

Capítulo 4 Diseño e implementación

4.1 Diseño

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. El objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente [Pressman, 1998].

Hay tres características que sirven como parámetros generales para la evaluación de un buen diseño [McGlaughlin, 1991]. Estos parámetros son los siguientes:

1. El diseño debe implementar todos los requisitos explícitos obtenidos en la etapa de análisis.
2. El diseño debe ser una guía que puedan leer y entender los que construyen el código y los que prueban y mantienen el software.
3. El diseño debe proporcionar una idea completa de lo que es el software.

El diseño es la primera de las tres actividades técnicas que implica un proceso de ingeniería de software; estas etapas son diseño, codificación (en el caso de este proyecto Desarrollo e Implementación) y pruebas [Pressman, 1998]. Generalmente la fase de diseño produce:

- **Diseño de datos** esencialmente se encarga de transformar el modelo de dominio de la información creado durante el análisis. Para el tipo de datos manejado en este proyecto es de suma importancia este diseño, ya que de él dependerá el buen funcionamiento de la aplicación. Éste diseño está basado en el esquema entidad-relación obtenido en análisis, del cual se originan las tablas que serán utilizadas por ésta herramienta computacional.
- **Diseño arquitectónico** se definen las relaciones entre los principales elementos estructurales del programa. Para un software empresarial, como el que se va a desarrollar e implementar, es importante tomar en cuenta esta fase de diseño, dado

que en esta representación se establece la estructura modular del software que se desarrolla, tipo de arquitectura, tipo de conexión entre la aplicación y el DBMS, el manejo de reportes, entre otros. La figura 4.1 muestra la arquitectura del sistema.

- **Diseño de interfaz** describe cómo se comunica el software con los sistemas que operan con él, y con los operadores que lo emplean. En el caso de la herramienta de software propuesta la comunicación que se genera con los operadores es a través de ambientes gráficos que llevan de la mano al usuario para que pueda realizar sus tareas cotidianas. La comunicación con otros sistemas, DBMS y Crystal Reports, se realiza a través de mecanismos de acceso a base de datos y componentes **VCL** (Visual Component Library).
- **Diseño procedural** transforma elementos estructurales de la arquitectura del programa en una descripción procedural de los componentes del software. Para el desarrollo de esta aplicación es importante éste diseño, debido a que del análisis previo se obtuvieron los módulos a desarrollar y esta parte definirá los pasos a seguir en cada uno de los procesos y funciones que formarán cada módulo, así como los procesos necesarios para conectar la aplicación con el DBMS y Crystal Reports.

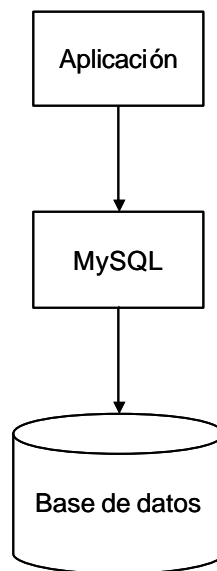


Figura 4.1 Esquema general

En el diseño de estructura y comportamiento de objetos los siguientes componentes son definidos:

- ¿Qué clases serán implementadas? Los tipos de objetos de AEO guiarán esta decisión.
- ¿Qué estructura de datos buscará una clase? La estructura de datos puede ser representada a través de un diagrama.
- ¿Qué operaciones ofrecerá cada clase y cuáles serían sus métodos? Las operaciones son listadas y eventualmente sus métodos son especificados.
- ¿Cómo serán implementadas las clases con herencia y cómo cambiarán los datos y las especificaciones de las operaciones?
- ¿Qué variantes existen? Se identifican variantes entre clases (“Igual que, excepto”)

En el apéndice C se encontrarán los diagramas que ilustran los puntos definidos anteriormente en cuanto al Diseño OO.

4.1.1 Modelo de comportamiento de workflows

Para el diseño de workflows es necesario comprender el modelado de comportamiento, el cual describe los pasos que se siguen durante la ejecución del workflow y que caracterizan su comportamiento. Esta descripción incluye el momento de disparo de cada una de las actividades, qué acciones se deben de realizar en caso de error y el criterio para la elección de agentes.

Se distinguen tres tipos de disparadores de tareas: por iniciativa de un agente, esto es, una tarea disparada por un empleado; por un evento externo, como la recepción de un mensaje; o por una señal de tiempo, dígame el procesamiento de una lista de órdenes comienza a las 6:00 en punto.

4.2 Reingeniería

Como ya se menciona la solución del Grupo Pro esta desarrollada en Excel, sus componentes gráficos son muy pobres y las funcionalidades no son las más adecuadas para las actividades de la empresa.

Actualmente los sistemas computacionales, en su mayoría, están diseñados con ventanas y formas que hagan más fácil su manejo y acceso a las funciones. Normalmente éstas ventanas están compuestas de un conjunto de componentes gráficos, como son los campos de texto, botones, etiquetas, checklist, entre otros; y que tienen la característica de que el usuario puede controlarlos con la ayuda del mouse sin excluir al teclado.

