

MODULOS DE INVENTARIO Y FACTURACION EN DISTRIFAGAR LTDA.

ALEXANDER JOSE FARIAS PEREZ

CARLOS HUMBERTO GAMBOA HERNANDEZ

JAVIER SANCHEZ MUÑOZ

GEOVANI FABIAN VAZQUEZ RADA

UNIVERSIDAD SIMON BOLIVAR

FACULTAD DE INGENIERIA DE SISTEMAS

AREA DE FORMATIVA III

DECIMO SEMESTRE A-NOCTURNO

BARRANQUILLA

2007

CONTENIDO

1. INTRODUCCION	3
2. PLANTEAMIENTO DEL PROBLEMA	4
2.1 DESCRIPCION DEL PROBLEMA	4
2.2 FORMULACION DEL PROBLEMA	5
2.3 SISTEMATIZACION DEL PROBLEMA	5
3. OBJETIVOS	6
3.1 OBJETIVO GENERAL	6
3.2 OBJETIVOS ESPECIFICOS	6
4. JUSTIFICACION	7
5. MARCO TEORICO	8
5.1 ANTECEDENTES HISTORICOS	8
5.2 MODELO LINEAL SECUENCIAL	12
6. MARCO CONCEPTUAL	15
7. MARCO LEGAL	17
8. METODOLOGIA	19
8.1 TIPO DE INVESTIGACIÓN	19
8.2 LINEA DE INVESTIGACIÓN	19
8.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN	19
9. RECURSOS	20
9.1 RECURSOS HUMANOS	20

9.2 RECURSOS MATERIALES	20
9.3 RECURSOS INSTITUCIONALES	21
9.4 RECURSOS FINANCIEROS	21
10. CRONOGRAMA DE ACTIVIDADES	22
11. INGENIERIA DE REQUISITOS	23
11.1 DESCRIPCION DEL SISTEMA ACTUAL	23
11.1.1 PROCESOS QUE INTERVIENEN EN LA COMPRA Y VENTA DE PRODUCTOS	23
12.CASOS DE USO	25
13. ESPECIFICACIÓN DE REQUISITOS	27
13.1 FUNCIONALES	27
13.2 NO FUNCIONALES	29
14. INGENIERIA DE INFORMACION	30
14.1 HISTORIA	30
14.2 MISION	30
14.3 VISION	31
15. ORGANIGRAMA	32
16. DISEÑO DEL SISTEMA ESTRUCTURA FUNCIONAL	34
17. DESCRIPCION DE LOS COMPONENTES DE LA ESTRUCTURA FUNCIONAL	35
18. MODELO RELACIONAL	39
19. DICCIONARIO DE DATOS DEL MODELO RELACIONAL.	40

20. DIAGRAMA FLUJO DEL SISTEMA ACTUAL	47
21. CASOS DE USO EXPANDIDO	49
22. DIAGRAMA DE SECUENCIAS	53
23. BIBLIOGRAFIA	57

1. INTRODUCCIÓN

Desde hace algunos años los sistemas informáticos se han convertido en prioridad para las pequeñas, medianas y grandes empresas ya que la información es uno de los recursos de mayor importancia para las organizaciones, del manejo que se le de a la información de la empresa dependerá en gran parte su crecimiento y el éxito que esta pueda alcanzar. La competencia actual en los diversos mercados requiere que las empresas cuenten con información disponible e integral sobre la administración, utilizando la menor cantidad de tiempo en el desarrollo de tareas repetitivas y rutinarias, enfocándose en las actividades relevantes para la empresa.

En el mercado del software administrativo se ofrecen diversas soluciones, las empresas que las adquieren deben adaptarse o encontrar un punto intermedio entre este recurso y su forma de trabajo.

La finalidad de la construcción de un software contable en Distrifagar Ltda. Es facilitar el trabajo, es evitar la repetición de tareas, el doble registro de información es decir, al capturar la información una vez, ésta se debe compartir con las demás áreas de forma integral.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 DESCRIPCION DEL PROBLEMA

Distrifagar Ltda. Es una microempresa dedicada a la venta y distribución de repuestos para automotores. Actualmente en esta la información se maneja manualmente y es aquí donde se derivan una gran cantidad de dificultades y limitantes que afectan directamente el óptimo funcionamiento de las actividades diarias efectuadas en la empresa, como son la actualización, chequeo, y modificación de inventario como consecuencia de cualquier registro contable como la compra y venta de mercancía, son muchas las situaciones donde se ve claramente que se necesita de un sistema que permita el manejo de estas operaciones de manera practica, sencilla y ágil.

Un sector de la competencia a la que se enfrenta Distrifagar, ha implementado nuevas técnicas de manejo de información, lo cual de una u otra forma deja a la empresa en un nivel inferior de competitividad.

Si se desea mantener el buen desempeño dentro de un campo tan exigente como lo es el comercio, se debe cambiar la forma en que se están manejando los datos en el negocio; se debe implementar una forma mas eficiente de manipulación de

datos, que permita satisfacer las necesidades de los clientes y del medio en el que subsiste la empresa. Es por eso que se plantea la construcción de un software que sea capaz de suplir las necesidades y los requerimientos que exige Distrifagar Ltda.

2.2 FORMULACION DEL PROBLEMA

¿De qué manera se puede agilizar la gestión del proceso de facturación e inventario en Distrifagar Ltda. ?

2.3 SISTEMATIZACION DEL PROBLEMA

- Como evitar la inconsistencia de la información de inventario y facturación?
 - Como disminuir errores y dificultades causados por una mala relación Software-Usuario?
 - Como optimizar el tiempo de búsqueda de información y reportes?
 - Como reducir los problemas de información generados por mala redacción, tachones o ambigüedad?
 - Como reducir el tiempo empleado para el proceso de facturación y control inventario?
 - Como disminuir el tiempo de búsqueda de datos ya registrados?
-

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Crear un sistema de inventario y facturación para Distrifagar Ltda.

3.2 OBJETIVOS ESPECIFICOS

- Construir una herramienta que administre la información de inventario y facturación.
 - Facilitar el manejo del sistema diseñando una interfaz de fácil manejo para el usuario final.
 - Permitir generar reportes actualizados de manera fácil y oportuna.
 - Aumentar el nivel de seguridad y confiabilidad de la información
 - Optimizar el procesamiento, almacenamiento y recuperación de la información de inventario y facturación.
 - Facilitar el acceso a información de periodos (días, meses) anteriores.
-

4. JUSTIFICACION

Este proyecto nos llevara a comprender de mejor manera problemas de análisis y solución de requerimientos para empresas y organizaciones, para nosotros es de vital importancia enriquecernos con los procedimientos, conclusiones y resultados vistos en el transcurso y finalización de la investigación. Estos conceptos y resultados aportaran a nuestra formación como ingenieros nos enseñaran a analizar problemas de modelos de sistemas de información, en donde se deba conocer el nivel de factibilidad de la creación e implementación de bases de datos.

Los múltiples inconvenientes que se presentan en Distrifagar Ltda. Por la manipulación manual de la información (Borriones, inconsistencia, perdida, entre otros), hacen vano cualquier esfuerzo de crecimiento ya que entorpecen el optimo desarrollo de la empresa, con este proyecto se busca establecer la mejor manera de manipular esa información de modo que a futuro se dedique mayor tiempo y esfuerzo a las actividades comerciales, que son la naturaleza de las operaciones fundamentales de este tipo de negocio. Por esto, se plantea la creación de un sistema de información contable en Distrifagar Ltda.

5. MARCO TEORICO

5.1 ANTECEDENTES HISTORICOS

Inventario, una palabra que no nos es desconocida del todo, aunque solo sea referencialmente, pero que no la conocemos tan a fondo, como para temerle o desafiarle, sin embargo sabemos que en estos tiempos modernos, la palabra inventario ha alcanzado una notoriedad muy alta, hemos escuchado que debemos mantener niveles óptimos de inventarios, pero ¿Que es eso?; ¿Cuanto inventario se deber mantener? Si se mantienen inventarios demasiado altos, el costo podría llevar a una empresa a tener problemas de liquidez financiera, esto ocurre porque un inventario "parado" inmoviliza recursos que podrían ser mejor utilizados en funciones mas productivas de la organización. Además, el inventario "parado" tiende a tornarse obsoleto, a quedar fuera de uso y corre el riesgo de dañarse. Por otro lado, si se mantiene un nivel insuficiente de inventario, podría no atenderse a los clientes de forma satisfactoria, lo cual genera reducción de ganancias y perdida de mercado, al no afirmar la confiabilidad de los clientes en la capacidad de reacción de la empresa, ante las fluctuaciones del mercado.

El manejo de inventarios ha llegado a la cumbre de los problemas de la administración de empresas debido a que es un componente fundamental de la

productividad. La empresa de hoy tiene que ser productiva para sobrevivir y prosperar. En mercados altamente competitivos, las empresas trabajan con inventarios cada vez más bajos y con niveles de servicios cada vez más altos.

En las empresas comerciales, cuyo objeto social es la compraventa de mercancías, los inventarios deben ser valorados técnicamente, ya que son fundamentales para determinar los resultados del ejercicio contable. Para esto es necesario una aplicación capaz de gestionar, controlar e informar en todo tipo de transacciones independientemente de que se retiren las copias duras, referencias a documentos escaneados, o búsqueda de documentos almacenados externamente, estas cualidades son relacionadas en el PowerPick Office. Este es un software aplicativo con módulos contables que trabaja con todos los sistemas operativos Windows comunes y puede conectarse a diferentes sistemas de host.

Las características principales del PowerPick Office son:

- Mejora de los tiempos de respuesta a las peticiones de los clientes / proveedores.
 - Mejora la productividad de una manera significativa al reducir el tiempo perdido buscando documentos trasapelados.
 - Incrementa la precisión al archivar, clasificar y confirmar todas las transacciones, eliminando el error humano.
-

- Ofrecen un completo abanico de informes detallando un historial de transacciones.

Todas estas cualidades de PowerPick deben ser incluidas en el desarrollo de la aplicación para Distrifagar Ltda.

Con el fin de registrar y controlar los inventarios, de acuerdo con la magnitud de los negocios, el volumen de ventas y de existencias de mercancías, los comerciantes pueden elegir el sistema que mejor se adapte a sus necesidades.

Los sistemas mas utilizados son:

- Sistema de inventario periódico o juego de inventarios.
- Sistema de inventario permanente o continuo.

En el **sistema periódico de inventarios** el costo de lo vendido en el sistema de juego de inventarios se determina tomando el valor del inventario inicial, más el costo de las mercancías compradas, menos el inventario final. Este juego de inventarios debe realizarse como mínimo una vez al año para efectos fiscales.

- *Inventario inicial* es la relación detallada y minuciosa de las existencias de mercancías que tiene una empresa al iniciar sus actividades, después de hacer un conteo físico.
-

➤ *Inventario final* es la relación de existencias al finalizar un periodo contable.

Mediante el **sistema de inventario permanente o perpetuo** la empresa conoce el valor de la mercancía en existencia y además se conoce el costo de la mercancía vendida en cualquier momento, sin necesidad de realizar un inventario físico, por que los movimientos de compraventa se registran directamente en la cuenta Mercancías no fabricadas por la Empresa a precio de costo.

La empresa que adopta este sistema requiere un fichero o auxiliar de mercancías denominado kardex, en el cual se registra cada artículo como subcuenta de la cuenta Mercancías. La suma de los saldos de cada subcuenta da el saldo de mercancías, por esta razón este sistema recibe el nombre de *inventario permanente o perpetuo*; si adoptan el sistema de inventario permanente deben implantar un procedimiento de rutina, que permita controlar con exactitud el movimiento de las mercancías. Para visualizar esta información se requiere un kardex.

El kardex de mercancías esta formado por tarjetas que permiten controlar las cantidades y costos de las entradas y salidas de un artículo determinado, y dar a

conocer las existencias en cualquier momento sin necesidad de realizar un inventario físico.¹

5.2 MODELO LINEAL SECUENCIAL

También llamado "Ciclo de vida básico" o "Modelo de cascada" tiene su origen en el "Modelo de cascada" ingeniado por Winston Royce, aunque omite los muchos bucles de este último. El MLS sugiere un enfoque sistemático o más bien secuencial del desarrollo de software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento. El MLS acompaña las siguientes actividades:

Primero se realiza el **análisis de los requerimientos del software**, fase en la cual se reúnen todos los requisitos que debe cumplir el software. En esta etapa es fundamental la presencia del cliente que documenta y repasa dichos requisitos.

En segundo lugar esta el **diseño**, etapa dirigida hacia la estructura de datos, la arquitectura del software, las representaciones de la interfaz y el detalle procedimental (algoritmo). En forma general se hace un esbozo de lo solicitado y se documenta haciéndose parte del software.

¹ Gudiño, Emma Lucía y Coral, Lucy del C. Contabilidad 2000 plus. Quinta edición. Bogotá D.C. Mac Graw Hill, 2005. p. 144 – 168.

Una vez establecido el diseño se procede a la **generación del código**, etapa en la cual se traduce el diseño para que sea comprensible por la máquina. Esta etapa va a depender estrechamente de lo detallado del diseño.

Luego se realizan las **pruebas**. Esta etapa se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores.

Por último el **mantenimiento**, debido a que el programa puede tener errores, puede no ser del completo agrado del cliente o puede necesitar, eventualmente acoplarse a los cambios en su entorno. Esto quiere decir que no se rehace el programa, sino que sobre la base de uno ya existente se realizan algunos cambios.

El MLS es el paradigma de desarrollo de software más antiguo que existe, sin embargo esto no ha impedido que se haya creado una desconfianza alrededor de él basada en los siguientes errores reales:

- Los proyectos raramente siguen el paradigma secuencial que propone el proyecto.
-

- A menudo es difícil que el cliente exponga exactamente todos los requisitos.
- El cliente debe tener paciencia.
- Los responsables del desarrollo de software siempre se retrasan innecesariamente.

Todo lo anteriormente expuesto es cierto pero este paradigma tiene un lugar bien definido e importante en el trabajo de la Ingeniería de Software a parte de proporcionar una plantilla en la que se encuentran métodos para análisis, diseño, codificación, pruebas y mantenimiento. Con todo y sus errores, sigue siendo el paradigma más utilizado en el desarrollo del software, siendo mucho mejor que un enfoque al azar.²

² Tomado de <http://www.google.com/definicion/cascada/modelo.html>

6. MARCO CONCEPTUAL

Debido al crecimiento que han tenido las PYMES en Colombia y su gran acogida en el mercado, estas deben satisfacer de la manera mas eficiente, las necesidades de la demanda y asegurar un buen papel en un mundo tan lleno de competidores fuertes y que están dispuestos a quedarse con el apoyo y aceptación del consumidor, el tiempo de espera en el acceso a datos y de respuesta, la información inconsistente y redundante son inconvenientes con los que se enfrentan a diario las pequeñas empresas comercializadoras y distribuidoras de productos. Un modulo de inventario y facturación es un sistema que permite optimizar el tiempo empleado en la realización de operaciones contables realizadas manualmente, de manera que se puedan obtener datos de manera sencilla y confiable, pues la redundancia e inconsistencia de los datos son reducidas a margen mínimo. Evita la ambigüedad de los datos pues no cabe lugar a tachones o enmendaduras ocurridas en las operaciones manuales.

Teniendo como propósito aclarar la utilización de algunos conceptos objetos de estudio, se utilizaran las principales definiciones:

- Modulo: Termino usado en programación para hacer referencia a componente de un paquete de software.
-

- Lenguaje de programación: Cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador
 - MySql: Motor de base de datos empleado para almacenar datos, cuya característica principal es ser un software GNU.
 - Aplicación: Conjunto integrado de programas que realizan una función en particular y que tiene actividades de entrada, procesamiento y salida de datos
 - Visual Basic: Lenguaje de programación orientado a objetos desarrollado por Microsoft.
 - Orientado a objetos (POO): Es una metodología de diseño de software y paradigma de programación que define a los paradigmas en términos de clases y objetos, que son entidades que combinan estados (Datos) y comportamiento (Proceso o método).
 - Usabilidad: Facilidad de uso intuitivo del software por parte del usuario
 - Objeto: Es algo que tiene atributos (Describen), asocian métodos (Acción) y las acciones las indicamos mediante mensajes.
-

7. MARCO LEGAL

El reglamento general y los principios o normas de contabilidad generalmente aceptados deben ser observados por todas las personas a quienes la ley obliga a llevar contabilidad, entre otros los siguientes sujetos:

Todos los comerciantes, es decir, las personas que directa o indirectamente se ocupan en algunas actividades que la ley considera mercantiles, en virtud del artículo 19, numeral 3 del código de comercio.

Los comerciantes que se encuentren sometidos a estatutos especiales, como es el caso de los vigilados por entidades gubernamentales, deberán aplicar, en virtud del principio hermenéutico de la especialidad, las disposiciones particulares para dichos entes y en forma supletiva las normas del Decreto Reglamentario 2649 de 1993.

Acerca del software. En cuanto al licenciamiento es del tipo software propietario, cuya principal característica es que no está disponible el código fuente. Además Distrifagar no podrá por ningún motivo distribuir o reproducir el software o cualquier parte del mismo de manera ilícita, de ser así será objeto de acciones judiciales correspondientes.

La propiedad intelectual del mismo corresponde únicamente a los integrantes del proyecto de investigación.

La ley 44 de 1993 especifica penas entre dos y cinco años de cárcel así como el pago de indemnizaciones por daños y perjuicios, a quienes cometan el delito de piratería del software. Se considera delito el uso o reproducción de un programa de computador de manera diferente a como está estipulado en la licencia. Los programas sin licencia son ilegales.

La reforma del Código de Procedimiento Legal, que entró en vigencia a partir del mes de Julio de 2001 convierte en no excarcelable los delitos en contra de la propiedad intelectual y los derechos de autor. Lo que significa que quienes sean encontrados usando, distribuyendo o copiando software sin licencia deberá estar en la cárcel por un periodo de cinco años.

8. METODOLOGIA

8.1 TIPO DE INVESTIGACIÓN

El proyecto de estudio corresponde al tipo de investigación descriptivo, dado que el objetivo principal es identificar los requerimientos y necesidades del sistema. Buscando obtener datos precisos, el tipo Investigación empleado es cualitativo, ya que los datos son obtenidos a partir de técnicas de recolección de información.

8.2 LINEA DE INVESTIGACIÓN

Debido a las características del sistema que se desarrollará la línea de investigación a seguir es Ingeniería del Software.

8.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

Las técnicas utilizadas durante la investigación fueron de Entrevistas para fortalecer la parte teórica del anteproyecto.

La **entrevista** fue dirigida a la persona encargada de la contabilidad en la empresa, al gerente y al subgerente. Con el fin de recolectar más información relacionada con los procesos contables que se llevan a cabo en la empresa. A través de ésta se obtuvo información general que nos permitirá un mejor desarrollo de las fases de diseño de la herramienta.

9. RECURSOS

9.1 RECURSOS HUMANOS

El Proyecto en curso esta siendo desarrollado por Estudiantes de X semestre nocturno de Ing. de sistemas:

- Alexander Farias, Javier Sánchez, Carlos Gamboa y Geonany Vázquez.

Además contribuyeron para la realización de dicho proyecto:

- Ing. Luis Briceño. Docente de la asignatura de Formativa II de la Universidad Simón Bolívar.
- Sra. Moramay farias Asistente contable de Distrifagar Ltda.

9.2 RECURSOS MATERIALES

- Computador instalado en las instalaciones de Distrifagar Ltda.
-

- Computadores personales de las estudiantes que conforman el proyecto de investigación.
- 1 Impresora Epson LX-300.
- 8 Disquetes.
- 150 Hojas empleadas en trabajo escrito y borrador.
- 6 Lapiceros.
- 2 Portaminas.
- 2 Correidores.
- 5 Carpetas de presentación.
- 3 CD - ROM

9.3 RECURSOS INSTITUCIONALES

Corporación Educativa Mayor del Desarrollo Simón Bolívar.

9.4 RECURSOS FINANCIEROS

Descripción	Cantidad	Valor Unitario	Valor Total
Fotocopias	600	70	42000
Hojas	150	100	15000
Lapiceros	6	750	4500

8. CRONOGRAMA DE ACTIVIDADES

ACTIVIDADES	DURACION DE ACTIVIDADES POR SEMANAS															
	2005															
	Agosto				Septiembre				Octubre				Noviembre			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Eleccion del tema para para la realizacion del proyecto																
2. Realizacion del planteamiento del problema, objetivos y Justificacion																
3. Realizacion del marco de referencia																
4. Hipotesis																
5. Realizacion de la metodologia																
6. Muestro																

ACTIVIDADES	DURACION DE ACTIVIDADES POR SEMANAS															
	2006															
	Febrero				Marzo				Abril				Mayo			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Ingenieria de requisitos (Especificacion del sistema actual, identificacion y analisis de requisitos)																
2. Ingenieria de la informacion (Mision, vision, historia, politicas y organigrama de la informacion)																
3. Analisis del sistema (Especificaciones de entidades, diagrama E/R, relacional y diccionario de datos)																
4. Diseño del sistema (Estructura funcional, descripcion y diseño de las interfaces (Prototipo))																

ACTIVIDADES	DURACION DE ACTIVIDADES POR SEMANAS															
	2006															
	Agosto				Septiembre				Octubre				Noviembre			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Desarrollo y elaboracion del software																
2. Investigar y colocar en practica en el proceso de codificacion																
3. Realizar pruebas con datos reales para verificar que se cumplan los requerimientos establecidos																

ACTIVIDADES	DURACION DE ACTIVIDADES POR SEMANAS															
	2007															
	Febrero				Marzo				Abril				Mayo			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Verificacion del diseño y desarrollo																
2. Desarrollar, elaborar e implementar los reportes dinamicos y las ayudas en linea de la aplicación																
3. Control de cambios de diseño y desarrollo																
4. Validacion del diseño y desarrollo																
5. Elaborar manuales como son el manual de usuario y sistema con el fin de sus entendimiento y facilitar el uso de la aplicación																

11. INGENIERIA DE REQUISITOS

11.1 DESCRIPCION DEL SISTEMA ACTUAL

11.1.1 PROCESOS QUE INTERVIENEN EN LA COMPRA Y VENTA DE PRODUCTOS

1. El cliente solicita los productos.
 2. Se procede a liquidar los productos para facturar.
 3. Realizada la factura pasa al auxiliar contable el cual procede a registrarla en una hoja de diario que es donde se realizan todos los movimientos de venta del día, este movimiento diario lleva un consecutivo o numeración.
 4. Las facturas ya registradas en el libro de Diario, pasan al contador quien se encarga de cargar o descargar el monto crédito o débito de cada cliente existente.
 5. En el caso de que el cliente compre de contado se le entregara copia de la factura generada, al momento de entregarle el producto, este dinero se almacena en caja y ahí se carga como "venta de contado".
-

6. Cuando el cliente esta registrado en el libro como clientes de crédito directo se le informa de forma escrita cuanto debe, estipulando fecha de inicio del crédito y fecha estipulada para pagar dicha factura, adjuntamente se le informa cuanto es el monto de dinero que debe cancelar para estar a paz y salvo con el almacén.

7. El pedido al proveedor se realiza llenando un formato numerado especificando la fecha, el nombre del proveedor y detallando los productos y su cantidad, posteriormente se le hace llegar al proveedor de forma telefónica (fax).

8. Cuando el pedido llega, el proveedor envía junto con los productos, la factura de venta, la cual contiene un número, fecha, subtotal sin IVA, total con IVA, y detallando los productos, su cantidad y subtotales.

9. Con esta información el jefe de almacén busca el kardex de cada producto y anota la entrada del insumo, contemplando la fecha, entrada por proveedor, la cantidad y además se compara el pedido contra la factura con el fin de verificar si todos los insumos llegaron completos, de no ocurrir así, se marca el pedido como incompleto y se elabora una relación de los insumos pendientes, para así poder generar otro pedido con los productos faltantes al mismo u otro proveedor, en una fecha posterior. La factura se almacena y se le agrega un sello de recibido.

12. CASOS DE USO

CASO DE USO:

Referencia del Caso: RF 001

Nombre del Caso: VENTA DE UN PRODUCTO

Definir Actores: VENDEDOR, CAJERO, CLIENTE (init)

Descripción: El cliente solicita productos el vendedor despacha
Productos y entrega factura, el cliente se dirige a la caja y
Decide su tipo de pago (Contado o crédito).
El cajero recibe el pago o la factura y registra la venta
El vendedor entrega el producto.

Referencias cruzadas: R2, R3, R9.

CASO DE USO:

Referencia del Caso: RF 002

Nombre del Caso: COMPRA DE PRODUCTO

Definir Actores: CAJERO, PROVEEDOR (init), ALMACENISTA.

Descripción: El proveedor entrega la mercadería y factura al almacenista, este
Revisa la mercancía y entrega la factura con el VoBo, el cajero
Ingresar datos de la factura en los egresos, descarga el dinero y
Lo entrega al proveedor si el pago es de contado, si el pago es a
Crédito el cajero descarga la factura en las cuentas por pagar
Del sistema y entrega al proveedor la factura que cobrara en
Una fecha pospuesta.

Referencias cruzadas: R1.

CASO DE USO:

Referencia del Caso: RF 003

Nombre del Caso: MOVIMIENTO DE CUENTAS POR COBRAR.

Definir Actores: CAJERO, CLIENTE (init).

Descripción: El cliente se acerca a la caja para cancelar una factura Pendiente, el cajero busca en el sistema la cuenta del cliente Y cierra esa cuenta por cobrar si el cliente, cancela el valor Total de esta, luego le entrega un recibo constando que la Deuda fue saldada o resta del saldo pendiente, si es un Abono. En cualquiera de los casos, el cajero resta el valor Recibido del movimiento de cuentas por cobrar al Respectivo Cliente y registra el mismo valor en el sistema.

Referencias cruzadas: R3, R4.

CASO DE USO:

Referencia del Caso: RF 004

Nombre del Caso: MOVIMIENTO DE CUENTAS POR PAGAR.

Definir Actores: CAJERO, PROVEEDOR (init).

Descripción: La fecha de cancelación de una ó varias facturas se vence, y El proveedor cobra el valor de esta. El cajero de contado el Valor total de la factura y resta en el movimiento de cuentas Por pagar del sistema, el valor cierra de esta manera la Cuenta por pagar, si el cajero solo hace un abono se resta el Valor del abono tanto en la factura como en el movimiento De cuentas por pagar del respectivo proveedor en la factura Del sistema.

Referencias cruzadas: R4, R11, R13.

CASO DE USO:

Referencia del Caso: RF 005

Nombre del Caso: REGISTRO Y CONSULTA DE CLIENTES.

Definir Actores: ALMACENISTA, CLIENTE (init)

Descripción: El registro del cliente llega al momento de realizar la factura, Se capturan todos sus datos para registrarlos en una base De datos de clientes donde de igual forma si se necesita el Nit u otra característica del cliente simplemente se busca el Nombre del cliente y se consulta todos sus datos.

Referencias cruzadas: R6, R8.

CASO DE USO:

Referencia del Caso: RF 006

Nombre del Caso: REPORTE DE PAGOS DE CLIENTES A CREDITO O DE CONTADO.

Definir Actores: CAJERO, CLIENTE (init)

Descripción: El cliente llega al almacén y se dirige a la caja donde el Cajero respectivamente le recibe el dinero dando por Cancelado la factura que obtuvo de compra a crédito que Realizo en el almacén.

Referencias cruzadas: R12, R14.

13. ESPECIFICACIÓN DE REQUISITOS FUNCIONALES

REFERENCIA	DESCRIPCION	CATEGORIA
R1	Generar reportes oportunos acerca del estado actual del inventario.	EVIDENTE
R2	Venta de productos de contado.	EVIDENTE
R3	Registrar y consultar movimientos de inventario, generados por	EVIDENTE

	cualquier concepto (Compra, Venta, devolución, entre otros.)	
R4	Pago de cuentas por cobrar.	EVIDENTE
R5	Controlar la existencia de productos en inventario.	OCULTOS
R6	Registrar y consultar clientes.	EVIDENTE
R7	Actualizaciones inmediatas de la información de los productos en inventario y clientes en la base de datos.	OCULTOS
R8	Ingresar nuevos clientes.	EVIDENTE
R9	Generar facturas de venta.	EVIDENTE
R10	Registrar y consultar productos.	EVIDENTE
R11	Pago a proveedores en efectivo o por cheque.	EVIDENTE
R12	Reporte de pago de clientes.	EVIDENTE
R13	Controlar el movimiento de cuentas por pagar.	OCULTOS
R14	Consulta de saldos de clientes existentes.	EVIDENTE
R15	Registrar y consultar proveedores.	EVIDENTE

13.2 NO FUNCIONALES

REFERENCIA	DESCRIPCIÓN
RN1	Fácil de usar.
RN2	Rendimiento optimo.
RN3	La base de datos debe ser realizada en mysql.
RN4	Tolerancia a fallos.
RN5	El sistema se maneja en una base de datos centralizada.
RN6	El entorno debe ser similar al de Windows en cuanto a apariencia de las ventanas y botones.

14. INGENIERIA DE INFORMACION

14.1 HISTORIA

Distrifagar Ltda... Debe su existencia al espíritu emprendedor de dos grandes personas Sr. José Farias y el Sr. Manuel García, el primero rectificador de cigüeñales y graduado del sena como técnico en Torno y Fresadora y el segundo egresado de estudios administrativos en Cajacopi, constituyeron este negocio con grandes expectativas el día 15 de noviembre de 1999. Desde entonces, ha sido propósito del almacén, promover y distribuir la venta de partes automotores brindando toda la confianza que necesitan los clientes, rectificadoras, y demás almacenes de todas las ciudades del país.

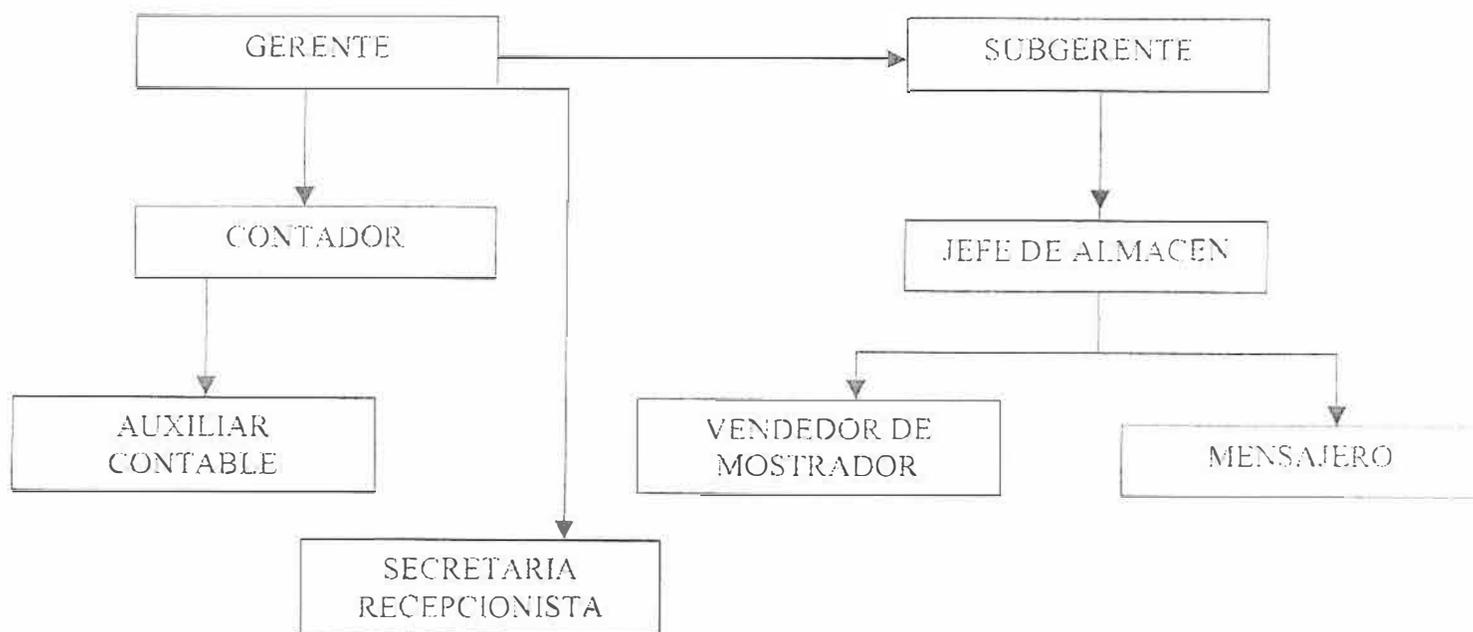
14.2 MISION

Distrifagar Ltda. es una microempresa dedicada a la venta de repuestos y partes de toda clase de motor, constituido por sociedad, para el servicio de la ciudad y orientado a el buen funcionamiento de sistemas automotores.

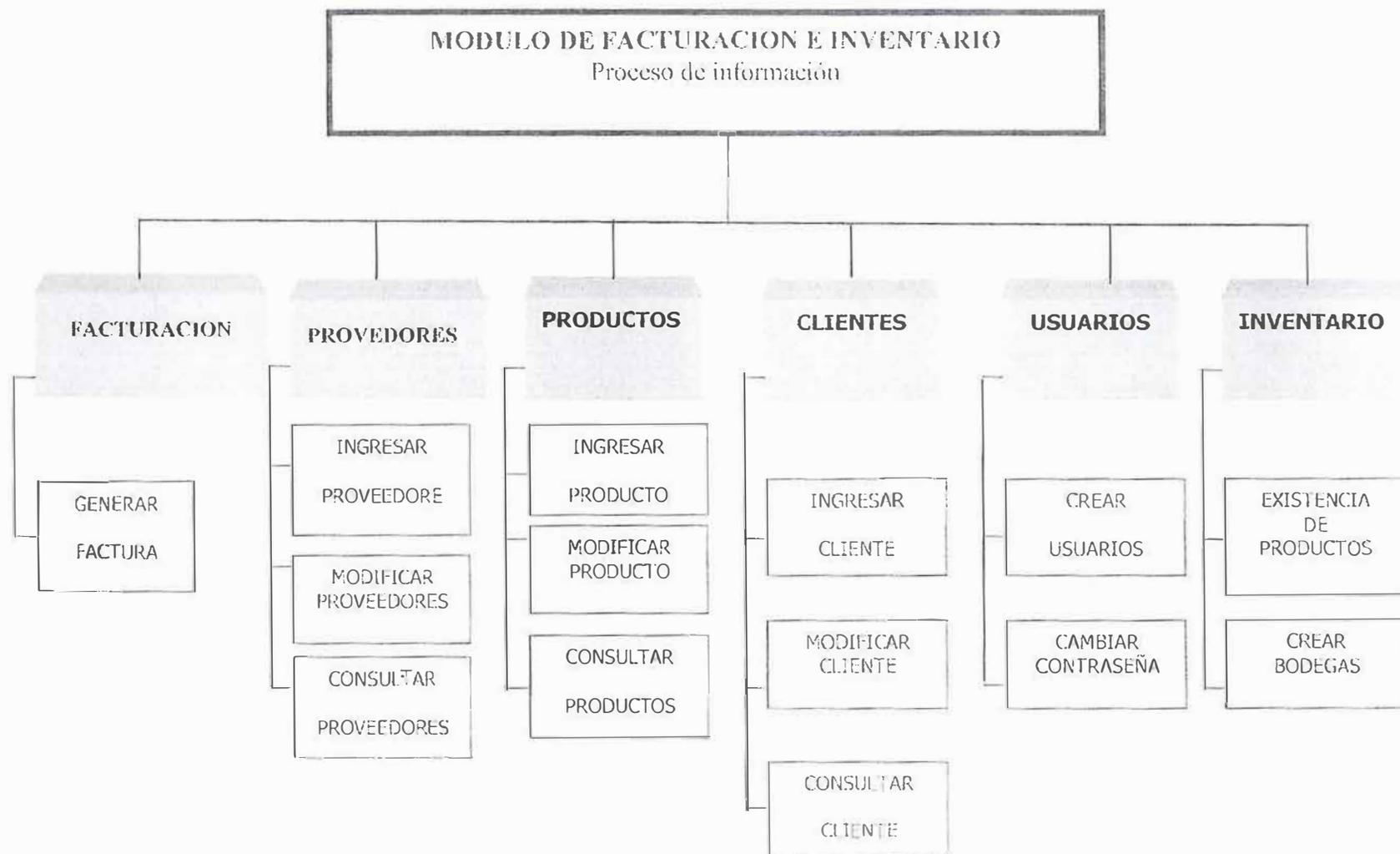
14.3 VISION

Ser en los próximos años el proveedor líder del mercado local, y brindar el mejor servicio a todos y cada uno de nuestros clientes, pues es por ellos que existimos y luchamos cada día por ser una empresa competente.

15. ORGANIGRAMA



16. DISEÑO DEL SISTEMA ESTRUCTURA FUNCIONAL



17. DESCRIPCION DE LOS COMPONENTES DE LA ESTRUCTURA FUNCIONAL

Facturación: En este menú se efectúan los respectivos procesos sobre una factura.

- **Generar factura:** En este modulo se registran los datos que van incluidos en la factura a generar.
- **Consultar factura:** En este modulo se consultan por medio del código de la factura, las facturas ya generadas (en la base de datos).
- **Anular factura:** en este modulo se les cambia el estado de una factura ya generada (a anulada).
- **Pago o cancelación de factura:** en este modulo se cancelan las facturas de ventas y de clientes.

Proveedores: En este menú se efectúan los respectivos procesos sobre los proveedores.

- **Ingresar proveedor:** En este modulo se ingresan los datos de los proveedores nuevos que entran a la base de datos.
 - **Modificar proveedor:** En este modulo se actualizan los datos de los proveedores ya ingresados a la base de datos.
-

- **Consultar proveedor:** En este modulo se consultan los proveedores ya creados en la base de datos.
- **Pagos de proveedores:** En este modulo se consultan los pagos realizados a proveedores.
- **Eliminar proveedor:** En este modulo se eliminan los proveedores de la base de datos, también se les puede cambiar el estado.

Productos: En este menú se efectúan los respectivos procesos sobre los productos.

- **Ingresar producto:** En este modulo se ingresan los datos de los productos nuevos que entran a la base de datos.
 - **Modificar producto:** En este modulo se actualizan los datos de los productos ya ingresados a la base de datos.
 - **Consultar productos:** En este modulo se consultan los productos ya creados en la base de datos.
 - **Compra de productos:** En este modulo se consulta la compra de productos de diferentes especificaciones.
 - **Eliminar producto:** En este modulo se eliminan los productos de la base de datos, también se les puede cambiar el estado.
-

Clientes: En este menú se efectúan los respectivos procesos sobre los clientes.

- **Ingresar cliente:** En este modulo se ingresan los datos de los clientes nuevos que entran a la base de datos.
- **Modificar clientes:** En este modulo se actualizan los datos de los clientes ya ingresados a la base de datos.
- **Consultar clientes:** En este modulo se consultan los clientes ya creados en la base de datos.
- **Eliminar clientes:** En este modulo se eliminan los clientes de la base de datos, también se les puede cambiar el estado.

Usuarios: En este menú se efectúan los respectivos procesos sobre los usuarios.

