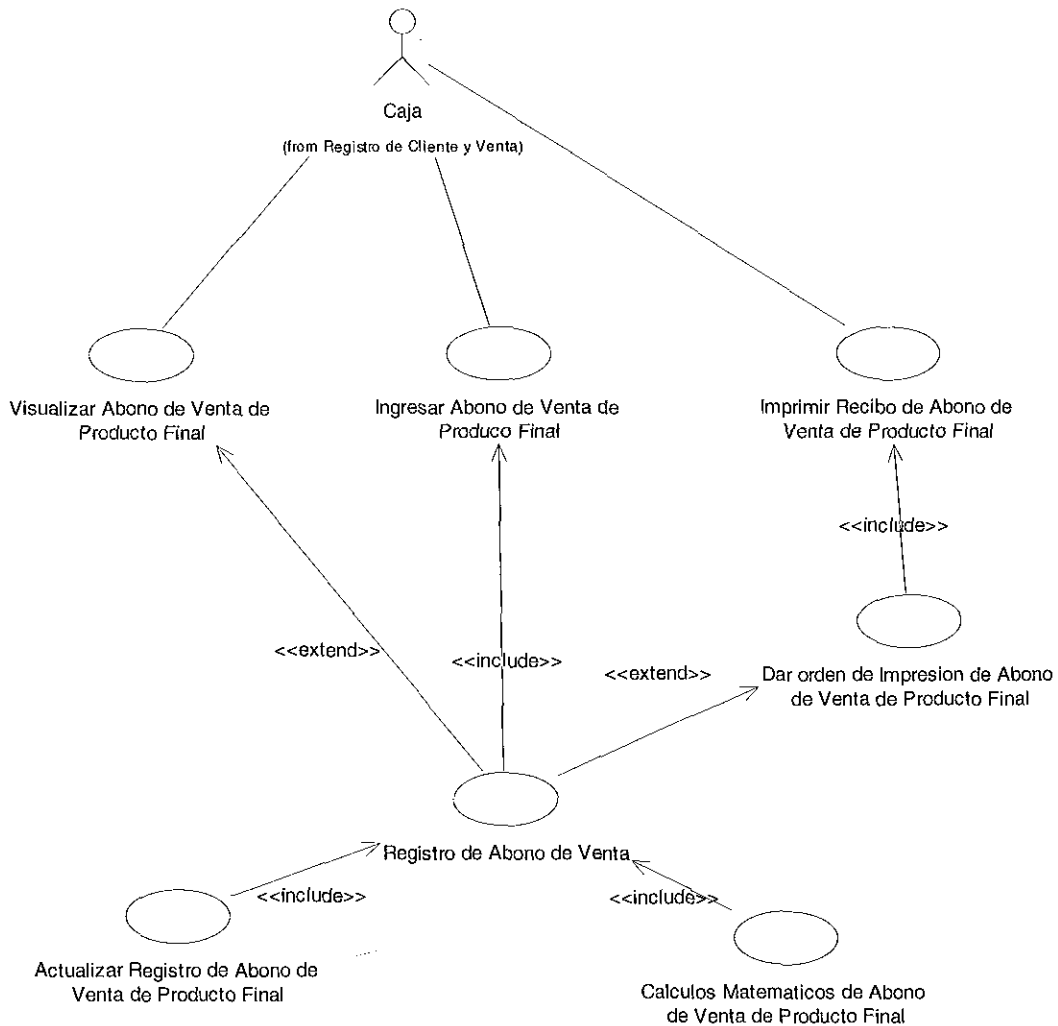
Pos condiciones: El total abonado no debe de ser igual al monto total de la factura

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1.Caja Introduce la cantidad abonada por la empres a los proveedores	2. Realizar cálculos matemáticos
	3. Introducir abono al registro de la tabla de abono de la base de datos
	4. Actualizar registros de la tabla de abono de la base de datos del sistema.
5.Caja imprime Recibo de abono de abono de compra	6.Dar orden de Impresión de recibo de abono de compra
7.Caja recibe impresión de recibo de abono de compra	

Frecuencia: Cuando el recolector de el proveedor solicite el abono correspondiente

Registrar e Imprimir Recibo de Abono de Venta de Producto Final



Caso de Uso: Registrar e Imprimir recibo de Abono de Venta de Producto Final

Objetivo: Llevar Un registro de todos los abonos realizado por parte del cliente a la empresa

Actor:

- Caja

Precondiciones: Deben de existir Ventas al crédito en la base de datos

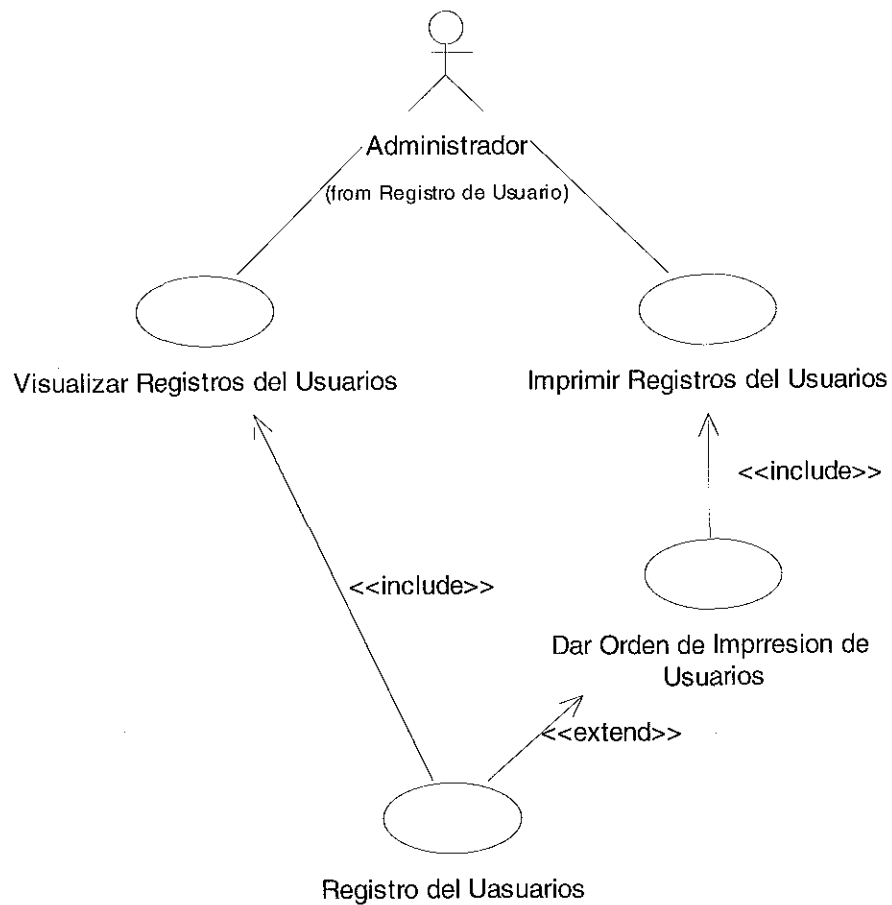
Pos condiciones: El total abonado no debe de ser igual al monto total de la factura

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1.Caja Introduce la cantidad abonada por el cliente a la empresa	2. Realizar cálculos matemáticos
	3. Introducir abono al registro de la tabla de abono de la base de datos
	4. Actualizar registros de la tabla de abono de la base de datos del sistema.
5.Caja imprime Recibo de abono de Venta	6.Dar orden de Impresión de recibo de Abono de venta
7.Caja recibe impresión de recibo de abono de venta	

Frecuencia: Cada Vez que el cliente le toque hacer el abono

Visualizar e Imprimir Registro del Usuario



Caso de Uso: Visualizar e imprimir registro del usuario

Objetivo: llevar un registro con los datos personales de los usuarios

Actor:

- Administrador

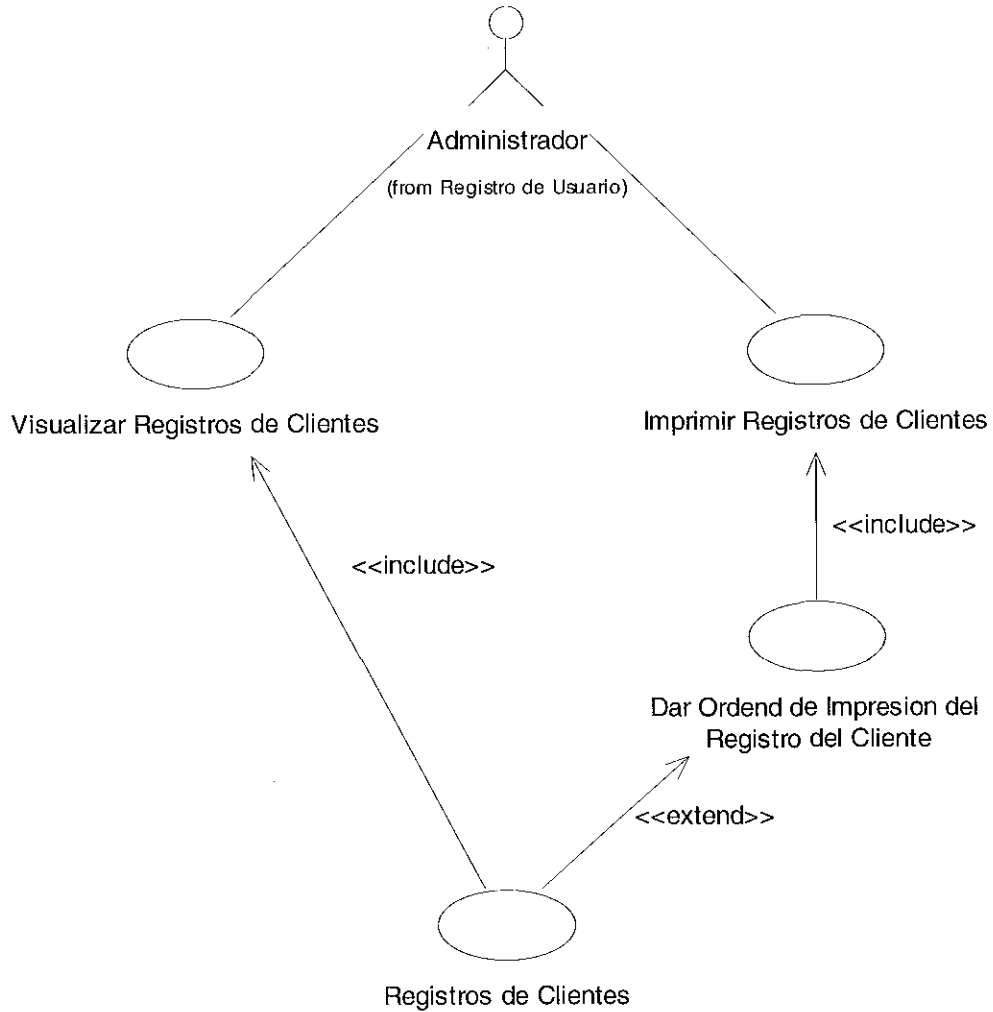
Precondiciones: Deben de existir Registro de Usuarios en la base de datos

Flujo Básico:

<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sistema</i></u>
<i>1. Administrador visualiza los datos personales del usuario</i>	<i>2. Muestra los registros de la tabla de usuario de la base de datos del sistema</i>
<i>3. Administrador imprime los registros del usuario</i>	

Frecuencia: Cada vez que el Administrador lo desee hacer

Visualizar e Imprimir Registro del Cliente



Caso de Uso: Visualizar e imprimir registro del Cliente

Objetivo: llevar un registro con los datos personales del Cliente solicitados por el vendedor

Actor:

- Administrador

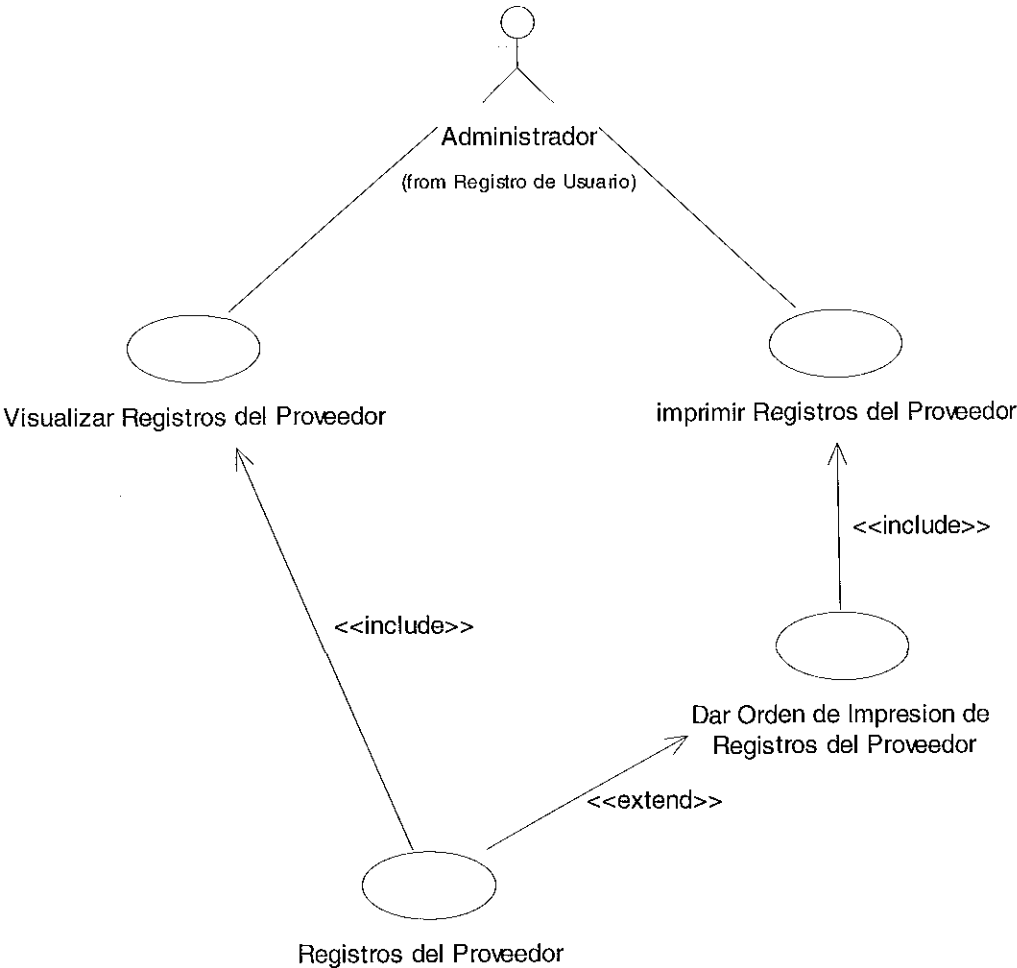
Precondiciones: Deben de existir Registro de Cliente en la base de datos

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sistema</u>
1. Administrador visualiza los datos personales del cliente	2. Muestra los registros de la tabla del cliente de la base de datos del sistema
3. Administrador imprime los registros del cliente	

Frecuencia: Cada vez que el Administrador lo desee hacer

Visualizar e Imprimir Registro del Proveedor



Caso de Uso: Visualizar e imprimir Registros del Proveedor

Objetivo: llevar un registro con los datos personales del Proveedor solicitados por la cajera

Actor:

- Administrador

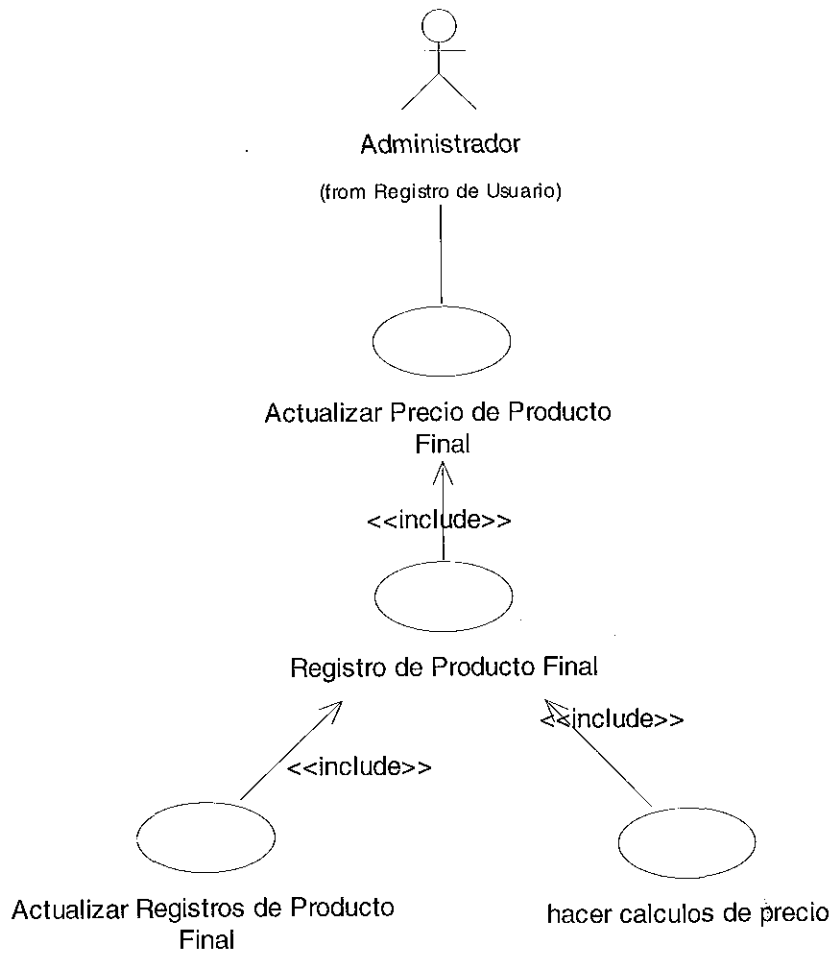
Precondiciones: Deben de existir Registro del Proveedor en la base de datos

Flujo Básico:

<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sistema</i></u>
<i>1. Administrador visualiza los datos personales del Proveedor</i>	<i>2. Muestra los registros de la tabla del Proveedor de la base de datos del sistema</i>
<i>3. Administrador imprime los registros del Proveedor</i>	

Frecuencia: Cada vez que el Administrador lo desee hacer

Actualizar Precio de Producto Final



Caso de Uso: Actualizar Precio de Producto Final

Objetivo: Poder ponerle precios al Producto Final

Actor:

- Administrador

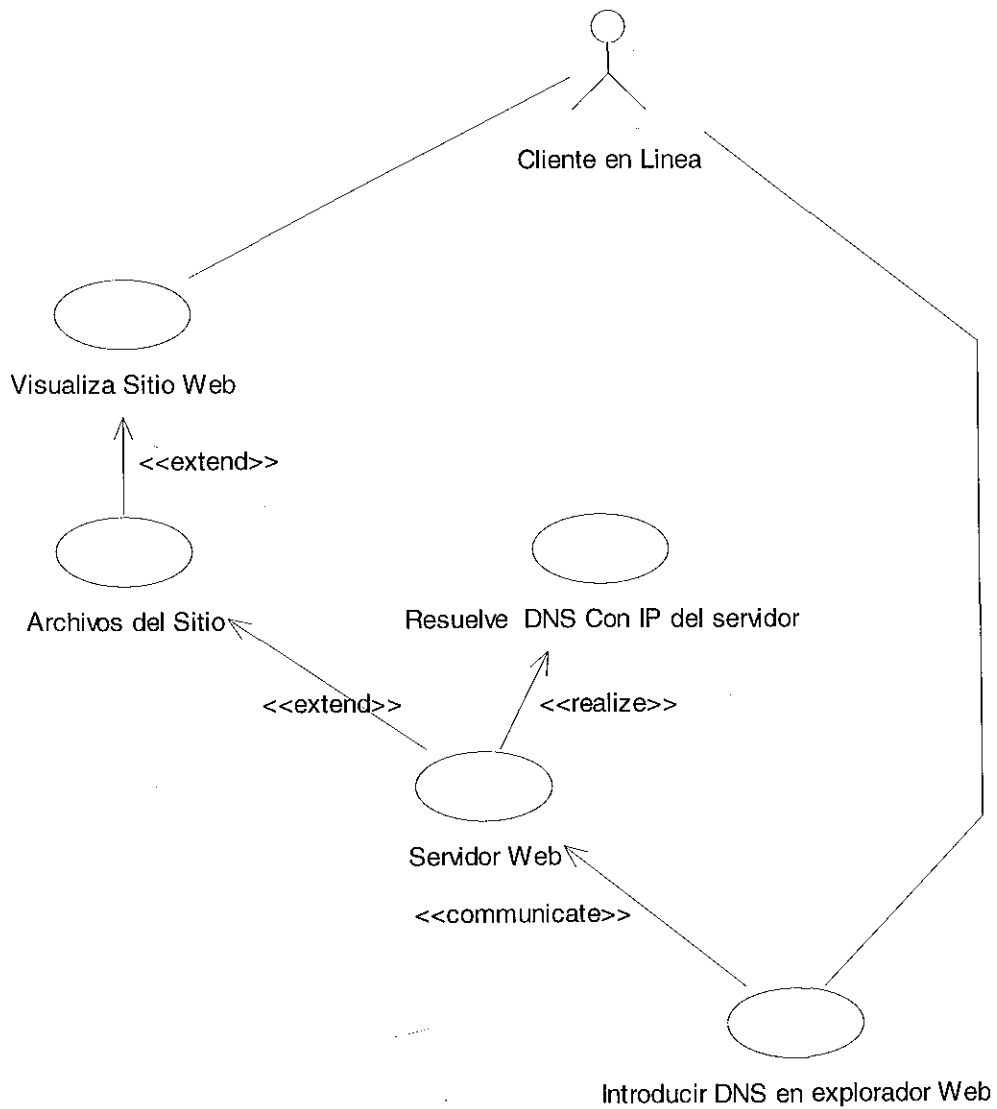
Flujo Básico:

<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sistema</i></u>
<i>1. Administrador Actualiza precios de Producto Final</i>	<i>2. Actualiza la Tabla de Producto Final de la Base de datos del Sistema</i>

Frecuencia: Cada vez que se necesite

2.1.2 Aplicación Web

Ingresar al Sitio Web



Caso de Uso: Ingresar al Sitio Web

Objetivo: Llevar Un registro de todos los proveedores y compras que se realicen

Actor:

- Cliente On Line

Precondiciones: Tener Un Explorador Web y conexión a Internet

Pos condiciones: Se Visualizara El Sitio Web de Sáenz Bloques Adoquines Y mas...

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sevidor</u>
1. Cliente On Line Introduce el DNS en el explorador Web	2. Resolver DNS comparando con su IP
	3. Mostrar la pagina Web Principal del sitio
4. Cliente On Line Visualiza el Sitio Web	5. El Sistema Registrar Compra de Materia Prima
	6. Actualizar Registros de Compra de MP

Flujo Alternativo: 2. Resuelve DNS con su IP:

El servidor compara el DNS con su número IP asignado por un ISP

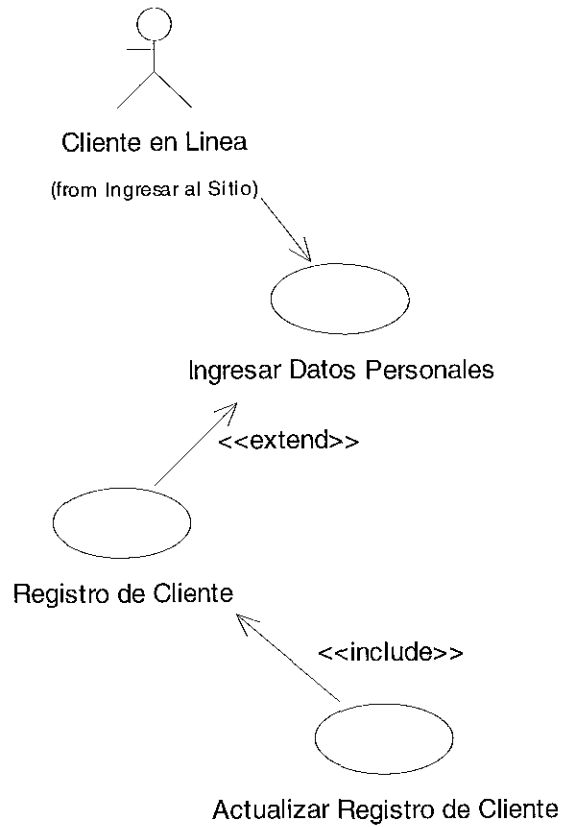
3. Mostrar Pagina Wb del Sitio

Por defecto la pagina de inicio del sitio Web es default.asp

Requisitos Especiales: Tener Instalado en el ordenador del cliente Un Explorador Web

Frecuencia: Cada Vez Que el cliente desee entrar al sitio Web

Ingresar Cliente



Caso de Uso: Ingresar Cliente

Objetivo: Llevar Un registro de todos los proveedores y compras que se realicen

Actor:

- Cliente On Line

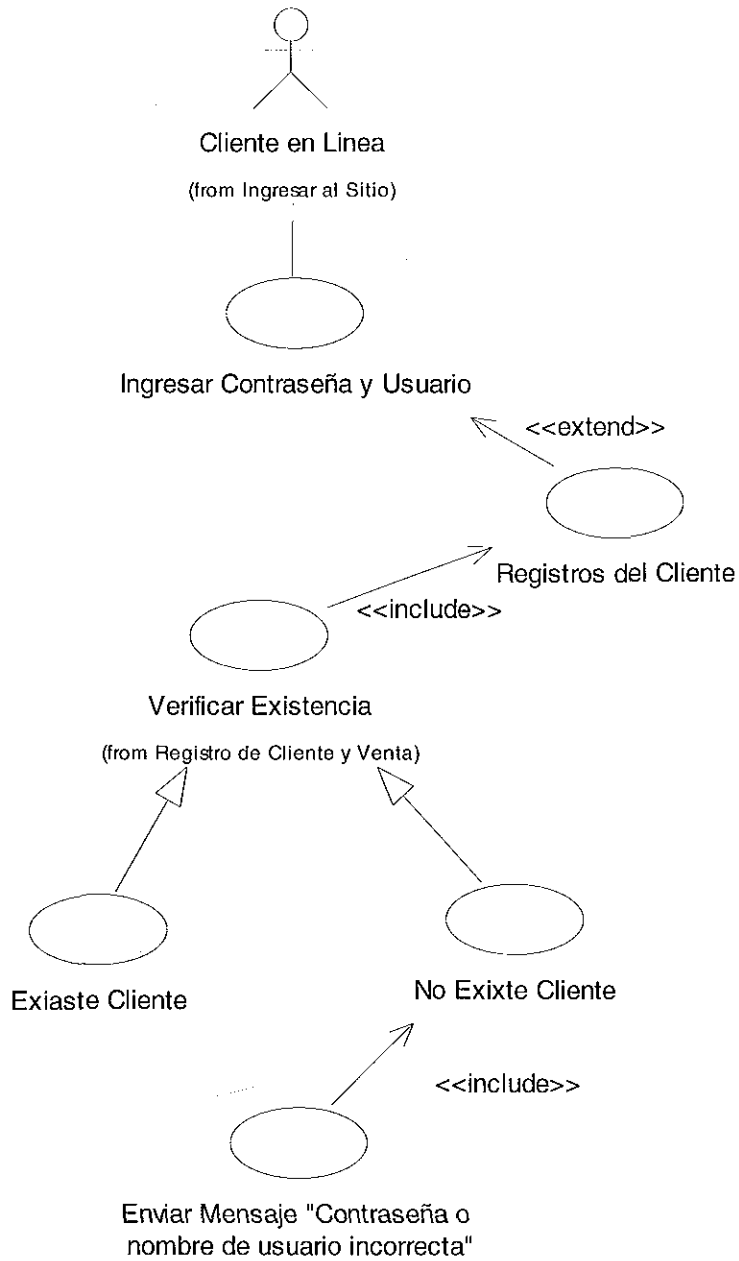
Precondiciones: Estar Dentro del Sitio Web de la empresa

Flujo Básico:

<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sitio Web</i></u>
<i>1. Cliente On Line Ingresa Datos Personales</i>	<i>2. Actualiza La tabla de Cliente de la base de datos</i>

Frecuencia: Cuando desee el cliente

Logearse en el sitio Web



Caso de Uso: Validarse en el Sitio

Objetivo: Poder acceder a la página de compra

Actor:

- Cliente On Line

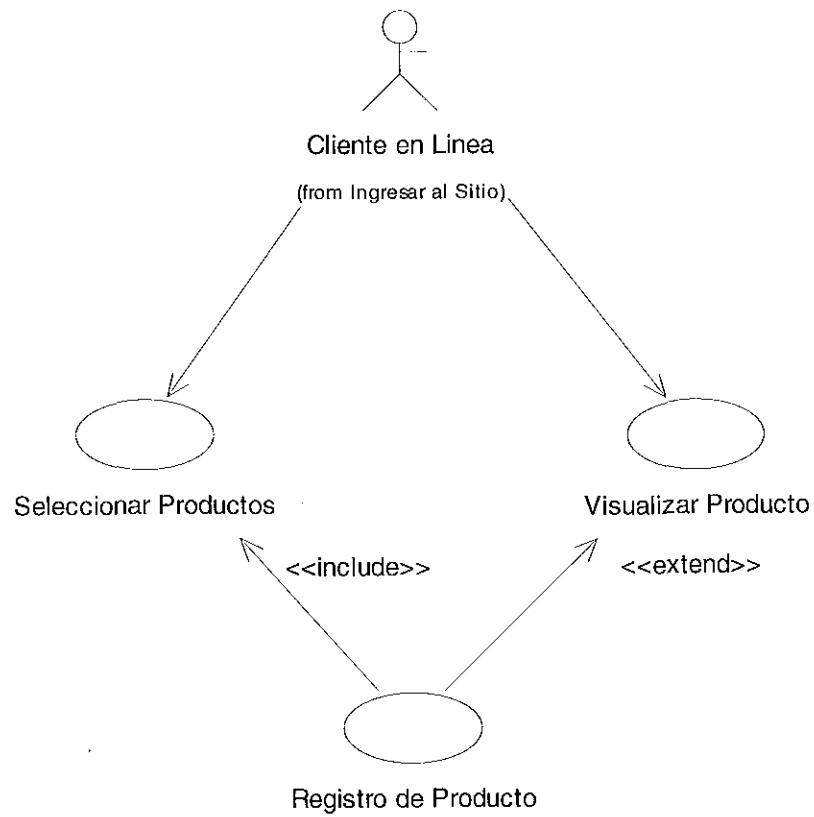
Precondiciones: Haberse registrado anteriormente

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sitio Web</u>
1. <i>Cliente On Line Ingresa Contraseña y Nombre de usuario</i>	2. <i>Verifica Existencia</i>
	3. <i>Si existe Cliente en la tabla de cliente de la base de datos abrir pagina de Compras. Si no Existe mandar Un Mensaje que diga "Contraseña o nombre de usuario Incorrecta "</i>

Frecuencia: Cada Vez Que el cliente desee hacer una compra

Seleccionar Los Productos



Caso de Uso: Selección de Los Productos

Objetivo: Seleccionar los producto que se van a comprar

Actor:

- Cliente On Line

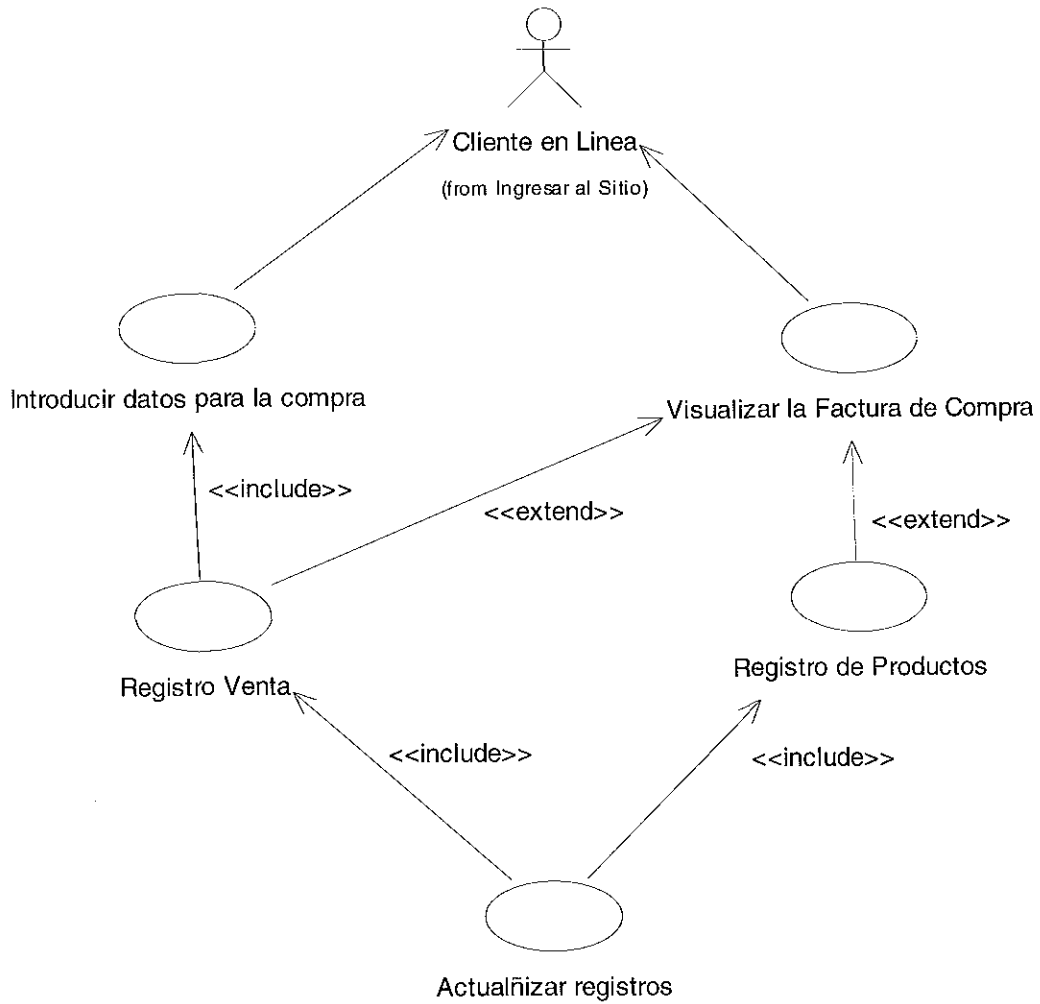
Precondiciones: Haberse Validado en el sitio

Flujo Básico:

<u>Acción del Actor</u>	<u>Responsabilidad del Sitio Web</u>
1. Cliente On Line Selecciona el Producto y cantidad a comprar	2. Verifica Existencia
	3. Si existen productos se visualiza la cantidad y el nombre del producto Si no Existe mandar Un Mensaje que diga "Lo sentimos por el momento no tenemos en existencia ese tipo producto "

Frecuencia: Cada Vez Que el cliente desee hacer una compra

Hacer el Pago de la compra



Caso de Uso: Hacer Pago de la compra

Objetivo: Realizar pago de productos anteriormente seleccionados

Actor:

- Cliente On Line

Precondiciones: Haberse registrado anteriormente

Flujo Básico:

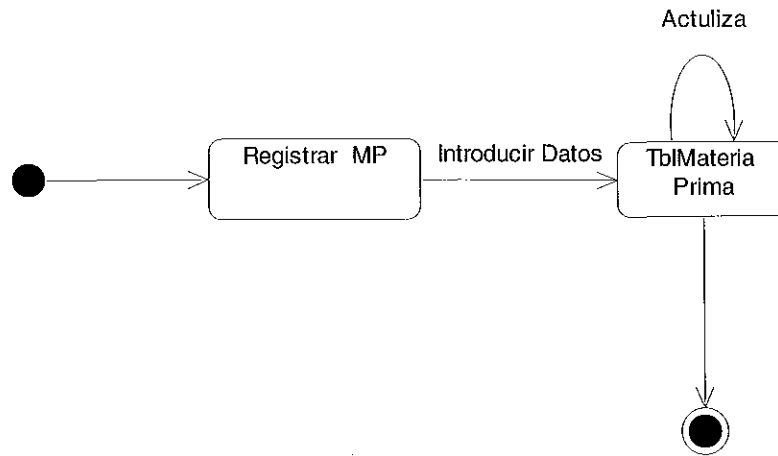
<u><i>Acción del Actor</i></u>	<u><i>Responsabilidad del Sitio Web</i></u>
<i>1. Cliente On Line Ingresar datos de la tarjeta de crédito o débito</i>	<i>2. Verifica número de tarjeta con base de datos de tarjeta de los bancos</i>
	<i>3. Si es válida se actualiza la base de datos</i>
	<i>Si no Existe mandar Un Mensaje que diga "Numero de tarjeta invalido"</i>

Frecuencia: Cada Vez Que el cliente desee hacer una compra

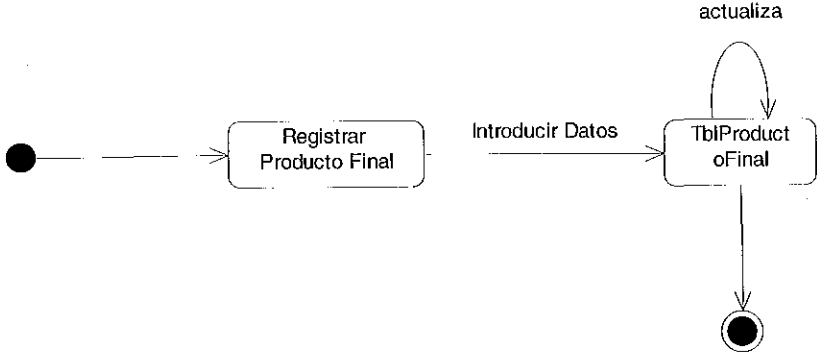
2.2 Diagramas de actividades

2.2.1 Aplicación Windows

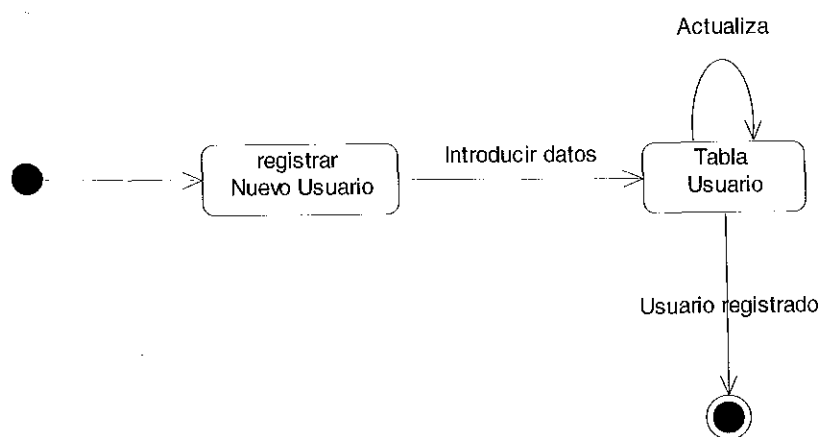
Registro de Materia Prima



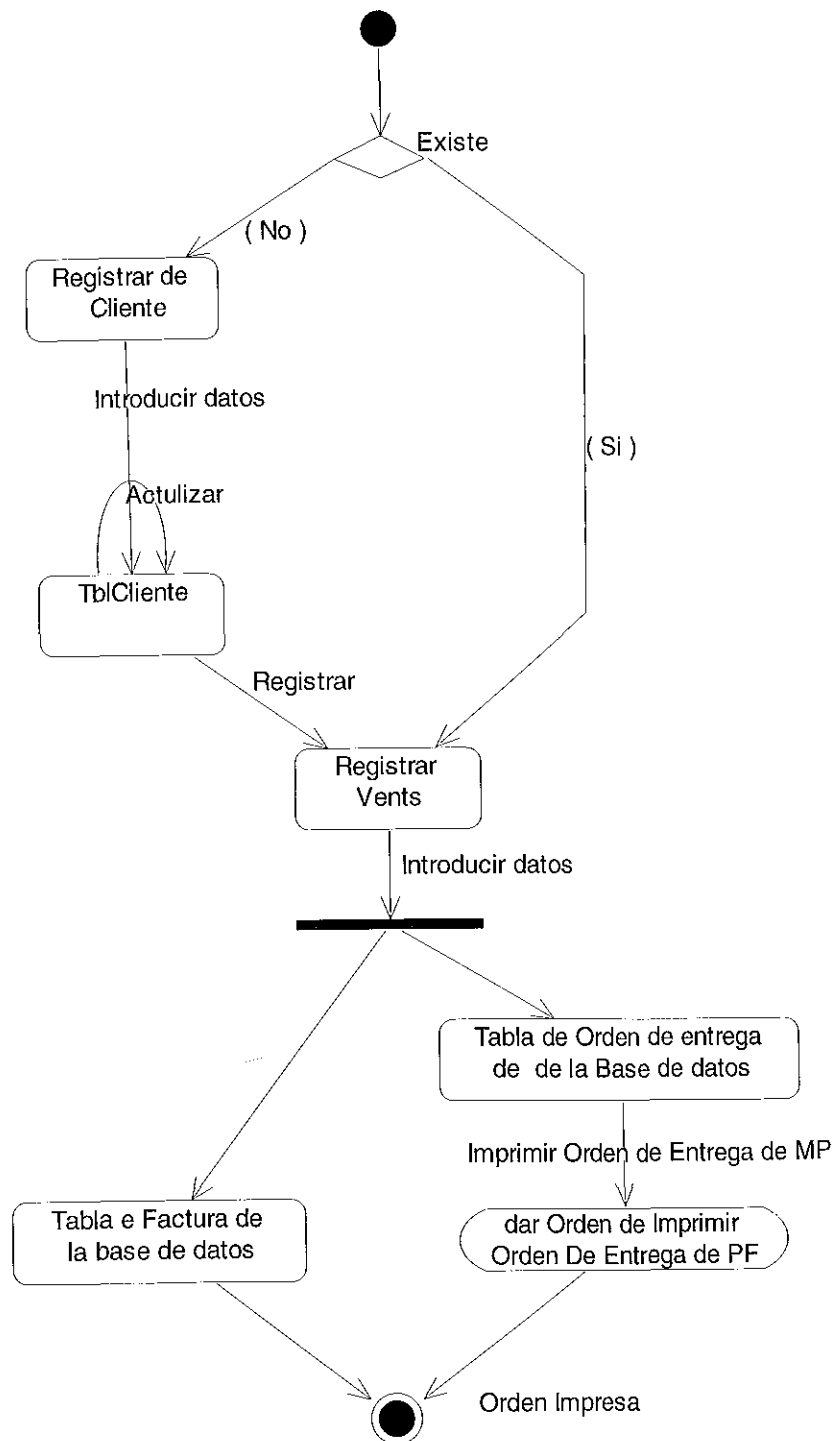
Registro de Producto Final



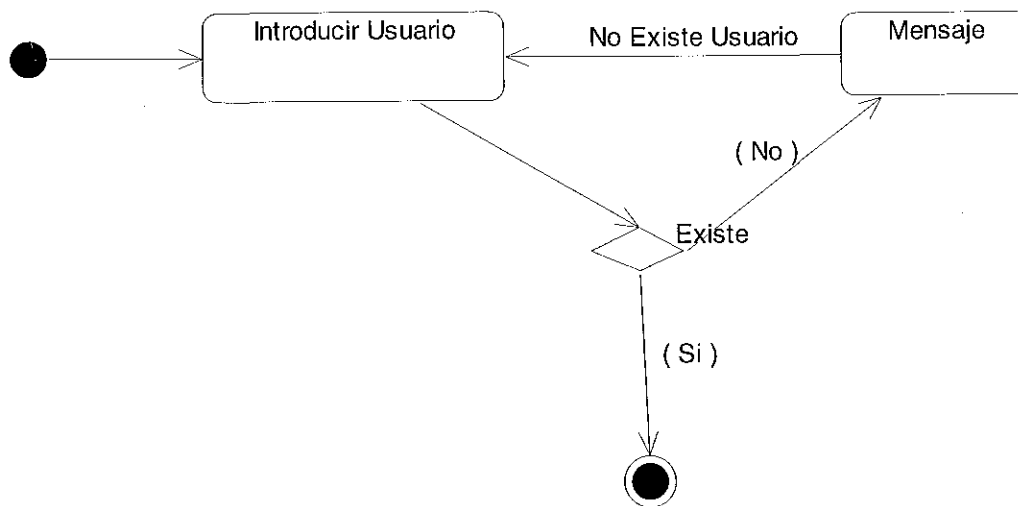
Registro de Usuario



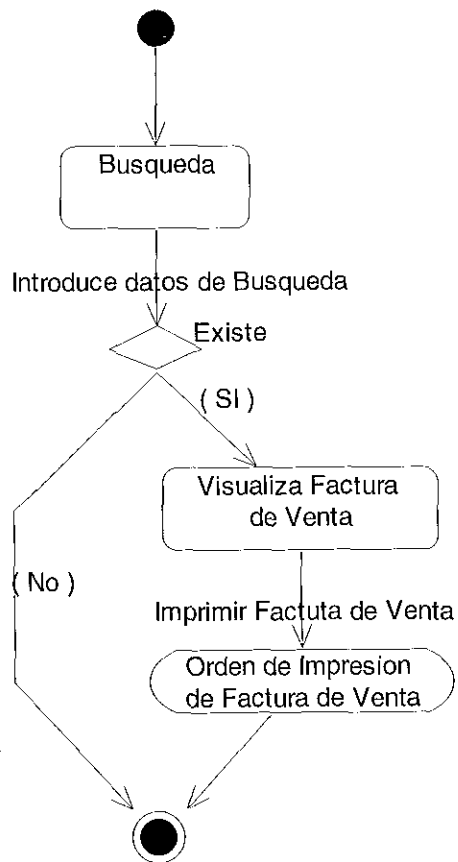
Registro de Cliente y Venta



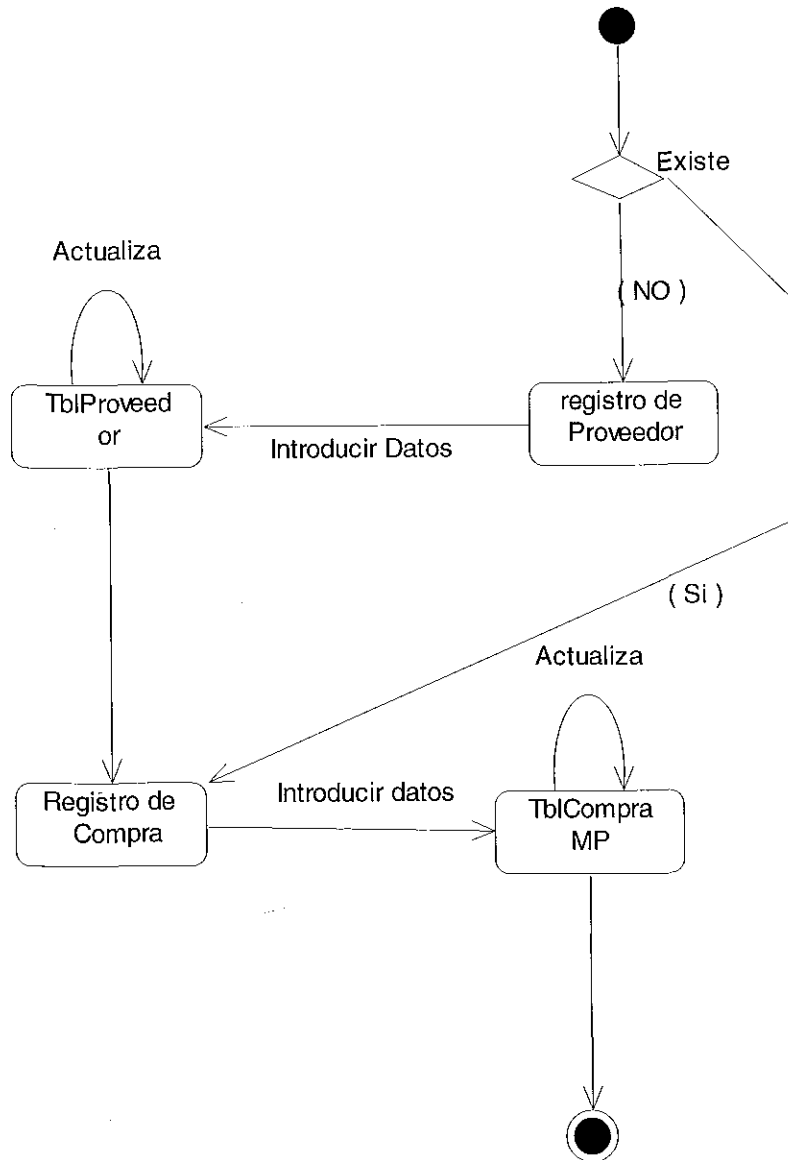
Login de Usuario



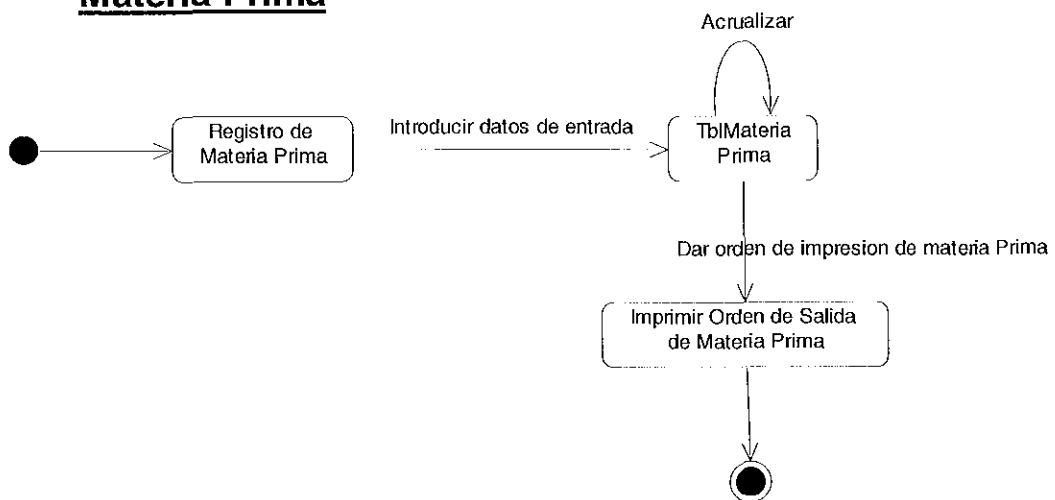
Busqueda e Impresion de Factura de Venta del Producto Final



Registro de Proveedor y Compra



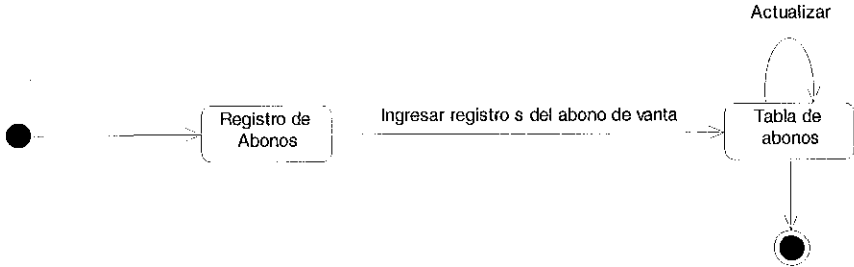
Registro e Impresion de Orden de Entrega de Materia Prima



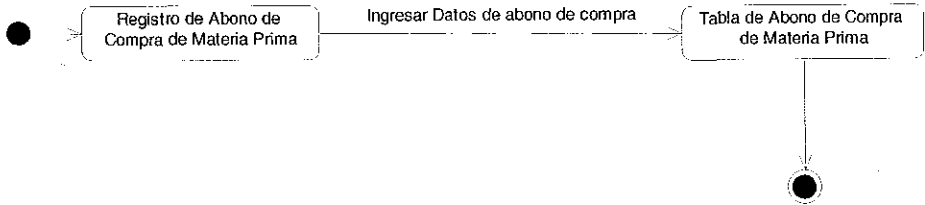
Actualizar Precios del Producto Final



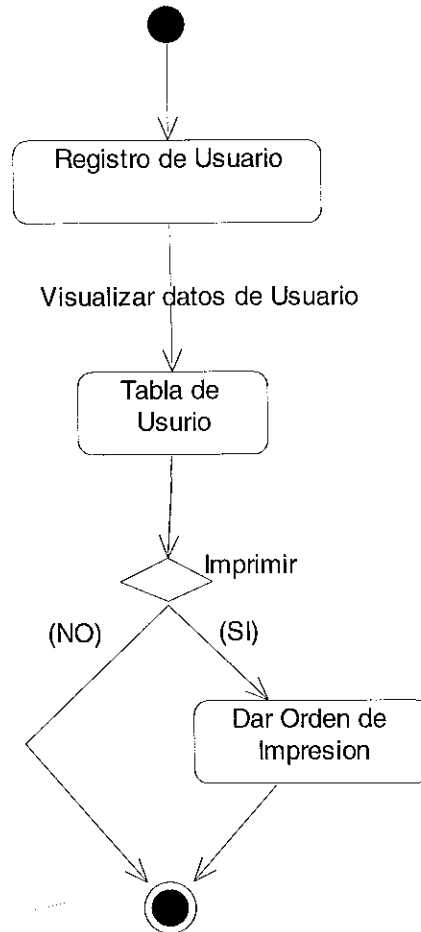
Registro de Abono de Venta de Producto Final



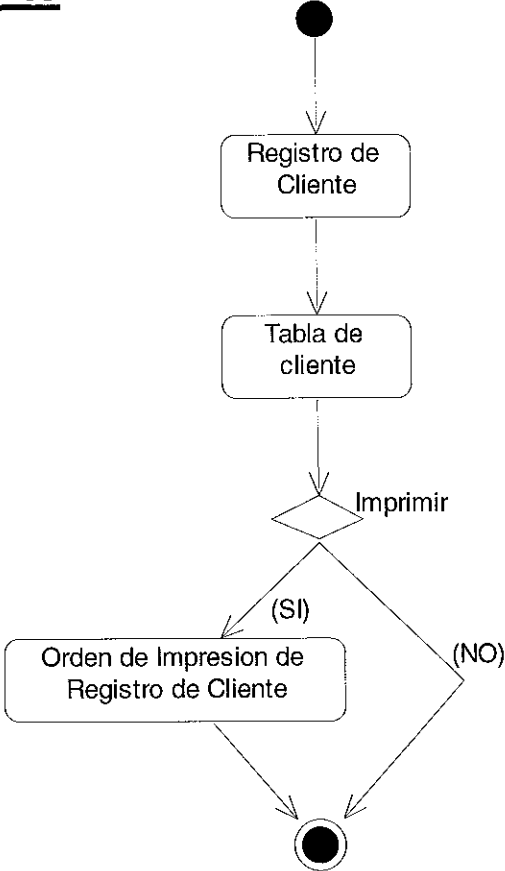
Registro de Abono de Compra de Materia Prima