Dados estos cambios en el desarrollo y diseño a través del tiempo, nuestra propuesta es el desarrollo de un sistema que sustituya la solución en Excel, haciendo un proceso de reingeniería de la solución, usando herramientas actuales y adaptables para los usuarios finales. Para este fin se ha tomado en cuenta las operaciones básicas realizadas por la solución de Excel, aunque para el nuevo sistema se realizó un nuevo análisis, haciendo modificaciones en el manejo de algunas de las operaciones y desarrollando nuevas funcionalidades.

El código desarrollado en el proyecto es totalmente nuevo, solo se tomó en cuenta la forma de operación de la empresa para el desarrollo de los mismos. Para la creación de las interfaces gráficas se tomó como guía la manera en que Excel presentaba los campos de ingreso de datos, para que al usuario final no le fuera tan difícil el adaptarse a este nuevo sistema; sin embargo en algunos casos fue necesario hacer un diseño totalmente nuevo debido a que no se encontraba nada desarrollado en Excel.

Como consecuencia de las restricciones de Excel, la información almacenada en sus libros contenía incongruencia, errores redundancias. Esto llevó a hacer una reingeniería en cuanto a que se tomó la información valiosa y se desechó la redundancia, sin alterar la gramática de sus datos.

4.2.1 Herramientas utilizadas

Delphi

Para la codificación de SIC se decidió utilizar Delphi, en su versión 7, debido a que cuenta con varios mecanismos de acceso a datos lo cual permite tener conexión con varios manejadores de datos, Oracle, InterBase, MySQL, entre otros; la manipulación de los datos

es ágil, se genera un único ejecutable, es flexible, de fácil manejo, por otro lado es una herramienta **RAD** (Rapid Application Development). Se debe aclarar que Delphi es una herramienta de desarrollo, no un lenguaje de programación; Object Pascal es el lenguaje que utiliza Delphi, el cual es una evolución de Pascal.

MySQL

Como manejador de base de datos se empleó MySQL, ya que es un sistema gratuito que se encuentra disponible a través de Internet; cuenta con lo necesario para poder llevar a cabo un buen funcionamiento del sistema y es aceptable para los fines del mismo.

Las razones para escoger MySQL como solución de misión crítica para la administración de datos son [Gilfillan, 2003]:

- **Coste:** el coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico
- **Asistencia:** MySQL AB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL
- **Velocidad:** MySQL es mucho más rápido que la mayor parte de sus rivales
- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación. Así mismos, se desarrolla y actualiza de forma mucho más rápida que muchos de sus rivales, por lo que prácticamente todas las funciones estándar de MySQL todavía no están en fase de desarrollo
- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

Crystal Reports

Para el desarrollo de los reportes que se utilizaran en la aplicación se utilizó el reporteador Crystal Reports en su versión 10, debido a que es uno de los más poderosos reporteadores

disponibles con la habilidad de obtener datos de cualquier fuente de datos [Taylor, 2004]. Puede utilizarse Crystal Reports para generar informes de cualquiera de los programas de base de datos estándares para PC; además tiene un poderoso servidor de informes Web, que permite distribuir los reportes a través del Web [Harvest, 2004].

4.3 Implementación

La figura 4.2 muestra la arquitectura del sistema desarrollado en este proyecto.

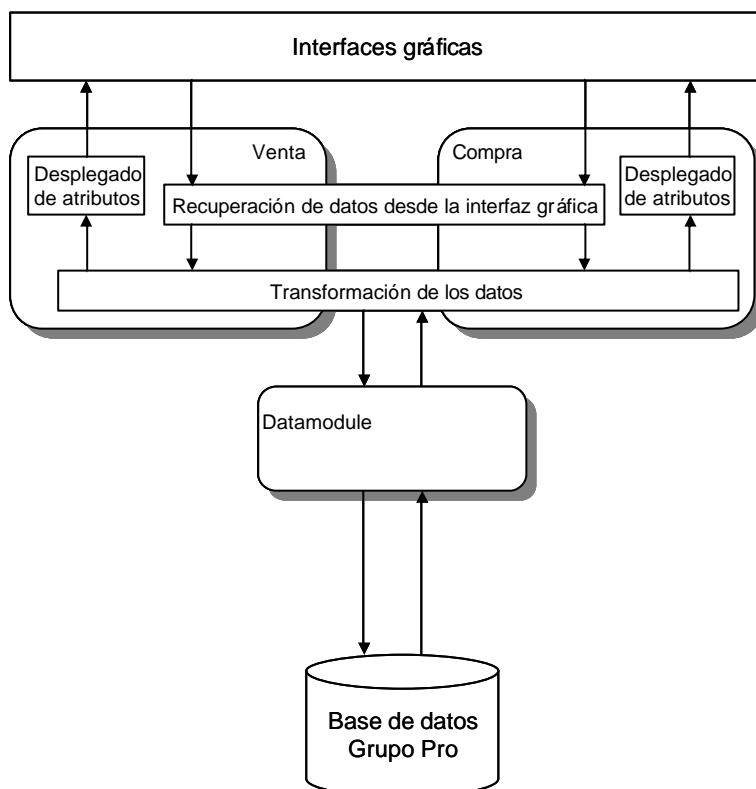


Figura 4.2 Arquitectura del sistema

Para la implementación se organizaron los módulos por secciones, de acuerdo a su funcionalidad; esto quiere decir que se agruparon las funciones del software en conjuntos bien definidos de acuerdo a los resultados que ofrece cada una de éstas. Las divisiones se presentan en el software como menús, en donde cada uno agrupa un número de funciones.

