



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ALMACÉN PARA UNA COMPAÑÍA REFRESQUERA.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

MARÍA PETRA GONZÁLEZ VARGAS

RUTH ERIKA HERRERA CHÁVEZ

NYDIA NAVARRETE LOZANO

LEOPOLDO ZAVALA AGUAYO

288073

DIRECTOR DE LA TESIS: M. EN I. JUAN CARLOS ROA BEIZA

MÉXICO, D.F.

200/





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

MARÍA PETRA GONZÁLEZ VARGAS

*A mi hermano Odi
Por todo su apoyo en mi formación educativa.*

*A mi familia
Por su comprensión y amor.*

*A mis maestros
Por su ejemplo y conocimientos para poder llegar a este momento.*

A aquellas personas que me apoyaron y motivaron en las diferentes etapas de mi vida para lograr mis objetivos.

RUTH ERIKA HERRERA CHÁVEZ

A mis padres:

Por haber hecho de mi lo que soy, porque gracias a su apoyo, comprensión y cariño he logrado una de mis más grandes metas.

Porque han sido unos padres ejemplares, pues de ellos he aprendido las cosas de más valía en mi vida. Muchas gracias, los amo.

A mi esposo:

Porque ha sido mi mayor motivación para la culminación de mis metas, porque gracias a su apoyo y amor me he superado y crecido como mujer y profesionalista.

Por la fortuna de tenerlo a mi lado en cada instante de mi vida y hacerme la mujer más dichosa y agraciada.

A mis hermanos:

Por haber compartido conmigo las etapas más valiosas de mi vida. Porque el haber caminado juntos es algo único, que se ha visto reflejado en cada uno de mis logros.

A mis sobrinos:

Porque me han enseñado las cosas más importantes que existen en la vida, gracias por su amor y su alegría de vivir.

A mis profesores:

Por haberme obsequiado los más inestimables conocimientos, que han dejado una huella que perdurará por el resto de mi vida.

NYDIA NAVARRETE LOZANO

*Gracias por estar siempre conmigo,
por que el apoyo recibido
es el mejor regalo que he tenido.
Por dejarme ser y hacer.
Por formar parte de este esfuerzo.
Y por que este logro obtenido
no solo es mío...
si no de ustedes también.
Papa y Mama los quiero mucho.
Ojala tuvieran unos padres, los mejores...
como los tuve yo.
Los quiere y ama su hija.
Josefina y Javier*

*Por su cariño, por su aprecio,
por sus cuidados...
por ser confidentes y amigos...
Por ser lo que son, mis hermanos...
así de sencillo..
Por que los quiero
y por que gracias a ustedes
mi esfuerzo es compartido
Los quiero mucho a los dos.
Javito y Davicho*

*Por alentarme a seguir estos últimos pasos,
por aguantarme y estar ahí...
por que te Amo...
DANi
Gracias también.*

LEOPOLDO ZAVALA AGUAYO

A mi esposa

Carolina

Por darme el coraje y el ánimo necesario para culminar este trabajo y apoyarme en mi carrera profesional.

A mis hijos

Leopoldo, María Fernanda y Hernán

A mis queridos hijos por ser ellos la recompensa a todos mis esfuerzos y la fuente de inspiración para fijarme nuevas metas.

A mis padres

Leopoldo y Mercedes

Gracias por haberme dado la vida y la gran oportunidad de terminar una carrera, por apoyarme en todo momento y por hacerme quien soy.

INDICE

CAPÍTULO 1. REGLAS DEL NEGOCIO.....	3
1.1. CONCEPTOS BÁSICOS DE CONTROL DE INVENTARIOS.	3
1.1.1. Insumos.....	4
1.1.2. Canales.....	5
1.1.3. Rutas.....	6
1.1.4. Control de producción.	8
1.1.5. Control Monitor.	8
1.2. CONCEPTOS FUNDAMENTALES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	8
1.2.1. Que es la programación orientada a objetos.....	8
1.2.2. El objeto.	8
1.2.3. Ejemplo de objeto.....	9
1.2.4. Métodos y mensajes.....	10
1.2.5. Clases.	10
1.2.6. Estructura interna de un objeto.	10
1.2.7. Herencia.	11
1.2.8. Ligadura dinámica.	14
1.2.9. Polimorfismo.	14
1.2.10. Reutilización con orientación a objetos.....	15
1.3. MODELADO DE OBJETOS.	15
1.3.1. Metodología del desarrollo de sistemas de información.	16
1.3.2. Definición de la metodología del desarrollo de sistemas.	16
1.3.3. Lenguaje Unificado para Modelado.	18
1.3.4. Modelado de objetos con UML.....	19
1.3.5. Componentes estructurales de los sistemas de información.....	25
1.4. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE VISUAL BASIC..	27
1.5. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE SQL-SERVER.....	35
CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN.....	42
2.1. PROBLEMÁTICA ACTUAL.	42
2.2. EL USUARIO Y SUS REQUERIMIENTOS.....	48
2.3. BÚSQUEDA Y ANÁLISIS DE LA INFORMACIÓN.....	54
2.4. IDENTIFICACIÓN DEL PROBLEMA.	57
2.5. ÁREAS INVOLUCRADAS EN EL PROCESO.....	60
2.6. OPCIONES DE SOLUCIÓN Y SELECCIÓN DEL BACK-END.	70
2.7. OPCIONES DE SOLUCIÓN Y SELECCIÓN DEL FRONT-END.	79
CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA.	90
3.1. APLICACIÓN DE LA METODOLOGÍA ELEGIDA (BACK-END PARA CADA MÓDULO.	90
3.1.1. Diagramas de contexto.	90
3.1.2. Diagrama Dinámico de eventos.....	97
3.1.3. Diccionario de Datos.	104
3.1.4. El Modelo Entidad-Relación.....	104

CAPÍTULO 1. REGLAS DEL NEGOCIO.

1.1. CONCEPTOS BÁSICOS DE CONTROL DE INVENTARIOS.

El tema de los inventarios, así como todo lo que a ellos se refiere –compras, almacenamiento, costo de ventas, etc.–, es el tema central de cualquier negocio en el que se manejen mercancías; concretamente en todo comercio, en toda industria, y en toda negociación agrícola.

También desde el punto de vista de Contabilidad las existencias ofrecen aspectos interesantes, tanto de control como de estimación e interpretación de resultados. Conviene recordar al efecto que:

1. La mercancía se compra a un precio (precio de costo) y se vende a otro (precio de venta), normalmente superior; pero que en ciertos casos puede ser inferior al primero.
2. De la diferencia entre precio de costo y precio de venta se forma el “resultado” que puede ser utilidad o pérdida bruta.
3. Para conocer el resultado de las operaciones realizadas con mercancías durante un ejercicio es necesario calcular el costo de lo vendido.
4. Este cálculo puede hacerse de dos maneras:
 - a) Multiplicando el costo de cada unidad vendida por el número de unidades vendidas (sistema de inventario perpetuo), o bien
 - b) Sumando las compras del ejercicio al inventario inicial y restándole el inventario final (sistema de comparación de inventarios)
5. El primer procedimiento es más exacto pero laborioso; el segundo es más sencillo, pero menos preciso ya que el costo de lo vendido se incluirían mermas, faltantes y sobrantes sin determinar su importe.
6. En ambos casos es necesario tomar inventarios físicos con cierta periodicidad.

Con estas ideas como base podemos entrar a ciertos pormenores relativos a la organización del almacén.

Los temas centrales que comprende la contabilidad de almacenes son los siguientes:

1. La compra, que implica el concepto del costo y la entrada de mercancía al almacén.
2. La salida de la mercancía.
3. La toma de inventarios.
4. El almacenamiento o custodia de la mercancía.

Como de costumbre la contabilidad ejerce un control sobre estas operaciones con el doble fin de proteger los intereses de la empresa (control interno) y de calcular los resultados de una manera razonablemente exacta.

LOS INVENTARIOS FÍSICOS.

Cualquiera que sea el sistema de contabilidad que se adopte para el control del almacén, es indispensable practicar inventarios físicos con la frecuencia que las circunstancias lo indiquen. Esta es una labor de trascendencia fundamental, sobre todo por los efectos que tiene para el cálculo de la utilidad – o de la pérdida- resultante de las operaciones del ejercicio.

Recuerde que al hacer los ajustes por inventario, la utilidad o la pérdida va en razón directa del valor del inventario; o sea que a mayor inventario, mayor utilidad –o menor pérdida-; y que, a menor inventario, menor utilidad, o mayor pérdida.

No obstante su trascendencia contable, el tomar un inventario es esencialmente una labor de carácter físico que requiere sujetarse a ciertas normas y seguir ciertos procedimientos.

El conocimiento del material o mercancía que maneje la industria así como de sus actividades y costumbres, es indispensable, no sólo para hacer el recuento físico de dicho material, sino también para elegir el procedimiento de recuento que sea más práctico en vista del caso especial de que se trate.

Un inventario es, esencialmente, una lista; y para formarla debe procederse con cierto orden.

1.1.1. Insumos.

Los insumos requeridos para la elaboración de un producto desde su concepción hasta que llega al consumidor es ardua, desde el punto de vista de inversiones a menudo riesgosa. Por cada insumo que se requiere es necesario elaborar un método muy seguro de adquisición y conservación hasta su uso para garantizar la calidad de los productos que se elaboran en la compañía.

Numerosas pruebas con los productos requeridos y técnicas avanzadas de selección proporcionan los estándares para identificar la vida útil del producto.

Dentro de la selección de materiales se debe conocer en extremo el uso del producto, controlar su distribución, llevar registros ordenados y controlar el inventario como una de las partes más importantes para la empresa.

Es requerido un exhaustivo control en las materias primas, procesos internos y productos terminados. Se deben realizar análisis complejos, oportunos y exactos; los análisis son requeridos para evaluar la aceptabilidad de un producto.

En el curso de la producción de los productos se generan documentos y registros. A cada lote se le asigna un código o número específico que se conoce como "registro de lote" y comprende datos sobre cada fase significativa de la producción, control y distribución. Estos procedimientos son realizados por gente responsable y capacitada para contribuir en la seguridad e integridad del producto. La figura 1.1.1.1 muestra el proceso de servicio.

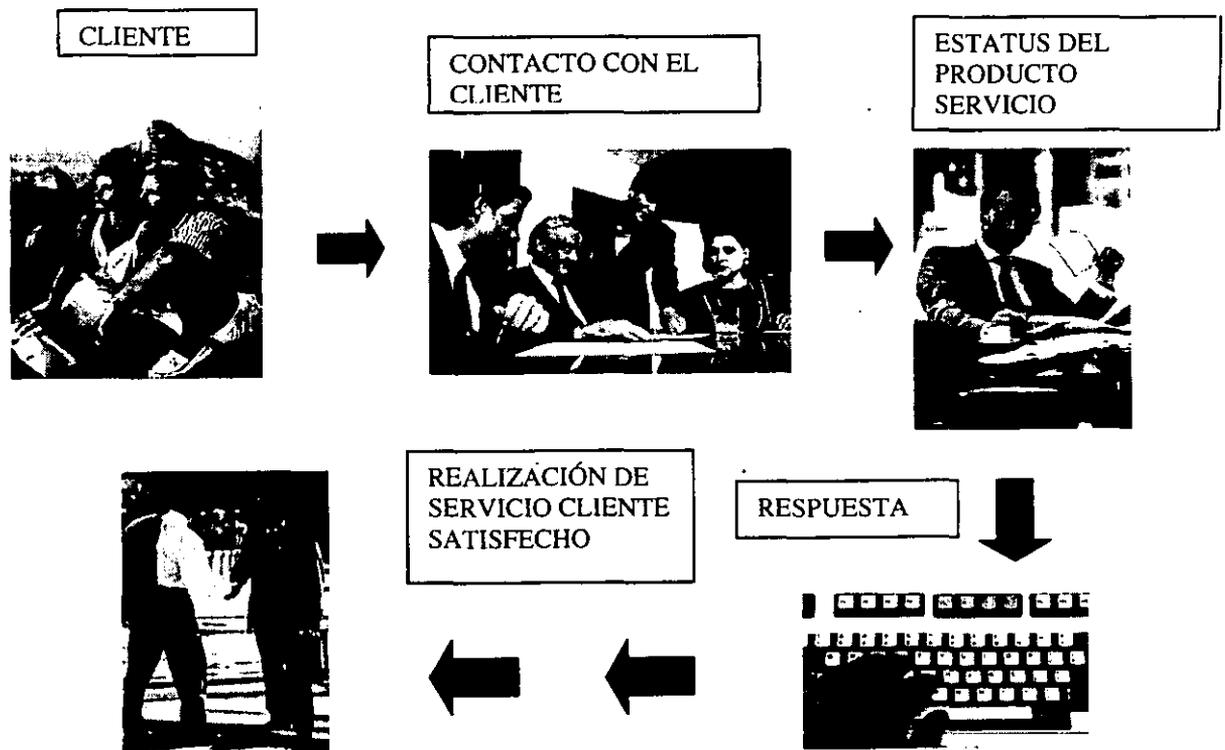


Figura 1.1.1.1. Proceso de Servicio.

El procesamiento electrónico de datos se ha convertido en otro recurso útil para estimar procesos y parámetros de prueba y para analizar los datos recogidos durante la producción. El control de operaciones por computadora y microprocesadores provee capacidad para elaborar productos de mayor calidad. Estos sistemas han venido desplazando a los más antiguos y han

dado lugar a nuevos enfoques en el control de procesos interno, en la recolección y análisis de datos.

Los materiales de etiquetado y envasado se recibirán, identificarán, almacenarán, muestrearán y ensayarán de acuerdo con procedimientos detallados por escrito.

Se ejercerán controles especiales sobre el etiquetado para asegurarse de que solo se emitan las etiquetas correctas para el envasado de un producto dado con la calidad óptima.

Luego de los estudios de estabilidad que corresponden a las condiciones de temperatura prescrita, los productos que salen al mercado deben de tener una fecha de vencimiento para tener la seguridad de que se usarán dentro de su vida de almacenamiento prevista.

Durante la elaboración de los productos se deben generar detalles de los diversos registros y documentos que deben estar disponibles para examinarlos. Se debe preparar un registro maestro de producción para cada producto, describiendo todos los aspectos de su elaboración, envasado y control. De este registro maestro se obtienen los registros de los lotes individuales.

Se debe contar con un archivo de quejas para mantener el registro de las quejas recibidas de los consumidores y profesionales, junto con el informe y su respuesta.

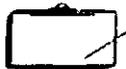
1.1.2. Canales.

Los registros de distribución comprenden anotaciones de embarque, remisiones, notas de carga y todos los documentos relacionados con la distribución, los canales que se establecen deben de ser definidos correctamente para cumplir con las normas nacionales e internacionales.

A continuación se describen los requisitos para almacenamiento y distribución de los productos y su tenencia en condiciones apropiadas de luz, temperatura y humedad, los productos distribuidos en la empresa no son perecederos por lo cual se facilita el transporte y distribución de los mismos, sin embargo el cuidado en el tipo de transporte requerido es básico por la delicadeza del envase que contiene el producto final, ya que se requiere un diseño especial en el contenedor de envases y en el proceso de carga y descarga para la distribución Nacional e Internacional del producto.



Consulta de la información en almacén, físicamente y/o mediante tarjetas de consumo



Se genera un problema al llevar un registro manual de cada producto en el documento de inventario.

Detección de inconsistencias.

Obtención de información estadística



Registro contable para determinación del valor total.

Figura 1.1.2.1. Diagrama de elaboración del inventario para la distribución del producto.

Los diferentes canales por los que se distribuye el producto están identificados dentro de las rutas establecidas, así mismo estos determinan correctamente cuales son los medios permitidos para su distribución y entrega al consumidor.

Se debe llevar un registro de los productos devueltos por los canales de distribución y los motivos de su devolución. La figura 1.1.2.1 muestra un diagrama de elaboración del inventario para la distribución del producto.

1.1.3. Rutas.

Para comprender mejor todo lo que gira en torno a un pedido, es necesario tocar brevemente cada una de las partes que participan durante el proceso de realización del mismo, la parte proveedora o vendedora y la parte consumidora o compradora, para después enfocarnos a los requerimientos que resultan indispensables para su realización.

Comenzaremos con la parte que funge como proveedora, es decir, aquella entidad que se encarga de suministrar con base en una venta, sus productos, esta parte necesita tener amplios conocimientos sobre la dinámica del flujo de sus insumos, esto para garantizar un suministro constante. Lo que quiere decir que se debe tener conocimiento de la información que fluye tanto de una petición para reaprovisar sus almacenes como de una licitación hecha por una entidad externa que pudiera tener impacto sobre su capacidad de producción así como las operaciones contables que se generen por consecuencia de las ventas, sin dejar fuera la dinámica de distribución de los productos. Se comienzan a ver por tanto, un conjunto de elementos de producción, de control de calidad, de almacenaje, de distribución, hasta llegar al lugar propio de consumo, que solo si se comprenden adecuadamente podrán llevar a esta parte a cumplir con su papel; y sin perder de vista que como es una entidad que esencialmente se dedica a la venta, tiene que ser regulada por leyes vigentes tanto nacionales como internacionales.

Es en este punto, donde la ayuda de alguna herramienta que permita tener un mejor control de flujo de la información puede ser la diferencia entre un proveedor y otro. Esta herramienta puede tener un efecto más evidente al facilitar un control centralizado con un suministro descentralizado, es decir, se puede dar el caso de tener que proveer productos a regiones distantes del centro de almacenaje y por tanto del de producción, por lo que es necesario descentralizar el o los almacenes con los que se cuente, para garantizar una entrega oportuna. En la figura 1.1.3.1 se muestra el flujo de productos para un proveedor.

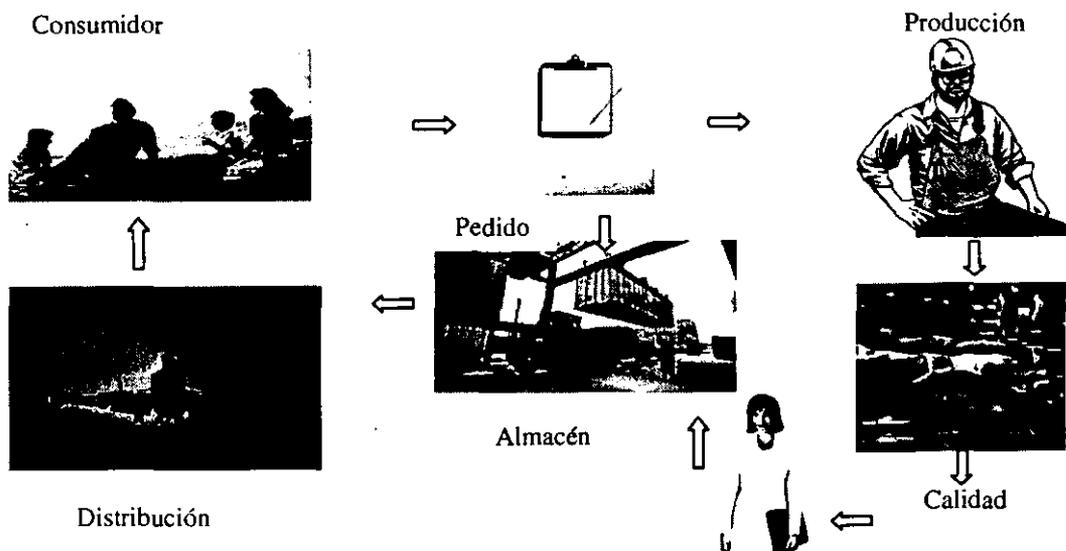


Figura 1.1.3.1. Flujo de productos para un proveedor.

También es evidente el manejo de las normas de suministros, las cuales se aplicarían para cualquier centro de distribución. No se quiere decir que este tipo de control es el que más conviene, ya que también puede tener sus defectos, por dirigir todos los pedidos hacia un punto

central que puede estar sujeto a fallas. Sin embargo ambos esquemas pudieran complementarse para obtener el mayor beneficio, ajustándose en grupos de insumos que se apeguen más a cada uno.

La otra parte que funge como solicitante y consumidora de los productos que el proveedor le proporciona, con base a una compra, es la encargada de solicitar un pedido que atienda a sus necesidades de abastecimiento y control interno. Hay que hacer notar que dentro de las mismas se encuentran implícitos una serie de departamentos que pueden existir o no como entidades una dentro de la otra. Como el de comprar que sería el punto de contacto indicado para realizar un pedido, de un almacén. Así aunque sabemos que el proceso de aprovisionamiento involucra el trabajo conjunto de una serie de departamentos los agruparemos a todos para hacer referencia a ellos como una sola parte, pero sin perder de vista las funciones propias de cada uno de ellos. Por lo que tenemos que mencionar primeramente a uno de los más importantes y que es el almacén, ya que éste tiene como responsabilidad vigilar el consumo de los artículos en general y sobre todo de aquellos cuya importancia sea evidente, de esta forma se puede llegar a detectar un consumo inadecuado ya sea por un pedido fuera de lo normal o un consumo desmedido dentro del mismo almacén. Tiene también la necesidad de realizar las requisiciones necesarias para mantener las reservas por encima de lo mínimo reglamentario.

En lo que respecta al departamento de compras, resalta su operación en forma centralizada. Ya que debe de converger hacia un lugar donde se lleva el control de los recursos financieros de la entidad. Mantiene además un inventario físico de los equipos, se encarga de revisar el almacenamiento o tiene conocimiento del mismo y algo muy importante, señala las especificaciones para la compra de materiales; entre otras cosas. Este departamento basa todas sus operaciones en órdenes de pedido que le hacen llegar de otros departamentos, las cuales una vez autorizadas pueden retornarse como copia al departamento que las ordenó, otra copia permanece en el mismo departamento de compras y el original es el que se manda al proveedor. Una vez que se recibió la mercancía en los términos especificados es turnada la factura al departamento de contabilidad para su registro y por supuesto su pago. Como se dijo con anterioridad, los departamentos tienen funciones propias aunque necesariamente ligados uno del otro. Es evidente establecer una política para el suministro, que sea lo más clara posible. En la figura 1.1.3.2 se muestra la división de proveedores que puede contemplar un Consumidor.

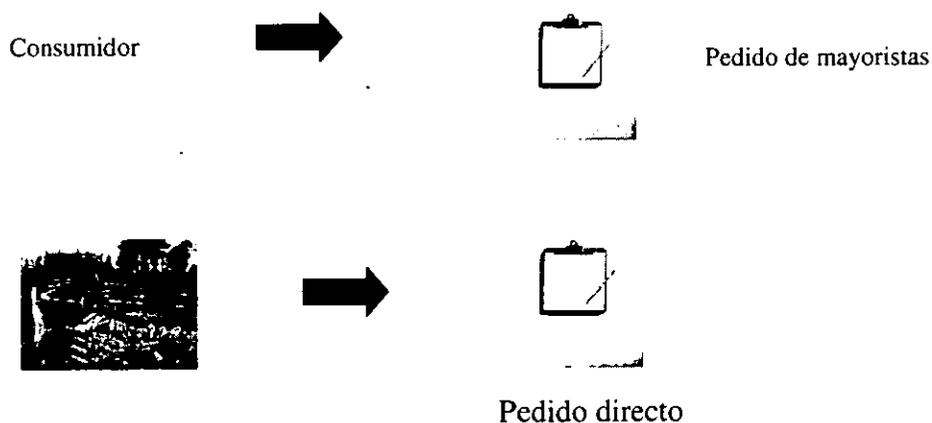


Figura 1.1.3.2. División de proveedores que puede contemplar un Consumidor.

1.1.4. Control de producción.

Los productos que han sido almacenados incorrectamente no se pueden rescatar para volver a usar.

Se debe de tener un especial cuidado con la información manejada durante el proceso de elaboración de los productos, ya que la mezcla de los productos es delicada y estrictamente confidencial, por ser marca registrada, el cuidado en la cantidad exacta de cada uno de los materiales repercute directamente en la calidad de los productos.

Es necesario establecer y mantener procedimientos apropiados para asegurar la total adherencia a los pertinentes códigos nacionales e internacionales para revisar y controlar todas sus actividades y materiales de promoción.

El personal encargado del control de la producción deberá estar adecuadamente capacitado y poseer suficientes conocimientos técnicos y experiencia en los procesos para presentar la información sobre los productos de manera precisa, responsable y ética.

1.1.5. Control Monitor.

Como en gran parte de las empresas el control de los estudios de poscomercialización y de vigilancia deberán ser llevados a cabo dentro de las normas establecidas para este fin. Deberá ser comunicada con prioridad a la autoridad sanitaria nacional competente toda información sustentada en datos, sobre reacciones adversas o inesperadas asociadas con los productos.

La promoción de los productos deberá realizarse de acuerdo a las reglas establecidas en la localidad, cumpliendo con lo establecido en las normas de la compañía.

1.2. CONCEPTOS FUNDAMENTALES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.

1.2.1. Que es la programación orientada a objetos.

La Programación Orientada a Objetos (POO) es una forma de programación que utiliza *objetos*, ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización de código. El objetivo de la POO es hacer de la programación algo más fácil. Como los usuarios exigen programas que tengan más características y programas con mayor complejidad, la necesidad de disponer de mejores herramientas para programar se hace cada día más importante, es aquí donde los lenguajes orientados a objetos entran en juego.

Los mecanismos básicos de la programación orientada a objetos son: *objetos*, *mensajes*, *métodos* y *clases*.

El futuro de la programación orientada a objetos luce muy prometedor por el énfasis incrementado respecto a distribución, la cual consiste, en poder trabajar partes de una misma aplicación en varios procesadores dentro de una red de computadoras y sistemas abiertos, donde el concepto de objetos, resulta ser el apoyo natural para tales aplicaciones.

1.2.2. El objeto.

Un programa tradicional se compone de procedimientos y datos. Un programa orientado a objetos consiste solamente en *objetos*. Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Dicho de otra forma, un *objeto* es una entidad que tiene atributos particulares, *propiedades*, y formas de operar sobre ellos, así como los *métodos*. Por lo tanto, un objeto contiene, por una parte, operaciones que definen su comportamiento, y por otra, variables manipuladas por esas operaciones que definen su estado.

El *estado de un objeto* abarca todas las propiedades del objeto, y los valores actuales de cada una de sus propiedades. Las propiedades de los objetos suelen ser estáticas, los valores de las propiedades cambian con el tiempo.

El cambio en el estado de un objeto es lo que se conoce como *evento*. Un evento ocurre cada vez que un sistema orientado a objetos y un sistema externo u operador intercambian información.

1.2.3. Ejemplo de objeto.

Como ejemplo de un objeto esta una ventana del sistema operativo Windows. El color de fondo de la ventana, el ancho, el alto, etc. son propiedades. Las rutinas, lógicamente transparentes al usuario, que permiten maximizar la ventana, minimizarla, etc. son métodos.

Algunos ejemplos de objetos pueden verse en la forma de manifestación de estos, mostrada en la figura 1.2.3.1. Los objetos pueden ser:

- Eventos u ocurrencias (ej. La terminación de una serie de movimientos de un robot) que ocurren dentro del contexto de operación del sistema.
- Cosas (ej. Cartas, informes) que son parte del dominio de la información del problema.
- Entidades externas (ej. Dispositivos, personas, otros sistemas) que producen o consumen información.
- Papeles o roles (ej. Director, ingeniero, vendedor) desempeñados por personas que interactúan con el sistema.
- Unidades organizacionales (ej. División, grupo, equipo) que son relevantes en una aplicación.
- Lugares (ej. Planta de producción, muelle de carga) que establece el contexto del problema y la función general del sistema.
- Estructuras (ej. Sensores, vehículos de cuatro ruedas, computadoras) que definen una clase de objetos o, en casos externos, clases relacionadas de objetos.

COMO SE MANIFIESTAN LOS OBJETOS

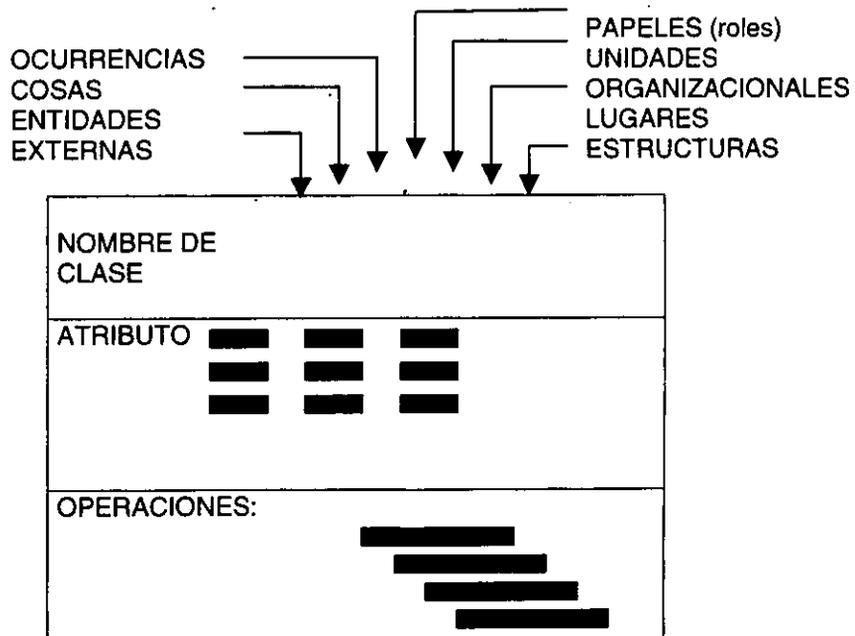


Figura 1.2.3.1. Manifestación de los objetos.

1.2.4. Métodos y mensajes.

Un *método* se implementa en una *clase* de objetos y determina cómo tiene que actuar el objeto cuando recibe un *mensaje*. Un *método* puede también enviar *mensajes* a otros objetos solicitando una acción o información.

Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo, interpretando y respondiendo a *mensajes* de otros objetos. Esto marca una clara diferencia con respecto a los elementos de datos pasivos de los sistemas tradicionales.

1.2.5. Clases.

Una *clase* es un tipo de objeto definido por el usuario. Una *clase* equivale a la generalización de un tipo específico de objetos. Dos de las características de la POO, la *abstracción* y la *encapsulación* están representadas por la *clase*. La *clase* es una **abstracción** porque en ella se definen las propiedades de un determinado conjunto de objetos con características comunes, y es una **encapsulación**, porque constituye una caja negra que encierra tanto los datos de que constan los objetos como los métodos que permiten manipularlos.

Una clase **encapsula** las abstracciones de datos y procedimientos que se requieren para describir el contenido y comportamiento de alguna entidad del mundo real. Especifica la estructura de datos y los métodos operativos permitidos que se aplican a cada uno de sus objetos.

Dentro de la estructura de clases se encuentran la superclase (colección de clases) y la subclase (instancias de clases).

Todos los objetos que existen dentro de una clase heredan sus atributos y las operaciones disponibles para su manipulación. Estas definiciones implican la existencia de una **jerarquía** (clasificación u ordenación de la relación que se produce entre clases) en la cual los atributos y operaciones de la superclase son heredadas por subclases que pueden añadir, cada una de ellas, atributos "privados" y métodos.

1.2.6. Estructura interna de un objeto.

La estructura más interna de un *objeto*, descrita en la figura 1.2.6.1, está oculta para otros usuarios y la única conexión que tiene con el exterior son los *mensajes*. Los datos que están dentro de un objeto solamente pueden ser manipulados por los *métodos* asociados al propio objeto.

Una característica importante de la programación orientada a objetos es la **encapsulación**, esta característica es la que permite que la estructura interna de un objeto este oculta, es decir, es lo que le da a los objetos su valor como elementos fundamentales de construcción. La **encapsulación** proporciona dos características importantes:

1. Coloca a los datos y a las funciones bajo un mismo techo.
2. Proporciona el recurso operacional de ocultar los datos.

En nuestra vida diaria confiamos constantemente en el principio de encapsulación, particularmente cuando está relacionada con máquinas complejas. Por ejemplo, conducir un automóvil. Cuando se está conduciendo, no se necesita saber si el automóvil tiene tracción delantera o trasera. Sólo se necesita saber que cuando se presiona el pedal del acelerador, el automóvil se mueve.

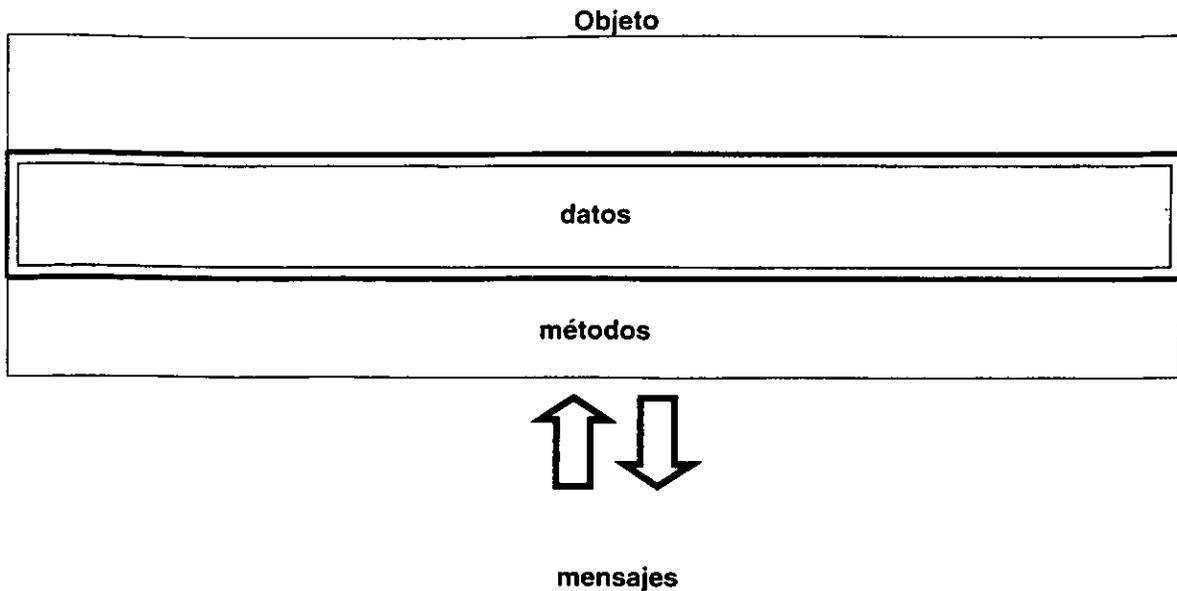


Figura 1.2.6.1. Estructura interna de un objeto.

Cuando se utilizan las técnicas de Programación Orientada a Objetos (POO) en los programas, el mismo principio se aplica. Por ejemplo, si se desarrolla un objeto para la base de datos, realmente no se quiere saber cómo está almacenado el objeto en el sistema de base de datos. Lo único que se quiere saber es qué rutinas se deben llamar para tener acceso al objeto.

Por lo general, la **encapsulación** tiene tres propósitos. Primero, protege a los datos de un objeto evitando que tengan demasiada exposición, ya que generalmente los datos sólo se accesan a través de las funciones que están definidas dentro del objeto. Segundo, la encapsulación hace más fácil utilizar los datos del objeto ya que todas las acciones tienen lugar a través de una interfaz bien definida (nuevamentepor medio de las operaciones asociadas con el objeto). Finalmente, la encapsulación también puede ser utilizada para ocultar los detalles de cómo están almacenados o instrumentados los datos.

1.2.7. Herencia.

La **herencia** es el mecanismo para compartir automáticamente métodos y datos entre clases y subclases de objetos, es decir, permite derivar una nueva clase de una clase existente.

La **herencia de clase** hace que la estructura de datos y operaciones sean disponibles para su reutilización por parte de sus subclases. La herencia de las operaciones de una superclase permite que las clases compartan el código en lugar de volverlo a definir. La herencia de estructura de datos permite la reutilización de la estructura.

Con esto se puede:

- Añadir datos y código a una clase sin tener que cambiar la clase original.
- Reutilizar el código.
- Modificar el comportamiento de una clase.

Herencia simple. Es aquella en la que una clase puede heredar la estructura de datos y operaciones de una superclase (Figura 1.2.7.1).

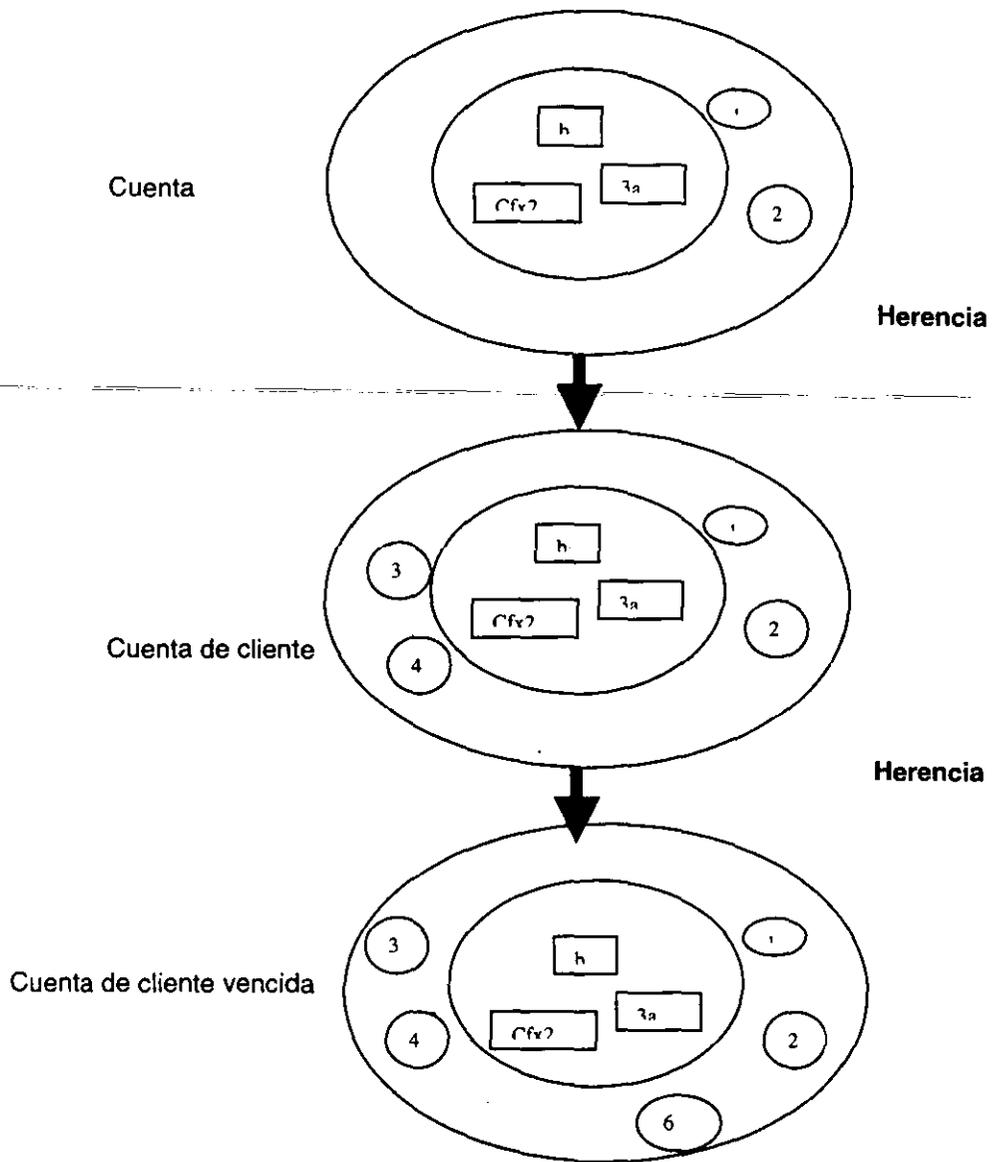


Figura 1.2.7.1. Herencia. La clase Cuenta de cliente hereda los métodos 1 y 2 de la clase Cuenta. Cuenta de cliente tiene sus propios métodos 3 y 4. La clase Cuenta de cliente vencida hereda los métodos 1, 2, 3 y 4 de Cuenta de cliente, además de tener su propio método 6.

Herencia múltiple. Una clase puede heredar estructuras de datos y operaciones de más de una superclase (Figura 1.2.7.2).

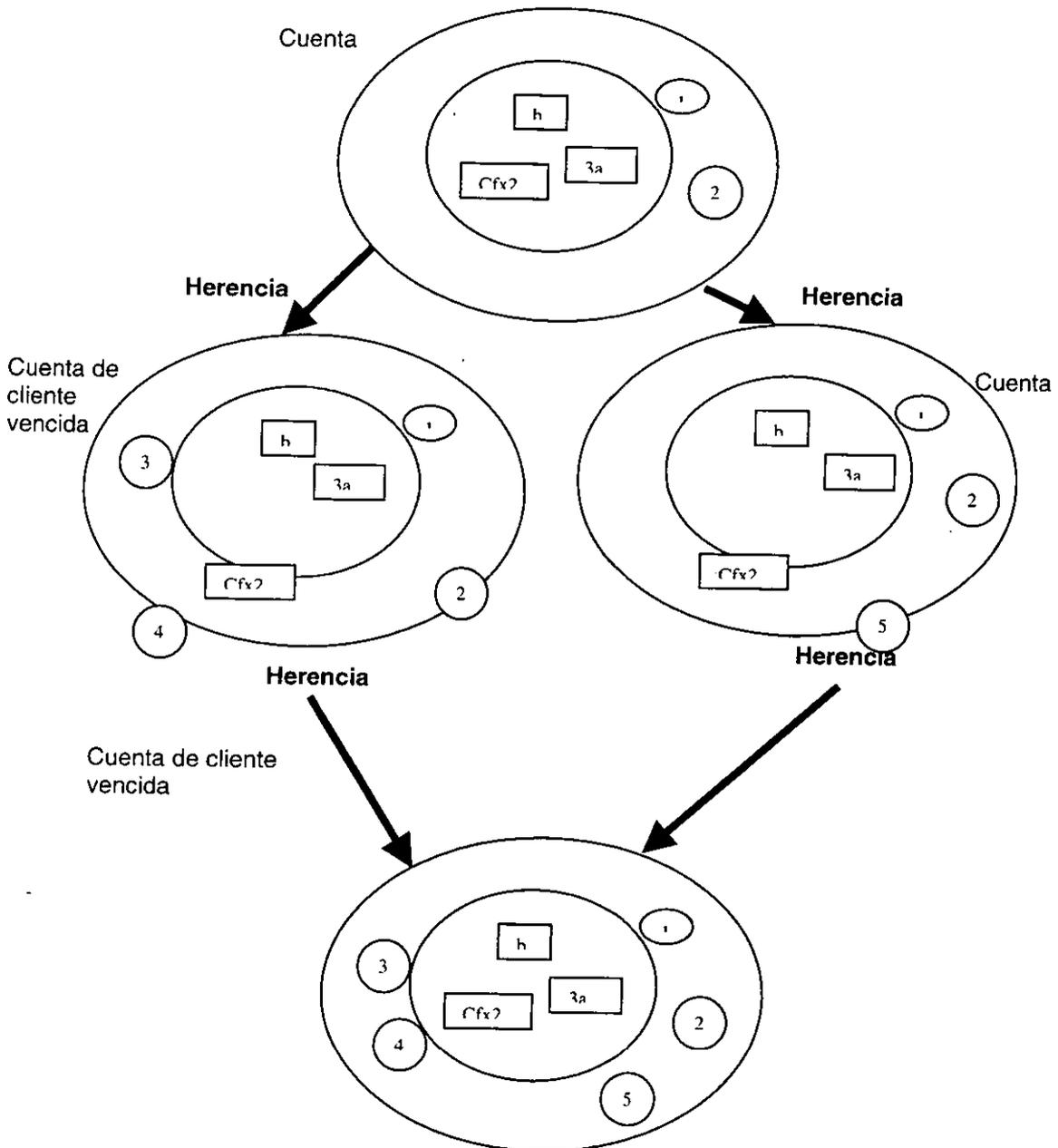


Figura 1.2.7.2. Las tres clases inferiores heredan los métodos de Cuenta. Cuenta de cliente vencida hereda los métodos de Cuenta de cliente y Cuenta vencida.

1.2.8. Ligadura dinámica.

El proceso que se utiliza para vincular una llamada de función con el código actual para esa función se denomina *función de ligadura*. En un procedimiento normal o en una llamada de función, el compilador determina la dirección de la rutina y la utiliza para crear un salto hasta el código de la rutina. Ya que esto se hace estáticamente en la fase de compilación, el proceso se llama *ligadura estática* o *ligadura temprana*.

Cuando una función miembro normal se llama a través del puntero de un objeto, el compilador se fija en el tipo de puntero para determinar que función llamar. Aun cuando se involucran punteros, la función de ligadura todavía se determina en la fase de compilación, de modo que la ligadura es estática. En contraste, las funciones utilizan la *ligadura dinámica* toda vez que ellas han sido declaradas virtuales. Por lo tanto, la dirección de la función se encuentra mediante una búsqueda que se realiza durante el tiempo de ejecución. Cada objeto tiene un puntero a una tabla y, si los objetos pertenecen a diferentes tipos, los punteros apuntan a tablas diferentes.

La decisión de utilizar la ligadura estática o la ligadura dinámica no está basada en si se utilizan o no los punteros a objetos. En su lugar, la decisión se hace basada en si se utilizan o no las funciones. Con las funciones virtuales, el mismo objeto siempre determina qué función obtiene la llamada. Esto le permite al polimorfismo trabajar correctamente aun cuando están involucradas asignaciones dudosas de puntero.

Para lograr la máxima flexibilidad con una clase, todas sus funciones miembro deben ser declaradas como virtuales. De esta forma, cualquier clase derivada que quiera omitir una función puede hacerlo fácilmente, sin necesidad de modificar el código de la clase base. Además, el proceso de omisión funcionará en ambas especificaciones: estáticas y dinámicas.

1.2.9. Polimorfismo.

Esta característica permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto hace que se pueda acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando exactamente el mismo medio de acceso.

Un aspecto importante del polimorfismo radica en que cada objeto en la familia puede tener métodos con los mismos nombres, pero el código para los métodos de cada objeto puede ser totalmente diferente. La figura 1.2.9.1 resume cómo opera el polimorfismo. Cuando el controlador de tráfico aéreo de la figura envía el mensaje "suba y mantenga 10 000 pies", el comportamiento de cada uno de los objetos es diferente. El controlador de tráfico aéreo que envía el mensaje realmente no sabe qué clase de objeto está recibiendo el mensaje. Tampoco sabe si el OVNI tiene un motor o si obtiene su potencia mediante pedales. El controlador sólo espera que sus instrucciones se cumplan, independientemente de quien las reciba.



Figura 1.2.9.1. Ejemplo de polimorfismo

1.2.10. Reutilización con orientación a objetos.

La reutilización del código es la capacidad de trasladar características de un objeto a otro, lo que se logra con alguna forma de herencia. En una jerarquía de clases, la herencia muestra cómo los objetos que se derivan de otros objetos más simples heredan su comportamiento.

Como muestra la figura 1.2.10.1, podemos utilizar la línea de ensamblaje de automóviles básica para derivar otras líneas de ensamblaje para fabricar un automóvil extravagante. Estas líneas de ensamblaje demuestran cómo trabaja la herencia porque le permiten al fabricante del automóvil ensamblar varios automóviles específicos a partir de un automóvil genérico como base. Por supuesto, el fabricante del automóvil realmente se beneficia de este proceso debido a que no tiene que comprometer extensos recursos para construir tres automóviles diferentes.

El punto importante a recordar del ejemplo de una línea de ensamblaje es que se está utilizando parte de una línea de ensamblaje para crear otras líneas de ensamblaje. El compartir los recursos de esta manera conduce hacia mejores herramientas y hacia productos más lucrativos. Esto mismo es lo que se está haciendo con los programas orientados a objetos, esto es, se construyen clases que se reutilizan de manera que no se tenga que rediseñar cada vez que se escriba una nueva aplicación.

Otra característica útil de la herencia involucra la habilidad de permitir modificar una clase de un objeto existente para hacer una nueva clase que tenga personalidad diferente. De este modo, se puede utilizar la herencia para crear múltiples objetos que desempeñen diferentes acciones aun cuando los objetos hayan sido derivados de la misma existencia.

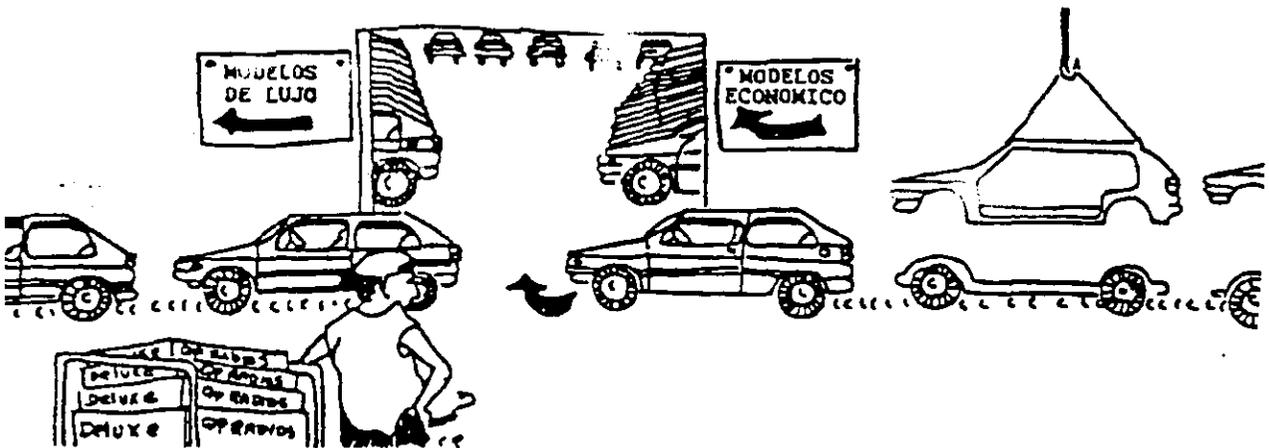


Figura 1.2.10.1. La fabricación en las líneas de ensamblaje a menudo se aprovecha para compartir recursos.

1.3. MODELADO DE OBJETOS.

Un programa completo consta siempre de dos partes, de un programa y de los datos sobre los que actúan estos programas. Según la forma en que se organicen los datos y programas, se distinguen diferentes formas de programar: programación procedimental (imperativo), programación lógica (basado en reglas) y programación orientada a objetos.

En la década de los ochenta, surgió un nuevo conjunto de técnicas que recibe el nombre de Programación Orientada a Objetos (P.O.O.). En esta forma de programar los datos forman la jerarquía básica.

La POO viene a ser una forma de programar opuesta a la programación clásica, en el sentido de que los datos y los programas desempeñan papeles diametralmente opuestas. En la programación procedimental, los programas están organizados mientras los datos no lo están. En la POO, son los datos los que forman una organización, pero no ocurre lo mismo con los programas. A su vez, ambas formas de programas se oponen a la programación lógica, donde ni los datos ni los programas tienen por qué estar organizados.

La programación comprende las técnicas de desarrollo formal para el diseño de software. Problemas frecuentes del software tales como: Aumento de la complejidad, cambios continuos, no confiable, dificultad para verificarlo, dificultad para especificar los requerimientos. Estas son algunas de las razones por la cual la programación orientada a objetos sea tan popular.

1.3.1. Metodología del desarrollo de sistemas de información.

La metodología del desarrollo de sistemas es el camino que siguen los analistas de sistemas al realizar su trabajo (Burch y Grudnitsky, 1996, pág. 73). Se emplea el término genérico de analista de sistemas para describir a la persona que tiene la responsabilidad principal de conjuntar los componentes estructurales, dándoles forma y sustancia en conformidad con las fuerzas del diseño para construir sistemas de información exitosos.

Fases principales de la metodología del desarrollo de sistemas.

“Las fases principales son el análisis de sistemas, el diseño general de sistemas, la evaluación de sistemas, el diseño detallado de sistemas y la implementación de sistemas. Dentro de cada fase se incluyen las principales actividades o tareas. Las primeras cuatro fases están dirigidas a proporcionar valores específicos para los componentes estructurales. La última fase se ocupa de que los componentes estructurales sean operacionales. Algunos autores, compañías y consultores dividen su metodología de desarrollo de sistemas en fases o etapas diferentes a las presentadas aquí” (Burch y Grudnitsky, 1996, pág. 73). Mostrada en la figura 1.3.1.1.

“Independientemente del número o nombres de las fases o etapas, la metodología del desarrollo de sistemas racionaliza y asigna una rutina al proceso de construcción de sistemas de información. Su sello distintivo lo forman sus fases discretas. La meta principal de la metodología del desarrollo de sistemas es reducir los inicios falsos, reciclamiento indebido, retrabajos y callejones sin salida. Además, aumenta la probabilidad de que el sistema que se construya e instale finalmente sea el que los usuarios desean y necesitan. Pero de ninguna manera la metodología del desarrollo de sistemas deberá ahogar la creatividad o la innovación. Proporciona un faro guía para el recorrido a lo largo de un sistema de información” (Burch y Grudnitsky, 1996, pág. 73).

1.3.2. Definición de la metodología del desarrollo de sistemas.

Durante la fase del análisis de sistemas, los analistas de sistemas ayudan a los usuarios a identificar qué información se necesita. Se llevan a cabo varias entrevistas y se plantean preguntas como: ¿Qué información está usted recibiendo actualmente?, ¿Qué clase de información necesita para realizar su trabajo?, Sin embargo, en este punto los analistas de sistemas deben ser más que escuchas pasivos o mecanógrafos; deben ser líderes y maestros. Con frecuencia los usuarios no tienen una idea completa de cuál es la información que realmente necesitan o cómo pueden obtenerla. En la mayoría de los casos, los analistas deben interactuar y determinar con los usuarios cuáles son sus verdaderas necesidades. Hacia el final de la fase de análisis, los analistas conceptualizan diseños generales.

Antes de iniciar el diseño general de sistemas, la fase de análisis debe completarse y los usuarios deben estar de acuerdo con los resultados. El diseño es el proceso de traducir los

requerimientos definidos durante el análisis a varias alternativas de diseño para la consideración de los usuarios. También implica descubrir todas las fuerzas de diseño para ver cómo impactarán e influirán en los diferentes diseños.