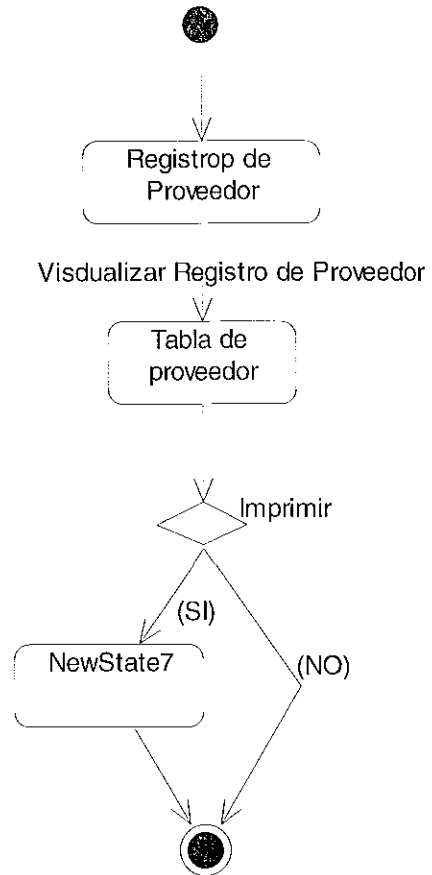
Visualizar e Imprimir Registro de usuario



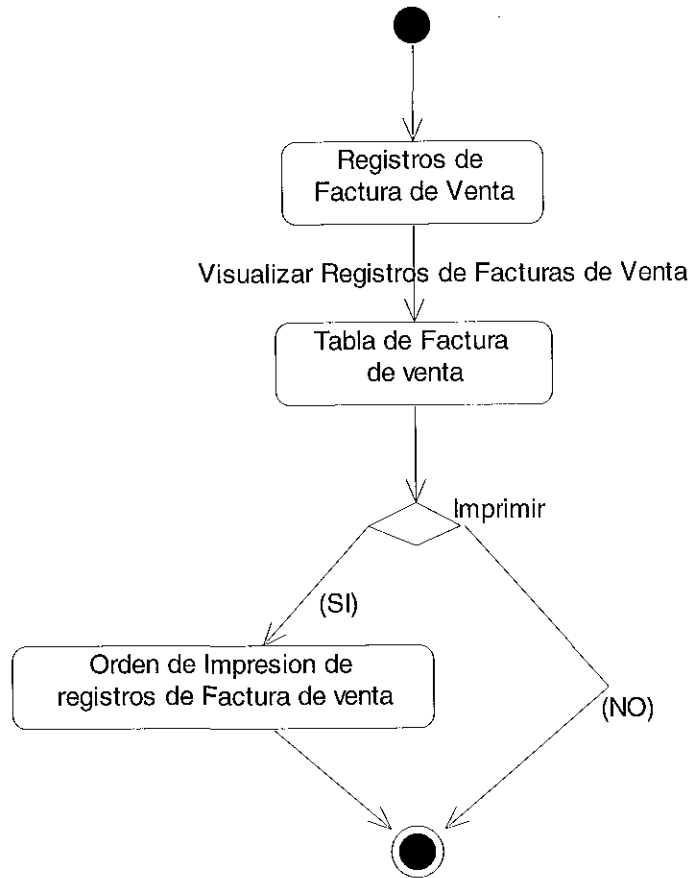
Visualizar e Imprimir Registro de Cliente



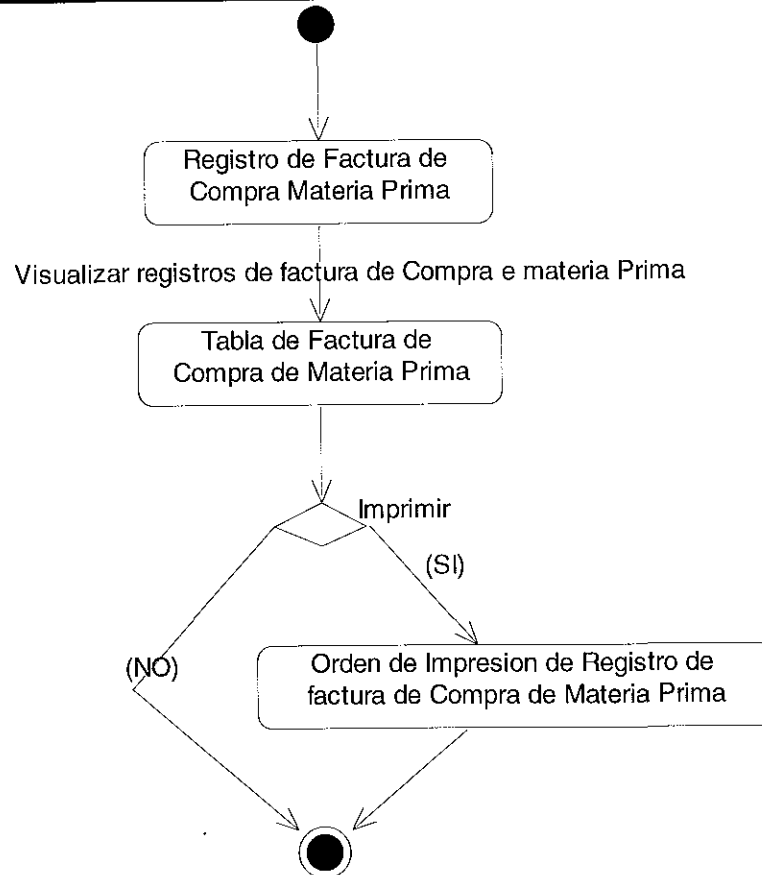
Visualizar e Imprimir Registro de Proveedor



Visualizar e Imprimir Registro de Factura de Venta

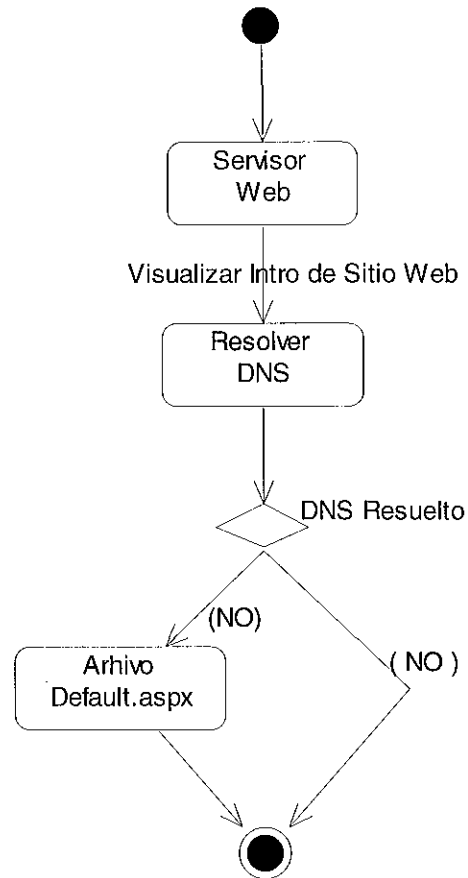


Visualizar e Imprimir Registro de Factura de Compra

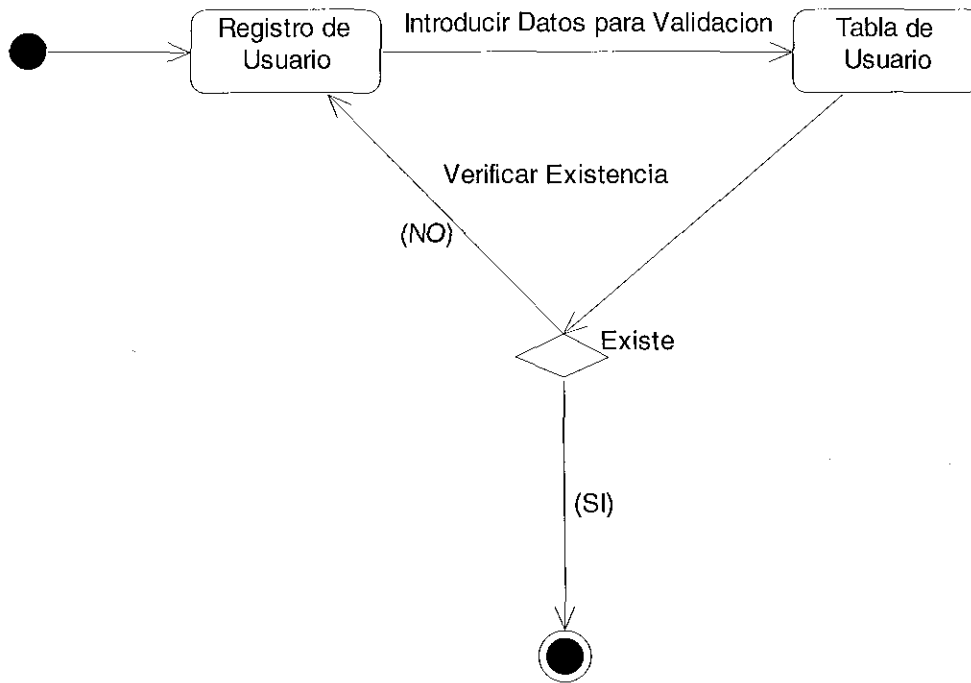


2.2.2 Aplicación Web

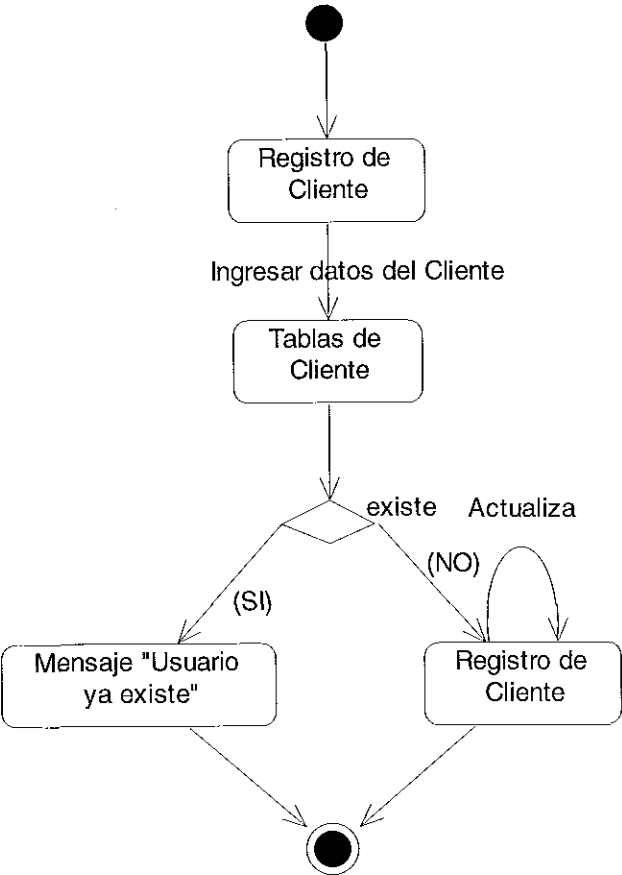
Visualizar Sitio Web



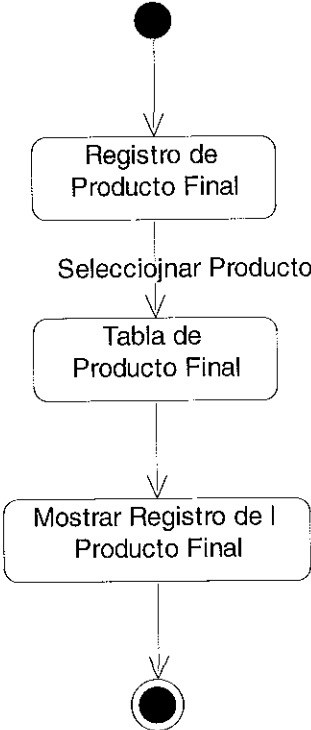
Login Usuario



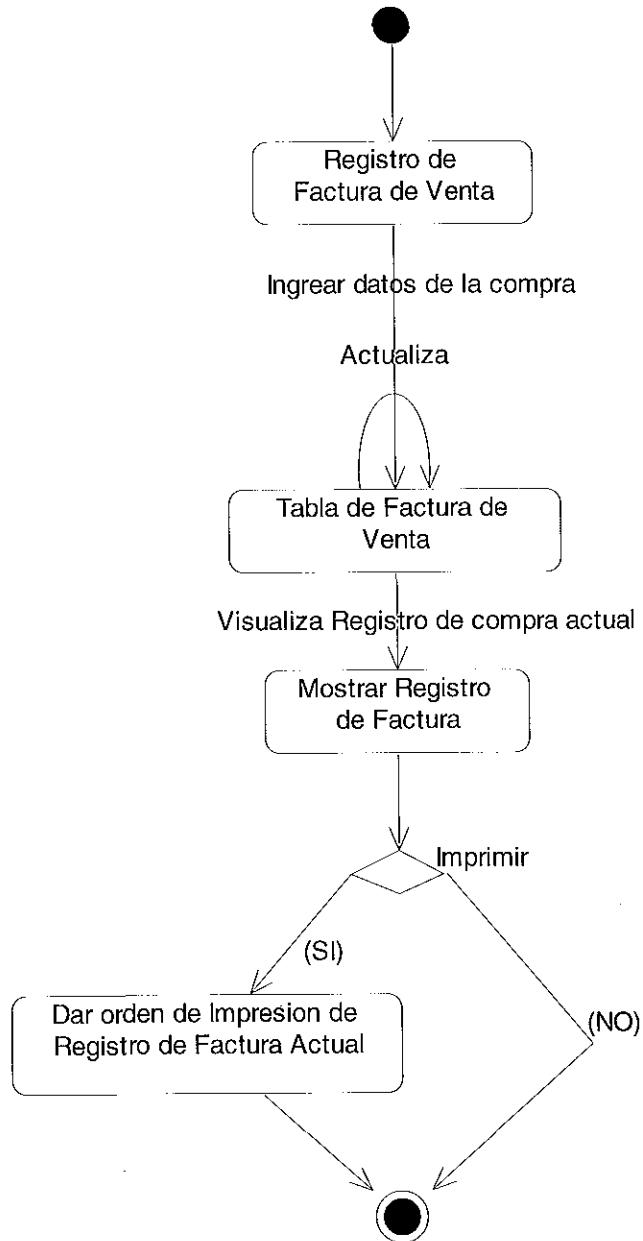
Registro de Cliente On Line



Seleccionar Los Productos



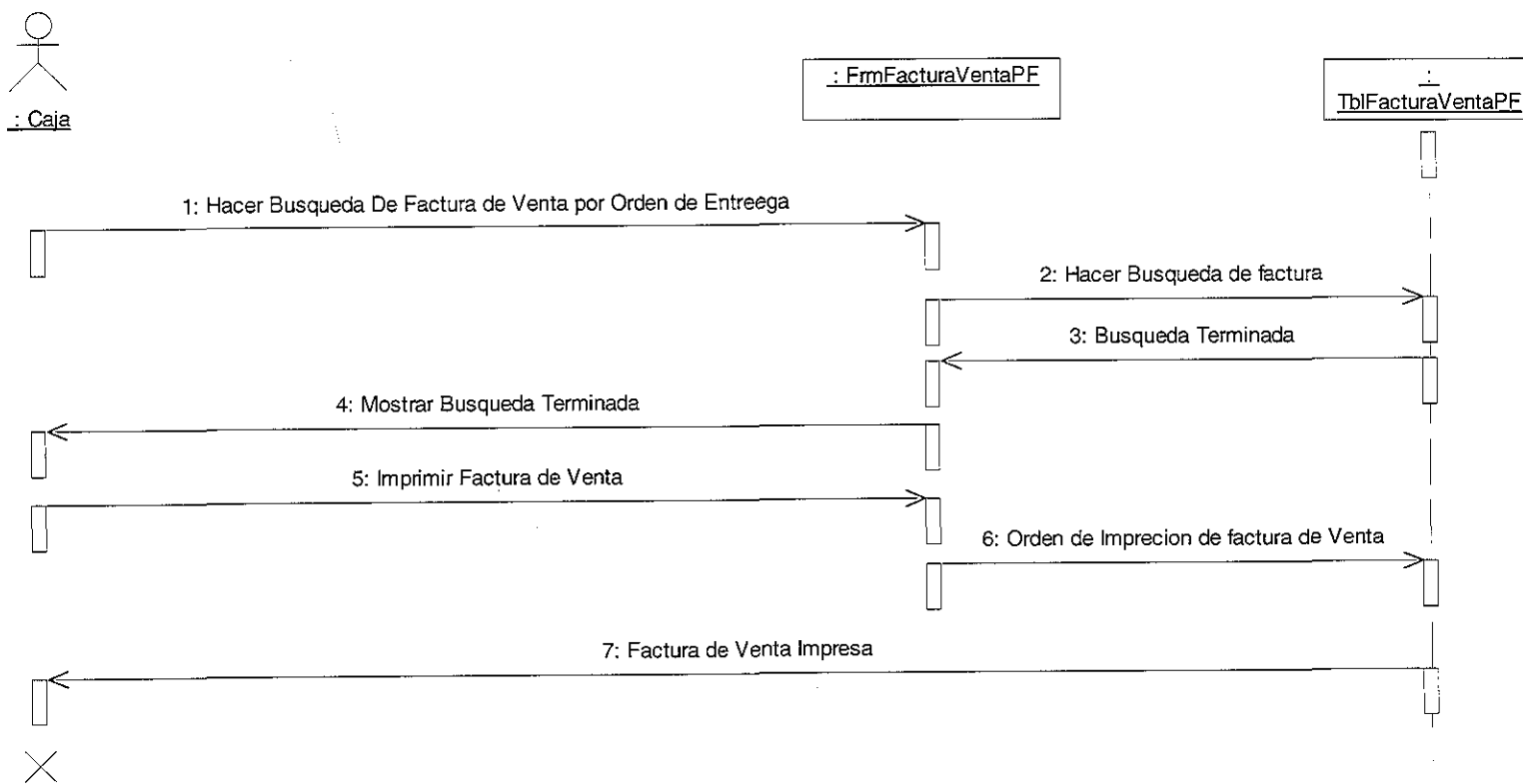
hacer Pago e imprimir factura de Compra



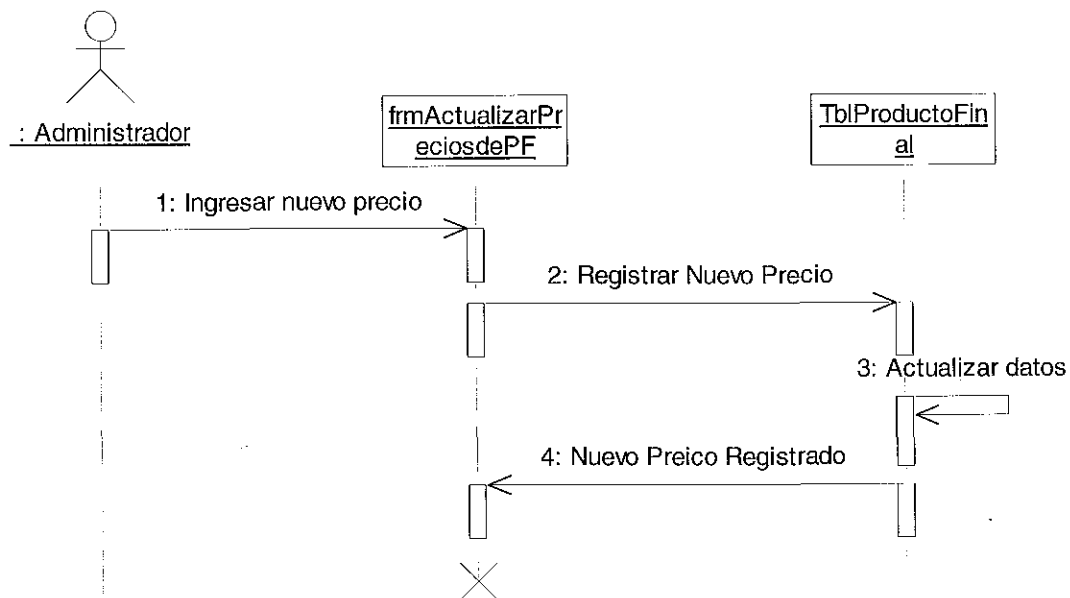
2.3 Diagrama de secuencia

2.3.1 Aplicación Windows

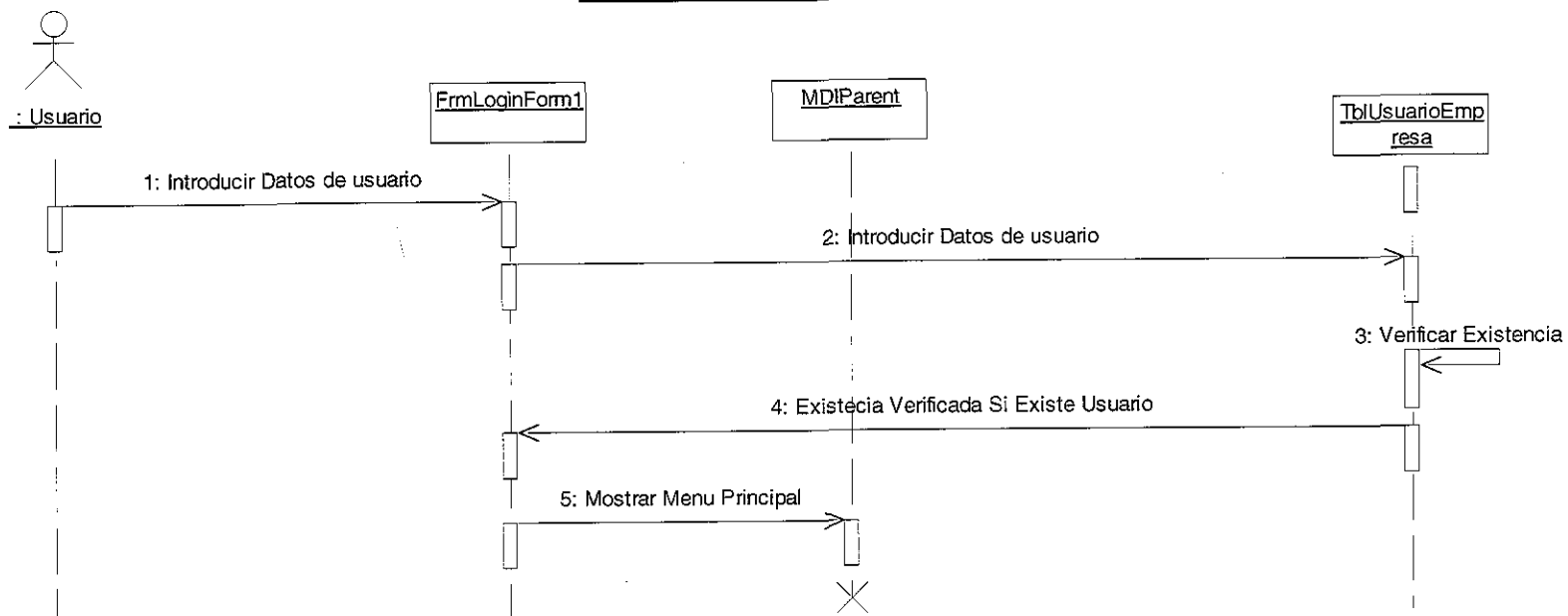
Búsqueda e Impresión de Factura de Venta



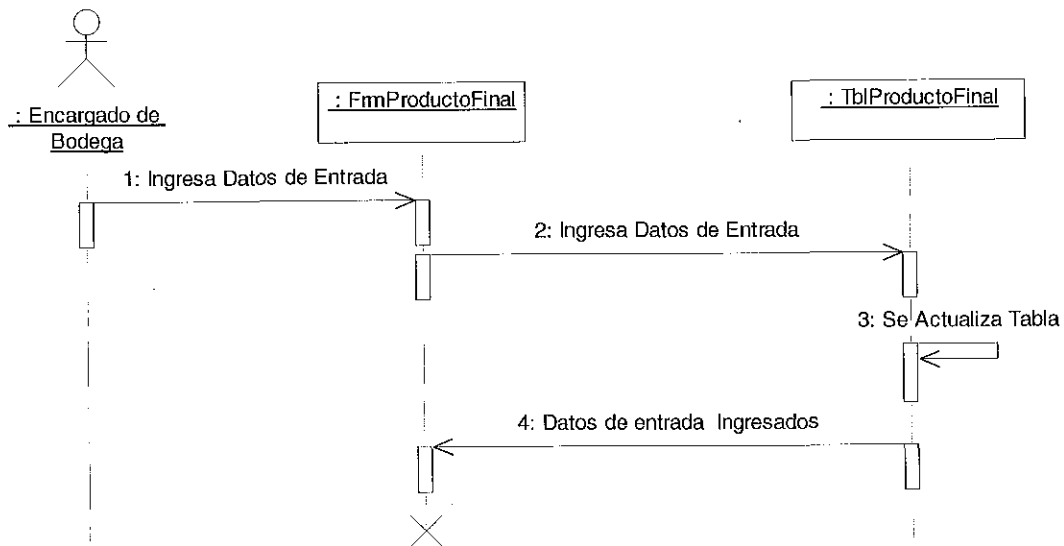
Actualizar Precios del Producto Final



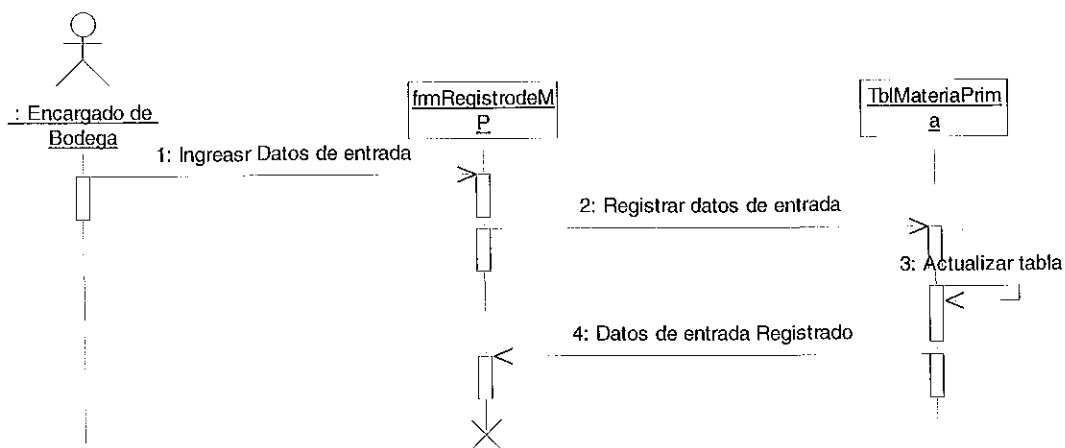
Login Usuario



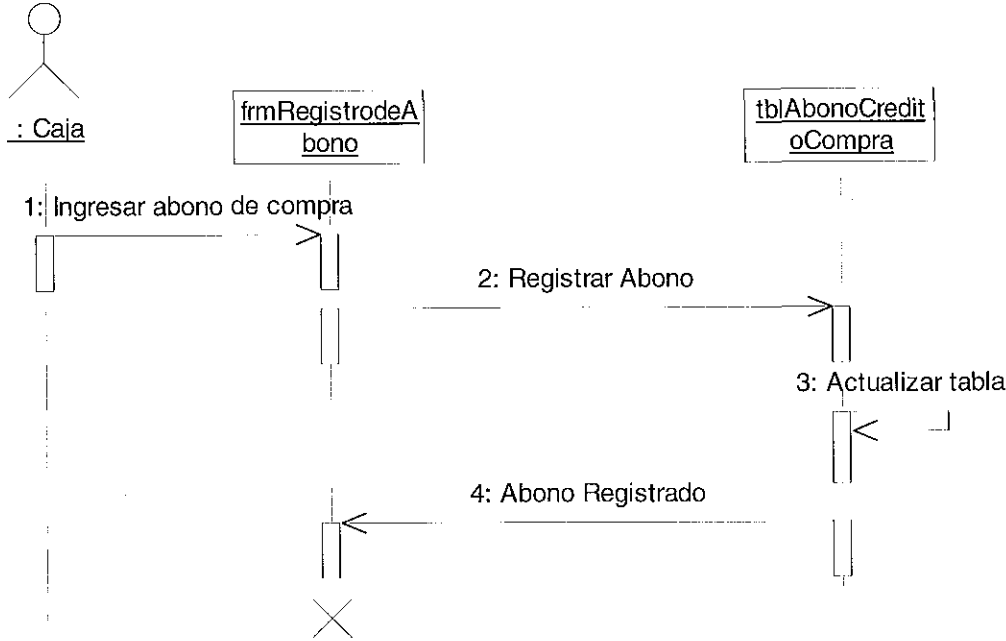
Registro de Producto Final



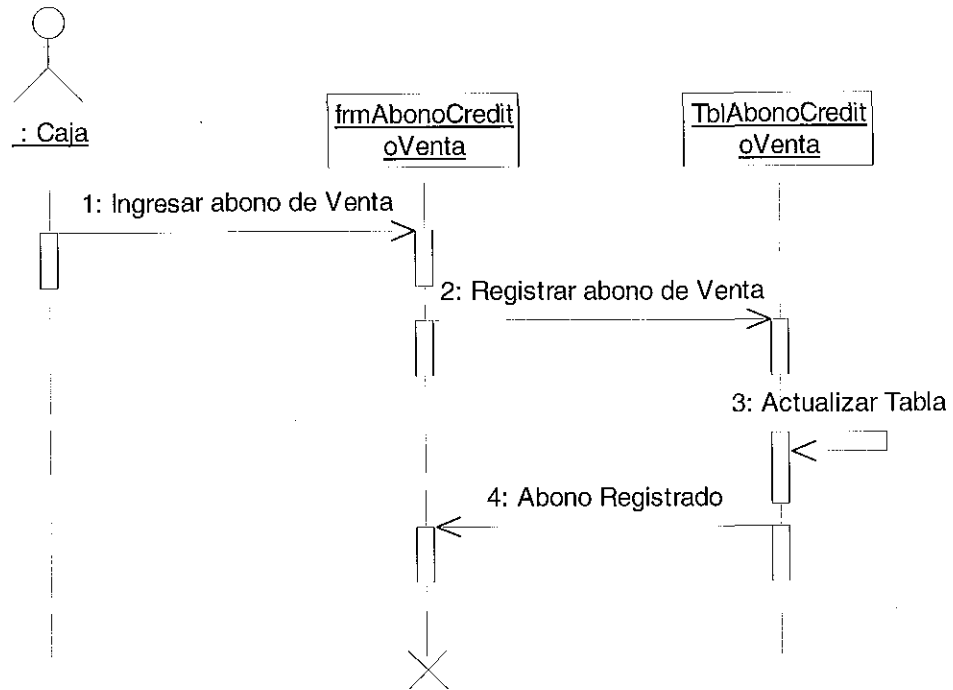
Registro de Materia Prima



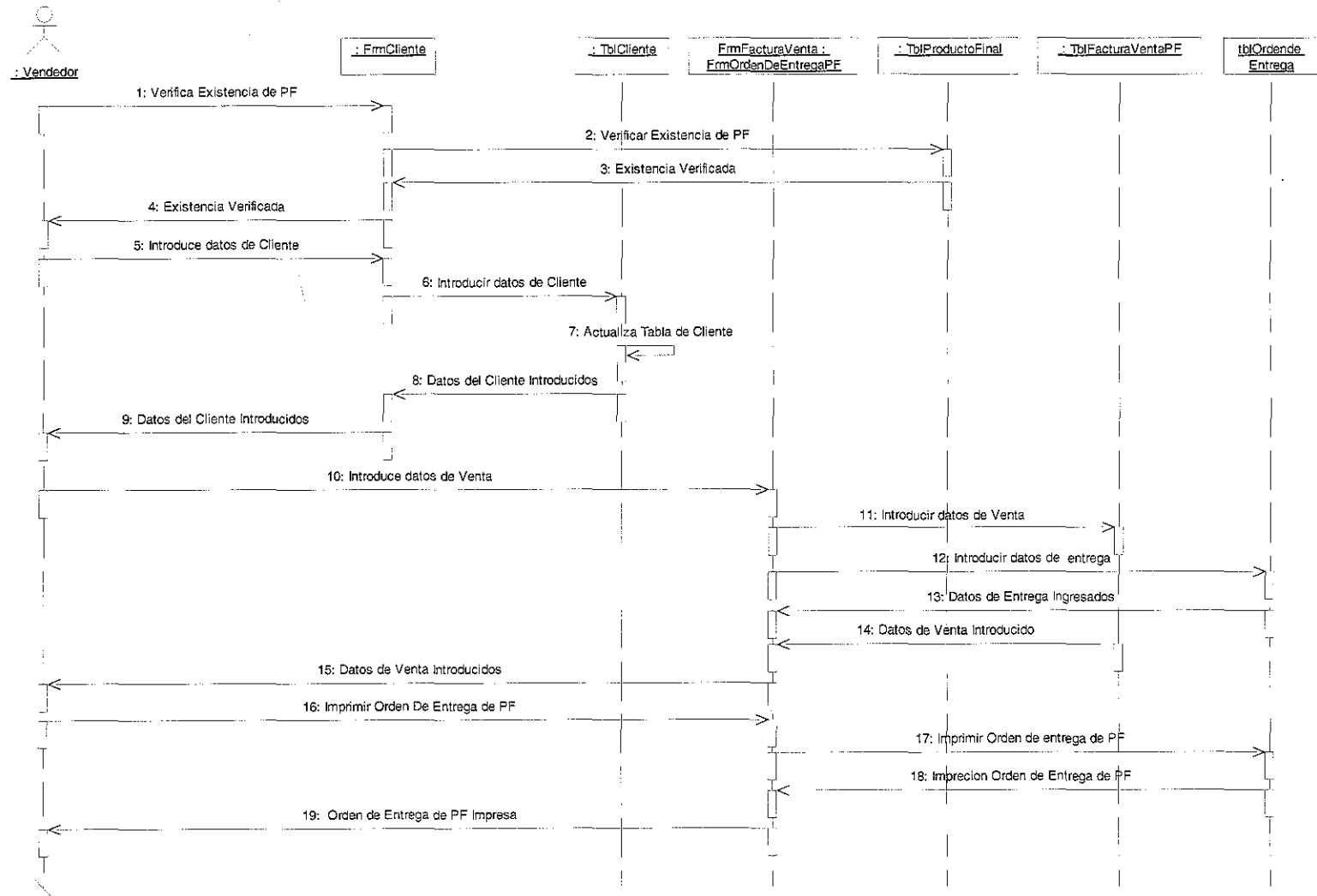
Registro de Abono de Compra



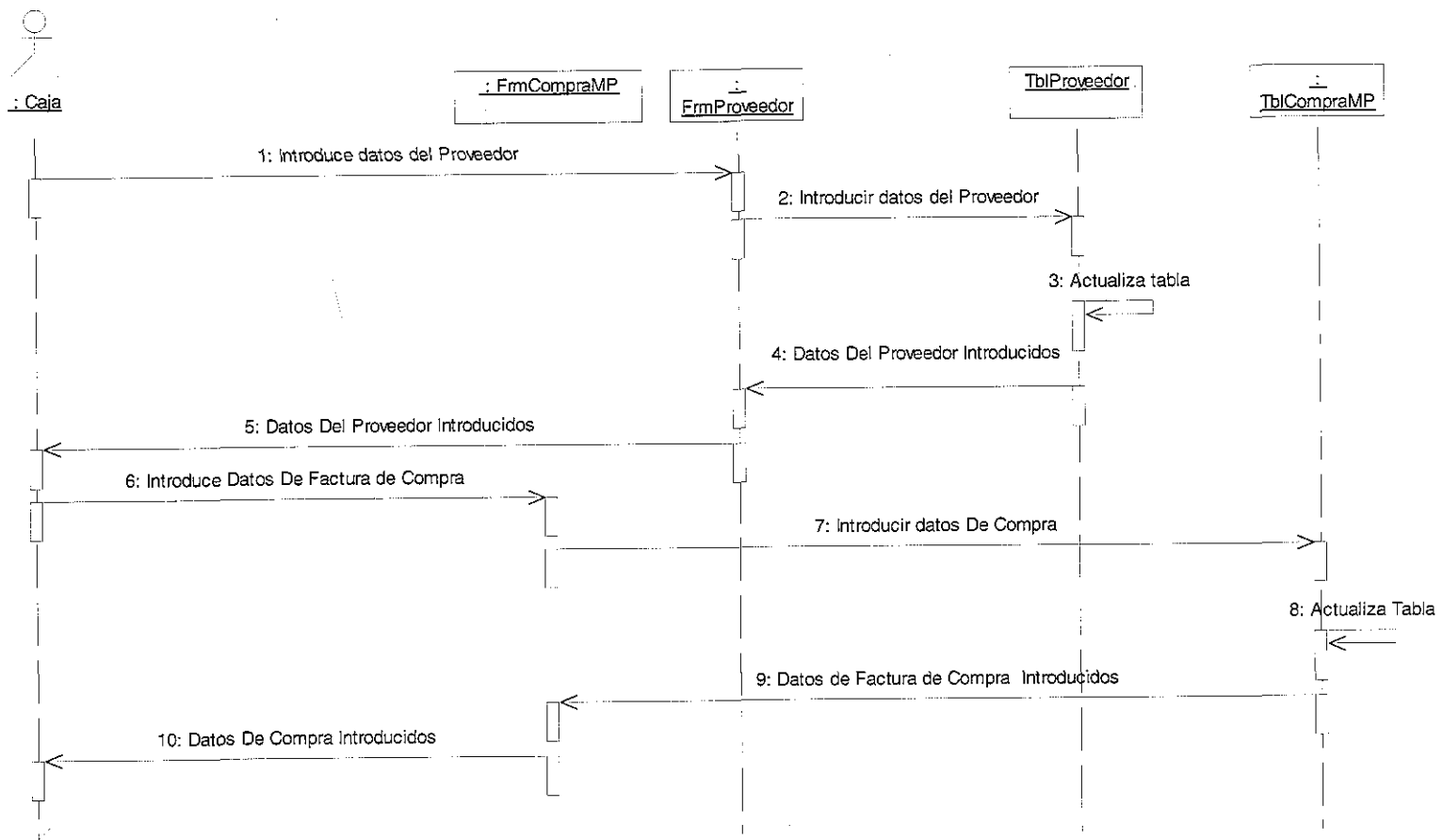
Registro de Abono de Venta



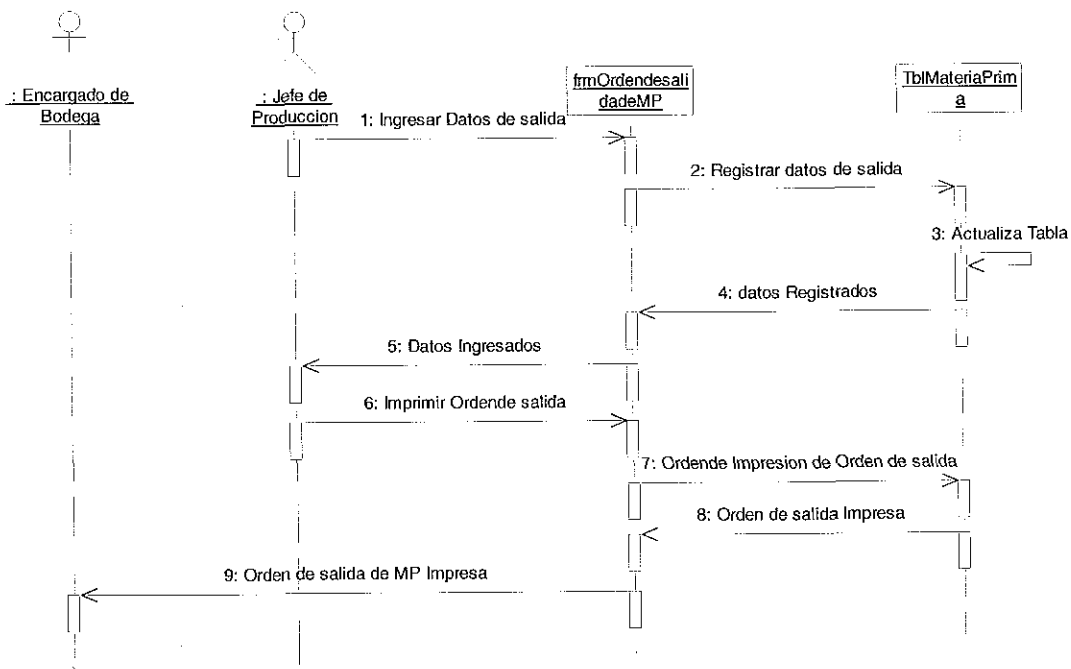
Registro de Cliente y Venta



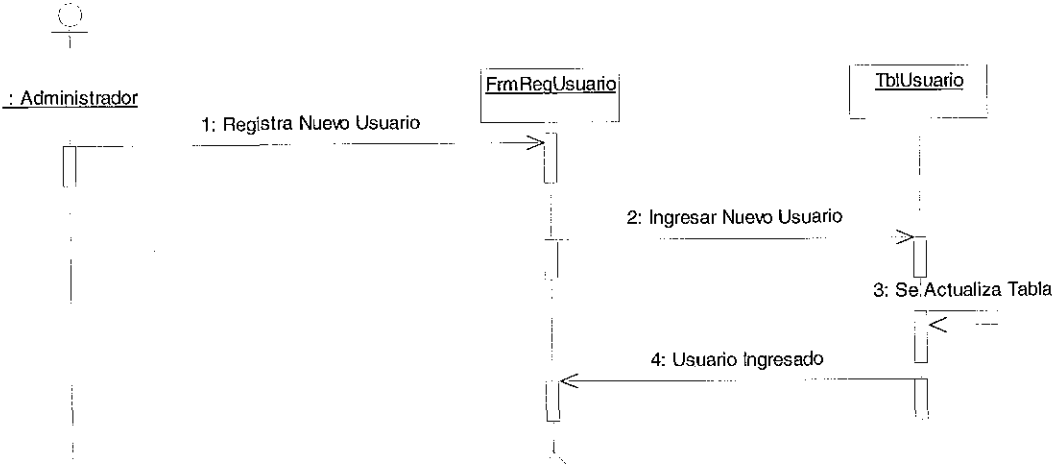
Registro de Proveedor y Compra



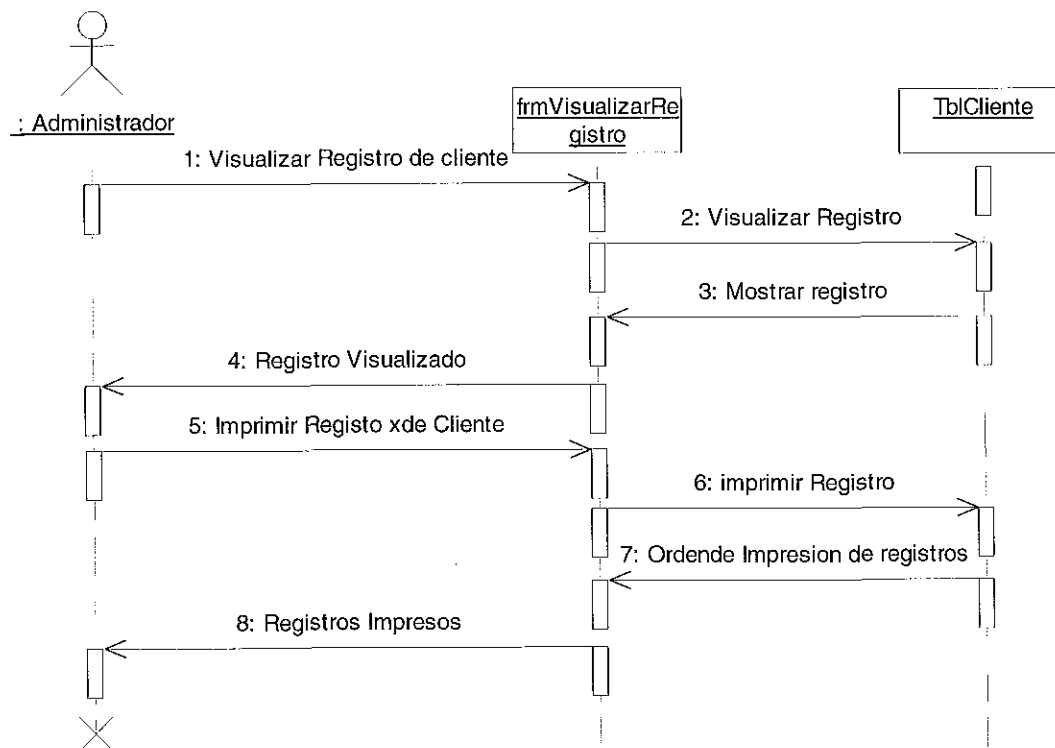
Salida de Materia Prima



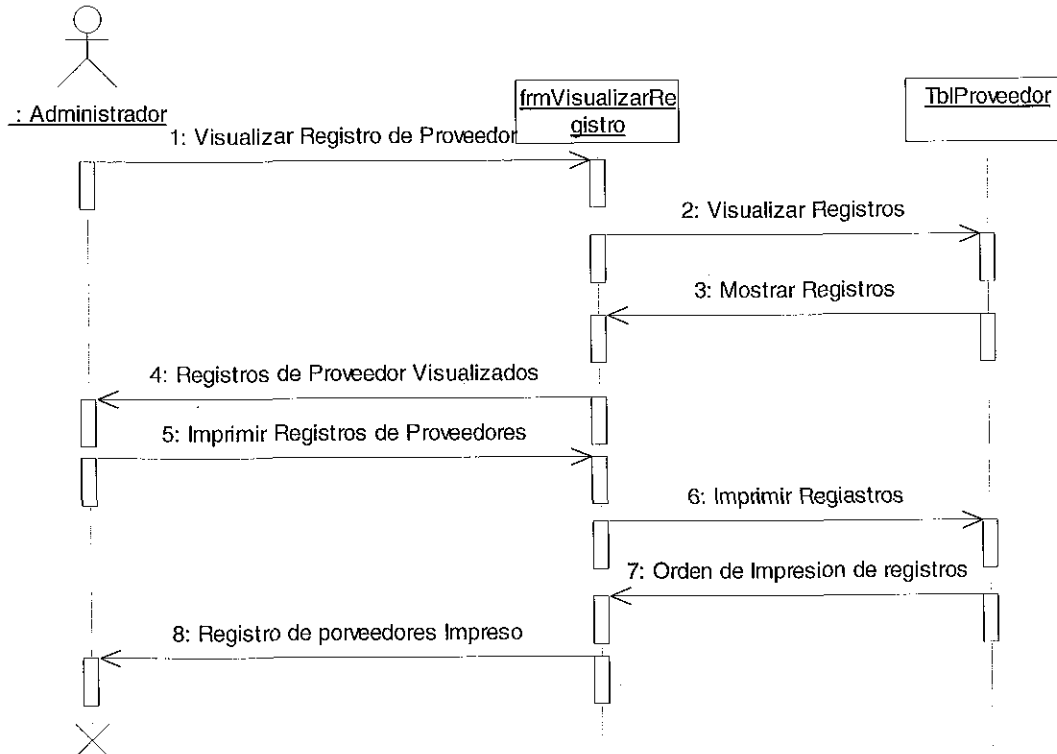
Registro de Usuario



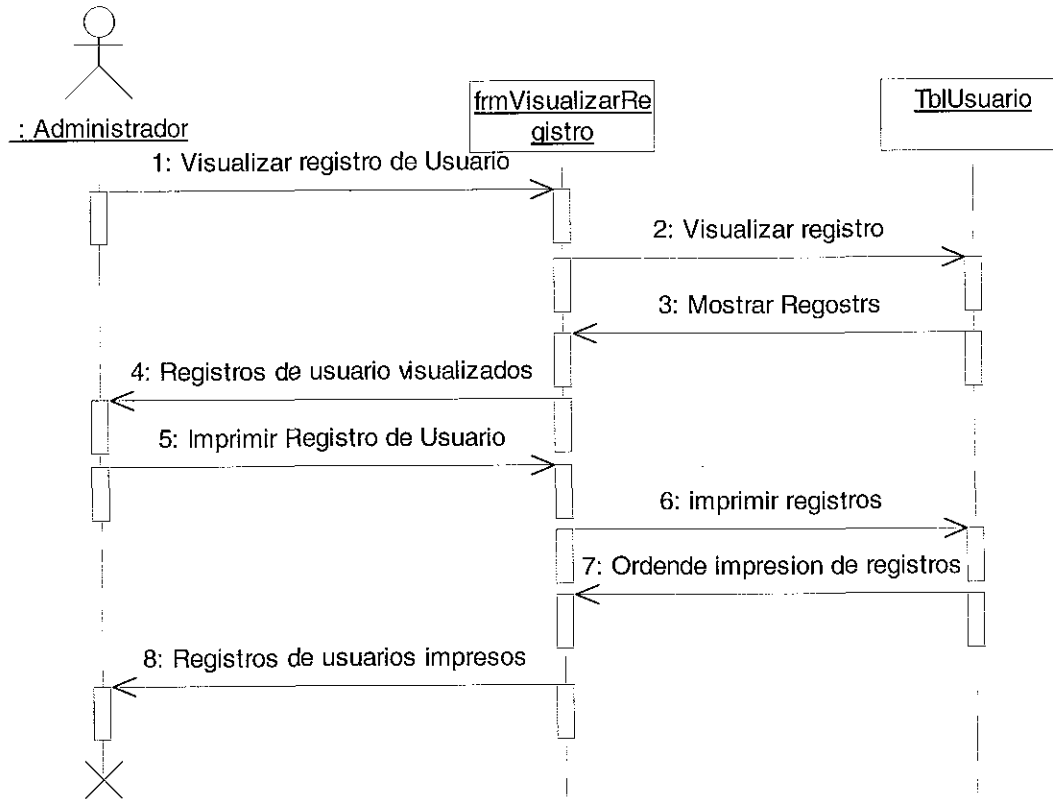
Visualización e Impresión de Registros del Cliente



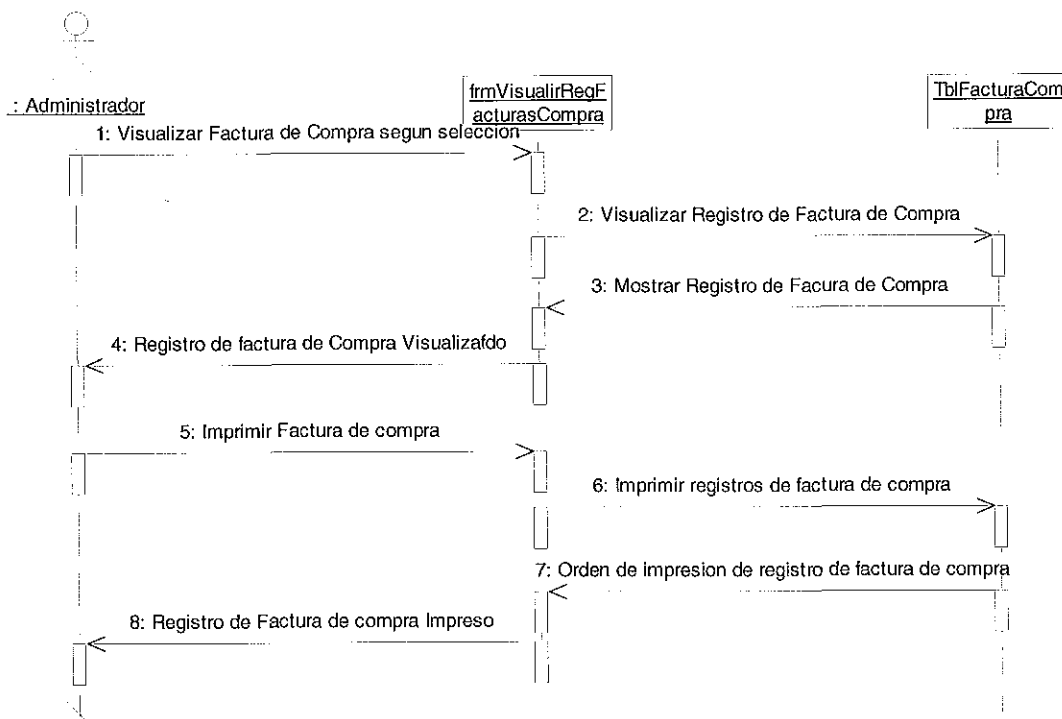
Visualización e Impresión de Registros del Proveedor



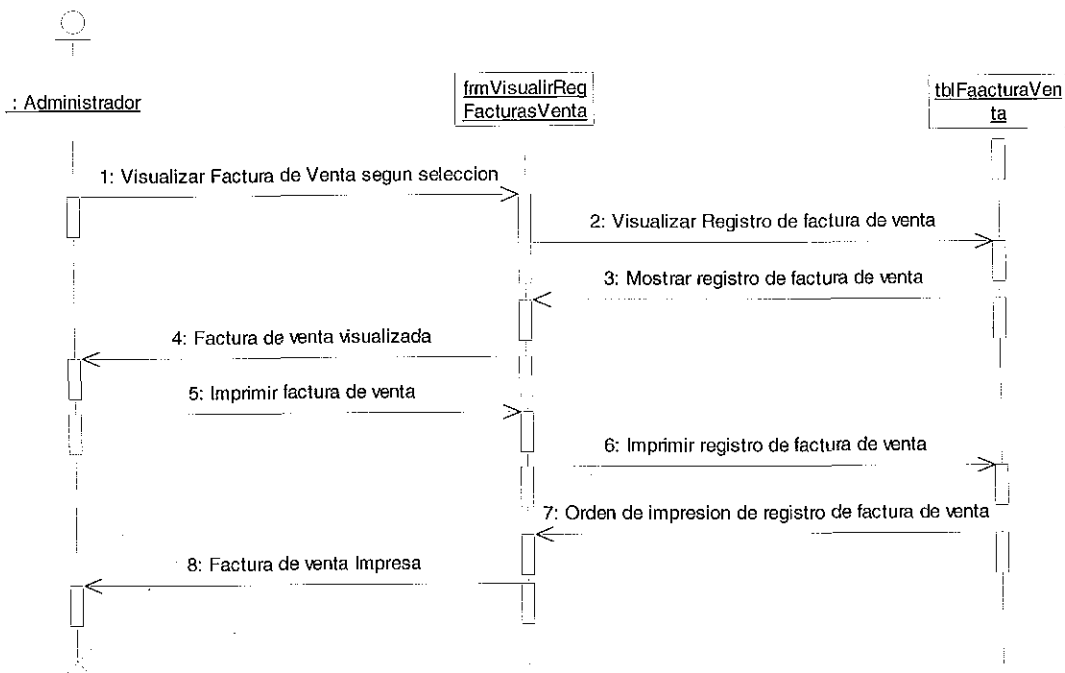
Visualización e Impresión de Registros del Usuario