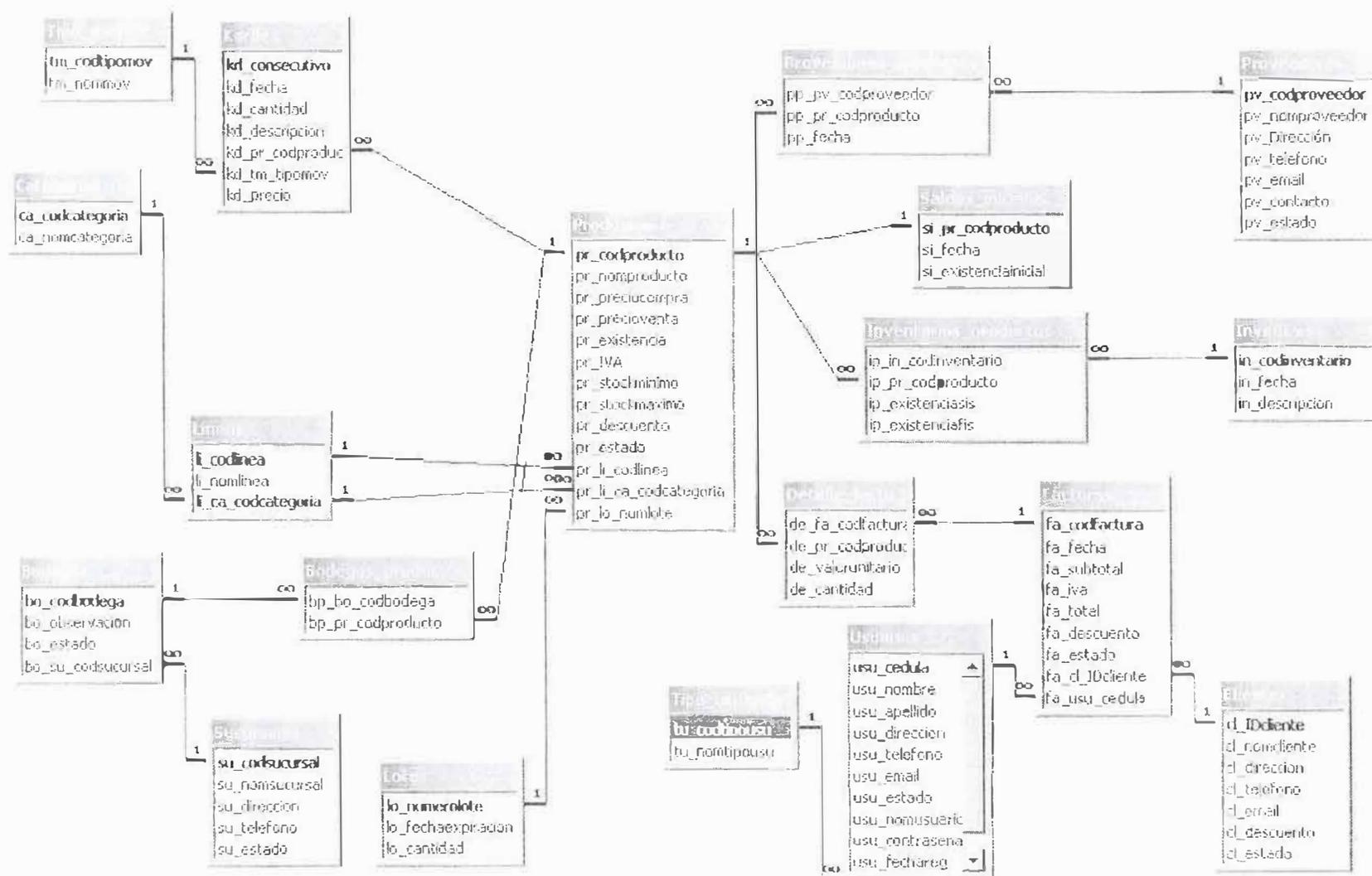
- **Crear usuarios:** En este modulo se ingresan los datos de los usuarios nuevos que entran a la base de datos.
- **Cambiar contraseña:** En este modulo se cambia la contraseña de ingreso al programa de los usuarios.

Inventario: En este menú se efectúan los respectivos procesos sobre el inventario de productos.

- **Existencia de productos en sistema:** En este modulo se consulta la cantidad general de los productos encontrados en sistema.
-

- **Existencia de productos en bodega (conteo físico):** En este modulo se ingresan la cantidad de productos encontrados en el inventario físico efectuado en la bodega.
 - **Cierre mes:** En este modulo se confrontan la información del inventario encontrado sistema con la información recopilada en el conteo físico.
-

18. MODELO RELACIONAL



19. DICCIONARIO DE DATOS DEL MODELO RELACIONAL.

Nombre de la tabla: Productos Aplicación: Inventario y Facturación					Descripción de la tabla:		
					Almacena la información de cada uno de los productos que ofrece el almacén de repuestos.		
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	pr_codproducto	PK	Numérico	10	0	No	Código del producto
2	pr_nomproducto		Texto	40	0	No	Nombre del producto
3	pr_preciocompra		Numérico	10	0	No	Precio de compra
4	pr_precioventa		Numérico	10	0	No	Precio de venta
5	pr_existencia		Numérico	10	0	No	Existencia actual del producto
6	pr_IVA		Numérico	5	0	No	IVA del producto
7	pr_stockminimo		Numérico	10	0	No	Stock mínimo del producto
8	pr_stockmaximo		Numérico	10	0	No	Stock máximo del producto
9	pr_descuento		Numérico	5	0	No	Descuento del producto
10	pr_estado		Texto	2	0	No	Estado del producto
11	pr_li_codlinea	FK	Numérico	10	0	No	Código de línea del producto
12	pr_li_ca_codcategoria	FK	Numérico	10	0	No	Código de la categoría del P.

Nombre de la tabla: Proveedores Aplicación: Inventario y Facturación					Descripción de la tabla:		
					Almacena la información de cada uno de los proveedores que tiene el almacén de repuestos.		
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	pv_codproveedor	PK	Numérico	10	0	No	Código del proveedor
2	pv_nomproveedor		Texto	40	0	No	Nombre del proveedor
3	pv_Dirección		Texto	40	0	No	Dirección del proveedor
4	pv_telefono		Numérico	15	0	Si	Teléfono del proveedor
5	pv_email		Texto	40	0	Si	Email del proveedor
6	pv_contacto		Texto	40	0	Si	Contacto del proveedor
7	pv_estado		Texto	2	0	No	Estado del proveedor

Nombre de la tabla: Sucursales
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las sucursales que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	su_codsucursal	PK	Numérico	10	0	No	Código de la sucursal
2	su_nomsucursal		Texto	40	0	No	Nombre de la sucursal
3	su_direccion		Texto	40	0	No	Dirección de la sucursal
4	su_telefono		Numérico	15	0	Si	Teléfono de la sucursal
5	su_estado		Texto	2	0	No	Estado de la sucursal

Nombre de la tabla: Bodegas
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las bodegas que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	bo_codbodega	PK	Numérico	10	0	No	Código de la bodega
2	bo_observacion		Texto	40	0	No	Observación de la bodega
3	bo_estado		Texto	2	0	No	Estado de la Bodega
4	bo_su_codsucursal	FK	Numérico	10	0	No	Código de la sucursal

Nombre de la tabla: Categorías
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las categorías de los productos que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	ca_codcategoria	PK	Numérico	10	0	No	Código de la categoría
2	ca_nomcategoria		Texto	40	0	No	Nombre de la categoría

Nombre de la tabla: Líneas
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las líneas de productos que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	li_codlinea	PK	Numérico	10	0	No	Código de la línea
2	li_nomlinea		Texto	40	0	No	Nombre de la línea
3	li_ca_codcategoria	FK	Numérico	10	0	No	Código de la categoría

Nombre de la tabla: Kardex

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las transacciones que se efectúa en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	kd_consecutivo	PK	Numérico	10	0	No	Código consecutivo del móv.
2	kd_fecha		Date	--	0	No	Fecha del movimiento
3	Kd_hora		time	--	0	No	Hora del movimiento
4	kd_cantidad		Numérico	10	0	No	Cantidad de productos
5	kd_descripcion		Texto	40	0	Si	Descripción del movimiento
6	kd_pr_codproducto	FK	Numérico	10	0	No	Código del producto
7	kd_tm_tipomov	FK	Numérico	10	0	No	Código del tipo de movimiento
8	tm_codsubtipomov	FK	Numérico	10	0	Si	Código del subtipo de móv.
9	kd_precio		Numérico	10	0	No	Precio del producto
10	Kd_usu_cedula	FK	Numérico	15	0	No	Cedula del usuario

Nombre de la tabla: Tipo_Movimientos

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada uno de los tipos de movimientos que se pueden efectuar en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	tm_codtipomov	PK	Numérico	10	0	No	Código del tipo de movimiento
2	tm_codsubtipomov	PK	Numérico	10	0	No	Código del subtipo de móv.
3	tm_nommov		Texto	40	0	No	Nombre del movimiento

Nombre de la tabla: Facturas

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las

facturas efectuadas en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	fa_codfactura	PK	Numérico	10	0	No	Código de la factura
2	fa_fecha		Date	--	0	No	Fecha de la factura
3	fa_subtotal		Numérico	10	0	No	Subtotal del valor de la factura
4	fa_iva		Numerico	--	0	No	IVA de valor total de la factura
5	fa_total		Numérico	10	0	No	Valor total de la factura
6	fa_descuento		Numérico	5	0	No	Descuento de la factura
7	fa_estado		Numérico	10	0	No	Estado de la factura
8	fa_cl_IDcliente	FK	Numérico	15	0	No	Identificación del cliente
9	fa_usu_cedula	FK	Numérico	15	0	No	Cedula de los usuarios

Nombre de la tabla: Detalle_facturas**Aplicación:** Inventario y Facturación**Descripción de la tabla:**

Almacena la información de cada uno de los detalles de las facturas que se efectúan en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	de_fa_codfactura		Numérico	10	0	No	Código de la factura
2	de_pr_codproducto		Numérico	10	0	No	Código del producto
3	de_valorunitario		Numérico	10	0	No	Valor unitario del producto
4	de_cantidad		Numérico	10	0	No	Cantidad de productos

Nombre de la tabla: Inventarios_productos**Aplicación:** Inventario y Facturación**Descripción de la tabla:**

Esta tabla nace de una relación muchos a muchos y almacena información de productos con su respectiva existencia asociados a un código de inventario en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	ip_in_codinventario		Numérico	10	0	No	Numero consecutivo de inv.
2	ip_pr_codproducto		Numérico	10	0	No	Código del producto
3	ip_existenciasis		Numérico	10	0	No	Existencia en sistema del prod.
4	ip_existenciafis		Numérico	10	0	No	Existencia física del producto

Nombre de la tabla: Inventarios Aplicación: Inventario y Facturación					Descripción de la tabla: Almacena la información de cada uno de los inventarios de productos realizados en el almacén de repuestos.		
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	in_codinventario	PK	Numérico	10	0	No	Código del inventario
2	In_fecha		Date	--	0	No	Fecha del inventario
3	in_descripcion		Numérico	10	0	No	Descripción del inventario

Nombre de la tabla: Clientes Aplicación: Inventario y Facturación					Descripción de la tabla: Almacena la información de cada uno de los clientes del almacén de repuestos.		
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	cl_IDcliente	PK	Numérico	15	0	No	Identificación del cliente
2	cl_nomcliente		Texto	40	0	No	Nombre del cliente
3	cl_direccion		Texto	40	0	No	Dirección del cliente
4	cl_telefono		Numérico	15	0	Si	Telefono del cliente
5	cl_email		Texto	40	0	Si	Email del cliente
6	cl_descuento		Numérico	5	0	No	Descuento del cliente
7	cl_estado		Texto	2	0	No	Estado del cliente

Nombre de la tabla: Usuarios Aplicación: Inventario y Facturación					Descripción de la tabla: Almacena la información de cada uno de los usuarios de la aplicación.		
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	usu_cedula	PK	Numérico	15	0	No	Cedula del usuario
2	usu_nombre		Texto	20	0	No	Nombre del usuario
3	usu_apellido		Texto	20	0	No	Apellido del usuario
4	usu_direccion		Texto	40	0	No	Dirección del usuario
5	usu_telefono		Numérico	15	0	Si	Telefono del usuario
6	usu_email		Texto	40	0	Si	Correo electrónico del usuario
7	usu_estado		Texto	2	0	No	Estado del usuario

8	usu_nomusuario		Texto	40	0	No	Nombre de usuario para la apli
9	usu_contrasena		Texto	40	0	Si	Contraseña del usuario
10	Usu_fechaereg		Date	--	0	No	Fecha de registro del usuario
11	usu_tu_codtipousu	FK	Numérico	10	0	No	Codigo del tipo de usuario

Nombre de la tabla: Tipo_usuarios
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada uno de los tipos de usuarios de la aplicación.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	tu_codtipousu	PK	Numérico	10	0	No	Código del tipo de usuario
2	tu_nomtipousu		Texto	40	0	No	Nombre del tipo de usuario

Nombre de la tabla: Bodegas_productos
Aplicación: Inventario y Facturación

Descripción de la tabla:

Esta tabla nace de una relación de muchos a muchos la cual almacena información de las bodegas asociadas a los productos de ella misma.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	bp_bo_codbodega	PK	Numérico	10	0	No	Código del tipo de usuario
2	bp_pr_codproducto		Texto	40	0	No	Nombre del tipo de usuario

Nombre de la tabla: Proveedores_productos
Aplicación: Inventario y Facturación

Descripción de la tabla:

Esta tabla nace de una relación de muchos a muchos la cual almacena los proveedores a asociados con sus productos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	pp_pv_codproveedor	FK	Numérico	10	0	No	Código del proveedor
2	pp_pr_codproducto	FK	Numérico	10	0	No	Código del producto
3	pp_fecha		Date	--	0	No	Fecha

Nombre de la tabla: Saldos_iniciales
Aplicación: Inventario y Facturación

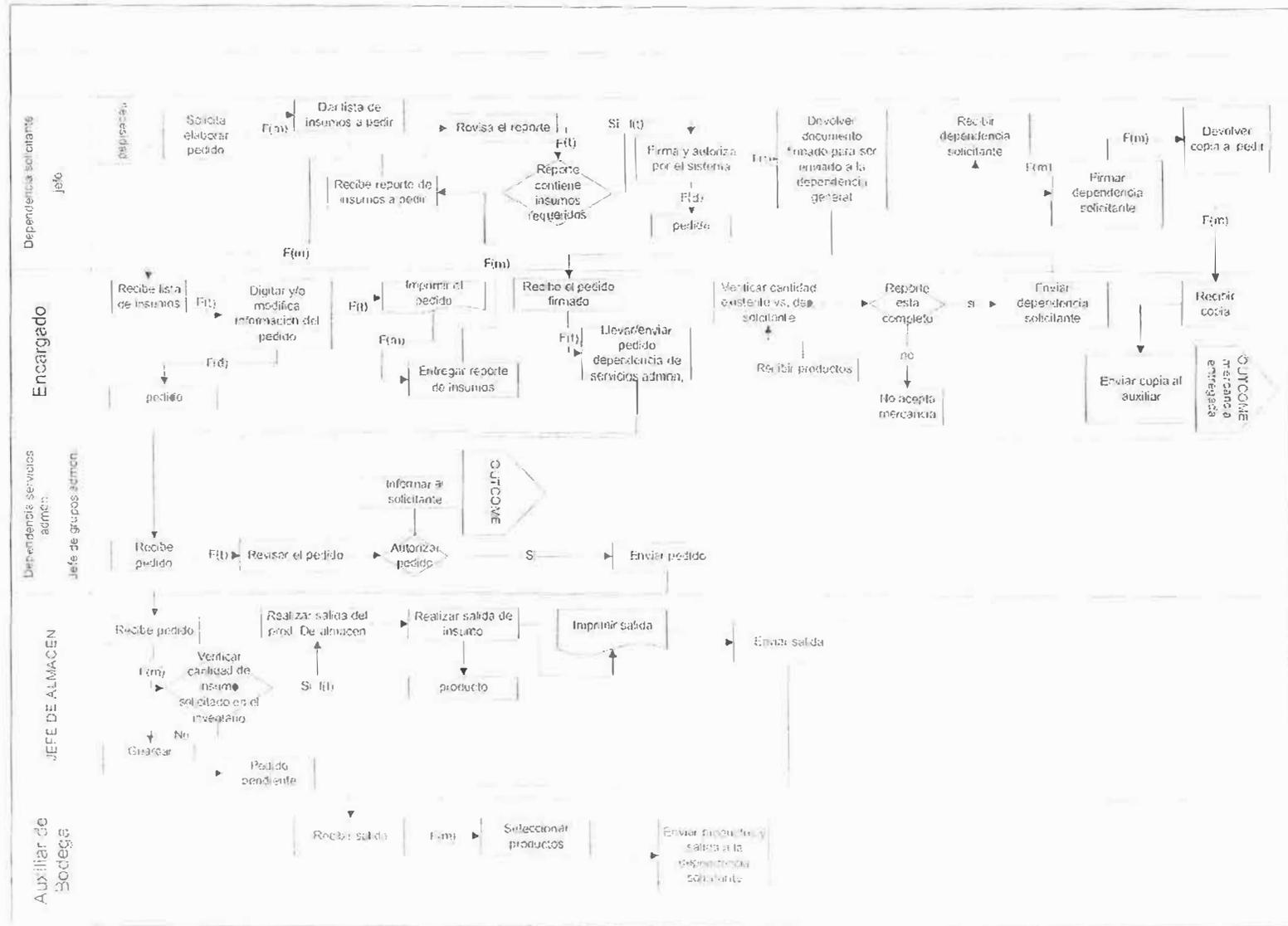
Descripción de la tabla:

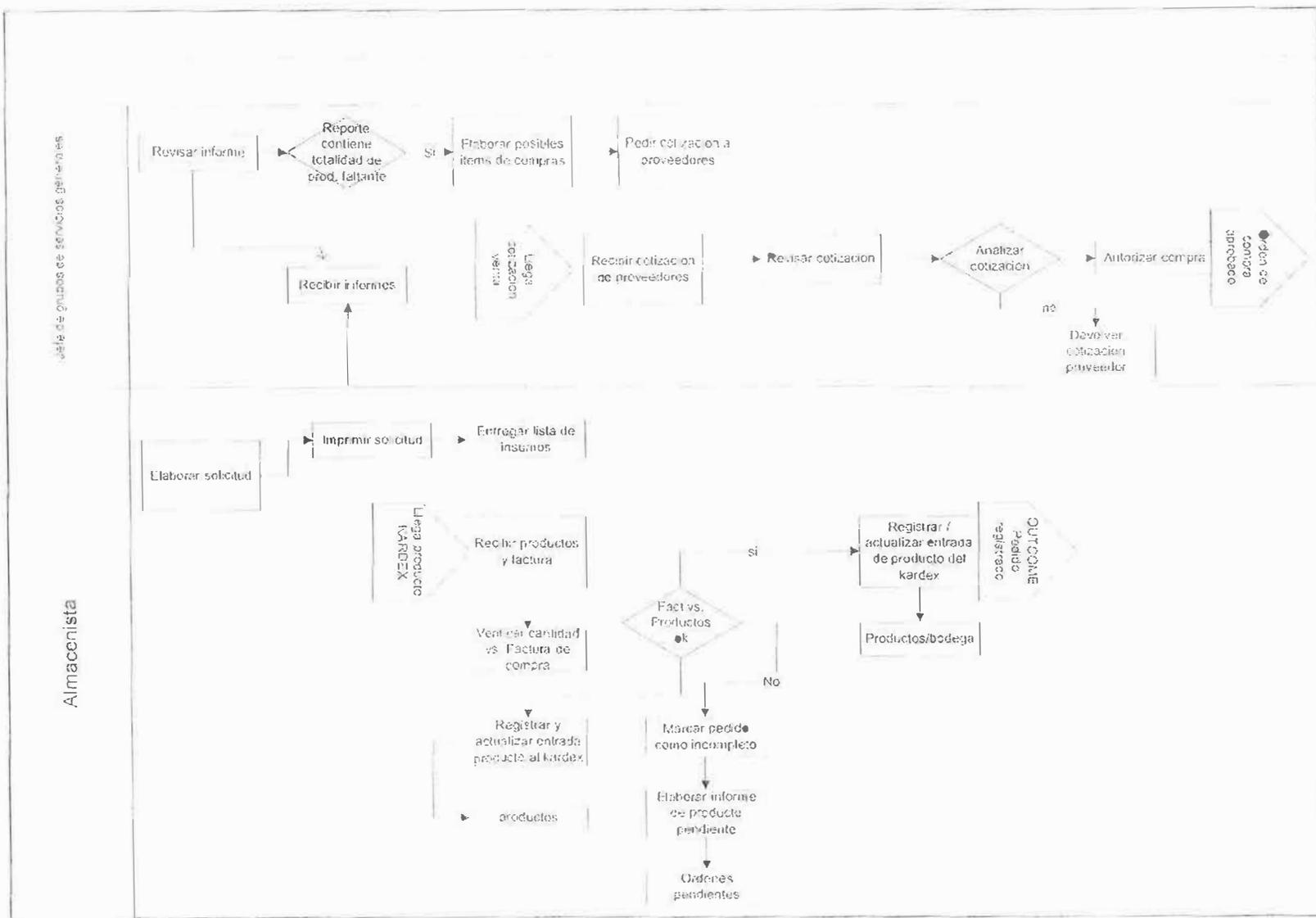
Almacena la información de cada uno de los productos con su respectiva existencia inicial en

el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	si_pr_codproducto	PK	Numérico	10	0	No	Código del producto
2	si_fecha		Numérico	10	0	No	Fecha de la existencia
3	si_existenciainicial		Numérico	10	0	No	Existencia inicial del producto

20. DIAGRAMA FLUJO DEL SISTEMA ACTUAL





21. CASOS DE USO EXPANDIDOS

CASO DE USO:

Referencia del Caso: RF 001

Nombre del Caso: VENTA DE UN PRODUCTO

Definir Actores: VENDEDOR, CAJERO, CLIENTE (init)

Descripción: El cliente solicita productos el vendedor despacha Productos y entrega factura, el cliente se dirige a la caja y Decide su tipo de pago (Contado o crédito). El cajero recibe el pago o la factura y registra la venta El vendedor entrega el producto.

Referencias cruzadas: R2, R3, R9.

Curso Típico de Eventos:

ACCION DE ACTORES

1. El cliente solicita un producto
2. El vendedor despacha los Productos
3. El cliente llega a la caja y decide Como pagar.
4. El cajero recibe pago o factura.
5. El cajero registra la venta
7. El vendedor entrega la mercarí a al Cliente.

RESPUESTA DEL SISTEMA

6. El sistema genera la venta y Descarga del inventario.

Curso Alternativo: Ninguno.

CASO DE USO:

Referencia del Caso: RF 002

Nombre del Caso: COMPRA DE PRODUCTO

Definir Actores: CAJERO, PROVEEDOR (init), ALMACENISTA.

Descripción: El proveedor entrega la mercarí a y factura al almacenista, este Revisa la mercancí a y entrega la factura con el VoBo, el cajero Ingresa datos de la factura en los egresos, descarga el dinero y Lo entrega al proveedor si el pago es de contado, si el pago es a Crédito el cajero descarga la factura en las cuentas por pagar Del sistema y entrega al proveedor la factura que cobrara en Una fecha pospuesta.

Referencias cruzadas: R1.

Curso Típico de Eventos:

ACCION DE ACTORES

1. El proveedor entrega mercarí a y Factura.
2. El almacenista recibe y da VoBo.
3. El cajero ingresa datos de factura

RESPUESTA DEL SISTEMA

4. El sistema genera el egreso y lo

En los egresos (contado) o en las Cuentas por pagar (Crédito).
 5. El cajero cancela factura al proveedor (Contado) ó firma la factura para su próximo cobro (Crédito).

Guarda (Contado) ó descarga en Cuentas por pagar (Crédito).

Curso Alternativo:

Línea 4: El sistema no genera nada, por devolución de mercaríá.

CASO DE USO:

Referencia del Caso: RF 003

Nombre del Caso: MOVIMIENTO DE CUENTAS POR COBRAR.

Definir Actores: CAJERO, CLIENTE (init).

Descripción: El cliente se acerca a la caja para cancelar una factura Pendiente, el cajero busca en el sistema la cuenta del cliente Y cierra esa cuenta por cobrar si el cliente, cancela el valor Total de esta, luego le entrega un recibo constando que la Deuda fue saldada o resta del saldo pendiente, si es un Abono. En cualquiera de los casos, el cajero resta el valor Recibido del movimiento de cuentas por cobrar al Respectivo Cliente y registra el mismo valor en el sistema.

Referencias cruzadas: R3, R4.

Curso Típico de Eventos:

ACCION DE ACTORES

RESPUESTA DEL SISTEMA

- | | |
|---|--|
| 1. El cliente llega a la caja a cancelar Factura pendiente. | |
| 2. El cajero busca cuentas a cobrar Pendientes. | 3. El sistema muestra todas las Cuentas por cobrar. |
| 4. El cajero entrega la factura que Adeuda el cliente. | |
| 5. El cliente recibe factura y cancela la Misma. | |
| 6. El cajero descarga el pago del Movimiento de cuentas por cobrar y Este dicho pago hay que restarlo al Saldo del cliente. | 7. El sistema genera el pago, Actualiza y muestra el nuevo Movimiento de cuentas por cobrar. |
| 8. El cliente recibe su factura ó Constancia de cancelado. | |

Curso Alternativo:

Línea 3: El sistema informa que no contiene cuentas que halla que cobrar.

Línea 7: El sistema no genera ningún movimiento de cuentas por cobrar al no Contener dichas cuentas.

CASO DE USO:**Referencia del Caso:** RF 004**Nombre del Caso:** MOVIMIENTO DE CUENTAS POR PAGAR.**Definir Actores:** CAJERO, PROVEEDOR (init).

Descripción: La fecha de cancelación de una ó varias facturas se vence, y El proveedor cobra el valor de esta. El cajero de contado el Valor total de la factura y resta en el movimiento de cuentas Por pagar del sistema, el valor cierra de esta manera la Cuenta por pagar, si el cajero solo hace un abono se resta el Valor del abono tanto en la factura como en el movimiento De cuentas por pagar del respectivo proveedor en la factura Del sistema.

Referencias cruzadas: R4, R11, R13.**Curso Típico de Eventos:****ACCION DE ACTORES****RESPUESTA DEL SISTEMA**

- | | |
|---|---|
| 1. El proveedor regresa a cobrar
La factura vencida. | |
| 2. El cajero desembolsa el dinero,
Cancela al proveedor. | |
| 3. El cajero descarga el pago que se
hizo de la factura (Total) ó
Descarga el abono realizado. | 4. El sistema resta del saldo que se le
Debe en el movimiento de cuentas
Por pagar al proveedor, actualiza y
Muestra la actualización. |
| 5. Activa impresión del reporte de
Pago, para que el proveedor lo
Firme y quede constancia del pago
Realizado. | 6. El sistema imprime el reporte. |
| 7. El cajero entrega el reporte y el
Proveedor lo firma. | |
| 8. El proveedor recibe una copia. | |

Curso Alternativo:

Línea 4: El sistema envía mensaje de que no existe ningún saldo de ese Proveedor y que se encuentra a paz y salvo.

Línea 6: El sistema no imprime el reporte por daño de la impresora.

CASO DE USO:**Referencia del Caso:** RF 005**Nombre del Caso:** REGISTRO Y CONSULTA DE CLIENTES.**Definir Actores:** ALMACENISTA, CLIENTE (init)

Descripción: El registro del cliente llega al momento de realizar la factura, Se capturan todos sus datos para registrarlos en una base De datos de clientes donde de igual forma si se necesita el Nit u otra característica del cliente simplemente se busca el Nombre del cliente y se consulta todos sus datos.

Referencias cruzadas: R6, R8.

Curso Típico de Eventos:**ACCION DE ACTORES**

- 1.El almacenista crea la factura y
Capta datos generales del cliente.
- 2.Registra en la base de datos al
Cliente y sus descripciones.

RESPUESTA DEL SISTEMA

- 3.El sistema captura ese registro, lo
Guarda y lo genera.

Curso Alternativo:

Línea 3: El sistema informa que ya existe ese cliente, con esas mismas
Descripciones.

CASO DE USO:

Referencia del Caso: RF 006

Nombre del Caso: REPORTE DE PAGOS DE CLIENTES A CREDITO O DE
CONTADO.

Definir Actores: CAJERO, CLIENTE (init)

Descripción: El cliente llega al almacén y se dirige a la caja donde el
Cajero respectivamente le recibe el dinero dando por
Cancelado la factura que obtuvo de compra a crédito que
Realizo en el almacén.

Referencias cruzadas: R12, R14.

Curso Típico de Eventos:**ACCION DE ACTORES**

- 1.El cliente entrega el dinero de la
Compra a crédito o de contado que
Realizó.
- 2.El cajero recibe el dinero, lo cuenta
Y dispone a descargar el pago.
3. El cajero modifica los pagos
Realizados por el cliente y actualiza
El reporte en el sistema.
- 5.El cajero llama en el sistema el
Nuevo reporte.
7. El cajero verifica si esta bien el
Nuevo reporte y pide al sistema
Que lo imprima.
9. El cajero entrega al cliente el
Nuevo reporte, para que el cliente
Tenga constancia de sus pagos
Realizados al almacén.

RESPUESTA DEL SISTEMA

- 4.El sistema actualiza y genera el
Nuevo reporte.
- 6.El sistema muestra el nuevo
Reporte.
8. El sistema imprime el nuevo
Reporte.

Curso Alternativo:

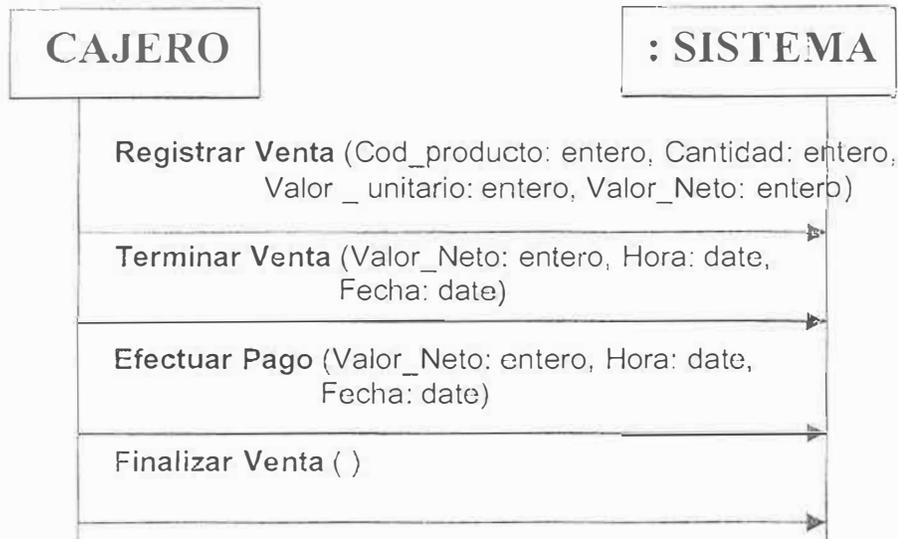
Línea 8: El sistema informa que no se puede imprimir el reporte por fallas en la
Impresora.

Línea 4: El sistema informa que no puede generar el reporte debido a
Inconsistencias en los datos del cliente.

22. DIAGRAMA DE SECUENCIA

Referencia del Caso: RF 001

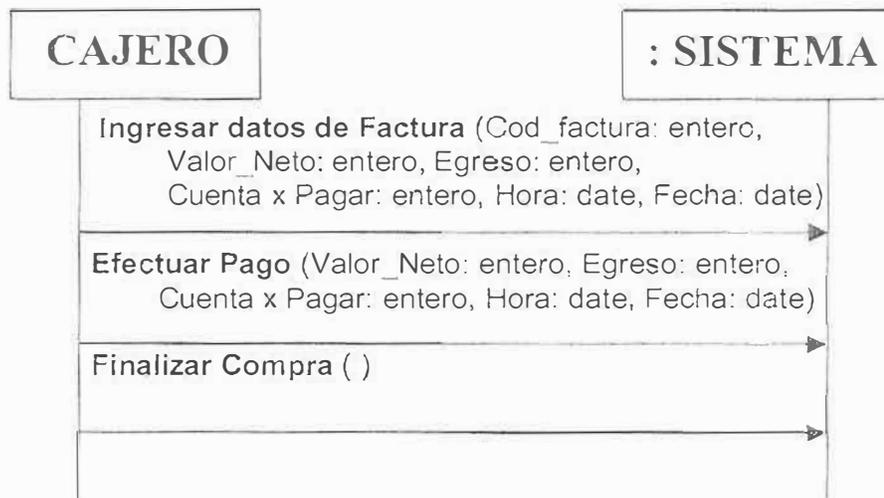
Nombre del Caso: VENTA DE UN PRODUCTO.



Curso Alternativo: Ninguno.

Referencia del Caso: RF 002

Nombre del Caso: COMPRA DE PRODUCTO.

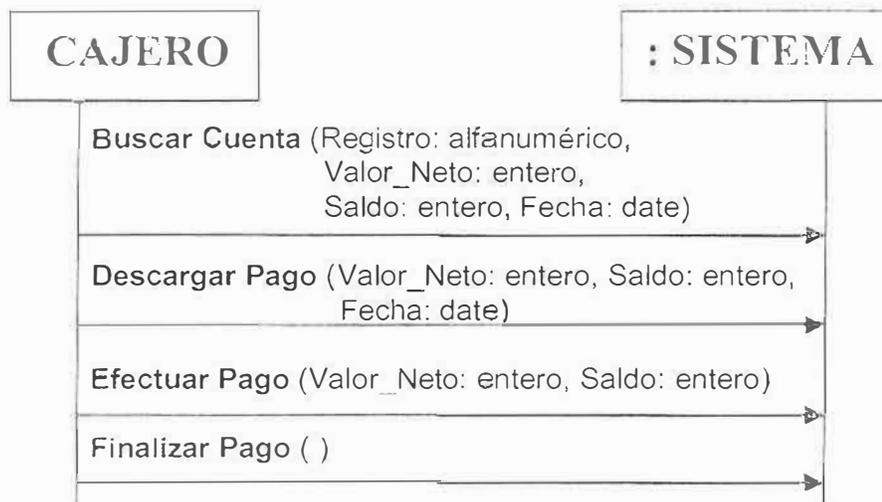


Curso Alternativo:

Línea 4: El sistema no genera nada, por devolución de mercancía.

Referencia del Caso: RF 003

Nombre del Caso: MOVIMIENTO DE CUENTAS POR COBRAR.



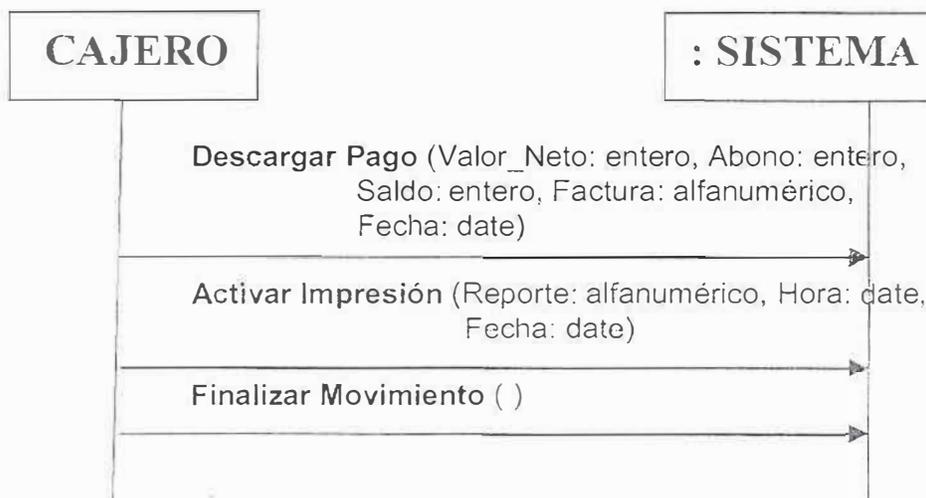
Curso Alternativo:

Línea 3: El sistema informa que no contiene cuentas que halla que cobrar.

Línea 7: El sistema no genera ningún movimiento de cuentas por cobrar al no Contener dichas cuentas.

Referencia del Caso: RF 004

Nombre del Caso: MOVIMIENTO DE CUENTAS POR PAGAR.



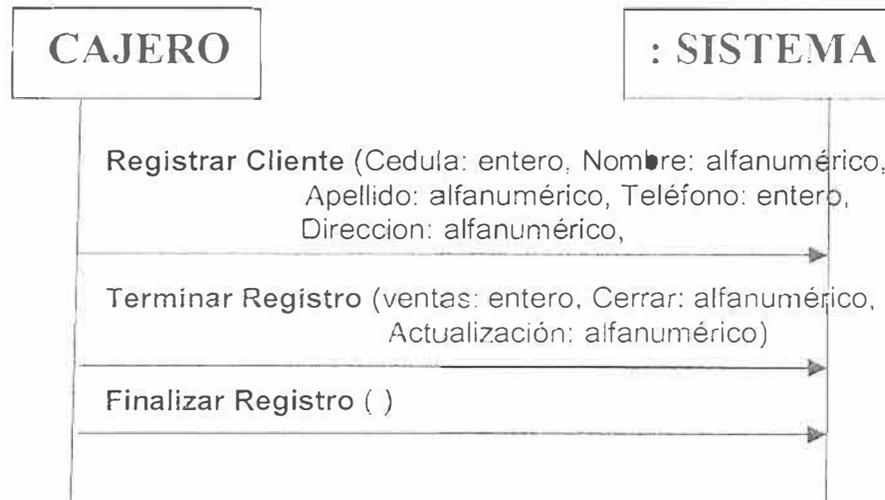
Curso Alternativo:

Línea 4: El sistema envía mensaje de que no existe ninguna deuda para ese Proveedor y que se encuentra el negocio a paz y salvo.

Línea 6: El sistema no imprime el reporte por daño de la impresión.

Referencia del Caso: RF 005

Nombre del Caso: REGISTRO Y CONSULTA DE CLIENTES.

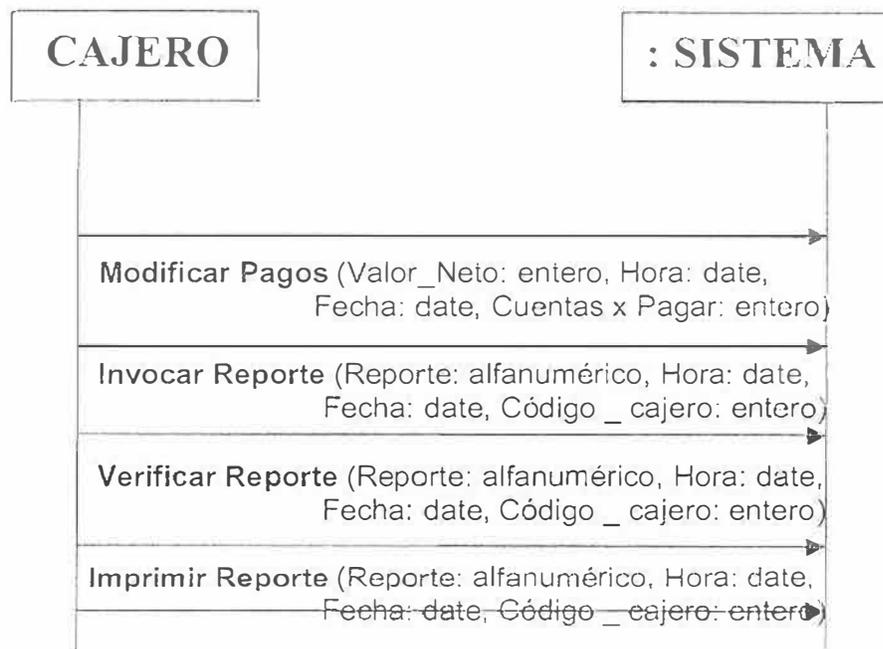


Curso Alternativo:

Línea 3: El sistema informa que ya existe ese cliente, con esas mismas Descripciones.