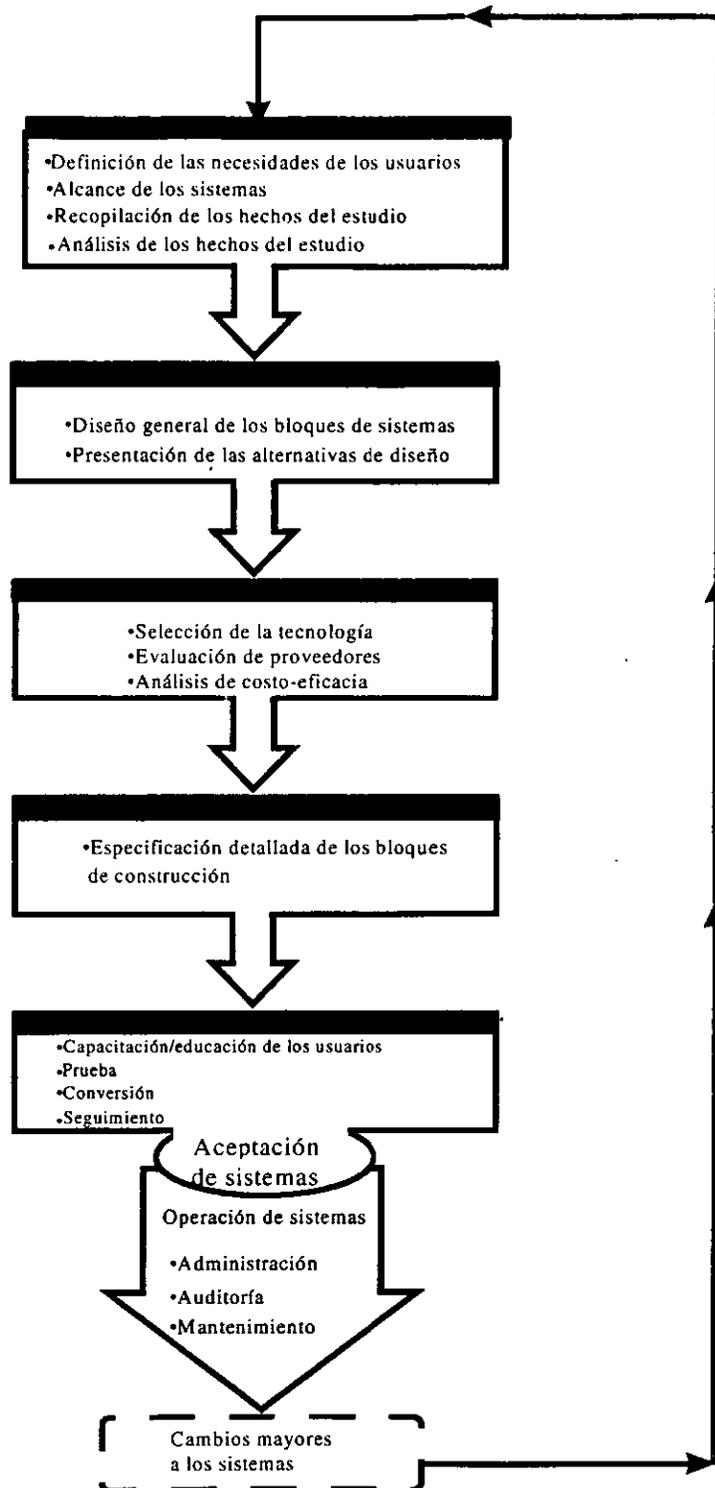


Figura 1.3.1.1. Fases de la Metodología del Desarrollo de Sistemas.
Fuente : (Buch y Grudnisty, 1996, pág. 74)

En la fase de diseño, los usuarios seleccionan dos o tres de los mejores diseños o le piden al analista que regrese al restirador para desarrollar un mejor diseño. Si se recopilaron y analizaron suficientes hechos del estudio en la fase de análisis, se reduce la probabilidad de tener que regresar a dicha fase. También se debe recalcar que durante el diseño los analistas de sistemas con frecuencia empezarán a interactuar con técnicos en administración, expertos en robótica, programadores, especialistas de formas y expertos en comunicaciones. En cualquier caso, los analistas concluyen la fase del diseño general con un entendimiento bastante claro de lo que se desea y de lo que tendrán que hacer para obtenerlo.

La evaluación de sistemas en realidad no es una fase discreta; en todo momento durante el trabajo en sistemas se realiza una gran cantidad de evaluación, pero para resaltarla, aquí se presenta como una fase separada. Esta fase implica la selección de la tecnología que soportará a los otros componentes estructurales, la evaluación de esta tecnología y los proveedores que la suministran, y un análisis completo de costo-eficacia de cada una de las alternativas propuestas del diseño de sistemas.

Obviamente, los analistas no desean tener que recorrer el camino largo y difícil hacia el diseño detallado de sistemas teniendo varias alternativas de diseño en el aire. Antes de iniciar este trabajo secuencial, los analistas desean asegurarse de que todas las partes hayan llegado a un consenso final en cuanto al sistema que desean que se implemente. Aquí, a cada componente de construcción se le da una definición precisa y detallada.

La implementación de sistemas es la capacitación y educación de los usuarios, la prueba y la conversión para hacer que el sistema sea operacional. Los analistas fijan las fechas límite, capacitan y coordinan al personal usuario, instruyen a los técnicos, prueban el nuevo sistema y eliminan errores, instalan nuevos procedimientos y formas, y ven si hay descuidos.

Poco después de su aceptación, el analista de sistemas deberá realizar un seguimiento para ver que el sistema está operando según lo esperado. En muchos casos, los analistas pueden hacer algunos ajustes menores que resulten en mejoras significativas. Después de que el sistema haya estado operando durante cierto tiempo, las necesidades de los usuarios cambiaron y los analistas de sistemas se verán involucrados en la preparación de prototipos especiales para cubrir necesidades particulares.

En ocasiones será necesario efectuar algún trabajo de mantenimiento. Finalmente en algún momento, varios años después de que el sistema haya estado en operación, el analista de sistemas se verá de nuevo fuertemente involucrado en un mantenimiento mayor del sistema, o en el desarrollo y diseño de un nuevo, y entonces se repetirá el ciclo de vida de la metodología del desarrollo de sistemas.

El papel de los analistas de sistemas y su empleo de la metodología del desarrollo de sistemas es, por supuesto, un cuadro compuesto. En una organización grande el sistema de información es una entidad vasta y compleja, y muchos analistas están involucrados en el diseño e implementación de sólo un subsistema de todo el sistema de información de la organización. En un momento dado, un analista podría muy bien estar realizando actividades relacionadas con más de una fase de la metodología del desarrollo de sistemas, y estas actividades se deben coordinar adecuadamente.

Para que el sistema sea más eficaz, el analista de sistemas participará en un amplio espectro de actividades. Al realizar éstas, el analista puede hacer uso de diversas herramientas de apoyo, tales como diagramas de flujo de datos, diagramas UML, diagramas Entidad-Relación, diccionarios de datos, gráficas, entrevistas y prototipos, entre otros. Estas técnicas se refuerzan entre sí y, cuando se ven en conjunto, proporcionan las herramientas básicas para el trabajo en sistemas.

1.3.3. Lenguaje Unificado para Modelado.

El Lenguaje Unificado para Modelado(UML) es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de *software* orientado a objetos. Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de *software*.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelado como OMT (Técnicas de Modelado de Objetos) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo. Las diferencias son muy marcadas y afectan a todas las fases del proceso.

Los Inicios. A partir del año 1994, Grady Booch (precursor de Booch '93) y Jim Rumbaugh (creador de OMT) se unen en una empresa común, Rational *Software* Corporation, y comienzan a unificar sus dos métodos (Lafuente, 1999, Introducción al uml). Un año más tarde, en octubre de 1995, aparece UML 0.8, la que se considera como la primera versión del UML. A finales de ese mismo año, Ivar Jacobson, creador de OOSE (Ingeniería de *Software* Orientada a Objetos) se añade al grupo. Como objetivos principales de la consecución de un nuevo método que aunara los mejores aspectos de sus predecesores, sus protagonistas se propusieron lo siguiente:

- El método debía ser capaz de modelar no sólo sistemas de *software* sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos.
- Crear un lenguaje para modelado utilizable a la vez por máquinas y por personas.
- Establecer un acoplamiento explícito de los conceptos y los artefactos ejecutables.
- Manejar los problemas típicos de los sistemas complejos de misión crítica.

Lo que se intenta es lograr con esto que los lenguajes que se aplican siguiendo los métodos más utilizados sigan evolucionando en conjunto y no por separado. Y además, unificar las perspectivas entre diferentes tipos de sistemas (no sólo *software*, sino también en el ámbito de los negocios), al aclarar las fases de desarrollo, los requerimientos de análisis, el diseño, la implementación y los conceptos internos de la programación orientada a objetos.

1.3.4. Modelado de objetos con UML.

En la especificación del UML podemos comprobar que una de las partes que lo componen es un metamodelo formal. Un metamodelo es un modelo que define el lenguaje para expresar otros modelos. Un modelo orientado a objetos es una abstracción cerrada semánticamente de un sistema y un sistema es una colección de unidades conectadas que son organizadas para realizar un propósito específico. Un sistema puede ser descrito por uno o más modelos, posiblemente desde distintos puntos de vista.

Una parte del UML define, entonces, una abstracción con significado de un lenguaje para expresar otros modelos (es decir, otras abstracciones de un sistema, o conjunto de unidades conectadas que se organizan para conseguir un propósito). Lo que en principio puede parecer complicado no lo es tanto si pensamos que uno de los objetivos del UML es llegar a convertirse en una manera de definir modelos, no sólo establecer una forma de modelo, de esta forma simplemente estamos diciendo que UML, además, define un lenguaje con el que podemos abstraer cualquier tipo de modelo (Lafuente, 1999, modelados).

El UML es un lenguaje de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación. Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.

Los modelos se utilizan en muchas actividades de la vida humana: antes de construir una casa el arquitecto utiliza un plano, los músicos representan la música en forma de notas musicales, etc. Unos y otros abstraen una realidad compleja sobre unos bocetos, modelos al fin y al cabo. La OMT, por ejemplo, intenta abstraer la realidad utilizando tres clases de modelos OO: el modelo de objetos, que describe la estructura estática; el modelo dinámico, con el que describe las relaciones temporales entre objetos; y el modelo funcional que describe las relaciones funcionales entre valores. Mediante estas tres fases de construcción de un modelos, se consigue una abstracción de la realidad que tiene en sí misma información sobre las principales características de ésta.

Los modelos además, al no ser una representación que incluya todos los detalles de los originales, permiten probar más fácilmente los sistemas que modelan y determinar los errores. Según se indica en la Metodología OMT(Lafuente, 1999, modelado), los modelos permiten una mejor comunicación con el cliente por distintas razones:

- Es posible enseñar al cliente una posible aproximación de lo que será el producto final.
- Proporcionan una primera aproximación al problema que permite visualizar cómo quedará el resultado.
- Reducen la complejidad del original en subconjuntos que son fácilmente tratables por separado.

Se consigue un modelo completo de la realidad cuando el modelo captura los aspectos importantes del problema y omite el resto. Los lenguajes de programación que estamos acostumbrados a utilizar no son adecuados para realizar modelos completos de sistemas reales porque necesitan una especificación total con detalles que no son importantes para el algoritmo que están implementando. En OMT se modela un sistema desde tres puntos de vista diferentes donde cada uno representa una parte del sistema y una unión lo describe de forma completa. En esta técnica de modelado se utilizó una aproximación al proceso de implementación de *software* habitual donde se utilizan estructuras de datos (modelo de objetos), las operaciones que se realizan con ellos tienen una secuencia en el tiempo (modelo dinámico) y se realiza una transformación sobre sus valores (modelo funcional).

UML utiliza parte de este planteamiento obteniendo distintos puntos de vista de la realidad que modela mediante los distintos tipos de diagramas que posee. Con la creación del UML se persigue obtener un lenguaje que sea capaz de abstraer cualquier tipo de sistema, sea informático o no, mediante los diagramas, es decir, mediante representaciones gráficas que contienen toda la información relevante del sistema. Un diagrama es una representación gráfica de una colección de elementos del modelo, que habitualmente toma forma de gráfica donde los arcos que conectan sus vértices son las relaciones entre los objetos y los vértices se corresponden con los elementos del modelo. Los distintos puntos de vista de un sistema real que se quieren representar para obtener el modelo se dibujan de forma que se resalten los detalles necesarios para entender el sistema.

Artefactos para el Desarrollo de Proyectos.

Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de *software*. Pueden ser artefactos un modelo, una descripción o un *software*. Los artefactos de UML se especifican en forma de diagramas, éstos, junto con la documentación sobre el sistema constituyen los artefactos principales que el modelador puede observar.

Se necesita más de un punto de vista para llegar a representar un sistema. UML utiliza los diagramas gráficos para obtener estos distintos puntos de vista de un sistema(Lafuente, 1999, diagramas del uml):

- Diagramas de Casos de Uso.

- Diagramas de Clases.
- Diagramas de Comportamiento o Interacción.
- Diagramas de Implementación.

Nuevas características del UML.

Además de los conceptos extraídos de métodos anteriores, se han añadido otros nuevos que vienen a suplir carencias antiguas de la metodología de modelado. Estos nuevos conceptos son los siguientes:

- Definición de estereotipos: un estereotipo es una nueva clase de elemento de modelado que debe basarse en ciertas clases ya existentes en el metamodelo y constituye un mecanismo de extensión del modelo.
- Responsabilidades.
- Mecanismos de extensibilidad: estereotipos, valores etiquetados y restricciones.
- Tareas y procesos.
- Distribución y concurrencia (para modelar por ejemplo ActiveX/DCOM y CORBA).
- Patrones/Colaboraciones.
- Diagramas de actividad (para reingeniería de proceso de negocios)
- Clara separación de tipo, clase e instancia.
- Refinamiento (para manejar relaciones entre niveles de abstracción).
- Interfaces y componentes.

El Proceso de Desarrollo.

UML no define un proceso concreto que determine las fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que utilicen UML serán los tipos de diagramas. UML es un método independiente del proceso. Los procesos de desarrollo deben ser definidos dentro del contexto donde se van a implementar los sistemas. A continuación se describen los diagramas utilizados en uml(Lafuente, 1999, descripción de diagramas del uml).

Diagramas de Casos de Uso.

Un diagrama de casos de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas. O lo que es igual , un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo.

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar como reacciona una respuesta a eventos que se producen en el mismo. En este tipo de diagrama intervienen algunos conceptos nuevos: un actor es una entidad externa al sistema que

se modela y que puede interactuar con él; un ejemplo de actor podría ser un usuario o cualquier otro sistema. Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un caso de uso.
- Un caso de uso extiende otro caso de uso.
- Un caso de uso usa otro caso de uso.

Diagramas de Clases.

Los diagramas de clases representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos. Algunos de los elementos que se pueden clasificar como estáticos son los siguientes:

Paquete: Es el mecanismo de que dispone UML para organizar sus elementos en grupos, se representa un grupo de elementos del modelo. Un sistema es un único paquete que contiene el resto del sistema, por lo tanto, un paquete debe poder anidarse, permitiéndose que un paquete contenga otro paquete.

Clases: Una clase representa un conjunto de objetos que tienen una estructura, un comportamiento y unas relaciones con propiedades parecidas. Describe un conjunto de objetos que comparte los mismos atributos, operaciones, métodos, relaciones y significado. En UML una clase es una implementación de un tipo. Los componentes de una clase son:

Atributo. Se corresponde con las propiedades de una clase o un tipo. Se identifica mediante un nombre. Existen atributos simples y complejos.

Operación. También conocido como método. Es un servicio proporcionado por la clase que puede ser solicitado por otras clases y que produce un comportamiento en ellas cuando se realiza.

Las clases pueden tener varios parámetros formales, son las clases denominadas plantillas. Sus atributos y operaciones vendrán definidas según sus parámetros formales. Las plantillas pueden tener especificados los valores reales para los parámetros formales, entonces reciben el nombre de clase parametrizada instanciada. Se puede usar en cualquier lugar en el que se podría aparecer su plantilla.

Relacionado con las clases nos encontramos con el término utilidad, que se corresponde con una agrupación de variables y procedimientos globales en forma de declaración de clase, también puede definirse como un estereotipo (o nueva clase generada a partir de otra ya existente) de un tipo que agrupa variables globales y procedimientos en una declaración de clase. Los atributos y operaciones que se agrupan en una utilidad se convierten en variables y operaciones globales. Una utilidad no es fundamental para el modelado, pero puede ser conveniente durante la programación.

Metaclase: Es una clase cuyas instancias son clases. Sirven como depósito para mantener las variables de clase y proporcionan operaciones (método de clase) para inicializar estas variables. Se utilizan para construir metamodelos (modelos que se utilizan para definir otros modelos).

Tipos: Es un descriptor de objetos que tiene un estado abstracto y especificaciones de operaciones pero no su implementación. Un tipo establece una especificación de comportamiento para las clases.

Interfaz: Representa el uso de un tipo para describir el comportamiento visible externamente de cualquier elemento del modelo.

Relación entre clases: Las clases se relacionan entre sí de distintas formas, que marcan los tipos de relaciones existentes:

Asociación:

Es una relación que describe un conjunto de vínculos entre clases. Pueden ser binarias o n-arias, según se implican a dos clases o más. Las relaciones de asociación vienen identificadas por los roles, que son los nombres que indican el comportamiento que tienen los tipos o las clases, en el caso del rol de asociación (existen otros tipos de roles según la relación a la que identifiquen). Indican la información más importante de las asociaciones. Es posible indicar el número de instancias de una clase que participan en una relación mediante la llamada multiplicidad. Cuando la multiplicidad de un rol es mayor que 1, el conjunto de elementos que se relacionan pueden estar ordenado. Las relaciones de asociación permiten especificar qué objetos van a estar asociados con otro objeto mediante un calificador. El calificador es un atributo o conjunto de atributos de una asociación que determina los valores que indican cuáles son los valores que se asociarán. Una asociación se dirige desde una clase a otra (o un objeto a otro), el concepto de navegabilidad se refiere al sentido en el que se recorre la asociación. Existe una forma especial de asociación, la agregación, que especifica una relación entre las clases donde el llamado "agregado" indica el todo y el "componente" es una parte del mismo.

Composición:

Es un tipo de agregación donde la relación de posesión es tan fuerte como para marcar otro tipo de relación. Las clases en UML tienen un tiempo de vida determinado, en las relaciones de composición, el tiempo de vida de la clase que es parte del todo (o agregado) viene determinado por el tiempo de vida de la clase que representa el todo, por tanto es equivalente a un atributo, aunque no lo es porque es una clase y puede funcionar como tal en otros casos.

Generalización:

Cuando se establece una relación de este tipo entre dos clases, una es una Superclase y la otra es una Subclase. La subclase comparte la estructura y el comportamiento de la superclase. Puede haber más de una clase que se comporte como subclase.

Dependencia:

Una relación de dependencia se establece entre clases (u objetos) cuando un cambio en el elemento independiente del modelo puede requerir un cambio en el elemento dependiente.

Relación de Refinamiento:

Es una relación entre dos elementos donde uno de ellos especifica de forma completa al otro que ya ha sido especificado con cierto detalle.

Diagramas de Interacción o Comportamiento.

Muestran las interacciones entre objetos ocurridas en un escenario (parte) del sistema. Hay varios tipos:

- Diagrama de secuencia

- Diagrama de colaboración
- Diagrama de estado

Diagrama de secuencia.

Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. En los diagramas de este tipo intervienen objetos, que tienen un significado parecido al de los objetos representados en los diagramas de colaboración, es decir son instancias concretas de una clase que participa en la interacción. El objeto puede existir sólo durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción. Un diagrama de secuencia representa una forma de indicar el período durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.

En este tipo de diagramas también intervienen los mensajes, que son la forma en que se comunican los objetos: el objeto origen solicita (llama a) una operación del objeto destino. Existen distintos tipos de mensajes según cómo se producen en el tiempo: simples, síncronos, y asíncronos. Los diagramas de secuencia permiten indicar cuál es el momento en el que se envía o se completa un mensaje mediante el tiempo de transición, que se especifica en el diagrama.

Diagrama de colaboración.

Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. A diferencia de un diagrama de secuencias, un diagrama de colaboraciones muestra las relaciones entre los objetos, no la secuencia en el tiempo en que se producen los mensajes. Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en una forma diferente.

Formando parte de los diagramas de colaboración nos encontramos con objetos, enlaces y mensajes. Un objeto es una instancia de una clase que participa como una interacción, existen objetos simples y complejos. Un objeto es activo si posee un hilo de control y es capaz de iniciar la actividad de control, mientras que un objeto es pasivo si mantiene datos pero no inicia la actividad.

Un enlace es una instancia de una asociación que conecta dos objetos de un diagrama de colaboración. El enlace puede ser reflexivo si conecta a un elemento consigo mismo. La existencia de un enlace entre dos objetos indica que puede existir un intercambio de mensajes entre los objetos conectados.

Los diagramas de interacción indican el flujo de mensajes entre elementos del modelo, el flujo de mensajes representa el envío de un mensaje desde un objeto a otro si entre ellos existe un enlace. Los mensajes que se envían entre objetos pueden ser de distintos tipos, también según como se producen en el tiempo. Durante la ejecución de un diagrama de colaboración se crean y destruyen objetos y enlaces.

Diagramas de estado.

Representan la secuencia de estados por los que un objeto o una interacción entre objetos pasa durante su tiempo de vida en respuesta a estímulos (eventos) recibidos. Representa lo que podemos denominar en conjunto una máquina de estados. Un estado en UML es cuando un objeto o una interacción satisface una condición, desarrolla alguna acción o se encuentra esperando un evento.

Cuando un objeto o una interacción pasa de un estado a otro por la ocurrencia de un evento se dice que ha sufrido una transición, existen varios tipos de transiciones entre objetos: simples (normales y reflexivas) y complejas. Además una transición puede ser interna si el estado del que parte el objeto o interacción es el mismo que al que llega, no se provoca un cambio de estado y se representan dentro del estado, no de la transición. Como en todas las

metodologías OO se envían mensajes, en este caso es la acción de la que puede enviar mensajes a uno o varios objetos destino.

Diagramas de Implementación.

Se derivan de los diagramas de proceso y módulos de la metodología de Booch, aunque presentan algunas modificaciones. Los diagramas de implementación muestran los aspectos físicos del sistema. Existen dos tipos:

- Diagramas de componentes
- Diagrama de plataforma de despliegue

Diagramas de componentes.

Muestra la dependencia entre los distintos componentes de *software*, incluyendo componentes de código fuente, binario y ejecutable. Un componente es un fragmento de código *software* (un fuente, binario o ejecutable) que se utiliza para mostrar dependencias en tiempo de compilación.

Diagrama de plataformas o despliegue.

Muestra la configuración de los componentes *hardware*, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formado por otros componentes.

El objetivo de este apartado consiste en describir de forma general el funcionamiento del UML, así como los diagramas que en este implementan. No se pretende presentar la metodología completa ya que para el desarrollo de la presente tesis, en el análisis del sistema de cómputo, se aplica este lenguaje de modelado utilizando los diagramas de clases y los diagramas de casos de uso.

1.3.5. Componentes estructurales de los sistemas de información.

Sin importar las organizaciones a las que sirven o la forma en que se desarrollan y diseñan, todos los sistemas de información están compuestos de los siguientes seis componentes estructurales (Burch y Grudnitski, 1996, pág. 58): entrada, modelos, salida, tecnología, base de datos y controles. Estos componentes pueden tomar diferentes formas, valores y contenido; pueden parecer diferentes y trabajar en forma diferente; algunos pueden soportar sistemas bien diseñados; otros pueden soportar sistemas diseñados con deficiencia; algunos pueden ser imperfectos; algunos pueden ser altamente sofisticados, todo ello es irrelevante. Éstos son los seis componentes estructurales básicos de todos los sistemas de información. El qué tan bien se combinen y el tipo de sistema de información que resulte depende del diseñador, que es el arquitecto de los sistemas. Los componentes estructurales de los sistemas de información pueden conjuntarse para obtener sistemas de información funcionales que satisfagan las necesidades de las organizaciones y de sus usuarios. La comprensión de estos componentes estructurales, sus relaciones y acoplamientos y su contenido lógico y físico, proporcionan los conocimientos básicos para describir, desarrollar y diseñar sistemas de información. A continuación se describen brevemente los componentes estructurales.

Bloque de entrada.

“La entrada representa a todos los datos, texto, voz e imágenes que entran al sistema de información y los métodos por los cuales se capturan e introducen. La entrada está compuesta de transacciones, solicitudes, consultas, instrucciones y mensajes. Por lo general, la entrada sigue un protocolo y un formato para que el contenido, la identificación, la autorización, el arreglo y el procedimiento sean adecuados. La introducción puede hacerse mediante escritura manual, formas en papel, reconocimiento de huellas digitales, teclados, ratones, voz, sensores táctiles, caracteres y códigos ópticos y magnéticos” (Burch y Grudnitsky, 1996, pág. 58).

“En la actualidad, los medios más comunes para la introducción de transacciones y texto son las lectoras de códigos de barras y el teclado, respectivamente. Con frecuencia se puede conseguir una eficiencia en la entrada combinando los métodos. Por ejemplo, la entrada por voz puede complementarse mediante un teclado para proporcionar comandos como introducir, archivar o pararse, para dar entrada a códigos numéricos, o para la entrada de mensajes. Una forma aún mejor sería desarrollar sistemas de entrada de voz como una alternativa viable al teclado. Con la entrada por voz, el cuello de botella durante la entrada se reduciría sustancialmente. El usuario probablemente trabaja mejor mediante entrada por voz, debido a que ambas manos quedan libres para otras tareas, y la entrada por voz mediante teléfono por ejemplo, es altamente eficiente debido a que elimina los pasos intermedios de anotar y volver a teclear mensajes. También, una diversidad de dispositivos con entrada eficiente, como las pantallas sensibles al tacto responden directamente a la presión de los dedos. También se cuenta con plumas luminosas y ratones, para ser empleados con visualización por pantalla y gráficas” (Burch y Grudnitsky, 1996, pág. 59).

Bloque de modelos.

“Este componente consta de modelos lógico-matemáticos que manipulan de diversas formas la entrada y los datos almacenados, para producir los resultados deseados o salida. Un modelo lógico-matemático puede combinar ciertos elementos de datos para proporcionar una respuesta adecuada a una consulta, o puede reducir o agregar volúmenes de datos para obtener un reporte conciso. Puede ser tan simple como: $Ganancias = Ingresos - Gastos$. El componente de modelos también contiene una descripción de algunas de las técnicas de modelado más populares empleadas por los analistas de sistemas para diseñar y documentar las especificaciones de los sistemas. Estas técnicas incluyen diagramas UML, diagramas de flujo de datos, el empleo de prototipos, entre otros” (Burch y Grudnitsky, 1996, pág. 59).

Bloque de salida.

“El producto del sistema de información es la salida de información de calidad y documentos para todos los niveles de la gerencia y para todos los usuarios dentro y fuera de la organización. La salida es, en gran medida, el componente que guía e influye en los otros componentes. Si el diseño de este componente no satisface las necesidades del usuario, entonces los otros componentes tienen poca importancia” (Burch y Grudnitsky, 1996, pág. 59).

La salida representa el otro extremo de la entrada y obviamente no puede ser mejor que la entrada y los modelos empleados para producirla. Con frecuencia, la entrada y la salida son interactivas. La entrada se convierte en salida; la salida se convierte en entrada.

“El estudio anterior se ocupó del aspecto referente al mensaje ¿Pero qué hay acerca del medio? La salida se puede producir en pantallas, impresoras, dispositivos de audio o video. En el pasado, el principal medio de salida eran pilas de reportes impresos, denominados listados de computadora. En la actualidad, las personas están eligiendo otras formas de salida que vayan más de acuerdo a sus gustos, como gráficas, video y audio. Debido a que la mayoría de los trabajadores pasan entre 50 y 90 por ciento de su tiempo manejando información y comunicándose, están buscando, de hecho, una gama más amplia de medios que se aproximen a los métodos humanos naturales de emitir información y de comunicarse con el sistema y entre sí.”

Otros métodos populares de salida incluyen a los graficadores y equipos de fotocomposición. También, por supuesto, los sistemas de respuesta por voz y correo electrónico son formas ideales para la salida de mensajes; estas salidas ciertamente son del gusto de la mayoría de la gente debido a su atractivo y conveniencia" (Burch y Grudnitsky, 1996, pág. 60).

Bloque de tecnología.

"La tecnología es la caja de herramientas del trabajo en sistemas de información. Captura la entrada, activa los modelos, almacena y accede datos, produce y transmite salida, y ayuda a controlar todo el sistema. Hace todo el trabajo pesado y une a todos los componentes estructurales. La tecnología consta de tres componentes principales: la computadora y el almacenamiento auxiliar, las telecomunicaciones y el *software*. Las telecomunicaciones comprenden el empleo de medios electrónicos y de transmisión de luz para la comunicación entre nodos a lo largo de una distancia. El *software* corresponde a los programas que hacen que funcione el *hardware* de la computadora y le dan instrucciones sobre la forma de procesar los modelos. El *hardware* está compuesto de una variedad de dispositivos que proporcionan el soporte físico para los componentes estructurales. Por ejemplo, una terminal sirve como dispositivo de entrada para las transacciones contables; una unidad central de procesamiento (CPU) acciona a los modelos contables con datos apropiados; las impresoras localizadas en varias divisiones a lo largo de todo el país y conectadas al Servidor de Archivos mediante satélites y estaciones terrestres producen como salida estados contables. La base de datos almacena los archivos maestros de contabilidad, los diarios y los libros; y un dispositivo de seguridad e integridad de la base de datos ayuda a controlar la confidencialidad de la contabilidad y demás información sensible a medida que se transmite y también mientras se almacena en la base de datos" (Burch y Grudnitsky, 1996, pág. 60).

En su esencia misma, la tecnología es un sustituto del esfuerzo humano. De los seis componentes estructurales, la tecnología es el más evidente. La mayoría de los sistemas de información actuales y del futuro estarán basados en la tecnología. Sin embargo, un peligro latente es el de un sobreencantamiento con la tecnología, en tanto se descuidan las necesidades de información de los usuarios. Otro peligro es el de suponer que la adquisición e instalación de una computadora y su tecnología relacionada es equivalente a implementar un sistema de información. Nada puede estar más alejado de la verdad" (Burch y Grudnitsky, 1996, pág. 61).

1.4. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE VISUAL BASIC.

Introducción a Visual Basic.

Visual Basic es un sistema creado por Microsoft para el desarrollo rápido y eficaz de aplicaciones visuales que se ejecutan sobre los sistemas operativos Windows NT, Windows 95 y Windows 98 (los programas creados con **Visual Basic 5** no pueden utilizarse con el ambiente operativo Windows 3.x, si desea crear programas para ese ambiente, podrá usar cualquier versión de Visual Basic 1 a 3 y la versión de 16 bits para la 4).

Visual Basic pertenece al conjunto de herramientas **RAD** (Desarrollo Rápido de Aplicaciones) de gran expansión en la actualidad, los sistemas RAD se caracterizan por su potencia y facilidad de uso (otro sistema muy popular, aunque un tanto más complejo que Visual Basic, es *Delphi* de Inprise).

Nuestro curso está orientado a VB5, sin embargo, todos los conceptos son 100% aplicables a la última versión VB6, ya que el contenido del curso es de introducción a la programación, salvo algunos cambios menores en la forma y arreglo de las pantallas del ambiente de desarrollo, no deberá tener problemas para usar la versión 6.

Existen tres versiones de **Visual Basic**:

1. Estándar
2. Profesional

3. Empresarial

La **versión estándar**, o de aprendizaje, permite crear aplicaciones visuales, e incluye controles generales, retículas similares a una hoja de cálculo, cajas de diálogo estándar para desplegar e imprimir información, y controles para el manejo de información guardada en disco (bases de datos).

La **versión profesional** incluye todas las características de la versión estándar, y añade el generador de informes *Crystal Report*, los controles ActiveX y controles para Internet.

La **versión empresarial** incluye todas las características de la versión profesional, y agrega las galerías de objetos, las herramientas para la administración de bases de datos, un sistema para la administración de proyectos, y un administrador para los procesos de automatización basado en componentes.

Existe la versión **Visual Basic 6** como parte de Visual Studio® 6.0, en el que se incluyen entre otros:

- Visual C++ 6.0
- Visual FoxPro 6.0
- Visual J++ 6.0
- Visual InterDev 6.0
- MSDN, Microsoft Developer Network Library, edición personal
- Windows NT 4.0 Option pack
- Visual SourceSafe 6.0

Herramientas de soporte de Aplicación del ciclo de vida.

- Herramientas de desarrollo para bases de datos empresariales
- Soporte de desarrollo del equipo
- BackOffice Server 4.5 Developer edition

Las características en VB incluyen un número de significativas ampliaciones de desempeño y productividad que, colocan a esta herramienta en lo más alto de la lista de herramientas de desarrollo Cliente/Servidor. Características avanzadas, como la compilación de código nativo, una alta velocidad de acceso a las bases de datos, y un mejor ambiente de desarrollo son sólo la punta del iceberg.

La mejor característica, es la implementación del componente con la capacidad de creación de la tecnología de ActiveX en VB 5.0. Antes, los desarrolladores requerían de un ambiente especial de desarrollo para ActiveX; sin embargo ahora se pueden crear aplicaciones basadas en los componentes destinados a Internet, Intranet, y los tradicionales ambientes Cliente/Servidor.

En adición al nuevo desarrollo enfocado hacia ActiveX, se incorporó una nueva versión de VBA (Visual Basic Application), en una nueva encarnación de la IDE (Integrated Development Environment), junto con la inclusión de la tecnología IntelliSense. Además, ahora tiene un verdadero repositorio (Microsoft Repository 1.0) para cuando algún desarrollador quiera construir desde la Base de datos. El Servidor de Transacciones Microsoft Edición Desarrollador finalmente permite a VB abarcar una escala de cientos de usuarios. El depurador T-SQL permite a los desarrolladores depurar procedimientos almacenados interactivamente al mismo tiempo, y en el IDE del Cliente. Por último, aunque no menos importante, es el Explorador de Desempeño de Aplicaciones (APE), el cual permite simular y analizar aplicaciones en escenarios distribuidos antes de comenzar a especificar alguna arquitectura. Este proceso permite optimizar una aplicación y una arquitectura para que alcance un buen desempeño.

VB permite construir nuestras interfaces mediante el ensamble de partes de componentes preconstruidos, y además aplica el mismo paradigma en la construcción de los

controles ActiveX, compartiendo las mismas características de desarrollo para las tradicionales aplicaciones de VB. Los desarrolladores pueden construir controles para cualquier cosa, desde los controles de subclases hasta la posibilidad de combinación de más de 1000 controles comerciales de ActiveX. Una vez creados dichos controles ActiveX, éstos pueden ser usados en otros ambientes de desarrollo que puedan usar Activex. .

VB viene en tres presentaciones: la Edición de Aprendizaje para quienes sólo desean hacer programas sencillos; la Edición Profesional para aquellos que desean crear básicamente aplicaciones Cliente/Servidor e Internet/Intranet (incluyendo en esta edición, los controles ActiveX, el repositorio y el compilador nativo); y, finalmente, la Edición Enterprise para aquellos que lo necesitan para toda una corporación. La edición Enterprise incluye SQL Server, Visual SourceSafe, Remote Data Objects (RDO), Servidor de transacciones Edición Desarrollador, y Herramientas de Bases de Datos Microsoft.

VBA, el lenguaje nativo para Visual Basic, es ahora el lenguaje para cientos de otros productos, incluyendo Excel, Word, PowerPoint, Access, Project, y una docena más de productos no Microsoft. VBA no sólo es un lenguaje: es toda una máquina de lenguaje (un editor, un visualizador de objetos, y una herramienta de depuración). Los desarrolladores pueden agregar valor en la creación de aplicaciones VBA porque el código es portable hacia otros ambientes y aplicaciones. Quienes consideraban a VBA como otro lenguaje de aplicaciones propietarias pueden ahora dar otro vistazo y cambiar su opinión.

Los desarrolladores armarán la interfaz usando controles GUI preconstruidos , y entonces darán a la aplicación su comportamiento con la colocación del código atrás de los controles o por cambios en las propiedades del control. Cabe señalar que éste es el mismo paradigma utilizado por las herramientas de la competencia, como Borland Delphi, y Oracle.

Lo realmente nuevo en herramientas son las características IntelliSense, soporte multiproyecto y editor ampliado, proyecto y ventana de propiedades, además de las nuevas herramientas de depuración.

Con esta versión se pueden cargar muchos proyectos a la vez, lo cual es una gran ayuda para crear controles de tal forma que los desarrolladores puedan cargar tanto el proyecto del control como el proyecto del host al mismo tiempo. El proyecto del host puede crear un archivo ejecutable que permitirá al host el control para pruebas. Desde la ventana de proyecto se pueden ver ahora todos los componentes de todos los proyectos cargados. Esto incluye a todos los controles ActiveX, formas, clases, módulos, y archivos de recursos. Cada proyecto es representado por una raíz en la ventana de proyecto. Haciendo click en el símbolo "más" se puede desplegar cuántas raíces hay y cuántas están incluidas en el proyecto.

En la ventana de propiedades desplegaba los atributos de controles, módulos, formas y clases. Con esta versión, usted puede ver las propiedades de los objetos por categorías o alfabéticamente. Ahora existen dos tabuladores, Alfabético y Categoría: mientras se utiliza el tabulador alfabético se podrán ver las propiedades alfabéticamente, y si se utiliza el tabulador por categoría se verán las propiedades por categoría.

Las características de depuración incluyen la habilidad de cargar ventanas locales, una ventana de watch, y una ventana inmediata. Las ventanas locales también incluyen un visualizador de llamadas al Stack, el cual muestra las variables corrientes y permite encontrar exactamente dónde están localizadas estas variables y dónde son referenciadas. La ventana de variables locales da la información de todas las variables declaradas en los procedimientos, así como de los valores almacenados en ellos. La ventana de Watch permite monitorear el valor de una variable o expresión en particular y permitir cambios al valor durante la ejecución. La ventana inmediata permite examinar cualquier expresión VBA, y el Stack de llamadas despliega una lista de todas las llamadas activas a procedimientos.

El Object Browser puede distinguir entre propiedades, métodos, propiedades personales, manejo de eventos y procedimientos definidos por el usuario. El objeto browser también permite desplegar funciones y retornar tipos y nombres de parámetros. Un mecanismo de hiperliga permite mover directamente al objeto referenciado y permite una fácil navegación de la jerarquía de clases. Además se pueden buscar objetos a través de todas las librerías a la vez.

Bloques de construcción.

Haciendo a un lado las nuevas características, el aspecto más destacado en VB es la habilidad de crear controles ActiveX, lo que permite crear controles mediante la modificación de controles existentes de reuso, a través de mecanismos de subclases pseudo orientadas a objetos, o mediante la combinación de algunos controles existentes. Una vez creados los controles usando VB, se pueden definir dichos controles incluyendo Páginas Web, documentos, aplicaciones Cliente/Servidor, y cualquier otro dispositivo de cómputo en red. Debemos recordar que ActiveX ofrece muchas de las características proporcionadas por Java.

Construir controles ActiveX, es una tarea relativamente fácil. Primero que nada se crea un test container para la creación de un estándar ejecutable usando VB; luego, se agrega un proyecto de control blanco ActiveX; y finalmente se dibuja la interfaz para el control. VB usa el mismo paradigma de desarrollo visual para construir controles ActiveX que los usados para otras aplicaciones. Todo es cuestión de dibujar la Interface, fijar las propiedades, y agregar algún código VBA de manejo de eventos. Una vez hecho lo anterior, es hora de usar y probar este control. Todo lo que se necesita es presionar F5 para correr la aplicación y ver cómo trabaja. Una vez que se está seguro que la aplicación funciona, se pueden utilizar los controles en un aplicación externa.

Más alto, más fuerte, más rápido.

Se puede observar que su desempeño es realmente bueno: no sólo el IDE corre rápido, sino que no requiere de intérprete, no tiene problemas de desempeño, es simple y fácil de utilizar. El desempeño de acceso a las bases de datos también es bueno. Sin embargo, yo encontré dificultades en determinar si el mayor desempeño es el resultado de la adición de un nuevo compilador nativo o si la capa de acceso a las bases de datos es más rápido.

La fiebre de los componentes.

Algunos de los problemas que se pudieron observar de Visual Basic es el cambio de paradigma de las últimas versiones. Microsoft está moviéndose de un enfoque de uso de servidores COM –donde los mecanismos participan de las aplicaciones– a la construcción y desarrollo de controles ActiveX. Si se comprende la estrategia de la Plataforma Active, con ActiveX como front-end, servidor Active como back-end, y servicios DCOM como enlaces, podemos creer que este paradigma tendrá más sentido. Sin embargo, Microsoft necesita enseñar a los desarrolladores cómo lograr que esta arquitectura pueda solucionar problemas de aplicación y desarrollo. Microsoft ahora vende dos, tres, y n soluciones, y da a cada una de ellas precios diferentes.

Es claro que Microsoft está basando todos sus productos en los controles ActiveX en estos días, y ésta es la forma en que se harán la mayoría de los programas Cliente/Servidor de las compañías en el futuro. Es claro además que Java empieza a interrelacionarse con Microsoft. Entonces, como podemos observar, VB está dando el salto a la fabricación de controles ActiveX que todos estaban esperando, pues hasta el momento la creación de controles ActiveX requería de un desarrollador con conocimientos en Visual C++ o alguna otra herramienta selecta.

La estrategia principal implica que, si se es un desarrollador de VB, VB puede ayudar a realizar una rápida actualización de lista. Si se es un desarrollador o Webmaster, VB ayudará a crear contenidos dinámicos ActiveX. Si se planea cualquier trabajo con componentes ActiveX, VB será la herramienta a utilizar. Sin embargo, los productos de Microsoft son algo difíciles de manejar con ActiveX y la Plataforma Active, y esto, quiérase o no, es un serio golpe. La idea, obviamente, es obligar al consumo.

ActiveX y OCX.

La característica más poderosa de **Visual Basic** es su capacidad para integrar controles de terceros (conocidos como controles ActiveX y OCX). Los controles ActiveX y OCX extienden

las capacidades de Visual Basic. Sin importar cual aplicación esté desarrollando, si la característica de programación no está incluida intrínsecamente en **Visual Basic 6**, se puede añadir esa característica mediante un control ActiveX u OCX.

Independiente de dispositivos.

Windows es un sistema operativo bastante popular, debido a que es independiente de dispositivos. Esto significa que no importa qué impresora esté utilizando, ya que si Windows la acepta (al momento en que instala la impresora), ésta deberá trabajar bien con cada programa que desarrolle. De hecho, el programador, no deberá preocuparse sobre cuál impresora utilizar. Nuevamente, siempre y cuando Windows haya aceptado la impresora, el programa deberá trabajar bien con la impresora instalada. Este concepto de independencia de dispositivos se aplica a las demás partes del hardware. Por lo tanto, sin importar qué tarjeta de sonido estén utilizando sus usuarios, los programas serán capaces de trabajar con ella sin que usted sepa el tipo de dicha tarjeta.

Otra cosa que hace que Windows sea tan popular es el hecho de que la interfaz de usuario es la misma para todas las aplicaciones de Windows. Esto es, los programas que aprenderá a diseñar en este curso son programas estándar de Windows que utilizan la interfaz normal para el usuario de Windows. Por lo tanto no tiene que indicar al usuario cómo hay que operar el programa.

Penetración de sistemas operativos por plataforma a fines de 1998 (Fuente Fortune):

Windows	374	millones
Macintosh	27	millones
Linux	7	millones
Unix	6	millones

Especificaciones técnicas de Visual Basic 6.0.

Para utilizar Visual Basic 6.0 Enterprise Edition, necesita:

- Una PC con procesador 486DX/66 MHz o superior (Pentium 90 recomendado).
- Sistema operativo Microsoft Windows 95 o Microsoft Windows NT 4.0 con Service Pack 3 (Service Pack 3 incluido) o posterior.
- 24 MB en RAM para Windows 95 y Windows NT (32 MB recomendado).
- Microsoft Internet Explorer 4.01 con Service Pack 1 (incluido).
- Espacio en disco duro requerido:

Instalación típica: 116 MB

Instalación máxima: 135 MB

- Espacio adicional requerido en disco duro para los siguientes productos:
 - Microsoft Internet Explorer: 43 MB típica, 59 MB máxima.
 - MSDN™: 57 MB típica, 493 MB máxima.
 - Windows NT 4.0 Option Pack: 20 MB para Windows 95 o posterior; 200 MB para Windows NT 4.0.
- Microsoft SQL Server 6.5 y Microsoft SNA Server 4.0 ejecutados únicamente en Windows NT 4.0, además de requerir el siguiente espacio adicional en disco duro:
 - SQL Server: 80 MB típica, 95 MB máxima
 - SNA Server: 50 MB típica, 100+ MB máxima
- Unidad CD ROM.
- Monitor VGA o de resolución más alta, se recomienda Super VGA.
- Microsoft Mouse o dispositivo indicador compatible.

Para utilizar Visual Basic 6.0 Professional Edition, necesita:

- Una PC con procesador 486DX/66 MHz o superior (Pentium 90 recomendado).

- Sistema operativo Microsoft Windows 95 o Microsoft Windows NT 4.0 con Service Pack 3 (Service Pack 3 incluido) o posterior.
- 16 MB en RAM para Windows 95 (32 MB recomendado); 24 MB para Windows NT 4.0 (32 MB recomendado).
- Microsoft Internet Explorer 4.01 con Service Pack 1 (incluido).
- Espacio en disco duro requerido:

Instalación típica: 76 MB

Instalación máxima: 94 MB

- Espacio adicional requerido en disco duro para los siguientes productos:
 - Microsoft Internet Explorer: 43 MB típica, 59 MB máxima.
 - MSDN™: 57 MB típica, 493 MB máxima.
 - Windows NT 4.0 Option Pack: 20 MB para Windows 95 o posterior; 200 MB para Windows NT 4.0.
- Unidad CD ROM.
- Monitor VGA o de resolución más alta, se recomienda Super VGA.
- Microsoft Mouse o dispositivo indicador compatible.

Para utilizar Visual Basic 6.0 Learning Edition, necesita:

- Una PC con procesador 486DX/66 MHz o superior (Pentium recomendado).
- Sistema operativo Microsoft Windows 95 o Microsoft Windows NT 4.0 con Service Pack 3 (Service Pack 3 incluido) o posterior.
- 16 MB en RAM para Windows 95 (32 MB recomendado); 24 MB para Windows NT 4.0 (32 MB recomendado).
- Microsoft Internet Explorer 4.01 con Service Pack 1 (incluido).
- Espacio en disco duro requerido:
 - Instalación típica: 52 MB
 - Instalación máxima: 65 MB
- Espacio adicional requerido en disco duro para los siguientes productos:
 - Microsoft Internet Explorer: 43 MB típica, 59 MB máxima.
 - MSDN™: 57 MB típica, 493 MB máxima.
 - *Learn Visual Basic Now*: 16 MB incluyendo prácticas de laboratorio.
- Unidad CD ROM.
- Monitor VGA o de resolución más alta, se recomienda Super VGA.
- Microsoft Mouse o dispositivo indicador compatible.

Características Estándar.

- **Compilador de código nativo.** Creación de aplicaciones y componentes del servidor y del cliente optimizados por el procesamiento del compilador de código nativo optimizado de Visual C++ 6.0.
- **ADO (ActiveX Data Objects, Objetos de datos ActiveX).** Visual Basic 6.0 presenta ADO como el nuevo estándar para acceso a datos. Entre los controladores OLE DB que se incluyen están SQL Server™ 6.5+, Oracle 7.3.3+, Microsoft Access, ODBC y SNA Server.
- **Visual Database Tools empresarial integrado.** Visual Basic 6.0 proporciona un conjunto completo de herramientas para integrar bases de datos con cualquier aplicación. Entre las características de bases de datos se incluyen herramientas de diseño para crear y modificar bases de datos de SQL Server 6.5, Oracle 7.3.3 o superior y AS/400.
- **Enlace automático de datos.** Prácticamente no se necesita nada de código para enlazar controles a los orígenes de datos. Para conectar el control a cualquier origen de datos sólo es necesario configurar dos propiedades en la ventana Propiedades.

- **Diseñador de entorno de datos.** Creación visual de objetos de comando reutilizables con la funcionalidad arrastrar y colocar. Enlace a múltiples orígenes de datos para agregar y manipular los datos.
- **Diseñador de informe de datos.** Los programadores pueden arrastrar y colocar rápidamente controles personalizados enlazados a datos para crear formularios o informes. Para crear jerarquías personalizadas de datos basta con completar un cuadro de diálogo y arrastrar el comando al formulario.
- **Creación de clientes y proveedores personalizados de datos.** Creación de controles COM personalizados de datos para el cliente o proveedores de OLE DB de la capa media y personalizados.
- **Herramientas de distribución y comprobación de la capa media** Depuración interactiva de los componentes empaquetados de Microsoft Transaction Server, y empaquete y distribuya componentes a servidores remotos.
- **Diseñador de WebClasses de Visual Basic.** Creación de aplicaciones y componentes del servidor a los que se tiene fácil acceso desde cualquier explorador de Web en cualquier plataforma.
- **Diseñador de páginas HTML dinámico.** Desarrollo de aplicaciones multimedia de gran impacto mediante el modelo de objeto de documento y HTML dinámico como su interfaz de usuario con un editor WYSIWYG.
- **Compatibilidad con equipos portátiles.** Aplicaciones cliente-servidor que funcionen con bases de datos que exista o no conexión con una red de área local (LAN) o con Web.
- **Visual Modeler.** Planee y modele visualmente arquitecturas complejas de aplicación y código para regenerar las interfaces de componentes a medida que se refinan los modelos.

Ventajas de Visual Basic.

El sistema de desarrollo Microsoft Visual Basic es la herramienta más productiva para crear de manera rápida soluciones de negocios que puedan ser ejecutadas en Windows® y en el web. Un ambiente de desarrollo integral ayuda a que los desarrolladores creen e instalen aplicaciones cliente/servidor, además permite programar para Internet utilizando las herramientas de desarrollo y técnicas de Visual Basic.

Enfoque en la productividad y rendimiento.

- El lenguaje de programación más popular de todo el mundo. Desarrolle rápidamente aplicaciones de alto rendimiento y componentes en un ambiente RAD.
- Estructuración aplicaciones más rápidas y de código nativo, componentes que utilicen la misma tecnología compiladora de clase mundial, como se hace en el sistema de desarrollo Microsoft Visual C++®.
- Es posible optimizar la velocidad y el tamaño de las aplicaciones, así como mejorar aún más el rendimiento.
- Conexión de sus aplicaciones y componentes sin necesidad de seguir una administración de conexión manual.
- Aumento en la velocidad y la productividad del desarrollo con la tecnología IntelliSense®. En lo que se refiere a la sintaxis y elaboración de textos utilice funciones como Complete Word, Quick Info y DataTips®.
- Creación fácil de una amplia variedad de componentes basados en COM, incluyendo los Controles ActiveX®, así como los componentes middle-tier y de servidor.

Creación fácil de aplicaciones de bases de datos.

- Desarrollo rápido de formatos con muchos datos o la utilización de Report Writer integrado para desarrollar reportes sofisticados, jerárquicos, todo con la función de "arrastrar y colocar".
- Creación de componentes de acceso a datos reutilizables usando el ambiente Data Environment, el cual se utiliza en otras herramientas de desarrollo o en el web, donde automáticamente se aprovecha la unión de datos.
- Creación de una variedad extensa de aplicaciones para el usuario móvil. Elaboración de aplicaciones cliente/servidor que trabajen con bases de datos, ya sea que estén conectadas a LAN o al web. Ahora, Visual Basic 6.0 soporta la interfaz para el acceso universal a datos con ActiveX Data Objects (ADO) versión 2.0.
- Alcance de altos niveles de productividad al trabajar con datos jerárquicos con el control FlexGrid. Despliegue fácilmente los datos *master-detail* de manera que le permita extender, colapsar, ocultar o mostrar información.

Incremento en la productividad de manera eficiente con las herramientas integradas de bases de datos visuales.

Visual Basic 6.0 proporciona las herramientas de bases de datos integradas, como: ver tablas, diseñador de consulta y el diseñador de base de datos. Sin salir de Visual Basic.

- Herramientas visuales recientemente integradas de base de datos para realizar las actividades más comunes, sin dejar de utilizar Visual Basic. Ver tablas, modificación de datos y creación de consultas SQL dentro del ambiente de desarrollo para cualquier base de datos compatible ODBC o OLE DB.
- Diseñador de base de datos completamente integrado que permite visualizar el diseño y modificar esquemas de bases de datos activos, así como otros objetos para las bases de datos Microsoft SQL Server™ 6.5+ y Oracle 7.3.3+.
- Diseñador de consulta para crear consultas visualmente y realizar tareas de bases de datos sofisticadas sin tener que conocer el lenguaje SQL.
- Creación, edición y modificación de procedimientos almacenados, todos dentro del ambiente Visual Basic. Igual que en el editor de código de Visual Basic, la sintaxis se codifica por color SQL y se verifica para su corrección. Esto hace que el código SQL sea más fácil de leer y menos susceptible a los errores.
- Optimización interactiva Microsoft SQL Server 6.5 o los procedimientos almacenados al último y ejecútelos al mismo tiempo, y en la misma estación de trabajo, a medida que se desarrolla y se optimizan las aplicaciones.

Aplicaciones Internet.

Con Visual Basic 6.0, se pueden crear aplicaciones poderosas para Internet, accesibles desde cualquier explorador o plataforma.

- Programación de aplicaciones basadas en Windows en un ambiente RAD para desarrollar aplicaciones para el web.
- Elaboración de aplicaciones y componentes web ejecutando Microsoft Internet Information Server. Estas aplicaciones pueden accederse desde cualquier explorador o plataforma.
- Instalación de aplicaciones y componentes Internet rápida y fácilmente utilizando el asistente Package y Deployment.

Funcionalidad de Internet mejorada con HTML dinámico.

- Beneficios de soporte completo de presentación, incluyendo configuración dinámica, transiciones gráficas, tipografía sólida, desplazamiento y reajuste automático de tamaño y más.

- Creación de páginas con HTML dinámico manejadas por eventos para que sean utilizadas en Microsoft Internet Explorer 4.0. El nuevo diseñador de páginas permite programar páginas web altamente interactivas tan fácil como con un formato de Visual Basic.
- Distribución del *software* automáticamente y mejoras tanto el HTML dinámico como la interfaz del explorador del usuario.
- Integración fácil del web, incluyendo las páginas con HTML dinámico, dentro de las aplicaciones Visual Basic, utilizando el control Internet Explorer 4.0 WebBrowser.

Escalabilidad en la empresa.

- Estructuración de una plataforma de clase mundial utilizando el poder y confiabilidad de las herramientas estándar en la industria. Microsoft Visual Basic se integra perfectamente con la familia Microsoft BackOffice®, la cual proporciona un ambiente confiable y seguro para aplicaciones empresariales sólidas, ya sean para intranet, o para Internet.
- Aprovechamiento de las ventajas que la plataforma Windows NT® le brinda. Proporcione aplicaciones de código nativo para los sistemas que se ejecutan en la plataforma Windows 95 y Windows NT, incluyendo las de Intel y DEC Alpha.
- Modelación visual de las arquitecturas e interfaces de componentes con Visual Modeler, el cual le permite generar código y modificarlo conforme realiza cambios en sus modelos.
- Creación de mejores soluciones utilizando Windows NT Option Pack. Microsoft Transaction Server 2.0, Microsoft Internet Information Server 4.0 y Microsoft Message Queue Server le permiten elaborar aplicaciones de negocios sofisticadas y para usuarios múltiples de web, LAN, o ambos.
- Utilización de Microsoft SQL Server 6.5 Developer Edition, que es un sistema de administración de base de datos escalable y de alto rendimiento para sistemas basados en Windows NT, para crear un prototipo y probar las aplicaciones empresariales.
- Asegurar el código de origen utilizando Microsoft Visual SourceSafe™. Este proporciona un control de versión fácil de usar, seguro y orientado a un proyecto. Un equipo de cualquier tamaño puede incrementar la productividad y proteger fácilmente documentos, códigos fuente y el contenido para el web o una PC.