Visualización e Impresión de Registros Factura de Compra

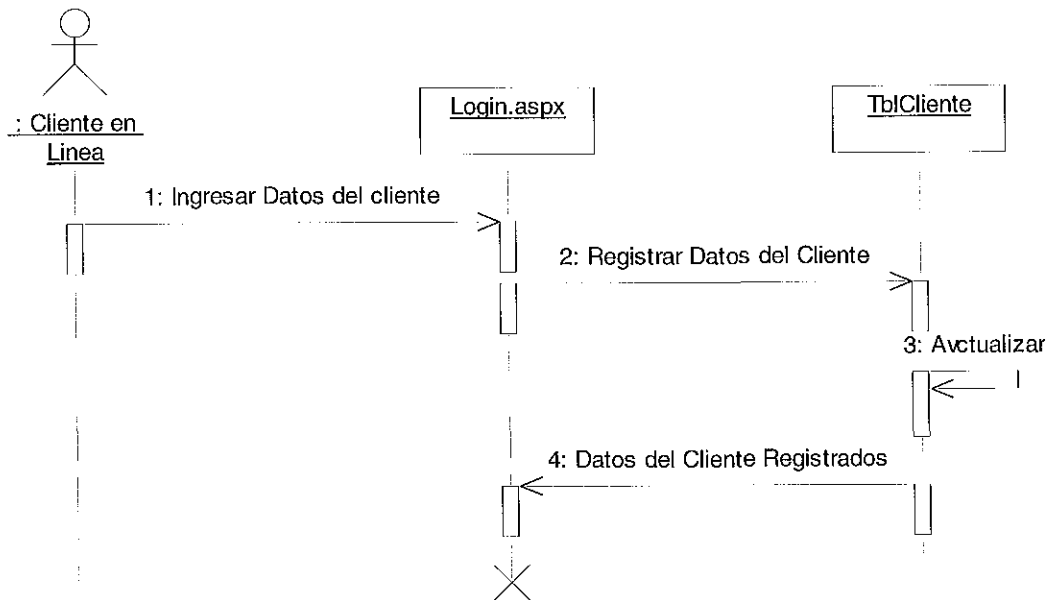


Visualización e Impresión de Registros Factura de Venta

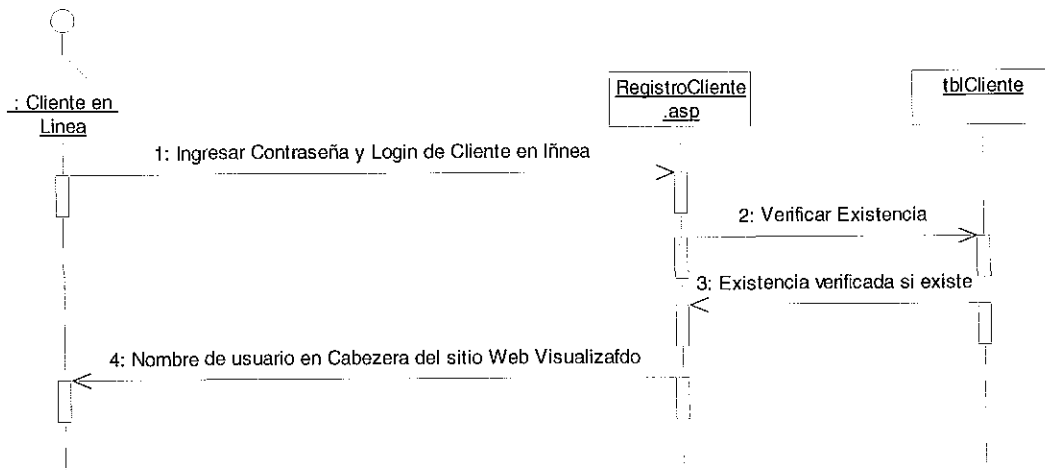


2.3.2 Aplicación Web

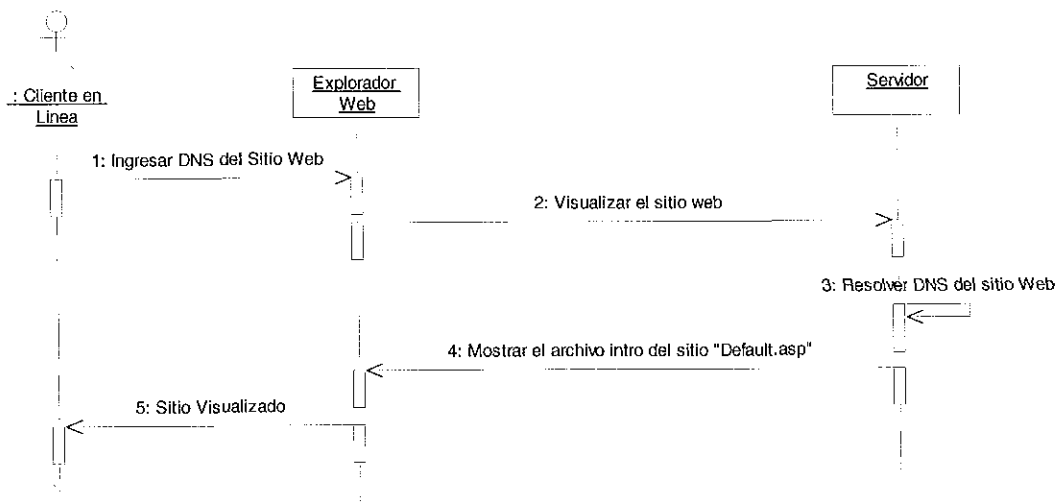
Registro de Cliente en Línea



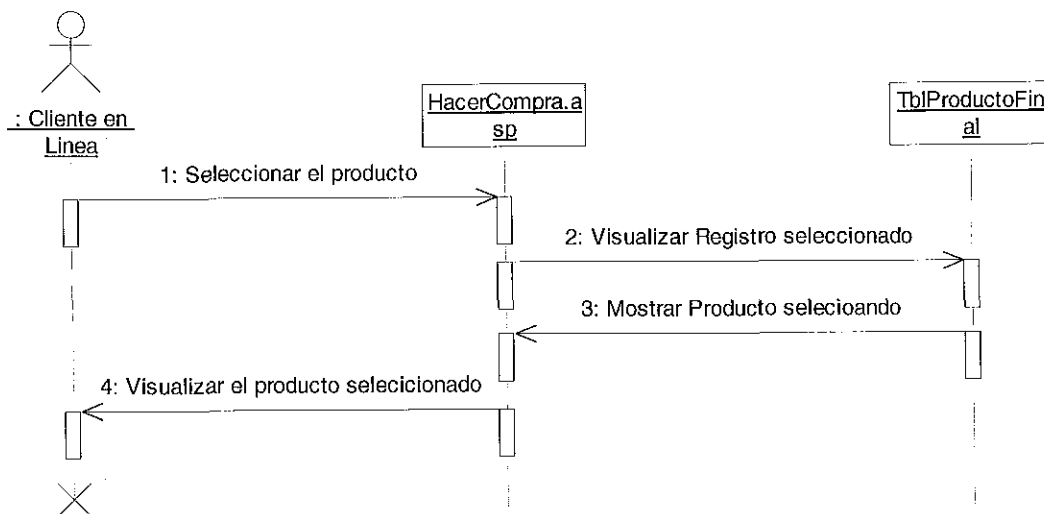
Login de usuario



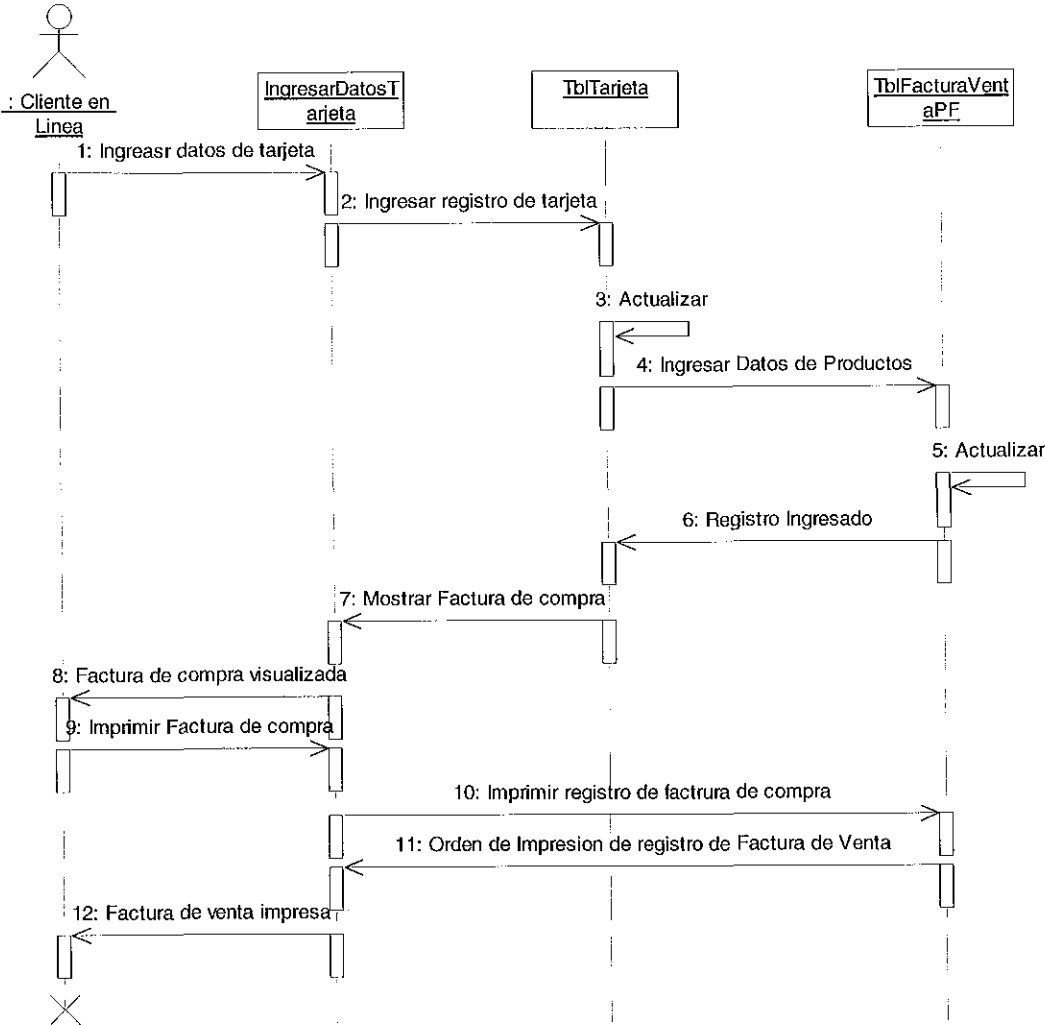
Ingresar al Sitio Web



Seleccionar Productos



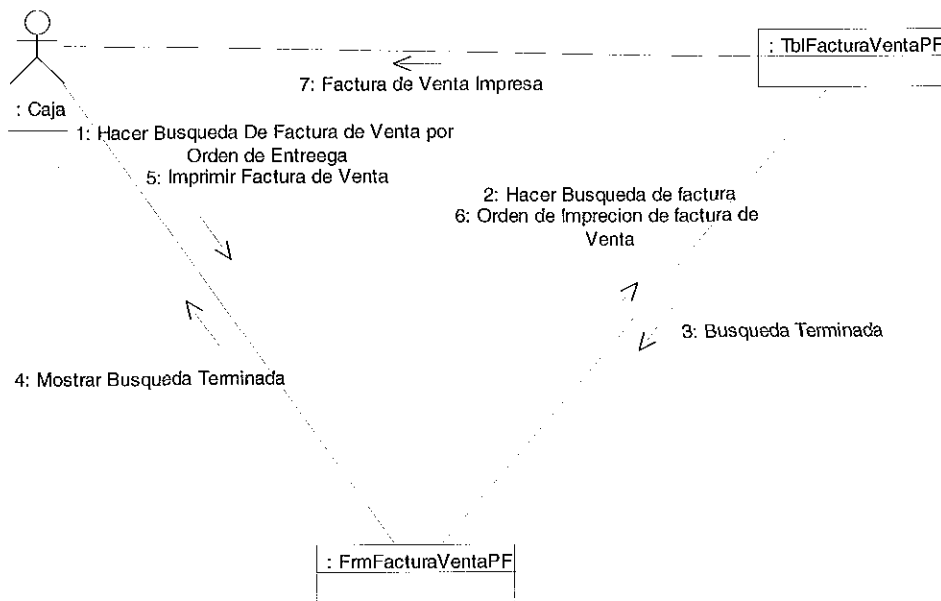
Hacer Pago de la Compra



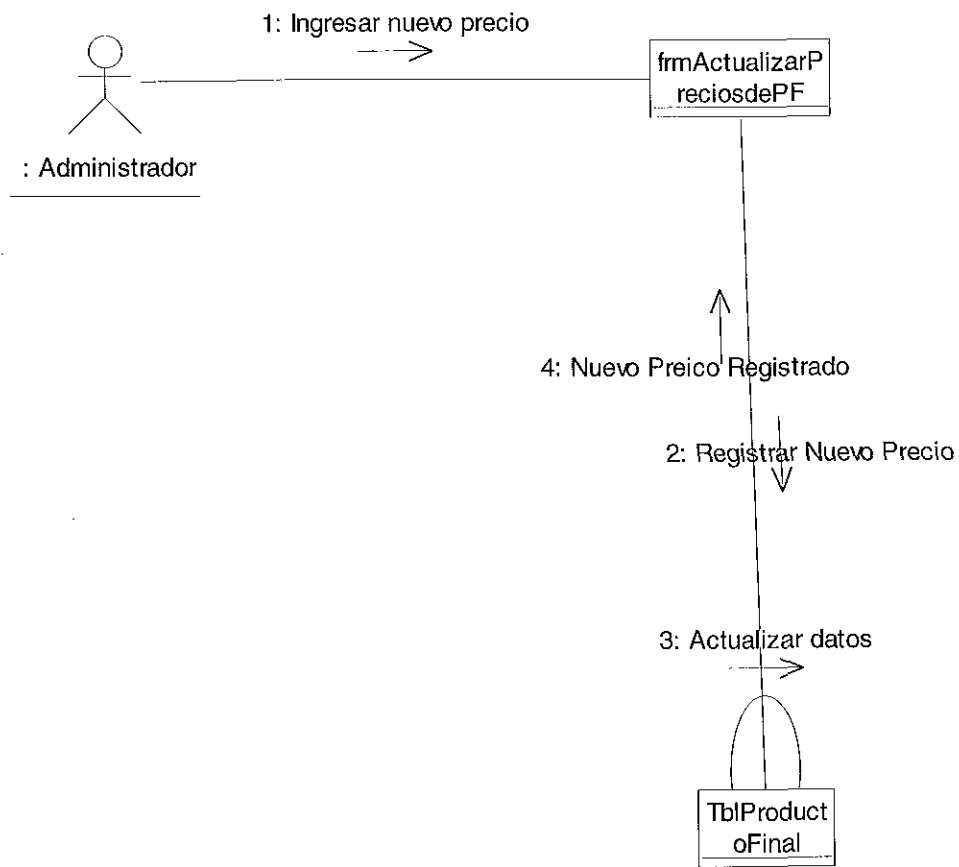
2.4 Diagrama de Colaboración

2.4.1 Aplicación Windows

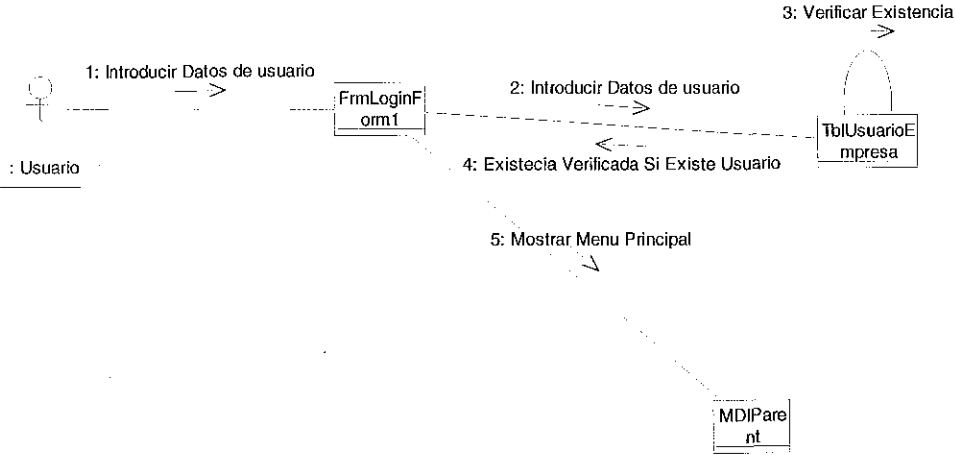
Búsqueda e Impresión de Factura de Venta



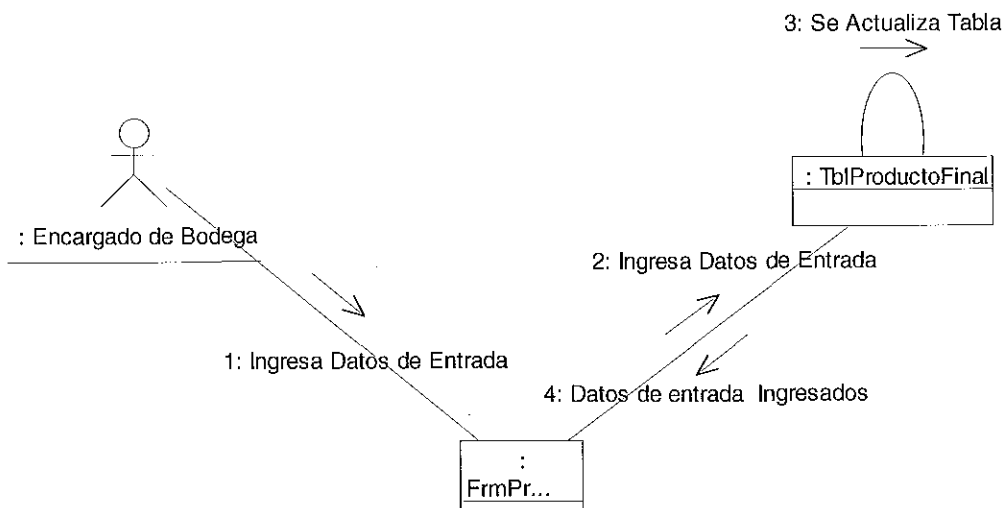
Actualizar Precios del Producto Final



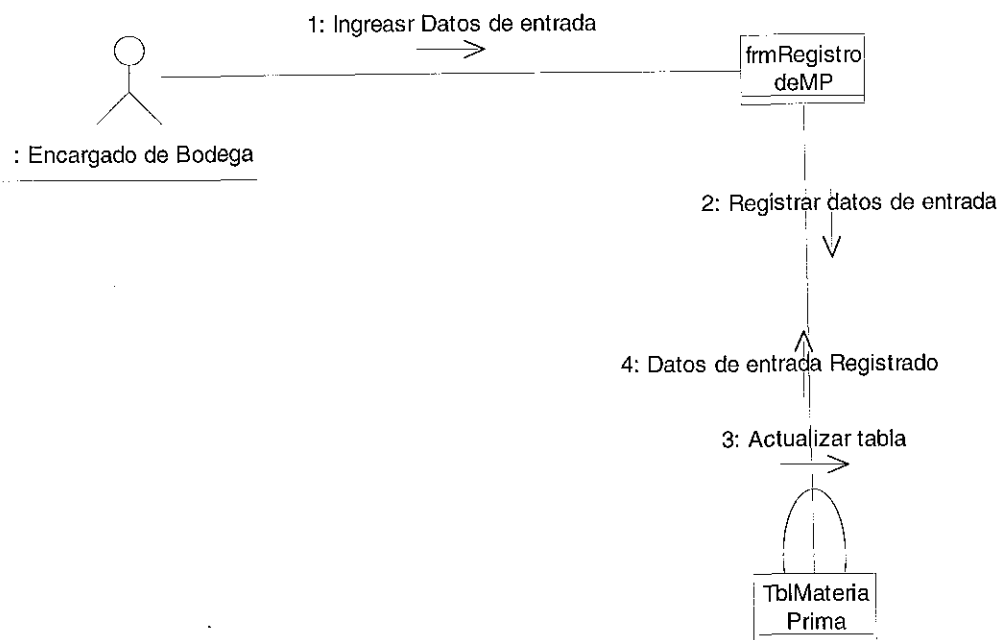
Login Usuario



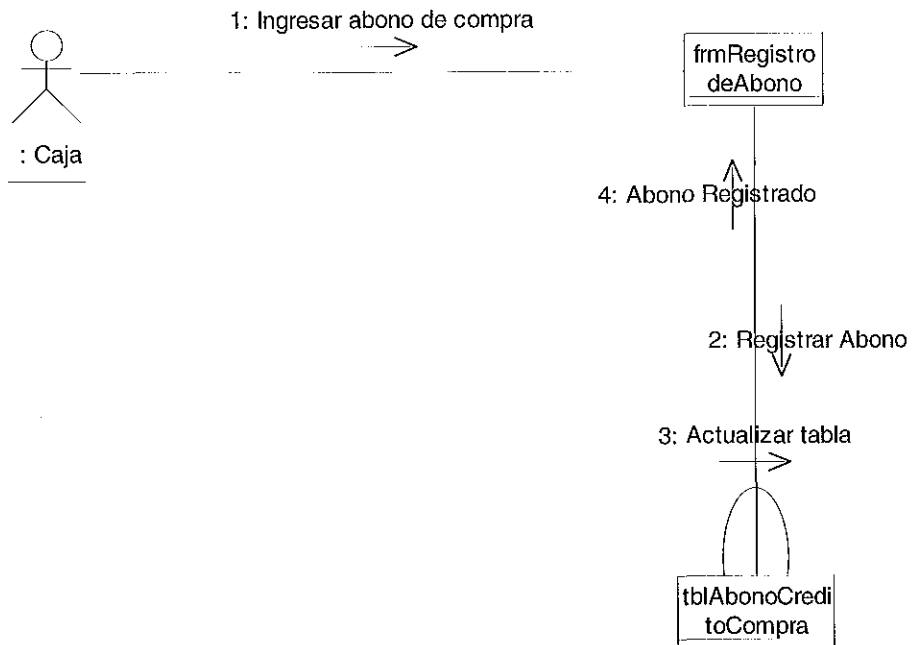
Registro de Producto Final



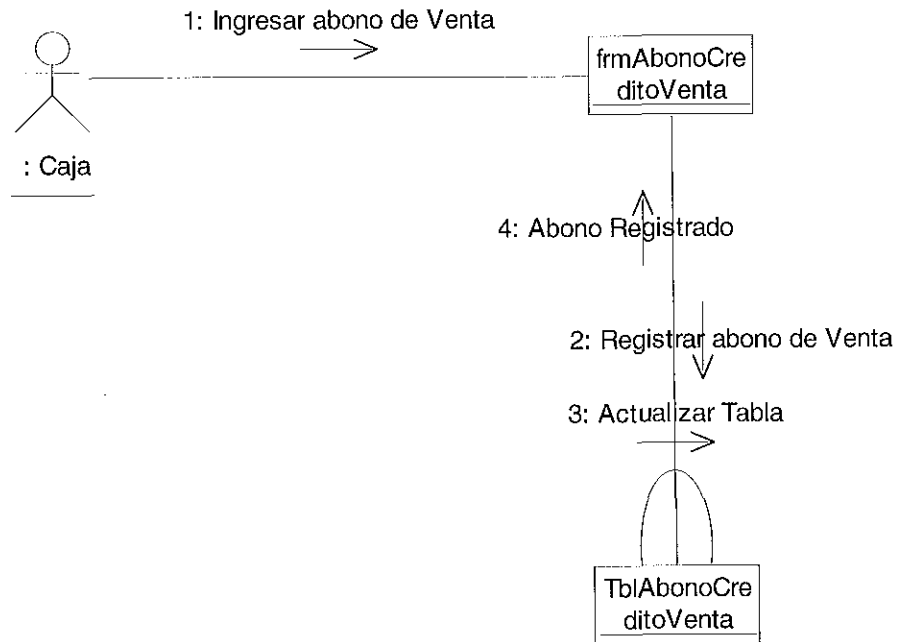
Registro de Materia Prima



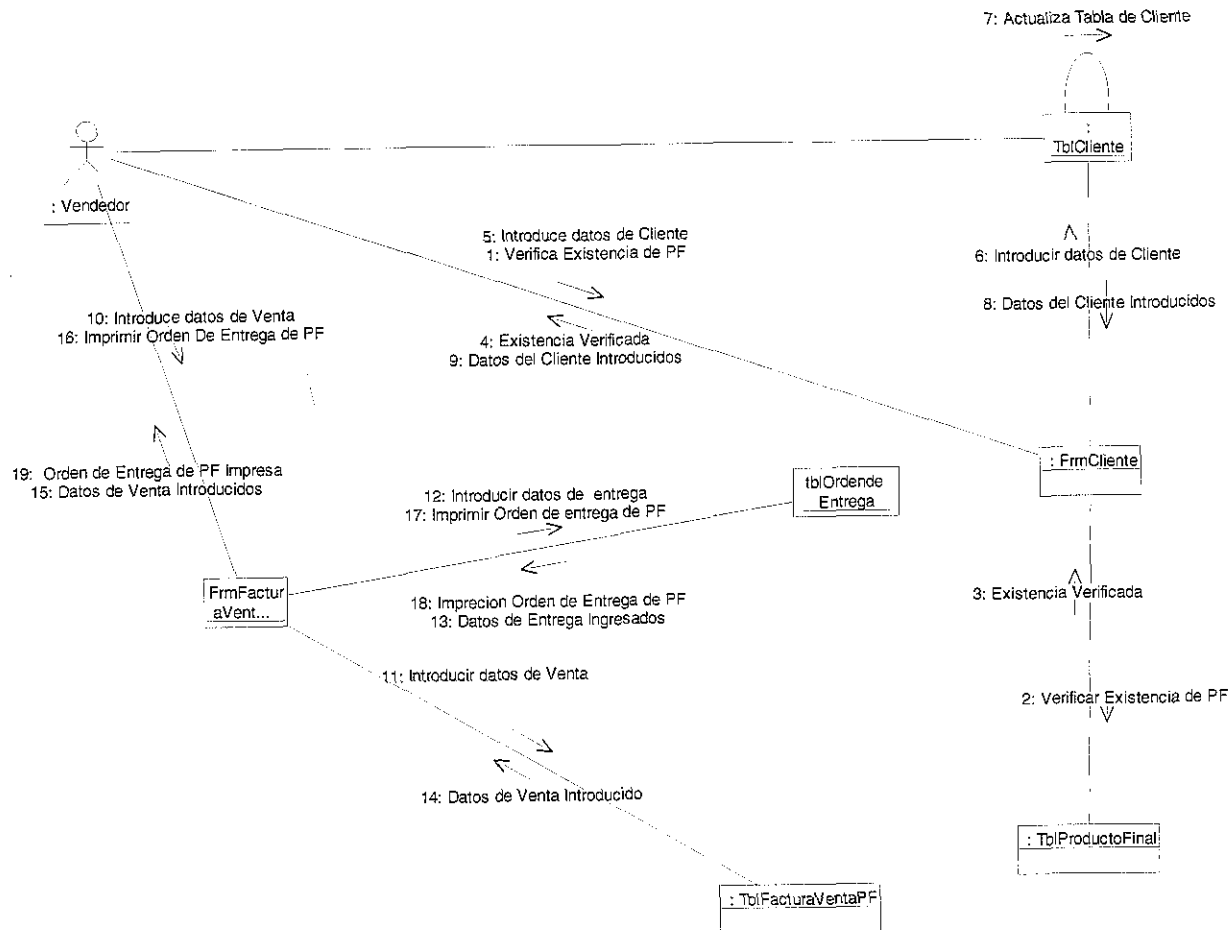
Registro de Abono de Compra



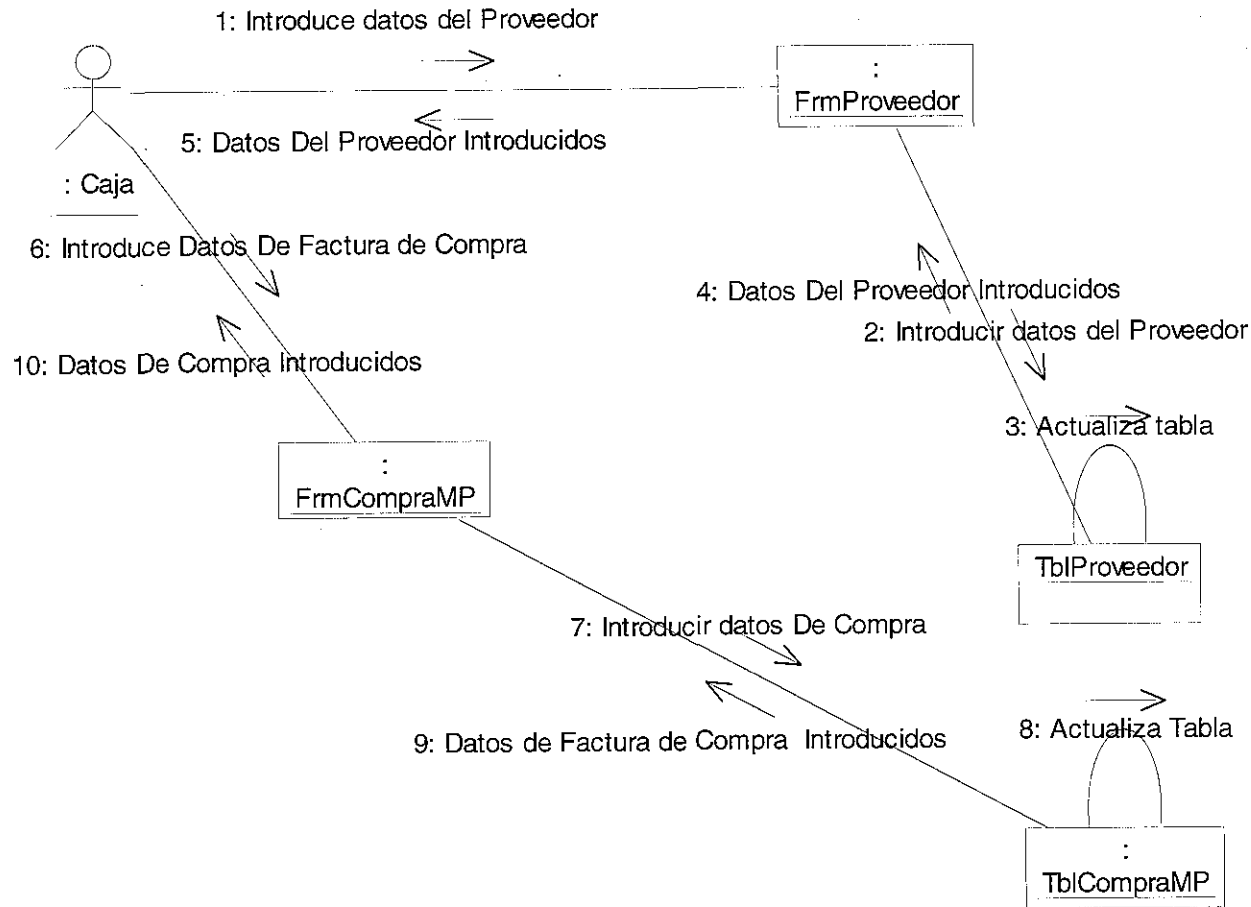
Registro de Abono de Venta



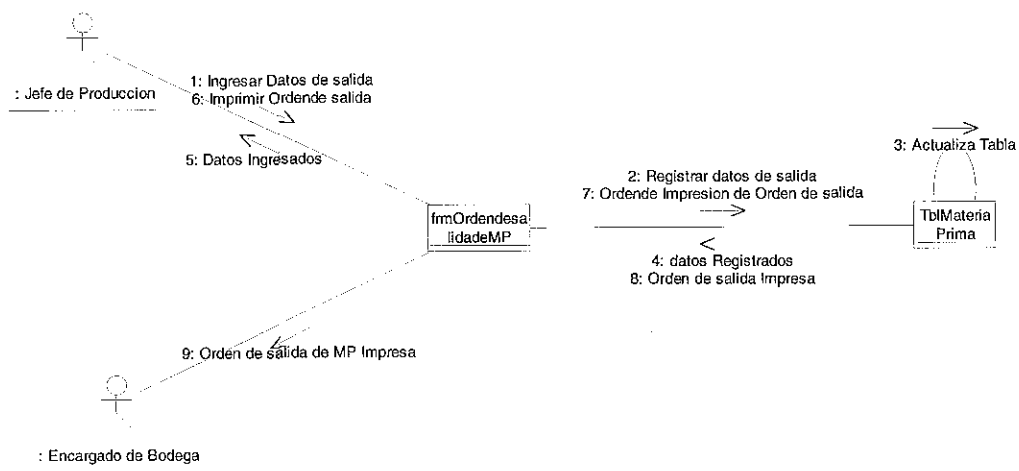
Registro de Cliente y Venta



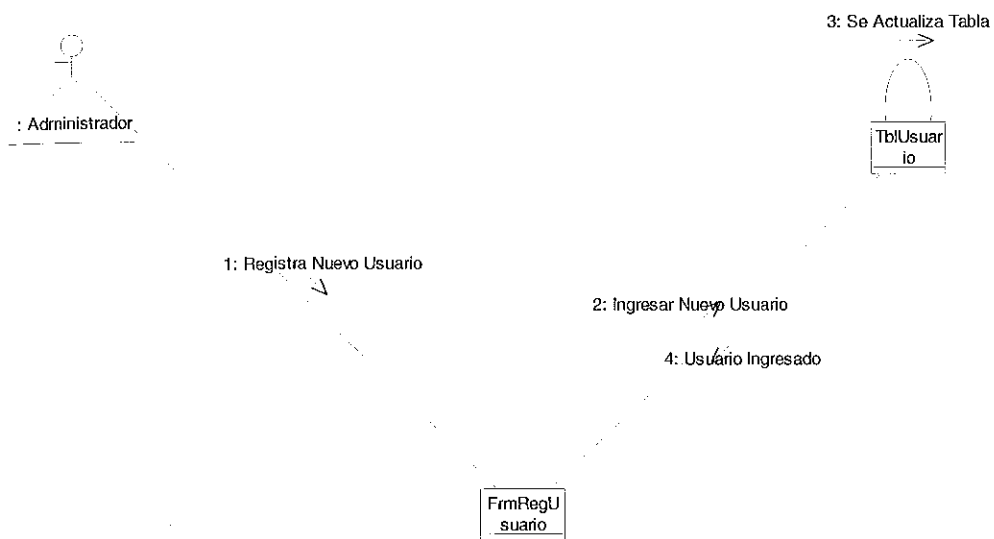
Registro de Proveedor y Compra



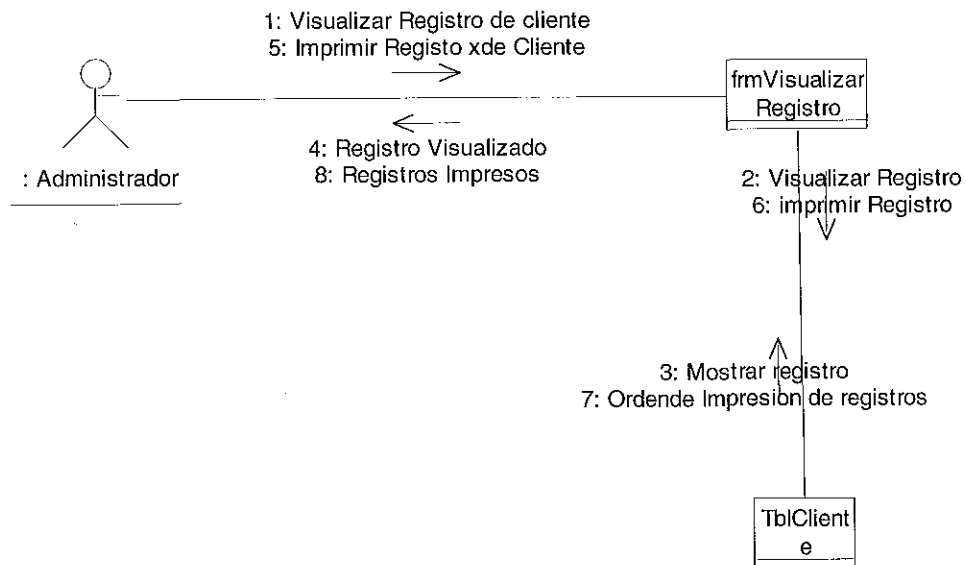
Salida de Materia Prima



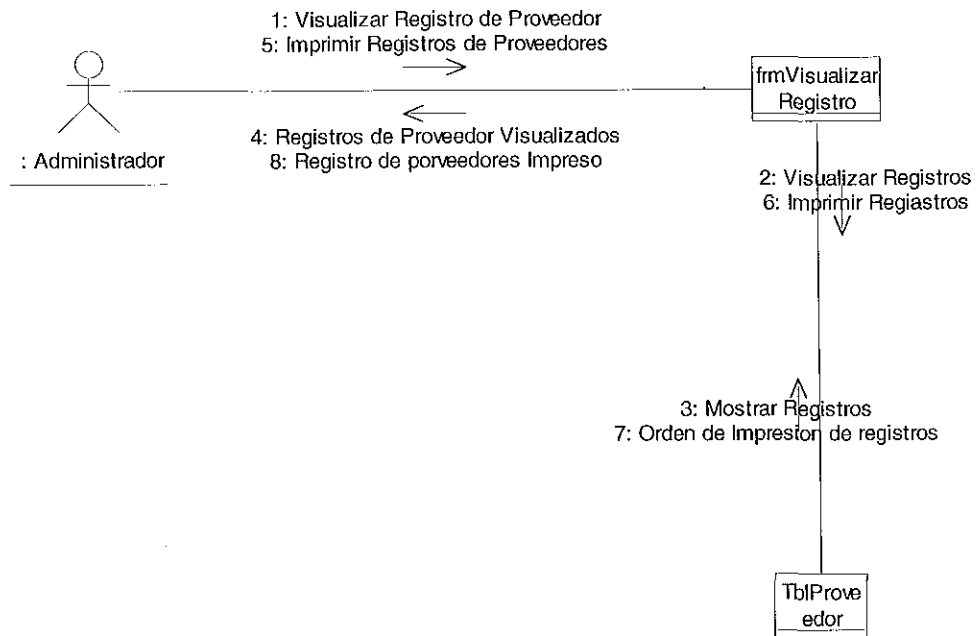
Registro de Usuario



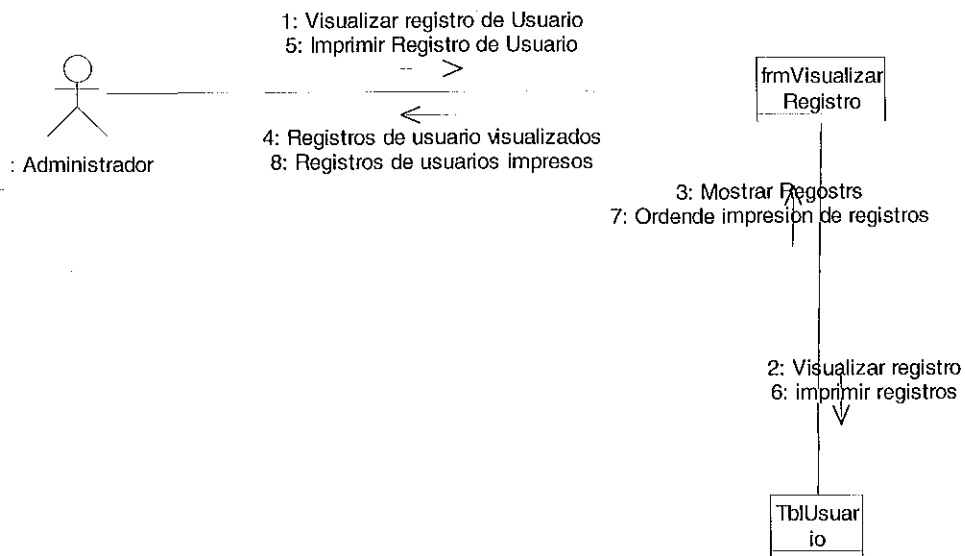
Visualización e Impresión de Registros del Cliente



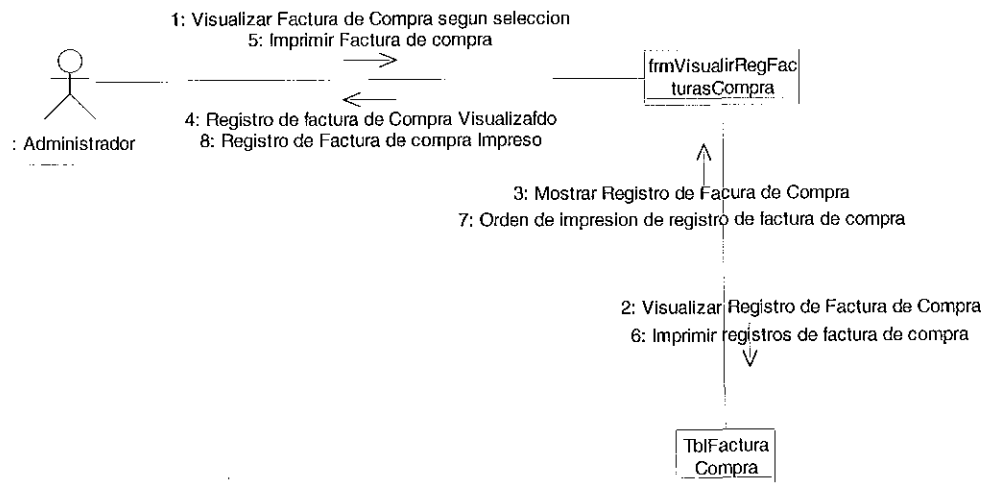
Visualización e Impresión de Registros del Proveedor



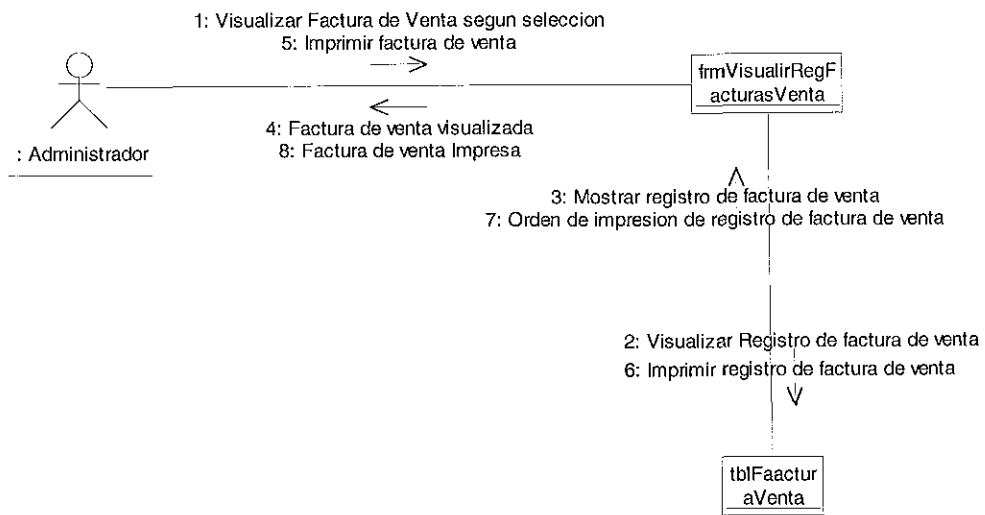
Visualización e Impresión de Registros del Usuario