En cuanto a al interfaz, los menús principales son las categorías de los procesos que surgieron en el análisis (compras, ventas, inventario, producción, administración del

sistema y catálogos). A su vez cada uno de estos menús principales está dividido en submenús, los cuales especifican las tareas que pueden ser realizadas según la función del menú principal. MainForm es la ventana principal; es donde se encuentra el menú desde el cual se manda a llamar a todas las demás ventanas de los submenús Figura 4.3.

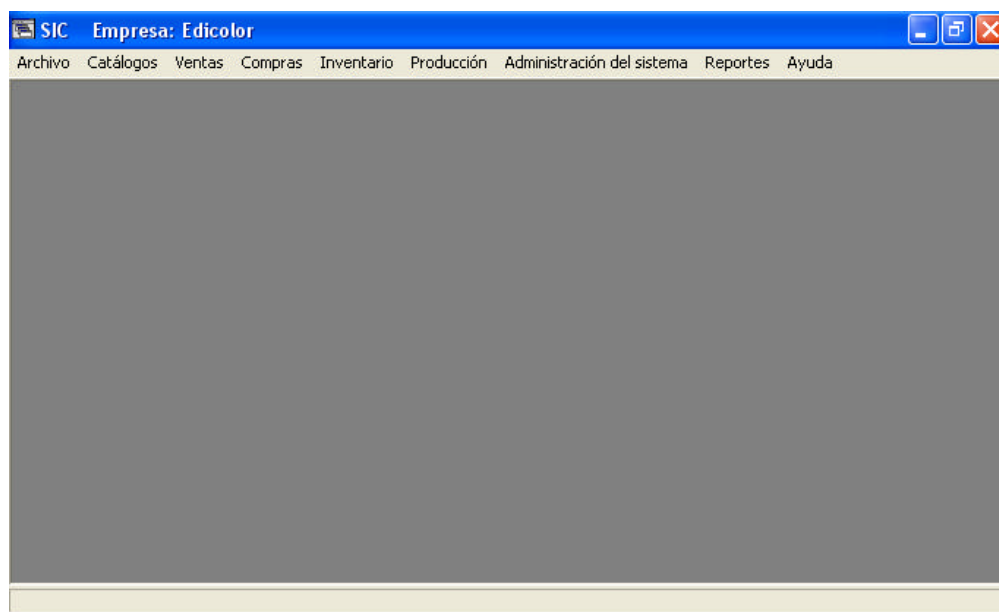


Figura 4.3 Pantalla principal

Es importante tener presente cada una de las categorías con las funciones que realiza, por lo que a continuación la tabla 4.1 muestra la estructura final de cada menú principal y sus submenús.

Compras	Orden de compra
	Registro de Compra
	Pago a proveedores
	Cancelaciones
	Cancelación total/parcial de orden de compra
	Notas de crédito
Ventas	Notas de cargo
	Pedido de venta.
	Facturas
	Pago de clientes
	Cancelaciones
	Cancelación total/parcial de pedidos
	Cancelación total de factura
	Notas de venta
	Notas de crédito
	Notas de cargo

Inventarios	Trasposos Facturación de trasposos Movimientos varios
Producción	Manejo de fórmulas Alta de fórmulas Baja de fórmulas Modificación de fórmulas Orden de producción Cierre de orden de producción
Administración del sistema	Permisos de usuarios Usuarios Alta de usuario Baja de usuario Modificación de usuario Consulta de usuario Compañías Alta de compañía Modificación de compañía Consulta de compañía Almacén Alta de almacén Modificación de almacén Consulta de almacén Respaldo
Catálogos	Tipo de cambio Clientes Alta de cliente Baja de cliente Modificación de cliente Consulta de cliente Proveedores Alta de proveedor Baja de proveedor Modificación de proveedor Consulta de proveedor Vendedores Alta de vendedor Baja de vendedor Modificación de vendedor Productos Alta de producto Modificación de producto Consulta de producto Familia Alta de familia Modificación de familia Consulta de familia Color Alta de color Modificación de color Consulta de color Clasificación Alta de clasificación Modificación de clasificación Consulta de clasificación Línea

Alta de línea
Modificación de línea
Consulta de línea
Marca
Alta de marca
Modificación de marca
Consulta de marca

Tabla 4.1 Organización del menú

Cabe mencionar que cada submenú del menú principal es una unidad; donde cada una de ellas realiza ciertas operaciones dependiendo de la actividad que se desea realizar, pedido, facturación, pagos, entre otros.