Desventajas de Visual Basic.

- Es un sistema dependiente de una plataforma.
- No cuenta con apuntadores estándar ni con programación orientada a objetos(OOP, sus siglas en ingles) real.
- En ocasiones, el manejo automático de la memoria (con datos de cadena y Variant) puede producir pequeños y muy ocasionales retrasos en los programas.
- Microsoft ha alterado en varias ocasiones la estrategia de acceso a datos(DAO, RDO y ADO), lo que ha traído serias consecuencias a los desarrolladores. A pesar de esto, la firma ha dicho que ADO es la estrategia a seguir y que los desarrolladores ya no deberán preocuparse al respecto.
- Los controles ActiveX no son independientes a la plataforma y no son compatibles con equipos UNIX, de uso tan extendido en Internet.

1.5. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE SQL-SERVER.

Visión general del producto.

A medida que las empresas simplifican los procesos y descentralizan la toma de decisiones, cada vez más se depende de la tecnología para acercar la información a los usuarios. Desde este punto de vista, las organizaciones están considerando a la informática

distribuida como el puente entre los datos y las decisiones de negocio. El rendimiento y la fiabilidad se han convertido actualmente en factores incluso más importantes debido al crecimiento en tamaño y número de usuarios de los sistemas transaccionales.

Microsoft SQL Server 7 es un sistema de administración de base de datos escalable de alto rendimiento, diseñado específicamente para el proceso distribuido cliente/servidor (C/S). La replicación de datos incluida, las potentes herramientas de gestión, la integración a Internet y su arquitectura de sistema abierto proporcionan la mejor plataforma para implantar soluciones de información efectivas en costo.

Microsoft SQL Server 7 es la última versión del sistema de gestión de bases de datos relacionales (SGBDR) que aprovecha la sólida base establecida por su predecesor SQL Server 6.5. Como la mejor base de datos para Windows NT, SQL Server 7 es el SGBDR ideal para un amplio espectro de clientes corporativos y fabricantes independientes de software (ISV). Las necesidades y requisitos del cliente han dado lugar a innovaciones significativas en SQL Server versión 7, entre las que se incluyen la facilidad de uso, escalabilidad, fiabilidad, y almacenamiento de datos.

Entre las más importantes innovaciones de Microsoft SQL Server 7 cabe citar:

- Primera base de datos que soporta la configuración automática y la auto-optimización.
- Primera base de datos con un servidor OLAP integrado.
- Primera base de datos con los servicios de transformación de datos (Data Transformation Services, DTS) integrados.
- El Data Warehousing Framework constituye el primer planteamiento de amplia cobertura, para la resolución de los problemas que plantea la utilización de metadatos.
- La primera base de datos que ofrece administración multiservidor para un gran número de servidores.
- Una gran variedad de opciones de duplicación de cualquier base de datos.
- La mejor integración con la familia Windows NT Server, Microsoft Office y BackOffice®.
- Acceso universal a los datos (Universal Data Access), la estrategia de Microsoft para permitir el acceso de alto rendimiento a una gran cantidad de fuentes de información.

Características principales del producto.

- **Servicios de ayuda a la toma de decisiones.** Servicios de ayuda a la toma de decisiones (DSS) es un componente de proceso analítico en línea (OLAP) con múltiples características de Microsoft® SQL Server™ 7.0.
- **Almacenamiento simplificado en disco.** La simplificada arquitectura de almacenamiento en disco permite la escalabilidad desde bases de datos de equipos portátiles hasta bases de datos empresariales de tamaño de terabyte.
- **Optimizador de consultas con múltiples fases.** El Optimizador de consultas con múltiples fases busca el plan óptimo para consultas para mejorar el rendimiento de consultas complejas.
- **Estadísticas automáticas.** Estadísticas automáticas extrae estadísticas mediante el análisis rápido de una muestra, habilitando el Optimizador de consultas para utilizar la información más reciente e incrementar la eficacia de las consultas.
- **Herramientas y programas.** Las herramientas y los programas se ejecutan mucho más rápido y están diseñados para que produzcan menos impacto en las operaciones del servidor.
- **Copia de seguridad activa.** Las copias de seguridad activas proporcionan copias de seguridad en línea de alto rendimiento con un impacto mínimo en los sistemas en funcionamiento.
- **Ayudante de Web.** El Asistente de Web le facilita la publicación de datos en el Web.
- **Duplicación de mezcla.** La duplicación de mezcla permite a los usuarios trabajar libre e independientemente, después combinar su trabajo y, finalmente, mediante una posterior resolución de conflictos por prioridades solucionar los conflictos de mezcla.

- **Servicios OLAP integrados.** Los servicios OLAP integrados agilizan el análisis eficaz de información compleja, esencial para crear informes, para el análisis de datos, para ayudar a tomar de decisiones y modelizar datos.
- **Servicios de transformación de datos.** Los servicios de transformación de datos sirven para importar, exportar y transformar datos heterogéneos.
- **DBCC.** Comprueba la coherencia física y lógica de la base de datos. El algoritmo patentado de un solo paso agiliza el rendimiento. Las nuevas características son compatibles y pueden resolver algunos problemas. La nueva arquitectura del motor de almacenamiento reduce la necesidad de DBCC, pero supone una buena práctica.
- **Error! Bookmark not defined.Soporte para servidores SMP más grandes.** Microsoft SQL Server está diseñado para proporcionar una excelente escalabilidad en servidores SMP de diversos fabricantes. La versión estándar está optimizada para ser utilizada en servidores SMP de hasta cuatro procesadores. La versión Enterprise Edition está diseñada y se licencia para ser utilizada en nuevos servidores SMP de más de cuatro procesadores para obtener una mayor escalabilidad.
- **Error! Bookmark not defined.Preparado para clusters de alta disponibilidad.** Microsoft SQL Server, Enterprise Edition incluye además soporte para Microsoft Cluster Server (MSCS), denominado inicialmente "Wolfpack". En una configuración de cluster de alta disponibilidad, Microsoft SQL Server proporciona el 100% de protección contra fallos de hardware para aplicaciones críticas. Para simplificar la gestión de clusters de alta disponibilidad, Microsoft SQL Server, Enterprise Edition proporciona herramientas gráficas fáciles de usar para instalar y configurar el cluster de dos nodos.
- **Error! Bookmark not defined.Soporte para memoria adicional.** El completo soporte de Windows NT Server para sintonizar hasta 4 GB de RAM (4GT), permite a Microsoft SQL Server aprovechar memoria adicional. La utilización de 4GT permite a Microsoft SQL Server direccionar hasta 3 GB de memoria real, proporcionando un mayor rendimiento para aplicaciones Data Warehouse. Esta característica está disponible para Microsoft SQL Server Enterprise Edition únicamente en servidores Intel de 32-bit. El soporte para memoria Very Large Memory (VLM) en procesadores de 64-bit como Digital Alpha estará disponible en una futura versión de Microsoft SQL Server, Enterprise Edition.
- **Interfaz de lenguaje natural.** Permite la recuperación de información de SQL Server Enterprise Edition utilizando Inglés, en lugar de un lenguaje de consultas tradicional, como SQL. Una aplicación que utilice el interfaz de lenguaje natural acepta comandos, sentencias y preguntas realizados en inglés como por ejemplo, *input*, y permite determinar su significado. A continuación escribe y ejecuta una consulta de base de datos en SQL Server y da formato a la respuesta.
- **Una plataforma para generar aplicaciones distribuidas fiables.** Además, Microsoft SQL Server, Enterprise Edition, conjuntamente con Windows NT Server, Enterprise Edition, es una plataforma completa para aplicaciones de base de datos distribuidas, fiables y a gran escala, utilizando Microsoft Transaction Server (MTS) y Microsoft Message Queue Server (MSMQ). Microsoft Transaction Server es un middleware basado en componentes para generar rápidamente aplicaciones transaccionales distribuidas, escalables y fáciles de administrar. MTS proporciona unos sencillos componentes que permiten generar y ejecutar transacciones complejas de forma fiable y eficiente a través de redes distribuidas, incluyendo el soporte integrado para aplicaciones basadas en la Web. MSMQ es un middleware de almacenamiento y reenvío de mensajes que garantiza la distribución de mensajes entre aplicaciones que se ejecutan en múltiples máquinas a través de una red. MSMQ es el entorno ideal para crear aplicaciones distribuidas a gran escala que incluyen sistemas móviles o se comunican a través de redes que en ocasiones pueden no resultar fiables.

Características Adicionales.

- **Servicios de transformación de datos.** DTS simplifica el proceso de importación y transformación de datos de orígenes múltiples y heterogéneos, tanto de manera interactiva como automática.

- **Estadísticas automáticas.** El optimizador de consultas administra la recopilación de estadísticas y garantiza una evaluación de planes eficaz. Los nuevos algoritmos de muestreo aumentan el rendimiento.
- **E/S mayor y más eficaz.** Los bloques de E/S son cuatro veces mayores, las páginas son de 8 Kb, las extensiones de 64 Kb y los recorridos usan bloques de 64 Kb. LA E/S inteligente es la tecnología clave para mejorar el rendimiento con grandes tamaños de E/S. Las lecturas anticipadas más eficientes, los recorridos en el orden físico de las filas y la E/S paralela también mejoran el rendimiento.
- **Estrategias de utilización de índices.** Las nuevas estrategias incluyen el uso de varios índices en una única tabla o en varias tablas, índices de cobertura múltiple y combinados, creación de índices paralelos en la misma tabla y mantenimiento automático de estadísticas de manera predeterminada.
- **Combinaciones.** En una única consulta se pueden utilizar varios tipos de combinaciones. El procesador de consultas reconoce ciertos tipos de combinaciones comunes.
- **Consultas paralelas.** El rendimiento mejora si se utiliza la ejecución en paralelo de consultas de una única consulta a través de múltiples procesadores. Los pasos en una única consulta se ejecutan en paralelo, entregando el tiempo de respuesta óptimo.
- **Memoria dinámica.** Mejora el rendimiento optimizando la ubicación y el uso de la memoria. Minimiza los conflictos con otros administradores de recursos.
- **Administrador de registro.** El diseño simplificado mejora el rendimiento en truncamientos, en copias de seguridad en línea y en operaciones de recuperación.
- **Lectura anticipada.** La lógica de la lectura anticipada inteligente mejora el rendimiento y elimina la necesidad de la optimización manual.
- **Bloqueo dinámico de filas.** El bloqueo completo de filas se ha implementado tanto para filas de datos como para entradas de índice. El bloqueo dinámico elige automáticamente el nivel de bloqueo óptimo (fila, página, página múltiple y tabla) para todas las operaciones de base de datos.
- **Administración dinámica del espacio.** Las bases de datos pueden crecer o reducirse automáticamente dentro de límites configurables, minimizando la necesidad de la intervención del administrador de bases de datos. Ya no es necesario asignar espacio y administrar las estructuras de datos.
- **Capacidad para utilizar cantidades mayores de memoria.** SQL Server 7.0 Enterprise admite direccionamientos de memoria superiores a 4 GB, junto con Windows NT Server 5.0, sistemas que utilizan procesadores Alpha y otras combinaciones.
- **Confiabilidad.** Las estructuras y algoritmos simplificados de datos mejoran la concurrencia, escalabilidad y confiabilidad. Las comprobaciones en tiempo de ejecución de estructuras de datos críticos hacen que la base de datos sea más eficaz, minimizando la necesidad de comprobaciones de coherencia.
- **Almacenamiento escalable.** El formato de disco y el subsistema de almacenamiento proporcionan un almacenamiento que es escalable desde bases de datos muy pequeñas hasta bases de datos muy grandes.
- **Unicode nativo.** Con API Unicode de ODBC y OLE DB, mejora el soporte multilingüe.
- **Copia de seguridad y restauración.** Las utilidades de restauración y copia de seguridad paralela escalan a las velocidades de dispositivo. Durante la copia de seguridad, totalmente sin desconexión, se mantiene un bajo impacto de los sistemas operacionales y un muy alto proceso de transacciones de servidor.
- **Carga de datos masiva.** La velocidad de importación y exportación de datos mejora considerablemente. Ahora utiliza OLE DB y funciona junto con el procesador de consultas para planear y optimizar las consultas.

Especificaciones.

1. Sistema utilizando un procesador Intel Pentium o Alpha AXP
2. Microsoft Windows NT Server, 3.51 o posterior (requiere NT Server Enterprise 4.0 para utilizar Microsoft Cluster Server, el soporte para sistemas <4 SMP y memoria adicional)
3. 16 MB de memoria (32 MB con replicación)

4. 80 MB de espacio disponible en disco (95 MB si se instala **books online**)
5. Unidad de CD-ROM

Opciones de red.

Las siguientes redes están soportadas utilizando protocolos nativos:

1. Microsoft Windows NT Server
2. Microsoft LAN Manager
3. Novell NetWare
4. Redes basadas en TCP/IP
5. IBM LAN Server
6. Banyan VINES
7. Digital PATHWORKS
8. Apple AppleTalk

Clientes soportados:

1. Sistema Operativo Microsoft Windows® versión 3.1
2. Microsoft Windows 95
3. Microsoft Windows para Trabajo en Grupo
4. Microsoft Windows NT Workstation
5. Sistema Operativo Microsoft MS-DOS®
6. UNIX**
7. Apple Macintosh**
8. IBM OS/2

Opciones flexibles de licencia.

Microsoft ofrece opciones flexibles y rentables para licenciar Microsoft SQL Server y otros productos de Microsoft BackOffice. Se adquiere una Licencia de Servidor para el producto y una Licencia de Acceso de Cliente para cada estación cliente que accede al producto. Las Licencias de Acceso de Cliente adquiridas para ser utilizadas con la edición estándar Microsoft SQL Server 6.5 pueden también utilizarse con la versión Enterprise Edition. Una Licencia de Acceso de Cliente permite a un ordenador acceder al producto instalado en cualquier servidor de red. Debido a que los modelos de utilización varían, también está disponible una opción de uso concurrente. Este modelo de licencias permite a los usuarios obtener exactamente lo que necesitan para su empresa, y las nuevas herramientas gráficas de gestión de licencias permiten que la opción escogida sea más fácil de administrar.

Ventajas de SQL Server.

El motor de Base de Datos para la Empresa.

- Microsoft SQL Server revoluciona el concepto de Base de Datos para la Empresa. Reúne en un sólo producto la potencia necesaria para cualquier aplicación empresarial crítica junto con unas herramientas de gestión que reducen al mínimo el coste de propiedad. Con Microsoft SQL Server, la empresa tiene todo de serie.
- Miles de Soluciones Disponibles: Libertad de elección, ya que todas las aplicaciones de gestión del mercado corren sobre Microsoft SQL Server.
- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sitios.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server. Posee los mejores registros de los **benchmarks** independientes (TCP) tanto en transacciones totales como en coste por transacción.

- Gestión: Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.

La base de datos para Internet, Internet y Extranet.

- Diseñada desde su inicio para trabajar en entornos Internet e Intranet, Microsoft SQL Server es capaz de integrar los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones "tradicionales". Es más, cada aplicación que desarrollemos para ser empleada en entornos de red local puede ser utilizada de forma transparente -en parte o en su totalidad- desde entornos Internet, Intranet o Extranet.
- Plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server y con las mejores herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.
- Diseñada para INTERNET: Es el único gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática.
- La Base de Soluciones Integradas: La Integración total con BackOffice permite resolver toda las necesidades de infraestructura de la empresa con un sólo paquete.
- Potente y Escalable: Microsoft SQL Server es la única base de datos cuyo rendimiento sobre Internet está publicado, ofreciendo registros espectaculares.
- Mínimo costo de Propiedad: La sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor costo.

Arquitectura RDBMS.

- Arquitectura de servidor simétrico y paralelo con balanceo automático de carga en múltiples procesadores.
- Kernel multithread real para mejor rendimiento transaccional y escalabilidad.
- Soporte grandes bases de datos (VLDB) (+1 TB).
- Completo proceso transaccional interactivo con rollback automático y recuperación de roll-forward.
- Optimizador de consultas mejorado basado en coste.
- Checkpointing mejorado para un mejor throughput de datos y tiempo de respuesta.
- Soporte E/S asíncrono para acceso en paralelo a múltiples dispositivos de disco para un mejor throughput.
- Bloqueo a nivel fija y página con escalación de bloqueos; resolución automática de deadlocks.

Datos distribuidos y replicación.

- Llamadas a procedimientos remotos servidor-a-servidor (procedimientos almacenados remotos).
- Replicación asíncrona o continua basada en registros, o sincronización planificada de tablas point-in-time.
- Configuración de replicación gráfica y características de gestión.
- Replicación de subscritores ODBC, incluyendo IBM DB2, ORACLE, SYBASE y Microsoft Access.
- El Distributed Transaction Coordinator gestiona transacciones que involucran a dos o más servidores SQL (proceso Two Phase Commit 2PC) transparente.
- Replicación de tipos de datos Texto e Imagen.

Desventajas de SQL Server.

La instalación y operación requiere del Internet Explorer (IE) 4.0. Le guste o no, la interfaz del navegador de Web sigue siendo cada vez más habitual, y su uso es lo último en desarrollo de interfaces. Podemos entender por qué Microsoft quiere usarlo con el Servidor SQL, ya que también es un producto de la compañía. Sin embargo, no tenemos ninguna utilidad para un navegador de Web en nuestro servidor de la base de datos, y su instalación es un problema que posiblemente, a más de uno le gustaría evitar. En nuestra opinión, la configuración de los canales de IE 4.0 para las características que nunca habíamos pensado usar nos parece algo frívolo.

La migración requiere un reinicio de la base de datos. El reinicio de todos los datos en una base de datos es un trabajo serio que invita a la potencial pérdida de datos. Cuanto más grande sea la base de datos, más onerosa será esta obligación. Sin embargo, después de mirar las herramientas de migración del SQL Server 7.0, es obvio que Microsoft se ha planteado esta operación como algo muy serio..

Ausencia de integridad referencial declarativa en cascada (DRI). La ausencia de una integridad referencial en cascada podría ser la desventaja más grande del Servidor SQL en comparación con las otras bases de datos dentro del mercado NT. Incluso Access ofrece soporte de este estilo. Se pueden utilizar triggers para compensar esta desventaja, aunque en otras bases de datos esta técnica no es necesaria, así que no es lógico que deba utilizar para trabajar con SQL Server 7.0. Al considerar las otras nuevas características de SQL Server 7.0, es una pena que ésta no esté incluida.

CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN.

2.1. PROBLEMÁTICA ACTUAL.

En los últimos años del siglo pasado nació en México la Industria de Refrescos la cual forma parte de una importante cadena productiva en la que destacan la industria de los envases, azúcar, plásticos, automotriz, refrigeradores, comunicación y las plantas embotelladoras.

Actualmente, existen más de 230 plantas embotelladoras en el territorio nacional para atender a casi un millón de puntos de venta que favorecen la posibilidad de adquirir en cualquier lugar de México un refresco embotellado y frío.

Los principales grupos embotelladores en México del Sistema Ultra-Ciel son: Ultra Ciel LAMSA (KOF), Grupo Azteca (Panamaco), Grupo Continental (CONTAL), Sistema Argos (ARGOS), Procor, Grupo Ponce y Grupo Tampico. (Flesihman).

Las franquicias de Ultra-Ciel son otorgadas a empresarios locales independientes que están autorizados por "The Ultra-Ciel Company" (La compañía Coca-cola) para elaborar y distribuir sus productos.

La empresa embotelladora Ultra-Ciel LAMSA, S.A. de C.V. ("KOF") es el embotellador de productos más grande en México, y uno de los dos embotelladores "ancla" de Ultra-Ciel en Latinoamérica.

Ultra-Ciel LAMSA es una compañía mexicana cuyas subsidiarias operan en 3 territorios: Territorio del Valle de México.- Comprende el área metropolitana de la Ciudad de México, incluyendo gran parte del adyacente estado de México.

Territorio del Sureste.- Comprende los estados de Tabasco y Chiapas y parte de los estados de Oaxaca y Veracruz.

Territorio de Buenos Aires.- Comprende el Distrito Federal de Buenos Aires, Argentina.

La zona del Valle de México y la del Sureste conforman el área de la compañía en México (los "Territorios Mexicanos"). La figura 2.1.1 muestra la distribución de Ultra-Ciel LAMSA así como las principales instalaciones de producción al 31 de Diciembre de 1998.

Las distribuidoras del Valle de México son:

Azcapotzalco	Hixquilucan	Santa Clara
Atizapán	Los Angeles	Texcoco
Centro Histórico	Mega Iztapalapa	Tlanepantla
Coacalco	Mega Los Reyes	Xocongo
Cuautitlán	Mega Vallejo	Xochimilco
Chalco	Mixcoac	Zaragoza

Las Plantas del Valle de México son:

Cedro	Los Reyes	Toluca
Cuautitlán	Tlanepantla	Churubusco
Tlalpan II		

Al 1 de Junio, la compañía tenía 6 plantas de distribución en el valle de México y Toluca. La planta de Churubusco fue cerrada en Enero de 1999.

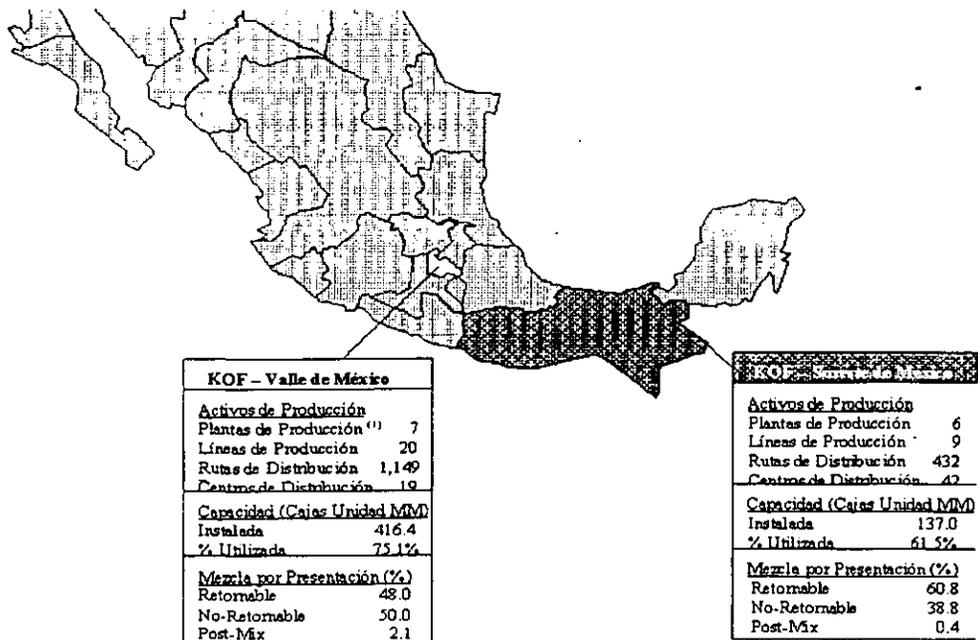


Figura 2.1.1. Zonas Ultra-Ciel.

Las distribuidoras del Sureste son:

Acuyacan	Comitán	Independencia
Yabotecum	Agua Dulce	Villa Flores
Huajuapán	Paraíso	Huatulco
Macuspana	Pijijiapan	Yajalón
Arriaga	Zapata	Ecológica Villa Hermosa
Palenque	Mapastepec	Pto. Escondido
Bochil	Huixtla	Matías Romero
Pujilic	Cárdenas	Escuintla
Motuzintla	San Andrés	Cintalapa
Nochistlán	Tehuantepec	Coatzacoalcos
Ocosingo	Tenosique	Comalcalco
Ocozacuautla	Tuxtla Gutiérrez	Comalapa
Ecológica Tapachula	Ecológica Oaxaca	Ecológica San Cristóbal

Las Plantas del Sureste son:

Ixtacomitán, Tab	Minatitlán, Ver.	San Cristóbal de las Casas
Juchitán, Oax.	Oaxaca, Oax.	Tapachula, Chis

Esta subdivisión en territorios, permite la operación de la empresa de manera que cada una de ellas se encarga de atender la demanda de su región. Dentro de ellos, existe un área corporativa que dicta las normas para la operación de la empresa, y dentro de esta área se encuentra la de Control de Inventarios.

Esta área tiene a su cargo el determinar el volumen de materias primas que se deben adquirir y en que proporción, de tal forma que se tengan estas materias en tiempo y cantidades

oportunas que permitan el buen funcionamiento de las plantas de producción, cuidando que los niveles de inventario no provoquen excesivos costos de almacenaje.

Control de Inventarios es el área responsable de mantener el equilibrio óptimo entre las existencias de las divisiones, así como el desarrollo y manejo de traspasos entre almacenes.

La adquisición de estas materias primas se realiza sobre la base de una proyección de varios meses para poder aprovechar las ventajas de comprar por volumen, pero con entregas parciales programadas para no sobre-inventariar los almacenes.

La Logística.

Para el manejo y distribución de sus productos la empresa creó LAMSA Logística como una división de negocios mas del Grupo LAMSA, para poder atender las necesidades del mercado nacional y ofrecer servicios de valor agregado a la cadena logística.

LAMSA Logística tuvo su origen como la unidad de transporte de Cervecería Cacique Moctezuma, hace 25 años a partir de su creación, se enfocó constantemente a mejorar sus procesos conservando siempre una completa alineación entre sus metas y las de sus clientes.

La importancia que la empresa deposita en su logística se ve influenciada por los siguientes puntos:

- Los canales de comercialización de esta empresa son principalmente abarrotes, misceláneas y hogares con ventas en donde se concentra el 75% del total. Los restaurantes, escuelas, clubes, hoteles y lugares de entretenimiento representan el 24% de las ventas y los supermercados sólo el 1%.
- Su distribución es intensiva y extensiva por la demanda que se genera, y por ser un producto de consumo inmediato, sale de la fabrica y se distribuye en tiendas a nivel detalle. La amplia distribución geográfica de las zonas de venta, implica un mayor grado de complejidad en la definición de las directivas de control.
- El sistema de embotelladoras de Ultra-Ciel tiene más del 60% del mercado mexicano con ventas que superan los 7 mil millones de litros. Con ventas anuales de 12 mil millones de litros de refrescos, México es el segundo mercado más grande en el mundo, después del de Estados Unidos de América
- En el mercado mexicano compiten aproximadamente 100 marcas, teniendo una mezcla de refrescos de 66% en colas y 34% en sabores.
- El 80% de las ventas se realiza en envase retornable.
- Los refrescos dietéticos representan el 2% de las ventas totales de la industria.
- KOF cotiza en la Bolsa Mexicana de Valores, y junto con los grupos CONTAL, ARGOS, GGMEX, GEUPEC y EMVASA generan el 50% de las ventas totales.

La distribución.

La línea que sigue el producto desde la planta distribuidora hasta el consumidor final es muy breve , su consumo es inmediato, este sale de la embotelladora y se distribuye a nivel detalle en los centros de ventas, estos se visitan periódicamente para su abastecimiento, como se muestra en la figura 2.1.2.

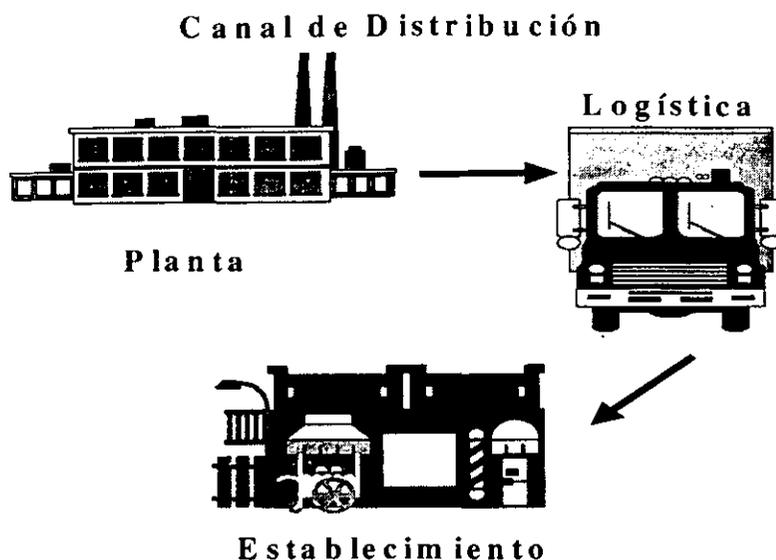


Figura 2.1.2. El modelo de distribución.

Son parte fundamental para este tipo de empresas los canales de distribución y de venta, en la misma medida que lo es su programa de publicidad, los riesgos generados por un control insuficiente se magnifican en un mercado en constante crecimiento y con una poblada competencia por los mercados de consumo.

El manejo de los productos por las líneas de distribución desde las plantas, ya sea tanto de producción como de distribución, presenta continuamente incidencias que manipulan considerablemente la actualización de la información en bitácoras y registros, corriendo riesgos inminentes, tales como el desconocimiento del flujo de la misma.

Cuando existen requerimientos adicionales o de carácter extraordinario se rompe con los esquemas planteados y es necesario un reordenamiento en el menor tiempo posible. Al presentarse estos casos se desata una secuencia de acciones con el fin de restablecer el equilibrio, tales como revisiones físicas de inventario, solicitudes de traspasos, generación de compras extraordinarias o adelantamiento de las fechas de entrega de las materias primas calendarizadas con anterioridad.

Estas actividades se presentan de manera muy habitual, y se cae en problemas como el retardo en los tiempos de entrega tanto de productos, como de materias primas, sobreinventario por cancelación de pedidos, ventas o traspasos no actualizados.

Si se toma en cuenta que el sistema de embotelladoras Ultra-Ciel tiene más del 60% del mercado mexicano con ventas que superan los 7 mil millones de litros y de que el consumo per cápita estimado de refrescos en México fue de 125 litros en 1995, el segundo per cápita más alto en el mundo y con posibilidades de alcanzar al de Estados Unidos que es de 192 litros; se explica la razón por la pugna por encontrar los medios necesarios para ajustarse lo más rápido posible a las demandas del mercado.

El hecho de que se tenga un precio liberado bajo, obliga a la producción y venta de grandes volúmenes por parte de la empresa, que conlleva a realizar esfuerzos en el sentido del

abaratamiento de los costos de elaboración de sus productos y de su colocación dentro de sus regiones.

La situación Actual.

Actualmente para realizar sus funciones, se cuenta con una automatización parcial del proceso diseñado para el Control de Inventarios, una parte importante del calculo de las estimaciones y de la elaboración de reportes, concentrados, etc. se llevan en hojas de calculo de Excel, como se muestra en la figura 2.1.3.

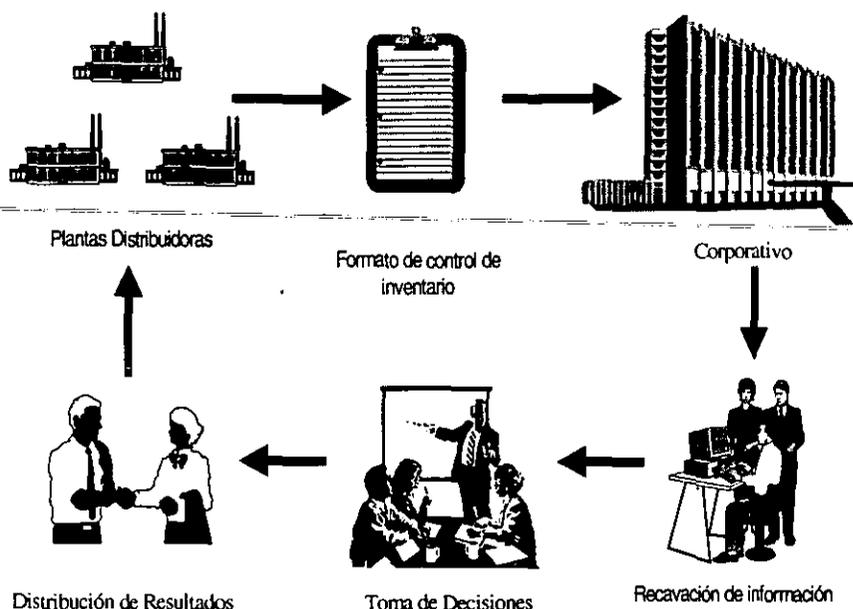


Figura 2.1.3. Esquema de operación Actual.

En la figura 2.1.2, el flujo de la información generada en las plantas distribuidoras, como en las de producción, esta información se recibe vía módem en un formato de hoja de cálculo, para su posterior procesamiento, los reportes que arrojan por resultado los cálculos desarrollados, se turnan al Corporativo, quién a su vez realiza una parte del proceso al fungir como órgano receptor de la información del cual se extrae toda la información de consumos para generar un promedio por almacén y por división.

La rapidez con que fluye esta información para retroalimentar las condiciones iniciales de operación, es significativamente baja y genera una situación desventajosa por parte de la empresa con respecto a sus competidores.

Adicionalmente, las tareas de consolidación se realizan en distintos equipos, con el riesgo de no contar con el concentrado final.

Por el formato de la información ésta es susceptible a la acción de virus informáticos, atentando contra la integridad de los datos almacenados.

Un problema que se presenta continuamente y que retrasa de manera significativa los trabajos de consolidación, es la falta de normalización de las tablas de las hojas de cálculo. Con frecuencia algunos datos de las columnas o renglones se guardan en un formato distinto del patrón normal, este pequeño detalle puede alterar el resultado de un informe global al omitirse cantidades, fechas, o números de claves de los registros, para lo cual se emplea un tiempo adicional en la verificación del contenido de estas tablas, para lograr la uniformidad de los datos, que nos garanticen el 100% de captura y de integridad.

Situación a futuro.

De lo anterior se desprende la necesidad de modificar el modelo de operación para hacerlo más eficiente.

Actualmente se cuenta con un gran número de herramientas de diseño de Sistemas de Información, cuyas características se adaptan a las necesidades de cada empresa en particular, obteniendo desarrollos fáciles de usar y de bajo costo de operación y mantenimiento. De ellos su atractivo principal son las interfaces gráficas que poseen la particularidad de ser amigables con el usuario que las opera, reduciendo el tiempo normal que se invierte en los trabajos de capacitación.

En la figura 2.1.4 se muestra el modelo que se pretende como solución del problema descrito con anterioridad.

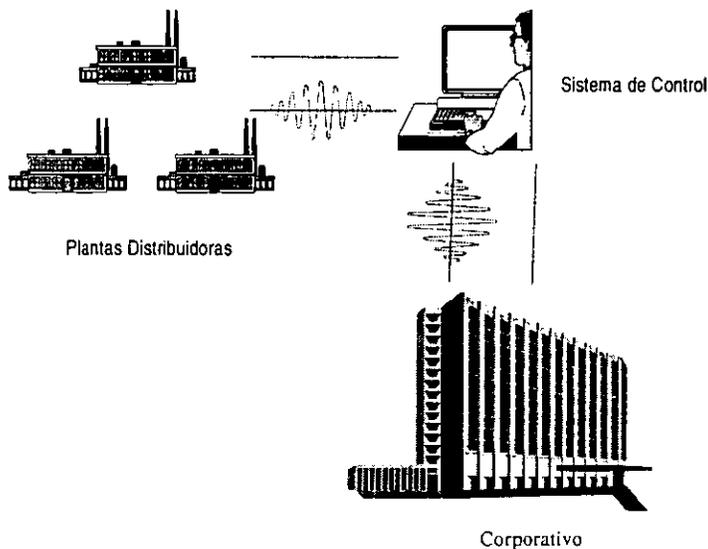


Figura 2.1.4. El modelo propuesto.

En esta situación se plantea el utilizar la organización implementada en el ámbito operativo por la sección de LAMSA Logística, para la organización de los canales y el trazado de las rutas de distribución, que serán susceptibles de modificarse al tenerse los resultados del ejercicio de los primeros periodos de producción estimados, dentro del sistema. Con ello se asegura un cambio gradual de las formas establecidas a la nueva dinámica que se pretende introducir.

Las rutinas del envío de información a través de comunicación vía protocolo TCP/IP por parte de las distribuidoras se mantienen, evitando con ello un gasto adicional en la implementación del nuevo diseño. El cambio substancial reside en primera instancia en el trabajo de concentración de las tablas de trabajo en un solo archivo, este paso por sí mismo economiza un tiempo considerable en desempeño de operación, agregando adicionalmente un rango de falibilidad considerablemente bajo.

El Corporativo podrá contar con los reportes específicos que se requieren en un lapso mucho menor, forzando la toma de decisiones en lapsos cortos generando continuamente las condiciones primarias de producción.

Los reportes generados serán un soporte decisivo en la planeación que realice el área de Presupuestos, tomando las previsiones que se consideren convenientes, manteniendo un estado de las finanzas sano. Punto de esencial importancia, ya que es la imagen de la empresa

la que se refleja en los boletines de sus estados financieros, al cotizar en la banca se debe poseer un alto grado de rentabilidad que atraiga la inversión de nuevos capitales y evita la migración de los ya captados, incrementando la solidez de la empresa.

El objetivo principal consiste en disminuir los tiempos, de procesamiento, de personal especializado y la optimización al 100% de los recursos con que cuenta Ultra-Ciel LAMSA, disminuyendo los costos de inventario. La competitividad del mercado marca la pauta y la diversidad del grupo permite integrar un frente bastante sólido para posibles expansiones dentro del mercado mexicano.

Como objetivos a lograr se enumeran los siguientes:

- Disminución de costos.
- Aumento de Utilidades.
- Plena identificación de mercados, productores consumidores e intermediarios incluyendo su ubicación geográfica.
- Optimización de instalaciones y recursos.
- Elevación de los niveles de producción.
- Eliminar la pérdida de materias primas por un inadecuado manejo de inventarios.
- Afianzar la estabilidad de la empresa mediante la expansión y diversificación.

Como integrante de Fomento Económico Mexicano, S.A. de C.V. (LAMSA), las acciones que se toman para alcanzar estas metas estipuladas, llevan una profunda relación con las actividades de los otros integrantes del grupo LAMSA.

2.2. EL USUARIO Y SUS REQUERIMIENTOS.

El levantamiento de requerimientos para el desarrollo de una aplicación, normalmente involucra el accionar de más de un área dentro de una organización o empresa, esto incide directamente en el modelado del diseño, las demandas del usuario (llamado así en forma genérica) se agrupan de acuerdo al origen donde se generan.

Una vez realizados los trabajos correspondientes. La empresa presenta el siguiente esquema de requerimientos:

- **Generales**
- **Específicos**

Generales.

El usuario requiere el apoyo en el desarrollo de los siguientes puntos básicos.

- Procesamiento de información
- Emisión de reportes
- Toma de decisiones
- Integridad
- Seguridad

Procesamiento de la información.

Consistente en la carga de archivos planos generados tanto por Plantas (Producción, Materia Prima y Recursos Humanos) como de Distribuidoras (Clientes, Equipos, Ventas y Recursos Humanos); estos archivos se reciben vía protocolo TCP/IP y se precisa su consolidación ya sea dentro del sistema o antes de ingresar, siendo lo más adecuado un proceso de integración secuencial para que de esta forma se normalice el formato de entrada de la información.

Dicha información se entrega de manera periódica (bimestral, trimestral, etc.), según su especificación, es decir el área involucrada.

Generación de Reportes.

La empresa requiere reportes de actividades en diferentes periodos de tiempo de tal forma que le permita evaluar lapsos de producción específicos y realizar los ajustes pertinentes.

Este es el punto de partida del proceso de evaluación de cualquier periodo de tiempo de producción, la información se normalizará y se procesará mediante procesos predefinidos en oportunidad y en consideración con el proyecto diseñado.

Toma de decisiones.

Sustentada en la información emitida por los diferentes reportes contenidos en el sistema, la toma de decisiones, ya que con los reportes se puede observar de manera detallada y precisa que distribuidora vende mas, que planta produce más con menos recursos, etc. y de esta manera conseguir el balance que se persigue en la elaboración producción y distribución de los distintos productos que ofrece la empresa al público consumidor.

Requerimientos de Integridad.

La información se debe concentrar en un solo equipo de computo que permita el manejo de consolidados de información para las diferentes áreas. Esto reduce la posibilidad de dispersión de la información y tomar los datos del universo entero. La desventaja que representa la concentración de la información en un solo punto, ante eventuales caídas o catástrofes al equipo de cómputo concentrador, se desvanece con la implementación de los respaldos de la Base de Datos.

Este respaldo se realizará en función de las políticas internas del Area de Sistemas, tomando siempre en consideración la necesidad de contar con la información siempre en línea y en activo.

Requerimientos de seguridad.

Requiere de un manejo de seguridad, el ingreso al sistema mediante un password que lo valide como usuario y que posteriormente limite los derechos del usuario a acceder a los diferentes módulos del mismo, con la finalidad de proteger aquellas secciones que contemplen o desplieguen información clasificada para los usuarios, esta medida permitirá adicionalmente el monitorear la actividad de los movimientos registrados en las tablas de trabajo.

El contenido de algunos reportes representa para la empresa una información de tipo confidencial, al cual sólo un grupo selecto de su personal debe tener acceso. El que se registre el nombre del usuario y este aparezca en los encabezados de las pantallas de captura, representa una medida adicional para concientizar al usuario de la importancia que tiene el poseer un password de acceso a la aplicación y que sus ingresos al mismo son registrados para normar su actividad dentro del entorno de trabajo.

Que las pantallas de captura muestren el usuario que ingresa, fecha y hora en que lo hace.

De esta manera se trata de inhibir la práctica común de compartir claves de usuario entre los empleados, al hacer responsables de los movimientos registrados a las claves que resulten ser las detectadas.

Al definirse todo un esquema de seguridad por parte del usuario, se requiere de un modulo de administración de seguridad, que permita el mantenimiento de los accesos y permisos al sistema, ya que eventualmente se producen cambios y movimientos en el personal de cualquier empresa, teniéndose entre las variantes, el ingreso de personal nuevo, la migración de personal a diferentes áreas, y el término de la relación laboral entre empleados y la empresa. Estos aspectos obligan a una constante revisión de la plantilla de personal que posee una clave

de acceso al sistema y de los derechos que posee dentro del sistema, esta tarea de mantenimiento de la seguridad de la aplicación se deberá complementar con las políticas del Administrador de Red del Área de Sistemas, para elevar el nivel de seguridad demandado por el usuario.

La eficiente definición de permisos sobre los depósitos de almacenamiento de las tablas de información, inhibe la posibilidad de que usuarios con un cierto nivel de cultura informática y poseedores de las herramientas necesarias fuera del personal de desarrollo puedan acceder a la información almacenada por otros medios alternos y estén en posibilidad de modificarlos.

Específicos.

Los requerimientos específicos se ubican en dos divisiones, aquellos que involucran la plataforma de trabajo presente instalada y de los requerimientos en particular que cada área demanda.

- De implementación
- Por área

Requerimientos de Implementación.

El ambiente debe poseer una interfaz gráfica, compatible con el sistema operativo Windows, Windows 95 o Windows 98. La plataforma principal de la empresa se basa en los productos derivados de la Interfaz Gráfica desarrollada por la empresa Microsoft, por lo cual se requiere asegurar la compatibilidad de la aplicación con su entorno, y que los equipos de cómputo de cualquier punto de la empresa, en un momento dado, puedan convertirse en un cliente del sistema, sin realizar labores adicionales a su configuración e instalación.

Es importante tomar en cuenta esta consideración, por la diversidad de equipos con que cuenta la empresa y cuyos recursos son variados, por lo que se hace necesario establecer un valor promedio para definir las capacidades y requerimientos mínimos de hardware, al realizar el desarrollo correspondiente.

Normalmente se presentan problemas de resolución, de velocidad de procesamiento y de versiones distintas de la plataforma del software de sistema operativo utilizado, que repercuten en el desempeño de la aplicación generada.

Debe de ser simple y amigable, para que no se requiera personal altamente calificado que lo opere, así como en el desarrollo de sus procesos que sean de fácil comprensión para el usuario.

La empresa desea facilitar su uso a los altos directivos de sus áreas estratégicas, cuya cultura informática no tenga un gran nivel, al automatizarse las tareas de mantenimiento diseñadas para el sistema, sólo basta con que un operador calificado las ejecute observando los lineamientos predeterminados para tal efecto.

El sistema debe ser rápido en su acceso al mismo y permitir la emisión de rutinas y reportes cuantas veces sean necesarios sin tener que realizar rutinas especiales para su elaboración.

El diseñar procesos de baja complejidad repercute directamente en el aspecto de capacitación del personal que lo opera. La curva de aprendizaje disminuye en la medida que las complejidades desaparecen, se reduce el margen de error por parte del usuario que se encuentra normalmente en constante contacto con las aplicaciones que se desarrollan y que gradualmente va adquiriendo un conocimiento mayor de su operación, que en algunos casos se sobrepasa al de los diseñadores.

La incorporación eventual de nuevos operadores, también se ve beneficiado de este aspecto ya que forma un punto de calidad en la aplicación.

Al elegir una plataforma cliente servidor como se muestra en la figura 2.2.1. , Se asegura que la información resida en un ente concentrador que mantendrá la integridad de los datos apoyado en las políticas de respaldo y mantenimiento. Los clientes se concentrarán exclusivamente en el uso de los datos almacenados aprovechando todas las bondades que el modelo ofrece.

Debe permitir la concentración de la información que provienen de las comunicaciones vía protocolo TCP/IP por parte de las distribuidoras, la normalización de la información de entrada al sistema es la base de operación del mismo, los archivos enviados por las diferentes instancias conformarán las tablas de trabajo para realizar un periodo de operación. Esto con el fin de aprovechar la logística básica desarrollada por la sección especializada de la empresa como se muestra en el mapa de la figura 2.2.2.

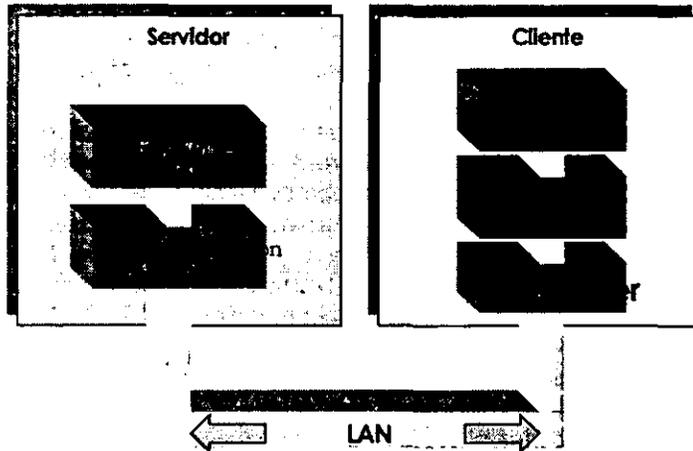


Figura 2.2.1. El Modelo Cliente Servidor.

La distribución territorial de las fuentes de información se encuentra concentrada en dos grandes puntos pero a los cuales los separa un espacio considerablemente amplio para facilitar el contacto directo.

El uso de los medios de transmisión electrónica reduce los tiempos necesarios para el envío y recepción de la información requerido en tiempo y modo, permitiendo poseer en un lapso considerablemente corto las incidencias completas, del ejercicio de producción anterior. La posibilidad de rectificar informes de último minuto también se ve favorecido por el uso de los medios.

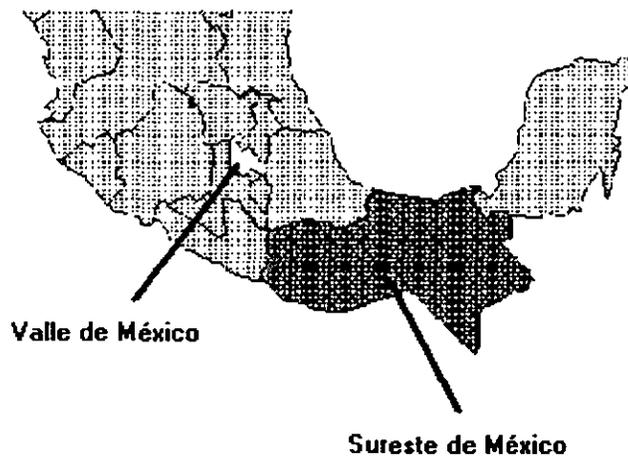


Figura 2.2.2. Distribución Geográfica.

El sistema debe permitir el almacenamiento de grandes volúmenes de información, incluyendo periodos tan largos como un año de operación, respaldo de la información, y de los movimientos del día, en caso de pérdida parcial o total de los registros contenidos en la Base de Datos.

El crecimiento de la misma esta regulado por el Manejador de Base de Datos elegido por la empresa, (Debido a que ya cuenta con las licencias correspondientes) y su eventual crecimiento se encuentra estimado y evaluado para no correr un riesgo innecesario que afecte la operación del sistema.

Ocasionalmente las consultas realizadas por parte de los usuarios son de periodos anuales completos incluyendo los anteriores, por lo cual se requiere que esta posibilidad sea factible y no represente un problema técnico al presentarse este requerimiento por parte del usuario.

El sistema deberá contemplar una gran riqueza en cuanto a emisión de reportes para las distintas áreas que se verán beneficiadas de la información generada por la actividad del sistema.

Los reportes son la parte final y el producto de la actividad del accionar de la aplicación, estos revisten una importancia adicional para la empresa ya que incluso estos mismos son parte de un proceso de toma de decisiones denominado Cubo de Decisiones.

Estos deberán poderse imprimir tanto en impresoras de inyección de tinta como en impresoras láser, sin perder la configuración de diseño y calidad de imagen, no se requiere el uso de impresoras de matriz de punto, debido a la ausencia de requerimientos por parte del usuario del uso de formas continuas para formatos preimpresos.

Todos deberán llevar el logotipo distintivo de la empresa así como la leyenda que hace alusión al contenido de la información impresa en la forma, la fecha y hora de emisión.

Estos reportes se deberán exportar adicionalmente a Hojas de Calculo de Microsoft Excel (de la versión estándar que posea el cliente), como se muestra en la figura 2.2.3.

La particularidad de exportación, radica en la necesidad de la empresa por emitir reportes ejecutivos en esta herramienta, a partir de los resultados obtenidos del procesamiento de la información recabada, y el manejo subsecuente de la misma en niveles descendentes.

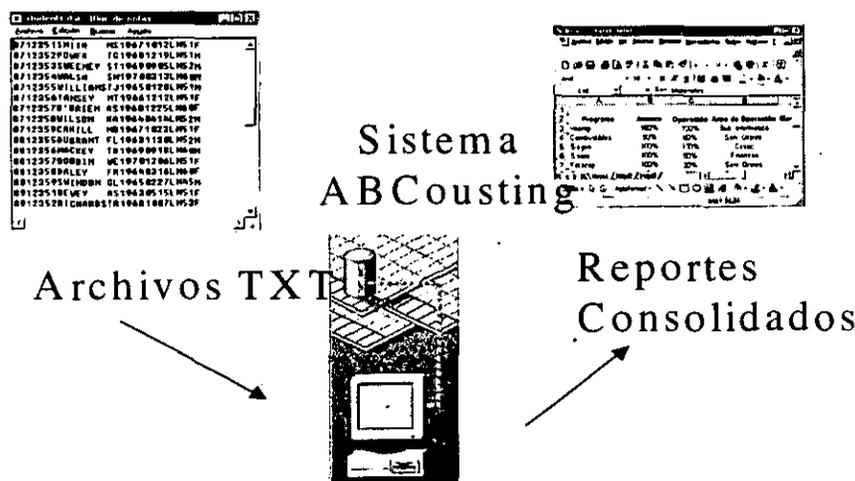


Figura 2.2.3. El procesamiento de la información.

Por área.

- Almacén.
- Recursos Humanos.
- Mantenimiento.
- Informática.

Almacén.

El área de almacenes e inventarios requieren información constante sobre los niveles de stock existentes al día, así como información referente a los sistemas PEPS (Primeras Entradas - Primeras Salidas), y UEPS (Últimas Entradas - Primeras Salidas), concerniente tanto a materia prima como jarabes, empaques, glucosa, etcétera. ; y de producto terminado por líneas de productos y marcas.

Recursos Humanos.

Requiere de mayor control sobre el manejo de la plantilla de personal, en aspectos como la distribución de personal en las diferentes áreas que integran el grupo administrativo y operativo de la empresa, así como del perfil de cada uno de los integrantes y los puestos que integran dicha unidad económica.

De esta forma el área de recursos humanos podrá contar con los elementos necesarios para realizar los proyectos de expansión, integración y capacitación propios de cada área.

Mantenimiento.

La información requerida por esta área va en función de la totalidad del equipo con el que cuenta la empresa para la elaboración de sus productos y de los procesos de producción se desprende la necesidad de realizar tareas conjuntas de acuerdo a los periodos establecidos por la unidad de planeación con la finalidad de optimizar y alargar la vida útil de la infraestructura y así mismo de proyectar las adquisiciones que se consideren pertinentes.

Informática.

Informática requiere de un proceso de depuración de su información almacenada, que permita disminuir el volumen acumulado de almacenamiento y eliminar aquella considerada sin valor, realizar cargas manuales de valores adicionales a los procesos iniciales como producto de las actualizaciones constantes del flujo de información manejado.

Adicionalmente requiere de los siguientes manuales:

- Manual de Usuario.
- Manual Técnico.
- Guía de Fallas.

Manual de Usuario.

Este manual deberá ser una guía de los pasos que se deben llevar a cabo en el proceso de evaluación de un periodo determinado de producción.

El manual deberá ser totalmente explícito y gráfico, definiendo con claridad los procedimientos a seguir en el manejo de cada uno de los módulos creados para la operación de la aplicación por parte del operador.

Cada Submenu del sistema se explicará ampliamente, indicando la manera de realizar las operaciones básicas que se pueden realizar, de ir acompañadas de una breve explicación del propósito del módulo en cuestión y de ser el caso del orden que guarda en la serie de procedimientos articulados para la óptima operación.

Manual Técnico.

Este documento permitirá a los administradores del sistema conocer el detalle del desarrollo de la aplicación, proporcionando información desde la configuración de los equipos del cliente para la conexión a la Base de Datos que se encuentra en el servidor, como también la estructura de la Base de Datos, desde el nombramiento de las tablas y sus campos con sus

correspondientes tipos de datos y todas aquellas funciones y procedimientos necesarios para el funcionamiento de la aplicación.

Guía de Fallas.

Se deben incluir un listado de las posibles fallas o errores por los cuales la aplicación podría no tener un buen funcionamiento, como así de sus correspondientes soluciones.

Con la finalidad de orientar al usuario en el aprendizaje del manejo de los errores de operación de los de diseño.

Esta guía deberá ser concisa y seriada en cuanto a sus posibles orígenes, de tal forma que su manejo sea eficaz.