Visualización e Impresión de Registros Factura de Compra

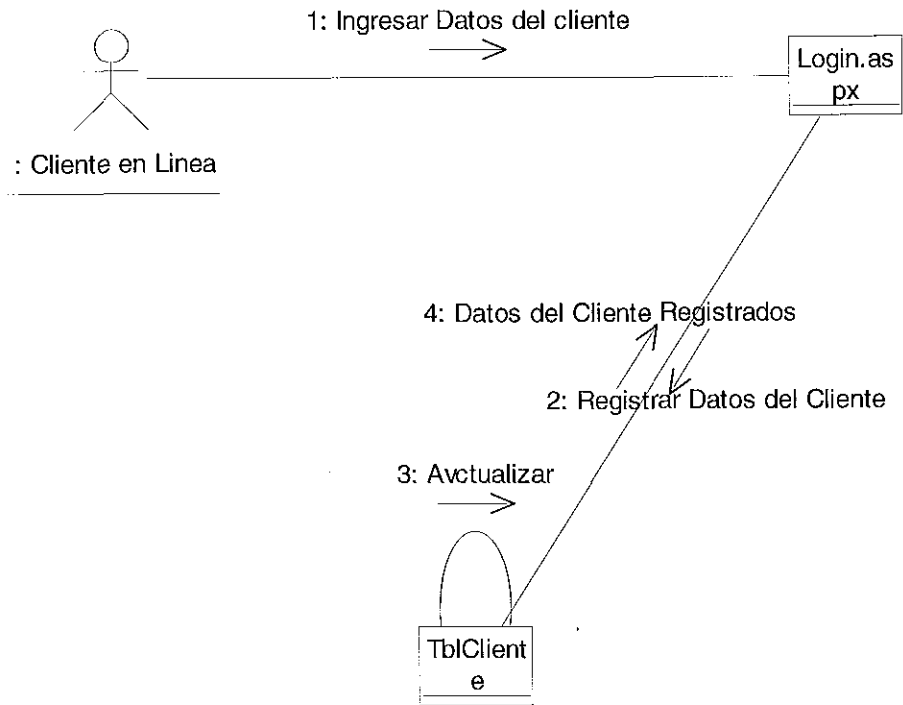


Visualización e Impresión de Registros Factura de Venta

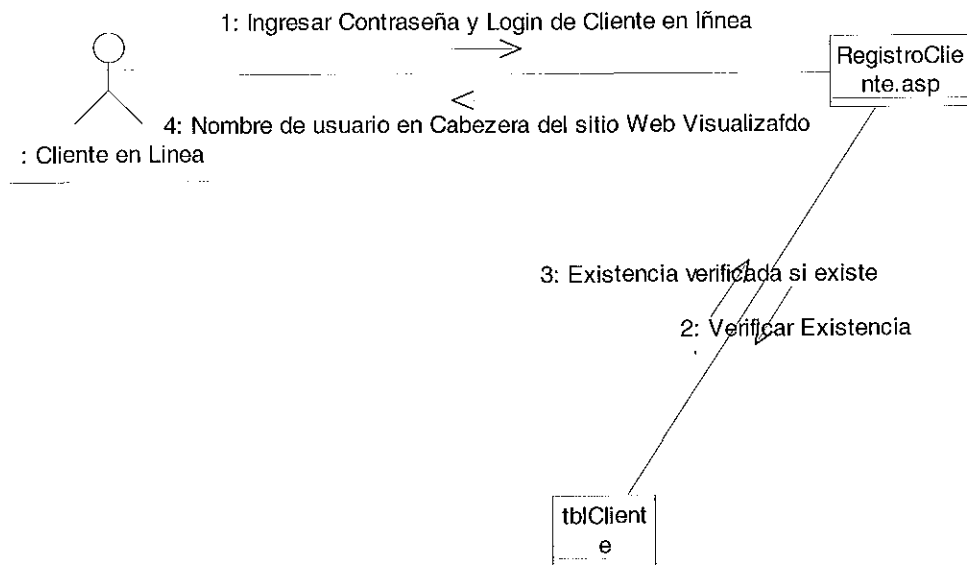


2.4.2 Aplicación Web

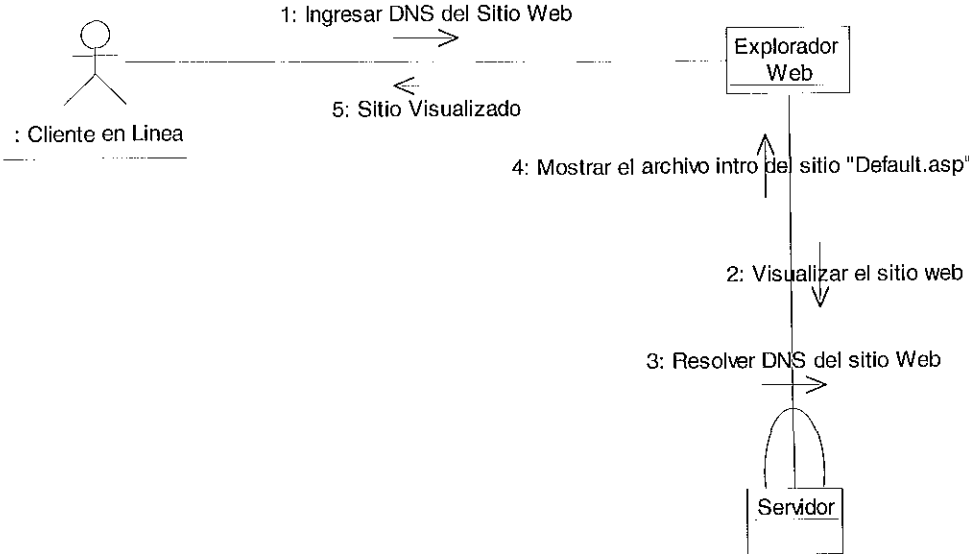
Registro de Cliente en Línea



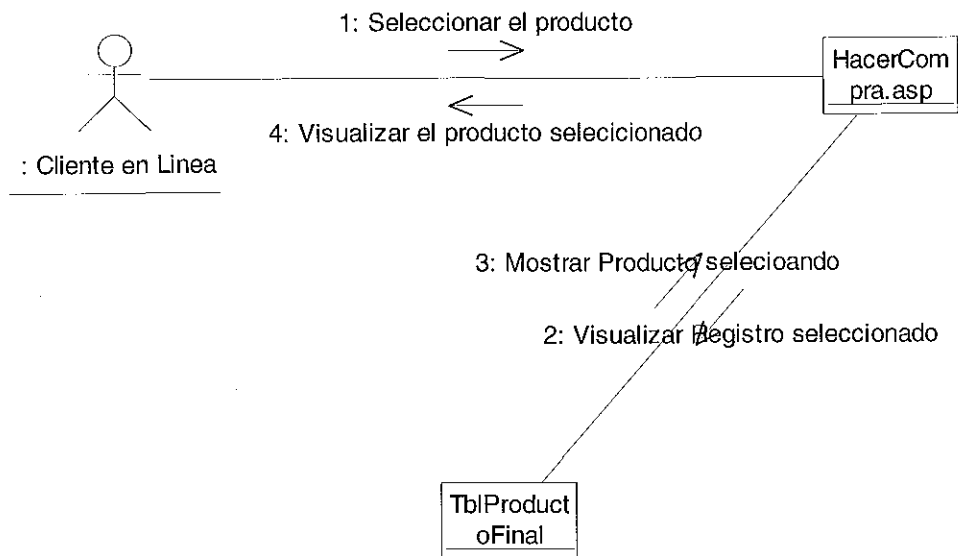
Login del Cliente en Línea



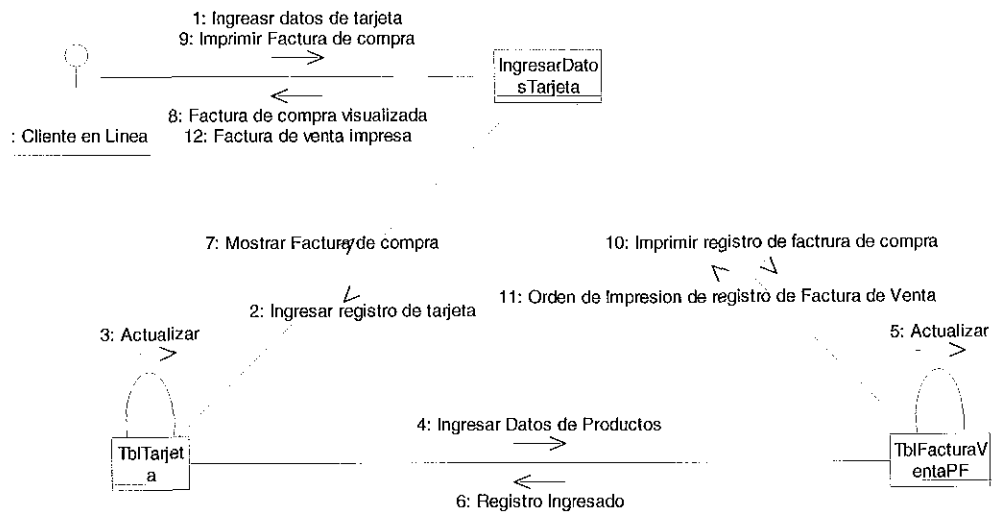
Ingresar al Sitio Web



Seleccionar Productos



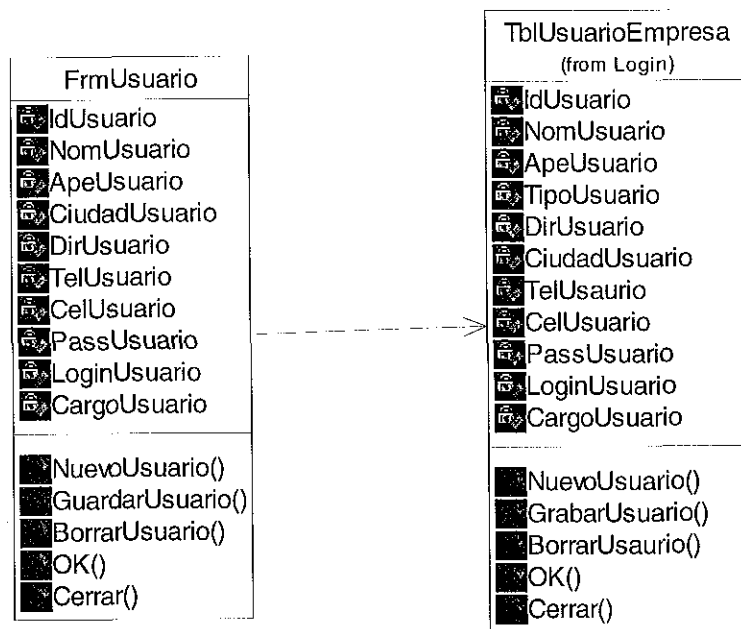
Hacer Pago de la Compra



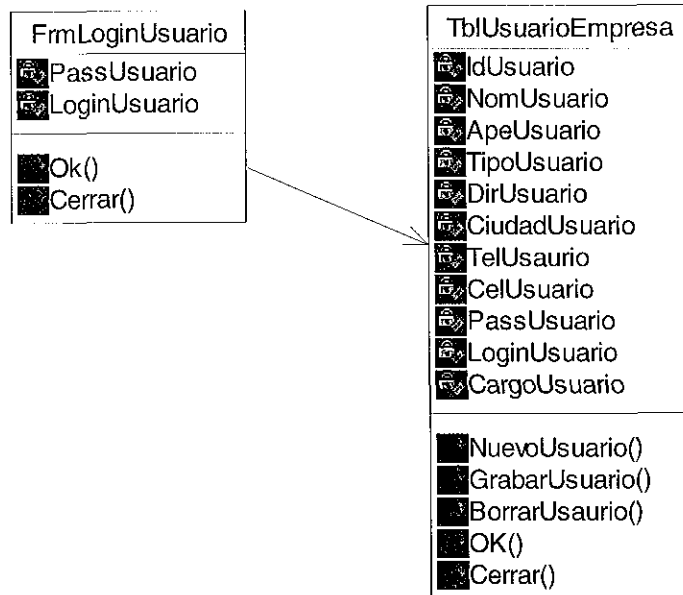
2.5 Diagrama de clases

a. Aplicación Windows

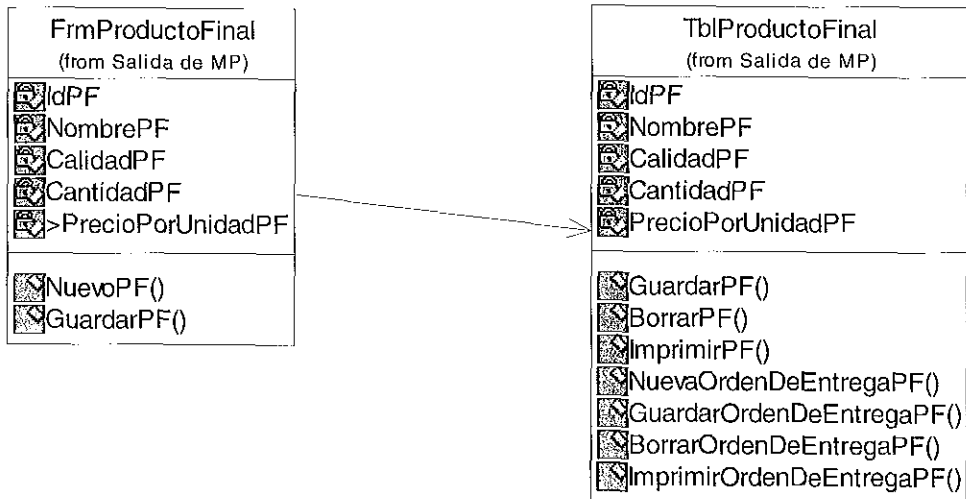
Registrar Usuario



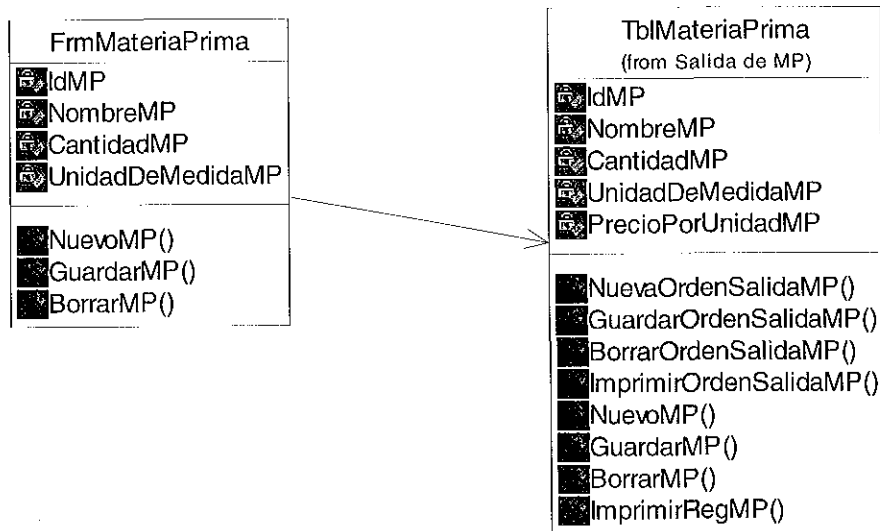
Login



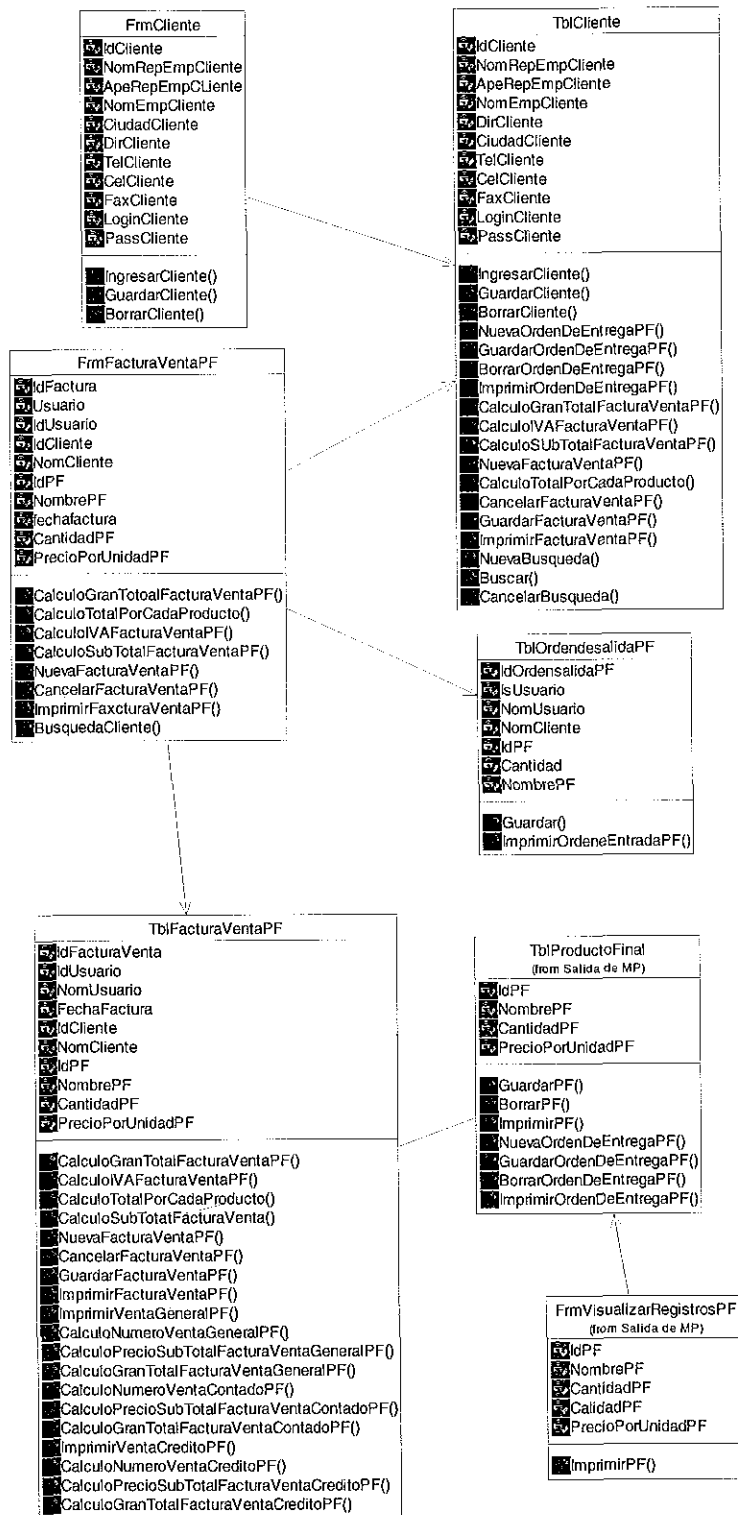
Registro de Producto Final



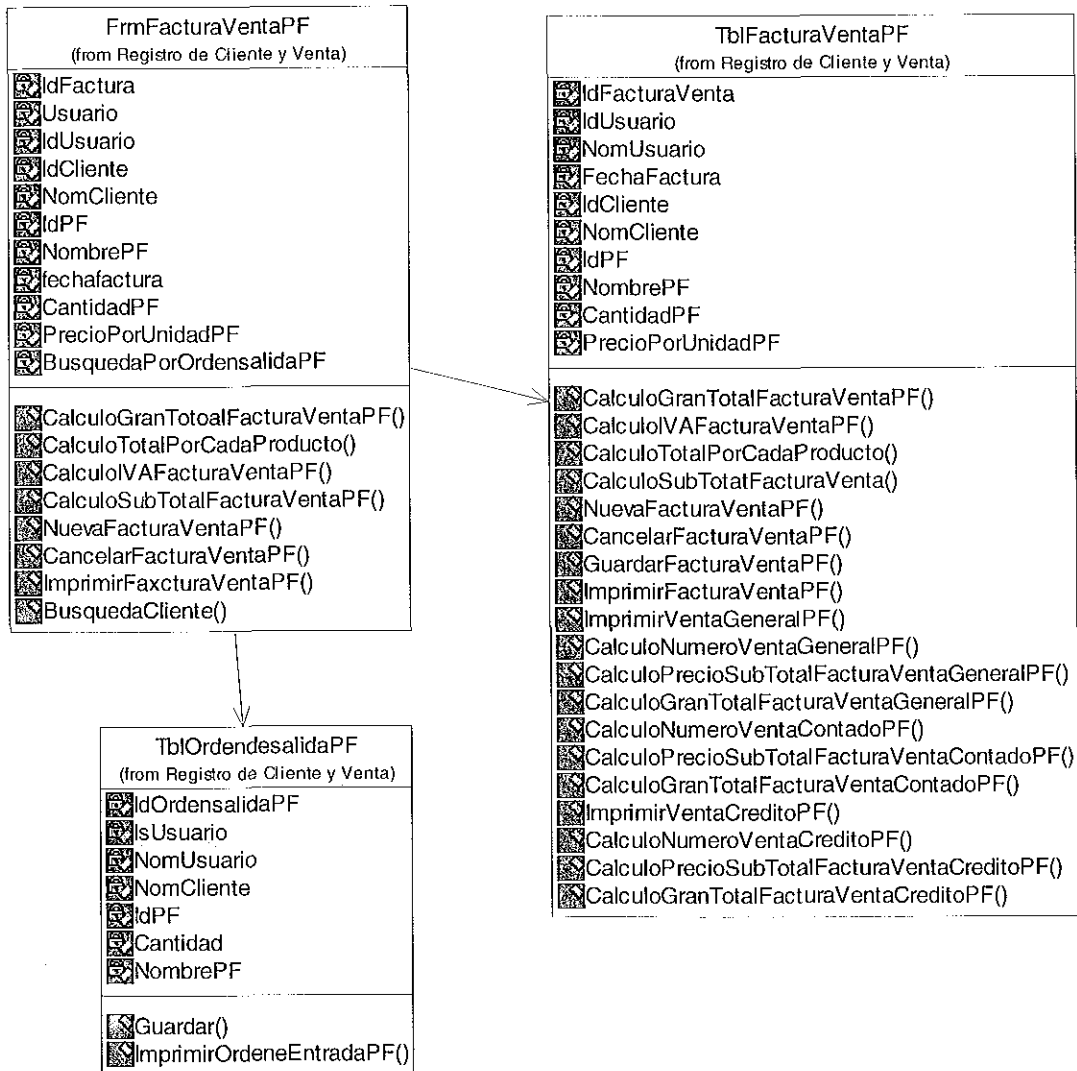
Registro de MP



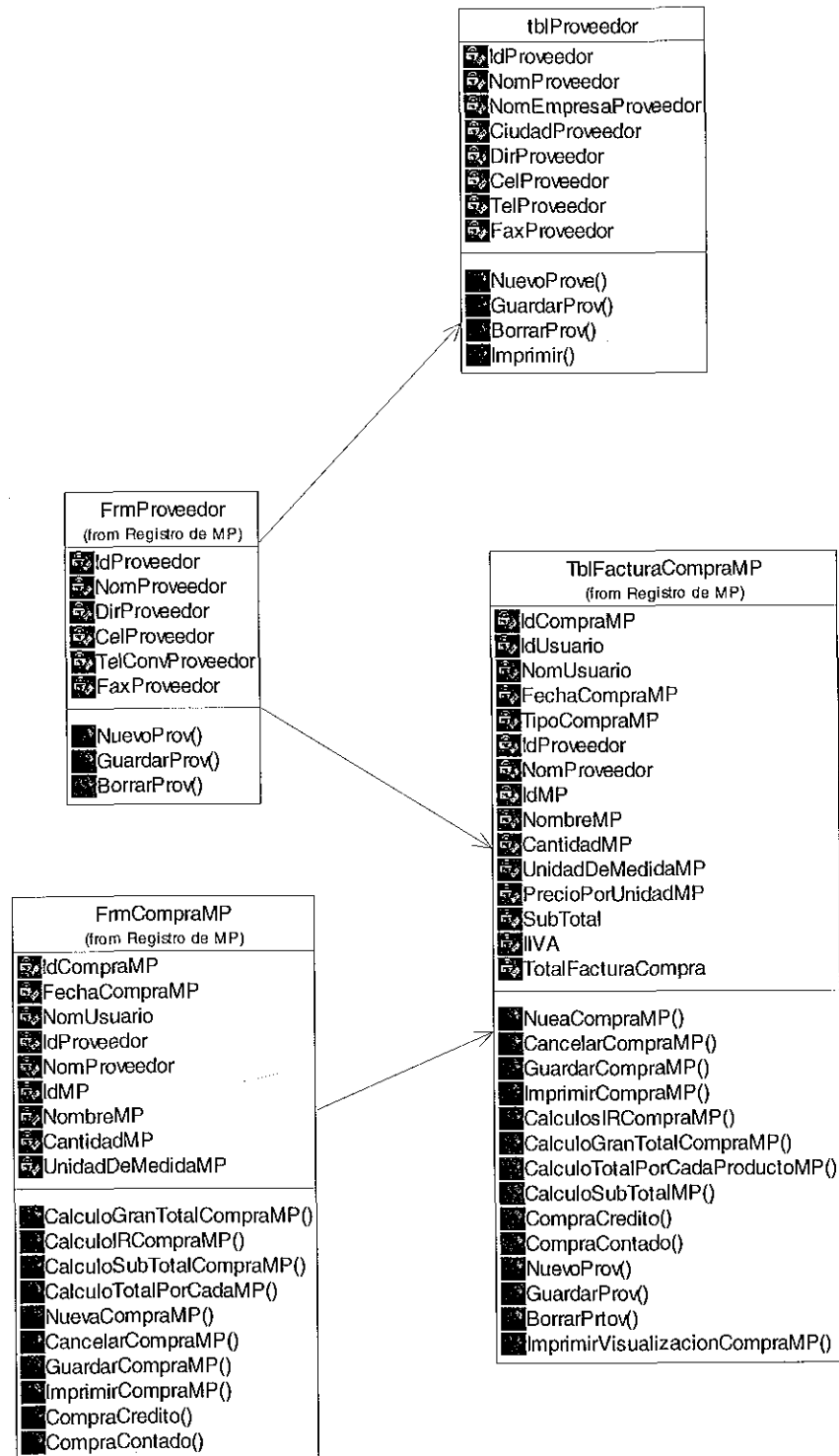
Registro Clientes y Venta



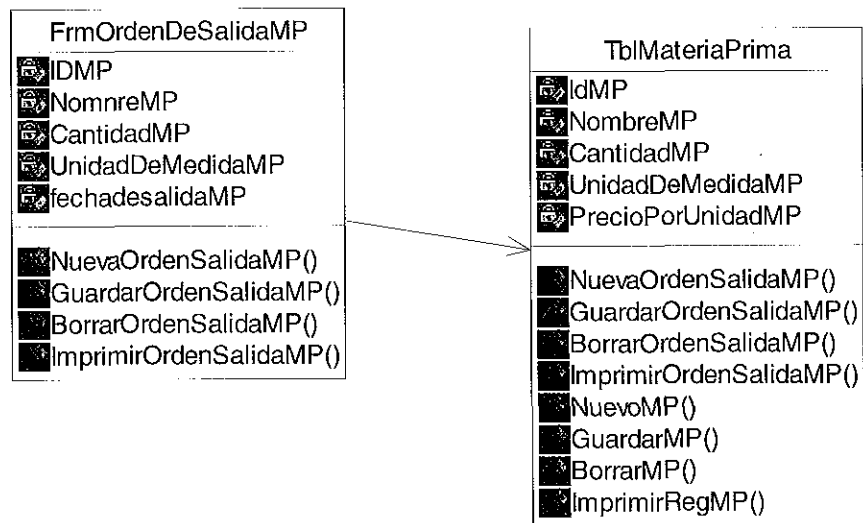
Busqueda e Impresion de Factura de Venta de PF



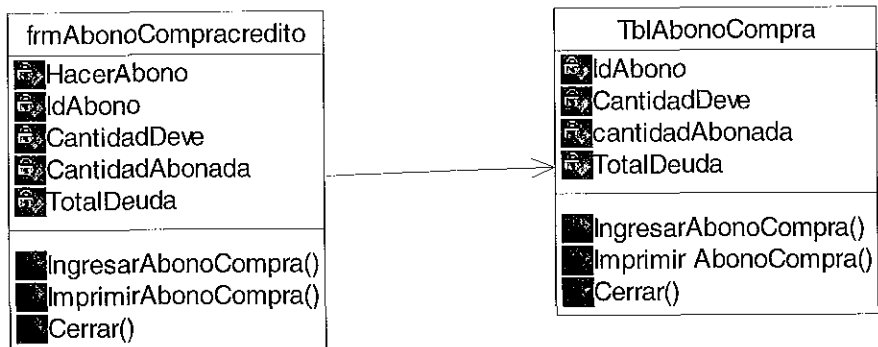
Registro de Proveedor y compra



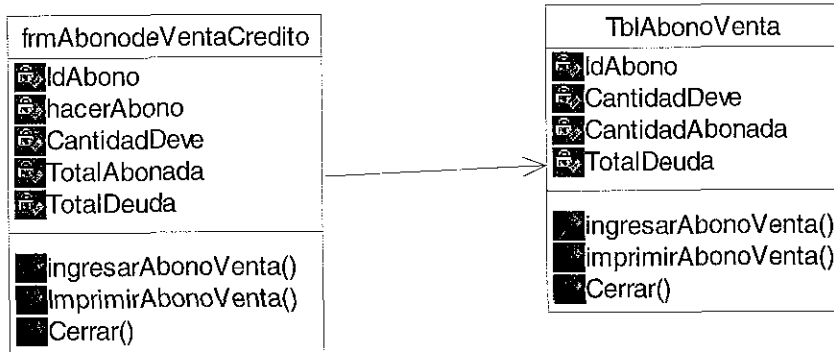
Salida de materia Prima



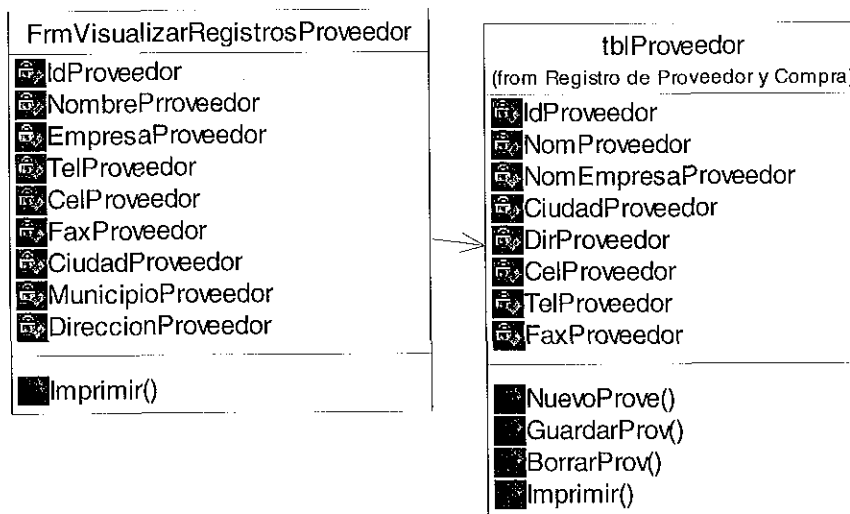
Registro de Abono de Compra



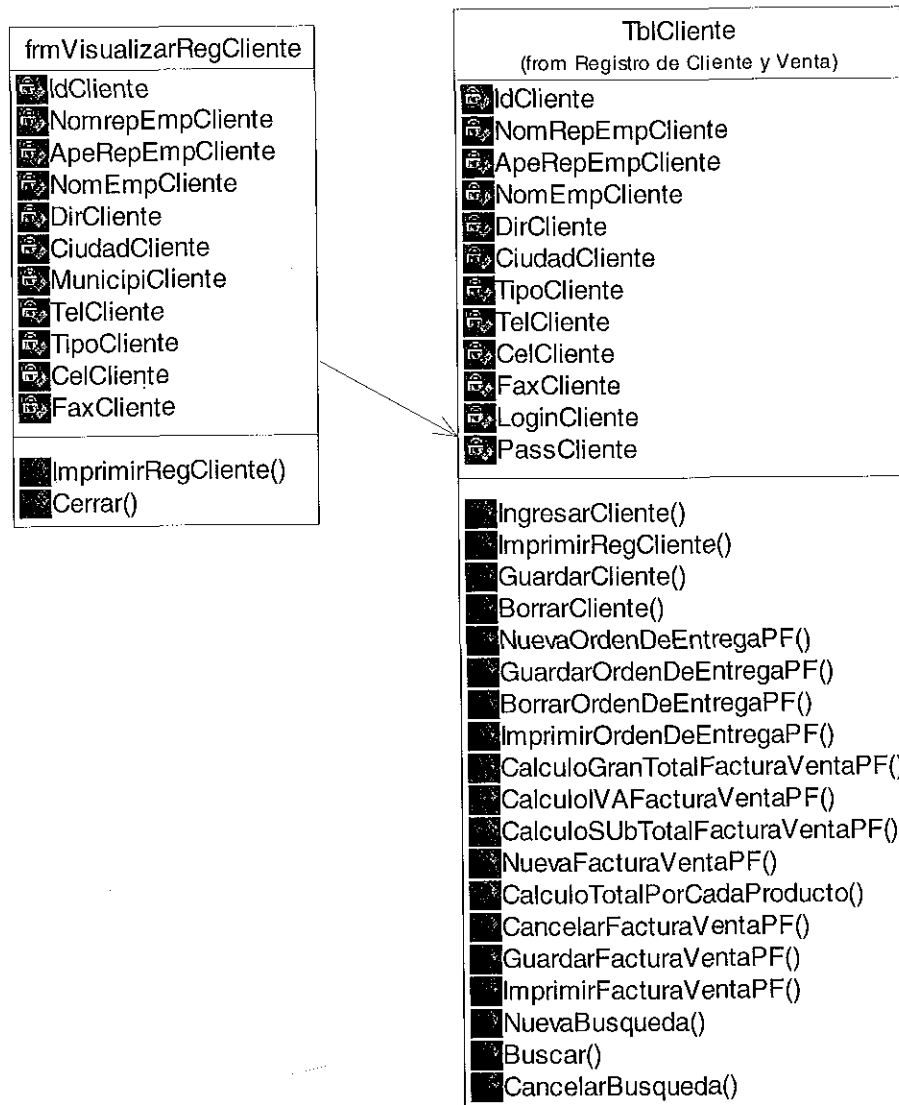
Registro de Abono de Venta



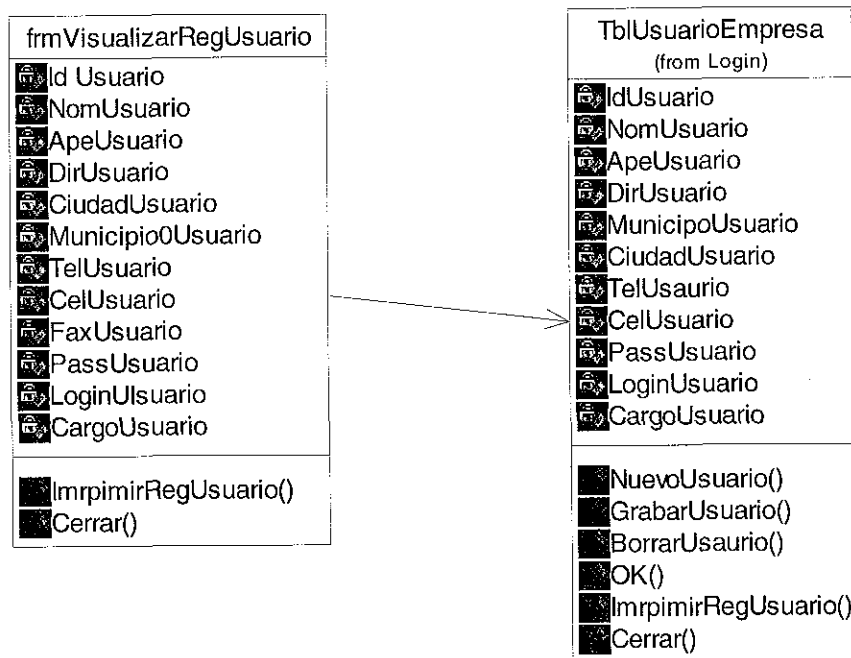
Visualizar e Imprimir Registro de Proveedor



Visualizar e Imprimir Registro de Cliente



Visualizar e Imprimir Registro de Usuario

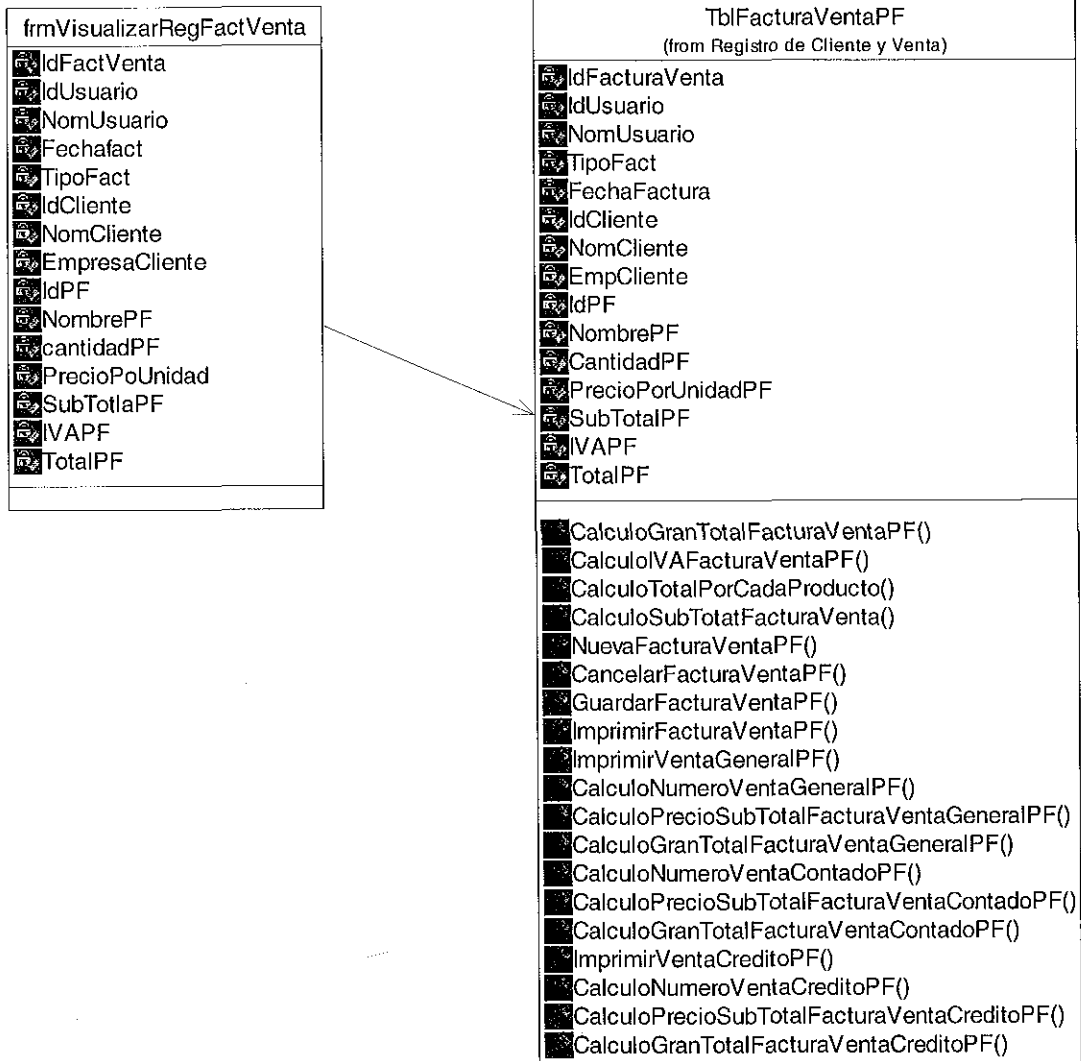


Visualizar e Imprimir Registro de Factura de Compra

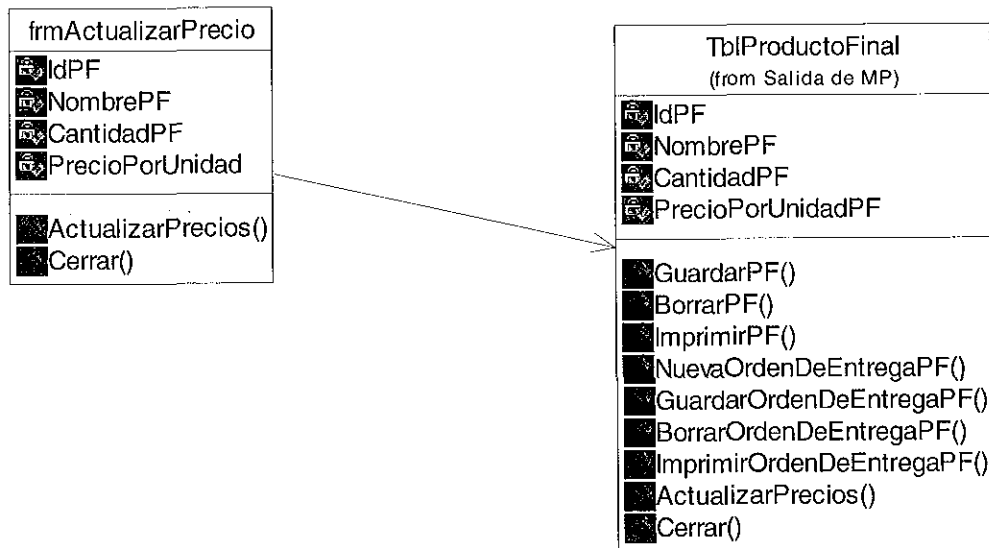
frmRegFacturaCompra
IdCompraMP
IdUsuario
NomUsuario
FechaCompra
TipoCompra
IdProveedor
NomProveedor
EmpresaProveedor
IdMP
NomMP
cantidadMP
UnidaddeMedidaMP
PrecioPorUnidadMP
SubTotal
IVA
TotalFactCompra
ImprimirFactCompraMP()
Cerrar()

TblFacturaCompraMP (from Registro de MP)
IdCompraMP
IdUsuario
NomUsuario
FechaCompraMP
TipoCompraMP
IdProveedor
NomProveedor
IdMP
NombreMP
CantidadMP
UnidadDeMedidaMP
PrecioPorUnidadMP
SubTotal
IIVA
TotalFacturaCompra
NueaCompraMP()
CancelarCompraMP()
GuardarCompraMP()
ImprimirCompraMP()
CalculosIRCompraMP()
CalculoGranTotalCompraMP()
CalculoTotalPorCadaProductoMP()
CalculoSubTotalMP()
CompraCredito()
CompraContado()
NuevoProv()
GuardarProv()
BorrarPrtov()
ImprimirVisualizacionCompraMP()

Visualizar e Imprimir Registro de Factura de Venta

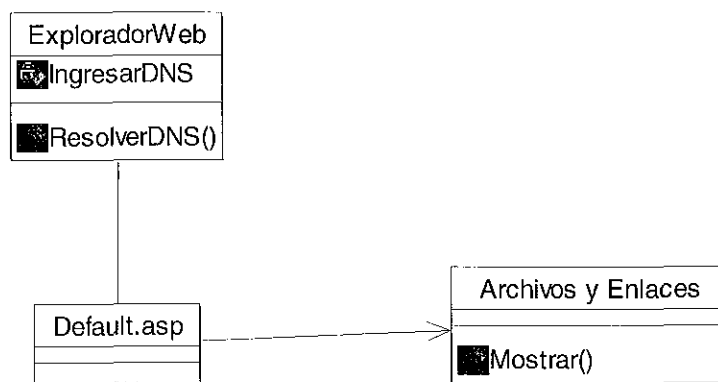


Actualizar Precio

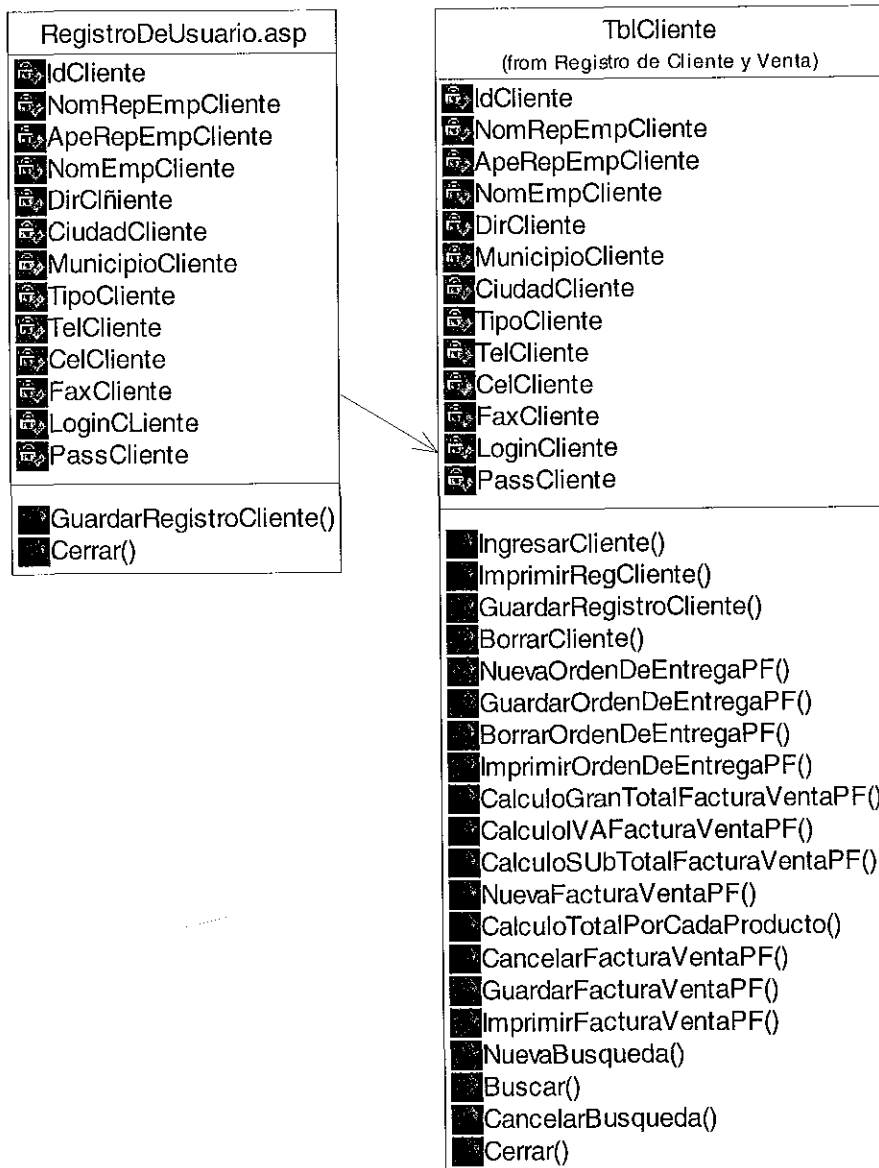


b. Aplicación Web

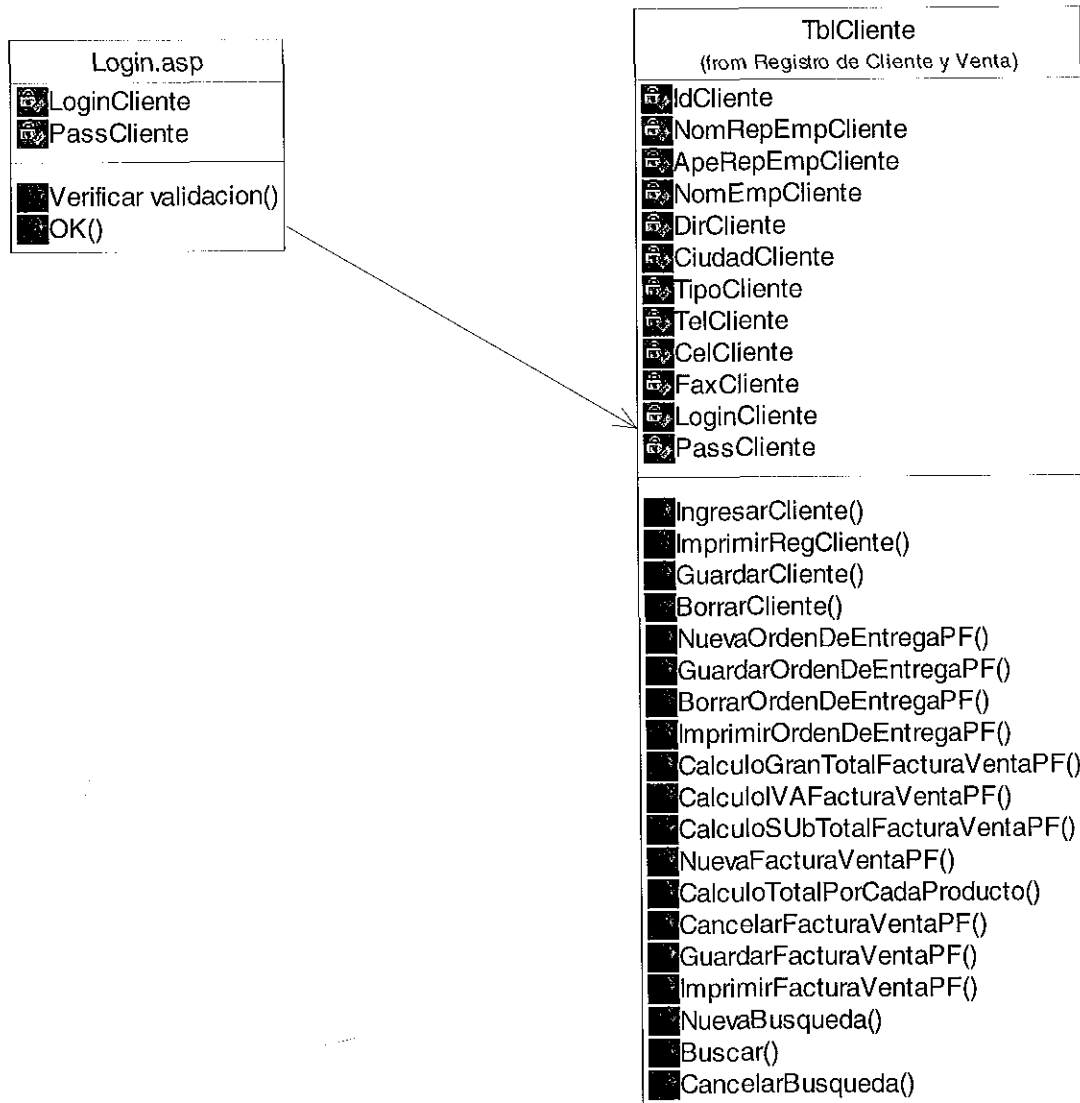
Ingresar al Sitio Web



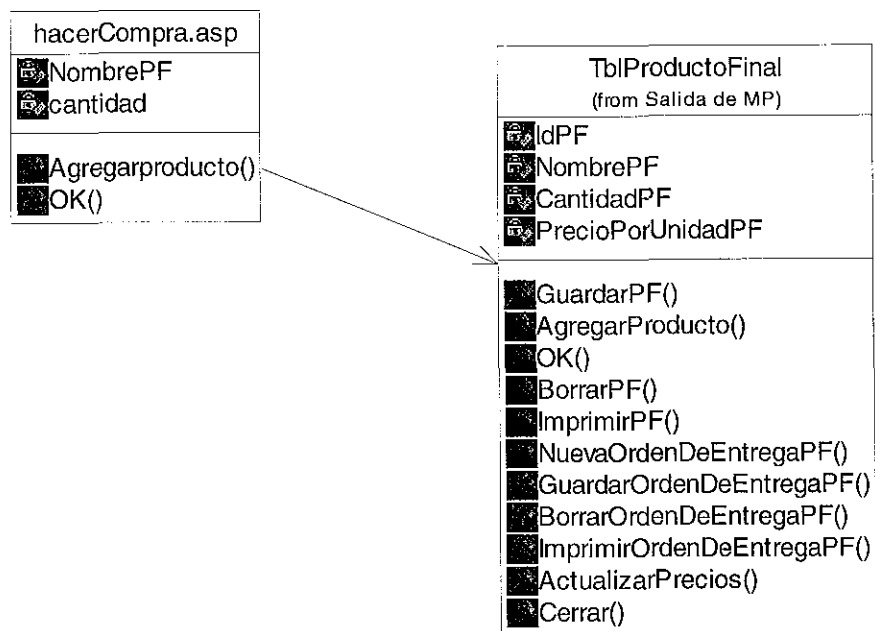
Registro del Cliente



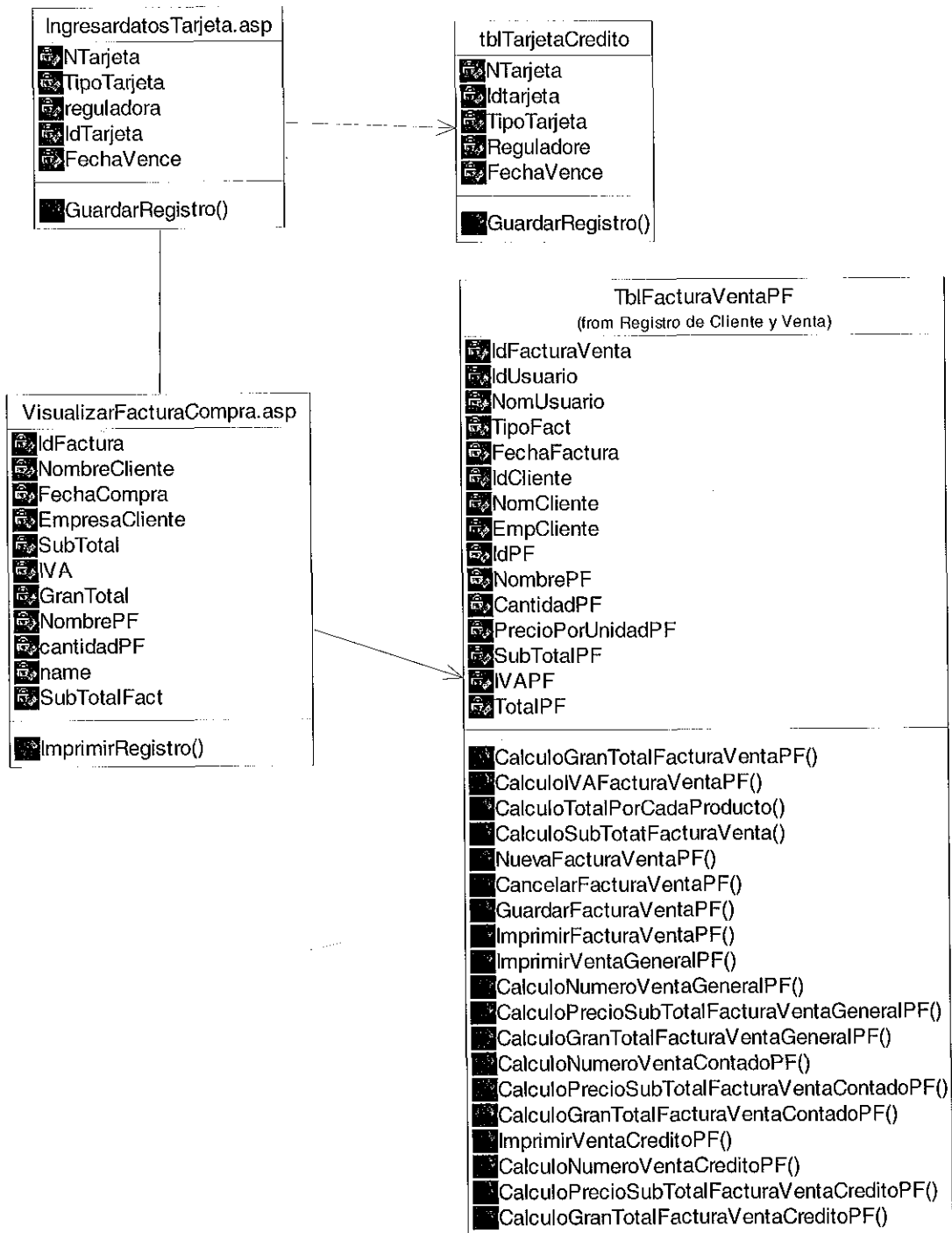
Login Cliente



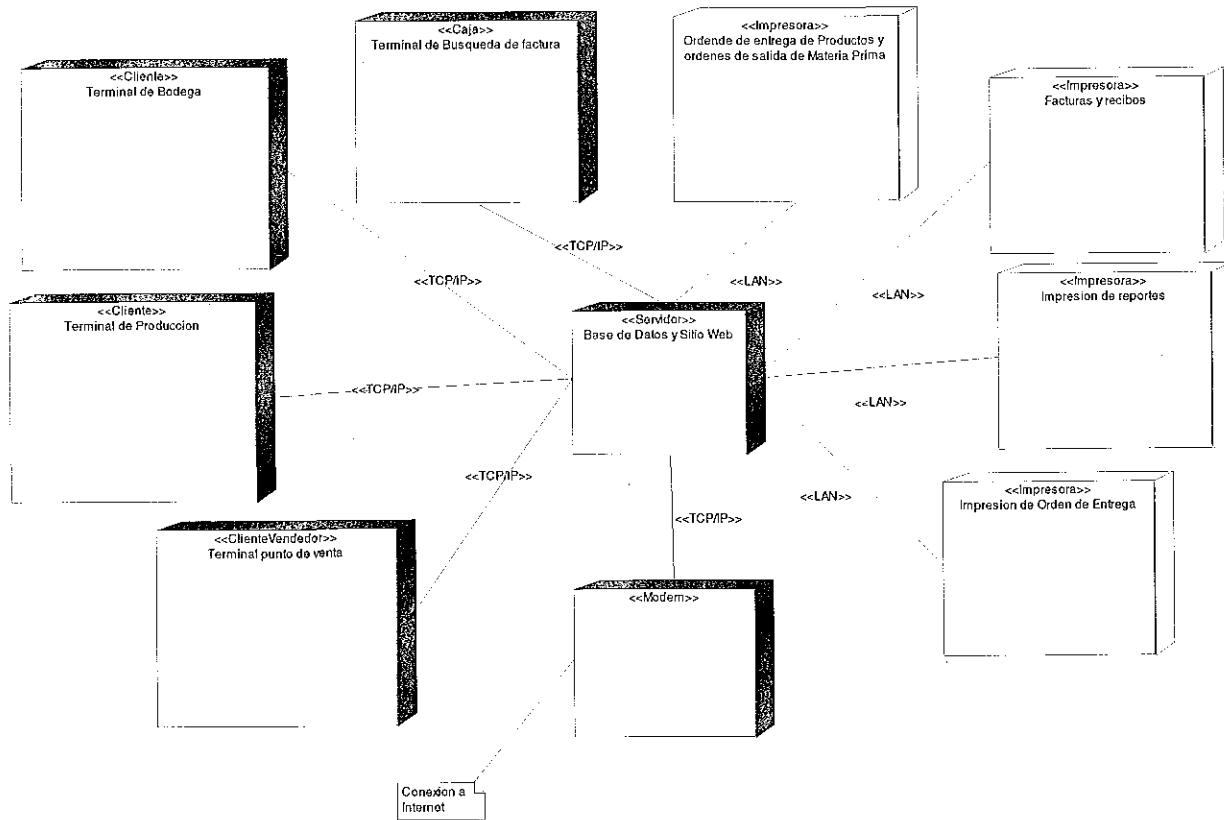
Agregar productos



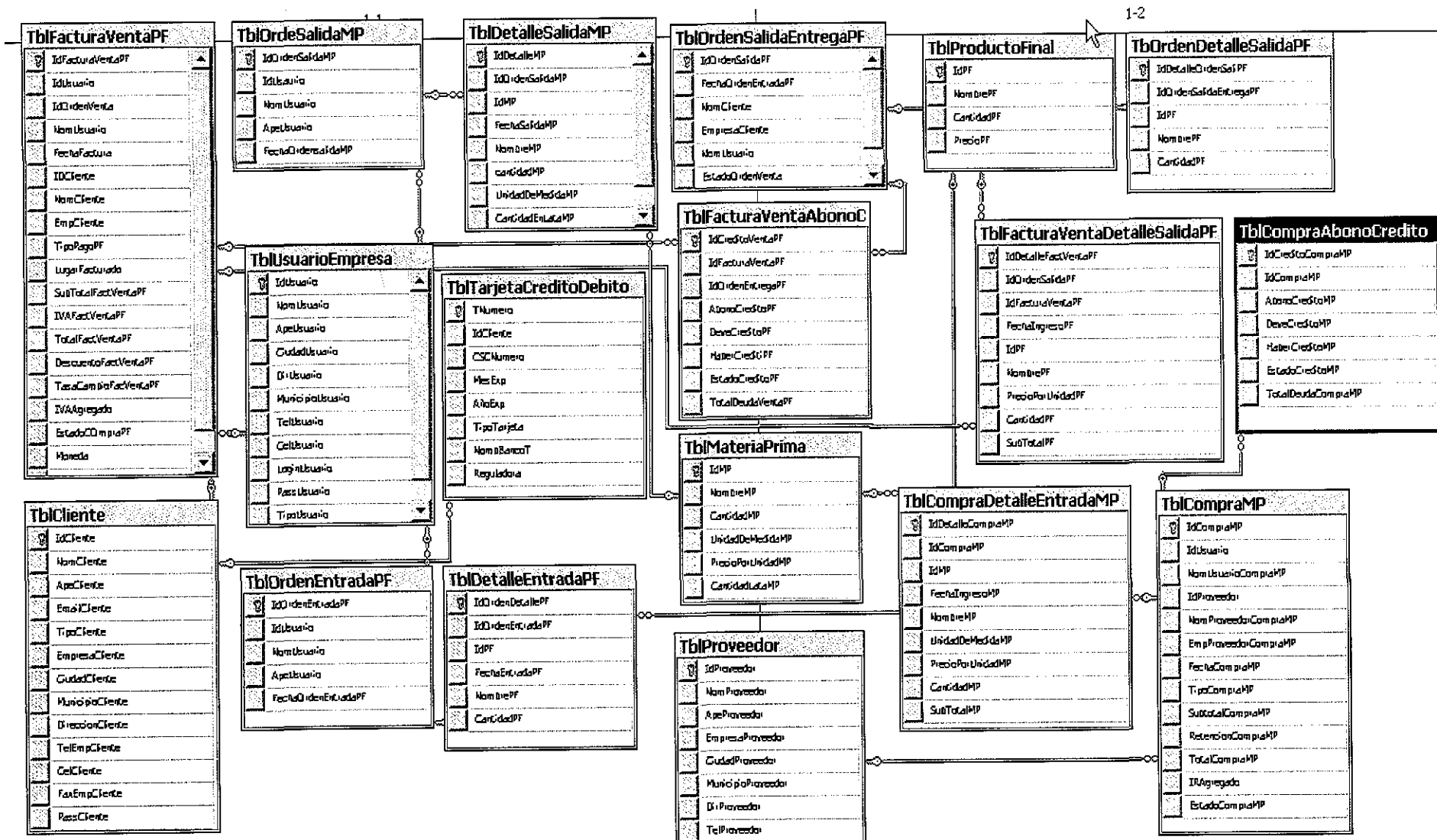
Hacer el pago de la Compra



2.6 Diagrama de despliegue / Distribución



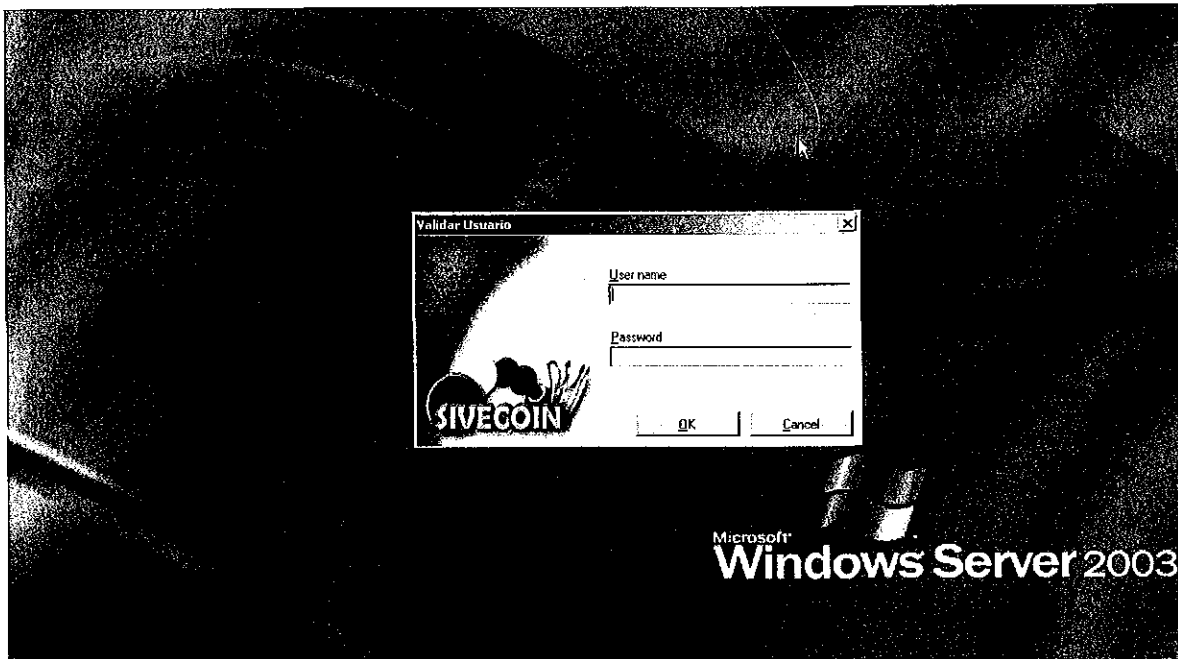
2.7 Tablas de la base de datos



VIII. Diseño físico y Despliegue

1. Pantalla de captación de datos y codificación

1.1 Aplicación Windows



Esta es la ventana de validación de Usuario, desde aquí se puede ingresar al sistema, siempre y cuando este en registro los datos del usuario, en caso contrario mandara un mensaje de "Nombre de Usuario y Contraseña incorrecta"