Referencia del Caso: RF 006

Nombre del Caso: REPORTE DE PAGOS DE CLIENTES A CREDITO.



Finalizar Movimiento ()

Curso Alternativo:

Línea 4: El sistema informa que no puede generar el reporte debido a Incoherencias en los datos del cliente.

Línea 8: El sistema informa que no se puede imprimir el reporte por fallas en la Impresión.

23. BIBLIOGRAFIA

Gudiño, Emma Lucia y Coral, Lucy del C. Contabilidad 2000 plus. Quinta edición. Bogotá D.C. Mac Graw Hill, 2005. p. 144 – 168.

MONOGRAFÍAS.COM. Documentación contable

<http://www.monografias.com/trabajos11/ladocont/ladocont.shtml>

(Consulta : 5 de Noviembre de 2005).

REGLAMENTO GENERAL DE LA CONTABILIDAD. Decreto Reglamentario 2649 de 1993. p. 62-63.

GOOGLE.COM. Modelo lineal secuencial

<http://www.google.com/definicion/cascada/modelo.html>

(Consulta 25 de Octubre de 2005).

MODULOS DE INVENTARIO Y FACTURACION EN DISTRIFAGAR LTDA.

LINEA DE INVESTIGACION: INGENIERIA DE SOFTWARE

ALEXANDER JOSE FARIAS PEREZ

JAVIER SANCHEZ MUÑOZ

CARLOS HUMBERTO GAMBOA HERNANDEZ

GEOVANI FABIAN VAZQUEZ RADA

UNIVERSIDAD SIMON BOLIVAR

FACULTAD DE INGENIERIA DE SISTEMAS

AREA DE FORMATIVA IV

UNDECIMO SEMESTRE A-NOCTURNO

BARRANQUILLA

2007

MANUAL DEL SISTEMA PARA EL SOFTWARE

MINFA

CONTENIDO

INTRODUCCION	3
1. OBJETIVO	4
1.1 Objetivo general	4
1.2 Objetivos específicos	4
2. INSTALACIÓN	5
2.1 Hardware Requerimientos mínimos	5
2.2 Software	5
2.3 Instalación en el Servidor de Bases de Datos	5
2.3.1 Instalación de MySQL Servers and Clients 5.0.1. - Alpha	6
2.4 Instalar MINFA	8
3. ADMINISTRACIÓN DE USUARIOS	8
4. DEFINICION DE LAS BASES DE DATOS	9
4.1 Tablas del Sistema	9
4.2 Listado de Tablas	9
4.3 Script Para La Creacion De La Base De Datos	10
4.4 Diccionario De Datos Del Modelo Relacional	15
4.5 Modelo Relacional	20
4.6 Diagrama Flujo Del Sistema Actual	21
5. EJECUCION DE LA CODIFICACION	23

INTRODUCCION

Este Manual está dirigido al personal técnico responsable de instalar inicialmente MINFA; por tanto se da por hecho que quien tenga acceso a este documento está familiarizado con ciertos conceptos básicos de administración de usuarios para la aplicación de los conocimientos explicados aquí, posee un usuario Windows que tiene los permisos adecuados (Un usuario Administrador de Windows).

MINFA es un software diseñado para llevar un control sistematizado de las entradas y salidas de productos en su negocio, por los conceptos de compra, venta de productos; también facilita la buena administración de la locación de sus productos en bodega y permite de manera rápida y segura hacer consultas. Sugerimos que su instalación cuente con la supervisión de personal técnico.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Crear un sistema de inventario y facturación para Distrifagar Ltda.

1.2 OBJETIVOS ESPECIFICOS

- Construir una herramienta que administre la información de inventario y facturación.
 - Facilitar el manejo del sistema diseñando una interfaz de fácil manejo para el usuario final.
 - Permitir generar reportes actualizados de manera fácil y oportuna.
 - Aumentar el nivel de seguridad y confiabilidad de la información
 - Optimizar el procesamiento, almacenamiento y recuperación de la información de inventario y facturación.
 - Facilitar el acceso a información de periodos (días, meses) anteriores.
-

2. INSTALACION

2.1 Hardware Requerimientos mínimos:

PROCEDADOR: 333mhz

RAM: 128MB

DISCO DURO: 10GB

2.2 Software.

Sistemas Operativos: Win9X, Windows 2000/XP/Vista

Lenguaje de Desarrollo: Visual Basic 6.0, Cristal Report

Base de Datos: MySql Servers and Clients 5.0.1. - alpha

Modelo de acceso a Base de Datos: Cliente / Servidor

Conexión a Base de Datos: MySql ODBC 3.51 Driver DSN

Este esquema requiere que en el servidor esté instalado el respectivo motor de Base de Datos a usar, y que en el equipo cliente estén instalado el cliente del motor de Base de Datos. El aplicativo de MINFA puede estar instalado en el equipo cliente o sobre un equipo remoto; pero debe ser ejecutado en el cliente.

El diseño de MINFA se ciño a una metodología de portabilidad para aprovechar que el aplicativo soporte diferentes motores de Bases de Datos basados en SQL. En ese orden de ideas, usted puede escoger el motor que más le convenga, que entre otras incluyen controladores nativos para versiones de Oracle, SQL Server, DB2, Interbase y que además soporta conectividad ODBC, ampliando por tanto la gama de motores soportados.

2.3 Instalación en el Servidor de Bases de Datos

La instalación de MINFA en el servidor requiere una copia de las Bases de Datos de MINFA o simplemente correr un Script de la misma y la instalación del motor de Base de Datos en que estas se encuentra originalmente, es decir MySql Servers and Clients 5.0.1. - alpha.

2.3.1 Instalación de MySQL Servers and Clients 5.0.1. - Alpha.

Busque en el CD el directorio MySQL 5.0.1.\mysql-5.0.1-alpha-snapshot-win, dentro de este subdirectorio hay un archivo llamado SETUP.exe. Ejecútelo y siga los pasos de la instalación.

Después de la instalación de la Base de Datos reinicie el equipo para que Windows inicie el servicio de MySQL y se puedan generar las conexiones a la Base de Datos.

El aplicativo MINFA se comunica con la Base de Datos mediante un intermediario conocido como ODBC, el cual realiza la conexión con MySQL Servers and clients 5.0.1 – alpha, con un origen de base de datos de MySQL ODBC 3.51 Driver DSN como muestra la Figura No 1.

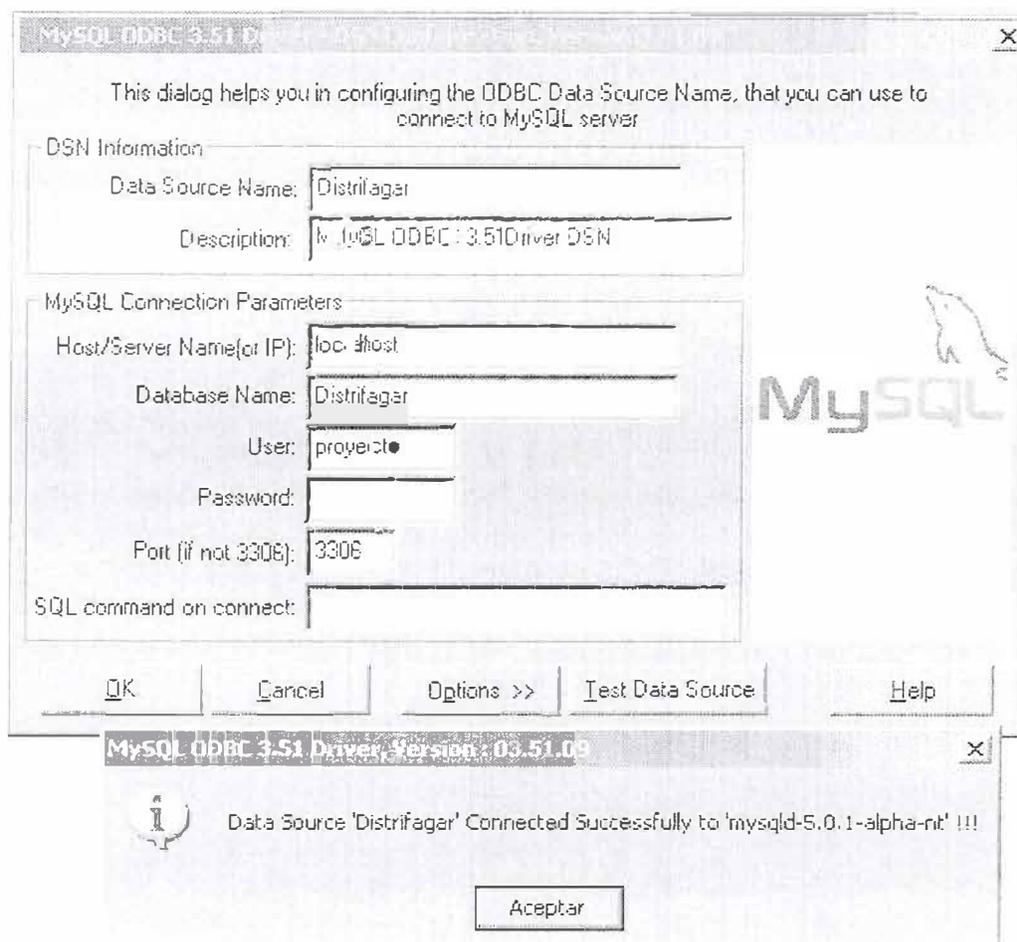


Figura No. 1: Comunicación entre aplicativo MINFA y Base de Datos MySQL Servers and Clients 5.0.1. – alpha, a través del ODBC.

Posteriormente a la instalación de la Base de Datos se instala el controlador

ODBC (es el que permite la conexión del aplicativo con la Base de Datos), el cual se encuentra en el CD, en el directorio MySQL 5.0.1., dentro de este subdirectorio hay un archivo llamado MyODBC-standard-3.51.9-win.exe. Ejecútelo y siga los pasos de Instalación.

Y por ultimo para un mejor manejo de la Base de Datos con una interface grafica se instala la consola de Base de Datos SQLyog51, la cual se encuentra en el CD, en el directorio MySQL 5.0.1., dentro de este subdirectorio hay un archivo llamado SQLyog512.exe. Ejecútelo y siga los pasos de instalación. (Opcional)

Esta última le facilita y agiliza la respectiva creación y modificación de la base de datos pues viene en ambiente Windows, ver Figura No 2.

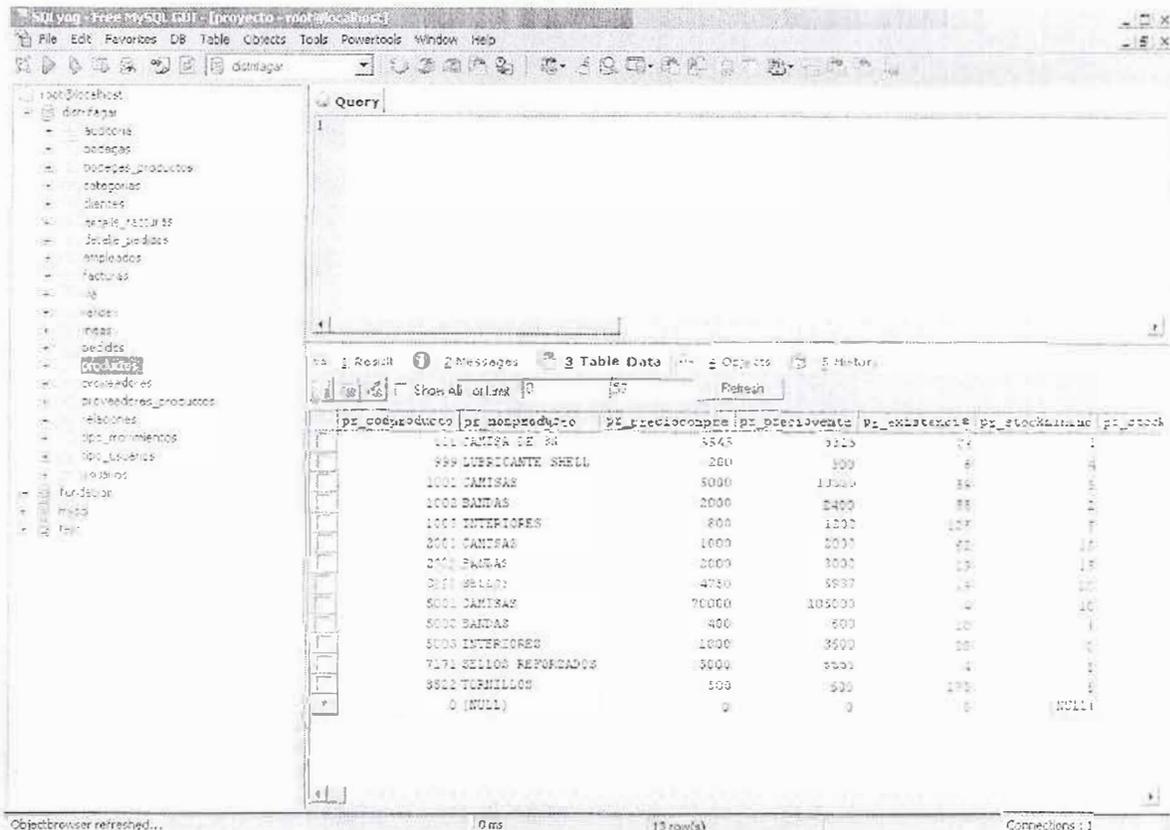


Figura No. 2: Consola grafica de MySQL para mejor manejo de las operaciones en la base de datos

El instalador MINFA coloca el aplicativo en el equipo cliente; pero en sistemas de red, es frecuente que se prefiera mantener el ejecutable en un servidor de aplicaciones debido a que la actualización del aplicativo sólo se realiza una vez sobre dicho servidor, y no sobre cada equipo cliente. Para esta estrategia administrativa de aplicativos dirigidos a un servidor,

MINFA.exe se instala en el menú de programas, para dar acceso a sus diferentes módulos.

Luego, solo deberá re direccionar manualmente el enlace del aplicativo de MINFA para que este sea tomado del servidor de aplicaciones.

2.4 Instalar MINFA

Para instalar MINFA en el cliente se deben realizar los siguientes pasos:

Busque el directorio MINFA del CD de Instalación y ejecute el programa SETUP.exe, Siga los pasos que ahí aparecen: seguidamente es necesario configurar el ODBC, recuerde que el ODBC es el mecanismo usado por MINFA para comunicarse con la Base de Datos y por ultimo re dirija el ejecutable de el aplicativo al servidor de aplicaciones si este es el caso.

3. ADMINISTRACIÓN DE USUARIOS

El esquema de seguridad de MINFA maneja las conexiones al Aplicativo en dos formas distintas, dependiendo del usuario y el módulo que se ejecute.

Existen varias categorías de usuarios MINFA dependiendo del perfil que tenga el empleado en la empresa: **Usuarios Administradores**, **Usuarios Digitadores**. Los usuarios administradores de MINFA tienen acceso a todos los módulos del aplicativo, y tienen todos los privilegios. Los usuarios digitadores solo realizan labores de digitaciones dependiendo al rol que le asignen.

4. DEFINICION DE LAS BASES DE DATOS

En la elaboración de este software se utiliza **MySQL**, el cual es un sistema de gestión de Base de Datos multiusuario, **MySQL** es un software libre.

4.1 Tablas Del Sistema.

En el proceso de la realización del sistema MINFA fue necesario utilizar un número determinado de tablas que permitiera almacenar y organizar la información. Las tablas que se utilizan para este fin son las siguientes: Auditoria, bodegas, bodegas_productos, categorías, clientes, detalle_facturas, detalle_pedidos, empleados, facturas, iva, kardex, líneas, pedidos, productos, proveedores, proveedores_productos, tipo_movimientos, tipo_usuarios, usuarios.

4.2 Listado De Tablas.

Tablas	Definición
Auditoria	Almacena información de los movimientos generados en la aplicación ya sean inserciones o modificaciones de los datos.
Bodegas	Almacena información de las bodegas de almacenamiento de los productos del almacén.
Bodegas_productos	Almacena las llaves primarias de las tabla bodegas y productos las cuales Tienen una relación de muchos a muchos.
Categorías	Almacena información de las categorías de los productos.
Clientes	Almacena información de los clientes del almacén.
Detalle_facturas	Almacena el detalle de las facturas: Código de la factura, código del producto, valor unitario, cantidad y valor total por cantidad de productos.
Detalle_pedidos	La estructura de esta tabla es parecida a la de detalle_facturas, pero en este caso se almacenan los detalles de los pedidos.
Empleados	Almacena información de los empleados del almacén.
Facturas	Almacena información de las facturas con el total de las mismas.
Iva	Almacena información de los distintos valores de iva que maneje el almacén.
Kardex	Almacena información de los movimientos de entrada y salida de productos al almacén.
Líneas	Almacena información de las líneas de los productos.
Pedidos	Almacena información de los pedidos con el total de los mismos.
Productos	Almacena información de los productos que provee el almacén a los clientes.
Proveedores	Almacena información de los proveedores que surten los productos al almacén.
Proveedores_productos	Almacena las llaves primarias de las tabla proveedores y productos las cuales Tienen una relación de muchos a muchos.
Tipo_movimientos	Almacena información los tipos de movimientos que se generan a nivel de

	Producción.
Tipo_usuarios	Almacena información de los tipos de usuarios del aplicativo.
Usuarios	Almacena información de los usuarios del aplicativo.

4.3 SCRIPT PARA LA CREACION DE LA BASE DE DATOS.

```
CREATE DATABASE IF NOT EXISTS `distrifagar`;
USE `distrifagar`;
```

```
CREATE TABLE `auditoria` (
  `au_codigo` int(10) NOT NULL auto_increment,
  `au_operacion` varchar(20) default NULL,
  `au_tabla` varchar(50) default NULL,
  `au_fecha` date default NULL,
  `au_hora` time default NULL,
  `us_nombre` varchar(50) default NULL,
  PRIMARY KEY (`au_codigo`),
  KEY `FK_auditoria` (`us_nombre`),
  CONSTRAINT `auditoria_ibfk_1` FOREIGN KEY (`us_nombre`)
REFERENCES `usuarios` (`us_nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `bodegas` (
  `bo_codbodega` int(10) NOT NULL default '0',
  `bo_observacion` varchar(40) default NULL,
  `bo_cupo` int(10) default NULL,
  `bo_estado` int(2) default NULL,
  PRIMARY KEY (`bo_codbodega`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `bodegas_productos` (
  `bo_codbodega` int(10) default NULL,
  `pr_codproducto` int(10) default NULL,
  KEY `FK_bodegas_productos_pr` (`pr_codproducto`),
  KEY `FK_bodegas_productos_bo` (`bo_codbodega`),
  CONSTRAINT `bodegas_productos_ibfk_1` FOREIGN KEY
(`bo_codbodega`) REFERENCES `bodegas` (`bo_codbodega`),
  CONSTRAINT `bodegas_productos_ibfk_2` FOREIGN KEY
(`pr_codproducto`) REFERENCES `productos` (`pr_codproducto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `categorias` (
```

```

`ca_codcategoria` int(10) NOT NULL default '0',
`ca_nomcategoria` varchar(40) default NULL,
`li_codlinea` int(10) NOT NULL default '0',
PRIMARY KEY (`ca_codcategoria`,`li_codlinea`),
UNIQUE KEY `ca_codcategoria` (`ca_codcategoria`),
KEY `FK_categorias_lineas` (`li_codlinea`),
CONSTRAINT `categorias_ibfk_1` FOREIGN KEY (`li_codlinea`)
REFERENCES `lineas` (`li_codlinea`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `clientes` (
  `cl_idcliente` int(15) NOT NULL default '0',
  `cl_nomcliente` varchar(40) default NULL,
  `cl_direccion` varchar(40) default NULL,
  `cl_telefono` varchar(40) default NULL,
  `cl_email` varchar(40) default NULL,
  `cl_descuento` int(10) default NULL,
  `cl_estado` int(2) default NULL,
  PRIMARY KEY (`cl_idcliente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `detalle_facturas` (
  `fa_codfactura` int(10) default '0',
  `pr_codproducto` int(10) default NULL,
  `de_valorunitario` int(10) default NULL,
  `de_cantidad` int(10) default NULL,
  `de_total` int(10) default NULL,
  KEY `FK_detalle_facturas` (`pr_codproducto`),
  KEY `FK_detalle_factura` (`fa_codfactura`),
  CONSTRAINT `detalle_facturas_ibfk_1` FOREIGN KEY (`fa_codfactura`)
REFERENCES `facturas` (`fa_codfactura`),
  CONSTRAINT `detalle_facturas_ibfk_2` FOREIGN KEY (`pr_codproducto`)
REFERENCES `productos` (`pr_codproducto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `detalle_pedidos` (
  `pe_codpedido` varchar(20) default NULL,
  `pr_codproducto` int(10) default NULL,
  `de_valorunitario` int(10) default NULL,
  `de_cantidad` int(10) default NULL,
  `de_total` int(10) default NULL,
  KEY `detalle_pedidos_ibfk_1` (`pe_codpedido`),
  KEY `detalle_pedidos_ibfk_2` (`pr_codproducto`),

```

```

    CONSTRAINT `detalle_pedidos_ibfk_1` FOREIGN KEY (`pe_codpedido`)
REFERENCES `pedidos` (`pe_codpedido`),
    CONSTRAINT `detalle_pedidos_ibfk_2` FOREIGN KEY (`pr_codproducto`)
REFERENCES `productos` (`pr_codproducto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `empleados` (
  `em_cedula` int(15) NOT NULL default '0',
  `em_nombre` varchar(40) default NULL,
  `em_apellido` varchar(40) default NULL,
  `em_direccion` varchar(40) default NULL,
  `em_telefono` varchar(40) default NULL,
  `em_email` varchar(40) default NULL,
  `em_fechareg` date default NULL,
  PRIMARY KEY (`em_cedula`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `facturas` (
  `fa_codfactura` int(10) NOT NULL default '0',
  `fa_fecha` date default NULL,
  `fa_subtotal` int(10) default NULL,
  `fa_iva` int(5) default NULL,
  `fa_total` int(10) default NULL,
  `fa_descuento` int(10) default NULL,
  `fa_estado` int(2) default NULL,
  `cl_idcliente` int(15) default NULL,
  `us_nombre` varchar(20) default NULL,
  PRIMARY KEY (`fa_codfactura`),
  KEY `FK_facturas_cli` (`cl_idcliente`),
  KEY `FK_facturas_em` (`us_nombre`),
  CONSTRAINT `facturas_ibfk_2` FOREIGN KEY (`us_nombre`)
REFERENCES `usuarios` (`us_nombre`),
  CONSTRAINT `facturas_ibfk_3` FOREIGN KEY (`cl_idcliente`)
REFERENCES `clientes` (`cl_idcliente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `iva` (
  `iva_codiva` int(5) NOT NULL default '0',
  `iva_valoriva` int(5) default NULL,
  PRIMARY KEY (`iva_codiva`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `kardex` (
  `kd_consecutivo` int(10) NOT NULL auto_increment,

```

```

`kd_fecha` date default NULL,
`kd_cantidad` int(10) default NULL,
`kd_precio` int(10) default NULL,
`pr_codproducto` int(10) default NULL,
`tm_codtipomov` int(10) default NULL,
`us_nombre` varchar(20) default NULL,
PRIMARY KEY (`kd_consecutivo`),
KEY `FK_kardex` (`pr_codproducto`),
KEY `FK_kardex_em` (`us_nombre`),
KEY `FK_kardex_mov` (`tm_codtipomov`),
CONSTRAINT `kardex_ibfk_1` FOREIGN KEY (`pr_codproducto`)
REFERENCES `productos` (`pr_codproducto`),
CONSTRAINT `kardex_ibfk_2` FOREIGN KEY (`tm_codtipomov`)
REFERENCES `tipo_movimientos` (`tm_codtipomov`),
CONSTRAINT `kardex_ibfk_3` FOREIGN KEY (`us_nombre`)
REFERENCES `usuarios` (`us_nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `líneas` (
  `li_codlinea` int(10) NOT NULL default '0',
  `li_nomlinea` varchar(40) default NULL,
  PRIMARY KEY (`li_codlinea`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `pedidos` (
  `pe_codpedido` varchar(20) NOT NULL default "",
  `pe_fecha` date default NULL,
  `pe_subtotal` int(20) default NULL,
  `pe_iva` int(20) default NULL,
  `pe_total` int(20) default NULL,
  `pe_estado` int(2) default NULL,
  `pv_codproveedor` int(10) default NULL,
  `us_nombre` varchar(20) default NULL,
  PRIMARY KEY (`pe_codpedido`),
  KEY `pedidos_ibfk_1` (`pv_codproveedor`),
  KEY `pedidos_ibfk_2` (`us_nombre`),
  CONSTRAINT `pedidos_ibfk_1` FOREIGN KEY (`pv_codproveedor`)
REFERENCES `proveedores` (`pv_codproveedor`),
  CONSTRAINT `pedidos_ibfk_2` FOREIGN KEY (`us_nombre`)
REFERENCES `usuarios` (`us_nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `productos` (

```

```

`pr_codproducto` int(10) NOT NULL default '0',
`pr_nomproducto` varchar(40) default NULL,
`pr_preciocompra` int(10) default '0',
`pr_precioventa` int(10) default '0',
`pr_existencia` int(10) default '0',
`pr_stockminimo` int(10) default NULL,
`pr_stockmaximo` int(10) default NULL,
`iva_codiva` int(5) default NULL,
`pr_estado` int(2) default NULL,
`li_codlinea` int(10) default NULL,
`ca_codcategoria` int(10) default NULL,
PRIMARY KEY (`pr_codproducto`),
KEY `FK_productos_li` (`li_codlinea`),
KEY `FK_productos_iva` (`iva_codiva`),
KEY `FK_productos` (`ca_codcategoria`,`li_codlinea`),
CONSTRAINT `productos_ibfk_1` FOREIGN KEY (`ca_codcategoria`,`li_codlinea`) REFERENCES `categorias` (`ca_codcategoria`,`li_codlinea`),
CONSTRAINT `productos_ibfk_2` FOREIGN KEY (`iva_codiva`) REFERENCES `iva` (`iva_codiva`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `proveedores` (
`pv_codproveedor` int(10) NOT NULL default '0',
`pv_nomproveedor` varchar(40) default NULL,
`pv_direccion` varchar(40) default NULL,
`pv_telefono` varchar(40) default NULL,
`pv_email` varchar(40) default NULL,
`pv_contacto` varchar(40) default NULL,
`pv_estado` int(2) default NULL,
PRIMARY KEY (`pv_codproveedor`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `proveedores_productos` (
`pv_codproveedor` int(10) default NULL,
`pr_codproducto` int(10) default NULL,
KEY `FK_proveedores_productos_pro` (`pv_codproveedor`),
KEY `FK_proveedores_productos` (`pr_codproducto`),
CONSTRAINT `proveedores_productos_ibfk_1` FOREIGN KEY (`pr_codproducto`) REFERENCES `productos` (`pr_codproducto`),
CONSTRAINT `proveedores_productos_ibfk_2` FOREIGN KEY (`pv_codproveedor`) REFERENCES `proveedores` (`pv_codproveedor`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```
CREATE TABLE `tipo_movimientos` (
  `tm_codtipomov` int(10) NOT NULL default '0',
  `tm_nommov` varchar(50) default NULL,
  PRIMARY KEY (`tm_codtipomov`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `tipo_usuarios` (
  `tu_codigo` int(10) NOT NULL default '0',
  `tu_descripcion` varchar(50) default NULL,
  PRIMARY KEY (`tu_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `usuarios` (
  `us_nombre` varchar(20) NOT NULL default "",
  `us_clave` varchar(20) default NULL,
  `tu_codigo` int(10) default NULL,
  `em_cedula` int(15) NOT NULL default '0',
  `us_estado` int(2) default NULL,
  PRIMARY KEY (`us_nombre`,`em_cedula`),
  KEY `FK_usuarios` (`tu_codigo`),
  KEY `FK_usuarios_emp` (`em_cedula`),
  CONSTRAINT `usuarios_ibfk_1` FOREIGN KEY (`tu_codigo`)
REFERENCES `tipo_usuarios` (`tu_codigo`),
  CONSTRAINT `usuarios_ibfk_2` FOREIGN KEY (`em_cedula`)
REFERENCES `empleados` (`em_cedula`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.4 DICcionario DE DATOS DEL MODELO RELACIONAL.

Nombre de la tabla: Productos Aplicación: Inventario y Facturación					Descripción de la tabla:		
					Almacena la información de cada uno de los productos que ofrece el almacén de repuestos.		
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	pr_codproducto	PK	Numérico	10	0	No	Código del producto
2	pr_nomproducto		Texto	40	0	No	Nombre del producto
3	pr_preciocompra		Numérico	10	0	No	Precio de compra
4	pr_precioventa		Numérico	10	0	No	Precio de venta
5	pr_existencia		Numérico	10	0	No	Existencia actual del producto
6	pr_IVA		Numérico	5	0	No	IVA del producto
7	pr_stockminimo		Numérico	10	0	No	Stock mínimo del producto

8	pr_stockmaximo		Numérico	10	0	No	Stock máximo del producto
9	pr_descuento		Numérico	5	0	No	Descuento del producto
10	pr_estado		Texto	2	0	No	Estado del producto
11	pr_li_codlinea	FK	Numérico	10	0	No	Código de línea del producto
12	pr_li_ca_codcategoria	FK	Numérico	10	0	No	Código de la categoría del P.

Nombre de la tabla: Proveedores
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada uno de los proveedores que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	pv_codproveedor	PK	Numérico	10	0	No	Código del proveedor
2	pv_nomproveedor		Texto	40	0	No	Nombre del proveedor
3	pv_Dirección		Texto	40	0	No	Dirección del proveedor
4	pv_telefono		Numérico	15	0	Si	Teléfono del proveedor
5	pv_email		Texto	40	0	Si	Email del proveedor
6	pv_contacto		Texto	40	0	Si	Contacto del proveedor
7	pv_estado		Texto	2	0	No	Estado del proveedor

Nombre de la tabla: Bodegas
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las bodegas que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	bo_ccdbodega	PK	Numérico	10	0	No	Código de la bodega
2	bo_observacion		Texto	40	0	No	Observación de la bodega
3	bo_estado		Texto	2	0	No	Estado de la Bodega
4	bo_su_codsucursal	FK	Numérico	10	0	No	Código de la sucursal

Nombre de la tabla: Categorías
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las categorías de los productos que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	ca_codcategoria	PK	Numérico	10	0	No	Código de la categoría
2	ca_nomcategoria		Texto	40	0	No	Nombre de la categoría

Nombre de la tabla: Líneas
Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las líneas de productos que tiene el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	li_codlinea	PK	Numérico	10	0	No	Código de la línea
2	li_nomlinea		Texto	40	0	No	Nombre de la línea
3	li_ca_codcategoria	FK	Numérico	10	0	No	Código de la categoría

Nombre de la tabla: Kardex

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las transacciones que se efectúa en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	kd_consecutivo	PK	Numérico	10	0	No	Código consecutivo del móv.
2	kd_fecha		Date	--	0	No	Fecha del movimiento
3	Kd_hora		time	--	0	No	Hora del movimiento
4	kd_cantidad		Numérico	10	0	No	Cantidad de productos
5	kd_descripcion		Texto	40	0	Si	Descripción del movimiento
6	kd_pr_codproducto	FK	Numérico	10	0	No	Código del producto
7	kd_tm_tipomov	FK	Numérico	10	0	No	Código del tipo de movimiento
8	tm_codsubtipomov	FK	Numérico	10	0	No	Código del subtipo de móv.
9	kd_precio		Numérico	10	0	No	Precio del producto
10	Kd_usu_cedula	FK	Numérico	15	0	No	Cedula del usuario

Nombre de la tabla: Tipo_Movimientos

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada uno de los tipos de movimientos que se pueden efectuar en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	tm_codtipomov	PK	Numérico	10	0	No	Código del tipo de movimiento
2	tm_codsubtipomov	PK	Numérico	10	0	No	Código del subtipo de móv.
3	tm_nommov		Texto	40	0	No	Nombre del movimiento

Nombre de la tabla: Facturas

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada una de las facturas efectuadas en el almacén de repuestos.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	fa_codfactura	PK	Numérico	10	0	No	Código de la factura
2	fa_fecha		Date	--	0	No	Fecha de la factura
3	fa_subtotal		Numérico	10	0	No	Subtotal del valor de la factura
4	fa_iva		Numerico	--	0	No	IVA de valor total de la factura

5	fa_total		Numérico	10	0	No	Valor total de la factura
6	fa_descuento		Numérico	5	0	No	Descuento de la factura
7	fa_estado		Numérico	10	0	No	Estado de la factura
8	fa_cl_IDcliente	FK	Numérico	15	0	No	Identificación del cliente
9	fa_usu_cedula	FK	Numérico	15	0	No	Cedula de los usuarios

Nombre de la tabla: Detalle_facturas					Descripción de la tabla:		
Aplicación: Inventario y Facturación					Almacena la información de cada uno de los detalles de las facturas que se efectúan en el almacén de repuestos.		

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	de_fa_codfactura		Numérico	10	0	No	Código de la factura
2	de_pr_codproducto		Numérico	10	0	No	Código del producto
3	de_valorunitario		Numérico	10	0	No	Valor unitario del producto
4	de_cantidad		Numérico	10	0	No	Cantidad de productos

Nombre de la tabla: Clientes					Descripción de la tabla:		
Aplicación: Inventario y Facturación					Almacena la información de cada uno de los clientes del almacén de repuestos.		

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	cl_IDcliente	PK	Numérico	15	0	No	Identificación del cliente
2	cl_nomcliente		Texto	40	0	No	Nombre del cliente
3	cl_direccion		Texto	40	0	No	Direccion del cliente
4	cl_telefono		Numérico	15	0	Si	Telefono del cliente
5	cl_email		Texto	40	0	Si	Email del cliente
6	cl_descuento		Numérico	5	0	No	Descuento del cliente
7	cl_estado		Texto	2	0	No	Estado del cliente

Nombre de la tabla: Usuarios					Descripción de la tabla:		
Aplicación: Inventario y Facturación					Almacena la información de cada uno de los usuarios de la aplicación.		

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	usu_cedula	PK	Numérico	15	0	No	Cedula del usuario
2	usu_nombre		Texto	20	0	No	Nombre del usuario
3	usu_apellido		Texto	20	0	No	Apellido del usuario
4	usu_direccion		Texto	40	0	No	Direccion del usuario
5	usu_telefono		Numérico	15	0	Si	Telefono del usuario
6	usu_email		Texto	40	0	Si	Correo electrónico del usuario

7	usu_estado		Texto	2	0	No	Estado del usuario
8	usu_nomusuario		Texto	40	0	No	Nombre de usuario para la apli
9	usu_contrasena		Texto	40	0	Si	Contraseña del usuario
10	Usu_fechaereg		Date	--	0	No	Fecha de registro del usuario
11	usu_tu_codtipousu	FK	Numérico	10	0	No	Codigo del tipo de usuario

Nombre de la tabla: Tipo_usuarios

Aplicación: Inventario y Facturación

Descripción de la tabla:

Almacena la información de cada uno de los tipos de usuarios de la aplicación.

Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	tu_codtipousu	PK	Numérico	10	0	No	Código del tipo de usuario
2	tu_nomtipousu		Texto	40	0	No	Nombre del tipo de usuario

Nombre de la tabla: Bodegas_productos

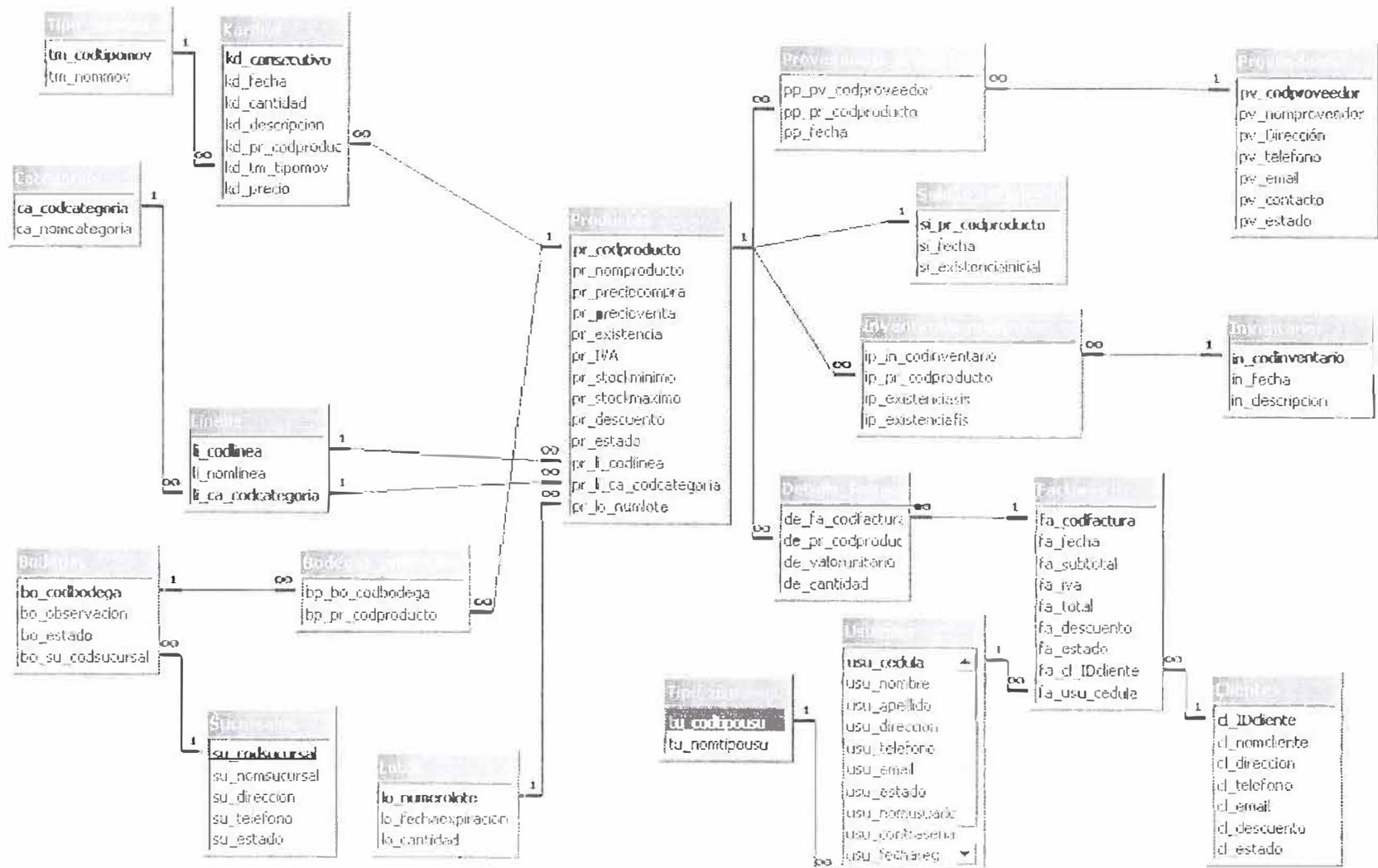
Aplicación: Inventario y Facturación

Descripción de la tabla:

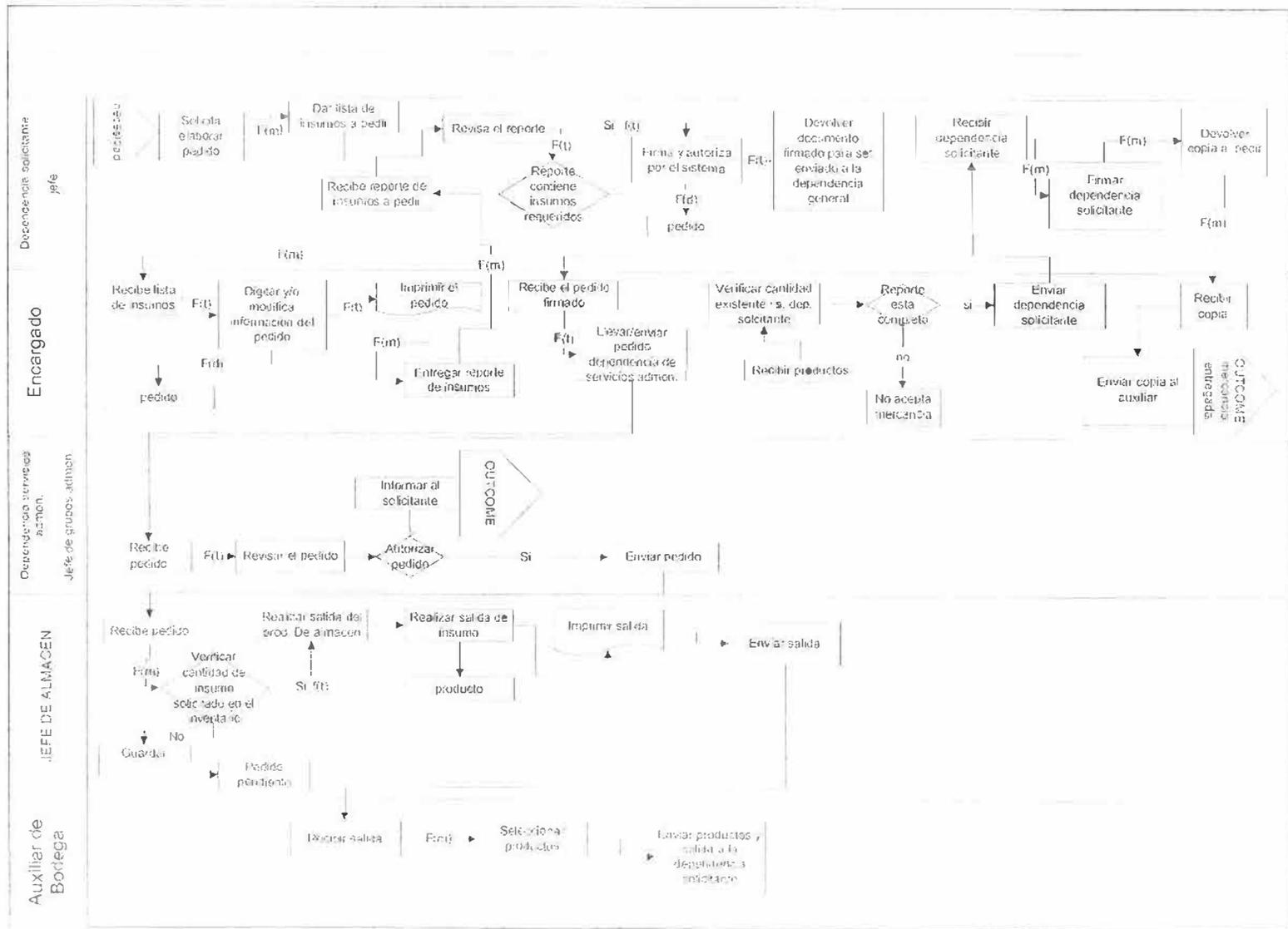
Esta tabla nace de una relación de muchos a muchos la cual almacena información de las bodegas asociadas a los productos de ella misma.

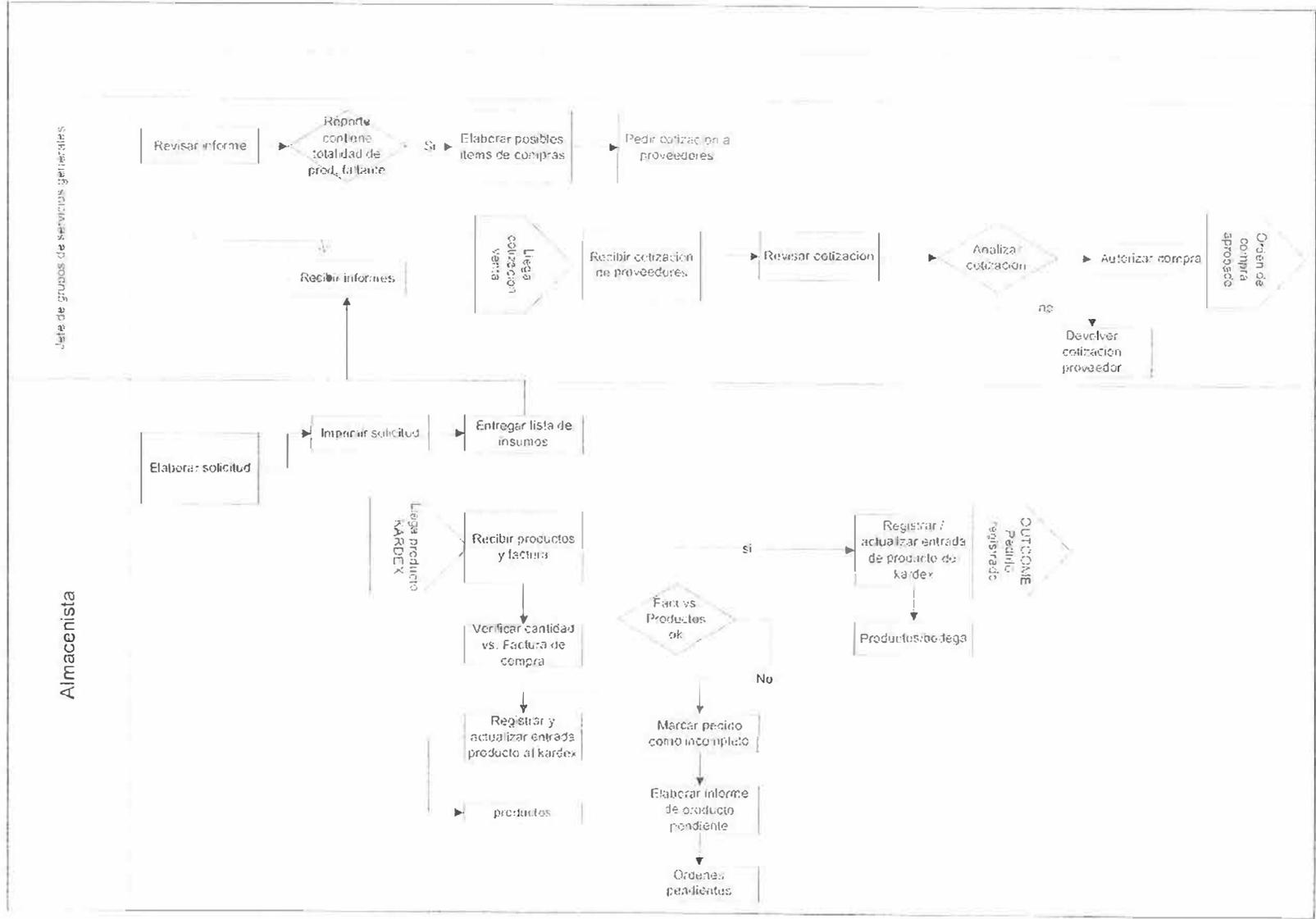
Nro	Nombre Atributo	Llaves	Tipo de dato	Long	Dec	Null	Descripción
1	bp_bo_codbodega	PK	Numérico	10	0	No	Código del tipo de usuario
2	bp_pr_codproducto		Texto	40	0	No	Nombre del tipo de usuario

4.5 MODELO RELACIONAL



4.6 DIAGRAMA FLUJO DEL SISTEMA ACTUAL





5. EJECUCION DE LA CODIFICACION

'DECLARACION DE VARIABLES LOCALES PARA ESTE FORMULARIO