2.3. BÚSQUEDA Y ANÁLISIS DE LA INFORMACIÓN.

Una vez determinada la problemática actual que vive la compañía y las necesidades de los usuarios así como sus requerimientos, contamos con bastantes elementos para poder realizar un análisis de toda esta información y dar forma a lo que en un futuro será la propuesta de solución. En este caso utilizaremos la observación como técnica para el análisis, ya que, para los capítulos anteriores se ha requerido de visitas de campo, entrevistas y cuestionarios para conocer el estado actual y sus deficiencias o necesidades.

Como una necesidad de mantener el control de la información generada en las plantas de producción y los canales de venta, es necesario crear una metodología que integre toda información generada que ayude a supervisar el rendimiento y eficiencia de la producción identificando cual o cuales son las plantas productivas con mayor capacidad para producir, al mismo tiempo, identificar cual es el mejor canal de venta y así lograr elevar la competitividad y ganancia de la empresa.

Para ello es necesario integrar en el menor tiempo posible y de la manera mas segura los datos generados en cada uno de estos canales considerando intervalos de tiempo predeterminados. Por otro lado la información que se envía debe tener una ruta de acceso para cargar al sistema general y de esta forma la información generada como resultado de este proceso tenga congruencia.

La información contenida en los reportes empleados para la compañía tanto de los canales de venta como de las plantas se detalla como sigue:

Archivo de existencias. En este archivo se tiene información referente al material que existe en cada uno de los almacenes, contando con la clave del almacén, clave de material, la cantidad de material en existencia y si hay en reserva; el contenido del archivo es como se muestra a continuación en la tabla 2.3.1.

ALMACEN	NUM_MAT	CANTIDAD	RESERVADOS	DISPONIBLE
01	0110207	3776.000	0.000	3776.000
01	0110403	100.000	0.000	100.000
01	0110458	222.000	0.000	222.000
01	0110567	60.000	0.000	60.000
01	0110786	340.000	0.000	340.000
01	0119876	237.000	0.000	237.000
01	0110665	237.000	0.000	237.000
01	0110765	876.000	0.000	876.000
01	0110458	328.000	0.000	328.000
01	0110725	666.000	0.000	666.000
01	0110987	89.000	0.000	89.000

Figura 2.3.1. Contenido archivo de existencias.

Descripción de los campos que conforman el archivo de existencias, que se muestra en la tabla 2.3.1.

Campo	Descripción
ALMACEN	Define la clave asignada para cada uno de los almacenes.
NUM_MAT	Define la clave de material, esta clave es la que se tiene definida en el catálogo de almacenes.
CANTIDAD	Define la cantidad de material que se tiene en el almacén.
RESERVADOS	Define la cantidad de material que se tiene en reserva por las necesidades de alguna materia prima especial o la demanda de la misma.
DISPONIBLE	Es la cantidad de material que en un momento dado se puede disponer para uso interno del almacén o en su defecto para traspaso a las plantas de producción.

MATERIAL	DESCRIPCION	ALMCEN_DESTINO	COMPRA_POR_MES
1225001		06	4000.000
1123455		07	975.000
2335454		30	567.000
1234565		12	67.000
3324566		67	5.000
3456755		80	56.000
1123332		34	456.000
1324567		23	45.000
2345643		80	33.000
0020034		15	1234.000
0034566		23	1256.000
0123434		30	1265.000

Figura 2.3.2. Contenido archivo de peticiones de compra de materiales.

En el área de compras se utiliza la información generada por el control de inventarios, que es la petición de materias primas o materiales a comprar para cada almacén o planta de producción como se muestra en la tabla 2.3.2.

Campo	Descripción
MATERIAL	Clave de material según catálogo
DESCRIPCION	Descripción del material.
ALMACEN_DESTINO	Clave del almacén donde se requiere la materia prima o material.
COMPRA_POR_MES	Las compras que deben efectuarse cada mes.

Figura 2.3.3. Formato de Control de Producción.

Para el control de la mercancía en almacén es necesario un reporte de control de producto como lo muestra la figura 2.3.3.

Campo	Descripción
CLAVE	Clave de Producto según catálogo
DESCRIPCION	Descripción del Producto.
PRODUCTO ACUMULABLE	Clave del producto si es acumulable.
CAJAS POR PALLETTE	Cantidad de cajas empacadas por pallette
FACTOR CUBICO	Factor para determinar el volumen
CAJAS	numero de descripción de las cajas
MARCA	clave de la marca del producto
EMPAQUE	clave de tipo de empaque empleado

Figura 2.3.4. Formato de control de ruta.

Para la distribución en tiempo se utiliza el reporte de Ruta como lo muestra la figura 2.3.4.

Campo	Descripción
CLAVE DE RUTA	Clave de la Ruta
UNIDAD OPERATIVA	Descripción de la unidad operativa empleada.
TIPO DE VENTA	Clave de el tipo de venta

Para el control de la Materia Prima y su manejo se utiliza la forma que aparece en la figura 2.3.5.

Campo	Descripción
CLAVE	Clave de materia Prima catálogo
DESCRIPCION	Descripción de materia prima.
MEDIDA	Medida de la materia prima

Figura 2.3.5. Formato de control para la Materia prima

En el manejo de la Unidad operativa se utiliza el reporte que se muestra en la figura 2.3.6.

Campo	Descripción
CLAVE ABC	Clave de ABC según catálogo
CLAVE FUENTE	Clave asignada de fuente.
DESCRIPCION	descripción de la unidad
TIPO(PLANTA / DISTRIBUIDORA)	tipo de unidad operativa.
CLAVE RECURSOS HUMANOS	Clave asignada por recursos humanos.
CLAVE OROS	Clave de oros
DESCRIPCIÓN OROS	Descripción de oros
ZONA	Código de la zona asignada

Figura 2.3.6. Formato de unidad Operativa.

Para el reporte de consumo teórico se genera el de la figura 2.3.7.

Campo	Descripción
UNIDAD OPERATIVA	Clave de la unidad operativa
PRODUCTO	clave del producto.
MATERIA PRIMA	Clave de la materia prima.
CONSUMO	cantidad del consumo por unidad de tiempo.

Debido a la cantidad de materiales que se manejan en la empresa se requerirá de un área que permita organizar y centralizar la información de todos los materiales que en un momento se requiera transportar entre los diferentes almacenes. Es decir, cuando en un almacén o planta de producción se requiera de algún material que exista en otro almacén y del cual se pueda traspasar para cubrir las demandas de éstos, es necesario contar con una logística para su transportación y traslado.

Figura. 2.3.7. Formato de Consumo teórico.

En el área de control de inventarios se generan los reportes de todos los materiales que en un momento dado se requiere traspasar, el área de logística utiliza esta información para la organización y control de los diferentes movimientos de materiales que se tengan que efectuar entre los almacenes y las plantas de producción.

2.4. IDENTIFICACIÓN DEL PROBLEMA.

Como gran parte de las empresas que venden sus productos a una amplia variedad de clientes y que con procesos manuales, manipulan un considerable volumen de información; se corren riesgos inminentes, tales como:

- el desconocimiento del flujo de la misma
- su explotación e interpretación incorrecta
- su archivamiento inadecuado
- la imposibilidad de llevar a cabo cualquier cambio en el esquema del funcionamiento actual; que sin duda afectan el rendimiento general de la Organización.

Es necesario por tanto, identificar los puntos en donde tales empresas se encuentren con obstáculos que les impidan llevar un control interno eficaz.

Los riesgos de enfrentarse frecuentemente con problemas generados por un control insuficiente de la información, no son deseables para ningún proveedor y mucho menos si tomamos en cuenta que se encuentran inmersos en un mercado con una demanda en continuo aumento y con una nutrida competencia, que a su vez tratará de esforzarse no sólo de evitar sus propios problemas sino de encontrar los medios necesarios para no quedarse atrás y ajustarse lo más rápido posible a los cambios que se puedan presentar.

La necesidad de una correcta interacción del proveedor y los clientes resulta más evidente cuando se llegan a presentar problemas que los hagan entrar en contradicciones, los cuales pueden repetirse de un cliente a otro e ir incrementando el grado de descontrol de una o ambas partes. En el caso del proveedor, las diferencias que se tengan con un cliente no serán evidentes al momento de cumplir con el pedido, sino que muchas veces, éstas pueden tener tras de sí un largo tiempo de haberse efectuado, por lo que para su aclaración habría que remontarse atrás un tiempo equivalente, para muchas veces llegar a la conclusión de que todo fue provocado por un error interno. Sin embargo, cualquiera que sea el caso que ocasione el error, se debe procurar una respuesta aceptable en ambas partes y que en un futuro no se vuelva a repetir o su ocurrencia sea mínima. No es sino cuando ocurren los problemas, que son más evidentes las deficiencias de una o las dos partes a la vez, muchas veces por incompatibilidad en las fuentes de información que fluyen entre ambas, y otras tantas por deficiencias de control interno propias de cada una. Obviamente las carencias para enfrentar los problemas comunes que involucren a un proveedor y a un cliente, no deberían tener una solución aislada de alguna de las partes, si es que se desea corregir de verdad; si no más bien una colaboración conjunta pudiese llevar a una solución más efectiva. Puede entonces surgir la necesidad de formar un medio común a través del cual puedan establecer un puente de comunicación común que les facilite la detección y corrección oportuna de sus problemas, mucho antes de que estos se agraven.

El procesamiento manual de datos puede complicarse con el transcurrir del tiempo por el incremento en la cantidad de información que se tiene que manejar para:

- registrar la entrada/salida de productos de un almacén debido al cumplimiento de algún pedido
- elaborar de forma continua un inventario de las existencias
- llevar un control de los clientes, entre otras cosas, que desembocarán en las expectativas del negocio dependiendo en gran medida del balance de esta información complementada con su situación financiera.

De lo anterior, podemos decir que se trata de los puntos en donde se tienen que enfocar los esfuerzos para agilizar tanto los tiempos de respuesta como el control que cada uno requiera. La dinámica de un almacén puede arrojar datos muy interesantes desde el punto de vista del control, que pueden repercutir en datos que proporcionen información sobre costos de ciertos productos que no se consuman del todo, es decir, que permanezcan en el almacén un determinado tiempo tal que se deprecien con el transcurrir del tiempo, siendo que otros

representa dinero invertido que se sumará al que se posea en efectivo para arrojar en general el estado financiero de la entidad.

En ocasiones los inventarios son considerados sólo como un conteo de productos siendo que pueden contribuir a representar la verdadera situación económica que guarda o guardó un Proveedor. Así como para el almacén pueden ser utilizadas herramientas manuales como las tarjetas de consumo, en el proceso de inventario se puede utilizar una herramienta análoga que permita comunicar a las partes correspondientes sus resultados, que posteriormente serán interpretados, o en todo caso, corregidos, para pasar finalmente a archivarlos. Algo muy importante que se debe asentar en una hoja o forma de inventario es el costo original de una mercancía, así como el tiempo que se cree que estará almacenada, en caso de que ésta tienda a permanecer por un tiempo en el almacén, ya que estos datos son muy útiles para el cálculo de la depreciación. No es raro que en ocasiones se requiera consultar estas formas de inventario de mercancías para rendir alguna aclaración o también para generar información estadística. En la figura 2.4.2 se muestra un diagrama que señala el curso para la elaboración del inventario.

De los procesos de almacenaje, proveeduría y de inventario se puede destacar un punto desfavorable que es inevitable durante su desempeño y que resulta necesario si es que ambos procesos se desempeñan en forma manual, éste es el de la redundancia en la captura de datos, es decir, dentro del almacén es necesario llevar un registro de este producto el cual será nuevamente registrado para asentar su estado en la forma correspondiente por lo que, tanto el producto, como sus atributos, tendrán que ser apuntados por lo menos una vez durante cada inventario. Si ahora, se incrementa el número de mercancías, esto resultaría cada vez más complicado y requeriría, como se mencionó, mayor consumo de recursos humanos y financieros así como la tendencia a cometer más errores. Si no se tiende a caer en este doble registro. También puede presentarse el caso en que la decisión de entrada o no a un concurso para proveer de alguna materia prima a la institución dependa de la rapidez y confiabilidad con la que ambos procesos proporcionen su información para complementar a la empresa y generar una evaluación definitiva que facilite esa decisión.

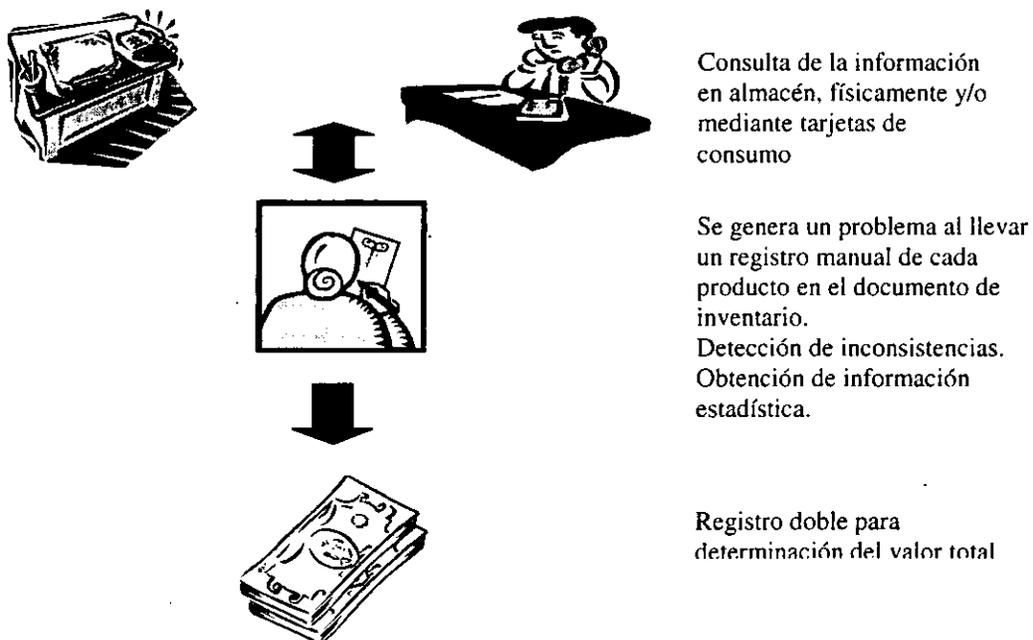


Figura 2.4.2. Diagrama que señala el curso para la elaboración del inventario.

El suministro a los clientes por parte del proveedor implica cumplir una serie de requisitos muchas veces dictados por instituciones gubernamentales y otras veces establecidos de común acuerdo entre el vendedor y el comprador.

Una vez más el proceso manual de información que se maneja en ambas partes durante la realización del pedido por el consumidor y su cumplimiento por el proveedor, pueden generar inconsistencia al momento de cotejar la información de su contraparte, provocando en su caso, una o varias devoluciones de pedidos completos que requerirán de una revisión y evaluación cuidadosa del proceso de suministro y/o distribución de productos. Si sumamos a estos la diversidad de ventas que se pueden hacer: ventas a mayoristas, minoristas, ventas a instituciones e inclusive exportaciones; así como la diversidad de clientes de cada uno de estos tipos de venta y en donde cada uno puede proporcionar una lista propia que contenga los requisitos necesarios para su satisfacción; resulta cada vez más complejo adecuarse a esa variedad e implica tener un conocimiento particular de cada cliente y de los productos que consume, dicho conocimiento se adquirirá sólo con base en el trato y la experiencia con él, lo cual no es deseable por ningún proveedor, aunque estos pedidos contengan frecuentemente los mismos tipos de productos cada vez que se solicitan variando sólo en la cantidad; sin embargo no siempre suele ser así ya que otras veces tanto la cantidad y frecuencia de pedidos como la diversidad de productos incluidos, dificulta el proceso de suministro y por lo tanto de cotejo para ambas partes. Al cliente no basta con satisfacerle su cuadro de requisitos establecidos, sino que además se deberán cumplir los tiempos de respuesta en caso de contingencia, las consultas o aclaraciones serán aceptadas ya que pueden ser determinantes para que el proveedor sea incluido dentro de la lista de las instituciones consumidoras. Siempre que el vendedor carezca de un buen funcionario interno que le permita reaccionar en caso de fallas, tenderá a recurrir continuamente a estos obligados para sobrellevarlas.

Los problemas presentados para una empresa de este tipo pueden tener variantes dependiendo sobre todo de su esquema operativo, sin embargo los definidos aquí pueden aplicarse a la mayoría de los casos. El problema fundamental radica en la integración adecuada de la información generada en los diferentes departamentos, su correcta explotación y sobre todo la interpretación y aprovechamiento que de ésta pueda efectuar la parte consumidora. Es importante apuntar que el incremento del procesamiento manual de la información, no implica necesariamente caer en la obsolescencia y en la ineficacia, aunque resulta evidente que conforme un proveedor incrementa la variedad de sus productos, de sus ventas y de sus clientes requerirá de inyectar mayores recursos que le permitan sostener su esquema o modificarlo de tal forma que no frene su crecimiento.

2.5. ÁREAS INVOLUCRADAS EN EL PROCESO.

En la empresa existen direcciones divisionales, las cuales son las responsables de atender todas y cada una de las demandas de materiales de la misma, para lograr esta función, en cada una de las divisiones existen las siguientes áreas:

- Área de Control de Inventarios.
- Área de Compras.
- Área de Contabilidad de Planta.
- Área de Auditoría.
- Área de Almacenes.

Área de Control de Inventarios. En esta área se realizan todos los procesos necesarios para determinar la cantidad de material que debe comprarse, para los diferentes almacenes, como se muestra en la figura 2.5.1 nota de entrada al almacén.

Problemática para el área de control de inventarios.

Para el área de control de inventarios, se sabe que la información es generada en el sistema y esta agrupada en archivos, la obtención de estos archivos se tiene que hacer de manera manual.

Área de compras. En esta área, una vez realizados los procesos y reportes de compra para los diferentes almacenes, estos reportes son analizados para realizar los pedidos de compra, y la asignación a proveedores, en las figuras 2.5.2 y 2.5.3 se muestran formatos utilizados en esta área.

En la figura 2.5.4 se muestra el diagrama de bloques de esta área.

Nota de Entrada al almacén			
Fecha _____		Número _____	
Recibido de _____		Orden de Compra Núm _____	
Cantidad	Unidad	Descripción	Observaciones
Recibió _____		Autorizó _____	

Figura 2.5.1. Nota de entrada al almacén.

NOMBRE DEL NEGOCIO					
Requisición de Compra					
Fecha _____			Número _____		
ARTICULO	Existencia en Almacén	Motivo de compra	Cantidad Solicitada	Precio Unitario	Observaciones
Recibió _____			Autorizó _____		

Figura 2.5.2. Requisición de Compra.

OMBRE DEL NEGOCIO México, D.F.	ORDEN DE COMPRA Núm. _____ Este número debe mencionarse en toda Correspondencia, factura, bultos, etc. Fecha: Entrega en: Fecha en que se requiere: Orden de Pago: Sírvase surtir las siguientes mercancías sujetas a las condiciones mencionadas al reverso:
Firma	_____

Cantidad	Unidad	Descripción	Precio	Importe

Figura 2.5.3 Orden de Compra o pedido.

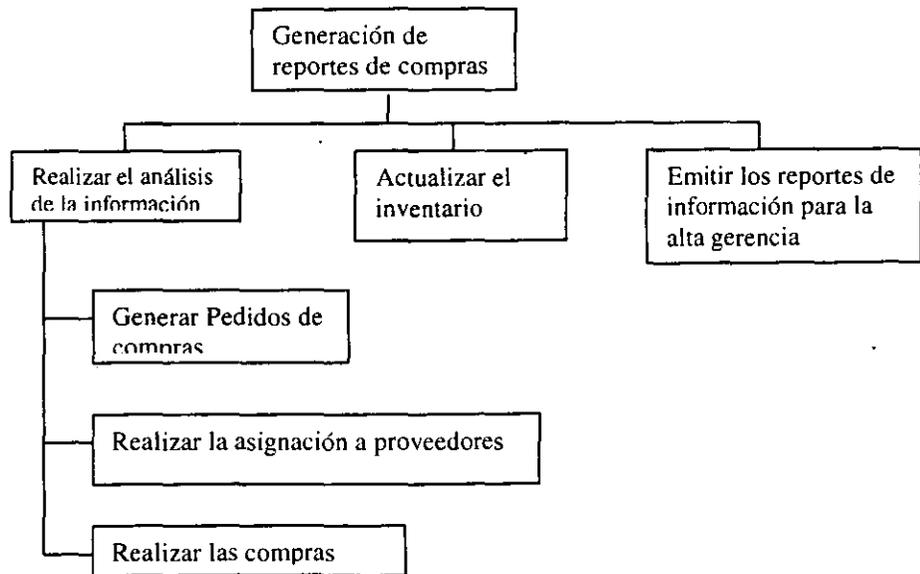


Figura 2.5.4. Diagrama de bloques del área de compras.

Área de contabilidad de planta. La función principal en esta área es analizar los consumos que se tienen en los diferentes almacenes y de esta manera tener el control de los gastos realizados en los mismos. En la figura 2.5.5 se muestra el diagrama de bloques de esta área.

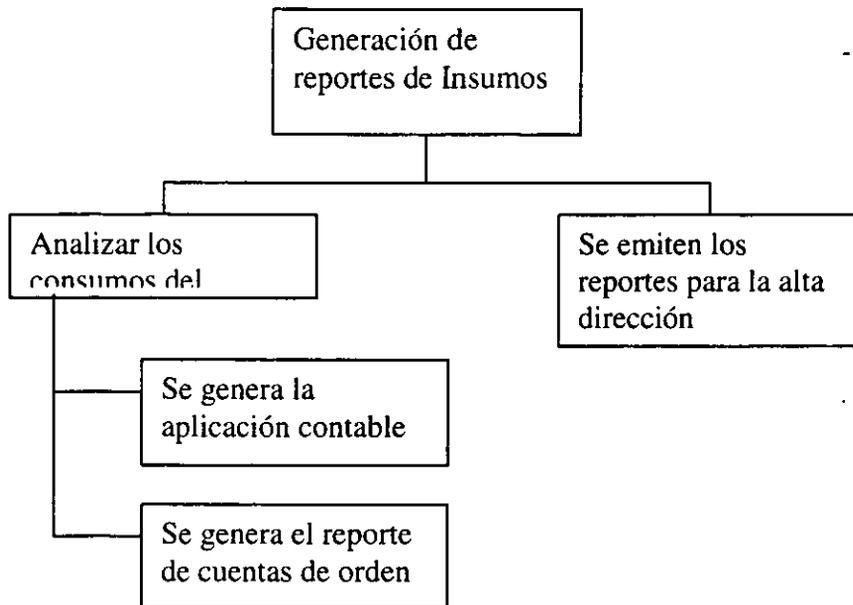


Figura 2.5.5. Diagrama de bloques del área de contabilidad.

Área de auditoría. En esta área se realizan las funciones de supervisar todas y cada una de las tareas realizadas por las áreas antes mencionadas, con el fin de que todo se realice en orden y en función de los procedimientos definidos por la empresa, en la figura 2.5.6 se muestra el diagrama de bloques de esta área.

Área de almacenes. Estas son las entidades de la empresa donde se encuentran los diferentes materiales, en ellas se generan las peticiones de compra y en otros casos el traspaso de materiales.

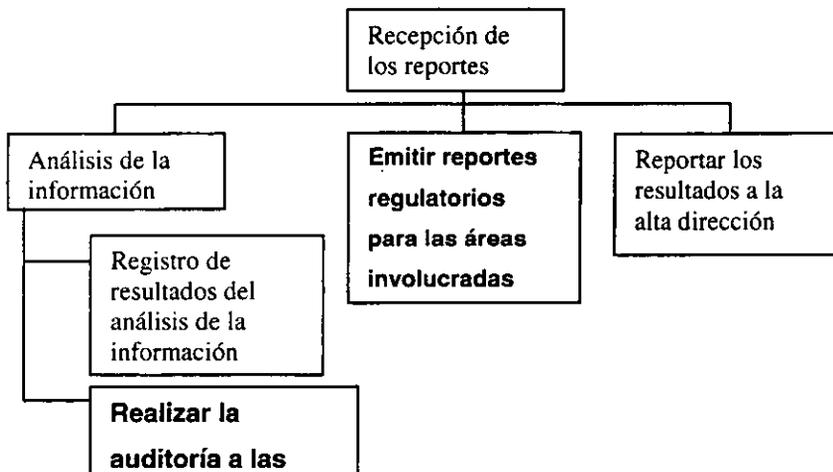


Figura 2.5.6. Diagrama de bloques del área de auditoría.

Área de control de inventarios. La información que se utiliza en esta área se deriva de las bases de datos del sistema.

En la empresa hay una área de sistemas responsable de la operación y control de este sistema, y en esta área se genera información de vital importancia para el área de control de inventarios, esta información esta constituida en tres archivos y son:

El archivo de existencias. En este archivo se tiene información referente al material que existe en cada uno de los almacenes, contando con la clave del almacén, clave de material, la cantidad de material en existencia y si hay material en reserva.

Área de compras. En esta área se utiliza la información generada por el área de control de inventarios, que es la petición de materiales a comprar para cada almacén, en la figura 2.5.8 se muestra un diagrama de bloques de esta área.

Cuando el área de compras crea necesario, solicitará información adicional para la generación de algún pedido en particular, genera un archivo en Excel como se muestra en la tabla 2.5.4, donde se contemplan todos los materiales que requieran ser analizados. Para efectuar este análisis es necesario tener cierta información que el área de control de inventarios debe proporcionar y es:

- Cual es el consumo.
- Cual la existencia.
- Cual es la existencia nacional.
- Que pronósticos se tienen adicionales.
- Etc.

MATERIA PRIMA	ALMACEN_DESTINO	COMPRA_POR_MES
AZUCAR	01	4000.000
COLORANTES	02	975.000
GAS CARBÓNICO	10	630.000
LATAS	08	630.000
ENVASES ½ Lt.	12	3.000
ENVASES ¾ Lt.	30	2043.000
ENVASES 1 Lt.	07	98.000
ENVASES 2 Lt.	08	1550.000
ETIQUETAS	03	12699.000
CORCHOLATAS	107	12699.000

Tabla 2.5.7. Contenido de archivo de análisis de compra.

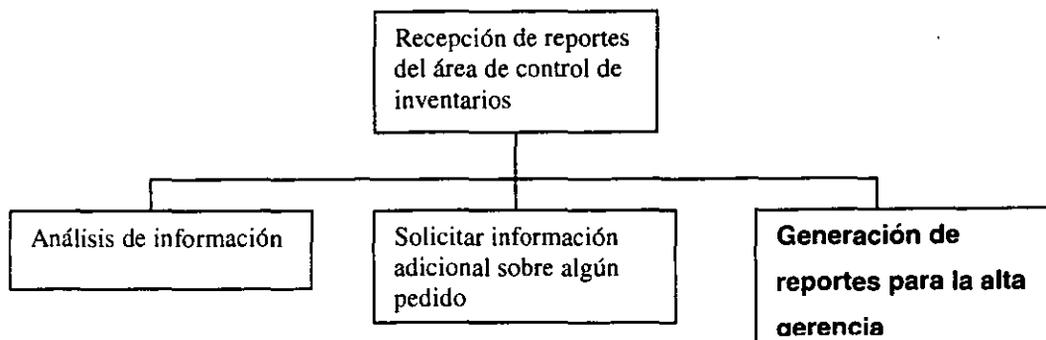


Figura 2.5.8. Diagrama de bloques del área de auditoría.

Área de contabilidad de planta. En esta área la información que se maneja la podemos dividir en dos grupos, el primero sería el de verificar y evaluar la cantidad de material que se está consumiendo en cada almacén, como se muestra en la figura 2.5.9 la existencia y los pedidos de los mismos; para efectuar esta tarea se utiliza el sistema, para poder proporcionar a el área de control de inventarios los niveles de inventarios en cada uno de los almacenes. El segundo grupo es la responsabilidad de generar el archivo de precios por material, que es utilizado por las áreas de compras y control de inventarios, como se muestra en la figura 2.5.10.

Área de auditoría. Esta área como se menciona al principio, la función principal es de verificar que todas las tareas de las áreas antes mencionadas, se realicen conforme a las normas establecidas por la empresa.

La información que utiliza esta área, es un compendio de información de las otras áreas como puede ser: listados de precios, catálogos de materiales, contratos, normas de materiales, garantías, etc.

Cuando a esta área se le solicita que efectúe una auditoría en alguna otra área en particular referente a materiales y que pueden ser en: uso de los mismos, contratos, garantías, cantidades asignadas, etc.

(NOMBRE DE LA NEGOCIACIÓN) REGISTRO DE COMPRAS Y GASTOS DE COMPRA										
Pedido		Número de la Factura	Proveedor	Precio en Dólares	Precio en moneda Nacional	Gastos de Compra				Observaciones
Fecha	úm.					Fletes	Derechos	Gastos	Total	
Jun 8		X3847	J.P. Smith	1,000	22,500	328	580	240	23,648	
Jun 20		27483	U.S. Export	3,250	73,125	975	1600	625	76,320	

Figura 2.5.9. Registros de compras y gastos de compras.

ASIENTO DIARIO DE CONCENTRACIÓN
CONTABILIDAD AUXILIAR DE
1020 A L M A C E N

30 de junio de 1999
 FECHA

FOLIO 91

CUENTAS	DEBE			HABER		
1029 COMPRAS				7	00	0
5000 COSTOS DE VENTAS	98	5				
SUMAS						
	b)5	98	5	a)27	00	0
Movimiento de la Cuenta Colectiva						
Cargos del día 1020	a)27	00	0			
Abonos del día 1020				b)5	98	5
Movimiento Anterior	7	80	0	9	43	5
Movimiento Acumulado	4	80	0	5	42	0
HECHO POR:		REGISTRADO POR:				
_____		_____				
AUXILIAR		CONTADOR GENERAL				

Figura 2.5.10. Asiento diario de concentración del movimiento del almacén.

La información que se utilizara va a ser muy variable y depende del alcance de la misma, algunos de los formatos utilizados se muestran en las figuras 2.5.11 Informe de valores recibidos por correo, 2.5.12 Registro de envíos C.A.E. y 2.5.13 Comprobantes de caja chica.

Área de almacenes. En estas entidades es donde se hace la recepción y entrega de materiales, para realizar esto se utilizan formatos internos de la empresa.

Otro tipo de información que se ve involucrada en el proceso, son las peticiones de materiales proyectadas a ciertos periodos por parte del usuario. Es decir, en función de los proyectos asignados, el usuario puede programar las necesidades de materiales que puede necesitar para cumplir con las expectativas de los proyectos y que por falta de los mismos no se cumplan. Para lograrlo, el usuario organiza sus requerimientos y los hace llegar al área de control de inventarios para que, cuando se realicen las peticiones de pedidos al área de compras, se contemplen dichas peticiones; esta información es generada en diversos formatos que pueden ser archivo en:

- Excel
- Word
- Escritos a mano
- Etc.

El área de control de inventarios genera un formato de traspaso de materiales, como se muestra en la figura 2.5.14, este formato se manda al área de logística y a los almacenes involucrados para que realicen el movimiento. La información que contiene este formato es:

Campo	Descripción
Clave de la materia prima	Clave de la materia prima según catalogo.
Descripción	Descripción de la materia prima
Unidad de medida	La unidad de medida (kg., m, piezas, lst, etc.).
Almacén origen	Clave de almacén donde esta la materia prima.
Almacén destino	Clave de almacén a donde se paso la materia prima.
Cantidad	Cantidad de la materia prima
Consumo promedio	Consumo promedio de la materia prima que se traspasa en el almacén origen.
Existencias promedio	Existencia promedio de la materia prima que se traspasa en el almacén origen.
Excedente	Excedente de la materia prima que se traspasa en el almacén origen.

Debido a la cantidad de materiales que se manejan en la empresa se requiere de un área que permita organizar y centralizar la información de todos los materiales que en un momento dado se requiera transportar entre diferentes almacenes. Es decir, cuando en un almacén se requiera de algún material que exista en otro almacén y del cual se pueda traspasar a los almacenes que lo requieran debe de existir una logística para realizar esta tarea.

El área de control de inventarios genera los reportes de todos los materiales que en un momento dado se requiera traspasar, el área de logística utiliza esta información para la organización y control de los diferentes movimientos de materiales que se tengan que realizar entre los diferentes almacenes.

Cabe mencionar que la función que desempeña esta área es de vital importancia para que los materiales necesarios se entreguen en tiempo y orden a las diferentes entidades donde sean requeridos.

La aparición de la computadora personal en el mercado, y su rápida evolución y tecnología, así como el abatimiento del costo, ha propiciado su uso en gran escala. En consecuencia ha surgido un gran número de software de BD.

Por la facilidad de implantación del modelo relacional, el desarrollo de los administradores de BD se ha orientado a éste. El poder de la facilidad de uso son los aspectos fundamentales para los usuarios de bases de datos relacionales.

El manejador de BD es un módulo de programas que constituye la interfaz entre los datos de bajo nivel almacenados en la BD y los programas de aplicaciones y las consultas hechas al sistema. El manejador de BD es responsable de las siguientes tareas:

- Interacción con el manejador de archivo. Los datos sin procesar se almacenan en el disco mediante el sistema de archivos proporcionado normalmente por un sistema operativo convencional. El manejador de BD traduce las diferentes proposiciones en DML, a comandos de sistemas de archivos de bajo nivel. Así, el manejador de BD se encarga de realmente del almacenamiento, recuperación y actualización de los datos en la BD.
- Implantación de la integridad. Los valores de los datos almacenados en la BD deben satisfacer ciertos tipos de limitantes de consistencias. El administrador de la BD debe especificar estas limitantes en forma explícita. Si se especifican estas limitantes entonces el manejador de la BD puede verificar si las actualizaciones a esta resultan en la violación de cualquiera de estas limitantes, y si es así se podrá realizar la acción apropiada.
- Puesta en práctica de la seguridad. Es preciso que todos los usuarios de la BD tengan acceso a todo su contenido. Es labor del manejador de la BD que éstos tengan acceso a su contenido y de que se cumplan estos requisitos de seguridad.
- Respaldo y recuperación. Un sistema de cómputo, como cualquier otro tipo de dispositivo mecánico o eléctrico, está sujeto a fallas. Es responsabilidad del manejador de la BD detectar estas fallas y restaurarla al estado que existía antes de presentarse la falla.
- Control de concurrencia. Cuando varios usuarios actualizan la BD en forma concurrente, es posible que no se conserve la consistencia de los datos. Es necesario que el sistema controle la interacción entre los usuarios concurrentes; lograr dicho control es una de las tareas del manejador de la BD.

Los sistemas administradores de BD de datos relacionales (RDBMS) han sido altamente aceptados por la forma en que se manejan los datos, los sistemas relacionales ofrecen los siguientes beneficios;

- Permiten un sencillo acceso a los datos.
- Flexibilidad en el modelado de los datos
- Disminuye la redundancia de datos y el nivel de almacenaje
- Existe independencia entre el almacenaje físico y el diseño lógico de datos
- Nivel alto en la manipulación de datos.

ORACLE.

Oracle ofrece una administración eficiente, confiable y segura de datos para todo tipo de aplicaciones, desde sistemas de alto volumen de transacciones en línea hasta aplicaciones de consulta intensiva data warehouse. Oracle no sólo soporta las necesidades de la compleja administración de datos, sino también, brinda las herramientas para administrar los sistemas, la flexibilidad para distribuir efectiva y eficientemente los datos a los usuarios y la escalabilidad para alcanzar el rendimiento óptimo de todos los recursos de computación disponibles.

La arquitectura de Oracle ofrece aplicaciones OLTP con la escalabilidad para soportar un gran número de usuarios y cargas de trabajo de alto volumen de transacciones. Oracle brinda escalabilidad excepcional en máquinas SMP, clusters y MPP.

Las aplicaciones OLTP utilizan al máximo la arquitectura en paralelo de Oracle al distribuir las tareas a lo largo de múltiples procesadores o máquinas, tal como en un ambiente cluster, lo cual mejora los tiempos de respuesta de las transacciones individuales y el desempeño general del sistema. Una capacidad dinámica, automática y autoafinable, balancea equitativamente la carga de trabajo del procesamiento a través de los recursos asignados de hardware y el sistema operativo.

Para proporcionar un alto rendimiento en el procesamiento de transacciones, la arquitectura de multiconexión y multiservidor de Oracle coordina miles de requerimientos simultáneos. Los requerimientos individuales son puestos en cola y servidos por un mínimo de procesos del servidor.

Oracle7 ofrece un número de caminos diferentes de acceso para localizar datos transaccionales rápida y efectivamente, incluyendo búsquedas rápidas de tablas completas, búsquedas de una columna en árbol B, índices concatenados, tablas cluster ("pre-join"), índices "hash" cluster (con el uso de una sola columna o una función hash SQL específica de la aplicación) e identificadores únicos de fila. El optimizador en base al costo de Oracle selecciona dinámicamente el camino de acceso más rápido disponible y satisface los requerimientos de consulta directamente de los índices, siempre que sea factible.

La arquitectura de las consultas en paralelo de Oracle mejora el rendimiento de las operaciones de la base de datos, ya que, subdivide dinámicamente estas operaciones en distintas tareas y así, distribuye la carga de trabajo a través de múltiples procesadores. Oracle7 paraleliza más operaciones que ningún otro producto de base de datos.

La característica más importante de la arquitectura en paralelo de Oracle es la capacidad de ejecución en todo tipo de sistemas de hardware en paralelo. La consecución de las consultas en paralelo de Oracle es adaptable, o sea, elige inteligentemente el método de procesamiento en paralelo más apropiado para cada ambiente de hardware.

Oracle incluye el servidor Oracle WebServer™, el cual puede utilizarse para integrar totalmente sus aplicaciones de negocios de Oracle7 con la tecnología Web e implementarlas de forma segura dentro o fuera de los límites de su corporación.

FOXPRO.

La interfaz de FoxPro incluye menús presentado junto con una ventana de comandos conveniente, para utilizar los menús o escribir los comandos sin que ninguno de los elementos de la interfaz interfieran con otro. Su soporte al ratón es el mejor de todos los paquetes basados en iconos. Asimismo, la ventana de comando mantiene una historia corrida de las instrucciones (al igual que Dbase III plus), esto facilita la repetición de los comandos utilizados a través de una sesión. También se puede seleccionar parte de la historia de los comandos, y anexarla a sus aplicaciones.

La ventana presenta gráficamente todas las áreas de trabajo disponible; se puede seleccionar un área de trabajo disponible y abrir en ella una BD, accionando un botón de comando. FoxPro ofrece una caja de diálogo con una lista de los campos principales.

El constructor de pantalla comienza en blanco, en el cual se puede capturar el texto y colocar los campos, también se pueden crear botones de comando, casillas para marcar con "palomas", botones de radio y lista de extraer. Se pueden agregar pedacitos de código a cualquier código u objeto, incluso a los campos.

Desde la pantalla de establecimiento, se puede ejecutar código antes y después de la pantalla de captura, cuando se genere el código para la pantalla diseñada, se pueden anexar otras pantallas a ella, lo cual ahorra tiempo una vez formulada una biblioteca de pantallas genéricas.

El funcionamiento de FoxPro es magnífico gracias a su tecnología exclusiva de Rushmore, siendo su único inconveniente su voluminosa documentación, ya que es un problema localizar la información rápidamente, lo cual no obstante disminuye gracias a su fuerte función de ayuda en línea.

INFORMIX-SQL.

Informix SQL es cien por ciento una BD con lenguaje de consulta estructurado(SQL), el aspecto del programa es muy austero: no se encontraría con pantallas de colores múltiples con menús y ventanas a la vista. Las pantallas del paquete, estilo Lotus impulsadas por menús, automatizan las operaciones mas significativas de la BD, tales como la creación de tablas, la definición de los campos, etc. Informix también incluye un generador de reportes, y un sistema para la ejecución de archivos de la definición de formas diseñadas.

En Informix se construye una forma, no moviendo un cursor sobre la pantalla con un ratón o teclas de fecha, sino escribiendo en la forma, un tipo de definición de pantalla acompañada con instrucciones ejecutables.

Dicho archivo comprende cinco partes: una sección de tablas serán accesadas por la que describe cada campo exhibido por la forma y una sección opcional de instrucciones que define las operaciones que habrán de ser llevadas a cabo sobre los campos dentro de la forma.

Las secciones y tablas informan al sistema que debe presentar; la sección de pantalla indica en donde mostrarlo, la sección de atribuciones indica como presentarlo y la sección de instrucciones le dice al sistema que hace antes, mientras y después de presentarlo.

Se incluye con Informix-SQL varios paquetes de servicios: BECHECK verifica la integridad de los índices, si encuentra una discrepancia entre un archivo de datos y uno de sus índices, le permite reformar él índice: DRLINK y DBLOAD son de utilidad para el traslado de los datos entre Informix y el mundo exterior de los archivos de Lotus 1-2-3, Dbase, o ASCII: con DBSCHEMA, se pueden elaborar las instrucciones de SQL requeridas para crear una tabla o una BD.

PARADOX.

Una de las características mas interesantes de PARADOX es su velocidad, destacándose su rápido tiempo de respuesta para la lectura y edición de las tablas. Paradox soporta los formatos de importación y exportación mas importantes, pero existen problemas en la importación almacenada en el formato ASCII. Con Paradox, es necesario planear con cuidado antes de precipitarse a la construcción de una forma, aplicando esto principalmente a las formas que acceden a múltiples tablas. El diseño de formas con tablas múltiples exige la creación de una forma maestra, después existe también la introducción, en dicha forma maestra de las formas incluidas en las otras tablas. Así que para crear la forma maestra completa, deben diseñarse las formas que se incluirán primero.

El lenguaje para el manejo de la BD PARADOX es PAL, aunque es erróneo representar a PAL como simplemente un DML. Lo que dificulta la programación con PAL es que maneja la introducción de comandos como si el programador fuera un robot sentado en un teclado operando PARADOX; sin embargo, si simplemente no se quiere tener nada que ver con PAL, podrá instalarse el programador personal. Este programa es esencialmente un constructor de aplicaciones que através de la creación de una aplicación completa con menús y formas ofreciendo el código PAL como su producto final.

SQLSERVER.

SQL Server ha llegado a ser uno de los más populares administradores de bases de datos relacional lanzado en el medio informático, debido a todas las posibilidades que integra. Es el producto mas escalable en su clase, implanta un impresionante grupo de características nuevas y mejoras a la facilidad de uso, incluyendo varios Wizards nuevos. Los Wizard son módulos orientados a ciertas tareas, que hacen una serie de preguntas, luego generan los objetos apropiados, según las respuestas.

Cuenta con asignación dinámica de recursos la cual es una característica muy útil ya que la asignación dinámica de recursos permite la escalabilidad del uso del disco y memoria para

acomodarse a las necesidades de la base de datos en cada momento. Esta flexibilidad permite un mejor rendimiento y simplifica la administración del software. La eliminación de dispositivos también es una ventaja añadida.

Cuenta con Analizador Gráfico de Consultas. El programa ISQL/w permite construir y ejecutar las sentencias interactivas de SQL. No solo se puede construir procedimientos guardados y ejecutar unas consultas interactivas, sino que también se puede enseñar gráficamente los pasos que el procesador de consultas usa para ejecutar la consulta.

Los Servicios de Transformación de Datos (DTS). Es la nueva característica DTS de SQL Server. Aunque Microsoft la ha diseñado pensando en facilitar el almacenamiento de datos, la utilidad del producto no acaba allí. DTS simplifica la importación y la exportación de datos entre dos bases de datos compatibles con OLE DB. DTS también genera scripts Visual Basic (VBScript) que se pueden ejecutar desde el WSH (Windows Scripting Host) u otros entornos COM (Component Object Model).

Las funciones del Enterprise Manager (EM). Además de implementar el SQL Server Enterprise Manager como un snap-in del MMC (Microsoft Management Console), se mejoraron sus funciones, una de sus características es la posibilidad de mirar los contenidos de una tabla directamente desde el EM. Otra función muy útil es la posibilidad de cambiar directamente los tipos de datos de las tablas existentes.

Diseñado desde su inicio para trabajar en entornos Internet e Intranet, SQL Server es capaz de integrar los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones "tradicionales". Cada aplicación que sea desarrollada para ser empleada en entornos de red local que puede ser utilizada de forma transparente -en parte o en su totalidad- desde entornos Internet, Intranet o Extranet.

Cuenta con una plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server y con las herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.

Es el único gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática.

Tiene una ventana gráfica de Relationship que muestra todas las tablas en la BD con líneas entre ellas para representar las relaciones activas. Simplemente se arrastra de un campo en una tabla a otro en otra tabla para definir una relación y luego se personaliza el enlace.

Puede usar datos de varias bases de datos como: Dbase III y IV, FoxPro 2.0 y 2.5, Paradox 3.x y 4.x así como Microsoft, Oracle, Sybase y el lenguaje SQL.

SQL Server es una aplicación que se desarrollo bajo Windows, de esta forma facilita construir tareas rápidas por medio de ventanas y sus elementos.

Características de SQL Server.

- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server. Posee los mejores registros de los benchmarks independientes (TCP) tanto en transacciones totales como en coste por transacción.
- Gestión: Con una completa interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.
- Plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server y con las mejores herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.
- Diseñada para INTERNET: Es el único gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de manera automática.

- Potente y Escalable: Microsoft SQL Server es la única base de datos cuyo rendimiento sobre Internet está publicado, ofreciendo registros espectaculares.
- Mínimo costo de Propiedad: La sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor costo.
- Integración con herramientas de Usuario: Microsoft EXCEL, Microsoft ACCESS y muchas otras herramientas que utilizan de forma mayoritaria los usuarios finales para su trabajo diario. La confidencialidad de los datos está garantizada ya que la seguridad se integra con la de Windows NT.
- Arquitectura de servidor simétrico y paralelo con balanceo automático de carga en múltiples procesadores.
- Kernel multithread real para mejor rendimiento transaccional y escalabilidad.
- Soporte a grandes bases de datos (VLDB) (+1 TB).
- Completo proceso transaccional interactivo con rollback automático y recuperación de roll-forward.
- Optimizador de consultas mejorado basado en coste.
- Checkpointing mejorado para un mejor throughput de datos y tiempo de respuesta.
- Soporte E/S asíncrono para acceso en paralelo a múltiples dispositivos de disco para un mejor desempeño, capacidad de canal.
- Bloqueo a nivel fila y página con escalación de bloqueos; resolución automática de deadlocks.
- Llamadas a procedimientos remotos servidor-a-servidor (procedimientos almacenados remotos).
- Replicación asíncrona o continua basada en registros, o sincronización planificada de tablas point-in-time.
- Configuración de replicación gráfica y características de gestión.
- Replicación de subcriptores ODBC, incluyendo IBM DB2, ORACLE, SYBASE y Microsoft Access.
- Replicación de tipos de datos Texto e Imagen.
- Backup/restore en paralelo de alta velocidad que soporta hasta 32 discos o dispositivos de cinta.
- Copias de seguridad en cinta y recuperación de tablas individuales.
- Reconstrucción de índice sin realizar dropping y recreación del índice.

Elección de un Administrador de Base de Datos para PC's.

Uno de los aspectos importantes de los administradores de bases de datos mas recientes es que logran que entre la característica de poder que ofrecen la mayoría de los fabricantes, se cuenta con la posibilidad integrada de enlace en redes, la posibilidad de colocar un programa nuevo de software en la red, para ofrecer acceso compartido a datos y lenguajes de programación que se puedan utilizar para desarrollar aplicaciones adecuadas a las necesidades de cada uno de los usuarios, incluyendo menús y reportes y que ese poder sea mas accesible con una serie de menús, editores de toda la pantalla y características automáticas que vuelvan relativamente sencilla la realización de operaciones relacionales. Productos pioneros como Dbase por ejemplo, era predominantemente controlados por líneas de comandos. Para aprovechar este software tenia que aprender un gran numero de comandos y procedimientos. Esto ya no sucede con los productos nuevos, tal es el caso de SQL Server. La demanda del usuario de acceso a BD ha apresurado a los productores de software a diseñar interfaces realistas. Recientes adelantos técnicos (implantación de gráficos en Bases de Datos), como se menciona anteriormente con SQL Server se puede trabajar con la implementación de los avances OLE (Object Link Embedd) en versión 2.0 incorpora datos de Word , Excel, Power Point y Microsoft mail. SQL Server es una aplicación que se desarrollo bajo Windows, de esta forma facilita construir tareas rápidas por medio de ventanas y sus elementos.

Para poder elegir la BD a nuestras necesidades, fue necesario realizar un criterio de selección el cual se muestra en la Tablas 2.6.1 –2.6.12

PRODUCTO	COMPANÍA
Oracle	Oracle Corp.
FoxPro	Microsoft Corp.
Informix - SQL	Informix
Paradox	Corel Corp.
SQL Server	Microsoft Corp.

Para fines prácticos, se abreviaron con las siguientes letras a los productos:

Oracle = O
 FoxPro = F
 Informix-SQL = I
 Paradox = P
 SQL Server = S

Tiempo de Procesamiento de Batch	O	F	I	P	S
Bueno	X				X
Regular		X		X	
Malo			X		

Tabla 2.6.1. Evaluación de tiempo de procesamiento en batch.

Tiempo Importación y Generación de Índices	O	F	I	P	S
Bueno	X	X		X	X
Regular					
Malo			X		

Tabla 2.6.2. Evaluación de tiempo de importación y generación de índices.

Soporte de No. Máximo de Archivos Abiertos	O	F	I	P	S
Bueno	X		X		
Regular		X		X	X
Malo					

Tabla 2.6.3. Evaluación de soporte de No. Máximo de archivos abiertos.

Facilidad de Uso	O	F	I	P	S
Bueno		X		X	X
Regular	X		X		
Malo					

Tabla 2.6.4. Evaluación de facilidad de uso.

Proporción Seguridad	O	F	I	P	S
Bueno	X	X	X	X	X
Regular					
Malo					

Tabla 2.6.5. Evaluación que proporciona seguridad.

Integridad en Los Datos	O	F	I	P	S
Bueno	X			X	X
Regular		X	X		
Malo					

Tabla 2.6.6. Evaluación de la integridad de datos.

Presentación de Calidad En los Reportes y Formas	O	F	I	P	S
Bueno	X				X
Regular		X	X	X	
Malo					

Tabla 2.6.7. Evaluación de la presentación de calidad en los reportes y formas.

Facilidad de Programación	O	F	I	P	S
Bueno					X
Regular		X		X	
Malo	X		X		

Tabla 2.6.8. Evaluación de la presentación de calidad en los reportes y formas.

Compartición de los datos Con otras Bases de Datos	O	F	I	P	S
Bueno	X				X
Regular		X	X	X	
Malo					

Tabla 2.6.9. Evaluación en la compartición de los datos con otras Bases de Datos.

Diversidad en Tipos de Datos	O	F	I	P	S
Bueno	X				X
Regular		X		X	
Malo			X		

Tabla 2.6.10. Evaluación en la diversidad en tipos de datos.

Manejo de Gráficos para Formularios	O	F	I	P	S
Bueno					
Regular	X				X
Malo		X	X	X	

Tabla 2.6.11. Evaluación en el manejo de gráficos para formularios.

Para establecer la puntuación que obtuvo cada producto se determinó un valor para cada categoría determinada de la siguiente manera:

Bueno = 2 puntos
 Regular = 1 punto
 Malo = 0 puntos

Obteniendo finalmente la siguiente tabla de puntuaciones: tabla 2.6.12

Características	O	F	I	P	S
Tiempo de procesamiento en Batch	2	1	0	1	2
Tiempo de importación y generación de índices	2	2	0	2	2
Soporte de No. De Máximo de archivos abiertos	2	1	2	1	1
Facilidad de Uso	0	2	1	2	2
Proporciona Seguridad	2	2	2	2	2
Integridad de Datos	2	1	1	2	2
Presentación de Calidad en los Reportes y Formas	2	1	1	1	2
Facilidad de Programación	2	1	0	1	2
Compartición de Datos con otras Bases de Datos	2	1	1	1	2
Diversidad de tipos de datos	2	1	0	1	2
Manejo de gráficos para Formularios	1	0	0	0	1
Total	19	13	8	14	20

Tabla 2.6.12. Evaluación para la puntuación final.

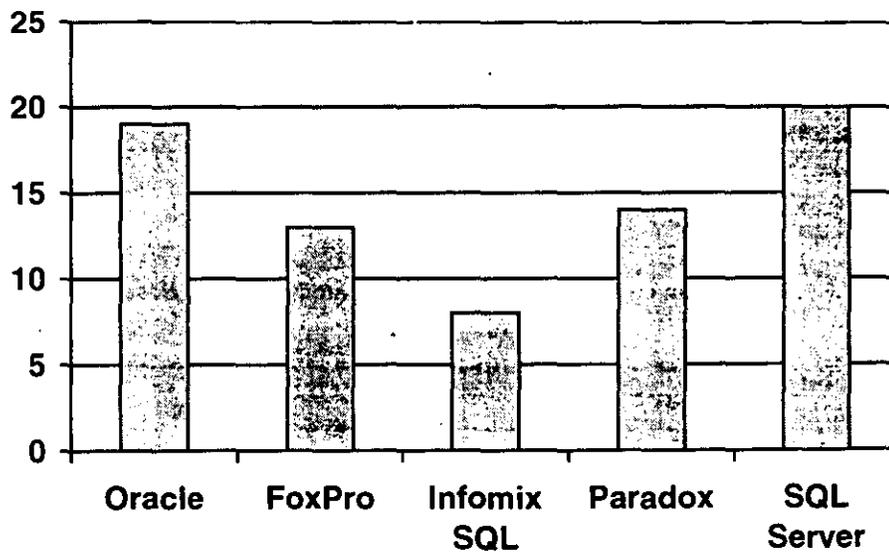


Tabla 2.6.13. Puntuación de la comparación.

La elección de la BD se basó, además de todo lo anterior, en la disponibilidad comercial. Por lo antes mencionado y en relación con las capacidades de manejo, programación y de conocimiento, entre otras, asociadas, se determinó el utilizar SQL-Server para el desarrollo del sistema.

2.7. OPCIONES DE SOLUCIÓN Y SELECCIÓN DEL FRONT-END.

El Front-end de un sistema es un programa de computadora que oculta detalles del acceso a datos. En un sentido, todo programa es un front-end que evita que los usuarios conozcan los detalles reales del intrincado funcionamiento de la computadora.

En otras palabras, un front-end es un procesador de formatos, es decir, un procesador de entrada/salida utilizado para realizar el formato de datos de entrada y/o procesarlos. A veces se utiliza el termino para referirse a un programa de comunicación

Programación Visual.

El éxito de Windows en el entorno de PC ha llevado a los programadores a desarrollar aplicaciones que respeten el "look" y las necesidades de interfaz gráfica. La llamada "programación visual" supone un paso más en la búsqueda de nuevas herramientas que faciliten el desarrollo de software para Windows.

Empresas como Microsoft y Borland se han apresurado a lanzar al mercado versiones de sus productos con la palabra "Visual". Se puede decir que "Programación Visual" se entiende como un procesador de desarrollo interactivo de la interfaz de usuario para sus aplicaciones. La idea es que la "Programación Visual" permite que la mayor parte del proceso necesario para crear cualquier aplicación se pueda utilizar en forma "automática".