Codigo de Validacion de Usuario

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class LoginForm1

Protected Const SQL_CONNECTION_STRING As String =

—
"Server=MAMA\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Dim bool As Boolean

Private ConnectionString As String =

SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub Cancel_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Cancel.Click

Me.Close()

End Sub

Private Sub OK_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles OK.Click

validarUsuario()

End Sub

Private Sub validarUsuario()

```
Dim menu As New MDIParent
If validarUsuario(PasswordTextBox.Text) = 1 And
validarUsuario(UsernameTextBox.Text) = 1 Then
    menu.Show()
    Me.Visible = False

Else
    MsgBox("Nombre de Usuario o Contraseña Invalida,
Por favor intente de nuevo", MsgBoxStyle.Critical, "Validacion
Incorrecto")
    limpiarCajas()
End If
End Sub

Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function

Private Sub limpiarCajas()
    UsernameTextBox.Text = ""
    PasswordTextBox.Text = ""
End Sub

Private Function ValidarUsuario(ByVal strValue As String)
As Integer
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
```

```
Dim strSQL As String
```

```
Try
```

```
    strSQL = "Select LoginUsuario, PassUsuario FROM  
TblUsuarioEmpresa WHERE LoginUsuario=" &  
PrepareStr(UsernameTextBox.Text) & _  
" AND PassUsuario=" &  
PrepareStr>PasswordTextBox.Text)  
    cnSQL = New SqlConnection(ConnectionString)  
    cnSQL.Open()  
  
    cmSQL = New SqlCommand(strSQL, cnSQL)  
    drSQL = cmSQL.ExecuteReader()  
    If drSQL.Read() Then  
        UsernameTextBox.Text =  
drSQL.Item("LoginUsuario").ToString()  
        PasswordTextBox.Text() =  
drSQL.Item("PassUsuario").ToString()  
        Return 1  
    Else  
        Return 0  
    End If  
    ' Cerrar y limpiar objetos  
    drSQL.Close()  
    cnSQL.Close()  
    cmSQL.Dispose()  
    cnSQL.Dispose()
```

```
Catch e As SqlException
```

```
        MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL  
Error")
```

```
    Catch e As Exception
```

```
        MsgBox(e.Message,    MsgBoxStyle.Critical,    "General  
Error")
```

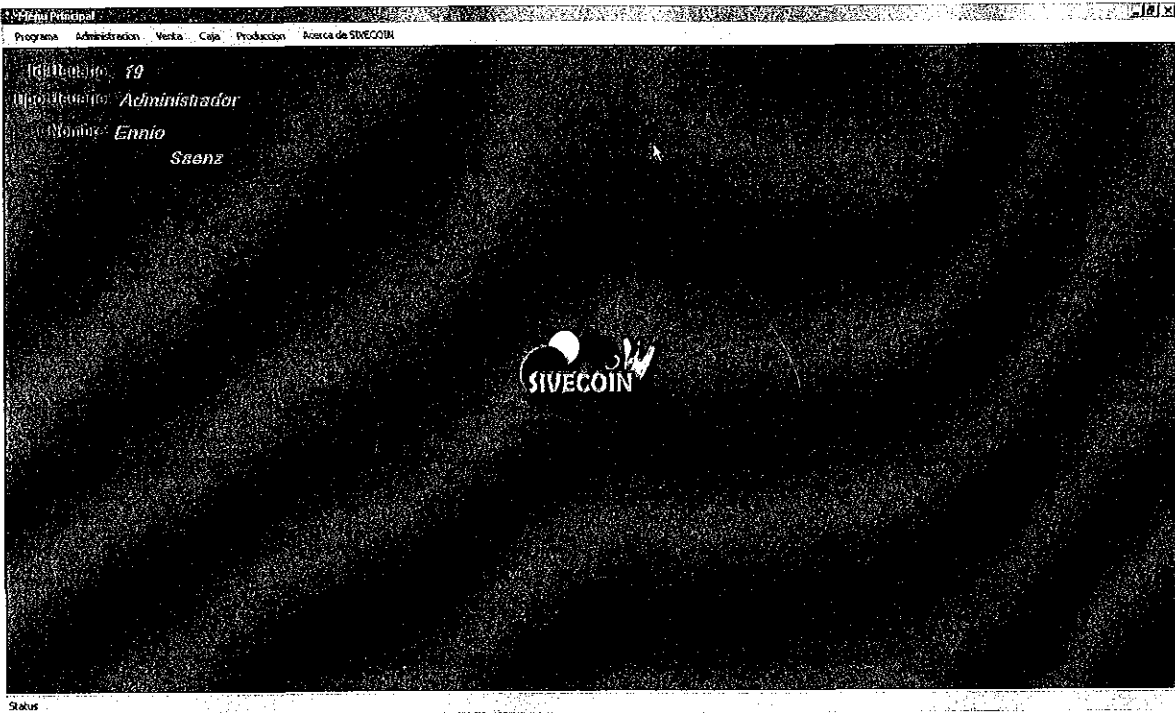
```
    End Try
```

```
End Function
```

```
Private Sub LoginForm1_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
End Sub
```

```
End Class
```



Esta es la ventana principal, en la barra de menú se visualizan los diferentes cargos que afectara el proyecto, cada cargo es un sub menú. Cada sub menú se activara dependiendo del cargo del usuario, y los demás permanecerán desactivada para restringir el acceso

Código de menú principal

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Imports System.Windows.Forms

Public Class MDIParent

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

-

"Server=MAMA\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Dim bool As Boolean

Private ConnectionString As String =

SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs)

End Sub

Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)

Dim OpenFileDialog As New OpenFileDialog

OpenFileDialog.InitialDirectory =

My.Computer.FileSystem.SpecialDirectories.MyDocuments

```
OpenFileDialog.Filter = "Archivos de texto  
(*.*)|*.txt|Todos los archivos (*.*)|*.*"  
If (OpenFileDialog.ShowDialog(Me) =  
System.Windows.Forms.DialogResult.OK) Then  
Dim FileName As String = OpenFileDialog.FileName  
' TODO: Agregar código aquí para abrir el archivo.  
End If  
End Sub
```

```
Private Sub SaveAsToolStripMenuItem_Click(ByVal sender  
As Object, ByVal e As EventArgs)  
Dim SaveFileDialog As New SaveFileDialog  
SaveFileDialog.InitialDirectory =  
My.Computer.FileSystem.SpecialDirectories.MyDocuments  
SaveFileDialog.Filter = "Archivos de texto  
(*.*)|*.txt|Todos los archivos (*.*)|*.*"  
If (SaveFileDialog.ShowDialog(Me) =  
System.Windows.Forms.DialogResult.OK) Then  
Dim FileName As String = SaveFileDialog.FileName  
' TODO: agregar código aquí para guardar el contenido  
actual del formulario en un archivo.  
End If  
End Sub
```

```
Private Sub ExitToolStripMenuItem_Click(ByVal sender As  
Object, ByVal e As EventArgs) Handles  
ExitToolStripMenuItem.Click  
Global.System.Windows.Forms.Application.Exit()  
End Sub
```

```
Private Sub CutToolStripMenuItem_Click(ByVal sender As  
Object, ByVal e As EventArgs)
```

```
    ' Utilice My.Computer.Clipboard para insertar el texto o  
    las imágenes seleccionadas en el Portapapeles
```

```
End Sub
```

```
Private Sub CopyToolStripMenuItem_Click(ByVal sender As  
Object, ByVal e As EventArgs)
```

```
    ' Utilice My.Computer.Clipboard para insertar el texto o  
    las imágenes seleccionadas en el Portapapeles
```

```
End Sub
```

```
Private Sub PasteToolStripMenuItem_Click(ByVal sender  
As Object, ByVal e As EventArgs)
```

```
    'Utilice My.Computer.Clipboard.GetText() o  
    My.Computer.Clipboard.GetData para recuperar la  
    información del Portapapeles.
```

```
End Sub
```

```
Private Sub CascadeToolStripMenuItem_Click(ByVal  
sender As Object, ByVal e As EventArgs)
```

```
    Me.LayoutMdi(MdiLayout.Cascade)
```

```
End Sub
```

```
Private Sub TileVerticleToolStripMenuItem_Click(ByVal  
sender As Object, ByVal e As EventArgs)
```

```
    Me.LayoutMdi(MdiLayout.TileVertical)
```

```
End Sub
```

```
Private Sub TileHorizontalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.LayoutMdi(MdiLayout.TileHorizontal)
End Sub
```

```
Private Sub ArrangeIconsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.LayoutMdi(MdiLayout.ArrangeIcons)
End Sub
```

```
Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    ' Cierre todos los formularios secundarios del primario.
    For Each ChildForm As Form In Me.MdiChildren
        ChildForm.Close()
    Next
End Sub
```

```
Private m_ChildFormNumber As Integer = 0
```

```
Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    AboutBox.Show()
End Sub
```

```
Private Sub PersonalesYEmpresasToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmVisualRegPersonasEmpresas.Show()
End Sub
```

```
Private Sub ProduccionToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmVisualRegProduccion.Show()
End Sub
```

```
Private Sub ComprasYVentasToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmVisualRegComprayVentas.Show()
End Sub
```

```
Private Sub IntroducirToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles menuAdministrador.Click
End Sub
```

```
Private Sub MDIParent_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    LlenarInfoUsuario()
    PermisosMenu()
End Sub
```

```
Private Sub MDIParent_ControlRemoved(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    Global.System.Windows.Forms.Application.Exit()
End Sub
```

```
Private Sub PermisosMenu()
    If Label1.Text = "Administrador" Then
```

```
        menuAdministrador.Enabled = True
        MenuCaja.Enabled = True
        MenuProduccion.Enabled = True
        MenuVenta.Enabled = True
    ElseIf Label1.Text = "Caja" Then
        frmIntroRegVentaPF.btnVisFact.Visible = False
        MenuCaja.Enabled = True
        frmIntroRegVentaPF.btnPrinter.Text =
"Visualizar/Imprimir Factura"
        frmIntroRegVentaPF.gbBotonesFactVentaPF.Visible =
False
    ElseIf Label1.Text = "Produccion" Then
        MenuProduccion.Enabled = True
    ElseIf Label1.Text = "Venta" Then
        frmIntroRegVentaPF.gbAnular.Visible = False
        frmIntroRegVentaPF.btnVisFact.Visible = False
        MenuVenta.Enabled = True
        frmIntroRegVentaPF.gbBusqueda.Visible = False
        frmIntroRegVentaPF.btnAgregarAbono.Visible = False
        frmIntroRegVentaPF.chkAnularFactura.Visible = False
        frmIntroRegVentaPF.gbBusqueda.Visible = False

    End If
End Sub
Private Sub LlenarInfoUsuario()
    Dim drSQL As SqlDataReader
    Dim cnSQL As SqlConnection
    Dim strSQL As String
    Dim cmSQL As SqlCommand
    Try
```

```
        strSQL = "SELECT * FROM TblUsuarioEmpresa
WHERE          LoginUsuario="          &
PrepareStr(LoginForm1.UsernameTextBox.Text) & _
        "          AND          PassUsuario="          &
PrepareStr(LoginForm1.PasswordTextBox.Text)
        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        drSQL = cmSQL.ExecuteReader()
        If drSQL.Read() Then
            Label1.Text = drSQL.Item("TipoUsuario").ToString()
            Label2.Text = drSQL.Item("NomUsuario").ToString()
            Label3.Text = drSQL.Item("ApeUsuario").ToString()
            Label6.Text = drSQL.Item("IdUsuario").ToString()
        End If
        drSQL.Close()
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

        Catch exp As SqlException
            MsgBox(exp.Message,  MsgBoxStyle.Critical,  "SQL
Error")

        Catch exp As Exception
            MsgBox(exp.Message,  MsgBoxStyle.Critical,  "General
Error")

        End Try
```

```
End Sub
Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function
```

```
Private Sub SalidaMateriaPrimaToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    frmOrdenSalidaMP.Show()
End Sub
```

```
Private Sub MenuStrip_ItemClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs)
Handles MenuStrip.ItemClicked
```

```
End Sub
```

```
Private Sub UsuarioToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
UsuarioToolStripMenuItem.Click
```

```
End Sub
```

```
Private Sub PersonalesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmVisualRegPersonasEmpresas.Show()
End Sub
```

```
Private Sub ProduccionToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmVisualRegProduccion.Show()
End Sub
```

```
Private Sub AcercaDeSIVECOINToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuAcercaDeSivecoin.Click
    AboutBox.Show()
End Sub
```

```
Private Sub OrdenDeSalidaDeMateriaPrimaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OrdenDeSalidaDeMateriaPrimaToolStripMenuItem.Click
    frmOrdenSalidaMP.Show()
End Sub
```

```
Private Sub IngresarNuevoProductoFinalToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IngresarNuevoProductoFinalToolStripMenuItem.Click
    frmOrdenEntradaPF.Show()
End Sub
```

```
Private Sub VisualizarRegistrosDeProduccionToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmVisualRegProduccion.Show()
End Sub
```

```
Private Sub RegistrarNuevoClienteToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmIntroCliente.Show()
End Sub
```

```
Private Sub RegistrarNuevoProveedorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmIntroProveedor.Show()
End Sub
```

```
Private Sub RegistrarCompraToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RegistrarCompraToolStripMenuItem.Click
    frmIntroRegCompraMP.Show()
End Sub
```

```
Private Sub ProductoToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    frmIntroPF.Show()
End Sub
```

```
Private Sub MaterialToolStripMenuItem_Click(ByVal sender  
As System.Object, ByVal e As System.EventArgs)  
    frmIntroMP.Show()  
End Sub
```

```
Private Sub IniciarSecionConOtroUsuarioToolStripMenuItem_Click(ByVal  
sender As System.Object, ByVal e As System.EventArgs)  
    LoginForm1.Visible = True  
    LoginForm1.UsernameTextBox.Text = ""  
    LoginForm1.PasswordTextBox.Text = ""  
End Sub
```

```
Private Sub RegistrarVentaToolStripMenuItem_Click(ByVal  
sender As System.Object, ByVal e As System.EventArgs)  
Handles RegistrarVentaToolStripMenuItem.Click  
    frmIntroRegVentaPF.Show()  
End Sub
```

```
Private Sub MenuProduccion_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MenuProduccion.Click  
  
End Sub
```

```
Private Sub LoginToolStripMenuItem_Click(ByVal sender  
As System.Object, ByVal e As System.EventArgs)  
  
End Sub
```

```
Private Sub VerReportesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    ReportParam.Show()
End Sub
```

```
Private Sub HacerBusquedaDeFacturaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles HacerBusquedaDeFacturaToolStripMenuItem.Click
    frmIntroRegVentaPF.Show()
End Sub
```

```
Private Sub ReportesDeFacturasDeVentaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ReportesDeFacturasDeVentaToolStripMenuItem.Click
    ReporteFacturasVentaPF.Show()
End Sub
```

```
Private Sub ActualizarPreciosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ActualizarPreciosToolStripMenuItem.Click
End Sub
```

```
Private Sub MateriaPrimaToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MateriaPrimaToolStripMenuItem2.Click
    frmIntroMP.Show()
End Sub
```

End Sub

```
Private Sub ProductoFinalToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ProductoFinalToolStripMenuItem1.Click  
    frmIntroPF.Show()  
End Sub
```

End Sub

```
Private Sub UsuariosToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UsuariosToolStripMenuItem1.Click
```

End Sub

```
Private Sub ProductosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ProductosToolStripMenuItem.Click
```

End Sub

```
Private Sub MateriaPrimaToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MateriaPrimaToolStripMenuItem1.Click
```

End Sub

```
Private Sub UsuarioToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UsuarioToolStripMenuItem1.Click  
    frmIntroRegUsuarios.Show()  
End Sub
```

End Sub

```
Private Sub ProveedorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ProveedorToolStripMenuItem.Click  
    frmIntroProveedor.Show()  
End Sub
```

```
Private Sub ClienteToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ClienteToolStripMenuItem.Click  
    frmIntroCliente.Show()  
End Sub  
End Class
```

Desde este formulario se registro de las ventas y búsqueda de factura por orden de entrega. De este formulario existen varias vistas según el cargo del usuario

Registro de Venta

Información de Venta

Datos de la Orden

N Factura:

Facturas: 1, 2, 3

Tipo de Venta:

Agregar Abono

Datos del Usuario

Id Usuario: 19

Nombre del Usuario: Ernio

No Anulado

Datos del Cliente

ID:

Nombre:

Empresa:

Nuevo Cliente

Fecha de Venta: 21/11/2007 13: Lugar Facturado: LOCAL

Descuento: 0 % IVA: 15 %

Hacer Búsqueda de Factura

Nº Orden:

Busqueda

Anular Factura

Anular OK

Nuevo Guardar Refresca Detalle

Detalles de Venta

Sub Total: 0

Descuento: 0

IVA: 0

Gran Total: 0

Visualizar/Imprimir Factura Visualizar/Imprimir Orden Cerrar

Vista del Administrador, aquí se ingresan los datos de venta de la factura y datos de la orden de entrega

The screenshot shows a software window titled "Registro de Venta". It is divided into several sections:

- Información de Venta:**
 - Datos de la Orden:** Includes fields for "N Factura:" (empty), "Facturas:" (a list with items 1, 2, and 3), and "Tipo de Venta:" (a dropdown menu).
 - Datos del Usuario:** Shows "Id Usuario: 19" and "Nombre del Usuario: Errio".
 - Datos del Cliente:** Includes "ID:" (empty), "Nombre:" (a dropdown menu), "Empresa:" (empty), and a "Nuevo Cliente" button.
 - Fecha de Venta:** 21/11/2007 13: Lugar Facturado: LOCAL
 - Descuento:** 0 %
 - IVA:** 15 %
- Botones:** "Nuevo", "Guardar", "Refresca", and "Detalle".
- Detalles de Venta:** A large, mostly blank area with a "Detalles de Venta" label.
- Logo:** "SIVECOIN" logo featuring a hand holding a coin.
- Totales:** "Sub Total: 0", "Descuento: 0", "IVA: 0", and "Gran Total: 0".
- Botones de Pie:** "Visualizar/Imprimir Factura", "Visualizar/Imprimir Orden", and "Cerrar".

Vista del Vendedor, desde aquí se introducen datos de la factura y orden de entrega

Registro de Venta

Información de Venta

Datos de la Orden

N Factura:

Facturas:
1
2
3

Datos del Usuario

Id Usuario: 19

Nombre del Usuario: Ennio

Datos del Cliente

ID:

Nombre:

Empresa:

Tipo de Venta:

Fecha de Venta: 21/11/2007 13 Lugar Facturado: LOCAL


Agregar Abono Descuento: 0 % IVA: 15 %

Hacer Búsqueda de Factura

Nº Orden: Anular Factura

Detalles de Venta

Detalles de Venta

	Sub Total: 0
	Descuento: 0
	IVA: 0
	Gran Total: 0

Vista de la caja, desde esta vista se realizan las búsquedas de las facturas por número de orden de entrega.

Codig de Factura de Venta

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmIntroRegVentaPF

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—
"Server=MAMA\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub btnNuevoCliente_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnNuevoCliente.Click

frmIntroCliente.Show()

Me.Enabled = False

End Sub

```
Private Sub NuevaFactura()  
    Dim drSQL As SqlDataReader  
    Dim cnSQL As SqlConnection  
    Dim strSQL As String  
    Dim cmSQL As SqlCommand  
    Try  
        strSQL = "SELECT IdFacturaVentaPF FROM  
TblFacturaVentaPF WHERE IdFacturaVentaPF = (SELECT  
Max(IdFacturaVentaPF)FROM TblFacturaVentaPF )"  
        cnSQL = New SqlConnection(ConnectionString)  
        cnSQL.Open()  
  
        cmSQL = New SqlCommand(strSQL, cnSQL)  
        drSQL = cmSQL.ExecuteReader()  
        If drSQL.Read() Then  
            txtIdFactura.Text =  
drSQL.Item("IdFacturaVentaPF").ToString()  
        End If  
        drSQL.Close()  
        cnSQL.Close()  
        cmSQL.Dispose()  
        cnSQL.Dispose()  
  
    Catch exp As SqlException  
        MsgBox(exp.Message, MsgBoxStyle.Critical, "SQL  
Error")  
  
    Catch exp As Exception
```

```
MsgBox(exp.Message, MsgBoxStyle.Critical, "General  
Error")
```

```
End Try
```

```
If txtIdFactura.Text = CStr("") Then
```

```
txtIdFactura.Text = CStr(CInt(0))
```

```
Else
```

```
txtIdFactura.Text = CStr(CInt(txtIdFactura.Text) +  
CInt(1))
```

```
End If
```

```
End Sub
```

```
Private Sub LimpiarCajas()
```

```
txtDescuento.Text = CStr(CDbl(0))
```

```
lblLugarfact.Text = "LOCAL"
```

```
lblFecha.Text = CStr(Now())
```

```
txtIVAPF.Text = CStr(CDbl(15))
```

```
lblSubTotalPF.Text = CStr(CDbl(0))
```

```
lblDescuentoPF.Text = CStr(CDbl(0))
```

```
lblIVAPF.Text = CStr(CDbl(0))
```

```
lblGranTotalPF.Text = CStr(CDbl(0))
```

```
txtIdFactura.Text = ""
```

```
cbTipoVenta.Text = ""
```

```
cbCliente.Text = ""
```

```
txtEstadoVenta.Text = "No Anulado"
```

```
End Sub
```

```
Private Sub frmIntroRegCompraMP_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    LimpiarCajas()  
    LlenarInfoUsuario()  
    LlenarListaFormuilario()  
    LlenarComoboClienteNom()
```

```
End Sub
```

```
Private Sub LlenarInfoUsuario()
```

```
    Dim drSQL As SqlDataReader  
    Dim cnSQL As SqlConnection  
    Dim strSQL As String  
    Dim cmSQL As SqlCommand  
    Try  
        strSQL = "SELECT * FROM TblUsuarioEmpresa  
WHERE          LoginUsuario="          &  
PrepareStr(LoginForm1.UsernameTextBox.Text) & _  
        "          AND          PassUsuario="          &  
PrepareStr(LoginForm1.PasswordTextBox.Text)  
        cnSQL = New SqlConnection(ConnectionString)  
        cnSQL.Open()  
  
        cmSQL = New SqlCommand(strSQL, cnSQL)  
        drSQL = cmSQL.ExecuteReader()  
        If drSQL.Read() Then  
            lblNombreUsuario.Text          =  
drSQL.Item("NomUsuario").ToString()  
            lblIdUsuario.Text          =  
drSQL.Item("IdUsuario").ToString()
```

```
End If
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch exp As SqlException
    MsgBox(exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch exp As Exception
    MsgBox(exp.Message, MsgBoxStyle.Critical, "General
Error")

End Try

End Sub
Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function
Private Sub LlenarListaFormulario()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
```

Try

' Construir la sentencia Select statement para
encuestar los nombres de productos

```
strSQL = "SELECT IdFacturaVentaPF FROM  
TbIFacturaVentaPF"
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
drSQL = cmSQL.ExecuteReader()
```

```
IstVenta.Items.Clear()
```

' Lazo de control de lectura sobre la clase datareader.

```
Do While drSQL.Read()
```

```
    IstVenta.Items.Add(drSQL("IdFacturaVentaPF"))
```

```
Loop
```

```
If IstVenta.Items.Count > 0 Then
```

```
    'IstOrders.SetSelected(0, True)
```

```
End If
```

' Close and Clean up objects

```
drSQL.Close()
```

```
cnSQL.Close()
```

```
cmSQL.Dispose()
```

```
cnSQL.Dispose()
```

```
        Catch e As SqlException
            MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")
```

```
        Catch e As Exception
            MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
```

```
    End Try
```

```
End Sub
```

```
Private Sub Checked()
```

```
    If chkAnularFactura.Checked = True Then
```

```
        txtEstadoVenta.Text = CStr("Anulado")
```

```
    ElseIf chkAnularFactura.Checked = False Then
```

```
        txtEstadoVenta.Text = CStr("No Anulado")
```

```
    End If
```

```
End Sub
```

```
Private Sub AgregarOrdenEVentaPF()
```

```
    Dim cnSQL As SqlConnection
```

```
    Dim cmSQL As SqlCommand
```

```
    Dim strSQL As String
```

```
    Dim Fecha As String = Format(lblFecha.Text, "Short
Date")
```

```
    Try
```

```
        ' Build Sentencia Insert para insertar un nuevo
producto dentro de la tabla products
```

```
        strSQL = "INSERT INTO TblOrdenSalidaEntregaPF
VALUES (" & _
```

```
"" & Fecha & "" & "," & _  
PrepareStr(txtNombre.Text) & "," & _  
PrepareStr(txtEmpresa.Text) & "," & _  
PrepareStr(lblNombreUsuario.Text) & "," & _  
"Activa")"  
  
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()  
  
cmSQL = New SqlCommand(strSQL, cnSQL)  
cmSQL.ExecuteNonQuery()  
  
' Cerrar y Limpiar los objetos utilizados  
cnSQL.Close()  
cmSQL.Dispose()  
cnSQL.Dispose()  
  
' Refresh Product List  
  
Catch Exp As SqlException  
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL  
Error")  
  
Catch Exp As Exception  
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General  
Error")  
End Try  
  
End Sub  
  
Private Sub AgregarVenta()
```

```
If Not IsValidForm() Then
```

```
Exit Sub
```

```
End If
```

```
Dim cnSQL As SqlConnection
```

```
Dim cmSQL As SqlCommand
```

```
Dim strSQL As String
```

```
Dim NFactua As String = txtIdFactura.Text
```

```
Dim Fecha As String = Format(lblFecha.Text, "Short  
Date")
```

```
Try
```

```
    ' Build Sentencia Insert para insertar un nuevo  
    producto dentro de la tabla products
```

```
    strSQL = "INSERT INTO TbFacturaVentaPF (  
    IdFacturaVentaPF" & _
```

```
    ",IdUsuario,IdOrdenVenta,NomUsuario,FechaFactura,IDClient  
    e,TipoPagoPF,LugarFacturado," & _
```

```
    "SubTotalFactVentaPF,IVAFactVentaPF,TotalFactVentaPF,De  
    scuentoFactVentaPF,IVA Agregado," & _
```

```
    "EstadoCOMpraPF) VALUES (" & _
```

```
    CInt(NFactua) & "," & _
```

```
    CInt(lblIdUsuario.Text) & "," & _
```

```
    CInt(txtIdOrdenEntregaPF.Text) & "," & _
```

```
    PrepareStr(lblNombreUsuario.Text) & "," & _
```

```
    "" & Fecha & "" & "," & _
```

```
    CType(cbCliente.Items(cbCliente.SelectedIndex),
```

```
    ListItem).ID & "," & _
```

```
    PrepareStr(cbTipoVenta.Text) & "," & _
```

```
PrepareStr(lblLugarfact.Text) & "," & _  
Cdbl(lblSubTotalPF.Text) & "," & _  
Cdbl(lblIVAPF.Text) & "," & _  
Cdbl(lblGranTotalPF.Text) & "," & _  
Cdbl(txtDescuento.Text) & "," & _  
Cdbl(txtIVAPF.Text) & "," & _  
PrepareStr(txtEstadoVenta.Text) & ")"  
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()  
  
cmSQL = New SqlCommand(strSQL, cnSQL)  
cmSQL.ExecuteNonQuery()  
  
' Cerrar y Limpiar los objetos utilizados  
cnSQL.Close()  
cmSQL.Dispose()  
cnSQL.Dispose()  
  
' Refresh Product List  
LlenarListaFormuilario()  
FindProductByName(lstVenta, NFactua)  
  
Catch Exp As SqlException  
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL  
Error")  
  
Catch Exp As Exception  
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General  
Error")  
End Try  
End Sub
```

```
Private Sub ActualizarClienteYEmp()  
    Dim cnSQL As SqlConnection  
    Dim cmSQL As SqlCommand  
    Dim strSQL As String  
    Dim intRowsAffected As Integer  
    Dim Fecha As String = Format(lblFecha.Text, "Short  
Date")
```

```
Try
```

```
    ' Construir una sentencia update para la tabla product  
    con los datos
```

```
    ' en la forma.
```

```
    strSQL = "UPDATE TbFacturaVentaPF SET  
NomCliente = " & PrepareStr(txtNombre.Text) & _  
    ",EmpCliente = " & PrepareStr(txtEmpresa.Text) & _  
    " WHERE IdFacturaVentaPF = " &  
    Cint(txtIdFactura.Text)
```

```
    cnSQL = New SqlConnection(ConnectionString)  
    cnSQL.Open()
```

```
    cmSQL = New SqlCommand(strSQL, cnSQL)  
    intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
    If intRowsAffected <> 1 Then
```

```
        MsgBox("Update Failed.", MsgBoxStyle.Critical,  
"Update")
```

```
    End If
```

```
' Cerrar y Limpiar los objetos
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

Catch e As Exception
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
End Try
End Sub
Private Sub LeerCliente()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String

    Try
        strSQL = "SELECT    NomCliente,EmpresaCliente
FROM    TblCliente    Where    IdCliente    =    " &
CType(cbCliente.Items(cbCliente.SelectedIndex), ListItem).ID

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        drSQL = cmSQL.ExecuteReader()
```

```
IstVenta.Items.Clear()

' Lazo de control de lectura sobre la clase datareader.
Do While drSQL.Read()
    txtNombre.Text = (drSQL("NomCliente").ToString())
    txtEmpresa.Text =
(drSQL("EmpresaCliente").ToString())
Loop

If IstVenta.Items.Count > 0 Then
    'IstOrders.SetSelected(0, True)
End If

' Close and Clean up objects
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

Catch e As Exception
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
End Try
End Sub

Private Sub ActualizarVentaPF()
    Dim cnSQL As SqlConnection
```

```
Dim cmSQL As SqlCommand
Dim strSQL As String
Dim intRowsAffected As Integer
Dim Fecha As String = Format(lblFecha.Text, "Short
Date")
```

```
Try
```

```
    ' Construir una sentencia update para la tabla product
con los datos
```

```
    ' en la forma.
```

```
    strSQL = "UPDATE TblFacturaVentaPF SET " & _
```

```
    "IdUsuario = " & CInt(lblIdUsuario.Text) & _
```

```
    ",NomUsuario = " & PrepareStr(lblNombreUsuario.Text) & _
```

```
    PrepareStr(lblNombreUsuario.Text) & _
```

```
    ",FechaFactura = " & "" & Fecha & "" & _
```

```
    ",IDCliente = " & CType(cbCliente.Items(cbCliente.SelectedIndex), ListItem).ID
```

```
    & _
```

```
    ",TipoPagoPF = " & PrepareStr(cbTipoVenta.Text) & _
```

```
    ",LugarFacturado = " & PrepareStr(lblLugarfact.Text) & _
```

```
    ",SubTotalFactVentaPF = " & CDbl(lblSubTotalPF.Text) & _
```

```
    CDbl(lblSubTotalPF.Text) & _
```

```
    ",IVAFactVentaPF = " & CDbl(lblIVAPF.Text) & _
```

```
    ",TotalFactVentaPF = " & CDbl(lblGranTotalPF.Text) & _
```

```
    ",IVA Agregado = " & CDbl(txtIVAPF.Text) & _
```

```
    ",EstadoCOMpraPF = " & PrepareStr(txtEstadoVenta.Text) & _
```

```
    PrepareStr(txtEstadoVenta.Text) & _
```

```
WHERE IdFacturaVentaPF = " &  
CInt(txtIdFactura.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
If intRowsAffected <> 1 Then  
    MsgBox("Update Failed.", MsgBoxStyle.Critical,  
"Update")  
End If
```

```
' Cerrar y Limpiar los objetos  
cnSQL.Close()  
cmSQL.Dispose()  
cnSQL.Dispose()
```

```
Catch e As SqlException  
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")
```

```
Catch e As Exception  
    MsgBox(e.Message, MsgBoxStyle.Critical, "General  
Error")
```

```
End Try  
End Sub
```

```
Private Sub IstCompras_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IstVenta.SelectedIndexChanged
    chkAnularFactura.Checked = False
    LlenarFormulario()
    LeerCliente()
    LlenarDGridDeacuerdoLista()
    AddChecked()

    Mode = "Update"
End Sub
Private Sub AddChecked()
    If txtEstadoVenta.Text = "Anulado " Then
        chkAnularFactura.Checked = True
    ElseIf txtEstadoVenta.Text = "No Anulado" Then
        chkAnularFactura.Checked = False
    End If
End Sub
Private Sub LlenarDGridDeacuerdoLista()
    Dim cnSQL As New SqlConnection
    Dim adSQL As SqlDataAdapter
    Dim DSet As New DataSet()
    Dim strSQL As String

    Try

        strSQL = "SELECT FechaIngresoPF AS
Fecha,NombrePF AS Producto," & _
        "PrecioPorUnidadPF AS 'Precio Por
Unidad',CantidadPF AS Cantidad,SubTotalPF AS 'Sub Total'
```

```
from      TblFacturaVentaDetalleSalidaPF      WHERE
IdFacturaVentaPF =" & CInt(txtIdFactura.Text)
' cargar el grid de personal
adSQL     =     New     SqlDataAdapter(strSQL,
ConnectionString)
adSQL.Fill(DSet, "TblFacturaVentaDetalleSalidaPF")
grdDetalleVenta.DataSource                      =
DSet.Tables("TblFacturaVentaDetalleSalidaPF")

Catch Exp As SqlException
MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try

cnSQL.Close()
cnSQL.Dispose()
End Sub
Private Sub LlenarFormulario()
Dim cnSQL As SqlConnection
Dim cmSQL As SqlCommand
Dim drSQL As SqlDataReader
Dim strSQL As String
Dim strID As String

Try
```

' Construir la sentencia Select para encuestar la información

```
strSQL = "SELECT * FROM TblFacturaVentaPF " & _  
        "WHERE IdFacturaVentaPF = " &  
CType(lstVenta.SelectedItem, String)
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
drSQL = cmSQL.ExecuteReader()
```

```
If drSQL.Read() Then
```

```
' Llenar la forma con los datos
```

```
txtIdFactura.Text =
```

```
drSQL.Item("IdFacturaVentaPF").ToString()
```

```
lblIdUsuario.Text =
```

```
drSQL.Item("IdUsuario").ToString()
```

```
txtIdOrdenEntregaPF.Text =
```

```
drSQL.Item("IdOrdenVenta").ToString()
```

```
lblNombreUsuario.Text() =
```

```
drSQL.Item("NomUsuario").ToString()
```

```
lblFecha.Text() =
```

```
drSQL.Item("FechaFactura").ToString()
```

```
lblIdCliente.Text =
```

```
drSQL.Item("IDCliente").ToString()
```

```
cbTipoVenta.Text() =
```

```
drSQL.Item("TipoPagoPF").ToString()
```

```
lblLugarfact.Text() =
```

```
drSQL.Item("LugarFacturado").ToString()
```

```
        lblSubTotalPF.Text() =  
        drSQL.Item("SubTotalFactVentaPF").ToString()  
        lblIVAPF.Text() =  
        drSQL.Item("IVAFactVentaPF").ToString()  
        lblGranTotalPF.Text() =  
        drSQL.Item("TotalFactVentaPF").ToString()  
        txtDescuento.Text() =  
        drSQL.Item("DescuentoFactVentaPF").ToString()  
        txtIVAPF.Text() =  
        drSQL.Item("IVA Agregado").ToString()  
        txtEstadoVenta.Text() =  
        drSQL.Item("EstadoCOMpraPF").ToString()  
        strID = drSQL.Item("IDCliente").ToString()  
        FindItemByID(cbCliente, strID)
```

End If

' Cerrar y limpiar objetos

```
drSQL.Close()
```

```
cnSQL.Close()
```

```
cmSQL.Dispose()
```

```
cnSQL.Dispose()
```

Catch e As SqlException

```
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")
```

Catch e As Exception

```
        MsgBox(e.Message, MsgBoxStyle.Critical, "General  
Error")
```

```
End Try
End Sub

Private Sub FindItemByID(ByVal cbCombo As ComboBox,
ByVal strID As String)
    'Esta Sub es utilizada para encontrar un elemento de la
    combobox y
    ' situar el selected index de la combo box a ese elemento
    Dim bFound As Boolean
    Dim ListItem As ListItem
    Dim itemCount As Integer

    ListItem = New ListItem()

    itemCount = 0
    While Not bFound Or itemCount <=
cbCombo.Items.Count - 1
        ListItem = CType(cbCombo.Items(itemCount),
ListItem)

        If ListItem.ID = CInt(strID) Then
            cbCombo.SelectedIndex = itemCount
            bFound = True
        End If
        itemCount += 1
    End While

    If Not bFound Then
        cbCombo.SelectedIndex = -1
    End If
```

End Sub

```
Private Sub FindProductByName(ByVal lbList As ListBox,  
ByVal strValue As String)
```

```
' Esta sub es utilizada para encontrar un Product en la  
List Box product
```

```
' y seleccionar un selected index de la list box a ese  
elemento.
```

```
Dim iposition As Integer
```

```
iposition = lstVenta.FindString(strValue)
```

```
' Lazo de control a la lista hasta encontrar el elemento
```

```
If iposition = ListBox.NoMatches Then
```

```
    lstVenta.SelectedIndex = 0
```

```
Else
```

```
    lstVenta.SelectedIndex = iposition
```

```
End If
```

End Sub

```
Private Function IsValidForm() As Boolean
```

```
' Validación de los campos
```

```
If cbCliente.SelectedIndex = -1 Or _
```

```
    cbTipoVenta.Text = "" Or _
```

```
    Not IsNumeric(txtIdFactura.Text) Or _
```

```
    Not IsDate(lbIFecha.Text) Then
```

```
    MsgBox("Favor introduzca un valor válido para cada  
campo", _
```

```
        MsgBoxStyle.Exclamation, Me.Text)
```

```
        Return False
    Else
        Return True
    End If
End Function
Private Sub LlenarComoboClienteNom()
    ' Este procedimiento llena la list box en la forma
    ' con la lista de Categories desde la base de datos
    ' Northwind.

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem

    ' Mostrar un mensaje de estado de conexión diciendo que
    se intenta conectar.
    Dim frmStatusMessage As New frmStatus()
    If Not HasConnected Then
        frmStatusMessage.Show("Conectando a la Base de
        Datos del Servidor")
    End If

    ' Intento de conexión a SQL server
    Dim IsConnecting As Boolean = True
    While IsConnecting
        Try

            strSQL = "SELECT IdCliente, NomCliente FROM
            TblCliente"
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

cbCliente.Items.Clear()

' Lazo de control para llenar la combo box con los
nombres de categoría      ' names to the combo box.
Do While drSQL.Read()
    objListItem = New
ListItem(drSQL.Item("NomCliente").ToString(), _
        CInt(drSQL.Item("IdCliente")))
    cbCliente.Items.Add(objListItem)
Loop

IsConnecting = False
HasConnected = True

' Cerrar y Limpiar los objetos
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
' No fué posible conectar con SQL Server
frmStatusMessage.Close()
```

```
        MsgBox("Error al efectuar la conexión con SQL
Server Express" & _
        " Debe tener la base de datos DBSBAM
instalada.", MsgBoxStyle.Critical, _
        "SQL Server Express not found")
        ' Salida del Programa sin realizar conexión.
    End
Catch e As Exception
    MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
End Try
frmStatusMessage.Close()
End While
End Sub
```

```
Private Sub btnClose_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnClose.Click
    Me.Close()
End Sub
```

```
Private Sub btnAgregarAbono_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAgregarAbono.Click
    MostrarFrmAbono()
    If txtTipoVenta.Text = "Credito" Then

        frmAbonosCreditVentaPF.Show()
    Else
```

```
        MsgBox("Esta Factura no es de Credito, Por favor  
verifique bien", MsgBoxStyle.Exclamation, "Operacion  
Incorrecta")
```

```
    End If
```

```
End Sub
```

```
Private Sub MostrarFrmAbono()
```

```
    Dim cnSQL As SqlConnection
```

```
    Dim cmSQL As SqlCommand
```

```
    Dim drSQL As SqlDataReader
```

```
    Dim strSQL As String
```

```
Try
```

```
    ' Construir la sentencia Select para encuestar la  
información
```

```
    strSQL = "SELECT * FROM TblFacturaVentaPF " & _  
            "WHERE IdFacturaVentaPF = " &  
CInt(txtIdFactura.Text)
```

```
    cnSQL = New SqlConnection(ConnectionString)
```

```
    cnSQL.Open()
```

```
    cmSQL = New SqlCommand(strSQL, cnSQL)
```

```
    drSQL = cmSQL.ExecuteReader()
```

```
    If drSQL.Read() Then
```

```
        txtTipoVenta.Text() =  
drSQL.Item("TipoPagoPF").ToString()
```

```
End If

drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch exp As SqlException
    MsgBox(exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch exp As Exception
    MsgBox(exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub

Private Sub cbCliente_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs)
Handles cbCliente.SelectedIndexChanged
    LeerCliente()
End Sub

Private Sub btnPrinter_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnPrinter.Click
    If btnPrinter.Text = "Visualizar/Imprimir Factura" Then
        FrmFacturaVenta.Show()
    Else
```

```
        frmOrdenEVenta.show()
    End If
End Sub

Private Sub btnBusqueda_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnBusqueda.Click

    chkAnularFactura.Checked = False
    ValidarBusqueda()
    AddChecked()

End Sub

Private Sub ValidarBusqueda()

    If Busqueda(txtBusquedaFAct.Text) = 1 Then

        Dim cnSQL As SqlConnection
        Dim cmSQL As SqlCommand
        Dim drSQL As SqlDataReader
        Dim strSQL As String
        Dim strID As String

        Try

            ' Construir la sentencia Select para encuestar la
información
            strSQL = "SELECT * FROM TbFacturaVentaPF " &
            "WHERE IdOrdenVenta = " &
            CInt(txtBusquedaFAct.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

If drSQL.Read() Then
    ' Llenar la forma con los datos
    txtIdFactura.Text =
drSQL.Item("IdFacturaVentaPF").ToString()
    lblIdUsuario.Text =
drSQL.Item("IdUsuario").ToString()
    lblNombreUsuario.Text =
drSQL.Item("NomUsuario").ToString()
    lblFecha.Text =
drSQL.Item("FechaFactura").ToString()
    lblIdCliente.Text =
drSQL.Item("IDCliente").ToString()
    cbTipoVenta.Text =
drSQL.Item("TipoPagoPF").ToString()
    lblLugarfact.Text =
drSQL.Item("LugarFacturado").ToString()
    lblSubTotalPF.Text =
drSQL.Item("SubTotalFactVentaPF").ToString()
    lblIVAPF.Text =
drSQL.Item("IVAFactVentaPF").ToString()
    lblGranTotalPF.Text =
drSQL.Item("TotalFactVentaPF").ToString()
    txtDescuento.Text =
drSQL.Item("DescuentoFactVentaPF").ToString()
```

```
        txtIVAPF.Text() =  
drSQL.Item("IVA Agregado").ToString()  
        txtEstadoVenta.Text() =  
drSQL.Item("EstadoCompraPF").ToString()  
        strID = drSQL.Item("IDCliente").ToString()  
        FindItemByID(cbCliente, strID)  
  
    End If  
    LlenarDGridDeacuerdoLista()  
    ' Cerrar y limpiar objetos  
    drSQL.Close()  
    cnSQL.Close()  
    cmSQL.Dispose()  
    cnSQL.Dispose()  
  
    Catch e As SqlException  
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")  
  
    Catch e As Exception  
        MsgBox(e.Message, MsgBoxStyle.Critical, "General  
Error")  
    End Try  
  
    Else  
        MsgBox("Este Numeo de Orden No existe, Por favor  
Verifique bien", MsgBoxStyle.Critical, "Busqueda Invalida")  
        LimpiarCajas()  
    End If  
End Sub
```

```
Private Function Busqueda(ByVal strValue As String) As Integer
```

```
    Dim cnSQL As SqlConnection  
    Dim cmSQL As SqlCommand  
    Dim drSQL As SqlDataReader  
    Dim strSQL As String
```

```
Try
```

```
    strSQL = "SELECT * FROM TblFacturaVentaPF  
WHERE IdOrdenVenta=" & CInt(txtBusquedaFact.Text)  
    cnSQL = New SqlConnection(ConnectionString)  
    cnSQL.Open()
```

```
    cmSQL = New SqlCommand(strSQL, cnSQL)  
    drSQL = cmSQL.ExecuteReader()
```

```
    If drSQL.Read() Then
```

```
        txtBusquedaFact.Text =  
drSQL.Item("IdOrdenVenta").ToString()
```

```
        Return 1
```

```
    Else
```

```
        Return 0
```

```
    End If
```

```
    ' Cerrar y limpiar objetos
```

```
    drSQL.Close()
```

```
    cnSQL.Close()
```

```
    cmSQL.Dispose()
```

```
    cnSQL.Dispose()
```

```
    Catch e As SqlException  
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")
```

```
    Catch e As Exception  
        MsgBox(e.Message, MsgBoxStyle.Critical, "General  
Error")
```

```
End Try
```

```
End Function
```

```
Private Sub btnAdd_Click_2(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnAdd.Click
```

```
    If Mode = "Add" Then
```

```
        AgregarOrdenEVentaPF()
```

```
        llenarIdOrdenEPF()
```

```
        AgregarVenta()
```

```
        LeerCliente()
```

```
        Mode = "Update"
```

```
        ActualizarClienteYEmp()
```

```
        btnAdd.Enabled = False
```

```
        btnDetalle.Enabled = True
```

```
        btnRefresh.Enabled = False
```

```
    Else
```

```
        Checked()
```

```
        ActualizarVentaPF()
```

```
        LeerCliente()
```

```
        ActualizarClienteYEmp()
```

```
End If
End Sub
Private Sub llenarIdOrdenEPF()
    Dim drSQL As SqlDataReader
    Dim cnSQL As SqlConnection
    Dim strSQL As String
    Dim cmSQL As SqlCommand
    Try
        strSQL = "SELECT IdOrdenSalidaPF FROM
TbIOrdenSalidaEntregaPF WHERE IdOrdenSalidaPF =
(SELECT
                                Max(IdOrdenSalidaPF)FROM
TbIOrdenSalidaEntregaPF )"
        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        drSQL = cmSQL.ExecuteReader()
        If drSQL.Read() Then
            txtIdOrdenEntregaPF.Text =
drSQL.Item("IdOrdenSalidaPF").ToString()
        End If
        drSQL.Close()
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

    Catch exp As SqlException
        MsgBox(exp.Message, MsgBoxStyle.Critical, "SQL
Error")
    End Try
End Sub
```

```
    Catch exp As Exception  
        MsgBox(exp.Message, MsgBoxStyle.Critical, "General  
Error")
```

```
    End Try
```

```
End Sub
```

```
Private Sub btnRefresh_Click_1(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnRefresh.Click
```

```
    btnAgregarAbono.Enabled = True
```

```
    btnAdd.Enabled = True
```

```
    btnNew.Enabled = True
```

```
    lblFecha.Text = CStr(Now())
```

```
    LlenarInfoUsuario()
```

```
    LlenarListaFormuilario()
```

```
    LlenarComoboClienteNom()
```

```
End Sub
```

```
Private Sub btnDetalle_Click_1(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnDetalle.Click
```

```
    frmDetallesVentaPF.show
```

```
    Me.Enabled = False
```

```
    btnDetalle.Enabled = False
```

```
    btnRefresh.Enabled = True
```

```
End Sub
```

```
Private Sub btnNew_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles btnNew.Click
```

```
Mode = "Add"  
LlenarInfoUsuario()  
LimpiarCajas()  
NuevaFactura()  
btnAgregarAbono.Enabled = False  
btnAdd.Enabled = True  
btnNew.Enabled = False  
End Sub
```

```
Private Sub btnVisFact_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnVisFact.Click  
FrmFacturaVenta.Show()  
End Sub
```

```
Private Sub btnAnularFactu_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnAnularFactu.Click
```

```
AddAnuladoCHK()
```

```
AnularFact()
```

```
End Sub
```

```
Private Sub AddAnuladoCHK()
```

```
If chkAnularFactura.Checked = True Then
```

```
txtEstadoVenta.Text = "Anulado "
```

```
Elseif chkAnularFactura.Checked = False Then
```

```
txtEstadoVenta.Text = "No Anulado"
```

```
End If
```

```
End Sub
```

```
Private Sub AnularFact()
```

```
Dim cnSQL As SqlConnection
```

```
Dim cmSQL As SqlCommand
Dim strSQL As String
Dim intRowsAffected As Integer
Dim Fecha As String = Format(lblFecha.Text, "Short
Date")
```

```
Try
```

```
    ' Construir una sentencia update para la tabla product
con los datos
```

```
    ' en la forma.
```

```
    strSQL = "UPDATE TbIFacturaVentaPF SET " & _
```

```
    "EstadoCOMpraPF          =          "          &
```

```
PrepareStr(txtEstadoVenta.Text) & _
```

```
    " WHERE IdFacturaVentaPF = "          &
```

```
CInt(txtIdFactura.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)
```

```
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)
```

```
intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
If intRowsAffected <> 1 Then
```

```
    MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")
```

```
End If
```

```
' Cerrar y Limpiar los objetos
```

```
cnSQL.Close()
```

```
        cmSQL.Dispose()
        cnSQL.Dispose()

    Catch exp As SqlException
        MsgBox(exp.Message, MsgBoxStyle.Critical, "SQL
Error")

    Catch exp As Exception
        MsgBox(exp.Message, MsgBoxStyle.Critical, "General
Error")
    End Try
End Sub

End Class
```

The screenshot shows a software window titled "Introducir Proveedor" with a standard Windows-style title bar. The window contains the following elements:

- Input fields for "Nombre:", "Apellido:", "Empresa:", "Ciudad:", and "Municipio:".
- A larger input field for "Dirección:".
- Input fields for "Teléfono:", "Celular:", and "FAX:".
- A logo for "SIVECOIN" featuring a hand holding a coin.
- Two buttons at the bottom right: "Guardar" and "Cerrar".

Esta es la ventana para registrar a los proveedores

Codigo de registro de Proveedor

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmIntroProveedor

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

–

"Server=MAMA\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=5"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub btnDelete_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)

 Me.Close()

End Sub

Private Sub btnUpdate_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnGuardar.Click

 Mode = "Add"

 AgregarProveedor()

```
MsgBox("Su Registro ha sido agregado  
satisfactoriamente")
```

```
btnGuardar.Enabled = False
```

```
End Sub
```

```
Private Sub AgregarProveedor()
```

```
    If Not IsValidForm() Then
```

```
        Exit Sub
```

```
    End If
```

```
    Dim cnSQL As SqlConnection
```

```
    Dim cmSQL As SqlCommand
```

```
    Dim strSQL As String
```

```
    Try
```

```
        ' Build Sentencia Insert para insertar un nuevo  
        producto dentro de la tabla de Usuario
```

```
        strSQL = "INSERT TblProveedor VALUES (" & _  
            PrepareStr(txtNombreP.Text) & "," & _  
            PrepareStr(txtApellidoP.Text) & "," & _  
            PrepareStr(txtEmpresaP.Text) & "," & _  
            PrepareStr(txtCiudadP.Text) & "," & _  
            PrepareStr(txtMunicipioP.Text) & "," & _  
            PrepareStr(txtDireccionP.Text) & "," & _  
            PrepareStr(txtTelefonoP.Text) & "," & _  
            PrepareStr(txtCelularP.Text) & "," & _  
            PrepareStr(txtFaxP.Text) & ")"
```

```
        cnSQL = New SqlConnection(ConnectionString)
```

```
        cnSQL.Open()
```

```
        cmSQL = New SqlCommand(strSQL, cnSQL)
```

```
cmSQL.ExecuteNonQuery()

' Cerrar y Limpiar los objetos utilizados
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch Exp As SqlException
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub

Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function

Private Function IsValidForm() As Boolean
    ' Validación de los campos
    If txtNombreP.Text = "" Or _
        txtApellidoP.Text = "" Or _
        txtCiudadP.Text = "" Or _
```

```
txtMunicipioP.Text = "" Or _
txtDireccionP.Text = "" Or _
txtTelefonoP.Text = "" Or _
txtCelularP.Text = "" Or _
txtFaxP.Text = "" Then

MsgBox("Favor introduzca un valor válido para cada
campo", _
MsgBoxStyle.Exclamation, Me.Text)
Return False
Else
Return True
End If
End Function

Private Sub btnCerrar_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnCerrar.Click
Me.Close()
frmIntroRegCompraMP.Enabled = True
End Sub

Private Sub frmIntroProveedor_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
 MyBase.Load

End Sub
End Class
```

The screenshot shows a Windows application window titled "Introducir y editar Usuario". The window contains a form for user management. The form has the following fields:

- Id Usuario:** Labeled "Label12".
- Nombre:** Text input field.
- Apellido:** Text input field.
- Ciudad:** Text input field.
- Direccion:** Text input field.
- Municipio:** Text input field.
- Telefono:** Text input field.
- Celular:** Text input field.
- Login:** Text input field.
- Contraseña:** Text input field.
- Tipo Usuario:** Dropdown menu.