Delphi utiliza unidades, las cuales tiene una estructura fundamental para su funcionamiento. Esta estructura se muestra a continuación [Cantú, 2003]:

1. Palabra reservada **unit** seguida del nombre de la unidad
2. La palabra reservada **interface**, la cual continúa hasta el inicio de la sección **implementation**. En esta sección se declaran las constantes, tipos, variables, procedimientos y funciones, las cuales estarán disponibles para los clientes, esto es, para otras unidades o programas que deseen usar los elementos de esta sección. Estas entidades son llamadas públicas, debido a que un cliente puede acceder a ellas, como si éstas estuvieran declaradas en el mismo cliente. La declaración de interfaz de un procedimiento o función incluye sólo el título de la rutina. Esta sección puede incluir su propia cláusula **uses**, la cual debe aparecer inmediatamente después de la palabra **interface**.
3. La cláusula **uses** consiste de la palabra reservada **uses**, seguida de una o más nombres de unidades delimitados por comas, seguidos de un punto y coma (;). En la mayoría de los casos, todas las unidades necesarias son colocadas en la cláusula **uses** cuando se genera el proyecto y mantiene un archivo de origen.
4. Palabra reservada **implementation**, la cual continúa hasta el inicio de la sección **initialization** o, si no existe esta sección, hasta el fin de la unidad. Esta sección define los procesos y funciones que se han declarado en la sección **interface**. Adicionalmente a la definición de procesos públicos y funciones, se pueden declara constantes, tipos (incluyendo clases), variables, procedimientos, y las

funciones que son privadas, es decir, inaccesibles a los clientes. Además puede contener su propia cláusula **uses**, la cuál debe aparecer inmediatamente después de la palabra **implementation**.

5. Para hacer comentarios en Delphi se puede hacer de diferentes formas: “//” para comentar una línea y “{ }” para comentar una línea o un bloque.

En el ejemplo siguiente se muestra el uso de la estructura de las unidades de Delphi.

```

unit usuario_inicio;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls,
  ExtCtrls, ComCtrls, Mask;
type
  Tinicio = class (TForm)
    usuario: TLabeledEdit;
    aceptarbtn: TButton;
    procedure aceptarbtnClick(Sender: TObject);
  private
    { Private declarations }
  public
    nombre,compania, usuarios: string;
    { Public declarations }
  end;
var
  inicio: Tinicio;

implementation
uses MAIN, datamodule;

procedure Tinicio.aceptarbtnClick(Sender: TObject);
begin
  // implementación del proceso.
end;
end.

```

Figura 4.4 Unidad de Delphi

Las unidades Datamodule y Rutinas son utilizadas por la mayoría de las unidades, debido a que contienen funciones y procesos necesarios ya sea, para el manejo de la información o, acciones recurrentes, por lo tanto es conveniente mencionar su funcionamiento.

La unidad Rutinas, cuenta con los procesos y funciones, como formato de moneda, el cual es requerido para la realización de ciertas operaciones comerciales.

La unidad Datamodule cuenta con los componentes necesarios para la conexión a la base de datos y para la recuperación de la información. A través de esta unidad las demás unidades podrán recuperar, manipular y actualizar la información. Ésta unidad utiliza los componentes del mecanismo de acceso a bases de datos **DBX (dbExpress)** de Delphi para la conexión con MySQL, debido a que está basado en la arquitectura cliente/servidor, sus controladores son mucho más sencillos y eficientes que los controladores **BDE** (Borland Database Engine) o **ADO** (Active Data Objects), es una solución multiplataforma, que permite hacer la conexión con el servidor, recuperar datos, manipularlos y efectuar actualizaciones [Charte, 2003]. Los componentes empleados para dicha conexión son:

- **TSQLConnection**. Para establecer la conexión con una base de datos. Éste componente precisa tres datos fundamentales: el nombre del controlador, el nombre de la función de entrada ha dicho controlador y el nombre de la biblioteca del cliente de la base de datos.
- **TSQLQuery**. Para establecer los queries necesarios para la recuperación del conjunto de datos.

Para lograr un mejor entendimiento de las funciones del sistema, la explicación de los componentes de estos será dividida según las operaciones comerciales que se realizan, y que podemos dividirlos en los grandes grupos: Compras Ventas, Manejo de Inventario, Productos, administración del sistema y los procesos que se modelaron como workflows.

4.3.1 Venta

El proceso de venta que se lleva a cabo en las compañías del Grupo Pro se debe componer de la siguiente manera.

4.3.1.1 Pedido

Este proceso comprende el listado de los productos que un cliente desea comprar, o bien ordenar producir. En el pedido se incluyen las claves de los productos y sus cantidades, a parte de los datos del cliente y un presupuesto aproximado del total.

Para la implementación de este proceso se creó el objeto Pedido (mostrado desde el análisis). Este objeto contiene los datos de lo que podemos llamar el encabezado del pedido. Por otro lado tenemos el objeto Línea_Pedido el cuál define, principalmente, la clave del producto, sus características, su cantidad surtida, por surtir y precio unitario.

Los métodos del objeto Pedido son: alta de pedido, cancelación total o parcial de pedido, obtención de la moneda en la que se presupuestó el pedido y cálculo del importe total.

El objeto Línea_Pedido contiene métodos que permiten la agregación, eliminación y modificación de cada una de ellas. También contiene métodos de validación de cantidad y de precio, así como el cálculo del importe por línea.