```
Dim CADCONNECT As String
Dim CONECT As New ADODB.Connection
Dim REGISTER As New ADODB.Recordset
```

'AL MOMENTO DE CARGAR EL FORMULARIO SE GENERA LA CONEXION A LA BASE DE DATOS

```
Private Sub Form_Load()
    CADCONNECT = "DSN=DISTRIFAGAR"
    CONECT.Open CADCONNECT
    INICIOSESION.KeyPreview = True
End Sub
```

'AL MOMENTO PERDER EL FOCO LA CAJA DE TEXTO EJECUTA EL PROCEDIMIENTO ALMACENADO EN EL BOTON ACEPTAR

```
Private Sub TXTCONTRASENA_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        ACEPTAR_Click
    End If
End Sub
```

'BOTON ACEPTAR PARA INGRESAR AL PROGRAMA

```
Private Sub ACEPTAR_Click()
```

'CERRAR EL REGISTRO EN CASO DE ESTAR ABIERTO (LOCAL)

```
    If REGISTER.State = adStateOpen Then
        REGISTER.Close
    End If
```

'CONSULTA PARA VERIFICAR SI EL NOMBRE DE USUARIO EXISTE EN LA BD

```
QUERY = "SELECT * FROM USUARIOS U,EMPLEADOS E WHERE U.US_NOMBRE = '" & Trim(TXTUSUARIO) & "' AND E.EM_CEDULA = U.EM_CEDULA"
REGISTER.Open QUERY, CONECT, adOpenDynamic, adLockBatchOptimistic
```

```
    If Not REGISTER.EOF Then
```

'DECODIFICA LA CONTRASEÑA DEL USUARIO PARA INGRESAR AL SISTEMA

```
        TEMP = EncryptString(REGISTER.Fields("US_CLAVE"), "UJUJU")
        If TXTCONTRASENA = "" Then
            MsgBox "No ha Digitado la Contraseña", vbInformation, "AVISO"
            TXTCONTRASENA.SetFocus
        Else
            If TXTCONTRASENA <> TEMP Then
                MsgBox "Contraseña Incorrecta", vbInformation, "AVISO"
                TXTCONTRASENA.SetFocus
            End If
        End If
```

```

Else
    US_NOMBRE = TXTUSUARIO
    COMENTARIO2.Visible = True
    CONECT.Close
    Unload Me
    MENU.Show
End If
End If
Else
    If TXTUSUARIO = "" Then
        MsgBox "No ha Digitado el Usuario", vbInformation, "AVISO"
        TXTUSUARIO.SetFocus
    Else
        MsgBox "El Usuario no Existe", vbInformation, "AVISO"
        TXTUSUARIO.SetFocus
    End If
End If
End Sub

```

'DECLARACION DE VARIABLES PUBLICAS DE LA APLICACION

```

Public CADCONNECT As String
Public CONECT As New ADODB.Connection
Public REGISTER As New ADODB.Recordset

```

```

Public CADENA As String
Public CONEXION As New ADODB.Connection
Public REGISTRO As New ADODB.Recordset

```

'AL MOMENTO DE CARGAR EL FORMULARIO SE GENERA LA CONEXION A LA BASE DE DATOS

```

Private Sub MDIForm_Load()
    Dim NODOS As Node
    Form1.Show
    Form1.ZOrder
    Form1.Visible = False
    CADENA = "DSN=DISTRIFAGAR"
    CONEXION.Open CADENA
    LABELUSUARIO = US_NOMBRE

```

'ENLACE DE CRYSTAL REPORT CON LA BASE DE DATOS

```

    CrystalReport.Connect = "DSN=DISTRIFAGAR;USER=PROYECTO;PWD="

```

'CONSULTA PARA VERIFICAR QUE TIPO DE USUARIO INGRESO A LA APLICACION

```

QUERY = " SELECT U.TU_CODIGO FROM USUARIOS U, TIPO_USUARIOS T
WHERE U.US_NOMBRE = " & Trim(US_NOMBRE) & " AND U.TU_CODIGO =
T.TU_CODIGO "

```

```
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
```

```
If Not MENU.REGISTRO.EOF Then
```

```
TEMPUSUARIO = MENU.REGISTRO.Fields("TU_CODIGO")
```

'DESPLIEGA EL MENU DE TODOS LOS MÓDULOS A LOS QUE SE PUEDE TENER ACCESO EN LA APLICACION

```
If TEMPUSUARIO = 1 Then
```

```
Set NODOS = TreeView1.Nodes.Add(, , "MENU", "MENU PRINCIPAL", 5, 7)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION0",
"PARAMETROS", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION0", tvwChild, "OPCION0.1",
"RANGO FACTURACION", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION0", tvwChild, "OPCION0.2",
"VALORES IVA", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION1",
"FACTURACION", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION1", tvwChild, "OPCION1.1",
"GENERAR FACTURA", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION2",
"PEDIDO", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION2", tvwChild, "OPCION2.1",
"GENERAR PEDIDO", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION3",
"PROVEEDORES", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION3", tvwChild, "OPCION3.1",
"INGRESAR PROVEEDOR", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION4",
"PRODUCTOS", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION4", tvwChild, "OPCION4.1",
"INGRESAR PRODUCTO", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION4", tvwChild, "OPCION4.2",
"LINEAS DE PRODUCTOS", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION4", tvwChild, "OPCION4.3",
"CATEGORIAS DE PRODUCTOS", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION5",
"CLIENTES", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION5", tvwChild, "OPCION5.1",
"INGRESAR CLIENTE", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION6",
"USUARIOS", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION6", tvwChild, "OPCION6.1",
"INGRESAR USUARIOS", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION7",
"CONTRASEÑAS", 1, 2)
```

```
Set NODOS = TreeView1.Nodes.Add("OPCION7", tvwChild, "OPCION7.2",
"CAMBIAR CONTRASEÑA", 6)
```

```
Set NODOS = TreeView1.Nodes.Add("MENU", tvwChild, "OPCION8",
```

```

"INVENTARIO", 1, 2)
Set NODOS = TreeView1.Nodes.Add("OPCION8", twwChild, "OPCION8.1",
"CREAR BODEGAS", 6)
Set NODOS = TreeView1.Nodes.Add("OPCION8", twwChild, "OPCION8.2",
"REPORTES", 6)
Set NODOS = TreeView1.Nodes.Add("MENU", twwChild, "OPCION9",
"AUDITORIA", 1, 2)
Set NODOS = TreeView1.Nodes.Add("OPCION9", twwChild, "OPCION9.1",
"REPORTES DINAMICOS", 6)
Set NODOS = TreeView1.Nodes.Add("OPCION9", twwChild, "OPCION9.2",
"LOG DE TRANSACCIONES", 6)
Set NODOS = TreeView1.Nodes.Add("OPCION9", twwChild, "OPCION9.3",
"EXPORTACION DE TABLAS", 6)
Set NODOS = TreeView1.Nodes.Add("MENU", twwChild, "SALIR", "SALIR", 4)
TreeView1.Nodes(1).Expanded = True
End If

```

***DESPLIEGA EL MENU DE TODOS LOS MODULOS A LOS QUE SE PUEDE TENER ACCESO EN LA APLICACION**

```

If TEMPUSUARIO = 2 Then
Set NODOS = TreeView1.Nodes.Add(, "MENU", "MENU PRINCIPAL", 5)
Set NODOS = TreeView1.Nodes.Add("MENU", twwChild, "OPCION1",
"FACTURACION", 1, 2)
Set NODOS = TreeView1.Nodes.Add("OPCION1", twwChild, "OPCION1.1",
"GENERAR FACTURA", 6)
Set NODOS = TreeView1.Nodes.Add("MENU", twwChild, "OPCION2",
"PEDIDO", 1, 2)
Set NODOS = TreeView1.Nodes.Add("OPCION2", twwChild, "OPCION2.1",
"GENERAR PEDIDO", 6)
Set NODOS = TreeView1.Nodes.Add("MENU", twwChild, "SALIR", "SALIR", 4)
TreeView1.Nodes(1).Expanded = True
End If
Else
MsgBox ("Ha ocurrido un error")
End If
End Sub

```

***CARGAR FORMULARIO CLIENTES**

```

Private Sub ac_Click()
CLIENTE.Show
CLIENTE.ZOrder
End Sub

```

***CARGAR FORMULARIO AYUDAS**

```

Private Sub acerca_Click()
AYUDA.Show
AYUDA.ZOrder

```

End Sub

***CARGAR FORMULARIO PROVEEDOR**

```
Private Sub ap_Click()  
    PROVEEDOR.Show  
    PROVEEDOR.ZOrder  
End Sub
```

***CARGAR FORMULARIO PRODUCTO**

```
Private Sub apro_Click()  
    PRODUCTO.Show  
    PRODUCTO.ZOrder  
End Sub
```

***CARGAR FORMULARIO BODEGAS**

```
Private Sub bodega_Click()  
    BODEGA.Show  
    BODEGA.ZOrder  
End Sub
```

***CARGAR FORMULARIO PARA CAMBIO DE CONTRASEÑAS**

```
Private Sub cen_Click()  
    CONTRASEÑA.Show  
    CONTRASEÑA.ZOrder  
End Sub
```

***CARGAR FORMULARIO FACTURAS**

```
Private Sub gf_Click()  
    FACTURA.Show  
    FACTURA.ZOrder  
End Sub
```

***CARGAR FORMULARIO DE REPORTE**

```
Private Sub reporte_Click()  
    INVENTARIO.Show  
    INVENTARIO.ZOrder  
End Sub
```

***CARGAR FORMULARIO DE USUARIOS**

```
Private Sub usuario_Click()  
    USUARIOS.Show  
End Sub
```

***AL MOMENTO DE SALIR DE LA APLICACIÓN MUESTRA UN MENSAJE DE CONFIRMACION**

```
Private Sub Sal_Click()
```

```

RESPUESTA = MsgBox("¿Desea Salir de la Aplicacion?", vbYesNo, "SALIR")
If RESPUESTA = 6 Then
    CONEXION.Close
End
End If
End Sub

```

'AL MOMENTO DE DAR CLICK EN EL ARBOL DE OPCIONES SELECCIONAS EL FORMULARIO AL QUE QUIERE ACCEDER EL USUARIO

```
Private Sub TreeView1_NodeClick(ByVal Node As MSCometLib.Node)
```

```

    Select Case Node.text
        Case Is = "REPORTES"
            INVENTARIO.Show
            INVENTARIO.ZOrder

        Case Is = "GENERAR FACTURA"
            FACTURA.Show
            FACTURA.ZOrder
        Case Is = "GENERAR PEDIDO"
            PEDIDO.Show
            PEDIDO.ZOrder
        Case Is = "INGRESAR PROVEEDOR"
            PROVEEDOR.Show
            PROVEEDOR.ZOrder
        Case Is = "INGRESAR PRODUCTO"
            PRODUCTO.Show
            PRODUCTO.ZOrder
        Case Is = "CREAR IVA"
            IVA.Show
            IVA.ZOrder
        Case Is = "LINEAS DE PRODUCTOS"
            LINEAS.Show
            LINEAS.ZOrder
        Case Is = "CATEGORIAS DE PRODUCTOS"
            CATEGORIAS.Show
            CATEGORIAS.ZOrder
        Case Is = "INGRESAR CLIENTE"
            CLIENTE.Show
            CLIENTE.ZOrder
        Case Is = "INGRESAR USUARIOS"
            USUARIOS.Show
            USUARIOS.ZOrder
        Case Is = "REPORTES DINAMICOS"
            GENERADOR.Show
            GENERADOR.ZOrder
        Case Is = "LOG DE TRANSACCIONES"
            AUDITORIA.Show

```

```

    AUDITORIA.ZOrder
Case Is = "CAMBIAR CONTRASEÑA"
    CONTRASENA.Show
    CONTRASENA.ZOrder
Case Is = "CREAR BODEGAS"
    BODEGA.Show
    BODEGA.ZOrder
Case Is = "EXPORTACION DE TABLAS"
    EXPORTAR.Show
    EXPORTAR.ZOrder
Case Is = "AYUDA"
    Form1.Show
    Form1.ZOrder
Case Is = "SALIR"
    RESPUESTA = MsgBox("¿Desea Salir de la Aplicacion?", vbYesNo, "SALIR")
    If RESPUESTA = 6 Then
        CONEXION.Close
    End
End If
End Select
End Sub

```

'PROCEDIMIENTO PUBLICO PARA CERRAR UN REGISTRO QUE SE ENCUENTRE ABIERTO (CONEXION)

```

Public Sub CERRAR()
    If MENU.REGISTRO.State = adStateOpen Then
        MENU.REGISTRO.Close
    End If
End Sub

```

'AL MOMENTO DE SALIR DE LA APLICACIÓN MUESTRA UN MENSAJE DE CONFIRMACION

```

Private Sub MDIForm_Unload(Cancel As Integer)
    RESPUESTA = MsgBox("¿Desea Salir de la Aplicacion?", vbYesNo, "SALIR")
    If RESPUESTA = 6 Then
        CONEXION.Close
    End
End If
End Sub

```

'AL MOMENTO QUE SE CARGA EL FORMULARIO AUDITORIA

```

Private Sub Form_Load()
    Me.Top = 0
    Me.Left = 0
    CONSULTA_AUDITORIA.Rows = 1
    CONSULTA_AUDITORIA.Cols = 6

```

```

CONSULTA_AUDITORIA.ColWidth(0) = 800
CONSULTA_AUDITORIA.ColWidth(1) = 1500
CONSULTA_AUDITORIA.ColWidth(2) = 1800
CONSULTA_AUDITORIA.ColWidth(3) = 1400
CONSULTA_AUDITORIA.ColWidth(4) = 1200
CONSULTA_AUDITORIA.ColWidth(5) = 2000
CONSULTA_AUDITORIA.TextMatrix(0, 0) = "Item"
CONSULTA_AUDITORIA.TextMatrix(0, 1) = "Operacion"
CONSULTA_AUDITORIA.TextMatrix(0, 2) = "Tabla"
CONSULTA_AUDITORIA.TextMatrix(0, 3) = "Fecha"
CONSULTA_AUDITORIA.TextMatrix(0, 4) = "Hora"
CONSULTA_AUDITORIA.TextMatrix(0, 5) = "Usuario"
DATO = Date

```

'SE CARGA EL RANGO DE FECHAS. (SON LAS DEL SISTEMA)

```

FECHAINI.Value = DATO
FECHAFIN.Value = DATO

```

'SE CARGAN LOS USUARIOS EN EL COMBOUSUARIO

```

MENU.CERRAR

```

```

    QUERY = "SELECT U.US_NOMBRE, E.EM_CEDULA FROM USUARIOS U,
    EMPLEADOS E WHERE E.EM_CEDULA = U.EM_CEDULA"

```

```

    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic

```

```

If MENU.REGISTRO.EOF Then

```

```

    MsgBox ("La tabla Usuarios esta vacia")

```

```

Else

```

```

    Do While MENU.REGISTRO.EOF = False

```

```

        COMBOUSUARIO.AddItem MENU.REGISTRO.Fields("US_NOMBRE")

```

```

        COMBOUSUARIO.ItemData(COMBOUSUARIO.ListCount - 1) =

```

```

        MENU.REGISTRO("EM_CEDULA")

```

```

        MENU.REGISTRO.MoveNext

```

```

    Loop

```

```

    COMBOUSUARIO.ListIndex = 0

```

```

End If

```

'SE CARGAN LAS OPERACIONES EN EL COMBOOPERACION

```

COMBOOPERACION.AddItem "INSERCIÓN"

```

```

COMBOOPERACION.AddItem "MODIFICACION"

```

```

COMBOOPERACION.ListIndex = 0

```

'SE CARGAN TODOS LOS DATOS DE LA TABLA AUDITORIA EN EL FLEX

```

MENU.CERRAR

```

```

QUERY = "SELECT * FROM AUDITORIA"

```

```

    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic

```

```

Do While MENU.REGISTRO.EOF = False
    CONSULTA_AUDITORIA.AddItem CONSULTA_AUDITORIA.Rows & vbTab
    & MENU.REGISTRO("AU_OPERACION") & vbTab &
    MENU.REGISTRO("AU_TABLA") & vbTab &
    MENU.REGISTRO("AU_FECHA") & vbTab &
    MENU.REGISTRO.Fields("AU_HORA") & vbTab &
    MENU.REGISTRO.Fields("US_NOMBRE")
    MENU.REGISTRO.MoveNext
Loop
End Sub

```

```

Private Sub MOVIMIENTOS_Click()
    CONSULTA_AUDITORIA.Rows = 1
    If Format(FECHAFIN.Value, "yyyy.MM.dd") < Format(FECHAINI.Value,
"yyyy.MM.dd") Then
        MsgBox "La fecha final no puede ser anterior a la inicial", vbInformation,
"AVISO"
        FECHAFIN.SetFocus
        Exit Sub
    End If

```

'SE CARGA EL MS_FLEX CON LOS MOVIMIENTOS A CONSULTAR

```

MENU.CERRAR
    QUERY = "SELECT * FROM AUDITORIA WHERE US_NOMBRE = '" &
Trim(COMBOUSUARIO.List(COMBOUSUARIO.ListIndex)) & "' AND
AU_OPERACION = '" &
Trim(COMBOOPERACION.List(COMBOOPERACION.ListIndex)) & "' AND
(AU_FECHA BETWEEN '" & Trim(Format(FECHAINI.Value, "YYYY-MM-
DD")) & "' AND '" & Trim(Format(FECHAFIN.Value, "YYYY-MM-DD")) &
"')"
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If MENU.REGISTRO.EOF Then
    MsgBox "No hay Transacciones para mostrar", vbInformation, "AVISO"
    Exit Sub
End If
Do While MENU.REGISTRO.EOF = False
    CONSULTA_AUDITORIA.AddItem CONSULTA_AUDITORIA.Rows & vbTab
    & MENU.REGISTRO("AU_OPERACION") & vbTab &
    MENU.REGISTRO("AU_TABLA") & vbTab &
    MENU.REGISTRO("AU_FECHA") & vbTab &
    MENU.REGISTRO.Fields("AU_HORA") & vbTab &
    MENU.REGISTRO.Fields("US_NOMBRE")
    MENU.REGISTRO.MoveNext
Loop
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO BODEGAS

```
Private Sub Form_Load()
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
    BODEGA.KeyPreview = True
End Sub
```

'SE CARGA EL FORMULARIO DE CONSULTA DE BODEGAS

```
Private Sub CONSULTAR_Click()
    CONSULTAR_BODEGA.Show
    CONSULTAR_BODEGA.ZOrder
End Sub
```

```
Private Sub INGRESAR_Click()
    If ACTIVO.Value = True Then
        TEMP = 1
    Else
        If INACTIVO.Value = True Then
            TEMP = 2
        End If
    End If
End Sub
```

'SE INGRESAN LOS DATOS EN LA TABLA BODEGAS

```
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO BODEGAS VALUES(" &
    Trim(TXTBODEGA) & ", " & UCase(Trim(TXTOBSERVACIONES)) & ", " &
    Trim(TXTCUPO) & ", " & Trim(TEMP) & ")")
```

```
MsgBox "Se ha creado la bodega correctamente", vbInformation, "AVISO"
```

'SE INGRESAN LOS DATOS EN LA TABLA AUDITORIA

```
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('INSERCIÓN','BODEGAS','" & Trim(Format(Date, "YYYY-MM-DD")) & "','" &
Trim(Format(Time, "hh:mm:ss")) & "','" & UCase(Trim(US_NOMBRE)) & ")")
```

```
TXTOBSERVACIONES = ""
TXTBODEGA = ""
TXTCUPO = ""
INGRESAR.Enabled = False
MODIFICAR.Enabled = False
TXTBODEGA.SetFocus
End Sub
```

'SE LIMPIAN LOS CAMPOS DEL FORMULARIO BODEGAS

```
Private Sub LIMPIAR_Click()
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
```

```

    TXTBODEGA.Enabled = True
    TXTBODEGA = ""
    TXTCUPO = ""
    TXTOBSERVACIONES = ""
    ACTIVO.Value = True
    TXTBODEGA.SetFocus
End Sub

Private Sub MODIFICAR_Click()
    If ACTIVO.Value = True Then
        TEMP = 1
    Else
        TEMP = 2
    End If
    If TXTCUPO <> "" Then
        'SE ACTUALIZAN DATOS EN LA TABLA BODEGAS
        MENU.CERRAR
        MENU.CONEXION.Execute ("UPDATE BODEGAS SET BO_OBSERVACION
        = " & Trim(TXTOBSERVACIONES) & " , BO_CUPO = " & Trim(TXTCUPO)
        & " , BO_ESTADO = " & Trim(TEMP) & " WHERE BO_CODBODEGA = " &
        Trim(TXTBODEGA))
        MsgBox "Datos modificados correctamente", vbInformation, "AVISO"

        'SE INGRESAN DATOS EN LA TABLA AUDITORIA
        MENU.CERRAR
        MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
        ('MODIFICACION','BODEGAS'," & Trim(Format(Date, "YYYY-MM-DD")) &
        ",'" & Trim(Format(Time, "hh:mm:ss")) & "','" & UCase(Trim(US_NOMBRE)) &
        "')")
        TXTBODEGA = ""
        TXTOBSERVACIONES = ""
        TXTCUPO = ""
        TXTBODEGA.Enabled = True
        MODIFICAR.Enabled = False
        TXTBODEGA.SetFocus
    Else
        MsgBox "Digite el cupo de la bodega", vbInformation, "AVISO"
    End If
End Sub

Private Sub TXTBODEGA_LostFocus()
    If TXTBODEGA <> "" Then
        'SE CONSULTAN LAS BODEGAS EN LA BASE DE DATOS
        MENU.CERRAR
        QUERY = "SELECT * FROM BODEGAS WHERE BO_CODBODEGA = " &
        Trim(TXTBODEGA)
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic.
    End If
End Sub

```

```

    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
'SE ASIGNAN LOS VALORES DE LA CONSULTA A LA RESPECTIVA CAJA DE
TEXTO
        TEMP = MENU.REGISTRO.Fields("BO_ESTADO")
        TXTOBSERVACIONES = MENU.REGISTRO.Fields("BO_OBSERVACION")
        TXTCUPO = MENU.REGISTRO.Fields("BO_CUPO")
        If TEMP = 1 Then
            ACTIVO.Value = True
        Else
            INACTIVO.Value = True
        End If
        TXTBODEGA.Enabled = False
        INGRESAR.Enabled = False
        MODIFICAR.Enabled = True
    Else
        INGRESAR.Enabled = True
        MODIFICAR.Enabled = False
    End If
    Else
        INGRESAR.Enabled = False
    End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CATEGORIAS

```

Private Sub Form_Load()
    CATEGORIAS.KeyPreview = True
    MODIFICAR.Enabled = False
    INGRESAR.Enabled = False

```

'SELECCIONA TODOS LOS DATOS DE LA TABLA LINEAS

```

MENU.CERRAR
QUERY = "SELECT * FROM LINEAS ORDER BY LI_NOMLINEA"
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If MENU.REGISTRO.EOF Then
    MsgBox ("La tabla lineas esta vacia")
Else

```

'AGREGA AL COMBOLIST DE LINEAS LOS NOMBRES DE LAS LINEAS

```

Do While MENU.REGISTRO.EOF = False
    COMBOLINEA.AddItem MENU.REGISTRO(1)
    COMBOLINEA.ItemData(COMBOLINEA.ListCount - 1) =
    MENU.REGISTRO(0)
    MENU.REGISTRO.MoveNext
Loop
End If

```

**'CADA VES Q SE CARGUE EL FORMULARIO QUEDARA SELECCIONADO
POR DEFECTO EL PRIMERO**

```
    COMBOLINEA.ListIndex = 0
End Sub
```

'SE CARGA EL FORMULARIO DE CONSULTA DE CATEGORIAS

```
Private Sub CONSULTAR_Click()
    CONSULTAR_CATEGORIA.Show
    CONSULTAR_CATEGORIA.ZOrder
End Sub
```

```
Private Sub INGRESAR_Click()
```

```
    If TXTNOMBRE <> "" Then
```

'SE INGRESAN LOS DATOS EN LA TABLA CATEGORIAS

```
    MENU.CERRAR
```

```
    MENU.CONEXION.Execute ("INSERT INTO CATEGORIAS VALUES(" &
        Trim(TXTCATEGORIA) & "," & UCase(Trim(TXTNOMBRE)) & ", " &
        COMBOLINEA.ItemData(COMBOLINEA.ListIndex) & ")")
```

```
    MsgBox "Datos modificados correctamente", vbInformation, "AVISO"
```

'SE INGRESAN LOS DATOS EN LA TABLA AUDITORIA

```
    MENU.CERRAR
```

```
    MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
        (INSERCIÓN,CATEGORIAS,'" & Trim(Format(Date, "YYYY-MM-DD")) &
        "','" & Trim(Format(Time, "hh:mm:ss")) & "','" & UCase(Trim(US_NOMBRE)) &
        ")")
```

```
    TXTCATEGORIA = ""
```

```
    TXTNOMBRE = ""
```

```
    MODIFICAR.Enabled = False
```

```
    INGRESAR.Enabled = False
```

```
    TXTCATEGORIA.Enabled = True
```

```
    COMBOLINEA.ListIndex = 0
```

```
    TXTCATEGORIA.SetFocus
```

```
Else
```

```
    MsgBox "Digite el nombre de la categoria", vbInformation, "AVISO"
```

```
    TXTNOMBRE.SetFocus
```

```
End If
```

```
End Sub
```

'SE LIMPIAN LOS DATOS DEL FORMULARIO

```
Private Sub LIMPIAR_Click()
```

```
    TXTCATEGORIA = ""
```

```
    TXTNOMBRE = ""
```

```
    COMBOLINEA.ListIndex = 0
```

```
    INGRESAR.Enabled = False
```

```
    MODIFICAR.Enabled = False
```

```
    TXTCATEGORIA.Enabled = True
```

```
    TXTCATEGORIA.SetFocus
```

End Sub

Private Sub MODIFICAR_Click()

If TXTNOMBRE <> "" Then

'ACTUALIZA LOS DATOS DE LA TABLA CATEGORIA

MENU.CERRAR

```
MENU.CONEXION.Execute ("UPDATE CATEGORIAS SET
CA_NOMCATEGORIA = " & UCase(Trim(TXTNOMBRE)) & ",
LI_CODLINEA = " & COMBOLINEA.ItemData(COMBOLINEA.ListIndex) & "
WHERE CA_CODCATEGORIA = " & Trim(TXTCATEGORIA))
MsgBox "Datos modificados correctamente", vbInformation, "AVISO"
```

'SE INGRESAN LOS DATOS EN LA TABLA AUDITORIA

MENU.CERRAR

```
MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('MODIFICACION','CATEGORIAS'," & Trim(Format(Date, "YYYY-MM-DD"))
& "," & Trim(Format(Time, "hh:mm:ss")) & "," & UCase(Trim(US_NOMBRE))
& ")")
```

TXTCATEGORIA = ""

TXTNOMBRE = ""

MODIFICAR.Enabled = False

INGRESAR.Enabled = False

TXTCATEGORIA.Enabled = True

COMBOLINEA.ListIndex = 0

TXTCATEGORIA.SetFocus

Else

MsgBox "Digite el nombre de la categoria", vbInformation, "AVISO"

TXTNOMBRE.SetFocus

End If

End Sub

Private Sub TXTCATEGORIA_LostFocus()

If TXTCATEGORIA <> "" Then

'SELECCIONA LOS DATOS DE LA TABLA CATEGORIA SEGUN LA CLAUSULA

MENU.CERRAR

```
QUERY = "SELECT * FROM CATEGORIAS WHERE CA_CODCATEGORIA =
" & Trim(TXTCATEGORIA)
```

```
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
```

If Not MENU.REGISTRO.EOF Then

'SE COLOCA LOS DATOS DE LA CONSULTA EN SUS RESPECTIVOS CAMPOS

TXTCATEGORIA.Enabled = False

TXTNOMBRE = MENU.REGISTRO.Fields("CA_NOMCATEGORIA")

TEMPLINEA = MENU.REGISTRO.Fields("LI_CODLINEA")

For i = 0 To COMBOLINEA.ListCount - 1

If COMBOLINEA.ItemData(i) = TEMPLINEA Then

```

        COMBOLINEA.ListIndex = i
    End If
Next
MODIFICAR.Enabled = True
INGRESAR.Enabled = False
TXTCATEGORIA.Enabled = False
Else
    INGRESAR.Enabled = True
    MODIFICAR.Enabled = False
    TXTNOMBRE = ""
End If
Else
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
    TXTNOMBRE = ""
End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CLIENTES

```

Private Sub Form_Load()
    Me.Left = 0
    Me.Top = 0
    MODIFICAR.Enabled = False
    INGRESAR.Enabled = False
    CLIENTE.KeyPreview = True
    ACTIVO.Enabled = True
End Sub

```

'SE CARGA EL FORMULARIO DE CONSULTA DE CLIENTES

```

Private Sub CONSULTAR_Click()
    FORM_ACTUAL_CLIENTE = "CLIENTE"
    CONSULTAR_CLIENTE.Show
    CONSULTAR_CLIENTE.ZOrder
End Sub

```

```

Private Sub INGRESAR_Click()
    If TXTIDCLIENTE = "" Then
        MsgBox "Digite la identificación del cliente", vbInformation, "Aviso"
        TXTIDCLIENTE.SetFocus
        Exit Sub
    Else
        If TXTNOMBRE = "" Then
            MsgBox "Digite el nombre del cliente", vbInformation, "Aviso"
            TXTNOMBRE.SetFocus
            Exit Sub
        Else
            If ACTIVO.Value = False And INACTIVO.Value = False Then

```

```

        MsgBox "Asignele un estado al cliente", vbInformation, "Aviso"
    Exit Sub
End If
End If
End If
If ACTIVO.Value = True Then
    TEMP = 1
Else
    If INACTIVO.Value = True Then
        TEMP = 2
    End If
End If
'SE INGRESAN LOS DATOS EN LA TABLA CLIENTES
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO CLIENTES VALUES(" &
    Trim(TXTIDCLIENTE) & ", " & UCase(Trim(TXTNOMBRE)) & ", " &
    UCase(Trim(TXTDIRECCION)) & ", " & Trim(TXTTELEFONO) & ", " &
    UCase(Trim(TXTEMAIL)) & ", " & 0 & ", " & Trim(TEMP) & " ")")
'SE INGRESAN LOS DATOS EN LA TABLA AUDITORIA
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
(INSERCION,'CLIENTE'," & Trim(Format(Date, "YYYY-MM-DD")) & ", " &
    Trim(Format(Time, "hh:mm:ss")) & ", " & UCase(Trim(US_NOMBRE)) & " ")")
MsgBox "cliente ingresado correctamente", vbInformation, "AVISO"
TXTIDCLIENTE = ""
TXTNOMBRE = ""
TXTDIRECCION = ""
TXTTELEFONO = ""
TXTEMAIL = ""
INGRESAR.Enabled = False
ACTIVO.Enabled = True
TXTIDCLIENTE.Enabled = True
TXTIDCLIENTE.SetFocus
End Sub

Private Sub MODIFICAR_Click()
    If ACTIVO.Value = True Then
        TEMP = 1
    Else
        If INACTIVO.Value = True Then
            TEMP = 2
        End If
    End If
    If TXTIDCLIENTE <> "" Then
'SE ACTUALIZAN LOS DATOS DE LA TABLA CLIENTES
        MENU.CERRAR
        MENU.CONEXION.Execute ("UPDATE CLIENTES SET CL_NOMCLIENTE =