In the center of the window, there is a logo for "SIVECOIN" featuring a basket of produce. To the right of the logo is a table titled "Lista de Usuarios" with a single column header "lstUsuario".

At the bottom of the window, there are five buttons: "Nuevo", "Guardar", "Refrescar", "Borrar", and "Cerrar".

Desde esta ventana se Agregan, Modifican y borran registros de Usuarios

Codigo de registro de Usuario

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmIntroRegUsuarios

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—
"Server=(Local)\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub Button3_Click(ByVal sender As System.Object, :
ByVal e As System.EventArgs) Handles btnClose.Click

Me.Close()

End Sub

Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAdd.Click

ClearForm()

Mode = "Add"

```
        btnAdd.Enabled = False
        btnDelete.Enabled = False
    End Sub
    Private Sub ClearForm()

        ' Limpiar la forma de entrada de datos.
        txtIdUsuario.Text = ""
        txtNombreUsuario.Text = ""
        txtApellidoUsuario.Text = ""
        txtCiudadUsuario.Text = ""
        txtDireccionUsuario.Text = ""
        txtMunicipioUsuario.Text = ""
        txtTelefonoUsuario.Text = ""
        txtCelularUsuario.Text = ""
        txtLoginUsuario.Text = ""
        txtPassUsuario.Text = ""

    End Sub
    Private Sub frmIntroRegUsuarios_Load(ByVal sender As
System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        ' Rellenar todas las listas y combo utilizados en la forma
        PopulateProductList()
    End Sub

    Private Sub btnDelete_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
        DeleteProduct()
    End Sub
```

```
End Sub
Private Sub DeleteProduct()
    ' Esta sub es utilizada para eliminar el registro de
    productos
    ' cuando el usuario haga clicks en el boton Delete

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer

    Try
        ' Contruir la sentenecia Delete
        strSQL = "DELETE FROM TblUsuarioEmpresa " & _
            "WHERE IdUsuario = " & CInt(txtIdUsuario.Text)

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        intRowsAffected = cmSQL.ExecuteNonQuery()

        If intRowsAffected <> 1 Then
            MsgBox("Borrado Fallido. Id Usuario " &
                txtIdUsuario.Text & _
                " not found.", MsgBoxStyle.Critical, "Delete")
        End If

        ' Cierre y Limpieza de los objetos
        cnSQL.Close()
        cmSQL.Dispose()
    End Try
End Sub
```

```
cnSQL.Dispose()

PopulateProductList()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

Catch e As Exception
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
End Try
End Sub
Private Sub PopulateForm()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem

Try

    ' Tomar la llave primaria de la Listbox
    objListItem = CType(lstUsuario.SelectedItem, ListItem)

    ' Construir la sentencia Select para encuestar la
información
    strSQL = "SELECT * FROM TblUsuarioEmpresa " & _
        "WHERE IdUsuario = " & objListItem.ID
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

If drSQL.Read() Then
    ' Llenar la forma con los datos
    txtIdUsuario.Text =
drSQL.Item("IdUsuario").ToString()
    txtNombreUsuario.Text() =
drSQL.Item("NomUsuario").ToString()
    txtApellidoUsuario.Text() =
drSQL.Item("ApeUsuario").ToString()
    txtCiudadUsuario.Text() =
drSQL.Item("CiudadUsuario").ToString()
    txtDireccionUsuario.Text() =
drSQL.Item("DirUsuario").ToString()
    txtMunicipioUsuario.Text() =
drSQL.Item("MunicipioUsuario").ToString()
    txtTelefonoUsuario.Text() =
drSQL.Item("TelUsuario").ToString()
    txtCelularUsuario.Text() =
drSQL.Item("CelUsuario").ToString()
    txtLoginUsuario.Text() =
drSQL.Item("LoginUsuario").ToString()
    txtPassUsuario.Text() =
drSQL.Item("PassUsuario").ToString()
    cbTipoUsuario.Text() =
drSQL.Item("TipoUsuario").ToString()
```

```
End If

' Cerrar y limpiar objetos
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

Catch e As Exception
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
End Try
End Sub
Private Sub PopulateProductList()
' Este procedimiento llena la list box en la
' form con una lista de productos disponibles desde la
' database Northwind.

Dim cnSQL As SqlConnection
Dim cmSQL As SqlCommand
Dim drSQL As SqlDataReader
Dim strSQL As String
Dim objListItem As ListItem

Try
    ' Construir la sentencia Select statement para
encuestar los nombres de productos
```

```
strSQL = "SELECT NomUsuario, IdUsuario FROM  
TbUsuarioEmpresa"
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
drSQL = cmSQL.ExecuteReader()
```

```
IstUsuario.Items.Clear()
```

```
' Lazo de control de lectura sobre la clase datareader.  
Do While drSQL.Read()
```

```
    objListItem = New  
    ListItem(drSQL.Item("NomUsuario").ToString(), _  
             CInt(drSQL.Item("IdUsuario")))
```

```
    IstUsuario.Items.Add(objListItem)
```

```
Loop
```

```
If IstUsuario.Items.Count > 0 Then
```

```
    IstUsuario.SetSelected(0, True)
```

```
End If
```

```
' Close and Clean up objects
```

```
drSQL.Close()
```

```
cnSQL.Close()
```

```
cmSQL.Dispose()
```

```
cnSQL.Dispose()
```

```
Catch e As SqlException
```

```
MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")
```

```
Catch e As Exception
```

```
MsgBox(e.Message, MsgBoxStyle.Critical, "General  
Error")
```

```
End Try
```

```
End Sub
```

```
Private Sub FindItemByID(ByVal cbCombo As ComboBox,  
ByVal strID As String)
```

'Esta Sub es utilizada para encontrar un elemento de la
combobox y

' situar el selected index de la combo box a ese elemento

```
Dim bFound As Boolean
```

```
Dim ListItem As ListItem
```

```
Dim itemCount As Integer
```

```
ListItem = New ListItem()
```

```
itemCount = 0
```

```
While Not bFound Or itemCount <=  
cbCombo.Items.Count - 1
```

```
ListItem = CType(cbCombo.Items(itemCount),  
ListItem)
```

```
If ListItem.ID = CInt(strID) Then
```

```
cbCombo.SelectedIndex = itemCount
```

```
bFound = True
```

```
End If
```

```
itemCount += 1
```

```
End While
```

```
If Not bFound Then  
    cbCombo.SelectedIndex = -1  
End If
```

```
End Sub
```

```
Private Sub FindProductByName(ByVal lbList As ListBox,  
ByVal strValue As String)
```

```
    ' Esta sub es utilizada para encontrar un Product en la  
List Box product
```

```
    ' y seleccionar un selected index de la list box a ese  
elemento.
```

```
Dim blsFound As Boolean
```

```
Dim ListItem As ListItem
```

```
Dim itemCount As Integer
```

```
ListItem = New ListItem()
```

```
itemCount = 0
```

```
' Lazo de control a la lista hasta encontrar el elemento
```

```
While Not blsFound And itemCount <= lbList.Items.Count
```

```
- 1
```

```
    ListItem = CType(lbList.Items(itemCount), ListItem)
```

```
    If ListItem.Value = strValue Then
```

```
        lbList.SelectedIndex = itemCount
```

```
        blsFound = True
```

```
    End If
```

```
    itemCount += 1
```

```
End While
```

```
If Not blsFound Then
    lbList.SelectedIndex = 0
End If
End Sub
Private Function IsValidForm() As Boolean
    ' Validación de los campos
    If txtNombreUsuario.Text = "" Or _
        txtApellidoUsuario.Text = "" Or _
        txtCiudadUsuario.Text = "" Or _
        txtDireccionUsuario.Text = "" Or _
        txtMunicipioUsuario.Text = "" Or _
        txtTelefonoUsuario.Text = "" Or _
        txtCelularUsuario.Text = "" Or _
        txtLoginUsuario.Text = "" Or _
        txtPassUsuario.Text = "" Or _
        cbTipoUsuario.Text = "" Then

        MsgBox("Favor introduzca un valor válido para cada
campo", _
            MsgBoxStyle.Exclamation, Me.Text)
        Return False
    Else
        Return True
    End If
End Function

Private Sub btnUpdate_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnUpdate.Click
    ' Comprobar Mode para determinar la acción
```

```
If Mode = "Add" Then
    AddUsuario()
    btnAdd.Enabled = True
Else
    UpdateUsuario()
End If
```

```
End Sub
```

```
Private Sub AddUsuario()
```

```
    If Not IsValidForm() Then
        Exit Sub
    End If
```

```
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim NomUsuario As String = txtNombreUsuario.Text
```

```
    Try
```

```
        ' Build Sentencia Insert para insertar un nuevo
        producto dentro de la tabla de Usuario
```

```
        strSQL = "INSERT TblUsuarioEmpresa VALUES (" & _
            PrepareStr(NomUsuario) & "," & _
            PrepareStr(txtApellidoUsuario.Text) & "," & _
            PrepareStr(txtCiudadUsuario.Text) & "," & _
            PrepareStr(txtDireccionUsuario.Text) & "," & _
            PrepareStr(txtMunicipioUsuario.Text) & "," & _
            PrepareStr(txtTelefonoUsuario.Text) & "," & _
            PrepareStr(txtCelularUsuario.Text) & "," & _
            PrepareStr(txtLoginUsuario.Text) & "," & _
```

```
PrepareStr(txtPassUsuario.Text) & "," & _  
  
PrepareStr(CStr(cbTipoUsuario.SelectedItem).ToString()) & ")"  
  
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()  
  
cmSQL = New SqlCommand(strSQL, cnSQL)  
cmSQL.ExecuteNonQuery()  
  
' Cerrar y Limpiar los objetos utilizados  
cnSQL.Close()  
cmSQL.Dispose()  
cnSQL.Dispose()  
  
' Refresh Product List  
PopulateProductList()  
FindProductByName(lstUsuario, NomUsuario)  
  
Catch Exp As SqlException  
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL  
Error")  
  
Catch Exp As Exception  
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General  
Error")  
End Try  
End Sub  
Private Sub frmDataEntry_Load(ByVal sender As  
System.Object, _
```

```
ByVal e As System.EventArgs) _  
Handles MyBase.Load  
' Rellenar todas las listas y combo utilizados en la forma  
  
PopulateProductList()  
  
End Sub  
  
Private Sub IstUsuario_SelectedIndexChanged(ByVal  
sender As System.Object, _  
ByVal e As System.EventArgs)  
-  
Handles  
IstUsuario.SelectedIndexChanged  
PopulateForm()  
btnDelete.Enabled = True  
btnAdd.Enabled = True  
Mode = "Update"  
  
End Sub  
Private Sub UpdateUsuario()  
' Esta sub es utilizada para actualizar un registro  
existente  
Dim cnSQL As SqlConnection  
Dim cmSQL As SqlCommand  
Dim strSQL As String  
Dim intRowsAffected As Integer  
  
' Validar el formulario.  
If Not IsValidForm() Then  
Exit Sub
```

End If

Try

' Construir una sentencia update para la tabla product
con los datos

' en la forma.

```
strSQL = "UPDATE TbUsuarioEmpresa SET" & _  
        "      NomUsuario      =      "      &  
PrepareStr(txtNombreUsuario.Text) & _  
        "      ,ApeUsuario      =      "      &  
PrepareStr(txtApellidoUsuario.Text) & _  
        "      ,CiudadUsuario      =      "      &  
PrepareStr(txtCiudadUsuario.Text) & _  
        "      ,DirUsuario      =      "      &  
PrepareStr(txtDireccionUsuario.Text) & _  
        "      ,MunicipioUsuario      =      "      &  
PrepareStr(txtMunicipioUsuario.Text) & _  
        "      ,TelUsuario      =      "      &  
PrepareStr(txtTelefonoUsuario.Text) & _  
        "      ,CelUsuario      =      "      &  
PrepareStr(txtCelularUsuario.Text) & _  
        "      ,LoginUsuario      =      "      &  
PrepareStr(txtLoginUsuario.Text) & _  
        "      ,PassUsuario      =      "      &  
PrepareStr(txtPassUsuario.Text) & _  
        "      ,TipoUsuario      =      "      &  
PrepareStr(CStr(cbTipoUsuario.SelectedItem).ToString()) & _  
        "WHERE IdUsuario = " & CInt(txtIdUsuario.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)
intRowsAffected = cmSQL.ExecuteNonQuery()

If intRowsAffected <> 1 Then
    MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")
End If

' Cerrar y Limpiar los objetos
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch e As Exception
    MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub

Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function
```

End If

End Function

```
Private Sub btnRefresh_Click_1(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnRefresh.Click
```

```
PopulateProductList()
```

```
End Sub
```

```
End Class
```

The image shows a web form titled "Introducir Cliente" (Enter Customer). The form contains several input fields for customer information:

- Nombre: [Text Input]
- Apellido: [Text Input]
- Correo Electrónico: [Text Input]
- Lugar de Registro: [Dropdown Menu, currently showing "LOCAL"]
- Empresa: [Text Input]
- Ciudad: [Text Input]
- Municipio: [Text Input]
- Dirección: [Text Input]
- Teléfono: [Text Input]
- Celular: [Text Input]
- FAX: [Text Input]

At the bottom right of the form, there is a logo for "SIVECOIN" and two buttons: "Guardar" (Save) and "Cerrar" (Close).

Desde este formulario se pueden Ingresar registros de clientes Nuevos.

Codigo para el registro del cliente

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmIntroCliente

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—

"Server=MAMA\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =

SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click

Me.Close()

frmIntroRegVentaPF.Enabled = True

End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

Mode = "Add"

AgregarCliente()

MsgBox("Su Registro ha sido agregado
satisfactoriamente")

```
        Button1.Enabled = False
    End Sub
Private Sub AgregarCliente()
    If Not IsValidForm() Then
        Exit Sub
    End If

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String

    Try
        ' Build Sentencia Insert para insertar un nuevo
        producto dentro de la tabla de Usuario
        strSQL = "INSERT TbICliente VALUES (" & _
            PrepareStr(txtNombre.Text) & "," & _
            PrepareStr(txtApellido.Text) & "," & _
            PrepareStr(txtEmail.Text) & "," & _
            PrepareStr(txtTipoCliente.Text) & "," & _
            PrepareStr(txtEmpresa.Text) & "," & _
            PrepareStr(txtCiudad.Text) & "," & _
            PrepareStr(txtMunicipio.Text) & "," & _
            PrepareStr(txtDireccion.Text) & "," & _
            PrepareStr(txtTelefono.Text) & "," & _
            PrepareStr(txtCelular.Text) & "," & _
            PrepareStr(txtFax.Text) & "," & _
            PrepareStr(txtContraseña.Text) & ")"

        cnSQL = New SqlConnection(ConnectionString)
```

```
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
cmSQL.ExecuteNonQuery()

' Cerrar y Limpiar los objetos utilizados
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch Exp As SqlException
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub

Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function

Private Function IsValidForm() As Boolean
    ' Validación de los campos
```

```
If txtNombre.Text = "" Or _  
txtApellido.Text = "" Or _  
txtCiudad.Text = "" Or _  
txtMunicipio.Text = "" Or _  
txtDireccion.Text = "" Or _  
txtTelefono.Text = "" Or _  
txtCelular.Text = "" Or _  
txtFax.Text = "" Then
```

```
MsgBox("Favor introduzca un valor válido para cada  
campo", _
```

```
MsgBoxStyle.Exclamation, Me.Text)
```

```
Return False
```

```
Else
```

```
Return True
```

```
End If
```

```
End Function
```

```
Private Sub frmIntroCliente_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
txtTipoCliente.Text = "LOCAL"
```

```
End Sub
```

```
End Class
```

Realizar Abonos de Venta

N Factura: Label6 Estado de la Factura: Label6 Orden Prden de Entrega: Label6

Estado Actual del Credito

Deve: Label6 Total del Credito: Label7

Abono de Venta al Credito			
Abonos	Deuda	Total Abonos	Estado de la Orden

Abono:

Agregar Abono OK

Esta venta es para poder hacer abonos a las facturas de creditos

Codigo de Abono de Venta

Option Strict On

End Sub

```
Private Sub frmAbonosCreditCompraMP_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    frmIntroRegVentaPF.Enabled = False
```

```
    leerTblVentaAbonoCredito()
```

```
    LeerDGAbono()
```

End Sub

```
Private Sub LeerDGAbono()
```

```
    Dim cnSQL As New SqlConnection
```

```
    Dim adSQL As SqlDataAdapter
```

```
    Dim DSet As New DataSet()
```

```
    Dim strSQL As String
```

```
    Try
```

```
        strSQL = "SELECT AbonoCreditoPF AS Abonos,DeveCreditoPF AS Deuda,HaberCreditoPF AS 'Total Abonos',EstadoCreditoPF AS 'Estado de la Orden'" & _
```

```
        " FROM TblFacturaVentaAbonoCreditoPF where IdFacturaVentaPF = " &
```

```
        CInt(frmIntroRegVentaPF.txtIdFactura.Text)
```

```
        ' cargar el grid de personal
```

```
        adSQL = New SqlDataAdapter(strSQL, ConnectionString)
```

```
        adSQL.Fill(DSet, "TblFacturaVentaAbonoCreditoPF")
```

```
        grdAbonoVenta.DataSource =
```

```
        DSet.Tables("TblFacturaVentaAbonoCreditoPF")
```

```
    Catch Exp As SqlException
        MsgBox(Exp.Message,  MsgBoxStyle.Critical,  "SQL
Error")

    Catch Exp As Exception
        MsgBox(Exp.Message,  MsgBoxStyle.Critical,  "General
Error")
    End Try

    cnSQL.Close()
    cnSQL.Dispose()
End Sub
Private Sub leerTbIVentaAbonoCredito()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String

    Try
        strSQL = "SELECT
IdFacturaVentaPF,IdOrdenEntregaPF,DeveCreditoPF,Estado
CreditoPF,TotalDeudaVentaPF " & _
        " FROM  TblFacturaVentaAbonoCreditoPF  WHERE
IdFacturaVentaPF = " &
CInt(frmIntroRegVentaPF.txtIdFactura.Text) & _
        " AND  IdCreditoVentaPF = (Select
max(IdCreditoVentaPF)
from
TblFacturaVentaAbonoCreditoPF where IdFacturaVentaPF ="
& CInt(frmIntroRegVentaPF.txtIdFactura.Text) & ")"
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

' Lazo de control de lectura sobre la clase datareader.
If drSQL.Read() Then
    lblIdFactura.Text =
(drSQL("IdFacturaVentaPF").ToString())
    lblOrdenFact.text() =
(drSQL("IdOrdenEntregaPF").ToString())
    lblDeve.Text = (drSQL("DeveCreditoPF").ToString())
    lblEstadoCredito.Text =
(drSQL("EstadoCreditoPF").ToString())
    lblTotalCredito.Text =
(drSQL("TotalDeudaVentaPF").ToString())
End If
' Close and Clean up objects
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

Catch e As Exception
```

```
        MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
    End Try
End Sub
Private Function PrepareStr(ByVal strValue As String) As
String
    'Rellenar con NULOS las cajas de texto vacias.
    If strValue.Trim() = "" Then
        Return "NULL"
    Else
        Return "" & strValue.Trim() & ""
    End If
End Function
Private Sub AgregarAbono()
    If Not IsValidForm() Then
        Exit Sub
    End If

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim IdVenta As String = lblIdFactura.Text

    Try
        ' Build Sentencia Insert para insertar un nuevo
producto dentro de la tabla products
        strSQL = "INSERT TblFacturaVentaAbonoCreditoPF
VALUES (" & _
            CInt(IdVenta) & "," & _
            CInt(lblOrdenFact.Text) & "," & _
            CDbI(txtAbonoCreditoVentaPF.Text) & "," & _
```

```
        CDb1(IblDeve.Text)
    CDb1(txtAbonoCreditoVentaPF.Text) & "," & _
        CDb1(0) & "," & _
        PrepareStr(IblEstadoCredito.Text) & "," & _
        CDb1(IblTotalCredito.Text) & ")"
    cnSQL = New SqlConnection(ConnectionString)
    cnSQL.Open()

    cmSQL = New SqlCommand(strSQL, cnSQL)
    cmSQL.ExecuteNonQuery()

    ' Cerrar y Limpiar los objetos utilizados
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

Catch Exp As SqlException
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub

Private Function IsValidForm() As Boolean
    ' Validación de los campos
    If txtAbonoCreditoVentaPF.Text = "" Then
        MsgBox("Favor Introduzca el Abono", _
```

```
        MsgBoxStyle.Exclamation, Me.Text)
    Return False
Else
    Return True
End If
End Function

Private Sub btnAgregarAbono_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAgregarAbono.Click

    leerTbIVentaAbonoCredito()
    AgregarAbono()
    leerTbIVentaAbonoCredito()
    ActualizarHaberCreditoPF()
    leerTbIVentaAbonoCredito()
    LeerDGAbono()
End Sub

Private Sub ActualizarHaberCreditoPF()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer
    Try
        ' Construir una sentencia update para la tabla product
con los datos
        ' en la forma.
        strSQL = "UPDATE TblFacturaVentaAbonoCreditoPF
SET " & _
        "HaberCreditoPF = " & CDbl(lblTotalCredito.Text) -
CDbl(lblDeve.Text) & _
```

```
        WHERE    IdCreditoVentaPF    =    (Select  
max(IdCreditoVentaPF)                from  
TblFacturaVentaAbonoCreditoPF) AND IdFacturaVentaPF = "  
& CInt(frmIntroRegVentaPF.txtIdFactura.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
If intRowsAffected <> 1 Then
```

```
    MsgBox("Update    Failed.",    MsgBoxStyle.Critical,  
"Update")
```

```
End If
```

```
' Cerrar y Limpiar los objetos
```

```
cnSQL.Close()
```

```
cmSQL.Dispose()
```

```
cnSQL.Dispose()
```

```
Catch e As SqlException
```

```
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL  
Error")
```

```
Catch e As Exception
```

```
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General  
Error")
```

```
End Try
```

```
End Sub
```

```
End Class
```

Detalle de Venta

Id Factura: Label3 Id Producto: Id Material Fecha de Salida: Label7

Lista de Productos:

IstProducto

Nombre:


Precio por Unidad:

Cantidad en existencia:

Datos a Introducir

Cantidad a Ingresar:

CbTipoCompra



Sub Total: SubTotal

Descuento: Descuento

Total: TOTAL

Agregar Producto OK

Ventana de Detalles de venta, desde aquí se ingresan los detalles de las facturas de ventas

Codigo Detalle Venta

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmDetallesVentaPF

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

 "Server=MAMA\SQLEXPRESS;" & _

 "DataBase=DBSBAM;" & _

 "Integrated Security=SSPI;Connect Timeout=20"

 Private ConnectionString As String =

SQL_CONNECTION_STRING

 Private HasConnected As Boolean = False

 Private Mode As String = "Update"

 Dim SubTotal As Double

 Dim GranTotal As Double

 Dim IVA As Double

 Dim Descuento As Double

 Private Sub btnOK_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles btnOK.Click

 frmIntroRegVentaPF.Enabled = True

 Me.Close()

```
End Sub
Private Sub frmDetalleCompraMP_ControlRemoved(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    frmIntroRegVentaPF.Enabled = True
End Sub
```

```
Private Sub frmDetalleCompraMP_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    lblTipoComp.Text = CStr(frmIntroRegVentaPF.cbTipoVenta.Text)
    txtTipoComp.Text = CStr(frmIntroRegVentaPF.cbTipoVenta.Text)
    lblDescuento.Text = CStr(CDbl(0))
    lblSubTotal.Text = CStr(CDbl(0))
    lblTotal.Text = CStr(CDbl(0))
    lblIdVenta.Text = frmIntroRegVentaPF.txtIdFactura.Text
    lblFecha.Text = CStr(Now())
    LlenarListaFormulario()
End Sub
```

```
Private Sub LlenarListaFormulario()
```

```
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem
```

```
Try
    ' Construir la sentencia Select statement para
encuestar los nombres de productos
    strSQL = "SELECT NombrePF,IdPF FROM
TbIProductoFinal"
    cnSQL = New SqlConnection(ConnectionString)
    cnSQL.Open()

    cmSQL = New SqlCommand(strSQL, cnSQL)
    drSQL = cmSQL.ExecuteReader()

    IstProducto.Items.Clear()

    ' Lazo de control de lectura sobre la clase datareader.
    Do While drSQL.Read()
        objListItem = New
ListItem(drSQL.Item("NombrePF").ToString(), _
        CInt(drSQL.Item("IdPF")))
        IstProducto.Items.Add(objListItem)
    Loop

    If IstProducto.Items.Count > 0 Then
        IstProducto.SetSelected(0, True)
    End If

    ' Close and Clean up objects
    drSQL.Close()
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()
```

```
        Catch e As SqlException
            MsgBox(e.Message,   MsgBoxStyle.Critical,   "SQL
Error")
```

```
        Catch e As Exception
            MsgBox(e.Message,   MsgBoxStyle.Critical,   "General
Error")
```

```
    End Try
```

```
End Sub
```

```
Private Function PrepareStr(ByVal strValue As String) As
String
```

```
    'Rellenar con NULOS las cajas de texto vacias.
```

```
    If strValue.Trim() = "" Then
```

```
        Return "NULL"
```

```
    Else
```

```
        Return "" & strValue.Trim() & ""
```

```
    End If
```

```
End Function
```

```
Private Sub IstMaterial_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles IstProducto.SelectedIndexChanged
```

```
    LlenarFormulario()
```

```
    Mode = "Update"
```

```
End Sub
```

```
Private Sub FindProductByName(ByVal lbList As ListBox,
ByVal strValue As String)
```

```
    ' Esta sub es utilizada para encontrar un Product en la
List Box product
```

```
    ' y seleccionar un selected index de la list box a ese
elemento.
```

```
Dim iposition As Integer
iposition = IstProducto.FindString(strValue)

' Lazo de control a la lista hasta encontrar el elemento

If iposition = ListBox.NoMatches Then
    IstProducto.SelectedIndex = 0
Else
    IstProducto.SelectedIndex = iposition
End If

End Sub

Private Sub LlenarFormulario()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem
    Try
        objListItem = CType(IstProducto.SelectedItem,
ListItem)

        ' Construir la sentencia Select para encuestar la
información
        strSQL = "SELECT * FROM TblProductoFinal " & _
            "WHERE IdPF = " & objListItem.ID

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

If drSQL.Read() Then
    ' Llenar la forma con los datos
    lblIdPF.Text = drSQL.Item("IdPF").ToString()
    txtNombrePF.Text() =
drSQL.Item("NombrePF").ToString()
    txtCantidadExistencia.Text =
drSQL.Item("CantidadPF").ToString()
    txtPrecio.Text = drSQL.Item("PrecioPF").ToString()

End If

' Cerrar y limpiar objetos
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch e As Exception
    MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")

End Try
End Sub
```

```
Private Sub btnAgregar_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAgregar.Click
    If Cdbl(txtCantidadExistencia.Text) > Cdbl(0) Then
        CalculoSubTotal()
        CalculoTotal()
        ActualizarCantidad()
        AgregarProducto()
        AgregarODetalledeOrden()
        ActualizarVenta()
        LlenarDGridDeacuerdoLista()
    Else
        MsgBox("No Hay Producto en existencia",
MsgBoxStyle.Critical, "Advertencia")
    End If
End Sub
Private Sub Descuentos()
    lblDescuento.Text =
CStr(Cdbl(frmIntroRegVentaPF.txtDescuento.Text) / 100 *
Cdbl(lblTotal.Text))
End Sub
Private Sub AgregarODetalledeOrden()
    If Not IsValidForm() Then
        Exit Sub
    End If

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
```

```
Try
    ' Build Sentencia Insert para insertar un nuevo
    producto dentro de la tabla products
    strSQL = "INSERT TbOrdenDetalleSalidaPF VALUES
    (" & _
    CInt(frmIntroRegVentaPF.txtIdOrdenEntregaPF.Text)
    & "," & _
    CInt(lbIdPF.Text) & "," & _
    PrepareStr(txtNombrePF.Text) & "," & _
    CInt(txtCantidadAIngresar.Text) & ")"

    cnSQL = New SqlConnection(ConnectionString)
    cnSQL.Open()

    cmSQL = New SqlCommand(strSQL, cnSQL)
    cmSQL.ExecuteNonQuery()

    ' Cerrar y Limpiar los objetos utilizados
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

    ' Refresh Product List

Catch Exp As SqlException
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
    Error")

Catch Exp As Exception
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
    Error")
```

End Try

End Sub

Private Sub ActualizarCantidad()

Dim cnSQL As SqlConnection

Dim cmSQL As SqlCommand

Dim strSQL As String

Dim intRowsAffected As Integer

Try

' Construir una sentencia update para la tabla product
con los datos

' en la forma.

strSQL = "UPDATE TblProductoFinal SET " & _

"CantidadPF = " & CInt(CDb1(txtCantidadExistencia.Text)

CInt(CDb1(txtCantidadAIngresar.Text)) & _

"WHERE IdPF = " & CInt(IblIdPF.Text)

cnSQL = New SqlConnection(ConnectionString)

cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)

intRowsAffected = cmSQL.ExecuteNonQuery()

If intRowsAffected <> 1 Then

MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")

```
End If

' Cerrar y Limpiar los objetos
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

Catch e As Exception
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
End Try

End Sub

Private Sub ActualizarVenta()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer
    SubTotal =
Cdbl(frmIntroRegVentaPF.lblSubTotalPF.Text)
    IVA = Cdbl(frmIntroRegVentaPF.lblIVAPF.Text)
    GranTotal =
Cdbl(frmIntroRegVentaPF.lblGranTotalPF.Text)
    Descuento = Cdbl(lblDescuento.Text)

Try
```

' Construir una sentencia update para la tabla product
con los datos

' en la forma.

```
strSQL = "UPDATE TbIFacturaVentaPF SET " & _  
"SubTotalFactVentaPF = " & SubTotal & _  
",IVAFactVentaPF = " & IVA & _  
",TotalFactVentaPF = " & GranTotal & _  
",DescuentoFactVentaPF = " & Descuento & _  
"WHERE IdFacturaVentaPF = " &  
CInt(frmIntroRegVentaPF.txtIdFactura.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
If intRowsAffected <> 1 Then  
    MsgBox("Update Failed.", MsgBoxStyle.Critical,  
"Update")  
End If
```

```
' Cerrar y Limpiar los objetos  
cnSQL.Close()  
cmSQL.Dispose()  
cnSQL.Dispose()
```

```
Catch e As SqlException  
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")
```

```
        Catch e As Exception
            MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
        End Try
    End Sub
Private Sub LlenarDGridDeacuerdoLista()
    Dim cnSQL As New SqlConnection
    Dim adSQL As SqlDataAdapter
    Dim DSet As New DataSet()
    Dim strSQL As String

    Try

        strSQL = "select * from
TbIFacturaVentaDetalleSalidaPF where IdFacturaVentaPF ="
& CInt(frmIntroRegVentaPF.txtIdFactura.Text)

        adSQL = New SqlDataAdapter(strSQL,
ConnectionString)
        adSQL.Fill(DSet, "TbIFacturaVentaDetalleSalidaPF")
        frmIntroRegVentaPF.grdDetalleVenta.DataSource =
DSet.Tables("TbIFacturaVentaDetalleSalidaPF")

    Catch Exp As SqlException
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

    Catch Exp As Exception
```

```
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
    End Try

    cnSQL.Close()
    cnSQL.Dispose()
End Sub
Private Sub CalculoSubTotal()
    If Not IsValidForm() Then
        Exit Sub
    End If

    lblSubTotal.Text = CStr(CDbl(txtCantidadAlIngresar.Text)
* CDbl(txtPrecio.Text))

End Sub
Private Sub CalculoTotal()
    lblTotal.Text = CStr(CDbl(lblSubTotal.Text) +
CDbl(lblTotal.Text)
Descuentos()
    If lblDescuento.Text = CStr(CDbl(0)) Then
        frmIntroRegVentaPF.lblSubTotalPF.Text =
CStr(CDbl(lblTotal.Text))
        frmIntroRegVentaPF.lblIVAPF.Text =
CStr(CDbl(frmIntroRegVentaPF.lblSubTotalPF.Text) / 100 *
CDbl(frmIntroRegVentaPF.txtIVAPF.Text))
        frmIntroRegVentaPF.lblGranTotalPF.Text =
CStr(CDbl(frmIntroRegVentaPF.lblSubTotalPF.Text) +
CDbl(frmIntroRegVentaPF.lblIVAPF.Text))
        frmIntroRegVentaPF.lblDescuentoPF.Text =
CStr(CDbl(lblDescuento.Text))
```

```
Else
    frmIntroRegVentaPF.lblDescuentoPF.Text =
CStr(CDb1(lblDescuento.Text))
    frmIntroRegVentaPF.lblSubTotalPF.Text =
CStr(CDb1(lblTotal.Text))
    frmIntroRegVentaPF.txtSubTotal2.Text =
CStr(CDb1(frmIntroRegVentaPF.lblSubTotalPF.Text)
Cdb1(frmIntroRegVentaPF.lblDescuentoPF.Text))
    frmIntroRegVentaPF.lblIVAPF.Text =
CStr(CDb1(frmIntroRegVentaPF.txtSubTotal2.Text) / 100 *
Cdb1(frmIntroRegVentaPF.txtIVAPF.Text))
    frmIntroRegVentaPF.lblGranTotalPF.Text =
CStr((CDb1(frmIntroRegVentaPF.txtSubTotal2.Text)
Cdb1(frmIntroRegVentaPF.lblIVAPF.Text)))
```

End If

End Sub

Private Sub AgregarProducto()

If Not IsValidForm() Then

Exit Sub

End If

Dim cnSQL As SqlConnection

Dim cmSQL As SqlCommand

Dim strSQL As String

Dim Fecha As String = Format(lblFecha.Text, "Short
Date")

```
Try
    ' Build Sentencia Insert para insertar un nuevo
    producto dentro de la tabla products
    strSQL = "INSERT INTO
    TbFacturaVentaDetalleSalidaPF
    (IdFacturaVentaPF,IdPF,FechaIngresoPF,NombrePF" & _
    ",PrecioPorUnidadPF,CantidadPF,SubTotalPF)
    VALUES (" & _
    CInt(lblIdVenta.Text) & "," & _
    CInt(lblIdPF.Text) & "," & _
    "" & Fecha & "" & "," & _
    PrepareStr(txtNombrePF.Text) & "," & _
    CDb(txtPrecio.Text) & "," & _
    CDb(txtCantidadAlIngresar.Text) & "," & _
    CDb(lblSubTotal.Text) & ")".

    cnSQL = New SqlConnection(ConnectionString)
    cnSQL.Open()

    cmSQL = New SqlCommand(strSQL, cnSQL)
    cmSQL.ExecuteNonQuery()

    ' Cerrar y Limpiar los objetos utilizados
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

    ' Refresh Product List
    LlenarListaFormulario()

Catch Exp As SqlException
```

```
        MsgBox(Exp.Message,  MsgBoxStyle.Critical,  "SQL
Error")

        Catch Exp As Exception
            MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
        End Try
    End Sub
    Private Function IsValidForm() As Boolean
        ' Validación de los campos
        If Not IsNumeric(txtCantidadAlIngresar.Text) Then
            MsgBox("Favor introduzca un valor válido para cada
campo", _
                MsgBoxStyle.Exclamation, Me.Text)
            Return False
        Else
            Return True
        End If
    End Function
End Class
```

The screenshot shows a software window titled 'InventarioMP'. On the left, there is a list box labeled 'Productos agregados:' containing the text 'IstMaterial'. To the right of this list box, the text 'Id Materia: Label8' is displayed. Below this, there are five input fields with labels: 'Material:', 'Cantidad:', 'Unidad de Medida:', 'Precio Por Unidad:', and 'Cantidad en Lata:'. To the right of these fields is a small image of a product label with the text 'SIVECOIN' and a graphic of hands. At the bottom of the window, there is a horizontal bar with four buttons: 'Nuevo', 'Borrar', 'Agregar', and 'Cerrar'.

Ventana de registro de materia prima, desde aquí se haran los registros de Materia Prima

Codigo de Registro de Materia Prima

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmIntroMP

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—

"Server=(Local)\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCerrar.Click
 Me.Close()

End Sub

Private Sub btnNew_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnNew.Click

 ClearForm()

 Mode = "Add"

 btnAdd.Enabled = True

```
    btnDelete.Enabled = False
    btnNew.Enabled = False
End Sub
```

```
Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click
    BorrarMaterial()
End Sub
```

```
Private Sub btnAdd_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAdd.Click
    txtCantidadLata.Text = CStr(CDbl(txtCantidad.Text) / 42)
    AgregarMaterial()
    btnAdd.Enabled = False
    btnNew.Enabled = True
End Sub
```

```
Private Sub frmIntroMP_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    PopulateProductList()
End Sub
```

```
Private Sub lstMaterial_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lstMaterial.SelectedIndexChanged
    PopulateForm()
    btnDelete.Enabled = True
```

```
Mode = "Update"
End Sub
Private Sub AgregarMaterial()
    If Not IsValidForm() Then
        Exit Sub
    End If

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim NomMaterial As String = txtNombre.Text
    Try
        ' Build Sentencia Insert para insertar un nuevo
        producto dentro de la tabla de Usuario
        strSQL = "INSERT TbIMateriaPrima VALUES (" & _
            PrepareStr(NomMaterial) & "," & _
            CDbI(txtCantidad.Text) & "," & _
            PrepareStr(txtUnidadDeMedida.Text) & "," & _
            CDbI(txtPrecioPorUnidad.Text) & "," & _
            CDbI(txtCantidadLata.Text) & ")"

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        cmSQL.ExecuteNonQuery()

        ' Cerrar y Limpiar los objetos utilizados
        cnSQL.Close()
        cmSQL.Dispose()
    End Try
End Sub
```

```
cnSQL.Dispose()

' Refresh Product List
PopulateProductList()
FindProductByName(lstMaterial, NomMaterial)
Catch Exp As SqlException
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub

Private Function IsValidForm() As Boolean
' Validación de los campos
If txtNombre.Text = "" Or _
    Not IsNumeric(txtCantidad.Text) Or _
    txtUnidadDeMedida.Text = "" Or _
    Not IsNumeric(txtPrecioPorUnidad.Text) Or _
    txtCantidadLata.Text = "" Then

    MsgBox("Favor introduzca un valor válido para cada
campo", _
        MsgBoxStyle.Exclamation, Me.Text)
    Return False
Else
    Return True
End If
End Function
```

```
Private Function PrepareStr(ByVal strValue As String) As  
String
```

```
'Rellenar con NULOS las cajas de texto vacias.
```

```
If strValue.Trim() = "" Then
```

```
Return "NULL"
```

```
Else
```

```
Return "" & strValue.Trim() & ""
```

```
End If
```

```
End Function
```

```
Private Sub BorrarMaterial()
```

```
' Esta sub es utilizada para eliminar el registro de  
productos
```

```
' cuando el usuario haga clicks en el boton Delete
```

```
Dim cnSQL As SqlConnection
```

```
Dim cmSQL As SqlCommand
```

```
Dim strSQL As String
```

```
Dim intRowsAffected As Integer
```

```
Try
```

```
' Contruir la sentenecia Delete
```

```
strSQL = "DELETE FROM TbIMateriaPrima " & _  
"WHERE IdMP = " & CInt(lblIdMP.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)  
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)  
intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
If intRowsAffected <> 1 Then
```

```
        MsgBox("Borrado Fallido. Id Usuario " &
        lblIdMP.Text & _
            " not found.", MsgBoxStyle.Critical, "Delete")
    End If

    ' Cierre y Limpieza de los objetos
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

    PopulateProductList()

    Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
        Error")

    Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General
        Error")
    End Try
End Sub

Private Sub ClearForm()
    lblIdMP.Text = ""
    txtNombre.Text = ""
    txtCantidad.Text = ""
    txtUnidadDeMedida.Text = ""
    txtPrecioPorUnidad.Text = ""
    txtCantidadLata.Text = ""
End Sub

Private Sub PopulateForm()
    Dim cnSQL As SqlConnection
```

```
Dim cmSQL As SqlCommand
Dim drSQL As SqlDataReader
Dim strSQL As String
Dim objListItem As ListItem
```

```
Try
```

```
' Tomar la llave primaria de la Listbox
objListItem = CType(lstMaterial.SelectedItem, ListItem)
```

```
' Construir la sentencia Select para encuestar la
información
```

```
strSQL = "SELECT * FROM TblMateriaPrima " & _
        "WHERE IdMP = " & objListItem.ID
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()
```

```
cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()
```

```
If drSQL.Read() Then
```

```
' Llenar la forma con los datos
lblIdMP.Text = drSQL.Item("IdMP").ToString()
txtNombre.Text() =
drSQL.Item("NombreMP").ToString()
txtCantidad.Text =
drSQL.Item("CantidadMP").ToString()
txtUnidadDeMedida.Text() =
drSQL.Item("UnidadDeMedidaMP").ToString()
```

```
        txtPrecioPorUnidad.Text =  
drSQL.Item("PrecioPorUnidadMP").ToString()  
        txtCantidadLata.Text =  
drSQL.Item("CantidadLataMP").ToString()  
  
    End If  
  
    ' Cerrar y limpiar objetos  
    drSQL.Close()  
    cnSQL.Close()  
    cmSQL.Dispose()  
    cnSQL.Dispose()  
  
    Catch e As SqlException  
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL  
Error")  
  
    Catch e As Exception  
        MsgBox(e.Message, MsgBoxStyle.Critical, "General  
Error")  
    End Try  
End Sub  
Private Sub PopulateProductList()  
    ' Este procedimiento llena la list box en la  
    ' form con una lista de productos disponibles desde la  
    ' database Northwind.  
  
    Dim cnSQL As SqlConnection  
    Dim cmSQL As SqlCommand  
    Dim drSQL As SqlDataReader
```

Dim strSQL As String

Dim objListItem As ListItem

Try

' Construir la sentencia Select statement para
encuestar los nombres de productos

strSQL = "SELECT NombreMP, IdMP FROM
TbIMateriaPrima"

cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

IstMaterial.Items.Clear()

' Lazo de control de lectura sobre la clase datareader.

Do While drSQL.Read()

objListItem = New
ListItem(drSQL.Item("NombreMP").ToString(), _
CInt(drSQL.Item("IdMP")))

IstMaterial.Items.Add(objListItem)

Loop

If IstMaterial.Items.Count > 0 Then

IstMaterial.SetSelected(0, True)

End If

' Close and Clean up objects

drSQL.Close()

```
cnSQL.Close()  
cmSQL.Dispose()  
cnSQL.Dispose()
```

```
Catch e As SqlException  
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL  
Error")
```

```
Catch e As Exception  
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "General  
Error")
```

```
End Try
```

```
End Sub
```

```
Private Sub FindProductByName(ByVal lbList As ListBox,  
ByVal strValue As String)
```

```
    ' Esta sub es utilizada para encontrar un Product en la  
List Boxt product
```

```
    ' y seleccionar un selected index de la list box a ese  
elemento.
```

```
Dim blsFound As Boolean
```

```
Dim ListItem As ListItem
```

```
Dim ItemCount As Integer
```

```
ListItem = New ListItem()
```

```
ItemCount = 0
```

```
' Lazo de control a la lista hasta encontrar el elemento
```

```
While Not blsFound And ItemCount <= lbList.Items.Count
```

```
- 1
```

```
ListItem = CType(lbList.Items(ItemCount), ListItem)
```

```
    If ListItem.Value = strValue Then
        IList.SelectedIndex = itemCount
        blsFound = True
    End If
    itemCount += 1
End While

If Not blsFound Then
    IList.SelectedIndex = 0
End If
End Sub

Private Sub ActualizarPrecio()
    ' Esta sub es utilizada para actualizar un registro
    existente
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer

    ' Validar el formulario.
    If Not IsValidForm() Then
        Exit Sub
    End If

    Try
        ' Construir una sentencia update para la tabla product
        con los datos
        ' en la forma.
        strSQL = "UPDATE TbMateriaPrima SET" & _
```

```
        "      PrecioPorUnidadMP      =      "      &
PrepareStr(txtPrecioPorUnidad.Text) & _
        "WHERE IdMP = " & Cint(IblIdMP.Text)

cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
intRowsAffected = cmSQL.ExecuteNonQuery()

If intRowsAffected <> 1 Then
    MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")
End If

' Cerrar y Limpiar los objetos
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch e As Exception
    MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub
End Class
```

The screenshot shows a software window titled "Registrar Nuevo Producto" with standard window controls (minimize, maximize, close) in the top right corner. The window contains the following elements:

- Productos agregados:** A list box containing the text "IstProducto".
- Id Producto:** A text field containing the value "Label6".
- Producto:** An empty text input field.
- Cantidad:** An empty text input field.
- Precio Por Unidad:** An empty text input field.
- Actualizar Precio:** A button with a small icon of a hand holding a coin and the text "SIVECOIN" above it.
- Navigation Buttons:** A row of four buttons at the bottom: "Nuevo", "Borrar", "Agregar", and "Cerrar".

Ventana de registro y actualizacion de precios, desde aquí el administrador podra actualizar los precios, y tanto el adminisyrador como el encargado de bodega podran registrar los productos nuevos

Codigo de Registro de producto final y actualizacion de Precios del producto final

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmIntroPF

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—
"Server=(Local)\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =

SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub Button4_Click(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles btnCerrar.Click

Me.Close()

End Sub

Private Sub btnAdd_Click(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles btnAdd.Click

AgregarProducto()

btnAdd.Enabled = False

```
        btnNew.Enabled = True
    End Sub
    Private Sub AgregarProducto()
        If Not IsValidForm() Then
            Exit Sub
        End If

        Dim cnSQL As SqlConnection
        Dim cmSQL As SqlCommand
        Dim strSQL As String
        Dim NomProducto As String = txtProducto.Text
        Try
            ' Build Sentencia Insert para insertar un nuevo
            producto dentro de la tabla de Usuario
            strSQL = "INSERT TblProductoFinal VALUES (" & _
                PrepareStr(NomProducto) & "," & _
                CInt(txtCantidad.Text) & "," & _
                CDbI(txtPrecioPorUnidad.Text) & ")"

            cnSQL = New SqlConnection(ConnectionString)
            cnSQL.Open()

            cmSQL = New SqlCommand(strSQL, cnSQL)
            cmSQL.ExecuteNonQuery()

            ' Cerrar y Limpiar los objetos utilizados
            cnSQL.Close()
            cmSQL.Dispose()
            cnSQL.Dispose()
        End Try
    End Sub
End Class
```

```
' Refresh Product List
PopulateProductList()
FindProductByName(lstProducto, NomProducto)
Catch Exp As SqlException
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
    MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub
Private Function IsValidForm() As Boolean
' Validación de los campos
If txtProducto.Text = "" Or _
    Not IsNumeric(txtCantidad.Text) Or _
    Not IsNumeric(txtPrecioPorUnidad.Text) Then

    MsgBox("Favor introduzca un valor válido para cada
campo", _
        MsgBoxStyle.Exclamation, Me.Text)
    Return False
Else
    Return True
End If
End Function
Private Function PrepareStr(ByVal strValue As String) As
String
'Rellenar con NULOS las cajas de texto vacias.
If strValue.Trim() = "" Then
    Return "NULL"
```

```
Else
    Return "" & strValue.Trim() & ""
End If
End Function

Private Sub btnDelete_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
    BorrarProducto()
End Sub
Private Sub BorrarProducto()
    ' Esta sub es utilizada para eliminar el registro de
productos
    ' cuando el usuario haga clicks en el boton Delete

Dim cnSQL As SqlConnection
Dim cmSQL As SqlCommand
Dim strSQL As String
Dim intRowsAffected As Integer

Try
    ' Contruir la sentenecia Delete
    strSQL = "DELETE FROM TblProductoFinal " & _
        "WHERE IdPF = " & CInt(txtIdPF.Text)

    cnSQL = New SqlConnection(ConnectionString)
    cnSQL.Open()

    cmSQL = New SqlCommand(strSQL, cnSQL)
    intRowsAffected = cmSQL.ExecuteNonQuery()
```

```
        If intRowsAffected <> 1 Then
            MsgBox("Borrado Fallido. Id Usuario " &
                txtIdPF.Text & _
                    " not found.", MsgBoxStyle.Critical, "Delete")
        End If

        ' Cierre y Limpieza de los objetos
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

        PopulateProductList()

    Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

    Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
    End Try
End Sub

Private Sub btnNew_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btnNew.Click
    ClearForm()
    Mode = "Add"
    btnAdd.Enabled = True
    btnDelete.Enabled = False
    btnNew.Enabled = False
End Sub
```

```
Private Sub ClearForm()  
    txtIdPF.Text = ""  
    txtProducto.Text = ""  
    txtCantidad.Text = ""  
    txtPrecioPorUnidad.Text = ""  
End Sub
```

```
Private Sub frmIntroPF_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
    PopulateProductList()
```

```
End Sub
```

```
Private Sub PopulateForm()
```

```
    Dim cnSQL As SqlConnection
```

```
    Dim cmSQL As SqlCommand
```

```
    Dim drSQL As SqlDataReader
```

```
    Dim strSQL As String
```

```
    Dim objListItem As ListItem
```

```
Try
```

```
    ' Tomar la llave primaria de la Listbox
```

```
    objListItem = CType(lstProducto.SelectedItem,  
ListItem)
```

```
    ' Construir la sentencia Select para encuestar la  
información
```

```
    strSQL = "SELECT * FROM TblProductoFinal " & _  
"WHERE IdPF = " & objListItem.ID
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

If drSQL.Read() Then
    ' Llenar la forma con los datos
    txtIdPF.Text = drSQL.Item("IdPF").ToString()
    txtProducto.Text = drSQL.Item("NombrePF").ToString()
    txtCantidad.Text = drSQL.Item("CantidadPF").ToString()
    txtPrecioPorUnidad.Text = drSQL.Item("PrecioPF").ToString()

End If

' Cerrar y limpiar objetos
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch e As Exception
```

```
        MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
    End Try
End Sub
Private Sub PopulateProductList()
    ' Este procedimiento llena la list box en la
    ' form con una lista de productos disponibles desde la
    ' database Northwind.

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem

    Try
        ' Construir la sentencia Select statement para
        encuestar los nombres de productos
        strSQL = "SELECT NombrePF, IdPF FROM
TbIProductoFinal"

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        drSQL = cmSQL.ExecuteReader()

        IstProducto.Items.Clear()

        ' Lazo de control de lectura sobre la clase datareader.
        Do While drSQL.Read()
```

```
        objListItem = New
        ListItem(drSQL.Item("NombrePF").ToString(), _
                CInt(drSQL.Item("IdPF")))
        IstProducto.Items.Add(objListItem)
    Loop

    If IstProducto.Items.Count > 0 Then
        IstProducto.SetSelected(0, True)
    End If

    ' Close and Clean up objects
    drSQL.Close()
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

    Catch e As SqlException
        MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")

    Catch e As Exception
        MsgBox(e.Message,    MsgBoxStyle.Critical,    "General
Error")
    End Try
End Sub

Private Sub FindProductByName(ByVal lbList As ListBox,
ByVal strValue As String)
    ' Esta sub es utilizada para encontrar un Product en la
List Box product
    ' y seleccionar un selected index de la list box a ese
elemento.
```

```
Dim blsFound As Boolean
Dim ListItem As ListItem
Dim itemCount As Integer

ListItem = New ListItem()

itemCount = 0
' Lazo de control a la lista hasta encontrar el elemento
While Not blsFound And itemCount <= lbList.Items.Count
- 1
    ListItem = CType(lbList.Items(itemCount), ListItem)

    If ListItem.Value = strValue Then
        lbList.SelectedIndex = itemCount
        blsFound = True
    End If
    itemCount += 1
End While

If Not blsFound Then
    lbList.SelectedIndex = 0
End If
End Sub

Private Sub lstProducto_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lstProducto.SelectedIndexChanged
    PopulateForm()
    btnDelete.Enabled = True
    Mode = "Update"
```

End Sub

```
Private Sub btnUpdatePrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdatePrice.Click
```

```
    ActualizarPrecio()
```

```
    MsgBox("El Precio se Ha actualizado correctamente", MsgBoxStyle.Information, "ACTUALIZACION")
```

```
End Sub
```

```
Private Sub ActualizarPrecio()
```

```
    ' Esta sub es utilizada para actualizar un registro existente
```

```
    Dim cnSQL As SqlConnection
```

```
    Dim cmSQL As SqlCommand
```

```
    Dim strSQL As String
```

```
    Dim intRowsAffected As Integer
```

```
    ' Validar el formulario.
```

```
    If Not IsValidForm() Then
```

```
        Exit Sub
```

```
    End If
```

```
    Try
```

```
        ' Construir una sentencia update para la tabla product con los datos
```

```
        ' en la forma.
```

```
        strSQL = "UPDATE TblProductoFinal SET" & _
```

```
            "        PrecioPF        =        "        &
```

```
PrepareStr(txtPrecioPorUnidad.Text) & _
```

```
            "WHERE IdPF = " & CInt(txtIdPF.Text)
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
intRowsAffected = cmSQL.ExecuteNonQuery()

If intRowsAffected <> 1 Then
    MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")
End If

' Cerrar y Limpiar los objetos
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch e As Exception
    MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub
End Class
```

The screenshot shows a window titled "Realizar Abonos de Compra". At the top, there are two labels: "N Orden: Label6" and "Estado de la Orden: Label6". Below these is a section titled "Estado Actual del Credito" which contains two more labels: "Debe: Label6" and "Total del Credito: Label7". The main area of the window is divided into two parts. On the left, there is a section labeled "Abono" with a sub-label "Abono:" followed by a text input field. On the right, there is a table titled "Abono de Compra al Credito" which is currently empty. At the bottom of the window, there are two buttons: "Agregar Abono" and "OK".