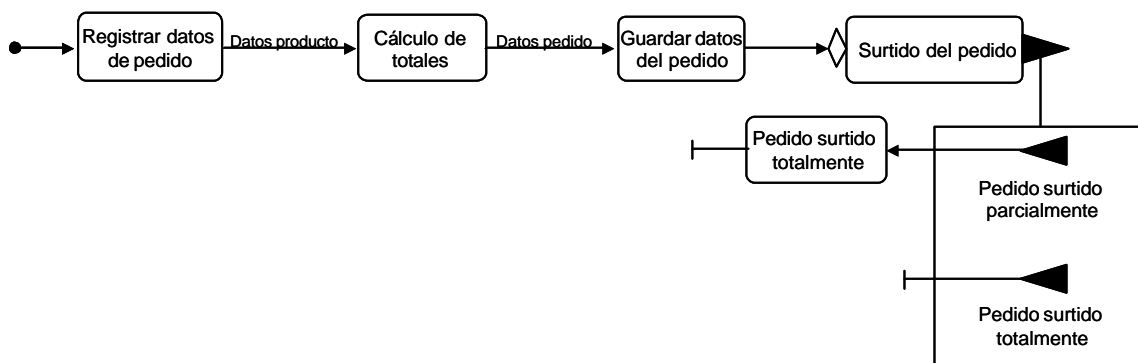


Figura 4.5 Diagrama de transición de estados del pedido

4.3.1.2 Factura

Especifica la materialización de la venta. Se debe mencionar que el proceso de facturación se puede realizar de dos formas: (1) La facturación directa de un pedido; (2) Facturación de

productos que no se encuentran en el catálogo y no atañen al giro de la empresa, dígame venta de mobiliario.

El objeto Factura contiene los datos generales del cliente, así como los datos indispensables que permiten la generación de la factura. También se tiene el objeto Línea_Factura, el cuál contiene los datos necesarios de los productos vendidos al cliente.

Los métodos concernientes al objeto Factura son: alta de factura, cancelación total de factura, calculo del importe total. Debe verificar la fecha que se indica en la elaboración de la factura, pues existe la posibilidad de que ésta sea anterior o posterior al día del movimiento, lo cual conlleva a manejo de autorizaciones. Partiendo de la fecha indicada, se debe obtener el tipo de cambio que será aplicado. En caso de que ésta provenga de un pedido, los datos del cliente deben ser obtenidos de éste; si se factura un producto ajeno al giro comercial de consorcio, los datos pueden ser obtenidos o simplemente ingresados por el usuario. Línea_Factura puede ser editada y agregada.

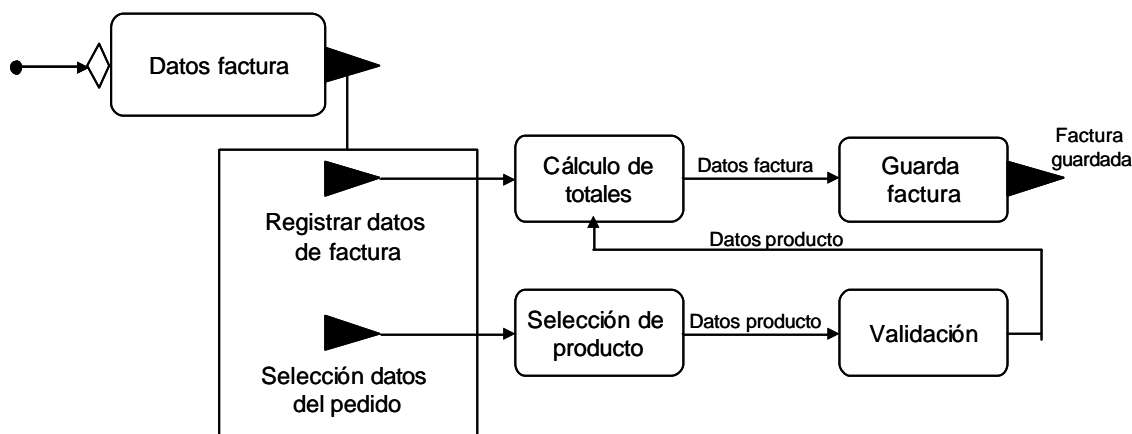


Figura 4.6 Diagrama de transición de estados de la factura

4.3.1.3 Nota de venta

Este objeto y su objeto asociado, Línea_Notaventa, tiene un comportamiento equivalente a Factura y Línea_Factura, con la diferencia comercial de que ésta no tiene implicaciones fiscales ni muchos menos está asociada a algún pedido. Son ventas de mostrador.



Figura 4.7 Diagrama de transición de estados de nota de venta

4.3.1.4 Nota de cargo

Este objeto se utiliza cuando, en la elaboración de una factura, hubo un error humano en el ingreso de los precios, siendo estos menores a los correctos. Esto implica generar un documento en el que se especifique la diferencia a pagar por el cliente. En este objeto también se maneja un encabezado que contiene los datos de la factura asociada. El objeto `Linea_NotaCargo` indica los atributos obtenidos del objeto `Linea_Factura` a los cuales se aplicará el cargo.