Así por ejemplo, como la mayoría de las aplicaciones de Windows disponen de menús desplegables, parece razonable desarrollar una herramienta para crearlos sin tener que recurrir a paginas de código indescifrable. Pero los beneficios no terminan aquí, debido a que lo mencionado anteriormente ya se encontraba disponible en los editores de recurso, sin embargo actualmente se cuenta con mayor integración entre el desarrollo de aplicaciones y el diseño de la interfaz de usuario.

Los editores de recursos que pueden encontrarse en el mercado (en los compiladores de C para Windows siempre se incluye alguno) siguen las siguientes pautas:

Primero, se crea el recurso deseado que puede ser una ventana, un menú, un icono, algún tipo de letra, un botón, una caja de texto o cualquier otro elemento típico de Windows, después se genera el código correspondiente al recurso y, por último con mas o menos pasos intermedios, se acaba uniéndolo al resto del código fuente de un programa como se puede apreciar en la figura 2.7.1

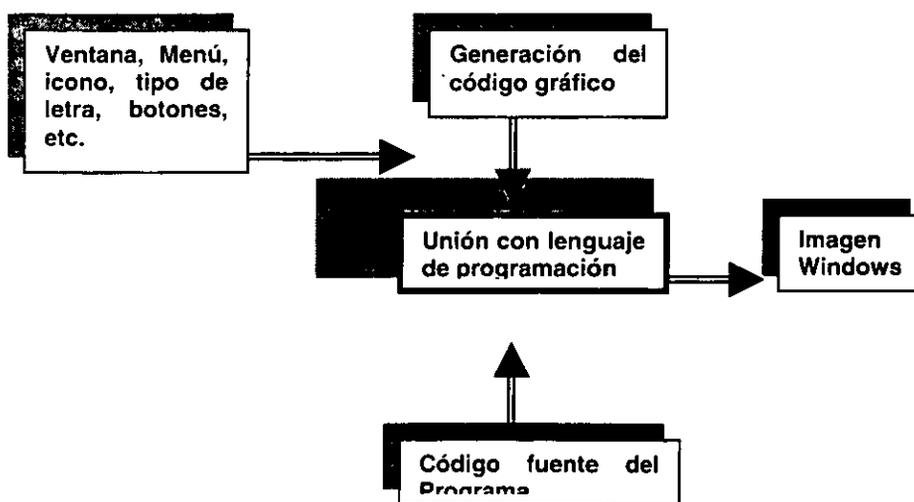


Figura 2.7.1. Proceso para realizar una aplicación con ventanas.

Con la programación utilizando herramientas visuales, se pretende simplificar al máximo todo el proceso, evitando cuantos pasos intermedios sean posibles. Lo más común es que se parta de la creación de una interfaz de usuario y se llegue a la escritura del código (de ser necesario) en algún lenguaje de presentación.

Por el momento, con la llegada de herramientas visuales para el desarrollo de aplicaciones, se ha dado un enorme paso para facilitar la vida de los desaprobadores; hasta entonces, tendríamos que seguir trabajando con las herramientas disponibles, muchas de las cuales ya han comenzado su transición a la "Programación Visual".

Desde que la informática abandonó el uso de tarjetas perforadas, la mayor preocupación de los programadores ha sido crear aplicaciones que "entren por los ojos"; es decir que no solo fueran fáciles de usar, sino también atractivos. En pocos años hemos vivido una verdadera revolución en la manera de entender las aplicaciones, esta revolución ha venido acompañada de nuevas y poderosas PC's, así como de sistemas operativos totalmente gráficos (Windows NT) y de periféricos tales como módem, impresora láser, escáner, etc.

Además, la proliferación de las interfaces gráficas de usuario -los llamados ambientes de ventanas o GUI- no ha hecho más que acentuar lo que existía entre las técnicas de programación de los años ochenta y las actuales. Muchos han sido los desarrolladores que no han sabido adaptarse a los nuevos tiempos y se han quedado obsoletos, sin embargo, aquellos que han actualizado sus conocimientos reconocen que ahora cuenta con herramientas mejores y más poderosas para realizar su trabajo.

Al mismo tiempo, las herramientas de programación han evolucionado hasta convertirse en verdaderos ambientes de desarrollo integrados que se basan en la misma filosofía de sencillez que gobiernan las interfaces gráficas. En el campo de la integración de herramientas, Borland fue pionera con su Turbo Pascal, cuyas primeras versiones aparecieron para el sistema operativo CPM de la familia de microprocesadores Z80 de Zilog. Por otra parte, Microsoft también ha sido pionera con el concepto de "Programación Visual" a través de su herramienta Visual Basic desde su versión para DOS y que actualmente con sus versiones para Windows se ha convertido en best seller.

Definición.

Anteriormente se mencionó que el término en sí era demasiado genérico y que procedía principalmente del nombre que las propias casas de software daban a sus IDE (Integrated Development Environment. Ambientes de Desarrollo Integrado) para lanzarlos publicitariamente con la idea de ser algo intuitivo y fácil de usar.

Sin embargo, con la aparición de los últimos productos de aplicaciones bajo ambientes gráficos se va haciendo necesario una terminología adecuada que los clasifique de alguna manera.

Un ambiente integrado de desarrollo es aquel que permite desde un único programa realizar las tareas más habituales de desarrollo y depuración de una aplicación (incluido en algunos casos la optimización y realización automática de pruebas).

Si el ambiente integrado es de "Programación Visual", se habla de un producto que integra una serie de herramientas adicionales a las habituales y que son necesarias para el desarrollo de programas bajo una interfaz de usuario gráfica, todo ello con la mínima intervención del programador en las tareas más repetitivas y por tanto, más automatizadas.

Se puede decir que la "Programación Visual" se sitúa directamente por debajo de las llamadas herramientas CASE (Computer Aided Software Engineering). Los programas CASE permiten directamente el diseño de una aplicación (diagramas de flujo, estructuras de datos) a producir la aplicación propiamente dicha.

Programar la Interfaz.

Con esta premisa, está claro que "Programación Visual" a pasado de ser una curiosa novedad, a constituirse como un conjunto de herramientas imprescindibles para el desarrollo de las complejas aplicaciones basadas en GUI's que se comercializan actualmente.

La complejidad de programar bajo una interfaz gráfica proviene del propio diseño de los ambientes. De hecho, desarrollar el programa mas sencillo para un ambiente gráfico puede convertirse en una tarea desquiciante si no se dispone de las herramientas adecuadas.

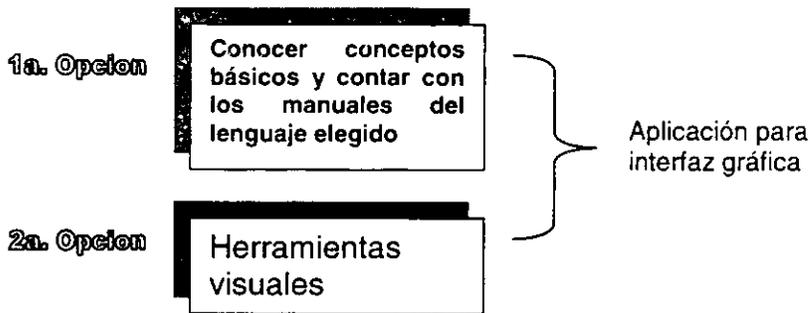


Figura 2.7.2. Desarrollo de una aplicación para una interfaz gráfica.

Para afrontar el desarrollo de una aplicación bajo una interfaz gráfica (Windows u OS/2 en el mundo PC) pueden seguirse dos caminos. El primero es enfrentarse con los conceptos básicos de ambiente, lo cual resulta una tarea tediosa y poco gratificante ya que a veces no es efectiva ante las funcionalidades ya existentes (ole; Object Linking Embedding) o los nuevos sistemas operativos (Windows 98). El segundo camino se simplifica al utilizar las mencionadas herramientas visuales de desarrollo.

Estos compiladores simplifican el diseño al proporcionarnos formatos, patrones tipo que pueden aplicarse a los desarrollos propios, al mismo tiempo que ofrece una elevada confiabilidad debido a que se tratan de patrones probados en muchos otros desarrollos. Todo ello redundando en una mayor eficiencia y un gran ahorro de tiempo de codificación.

VISUAL BASIC.

Tomando como referencia el punto Características, ventajas y desventajas de Visual Basic.

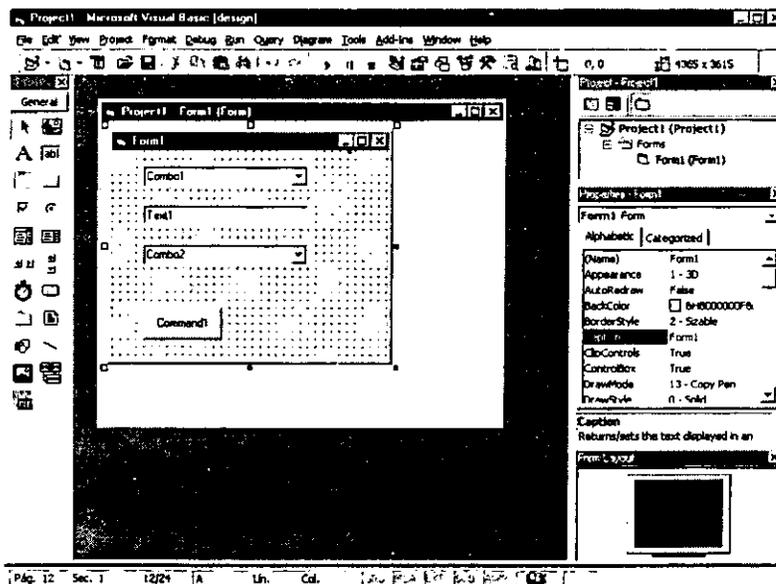


Figura 2.7.3. Ventana de Diseño de Visual Basic.

Ventajas.

- Es un sistema productivo para crear soluciones bajo ambiente Windows.
- Posibilidad de incorporar controles preconstruidos por terceros (Custom Control).
- Acceso a bases de datos (Access, SQL Server, Oracle, FoxPro, etc.)
- Manejo de Multimedia
- Permite ensamblar fácil y rápidamente una interfaz de usuario con componentes prefabricados.
- Ofrece una gran capacidad y velocidad en su Debugger.
- Permite manipular otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Word, Excel, Project, etc.) siempre y cuando dichas aplicaciones soporten OLE Automation.
- Interfaz amigable al usuario
- Manejo de ayudas en líneas robustas
- Interprete que permite la detección de errores al momento de editar el programa
- Maneja aplicaciones Cliente-Servidor
- Incluye motor de BD vía DAO (Data Access Object)
- Un usuario inexperto puede ser productivo con algunas semanas de utilización de Visual Basic.
- Puede ser utilizado como un servidor de OLE Automation local o remoto.
- Módulos de clases. Objetos y colecciones reutilizables
- Procedimientos para propiedades
- Compilación de aplicaciones en formato DLL para la reutilización de componentes.
- Aunque Visual Basic no cuenta con herencia, esto hace que los componentes reutilizados sean más fáciles de comercializar.
- Creación de AddIns. Son servidores OLE que pueden integrarse al ambiente de desarrollo de Visual Basic para facilitar al programador la elaboración de aplicaciones.

Desventajas.

- No cuenta con herencia, característica muy importante de la programación orientada a objetos.
- Por ser código interpretado, los ejecutables producidos por Visual Basic resultan lentos en su desempeño
- Su código no es compatible con plataformas no Windows
- La utilización de componentes remotos es muy lenta

DELPHI.

Delphi es una herramienta para construir aplicaciones que utiliza el lenguaje de programación Object Pascal de Borland -sin lugar a dudas se le pudo haber llamado Visual Pascal.

Realizar una aplicación en Delphi es como crear una aplicación en Visual Basic, uno crea las formas y selecciona componentes de una caja de herramientas, para ponerlas dentro de las formas y se procede a programar los eventos necesarios para manipular las entradas del usuario.

Por otra parte; Delphi es una aplicación basada en componentes bajo un ambiente de desarrollo que da soporte a una implementación rápida y eficiente de programas basados en Windows con un mínimo de código

Muchos de los requerimientos tradicionales de la programación para Windows son manejados dentro de una librería de clases de Delphi, eliminando los problemas de tareas repetitivas de programación.

Delphi proporciona herramientas de diseño tales como formas para aplicaciones, de tal manera que se pueden crear y probar los prototipos de la aplicación.

Mediante el uso de una gran cantidad de componentes y la generación intuitiva de código, las aplicaciones se pueden modificar generando programas completos y propios del usuario.

Las herramientas de bases de datos de Delphi permiten el desarrollo de aplicaciones Cliente-Servidor. Se puede ver la información en tiempo real de tal manera que se conocen los resultados de las consultas inmediatamente.

Ventajas.

- Presenta programación estructurada y programación orientada a objetos
- Requiere de una sintaxis precisa
- Su compilador realiza su programa ejecutable automáticamente
- Tiene edición multiarchivos
- Su curva de aprendizaje es de nivel medio
- Provee encapsulamiento para proyectos grandes o en grupo, lo cual significa que se tiene control de acceso a variables de una manera más sencilla que el esquema procedimiento - modulo- global utilizado en Visual Basic
- Presenta un modulo separado de encapsulamiento donde las declaraciones en un archivo pueden o no formar parte de la interfaz

Desventajas.

- Es una herramienta poco difundida
- Su limite de cadenas es de 2048 caracteres, sin embargo existen cadenas terminación cero que pueden ser de 65,535 caracteres de extensión, pero las operaciones que se pueden realizar con ellas no son tan poderosas como las de Visual Basic.
- El limite de 2048 caracteres también se aplica a algunas propiedades de control.
- No existen arreglos re-dimensionables. Sin embargo, Delphi tiene colecciones que pueden cambiarse de tamaño, las cuales son más poderosas que los arreglos

MICROSOFT VISUAL C++.

Aunque el lenguaje favorito de Microsoft es el Visual Basic para aplicaciones, son muchos los programadores que por muchos motivos prefieren el C o Pascal para sus aplicaciones. Para estos usuarios, existe una herramienta muy poderosa: Microsoft Visual C++.

Hasta hace unos años, Microsoft no había adoptado la filosofía de los ambientes integrados de programación en sus productos. Sus compiladores se invocaban desde la línea de comandos, teniendo que incluir multitud de parámetros para conseguir los resultados deseados. Sin embargo, su entrada a este campo ha sido espectacular. El nacimiento de Windows supuso un paso decisivo en general hacia el desarrollo de herramientas integradas e intuitivas, pero el mas beneficiado en este aspecto ha sido Microsot. Casi nadie esperaba que alcanzaría a Borland -pionero en ambientes bajo DOS- en cuanto a potencia y facilidad de uso en sus ...ambientes integrados de programación.

Visual C++ supone la competencia que Microsoft opone a Borland en este lenguaje que cuenta con tantos adeptos entre los programadores. Se supone que el inventor de Windows debe disponer de los mejores lenguajes de programación para las aplicaciones de este ambiente.

Una gran diferencia entre los programas elaborados en C a los programas elaborados bajo Windows C++, es que los programas en C llaman al sistema operativo para utilizar una salida, en cambio Windows utiliza las salidas a través de mensajes. Los programas elaborados en C son escritos directamente a la memoria de vídeo y al puerto de impresión, la desventaja de esta técnica es que se necesita un driver o manejador de software para los periféricos.

Windows y C++ introdujeron un nivel de abstracción llamado la interfaz de dispositivos gráficos (GDI), para que así el progama no necesite conocer el tipo de tarjeta de vídeo y el tipo de dispositivo de impresión que se tiene. En lugar de localizar el hardware, el programa llama a

la función GDI la cual hace referencia a una estructura de datos llamada dispositivo de contexto (DC), Windows convierte el DC a un dispositivo físico y establece las instrucciones apropiadas de entrada/salida. EL GDI es tan rápido como el acceso directo a vídeo y permite el uso a diferentes aplicaciones mientras se escribe.

Windows junto con C++ ofrece un manejador de memoria principal, resultando así que la memoria ya no es un problema. Simplemente se coloca la memoria que se necesita y Windows se encarga de lo demás, ejecuta el programa, administra los recursos, automáticamente lo cambia a disco y después lo coloca en memoria física, y los cambios son tan buenos que la computadora tendrá mucha memoria disponible.

Ventajas.

- Incluye librerías preconstruidas
- Controles visuales
- Reutilización de código total
- Manejo completo del esquema de la programación orientada a objetos (herencia, polimorfismo, etc.)
- Todo el manejo de dispositivos de entrada/salida se hacen a través de librerías.
- Cuenta con un compilador, no con un intérprete.
- A través de él se pueden crear clientes de OLE.
- Es el único lenguaje que al salir una nueva versión de Windows, ya cuenta con las herramientas necesarias para trabajar con él.
- Manejo de multitarea.
- Rapidez en ejecución de programas.
- Permite obtener funcionalidades de otras aplicaciones para utilizarlas como componentes en aplicaciones propias, por ejemplo Word, Excel, Project, Visual Basic 4.0, etc.
- Manejo de OLE automation. Permite la creación de servidores OLE.
- Total libertad para manejo de memoria y accesos al sistema operativo
- Utilización de librería MFC (Microsoft Foundation Classes) para el desarrollo de aplicaciones bajo plataforma Windows.

Desventajas.

- Curva de aprendizaje alta
- Tiempos de compilación largos
- Si se utiliza MFC se pierde la compatibilidad de código entre plataformas

VISUAL JAVA ++.

Microsoft Visual J++ 6.0 Edición profesional permite aprovechar la productividad del lenguaje Java y la eficacia de Windows para generar y distribuir soluciones cliente-servidor de alto rendimiento para Windows y el Web. Diseñar, depurar y distribuir visualmente aplicaciones de importantes características en el entorno de aplicación rápida de alto rendimiento (RAD).

Beneficios.

El entorno RAD de alta productividad ofrece un diseño visual bidireccional de aplicaciones Windows y Web de alto rendimiento y especiales características. La utilización Windows Foundation Classes (WFC) crea rápidamente aplicaciones, componentes y objetos de negocios de nivel intermedio. Genera eficaces aplicaciones de bases de datos con acceso a una gran variedad de datos y utilizando Visual Database Tools se crean, actualizan y visualizan rápidamente tablas de bases de datos y diseño gráfico de consultas SQL. Simplificación de la configuración y distribución de software con herramientas de un solo botón

para empaquetar, firmar y distribuir la entrega de aplicaciones completas a un archivo remoto en servidores Web.

Características Estándar.

Diseñador visual de formularios El diseñador visual ayuda a generar aplicaciones y componentes basados en Windows arrastrando y colocando.

Diseñador visual de HTML Simplifica el desarrollo de aplicaciones Web del cliente con las herramientas de creación y automatización integradas WYSIWYG. Desarrollo MTS en un solo paso. Con presionar un botón convierte cualquier clase Java en un componente de Microsoft Transaction Server (MTS)/Foundation Classes (WFC).

Este marco orientado a objetos encapsula, simplifica y unifica los modelos de programación Win 32 y HTML dinámico. Utiliza más de 100 componentes generados previamente para simplificar la construcción de interfaces de usuario, objetos reutilizables y componentes del servidor.

Tecnología IntelliSense. Aumenta la velocidad del desarrollo con la ayuda de la programación sobre la marcha, incluyendo la terminación de instrucciones, información de lista de parámetros y verificación de sintaxis.

Oracle.

Oracle Developer ofrece un juego de herramientas integradas que soporta las necesidades de productividad de las tareas de desarrollo a gran escala. Oracle Developer satisface las necesidades de escalabilidad de las aplicaciones para la gestión de toda la empresa, la satisfacción de requerimientos en constante evolución y la realidad de entornos heterogéneos. En pocas palabras, Oracle Developer cumple la promesa de la informática por Internet.

Oracle Developer incorpora un juego integrado de generadores (figura 2.7.4) para la creación de pantallas, informes, gráficos, consultas, objetos de base de datos y procedimientos. Estos componentes emplean potentes funciones declarativas, que permiten crear aplicaciones directamente a partir de definiciones de base de datos, sin tener que escribir ni una sola línea de código. Oracle Developer establece nuevos estándares de productividad y facilidad de uso para el desarrollo de cliente/servidor y Web mediante la utilización de técnicas de desarrollo rápido de aplicaciones (RAD), orientación a objetos, autoformación asistida por ordenador y arquitectura unificada de cliente, servidor de aplicaciones y servidor de bases de datos. Los estándares de desarrollo de la organización se aplican fácilmente mediante las plantillas de aplicación personalizables y la configuración por defecto de objetos de aplicación. Para obtener la máxima productividad, los desarrolladores tienen la opción de generar aplicaciones Oracle Developer completas directamente a partir de modelos creados con Oracle Designer.

Escalabilidad La escalabilidad es inherente a Oracle Developer. Inherente a la arquitectura de varias capas del producto. Explícita en el soporte sin igual de funcionalidad de servidor, como DML, cursores de base de datos, variables de transferencia, puntos de interrupción y juegos de resultados. Definitiva en el particionamiento de objetos cliente/servidor mediante el "arrastrar y soltar", que reduce el tráfico de red entre cliente y servidor a un solo viaje de ida y vuelta.

Las funciones integradas de Oracle Developer permiten a los clientes pasar de la escala de 10 a la de 10.000 usuarios, de megabytes a gigabytes de datos y de aplicaciones de soporte a la decisión a OLTPs complejos. Además, la arquitectura flexible de varias capas de Oracle Developer soporta Oracle Application Server o servidores de aplicaciones de terceros, como monitores TP, garantizando así la escalabilidad desde el puesto de trabajo hasta la empresa en su conjunto.

FUNCIONES DE ORACLE DEVELOPER.

Interfaces basadas en asistentes La interfaz de Oracle Developer se compone de la combinación de navegadores de objeto, cuadros de diálogo con pestañas, paletas de propiedades y una interfaz GUI de tipo WYSIWYG. Las aplicaciones se crean rápidamente mediante asistentes reentrantes o mediante la recuperación de componentes reutilizables de la Object Library.

Oracle Developer permite que las organizaciones respondan a las presiones de los mercados globales y altamente competitivos, de hoy en día. Produce aplicaciones de base de datos, complejas, críticas y a gran escala en un espacio de tiempo menor y con menos recursos, soporta una amplia diversidad de plataformas de implantación con un solo código origen, y proporciona la escalabilidad necesaria para dar soporte a las exigencias de las soluciones de las empresas.

Selección del front-end.

En cuanto a la selección del software para la generación del front-end del sistema, para poder elegir el software a nuestras necesidades; fue necesario realizar un criterio de selección el cual se muestra en la Tablas 2.7.8 -2.7.12.

PRODUCTO	COMPANÍA
Visual C++	Microsoft Corp.
Visual Basic	Microsoft Corp.
Delphi	Inprise Corp.
Visual J	Microsoft Corp.
Oracle	Oracle Corp.

Para fines prácticos se abreviaron con las siguientes letras a los productos

Visual C++ = VC
 Visual Basic = VB
 Delhi = D
 Visual J = VJ
 Oracle = O

Facilidad de Uso	VC	VB	D	VJ	O
Bueno	X	X			
Regular			X	X	X
Malo					

Tabla 2.7.8. Evaluación Facilidad de Uso.

Facilidad de Programación	VC	VB	D	VJ	O
Bueno	X	X			
Regular			X	X	X
Malo					

Tabla 2.7.9. Evaluación Facilidad de Programación.

Tiempo de Procesamiento en Batch	VC	VB	D	VJ	O
Bueno			X		
Regular		X		X	X
Malo	X				

Tabla 2.7.10. Evaluación de Tiempo de Procesamiento en Batch.

Tiempo de Importación y generación de índices	VC	VB	D	VJ	O
Bueno					
Regular		X	X	X	X
Malo	X				

Tabla 2.7.11. Evaluación de Tiempo de Importación y generación de índices.

Número máximo de archivos abiertos	VC	VB	D	VJ	O
Bueno	X				X
Regular		X	X	X	
Malo					

Tabla 2.7.12. Evaluación de Número máximo de archivos abiertos.

Seguridad	VC	VB	D	VJ	O
Bueno	X	X	X		X
Regular				X	
Malo					

Tabla 2.7.13. Evaluación de Seguridad.

Diversidad en tipos de datos	VC	VB	D	VJ	O
Bueno					X
Regular	X	X	X	X	
Malo					

Tabla 2.7.14. Evaluación de Diversidad de tipos de datos.

Manejo de gráfico para formularios	VC	VB	D	VJ	O
Bueno	X	X	X		
Regular				X	X
Malo					

Tabla 2.7.15. Evaluación de Manejo de gráfico para formularios.

Para establecer la puntuación que obtuvo cada producto se denominó con un valor para cada categoría determinada de la siguiente manera:

Bueno = 2 puntos
 Regular = 1 punto
 Malo = 0 puntos

Obteniendo finalmente la siguiente tabla de puntuaciones 2.7.16., mostrada en la figura 2.7.17.:

Características	Visual C++	Visual Basic	Delphi	Visual J	Oracle
Facilidad de Uso	2	2	1	1	1
Facilidad de Programación	2	2	1	1	1
Tiempo de procesamiento Batch	0	1	2	1	1
Tiempo de importación y generación de índices	0	1	1	1	1
Número máximo de archivos abiertos	2	1	1	1	2
Seguridad	2	2	2	1	2
Diversidad en tipos de datos	1	1	1	1	2
Manejo de gráficos para formularios	2	2	2	1	1
TOTAL	11	12	11	8	11

Tabla 2.7.16. Tabla de puntuaciones.

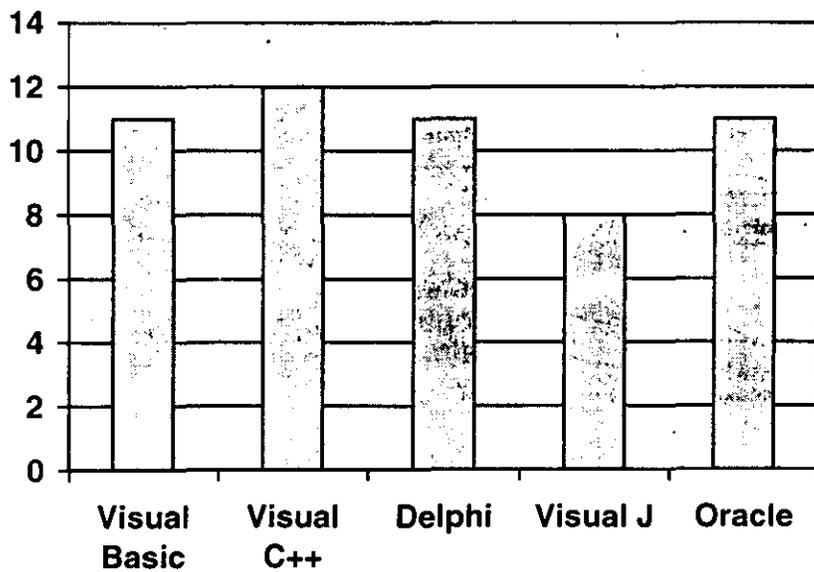


Tabla 2.7.17. Puntuación de la comparación.

Por lo tanto para la generación del front-end del sistema, se utilizara Visual Basic debido a todas las características que observamos y también aquellas que se mencionaron anteriormente.

CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA.

3.1. APLICACIÓN DE LA METODOLOGÍA ELEGIDA (BACK-END PARA CADA MÓDULO).

3.1.1. Diagramas de contexto.

Para poder realizar un diseño más rápido y sencillo necesitamos tener un panorama general de lo que se requiere hacer sin importar la manera en que se realice esto durante el análisis y posteriormente en el diseño se toman las decisiones de cómo se van a llevar a cabo las cosas. Todo en orden jerárquico de un nivel alto generalizado y posteriormente de una manera más detallada.

Al seleccionar la propuesta básica para resolver el problema, la propuesta de sistema es dividida en módulos o subsistemas. Esta manera de descomposición nos ayuda a entender de una manera más rápida el problema y tomar decisiones que afecten al sistema entero y a su vez poder trabajar de manera independiente cada módulo o subsistema.

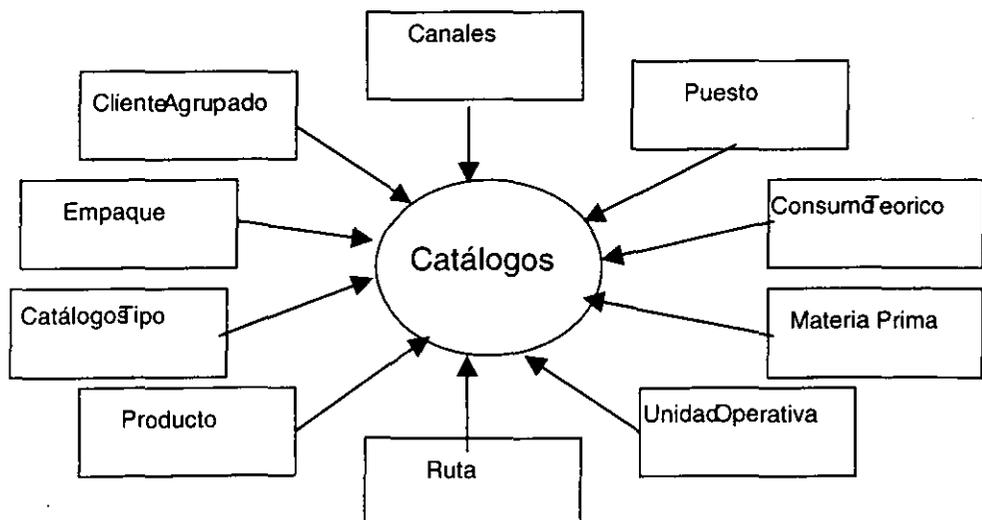
En la figura 3.1.1.1. podemos observar el diagrama de contexto del sistema de control de almacén en donde se tienen identificados los siguientes objetos:

Catálogos, Reportes, Consultas, Cargas y Acumulados, Administración y Seguridad.

En la figura 3.1.1.2. nos muestra el diagrama de contexto de Catálogos en donde se da mantenimiento a los catálogos de canales, clientes, puesto, consumo teórico, materia prima, unidad operativa, ruta, producto, catálogos tipo, empaque y cliente agrupado.

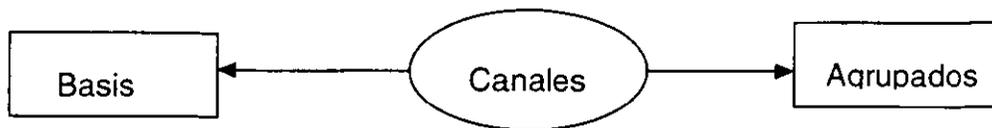


3.1.1.1. Diagrama de contexto del sistema de control de almacén



3.1.1.2. Diagrama de contexto de Catálogos

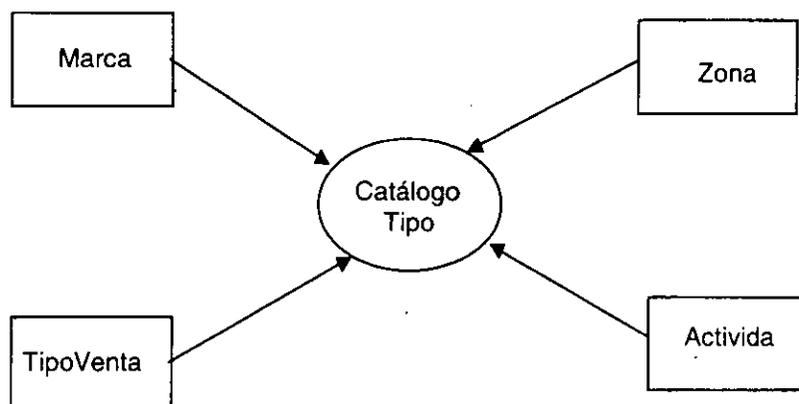
En la figura 3.1.1.3. nos muestra el diagrama de contexto de reportes en donde se generan reportes de bonificación y otros.



3.1.1.2.1. Diagrama de contexto de Canales

En la figura 3.1.1.4. nos muestra el diagrama de contexto de consultas en donde se realizan

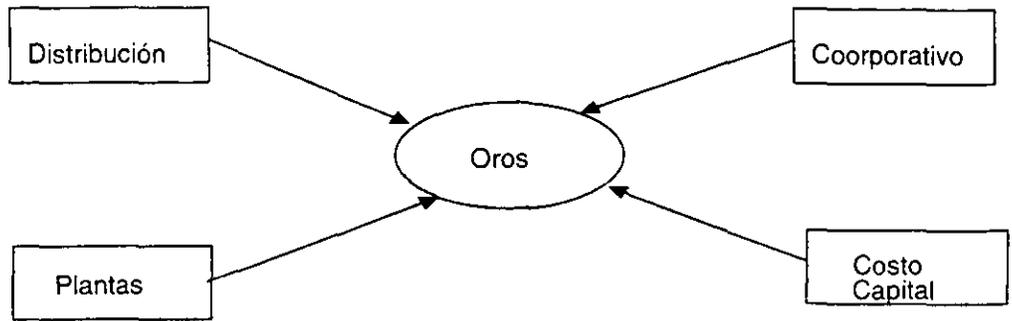
las consultas a asignación de actividades, canal por unidad operativa, ventas, cuadro de información y cajas físicas por distribuidora.



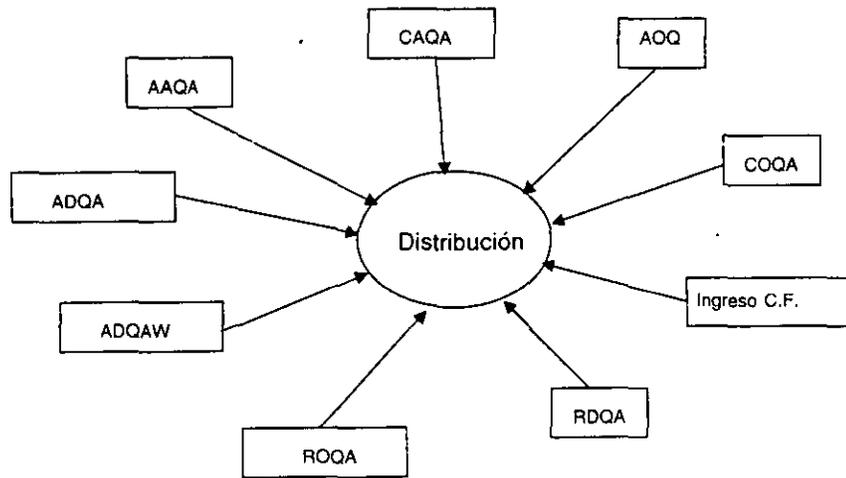
3.1.1.2.2. Diagrama de contexto de catálogo tipo



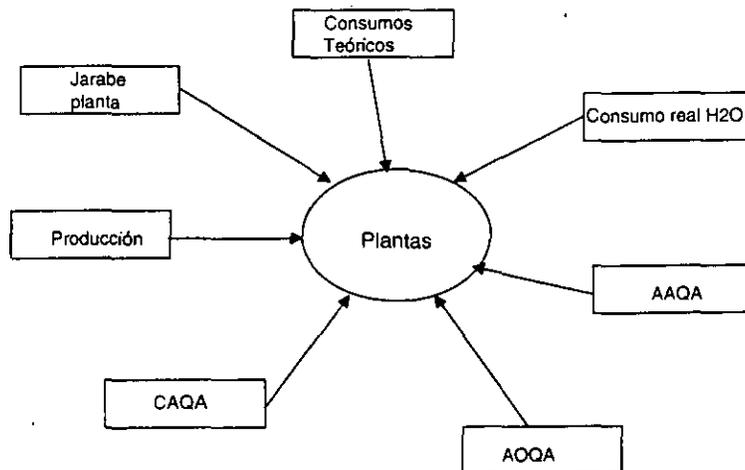
3.1.1.3. Diagrama de contexto de Reportes



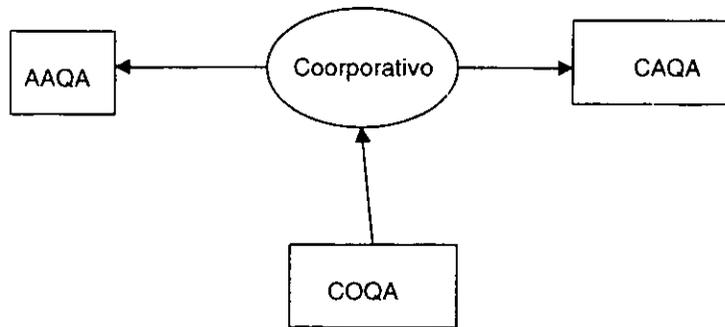
3.1.1.3.1. Diagrama de contexto de reporte



3.1.1.3.1.1 Diagrama de contexto de Reporte



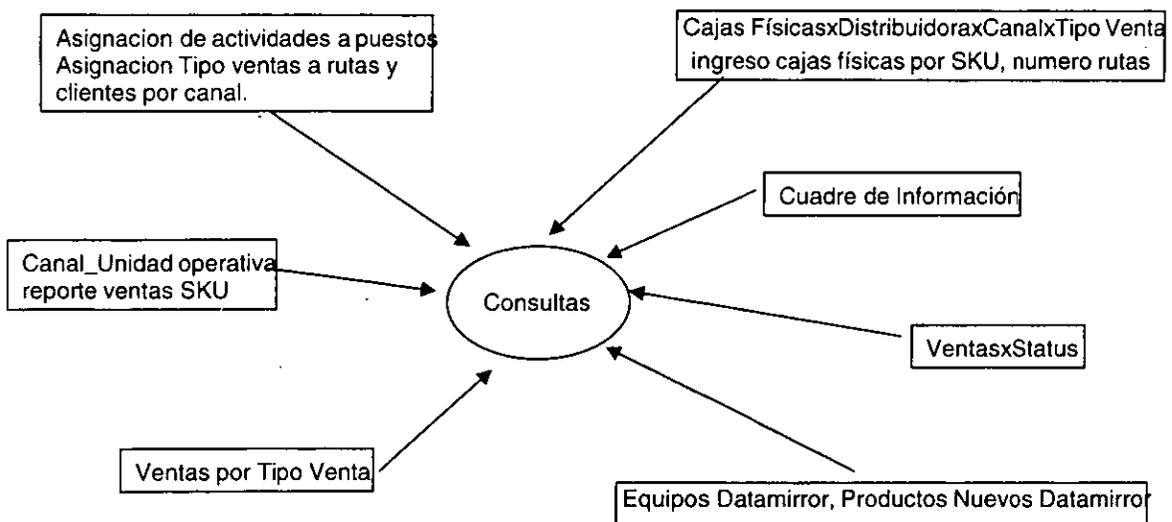
3.1.1.3.1.2. Diagrama de contexto de reporte de plantas



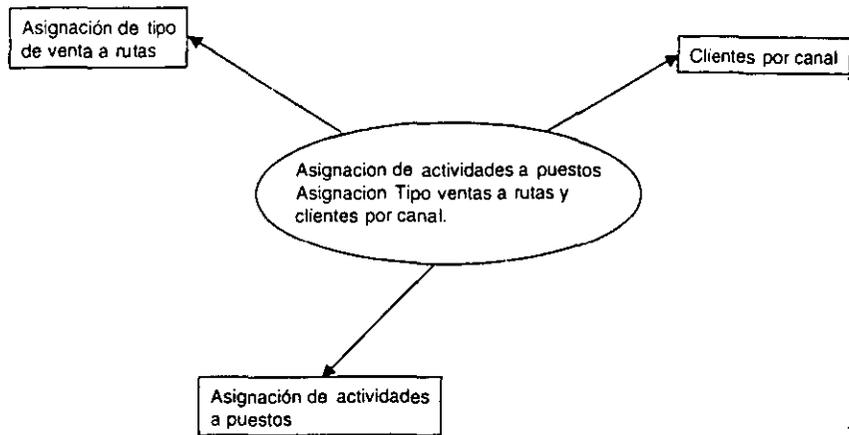
3.1.1.3.1.3. Diagrama de contexto de Reporte cooperativo



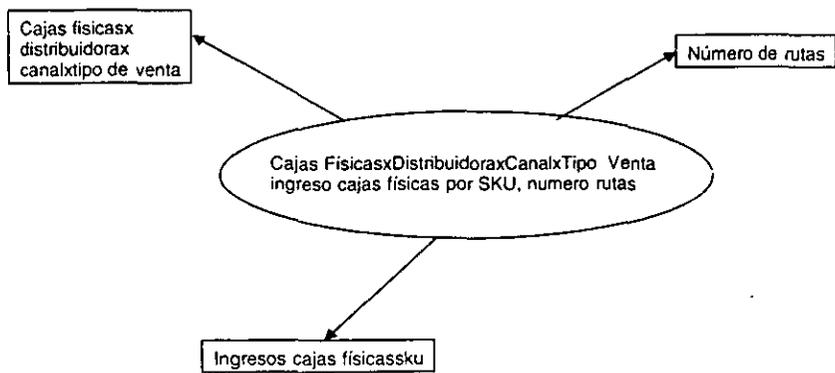
3.1.1.3.1.4. Diagrama de contexto de Reporte de Capital



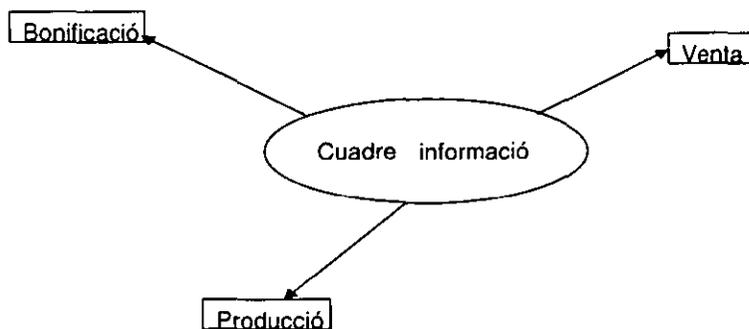
3.1.1.4. Diagrama de contexto de Consulta



3.1.1.4.1. Diagrama de contexto de Asignación de actividades



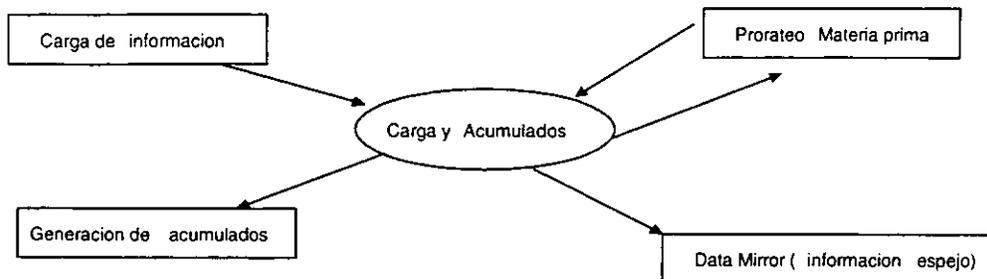
3.1.1.4.2. Diagrama de contexto de cajas



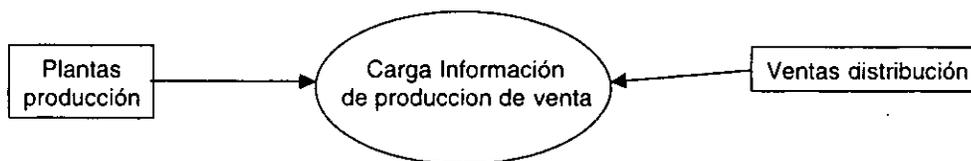
3.1.1.4.3. Diagrama de contexto de cuadro de Información

En la figura 3.1.1.5. nos muestra el diagrama de contexto de carga y acumulados, en donde se carga vía disquete los acumulados de cada punto y se generan los acumulados.

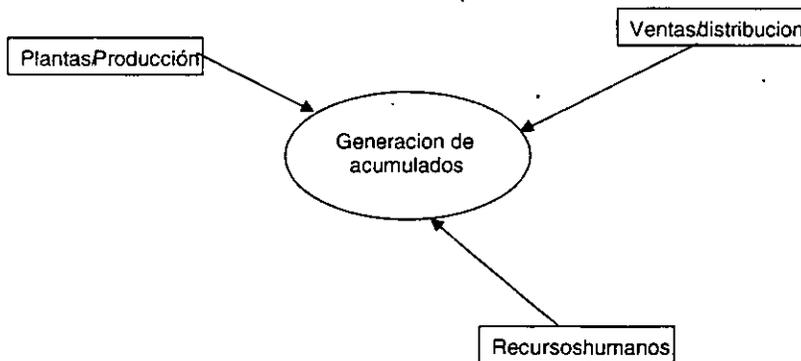
En la figura 3.1.1.6. nos muestra el diagrama de contexto de administración donde se registra una bitácora y se lleva un control monitor.



3.1.1.5. Diagrama de contexto de Carga y



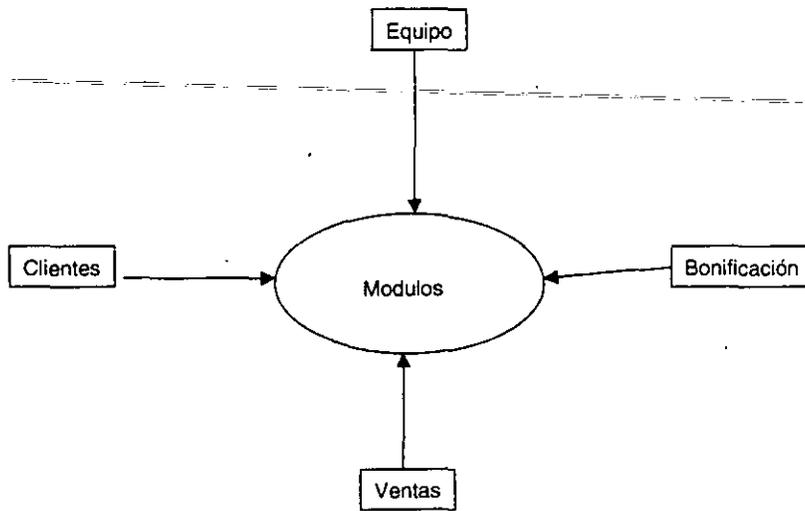
3.1.5.1.1. Diagrama de contexto de carga de información de Producción de venta



3.1.1.5.2. Diagrama de contexto de generación de acumulados



3.1.1.5.3. Diagrama de contexto data mirror

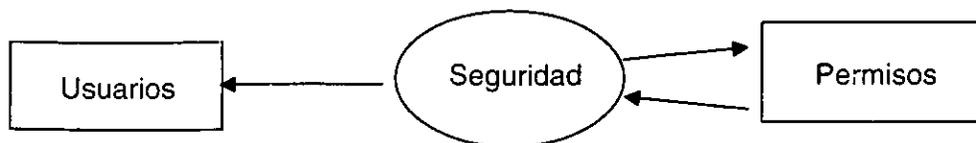


3.1.1.5.3.1. Diagrama de contexto de Módulos



3.1.1.6. Diagrama de contexto de Administración

En la figura 3.1.1.7. se muestra el diagrama de contexto de seguridad.



3.1.1.7. Diagrama de contexto de

3.1.2. Diagrama Dinámico de eventos.

En el análisis orientado a objetos el mundo se describe en términos de los objetos y sus estados, así como de los eventos que modifican esos estados. Como ya se mencionó en el capítulo 1, un evento es un cambio en el estado de un objeto; sin los eventos, el mundo no cambiaría. En un mundo sin eventos podríamos construir y generalizar bases de datos sin preocuparnos por actualizarlas. Sin embargo, en la mayoría de las aplicaciones, si debe cambiar el contenido de las bases de datos, puesto que deseamos saber de esos cambios y reaccionar en forma adecuada ante ellos, deberemos entender y modelar los eventos.

Una operación hace que los eventos ocurran. Los eventos indican los puntos en el tiempo en que se da el cambio de estado de un objeto. Según el área que se modele, puede ocurrir más de un evento al terminar una operación, y cada uno de estos puede activar operaciones independientes.

Las operaciones externas también se presentan y se utiliza un reloj cuando el tipo de evento lo provoca un reloj externo.

Una condición de control garantiza que un conjunto de eventos está completo antes de proceder con una operación. Antes de invocar de hecho a la operación, se verifica su condición de control. Si los resultados de la evaluación de la condición son verdaderos, se invoca a su operación, si son falsos no lo hace.

El diagrama de estado (Dinámico) nos muestra la secuencia del evento y estados permitidos en un sistema para una clase de objetos. Los diagramas de estados se refieren a otros modelos. Las acciones en el diagrama de estados corresponden a funciones del modelo funcional, los eventos en un diagrama de estados se convierten en operaciones en el modelo de objeto. Y todo lo anterior junto nos da una descripción funcional de todo el sistema.

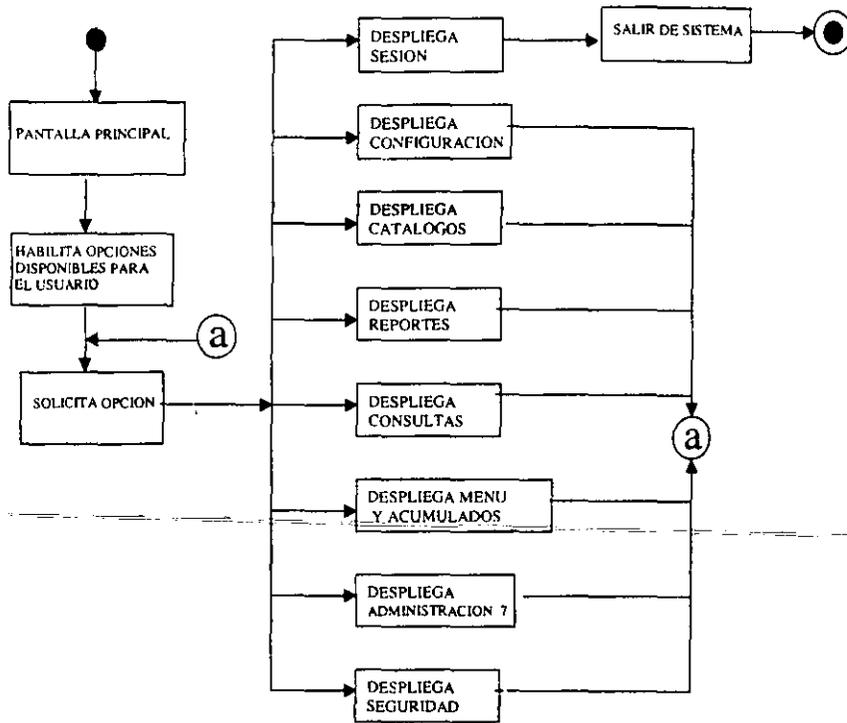


Figura 3.1.2. Diagrama de eventos general menu.

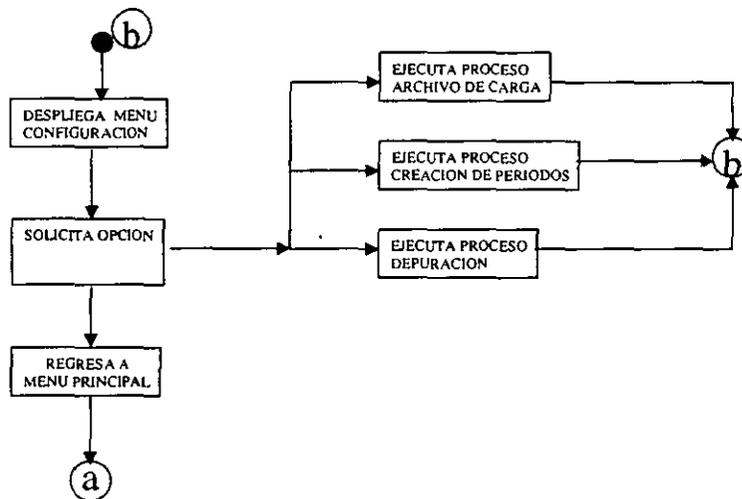


Figura 3.1.2.1. Diagrama de eventos configuración.

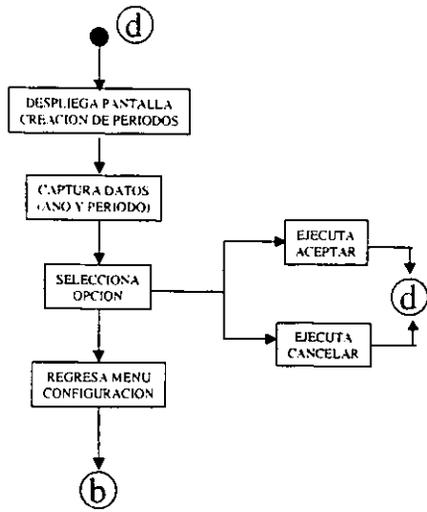


Figura 3.1.2.1.2. Diagrama de eventos configuración, creación de periodos.

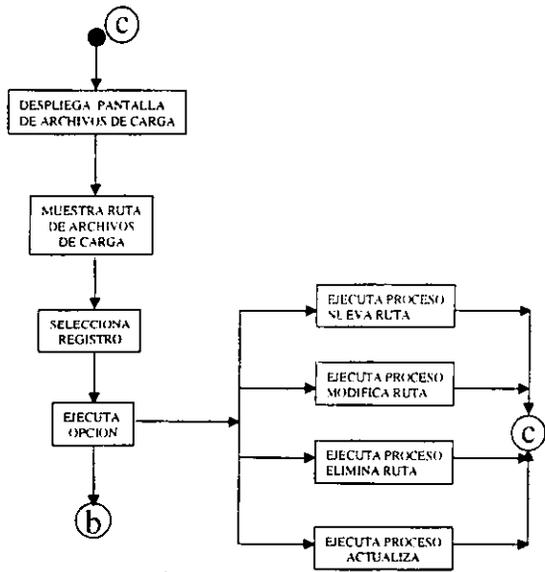


Figura 3.1.2.1.1. Diagrama de eventos configuración, archivos de carga.

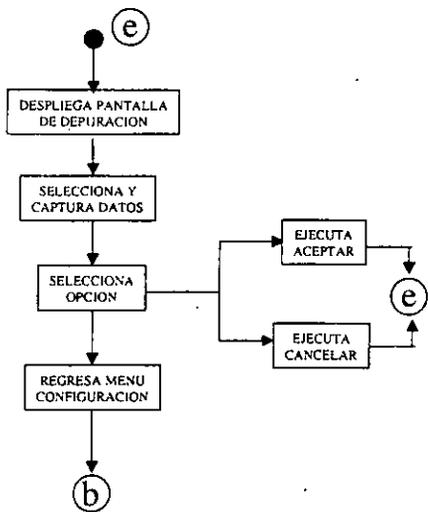


Figura 3.1.2.1.3. Diagrama de eventos configuración, proceso de depuración.