En esta Ventana se Podran hacer los abonos de las facturas de compra al credito

Codigo de Abono de Compras

Option Strict On

Option Explicit On

Imports System.Data

Imports System.Data.SqlClient

Public Class frmAbonosCreditCompraMP

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—
"Server=MAMA\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Private Mode As String = "Update"

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click

frmIntroRegCompraMP.Enabled = True

Me.Close()

End Sub

Private Sub
frmAbonosCreditCompraMP_ControlRemoved(ByVal sender
As Object, ByVal e As

```
System.ComponentModel.CancelEventArgs) Handles  
MyBase.Closing
```

```
    frmIntroRegCompraMP.Enabled = True
```

```
End Sub
```

```
Private Sub frmAbonosCreditCompraMP_Load(ByVal  
sender As System.Object, ByVal e As System.EventArgs)  
Handles MyBase.Load
```

```
    frmIntroRegCompraMP.Enabled = False
```

```
    leerTblCompraAbonoCredito()
```

```
    LeerDGAbono()
```

```
End Sub
```

```
Private Sub LeerDGAbono()
```

```
    Dim cnSQL As New SqlConnection
```

```
    Dim adSQL As SqlDataAdapter
```

```
    Dim DSet As New DataSet()
```

```
    Dim strSQL As String
```

```
Try
```

```
    strSQL = "SELECT AbonoCreditoMP AS  
Abonos,DeveCreditoMP AS Deuda,HaberCreditoMP AS 'Total  
Abonos',EstadoCreditoMP AS 'Estado de la Orden'" & _
```

```
    " FROM TblCompraAbonoCredito where IdCompraMP  
=" & CInt(frmIntroRegCompraMP.txtIdCompra.Text)
```

```
    ' cargar el grid de personal
```

```
    adSQL = New SqlDataAdapter(strSQL,  
ConnectionString)
```

```
    adSQL.Fill(DSet, "TblCompraAbonoCredito")
```

```
        grdAbonoCompra.DataSource =
DSet.Tables("TblCompraAbonoCredito")

        Catch Exp As SqlException
            MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

        Catch Exp As Exception
            MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
        End Try

        cnSQL.Close()
        cnSQL.Dispose()
    End Sub
    Private Sub leerTblCompraAbonoCredito()
        Dim cnSQL As SqlConnection
        Dim cmSQL As SqlCommand
        Dim drSQL As SqlDataReader
        Dim strSQL As String

        Try
            strSQL = "SELECT
            IdCompraMP,DeveCreditoMP,EstadoCreditoMP,TotalDeudaC
            ompraMP " & _
            " FROM TblCompraAbonoCredito where IdCompraMP
            = " & CInt(frmlIntroRegCompraMP.txtIdCompra.Text) & _
            " AND IdCreditoCompraMP = (Select
            max(IdCreditoCompraMP) from TblCompraAbonoCredito
```

```
where          IdCompraMP          ="          &
CInt(frmIntroRegCompraMP.txtIdCompra.Text) & ")"
```

```
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
drSQL = cmSQL.ExecuteReader()

' Lazo de control de lectura sobre la clase datareader.
If drSQL.Read() Then
    lblIdOrden.Text          =
(drSQL("IdCompraMP").ToString())
    lblDeve.Text          =
(drSQL("DeveCreditoMP").ToString())
    lblEstadoCredito.Text          =
(drSQL("EstadoCreditoMP").ToString())
    lblTotalCredito.Text          =
(drSQL("TotalDeudaCompraMP").ToString())
End If

' Close and Clean up objects
drSQL.Close()
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

Catch e As SqlException
    MsgBox(e.Message,    MsgBoxStyle.Critical,    "SQL
Error")
```

```
        Catch e As Exception
            MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
        End Try
    End Sub
    Private Function PrepareStr(ByVal strValue As String) As
String
        'Rellenar con NULOS las cajas de texto vacias.
        If strValue.Trim() = "" Then
            Return "NULL"
        Else
            Return "" & strValue.Trim() & ""
        End If
    End Function
    Private Sub AgregarAbono()
        If Not IsValidForm() Then
            Exit Sub
        End If

        Dim cnSQL As SqlConnection
        Dim cmSQL As SqlCommand
        Dim strSQL As String
        Dim IdCompra As String = lblIdOrden.Text

        Try
            ' Build Sentencia Insert para insertar un nuevo
            producto dentro de la tabla products
            strSQL = "INSERT TblCompraAbonoCredito VALUES
(" & _
                CInt(IdCompra) & "," & _
                CDbI(txtAbonoCreditoCompraMP.Text) & "," & _
```

```
CDbl(lblDeve.Text)
CDbl(txtAbonoCreditoCompraMP.Text) & "," & _
CDbl(0) & "," & _
PrepareStr(lblEstadoCredito.Text) & "," & _
CDbl(lblTotalCredito.Text) & ")"
cnSQL = New SqlConnection(ConnectionString)
cnSQL.Open()

cmSQL = New SqlCommand(strSQL, cnSQL)
cmSQL.ExecuteNonQuery()

' Cerrar y Limpiar los objetos utilizados
cnSQL.Close()
cmSQL.Dispose()
cnSQL.Dispose()

'Try
' Construir una sentencia update para la tabla product
con los datos
' en la forma.
'strSQL = "UPDATE TblCompraMP SET
NomProveedorCompraMP = " & PrepareStr(txtNombre.Text) &
",EmpProveedorCompraMP = " &
PrepareStr(txtEmpresa.Text) & " WHERE IdCompraMP = " &
CInt(txtIdCompra.Text)

'cnSQL = New SqlConnection(ConnectionString)
'cnSQL.Open()

'cmSQL = New SqlCommand(strSQL, cnSQL)
```

```
'intRowsAffected = cmSQL.ExecuteNonQuery()

'If intRowsAffected <> 1 Then
'MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")
'End If

Catch Exp As SqlException
MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL
Error")

Catch Exp As Exception
MsgBox(Exp.Message, MsgBoxStyle.Critical, "General
Error")
End Try
End Sub
Private Function IsValidForm() As Boolean
' Validación de los campos
If txtAbonoCreditoCompraMP.Text = "" Then
MsgBox("Favor Introduzca el Abono", _
MsgBoxStyle.Exclamation, Me.Text)
Return False
Else
Return True
End If
End Function

Private Sub btnAgregarAbono_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAgregarAbono.Click
AgregarAbono()
```

```
leerTblCompraAbonoCredito()
ActualizarHaberCreditoMP()
leerTblCompraAbonoCredito()
LeerDGAbono()
End Sub
Private Sub ActualizarHaberCreditoMP()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer

    Try
        ' Construir una sentencia update para la tabla product
con los datos
        ' en la forma.
        strSQL = "UPDATE TblCompraAbonoCredito SET " &
-
        "HaberCreditoMP = " & CDbl(lblTotalCredito.Text) -
CDbl(lblDeve.Text) & _
        " WHERE IdCreditoCompraMP = (Select
max(IdCreditoCompraMP) from TblCompraAbonoCredito)
AND IdCompraMP = " & CInt(lblIdOrden.Text)

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        intRowsAffected = cmSQL.ExecuteNonQuery()

        If intRowsAffected <> 1 Then
```

```
        MsgBox("Update Failed.", MsgBoxStyle.Critical,
"Update")
    End If

    ' Cerrar y Limpiar los objetos
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

    Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL
Error")

    Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General
Error")
    End Try
End Sub

Private Sub GroupBox1_Enter(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
GroupBox1.Enter

    End Sub
End Class
```

The screenshot shows a window titled "Orden de Entrega" (Delivery Order) with a toolbar at the top. Below the toolbar, there is a "Main Report" tab. The main content area displays the following information:

ORDEN DE ENTREGA

Orden N°: 41 Usuario: Fecha de Orden: Olive

Cliente: Saturday Empresa: Subreport

Producto	CantidadPF
Red	8.723
Fuchsia	27.529
Lime	12.316
Fuchsia	22.190
Lime	288
Silver	9.040
Blue	19.264
Blue	27.446
Gray	15.890
Yellow	24.370

Below the table, there is a summary row:

Producto	CantidadPF
Red	8.723

Este es el formulario de la orden de entrega del producto, aquí aparecerán los productos comprados por el cliente

Codigo Orden de Entrega de PF

Option Strict On

Option Explicit On

Imports CrystalDecisions.Shared

Imports System.data.Sql

Imports CrystalDecisions.CrystalReports

Imports CrystalDecisions.ReportSource

Imports System.Data

Imports System.Data.SqlClient

Public Class frmOrdenEVenta

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—

"Server=(Local)\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Friend WithEvents crFactVenta1 As SIVCOIN.crFactVenta

Private Mode As String = "Update"

Private Sub frmOrdenEVenta_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

llenarRreporteOrdenVenta()

End Sub

Private Sub llenarRreporteOrdenVenta()

Dim sqlConn As SqlConnection

```
Dim sqlDaProd As SqlDataAdapter
Dim sqlDaCate As SqlDataAdapter
Dim dsPc As New DBSBAMDataSet
Dim StrCommCate As String = "SELECT * From
TbIOrdenSalidaEntregaPF WHERE IdOrdenSalidaPF = " &
frmIntroRegVentaPF.txtIdOrdenEntregaPF.Text
Dim strCommProd As String = "SELECT * From
TbOrdenDetalleSalidaPF WHERE IdOrdenSalidaEntregaPF =
" & frmIntroRegVentaPF.txtIdOrdenEntregaPF.Text

Try
    'Crear los DataAdapters
    sqlConn = New SqlConnection(ConnectionString)
    sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
    sqlDaProd = New SqlDataAdapter(strCommProd,
sqlConn)

    'Poblar las tablas del dataset desde los dataAdaperts
    sqlDaCate.Fill(dsPc, "TbIOrdenSalidaEntregaPF")
    sqlDaProd.Fill(dsPc, "TbOrdenDetalleSalidaPF")
    'Poblar el informe con el dataSet y mostrarlo
    Dim info As New crOrdenVenta
    info.SetDataSource(dsPc)
    crwOrdenVentaPF.ReportSource = info

Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try
End Sub
```

```
Private Sub FrmFacturaVenta_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load  
    llenarRreporteOrdenVenta()  
End Sub  
End Class
```

Reportes de Facturas

Datos de Búsqueda

Tipo de búsqueda

Numero de Factura Lugar Lugar: inicio: Fecha Final:

Usuario Estado de la Factura

Fecha Tipo de Venta

Cliente En General

Main Report

REPORTE DE FACTURA

N Factur	Usuario	Fecha	Cliente	Empresa	Pago	Lugar	Descuento	Sub Total	IVA	Total	Estado
41	Olive	Saturday	Subreport		Date	3D Ris	57,05	293,58	269,62	244,64	Data titl
23.281	White	Sunday	Mail Label	Box	Boolean	Bubble	39,02	54,36	323,91	146,04	Legend
292	Silver	Friday	Mail Label		Currency	Area	18,69	217,26	147,71	115,38	Footnot
25.667	Red	Thursday	Cross-Tab		Number	3D Ris	151,41	303,33	176,73	46,64	Subtitle
28.253	Lime	Thursday	Mail Label	Box	Number	3D Sur	275,29	128,59	87,23	97,41	Data lak
12.316	Fuchsia	Sunday	Drill Down	Line	Time	Bubble	274,46	89,42	192,64	226,48	Data titl
15.890	Yellow	Wednesd	Mail Label	Line	Boolean	3D Sur	240,84	35,48	196,29	126,23	Group ti
18.766	Blue	Wednesd	Mail Label		Date	Doughi	55,37	324,39	246,26	113,23	Data lak
16.118	Blue	Thursday	Document		Boolean	Radar	139,77	296,58	227,04	99,30	Data2 tr
31.673	Green	Tuesday	Document		Number	Bar	50,97	58,29	267,77	155,73	Footnot

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Esta es la ventana del reporte de materia prima, donde se seleccionara el tipo de reporte que se dese crear, ya sea por fecha de faxcturacion, tipo de factura, lugar de compra y mas

Reporte de Facturas de venta

Option Strict On

Option Explicit On

Imports CrystalDecisions.Shared

Imports System.data.Sql

Imports CrystalDecisions.CrystalReports

Imports CrystalDecisions.ReportSource

Imports System.Data

Imports System.Data.SqlClient

Public Class ReporteFacturasVentaPF

Inherits System.Windows.Forms.Form

Protected Const SQL_CONNECTION_STRING As String =

—

"Server=(Local)\SQLEXPRESS;" & _

"DataBase=DBSBAM;" & _

"Integrated Security=SSPI;Connect Timeout=20"

Private ConnectionString As String =
SQL_CONNECTION_STRING

Private HasConnected As Boolean = False

Friend WithEvents crFactVenta1 As SIVECOIN.crFactVenta

Private Mode As String = "Update"

Private Sub ComboBox2_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs)

End Sub

```
Private Sub ReporteFacturasVentaPF_Load(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
    CheckBox1.Checked = False  
    CheckBox2.Checked = False  
    CheckBox3.Checked = False  
    CheckBox4.Checked = False  
    CheckBox5.Checked = False  
    CheckBox6.Checked = False  
    CheckBox7.Checked = False  
    CheckBox8.Checked = False
```

```
End Sub
```

```
Private Function PrepareStr(ByVal strValue As String) As  
String
```

```
    'Rellenar con NULOS las cajas de texto vacias.
```

```
    If strValue.Trim() = "" Then
```

```
        Return "NULL"
```

```
    Else
```

```
        Return "" & strValue.Trim() & ""
```

```
    End If
```

```
End Function
```

```
Private Sub btnHacBusqueda_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnHacBusqueda.Click
```

```
    If CheckBox8.Checked = True Then
```

```
        EnGeneral()
    Else
        BusqChecked()
    End If
End Sub
Private Sub BusqChecked()
    If CheckBox1.Checked = True Then
        BusqNFact()
    ElseIf CheckBox2.Checked = True Then
        BusqUsuario()
    ElseIf CheckBox3.Checked = True Then
        BusqFecha()
    ElseIf CheckBox4.Checked = True Then
        BusqCliente()
    ElseIf CheckBox5.Checked = True Then
        BusqLugar()
    ElseIf CheckBox6.Checked = True Then
        BusqEstadoFact()
    ElseIf CheckBox7.Checked = True Then
        BusqTipoVenta()
    ElseIf CheckBox8.Checked = True Then
        EnGeneral()
    End If
End Sub

Private Sub BusqNFact()
    Dim sqlConn As SqlConnection
    Dim sqlDaCate As SqlDataAdapter
    Dim dsPc As New DBSBAMDataSet
```

```
Dim StrCommCate As String = "SELECT * From  
TbIFacturaVentaPF Where IdFacturaVentaPF = " &  
CInt(txtValor1.Text)
```

```
Try
```

```
'Crear los DataAdapters
```

```
sqlConn = New SqlConnection(ConnectionString)
```

```
sqlDaCate = New SqlDataAdapter(StrCommCate,  
sqlConn)
```

```
'Poblar las tablas del dataset desde los dataAdaperts
```

```
sqlDaCate.Fill(dsPc, "TbIFacturaVentaPF")
```

```
'Poblar el informe con el dataSet y mostrarlo
```

```
Dim info As New crReporteFacturaVentaPF
```

```
info.SetDataSource(dsPc)
```

```
crwVentaFechaPF.ReportSource = info
```

```
Label1.Visible = True
```

```
txtValor1.Visible = True
```

```
Catch ex As Exception
```

```
MessageBox.Show(ex.ToString)
```

```
End Try
```

```
End Sub
```

```
Private Sub BusqUsuario()
```

```
Dim sqlConn As SqlConnection
```

```
Dim sqlDaCate As SqlDataAdapter
```

```
Dim dsPc As New DBSBAMDataSet
```

```
Dim StrCommCate As String = "SELECT * From  
TbIFacturaVentaPF Where NomUsuario = " &  
PrepareStr(txtValor1.Text)
```

```
Try
```

```
'Crear los DataAdapters
```

```
        sqlConn = New SqlConnection(ConnectionString)
        sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
        'Poblar las tablas del dataset desde los dataAdaperts
        sqlDaCate.Fill(dsPc, "TblFacturaVentaPF")
        'Poblar el informe con el dataSet y mostrarlo
        Dim info As New crReporteFacturaVentaPF
        info.SetDataSource(dsPc)
        crwVentaFechaPF.ReportSource = info
        Label1.Visible = True
        txtValor1.Visible = True
    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
End Sub
Private Sub BusqFecha()
    Dim sqlConn As SqlConnection
    Dim sqlDaCate As SqlDataAdapter
    Dim dsPc As New DBSBAMDataSet
    Dim StrCommCate As String = "SELECT * From
TblFacturaVentaPF Where FechaFactura Between " &
PrepareStr(txtValor1.Text) & " And " &
PrepareStr(txtValor2.Text)

    Try
        'Crear los DataAdapters
        sqlConn = New SqlConnection(ConnectionString)
        sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
        'Poblar las tablas del dataset desde los dataAdaperts
```

```
        sqlDaCate.Fill(dsPc, "TbIFacturaVentaPF")
        'Poblar el informe con el dataSet y mostrarlo
        Dim info As New crReporteFacturaVentaPF
        info.SetDataSource(dsPc)
        crwVentaFechaPF.ReportSource = info

    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
End Sub

Private Sub BusqCliente()
    Dim sqlConn As SqlConnection
    Dim sqlDaCate As SqlDataAdapter
    Dim dsPc As New DBSBAMDataSet
    Dim StrCommCate As String = "SELECT * From
TbIFacturaVentaPF Where IDCliente = " &
PrepareStr(txtValor1.Text) & " And EmpCliente = " &
PrepareStr(txtValor2.Text)

    Try
        'Crear los DataAdapters
        sqlConn = New SqlConnection(ConnectionString)
        sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
        'Poblar las tablas del dataset desde los dataAdaperts
        sqlDaCate.Fill(dsPc, "TbIFacturaVentaPF")
        'Poblar el informe con el dataSet y mostrarlo
        Dim info As New crReporteFacturaVentaPF
        info.SetDataSource(dsPc)
        crwVentaFechaPF.ReportSource = info
    End Try
End Sub
```

```
    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
End Sub
Private Sub BusqLugar()
    Dim sqlConn As SqlConnection
    Dim sqlDaCate As SqlDataAdapter
    Dim dsPc As New DBSBAMDataSet
    Dim StrCommCate As String = "SELECT * From
TbIFacturaVentaPF Where LugarFacturado = " &
PrepareStr(cbValor1.Text)

    Try
        'Crear los DataAdapters
        sqlConn = New SqlConnection(ConnectionString)
        sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
        'Poblar las tablas del dataset desde los dataAdaperts
        sqlDaCate.Fill(dsPc, "TbIFacturaVentaPF")
        'Poblar el informe con el dataSet y mostrarlo
        Dim info As New crReporteFacturaVentaPF
        info.SetDataSource(dsPc)
        crwVentaFechaPF.ReportSource = info

    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
End Sub
Private Sub BusqEstadoFact()
```

```
Dim sqlConn As SqlConnection
Dim sqlDaCate As SqlDataAdapter
Dim dsPc As New DBSBAMDataSet
Dim StrCommCate As String = "SELECT * From
TbIFacturaVentaPF Where EstadoCOMpraPF = " &
PrepareStr(cbAnulado.Text)
```

```
Try
```

```
    'Crear los DataAdapters
    sqlConn = New SqlConnection(ConnectionString)
    sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
    'Poblar las tablas del dataset desde los dataAdaperts
    sqlDaCate.Fill(dsPc, "TbIFacturaVentaPF")
    'Poblar el informe con el dataSet y mostrarlo
    Dim info As New crReporteFacturaVentaPF
    info.SetDataSource(dsPc)
    crwVentaFechaPF.ReportSource = info
```

```
Catch ex As Exception
```

```
    MessageBox.Show(ex.ToString)
```

```
End Try
```

```
End Sub
```

```
Private Sub BusqTipoVenta()
```

```
    Dim sqlConn As SqlConnection
```

```
    Dim sqlDaCate As SqlDataAdapter
```

```
    Dim dsPc As New DBSBAMDataSet
```

```
    Dim StrCommCate As String = "SELECT * From
TbIFacturaVentaPF Where TipoPagoPF = " &
PrepareStr(cbValor2.Text)
```

```
Try
    'Crear los DataAdapters
    sqlConn = New SqlConnection(ConnectionString)
    sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
    'Poblar las tablas del dataset desde los dataAdaperts
    sqlDaCate.Fill(dsPc, "TblFacturaVentaPF")
    'Poblar el informe con el dataSet y mostrarlo
    Dim info As New crReporteFacturaVentaPF
    info.SetDataSource(dsPc)
    crwVentaFechaPF.ReportSource = info

Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try
End Sub
Private Sub EnGeneral()
    Dim sqlConn As SqlConnection
    Dim sqlDaCate As SqlDataAdapter
    Dim dsPc As New DBSBAMDataSet
    Dim StrCommCate As String = "SELECT * From
TblFacturaVentaPF"

Try
    'Crear los DataAdapters
    sqlConn = New SqlConnection(ConnectionString)
    sqlDaCate = New SqlDataAdapter(StrCommCate,
sqlConn)
```

```
'Poblar las tablas del dataset desde los dataAdaperts  
sqlDaCate.Fill(dsPc, "TbIFacturaVentaPF")  
'Poblar el informe con el dataSet y mostrarlo  
Dim info As New crReporteFacturaVentaPF  
info.SetDataSource(dsPc)  
crwVentaFechaPF.ReportSource = info
```

```
Catch ex As Exception  
    MessageBox.Show(ex.ToString)  
End Try  
End Sub
```

```
Private Sub btnBusqOK_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnBusqOK.Click
```

```
    Acciones()
```

```
End Sub
```

```
Private Sub Acciones()
```

```
    If CheckBox1.Checked = False And CheckBox2.Checked  
= False And CheckBox3.Checked = False And  
CheckBox4.Checked = False And CheckBox5.Checked =  
False And CheckBox6.Checked = False And  
CheckBox7.Checked = False And CheckBox8.Checked =  
False Then
```

```
        txtValor2.Visible = False
```

```
        Label5.Visible = False
```

```
        Label1.Visible = False
```

```
        txtValor1.Visible = False
```

```
        Label2.Visible = False
```

```
        cbValor1.Visible = False
```

```
MsgBox("Seleccione un tipo de busqueda",  
MsgBoxStyle.Exclamation, "Sugerencia")
```

```
Else
```

```
  If CheckBox1.Checked = True Then
```

```
    Label1.Text = "Numero de Factura:"
```

```
    Label1.Visible = True
```

```
    txtValor1.Visible = True
```

```
    Label5.Visible = False
```

```
    txtValor2.Visible = False
```

```
    Label2.Visible = False
```

```
    cbValor1.Visible = False
```

```
    cbValor2.Visible = False
```

```
    cbAnulado.Visible = False
```

```
  ElseIf CheckBox2.Checked = True Then
```

```
    Label1.Text = "Nombre:"
```

```
    Label1.Visible = True
```

```
    txtValor1.Visible = True
```

```
    Label5.Visible = False
```

```
    txtValor2.Visible = False
```

```
    Label2.Visible = False
```

```
    cbValor1.Visible = False
```

```
    cbValor2.Visible = False
```

```
    cbAnulado.Visible = False
```

```
  ElseIf CheckBox3.Checked = True Then
```

```
    Label1.Text = "Fecha Inicio:"
```

```
    Label1.Visible = True
```

```
    txtValor1.Visible = True
```

```
    Label5.Text = "Fecha Final:"
```

```
    Label5.Visible = True
```

```
    txtValor2.Visible = True
```

```
    Label2.Visible = False
```

```
cbValor1.Visible = False
cbValor2.Visible = False
cbAnulado.Visible = False
Elseif CheckBox4.Checked = True Then
Label1.Text = "Nombre:"
Label1.Visible = True
txtValor1.Visible = True
Label5.Text = "Empresa:"
Label5.Visible = True
txtValor2.Visible = True
Label2.Visible = False
cbValor1.Visible = False
cbValor2.Visible = False
cbAnulado.Visible = False
Elseif CheckBox5.Checked = True Then
Label1.Visible = False
txtValor1.Visible = False
Label5.Visible = False
txtValor2.Visible = False
Label2.Text = "Lugar Facturado:"
Label2.Visible = True
cbValor1.Visible = True
cbValor2.Visible = False
cbAnulado.Visible = False
Elseif CheckBox6.Checked = True Then
Label1.Visible = False
txtValor1.Visible = False
Label5.Visible = False
txtValor2.Visible = False
Label2.Text = "Estado de la Factura:"
Label2.Visible = True
```

```
        cbValor1.Visible = False
        cbValor2.Visible = False
        cbAnulado.Visible = True
    ElseIf CheckBox7.Checked = True Then
        Label1.Visible = False
        txtValor1.Visible = False
        Label5.Visible = False
        txtValor2.Visible = False
        Label2.Text = "Tipo de Venta:"
        Label2.Visible = True
        cbValor1.Visible = False
        cbValor2.Visible = True
        cbAnulado.Visible = False
    ElseIf CheckBox8.Checked = True Then
        Label1.Visible = False
        txtValor1.Visible = False
        Label5.Visible = False
        txtValor2.Visible = False
        Label2.Visible = False
        cbValor1.Visible = False
        cbValor2.Visible = False
        cbAnulado.Visible = False
    End If
End If
End Sub
```

```
Private Sub btnCerrar_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnCerrar.Click
    Me.Close()
End Sub
```

```
Private Sub crwVentaFechaPF_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
crwVentaFechaPF.Load
```

```
End Sub  
End Class
```

Codigo del List Item

Este codgo se usa para enlistar por ID los datos de una tabla

Option Strict On
Option Explicit On

' This class is used to store the text and primary key of a record
' that you want to list in list box.

Public Class ListItem

Private mValue As String ' Stores a named description of the item

Private mID As Integer ' Stores a primary key to the record

Public Sub New(ByVal strValue As String, ByVal intID As Integer)

mValue = strValue

mID = intID

End Sub

Public Sub New()

mValue = ""

mID = 0

End Sub

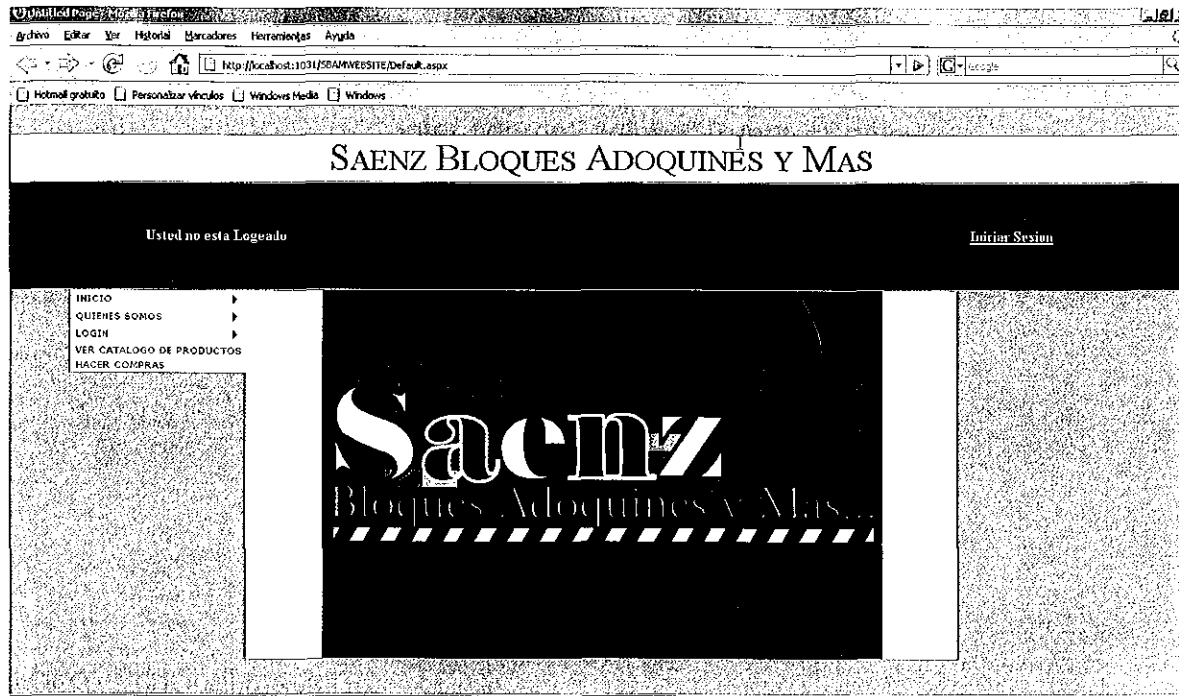
Property ID() As Integer

```
    Get
        Return mID
    End Get
    Set(ByVal Value As Integer)
        mID = Value
    End Set
End Property

Property Value() As String
    Get
        Return mValue
    End Get
    Set(ByVal Value As String)
        mValue = Value
    End Set
End Property

' When a list box displays an item in its collection
' it calls the ToString method to get the value to
' display so we will need to override this method
' so our class will return mValue.
Public Overrides Function ToString() As String
    Return mValue
End Function
End Class
```

1.2 Aplicación Web



Esta pantalla es el Intro del Sitio Web de La empresa

Codigo del Intro del Sitio Web "Default.aspx"

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"
AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="_Default" %>

    <asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
    <object style="width: 100%; height: 400px"
data="Multimedia/SaenzBloquesAdYMas.swf" name="SBAM">
    </object>
</asp:Content>
```

Marco Principal.master

```
<%@ Master Language="VB" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">

    Protected Sub Menu1_MenuItemClick(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.MenuEventArgs)

    End Sub

    Protected Sub LoginStatus1_LoggingOut(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.LoginCancelEventArgs)
```

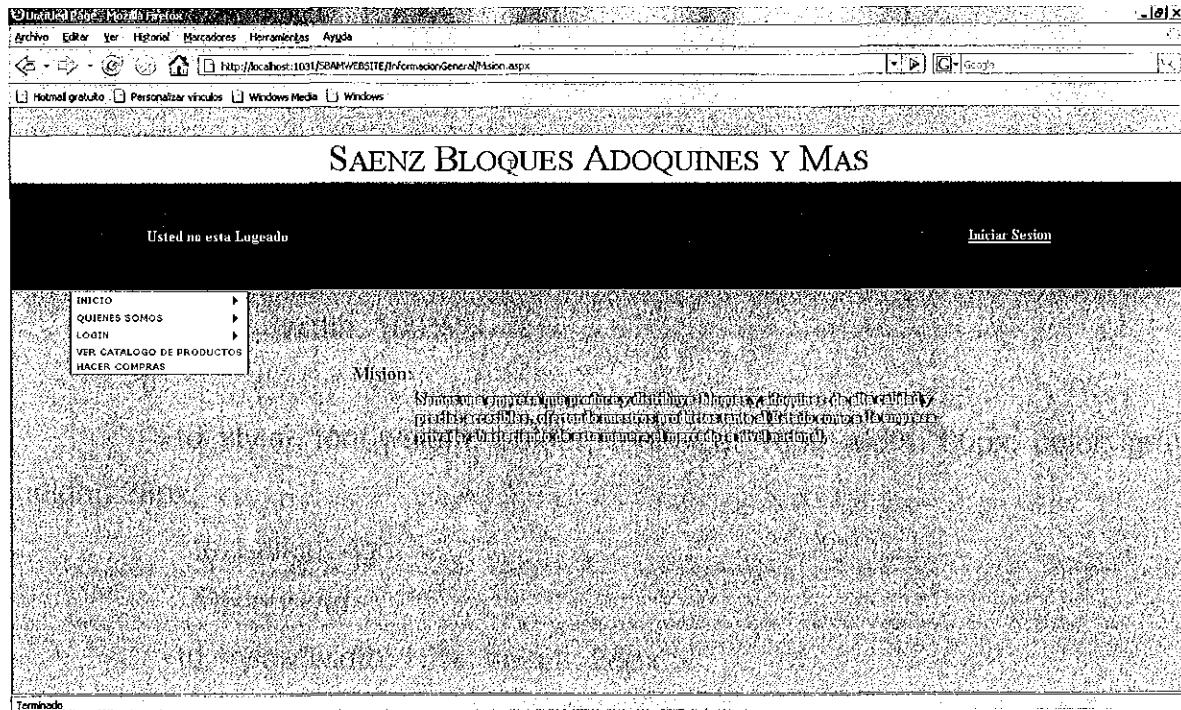
```
End Sub
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs)
End Sub
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body style="width: 100%; margin: 0px; clip: rect(0px 0px 0px 0px); height: 100%;"
bgcolor="#ff9933">
  <form id="form1" runat="server">
    <div style="padding-right: 0px; padding-left: 0px; padding-bottom: 0px; margin:
0px; width: 100%; padding-top: 0px; height: 100%">
      <table border="0" cellpadding="0" cellspacing="0" style="width: 100%;">
        <tr>
          <td colspan="3" style="height: 200px; vertical-align: bottom; text-align:
center;" align="center" valign="top">
            <table width="100%">
              <tr>
                <td style="font-size: 40px; background-color: yellow; font-variant:
small-caps;" align="center" valign="top">
                  Saenz Bloques Adoquines y Mas</td>
              </tr>
            </table>
          </td>
        </tr>
        <tr>
          <td style="width: 100%; background-color: maroon; height:
115px;">
            <table style="width: 100%">
              <tr>
                <td style="font-weight: bolder; width: 35%; color: white">
                  <asp:LoginView ID="LoginView1" runat="server">

```

```
<LoggedInTemplate>
    Usted Esta "CONECTADO"
</LoggedInTemplate>
<AnonymousTemplate>
    Usted no esta Logeado&nbsp;
</AnonymousTemplate>
</asp:LoginView>
</td>
<td style="font-weight: bolder; width: 35%; color: white">
    <asp:LoginName ID="LoginName1" runat="server" />
</td>
<td style="font-weight: bolder; width: 100%; color: white">
    <asp:LoginStatus ID="LoginStatus1" runat="server"
ForeColor="White" LoginText="Iniciar Sesion"
    LogoutText="Cerrar Sesion" />
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr>
    <td style="width: 20%; height: 250px;" align="right" valign="top">
        <asp:Menu ID="Menu1" runat="server" BackColor="#FFFBD6"
DynamicHorizontalOffset="2"
        Font-Names="Verdana" Font-Size="0.8em" ForeColor="#990000"
StaticSubMenuIndent="10px">
            <StaticMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" />
```

```
<DynamicHoverStyle BackColor="#990000" ForeColor="White" />
<DynamicMenuStyle BackColor="#FFFBD6" />
<StaticSelectedStyle BackColor="#FFCC66" />
<DynamicSelectedStyle BackColor="#FFCC66" />
<DynamicMenuItemStyle          HorizontalPadding="5px"
VerticalPadding="2px" />
  <Items>
    <asp:MenuItem  NavigateUrl="~/Default.aspx"  Text="INICIO"
Value="INICIO">
      <asp:MenuItem
NavigateUrl="~/InformacionGeneral/Mision.aspx"          Text="MISION"
Value="MISION">
        </asp:MenuItem>
      <asp:MenuItem
NavigateUrl="~/InformacionGeneral/Vision.aspx" Text="VISION" Value="VISION">
        </asp:MenuItem>
      <asp:MenuItem
NavigateUrl="~/InformacionGeneral/Objetivos.aspx" Text="OBJETIVOS"
Value="OBJETIVOS"></asp:MenuItem>
        </asp:MenuItem>
      <asp:MenuItem
NavigateUrl="~/InformacionGeneral/QuienesSomos.aspx"          Text="QUIENES
SOMOS"
Value="QUIENES SOMOS">
        <asp:MenuItem
NavigateUrl="~/InformacionGeneral/NuestrosProductos.aspx"  Text="NUESTROS
PRODUCTOS"
Value="NUESTROS PRODUCTOS"></asp:MenuItem>
        </asp:MenuItem>
      <asp:MenuItem          Text="LOGIN"          Value="LOG
IN"
NavigateUrl="~/Login.aspx">
```

```
<asp:MenuItem Text="NUEVO USUARIO" Value="NUEVO
USUARIO" NavigateUrl="~/Registro/NuevoUsuario.aspx"></asp:MenuItem>
<asp:MenuItem Text="CAMBIAR CONTRASE&#209;A"
Value="CAMBIAR CONTRASE&#209;A"
NavigateUrl="~/UsuarioRegistrado/ChangePass.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="VER CATALOGO DE PRODUCTOS"
Value="CATALOGO DE PRODUCTOS"
NavigateUrl="~/InformacionGeneral/Catalogo.aspx"></asp:MenuItem>
<asp:MenuItem
NavigateUrl="~/UsuarioRegistrado/ConfirmarUsuario.aspx" Text="HACER
COMPRAS"
Value="HACER COMPRAS"></asp:MenuItem>
</Items>
<StaticHoverStyle BackColor="#990000" ForeColor="White" />
</asp:Menu>
&nbsp; &nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
</td>
<td style="width: 60%; height: 250px">
<asp:contentplaceholder id="ContenedordePaginas" runat="server">
</asp:contentplaceholder>
</td>
<td style="width: 20%; height: 250px">
</td>
</tr>
</table>
</div>
</form>
</body>
</html>
```



En esta pantalla se muestra la Misión de la empresa

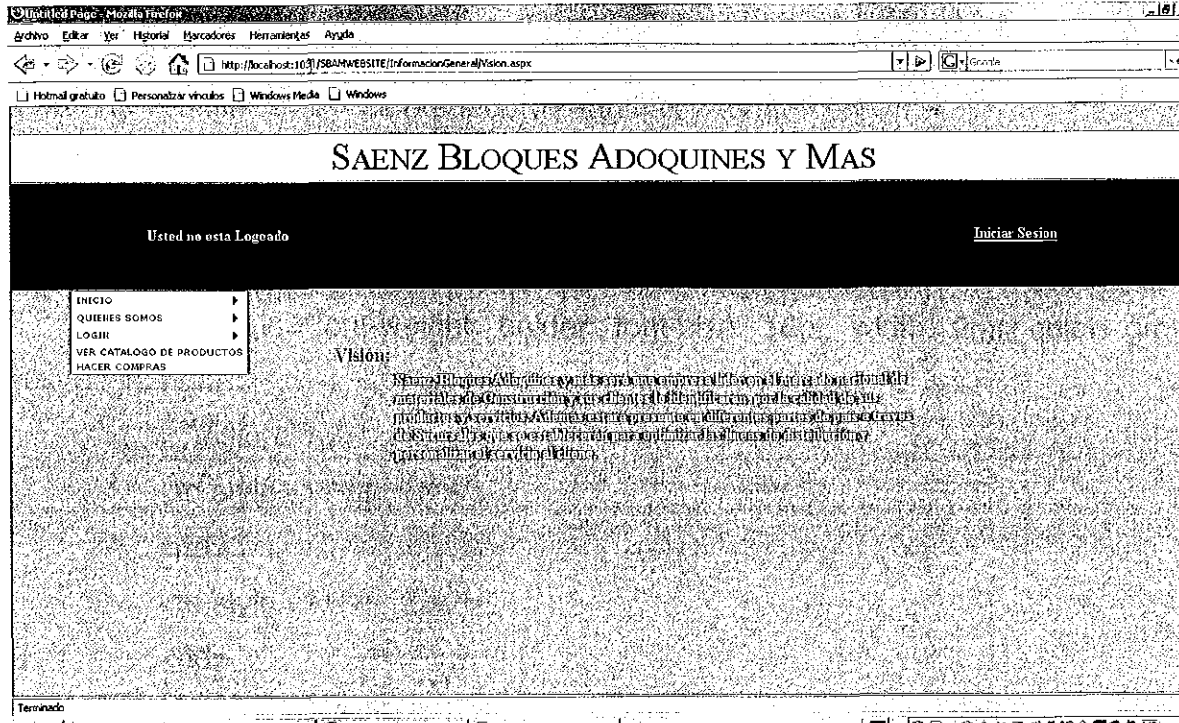
Codigo de "Mision.aspx"

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"%>

<asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
  <div>
    <table style="width: 100%">
      <tr>
        <td style="font-weight: bolder; font-size: 16pt; width: 86px; color: gray;
height: 28px;
text-align: right">
          Mision:</td>
        <td style="width: 75%; height: 28px">
          </td>
        <td style="width: 2px; height: 28px">
          </td>
      </tr>
      <tr>
        <td style="width: 86px">
          </td>
        <td style="font-weight: bolder; width: 75%; color: white">
          Somos una empresa que produce y distribuye &nbsp;bloques y
          adoquines&nbsp; de alta
          calidad y precios accesibles,&nbsp; ofertando nuestros productos
          tanto al Estado
          como a &nbsp;la empresa privada. abasteciendo de esta manera,el
          mercado&nbsp; a
          nivel nacional,&nbsp;<br />
          </td>
        <td style="width: 2px; height: 21px">
          </td>
      </tr>
    </table>
  </div>
</asp:Content>
```

```
        </td>
    </tr>
    <tr>
        <td style="width: 2px">
        </td>
    </tr>
</table>

</div>
</asp:Content>
```



En esta ventana se visualizara la Vision de la empresa

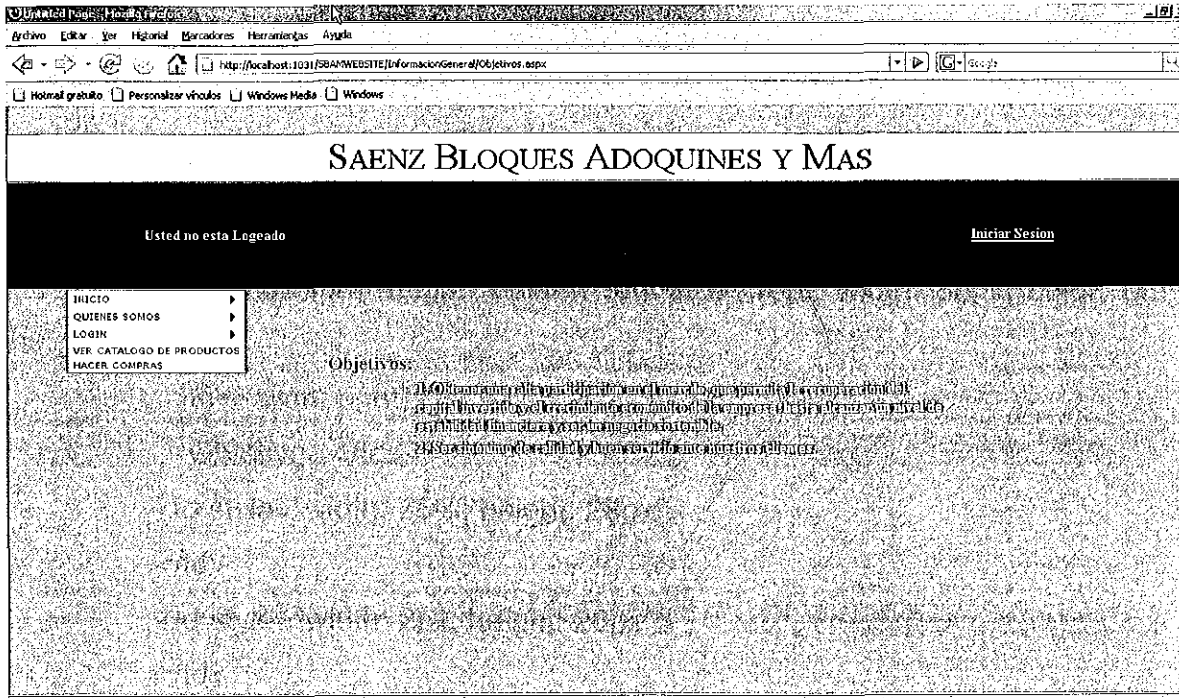
Codigo de "Vision.aspx"

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"%>

<asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
  <div>
    <table style="width: 100%">
      <tr>
        <td style="font-weight: bolder; font-size: 16pt; width: 2px; color: gray;
text-align: right">
          Visión:</td>
        <td style="width: 75%">
          </td>
        <td style="width: 2px">
          </td>
      </tr>
      <tr>
        <td style="width: 20%">
          </td>
        <td style="font-weight: bolder; width: 75%; color: white">
          Sáenz, Bloques Adoquines y más será una empresa líder en el
mercado nacional de
          materiales de Construcción y sus clientes lo identificarán&nbsp;   por la
calidad
          de sus productos y servicios. Además estará presente en diferentes
partes de país
          a traves de Sucursales que se establecerán para optimizar las líneas
de disteibución
          y personalizar el servicio al cliene.<br />
        </td>
      </tr>
    </table>
  </div>
</asp:Content>
```

```
<td style="width: 2px; height: 21px">
</td>
</tr>
<tr>
<td style="width: 2px">
</td>
<td style="width: 75%">
</td>
<td style="width: 2px">
</td>
</tr>
</table>

</div>
</asp:Content>
```



En esta ventana se visualizarán los objetivos de la empresa

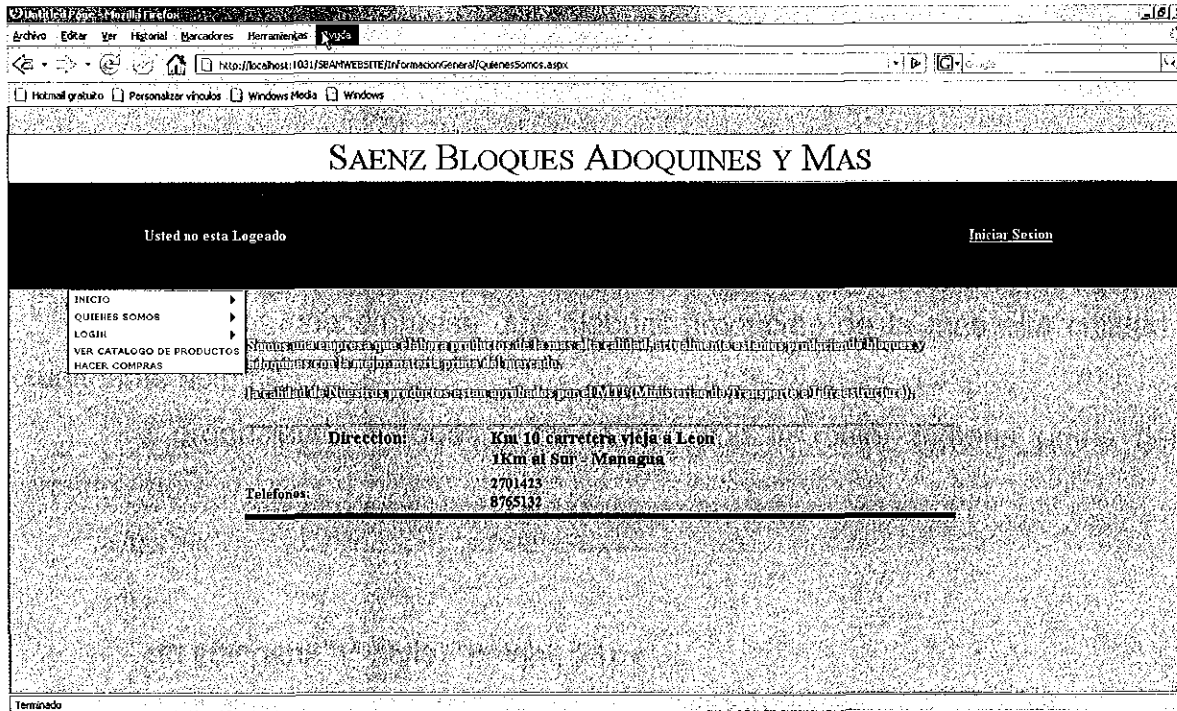
Codigo de "Objetivos.aspx"

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"%>

<asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
  <div>
    <table style="width: 100%">
      <tr>
        <td style="font-weight: bolder; font-size: 16pt; width: 86px; color: gray;
height: 28px;
text-align: right">
          Objetivos:</td>
        <td style="width: 75%; height: 28px">
          </td>
        <td style="width: 2px; height: 28px">
          </td>
      </tr>
      <tr>
        <td style="width: 86px">
          </td>
        <td style="font-weight: bolder; width: 75%; color: white">
          1. Obtener una&nbsp;alta participación en el mercd, que permita la
recuperación
          del capital invertido y el crecimiento económico de la empresa
          &nbsp;hasta alcanzar
          un nivel de estabilidad financiera y ser un negocio sostenible.<br />
        </td>
        <td style="width: 2px; height: 21px">
          </td>
      </tr>
    </table>
  </div>
</asp:Content>
```

```
</tr>
<tr>
  <td style="width: 86px">
  </td>
  <td style="font-weight: bolder; width: 75%; color: white">
    2. Ser sinónimo de calidad y buen servicio ante nuestros
clientes.</td>
  <td style="width: 2px">
  </td>
</tr>
</table>

</div>
</asp:Content></asp:Content>
```



Desde esta ventana se vera una pequeña descripción de la empresa

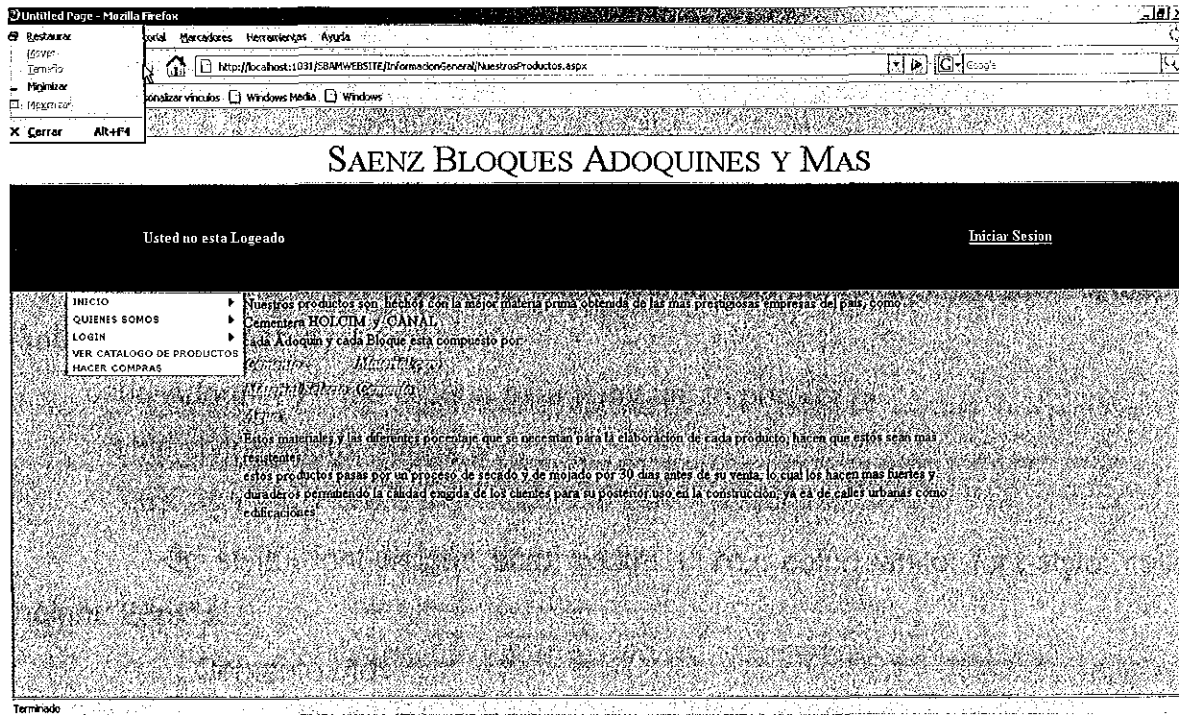
Codigo "QuienesSomos.aspx"

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"%>
<asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
  <div>
    <table style="margin-top: 5%; font-weight: bolder; color: white" width="100%">
      <tr>
        <td style="height: 57px">
          Somos una empresa que elabora productos de la mas alta calidad,
          actualmente estamos
          produciendo bloques y adoquines con la mejor materia prima del
          mercado.</td>
      </tr>
      <tr>
        <td colspan="1" style="height: 21px">
          la calidad de Nuestros productos estan aprobados por el MTI
          (Ministerio de Transporte
          e Infraestructura),
        </td>
      </tr>
      <tr>
        <td rowspan="2" style="height: 21px">
        </td>
      </tr>
    </table>
    <table cellpadding="0" cellspacing="0" style="width: 100%">
      <tr>
        <td align="right" style="padding-right: 0px; padding-left: 0px; font-weight:
bolder;
```

```
font-size: 14pt; padding-bottom: 0px; vertical-align: top; width: 88px;
color: blue;
border-top-style: outset; padding-top: 0px; height: 48px; text-align:
center"
valign="top">
Direccion:
</td>
<td align="left" style="padding-right: 0px; padding-left: 0px; font-weight:
bolder;
font-size: 14pt; padding-bottom: 0px; vertical-align: top; width: 88px;
color: blue;
border-top-style: outset; padding-top: 0px; height: 52px; text-align: left"
valign="top">
Km 10 carretera vieja a Leon 1Km al Sur - Managua</td>
<td style="padding-right: 0px; padding-left: 0px; font-weight: bolder; font-
size: 14pt;
padding-bottom: 0px; vertical-align: top; width: 88px; color: blue;
border-top-style: outset;
padding-top: 0px; height: 52px; text-align: left">
</td>
</tr>
<tr>
<td align="left" style="font-weight: bolder; color: blue; height: 48px; text-
align: left; border-bottom-style: outset;"
valign="middle">
Telefonos:
</td>
<td align="left" style="font-weight: bolder; color: blue; height: 48px; text-
align: left; border-bottom-style: outset;"
valign="middle">
2701423<br />
```

```
8765132</td>
<td align="left" style="font-weight: bolder; color: blue; height: 48px; text-align: left; border-bottom-style: outset;"
    valign="middle">
</td>
</tr>
<tr>
<td colspan="1" rowspan="1" style="padding-right: 0px; padding-left: 0px; font-weight: bolder; font-size: 14pt; padding-bottom: 0px; vertical-align: top; width: 333px; color: blue; border-top-style: solid; padding-top: 0px; text-align: right">
</td>
<td style="padding-right: 0px; padding-left: 0px; font-weight: bolder; font-size: 14pt; padding-bottom: 0px; vertical-align: top; width: 333px; color: blue; border-top-style: solid; padding-top: 0px; text-align: right">
</td>
<td style="padding-right: 0px; padding-left: 0px; font-weight: bolder; font-size: 14pt; padding-bottom: 0px; vertical-align: top; width: 333px; color: blue; border-top-style: solid; padding-top: 0px; text-align: right">
</td>
</tr>
</table>
<br />
<br />

</div>
</asp:Content>
```



Aquí se Visualizara una pequeña descripción de los producto elaborados en la empresa

Codigo de NuestrosProductosas.aspx

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"%>

<asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
  <div>
    Nuestros productos son&nbsp; hechos con la mejor materia prima obtenida
de las mas
    prestigiosas empresas del pais, como Cementera HOLCIM &nbsp;y
&nbsp;CANAL .<br />
    cada Adoquin y cada Bloque esta compuesto por:<br />
    <table style="width: 394px">
      <tr>
        <td style="font-weight: bold; width: 117px; color: white; font-style: italic;
height: 26px">
          Cemento</td>
        <td style="font-weight: bold; width: 105px; color: white; font-style: italic;
height: 26px">
          Material cero</td>
        <td style="height: 26px">
        </td>
      </tr>
      <tr>
        <td style="font-weight: bold; width: 117px; color: white; font-style: italic;
height: 26px">
          Material Selecto</td>
        <td style="font-weight: bold; width: 105px; color: white; font-style: italic;
height: 26px">
          Cemento</td>
        <td style="height: 9px">
```

```
</td>
</tr>
<tr>
  <td style="font-weight: bold; width: 117px; color: white; font-style: italic;
height: 26px">
    Agua</td>
  <td style="font-weight: bold; width: 105px; color: white; font-style: italic;
height: 26px">
</td>
<td>
</td>
</tr>
</table>
```

Estos materiales y las diferentes pocentaje que se necesitan para la elaboracion

de cada producto, hacen que estos sean mas resistentes.