Figura 4.8 Diagrama de transición de estados de nota de cargo

Contiene los métodos: alta, obtención de la factura asociada y sus datos, cálculo de las diferencias y totales. `Linea_NotaCargo` obtiene los datos de `Linea_Factura` de la factura asociada y el cálculo de los importes por línea.

4.3.1.5 Nota de crédito

Este documento puede ser generado por dos causas: (1) la devolución de productos por parte del cliente; (2) la factura fue generada por un importe mayor al debido. Estas situaciones llevan a otorgar al cliente un crédito.

Contiene los métodos: alta, obtención de la factura asociada y sus datos, cálculo de las diferencias y totales. `Linea_NotaCrédito` obtiene los datos de `Linea_Factura` de la factura asociada y el cálculo de los importes por línea.

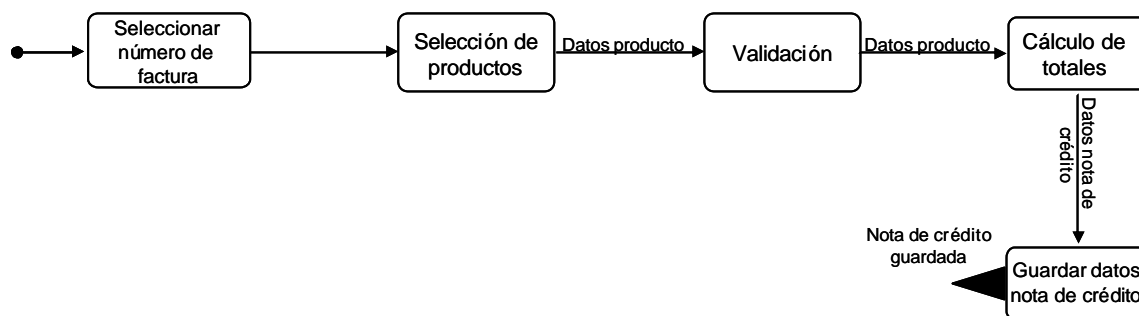


Figura 4.9 Diagrama de transición de estados de nota de crédito

4.3.1.6 Cliente

Son las personas a las cuales el consorcio realiza la venta de productos, contiene atributos para la identificación y manejo de los mismos. Este objeto está asociado a los objetos de Venta. Tiene los métodos principales alta, baja, modificación, consulta y pagos. Estos últimos se efectúan sobre las notas de venta y facturas, y pueden estar afectados por las Notas de Cargo y Crédito respectivas. Además existen métodos auxiliares como: verificación de saldos y de RFC.

4.3.1.7 Vendedor

Es la persona asignada a la atención de cliente, cada cliente tiene asignado un único vendedor. Para el momento de desarrollo de este proyecto el manejo de la información de los vendedores no es de gran relevancia, pero servirá de base para el desarrollo posterior del proceso de cálculo de comisiones.

4.3.2 Compra

El proceso de compra que se lleva a cabo en las compañías del Grupo Pro se debe componer de la siguiente manera.

4.3.2.1 Orden de compra

Es un listado de productos requeridos para compra seleccionados según la existencia física actual, los cuales serán solicitados a un proveedor. Tiene una estructura maestro-detalle; donde el maestro tiene los datos del proveedor y características de la orden, y el detalle contiene los productos así como sus características y cantidades a solicitar. Como métodos

se encuentran el alta de la orden, registro de la entrega de los productos y sus cantidades, cancelación de la orden, el cálculo del total de la orden. Linea_OrdenCompra contiene los métodos agrega, edita y elimina línea, verifica cantidad y precio, así como el cálculo del importe por línea.

4.3.2.2 Nota de crédito y cargo

Son objetos cuyo comportamiento es similar al de Nota de Crédito y Cargo de ventas, con la diferencia esencial de que afectan los cobros que hace el proveedor a la compañía.

4.3.2.3 Proveedor

Son mas personas o empresas a las cuales se les solicitan productos y materias primas. Este objeto contiene atributos para la identificación y manejo de los proveedores; por su parte estos están asociados a los objetos involucrados en las compras. Sus métodos principales son alta, baja, modificación, consulta y pagos. Los pagos se efectúan una vez que han sido registrados los productos, las cantidades a pagar se ven afectadas por las notas de crédito y cargo respectivas. Por otro lado existen métodos secundarios como: verificación de saldos y de RFC.

4.3.3 Inventario

Tiene los atributos correspondientes a cada producto involucrado en un movimiento de inventario, además posee datos que identifican y permiten manipular los movimientos de inventario. Los métodos contenidos en inventario son: Movimientos de entrada y salida, trasposos intercompañías y trasposos entre almacenes, mismos que están conformados por movimientos de entrada y salida, verificación de existencias por producto, el cual será detallado a fondo en la implementación de workflows.