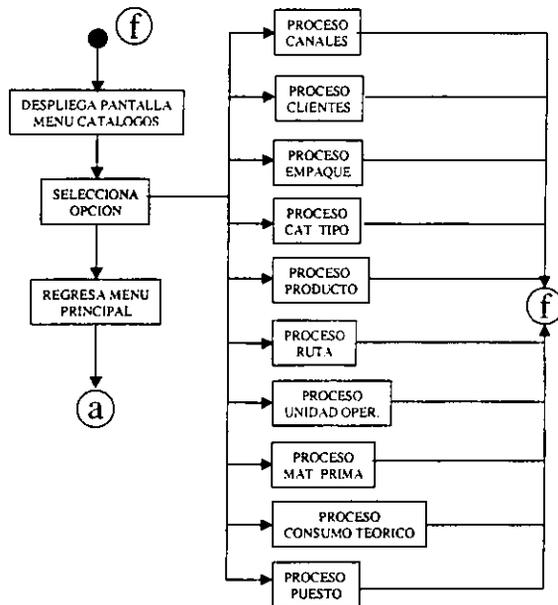


Figura 3.1.2.2. Diagrama de eventos catálogos.

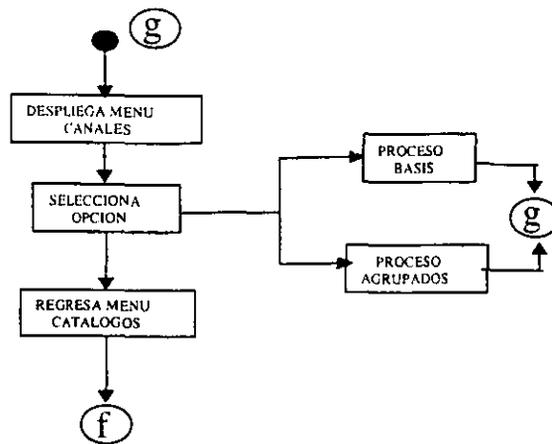


Figura 3.1.2.2.1. Diagrama de eventos canales

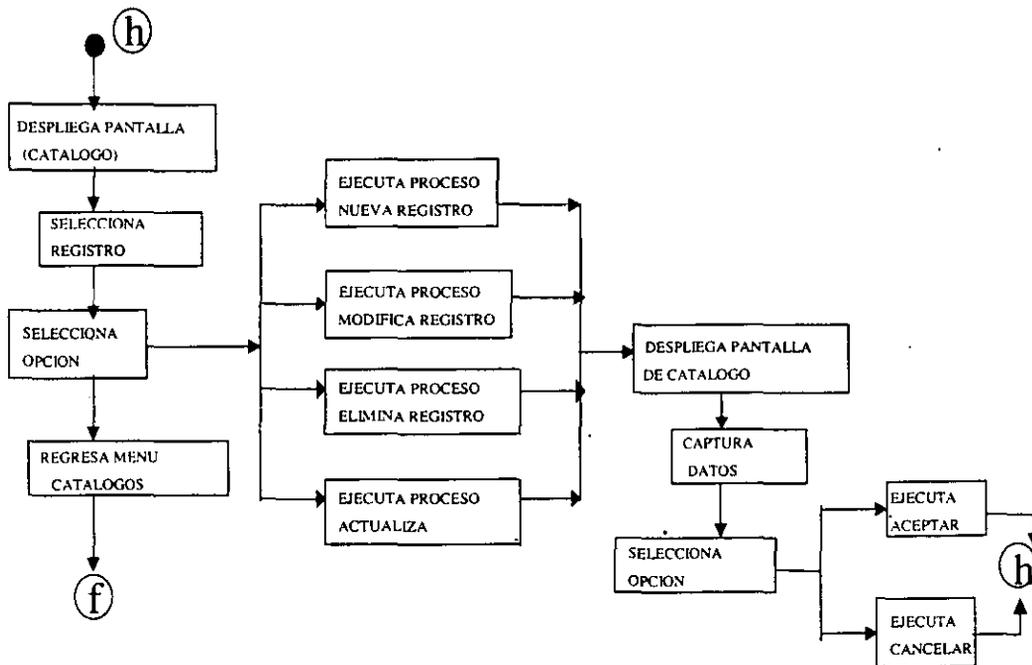


Figura 3.1.2.2.2. Diagrama de eventos, catálogo, canales, clientes, empaque, catalogo tipo, producto, ruta, unidad operativa, materia prima, consumo teórico y puesto.

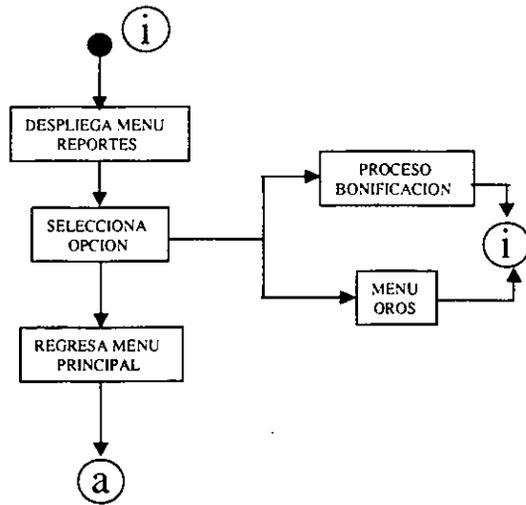


Figura 3.1.2.3. Diagrama de eventos reportes.

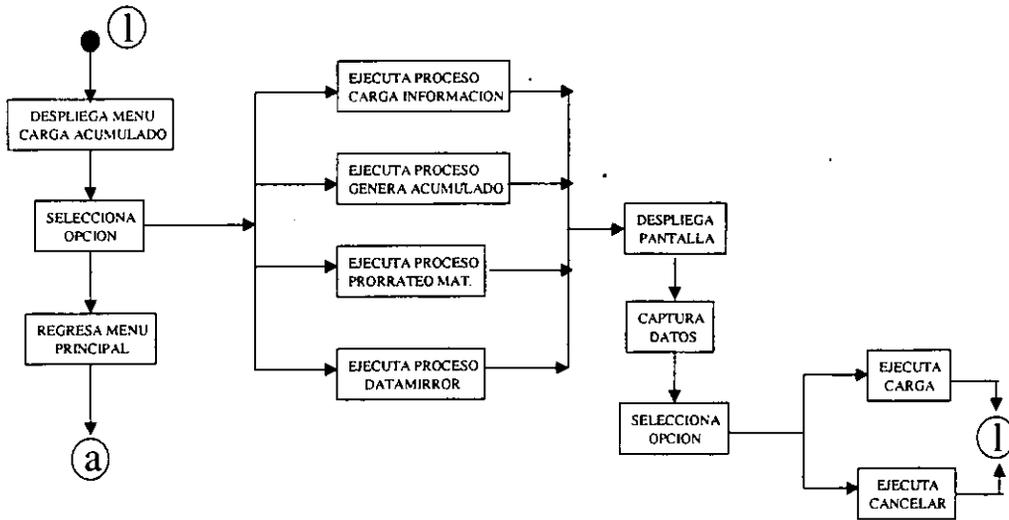


Figura 3.1.2.5. Diagrama de eventos carga

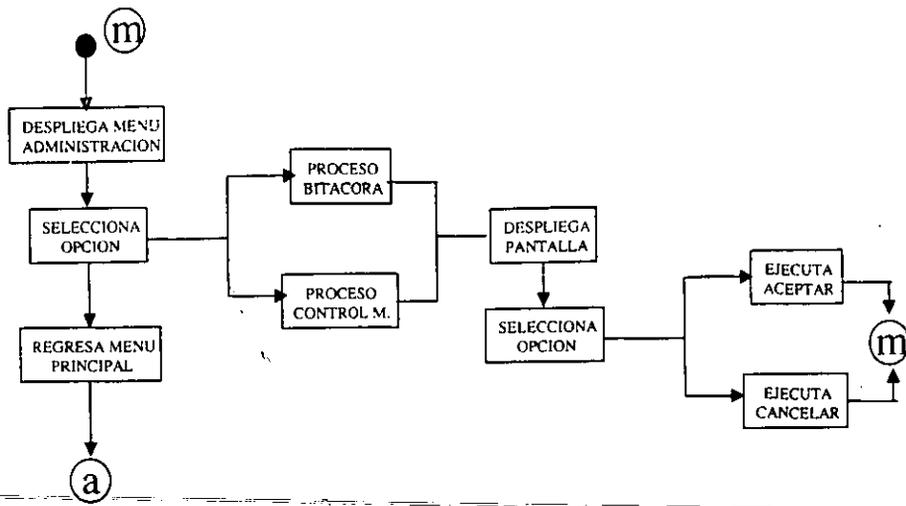


Figura 3.1.2.4. Diagrama de eventos consultas.

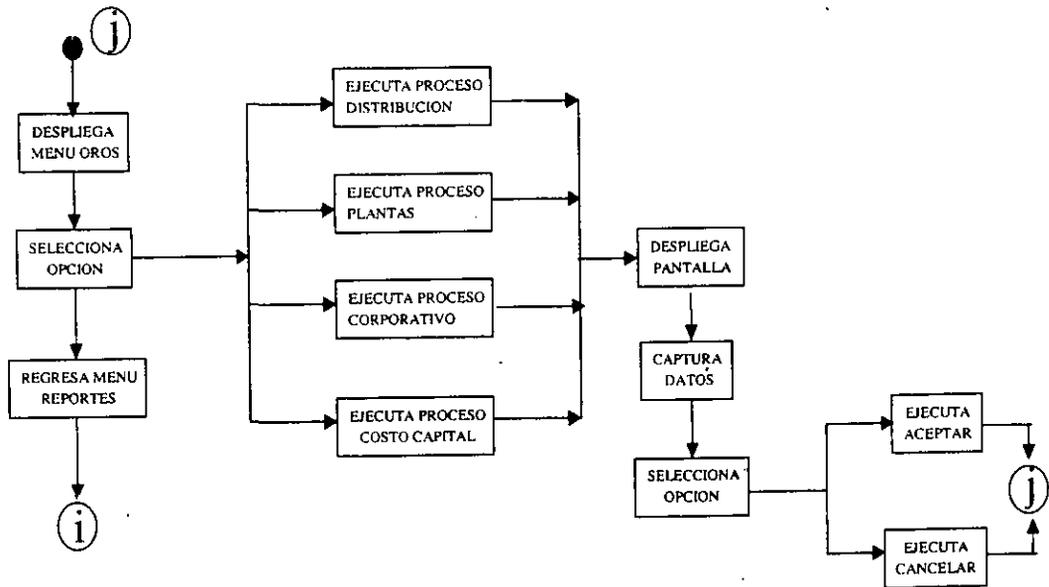


Figura 3.1.2.3.1. Diagrama de eventos reportes.

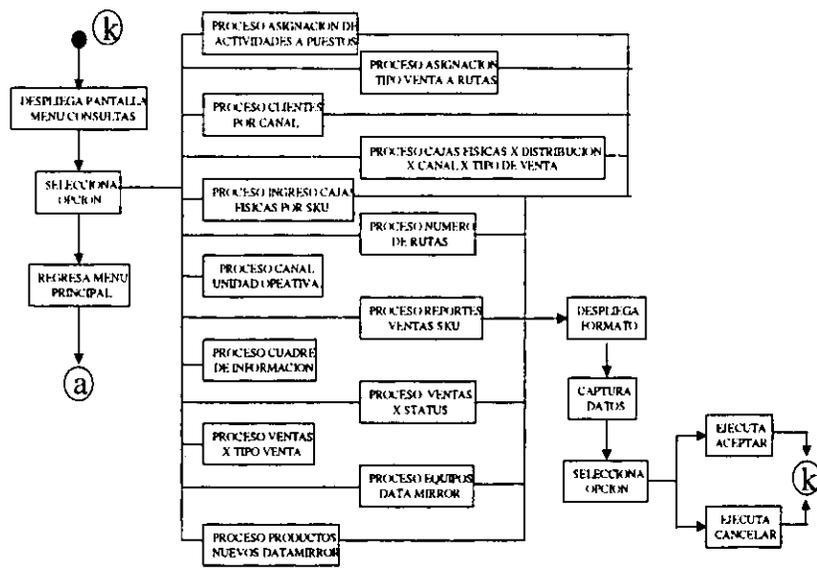


Figura 3.1.2.6. Diagrama de eventos administración.

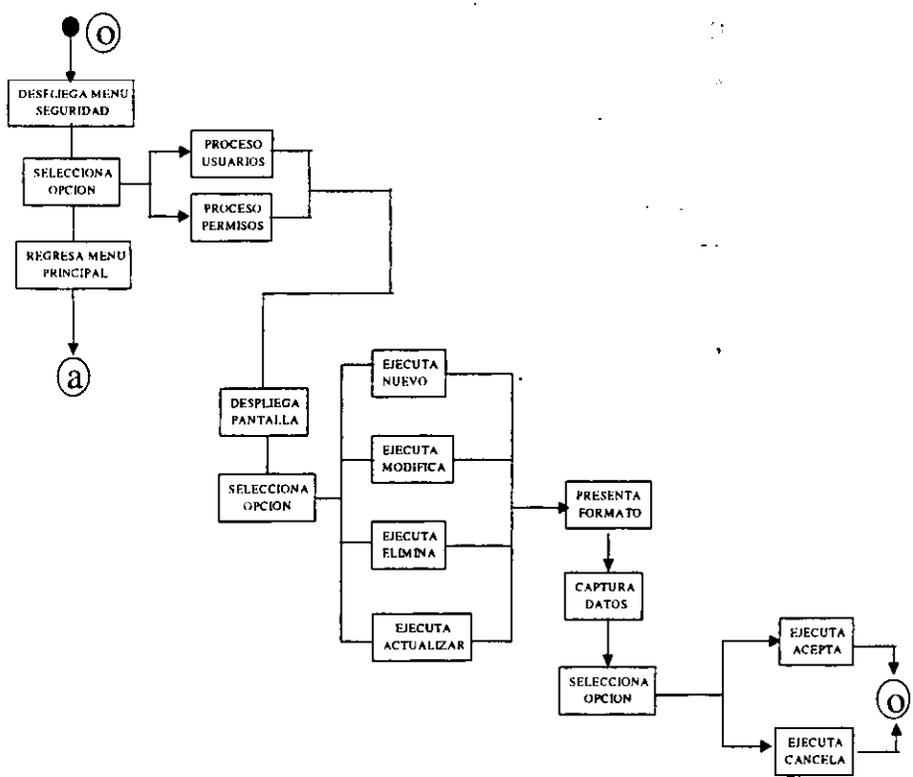


Figura 3.1.2.7. Diagrama de eventos seguridad.

3.1.3. Diccionario de Datos.

El diccionario de datos se utiliza para definir el significado, uso, características y otros datos relevantes de todos los campos, entidades de referencias cruzadas y las relaciones que existen entre ellos. Es decir, el diccionario de datos almacena información acerca de la estructura de la base de datos y la información de autorización, como las restricciones de la clave.

En un sistema de bases de datos se necesita tener información acerca de los datos y sus relaciones. Esta información se denomina *diccionario de datos*, o catálogo de sistema. Entre los tipos de información que el sistema debe almacenar están:

- Los nombres de las relaciones.
- Los nombres de los atributos de cada relación.
- Los dominios de los atributos.
- Los nombres de las vistas definidas en la base de datos y la definición de esas vistas.
- Las restricciones de integridad de cada relación (por ejemplo, las restricciones de clave).

Además de esto es recomendable que en la mayoría de los sistemas se conserven los siguientes datos:

- Nombres de los usuarios autorizados.
- Información contable acerca de los usuarios.

En los sistemas que utilizan estructuras altamente sofisticadas para almacenar relaciones, pueden conservarse datos estadísticos y descriptivos acerca de las relaciones:

- Nombre del índice.
- Nombre de la relación que se indexa.
- Atributos sobre los que está el índice.
- Tipo de índice.

Toda la información constituye, de hecho, una base de datos en miniatura. Generalmente es preferible almacenar los datos acerca de la base de datos en la misma base de datos, ya que ocasionalmente en otros sistemas se almacena esta información en otro lado.

Para todos los objetos que residen en nuestra base de datos se almacenan en lo que en nuestro medio se conoce como metadatos (datos definidos por datos). Este diccionario se almacena en una ubicación especial del resto de las tablas de trabajo o datos y es utilizada exclusivamente por el **Kernel** (núcleo) del manejador de bases de datos que utilizamos, es decir SQL Server.

Este diccionario de datos también nos permite validar la existencia de los objetos definidos en la base de datos proporcionando un acceso y mapas de ubicación física de almacenamiento.

Es una base de datos que contiene datos acerca de los datos. Es importante señalar que el diccionario de datos se tiene que analizar para saber cómo está formada la base de datos en su estructura.

Se debe tener especial cuidado en la creación del diccionario de datos puesto que determina una ruta de acceso óptimo para guardar o extraer los datos.

3.1.4. El Modelo Entidad-Relación.

Aunque el diagrama de flujos de datos es una herramienta muy útil para modelar sistemas, sólo resalta un aspecto principal de un sistema: sus funciones. La notación de los

R9	Entregas de productos por Unidad Operativa
R10	Canal por Alterno
R11	Clientes por Ruta de Distribución
R12	Equipo de producción por Cliente
R13	Ventas por Cliente
R14	Equipo por Ruta
R15	Ventas mensuales por Ruta
R16	Ventas mensuales por Cliente
R17	Ventas mensuales por Canal alternativo
R18	Ventas por mes
R19	Entregas por Canal Agrupado
R20	Entregas por Tipo de Venta
R21	Tipo de Venta por Zona
R22	Ventas mensuales por Unidad Operativa
R23	Ventas mensuales por producto
R24	Acumulado de ventas por mes
R25	Consumo de Materia Prima por Periodo
R26	Consumo de Materia Prima por Tipo
R27	Consumo Real por Tipo
R28	Estado que guarda cada equipo de Producción
R29	Cuadre de producción por Producto
R30	Producción por Producto
R31	Cuadre de Producción
R32	Producción acumulada por mes
R33	Producción acumulada al mes por producto
R34	Consumo real de Materia Prima
R35	Consumo real por acumulados
R36	Consumo Teórico por acumulados
R37	Consumo Real versus Consumo Teórico
R38	Consumo Teórico de Materia Prima
R39	Producción Acumulada por mes
R40	Producción Acumulada al mes por Unidad Operativa
R41	Histórico de Recursos Humanos por Periodo
R42	Equipo Acumulado por Periodo
R43	Equipo Acumulado al mes
R44	Productos con estándar de calidad
R45	Productos con estándar COQA por Unidad Operativa
R46	Recursos Humanos por Unidad Operativa
R47	Acumulados de Recursos Humanos
R48	Acumulados de Recursos Humanos por Mes

agregados de datos en los diagramas de flujo de datos muestra la existencia de uno o más grupos de datos almacenados, pero deliberadamente dice muy poco acerca de sus detalles.

Todos los sistemas almacenan y usan información acerca del ambiente en el cual interactúan, a veces, la información es mínima, pero en la mayoría de los sistemas actuales es bastante compleja. No sólo deseamos conocer en detalle qué información hay de cada agregado de datos, sino que también queremos conocer la relación que existe entre agregados. Este aspecto del sistema no es resaltado por el diagrama de flujo de datos, pero sí lo hace otra herramienta: el diagrama de entidad-relación.

El diagrama de entidad-relación consta de dos componentes básicos principales:

1. Tipos de objetos. Se representan por medio de un rectángulo en el diagrama. Esto representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema; pueden además ser identificados de manera única y ser descritos por uno o más atributos.

2. Relaciones. Se representan por medio de rombos en el diagrama y son la serie de conexiones o asociaciones entre los tipos de objetos que están conectados con la relación por medio de flechas

El objeto es algo del mundo material del mundo real, y el tipo de objeto es su representación en el sistema. Sin embargo un objeto también puede ser algo no material: por ejemplo, horarios, planes, estándares, estrategias y mapas.

Los objetos se conectan entre sí mediante relaciones. Una relación representa un conjunto de conexiones entre objetos, y se representa por medio de un rombo, como se muestra en la figura 3.1.4.1.

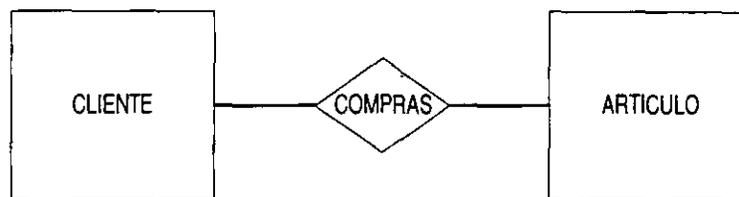


Figura 3.1.4.1. Una relación.

Es importante reconocer que la relación representa un conjunto de conexiones. Cada instancia de la relación representa una asociación entre cero o más ocurrencias de un objeto y cero o más ocurrencias del otro.

Relaciones del diagrama de Entidad-Relación.

Relación	Descripción
R1	Clientes por Unidad Operativa
R2	Equipo de producción por Unidad Operativa
R3	Clientes por Grupo
R4	Clientes por Canal
R5	Bonificación generada por Unidad Operativa
R6	Bonificación por Cliente
R7	Ventas que se adjudican por Unidad Operativa
R8	Bonificación que se genera por Canal

R49	Actividad por Puesto
R50	Vista de Equipo
R51	Puestos en Recursos Humanos
R52	Histórico de Recursos Humanos por Unidad Operativa
R53	Puestos por Unidad Operativa
R54	Equipo Acumulado por Unidad Operativa

Los diagramas Entidad-Relación se encuentran plasmados en el Apéndice A.

3.1.5. Normalización.

Codd ha identificado ciertas características estructurales en las relaciones que crean problemas de recuperación y actualización. Estas características indeseables pueden eliminarse descomponiendo una relación por otras, de estructuras deseables. Este proceso se conoce como normalización. Existen tres tipos de propiedades indeseables:

- Grupos de repetición.
- Uso múltiple del mismo atributo.
- Múltiple ocurrencia del mismo hecho.

Definición:

La normalización es un método que logra que una base de datos se encuentre en una consistencia lógica con un mínimo de redundancia, es decir, el objetivo de la normalización y del buen diseño de una base de datos es reducir, no necesariamente eliminar, la redundancia de los datos. Puede haber ocasiones, sin embargo en las cuales una limitada redundancia de datos es necesaria por razones de rendimiento.

Beneficios:

- Bases de datos más pequeñas, lo que hace más rápida la entrada/salida.
- Las actualizaciones en la información son más rápidas.
- Las tablas y renglones son más pequeños.

Desventajas:

- Existe un gran número de tablas.
- Incrementa el número de joins.
- No permite sumarizar datos cuando se necesitan.

La teoría de la normalización esta basada en el concepto de formas normales. Se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones.

Se han definido numerosas formas normales. Codd definió originalmente la primera, segunda y tercera formas normales (1FN, 2FN y 3FN). Ahora bien, se dice que todas las relaciones normalizadas están en 1FN; algunas relaciones en 2FN también están en 3FN. La motivación subyacente a la definición era que la 2FN resultaba "más deseable" que la 1FN y la 3FN, a su vez más deseable que la 2FN. Es decir, el diseñador, por lo general, debe escoger relaciones en 1FN o en 2FN; aunque esta proposición no debe interpretarse como ley, el único requisito riguroso es que las relaciones estén al menos en primera forma normal.

Primera forma normal (1FN).

Una tabla está en la primera forma normal (1FN) sólo si tiene una clave principal y sus atributos son todos tipos de datos simples sin atributos repetidos. Para cumplir con las restricciones de 1FN, los dominios de los atributos de una tabla deben ser valores atómicos, y no pueden ser grupos repetidos de atributos. Todos los grupos repetidos de atributos se deberán mover a una nueva tabla. Cuando una tabla se ajusta a 1FN, se le considera como una tabla normalizada, pareciéndose a un archivo plano de dos dimensiones. Los elementos que contienen grupos repetidos se consideran como no normalizados.

Por tanto, la finalidad de la primera forma es simplificar la estructura de una relación asegurando que contenga solamente datos elementales y no agrupaciones de datos. Es decir cada identificador en una relación debe poseer un solo valor de cada uno de los atributos y no múltiples valores de estos.

La información que se manejará para la explicación de la normalización corresponde a una información en bruto mostrada en la figura 3.1.5.1, a la cual se le aplicará dicho proceso.

Mirando a la tabla "Producción" una de las principales se puede descubrir que existen tres grupos repetidos, los cuales son los que tienen que ver con productos, unidades operativas y rutas, por tanto, la información repetitiva se trasladará a tres tablas diferentes de la de producción, quedando las tablas en primera forma normal (1FN) mostradas en la figura 3.1.5.2.

Producción
Produccion_id
Produccion_fecha
Produccion_linea
Produccion_prodcf
Produccion_numpersonas
Produccion_paradasP
Produccion_paradasA
Produccion_parosxmto
Produccion_hrsNP
Produccion_cajas
Produccion_hrs_pagform
Produccion_hrsest
Produccion_rotura_bod
Produccion_rotura_dis
Produccion_rotura_prod
Produccion_matprima
Produccion_status
UnidadOper_cvedsc
UnidadOper_dsc
UnidadOper_cvefuente
UnidadOper_tipo
UnidadOper_cve_rechum
UnidadOper_oros_id
UnidadOper_oros_dsc
UnidadOper_sap
Zona_id
Zona_dsc
Ruta_id
TipoVenta_id
TipoVenta_dsc
Producto_id
Producto_dsc
Producto_cajasXpal
Producto_faccubico

Producto_cajas
Producto_acumulable
Producto_idacumulado
Empaque_id
Empaque_dsc
Empaque_tipo
Empaque_botellaxcaja
Empaque_cantidad
Empaque_factorCub
Empaque_dsc_oros
Marca_id
Marca_dsc
Marca_oros_dsc

Figura 3.1.5.1 Tabla Producción.

Producción	Producto	Unidad Operativa	Ruta
Produccion_id	Producto_id	UnidadOper_cvedsc	UnidadOper_cvedsc
UnidadOper_cvedsc	Producto_dsc	UnidadOper_dsc	Ruta_id
Producto_id	Producto_cajasXpal	UnidadOper_cvefuente	TipoVenta_id
Produccion_fecha	Producto_faccubico	UnidadOper_tipo	TipoVenta_dsc
Produccion_linea	Producto_cajas	UnidadOper_cve_rechum	
Produccion_prodcf	Producto_acumulable	UnidadOper_oros_id	
Produccion_numpersonas	Producto_idacumulado	UnidadOper_oros_dsc	
Produccion_paradasP	Empaque_id	UnidadOper_sap	
Produccion_paradasA	Empaque_dsc	Zona_id	
Produccion_parosxmtto	Empaque_tipo	Zona_dsc	
Produccion_hrsNP	Empaque_botellaxcaja		
Produccion_cajas	Empaque_cantidad		
Produccion_hrs_pagform	Empaque_factorCub		
Produccion_hrsest	Empaque_dsc_oros		
Produccion_rotura_bod	Marca_id		
Produccion_rotura_dis	Marca_dsc		
Produccion_rotura_prod	Marca_oros_dsc		
Produccion_matprima			
Produccion_status			

Figura 3.1.5.2 Tabla Producción, Tabla Producto, Tabla Unidad Operativa y Tabla Ruta en 1FN.

Ahora que se han normalizado elementos (conforme a 1FN), se puede proceder con el siguiente paso de la normalización.

Segunda forma normal (2FN).

Una tabla está en segunda forma normal (2FN), si y sólo si está en primera forma y cada atributo que no es llave es funcionalmente dependiente de la llave principal (todos los atributos no-llave deben ser totalmente dependientes de la llave principal).

En la tabla "Ruta", la llave principal consta de la clave de unidad operativa, la clave de la ruta y de la clave del tipo de venta. El atributo "TipoVenta_dsc" no es totalmente dependiente de toda la llave principal, ya que sólo es dependiente en parte de la llave principal: la clave del tipo de venta. El atributo "TipoVenta_dsc" no tiene dependencia de la otra parte de la llave principal.

Para cumplir las restricciones de la segunda forma normal, se crea una tabla "Tipo de Venta", y se traslada el atributo que no es totalmente dependiente de la llave principal de la tabla "Ruta" a la tabla "Tipo de Venta".

Ya tenemos cinco elementos en la segunda forma normal, como se ve en la figura 3.1.5.3. Ahora que todos los elementos están en 2FN, se puede refinar el modelo un poco más.

Produccion	Producto	Unidad Operativa	Ruta
Produccion_id	Producto_id	UnidadOper_cvedsc	UnidadOper_cvedsc
UnidadOper_cvedsc	Producto_dsc	UnidadOper_dsc	Ruta_id
Producto_id	Producto_cajasXpal	UnidadOper_cvefuente	TipoVenta_id
Producc_fecha	Producto_faccubico	UnidadOper_tipo	
Producc_linea	Producto_cajas	UnidadOper_cve_rechum	
Producc_prodcf	Producto_acumulable	UnidadOper_oros_id	
Producc_numpersonas	Producto_idacumulado	UnidadOper_oros_dsc	
Producc_paradasP	Empaque_id	UnidadOper_sap	
Producc_paradasA	Empaque_dsc	Zona_id	
Producc_parosxmto	Empaque_tipo	Zona_dsc	
Producc_hrsNP	Empaque_botellaxcaja		
Producc_cajas	Empaque_cantidad		
Producc_hrs_pagform	Empaque_factorCub		
Producc_hrsest	Empaque_dsc_oros		
Producc_rotura_bod	Marca_id		
Producc_rotura_dis	Marca_dsc		
Producc_rotura_prod	Marca_oros_dsc		
Producc_matprima			
Producc_status			
Tipo Venta			
TipoVenta_id			
TipoVenta_dsc			

Figura 3.1.5.3. Tabla Producción, Tabla Producto, Tabla Unidad Operativa, Tabla Ruta y Tabla Tipo Venta en 2FN.

Tercera forma normal (3FN).

Una relación esta en tercera forma normal (3FN) si está en la segunda (2FN) y no contiene ninguna dependencia transitiva. Las dependencias transitivas son dependencias funcionales entre atributos que no son llave.

Cualquier atributo sin llave que sea funcionalmente dependiente de otro atributo sin llave de la misma tabla genera una dependencia transitiva y debe ser trasladado a una nueva tabla.

Para este caso, los atributos del empaque en la tabla "Producto" son funcionalmente dependientes del número de empaque (Empaque_id), que no forma parte de la llave principal. El número de empaque es funcionalmente dependiente del número de producto (Producto_id), haciendo a los atributos del empaque transitivamente dependientes del número de producto.

Produccion	Producto	Unidad Operativa	Ruta
Produccion_id	Producto_id	UnidadOper_cvedsc	UnidadOper_cvedsc
UnidadOper_cvedsc	Empaque_id	Zona_id	Ruta_id
Producto_id	Marca_id	UnidadOper_dsc	TipoVenta_id
Producc_fecha	Producto_dsc	UnidadOper_cvefuente	
Producc_linea	Producto_cajasXpal	UnidadOper_tipo	
Producc_prodcf	Producto_faccubico	UnidadOper_cve_rechum	
Producc_numpersonas	Producto_cajas	UnidadOper_oros_id	
Producc_paradasP	Producto_acumulable	UnidadOper_oros_dsc	
Producc_paradasA	Producto_idacumulado	UnidadOper_sap	
Producc_parosxmto			
Producc_hrsNP			
Producc_cajas			
Producc_hrs_pagform			
Producc_hrsest			
Producc_rotura_bod			

Producc_rotura_dis Producc_rotura_prod Producc_matprima Producc_status			
Tipo Venta	Empaque	Marca	Zona
TipoVenta_id TipoVenta_dsc	Empaque_id Empaque_dsc Empaque_tipo Empaque_botellaxcaja Empaque_cantidad Empaque_factorCub Empaque_dsc_oros	Marca_id Marca_dsc Marca_oros_dsc	Zona_id Zona_dsc

Figura 3.1.5.4. Tabla Producción, Tabla Producto, Tabla Unidad Operativa, Tabla Ruta, Tabla Tipo Venta, Tabla Empaque, Tabla Marca y Tabla Zona en 3FN.

Se deberían trasladar todos los atributos de empaque en la tabla "Producto" a una nueva tabla "Empaque" (con el número de empaque como la llave principal). El número de empaque continuará en la tabla "Producto" como la clave foránea. También deberían trasladarse los atributos de marca en la tabla "Producto" y los atributos de zona en la tabla "Unidad Operativa" a un nuevo elemento. En la figura 3.1.5.4., se han añadido estos tres nuevos elementos: Empaque, Marca y Zona.

3.2. GENERACIÓN DE CÓDIGO PARA EL PROCESAMIENTO DE LA INFORMACIÓN.

3.2.1. Construcción del Back End.

Una vez concluida la fase de diseño de la estructura de la Base de Datos en la sección anterior, se procederá a la implementación de la misma.

Cumpliendo con la línea de los productos Microsoft, SQL Server posee una amigable interfaz gráfica para el manejo de todos sus recursos que nos provee como Manejador de Base de Datos Objeto/Relacional.

Para la implementación de una Base de Datos en SQL Server es necesario que primero exista un objeto denominado "device". Un device es un archivo de operación de sistema (de SQL), en el cual se almacena la BD. En este device adicionalmente se alberga el "transaction" log (registro de transacciones), que es un área reservada por SQL para almacenar los cambios realizados sobre la BD. Cada BD posee su propio transaction log.

Cada cambio hecho sobre la BD se graba automáticamente en este transaction log; es decir funge como una bitácora de operaciones.

El device define físicamente el espacio que nuestra BD tendrá en cuestión, puede contener más de una BD. Con el uso del comando DIR se puede visualizar el archivo de extensión .dat que se genera al crearlo. La o las BD's pueden o no ocupar la totalidad de espacio reservado en el device.

Un device se crea a partir de un mínimo de 1 MB, éste solo puede ser creado desde la computadora local, es decir desde el servidor; y posee dos nombres, un físico y un lógico. En SQL se pueden crear los devices de dos maneras distintas:

- **De forma gráfica**
- **Desde la línea de comandos de SQL.**

Forma gráfica. (SQL Enterprise Manager).

Se utiliza el administrador de SQL, y para acceder a este, se deben tener permisos de administrador, ver figura 3.2.1.1.

El SQL Enterprise Manager es uno de los servicios que nos proporciona el manejador y que normalmente se configura para autocargarse al inicio de la sesión del servidor, y es la herramienta gráfica con que cuenta el producto para la creación de las instancias necesarias en nuestra construcción de una BD.

Los requerimientos de operación de la interfaz gráfica son los siguientes:

- **Iniciado el servicio Executive de SQL**
- **Iniciado el servicio del Enterprise Manager**

Estos dos servicios permiten la operación propiamente de SQL. Normalmente se predeterminan para que al levantarse el servidor de Windows NT se autoinicien los servicios denominados de aplicación, que son los de Server de SQL y el Executive.

Si se configuró de manera distinta, será necesario el iniciar estos servicios antes de utilizar el Enterprise Manager, mostrado en la figura 3.2.1.1.

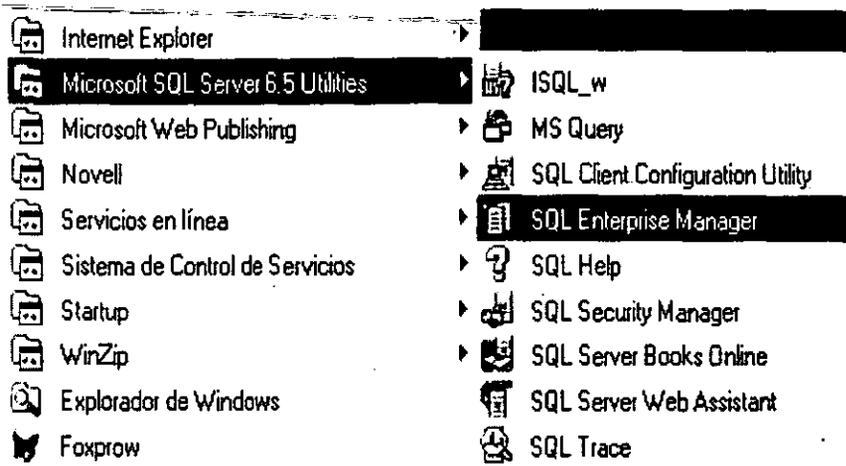


Figura 3.2.1.1 El SQL Enterprise Manager.

Si los servicios están bien configurados se ingresa a la interfaz gráfica.

De entrada esta ventana nos muestra las BD's , los devices de respaldo, los devices y logins del servidor.

Para crear un nuevo device:

Desde la barra de menús, seleccionamos Databases Devices.

Al aparecer esta ventana de creación, se ingresa el nombre del device, que será el nombre lógico con el que SQL lo reconocerá, la dirección, el drive y el nombre físico donde se guardará el archivo .dat; y el tamaño del contenedor.

Se debe seleccionar un tamaño adecuado para el tamaño de nuestro device, producto del análisis de la BD.

La ventana nos muestra el espacio disponible de la unidad actual y habilita el botón de creación.

Una vez hecho lo anterior se actualiza la pantalla con F5 y nos aparece dentro de la sección de Databases Devices la nueva instancia.

Cuando un Databases Devices es creado, sus nombres físico y lógico son ingresados en las tablas de información de SQL.

Forma manual. (Usando la sentencia DISK INIT).

Mediante esta forma, solo como "sa" se puede realizar la operación. El "sa" es el super administrador de SQL .

Este método de creación se recomienda por las ventajas que posteriormente se mencionarán.

La sentencia DISK INIT es propia del lenguaje de SQL Server y la sintaxis es la siguiente:

DISK INIT

NAME='Nombre Lógico',
PHYSNAME='Nombre Físico',
VDEVNO=Número virtual de device,
SIZE=Tamaño en bloques de 2k

Para nuestro caso la sentencia es como se muestra a continuación:

DISK INIT

NAME = 'ABCCOSTING',
PHYSNAME = 'c:\sqldata\abccosting.dat',
VDEVNO = 3,
SIZE = 64000

Donde el VDEVNO es un número entre 1 y 255, que identifica numéricamente al contenedor, y el tamaño se designa en bloques de 2K.

Normalmente esta sentencia se guarda como un archivo del tipo script de SQL con extensión .sql, esto con la finalidad de recrear la BD ante una eventualidad, ya que se debe crear en el mismo orden en que se hizo por primera vez. Es decir si eventualmente crece la BD y posteriormente sufre una corrupción total, el transaction log junto con los backups nos permitirán recuperar la información, pero se debe recrear la BD en la secuencia de inicio, si inicialmente se tenía un tamaño definido y se incrementó, al momento de recrearla, se debe hacer con el tamaño inicial y posteriormente incrementarse, de otra forma no se podrá reconstruir la información almacenada..

El archivo con el que se guarda es un formato propio del producto, que identifica a las sentencias declaradas en el mediante el lenguaje de consultas, y las reconoce como archivos de ejecución al momento de ejecutarlas dentro de las herramientas que poseen una línea de comandos para la creación de estos procedimientos almacenados y que se pueden desencadenar tantas veces como sean necesarios y posibles dentro de la integridad que nos marca los lineamientos del manejador de BD's.

En la barra de menús del Enterprise , y dentro de Tools, se tiene la opción SQL Query Tool, que es la herramienta que nos proporciona una línea de comandos para ejecutar sentencias sql , es decir, programar en el lenguaje de consultas.

Al ingresar aparece la ventana de comandos.

Dentro de esta ventana se agrega el código correspondiente y se ejecuta su contenido.

Hecho lo anterior se consigue el mismo objetivo que la forma gráfica, pero con la posibilidad de guardar el script como un programa almacenado.

Ya que se construyó el contenedor , podemos disponernos a crear la BD.

Al igual que con el device se puede crear por dos métodos diferentes nuestra BD.

Creación de la BD mediante el uso del SQL Enterprise Manager.

Para crear la BD de la barra de menús seleccionamos Manage y posteriormente Databases.

Al seleccionar lo anterior se nos muestra una ventana similar a la de creación de un nuevo device, se nos muestra las BD existentes. Optamos por crear una nueva BD desde la barra de menús o mediante el botón de nueva BD.

Seguidamente nos aparece la ventana de creación.
En esta ventana se debe ingresar:

- El nombre de la BD. El nombre debe cumplir con las reglas de SQL para su identificación.
- El nombre del device sobre el cual se almacena.
- El monto del total del espacio disponible del device para la creación de la BD.
- El nombre del device sobre el que se realizarán las tareas del transaction log.

Cumplido lo anterior se habilita el botón de creación y nos aparece un mensaje que nos indica la creación satisfactoria de la BD después de pulsar el botón Create Now.

Creación de la BD mediante el uso de la sentencia Create Database.

Este método requiere del uso de alguna de las herramientas que poseen una línea de comandos.

La sintaxis es la siguiente:

```
Create Database nombre de la BD  
ON Database_Device = tamaño  
LOG ON Database_Device = tamaño
```

Al igual que en la creación del device se recomienda almacenar la sentencia como un script, para apoyar las tareas de una eventual reconstrucción de la BD.

Ya que se tienen creados el device y la BD se procede a crear los objetos y estructuras que integrarán la colección de datos denominada BD. Esto se consigue mediante la ejecución del programa almacenado de extensión .sql, que implementa las tablas, vistas, y procedimientos almacenados que se diseñaron con anterioridad pero ya implementados en el lenguaje de consultas de SQL.

Para lo cual, desde el SQL Query Tool, nos colocamos en la BD recién creada y se corre el script que generará las instancias necesarias para completar la fase de creación del Back End. Ver figura 3.2.1.14.

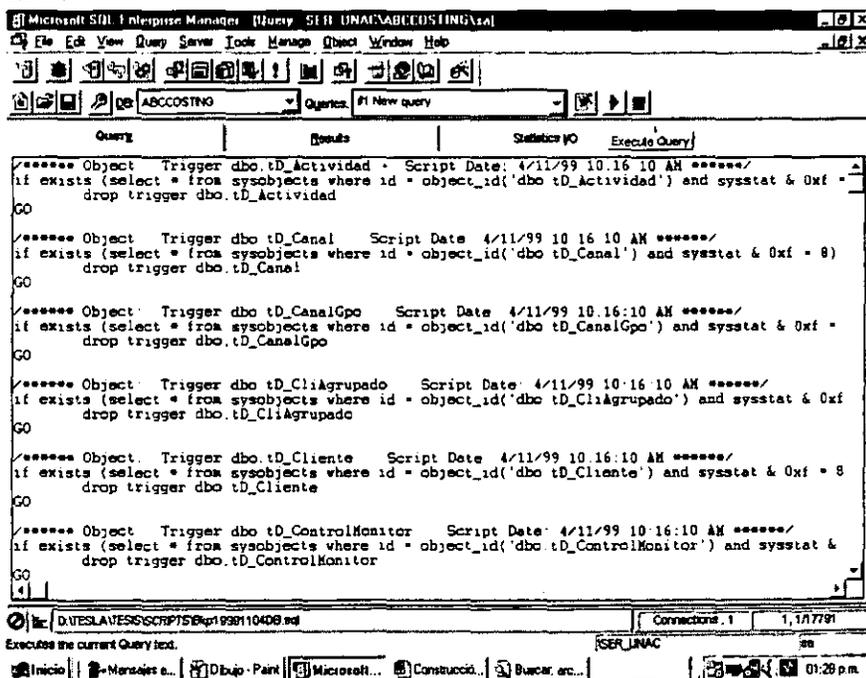


Figura 3.2.1.14. Finalizando la creación de la BD.

3.3. DISEÑO Y CONSTRUCCIÓN DEL FRONT-END.

De acuerdo al análisis presentado en la sección de ventajas y desventajas del software elegido, se conviene el diseño y construcción del entorno para acceder a la Base de Datos implementada.

Visual Basic en general provee de una interfaz gráfica sumamente amable en cuanto a la generación de aplicaciones como herramienta. En esta aplicación se trabaja basándose en instancias denominadas proyectos, y en ellas se alojan todos los elementos necesarios por la herramienta para realizar esta función.

Normalmente se requiere de la generación de un nuevo proyecto, en el que gradualmente se van agregando los elementos que conforman nuestro diseño proyectado. Este espacio llamado proyecto inicialmente se trata de una plantilla en blanco que se almacena con la información generada con un nuevo nombre. Al ejecutarse VB (Visual Basic), se nos pregunta sobre el tipo de aplicación a generar.

Dentro de las opciones se encuentra la posibilidad de crear controles especiales, librerías, documentos y la opción de asistente para aplicaciones; para este caso en particular se elige la forma estándar, una vez realizado lo anterior en la ventana de trabajo de VB aparece lo que se denomina el ambiente de diseño. Como se muestra en la Figura 3.3.1.

Los elementos básicos que lo conforman son los siguientes:

- **Toolbox**
- **Form**
- **Project Window**
- **Properties Window**
- **Code Window**
- **Toolbar**

Toolbox.

El Toolbox es un menú flotante que contiene todos los objetos y controles que se pueden agregar a las formas para la creación de nuestra aplicación. Cabe señalar que inicialmente se muestran sólo aquellos controles que se denominan estándar, quedando abierta la posibilidad de agregar nuevos objetos y controles, incluso de otros fabricantes distintos a Microsoft.

Form.

Que es un objeto contenedor general, sobre el que se depositan controles y objetos en tiempo de diseño, su apariencia es la de una ventana que funciona como la interfaz de nuestra aplicación.

Project Window.

El Project Window nos lista el código de los módulos y los archivos de apoyo que maneja nuestro actual proyecto. El proyecto se define como la colección de archivos que se utilizan para construir una aplicación. Esta herramienta nos permite el rápido acceso entre formas y código.

Properties Window.

El Properties Window nos muestra la lista de propiedades que se pueden manipular de controles y formas al tiempo de que son ejecutados. Una propiedad se define como el valor de un objeto como lo puede ser su tamaño, nombre o color.

Code Window.

En la ventana Code Window se visualiza las sentencias de código para su ingreso y edición.

Toolbar.

El Toolbar es la barra de herramientas estándar por excelencia de cualquier aplicación de una interfaz gráfica, en ella se encuentran los principales comandos usados en VB, como lo son Salvar el proyecto, Salvar, Propiedades, Iniciar, Parar, Salir, etc.

Adicionalmente al trabajar con VB, se debe estar familiarizado con los siguientes términos:

- **DesignTime vs. Run Time (Tiempo de diseño versus Tiempo de ejecución).**
- **Forms (Formas).**
- **Controls (Controles).**
- **Objects (Objetos).**
- **Properties (Propiedades).**
- **Methods (Métodos).**
- **Events (Eventos).**

Design Time vs. Run Time.

Design Time se define como el tiempo en que se está construyendo la aplicación en el ambiente de VB. Run Time es un tiempo donde se ejecuta la aplicación y se está en condiciones de interactuar con la misma, esto normalmente ocurre cuando se realiza un seguimiento para detectar un error lógico o de sintaxis en el programa. Es importante el saber conocer la diferencia entre estos términos ya que en ocasiones se realizan tareas muy específicas para alguno de estos tiempos en particular. Es decir hay controles que sólo en tiempo de ejecución se pueden visualizar o editar, mientras que en tiempo de diseño se trabajan la mayoría.

Forms.

Una forma funge como una ventana de interfaz entre el usuario y la aplicación, o como una ventana de diálogo. En ella se adicionan los controles, gráficos e imágenes para dar la apariencia que se requiere tanto estéticamente como funcionalmente. Una forma puede existir individualmente o puede servir como un documento o interfaz múltiple de documentos, donde un número de formas pequeñas existen sin una forma mayor.

Controls.

Son la representación gráfica de objetos como lo son botones, listas, cajas de edición que el usuario manipula para proveer de información a la aplicación.

Properties.

Son los valores de un objeto, tamaño, color, nombre, etc.

Methods.

Son las acciones que un objeto puede desempeñar o debe desempeñar sobre sí mismo, como refrescar imagen, insertar, borrar, etc.

Events.

Los eventos son acciones reconocidas por una forma o control. Los eventos ocurren cuando el usuario interactúa con los objetos de un programa. Dando clic en los controles, tecleando y moviendo el puntero del Mouse, son ejemplos de eventos.

Creando la Interfaz.

Una de las pantallas más representativas del sistema es la generación de las formas que contienen los catálogos de operación del sistema.

Se parte de la creación de una nueva forma. En el Project Window se hace clic con el botón derecho sobre la carpeta formularios y posteriormente en agregar, de donde surge un Submenu del cual se elige formulario

Una vez realizado lo anterior aparece una ventana que nos muestra varias opciones, haciendo válida la que indica Formulario (Para versiones en Español).

Ya anexada la forma en limpio se procede a agregar los controles necesarios para crear la apariencia estética y funcional que nuestra forma va a desempeñar. Para ello no servimos del Toolbar y del Toolbox.

Del Toolbox tomamos los controles a utilizar en nuestra ventana interfaz, haciendo clic sobre el control requerido y delineando sobre el área de trabajo el control. Al soltar el Mouse el control se dibuja en la superficie y se puede empezar a trabajar con él, se le da la estética requerida al control a través de la ventana de propiedades.

En la ventana de código (Code Window) se ingresan las sentencias de código que el control deberá ejecutar ya sea por métodos o eventos.

Eventualmente los controles se van acumulando sobre nuestra ventana, teniendo cuidado de salvar periódicamente los cambios sobre el proyecto a fin de no perder lo recién creado.

Ocasionalmente no se encuentra dentro del Toolbox todos los componentes que se requieren para nuestra forma y se deben de adicionar e instalar cuando así se requiera, la barra de herramientas nos lista únicamente los componentes denominados estándar, de tal forma que debemos recurrir al Toolbar.

Dentro del Toolbar se encuentra la opción de Proyecto y dentro de esta la opción Componentes.

Se pueden activar los que se muestran o en su caso los de otro fabricante de controles, (para el caso práctico de esta aplicación fue necesario por así pedirlo expresamente le empresa el uso de controles de otro fabricante que reunían ciertas características estéticas muy apreciadas por la misma) se pueden realizar múltiples selecciones, al aceptar los cambios estos controles se podrán habilitar desde nuestra Toolbox.

Como se muestra , los controles más básicos utilizados son los de tipo Etiquetas, Botones, Cajas de Texto, Frame (Un marco de relieve) y Grid (Un control que muestra una vista de la tabla de trabajo). Cada control se ingresa por separado, aunque también es posible dar de alta matrices de controles.

Los controles pueden ser referenciados dentro de la misma forma o ser llamados o modificados desde otra instancia.

Este caso se aplica de manera muy marcada en la generación de la forma del menú principal de la aplicación.

Para ello se requiere del Editor de Menús que se encuentra en el Toolbar, en la opción Herramientas y dentro de este Submenu, el editor o puede ser activado con la secuencia de teclas Ctrl+E.

Dentro de este editor se configuran los componentes del menú principal de nuestra aplicación. Los conceptos se ingresan de acuerdo a si son cabezas de Submenu o integrantes de uno. El símbolo & antes de una letra especifica su posible activación mediante la combinación de la tecla Alt y de la letra sobre la que se aplique el símbolo de llamada.

El editor nos ayuda a crear el esqueleto de la barra de Menú, pero todo el código que se requiere, se debe ingresar en la ventana del Code Window, cada opción se trata por separado y se disparan mediante el manejo del evento clic sobre la forma.

Para construir una pantalla de selección se procede al lo siguiente:

1ero. Se agrega una nueva forma.

Se agregan los controles del tipo frame para resaltar datos

Se modifican las propiedades del control para darle la apariencia requerida, en este caso se modifica la propiedad Caption que da nombre al objeto:

Se agrega un control que semeja una caja de texto, pero que adicionalmente , le da un formato especial de entrada a los datos, denominado MaskEdBox.

Se agregan los controles de tipo etiqueta, que permiten colocar identificadores de objetos en la forma.

Se agregan los controles de caja de texto ,que son controles que permiten el ingreso de información en formato libre y que se puede restringir su acceso.

También dentro de los controles principales se encuentra el combo box, que permite editar un lista de valores para hacer referencia o selección.

Y finalmente se agregan los controles del tipo botón, con ellos se desencadenan procesos internos propios de programación alojados en los eventos .

La forma ya terminada presenta el siguiente aspecto.

La mayoría de los controles poseen una propiedad que les da nombre e identifica gráficamente ante el usuario, para permitir la navegación y manejo de sus secuencias de operación. A la par del diseño de la forma se va agregando el código que ejecutará el control en la ejecución del programa.

Este código se inserta ya sea en sus eventos, propiedades o métodos.

Las pantallas de captura presentan el siguiente proceso de desarrollo.

Después de agregar una nueva forma al proyecto, se procede a agregar los controles anteriormente mencionados, adicionalmente se inserta un control distinto, el Grid, que es un objeto que sirve de interfaz para visualizar el contenido de una tabla , vista o consulta.

El código del programa permite reutilizar el formato de las pantallas base, para de esta forma optimizar el tamaño de la aplicación, diferenciándose entre si por los parámetros que acompañan a los procesos desencadenados y utilizando las bondades de referencia que la herramienta nos ofrece.

Algunas pantallas observan formatos más simples: Se inicia con una forma en limpio.

De la ventana de proyecto se da un click con botón derecho del Mouse y aparece un Submenu con la opción Agregar, esta llama a su vez otro Submenu con un listado de instancias, seleccionamos formulario del tipo nuevo. Es posible agregar formas de proyectos ya existentes, recomendándose esta propiedad, cuando deben cumplir funciones idénticas dentro de sus respectivos proyectos.

Posteriormente se agrega el control de selección adecuado para esta forma, un ComboBox que es un control que permite mostrar una lista de variables, y que para este caso desencadena una consulta derivada del valor seleccionado. Se agregan objetos de estética, líneas de relieve, etiquetas de identificación.

Posteriormente se agrega el control de tipo panel para dar un efecto de realce al control depositado en su interior, un DBGrid.

Ya solo falta incluir los controles de ejecución y salida de la forma, los botones de comando.

Los controles del tipo botón son de un fabricante diferente de Microsoft, llevan modificados las propiedades del Caption y Picture, para dar nombre e imagen al botón, y se agrega el código correspondiente a los controles.

3.4. PRUEBAS E INTEGRACIÓN DEL SISTEMA.

Las pruebas de software es un elemento crítico para la garantía o seguridad de la calidad de un sistema y presenta la última revisión de las especificaciones, diseño y codificación del mismo.

Las pruebas son esenciales en el desarrollo de cualquier sistema de software, un sistema sin pruebas es únicamente una aplicación sin confirmación y podrá funcionar o no, pero sin pruebas, no existe ninguna forma de conocer esto antes de utilizarlo. Los desarrolladores de software tienen que gastar aproximadamente el 40% del costo del desarrollo y tiempo en la planeación y diseño de las pruebas.

Para las pruebas del Sistema fijamos los siguientes objetivos:

1. La prueba es un proceso de ejecución de un programa con la intención de encontrar un error en él.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de encontrar un error no reconocible.
3. Una prueba satisfactoria es una que encuentra un error que el programador no localizó en las pruebas generales de su programa.
4. Llegar a una aceptación del Sistema por parte del usuario.

El objetivo real de las pruebas es: diseñar un conjunto de pruebas que sistemáticamente identifiquen diferentes tipos de errores y que los realice con un mínimo gasto de tiempo y esfuerzo, y así obtener la aprobación por el usuario.

Las pruebas son aplicadas a diferentes niveles de acuerdo al ciclo de vida del desarrollo de software, sin embargo para cada una de las pruebas se definen diferentes objetivos y se consideran diferentes premisas de acuerdo al nivel que se necesite evaluar. El centro de todo el desarrollo de las pruebas, es encontrar errores, pero los diferentes tipos de errores que se encuentran son marcados para cada nivel evaluado.