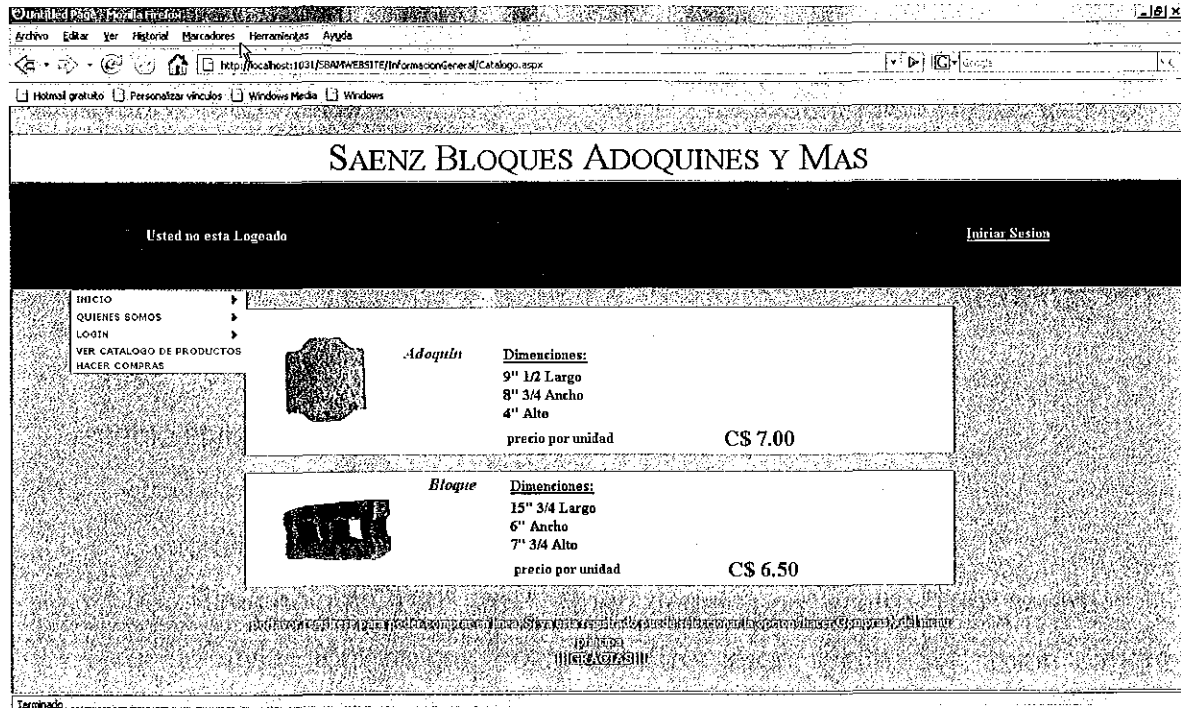
estos productos pasas por un proceso de secado y de mojado por 30 dias antes de

su venta, lo cual los hacen mas fuertes y duraderos permitiendo la calidad exigida

de los clientes para su posterior uso en la construccion, ya ea de calles urbanas

como edificaciones</div>

```
</asp:Content>
```



Esta es la Ventana de catalogo, aquí se mostraran los diferentes productos y sus precios

Codigo "Catalogo.aspx"

```
<%@ Page Language="VB" MasterPageFile = "~/MarcoPrincipal.master"
AutoEventWireup="false" CodeFile="Catalogo.aspx.vb"
Inherits="UsuarioRegistrado_Catalogo" %>
<asp:Content ID="ContenedorWeb"
ContentPlaceHolderID="ContenedordePaginas" Runat="Server">
<div>
<br>
<table style="width: 100%; background-color: yellow">
<tr>
<td align="center" rowspan="3" style="width: 3%">
</td>
<td style="font-weight: bold; font-size: large; width: 5%; color: dimgray;
font-style: italic;
height: 60px" valign="bottom">
Adoquin &nbsp; &nbsp;</td>
<td style="font-weight: bold; width: 19%; color: black; height: 60px; text-
decoration: underline"
valign="bottom">
Dimenciones:</td>
<td style="vertical-align: bottom; width: 20%; text-align: center">
<a href="RegistroClienteWeb.aspx"></a>
</td>
</tr>
<tr>
<td style="width: 5%; height: 58px">
</td>
<td style="font-weight: bold; width: 19%; color: black; height: 58px"
valign="top">
9" 1/2 Largo<br />
```



```
<td align="center" style="width: 20%" valign="bottom">
  <a href="RegistroClienteWeb.aspx"></a>
</td>
</tr>
<tr>
  <td style="width: 5%; height: 59px">
  </td>
  <td style="font-weight: bold; width: 19%; color: black" valign="top">
    15" 3/4 Largo<br />
    6" Ancho<br />
    7" 3/4 Alto</td>
  <td style="width: 20%; height: 59px">
  </td>
</tr>
<tr>
  <td style="width: 5%">
    &nbsp;</td>
  <td style="font-weight: bold; width: 19%; color: blue">
    &nbsp;&nbsp;&nbsp;precio por unidad
  </td>
  <td style="font-weight: bold; font-size: x-large; width: 20%; color: blue">
    C$ 6.50
  </td>
</tr>
</table>
<br />
<table style="width: 100%">
  <tr>
    <td align="center" style="width: 100%; color: white; height: 78px">
      porfavor registre para poder comprar en linea; Si ya esta registrado
      puede seleccionar
```

```
        la opcion "hacer Compras", del menu principa.<br />
        !!!GRACIAS!!!</td>
    </tr>
</table>
</div>
</asp:Content>
```

3. Pruebas

Se realizaron varias pruebas a lo largo del diseño del sistema y sitio web, para verificar que la codificación estuviera bien plantada y no hubiera ningún error. Estas pruebas son necesarias para poder encontrar y corregir los errores provocados por la mala codificación por parte de los desarrolladores del proyecto.

Se Encontraron varios errores minúsculos difíciles de encontrar, solo pudieron ser localizados con varias pruebas minuciosas

IX. Conclusiones

Terminado el prototipo de la aplicación Windows y finalizada la primer etapa del sitio Web de la Empresa Sáenz Bloques Adoquines y Mas., se concluye que el trabajo realizado será de gran beneficio, para la empresa como para nosotros en calidad de profesionales, ya que hemos adquirido experiencia y conocimientos en el diseño, desarrollo e implementación del la aplicación Windows y aplicación web

La aplicación Windows aun se encuentra en diseño ya que el tiempo planificado no fue el suficiente, pero creemos que hasta este punto hemos hecho un buen trabajo en el diseño y codificación de la aplicación.

La aplicación web se dividió en dos partes, la primera parte es la que hemos finalizado, esta servirá para darle una mejor imagen a la empresa desde el punto de vista publicitario, aquí mostramos una pequeña introducción de la empresa, y presentamos los diferentes productos con su precio. Finalizada la segunda parte se podrán hacer compra de los productos con tarjeta de crédito o debito desde el internet.

En este trabajo se hace mención de la segunda parte, además contamos con un adelanto en el diseño y codificación

X. Recomendaciones

Se le recomienda a la empresa Sáenz Bloques Adoquines y Mas..., crear un área de Informática, en la cual el personal encargado de esta área estará a cargo de la administración del sistema y la aplicación Web, o contratar los servicios de un Experto cuando la empresa lo solicite.

El personal de informática debe de tener conocimientos de Visual basic.net, ASPX, HTML y SQL

Se le debe de Dar mantenimiento preventivo y correctivo a los equipos cada 2 o 4 meses para el óptimo funcionamiento del sistema y el sitio Web

.Se deben de crear respaldo (Back Up) a la base de datos desde el SQL

XI.Bibliografía

Modulo de Ingeniería Web	(Ing. Fausto Quiñones)
Modulo de Aplicaciones Web	(Ing. Fausto Quiñones)
Modulo Intranet	(Ing. Irineo Moody)
Modulo Proyectos Informáticos	(Ing. Manuel Rojas)
Tutorial de SQL	(Claudio Casares)
El libro de Visual Basic 2005	(James Foxal, ANAYA Multimedia)

Líneas de código obtenidas desde el Internet:

www.aulaclick.com

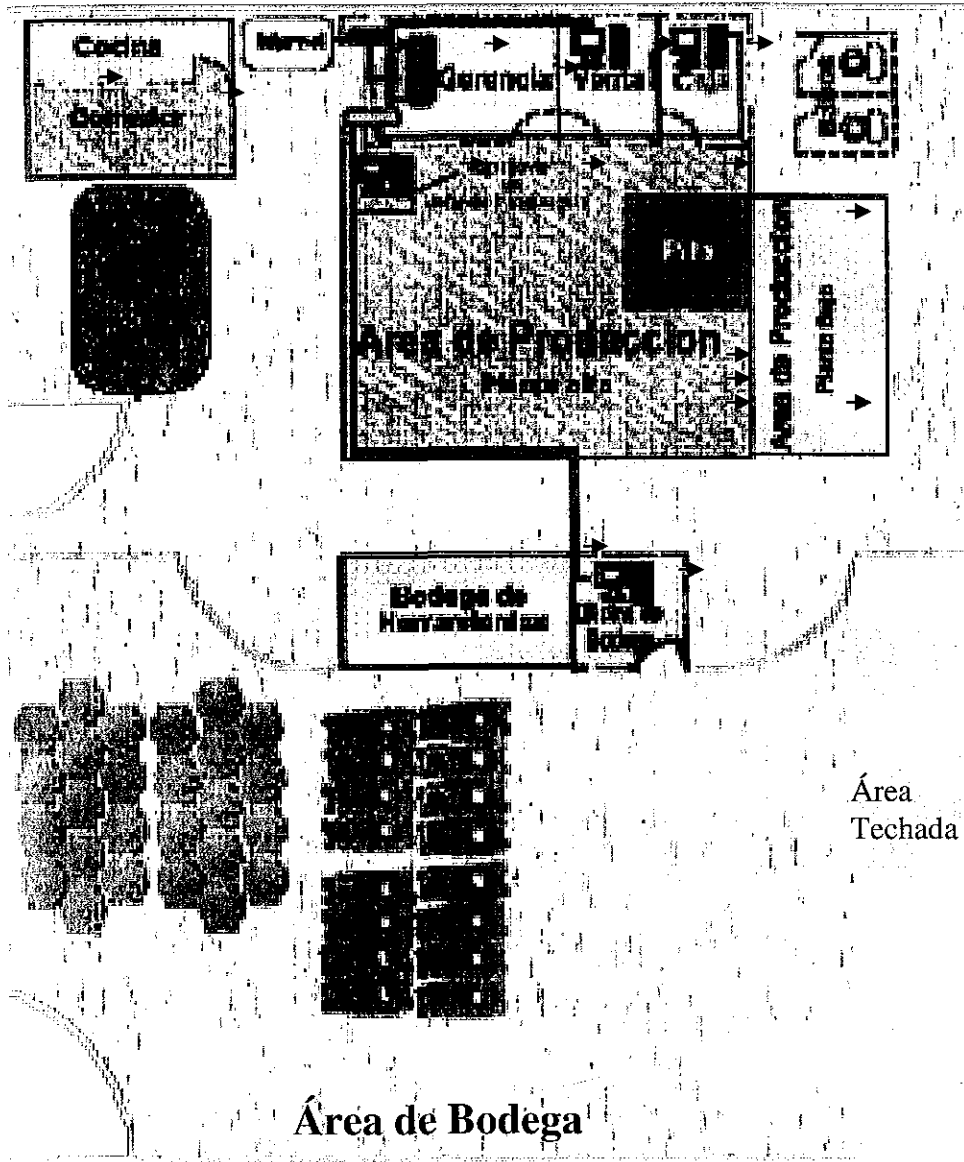
www.lawebdelprogramador.com

www.crystalab.com




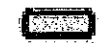











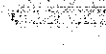

www.microsoftmsdn.com

XII. ANEXOS

Diagrama de la Organización de la Red y el sistema en la empresa Sáenz Bloques Adoquines y Mas...

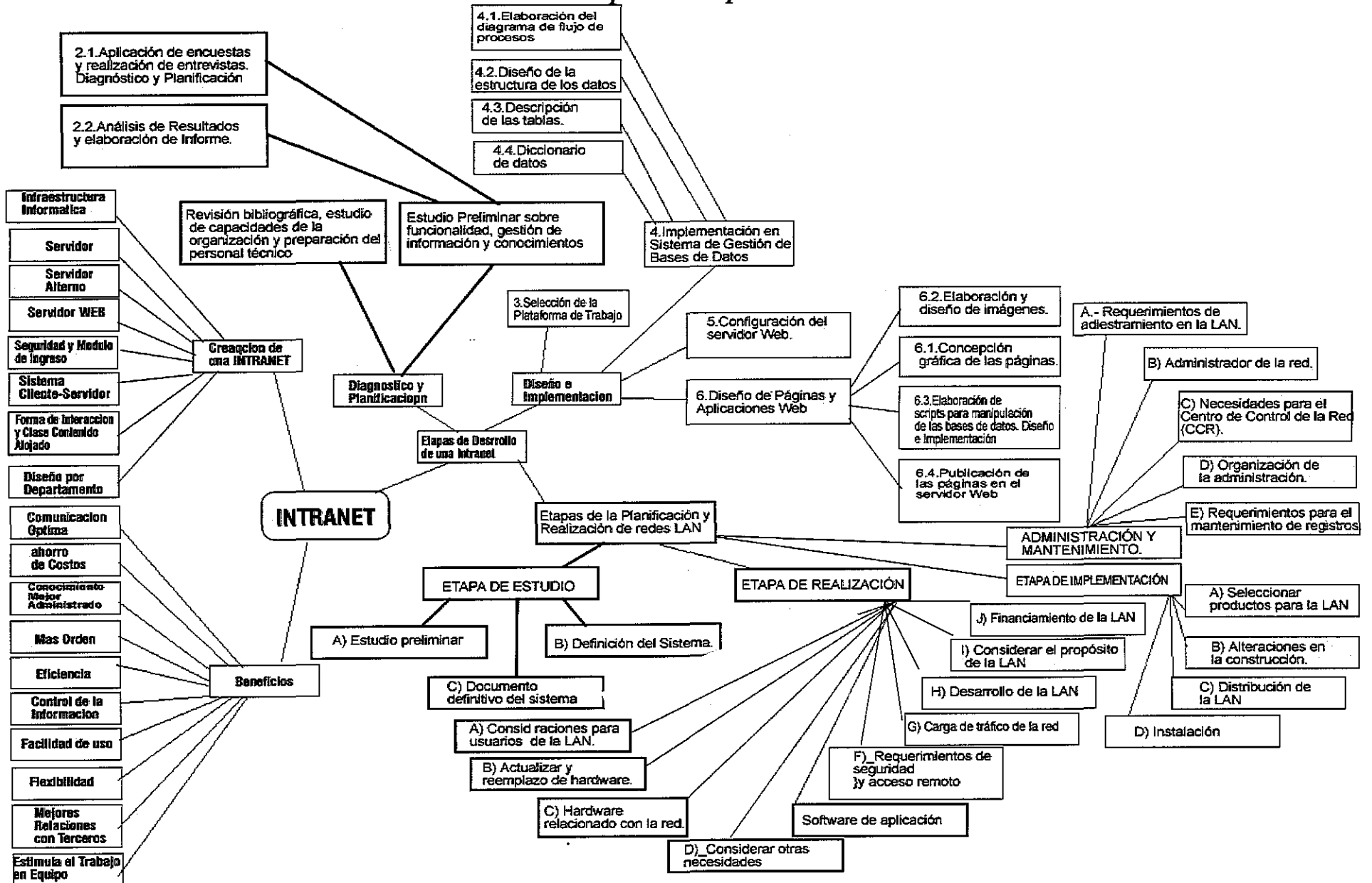


LEYENDA

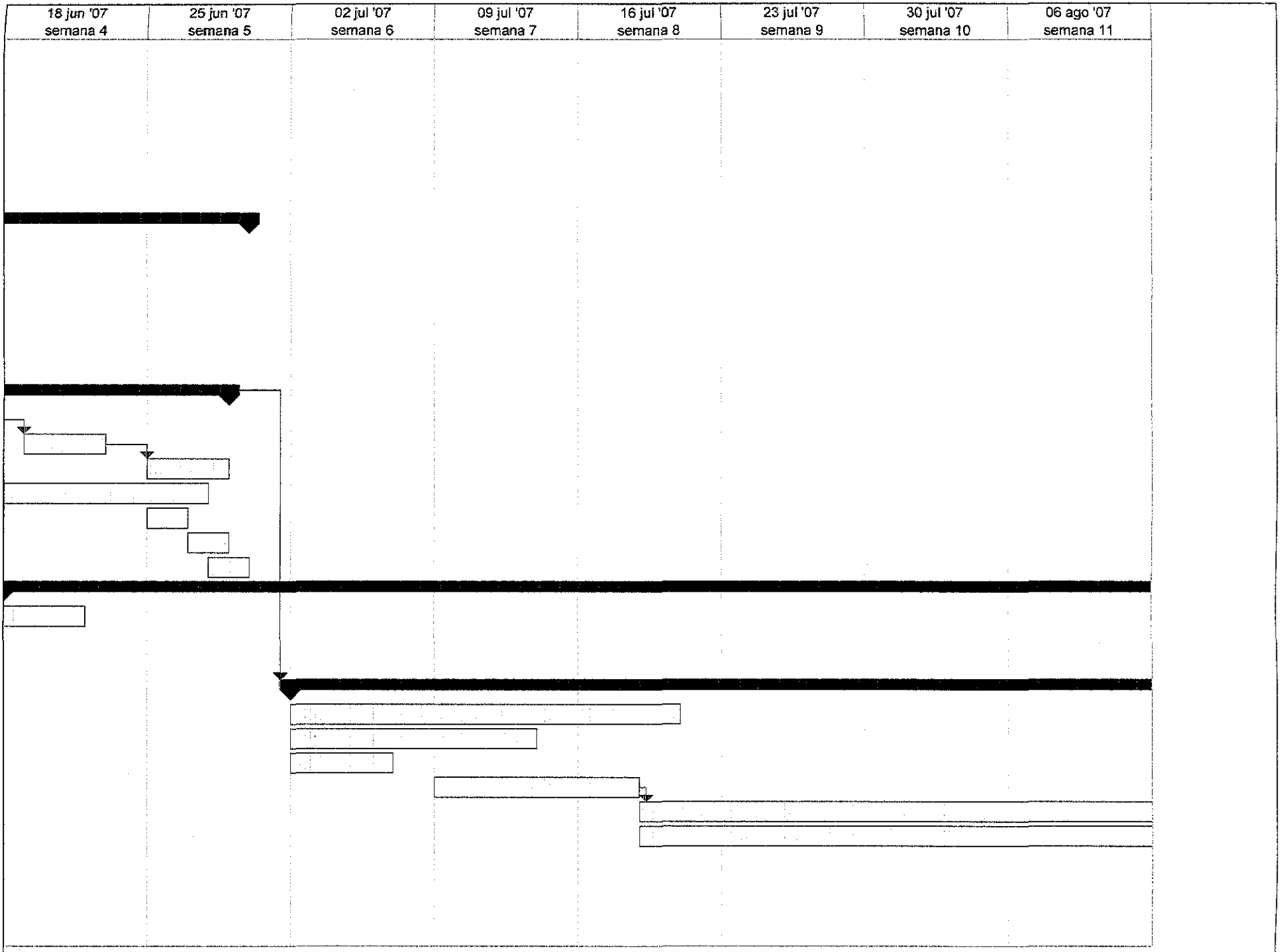
-  Servidor
-  Cliente
-  Modem
-  Switch 8-Ports
-  Cableado de Red
-  Internet
-  Escritorio del Jefe de Producción
-  Pipe de Agua
-  Servicios Higienicos
-  Parte externa del edificio
-  Parte interna del edificio
-  Granito-Patio
-  Ventanas
-  Adoquín
-  Bloque
-  Muro Externo
-  Panel

➔ Toma corriente 110W

Mapa Conceptual







Planificación Temporal



13 ago '07 semana 12	20 ago '07 semana 13	27 ago '07 semana 14	03 sep '07 semana 15	10 sep '07 semana 16	17 sep '07 semana 17	24 sep '07 semana 18	01 oct '07 semana 19	




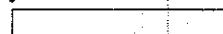
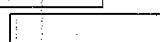
08 oct '07 semana 20	15 oct '07 semana 21	22 oct '07 semana 22	29 oct '07 semana 23	05 nov '07 semana 24	12 nov '07 semana 25	19 nov '07 semana 26	26 nov '07 semana 27

Id		Nombre de tarea	21 may '07 semana -1	28 may '07 semana 1	04 jun '07 semana 2	11 jun '07 semana 3
41		Capacitacion para el jefe de Produccion				
42		Capacitacion para el encargado de bodega				
34		Pruebas				
35		Prueba del sistema				
36		Prueba del Sitio Web				

18 jun '07 semana 4	25 jun '07 semana 5	02 jul '07 semana 6	09 jul '07 semana 7	16 jul '07 semana 8	23 jul '07 semana 9	30 jul '07 semana 10	06 ago '07 semana 11

13 ago '07 semana 12	20 ago '07 semana 13	27 ago '07 semana 14	03 sep '07 semana 15	10 sep '07 semana 16	17 sep '07 semana 17	24 sep '07 semana 18	01 oct '07 semana 19

--	--	--	--	--	--	--	--

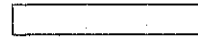
08 oct '07 semana 20	15 oct '07 semana 21	22 oct '07 semana 22	29 oct '07 semana 23	05 nov '07 semana 24	12 nov '07 semana 25	19 nov '07 semana 26	26 nov '07 semana 27
							
							

Proyecto: Planificación Temporal del
Fecha: mar 26/11/07

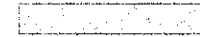
Tarea



Tarea resumida



Tareas externas



Progreso



Hilo resumido



Resumen del proyecto



Hito



Progreso resumido



Agrupar por síntesis



Resumen



División



Cotizaciones



Información general

Para registrar un dominio virtual haga lo siguiente:

1) Envíe por correo o FAX la siguiente documentación:

- Llenar el formato de Solicitud de Registro de Nuevo Dominio.
- Copia de Razón Social de la Empresa que solicita el dominio. O copia de Documento que lo acredite como apoderado LEGAL de la Empresa.
- Carta de Compromiso.

2) Pagar cuota ANUAL de USD \$ 50.00

Estos pagos puede hacerlos en córdobas o dólares a las cuentas a nombre de:

ADMINISTRACION NIC.NI

en cualquier sucursal de **BANCO UNO**.

Cuenta en Córdoba: No. 122002-11-6

Cuenta en Dólares : No. 397109-12-6

O bien en cheque a nombre de: **ADMINISTRACIÓN NIC.NI**

y depositarlos en nuestras oficinas, ubicadas en la Universidad Nacional de Ingeniería, 2do. Piso contiguo donde fue Rectoría.

PARA TODOS LOS PAGOS LES SOLICITAMOS ENVIARNOS UNA COPIA DEL RECIBO (escaneado) O TRANSFERENCIA BANCARIA A LOS CORREOS ELECTRONICOS **santos@nic.ni** y **marvin@nic.ni** O A NUESTRO TEL-FAX (+505) 278314



Web hosting

WEB HOSTING - UNICAMENTE HOSTING (no correo, no ftp, solo hosting)

Aprovecha nuestras ofertas de alojamiento::

PLAN BASICO

- 1. Espacio en disco 15 Mega**
- 2. Tasa de Transferencia 60 (enlace local)**
- 3. Actualización ON LINE.**
- 4. GRATIS!!**

PLAN INTERMEDIO

- 1. Espacio en disco 30 Mega**
- 2. Tasa de Transferencia 200**
- 3. Actualización ON-LINE**
- 4. USD 50.00 Anuales**

LLAMANOS PARA OBTENER MAS INFORMACIÓN AL 278 3141

MASTERTEC

TECNOLOGIA DEL FUTURO

Barrio Lanteynaga de la Iglesia La Merced To al sur, 10 varas abajo,
Managua Nic P.R.K. (505) 2507193, 2507194 Fax: 2491567

No. MV-1007

COTIZACIÓN

Cliente

Fecha: _____
 Empresa: **SIBECÓN**
 Atención: **JESI BODEN**
 Teléfono: **8228434** Fax: **2801720**

Vendedor **MARYALLY VALLE**
 E-mail: **mvalle@mastertec.com.ni**
 Teléfono: **2507193**
 Celular: **6847300**

Cant.	DESCRIPCION	PRECIO UNITARIO	TOTAL
3	IMPRESORA HP LASERJET P2015 1200 DPI 377PM LETTER 26PPM A4	\$318.00	\$954.00
1	MARCA EPSON MOD LX-300-II NUMERO DE PARTE C11C640011 TECNOLOGIA DE IMPRESION IMPACTO (MATRIZ DE PUNTOS 9 PINES) VELOCIDAD DE IMPRESION 337 CPS ANCHURA DE CARRO 80 COLUMNAS CONECTIVIDAD PARALELO USB 1.1/ SERIE (RS-232C BUFFER 64KB LENGUAJE DE IMPRESION EPSON ESC/P, IBM 2380 PLUS PLATAFORMA DE TRABAJO WINDOWS 95 WINDOWS 2000 WINDOWS ME WINDOWS XP MEDIOS SOPORTADOS PAPEL CONTINUO NO DE COPIAS : ORIGINAL +4 SUMINISTRO DE IMPRESION CINTA COLOR NEGRO NIVEL DE RUIDO APPROX.49 DB (A) CONTENIDO IMPRESORA MANUAL CABLE DE PODER DRIVER (CD) CINTA DE COLOR NEGRO CABLE USB	\$210.00	\$210.00

Detalles de Pago

Credito
 EFECTIVO
 CHEQUE

T/Entrega: **INMEDIATA**
 Garantía: **1 AÑO**
 Of./Valida: **16 DIA HABILES**

Descuento	\$1,164.00
Sub-Total	
L.V.A	\$1,164.00
TOTAL	\$174.60
	\$1,338.60



Firma _____
MARYALLY VALLE

Preco Sujeto a Cambio

DE : 3C_CONICO

NO. DE FAX : 2703860

22 NOV. 2007 10:39AM P1



FACTURA PROFORMA

Numero : 69318

Fecha : 22/11/2007

Cliente : 08688 CIDECOIN-270-1720

Vendedor : ELMER SANCHEZ

Zona : TODAS

Condiciones : Contado / EFECTIVO / 0 Dias

Orden Compra :

Notas :

8

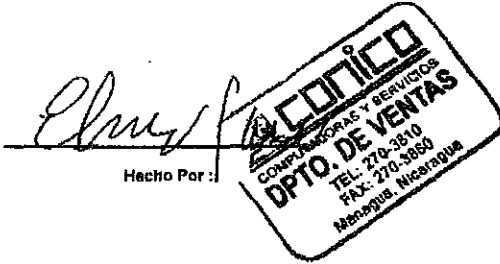
Usuario : VENTAS

Digitado : 22/11/2007 10:05:12

Linea	Producto	Cantidad	Precio	%Dec	BRUTO
1	US0001 CABLES USB	4.00	20.01	5.00	80.06
2	IMP04293 IMPRESORAS EPSON MATRIZ LX-300+	1.00	4,479.40	5.00	4,479.40
3	IMH4409 IMPRESORA LASER HP 1018	3.00	2,191.81	5.00	6,575.43
Totales :		8.00			11,134.89

TOTALES

Bruto :	11,134.89
Descuento :	556.74
Impuesto Ventas :	1,586.72
Otro Impuesto :	0.00
Transporte :	0.00
Neto :	12,184.87



Hecho Por :

Este documento no tiene ningun valor comercial.
Precios sujetos a cambio sin previo aviso.

La entrega se hara segun existencia al momento de efectuarse la venta.



DATOS DEL CLIENTE

Empresa : CIBECOIN

Tel-Fax : 270-1720

DATOS DE LA PROFORMA

Nº Ref.: MAGDA-171105

Fecha : 22 de Noviembre de 2007

Asunto: Presentación de oferta

COMTECH tiene el gusto de ofertarle, los accesorios y equipos de computación con las especificaciones técnicas solicitadas a precios económicos.

03	<p>Impresora láser Marca HEWLETPACKARD Modelo LJ 1020 Velocidad de impresión de 15 ppm Procesador RISC a 234MHZ 2MB de memoria Lenguaje de impresión HP PCL 5e Calidad de tipo 1.200 ppp (600 ppp con REt y FastRes) volumen mensual de impresión de 7,000 paginas 1 MB de memoria RAM-32KB de memoria ROM puerto USB y paralelo Incluye toner Q2612A y cable</p>	\$130.00	\$390.00
01	<p>Impresor Matricial Marca EPSON Modelo LX 300+II Referencia IMPM EP LX-300 + 9 (45) Descripción del producto Epson LX 300+II - impresora - B/W - matriz de puntos Tipo de impresora de grupo de trabajo Matriz de puntos - 9 espiga - monocromo Dimensiones (Ancho x Profundidad x Altura) 36.6 cm x 27.5 cm x 15.9 cm Peso 4.4 kg Tamaño máx. Soporte (estándar) Legal, Rollo (25,4 cm x 55,9 cm) Tamaño máximo de soporte (personalizado) 254 mm x 559 mm Velocidad de impresión Hasta 337 caracteres/s - borrador rápido Hasta 300 caracteres/s - borrador rápido Hasta 225 caracteres/s - borrador Hasta 270 caracteres/s - borrador Hasta 225 caracteres/s - condensado Hasta 56 caracteres/s - Letra Hasta 67 caracteres/s - Letra Resolución máxima (B/N) 240 ppp x 144 ppp Interfaz Paralelo, serial, USB RAM instalada (máx.) 8 KB Códigos de barras Código 39, EAN/JAN-8, EAN/JAN-13, UPC-A, UPC-E, Code</p>	\$196.00	\$196.00



Telefax: 2787883
 Telefono: 2526004



CLIENTE : ENNIO SAENZ
 ATENCIÓN: EL MISMO
 TELEFAX :



FECHA: 31/10/07
 VENDEDOR: INDIRA ESPINOZA
 EMAIL: ajein53@hotmail.com

COTIZACION

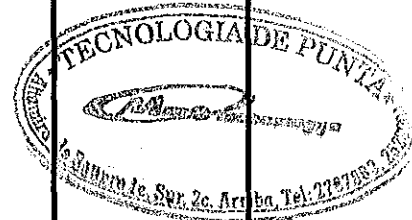
CEL.: 6791675

Estimado (s) Señor (es):

Es una enorme satisfacción y un privilegio que nuestra empresa pueda cotizar a usted (es) nuestros productos, esperamos que nuestros precios puedan llenar plenamente sus expectativas.

A continuación relacionamos los productos y precios con sus respectivas características

CANTIDAD	DESCRIPCION	PRECIO UNIT.	PRECIO TOTAL
	ACCESORIOS		
1	IMPRESORA MATRICIAL LX-300+II EPSON	\$213,04	\$213,04
2	IMPRESORA LASSERJET HP 1018		\$117,39
3	UPS DE 500VA FORZA CON REGULADOR	\$43,47	\$130,41
	**** **		
NUESTROS PRECIOS TIENDEN A VARIAR NO DUDE EN LLAMAR Y CONSULTARLO CON SU EJECUTIVO DE VENTAS. EL CK ES A NOMBRE DE MICROTECHNOLOGYS Y/O JUAN CARLOS NOVOA		SUB TOTAL	\$460,84
		15% IVA	\$ 69,13
		TOTAL	\$ 529,97



* TODOS NUESTROS EQUIPOS Y ACCESORIOS CUENTAN CON GARANTIA.

* OFRECEMOS SERVICIOS DE MANTENIMIENTO Y REDES.

***** **GRACIAS POR SU PREFERENCIA** *****



TOSHIBA





Telefax: 2787883
 Telefono: 2526004



CLIENTE : ENNIO SAENZ
 ATENCIÓN: EL MISMO
 TELEFAX :



FECHA: 31/10/07
 VENDEDOR: INDIRA ESPINOZA
 EMAIL: ajein53@hotmail.com

COTIZACION

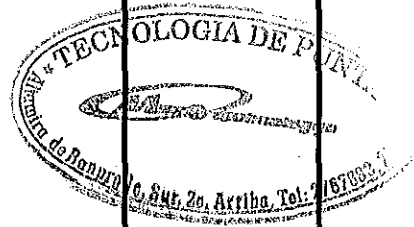
CEL.: 6791675

Estimado (s) Señor (es):

Es una enorme satisfacción y un privilegio que nuestra empresa pueda cotizar a usted (es) nuestros productos, esperamos que nuestros precios puedan llenar plenamente sus expectativas.

A continuación relacionamos los productos y precios con sus respectivas características

CANTIDAD	DESCRIPCION	PRECIO UNIT.	PRECIO TOTAL
	ACCESORIOS		
1	CAJA DE CABLE RJ45 UTP CAT. 5 305MTS	\$85,21	\$85,21
18	CONECTOR RJ45	\$0,25	\$4,50
1	SWITCH DE 8 PUEROTS 10/100 ENCORE	\$21,73	\$21,73
	**** **		
NUESTROS PRECIOS TIENDEN A VARIAR NO DUDE EN LLAMAR Y CONSULTARLO CON SU EJECUTIVO DE VENTAS. EL CK ES A NOMBRE DE MICROTECHNOLOGYS Y/O JUAN CARLOS NOVOA		SUB TOTAL	\$111,44
		15% IVA	\$ 16,72
		TOTAL	\$ 128,16



* TODOS NUESTROS EQUIPOS Y ACCESORIOS CUENTAN CON GARANTIA.

* OFRECEMOS SERVICIOS DE MANTENIMIENTO Y REDES.

***** GRACIAS POR SU PREFERENCIA *****



TOSHIBA





Telefax: 2787883
 Telefono: 2526004

CLIENTE : ENNIO SAENZ
 ATENCIÓN: EL MISMO
 TELEFAX :



FECHA: 31/10/07
 VENDEDOR: INDIRA ESPINOZA
 EMAIL: ajein53@hotmail.com



COTIZACION

CEL.: 6791675

Estimado (s) Señor (es):

Es una enorme satisfacción y un privilegio que nuestra empresa pueda cotizar a usted (es) nuestros productos, esperamos que nuestros precios puedan llenar plenamente sus expectativas.

A continuación relacionamos los productos y precios con sus respectivas características

CANTIDAD	DESCRIPCION	PRECIO UNIT.	PRECIO TOTAL
1	COMPUTADOR INTEL CORE 2 DUO MODELO: CASE ATX DE 450W PROC. INTEL CORE 2 DUO 2,0GHZ SOCK 775 MEMORIA DDR2 1GB PC 533 TARJETA MADRE ASROCK 775 DISCO DURO 160GB RPMS 7200 REV. SONIDO INTEGRADO CD ROM 52X INTERNO FLOPY DRIVE 1.44MB TECLADO, MOUSE OPTICO, PAD MOUSE PARLANTES 240W MONITOR DE 17" CRT SVGA *** 1 AÑO DE GARANTIA ***	\$503,04	\$503,04
NUESTROS PRECIOS TIENDEN A VARIAR NO DUDE EN LLAMAR Y CONSULTARLO CON SU EJECUTIVO DE VENTAS. EL CK ES A NOMBRE DE MICROTECNOLOGYS Y/O JUAN CARLOS NOVOA		SUB TOTAL	\$503,04
		15% IVA	\$ 75,46
		TOTAL	\$ 578,50



- * TODOS NUESTROS EQUIPOS Y ACCESORIOS CUENTAN CON GARANTIA.
- * OFRECEMOS SERVICIOS DE MANTENIMIENTO Y REDES.

***** GRACIAS POR SU PREFERENCIA *****



TOSHIBA





Telefax: 2787883
 Telefono: 2526004

CLIENTE : ENNIO SAENZ
 ATENCIÓN: EL MISMO
 TELEFAX :



FECHA: 05/06/07
 VENDEDOR: INDIRA ESPINOZA
 EMAIL: ajein53@hotmail.com



COTIZACION

CEL: 6791675

Estimado (s) Señor (es):

Es una enorme satisfacción y un privilegio que nuestra empresa pueda cotizar a usted (es) nuestros productos, esperamos que nuestros precios puedan llenar plenamente sus expectativas.

A continuación relacionamos los productos y precios con sus respectivas características

CANTIDAD	DESCRIPCION	PRECIO UNIT.	PRECIO TOTAL
4	COMPUTADOR INTEL CELERON MODELO: CASE ATX DE 450W PROC. INTEL CEL. 3.0GHZ SOCKET 775 MEMORIA DDR2 512MB PC 533 TARJETA MADRE ASROCK 775 DISCO DURO 80GB RPMS 7200 REV. SONIDO INTEGRADO CD ROM 52X INTERNO FLOPY DRIVE 1.44MB TECLADO, MOUSE OPTICO, PAD MOUSE PARLANTES 240W MONITOR DE 17" CRT SVGA *** 1 AÑO DE GARANTIA ***	\$363,04	\$1.452,16
NUESTROS PRECIOS TIENDEN A VARIAR NO DUDE EN LLAMAR Y CONSULTARLO CON SU EJECUTIVO DE VENTAS. EL CK ES A NOMBRE DE MICROTECHNOLOGYS Y/O JUAN CARLOS NOVOA		SUB TOTAL	\$1.452,16
		15% IVA	\$ 217,82
		TOTAL	\$ 1.669,98



- * TODOS NUESTROS EQUIPOS Y ACCESORIOS CUENTAN CON GARANTIA.
- * OFRECEMOS SERVICIOS DE MANTENIMIENTO Y REDES.

***** GRACIAS POR SU PREFERENCIA *****



TOSHIBA





OFERTA DE SERVICIOS DE CONECTIVIDAD

CLIENTE: CIDE COIN	FECHA: 21 Nov 2007
Atencion: Jessy Boden	CONTRATO: 10 días
Tel : 2701720	

CONCEPTO : ICABLE PYME

Ítem	Frecuencia del Cargo	Descripción	Cantidad	Precio Unitario US \$	Precio Total US \$
1	OTC	Activacion y Configuracion Icable	1	\$64,99	\$64,99
2	MRC	Velocidad de 128 Kbps Local Loops	1	\$49,99	\$49,99
	MRC	Velocidad de 256 kbps Local Loops	1	\$69,99	\$69,99
	MRC	Velocidad de 512 kbps Local Loops	1	\$99,99	\$99,99

TERMINOS Y CONDICIONES

Validez de la oferta: 10 días

- 1) Oferta Incluye Cable Modem
- 2) Incluye Cuenta de Correo Electronico de IBW
- 3) Ancho de Banda 4 a 1 Compartido
- 4) Costo de activacion Estesa 400 cordoba

Subtotal Cargo Unico	\$64.99
IVA	\$9.74
Total	\$74.73

Firma del Ejecutivo:

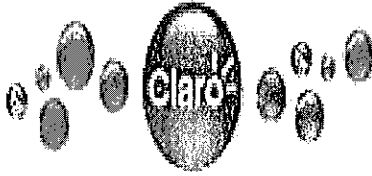
Lic Gabriela Flores Centeno
 gabriela.flores@ibw.com.ni
 Tel: 2786328 Ext: 4218 Cel: 8865617

21 e Nov 2007

Fecha

Aceptado por EL CLIENTE:

Fecha



PROFORMA DE SERVICIO TURBONETT INALAMBRICO

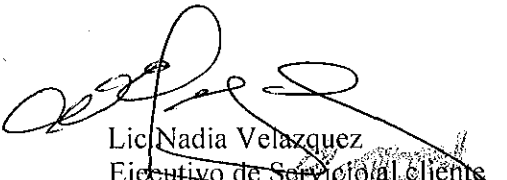
MODALIDAD	VELOCIDAD	PRECIO
Turbonett Residencial	128K	U\$19.99
Turbonett Residencial	256K	U\$29.99
Turbonett Residencial	512K	U\$39.99

Precios no incluyen IVA

COSTO DE SUSCRIPCION: US49.99 + IVA

Instalaciones mayores a 6 metros de altura que requieran Torre: tramo de cada 10 pies \$60.87 + IVA

Atentamente,


Lic. Nadia Velazquez
Ejecutivo de Servicio al Cliente
CAC BELLO HORIZONTE
Bello Horizonte

CC: Archivo

SILVA INTERNACIONAL, S.A
COTIZACION

Email: radial@sinsa.com.ni Tel.: 2782229 Fax: 2707919
Ciudad: MANAGUA Direccion: ROT. CRISTO REY 200m SUR

Documento: 000034787 Tienda: 15-RADIAL SANTO DOMINGO Fecha: 21-11-2007
Vendedor : 001017-FELIX BARBOZA
Cliente : 000000-- CYDECOIN. Telefono: 270-1720 Carnet:
Direccion:
Moneda : CORDOBAS (C\$) Condicion : CONTADO Dias de validez: 01

CODIGO	DESCRIPCION	U/M	CANT.	PRECIO	TOTAL
4272030500	GENERADOR ELECT. 2500W 5HP 120VAC	UND	1.00	16522.40	16,522.40
2270003000	&TUBO COND.1/2"X10 PVC CH ELECT_CH	UND	1.00	13.30	13.30
2205005000	&CURVA COND.1/2"X90° PVC CH ELECT_C	UND	1.00	2.66	2.66

Sub-Total C\$: 16,538.36
Impuesto C\$: 2,480.76
Total C\$: 19,019.12
Equiv. en US\$: 1,001.01
Factor de Cambio: 19.00

Observaciones:

Firma: Felix Barbosa Perez

Dpto. de ventas. Sinsa Radial Santo Domingo.

Tel: 278-22-29 Ext. 133. Cel: 806-9580

* Atencion Jessy Boden



Nota :No se aceptan cambios una vez aprobada la oferta, que fue hecha con base a datos _
suministrados. Los precios estan sujetos a cambio sin previo aviso.

Usuario: ventas1503 Fecha: 2007-11-21 Hora: 12:23:39

SILVA INTERNACIONAL, S.A
COTIZACION

Email: radial@sinsa.com.ni Tel.: 2782229 Fax: 2707919
Ciudad: MANAGUA Direccion: ROT. CRISTO REY 200m SUR

Documento: 34787 Tienda: 15-RADIAL SANTO DOMINGO Fecha: 21-11-2007
Vendedor : 1017-FELIX BARBOZA
Cliente : -- CYDECOIN. Telefono: 270-1720 Carnet:
Direccion: MANAGUA
Moneda : CORDOBAS (C\$) Condicion : CONTADO Dias de validez: 1

CODIGO	DESCRIPCION	U/M	CANT.	PRECIO	TOTAL
4272030500	GENERADOR ELECT. 2500W 5HP 120VAC	UND	1.00	16522.40	16,522.40
2270003000	&TUBO COND.1/2"X10 PVC CH ELECT_CH	UND	1.00	13.30	13.30
2205005000	&CURVA COND.1/2"X90° PVC CH ELECT_C	UND	1.00	2.66	2.66

Sub-Total C\$: 16,538.36
Impuesto C\$: 2,480.76
Total C\$: 19,019.12
Equiv. en US\$: 1,001.01
Factor de Cambio: 19.00

Observaciones:

Firma:

Felix Mauricio Barboza Perez. cel 806-9580.
** Ventas. Sinea. Radial. tel: 278-2229 Ext. 133.*



Atencioñ Jessy. Bodea

Nota: No se aceptan cambios una vez aprobada la oferta, que fue hecha con base a datos suministrados, Los precios estan sujetos a cambio sin previo aviso.

Usuario: ventas1503 Fecha: 2007-11-21 Hora: 10:34:34

FERRETERIA LUGO S.A.

KM 4 ¼ Carretera Masaya.
Teléfonos 287-0944, 287-5803, 270-9782 Fax 277-3913 e-Mail managua@ferreteriaLugo.com.ni
R.U.C No. 220870-8805 Apartado No. 1684

Cliente **SIBECOM (MOSTR)**
Ciudad **-FAX 2701720**
Fecha **21/11/2007**

PROFORMA **00-0000099120**
Vendedor
Almacen
Cajero Cajero GERALD

PROFORMA VALIDA POR 3 DIAS A PARTIR DE LA FECHA DE ELABORACION

Código	Descripción	Cantidad	Unidad Medida	Precio Venta	Sub - Total	Descuento	TOTAL
2000821	Generador Electrico Porter Cable 2500W(HONDA) 8HP	1.00	PZA/1	18,826.00	18,826.00	1,682.60	18,143.

Fecha de Impresión 21/11/2007 2:20:00PM

Sub - Total	18,826.00
Descuento	1,682.60
IGV	2,271.51
Retención	0.00
TOTAL	17,414.91

*** ESTIMADO CLIENTE ESTAMOS SUJETOS A RETENCIONES DEL 2 % ***

1 FERRETERIA JENNY CENTRAL

TODO EN FERRETERIA

MATERIALES DE CONSTRUCCION

TRANSPORTE GRATIS (MANAGUA)

PROFORMA

0000027088

DIA	MES	AÑO
21	Nov	2007

Fecha : 21/11/2007

Cliente : CIBE COIN

Teléfono : 2701720

FAX :

Dirección :

Contacto :

Vendedor : 14 ORLANDO FONSECA

Observaciones :

Código	Descripción	Cantidad	Precio Unit.	Subtotal
015180001	GENERADOR 3800W CUTLER HAMMER 30208 +	1.00	13,500.0000	13,500.00
ATENDIDO POR <u>Orlando F</u> TEL. <u>2578054</u>				Subtotal C\$: 13,500.00
ELABORAR CHEQUE A NOMBRE D "FERRETERIA JENNY S.A"				Impuesto C\$: 2,025.00
SOMOS EXENTOS DEL 2%				Total Neto C\$: 15,525.00
ENTRADA DEL HOSPITAL DEL NIÑO LA MASCOTA 75 VRS. AF				
TEL. 2480783 / 2490789 - 2530822 - 2482890 - 2400548 - 2530817 FAX # 2531503				
ESTOS PRECIOS PUEDEN VARIAR EN CUALQUIER MOMENTO SIN PREVIO AVISO				

Instrumentos

Entrevista

1. ¿Cuántos trabajadores tiene actualmente y en que área?

Actualmente la empresa tiene 16 trabajadores:

Gerencia:

Un gerente

Área de Ventas:

Dos vendedores

Un visitante

Un de planta

Contabilidad:

Una cajera

Área de Producción:

Un encargado de bodega

Un jefe de Producción

Dos Operarios

Un conductor

Un cuidador

Una de Limpieza

Ocho trabajadores Jornaleros

2. ¿Cómo son los procesos internos de su empresa?

Cuando un cliente nos hace un pedido, Primero el vendedor verifica con el encargado de bodega la existencia de producto final disponible.

Si no hay el encargado de bodega informa al vendedor cuanto se esta produciendo en el momento ya que el encargado de bodega maneja tanto las entradas y salidas de materia prima como de producto final,. El bodeguero debe garantizar siempre en el inventario exista un stock tanto de materia prima como de producto final.

Es decir, cuando el encargado de bodega observa que se está agotando la materia prima este debe informarle al jefe de producción las necesidades para que las compras de materia prima se hagan efectivas, el jefe de producción avisa al gerente quien se encarga de hacer los pedidos a los proveedores.

El jefe de producción controla y garantiza la producción del producto final en cuanto a la calidad requerida por el MTI y evitar los desperdicios de materia prima.

Cuando se termina d elaborar un bloque o adoquín, este inmediatamente es almacenado en bodega.

Luego que el vendedor verificar la disponibilidad, este informara al cliente la disponibilidad del producto y si se puede cubrir la cantidad solicitada por el mismo, de ser así el vendedor le hace una orden de entrega, después el cliente con su orden pasa a caja para la elaboración de su correspondiente factura

El cliente puede retirar su compra en ese mismo momento o indicarnos la fecha y dirección para la entrega de su compra ya que tenemos servicio a domicilio, claro, dependiendo de la cantidad Comprada por el cliente, si la cantidad supera las 50 unidades

El cliente tiene que llegar con una orden de compra en mano con la cantidad a entregar y el producto para que el encargado de bodega pueda entregar el producto al cliente, en el caso de mandárselo a dejar, el encargado de bodega le avisa al Jefe de producción que hay una entrega este verifica en caja, si todo está en orden el jefe de producción le da la orden de salida a bodega y bodeguero entrega el pedido al conductor del camión para que este efectúe la debida entrega.

El conductor deberá llevar una orden de entrega para que el cliente la firme de recibido, Todo lo llevamos por escrito, tenemos un archivero donde guardamos los recibos, facturas de compra y ventas, los informes de compra, informes de inventario de materia prima y producto final, con su respectivo fólder

3. ¿Con cuanto equipo cuenta para la elaboración del producto final?

Por el momento contamos con:

Dos mezcladoras de Cemento

Dos máquinas para fabricar bloques con 4 moldes

Dos máquinas para fabricar adoquines con 6 moldes

4. ¿Qué productos son los que elabora?

Adoquines y Bloques, pero estamos pensando en la posibilidad de producir nuevos productos como: Pisos, azulejos, lavadero, adoquines de lujo, piedra cantera y otros.

5. ¿Qué tipo de materia prima se necesitan para la elaboración del producto final?

La materia prima que usamos para la elaboración de los productos son:

Piedrín

Arena

Cemento

Piedra poma

Material Cero

Agua

6. ¿Su personal tiene algún conocimiento para el uso de un computador?

Yo creo que la mayoría si tiene conocimientos básicos como por ejemplo:

Encender la maquina

Apagar la maquina

Procesar textos

7. ¿Tiene algún equipo de cómputo en su empresa?

No

8. ¿Le gustaría tener un sistema de cómputo en su empresa? ¿porque?

Si, ya que Hace seis meses la producción ha aumentado en un 50 %, por lo que ahora nuestros clientes no son solo personas naturales y pequeñas empresas sino que también empresas constructoras y entidades del gobierno como las alcaldías y esto ha hecho que exista un descontrol en la producción, ay veces que no contamos con la suficiente materia prima para poder producir ya que se nos agota rápidamente por los pedidos grandes que hacen las constructoras, también existen desperdicios de materia prima, y Con un sistema Informático, creo que se Agilizarían los procesos y se llevaría un mejor control de materia prima y Producto final

9. ¿Que le gustaría que el sistema hiciera?

Que me lleve un control de las ventas y compras, un control del inventario de materia prima y de producto final, donde pueda ver cuánto exactamente hay en existencia, y poder visualizarlas y poder tener copias de ello

También me gustaría que agilizara las ventas y entregas de producto final para poder tener un buen servicio al cliente.

Imprimir las facturas de venta y reportes

10. ¿Le gustaría poder hacer ventas desde el internet? ¿porque?

Claro que me gustaría, de esta manera estoy dando una imagen de modernización, además ahora la gente para todo utiliza el Internet, además me permitiría capturar más clientes

11. ¿Cómo le gustaría que fuera su sitio web?

Me gustaría poder visualizar la misión, visión y objetivos de la empresa, una pequeña propaganda, un catalogo donde el cliente pueda ver nuestros productos y precios para poder hacer la compra por internet ya que sin el precio no se podría hacer la compra. Independientemente que la competencia pueda verlos ya que Estamos compitiendo en calidad no en precio.

También me gustaría visualizar e imprimir reportes de las ventas realizadas por internet

CORRESPONDENCIA

Carta de Presentación

Lic. Ligia Martínez Ovando

Gerente

Empresa:

Sáenz Bloques y Adoquines

Por este nos presentamos ante usted como Ingenieros en Sistema Egresados de la Universidad de Ciencias Comerciales (U.C.C.), y al mismo tiempo estamos ofreciéndole nuestro servicio como tal, dentro de los cuales podemos ofertarle lo siguiente:

- *Elaboración de un Sistema Informático para agilizar sus operaciones*
- *Elaboración de sitio Web Dinámico y estático.*
- *Instalación de Red.*
- *Capacitación del Personal para el manejo del Sistema*

Todo esto es posible con la ayuda de nuestro equipo, con el cual los servicios mencionados anteriormente se podrán elaborar con eficacia.

Sin otro particular y esperando poder brindarle Nuestro servicio nos despedimos

Ing. Ennio Sáenz

Ing. Karime Hurtado

Ing. Sharity Brown

Ing. Yessy Bodden