```

```

" & UCase(Trim(TXTNOMBRE)) & ", CL_DIRECCION = " &
UCase(Trim(TXTDIRECCION)) & " , CL_TELEFONO = " &
Trim(TXTTELEFONO) & " , CL_EMAIL = " & UCase(Trim(TXTEMAIL)) & " ,
CL_ESTADO = " & Trim(TEMP) & " WHERE CL_IDCLIENTE = " &
Trim(TXTIDCLIENTE))

```

'SE INGRESAN LOS DATOS EN LA TABLA AUDITORIA

MENU.CERRAR

```

MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('MODIFICACION','CLIENTE'," & Trim(Format(Date, "YYYY-MM-DD")) &
"," & Trim(Format(Time, "hh:mm:ss")) & "," & UCase(Trim(US_NOMBRE)) &
"")

```

TXTIDCLIENTE = ""

TXTNOMBRE = ""

TXTDIRECCION = ""

TXTTELEFONO = ""

TXTEMAIL = ""

MODIFICAR.Enabled = False

TXTIDCLIENTE.Enabled = True

TXTIDCLIENTE.SetFocus

ACTIVO.Value = True

End If

End Sub

Private Sub TXTIDCLIENTE_LostFocus()

'SE VALIDA QUE EL CARACTER INGRESADO SEA NUMERICO

If Not IsNumeric(TXTIDCLIENTE) And TXTIDCLIENTE <> "" Then

```

MsgBox "Codigo Invalido , este es un campo numerico", vbInformation, "AVISO"

```

TXTIDCLIENTE = ""

TXTIDCLIENTE.SetFocus

INGRESAR.Enabled = False

Exit Sub

End If

'SE SELECCIONAN LOS DATOS DE LA TABLA CLIENTES

MENU.CERRAR

```

QUERY = "SELECT * FROM CLIENTES WHERE CL_IDCLIENTE = " &
Trim(TXTIDCLIENTE)

```

If TXTIDCLIENTE <> "" Then

```

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

```

If Not MENU.REGISTRO.EOF Then

'SE COLOCAN LOS DATOS DE LA CONSULTA EN SUS RESPECTIVOS CAMPOS

```

TEMP = MENU.REGISTRO.Fields("CL_ESTADO")

```

TXTIDCLIENTE.Enabled = False

```

TXTNOMBRE = MENU.REGISTRO.Fields("CL_NOMCLIENTE")

```

```

TXTDIRECCION = MENU.REGISTRO.Fields("CL_DIRECCION")

```

```

    TXTTELEFONO = MENU.REGISTRO.Fields("CL_TELEFONO")
    TXTEMAIL = MENU.REGISTRO.Fields("CL_EMAIL")
    If TEMP = 1 Then
        ACTIVO.Value = True
    Else
        INACTIVO.Value = True
    End If
    TXTNOMBRE.SetFocus
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = True
Else
    INGRESAR.Enabled = True
    MODIFICAR.Enabled = False
    TXTNOMBRE = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
End If
Else
    MODIFICAR.Enabled = False
    INGRESAR.Enabled = False
End If
End Sub

```

'SE LIMPIAN LOS DATOS DEL FORMULARIO

```

Private Sub LIMPIAR_Click()
    TXTIDCLIENTE.Enabled = True
    TXTIDCLIENTE = ""
    TXTNOMBRE = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
    ACTIVO.Value = True
    TXTIDCLIENTE.SetFocus
End Sub

```

```

Private Sub TXTTELEFONO_LostFocus()

```

'SE VALIDA QUE EL CARÁCTER INGRESADO SEA NUMÉRICO

```

    If Not IsNumeric(TXTTELEFONO) And TXTTELEFONO <> "" Then
        MsgBox "Codigo Invalido , este es un campo numerico", vbInformation, "AVISO"
        TXTTELEFONO = ""
        TXTTELEFONO.SetFocus
    Exit Sub
End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO DE CONSULTAR BODEGAS

```
Private Sub Form_Load()
```

```
Me.Left = 0
```

```
Me.Top = 0
```

```
CONSULTA_BODEGA.Rows = 1
```

```
CONSULTA_BODEGA.Cols = 4
```

```
CONSULTA_BODEGA.ColWidth(0) = 800
```

```
CONSULTA_BODEGA.ColWidth(1) = 1000
```

```
CONSULTA_BODEGA.ColWidth(2) = 1500
```

```
CONSULTA_BODEGA.ColWidth(3) = 2500
```

```
CONSULTA_BODEGA.TextMatrix(0, 0) = "Item"
```

```
CONSULTA_BODEGA.TextMatrix(0, 1) = "Codigo"
```

```
CONSULTA_BODEGA.TextMatrix(0, 2) = "Cupo"
```

```
CONSULTA_BODEGA.TextMatrix(0, 3) = "Observacion"
```

'SE SELECCIONAN LOS DATOS DE LA TABLA BODEGAS EN ESTADO ACTIVO

```
MENU.CERRAR
```

```
QUERY = "SELECT * FROM BODEGAS WHERE BO_ESTADO = 1"
```

```
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,  
adLockBatchOptimistic
```

```
If Not MENU.REGISTRO.EOF Then
```

```
Do While MENU.REGISTRO.EOF = False
```

```
If MENU.REGISTRO.Fields("BO_ESTADO") = 1 Then
```

```
ESTADO = "ACTIVO"
```

```
Else
```

```
ESTADO = "INACTIVO"
```

```
End If
```

'SE AGREGAN AL FLEX LOS DATOS DE LA TABLA BODEGA

```
CONSULTA_BODEGA.AddItem CONSULTA_BODEGA.Rows & vbTab &
```

```
MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(2) & vbTab &
```

```
MENU.REGISTRO(1)
```

```
MENU.REGISTRO.MoveNext
```

```
Loop
```

```
Else
```

```
MsgBox "No hay Registro de productos", vbInformation, "AVISO"
```

```
Exit Sub
```

```
End If
```

```
End Sub
```

'SE COLOCAN LOS VALORES SELECCIONADOS EN EL FLEX EN EL FORMULARIO BODEGA

```
Private Sub CONSULTA_BODEGA_Click()
```

```
BODEGA.TXTBODEGA =
```

```
CONSULTA_BODEGA.TextMatrix(CONSULTA_BODEGA.RowSel, 1)
```

```
BODEGA.TXTCUPO =
```

```
CONSULTA_BODEGA.TextMatrix(CONSULTA_BODEGA.RowSel, 2)
```

```

        BODEGA.TXTOBSERVACIONES                                     =
        CONSULTA_BODEGA.TextMatrix(CONSULTA_BODEGA.RowSel, 3)
        BODEGA.TXTBODEGA.Enabled = False
        BODEGA.MODIFICAR.Enabled = True
        BODEGA.INGRESAR.Enabled = False
        Unload Me
    End Sub

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub TXTBUSCADOR_Change()
'SE DEPURAN LOS DATOS DEPENDE AL CRITERIO QUE SE SELECCIONE
    MENU.CERRAR
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM BODEGAS WHERE BO_CODBODEGA LIKE '"
        & Trim(TXTBUSCADOR) & "%' AND BO_ESTADO = 1"
    Else
        QUERY = "SELECT * FROM BODEGAS WHERE BO_OBSERVACION LIKE '"
        & Trim(TXTBUSCADOR) & "%' AND BO_ESTADO = 1"
    End If
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        CONSULTA_BODEGA.Rows = 1
        Do While MENU.REGISTRO.EOF = False
            CONSULTA_BODEGA.AddItem CONSULTA_BODEGA.Rows & vbTab &
            MENU.REGISTRO.Fields("BO_CODBODEGA") & vbTab &
            MENU.REGISTRO.Fields("BO_OBSERVACION") & vbTab &
            MENU.REGISTRO.Fields("BO_CUPO")
            MENU.REGISTRO.MoveNext
        Loop
    Else
        CONSULTA_BODEGA.Rows = 1
        Exit Sub
    End If
End Sub

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTAR CATEGORIAS
Private Sub Form_Load()
    Me.Left = 0
    Me.Top = 0
    CONSULTA_CATE.Rows = 1

```

```

CONSULTA_CATE.Cols = 4
CONSULTA_CATE.ColWidth(0) = 500
CONSULTA_CATE.ColWidth(1) = 800
CONSULTA_CATE.ColWidth(2) = 2000
CONSULTA_CATE.ColWidth(3) = 600
CONSULTA_CATE.TextMatrix(0, 0) = "Item"
CONSULTA_CATE.TextMatrix(0, 1) = "Codigo"
CONSULTA_CATE.TextMatrix(0, 2) = "Nombre"
CONSULTA_CATE.TextMatrix(0, 3) = "Linea"

```

'SELECCIONA LOS DATOS DE LAS TABLAS LINEAS Y CATEGORIAS - CRUCE DE TABLAS

MENU.CERRAR

```

QUERY = "SELECT * FROM CATEGORIAS C, LINEAS L WHERE
L.LI_CODLINEA = C.LI_CODLINEA"
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

```

If Not MENU.REGISTRO.EOF Then

Do While MENU.REGISTRO.EOF = False

'SE COLOCAN LOS DATOS DE LA CONSULTA EN EL FLEX

```

CONSULTA_CATE.AddItem CONSULTA_CATE.Rows & vbTab &
MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1) & vbTab &
MENU.REGISTRO(2)
MENU.REGISTRO.MoveNext

```

Loop

Else

MsgBox "No hay Registro de Categorias", vbInformation, "AVISO"

Exit Sub

End If

End Sub

Private Sub CONSULTA_CATE_Db1Click()

```

CATEGORIAS.TXTCATEGORIA =
CONSULTA_CATE.TextMatrix(CONSULTA_CATE.RowSel, 1)
CATEGORIAS.TXTNOMBRE =
CONSULTA_CATE.TextMatrix(CONSULTA_CATE.RowSel, 2)
COD_LINEA = CONSULTA_CATE.TextMatrix(CONSULTA_CATE.RowSel, 3)

```

'SE CONSULTAN LOS PRODUCTOS POR CODIGO

MENU.CERRAR

```

QUERY = "SELECT * FROM CATEGORIAS C, LINEAS L WHERE
C.LI_CODLINEA = " & Trim(COD_LINEA) & " AND L.LI_CODLINEA =
C.LI_CODLINEA "

```

```

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

```

COD_LINEA = MENU.REGISTRO.Fields("LI_CODLINEA")

```

For i = 0 To CATEGORIAS.COMBOLINEA.ListCount - 1
    If CATEGORIAS.COMBOLINEA.ItemData(i) = COD_LINEA Then
        CATEGORIAS.COMBOLINEA.ListIndex = i
    End If
Next
CATEGORIAS.TXTCATEGORIA.Enabled = False
CATEGORIAS.MODIFICAR.Enabled = True
CATEGORIAS.INGRESAR.Enabled = False
Unload Me
End Sub

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub TXTBUSCADOR_Change()
    MENU.CERRAR
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM CATEGORIAS WHERE CA_CODCATEGORIA
        LIKE '" & Trim(TXTBUSCADOR) & "%'"
    Else
        QUERY = "SELECT * FROM CATEGORIAS WHERE CA_NOMCATEGORIA
        LIKE '" & Trim(TXTBUSCADOR) & "%'"
    End If
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        CONSULTA_CATE.Rows = 1
        Do While MENU.REGISTRO.EOF = False
            CONSULTA_CATE.AddItem CONSULTA_CATE.Rows & vbTab &
            MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1) & vbTab &
            MENU.REGISTRO(2)
            MENU.REGISTRO.MoveNext
        Loop
    Else
        CONSULTA_CATE.Rows = 1
        Exit Sub
    End If
End Sub

'AL MOMENTO DE CARGAR EL FORMULARIO DE CONSULTAR CLIENTES
Private Sub Form_Load()
    Me.Left = 0

```

```

Me.Top = 0
CONSULTA_CLIENTE.Rows = 1
CONSULTA_CLIENTE.Cols = 7
CONSULTA_CLIENTE.ColWidth(0) = 500
CONSULTA_CLIENTE.ColWidth(1) = 1500
CONSULTA_CLIENTE.ColWidth(2) = 2500
CONSULTA_CLIENTE.ColWidth(3) = 2000
CONSULTA_CLIENTE.ColWidth(4) = 1500
CONSULTA_CLIENTE.ColWidth(5) = 3000
CONSULTA_CLIENTE.ColWidth(6) = 800
CONSULTA_CLIENTE.TextMatrix(0, 0) = "Item"
CONSULTA_CLIENTE.TextMatrix(0, 1) = "Cedula / NIT"
CONSULTA_CLIENTE.TextMatrix(0, 2) = "Nombre"
CONSULTA_CLIENTE.TextMatrix(0, 3) = "Direccion"
CONSULTA_CLIENTE.TextMatrix(0, 4) = "Telefono"
CONSULTA_CLIENTE.TextMatrix(0, 5) = "Email"
CONSULTA_CLIENTE.TextMatrix(0, 6) = "Estado"

'SE SELECCIONAN LOS DATOS DE LA TABLA CLIENTES
MENU.CERRAR
    QUERY = "SELECT * FROM CLIENTES WHERE CL_ESTADO = 1"
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        Do While MENU.REGISTRO.EOF = False
            'SE COLOCAN LOS DATOS DE LA CONSULTA EN EL FLEX
            CONSULTA_CLIENTE.AddItem CONSULTA_CLIENTE.Rows & vbTab &
            MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1) & vbTab &
            MENU.REGISTRO(2) & vbTab & MENU.REGISTRO(3) & vbTab &
            MENU.REGISTRO(4) & vbTab & MENU.REGISTRO.Fields("CL_ESTADO")
            MENU.REGISTRO.MoveNext
        Loop
    Else
        MsgBox "No hay Registro de clientes", vbInformation, "AVISO"
    End If
End Sub

Private Sub CONSULTA_CLIENTE_DblClick()
'SE COLOCAN LOS VALORES SELECCIONADOS EN EL FLEX EN EL
FORMULARIO FACTURA
    If FORM_ACTUAL_CLIENTE = "FACTURA" Then
        FACTURA.TXTIDCLIENTE.text =
        CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 1)
        FACTURA.TXTNOMBRE.text =
        CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 2)
        FACTURA.DETALLE.SetFocus
        FACTURA.KeyPreview = False
    End If
End Sub

```

End If

'SE COLOCAN LOS VALORES SELECCIONADOS EN EL FLEX EN EL FORMULARIO CLIENTE

```

If FORM_ACTUAL_CLIENTE = "CLIENTE" Then
    CLIENTE.TXTIDCLIENTE.text =
    CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 1)
    CLIENTE.TXTNOMBRE.text =
    CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 2)
    CLIENTE.TXTDIRECCION.text =
    CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 3)
    CLIENTE.TXTTELEFONO.text =
    CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 4)
    CLIENTE.TXTEMAIL.text =
    CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 5)
    TEMP = CONSULTA_CLIENTE.TextMatrix(CONSULTA_CLIENTE.RowSel, 6)
    If TEMP = 1 Then
        CLIENTE.ACTIVO.Value = True
    Else
        If TEMP = 2 Then
            CLIENTE.INACTIVO.Value = True
        End If
    End If
    CLIENTE.TXTIDCLIENTE.Enabled = False
    CLIENTE.TXTNOMBRE.SetFocus
    CLIENTE.MODIFICAR.Enabled = True
    CLIENTE.INGRESAR.Enabled = False
End If
Unload Me
End Sub

```

```

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

```

```

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

```

```

Private Sub TXTBUSCADOR_Change()
'DEPURA LA INFORMACION POR IDENTIFICACION DEL CLIENTE O NOMBRE DEL CLIENTE A MEDIDA QUE SE DIGITA UN CARACTER
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM CLIENTES WHERE CL_IDCLIENTE LIKE '" &
        Trim(TXTBUSCADOR) & "%'"
    Else
        QUERY = "SELECT * FROM CLIENTES WHERE CL_NOMCLIENTE LIKE '"

```

```

        & Trim(TXTBUSCADOR) & "%"
    End If
    MENU.CERRAR
    MENU.REGISTRO.Open QUERY. MENU.CONEXION. adOpenDynamic.
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        CONSULTA_CLIENTE.Rows = 1
        Do While MENU.REGISTRO.EOF = False
            'SE COLOCA EN EL FLEX LA CONSULTA DEPURADA
            CONSULTA_CLIENTE.AddItem CONSULTA_CLIENTE.Rows & vbTab &
            MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1) & vbTab &
            MENU.REGISTRO(2) & vbTab & MENU.REGISTRO(3) & vbTab &
            MENU.REGISTRO(4) & vbTab & MENU.REGISTRO(5) & vbTab &
            MENU.REGISTRO(6)
            MENU.REGISTRO.MoveNext
        Loop
    Else
        CONSULTA_CLIENTE.Rows = 1
        Exit Sub
    End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTAR FACTURA

```

Private Sub Form_Load()
    Me.Left = 0
    Me.Top = 0
    CONSULTA_FACTURA.Rows = 1
    CONSULTA_FACTURA.Cols = 9
    CONSULTA_FACTURA.ColWidth(0) = 800
    CONSULTA_FACTURA.ColWidth(1) = 1000
    CONSULTA_FACTURA.ColWidth(2) = 1400
    CONSULTA_FACTURA.ColWidth(3) = 1500
    CONSULTA_FACTURA.ColWidth(4) = 1500
    CONSULTA_FACTURA.ColWidth(5) = 1300
    CONSULTA_FACTURA.ColWidth(6) = 1500
    CONSULTA_FACTURA.ColWidth(7) = 800
    CONSULTA_FACTURA.ColWidth(8) = 1800
    CONSULTA_FACTURA.TextMatrix(0, 0) = "Item"
    CONSULTA_FACTURA.TextMatrix(0, 1) = "Codigo"
    CONSULTA_FACTURA.TextMatrix(0, 2) = "Fecha"
    CONSULTA_FACTURA.TextMatrix(0, 3) = "Cedula / NIT"
    CONSULTA_FACTURA.TextMatrix(0, 4) = "Subtotal"
    CONSULTA_FACTURA.TextMatrix(0, 5) = "Iva"
    CONSULTA_FACTURA.TextMatrix(0, 6) = "Total"
    CONSULTA_FACTURA.TextMatrix(0, 7) = "Estado"
    CONSULTA_FACTURA.TextMatrix(0, 8) = "Usuario"
    'SE CONSULTAN LAS FACTURAS GENERADAS

```

```

MENU.CERRAR
    QUERY = "SELECT * FROM FACTURAS"
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
    Do While MENU.REGISTRO.EOF = False
        If MENU.REGISTRO.Fields("FA_ESTADO") = 1 Then
            ESTADO = "OK"
        Else
            ESTADO = "ANULADA"
        End If
    End If

```

'SE CARGA LOS VALORES DE LA FACTURAS GENERADAS EN EL FLEX DE CONSULTA FACTURAS

```

CONSULTA_FACTURA.AddItem CONSULTA_FACTURA.Rows & vbTab &
MENU.REGISTRO.Fields("FA_CODFACTURA") & vbTab &
MENU.REGISTRO.Fields("FA_FECHA") & vbTab &
MENU.REGISTRO.Fields("CL_IDCLIENTE") & vbTab &
MENU.REGISTRO.Fields("FA_SUBTOTAL") & vbTab &
MENU.REGISTRO.Fields("FA_IVA") & vbTab &
MENU.REGISTRO.Fields("FA_TOTAL") & vbTab & ESTADO & vbTab &
MENU.REGISTRO.Fields("US_NOMBRE")
MENU.REGISTRO.MoveNext
Loop
Else
    MsgBox "No hay Registro de facturas", vbInformation, "AVISO"
Exit Sub
End If
End Sub

```

Private Sub CONSULTA_FACTURA_DblClick()

'SE ASIGNAN AL FORMULARIO FACTURA LOS VALORES DE LA FACTURA CONSULTADA

```

FACTURA.TXTFACTURA =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 1)
FACTURA.TXTFECHA =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 2)
FACTURA.TXTIDCLIENTE =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 3)
FACTURA.TXTSUBTOTAL =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 4)
FACTURA.TXTIVA =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 5)
FACTURA.TXTTOTAL =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 6)
FACTURA.TXTUSUARIO =
CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 8)

```

'SE CONSULTAN PARA TRAER EL NOMBRE DEL CLIENTE

```

MENU.CERRAR
  QUERY = "SELECT CL_NOMCLIENTE FROM CLIENTES WHERE
  CL_IDCLIENTE = " &
  Trim(CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 3))
MENU.REGISTRO.Open QUERY, MENU.CONEXIÓN, adOpenDynamic,
adLockBatchOptimistic
FACTURA.TXTNOMBRE = MENU.REGISTRO(0)

```

'SE CONSULTAN PARA TRAER EL DETALLE DE LA FACTURA

```

MENU.CERRAR
  QUERY = "SELECT * FROM DETALLE_FACTURAS D, PRODUCTOS P
  WHERE FA_CODFACTURA = " &
  Trim(CONSULTA_FACTURA.TextMatrix(CONSULTA_FACTURA.RowSel, 1))
  & " AND P.PR_CODPRODUCTO = D.PR_CODPRODUCTO"
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

```

```

If Not MENU.REGISTRO.EOF Then
  FACTURA.DETALLE.Rows = 1
  Do While MENU.REGISTRO.EOF = False

```

'SE AGREGA EL DETALLE DE LA FACTURA AL FLEX

```

  FACTURA.DETALLE.AddItem FACTURA.DETALLE.Rows & vbTab &
  MENU.REGISTRO.Fields("PR_CODPRODUCTO") & vbTab &
  MENU.REGISTRO.Fields("PR_NOMPRODUCTO") & vbTab &
  MENU.REGISTRO.Fields("DE_CANTIDAD") & vbTab &
  MENU.REGISTRO.Fields("DE_VALORUNITARIO") & vbTab &
  MENU.REGISTRO.Fields("DE_TOTAL")
  MENU.REGISTRO.MoveNext
  Loop
  FACTURA.DETALLE.Editable = False
  FACTURA.IMPRIMIR.Enabled = False
  FACTURA.BUSQUEDA.Enabled = False
  FACTURA.TXTIDCLIENTE.Enabled = False
End If
Unload Me
End Sub

```

```

Private Sub OPTIONCODIGO_Click()
  TXTBUSCADOR.SetFocus
End Sub

```

```

Private Sub OPTIONFECH_A_Click()
  TXTBUSCADOR.SetFocus
End Sub

```

```

Private Sub TXTBUSCADOR_Change()

```

**'DEPURA LA INFORMACION POR FECHA DE GENERACION O NOMBRE DEL
CLIENTE A MEDIDA QUE SE DIGITA UNA UN CARACTER**

```
If OPTIONCODIGO.Value = True Then
    QUERY = "SELECT * FROM FACTURAS WHERE CL_IDCLIENTE LIKE '" &
    Trim(TXTBUSCADOR) & "%' "
Else
    QUERY = "SELECT * FROM FACTURAS WHERE FA_FECHA LIKE '%" &
    Format(Trim(TXTBUSCADOR), "yyyy-MM-dd") & "%'"
```

```
End If
```

```
MENU.CERRAR
```

```
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
```

```
If Not MENU.REGISTRO.EOF Then
```

```
CONSULTA_FACTURA.Rows = 1
```

```
Do While MENU.REGISTRO.EOF = False
```

'SE COLOCAN EN EL FLEX LA CONSULTA DEPURADA

```
CONSULTA_FACTURA.AddItem CONSULTA_FACTURA.Rows & vbTab &
MENU.REGISTRO.Fields("FA_CODFACTURA") & vbTab &
MENU.REGISTRO.Fields("FA_FECHA") & vbTab &
MENU.REGISTRO.Fields("CL_IDCLIENTE") & vbTab &
MENU.REGISTRO.Fields("FA_SUBTOTAL") & vbTab &
MENU.REGISTRO.Fields("FA_IVA") & vbTab &
MENU.REGISTRO.Fields("FA_TOTAL") & vbTab & ESTADO & vbTab &
MENU.REGISTRO.Fields("US_NOMBRE")
MENU.REGISTRO.MoveNext
```

```
Loop
```

```
Else
```

```
CONSULTA_FACTURA.Rows = 1
```

```
Exit Sub
```

```
End If
```

```
End Sub
```

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTA LINEAS

```
Private Sub Form_Load()
```

```
Me.Left = 0
```

```
Me.Top = 0
```

```
CONSULTA_LIN.Rows = 1
```

```
CONSULTA_LIN.Cols = 3
```

```
CONSULTA_LIN.ColWidth(0) = 800
```

```
CONSULTA_LIN.ColWidth(1) = 1000
```

```
CONSULTA_LIN.ColWidth(2) = 2000
```

```
CONSULTA_LIN.TextMatrix(0, 0) = "Item"
```

```
CONSULTA_LIN.TextMatrix(0, 1) = "Codigo"
```

```
CONSULTA_LIN.TextMatrix(0, 2) = "Nombre"
```

```
MENU.CERRAR
```

```
QUERY = "SELECT * FROM LINEAS "
```

```
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
```

```

adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
    Do While MENU.REGISTRO.EOF = False
        CONSULTA_LIN.AddItem CONSULTA_LIN.Rows & vbTab &
        MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1)
        MENU.REGISTRO.MoveNext
    Loop
Else
    MsgBox "No hay Registro de productos", vbInformation, "AVISO"
Exit Sub
End If
End Sub

Private Sub CONSULTA_LIN_Db1Click()
    LINEAS.TXTLINEA = CONSULTA_LIN.TextMatrix(CONSULTA_LIN.RowSel, 1)
    LINEAS.TXTNOMBRE = CONSULTA_LIN.TextMatrix(CONSULTA_LIN.RowSel,
2)
    LINEAS.TXTLINEA.Enabled = False
    LINEAS.MODIFICAR.Enabled = True
    LINEAS.INGRESAR.Enabled = False
    Unload Me
End Sub

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub TXTBUSCADOR_Change()
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM LINEAS WHERE LI_CODLINEA LIKE " &
        Trim(TXTBUSCADOR) & "%"
    Else
        QUERY = "SELECT * FROM LINEAS WHERE LI_NOMLINEA LIKE " &
        Trim(TXTBUSCADOR) & "%"
    End If

MENU.CERRAR
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
    CONSULTA_LIN.Rows = 1
    Do While MENU.REGISTRO.EOF = False
        CONSULTA_LIN.AddItem CONSULTA_LIN.Rows & vbTab &

```

```

    MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1)
    MENU.REGISTRO.MoveNext
  Loop
Else
  CONSULTA_LIN.Rows = 1
  Exit Sub
End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTAR PEDIDOS

```

Private Sub Form_Load()
  Me.Left = 0
  Me.Top = 0
  CONSULTA_PEDIDO.Rows = 1
  CONSULTA_PEDIDO.Cols = 9
  CONSULTA_PEDIDO.ColWidth(0) = 800
  CONSULTA_PEDIDO.ColWidth(1) = 1000
  CONSULTA_PEDIDO.ColWidth(2) = 1400
  CONSULTA_PEDIDO.ColWidth(3) = 1500
  CONSULTA_PEDIDO.ColWidth(4) = 1500
  CONSULTA_PEDIDO.ColWidth(5) = 1500
  CONSULTA_PEDIDO.ColWidth(6) = 1500
  CONSULTA_PEDIDO.ColWidth(7) = 1000
  CONSULTA_PEDIDO.ColWidth(8) = 1800
  CONSULTA_PEDIDO.TextMatrix(0, 0) = "Item"
  CONSULTA_PEDIDO.TextMatrix(0, 1) = "Codigo"
  CONSULTA_PEDIDO.TextMatrix(0, 2) = "Fecha"
  CONSULTA_PEDIDO.TextMatrix(0, 3) = "Cedula / NIT proveedor"
  CONSULTA_PEDIDO.TextMatrix(0, 4) = "Subtotal"
  CONSULTA_PEDIDO.TextMatrix(0, 5) = "IVA"
  CONSULTA_PEDIDO.TextMatrix(0, 6) = "Total"
  CONSULTA_PEDIDO.TextMatrix(0, 7) = "Estado"
  CONSULTA_PEDIDO.TextMatrix(0, 8) = "Usuario"
  MENU.CERRAR
  QUERY = "SELECT * FROM PEDIDOS"
  MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
  adLockBatchOptimistic
  If Not MENU.REGISTRO.EOF Then
    Do While MENU.REGISTRO.EOF = False
      If MENU.REGISTRO.Fields("PE_ESTADO") = 1 Then
        ESTADO = "OK"
      Else
        ESTADO = "ANULADO"
      End If
      CONSULTA_PEDIDO.AddItem CONSULTA_PEDIDO.Rows & vbTab &
      MENU.REGISTRO.Fields("PE_CODPEDIDO") & vbTab &
      MENU.REGISTRO("PE_FECHA") & vbTab &

```

```

MENU.REGISTRO.Fields("PV_CODPROVEEDOR") & vbTab &
MENU.REGISTRO.Fields("PE_SUBTOTAL") & vbTab &
MENU.REGISTRO.Fields("PE_IVA") & vbTab &
MENU.REGISTRO.Fields("PE_TOTAL") & vbTab & ESTADO & vbTab &
MENU.REGISTRO.Fields("US_NOMBRE")
MENU.REGISTRO.MoveNext
Loop
Else
MsgBox "No hay Registro de PEDIDOS", vbInformation, "AVISO"
Exit Sub
End If
End Sub

Private Sub CONSULTA_PEDIDO_Db1Click()
Dim i As Integer
PEDIDO.TXTPEDIDO =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 1)
PEDIDO.TXTFECHA =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 2)
PEDIDO.TXTPROVEEDOR =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 3)
PEDIDO.TXTSUBTOTAL =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 4)
PEDIDO.TXTIVA =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 5)
PEDIDO.TXTTOTAL =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 6)
PEDIDO.TXTUSUARIO =
CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 7)
'CONSULTA PARA TRAER EL NOMBRE DEL PROVEEDOR
MENU.CERRAR
QUERY = "SELECT PV_NOMPROVEEDOR FROM PROVEEDORES WHERE
PV_CODPROVEEDOR = " &
Trim(CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 3))
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
PEDIDO.TXTNOMBRE = MENU.REGISTRO(0)
'CONSULTA PARA TRAER EL DETALLE DEI PEDIDO
MENU.CERRAR
QUERY = "SELECT * FROM DETALLE_PEDIDOS D, PRODUCTOS P
WHERE PE_CODPEDIDO = " &
Trim(CONSULTA_PEDIDO.TextMatrix(CONSULTA_PEDIDO.RowSel, 1)) & ""
AND P.PR_CODPRODUCTO = D.PR_CODPRODUCTO"
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
PEDIDO.DETALLE.Rows = 1

```

```

Do While MENU.REGISTRO.EOF = False
    PEDIDO.DETALLE.AddItem PEDIDO.DETALLE.Rows & vbTab &
    MENU.REGISTRO.Fields("PR_CODPRODUCTO") & vbTab &
    MENU.REGISTRO.Fields("PR_NOMPRODUCTO") & vbTab &
    MENU.REGISTRO.Fields("DE_CANTIDAD") & vbTab &
    MENU.REGISTRO.Fields("DE_VALORUNITARIO") & vbTab &
    MENU.REGISTRO.Fields("DE_TOTAL")
    MENU.REGISTRO.MoveNext
Loop
PEDIDO.DETALLE.Editable = False
PEDIDO.IMPRIMIR.Enabled = False
PEDIDO.BUSQUEDA.Enabled = False
PEDIDO.TXTPROVEEDOR.Enabled = False
PEDIDO.TXTPEDIDO.Enabled = False
End If
Unload Me
End Sub

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONFECHA_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub TXTBUSCADOR_Change()
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM PEDIDOS WHERE PV_CODPROVEEDOR LIKE '"
        & Trim(TXTBUSCADOR) & "%' "
    Else
        QUERY = "SELECT * FROM PEDIDOS WHERE PE_FECHA LIKE '" &
        Format(Trim(TXTBUSCADOR), "YYYY-MM-DD") & "%' "
    End If
    MENU.CERRAR
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        CONSULTA_PEDIDO.Rows = 1
        Do While MENU.REGISTRO.EOF = False
            CONSULTA_PEDIDO.AddItem CONSULTA_PEDIDO.Rows & vbTab &
            MENU.REGISTRO.Fields("PE_CODPEDIDO") & vbTab &
            MENU.REGISTRO.Fields("PE_FECHA") & vbTab &

```

```

        MENU.REGISTRO.Fields("PV_CODPROVEEDOR") & vbTab &
        MENU.REGISTRO("PE_SUBTOTAL") & vbTab &
        MENU.REGISTRO.Fields("PE_IVA") & vbTab &
        MENU.REGISTRO.Fields("PE_TOTAL") & vbTab & ESTADO & vbTab &
        MENU.REGISTRO.Fields("US_NOMBRE")
        MENU.REGISTRO.MoveNext
    Loop
Else
    CONSULTA_PEDIDO.Rows = 1
    Exit Sub
End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTAR PRODUCTOS

```

Private Sub Form_Load()
    Me.Left = 0
    Me.Top = 0
    CONSULTA_PRODUCTO.Rows = 1
    CONSULTA_PRODUCTO.Cols = 8
    CONSULTA_PRODUCTO.ColWidth(0) = 800
    CONSULTA_PRODUCTO.ColWidth(1) = 1000
    CONSULTA_PRODUCTO.ColWidth(2) = 2800
    CONSULTA_PRODUCTO.ColWidth(3) = 1800
    CONSULTA_PRODUCTO.ColWidth(4) = 1800
    CONSULTA_PRODUCTO.ColWidth(5) = 1300
    CONSULTA_PRODUCTO.ColWidth(6) = 1300
    CONSULTA_PRODUCTO.ColWidth(7) = 1000
    CONSULTA_PRODUCTO.TextMatrix(0, 0) = "Item"
    CONSULTA_PRODUCTO.TextMatrix(0, 1) = "Codigo"
    CONSULTA_PRODUCTO.TextMatrix(0, 2) = "Nombre"
    CONSULTA_PRODUCTO.TextMatrix(0, 3) = "Linea"
    CONSULTA_PRODUCTO.TextMatrix(0, 4) = "Categoria"
    CONSULTA_PRODUCTO.TextMatrix(0, 5) = "Prec. venta"
    CONSULTA_PRODUCTO.TextMatrix(0, 6) = "Prec. compra"
    CONSULTA_PRODUCTO.TextMatrix(0, 7) = "Existencia"
'CONSULTA DE PRODUCTOS SON SUS RESPECTIVAS CATEGORIAS Y
LINEAS
    MENU.CERRAR
    QUERY = "SELECT * FROM PRODUCTOS P, LINEAS L, CATEGORIAS C
    WHERE L.LI_CODLINEA = P.LI_CODLINEA AND C.CA_CODCATEGORIA =
    P.CA_CODCATEGORIA AND L.LI_CODLINEA = C.LI_CODLINEA AND
    PR_ESTADO = 1 ORDER BY PR_NOMPRODUCTO"
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then

```

```

Do While MENU.REGISTRO.EOF = False
  If MENU.REGISTRO.Fields("PR_ESTADO") = 1 Then
    ESTADO = "ACTIVO"
  Else
    ESTADO = "INACTIVO"
  End If
'ASIGNA EL PRODUCTO AL FLEX DE CONSULTA PRODUCTOS
  CONSULTA_PRODUCTO.AddItem CONSULTA_PRODUCTO.Rows & vbTab
  & MENU.REGISTRO("PR_CODPRODUCTO") & vbTab &
  MENU.REGISTRO("PR_NOMPRODUCTO") & vbTab &
  MENU.REGISTRO("LI_NOMLINEA") & vbTab &
  MENU.REGISTRO.Fields("CA_NOMCATEGORIA") & vbTab &
  MENU.REGISTRO.Fields("PR_PRECIOVENTA") & vbTab &
  MENU.REGISTRO("PR_PRECIOCOMPRA") & vbTab &
  MENU.REGISTRO("PR_EXISTENCIA")
  MENU.REGISTRO.MoveNext
Loop
Else
  MsgBox "No hay Registro de productos", vbInformation, "AVISO"
Exit Sub
End If
End Sub

Private Sub CONSULTA_PRODUCTO_Db1Click()
  Dim i As Integer, COD_PRODUCTO, COD_IVA, COD_LINEA, COD_CATEGORIA
  If FORM_ACTUAL_PRODUCTO = "FACTURA" Then
'VALIDA QUE NO DEJE UNA FILA VACIA CUANDO SE INGRESAN LOS PRODUCTOS
'DEBE CUMPLIRSE CUANDO SE INGRESA EL PRIMER PRODUCTO
    If FACTURA.DETALLE.Rows = 2 And FACTURA.DETALLE.TextMatrix(1, 1) = "" Then
      FACTURA.DETALLE.Rows = 1
    End If
'VALIDAR QUE NO SE AGREGUEN PRODUCTOS YA SELECCIONADOS
    For i = 0 To FACTURA.DETALLE.Rows - 1
      If FACTURA.DETALLE.TextMatrix(i, 1) = CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 1) Then
        MsgBox ("Este producto ya esta en la factura")
        Exit Sub
      End If
    Next
'ASIGNA AL FORMULARIO DE FACTURAS LA FACTURA CONSULTADA
    FACTURA.DETALLE.AddItem FACTURA.DETALLE.Rows & vbTab &
    CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 1) &
    vbTab &
    CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 2) &

```

```

vbTab      &      I      &      vbTab      &
CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 5) &
vbTab      &
CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 5)
FACTURA.TXTSUBTOTAL.text = 1 * FACTURA.TXTSUBTOTAL.text -
CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 5)
FACTURA.TXTIVA.text      =      FACTURA.TXTIVA.text      +
CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 6) *
16 / 100
FACTURA.TXTTOTAL.text    =    1    *    FACTURA.TXTIVA.text    +
FACTURA.TXTSUBTOTAL.text
Unload Me
FACTURA.DETALLE.Row = FACTURA.DETALLE.Rows - 1
FACTURA.DETALLE.Col = 3

```

End If

'SI SE CARGO DEL FORM PRODUCTO

If FORM_ACTUAL_PRODUCTO = "PRODUCTO" Then

COD_PRODUCTO

CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 1)

'CONSULTA DE PRODUCTOS POR CODIGO

MENU.CERRAR

QUERY = "SELECT * FROM PRODUCTOS WHERE PR_CODPRODUCTO = "
& Trim(COD_PRODUCTO)

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

PRODUCTO.LABELEXISTENCIA.Visible = True

PRODUCTO.TXTEXISTENCIA.Visible = True

'ASIGNA LOS DATOS DEL PRODUCTOS EN EL FORMULARIO FACTURA

PRODUCTO.TXTPRODUCTO.text =

MENU.REGISTRO.Fields("PR_CODPRODUCTO")