El proceso de trasposos en el caso de ser intercompañías se realiza de la siguiente manera: primero se selecciona el producto a traspasar de cualquiera de los almacenes disponibles, posteriormente se hace la selección de la empresa a la cual se desea traspasar, el almacén al que se desea ingresar, y selección de la clave correspondiente en la empresa destino. Una vez echa la selección se procede a sacar el almacén seleccionado el producto

para ingresarlo al almacén 9 de la empresa de salida, para posteriormente pasar el producto al almacén seleccionado de la empresa destino.

En el caso de que el proceso de traspaso sea entre almacenes se procede a seleccionar el almacén de salida, el producto de salida, el almacén de entrada y la clave del producto con la cual se ingresará al almacén deseado. Este traspaso puede ser utilizado cuando se hace la conversión de las unidades en las que se vende un producto.

4.3.4 Producto

Contiene los atributos correspondientes a los productos que son manejados por el consorcio. Tiene los métodos alta, modificación, consulta, verificación de contratipos, verificación de características, orden y cierre de producción, así como el alta, baja, modificación y consulta de fórmulas.

4.3.5 Administración del sistema

4.3.5.1 Usuario

Son las personas que manejarán la aplicación, cuenta con los atributos necesarios para su identificación, asignación y restricción en el uso de los diferentes procesos. Tiene los métodos alta, baja, modificación, consulta y verificación de permisos. El método verificación de permisos funciona utilizando algebra booleana, pues ésta permite el fácil “encendido” y “apagado” de los permisos. La operación AND permite la activación de los procesos; la operación OR nos ayuda a armar un número decimal que es tratado como binario, el cual indica que permisos tiene disponibles el usuario.

4.3.5.2 Almacén

Contiene la descripción de los almacenes disponibles. Tiene los métodos alta, modificación y consulta.

4.3.5.3 Compañía

Contiene los datos referentes a las compañías que conforman el consorcio. Tiene los métodos: alta, modificación y consulta.

4.3.6 Implementación de workflows

Para el desarrollo de los procesos verificación de existencia de productos, pagos pendientes de clientes y a proveedores, generación de archivo de contabilidad primero se especificarán los componentes necesarios para su funcionamiento, en base al modelado del comportamiento de workflows.

4.3.6.1 Verificación de existencias .

Este workflow es el encargado de, periódicamente, revisar las existencias de los productos a una fecha determinada.

Para esto es necesario realizar las siguientes tareas:

1. Calcular las existencias físicas en inventario de cada uno de los productos.
2. Obtener, de las ordenes de compra, qué productos y en qué cantidad, serán supuestamente recibidos a una fecha específica. La entrega de los productos se ve afectada por la puntualidad de los proveedores, es por eso que se toma un supuesto de las cantidades a recibir.
3. Verificar, de los pedidos de clientes, qué productos y en qué cantidad deben ser entregados en el periodo analizado.
4. Finalmente se debe obtener los productos y las cantidades de éstos que se están produciendo, para saber si se puede contar o no con ellos al día que se está consultando.
5. Ya teniendo estas cantidades se procede a realizar la siguiente operación: Existencia en inventario + cantidad a recibir de un proveedor + cantidad a obtener de producción – cantidad a entregar en los pedidos. Esta operación nos indica la existencia supuesta global a una fecha.

6. Finalmente se compara la existencia supuesta global con la cantidad mínima requerida en inventario para obtener un listado de los productos que deben ser, ya sea producidos, o bien, solicitados a los proveedores.

Las 4 primeras tareas pueden realizarse de manera paralela. Al tener los resultados de todas ellas, se procede a las últimas 2 de manera secuencial.

Este proceso, o workflow, va a ejecutarse en el primer ingreso al sistema en el día por parte del usuario correspondiente, esto es, aquel que tiene los permisos necesarios para visualizar esta información. Aquí podemos ver un ejemplo de la definición de roles requerida en el modelado de workflows.

En caso de error, se notificará al usuario el estatus del proceso y la causa del problema.

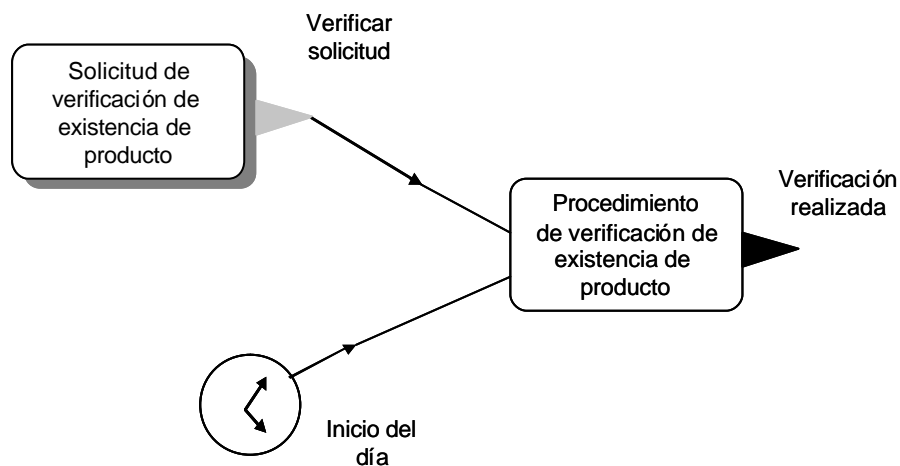


Figura 4.10 Diagrama de transición de estados de verificación de existencias

4.3.6.2 Pagos pendientes de clientes y a proveedores

Primero se debe aclarar que estos procesos son muy similares en cuanto a funcionamiento, su diferencia consiste en la información que maneja. En el control de pagos pendientes de clientes la información es obtenida de las tablas Factura y PagoCliente. En el caso de los pagos pendientes a proveedores se toman en cuenta las tablas Registro de Compra y PagoProveedor.