Condiciones importantes para la ejecución de las pruebas.

Condiciones y restricciones.

Generalmente son limitaciones o problemas de naturaleza técnica y están relacionadas con el desarrollo del proyecto en sí, la tecnología de pruebas, el estado de los ambientes de pruebas, etc.

Riesgos y suposiciones para las pruebas.

Los riesgos son aquellos factores que pueden afectar negativamente la ejecución de las pruebas. Las suposiciones son premisas que pueden afectar positiva o negativamente la ejecución de las pruebas complicando o facilitando las actividades de las pruebas.

Cobertura funcional de las pruebas.

Dentro de la cobertura funcional de las pruebas se deben describir y listar de manera clara y concisa las funciones, así como aquellas funciones a no ser probadas aún siendo parte del proyecto, ya que son necesarias especialmente cuando se requiere explicar él porque de su exclusión definiendo el alcance de las pruebas y delimitando responsabilidades.

Además se debe documentar el ciclo del sistema a ser simulado con el objeto de ejecutar cada una de las funciones objeto de las pruebas. Esta simulación suele ser realizada con muestras de datos fuera de especificaciones.

Descripción de la arquitectura del sistema.

Para ello se consideran las especificaciones del software sobre el cual esta construido el sistema tal como la plataforma, el software de base de datos, el sistema operativo, el lenguaje de programación, etc.

Infraestructura de pruebas.

La infraestructura de pruebas contempla lo siguiente:

Organización de las pruebas.

Definir la organización (puestos y responsabilidades) que es requerida para la construcción y ejecución de las pruebas.

Ambiente de pruebas.

Identificación de los ambientes donde se ejecutarán las pruebas, así como mencionar las características generales de los datos de prueba (que datos se necesitan y como se obtendrán). Tomando como base el modelo de datos del proyecto. Esta mención de los datos de prueba, es importante para saber cuántos y cuáles datos serán seleccionados, y para la estimación de la carga de trabajo necesaria para generarlos.

Herramienta de prueba.

En este punto es necesario identificar los productos a utilizar y el uso específico que se hace con ellos.

Básicamente se debe determinar si es preciso vigilar todos los componentes o solamente algunos, el interés de vigilar a determinados componentes, es justificado por la necesidad de verificar cual es el comportamiento interno de dicho componente, esto es, como se realiza el procesamiento de la información. En el caso de estar solamente interesados, en las entradas y salidas de los procesos, es suficiente muchas veces el verificar estas entradas y salidas sin tener en cuenta exactamente cómo se leen y como se generan.

La importancia de este punto radica en la decisión de adoptar el enfoque de "caja blanca" (vigilando el comportamiento interno de los componentes) lo cual implicará una carga adicional de trabajo debido al número adicional de casos de prueba.

Metodología de las pruebas.

En este punto es importante determinar si existe un procedimiento de pruebas dentro de la organización o si es necesario elaborarlo y en que medida este procedimiento está integrado con el resto de los sistemas de desarrollo y mantenimiento. Gracias a esta información, es posible estimar el esfuerzo adicional requerido para la construcción de estos procedimientos, con el objeto de realizar las pruebas.

Se deberá intentar localizar los procedimientos de pruebas de uso general que puedan ser de utilidad.

Puntos de control y aprobaciones.

Se deben especificar los puntos de control en el transcurso de la construcción y ejecución de las pruebas, tal como el determinar los puestos de las personas que tendrán que autorizar la continuación de las pruebas acorde con el plan original o asumiendo las variaciones incorporadas al mismo.

Criterios de suspensión y conclusión de las pruebas.

Estos criterios se refieren a la suspensión o terminación de la ejecución de los casos de prueba cuando son necesarios otros componentes que no tienen listos ó cuando el número de los defectos encontrados sobrepasan el límite de los esperados para lo cual es necesario regresar a la etapa de desarrollo y verificar las especificaciones.

Recursos Humanos.

Identificar las personas específicas a ocupar los puestos en la organización de pruebas. Esta información es de utilidad ya que al asociar personas a los puestos se descubren los factores que les hace falta para el efectivo cumplimiento de sus responsabilidades.

Preparación de pruebas.

Es necesario realizar todas las actividades que aseguren la disponibilidad de la infraestructura de pruebas acorde con las categorías de pruebas a ejecutar. Los criterios a tener en cuenta para decidir que tipo de plan necesitamos son los siguientes:

1. Tamaño y complejidad del Proyecto.
2. Lista de funciones, requerimientos, condiciones generales tanto funcionales como estructurales.
3. Arquitectura de la aplicación.
4. Condiciones y restricciones.
5. Ambiente de pruebas.

Criterios para iniciar la ejecución de las pruebas.- Se deben especificar los eventos de carácter general a acontecer con el objeto de iniciar la ejecución del plan de pruebas.

Preparación y verificación del ambiente de pruebas.- Especificar las actividades más relevantes para la preparación del ambiente de pruebas.

Una vez especificadas las actividades es posible identificar los procedimientos de pruebas necesarios (existentes y los que requieran ser creados).

A partir de los requerimientos funcionales y estructurales se deben generar los casos de pruebas.

Casos de pruebas.

Los casos de pruebas son las condiciones de prueba ejecutables, que incluye un conjunto de datos de entrada, resultados esperados y resultados obtenidos.

Los casos de prueba contemplados en la ejecución de estas se toman a partir de los requerimientos funcionales y estructurales con el siguiente criterio:

- Datos dentro de especificaciones.
- Datos fuera de especificaciones.

Procedimientos para las pruebas.

Así como la metodología de pruebas, en caso de no existir un procedimiento dentro de la organización para la realización de las pruebas, se hace necesario elaborar los pasos a seguir para terminar esta fase con óptimos resultados.

En la figura 3.4.1. se enlistan los elementos para llevar a cabo las pruebas de "integración del sistema", las cuales se descubrirán más adelante.

Las categorías de las pruebas se deducen directamente de la lista de requerimientos funcionales y estructurales del plan de pruebas. La especificación de las categorías de pruebas a usar es importante ya que basados en ellas podemos determinar la infraestructura de pruebas requerida. Las categorías de las mismas pueden ser: Unitarias, de Integración, de Regresión, de Concurrencia, de Volumen, de Aceptación, del Sistema, Manuales o Automáticas.

Categoría de Pruebas	Consideraciones para la ejecución de pruebas	Infraestructura de pruebas	Recursos Humanos
Unitarias	Riesgos y suposiciones	Ambiente de pruebas	Preparación de las pruebas
Integración	Condiciones y restricciones	Organización de las pruebas	Casos de Pruebas
Regresión	Cobertura funcional de las pruebas	Metodología de las pruebas	Procedimientos de las Pruebas
Volumen	Descripción de la arquitectura del sistema	Herramientas de Pruebas	
Aceptación del Usuario		Puntos de control y aprobaciones	
Caja Blanca		Criterios de suspensión y conclusión de las pruebas	
Caja Negra			
Estáticas			
Funcionales			
Estructurales			

Figura 3.4.1. Elementos para llevar a cabo las Pruebas de Integración del Sistema.

Los fundamentos de las pruebas definen los objetivos esenciales para las pruebas del software.

Es en esta fase donde el ingeniero intentará demoler todo el software que ha construido en las etapas de análisis y diseño.

Las pruebas requieren que se descarten las ideas acerca de lo correcto que es el software desarrollado y que al descubrir los errores, se logre superar cualquier conflicto en el sistema.

Las pruebas nos permiten:

- Sentar las bases para determinar los objetivos y un plan específico de pruebas.
- Asegurar la obtención y formalización de los requerimientos del usuario y verificar que son adquiridos de una manera completa, correcta y consistente.
- Verificar los requerimientos funcionales, así como los estructurales y establecerlos como fundamento para realizar las pruebas del sistema.
- Detectar y registrar defectos asociados a los requerimientos establecidos.
- Establecer la documentación de reportes para pruebas realizadas.

Pruebas Unitarias.

Categoría de pruebas realizadas sobre un programa o módulo con la intención de encontrar problemas funcionales en la lógica, así como problemas técnicos en el código. La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software "el modulo". Usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes, con el fin de descubrir errores dentro del ámbito del módulo. La complejidad relativa de las pruebas y de los errores descubiertos esta limitada por el alcance estricto establecido por la prueba de unidad. La prueba de unidad siempre esta orientada a la caja blanca y este paso se puede llevar a cabo en paralelo para múltiples módulos.

Para cada una de las unidades se probaron los siguientes aspectos:

- Se revisó la presentación de la pantalla, es decir, el tamaño de las fuentes y de los controles, la ortografía, el color, etc.
- Se probó que todos los controles que tienen asociado algún evento reaccionen adecuadamente, por medio del teclado y/o el ratón.

Pantalla de Acceso.

En la pantalla de acceso se verifico que en el momento de errar en la captura de la clave del usuario y de la contraseña, el sistema mostrara un mensaje como el que se muestra en la figura 3.4.2.

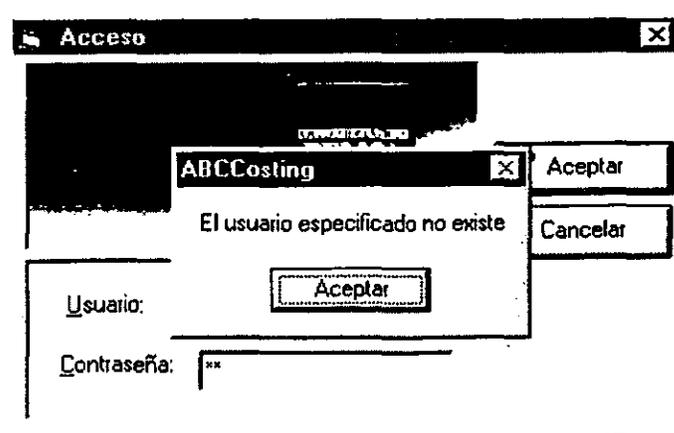


Figura 3.4.2. Pantalla que muestra el acceso denegado al sistema.

Pantalla principal.

- Se comprobó que la pantalla presentara todas las opciones que podrán manejar dentro del sistema.
- Se probó que aparecieran todas las opciones del menú principal.

MENÚ PRINCIPAL

1. SESIÓN
 - 1.1 SALIR
2. CONFIGURACIÓN
 - 2.1 ARCHIVOS DE CARGA
 - 2.2 CREACIÓN DE PERÍODOS
 - 2.3 DEPURACIÓN
3. CATÁLOGOS
 - 3.1 CANALES
 - 3.2 CLIENTE AGRUPADO
 - 3.3 EMPAQUE
 - 3.4 CATÁLOGOS TIPO (MARCA, TIPOS DE VENTA, ZONA, ACTIVIDAD)
 - 3.5 PRODUCTO
 - 3.6 RUTA
 - 3.7 UNIDAD OPERATIVA
 - 3.8 MATERIA PRIMA
 - 3.9 CONSUMO TEÓRICO
 - 3.10 PUESTO
4. REPORTES
 - 4.1 BONIFICACIÓN
 - 4.2 "OROS"
5. CONSULTAS
 - 5.1 CONSULTAS DE ASIGNACIÓN DE ACTIVIDADES A PUESTOS, ASIGNACIÓN TIPO DE VENTA A RUTAS Y CLIENTES POR CANAL
 - 5.2 CONSULTAS DE CAJAS FÍSICAS X DISTRIBUIDORA X CANAL X TIPOVENTA, INGRESO CAJAS FÍSICAS POR SKU Y NÚMERO DE RUTAS
 - 5.3 CANAL_UNIDADOPERATIVA
 - 5.4 REPORTE VENTAS SKU
 - 5.5 CONSULTAS DE CUADRE DE INFORMACIÓN
 - 5.6 VENTAS X STATUS
 - 5.7 VENTAS POR TIPOVENTA
 - 5.8 EQUIPOS DATAMIRROR, PRODUCTOS NUEVOS DATAMIRROR
6. CARGA Y ACUMULADOS
 - 6.1 CARGA DE INFORMACIÓN
 - 6.2 GENERACIÓN DE ACUMULADOS
 - 6.3 PRORRATEO MATERIA PRIMA
 - 6.4 DATAMIRROR
7. ADMINISTRACIÓN
 - 7.1 BITÁCORA
 - 7.2 CONTROL MONITOR
8. SEGURIDAD
 - 8.1 USUARIOS
 - 8.2 PERMISOS

Pantalla de configuración de archivos de carga.

Para la opción de configuración de archivos de carga se comprobó que cada una de las opciones que tiene la pantalla funcionara correctamente, por ejemplo para el caso de modificación o creación, se verifico que se almacenaran los datos introducidos por el usuario, como la nueva ruta para un modulo, si se van a hacer cargas automáticas o no, y en caso que se realicen cargas automáticas que se indique la calendarización de estas

en la parte inferior de la pantalla etiquetada como diario, esta pantalla se puede observar en la figura 3.4.4.

Pantalla de depuración de la información.

Para esta pantalla se verifica que aparezca un mensaje que indica que la información a sido depurada después de que el usuario ha introducido correctamente los datos.

The image shows a Windows-style dialog box titled "Configuración de Archivos - (Nuevo)". It has a standard title bar with a close button (X). The dialog contains several input fields and controls:

- Modulo:** A dropdown menu.
- Ruta:** A text input field with a browse button (...).
- Fuente de Datos:** Two radio buttons: "Texto" and "DataMirrow".
- Carga Automática:** Two radio buttons: "Activado" and "Desactivado".
- Hora:** A text input field containing "12:00".
- Fecha det.:** Two date input fields separated by "a:", each containing " / /".
- Diario:** A group box containing seven checkboxes for the days of the week: "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", and "Domingo".
- Buttons:** "Aceptar" (with a checkmark icon) and "Cancelar" (with an X icon) are located on the right side.

Figura 3.4.4. Pantalla que permite configurar una nueva ruta.

Catálogos.

Para cada catálogo se aplicaron pruebas muy similares, ya que como es un módulo de mantenimiento de catálogos, sus funciones son parecidas en cada opción. Las pruebas principalmente se basaron en probar la correcta realización de las altas, bajas y cambios en la base de datos.

Reportes y Consultas.

Las pruebas realizadas a los módulos de reportes y consultas, fue principalmente verificar que la información que se pedía era realmente la que se obtenía, y también que cuando se pedía información que no estuviera registrada apareciera el mensaje de "No se encontraron registros".

Carga de información.

Se verifica para este módulo que la información sea cargada correctamente en la base de datos de las rutas dadas por los procedimientos almacenados de la base de datos. También se verifica que conforme los archivos sean cargados aparezca un mensaje que indica que el archivo se cargo exitosamente y al finalizar la carga de todos los archivos que aparezca el mensaje de "Proceso terminado".

Pruebas de Integración.

Categoría de pruebas realizadas a un grupo de programas para asegurar que los datos y controles sean pasados adecuadamente entre controles. La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la iteración, el objetivo es tomar los módulos probados en unidad y construir una estructura de programas que este de acuerdo con lo que dicta el diseño.

Pruebas de Regresión.

Categorías de pruebas selectivas para detectar fallas que se hayan introducido durante las modificaciones a un sistema o componente, que permiten verificar que estas modificaciones no impacten en forma negativa y que se siga cumpliendo con los requerimientos planteados.

Pruebas de Volumen.

Categoría de pruebas realizadas para verificar el comportamiento adecuado y eficiente de una aplicación bajo condiciones de volumen (número de operaciones), competencia de recursos (conurrencia) y carga máxima (velocidad de petición de ejecución de una operación) así como el comportamiento eficiente bajo las condiciones de volumen máximo (cantidad de datos) en las aplicaciones.

Pruebas de Caja Blanca.

Pruebas basadas en el conocimiento sobre la lógica y estructura internas. Usualmente dirigidas a la lógica. Es un método diseñado para crear casos para prueba de software, aprovechando el control estructurado del diseño procedural de la aplicación. Utilizando los métodos de prueba de caja blanca, el ingeniero de software puede construir ejemplos para pruebas que garanticen que todas las rutas independientes de un módulo sean probadas al menos una vez, probar todas las decisiones lógicas en sus casos **TRUE** (VERDADERO) y **FALSE** (FALSO), probar todos los **loops** (ciclos) y sus fronteras dentro de operaciones definidas y probar la estructura de datos interna para asegurar su utilidad. A las pruebas de caja blanca también se les conocen como pruebas estructurales, porque utilizan las estructuras internas de un programa para derivar los casos de estudio.

Ya que se obtuvieron ciertos resultados para cada una de las unidades, construimos algunos ejemplos que nos dieron una visión más clara de qué es lo que deberíamos esperar de las entradas/salidas de cada una de estas unidades. Así para todas las opciones del módulo de Catálogos, por ejemplo, en la opción "Cliente agrupado", al capturar todos los datos que se requieren al dar de alta para un nuevo cliente agrupado se confirmó la correcta inserción del mismo en la base de datos, es decir la correcta validación de los datos de entrada para generar su correspondiente salida; caso contrario que si no se capturan todos los datos, pues no se podrá realizar la inserción de ese cliente agrupado, dando como resultado de la prueba, la correcta validación de los datos de entrada para que los de salida no sean generados inconsistentemente.

Pruebas de Caja Negra.

Pruebas funcionales basadas en los requerimientos sin conocimiento sobre cómo fue construido el sistema y usualmente dirigidas a los datos. Habilita al ingeniero de software para derivar un conjunto de condiciones de entradas y que serán ejecutadas para todos los requerimientos funcionales de un programa.

Estas pruebas son complementarias a las técnicas de la caja blanca, ya que cubren diferentes clases de errores. A diferencia de las pruebas de caja blanca, que se ejecutan al inicio del proceso de pruebas; las de caja negra son aplicadas durante las etapas finales de las pruebas.

Las pruebas de caja negra también se conocen como pruebas funcionales, pues se basan en la definición de que, un programa debe realizar eficientemente las especificaciones definidas, sin importar la estructura de como lo hace.

Para este tipo de pruebas construimos ejemplos que nos ayudaron para la correcta integración de todas las unidades. Fuimos enlazando unidad por unidad para confirmar que nuestro diseño cumplía con las funciones planteadas. La integración de la información fue una parte importante a considerar ya que la funcionalidad, de una unidad a otra, depende en gran medida de esta integridad.

Pruebas de Aceptación del Usuario.

Categoría de pruebas finales, ejecutadas por el usuario, para asegurar que el sistema satisfaga las necesidades de la organización o usuario final (validan que el sistema construido es el correcto).

Es virtualmente imposible para un desarrollador de software imaginar como el cliente realmente utilizará un programa. Instrucciones para utilizar el software pueden ser mal interpretadas; extrañas combinaciones de datos pueden ser regularmente usadas; la salida que parece clara para el desarrollador puede ser no entendible para un usuario. Cuando el software del cliente se construye, una serie de pruebas aceptables son conducidas para habilitar al cliente y validar todos sus requerimientos. Conducido por el usuario final, más que por el desarrollador del sistema, una prueba aceptable puede cambiar de una informal prueba de funcionamiento a planear y ejecutar sistemáticamente una serie de pruebas. En efecto, las pruebas de aceptación pueden ser conducidas por un período de semanas o meses, identificando errores que significan retrasos de la liberación del sistema.

Primera Prueba.

En un ambiente diseñado especialmente para esta prueba, se trató de recrear lo más posible la información cotidiana que procesan los usuarios, para que estos no fuesen influenciados adversamente. La presencia y participación de los analistas y desarrolladores del sistema fue evidente. La prueba generó el punto de vista de ambas partes para ser encaminada a una complementación que mejoró algunos aspectos del sistema.

Segunda Prueba.

Esta prueba nos arrojó resultados interesantes ya que fue en la que el usuario planteó casos que muchas veces no nos esperábamos. Aunque la retroalimentación entre los usuarios y nosotros no fue inmediata (ya que no nos encontrábamos en ese lugar) la generación de los resultados de la prueba hicieron evidentes algunos detalles que desembocaron en modificación de ciertas partes de la codificación. Uno de estos detalles fue la escasez de mensajes de error que orienten a los usuarios para una correcta operación del sistema.

Pruebas Estáticas.

Consiste en la revisión y validación de los documentos generados en las distintas fases de la vida de un proyecto. Verificación realizada sin ejecutar el código del sistema.

Pruebas Funcionales.

Validan los requerimientos de la organización (lo que se supone que el sistema debe hacer), pretenden descubrir errores cometidos en la implantación de dichos requerimientos.

Para nuestro sistema se revisaron los diferentes módulos que lo integran, observando su buen funcionamiento, a continuación se enlistan estos, describiendo lo que deben realizar cada uno:

Configuración .- Permite dar de alta las rutas y los periodos para los archivos de carga, así como también depurar la información contenida en las tablas de la base de datos, en caso de haberse generado un error en la carga de información.

Catálogos .- Realizan consultas a la base de datos, dependiendo del sub menú seleccionado.

Reportes .- Ejecuta Funciones (Store Procedures) , para extraer la información contenida en la base de datos, correspondiente al reporte seleccionado.

Consultas .- Realiza consultas de información específica extraída de la base de datos.

Carga y Acumulados .- Carga de información manual, que no fue programada en la configuración, genera los acumulados y el prorrateo.

Administración .- Guarda los errores y los procesos realizados dentro del sistema, además se registran aquellos archivos que han sido cargados, acumulados y en su caso prorrateados.

Seguridad .- Da de alta a los usuarios autorizados para el uso del sistema, asignando los permisos definidos para cada uno de ellos.

Pruebas Estructurales.

Validan la arquitectura del sistema confirmando que todas sus partes funcionen sincronizadamente y que la tecnología esta siendo usada apropiadamente. Se refieren a las características técnicas, como su comportamiento con grandes volúmenes de información, tiempos de respuesta, etc.

3.5. GENERACIÓN DE REPORTES PARA LA TOMA DE DECISIONES.

Todos los reportes se generan basándose en consultas realizadas a la Base de Datos por medio de llamadas del Front End que desencadenan los procedimientos dentro del manejador de Base de Datos, todos los procesos que finalizan en la obtención de relaciones de estado sobre las tablas de trabajo se ejecutan en el Back End.

La información seleccionada se muestra en una pantalla de resultados, ésta contiene una interfaz programada para importar los reportes a formatos de Excel, de tal manera que es la única forma de obtener listados impresos sobre el resultado de los diferentes procesos es la exportación de registros a Hojas de Calculo de un Libro de Excel.

Los reportes que se precisan para la toma de decisiones se agrupan en los siguientes conjuntos:

- **Distribución.**
 - **Plantas.**
 - **Corporativo.**
 - **Costo Capital.**

Distribución.

Esta subdivisión contempla información relacionada con las Plantas de Distribución, el volumen de productos y de que tipo se colocan en las rutas determinadas por los canales adecuados. Cada planta se encuentra enmarcada dentro de una Unidad Operativa de esta forma se puede medir el tráfico que maneja una región en particular.

Los reportes que se generan a partir de esta opción se realizan sobre la base de los periodos de producción que con anterioridad se configuraron.

Plantas.

Al igual que las correspondientes a las encargadas de distribución, estos generan información relacionada sobre los volúmenes de productos que se elaboran en las plantas por Unidad Operativa.

Corporativo y Costo Per cápita.

Los reportes corporativos junto con los de Costo Per cápita son los que representan el mayor interés por parte de la empresa ya que son el material básico para áreas vitales el Administrativo, Recursos Humanos, Finanzas y otros.

Estos consisten en una serie de consultas, sobre la Base de Datos para obtener los registros de los siguientes rubros: nominas de plantas y distribuidoras, clientes, cartera de

clientes, cuentas por pagar, pasivos, volúmenes de producción, existencias sobre inventarios y almacenes, etc.

Los reportes se construyen dentro del sistema de la siguiente manera. Dentro de la opción reportes cada elemento de los submenús tiene asociado una instrucción SQL en el Back End desencadenada desde el Front End. El Back End construye la información requerida delimitando el periodo de interés seleccionado. Ver Fig. 3.5.1

Una vez realizado lo anterior, se establece una conexión adicional para la tabla de paso construida y alimentada con la información generada por la llamada de ejecución del procedimiento SQL y se llena el objeto interfaz para visualizar el resultado en pantalla, y se realiza la exportación del reporte a Hojas de Calculo de Excel.

Los formatos con que se exportan estos reportes son planos es decir se exporta directamente un vista de la tabla de trabajo, esto con la finalidad de ser objeto de análisis posterior por parte de la empresa con una utilidad denominada cubo de decisiones. Esta herramienta junto con los listados de soporte permiten a la empresa planear y diseñar sus nuevos periodos de producción.

```
-----/*****Object: -Stored Procedure: dbo.spS_Cuadre_Ventas_DATAMIRROR Script
Date: 4/11/99 10:16:29 AM CREATE PROCEDURE
spS_Cuadre_Ventas_DATAMIRROR
    (@UnidadOper int,
     @FechaIni smalldatetime,
     @FechaFin smalldatetime)
/*-----*/
/*Objetivos: StoreProcedure para Borrar de la tabla */
/*de ConsTeorico */
/* Stored de donde son llamados: */
/*frmCuadreProduccion */
/*-----*/
AS
```

Figura 3.5.1.El script (Línea de código del Back End).

El formato es una vista directa de las tablas, encabezados y valores que se pasan tal cual, de los archivos de origen.

En la parte superior derecha se visualiza el botón exportar, que es el que desencadena la exportación de la vista del Grid (Control de Visual Basic) directamente a una hoja de cálculo de un Libro de Excel.

Dentro de esta pantalla se tiene la opción de exportar la información visualizada a un libro de Excel.

Para generar consultas de acumulados se procede de la siguiente manera:

Del menú consultas se hace válida alguna de sus opciones, ver figura 3.5.5.

Carga y Acumulados .- Carga de información manual, que no fue programada en la configuración, genera los acumulados y el prorrateo.

Administración .- Guarda los errores y los procesos realizados dentro del sistema, además se registran aquellos archivos que han sido cargados, acumulados y en su caso prorrateados.

Seguridad .- Da de alta a los usuarios autorizados para el uso del sistema, asignando los permisos definidos para cada uno de ellos.

Pruebas Estructurales.

Validan la arquitectura del sistema confirmando que todas sus partes funcionen sincronizadamente y que la tecnología esta siendo usada apropiadamente. Se refieren a las características técnicas, como su comportamiento con grandes volúmenes de información, tiempos de respuesta, etc.

3.5. GENERACIÓN DE REPORTES PARA LA TOMA DE DECISIONES.

Todos los reportes se generan basándose en consultas realizadas a la Base de Datos por medio de llamadas del Front End que desencadenan los procedimientos dentro del manejador de Base de Datos, todos los procesos que finalizan en la obtención de relaciones de estado sobre las tablas de trabajo se ejecutan en el Back End.

La información seleccionada se muestra en una pantalla de resultados, ésta contiene una interfaz programada para importar los reportes a formatos de Excel, de tal manera que es la única forma de obtener listados impresos sobre el resultado de los diferentes procesos es la exportación de registros a Hojas de Calculo de un Libro de Excel.

Los reportes que se precisan para la toma de decisiones se agrupan en los siguientes conjuntos:

- **Distribución.**
 - **Plantas.**
 - **Corporativo.**
 - **Costo Capital.**

Distribución.

Esta subdivisión contempla información relacionada con las Plantas de Distribución, el volumen de productos y de que tipo se colocan en las rutas determinadas por los canales adecuados. Cada planta se encuentra enmárcada dentro de una Unidad Operativa de esta forma se puede medir el tráfico que maneja una región en particular.

Los reportes que se generan a partir de esta opción se realizan sobre la base de los periodos de producción que con anterioridad se configuraron.

Plantas.

Al igual que las correspondientes a las encargadas de distribución, estos generan información relacionada sobre los volúmenes de productos que se elaboran en las plantas por Unidad Operativa.

Corporativo y Costo Per cápita.

Los reportes corporativos junto con los de Costo Per cápita son los que representan el mayor interés por parte de la empresa ya que son el material básico para áreas vitales el Administrativo, Recursos Humanos, Finanzas y otros.

Estos consisten en una serie de consultas, sobre la Base de Datos para obtener los registros de los siguientes rubros: nominas de plantas y distribuidoras, clientes, cartera de

clientes, cuentas por pagar, pasivos, volúmenes de producción, existencias sobre inventarios y almacenes, etc.

Los reportes se construyen dentro del sistema de la siguiente manera. Dentro de la opción reportes cada elemento de los submenús tiene asociado una instrucción SQL en el Back End desencadenada desde el Front End. El Back End construye la información requerida delimitando el periodo de interés seleccionado. Ver Fig. 3.5.1

Una vez realizado lo anterior, se establece una conexión adicional para la tabla de paso construida y alimentada con la información generada por la llamada de ejecución del procedimiento SQL y se llena el objeto interfaz para visualizar el resultado en pantalla, y se realiza la exportación del reporte a Hojas de Calculo de Excel.

Los formatos con que se exportan estos reportes son planos es decir se exporta directamente un vista de la tabla de trabajo, esto con la finalidad de ser objeto de análisis posterior por parte de la empresa con una utilería denominada cubo de decisiones. Esta herramienta junto con los listados de soporte permiten a la empresa planear y diseñar sus nuevos periodos de producción.

```
/****** Object: Stored Procedure dbo.spS_Cuadre_Ventas_DATAMIRROR Script
Date: 4/11/99 10:16:29 AM CREATE PROCEDURE
spS_Cuadre_Ventas_DATAMIRROR
    (@UnidadOper int,
     @Fechaini smalldatetime,
     @Fechafin smalldatetime)
/*-----*/
/*Objetivos: StoreProcedure para Borrar de la tabla */
/*de ConsTeorico */
/* Stored de donde son llamados: */
/*frmCuadreProduccion */
/*-----*/
AS
```

Figura 3.5.1.El script (Línea de código del Back End).

El formato es una vista directa de las tablas, encabezados y valores que se pasan tal cual, de los archivos de origen.

En la parte superior derecha se visualiza el botón exportar, que es el que desencadena la exportación de la vista del Grid (Control de Visual Basic) directamente a una hoja de calculo de un Libro de Excel.

Dentro de esta pantalla se tiene la opción de exportar la información visualizada a un libro de Excel.

Para generar consultas de acumulados se procede de la siguiente manera:

Del menú consultas se hace válida alguna de sus opciones, ver figura 3.5.5.

Microsoft Excel - empaques

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

H7 =

	A	B	C	D	E	
1	emp_iCve	emp_sDescrip	emp_sTipo	emp_tBotellaxCaja	emp_MiliBotella	emp
2		9 1 LT RET 14	RE		14	1000
3		23 2 LTS NR 6	NR		6	2000

Figura 3.5.11. Empaques.

Microsoft Excel - Productos

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

A1 = pto_iCve

	A	B	C	D	E	F	G
	pto_iCve	pto_iCveAcum	emp_iCve	pto_sDescrip	pto_iCajasxPai	pto_fFacCubico	pto_fFacCajasF
2	1	1	1	PLASTICO PET 12 DIV	1	1	1
3	2	2	2	PLASTICO PET 24 DIV	1	1	1
4	3	3	1	PLASTICO TAM. GRANDE	1	1	1
5	4	4	1	PLASTICO TAM. NORMAL	1	1	1
6	6	6	1	PLASTICO PET 6 CAV.	1	1	1
7	7	7	1	PLASTICO REF/PET 12	1	1	1
8	8	8	1	PLASTICO PET 8 CAV.	1	1	1
9	9	9	1	PLASTICO PET 8 CAV.	1	1	1
10	10	10	1	CANASTILLA 24 600 ML	1	1	1
11	110	110	1	COCA-COLA 6.5 ONZAS	42	0.029	1
12	120	120	5	COCA-COLA 12 ONZAS	36	0.034	1
13	130	130	7	COCA-COLA 26 OZ	24	0.039	1
14	140	140	17	FANTA N.J.A. 1/2 LTO	30	0.042	1
15	141	140	0	FANTA MANZANA 1/2 LT	0	0	1
16	145	145	1	ENVASE FANTA 1/2	1	1	1
17	150	150	17	COCA-COLA 1/2 LTO	30	0.042	1
18	155	155	1	ENVASE COCA-COLA 1/2	1	1	1
19	160	163	0	DIET COKE 12 OZ	0	0	1
20	161	161	1	C-C LIGHT 6.5 ONZAS	42	0.029	1
21	162	162	1	ENVASE C-C LIGHT 6.5 OZ	1	1	1
22	163	163	5	COCA-COLA LIGHT 12OZ	36	0.034	1

Figura 3.5.12. Productos.

Microsoft Excel - Ventas

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

F5 = 170

	A	B	C	D	E	F	G	H	I	
	año_iCve	mes_iCve	die_iCve	tu_iCve	caa_iCve	pto_iCve	avt_iCajasFis	avt_fImporteNav	avt_iCajasFis	avt_fI
1	1999					110	383	15,386.60		14
2	1999				1	1101	120	18577	749,568.84	372
3	1999				1	1101	163	68	3,695.65	0
4	1999				1	1101	170	439.5	23,865.86	0
5	1999				1	1101	180	491	22,201.74	20
6	1999				1	1101	190	9829	534,184.53	0
7	1999				1	1101	200	8	434.78	0
8	1999				1	1101	205	208	11,304.34	0
9	1999				1	1101	240	437	23,749.99	0
10	1999				1	1101	250	270.5	14,701.08	0
11	1999				1	1101	270	329	17,680.43	0
12	1999				1	1101	280	955	51,902.15	0
13	1999				1	1101	305	86.5	4,701.08	0
14	1999				1	1101	355	6880.5	440,687.18	0
15	1999				1	1101	356	2761	176,658.15	0
16	1999				1	1101	357	2780.5	177,091.65	0
17	1999				1	1101	358	1839	117,538.01	0
18	1999				1	1101	359	2952	193,208.03	0
19	1999				1	1101	360	65714	4,273,680.28	1
20	1999				1	1101	361	5205	333,329.37	0
21	1999				1	1101	363	6558	358,952.75	302
22	1999				1	1101	365	96089	5,180,448.26	3810
23	1999				1	1101	367	1896	102,219.05	64
24	1999				1	1101	368	5685	306,495.41	127
25	1999				1	1101	369	20000	20000.00	0

Figura 3.5.13. Ventas.

	A	B	C	D	E	F	G	H
1	0008786100009999	401	2417546	506 LA CASITA ARGENTINA			CTO PTE #10 L18	
2	0400194100009999	402	1104054	0542HOGAR CONSUMO			5 DE FEBRERO #2	54160
3	0400725100009999	40					PR	
4	0400773100009999	405	2416999	058 YOLANDA SALAZAR			AV INDUSTRIA S/N	54075
5	3200019100009999	408	2599349	349 BENMATIC NORTE, S A DE C V			HDA STA ANA Y LOBO #62	
6	3200020100009999	408	2730349	349 MOTORES MC MILLAN, S A DE			AV EDO DE MEXICO NO 4	
7	3200024100009999	408	2730349	349 DRESLAN SA DE CV			KM 27 5 CARR MEX-QRO	5494
8	3200099100009999	408	2731348	348 AUTOMOTORES DE SATELITE, S A			SOR JUANA INES CRUZ NO	
9	3200144100009999	406	2730348	348 FOSECO, S.A. DE C.V.			AV 16 DE SEPTIEMBRE NO. 100 5	
10	3200216100009999	408	2730347	347 EMPAQUES PLASTICOS INDUSTRIA-			BLVD. TOLUCA NO 18-A	
11	3200219100009999	408	2730348	348 SERV CONDESA DE ZACATECAS			AV STA. ROSA NO 1	
12	3200241100009999	408	2730348	348 DISTRIBUIDORA COMERCIAL DICO,			RECURSOS PETROLEROS N	
13	3200392100009999	408	2753348	348 TUM TRANSPORTITAS UNIDOS			EUCALIPTO NO 2	540
14	3200526020002100	406	1205349	349 COMERCIAL MEX VALLE DORADO			ATENAS NO 6	54
15	3200562100009999	408	2730347	347 ASTRA CHAMICAL, S.A. DE C.V			AV URBINA NO. 15	530
16	3200681100009999	408	2730347	347 ESPEC EN ESTERILIZ Y ENV.			GUILLERMO BARROSO NO 11-A	
17	3200700100009999	408	2730349	349 REFACCIONARIA ROGELIO, S.A.			DE FULTON NO. 17	541
18	3200780100009999	408	2731347	347 AUTOMOTRIZ NAUCALPAN, S.A.			DE HACIENDA DE ECHEGARAY	
19	3200806100009999	408	2730347	347 TERCASA, S.A. DE C V			AZTECAS NO 27	5315036
20	3200842100009999	406	2730349	349 FRANCISCO BAUTISTA			FILIBERTO GOMEZ NO 279	541

Figura 3.5:14. Clientes.

3.6. FACTIBILIDAD TÉCNICA Y OPERATIVA.

Dentro de los elementos viables que tenemos en este sistema, es sin duda la centralización de la información, es decir la definición y creación de una estructura de datos organizada, y normalizada, esto lo llevamos a cabo con la implementación de una base de datos orientada a objetos, de donde aplicamos las normas y procedimientos que están definidos para la realización de este tipo de bases de datos.

La centralización de la información permite de alguna manera tener mejor control de la misma, es decir, podemos contar con procedimientos de validación, de acceso, y políticas de respaldos, otra de las ventajas que podemos tener es la importancia de cuidar la integridad y evitar la duplicidad de la información, y con ello la disminución de recursos de cómputo y sin duda tiempo de proceso, para la obtención de resultados.

Utilizando las herramientas, podemos tener un mejor manejo y explotación de la información, tal como: consultas, reportes, estadísticas, etc., contar con esta flexibilidad, es sin duda un factor muy importante, ya que uno de los problemas con que contaba el área es en la generación de reportes de interés a ciertas entidades, y esto implicaba mucho tiempo, por falta de un sistema centralizado.

Con la concentración de la información e integración de un sistema, esto facilita las condiciones para que se integre a la red de datos, y por ende, todas las áreas involucradas en la compra de materiales, para tener la facilidad de acceder a la información vía red, y de esta manera realizar consultas en línea de la misma, que es una de las necesidades que se tenía en el área, la importancia de contar con un buen sistema de información.

Con la centralización de los diferentes procesos en un solo equipo, la integración de este a la red de datos y la implementación del sistema, se puede prescindir del uso de varios elementos que se involucraban en el proceso anterior, tales como: Equipo de cómputo, Impresoras, Software, Personal especializado, uso de diferentes formatos, etc., y esto para el área representa una gran ayuda, ya que se puede tener mejores tiempos de respuesta, información confiable y oportuna para las áreas involucradas (compras, Administrativas, Usuarios, etc.).

La parte más importante y crítica de una red la constituye el servidor. La función del servidor es la de compartir sus recursos a la red, ya sean estos servicios de disco, impresión o aplicaciones cliente/servidor. También debe proporcionar una plataforma robusta para la ejecución de aplicaciones de misión crítica.

Por todo lo anterior, se hace necesario que la computadora destinada a ser servidor, cumpla con requisitos específicos para realizar de manera eficiente todas las funciones anteriormente mencionadas. Estos incluyen lo concerniente a sus requerimientos mínimos de

hardware y software, sus requerimientos eléctricos y su ubicación física, la figura 3.6.2 muestra parte del equipo para el cual se requiere un lugar especialmente acondicionado.

Con el fin de ajustarse a diferentes requerimientos de uso y aplicación se han definido dos tipos de servidores: el servidor departamental y el servidor corporativo.

El servidor departamental esta enfocado a la ejecución de los servicios básicos de red (archivos, impresión, software institucional) esta proyectado para dar servicio hasta a 200 usuarios.

Por otra parte el servidor corporativo esta enfocado no solamente a la ejecución de los servicios básicos de red, sino también a la ejecución de aplicaciones de misión crítica (Servidores SQL, aplicaciones cliente/servidor etc.) y para redes con mas de 200 usuarios.

Los requerimientos mínimos de hardware para el servidor departamental son:

- Servidor Pentium III(500 Mhz)
- Memoria RAM ECC de 128 MB
- Unidad de CD-ROM 8x (SCSI) -interna – booteable
- Unidad de respaldo 24 GB (SCSI) –interna-
- Disco duro de 9 GB(SCSI)
- Drive interno de 3.5"
- 3 bahías disponibles para crecimiento futuro
- Tarjeta de red PCI (Operación Dual 10 BASET/100BASETX)
- 2 puertos seriales y 1 puerto paralelo
- Mouse (bus)
- Monitor SVGA color 15"
- UPS con interface al servidor (1200 Watts)

Los requerimientos mínimos de hardware para el servidor corporativo son:

- Servidor Dual Pentium III(700 Mhz)
- Memoria RAM ECC de 512 MB
- Unidad de CD-ROM 8x (SCSI) -interna – booteable
- Unidad de respaldo 24 GB (SCSI) –interna-
- RAID con 3 discos duros Hot-Swap de 9 GB Hot-Swap
- Drive interno de 3.5"
- 3 bahías Hot-Swap disponibles para crecimiento futuro
- 3 ranuras de expansión libres PCI
- Tarjeta de red PCI (Operación Dual 10 BASET/100BASETX)
- 2 puertos seriales y 1 puerto paralelo
- Mouse (bus)
- Monitor SVGA color 15"
- UPS con interface al servidor (1200 Watts)

Es importante señalar que las marcas y modelos a adquirir (CPU, UPS, tarjeta de red) estén registrados en el HCL(Hardware Compatability List) de Windows NT Server.

Requerimientos de SOFTWARE del servidor.

Los requerimientos mínimos de software son:

- Windows NT Server v4.x-English-
- Windows NT Service Pack 3 –English-

Requerimientos eléctricos del servidor.

El servidor debe estar alimentado por un UPS, y a su vez contar con una interfaz a esta para protegerlo de forma automática en caso de falla en el suministro eléctrico. Gracias a esta interface, es posible apagar de forma segura el servidor, evitándose así pérdida de información y posibles daños al equipo.

Ubicación física del servidor.

El servidor debe ser colocado en un área exclusiva y libre del paso (de preferencia en un lugar cerrado, y con aire acondicionado) , con el fin de que pueda ser operado sin dificultad cuando sea necesario.

Requerimientos de HARDWARE de las estaciones de trabajo:

La estación de trabajo es la computadora donde el usuario ejecuta aplicaciones y desde donde accesa y utiliza los recursos del servidor.

Para ejecutar todo el software institucional de manera rápida y eficiente las estaciones de trabajo deben de cumplir con ciertos requerimientos mínimos de hardware y software.

Procesador PENTIUM II de 450MHz
Memoria RAM de 64 MB
Disco Duro de 8 GB
Tarjeta de red a 10 MGB Ethernet
Monitor SVGA de 14 "
Memoria de vídeo de 4MB
Mouse (bus)
2 puertos seriales y 1 puerto paralelo

Requerimientos de SOFTWARE de las estaciones de trabajo.

Los requerimientos mínimos de software son:

Windows NT Workstation 4.x (de preferencia) ó
Windows 95
Visual Basic Ver 5.0.

Así mismo deberán utilizar la configuración de red en modo protegido (32 bits).

Impresoras.

Aquellas impresoras que sean conectadas directamente a la red(impresoras con tarjeta de red o a través de una interfaz externa) deberán soportar (y por lo tanto configurarse) por el protocolo TCP/IP. Las impresoras y los respectivos adaptadores de red tienen que estar en el HCL de Windows NT y Windows 95.

Puesta a punto.

Una vez que se tienen listos el servidor, las estaciones de trabajo e impresoras con sus respectivas tarjetas de red, configuradas e intercomunicadas a través del cableado, se debe realizar la puesta a punto, la cual comprende la instalación y configuración inicial del servidor.

Con el fin de simplificar, optimizar y homogeneizar los servidores Windows NT de la empresa, se han normalizado los siguientes puntos.

- a) Instalación
- b) Nombre de servidor y rol asignado
- c) Nombre de las estaciones de trabajo y grupos de trabajo
- d) Nombres de cuentas de usuarios
- e) Nombre de recursos
- f) Dominios
- g) Configuración TCP/IP
- h) Configuración IIS

- i) Servicios WINS/DNS/DHCP
- j) Servicio FPNW
- k) Directorio de aplicaciones (APPS)
- l) Directorio de grupos(GROUPS)
- m) Tareas y responsabilidades del administrador
- n) Configuración de las estaciones de trabajo

a) Instalación

Deberá consultarse antes de la instalación, todas los puntos contenidos en esta sección a fin de hacer la adecuada selección de los parámetros de instalación. Los siguientes son los primeros parámetros que deberán ajustarse:

Directorio de instalación Windows NT	C: WINNT
Modo de Licenciamiento	Per Seat

Se requiere que la partición del sistema (C:) tenga al menos 1 GB de espacio.

Recuerde, antes de proceder con la instalación, deberá contar con un nombre y rol asignados, la configuración TCP/IP correcta, deberá conocer así mismo los servicios a utilizar y asegurarse que el servidor este apropiadamente conectado a un nodo operacional.

b) Nombre del servidor y rol asignado

El servidor debe contar con un nombre único que lo identifique y evite problemas de comunicación con otras redes.

Un servidor Windows NT puede configurarse bajo tres esquemas diferentes de seguridad: como PDC(Primary Domain Controller), BDC (Backup Domain Controller) y como Server(Standalone Server). Este esquema de seguridad es conocido como Dominio. El nombre del servidor, así como el rol asignado deberá solicitarlo y registrarlo a la Gerencia de Comunicaciones en Sistemas.

Para servidores que pertenezcan al dominio de la empresa y que trabajan bajo el esquema de seguridad del Server, el nombre se formará de acuerdo a las siguientes reglas:

SRVDE <nombre-asignado> <consecutivo>

Donde:

SRVDE Para indicar que es un nodo tipo servidor dentro del dominio de la empresa

<nombre asignado> Corresponden al nombre asignado (siglas, ubicación física, proyecto etc.) Este deberá tener un máximo de 8 caracteres. Deberá solicitarlo a la Gerencia de comunicaciones.

<Consecutivo> El consecutivo estará en función del número de servidores que existan en la misma Empresa empezando por el número 1.

c) Nombre de las estaciones de trabajo y grupos de trabajo

El nombre asignado a las estaciones de trabajo (nombres NETBIOS) debe ser igual al login-name del usuario. Si algún usuario tiene asignado más de un equipo se deberá utilizar números consecutivos empezando por el número dos.

Ejemplos:

MGONZ
RHERR
LRAMIR

La configuración del grupo de trabajo deberá ser igual al de sus siglas organizativas, mínimos a nivel Subdirección (siglas correspondientes a su área).

d) Nombre de cuentas de usuarios (login-name)

El nombre del usuario (login-name) se formará con un máximo de 8 caracteres (todos ellos deben ser caracteres ASCII de 7 bits, esto significa que no deberán utilizarse caracteres como la ñ o letras acentuadas). EL primero de ellos corresponderá a la primer letra del primer nombre del usuario, los subsecuentes caracteres, corresponderán a su apellido paterno hasta completar un máximo de 8. Si se diera el caso de un login-name repetido se tomara entonces como segundo carácter la primer letra de su primer apellido paterno y los restantes corresponderán a su apellido materno hasta completar un máximo de 8 caracteres.

Solo en caso de que existan login-names repetido aun después de aplicar la regla anterior, se deberá aplicar la siguiente nomenclatura: El apellido paterno hasta un máximo de 7 caracteres y la primera letra de su primer nombre.

Ejemplos

Nombre del Usuario:	Login Name:
Rene Herrera Martínez	RHERRERA
Noe Navarrete López	NNAVARRE
Martha González Vega	MGONZALE
Víctor Aguilar Meza	VAGUILAR
Rosa Sánchez Ortega	RORTEGA
Miguel Segura López	MSEGURA

Tabla 3.6.1. Ejemplo de Login-names.

La asignación de la cuentas de usuario (login-names) definitivos para el dominio de la empresa lo asignará el Área de Comunicaciones. Aquí se asignan de tal forma que sean únicos, no solo a nivel servidor, sino a nivel Corporativo.

e) Nombre de Recursos

Un recurso es cualquier parte de una computadora interconectada a la red como una impresora, disco duro, unidad de CD-ROM que puede ser compartido.

Ahora bien, con el fin de asegurar el acceso homologado a ciertos recursos de los servidores Windows NT, se deberán utilizar los nombres que a continuación se señalan:

U1CDROM	Primera unidad de CD-ROM
U2CDROM	Segunda unidad de CD-ROM
TODOS	Directorios de acceso común (\usuarios\todos)

Tabla 3.6.2 Nombres de recursos reservados.

Para el caso de los nombres de recursos asignados a impresoras la terminología se deja abierta, sin embargo se recomienda utilizar la siguiente nomenclatura: <marca><modelo><tecnología><ubicación física>.

Ejemplos:

Hplj4laser-of1513
Hpdj602ink-of1402

f) Dominios

En Windows NT, al conjunto de servidores que comparten una base de datos de usuarios común se le llama dominio. Un dominio provee acceso centralizado a las cuentas de usuarios y de grupos. Todo esto es controlado por un administrador de dominio. Cada dominio debe tener un nombre único.

Ahora bien, el configurar servidores NT bajo un mismo dominio es una poderosa herramienta que facilita la administración del sistema, ya que posibilita "ver" a un conjunto de servidores NT como si fueran un sistema único.

En una seguridad basada en dominios un servidor NT puede jugar tres papeles: como PDC (Primary Domain Controller), BDC (Backup Domain Controller) y como Server (Standalone Server). Solo puede haber un PDC en un dominio.

En la empresa se sigue un esquema de seguridad llamado Master Domain Model, el que básicamente consiste en tener un dominio de cuentas (Domain Account) y dominios de recursos (Resources Domains).

Existe solo un dominio maestro (Account Domain) en el cual están contenidos las cuentas de todos los usuarios (a nivel corporativo).

g) Configuración TCP/IP

Uno de los puntos clave en la arquitectura bajo la cual se diseñó la red de servidores corporativos utilizando Windows NT Server, es la capacidad de este sistema de trabajar todos sus servicios a través del protocolo TCP/IP. Esto está estrechamente relacionado con los proyectos y esfuerzos orientados a construir una red WAN uniprotocolo, lo cual representa incrementar sustancialmente la velocidad de la red, reducir los tiempos de administración de esta al simplificar enormemente los parámetros de operación y configuración, para lograr así una red más robusta, segura y eficiente.

Por tal motivo, solo deberá habilitar el protocolo TCP/IP en el servidor. Otros protocolos como IPX/SPX, NETBEUI, DLC no deberán instalarse.

h) Configuración IIS

El IIS (Internet Information Server) es un servicio que permite publicar páginas WEB en la red. De hecho ofrece la plataforma ideal para crear la INTRANET de la empresa, al ofrecer los servicios de WWW (World Wide Web) y FTP (File Transfer Protocol) a los servidores Windows NT Server.

Deberá consultar la Normatividad para Intranet emitida por el Área de Desarrollo de Software. Esta Normatividad tiene como propósito establecer las políticas, normas y recomendaciones a seguir para la implantación de Intranet's en la compañía.

Los directorios de los servicios INTRANET deben ser los siguientes (son los directorios por omisión).

\\netpub\ftproot	(Home-Directory para los servicios de FTP)
\\netpub\gophroot	
\\netpub\wwwroot	

Así mismo es necesario crear un directorio virtual (Virtual Directory) llamado todos que apunte al directorio \\USUARIOS\\TODOS y con el fin de hacerlo visible al software FTP, se deberá crear el directorio \\netpub\ftproot\todos.

i) Servicios WINS/DNS/DHCP

El servicio de DHCP (Dynamic Host Configuration Protocol), ofrece la posibilidad de configurar automáticamente todos los parámetros TCP/IP en las estaciones de trabajo (Windows NT Workstation y Windows 95) DHCP provee una manera segura, confiable y simple para configurar TCP/IP en la red a la vez que ayuda a prevenir conflictos de

direcciones, ayudando así a conservar el uso de las direcciones IP y posibilitando la administración centralizada de esta importante función.

WINS (Windows Internet Name Service) y DNS (Domain Name Service) son los servicios de resolución de nombres. Estos se utilizan para resolver nombres NETBIOS (los nombres utilizados por los servidores y estaciones de trabajo) a direcciones IP.

La información de los servidores WINS/DNS registrados, así como de los servidores DHCP y los segmentos habilitados con este servicio los podrá consultar en el sitio WEB del Área de Comunicaciones.

j) Servicio FPNW

El FPNW (File and Print Services for Netware) es el servicio que permite operar a un servidor Windows NT como un servidor Novell Netware. En la primera revisión esta normatividad se enunciaba la necesidad de instalar este servicio a fin de facilitar los procesos de migración de plataforma, ya que permite el acceso a servicios de disco e impresión a los usuarios de DOS y Windows v3.x a través de la tradicional configuración de servidores Novell Netware.

Sin embargo, bajo esta nueva normatividad **NO DEBERA INSTALARSE** este servicio, ya que en este momento se están realizando esfuerzos por detener el incremento de servicios IPX/SPX, y la eliminación gradual del enrutamiento de este protocolo en la red.

k) Directorio de aplicaciones (APPS)

En la primera revisión de esta normatividad, se pretendía que todos los servidores tuvieran instaladas la mayor parte del software institucional (office, Windows 95, Antivirus, Utilerías, etc.) bajo una estructura de directorios estandarizada con el fin de ofrecer a los usuarios de estos servidores un acceso rápido y transparente al software.

Ahora en base a las nuevas políticas para el acceso al software institucional, surgidas de las recomendaciones de las auditorías realizadas en la empresa, la instalación de este software será exclusivamente a través de los servidores administrados por el área de Distribución de Software.

Por tal motivo los servidores **NO** deberán ser utilizados para la instalación de software institucional y el directorio asignado para este propósito (APPS) no se requiere. Deberá obtener del área de distribución de software los procedimientos para el acceso a estos servidores, así como las cuentas y passwords autorizados.

Finalmente, en base a la nueva filosofía de trabajo en red, el software institucional debe instalarse completo en el disco duro del usuario, **NO** como instalación de red. Los beneficios en términos de menor tráfico en la red y mayor tolerancia a fallas compensan el consumo de espacio utilizado en el disco de los usuarios.

l) Directorio de grupos (USUARIOS)

Este constituye el directorio de datos y aplicaciones específicas. A fin de uniformar los nombres de subdirectorios bajo USUARIOS(USUARIOS) se deberán seguir las siguientes reglas en la asignación de nombres de grupo.

Al menos se deberá crear el directorio \USUARIOS\TODOS en el que todos los usuarios tengan acceso, esto con el fin de que al menos exista un lugar donde cualquier usuario pueda compartir información con otro. Será necesario informar a los usuarios que no existe ningún compromiso de confidencialidad y permanencia de información sobre este directorio y que será completamente depurado en cuanto se requiera. Es importante mencionar que además este directorio puede ser accesado a través de FTP, para así posibilitar a los usuarios UNIX o bien a cualquiera que cuente con un servicio FTP acceder y grabar información en este directorio.

Los nombres de los subdirectorios de grupos deberán llamarse igual a los grupos para los cuales se tiene acceso. Así mismo se recomienda crear un recurso compartido por cada uno de estos directorios.