PRODUCTO.TXTNOMBRE.text =

MENU.REGISTRO.Fields("PR_NOMPRODUCTO")

PRODUCTO.TXTCOMPRA.text =

MENU.REGISTRO.Fields("PR_PRECIOCOMPRA")

PRODUCTO.TXTVENTA.text =

MENU.REGISTRO.Fields("PR_PRECIOVENTA")

PRODUCTO.TXTEXISTENCIA.text =

MENU.REGISTRO.Fields("PR_EXISTENCIA")

PRODUCTO.TXTSTOCKMINIMO.text =

MENU.REGISTRO.Fields("PR_STOCKMINIMO")

PRODUCTO.TXTSTOCKMAXIMO.text =

MENU.REGISTRO.Fields("PR_STOCKMAXIMO")

COD_IVA = MENU.REGISTRO.Fields("IVA_CODIVA")

COD_LINEA = MENU.REGISTRO.Fields("LI_CODLINEA")

COD_CATEGORIA = MENU.REGISTRO.Fields("CA_CODCATEGORIA")

TEMP = MENU.REGISTRO.Fields("PR_ESTADO")

'ESTE CALCULO ES PARA SACAR EL PORCENTAJE PARA ESTE PRODUCTO

EN EL COMBO

```
TEMPORAL = PRODUCTO.TXTVENTA.text - PRODUCTO.TXTCOMPRA.text
```

```
PORCENTAJE = (TEMPORAL * 100) / PRODUCTO.TXTCOMPRA.text
```

```
For i = 0 To PRODUCTO.COMBOPORCENTAJE.ListCount - 1
```

'LA CONVERSION ES PARA EVITAR ESPACIOS Q SE ESTAN DANDO EN EL COMBO Y POR ESO NUNCA SE CUMPLIA LA CONDICION

```
If Cdbl(PRODUCTO.COMBOPORCENTAJE.List(i)) = PORCENTAJE Then
```

```
    PRODUCTO.COMBOPORCENTAJE.ListIndex = i
```

```
End If
```

```
Next
```

'SE SELECCIONA LO Q VIENE DE LA TABLA TENIENDO EN CUENTA EL CODIGO Y NO EL NOMBRE, POR SEGURIDAD

```
For i = 0 To PRODUCTO.COMBOIVA.ListCount - 1
```

```
If PRODUCTO.COMBOIVA.ItemData(i) = COD_IVA Then
```

```
    PRODUCTO.COMBOIVA.ListIndex = i
```

```
End If
```

```
Next
```

```
For i = 0 To PRODUCTO.COMBOLINEA.ListCount - 1
```

```
If PRODUCTO.COMBOLINEA.ItemData(i) = COD_LINEA Then
```

```
    PRODUCTO.COMBOLINEA.ListIndex = i
```

```
End If
```

```
Next
```

```
For i = 0 To PRODUCTO.COMBOCATEGORIA.ListCount - 1
```

```
If PRODUCTO.COMBOCATEGORIA.ItemData(i) = COD_CATEGORIA Then
```

```
    PRODUCTO.COMBOCATEGORIA.ListIndex = i
```

```
End If
```

```
Next
```

```
For i = 0 To PRODUCTO.COMBOBODEGA.ListCount - 1
```

```
If PRODUCTO.COMBOBODEGA.ItemData(i) = COD_BODEGA Then
```

```
    PRODUCTO.COMBOBODEGA.ListIndex = i
```

```
End If
```

```
Next
```

```
If TEMP = 1 Then
```

```
    PRODUCTO.ACTIVO.Value = True
```

```
Else
```

```
    If TEMP = 2 Then
```

```
        PRODUCTO.INACTIVO.Value = True
```

```
    End If
```

```
End If
```

```
PRODUCTO.TXTPRODUCTO.Enabled = False
```

```
PRODUCTO.INGRESAR.Enabled = False
```

```
PRODUCTO.MODIFICAR.Enabled = True
```

```
Unload Me
```

```
End If
```

'SI SE CARGO DEL FORM PEDIDO

```
If FORM_ACTUAL_PRODUCTO = "PEDIDO" Then
```

```
    If PEDIDO.DETALLE.Rows = 2 And PEDIDO.DETALLE.TextMatrix(1, 1) = ""
```

```

Then
    PEDIDO.DETALLE.Rows = 1
End If
'ESTE FOR ES PARA VALIDAR QUE NO SE AGREGUEN PRODUCTOS YA SELECCIONADOS
For i = 0 To PEDIDO.DETALLE.Rows - 1
    If PEDIDO.DETALLE.TextMatrix(i, 1) = CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 1)
        Then
            MsgBox ("Este producto ya esta en el pedido")
            Exit Sub
        End If
    Next
    PEDIDO.DETALLE.AddItem PEDIDO.DETALLE.Rows & vbTab & CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 1) & vbTab & CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 2) & vbTab & 1 & vbTab & CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 6) & vbTab & CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 6)
    PEDIDO.TXTSUBTOTAL.text = 1 * PEDIDO.TXTSUBTOTAL.text + CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 6)
    PEDIDO.TXTIVA.text = PEDIDO.TXTIVA.text + CONSULTA_PRODUCTO.TextMatrix(CONSULTA_PRODUCTO.RowSel, 6) * 16 / 100
    PEDIDO.TXTTOTAL.text = 1 * PEDIDO.TXTIVA.text + PEDIDO.TXTSUBTOTAL.text
    Unload Me
    PEDIDO.DETALLE.Row = PEDIDO.DETALLE.Rows - 1
    PEDIDO.DETALLE.Col = 3
End If
End Sub

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub TXTBUSCADOR_Change()
'DEPURA LA INFORMACION POR CODIGO DE PRODUCTO O NOMBRE DEL PRODUCTO A MEDIDA QUE SE DIGITA UNA UN CARACTER
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM PRODUCTOS P, LINEAS L, CATEGORIAS C

```

```

WHERE P.PR_CODPRODUCTO LIKE " " & Trim(TXTBUSCADOR) & "%' AND
P.PR_ESTADO = 1 AND L.LI_CODLINEA = P.LI_CODLINEA AND
C.CA_CODCATEGORIA = P.CA_CODCATEGORIA"
Else
QUERY = "SELECT * FROM PRODUCTOS P. LINEAS L. CATEGORIAS C
WHERE P.PR_NOMPRODUCTO LIKE " " & Trim(TXTBUSCADOR) & "%'
AND P.PR_ESTADO = 1 AND L.LI_CODLINEA = P.LI_CODLINEA AND
C.CA_CODCATEGORIA = P.CA_CODCATEGORIA"
End If
MENU.CERRAR
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
CONSULTA_PRODUCTO.Rows = 1
Do While MENU.REGISTRO.EOF = False
'ASIGNA AL FLEX DE CONSULTA PRODUCTOS LOS PRODUCTOS QUE
TRAIGA DE LA CONSULTA
CONSULTA_PRODUCTO.AddItem CONSULTA_PRODUCTO.Rows & vbTab
& MENU.REGISTRO("PR_CODPRODUCTO") & vbTab &
MENU.REGISTRO("PR_NOMPRODUCTO") & vbTab &
MENU.REGISTRO("LI_NOMLINEA") & vbTab &
MENU.REGISTRO.Fields("CA_NOMCATEGORIA") & vbTab &
MENU.REGISTRO.Fields("PR_PRECIOVENTA") & vbTab &
MENU.REGISTRO("PR_PRECIOCOMPRA") & vbTab &
MENU.REGISTRO("PR_EXISTENCIA")
MENU.REGISTRO.MoveNext
Loop
Else
CONSULTA_PRODUCTO.Rows = 1
Exit Sub
End If
End Sub

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTAR
PROVEEDORES
Private Sub Form_Load()
Me.Left = 0
Me.Top = 0
CONSULTAR_PROVEEDOR.KeyPreview = True
CONSULTA_PROVEEDOR.Rows = 1
CONSULTA_PROVEEDOR.Cols = 8
CONSULTA_PROVEEDOR.ColWidth(0) = 500
CONSULTA_PROVEEDOR.ColWidth(1) = 1500
CONSULTA_PROVEEDOR.ColWidth(2) = 3000
CONSULTA_PROVEEDOR.ColWidth(3) = 3500
CONSULTA_PROVEEDOR.ColWidth(4) = 1500
CONSULTA_PROVEEDOR.ColWidth(5) = 3500

```

```

CONSULTA_PROVEEDOR.ColWidth(6) = 3000
CONSULTA_PROVEEDOR.ColWidth(7) = 800
CONSULTA_PROVEEDOR.TextMatrix(0, 0) = "Item"
CONSULTA_PROVEEDOR.TextMatrix(0, 1) = "NIT"
CONSULTA_PROVEEDOR.TextMatrix(0, 2) = "Nombre proveedor"
CONSULTA_PROVEEDOR.TextMatrix(0, 3) = "Direccion"
CONSULTA_PROVEEDOR.TextMatrix(0, 4) = "Telefono"
CONSULTA_PROVEEDOR.TextMatrix(0, 5) = "E-Mail"
CONSULTA_PROVEEDOR.TextMatrix(0, 6) = "Contacto"
CONSULTA_PROVEEDOR.TextMatrix(0, 7) = "Estado"
'CONSULTA LOS DATOS DE PROVEEDORES DONDE SU ESTADO SEA
ACTIVO = 1
MENU.CERRAR
QUERY = "SELECT * FROM PROVEEDORES WHERE PV_ESTADO = 1"
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
CONSULTA_PROVEEDOR.Rows = 1
CONT = 1
Do While MENU.REGISTRO.EOF = False
'ASIGNA AL FLEX LOS VALORES DE LOS PROVEEDORES
CONSULTA_PROVEEDOR.AddItem CONT & vbTab & MENU.REGISTRO(0)
& vbTab & MENU.REGISTRO(1) & vbTab & MENU.REGISTRO(2) & vbTab &
MENU.REGISTRO(3) & vbTab & MENU.REGISTRO(4) & vbTab &
MENU.REGISTRO(5) & vbTab & MENU.REGISTRO(6)
MENU.REGISTRO.MoveNext
CONT = CONT + 1
Loop
Else
MsgBox "No hay Registro de proveedores", vbInformation, "AVISO"
CONSULTA_PROVEEDOR.Rows = 1
Exit Sub
End If
End Sub

Private Sub CONSULTA_PROVEEDOR_Db1Click()
If FORM_ACTUAL_PROVEEDOR = "PROVEEDOR" Then
'ASIGNA LOS VALORES DEL PROVEEDOR CONSULTADO AL FORMULARIO
PROVEEDORES
PROVEEDOR.TXTPROVEEDOR.text =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 1)
PROVEEDOR.TXTNOMBRE.text =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 2)
PROVEEDOR.TXTDIRECCION.text =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 3)
PROVEEDOR.TXTTELEFONO.text =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 4)

```

```

PROVEEDOR.TXTEMAIL.text =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 5)
PROVEEDOR.TXTCONTACTO.text =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 6)
TEMP =
CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 7)
If TEMP = 1 Then
    PROVEEDOR.ACTIVO.Value = True
Else
    If TEMP = 2 Then
        PROVEEDOR.INACTIVO.Value = True
    End If
End If
PROVEEDOR.TXTPROVEEDOR.Enabled = False
PROVEEDOR.TXTNOMBRE.SetFocus
PROVEEDOR.MODIFICAR.Enabled = True
PROVEEDOR.INGRESAR.Enabled = False
End If
If FORM_ACTUAL_PROVEEDOR = "PEDIDO" Then
'ASIGNA LOS VALORES DE CONSULTA PROVEEDOR AL FORMULARIO
PEDIDOS
    PEDIDO.TXTPROVEEDOR.text =
    CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 1)
    PEDIDO.TXTNOMBRE.text =
    CONSULTA_PROVEEDOR.TextMatrix(CONSULTA_PROVEEDOR.RowSel, 2)
    PEDIDO.DETALLE.SetFocus
    PEDIDO.KeyPreview = False
End If
Unload Me
End Sub

Private Sub OPTIONCODIGO_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub OPTIONNOMBRE_Click()
    TXTBUSCADOR.SetFocus
End Sub

Private Sub TXTBUSCADOR_Change()
'DEPURA LA INFORMACION POR CODIGO DEL PROVEEDOR O NOMBRE
DEL PROVEEDOR A MEDIDA QUE SE DIGITA UNA UN CARACTER
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM PROVEEDORES WHERE PV_CODPROVEEDOR"
        QUERY = QUERY & " LIKE '" & Trim(TXTBUSCADOR) & "%' AND PV_ESTADO = 1 "
    Else
        QUERY = "SELECT * FROM PROVEEDORES WHERE

```

```

        PV_NOMPROVEEDOR LIKE " " & Trim(TXTBUSCADOR) & "%' AND
        PV_ESTADO = 1 "
    End If
    MENU.CERRAR
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        CONSULTA_PROVEEDOR.Rows = 1
        Do While MENU.REGISTRO.EOF = False
'ASIGNA AL FLEX LOS DATOS DEPURADOS
            CONSULTA_PROVEEDOR.AddItem CONSULTA_PROVEEDOR.Rows &
vbTab & MENU.REGISTRO(0) & vbTab & MENU.REGISTRO(1) & vbTab &
MENU.REGISTRO(2) & vbTab & MENU.REGISTRO(3) & vbTab &
MENU.REGISTRO(4) & vbTab & MENU.REGISTRO(5) & vbTab &
MENU.REGISTRO(6)
            MENU.REGISTRO.MoveNext
        Loop
    Else
        CONSULTA_PROVEEDOR.Rows = 1
        Exit Sub
    End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CONSULTA USUARIOS

```

Private Sub Form_Load()
    Me.Left = 0
    Me.Top = 0
    CONSULTAR_USUARIOS.KeyPreview = True
    CONSULTA_USUARIOS.Rows = 1
    CONSULTA_USUARIOS.Cols = 7
    CONSULTA_USUARIOS.ColWidth(0) = 500
    CONSULTA_USUARIOS.ColWidth(1) = 1500
    CONSULTA_USUARIOS.ColWidth(2) = 2000
    CONSULTA_USUARIOS.ColWidth(3) = 2000
    CONSULTA_USUARIOS.ColWidth(4) = 2200
    CONSULTA_USUARIOS.ColWidth(5) = 1800
    CONSULTA_USUARIOS.ColWidth(6) = 4000
    CONSULTA_USUARIOS.TextMatrix(0, 0) = "Item"
    CONSULTA_USUARIOS.TextMatrix(0, 1) = "Cedula / nit"
    CONSULTA_USUARIOS.TextMatrix(0, 2) = "Nombres"
    CONSULTA_USUARIOS.TextMatrix(0, 3) = "Apellidos"
    CONSULTA_USUARIOS.TextMatrix(0, 4) = "Direccion"
    CONSULTA_USUARIOS.TextMatrix(0, 5) = "Telefono"
    CONSULTA_USUARIOS.TextMatrix(0, 6) = "E-Mail"
    MENU.CERRAR
    QUERY = "SELECT * FROM EMPLEADOS E, USUARIOS U WHERE
        E.EM_CEDULA = U.EM_CEDULA AND US_ESTADO = 1 "

```

```

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
CONSULTA_USUARIOS.Rows = 1
CONT = 1
Do While MENU.REGISTRO.EOF = False
CONSULTA_USUARIOS.AddItem CONT & vbTab & MENU.REGISTRO(0) &
vbTab & MENU.REGISTRO(1) & vbTab & MENU.REGISTRO(2) & vbTab &
MENU.REGISTRO(3) & vbTab & MENU.REGISTRO(4) & vbTab &
MENU.REGISTRO(5) & vbTab & MENU.REGISTRO(6)
MENU.REGISTRO.MoveNext
CONT = CONT + 1
Loop
Else
MsgBox "No hay Registro de empleados", vbInformation, "AVISO"
CONSULTA_USUARIOS.Rows = 1
Text2.SetFocus
Exit Sub
End If
End Sub

Private Sub CONSULTA_USUARIOS_DblClick()
USUARIOS.TXTCEDULA.text =
CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel, 1)
USUARIOS.TXTNOMBRES.text =
CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel, 2)
USUARIOS.TXTAPELLIDOS.text =
CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel, 3)
USUARIOS.TXTDIRECCION.text =
CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel, 4)
USUARIOS.TXTTELEFONO.text =
CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel, 5)
USUARIOS.TXTEMAIL.text =
CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel, 6)
MENU.CERRAR
QUERY = "SELECT * FROM USUARIOS WHERE EM_CEDULA = " &
Trim(CONSULTA_USUARIOS.TextMatrix(CONSULTA_USUARIOS.RowSel,
1))
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
USUARIOS.TXTUSUARIO = MENU.REGISTRO.Fields("US_NOMBRE")
USUARIOS.TXTCONTRASENA = MENU.REGISTRO.Fields("US_CLAVE")
USUARIOS.TXTCONFIRMACION = MENU.REGISTRO.Fields("US_CLAVE")
TEMPTIPOUSU = MENU.REGISTRO.Fields("TU_CODIGO")
TEMP = MENU.REGISTRO.Fields("US_ESTADO")
End If

```

```

For i = 0 To USUARIOS.COMBOTIPO_USUARIO.ListCount - 1
    If USUARIOS.COMBOTIPO_USUARIO.ItemData(i) = TEMPTIPOUSU Then
        USUARIOS.COMBOTIPO_USUARIO.ListIndex = i
    End If
Next
If TEMP = 1 Then
    USUARIOS.ACTIVO.Value = True
Else
    If TEMP = 2 Then
        USUARIOS.INACTIVO.Value = True
    End If
End If
USUARIOS.TXTCEDULA.Enabled = False
USUARIOS.TXTNOMBRES.SetFocus
USUARIOS.MODIFICAR.Enabled = True
USUARIOS.INGRESAR.Enabled = False
Unload Me
End Sub

Private Sub TXTBUSCADOR_Change()
    MENU.CERRAR
    If OPTIONCODIGO.Value = True Then
        QUERY = "SELECT * FROM EMPLEADOS WHERE EM_CEDULA LIKE '" &
            Trim(TXTBUSCADOR) & "'"
    Else
        QUERY = "SELECT * FROM EMPLEADOS WHERE EM_NOMBRE LIKE '" &
            Trim(TXTBUSCADOR) & " '"
    End If
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
        adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        CONSULTA_USUARIOS.Rows = 1
        Do While MENU.REGISTRO.EOF = False
            CONSULTA_USUARIOS.AddItem CONSULTA_USUARIOS.Rows & vbTab &
                MENU.REGISTRO.Fields("EM_CEDULA") & vbTab &
                MENU.REGISTRO.Fields("EM_NOMBRE") & vbTab &
                MENU.REGISTRO.Fields("EM_APELLIDO") & vbTab &
                MENU.REGISTRO.Fields("EM_DIRECCION") & vbTab &
                MENU.REGISTRO.Fields("EM_TELEFONO") & vbTab &
                MENU.REGISTRO.Fields("EM_IMAIL")
            MENU.REGISTRO.MoveNext
        Loop
    Else
        CONSULTA_USUARIOS.Rows = 1
    End If
Exit Sub
End Sub
End Sub

```

AL MOMENTO DE CARGAR EL FORMULARIO CAMBIAR CONTRASEÑA

```

Private Sub Form_Load()
    Me.Top = 0
    Me.Left = 0
    ACTUALIZAR.Enabled = False
    CONTRASENA.KeyPreview = True
    TXTUSUARIO = US_NOMBRE
End Sub

Private Sub ACTUALIZAR_Click()
    If TXTACTUAL = "" Then
        MsgBox "Digite la contraseña actual", vbInformation, "AVISO"
        TXTACTUAL.SetFocus
        Exit Sub
    Else
        If TXTNUEVA = "" Then
            MsgBox "Digite la nueva contraseña", vbInformation, "AVISO"
            TXTNUEVA.SetFocus
            Exit Sub
        Else
            If TXTCONFIRMACION = "" Then
                MsgBox "Confirme la contraseña nueva", vbInformation, "AVISO"
                TXTCONFIRMACION.SetFocus
                Exit Sub
            Else
                MENU.CERRAR
                QUERY = "SELECT US_NOMBRE, US_CLAVE FROM USUARIOS
                WHERE US_NOMBRE= '" & Trim(TXTUSUARIO) & "'"
                MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
                adLockBatchOptimistic
                If Not MENU.REGISTRO.EOF Then
                    CONACTUAL = EncryptString(MENU.REGISTRO.Fields("US_CLAVE").
                    "UJUU")
                    If TXTACTUAL.text <> CONACTUAL Then
                        MsgBox "La contraseña digitada no coincide con la contraseña actual",
                        vbInformation, "AVISO"
                        TXTACTUAL = ""
                        TXTACTUAL.SetFocus
                        Exit Sub
                    Else
                        If TXTCONFIRMACION.text <> TXTNUEVA.text Then
                            MsgBox "La confirmacion no coincide con la contraseña nueva",
                            vbInformation, "AVISO"
                            Exit Sub
                        End If
                    End If
                End If
            End If
        End If
    End If
End Sub

```

```

    TextoCodificado = EncryptString(TXTNUEVA, "UJUJU")
    MENU.CERRAR
        MENU.CONEXION.Execute ("UPDATE USUARIOS SET US_CLAVE =
        " & Trim(TextoCodificado) & " WHERE US_NOMBRE = " &
        Trim(TXTUSUARIO) & " ")
        MsgBox "Contraseña modificada correctamente", vbInformation,
        "Confirmacion"
        TXTACTUAL = ""
        TXTCONFIRMACION = ""
        TXTNUEVA = ""
        ACTUALIZAR.Enabled = False
        TXTACTUAL.SetFocus
    End If
End If
End If
End Sub

Private Sub LIMPIAR_Click()
    TXTACTUAL = ""
    TXTNUEVA = ""
    TXTCONFIRMACION = ""
    ACTUALIZAR.Enabled = False
    TXTACTUAL.SetFocus
End Sub

Private Sub TXTACTUAL_LostFocus()
    If TXTACTUAL <> "" Then
        MENU.CERRAR
            QUERY = "SELECT US_CLAVE FROM USUARIOS WHERE US_NOMBRE =
            " & Trim(TXTUSUARIO) & ""
            MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
            If Not MENU.REGISTRO.EOF Then
                TEMPACTUAL = EncryptString(MENU.REGISTRO.Fields("US_CLAVE").
                "UJUJU")
                If TXTACTUAL <> TEMPACTUAL Then
                    MsgBox "La contraseña no coincide con la del usuario", vbInformation,
                    "AVISO"
                    TXTACTUAL.SetFocus
                    ACTUALIZAR.Enabled = False
                    Exit Sub
                End If
            End If
        End If
    Else
        ACTUALIZAR.Enabled = True
    End If
End Sub

```

End Sub

```
Private Sub TXTCONFIRMACION_LostFocus()
    ACTUALIZAR.Enabled = True
End Sub
```

'DECLARAR VARIABLES PARA ESTE FORMULARIO

```
Public con As New ADODB.Connection
Public COM As New ADODB.Command
Public REG1 As New ADODB.Recordset
Public COM1 As New ADODB.Command
Public NOMB As String
Public tabla As String
```

'AL MOMENTO DE CARGAR DEL FORMULARIO EXPORTAR/IMPORTAR DATOS

```
Private Sub Form_Load()
    con.Open CONEC.ConnectionString
    Dim NODOS As Node
    Me.Left = 0
    Me.Top = 0
    On Error GoTo ERRORES
    MENU.CERRAR
    COM.CommandText = "SHOW TABLES"
    MENU.REGISTRO.Open COM.CommandText, con, adOpenDynamic,
    adLockBatchOptimistic
    While Not MENU.REGISTRO.EOF
        LISTABD.AddItem (MENU.REGISTRO!tables_in_distribucion)
        MENU.REGISTRO.MoveNext
    Wend
ERRORES:
    If Err.Number <> 0 Then
        MsgBox Err.Description & " Proceso cancelado", vbInformation, "EXPORTACION"
    End If
End Sub
```

```
Private Sub CHECKTODAS_Click()
    If CHECKTODAS.Value Then
        For i = 0 To LISTABD.ListCount - 1
            LISTABD.Selected(i) = True
        Next i
    Else
        For i = 0 To LISTABD.ListCount - 1
            LISTABD.Selected(i) = False
        Next i
    End If
```

End Sub

```
Private Sub Form_Unload(Cancel As Integer)
    con.Close
End Sub
```

```
Private Sub BOTEXPORTAR_Click()
    EXPORTAR
    For i = 0 To LISTABD.ListCount - 1
        LISTABD.Selected(i) = False
    Next i
    CHECKTODAS.Value = False
End Sub
```

```
Private Sub BOTIMPORTAR_Click()
    TABLA_S.Show
End Sub
```

```
Private Sub BOTLIMPIAR_Click()
    ERROR.Clear
    CHECKTODAS.Value = False
    For i = 0 To LISTABD.ListCount - 1
        LISTABD.Selected(i) = False
    Next i
End Sub
```

```
Private Sub EXPORTAR()
    Dim OUTFILE, i, J As Long
    OUTFILE = FreeFile()
    CONT = 0
    For i = 0 To LISTABD.ListCount - 1
        If LISTABD.Selected(i) Then
            CONT = CONT + 1
        End If
    Next i
    If CONT = 0 Then
        MsgBox "Debe Seleccionar al menos una Tabla a Exportar", vbInformation, "AVISO"
        Exit Sub
    End If
    With DIGARCHIVO
        .Filter = "ARCHIVOS DE PRUEBA|*.txt"
        .ShowSave
        Archivo = .FileName
        If Archivo <> "" Then
            Open Archivo For Output As #OUTFILE
        Else
```

```

    Exit Sub
End If
End With
For i = 0 To LISTABD.ListCount - 1
    If LISTABD.Selected(i) Then
        COM.CommandText = "SELECT * FROM " & LISTABD.List(i)
        MENU.CERRAR
        MENU.REGISTRO.Open COM.CommandText, MENU.CONEXION,
        adOpenDynamic, adLockBatchOptimistic
        While MENU.REGISTRO.EOF = False
            TXTLINEA = ""
            For k = 0 To MENU.REGISTRO.Fields.Count - 1
                TXTLINEA = TXTLINEA & MENU.REGISTRO.Fields.Item(k).Value
                If (k < (MENU.REGISTRO.Fields.Count - 1)) Then
                    TXTLINEA = TXTLINEA & ","
                End If
            Next k
            Print #OUTFILE, TXTLINEA
            MENU.REGISTRO.MoveNext
        Wend
    End If
Next i
MsgBox "DATOS EXPORTADOS", vbInformation, "IMPORT/EXPORT"
Close #OUTFILE
Exit Sub
Close #OUTFILE
End Sub

Public Sub IMPORTAR()
    Dim FN As String, INFILE, NUM_CAR, CARACTERES, LINEAS As Long
    Dim TXTLINEA As String
    Dim TXTDATOS() As String, txt() As String
    Dim ERRORES, ERRORDATOS As Boolean
    On Error GoTo ERRORINPUT
    ERRORDATOS = True
    ERRORES = True
    With DIGARCHIVO
        .DialogTitle = "SELECCIONE EL ARCHIVO A PROCESAR"
        .Filter = "ARCHIVOS DE PRUEBA|*.txt"
        .ShowOpen
        FN = .FileName

        If FN = "" Then
            Exit Sub
        End If
    End With
    If FN <> vbNullString Then

```

```

INFILE = FreeFile()
Open FN For Input As #INFILE
If NOMB = "" Then
    MsgBox "DEBE ESPECIFICAR LA TABLA DONDE DESEA IMPORTAR LOS
    DATOS", vbInformation, "INPOT/EXPORT"
    Exit Sub
End If
MENU.CERRAR
COM.CommandText = "DESC " & NOMB
MENU.REGISTRO.Open COM.CommandText, con, adOpenDynamic,
adLockBatchOptimistic
LINEAS = 0
ERROR.Clear
Do While Not EOF(INFILE)
    LINEAS = LINEAS + 1
    Line Input #INFILE, TXTLINEA
    TXTDATOS = Split(TXTLINEA, ",")
    NUM_CAMPOS = CONTAR_CAMPOS(TXTLINEA)
    CONT = 0
    MENU.REGISTRO.MoveFirst
    While Not MENU.REGISTRO.EOF
        CONT = Val(CONT) + 1
        MENU.REGISTRO.MoveNext
    Wend
    If NUM_CAMPOS = CONT - 1 Then
        MENU.REGISTRO.MoveFirst
        For i = 1 To CONT
            txt = Split(MENU.REGISTRO!Type, "(")
            CAR = Trim(txt(0))
            Select Case CAR
                Case "varchar"
                    NUM_CAR = Val(txt(1))
                    CARACTERES = Len(TXTDATOS(i - 1))
                    If NUM_CAR >= CARACTERES Then
                        If MENU.REGISTRO!Null = "" Then
                            If TXTDATOS(i - 1) = "" Then
                                ERRORES = False
                                'ERROR
                                CRITICOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
                                RROR.AddItem "Error en la linea " & LINEAS & ". " &
                                TXTLINEA & " EL CAMPO " & REG.Fields.Item(0) & "
                                NO PUEDE SER NULO."
                            End If
                        End If
                    Else
                        'ERROR
                        GRAVISIMOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
                        ERROR.AddItem "Error en la Linea " & LINEAS & ". " & "El

```

```

        Campo " & TXTDATOS(i - 1) & " solo puede contener " &
        NUM_CAR & " Caracteres."
        ERRORES = False
    End If
Case "int"
    NUM_CAR = Val(txt(1))
    CARACTERES = Len(TXTDATOS(i - 1))
    If IsNumeric(TXTDATOS(i - 1)) Then
        If NUM_CAR >= CARACTERES Then
            If MENU.REGISTRO!Null = "" Then
                If TXTDATOS(i - 1) = "" Then
                    ERRORES = False 'ERROR
                    CRITICOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
                End If
            End If
        Else
            'ERROR
            GRAVISIMOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
            ERRORES = False
        End If
    Else
        'ERROR
        CRITICOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
        ERRORES = False
        ERROR.AddItem "Error en la linea " & LINEAS & ". " &
        TXTLINEA & " El campo " & MENU.REGISTRO.Fields.Item(0) &
        "(" & TXTDATOS(i - 1) & ") debe ser numerico"
    End If
End Select
MENU.REGISTRO.MoveNext
Next i
ElseIf NUM_CAMPOS < CONT - 1 Then
    ERRORES = False
    ERROR.AddItem "Error en la linea " & LINEAS & ". " & TXTLINEA & "
    NUMERO DE COLUMNAS INVALIDOS "
    MENU.REGISTRO.MoveFirst
    For j = 1 To CONT - 1
        txt = Split(MENU.REGISTRO!Type, "(")
        CAR = Trim(txt(0))
        Select Case CAR
            Case "varchar"
                NUM_CAR = Val(txt(1))
                CARACTERES = Len(TXTDATOS(i - 1))
                If NUM_CAR >= CARACTERES Then
                    If MENU.REGISTRO!Null = "" Then
                        If TXTDATOS(i - 1) = "" Then
                            ERRORES = False 'ERROR

```

```

CRITICOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
'MsgBox "ESTE CAMPO NO PUEDE SER NULO "
ERROR.AddItem "Error en la linea " & LINEAS & ". " &
'TXTLINEA & " EL CAMPO " & REG.Fields.Item(0) & "
NO PUEDE SER NULO."
End If
End If
Else
'ERROR
GRAVISIMOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
ERROR.AddItem "Error en la Linea " & LINEAS & ". " & "El Campo
" & TXTDATOS(i - 1) & " solo puede contener " & NUM_CAR & "
Caracteres."
ERRORES = False
End If
Case "int"
NUM_CAR = Val(txt(1))
CARACTERES = Len(TXTDATOS(i - 1))
If IsNumeric(TXTDATOS(i - 1)) Then
If NUM_CAR >= CARACTERES Then
If MENU.REGISTRO!Null = "" Then
If TXTDATOS(i - 1) = "" Then
ERRORES = False 'ERROR
CRITICOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
End If
End If
Else
'ERROR
GRAVISIMOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
ERRORES = False
End If
Else
'ERROR
CRITICOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
ERRORES = False
ERROR.AddItem "Error en la linea " & LINEAS & ". " &
TXTLINEA & " El campo " & REG.Fields.Item(0) & "(" &
TXTDATOS(i - 1) & ") debe ser numerico"
End If
End Select
MENU.REGISTRO.MoveNext
Next i
Else
ERRORES = False
ERROR.AddItem "Error en la linea " & LINEAS & ". " & TXTLINEA & "
NUMERO DE COLUMNAS INVALIDOS "
REG.MoveFirst

```



```

        & " El campo " & REG.Fields.Item(0) & "(" & TXTDATOS(i - 1) &
        ") debe ser numerico"
        ERRORES = False
    End If
End Select
REG.MoveNext
Next i
End If
Loop
Close #INFILE
If ERRORES Then
    LINEAS = 0
    INFILE = FreeFile()
    Open FN For Input As #INFILE
    MENU.REGISTRO.MoveFirst
    Do While Not EOF(INFILE)
        Line Input #INFILE, TXTLINEA
        LINEAS = LINEAS + 1
        TXTDATOS = Split(TXTLINEA, ",")
        COM.CommandText = "INSERT INTO " & NOMB & " VALUES ("
        For i = 0 To CONT - 1
            COM.CommandText = COM.CommandText & TXTDATOS(i)
            If i < CONT - 1 Then
                COM.CommandText = COM.CommandText & ", "
            End If
        Next i
        COM.CommandText = COM.CommandText & ")"
        COM1.CommandText = "SELECT * FROM " & NOMB & " WHERE " &
            MENU.REGISTRO.Fields(0) & "='" & TXTDATOS(0) & "'"
        REG1.Open COM1.CommandText, con, adOpenDynamic,
            adLockBatchOptimistic
        If REG1.EOF Then
            con.Execute COM.CommandText
        Else
            ERROR.AddItem "Error en la Linea " & LINEAS & "." & " No se pudo
                duplicar el Registro conCodigo " & TXTDATOS(0) & "."
            ERRORDATOS = False
        End If
        REG1.Close
    Loop
    Close #INFILE
    If ERRORDATOS = True Then
        ERROR.AddItem "0 ERRORES"
        MsgBox "LOS DATOS FUERON CARGADOS
            SATISFACTORIAMENTE", vbInformation, "INPORT/EXPORT"
    Else
        MsgBox "ES PROBLABLE QUE TODOS LOS REGISTROS NO SE

```

```

        HAYAN CARGADO SATISFACTORIAMENTE", vbInformation,
        "INPORT/EXPORT"
    End If
Else
    MsgBox "Los registros no se cargaron en la Base de Datos debido a errores
    presentados durante la Validacion del sistema ", vbCritical, "INPORT/EXPORT"
End If
End If
    ERRORINPUT:
If Err.Number <> 0 Then
    MsgBox Err.Description & ". Proceso Cancelado", vbCritical, "IMPORT/EXPORT"
    Close #INFILE
End If
End Sub

Private Function CONTAR_CAMPOS(ByVal CADENA As String) As Long
    CONTAR_CAMPOS = 0
    CONTAR = 0
    For i = 1 To Len(CADENA)
        If Mid(CADENA, i, 1) = "," Then
            CONTAR = Val(CONTAR) + 1
        End If
    Next i
    CONTAR_CAMPOS = CONTAR
End Function

```

'AL MOMENTO DE CARGAR EL FORMULARIO FACTURACION

```

Private Sub Form_Load()
    DATO = Date
    FACTURA.KeyPreview = True
    Me.Left = 0
    Me.Top = 0
    TXTUSUARIO = US_NOMBRE
    TXTFECHA = DATO
    DETALLE.Rows = 2
    DETALLE.Cols = 6
    DETALLE.ColWidth(0) = 500
    DETALLE.ColWidth(1) = 1000
    DETALLE.ColWidth(2) = 4000
    DETALLE.ColWidth(3) = 800
    DETALLE.ColWidth(4) = 2000
    DETALLE.ColWidth(5) = 2000
    DETALLE.TextMatrix(0, 0) = "Item"
    DETALLE.TextMatrix(0, 1) = "Codigo"
    DETALLE.TextMatrix(0, 2) = "Producto"
    DETALLE.TextMatrix(0, 3) = "Cantidad"
    DETALLE.TextMatrix(0, 4) = "Valor unitario"

```

```

DETALLE.TextMatrix(0, 5) = "Total"
DETALLE.TextMatrix(1, 0) = "1"
MENU.CERRAR
QUERY = "SELECT * FROM RANGO_FACTURAS"
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
        adLockBatchOptimistic, -1
If Not MENU.REGISTRO.EOF Then
    TEMPFACTURA = MENU.REGISTRO(0)
    TEMPINICIAL = MENU.REGISTRO(1)
    TEMPFINAL = MENU.REGISTRO(2)
Else
    MsgBox "No han asignado rango de facturas", vbInformation, "AVISO"
    Exit Sub
End If

```

'CARGAR EL FORMULARIO DE CONSULTA DE CLIENTES

```

Private Sub BUSQUEDA_Click()
    FORM_ACTUAL_CLIENTE = "FACTURA"
    CONSULTAR_CLIENTE.Show
    CONSULTAR_CLIENTE.ZOrder
End Sub

```

'CONSULTA DE FACTURAS GENERADAS

```

Private Sub CONSULTAR_Click()
    CONSULTAR_FACTURA.Show
End Sub

```

```

Private Sub DETALLE_Click()
    FACTURA.KeyPreview = False
End Sub

```

```

Private Sub DETALLE_KeyDownEdit(ByVal Row As Long, ByVal Col As Long,
    KeyCode As Integer, ByVal Shift As Integer)

```

'TRAER EL FORMULARIO DE CONSULTA DE PRODUCTOS CON LA TECLA F3

```

    If KeyCode = 114 And DETALLE.Editable = True Then
        FORM_ACTUAL_PRODUCTO = "FACTURA"
        CONSULTAR_PRODUCTO.Show
    End If

```

'PARA EVITAR SUPRIMIR LA INFORMACION DEL PRODUCTO EN EL GRID

```

    If KeyCode = 46 Then
        KeyCode = 113
    End If
End Sub

```

'SE CONTROLA Q SOLO SE DIGITEN NUMEROS EN LA COLUMNA CANTIDAD DEL GRID

```
Private Sub DETALLE_KeyPressEdit(ByVal Row As Long, ByVal Col As Long,
KeyAscii As Integer)
```

```
    If DETALLE.Col = 3 Then
```

```
        'LA TERCERA CONDICION VALIDA Q EL COSTO UNITARIO NO SEA VACIO
        (Q EXISTA UNA FILA)
```

```
        If DETALLE.TextMatrix(DETALLE.RowSel, 4) = "" Or Not
(IsNumeric(Chr(KeyAscii))) And KeyAscii <> 13 And KeyAscii <> 8 Then
```

```
            KeyAscii = 0
```

```
        Else
```

```
            If KeyAscii = 13 And DETALLE.TextMatrix(DETALLE.RowSel,
DETALLE.ColSel) <> "" Then
```

```
                'VALIDA QUE LA CANTIDAD DE PRODUCTO NO SEA 0
```

```
                If DETALLE.TextMatrix(DETALLE.RowSel, DETALLE.ColSel) = "0"
Then
```

```
                    MsgBox ("La cantidad no puede ser 0")
```

```
                    Exit Sub
```

```
                End If
```

```
                COSTO = DETALLE.TextMatrix(DETALLE.RowSel, DETALLE.ColSel)
* DETALLE.TextMatrix(DETALLE.RowSel, 4)
```