El control del crédito en los pagos de las facturas se efectúa de la siguiente manera:

1. Se deben obtener las facturas (o registros de compras) que no han sido saldadas en su totalidad.
2. Se verifica el tipo de crédito que tienen los clientes relacionados a las facturas obtenidas de la tarea anterior.
3. Se calculan los días transcurridos desde la realización de la factura hasta la fecha en la que se ejecuta el proceso.
4. Se comparan los días de crédito asignados con los días transcurridos.
5. Se presenta un reporte de las facturas (o registros de compras) a las que les restan 3 días de crédito y todas aquellas que ya son vencidas.

La sucesión de las acciones en este proceso son totalmente secuenciales. Este workflow se accionará, de la misma manera que el anterior, en el primer ingreso al sistema del usuario correspondiente.

En caso de error, el usuario visualizará un mensaje en el que se indica la razón de la falla.

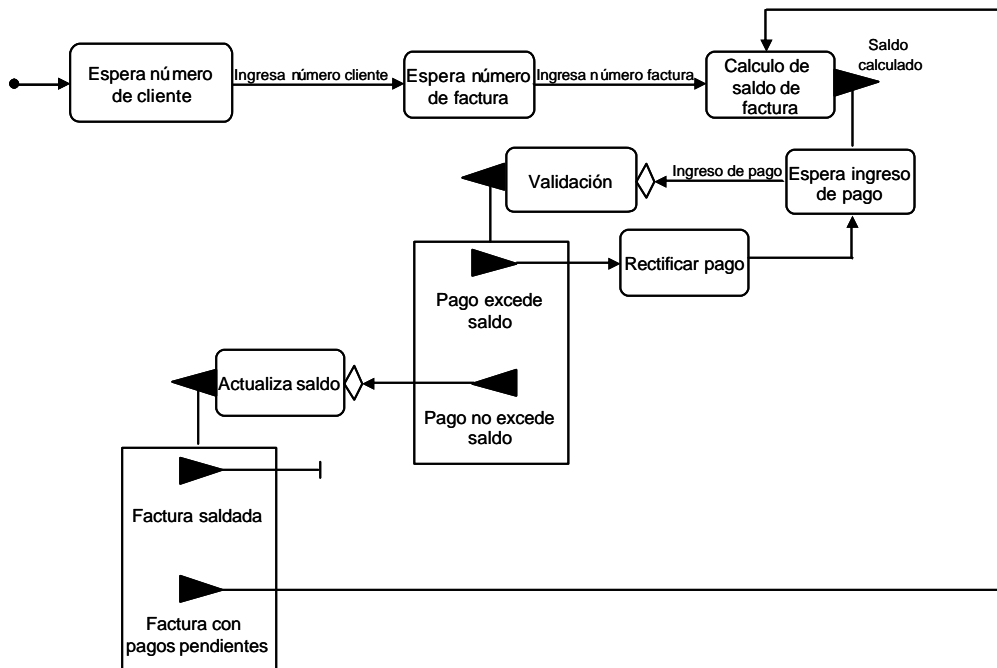


Figura 4.11 Diagrama de transición de estados de pagos

4.3.6.3 Generación de archivo de contabilidad

Como se mencionó al inicio del documento, este sistema no maneja información relativa a contabilidad, es por eso que se hace necesaria la exportación de datos de compras y ventas, para su ingreso en el sistema de contabilidad, completamente ajeno a este proyecto.

Para la obtención del un archivo de texto con la información requerida por el sistema de contabilidad se debe:

1. Obtener todos los movimientos comerciales realizados en el periodo de tiempo especificado por el usuario, sus importes, fechas y moneda en que se efectuaron. Estos movimientos son: facturas, órdenes de compra, notas de crédito y cargo en ventas, notas de crédito y cargo en compras, pagos de clientes y pagos a proveedores.
2. Relacionar los movimientos con los clientes y proveedores respectivos, así como las cuentas contables correspondientes a cada uno de ellos.
3. Agrupar estos registros por día de realización.
4. Convertir los importes de dólares americanos a su equivalencia en moneda nacional según el tipo de cambio que aplicó en la fecha del movimiento.
5. Calcular los importes totales por día, en moneda nacional y en dólares americanos.
6. Generar el archivo de texto según el formato especificado por el sistema de contabilidad.

La generación del archivo de exportación de datos se efectuará según el periodo definido por el usuario en la configuración del proceso; o bien, al momento en que el usuario haga la petición de ejecución.

Si llegase a ocurrir un error, el usuario será informado de la causa de éste y se detendrá la generación del documento.