Si se tiene privilegios de solo lectura, para un mismo directorio, se deberá de añadir el sufijo -RO al nombre de estos grupos.

m) Tareas y responsabilidades del administrador

Cada área deberá de nombrar a su administrador, el cual se encargara de todas las tareas relacionadas con la administración y mantenimiento del servidor.

El administrador deberá asumir las siguientes responsabilidades:

- Administración de usuario y grupo (altas/bajas/cambios)
- Administrar la seguridad del servidor (asignación de privilegios y passwords)
- Administrar los servicios de archivos e impresión.
- Respaldo de la información.
- Soporte técnico de primer nivel a sus usuarios.
- Instalación y soporte del software institucional a sus usuarios.
- Asegurar el cumplimiento de las normatividades de Sistemas.

Para cualquier problema relacionado con los puntos anteriores el administrador puede solicitar asesoría y apoyo al área de Sistemas que le asesora o bien solicitar soporte a través de HELP DESK. Para cualquier problema relacionado con la comunicación con otras redes, el supervisor deberá comunicarse a la Gerencia de Comunicaciones. Para ello se deberán proporcionar a esta todos los datos del supervisor, con el fin de coordinar con él la solución a cualquier problema existente.

n) Configuración de las estaciones de trabajo

Como se menciona anteriormente los clientes que se recomiendan para trabajar en redes NT son Windows NT Workstation y Windows 95 o 98. Es importante que estos sistemas se configuren con el software de red apropiado, es decir, que se utilicen drivers de red en modo protegido (32bits) específicamente diseñados para el sistema operativo que se trate.

Todos estos clientes deben configurarse con el stack de protocolo TCP/IP y de preferencia configurarse a través de los servicios de DHCP (siempre que sea posible). Así mismo, para asegurar la correcta resolución de nombres en la red se deberán configurar los servidores primarios y secundarios de WINS y DNS.

Capacitación de los usuarios y administrador del sistema.

Se ha determinado brindar la capacitación necesaria para los usuarios, por lo cual se contemplan los siguientes puntos:

- Problemática en los procesos que se desarrollan actualmente
- Solución y/o sistema propuesto
- Ventajas que proporciona el Sistema
- Funcionamiento del Sistema (manual de usuario)

El administrador del sistema deberá tomar la capacitación que se les brinda a los usuarios del sistema.

Al administrador y a dos personas más de las áreas involucradas al proceso se les brindará la siguiente capacitación.

- Características y ventajas principales de Visual Basic
- Desarrollo e implementación del sistema
- Administración y mantenimiento del programa
- Manual técnico

Descripción de Roles:

Coordinadores de Proyecto	Más de 8 años de experiencia coordinando proyectos de desarrollo de software en múltiples tecnologías. Amplias habilidades de comunicación y coordinación de personal. Amplio conocimiento de las tecnologías necesarias para el proyecto. Dominio de la metodología RUP (Rational Unified Process) y de la notación UML (Unified Modeling Language).
Consultor Tecnológico	Personal experto en aplicación de tecnología Microsoft. Participan como asesores de tiempo parcial en todos nuestros proyectos. Dominio de la metodología RUP y de la notación UML (Unified Modeling Language).
Líderes de Proyecto Sr.	Más de 3 años de experiencia coordinando proyectos de desarrollo de software en múltiples tecnologías. Amplias habilidades de comunicación y coordinación de personal. Amplio conocimiento de las tecnologías necesarias para el proyecto. Dominio de la metodología RUP y de la notación UML (Unified Modeling Language). Una certificación Microsoft en la mayoría de los casos.
Consultores SQA	Personal con amplia experiencia en control de proceso de desarrollo de software, incluyendo levantamiento de requerimientos, inspecciones de diseño, inspecciones de código, planes de pruebas, y planes de implantación. Expertos en la metodología RUP y en la notación UML (Unified Modeling Language).
Líderes de Proyecto	Más de 1 año de experiencia coordinando proyectos de desarrollo de software en múltiples tecnologías. Amplias habilidades de comunicación y coordinación de personal. Amplio conocimiento de las tecnologías necesarias para el proyecto. Conocimiento de la metodología RUP y de la notación UML (Unified Modeling Language). Una certificación Microsoft en algunos casos.
Analistas	Más de 3 años de experiencia analizando y desarrollando software en múltiples tecnologías. Buenas habilidades de comunicación y coordinación de personal. Conocimiento de las tecnologías necesarias para el proyecto. Conocimiento de la metodología RUP y de la notación UML (Unified Modeling Language). Una certificación Microsoft en algunos casos.
Programadores Senior	Más de 3 años de experiencia desarrollando software en múltiples tecnologías. Buenas habilidades de comunicación y coordinación de personal. Conocimiento de las tecnologías necesarias para el proyecto. Conocimiento de la metodología RUP y de la notación UML (Unified Modeling Language). Una certificación Microsoft en algunos casos. Productividad alta.
Programadores	Más de 1 año de experiencia desarrollando software en múltiples tecnologías. Conocimiento de las tecnologías necesarias para el proyecto. Conocimiento básico de la metodología RUP y de la notación UML (Unified Modeling Language). Productividad media.
Diseño Gráfico	Más de 2 años de experiencia como diseñador gráfico. Conocimiento amplio de herramientas de diseño gráfico por computadora.
Testers	Seis meses de experiencia ejecutando planes de prueba de sistemas de computo. Capacitados en procedimientos de prueba y reporte de errores de software.

Equipo de Trabajo.

Fase	Recursos
Elaboración	1 Consultor TEC. 1 Consultor SQA. 1 Analista. 1 Programador, 1 Diseñador Gráfico.
Construcción	1 Consultor TEC. 1 Consultor SQA. 1 Analista. 1 Programador, 1 Tester
Transición	1 Consultor TEC. 1 Consultor SQA. 1 Analista. 1 Programador, 1 Tester

Costo del Proyecto.

Los precios son en dólares americanos.

Fase	Personal	Duración	Horas	Costo	Total
Elaboración	1 Consultor SQA	4 Días	32	\$69	\$2,208
	1 Consultor MCSD	2 Días	16	\$84	\$1,344
	1 Líder de Proyecto	4 semanas	160	\$60	\$9,600
	1 Analista	4 semanas	160	\$48	\$7,680
Subtotal		4 semanas			\$20,832
Construcción	1 Consultor SQA	3 Días	24	\$69	\$1,656
	1 Consultor MCSD	5 Días	40	\$84	\$3,360
	1 Líder de Proyecto	8 semanas	320	\$60	\$19,200
	1 Analista	8 semanas	320	\$48	\$15,360
	1 Programador	8 semanas	320	\$37	\$11,840
	1 Tester	7 semanas	210	\$33	\$6,930
Subtotal		8 semanas			\$58,346
Transición	1 Consultor SQA	2 Días	16	\$69	\$1,104
	1 Consultor MCSD	2 Días	16	\$84	\$1,344
	1 Líder proyecto	2 semanas	80	\$60	\$4,800
	1 Programador	2 semanas	80	\$37	\$2,960
	1 Tester	2 semanas	90	\$33	\$2,970
Subtotal		2 semanas			\$13,178
Total		14 semanas			\$92,356

Calendario de Pagos.

- Un anticipo del 50% del costo total al inicio del proyecto.
- Un pago del 50% contra la aceptación y terminación del proyecto.

Cronograma del Proyecto.

ID	Task Name	Duration	Start	Finish	Resource Name	Dec	Jan	February	March	Apr
1	Proyecto	73 days	20/12/1999	31/03/2000	CSQA					
2	Elaboración	19.5 days	20/12/1999	18/01/2000	CSQA					
3	Visualización	8.5 days	20/12/1999	3/01/2000	CSQA					
4	Análisis profundo de la operativa	1.5 days	20/12/1999	21/12/1999	A1,D1					
5	Análisis de los requerimientos de las áreas	3 days	21/12/1999	27/12/1999	A1,D1					
6	Análisis de la plataforma del usuario	4 days	27/12/1999	3/01/2000	A1,D1					
7	Planeación	11 days	3/01/2000	18/01/2000	CSQA					
8	Diseño de la plataforma de red	3 days	3/01/2000	6/01/2000	CT1,A1					
9	Diseño de la base de datos	2 days	6/01/2000	10/01/2000	CT1,A1					
10	Diseño de diagramas del sistema	2 days	10/01/2000	12/01/2000	A1,D1					
11	Sesión de retroalimentación con el usuario	2 days	12/01/2000	14/01/2000	A1,D1					
12	Adecuaciones al diseño de la aplicación	1 day	14/01/2000	17/01/2000	A1,D1					
13	Elaboración de especificaciones técnicas	1 day	17/01/2000	18/01/2000	A1,P1					
14	Construcción	41 days	20/01/2000	16/03/2000	CSQA					
15	Plataforma	19 days	20/01/2000	15/02/2000	CT1					
16	Implementación de la red	4 days	3/02/2000	8/02/2000	CT1					
17	Implementación de la base de datos	5 days	9/02/2000	15/02/2000	CT1					
18	Construcción de la funcionalidad del sistema	10 days	20/01/2000	2/02/2000	A1,P1					
19	Pruebas	15 days	3/02/2000	23/02/2000	A1					
20	Pruebas Internas y de pruebas de usuario	8 days	3/02/2000	14/02/2000	A1,P1					
21	Modificaciones pedidas por el usuario	5 days	15/02/2000	21/02/2000	A1,P1					
22	Pruebas de usuario	2 days	22/02/2000	23/02/2000	USU					
23	Elaboración de manuales	10 days	24/02/2000	8/03/2000	P1					
24	Elaboración del manual del usuario y técnico	8 days	24/02/2000	6/03/2000	P1					
25	Elaboración de guía de solución a problemas	2 days	7/03/2000	8/03/2000	A1					
26	Capacitación	6 days	9/03/2000	16/03/2000	CSQA					
27	Capacitación al usuario de oficina central y cap.	6 days	9/03/2000	16/03/2000	A1					
28	Transición	11 days	17/03/2000	31/03/2000	CSQA					
29	Carga inicial de datos	3 days	17/03/2000	21/03/2000	CT1,USU					
30	Capacitación al usuario de los estados	3 days	17/03/2000	21/03/2000	A1					
31	Puesta en producción del sistema	1 day	20/03/2000	20/03/2000	CT1,A1					
32	Apoyo post-arranque	8 days	20/03/2000	29/03/2000	A1					
33	Firma de carta de terminación y aceptación	11 days	17/03/2000	31/03/2000	CSQA					

CONCLUSIONES.

El apoyo de los sistemas computacionales como herramientas de trabajo permite optimizar procesos de administración y dar eficiencia a los procedimientos de operación de cualquier compañía, siempre y cuando se tenga un buen diseño y análisis de la problemática y los requerimientos de la misma.

Al realizar el trabajo de tesis se logró plantear una solución a un problema específico y bien delimitado; para esto se tuvieron que realizar entre otras, las siguientes actividades:

- **Recolección de datos y selección de fuentes de información.**
- **Diseño de la aplicación:** consiste en definir las bases de datos, procesos, entradas y salidas de la aplicación en forma detallada.
- **Selección de herramientas de desarrollo:** dada la gran variedad de herramientas disponibles para implementar la aplicación, se hizo un estudio de cuáles son las más adecuadas. Seleccionando así la base de datos (SQL Server), el front end (Visual Basic 6), así como el ambiente de desarrollo.
- **Implementación y prueba de la aplicación:** consiste en escribir el código fuente y hacer las pruebas correspondientes
- **Escritura de la tesis o reporte**

Para poder desarrollar un buen proyecto se deben conocer las herramientas: el sistema operativo de la máquina, manejo de bibliotecas de programas, uso de generadores de aplicaciones, manejo de editores, utilerías, uso de depuradores etc. Se debe conocer el ambiente en que se desarrolla el programa, ya que usándolo de la manera en que fue previsto se puede hacer el trabajo en menor tiempo del planeado. Afirmando que el uso de herramientas es singularmente importante para el desarrollo de sistemas.

El trabajo de Tesis fue enfocado hacia la Tecnología Orientada a Objetos que nos trae grandes beneficios. Ya que nos permite escribir y diseñar mejores sistemas, y más modulares. Ayudando también a los usuarios, ya que obtienen programas mucho más fáciles de usar.

Lo que pasa con la Programación Orientada a Objetos es que no necesariamente se desarrollan Sistemas de Información con mayor velocidad. Simplemente, se tiene una mejor formación de nuestra programación, lo que nos permite escribir mejores programas.

Además, otro objetivo de la OOP es crear programas altamente reutilizables, lo que ha llevado a los programadores a pensar en el mejoramiento y reutilización de componentes que ayudaran a construir mejores y nuevos programas.

La gran desventaja de estas nuevas tecnologías es que es muy difícil que sean asimiladas. Lo que la programación Orientada a Objetos promete es aumentar el grado de modularización de un sistema, lo que es un requisito indispensable para dominar la complejidad de los nuevos programas.

Pero nosotros al haber utilizado como Front End, Visual Basic 6.0 nos limitó y no nos permitió un total acercamiento a la Programación Orientada a Objetos ya que este no cuenta con un lenguaje realmente Orientado a Objetos, eligiéndose esta herramienta por estar al alcance del cliente y sus necesidades.

Lo malo es que la programación Orientada a Objetos no es una solución total. Es relativamente fácil encontrar aún ejemplos donde la Programación Orientada a Objetos no es la solución. Lo que sucede es que la complejidad de los problemas es cada vez mayor, y esto exige cada vez mejores herramientas para resolverlos.

Se puede concluir del sistema, que cumplió con las necesidades planteadas desde el inicio del análisis. Se cubre la necesidad de aprovechar una información básica, proveniente de una plataforma anterior y que su remoción implica una inversión mayor de la planteada en la evaluación del proyecto; Se estandarizó el formato de entrada y se realizó la interfaz deseada

para importar los reportes de salida a una aplicación del tipo Hoja de Cálculo, con la finalidad de obtener una manipulación final.

Se creó un instrumento de monitoreo para realizar el seguimiento adecuado de las rutas de distribución, permitiendo el acceso en línea de clientes, productos y ventas por sector, apoyando sobremanera la administración de los territorios enmarcados en el área de actividad de la empresa.

El desarrollo del sistema planteó una problemática adicional, la necesidad de utilizar procesos básicos para la empresa, y que motivaron el diseño de una estrategia especial para incluir estos puntos como parte de los requerimientos del sistema, y posteriormente dentro de la propuesta de solución.

El sistema empleado por el Palacio de Minería en sus seminarios de Titulación, emplea una dinámica de grupo muy exitosa desde el punto de vista de los fines que se persiguen, permite acelerar el proceso de Titulación hasta el punto de la presentación del examen profesional, y dentro de esta dinámica los participantes encuentran un espacio de intercambio de experiencias profesionales, fomentándose la integración de la comunidad universitaria.

La elaboración de sistemas en la plataforma SQL SERVER ofrece beneficios a las empresas en el desarrollo de proyectos, lo suficientemente robustos para dar la solución en corto y mediano plazo, los cuales se ven reflejados en los costos.

Las aportaciones realizadas son la realización del análisis, diseño y construcción de un sistema aplicando todos los conocimientos adquiridos durante el transcurso de la carrera y los adquiridos con la experiencia profesional, para dar solución a un problema real de una empresa.

El presente trabajo proporciona una panorámica general en cuanto al diseño y desarrollo de sistemas utilizando el concepto de diseño orientado a objetos que tiene por objetivo la creación de sistemas fáciles de utilizar, portables y de sencilla modificación, sin que esto refleje una disminución de la calidad. Incluyendo conceptos teóricos y conjugándolos con los aspectos prácticos de manera general, intentando dar al lector un material que le sirva como base para la realización de otros sistemas.

Dentro de la problemática de desarrollo es importante observar lo complicado que resulta el poder manejar la información de tantas personas, por lo que se tuvo que cuidar que los datos devueltos por el sistema fueran los adecuados y que estos mostrarán la claridad necesaria para su posterior entendimiento. Es importante mencionar que en el desarrollo de todo sistema debe de existir una participación más completa por parte de todos los que de alguna u otra manera participan en el sistema y debe ser visto como algo que más que traer un perjuicio va a realizarse como beneficio de la empresa y por consiguiente de todos cuantos laboran ahí.

Uno de los puntos más importantes para la elaboración de un sistema es la comunicación que se tiene entre los miembros del equipo de trabajo, así como el ambiente creado por el mismo grupo, ya que se pudo constatar que un equipo bien acoplado asegura que los resultados obtenidos se obtengan con mayor eficiencia y rapidez, y entonces el equipo sea por consiguiente más eficaz. Esto resulta vitalmente importante, si se considera que la formación actual de cualquier ingeniero independientemente del ámbito laboral del que se trate se realiza bajo una formación de trabajo de grupo, y si ya desde ahora se logra tener una buena adaptación y entendimiento con la gente los resultados se verían reflejados no sólo en la calidad, sino también en un buen ambiente de trabajo, beneficiando de esta forma a la empresa en general y por consiguiente a uno mismo.

BIBLIOGRAFÍA.

SISTEMA PARA LA CLASIFICACION DE INSUMOS DE UN ALMACEN CENTRAL.
BUENDIA PEREZ ILDEFONSO ALFREDO.
MEXICO, 1998.

OPERATIONS RESEARCH IN PRODUCTION PLANNING, SCHEDULING AND INVENTORY CONTROL.
JOHNSON, LYNWOD A.
ED. JOHN WILEY & SONS
NEW YORK, 1974.

SISTEMAS DE PRODUCCION E INVENTARIO, PLANEACION Y CONTROL
ELWOOD S. BUFFA Y WIÑÑIAM H. TAUBERT.
ED. LIMUSA
MEXICO 1997.

CONTROL DE CALIDAD
DALE H. BESTERFIELD
CUARTA EDICIÓN
PRENTICE MAY

EL LIBRO DE WINDOWS NT.
HELEN CUSTER.
MCGRAWHILL.

VISUAL BASIC
PROGRAMACION CLIENTE SERVIDOR
ALFONS GONZÁLEZ
ED. ALFAOMEGA

APRENDIENDO VISUAL BASIC EN 24 HORAS
GREG PERRY
PRENTICE HALL, MEXICO 1998.

MICROSOFT VISUAL BASIC 6.0, INICIACION Y REFERENCIA.
LUIS JOYANES AGUILAR Y ANTONIO MUÑOZ CLEMENTE.
ED. MC GRAW HILL.
MADRID, 1999.

FUNDAMENTOS DE BASES DE DATOS.
KORTH, HENRY F. Y ABRAHAM SILBERSCHATZ
MC. GRAW HILL

INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS.
DATE, C. JADDISON
WESLEY IBEROAMERICANA.

MICROSOFT SQL SERVER 7.0
PERFORMANCE TUNING TECHNICAL REFERENCE
STEVE ADIM DE LUCA
ED. MICROSOFT PRESS, 2000.

Apuntes de :
INSTITUTO POLITECNICO NACIONAL
SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERIA CIENCIAS SOCIALES Y
ADMINISTRATIVAS

SISTEMAS DE CONTABILIDAD
ALEJANDRO PRIETO
SÉPTIMA EDICIÓN
EDITORIAL BANCA Y COMERCIO

PRINCIPIOS DE CONTABILIDAD GENERALMENTE ACEPTADOS
INSTITUTO MEXICANO DE CONTADORES PUBLICOS A.C.
DECIMOCUARTA EDICIÓN

COMUNICACIONES Y REDES DE PROCESAMIENTO DE DATOS
NESTOR GONZALEZ SAINZ
PRIMERA EDICIÓN
McGRAW-HILL

INGENIERIA DE SOFTWARE, UN ENFOQUE PRACTICO.
ROGER, S. PRESSMAN.
MC GRAW HILL

TCP/IP. ARQUITECTURA, PROTOCOLOS E IMPLEMENTACIÓN CON IPV6 Y SEGURIDAD IP.
DR. SIDNIE FEIT.
MC GRAW HILL, 1998, ESPAÑA.

ANALISIS ESTRUCTURADO MODERNO.
EDWARD YOURDON.
PRENTICE HALL IBEROAMERICANA, S.A.

ANALISIS Y DISEÑO ORIENTADO A PROCESOS PARA LA INTEGRACION DE SISTEMAS DE
INFORMACION
MIRELES BELMONTE JUAN ALONSO.
MEXICO, 1998.

MODELADO Y DISEÑO ORIENTADO A OBJETOS.
JAMES RUMBAUGH.
ED. PRENTICE HALL.
MADRID, 1996.

DIRECCIONES DE INTERNET:

<http://www.microsoft.com>
<http://msdn.microsoft.com/vbasic/>
<http://msdn.microsoft.es/>
<http://www.oracle.com/>
<http://www.delphi-training.com/>
<http://www.microsoft.com/Mexico/SQL/>
<http://microsoft.com/latam/sql/default.asp>

Apéndice A.

DIAGRAMAS ENTIDAD-RELACIÓN.

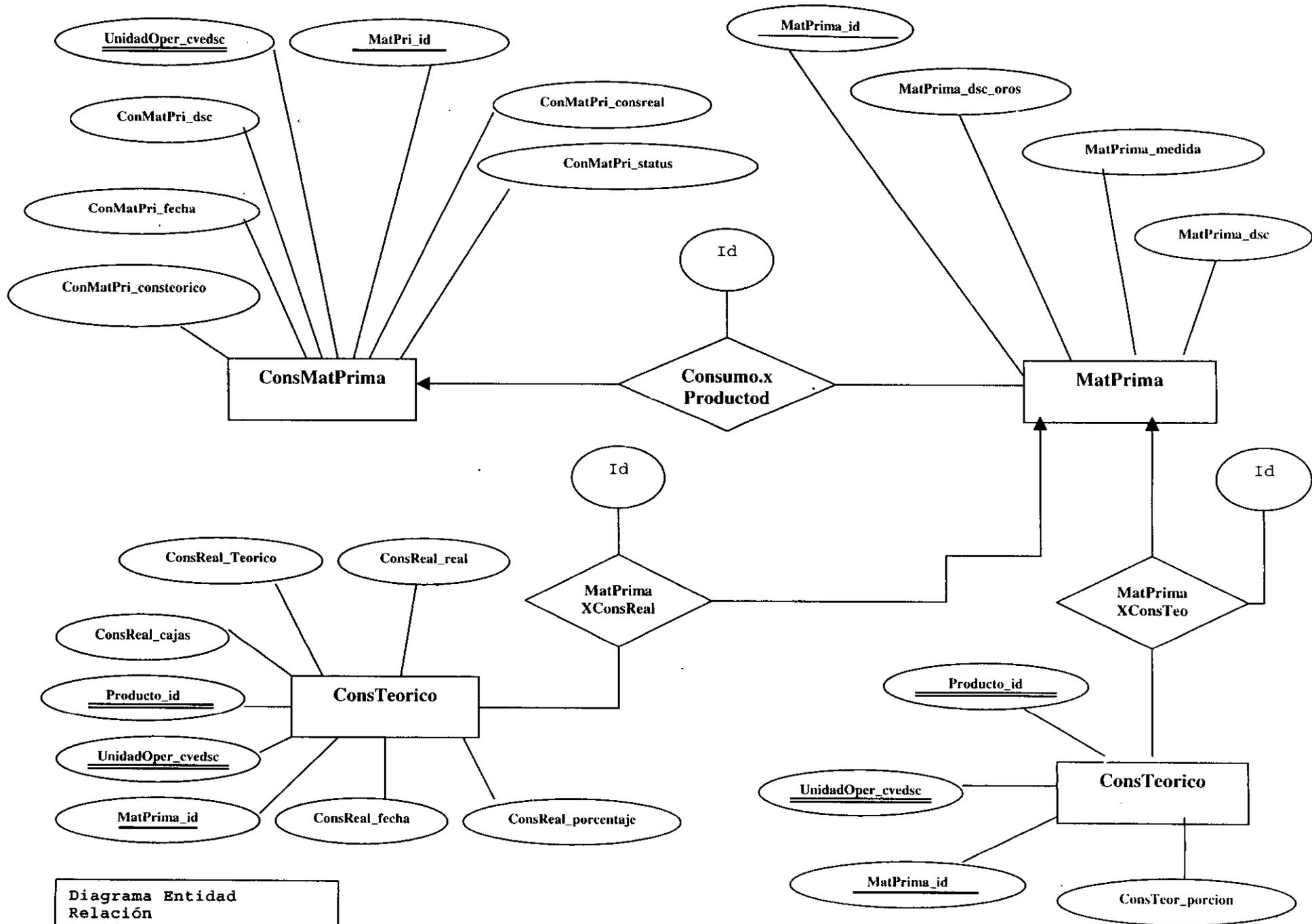


Diagrama Entidad Relación Entidad Producto

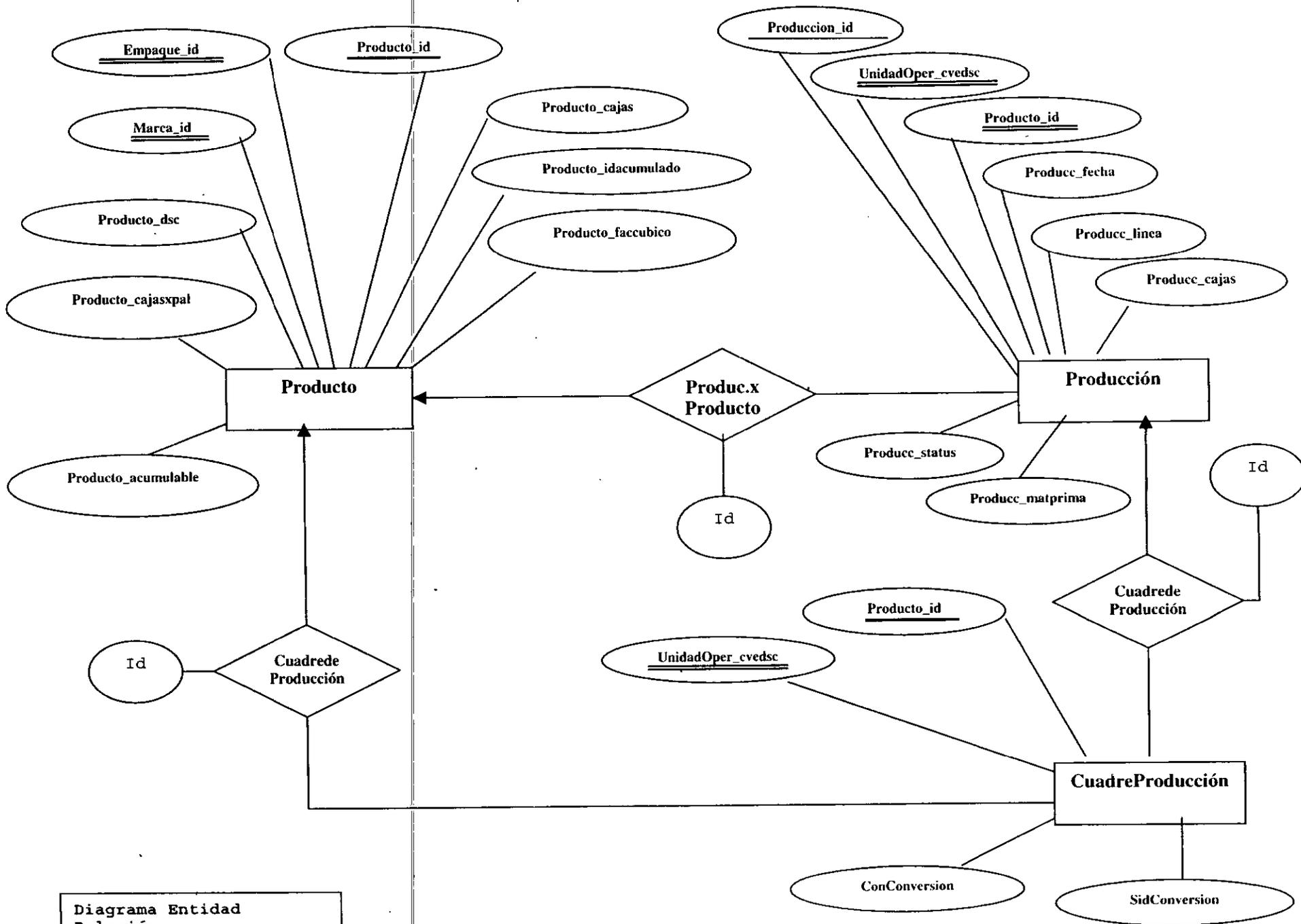


Diagrama Entidad
Relación
Entidad Producto

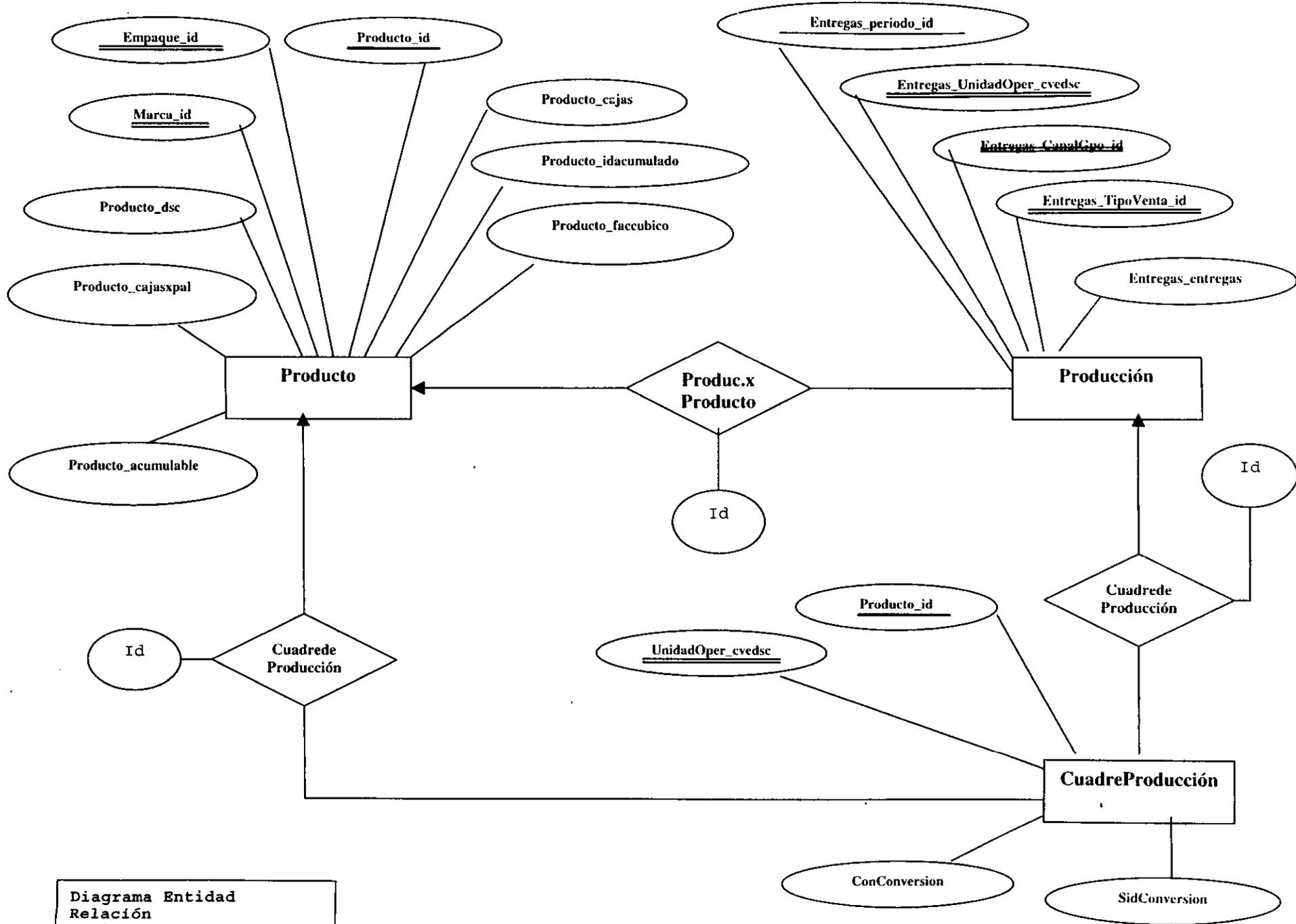


Diagrama Entidad
Relación
Entidad Producto

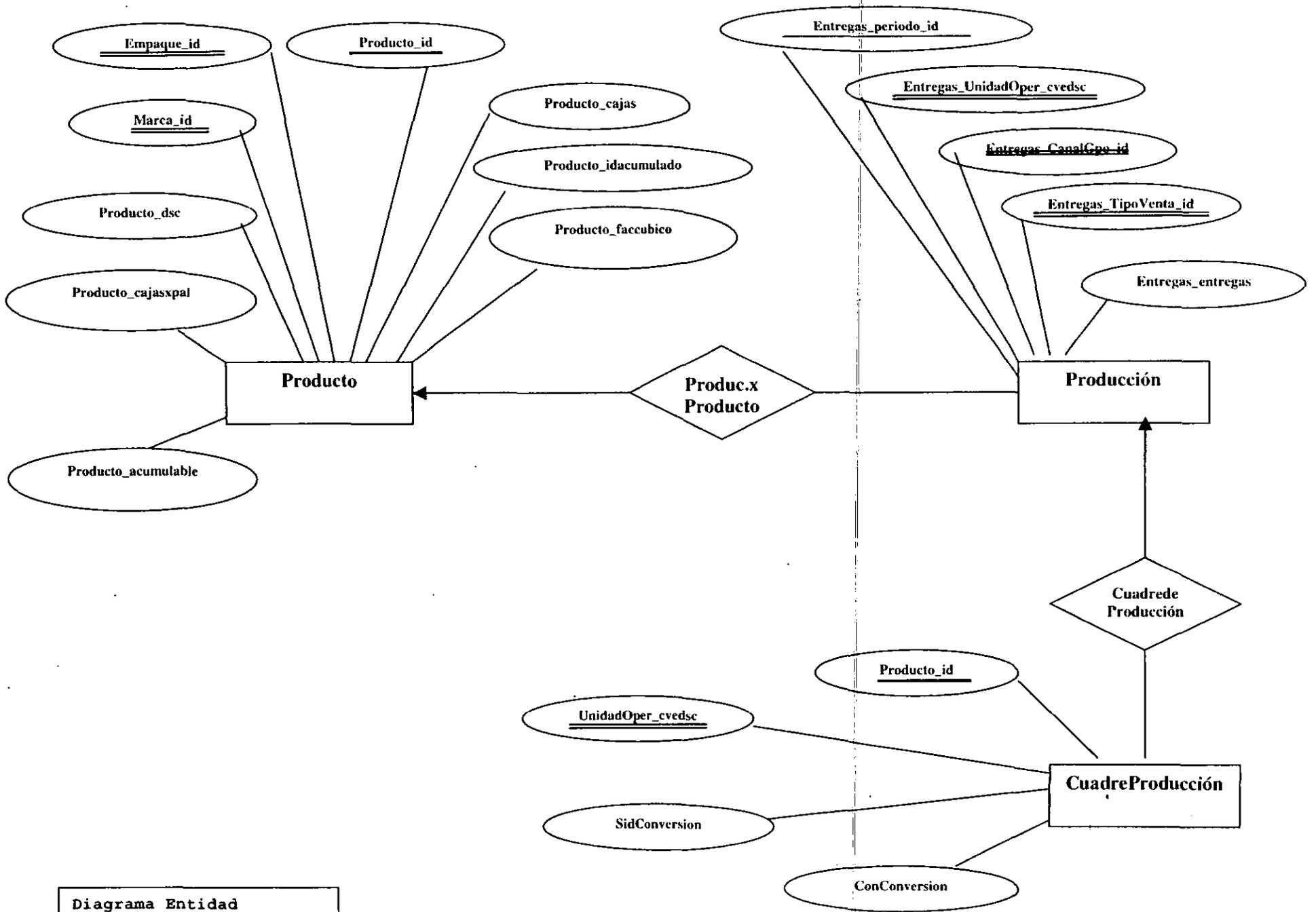


Diagrama Entidad Relación
Entidad Producto

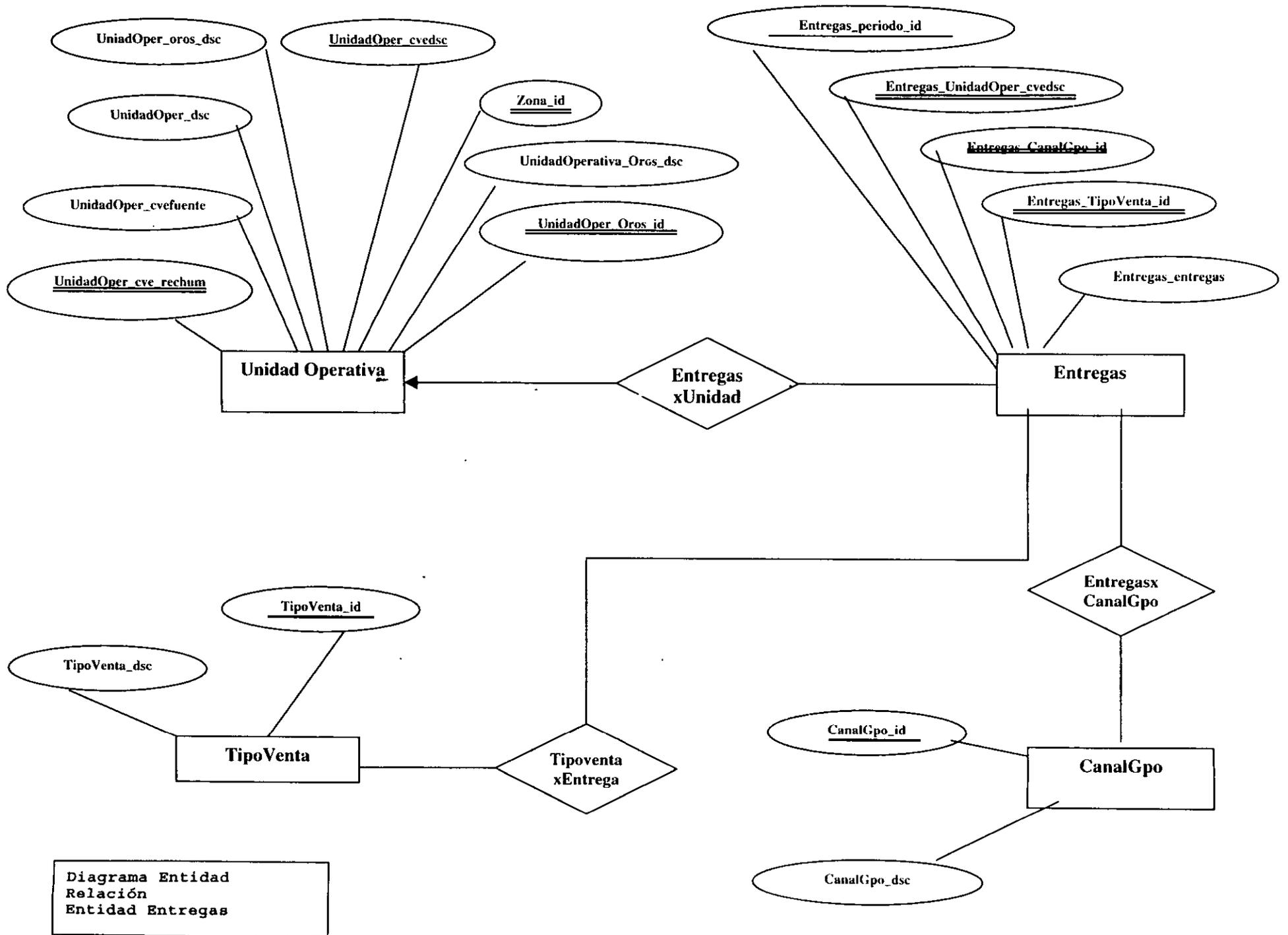


Diagrama Entidad
Relación
Entidad Entregas

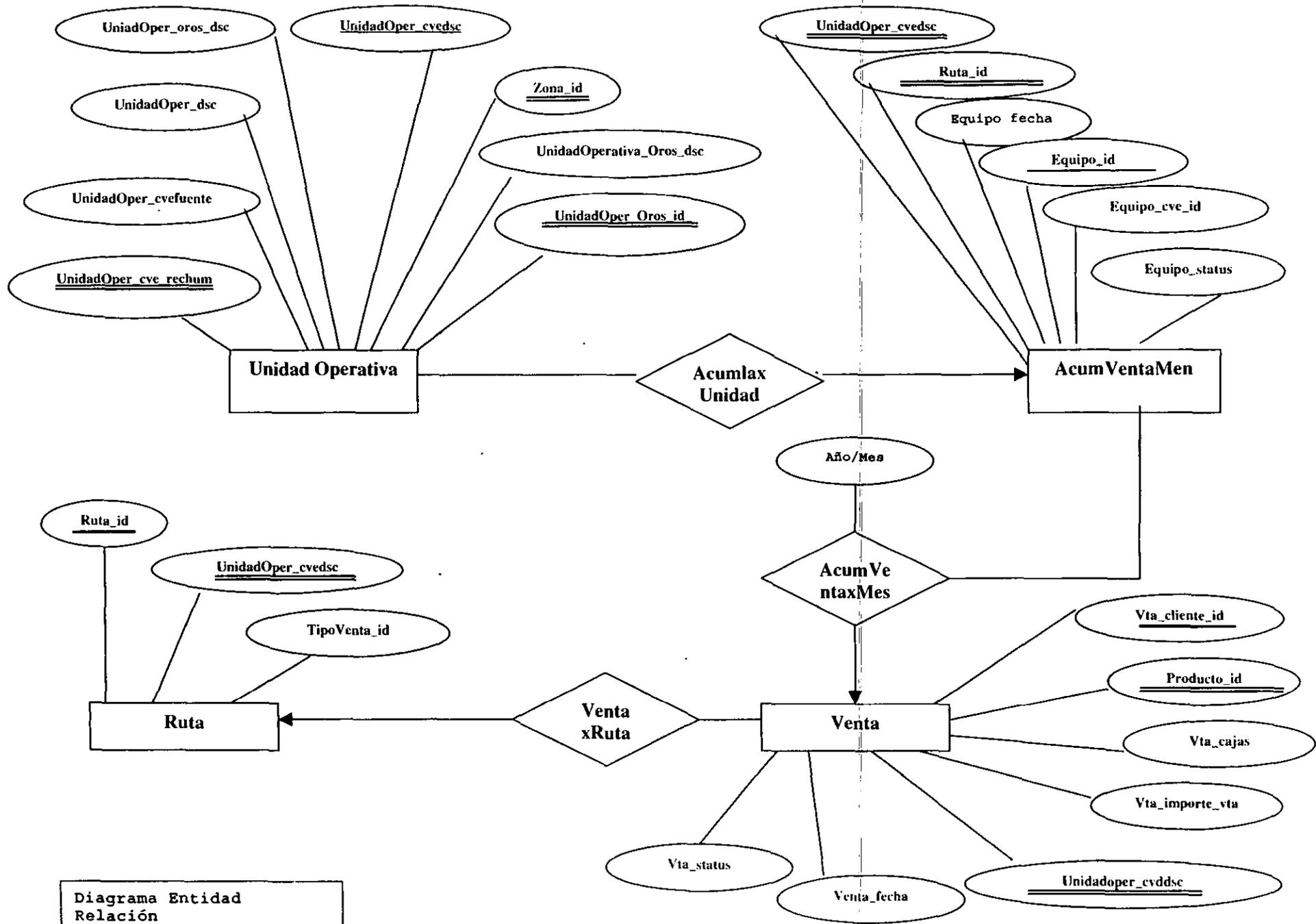


Diagrama Entidad Relación Entidad EQUIPO

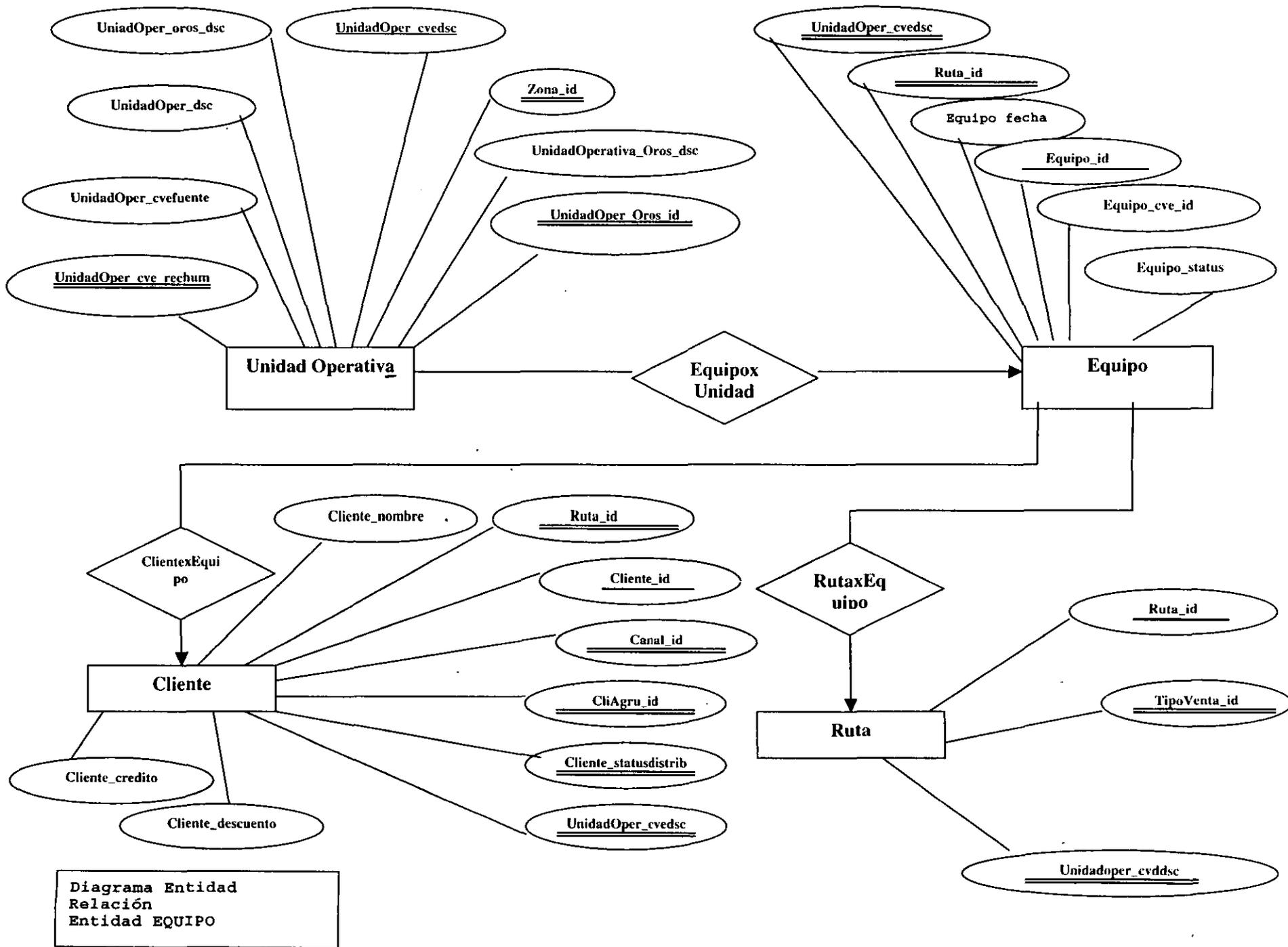


Diagrama Entidad
Relación
Entidad EQUIPO

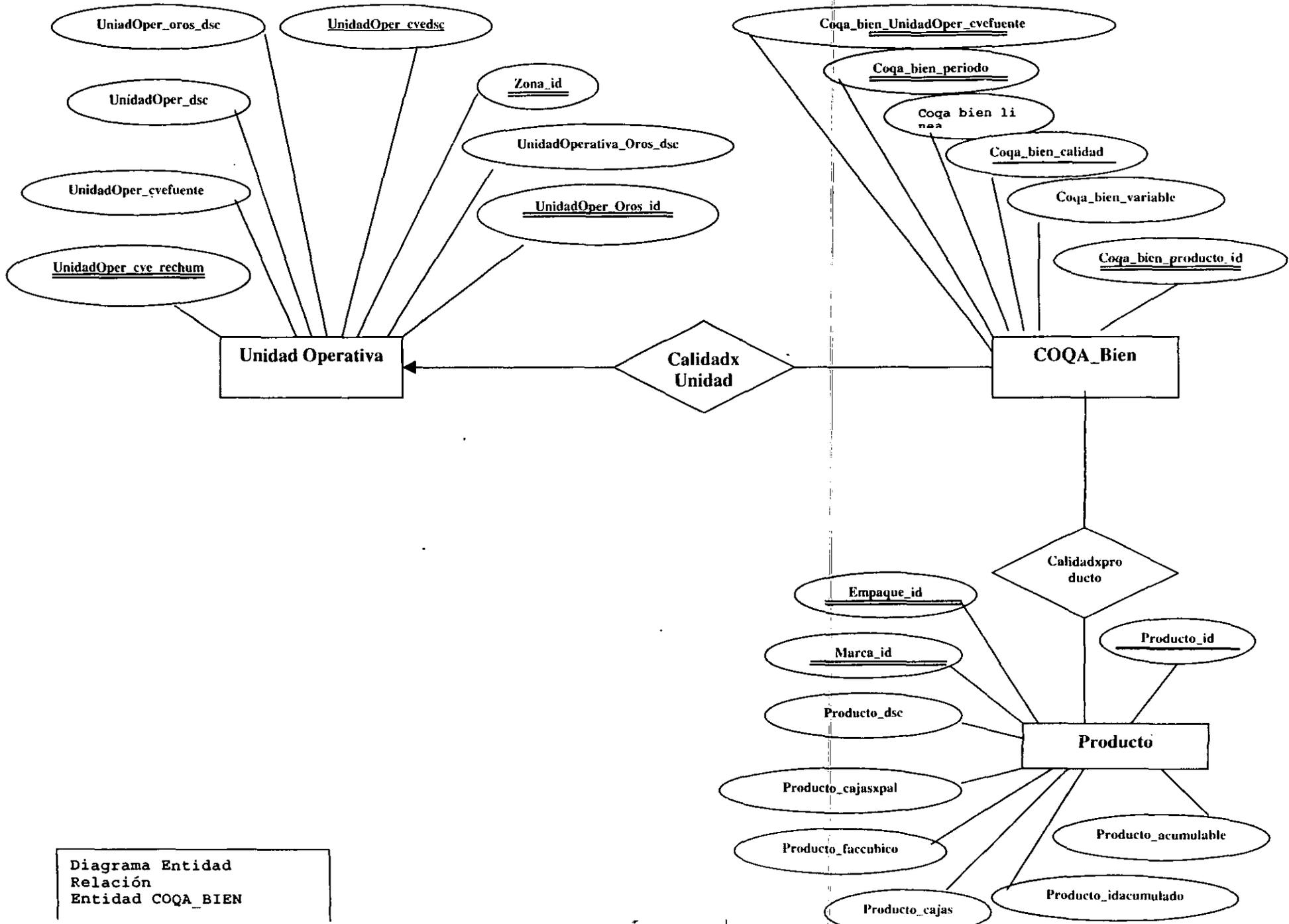


Diagrama Entidad Relación
Entidad COQA_BIEN

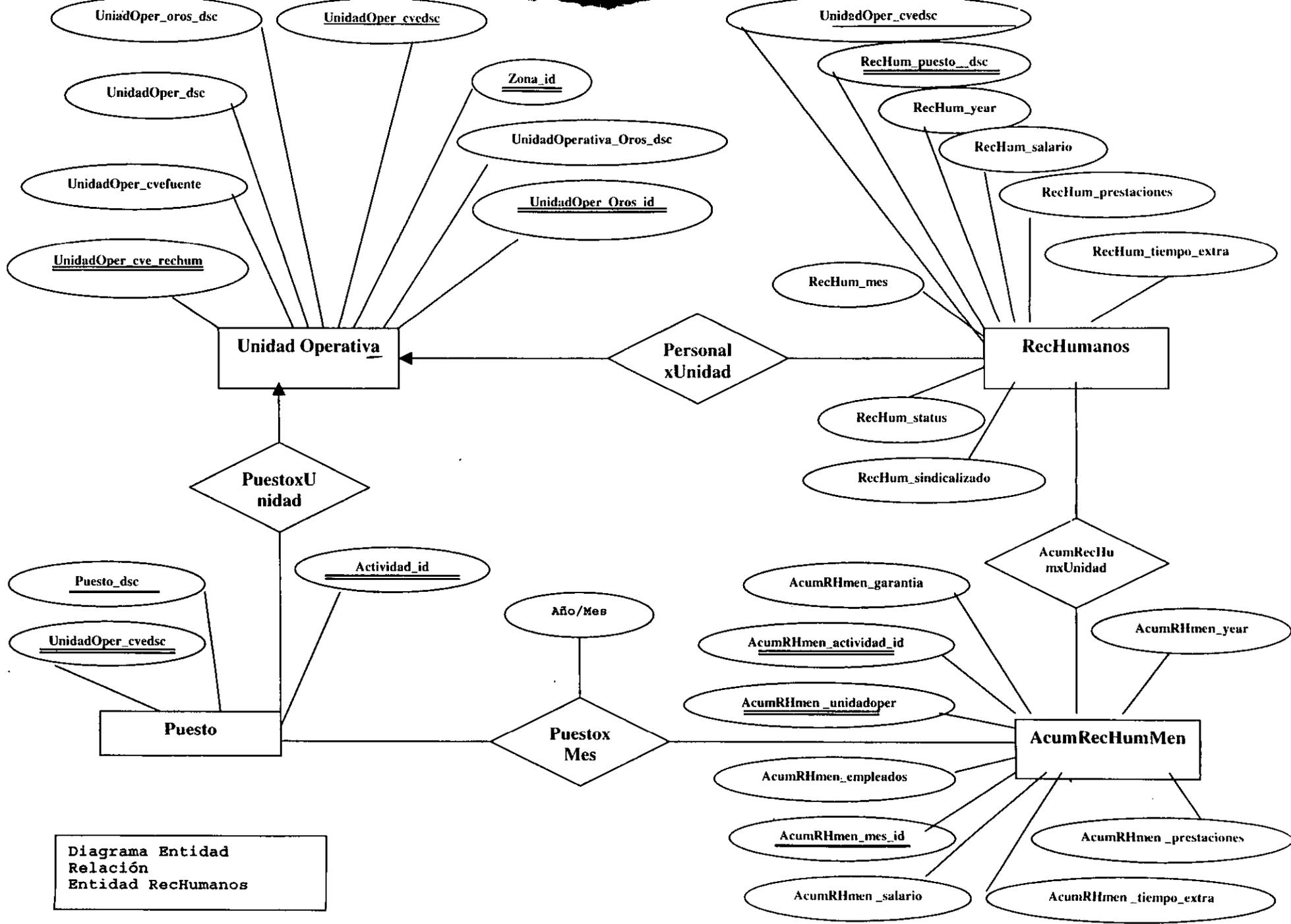


Diagrama Entidad
Relación
Entidad RecHumanos