```
                DETALLE.TextMatrix(DETALLE.RowSel, 5) = COSTO
```

```
                'CALCULO DEL SUBTOTAL SUMANDO TODA LA COLUMNA "TOTAL" DEL
GRID
```

```
                    TEMPSUBTOTAL = 0
```

```
                    For i = 1 To DETALLE.Rows - 1
```

```
                        TEMPSUBTOTAL = TEMPSUBTOTAL + DETALLE.TextMatrix(i, 5)
```

```
                    Next
```

```
                    TXTSUBTOTAL = TEMPSUBTOTAL
```

```
                    TXTIVA = FACTURA.TXTSUBTOTAL * 16 / 100
```

```
                    TXTTOTAL = TEMPSUBTOTAL + TXTIVA
```

```
                End If
```

```
            End If
```

```
        Else
```

```
            KeyAscii = 0
```

```
        End If
```

```
    End Sub
```

```
Private Sub IMPRIMIR_Click()
```

```
    If TXTFACTURA <> "" And TXTIDCLIENTE <> "" And DETALLE.Rows >= 2
And DETALLE.TextMatrix(1, 5) <> "" Then
```

```
        'LO PRIMERO Q SE HACE ES VALIDAR Q EL DETALLE (LOS PRODUCTOS)
EXISTAN EN LA BD
```

```
        For i = 1 To DETALLE.Rows - 1
```

```
            MENU.CERRAR
```

```
            QUERY = "SELECT * FROM PRODUCTOS WHERE
PR_CODPRODUCTO = " & Trim(DETALLE.TextMatrix(i, 1))
```

```
            MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
```

```

TEMPEXISTENCIA = MENU.REGISTRO.Fields("PR_EXISTENCIA")
TEMPNOMPRODUCTO = MENU.REGISTRO.Fields("PR_NOMPRODUCTO")
TEMPSTOCKMINIMO = MENU.REGISTRO.Fields("PR_STOCKMINIMO")
'VALIDAR LA EXISTENCIA DE LOS PRÓDUCTOS EN LAS BODEGAS
If TEMPEXISTENCIA < CInt(DETALLE.TextMatrix(i, 3)) Then
    MsgBox ("En bodega solo hay " + CStr(TEMPEXISTENCIA) + " "
    + TEMPNOMPRODUCTO)
    DETALLE.Row = i
    DETALLE.Col = 3
    Exit Sub
Else
    If TEMPEXISTENCIA - CInt(DETALLE.TextMatrix(i, 3)) <
    TEMPSTOCKMINIMO Then
        RESPUESTA = MsgBox("La existencia del producto " -
        TEMPNOMPRODUCTO + " no satisface el stockminimo (" +
        CStr(TEMPSTOCKMINIMO) + ") , desea continuar ?", vbYesNo,
        "CONFIRMACION")
        If RESPUESTA <> 6 Then
            Exit Sub
        End If
    End If
End If
Next
For i = 1 To DETALLE.Rows - 1
    MENU.CERRAR
    QUERY = "SELECT PR_EXISTENCIA, PR_NOMPRODUCTO FROM
    PRODUCTOS WHERE PR_CODPRODUCTO = " &
    Trim(DETALLE.TextMatrix(i, 1))
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    EXISTENCIAACTUAL = MENU.REGISTRO.Fields("PR_EXISTENCIA")
    EXISTENCIANUEVA = EXISTENCIAACTUAL -
    CInt(DETALLE.TextMatrix(i, 3))
'ACTUALIZO LA EXISTENCIA DE LOS PRODUCTOS
    MENU.CERRAR
    MENU.CONEXION.Execute ("UPDATE PRODUCTOS SET
    PR_EXISTENCIA = " & Trim(EXISTENCIANUEVA) & " WHERE
    PR_CODPRODUCTO = " & Trim(DETALLE.TextMatrix(i, 1)))
'SE INGRESA EN EL KARDEN
    MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO
    KARDEN(KD_FECHA,KD_CANTIDAD,KD_PRECIO,PR_CODPRODU
    CTO,TM_CODTIPOMOV,US_NOMBRE) VALUES (" &
    Trim(Format(TXTFECHA, "YYYY.MM.DD")) & ", " &

```

```
Trim(DETALLE.TextMatrix(i, 3)) & "," & Trim(DETALLE.TextMatrix(i,
5)) & "," & Trim(DETALLE.TextMatrix(i, 1)) & "," & Trim(3) & "," &
Trim(TXTUSUARIO) & ")")
```

```
Next
```

```
TEMPESTADO = 1
```

```
'SE INGRESAN DATOS EN LA TABLA FACTURAS
```

```
MENU.CERRAR
```

```
MENU.CONEXION.Execute ("INSERT INTO FACTURAS VALUES (" &
Trim(TXTFACTURA) & "," & Trim(Format(TXTFECHA,
"YYYY,MM,DD")) & "," & Round(Trim(TXTSUBTOTAL)) & "," &
Round(Trim(TXTIVA)) & "," & Round(Trim(TXTTOTAL)) & "," &
Trim(0) & "," & Trim(TEMPESTADO) & "," & Trim(TXTIDCLIENTE) &
"," & UCase(Trim(TXTUSUARIO)) & ")")
```

```
'SE INGRESAN DATOS EN LA TABLA AUDITORIA
```

```
MENU.CERRAR
```

```
MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('INSERCIÓN','FACTURA'," & Trim(Format(Date, "YYYY-MM-DD")) &
"," & Trim(Format(Time, "hh:mm:ss")) & "," &
UCase(Trim(US_NOMBRE)) & ")")
```

```
For i = 1 To DETALLE.Rows - 1
```

```
'SE INGRESAN DATOS EN LA TABLA DETALLE_FACTURAS
```

```
MENU.CERRAR
```

```
MENU.CONEXION.Execute ("INSERT INTO
DETALLE_FACTURAS VALUES (" & Trim(TXTFACTURA) &
"," & Trim(DETALLE.TextMatrix(i, 1)) & "," &
Trim(DETALLE.TextMatrix(i, 4)) & "," &
Trim(DETALLE.TextMatrix(i, 3)) & "," &
Trim(DETALLE.TextMatrix(i, 5)) & ")")
```

```
Next
```

```
DETALLE.Rows = 2
```

```
DETALLE.TextMatrix(1, 1) = ""
```

```
DETALLE.TextMatrix(1, 2) = ""
```

```
DETALLE.TextMatrix(1, 3) = ""
```

```
DETALLE.TextMatrix(1, 4) = ""
```

```
DETALLE.TextMatrix(1, 5) = ""
```

```
DETALLE.Cols = 6
```

```
TEMPFACTURA = Val(TXTFACTURA)
```

```
TXTFACTURA = Val(TEMPFACTURA) + 1
```

```
TXTIDCLIENTE = ""
```

```
TXTNOMBRE = ""
```

```
TXTSUBTOTAL = 0
```

```
TXTIVA = 0
```

```
TXTTOTAL = 0
```

```
TXTIDCLIENTE.SetFocus
```

```
FACTURA.KeyPreview = True
```

'SE MUESTRA LA FACTURA EN UN REPORTE - CRYSTAL REPORT

```

MENU.CrystalReport.ReportFileName = App.Path & "\REPORTES\FACTURA.rpt"
MENU.CrystalReport.SelectionFormula = "{facturas.fa_codfactura}=" &
(TEMPFACTURA)
MENU.CrystalReport.WindowState = crptNormal
MENU.CrystalReport.Action = 1
Else
MsgBox "Complete los datos de la factura", vbInformation, "AVISO"
If TXTIDCLIENTE = "" Then
TXTIDCLIENTE.SetFocus
Else
DETALLE.SetFocus
DETALLE.Row = 1
End If
End If
End Sub

```

'DEJA EL DOCUMENTO EN BLANCO PARA GENERAR NUEVAMENTE LA MISMA FACTURA

```

Private Sub LIMPIAR_Click()
DETALLE.Editable = True
DETALLE.Rows = 2
DETALLE.Cols = 6
DETALLE.TextMatrix(1, 1) = ""
DETALLE.TextMatrix(1, 2) = ""
DETALLE.TextMatrix(1, 3) = ""
DETALLE.TextMatrix(1, 4) = ""
DETALLE.TextMatrix(1, 5) = ""
TXTIDCLIENTE = ""
TXTNOMBRE = ""
TXTSUBTOTAL = 0
TXTIVA = 0
TXTTOTAL = 0
IMPRIMIR.Enabled = True
FACTURA.BUSQUEDA.Enabled = True
FACTURA.TXTIDCLIENTE.Enabled = True
TXTIDCLIENTE.SetFocus
MENU.CERRAR
QUERY = "SELECT * FROM RANGO_FACTURAS"
MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic, -1
If Not MENU.REGISTRO.EOF Then
TEMPFACTURA = MENU.REGISTRO(0)
TEMPINICIAL = MENU.REGISTRO(1)
TEMPEFINAL = MENU.REGISTRO(2)
Else
MsgBox "No han asignado rango de facturas", vbInformation, "AVISO"

```

```
Exit Sub
End If
```

```
Private Sub TXTFACTURA_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub
```

```
Private Sub TXTIDCLIENTE_GotFocus()
    TXTIDCLIENTE.SelStart = 0
    TXTIDCLIENTE.SelLength = Len(TXTIDCLIENTE)
    FACTURA.KeyPreview = True
End Sub
```

'VALIDA QUE SOLO DIGITEN NUMEROS

```
Private Sub TXTIDCLIENTE_KeyPress(KeyAscii As Integer)
    If Not (IsNumeric(Chr(KeyAscii))) And KeyAscii <> 8 And KeyAscii <> 13 Then
        KeyAscii = 0
    End If
End Sub
```

```
Private Sub TXTIDCLIENTE_LostFocus()
```

```
    If TXTIDCLIENTE <> "" Then
```

'CONSULTAR LOS DATOS DE LA TABLA CLIENTES

```
    MENU.CERRAR
```

```
    QUERY = "SELECT * FROM CLIENTES WHERE CL_IDCLIENTE = " &
        Trim(TXTIDCLIENTE)
```

```
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
```

```
    If Not MENU.REGISTRO.EOF Then
```

'ASIGNA A LA FACTURA EL NOMBRE DEL CLIENTE

```
    TXTNOMBRE = MENU.REGISTRO.Fields("CL_NOMCLIENTE")
```

```
    DETALLE.SetFocus
```

```
    DETALLE.Col = 1
```

```
Else
```

```
    RESPUESTA = MsgBox("El cliente no existe. ¿Desea ingresarlo a la base de
    datos?", vbYesNo, "CONFIRMACION")
```

```
    If RESPUESTA = 6 Then
```

```
        CLIENTE.Show
```

```
        CLIENTE.TXTIDCLIENTE = TXTIDCLIENTE
```

```
        CLIENTE.TXTIDCLIENTE.SetFocus
```

```
    Else
```

```
        TXTIDCLIENTE.SetFocus
```

```
        Exit Sub
```

```
    End If
```

```
End If
```

```
End If
```

```
FACTURA.KeyPreview = False
```

End Sub

```
Private Sub TXTSUBTOTAL_KeyDown(KeyCode As Integer, Shift As Integer)
    KeyCode = 0
End Sub
```

```
Private Sub TXTSUBTOTAL_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub
```

```
Private Sub TXTTOTAL_KeyDown(KeyCode As Integer, Shift As Integer)
    KeyCode = 0
End Sub
```

```
Private Sub TXTTOTAL_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub
```

```
Private Sub TXTIVA_KeyDown(KeyCode As Integer, Shift As Integer)
    KeyCode = 0
End Sub
```

```
Private Sub TXTIVA_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub
```

'DECLARACION DE VARIABLES PARA ESTE FORMULARIO

```
Public COMAN As New ADODB.Command
```

```
Public SW As Boolean
```

```
Public RELACION As Long
```

```
Public SWT As Boolean
```

'AL MOMENTO DE CARGAR EL FORMULARIO GENERADOR DE REPORTE

```
Private Sub Form_Load()
```

```
    Dim i As Long
```

```
    SWT = True
```

```
    RELACION = 0
```

```
    SW = True
```

```
    Dim TEMPTABLA As String
```

```
    Me.Left = 0
```

```
    Me.Top = 0
```

```
    Dim NODOS As Node
```

```
    Set NODOS = ARBOL.Nodes.Add(., "Menu", "Tablas")
```

'SE TRAEN LAS TABLAS DE LA BASE DE DATOS MEDIANTE LA SIGUIENTE CONSULTA

```
    COMAN.CommandText = "SHOW TABLES"
```

```
    MENU.CERRAR
```

```

    MENU.REGISTRO.Open    COMAN.CommandText,    MENU.CONEXION,
    adOpenDynamic, adLockBatchOptimistic
While Not MENU.REGISTRO.EOF
    TEMPTABLA = MENU.REGISTRO!tables_in_distribafagar
    If TEMPTABLA <> "bodegas_productos" Then
        If TEMPTABLA <> "proveedores_productos" Then
            If TEMPTABLA <> "relaciones" Then
'SE AGREGAN TABLAS AL ARBOL
                Set    NODOS    =    ARBOL.Nodes.Add("Menu",    twwChild,
                    MENU.REGISTRO!tables_in_distribafagar,
                    MENU.REGISTRO!tables_in_distribafagar)
                MENU.REGISTRO.MoveNext
            Else
                MENU.REGISTRO.MoveNext
            End If
        Else
            MENU.REGISTRO.MoveNext
        End If
    Else
        MENU.REGISTRO.MoveNext
    End If
Else
    MENU.REGISTRO.MoveNext
End If
Wend
MENU.CERRAR
CONT = 1
For i = 2 To ARBOL.Nodes.Count
    MENU.CERRAR
    COMAN.CommandText = "DESC " & ARBOL.Nodes(i).text
    MENU.REGISTRO.Open    COMAN.CommandText,    MENU.CONEXION,
    adOpenDynamic, adLockBatchOptimistic
    While Not MENU.REGISTRO.EOF
'PARA DIFERENCIAR LOS CAMPOS CON EL MISMO NOMBRE EN
DIFERENTES TABLAS, Y EVITAR ERROR POR LLAVES DUPLICADAS EN EL
ARBOL
        CONT = CONT + 1
'SE AGREGAN LOS CAMPOS A SUS RESPECTIVAS TABLAS EN EL ARBOL
        Set    NODOS    =    ARBOL.Nodes.Add(ARBOL.Nodes(i).text,    twwChild,
            MENU.REGISTRO!Field & CONT, MENU.REGISTRO!Field)
        MENU.REGISTRO.MoveNext
    Wend
Next i
MENU.CERRAR
ARBOL.Nodes(1).Expanded = True
For i = 0 To MATRIX.Rows = 1
    MATRIX.ColWidth(i) = 2500
Next i
MATRIX.Rows = 1
COMAN.CommandText = ""

```

```

MS_CAMPOS.Cols = 4
MS_CAMPOS.Rows = 2
MS_CAMPOS.ColWidth(0) = 2500
MS_CAMPOS.ColWidth(1) = 1000
MS_CAMPOS.ColWidth(2) = 2500
MS_CAMPOS.ColWidth(3) = 2500
MS_CAMPOS.Editable = True
MS_CAMPOS.TextMatrix(0, 0) = "ATRIBUTO"
MS_CAMPOS.TextMatrix(0, 1) = "CONDICION"
MS_CAMPOS.TextMatrix(0, 2) = "ATRIBUTO"
MS_CAMPOS.TextMatrix(0, 3) = "CONDICION"
GENERADOR.KeyPreview = True
End Sub

Private Sub ABRIR_Click()
    Dim FN, LISTAVAR As String, INFILE, i As Long
    Dim TXTLINEA As String
    On Error GoTo ABRIR
    COMAN.CommandText = ""
'SE LLAMA EL CUADRO PARA ABRIR LA CONSULTA GUARDADA *.TXT
    With DIGARCHIVO
        .DialogTitle = "SELECCIONE EL ARCHIVO A PROCESAR"
        .Filter = "*.txt"
        .ShowOpen
        FN = .FileName
    End With
'SE VERIFICA QUE EL ARCHIVO ESTE ABIERTO
    If FN <> vbNullString Then
        SWT = False
        INFILE = FreeFile()
        Open FN For Input As #INFILE
'SE RECORRE EL ARCHIVO LINEA POR LINEA Y SE ALMACENA EN UNA
VARIABLE COMAN...
        Do While Not EOF(INFILE)
            Line Input #INFILE, TXTLINEA
            COMAN.CommandText = COMAN.CommandText & TXTLINEA
            MsgBox COMAN.CommandText
        Loop
        Close #INFILE
        SWT = False
'SE EJECUTA LA CONSULTA ALMACENADA EN LA VARIABLE
        MENU.CERRAR
        MENU.REGISTRO.Open COMAN.CommandText, MENU.CONEXION,
            adOpenDynamic, adLockBatchOptimistic
        If Not MENU.REGISTRO.EOF Then
            MATRIX.Rows = 1

```

```

MATRIX.Cols = MENU.REGISTRO.Fields.Count
MATRIX.FixedCols = 0
For i = 0 To MATRIX.Cols - 1
    MATRIX.TextMatrix(0, i) = MENU.REGISTRO.Fields.Item(i).Name
    MATRIX.ColWidth(i) = 2500
Next i
'SE ESCRIBE EL RESULTADO DE LA CONSULTA DINAMICA GUARDADA EN EL FLEX
While Not MENU.REGISTRO.EOF
    LISTAVAR = ""
    For i = 0 To MENU.REGISTRO.Fields.Count - 1
        LISTAVAR = LISTAVAR & MENU.REGISTRO.Fields.Item(i)
        If i < MENU.REGISTRO.Fields.Count - 1 Then
            LISTAVAR = LISTAVAR & vbTab
        End If
    Next i
    MATRIX.AddItem LISTAVAR
    MENU.REGISTRO.MoveNext
Wend
End If
End If
ABRIR:
If Err.Number <> 0 Then
    MsgBox Err.Description & " Proceso cancelado", vbInformation, "GENERADOR"
End If
End Sub

Private Sub ARBOL_DblClick()
    EJECUTAR.Enabled = True
'SI TIENE NODOS HIJOS
    If ARBOL.Nodes(ARBOL.SelectedItem.Index).Children > 0 Then
'SE OBTIENE EL PRIMER HIJO SELECCIONADO
        Set nodo = ARBOL.Nodes(ARBOL.SelectedItem.Index).Child
        For i = 0 To ARBOL.Nodes(ARBOL.SelectedItem.Index).Children - 1
'VALIDAR DEVUELVE FALSO SI LA CADENA YA SE ENCUENTRA EN LA LISTA
            If (VALIDAR(ARBOL.Nodes(ARBOL.SelectedItem.Index).text & "." & nodo.text)) Then
                LISTA.AddItem ARBOL.Nodes(ARBOL.SelectedItem.Index).text & "." & nodo.text
'NODO.NEXT DEVUELVE EL SIGUIENTE NODO EN EL NIVEL ACTUAL (NIVEL SECUNDARIO HIJOS)
                Set nodo = nodo.Next
            End If
        Next
'VALIDAR DEVUELVE FALSO SI LA TABLA YA SE ENCUENTRA EN LA LISTA

```

```

If VALIDAR_TABLAS(ARBOL.Nodes(ARBOL.SelectedItem.Index).Index) Then
    TABLAS.AddItem ARBOL.Nodes(ARBOL.SelectedItem.Index).text
End If
Else
'VALIDAR DEVUELVE FALSO SI LA CADENA YA SE ENCUENTRA EN LA LISTA
    If (VALIDAR(ARBOL.Nodes(ARBOL.SelectedItem.Index).Parent.text & "." &
        ARBOL.Nodes(ARBOL.SelectedItem.Index).text)) Then
        LISTA.AddItem ARBOL.Nodes(ARBOL.SelectedItem.Index).Parent.text &
            "." & ARBOL.Nodes(ARBOL.SelectedItem.Index).text
    End If
'VALIDAR DEVUELVE FALSO SI LA TABLA YA SE ENCUENTRA EN LA LISTA
    If VALIDAR_TABLAS(ARBOL.Nodes(ARBOL.SelectedItem.Index).Parent.Index)
        Then
        TABLAS.AddItem ARBOL.Nodes(ARBOL.SelectedItem.Index).Parent.text
    End If
End If
End Sub

Private Sub FJECUTAR_Click()
    On Error GoTo ERRORES
    GUARDARCONS.Enabled = True
    If LISTA.ListCount = 0 Then
        MsgBox "DEBE SELECCIONAR LOS DATOS A CONSULTAR", vbInformation,
            "GENERADOR DE REPORTE"
        Exit Sub
    End If
    If TABLAS.ListCount = 0 Then
        MsgBox "DEBE SELECCIONAR LOS DATOS A CONSULTAR", vbInformation,
            "GENERADOR DE REPORTE"
        Exit Sub
    End If
    If CAMPO.Enabled = False Then
        CAMPO.Enabled = True
        CAMPO.Caption = ""
    End If
'LE ASIGNA A EL FLEX LA CANTIDAD DE CAMPOS DEL REPORTE DINAMICO
    MATRIX.Cols = LISTA.ListCount
    MATRIX.FixedCols = 0
'SE LLENA EL FLEX DONDE SE VA A MOSTRAR LOS CAMPOS DE LA CONSULTA A REALIZAR
    For i = 0 To LISTA.ListCount - 1
        MATRIX.TextMatrix(0, i) = LISTA.List(i)
        MATRIX.ColWidth(i) = 2000
    Next i

```

**'SE EMPIEZA A ARMAR LA SENTENCIA SQL PARA EL REPORTE DINAMICO
CON EL SELECT**

COMAN.CommandText = "SELECT "

'SE CONCATENAN A LA CONSULTA LOS CAMPOS

For i = 0 To LISTA.ListCount - 1

COMAN.CommandText = COMAN.CommandText & LISTA.List(i)

If i < LISTA.ListCount - 1 Then

COMAN.CommandText = COMAN.CommandText & ", "

End If

Next i

For i = 0 To TABLAS.ListCount - 1

For J = i + 1 To TABLAS.ListCount - 1

Var = TABLAS.List(i) & "_ " & TABLAS.List(J)

Next J

Next i

**'SE VERIFICA QUE LA RELACION EXISTA EN LA BASE DE DATOS CON LA
SIGUIENTE CONSULTA**

MENU.CERRAR

MENU.REGISTRO.Open "SELECT * FROM RELACIONES WHERE
RE_IDRELACION = " & Trim(Var) & "", MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

If Not MENU.REGISTRO.EOF Then

If (MENU.REGISTRO!RE_RELACIONAL <> 0) Then

If VALIDAR_TABLAS(Trim(1)) Then

SW = False

TABLAS.AddItem Trim(MENU.REGISTRO!RE_RELACIONAL)

TABLATEMP = Trim(MENU.REGISTRO!RE_RELACIONAL)

End If

End If

End If

'SE CONCATENAN A LA CONSULTA LAS TABLAS

COMAN.CommandText = COMAN.CommandText & " FROM "

For i = 0 To TABLAS.ListCount - 1

COMAN.CommandText = COMAN.CommandText & TABLAS.List(i)

If i < TABLAS.ListCount - 1 Then

COMAN.CommandText = COMAN.CommandText & ", "

End If

Next i

For i = MS_CAMPOS.Rows - 1 To 1 Step -1

If MS_CAMPOS.TextMatrix(i, 0) = "" Or MS_CAMPOS.TextMatrix(i, 1) = "" Then

MS_CAMPOS.RemoveItem (i)

End If

Next i

If MS_CAMPOS.Rows = 1 Then

CAMPO.Caption = ""

End If

COND = 0

'CONCATENAR A LA CONSULTA LA CLAUSULA WHERE

```

If CAMPO.Caption <> "" Then
    COMAN.CommandText = COMAN.CommandText & " WHERE "
    SW = False
    COND = 1
End If

```

'CONCATENAR LOS CRITERIOS DE LA CONSULTA DE LA MATRIZ, OJO SI HAY

```

For i = 1 To MS_CAMPOS.Rows - 1
    If MS_CAMPOS.TextMatrix(i, 0) <> "" Then
        COMAN.CommandText = COMAN.CommandText & " " &
            MS_CAMPOS.TextMatrix(i, 0) & " " & MS_CAMPOS.TextMatrix(i, 1) & " " &
            MS_CAMPOS.TextMatrix(i, 2) & " " & MS_CAMPOS.TextMatrix(i, 3)
    End If
Next i
CONDB = ""

```

'SE CONCATENA A LA CONSULTA EL CRUCE DE TABLAS CORRESPONDIENTE EN CASO QUE LA CONSULTA LLEVE MAS DE UNA TABLA

```

MENU.CERRAR
    MENU.REGISTRO.Open "SELECT RE_DESRELACION, RE_RELACIONAL
    FROM RELACIONES WHERE RE_IDRELACION = " & Trim(Var) & "",
    MENU.CONEXION, adOpenDynamic, adLockBatchOptimistic
SW = True
If Not MENU.REGISTRO.EOF Then
    SW = False
    CONDB = MENU.REGISTRO!RE_DESRELACION
End If
If SW Then
    If TABLAS.ListCount <> 1 Then
        MATRIX.Clear
        MsgBox "las Tablas posiblemente no estan relacionadas", vbInformation,
            "AVISO"
        Exit Sub
    End If
End If
If (CONDB <> "") Then
    If (COND = 0) Then
        COMAN.CommandText = COMAN.CommandText & " WHERE " & CONDB
    Else
        COMAN.CommandText = COMAN.CommandText & " AND " & CONDB
    End If
End If
SWT = True

```

'SE INGRESAN LOS CAMPOS DE LA CONSULTA EN EL FLEX CORRESPONDIENTE AL REPORTE DINAMICO

```

MENU.CERRAR
    MENU.REGISTRO.Open    COMAN.CommandText,    MENU.CONEXION,
        adOpenDynamic, adLockBatchOptimistic
MATRIX.Rows = 1
If Not MENU.REGISTRO.EOF Then
    MATRIX.Rows = 1
    While Not MENU.REGISTRO.EOF
        LISTAVAR = ""
        For i = 0 To MENU.REGISTRO.Fields.Count - 1
            LISTAVAR = LISTAVAR & MENU.REGISTRO.Fields.Item(i)
            If i < MENU.REGISTRO.Fields.Count - 1 Then
                LISTAVAR = LISTAVAR & vbTab
            End If
        Next i
        MATRIX.AddItem LISTAVAR
        MENU.REGISTRO.MoveNext
    Wend
End If
For i = 0 To TABLAS.ListCount - 1
    If TABLAS.List(i) = TABLATEMP Then
        TABLAS.RemoveItem (i)
    End If
Next i
ERRORES:
If Err.Number <> 0 Then
    MsgBox Err.Description & ". Proceso Cancelado", vbCritical, "REPORTES
        DINAMICOS"
End If
End Sub

Private Sub GUARDARCONS_Click()
    If SWT Then
'LLAMA EL PROCEDIMIENTO GUARDAR LA CONSULTA GUARDAR
        SWT = False
    End If
End Sub

'LIMPIA LOS PARAMETROS DE LA CONSULTA
Private Sub LIMPIAR_Click()
    GUARDARCONS.Enabled = False
    EJECUTAR.Enabled = False
    SWT = True
    RELACION = 0
    LISTA.Clear
    TABLAS.Clear
    CAMPO.Enabled = True
    CAMPO = ""

```

```

MATRIX.Clear
MATRIX.Rows = 1
COMAN.CommandText = ""
MS_CAMPOS.Rows = 1
MS_CAMPOS.Rows = 2
End Sub

```

'BORRA LOS CAMPOS SELECCIONADOS EN LA LISTA DE CAMPOS AL SELECCIONARLOS CON DOBLECLICK

```

Private Sub Lista_DblClick()
    LISTA.RemoveItem (LISTA.ListIndex)
End Sub

```

```

Private Sub MS_CAMPOS_BeforeEdit(ByVal Row As Long, ByVal Col As Long, Cancel
As Boolean)

```

'LLENA UN COMBOLIST DENTRO DEL FLEX CON LOS PARAMETROS DE LA CLAUSULA

```

    If MS_CAMPOS.Col = 1 Then
        MS_CAMPOS.ComboList = "=<><=>=<>"
    End If
    LISTAVAR = ""

```

'ALMACENA EN LISTAVAR LOS CAMPOS DE LA CONSULTA

```

    For i = 0 To LISTA.ListCount - 1
        LISTAVAR = LISTAVAR & LISTA.List(i) & "|"
    Next i

```

'LLENA UN COMBOLIST DENTRO DEL FLEX CON LOS CAMPOS QUE VENGAN EN LA VARIABLE LISTAVAR

```

    If MS_CAMPOS.Col = 0 Then
        MS_CAMPOS.ComboList = ""
        MS_CAMPOS.ComboList = LISTAVAR
    End If

```

'LLENA UN COMBOLIST CON LOS VALORES DEL CAMPO SELECCIONADO EN LA CLAUSULA WHERE

```

    If MS_CAMPOS.Col = 2 Then
        CAR = ""
        For i = 1 To Len(MS_CAMPOS.TextMatrix(MS_CAMPOS.RowSel, 0))

```

'MID ME DEVUELVE EL NOMBRE DE LA TABLA A LA QUE CORRESPONDE EL CRITERIO DE LA CONSULTA Y LO ALMACENA EN CAR PARA SU POSTERIOR USO

```

            If Mid(MS_CAMPOS.TextMatrix(MS_CAMPOS.RowSel, 0), i, 1) <> "." Then
                CAR = CAR & Mid(MS_CAMPOS.TextMatrix(MS_CAMPOS.RowSel, 0), i, 1)
            Else
                Exit For
            End If
        Next i

```

```

MENU.CERRAR
  If MS_CAMPOS.TextMatrix(MS_CAMPOS.RowSel, 0) <> "" And
  TABLAS.ListCount < 0 Then
'SE EJECUTA LA CONSULTA CON EL CRITERIO SELECCIONADO
    COMAN.CommandText = "SELECT DISTINCT " &
    MS_CAMPOS.TextMatrix(MS_CAMPOS.RowSel, 0) & " FROM " & CAR
    MENU.REGISTRO.Open COMAN.CommandText, MENU.CONEXION,
    adOpenDynamic, adLockBatchOptimistic
    LISTAVAR = ""
    While Not MENU.REGISTRO.EOF
      LISTAVAR = LISTAVAR & MENU.REGISTRO.Fields.Item(0) & "|"
      MENU.REGISTRO.MoveNext
    Wend
    MENU.CERRAR
    MS_CAMPOS.ComboList = LISTAVAR
  End If
End If
IfMS_CAMPOS.TextMatrix(1, 0) <> "" Then
  CAMPO.Caption = "."
End If
'LLENA UN COMBOLIST DENTRO DEL FLEX CON LOS CONDICIONALES DE LA CONSULTA
  If MS_CAMPOS.Col = 3 Then
    MS_CAMPOS.ComboList = "AND|OR|"
  End If
End Sub

'MOVERSE DE UN CAMPO A OTRO DENTRO DEL FLEX DEL CRITERIO DE LA CONSULTA
Private Sub MS_CAMPOS_KeyPress(KeyAscii As Integer)
  If KeyAscii = 9 Then
'PASAR DE COLUMNA EN COLUMNA
    If Not MS_CAMPOS.ColSel = 3 Then
      MS_CAMPOS.ColSel = MS_CAMPOS.ColSel + 1
    Else
'PASAR DE LINEA EN LINEA
      If MS_CAMPOS.TextMatrix(MS_CAMPOS.RowSel, 3) <> "" Then
        MS_CAMPOS.Rows = MS_CAMPOS.Rows + 1
      End If
    End If
  End If
End Sub

' SALIR DEL FORMULARIO GENERADOR DE REPORTES DINAMICOS
Private Sub SALIR_Click()
  If SWF And COMAN.CommandText <> "" Then
    RESPUESTA = MsgBox("DESEA GUARDAR LA CONSULTA?", vbYesNo,

```

```

"GENERADOR DE REPORTE DINAMICO")
If RESPUESTA = vbYes Then
    GUARDAR
ElseIf RESPUESTA = vbNo Then
    Unload Me
End If
End If
Unload Me
End Sub

```

' FUNCION QUE VALIDA QUE EL CAMPO NO SE REPITA EN LA LISTA PARA LA CONSULTA

```

Private Function VALIDAR(text As String) As Boolean
    Dim J As Long
    VALIDAR = True
    For J = 0 To LISTA.ListCount - 1
        If text = LISTA.List(J) Then
            VALIDAR = False
            Exit For
        End If
    Next J
End Function

```

'FUNCION QUE VALIDA QUE LA TABLA NO SE REPITA EN LA LISTA PARA LA CONSULTA

```

Private Function VALIDAR_TABLAS(i As Long)
    Dim J As Long
    VALIDAR_TABLAS = True
    For J = 0 To TABLAS.ListCount - 1
        If ARBOL.Nodes(i) = TABLAS.List(J) Then
            VALIDAR_TABLAS = False
            Exit For
        End If
    Next J
End Function

```

'GUARDAR LA CONSULTA EJECUTADA EN UN ARCHIVO DE TEXTO

```

Private Sub GUARDAR()
    Dim OUTFILE As Long
    Dim Archivo As String
    If LISTA.ListCount = 0 Then
        Exit Sub
    Else
        If COMAN.CommandText = "" Then
            MsgBox "NO HAY CONSULTA QUE GUARDAR", vbInformation,
                "GENERADOR DE REPORTE"
            Exit Sub
        End If
    End If

```

```

    End If
End If
OUTFILE = FreeFile()
'SE DA LA OPCION DE GUARDAR LA CONSULTA EN CUALQUIER PARTE
QUE QUIERA EL USUARIO
With DIGARCHIVO
    .Filter = "ARCHIVOS DE PRUEBA;*.txt"
    .ShowSave
    Archivo = .FileName
    If Archivo <> "" Then
        Open Archivo For Output As #OUTFILE
    Else
        Exit Sub
    End If
End With
'Open App.Path & "\ & Archivo & ".txt" For Output As #OUTFILE
Print #OUTFILE, COMAN.CommandText;
Close #OUTFILE
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO DE INVENTARIOS

```

Private Sub Form_Load()
    DATO = Date
    Me.Top = 1500
    Me.Left = 1500
    TXTFECHAINI = DATO
    TXTFECHAFIN = DATO
    MENU.CERRAR
        QUERY = "SELECT * FROM TIPO_MOVIMIENTOS"
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
            adLockBatchOptimistic
    If MENU.REGISTRO.EOF Then
        MsgBox ("La tabla Tipo de movimientos esta vacia")
    Else
        Do While MENU.REGISTRO.EOF = False
            COMBOTIPOMOVIMIENTO.AddItem MENU.REGISTRO(1)
            COMBOTIPOMOVIMIENTO.ItemData(COMBOTIPOMOVIMIENTO.ListCount
                - 1) = MENU.REGISTRO(0)
            MENU.REGISTRO.MoveNext
        Loop
        COMBOTIPOMOVIMIENTO.ListIndex = 0
    End If
End Sub

Private Sub MOVIMIENTOS_Click()
    If Format(FECHAFIN.Value, "yyyy.MM.dd") < Format(FECHAINI.Value,
        "yyyy.MM.dd") Then

```

```

        MsgBox "La fecha final no puede ser anterior a la inicial", vbInformation,
        "AVISO"
        TXTFECHAINI.SetFocus
    Exit Sub
End If
MENU.CrystalReport.ReportFileName = App.Path &
"\REPORTES\PRODUCTOSVENDIDOS.rpt"
MENU.CrystalReport.SelectionFormula = "{kardex.tm_codtipomov} = " &
Trim(COMBOTIPOMOVIMIENTO.ItemData(COMBOTIPOMOVIMIENTO.ListI
ndex)) & " AND {kardex.kd_fecha} in Date (" & Trim(Format(FECHAINI.Value,
"YYYY.MM.DD")) & ") TO Date (" & Trim(Format(FECHAFIN.Value,
"YYYY.MM.DD")) & ") "
MENU.CrystalReport.WindowState = crptMaximized
MENU.CrystalReport.Action = 1
End Sub
Private Sub EXISTENCIA_Click()
    MENU.CrystalReport.ReportFileName=App.Path&"\REPORTES\existenciaproduc
tos.rpt"
    MENU.CrystalReport.SelectionFormula = ""
    MENU.CrystalReport.WindowState = crptMaximized
    MENU.CrystalReport.Action = 1
End Sub

Private Sub STOCKMINIMO_Click()
    MENU.CrystalReport.ReportFileName = App.Path & "\REPORTES\stockminimo.rpt"
    MENU.CrystalReport.SelectionFormula = "{Productos.pr_existencia} <
{Productos.pr_stockminimo} and {Productos.pr_estado} = 1"
    MENU.CrystalReport.WindowState = crptMaximized
    MENU.CrystalReport.Action = 1
End Sub

'AL MOMENTO DE CARGAR EL FORMULARIO CREAR IVA
Private Sub Form_Load()
    IVA.KeyPreview = True
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
End Sub

Private Sub INGRESAR_Click()
    If TXTIVA = "" Then
        MsgBox "Debe digitar el codigo del IVA ", vbInformation, "Aviso"
        TXTIVA.SetFocus
    Exit Sub
    Else
        If TXTVALORIVA = "" Then
            MsgBox "Debe digitar el valor del IVA ", vbInformation, "Aviso"
            TXTIVA.SetFocus
        End If
    End If
End Sub

```

```

        Exit Sub
    End If
End If
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO IVA VALUES(" & Trim(TXTIVA)
& ", " & Trim(TXTVALORIVA) & " ")")
MsgBox "Se ha ingresado un nuevo valor del IVA en el sistema", vbInformation,
"AVISO"
TXTIVA = ""
TXTVALORIVA = ""
TXTIVA.SetFocus
INGRESAR.Enabled = False
End Sub

```

```

Private Sub LIMPIAR_Click()
    TXTIVA = ""
    TXTVALORIVA = ""
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
    TXTIVA.Enabled = True
    TXTIVA.SetFocus
End Sub

```

```

Private Sub MODIFICAR_Click()
    If TXTIVA <> "" And TXTVALORIVA <> "" Then
        MENU.CERRAR
        MENU.CONEXION.Execute ("UPDATE IVA SET IVA_VALORIVA = " &
Trim(TXTVALORIVA) & " ")
        MsgBox "Datos actualizados correctamente", vbInformation, "AVISO"
        TXTIVA = ""
        TXTVALORIVA = ""
        TXTIVA.Enabled = True
        TXTIVA.SetFocus
        MODIFICAR.Enabled = False
    Else
        If TXTIVA = "" Then
            MsgBox "Digite el codigo del iva", vbInformation, "AVISO"
            TXTIVA.SetFocus
        Else
            MsgBox "Digite el valor del iva", vbInformation, "AVISO"
            TXTVALORIVA.SetFocus
        End If
    End If
End Sub

```

```

Private Sub TXTIVA_LostFocus()

```

```

If Not IsNumeric(TXTIVA) And TXTIVA <> "" Then
    MsgBox "Codigo Invalido , este es un campo numerico", vbInformation, "AVISO"
    TXTIVA = ""
    TXTIVA.SetFocus
    INGRESAR.Enabled = False
    Exit Sub
End If
MENU.CERRAR
QUERY = "SELECT * FROM IVA WHERE IVA_CODIVA = " & Trim(TXTIVA)
If TXTIVA <> "" Then
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If Not MENU.REGISTRO.EOF Then
        TXTIVA.Enabled = False
        TXTVALORIVA = MENU.REGISTRO.Fields("IVA_VALORIVA")
        MODIFICAR.Enabled = True
        INGRESAR.Enabled = False
    Else
        INGRESAR.Enabled = True
        MODIFICAR.Enabled = False
        Exit Sub
    End If
Else
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CREAR LINEAS

```

Private Sub Form_Load()
    LINEAS.KeyPreview = True
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
End Sub

```

'LLAMA AL FORMULARIO CONSULTA DE LINEAS

```

Private Sub CONSULTAR_Click()
    CONSULTAR_LINEAS.Show
    CONSULTAR_LINEAS.ZOrder
End Sub

```

```

Private Sub INGRESAR_Click()
    If TXTLINEA = "" Then
        MsgBox "Debe digitar el codigo de la Linea ", vbInformation, "Aviso"
        TXTLINEA.SetFocus
        Exit Sub
    Else

```

'SE INGRESAN DATOS A LA TABLA LINEAS

MENU.CERRAR

MENU.CONEXION.Execute ("INSERT INTO LINEAS VALUES(" &
Trim(TXTLINEA) & ", " & Trim(TXTNOMBRE) & " ")")

MsgBox "Se ha creado la linea correctamente", vbInformation, "AVISO"

'SE INGRESAN DATOS A LA TABLA AUDITORIA

MENU.CERRAR

MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
(INSERCION,'LINEAS','" & Trim(Format(Date, "YYYY-MM-DD")) & "','" &
Trim(Format(Time, "hh:mm:ss")) & "','" & UCase(Trim(US_NOMBRE)) & "')")

TXTNOMBRE = ""

TXTLINEA = ""

INGRESAR.Enabled = False

TXTLINEA.SetFocus

End If

End Sub

'LIMPIA LOS DATOS DEL FORMULARIO

Private Sub LIMPIAR_Click()

TXTLINEA = ""

TXTNOMBRE = ""

INGRESAR.Enabled = False

MODIFICAR.Enabled = False

TXTLINEA.Enabled = True

TXTLINEA.SetFocus

End Sub

Private Sub MODIFICAR_Click()

If TXTLINEA <> "" Then

'SE ACTUALIZAN LOS DATOS DE LA TABLA LINEAS

MENU.CERRAR

MENU.CONEXION.Execute ("UPDATE LINEAS SET LI_CODLINEA = " &
Trim(TXTLINEA) & ", LI_NOMLINEA = " & Trim(TXTNOMBRE) & "
WHERE LI_CODLINEA = " & Trim(TXTLINEA))

MsgBox "Datos modificados correctamente", vbInformation, "AVISO"

'SE INGRESAN DATOS A LA TABLA AUDITORIA

MENU.CERRAR

MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
(MODIFICACION,'LINEAS','" & Trim(Format(Date, "YYYY-MM-DD")) & "','" &
& Trim(Format(Time, "hh:mm:ss")) & "','" & UCase(Trim(US_NOMBRE)) & "')")

TXTLINEA = ""

TXTNOMBRE = ""

MODIFICAR.Enabled = False

TXTLINEA.Enabled = True

TXTLINEA.SetFocus

End If

End Sub

```

Private Sub TXTLINEA_LostFocus()
    If TXTLINEA <> "" Then
        'CONSULTA LOS DATOS DE LA TABLA LINEAS
        MENU.CERRAR
        QUERY = "SELECT * FROM LINEAS WHERE LI_CODLINEA = " &
            Trim(TXTLINEA)
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
            adLockBatchOptimistic
        If Not MENU.REGISTRO.EOF Then
            'ASIGNA LOS DATOS A LOS CAMPOS CORRESPONDIENTES
            TEMPSUCURSAL = MENU.REGISTRO.Fields("LI_CODLINEA")
            TXTNOMBRE = MENU.REGISTRO.Fields("LI_NOMLINEA")
            MODIFICAR.Enabled = True
            INGRESAR.Enabled = False
            TXTLINEA.Enabled = False
        Else
            INGRESAR.Enabled = True
            MODIFICAR.Enabled = False
            TXTNOMBRE = ""
        End If
    Else
        INGRESAR.Enabled = False
        MODIFICAR.Enabled = False
        TXTNOMBRE = ""
    End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO GENERAR PEDIDO

```

Private Sub Form_Load()
    DATO = Date
    PEDIDO.KeyPreview = True
    Me.Left = 0
    Me.Top = 0
    TXTUSUARIO = US_NOMBRE
    TXTFECHA = DATO
    PEDIDO.DETALLE.Rows = 2
    PEDIDO.DETALLE.Cols = 6
    PEDIDO.DETALLE.ColWidth(0) = 500
    PEDIDO.DETALLE.ColWidth(1) = 1000
    PEDIDO.DETALLE.ColWidth(2) = 4000
    PEDIDO.DETALLE.ColWidth(3) = 800
    PEDIDO.DETALLE.ColWidth(4) = 2000
    PEDIDO.DETALLE.ColWidth(5) = 2000
    PEDIDO.DETALLE.TextMatrix(0, 0) = "Item"
    PEDIDO.DETALLE.TextMatrix(0, 1) = "Codigo"
    PEDIDO.DETALLE.TextMatrix(0, 2) = "Producto"

```

```

    PEDIDO.DETALLE.TextMatrix(0, 3) = "Cantidad"
    PEDIDO.DETALLE.TextMatrix(0, 4) = "Costo unitario"
    PEDIDO.DETALLE.TextMatrix(0, 5) = "Total"
    PEDIDO.DETALLE.TextMatrix(1, 0) = "1"
End Sub

```

'SE CARGA EL FORMULARIO DE CONSULTA DE PROVEEDORES

```

Private Sub BUSQUEDA_Click()
    FORM_ACTUAL_PROVEEDOR = "PEDIDO"
    CONSULTAR_PROVEEDOR.Show
    CONSULTAR_PROVEEDOR.ZOrder
End Sub

```

'CONSULTA LOS PEDIDOS GENERADOS

```

Private Sub CONSULTAR_Click()
    CONSULTAR_PEDIDOS.Show
    CONSULTAR_PEDIDOS.ZOrder
End Sub

```

```

Private Sub DETALLE_Click()
    PEDIDO.KeyPreview = False
End Sub

```

```

Private Sub DETALLE_KeyDown(KeyCode As Integer, Shift As Integer)

```

'SE CARGA EL FORMULARIO DE CONSULTA PRODUCTOS CON LA TECLA F1

```

    If KeyCode = 114 And DETALLE.Editable = True Then
        FORM_ACTUAL_PRODUCTO = "PEDIDO"
        CONSULTAR_PRODUCTO.Show
    End If

```

'PARA EVITAR SUPRIMIR LA INFORMACION DEL PRODUCTO EN EL GRID

```

    If KeyCode = 46 Then
        KeyCode = 113
    End If
End Sub

```

```

Private Sub DETALLE_KeyDownEdit(ByVal Row As Long, ByVal Col As Long,
    KeyCode As Integer, ByVal Shift As Integer)

```

'SE CARGA EL FORMULARIO DE CONSULTA PRODUCTOS CON LA TECLA F1

```

    If KeyCode = 114 And DETALLE.Editable = True Then
        FORM_ACTUAL_PRODUCTO = "PEDIDO"
        CONSULTAR_PRODUCTO.Show
    End If

```

'PARA EVITAR SUPRIMIR LA INFORMACION DEL PRODUCTO EN EL GRID

```

    If KeyCode = 46 Then
        KeyCode = 113
    End If

```

```

End If
End Sub

'SE CONTROLA Q SOLO SE DIGITEN NUMEROS EN LA COLUMNA
CANTIDAD DEL GRID
Private Sub DETALLE_KeyPressEdit(ByVal Row As Long, ByVal Col As Long,
KeyAscii As Integer)
    If DETALLE.Col = 3 Or DETALLE.Col = 4 Then
'SE CONTROLA LA TERCERA CONDICION VALIDA Q EL COSTO UNITARIO NO SEA VACIO
(Q EXISTA UNA FILA)
        If DETALLE.TextMatrix(DETALLE.RowSel, 4) = "" Or Not
(IsNumeric(Chr(KeyAscii))) And KeyAscii <> 13 And KeyAscii <> 8 Or KeyAscii =
Asc("0") Then
            KeyAscii = 0
        Else
            If KeyAscii = 13 And DETALLE.TextMatrix(DETALLE.RowSel,
DETALLE.ColSel) <> "" Then
                COSTO = DETALLE.TextMatrix(DETALLE.RowSel, 3) *
DETALLE.TextMatrix(DETALLE.RowSel, 4)
                DETALLE.TextMatrix(DETALLE.RowSel, 5) = COSTO
'SE CALCULA EL SUBTOTAL SUMANDO TODA LA COLUMNA "TOTAL"
DEL GRID
                TEMPSUBTOTAL = 0
                For i = 1 To DETALLE.Rows - 1
                    TEMPSUBTOTAL = 1 * TEMPSUBTOTAL + DETALLE.TextMatrix(i, 5)
                Next
                TXTSUBTOTAL = TEMPSUBTOTAL
                TXTIVA = PEDIDO.TXTSUBTOTAL * 16 / 100
                TXTTOTAL = 1 * TEMPSUBTOTAL + TXTIVA
            End If
        End If
    Else
        KeyAscii = 0
    End If
End Sub

Private Sub IMPRIMIR_Click()
    SW = 0
    For i = 1 To DETALLE.Rows - 1
        If DETALLE.TextMatrix(i, 3) = "" Then
            SW = 1
        End If
    Next
    For i = 1 To DETALLE.Rows - 1
'SE PREGUNTA SI EL PRODUCTO YA LO HA SUMINISTRADO EL
PROVEEDOR
        MENU.CERRAR
    
```

```

    QUERY = "SELECT * FROM PROVEEDORES_PRODUCTOS WHERE
    PV_CODPROVEEDOR = " & Trim(TXTPROVEEDOR) & " AND
    PR_CODPRODUCTO = " & DETALLE.TextMatrix(i, 1)
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    If MENU.REGISTRO.EOF Then
'SI NO LO HA SUMINISTRADO SE INGRESAN LOS DATOS EN
PROVEEDORES_PRODUCTOS
    MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO PROVEEDORES_PRODUCTOS
    VALUES( " & Trim(TXTPROVEEDOR) & ", " & Trim(DETALLE.TextMatrix(i,
    1)) & ")")
    End If
Next
If TXTPEDIDO <> "" And TXTPROVEEDOR <> "" And DETALLE.Rows >= 2 And
DETALLE.TextMatrix(1, 5) <> "" And SW = 0 Then
'PRIMERO HAY QUE VALIDAR LO Q SE ESTA INGRESANDO (CANTIDAD
CON SOTCK MINIMO Y MAXIMO)
    For i = 1 To DETALLE.Rows - 1
        MENU.CERRAR
        QUERY = "SELECT * FROM PRODUCTOS WHERE PR_CODPRODUCTO = "
        & Trim(DETALLE.TextMatrix(i, 1))
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
        adLockBatchOptimistic
        CANTIDAD = MENU.REGISTRO.Fields("PR_EXISTENCIA") +
        DETALLE.TextMatrix(i, 3) * 1
        TEMPSTOCKMINIMO = MENU.REGISTRO.Fields("PR_STOCKMINIMO")
        TEMPSTOCKMAXIMO = MENU.REGISTRO.Fields("PR_STOCKMAXIMO")
        TEMPNOMPRODUCTO = MENU.REGISTRO.Fields("PR_NOMPRODUCTO")
        If CANTIDAD < TEMPSTOCKMINIMO Then
            RESPUESTA = MsgBox("El stock minimo de " + TEMPNOMPRODUCTO
            + " es " + CStr(TEMPSTOCKMINIMO) + " , desea ingresar el pedido de
            todas formas ? ", vbYesNo, "CONFIRMACION")
            If RESPUESTA <> 6 Then
                Exit Sub
            End If
        Else
            If CANTIDAD > TEMPSTOCKMAXIMO Then
                RESPUESTA = MsgBox("El stock ma.ximo de " +
                TEMPNOMPRODUCTO + " es " + CStr(TEMPSTOCKMAXIMO) + " ,
                desea ingresar el pedido de todas formas ? ", vbYesNo,
                "CONFIRMACION")
                If RESPUESTA <> 6 Then
                    Exit Sub
                End If
            End If
        End If
    End For
End If
End If

```

Next

'SE ACTUALIZA LA EXISTENCIA DE LOS PRODUCTOS

For i = 1 To DETALLE.Rows - 1

MENU.CERRAR

QUERY = "SELECT * FROM PRODUCTOS WHERE PR_CODPRODUCTO = "
& Trim(DETALLE.TextMatrix(i, 1))

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

EXISTENCIAACTUAL = MENU.REGISTRO.Fields("PR_EXISTENCIA")

PRECIOCOMPRAACTUAL

MENU.REGISTRO.Fields("PR_PRECIOCOMPRA")

PRECIOVENTAACTUAL = MENU.REGISTRO.Fields("PR_PRECIOVENTA")

EXISTENCIANUEVA = EXISTENCIAACTUAL + CInt(DETALLE.TextMatrix(i,
3))

'ESTE CALCULO ES PARA SACAR EL PORCENTAJE PARA ESTE PRODUCTO

TEMPORAL = PRECIOVENTAACTUAL - PRECIOCOMPRAACTUAL

PORCENTAJE = (TEMPORAL * 100) / PRECIOCOMPRAACTUAL

TOTAL = DETALLE.TextMatrix(i, 5) * 1 **'ESTO SE HACE PARA PASAR LA
CADENA A ENTERO**

'AQUI SE APLICA EL INVENTARIO POR PROMEDIO PONDERADO

PRECIOCOMPRA*NUEVO = ((EXISTENCIAACTUAL
PRECIOCOMPRAACTUAL) + TOTAL) / EXISTENCIANUEVA

PRECIOVENTA+NUEVO = PRECIOCOMPRA+NUEVO
(PRECIOCOMPRA+NUEVO * PORCENTAJE) / 100

'SE ACTUALIZA EL PRECIO DE COMPRA Y DE VENTA DEL PRODUCTO

MENU.CERRAR

MENU.CONEXION.Execute ("UPDATE PRODUCTOS SET
PR_PRECIOCOMPRA = " & Trim(PRECIOCOMPRA+NUEVO) & "
PR_PRECIOVENTA = " & Trim(PRECIOVENTA+NUEVO) & "
PR_EXISTENCIA = " & Trim(EXISTENCIANUEVA) & " WHERE
PR_CODPRODUCTO = " & Trim(DETALLE.TextMatrix(i, 1)))

Next

TEMPESTADO = 1

'SE INGRESAN LOS DATOS EN LA TABLA PEDIDOS

MENU.CERRAR

MENU.CONEXION.Execute ("INSERT INTO PEDIDOS VALUES (" &
Trim(TXTPEDIDO) & "," & Trim(Format(TXTFECHA, "YYYY,MM,DD")) &
"," & Round(Trim(TXTSUBTOTAL)) & "," & Round(Trim(TXTTOTAL)) & ","
& Round(Trim(TXTIVA)) & "," & Trim(TEMPESTADO) & "," &
Trim(TXTPROVEEDOR) & "," & Trim(TXTUSUARIO) & ")")

'SE INGRESAN LOS DATOS EN LA TABLA AUDITORIA

MENU.CERRAR

MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
(INSECCION,'PEDIDOS'," & Trim(Format(Date, "YYYY-MM-DD")) & "," &
Trim(Format(Time, "hh:mm:ss")) & "," & UCase(Trim(US_NOMBRE)) & ")")

'SE INGRESAN LOS DATOS EN LA TABLA DETALLE_PEDIDO

```

For i = 1 To DETALLE.Rows - 1
    MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO DETALLE_PEDIDOS VALUES
    (" & Trim(TXTPEDIDO) & ", " & Trim(DETALLE.TextMatrix(i, 1)) & "," &
    Trim(DETALLE.TextMatrix(i, 4)) & ", " & Trim(DETALLE.TextMatrix(i, 3)) & ","
    & Trim(DETALLE.TextMatrix(i, 5)) & ")")
'SE INGRESAN LOS DATOS EN LA TABLA KARDEX
    MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO
    KARDEX(KD_FECHA,KD_CANTIDAD,KD_PRECIO,PR_CODPRODUCTO,T
    M_CODTIPOMOV,US_NOMBRE) VALUES (" & Trim(Format(TXTFECHA,
    "YYYY.MM.DD")) & "," & Trim(DETALLE.TextMatrix(i, 3)) & "," &
    Trim(DETALLE.TextMatrix(i, 5)) & ", " & Trim(DETALLE.TextMatrix(i, 1)) & ","
    & Trim(1) & ", " & Trim(TXTUSUARIO) & ")")
Next
MsgBox "Pedido ingresado a la base de datos"
DETALLE.Rows = 2
DETALLE.TextMatrix(1, 1) = ""
DETALLE.TextMatrix(1, 2) = ""
DETALLE.TextMatrix(1, 3) = ""
DETALLE.TextMatrix(1, 4) = ""
DETALLE.TextMatrix(1, 5) = ""
DETALLE.Cols = 6
TEMP = TXTPEDIDO
TXTPEDIDO = Val(TEMP) + 1
TXTPEDIDO = ""
TXTPROVEEDOR = ""
TXTNOMBRE = ""
TXTSUBTOTAL = 0
TXTIVA = 0
TXTTOTAL = 0
TXTPROVEEDOR.SetFocus
PEDIDO.KeyPreview = True
Else
    MsgBox "Complete los datos del pedido", vbInformation, "AVISO"
    If TXTPEDIDO = "" Then
        TXTPEDIDO.SetFocus
    Else
        If TXTPROVEEDOR = "" Then
            TXTPROVEEDOR.SetFocus
        Else
            DETALLE.SetFocus
            DETALLE.Row = 1
        End If
    End If
End If
End If
End Sub

```

'LIMPIAR LOS DATOS DEL FORMULARIO

```

Private Sub LIMPIAR_Click()
    DETALLE.Editable = True
    DETALLE.Rows = 2
    DETALLE.Cols = 6
    DETALLE.TextMatrix(1, 1) = ""
    DETALLE.TextMatrix(1, 2) = ""
    DETALLE.TextMatrix(1, 3) = ""
    DETALLE.TextMatrix(1, 4) = ""
    DETALLE.TextMatrix(1, 5) = ""
    TXTPEDIDO = ""
    TXTPROVEEDOR = ""
    TXTNOMBRE = ""
    TXTSUBTOTAL = 0
    TXTIVA = 0
    TXTTOTAL = 0
    IMPRIMIR.Enabled = True
    PEDIDO.BUSQUEDA.Enabled = True
    PEDIDO.TXTPROVEEDOR.Enabled = True
    PEDIDO.TXTPEDIDO.Enabled = True
    TXTPEDIDO.SetFocus
    PEDIDO.KeyPreview = True
End Sub

```

'VERIFICA QUE YA EXISTA UN PEDIDO CON ESTE CODIGO

```

Private Sub TXTPEDIDO_LostFocus()
    If TXTPEDIDO <> "" Then
        'CONSULTA DE PEDIDOS EN LA BASE DE DATOS
        MENU.CERRAR
        QUERY = "SELECT * FROM PEDIDOS WHERE PE_CODPEDIDO = " &
            Trim(TXTPEDIDO) & ""
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
            adLockBatchOptimistic
        If Not MENU.REGISTRO.EOF Then
            MsgBox ("Ya existe un pedido con ese codigo")
            TXTPEDIDO.SetFocus
        End If
    End If
End Sub

```

```

Private Sub TXTPROVEEDOR_LostFocus()
    If TXTPROVEEDOR <> "" Then
        'VERIFICA QUE EL USUARIO EXISTA
        MENU.CERRAR
        QUERY = "SELECT * FROM PROVEEDORES WHERE PV_CODPROVEEDOR
            = " & Trim(TXTPROVEEDOR)

```

```

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic
If Not MENU.REGISTRO.EOF Then
'COLOCA LOS DATOS DEL PROVEEDOR EN LOS CAMPOS
CORRESPONDIENTES
    TXTNOMBRE = MENU.REGISTRO.Fields("PV_NOMPROVEEDOR")
    DETALLE.SetFocus
    DETALLE.Col = 1
Else
    RESPUESTA = MsgBox("El proveedor no existe, ¿Desea ingresarlo a la base de
datos?", vbYesNo, "CONFIRMACION")
    If RESPUESTA = 6 Then
        PROVEEDOR.Show
        PROVEEDOR.TXTPROVEEDOR = TXTPROVEEDOR
        PROVEEDOR.TXTPROVEEDOR.SetFocus
    Else
        TXTPROVEEDOR.SetFocus
        Exit Sub
    End If
End If
End If
PEDIDO.KeyPreview = False
End Sub

Private Sub TXTTOTAL_KeyDown(KeyCode As Integer, Shift As Integer)
    KeyCode = 0
End Sub

Private Sub TXTTOTAL_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub

Private Sub TXTSUBTOTAL_KeyDown(KeyCode As Integer, Shift As Integer)
    KeyCode = 0
End Sub

Private Sub TXTSUBTOTAL_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub

Private Sub TXTIVA_KeyDown(KeyCode As Integer, Shift As Integer)
    KeyCode = 0
End Sub

Private Sub TXTIVA_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CREAR PRODUCTOS

Private Sub Form_Load()

Me.Left = 0

Me.Top = 0

PRODUCTO.KeyPreview = True

'ESTA CONSULTA TRAE TODAS LAS LINEAS Q TENGAN CATEGORIAS ASOCIADAS, SI LA LINEA ESTA CREADA PERO NO TIENE NINGUNA CATEGORIA ASOCIADA, ENTONCES NO SE CARGA, (PARA EVITAR ERRORES)

MENU.CERRAR

QUERY = "SELECT DISTINCT(L.LI_CODLINEA), L.LI_NOMLINEA FROM LINEAS L, CATEGORIAS C WHERE L.LI_CODLINEA = C.LI_CODLINEA ORDER BY LI_NOMLINEA"

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic, adLockBatchOptimistic

If MENU.REGISTRO.EOF Then

MsgBox ("La tabla lineas esta vacia")

Else

Do While MENU.REGISTRO.EOF = False

'SE AGREGAN LAS LINEAS AL COMBO LINEA

COMBOLINEA.AddItem MENU.REGISTRO(1)

COMBOLINEA.ItemData(COMBOLINEA.ListCount - 1) =

MENU.REGISTRO(0)

MENU.REGISTRO.MoveNext

Loop

End If

' CADA VES Q SE CARGUE EL FORM QUEDE SELECCIONADO POR DEFECTO EL PRIMER TEXTO

COMBOLINEA.ListIndex = 0

' CONSULTA LOS DATOS DE LA TABLA IVA

MENU.CERRAR

QUERY = "SELECT * FROM IVA"

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic, adLockBatchOptimistic

If MENU.REGISTRO.EOF Then

MsgBox ("La tabla IVA esta vacia")

Else

Do While MENU.REGISTRO.EOF = False

'AGREGA LOS VALORES DE IVA AL COMBO IVA

COMBOIVA.AddItem MENU.REGISTRO(1)

COMBOIVA.ItemData(COMBOIVA.ListCount - 1) = MENU.REGISTRO(0)

MENU.REGISTRO.MoveNext

Loop

End If

'CONSULTA LOS DATOS DE LAS BODEGAS QUE ESTEN EN ESTADO

ACTIVO = 1

MENU.CERRAR

QUERY = "SELECT * FROM BODEGAS WHERE BO_ESTADO = 1 "

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

If MENU.REGISTRO.EOF Then

MsgBox ("La tabla bodegas esta vacia")

Else

Do While MENU.REGISTRO.EOF = False

'AGREGA LAS BODEGAS AL COMBO BODEGAS

COMBOBODEGA.AddItem MENU.REGISTRO(1)

COMBOBODEGA.ItemData(COMBOBODEGA.ListCount - 1) =

MENU.REGISTRO(0)

MENU.REGISTRO.MoveNext

Loop

End If

'CADA VES Q SE CARGUE EL FORM QUEDE SELECCIONADO POR DEFECTO EL PRIMERO

COMBOIVA.ListIndex = 0

COMBOPORCENTAJE.ListIndex = 0

COMBOBODEGA.ListIndex = 0

ACTIVO.Value = True

INGRESAR.Enabled = False

MODIFICAR.Enabled = False

LABELEXISTENCIA.Visible = False

TXTEXISTENCIA.Visible = False

End Sub

'SE CALCULA EL PRECIO DE VENTA DEL PRODUCTO

Private Sub COMBOPORCENTAJE_Click()

If TXTCOMPRA <> "" And COMBOPORCENTAJE.text <> "" Then

TXTVENTA.text = TXTCOMPRA.text + (COMBOPORCENTAJE.text *
(TXTCOMPRA.text) / 100)

End If

End Sub

Private Sub COMBOLINEA_Click()

COMBOCATEGORIA.Clear

'SE CONSULTAN LAS CATEGORIAS DE UNA LINEA ESPECIFICA

MENU.CERRAR

QUERY = "SELECT * FROM CATEGORIAS WHERE LI_CODLINEA = " &
Trim(COMBOLINEA.ItemData(COMBOLINEA.ListIndex)) & " ORDER BY
CA_NOMCATEGORIA"

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

If Not MENU.REGISTRO.EOF Then

Do While MENU.REGISTRO.EOF = False

'CARGAN LAS CATEGORIAS AL COMBO CATEGORIA

```

COMBOCATEGORIA.AddItem MENU.REGISTRO(1)
COMBOCATEGORIA.ItemData(COMBOCATEGORIA.ListCount - 1) =
MENU.REGISTRO(0)
MENU.REGISTRO.MoveNext

```

```

Loop

```

'CADA VES Q SE CARGUE EL FORM QUEDE SELECCIONADO POR DEFECTO EL PRIMER TEXTO

```

COMBOCATEGORIA.ListIndex = 0
End If
End Sub

```

'SE CARGA EL FORMULARIO DE CONCLTA DE PRODUCTOS

```

Private Sub CONSULTAR_Click()
FORM ACTUAL_PRODUCTO = "PRODUCTO"
CONSULTAR_PRODUCTO.Show
CONSULTAR_PRODUCTO.ZOrder
End Sub

```

```

Private Sub INGRESAR_Click()

```

```

If TXTNOMBRE = "" Then
MsgBox "Digite el nombre del producto", vbInformation, "Aviso"
TXTNOMBRE.SetFocus
Exit Sub

```

```

Else

```

```

If TXTCOMPRA = "" Then
MsgBox "Digite el costo del producto", vbInformation, "Aviso"
TXTCOMPRA.SetFocus
Exit Sub

```

```

Else

```

```

If COMBOPORCENTAJE = "" Then
MsgBox "Seleccione un porcentaje para calcular el precio de venta",
vbInformation, "Aviso"
COMBOPORCENTAJE.SetFocus
Exit Sub

```

```

Else

```

```

If ACTIVO.Value = False And INACTIVO.Value = False Then
MsgBox "Asigne un estado al producto", vbInformation, "Aviso"
Exit Sub

```

```

End If

```

```

End If

```

```

End If

```

```

If ACTIVO.Value = True Then
TEMP = 1

```

```

Else

```

```

If INACTIVO.Value = True Then

```

```

        TEMP = 2
    End If
End If
'INGRESA LOS DATOS EN LA TABLA PRODUCTOS
MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO PRODUCTOS VALUES(" &
        Trim(TXTPRODUCTO) & ", " & Trim(TXTNOMBRE) & ", " &
        Trim(TXTCOMPRA) & ", " & Trim(TXTVENTA) & ", " & Trim(0) & ", " &
        Trim(TXTSTOCKMINIMO) & ", " & Trim(TXTSTOCKMAXIMO) & ", " &
        Trim(COMBOIVA.ItemData(COMBOIVA.ListIndex) & ", " & Trim(TEMP) & ",
        " & Trim(COMBOLINEA.ItemData(COMBOLINEA.ListIndex)) & ", " &
        Trim(COMBOCATEGORIA.ItemData(COMBOCATEGORIA.ListIndex)) & " ")")
    MsgBox "Producto ingresado correctamente", vbInformation, "AVISO"

'INGRESA LOS DATOS EN LA TABLA AUDITORIA
MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('INSERCIÓN','PRODUCTOS'," & Trim(Format(Date, "YYYY-MM-DD")) & ",." &
    & Trim(Format(Time, "hh:mm:ss")) & ", " & UCase(Trim(US_NOMBRE)) & " ")")
'INGRESA LOS DATOS EN LA TABLA BODEGAS PRODUCTOS
MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO BODEGAS_PRODUCTOS
VALUES (" & Trim(COMBOBODEGA.ItemData(COMBOBODEGA.ListIndex))
    & ", " & Trim(TXTPRODUCTO) & " ")")
TXTPRODUCTO = ""
TXTNOMBRE = ""
TXTDIRECCION = ""
TXTTELEFONO = ""
TXTEMAIL = ""
TXTCONTACTO = ""
TXTCOMPRA = ""
TXTVENTA = ""
TXTEXISTENCIA = ""
TXTSTOCKMINIMO = ""
TXTSTOCKMAXIMO = ""
COMBOIVA.ListIndex = 0
COMBOLINEA.ListIndex = 0
COMBOCATEGORIA.ListIndex = 0
COMBOBODEGA.ListIndex = 0
COMBOPORCENTAJE.ListIndex = 0
ACTIVO.Value = True
INGRESAR.Enabled = False
TXTPRODUCTO.SetFocus
End Sub

Private Sub MODIFICAR_Click()

```

```

If ACTIVO.Value = True Then
    TEMP = 1
Else
    If INACTIVO.Value = True Then
        TEMP = 2
    End If
End If
If TXTPRODUCTO <> "" Then
'ACTUALIZA LOS DATOS EN LA TABLA PRODUCTOS
    MENU.CERRAR
    MENU.CONEXION.Execute ("UPDATE PRODUCTOS SET
    PR_NOMPRODUCTO = " & Trim(TXTNOMBRE) & ", PR_PRECIOCOMPRA
    = " & Trim(TXTCOMPRA) & ", PR_PRECIOVENTA = " & Trim(TXTVENTA)
    & ", PR_EXISTENCIA = " & Trim(TXTEXISTENCIA) & ", IVA_CODIVA =
    " & Trim(COMBOIVA.ItemData(COMBOIVA.ListIndex)) & ",
    PR_STOCKMINIMO = " & Trim(TXTSTOCKMINIMO) & ",
    PR_STOCKMAXIMO = " & Trim(TXTSTOCKMAXIMO) & ", PR_ESTADO =
    " & Trim(TEMP) & ", LI_CODLINEA = " &
    Trim(COMBOLINEA.ItemData(COMBOLINEA.ListIndex)) & "
    CA_CODCATEGORIA = " &
    Trim(COMBOCATEGORIA.ItemData(COMBOCATEGORIA.ListIndex)) & "
    WHERE PR_CODPRODUCTO = " & Trim(TXTPRODUCTO))
    MsgBox "Datos modificados correctamente", vbInformation, "AVISO"
'INGRESA LOS DATOS EN LA TABLA AUDITORIA
    MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
    ('MODIFICACION','PRODUCTOS','" & Trim(Format(Date, "YYYY-MM-DD"))
    & "','" & Trim(Format(Time, "hh:mm:ss")) & "','" & UCase(Trim(US_NOMBRE))
    & "'")
    TXTPRODUCTO = ""
    TXTNOMBRE = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    TXTCONTACTO = ""
    TXTCOMPRA = ""
    TXTVENTA = ""
    TXTEXISTENCIA = ""
    TXTSTOCKMINIMO = ""
    TXTSTOCKMAXIMO = ""
    COMBOIVA.ListIndex = 0
    COMBOLINEA.ListIndex = 0
    COMBOCATEGORIA.ListIndex = 0
    COMBOBODEGA.ListIndex = 0
    COMBOPORCENTAJE.ListIndex = 0
    MODIFICAR.Enabled = False
    ACTIVO.Value = True

```

```

    TXTPRODUCTO.Enabled = True
    TXTPRODUCTO.SetFocus
End If
End Sub

```

'PARA CUANDO SE CAMBIE EL PRECIO DE COMPRA SE ACTUALIZE EL PRECIO DE VENTA

```

Private Sub TXTCOMPRA_LostFocus()
    If TXTCOMPRA <> "" And COMBOPORCENTAJE.text <> "" Then
        TXTVENTA.text = TXTCOMPRA.text + (COMBOPORCENTAJE.text *
            (TXTCOMPRA.text) / 100)
    End If
End Sub

```

```

Private Sub TXTPRODUCTO_LostFocus()

```

'SE CONSULTAN LOS PRODUCTOS

```

    MENU.CERRAR

```

```

    QUERY = "SELECT * FROM PRODUCTOS WHERE PR_CODPRODUCTO = "
    & Trim(TXTPRODUCTO)

```

```

    If TXTPRODUCTO <> "" Then

```

```

        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
        adLockBatchOptimistic

```

```

        If Not MENU.REGISTRO.EOF Then

```

'SE ASIGNAN LOS VALORES CORRESPONDIENTES DEL PRODUCTOS A LOS CAMPOS

```

        TXTNOMBRE = MENU.REGISTRO.Fields("PR_NOMPRODUCTO")

```

```

        TXTCOMPRA = MENU.REGISTRO.Fields("PR_PRECIOCOMPRA")

```

```

        TXTVENTA = MENU.REGISTRO.Fields("PR_PRECIOVENTA")

```

```

        TXTSTOCKMINIMO =

```

```

        MENU.REGISTRO.Fields("PR_STOCKMINIMO")

```

```

        TXTSTOCKMAXIMO =

```

```

        MENU.REGISTRO.Fields("PR_STOCKMAXIMO")

```

```

        TXTEXISTENCIA = MENU.REGISTRO.Fields("PR_EXISTENCIA")

```

```

        COD_IVA = MENU.REGISTRO.Fields("IVA_CODIVA")

```

```

        COD_LINEA = MENU.REGISTRO.Fields("LI_CODLINEA")

```

```

        COD_CATEGORIA =

```

```

        MENU.REGISTRO.Fields("CA_CODCATEGORIA")

```

```

        TEMP = MENU.REGISTRO.Fields("PR_ESTADO")

```

'ESTE CALCULO ES PARA SACAR EL PORCENTAJE PARA ESTE PRODUCTO EN EL COMBO

```

        TEMPORAL = PRODUCTO.TXTVENTA.text -
        PRODUCTO.TXTCOMPRA.text

```

```

        PORCENTAJE = (TEMPORAL * 100) / PRODUCTO.TXTCOMPRA.text

```

```

        For i = 0 To PRODUCTO.COMBOPORCENTAJE.ListCount - 1

```

'LA CONVERSION ES PARA EVITAR ESPACIOS Q SE ESTAN DANDO EN EL COMBO Y POR ESO NUNCA SE CUMPLIA EL IF

```

            If Cdbl(PRODUCTO.COMBOPORCENTAJE.List(i)) = PORCENTAJE

```

```

Then
    PRODUCTO.COMBOPORCENTAJE.ListIndex = i
End If
Next
'SE SELECCIONA LO Q VIENE DE LA TABLA TENIENDO EN CUENTA EL
CODIGO Y NO EL NOMBRE, POR SEGURIDAD
For i = 0 To PRODUCTO.COMBOIVA.ListCount - 1
    If PRODUCTO.COMBOIVA.ItemData(i) = COD_IVA Then
        PRODUCTO.COMBOIVA.ListIndex = i
    End If
Next
For i = 0 To PRODUCTO.COMBOLINEA.ListCount - 1
    If PRODUCTO.COMBOLINEA.ItemData(i) = COD_LINEA Then
        PRODUCTO.COMBOLINEA.ListIndex = i
    End If
Next
For i = 0 To PRODUCTO.COMBOCATEGORIA.ListCount - 1
    If PRODUCTO.COMBOCATEGORIA.ItemData(i) =
COD_CATEGORIA Then
        PRODUCTO.COMBOCATEGORIA.ListIndex = i
    End If
Next
For i = 0 To PRODUCTO.COMBOBODEGA.ListCount - 1
    If PRODUCTO.COMBOBODEGA.ItemData(i) = COD_BODEGA Then
        PRODUCTO.COMBOBODEGA.ListIndex = i
    End If
Next
If TEMP = 1 Then
    ACTIVO.Value = True
Else
    INACTIVO.Value = True
End If
TXTPRODUCTO.Enabled = False
TXTNOMBRE.SetFocus
INGRESAR.Enabled = False
MODIFICAR.Enabled = True
Else
    INGRESAR.Enabled = True
    MODIFICAR.Enabled = False
    TXTNOMBRE = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    TXTCONTACTO = ""
    TXTCOMPRA = ""
    TXTVENTA = ""
    TXTEXISTENCIA = ""

```

```

        End If
    Else
        INGRESAR.Enabled = False
    End If
End Sub

```

'LIMPIAR LOS DATOS DEL FORMULARIO

```

Private Sub LIMPIAR_Click()
    TXTPRODUCTO.Enabled = True
    TXTPRODUCTO = ""
    TXTNOMBRE = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    TXTCONTACTO = ""
    TXTCOMPRA = ""
    TXTVENTA = ""
    TXTEXISTENCIA = ""
    TXTSTOCKMINIMO = ""
    TXTSTOCKMAXIMO = ""
    COMBOIVA.ListIndex = 0
    COMBOLINEA.ListIndex = 0
    COMBOCATEGORIA.ListIndex = 0
    COMBOPORCENTAJE.ListIndex = 0
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
    ACTIVO.Value = True
    INACTIVO.Value = False
    LABELEXISTENCIA.Visible = False
    TXTEXISTENCIA.Visible = False
    TXTPRODUCTO.SetFocus
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CREAR PROVEEDORES

```

Private Sub Form_Load()
    Me.Left = 1800
    Me.Top = 1300
    MODIFICAR.Enabled = False
    INGRESAR.Enabled = False
    PROVEEDOR.KeyPreview = True
    ACTIVO.Enabled = True
End Sub

```

'LLAMA EL FORMULARIO CONSULTA PROVEEDOR

```

Private Sub CONSULTAR_Click()
    FORM_ACTUAL_PROVEEDOR = "PROVEEDOR"
    CONSULTAR_PROVEEDOR.Show

```

```
CONSULTAR_PROVEEDOR.ZOrder
End Sub
```

```
Private Sub INGRESAR_Click()
    If TXTPROVEEDOR = "" Then
        MsgBox "Digite el codigo del proveedor", vbInformation, "Aviso"
        TXTPROVEEDOR.SetFocus
        Exit Sub
    Else
        If TXTNOMBRE = "" Then
            MsgBox "Digite el nombre del proveedor", vbInformation, "Aviso"
            TXTNOMBRE.SetFocus
            Exit Sub
        Else
            If TXTDIRECCION = "" Then
                MsgBox "Digite la direccion del proveedor", vbInformation, "Aviso"
                TXTDIRECCION.SetFocus
                Exit Sub
            Else
                If ACTIVO.Value = False And INACTIVO.Value = False Then
                    MsgBox "Asignele un estado al proveedor", vbInformation, "Aviso"
                    Exit Sub
                End If
            End If
        End If
    End If
End If
If ACTIVO.Value = True Then
    TEMP = 1
Else
    If INACTIVO.Value = True Then
        TEMP = 2
    End If
End If
```

'INGRESA DATOS EN LA TABLA PROVEEDORES

```
MENU.CERRAR
```

```
MENU.CONEXION.Execute ("INSERT INTO PROVEEDORES VALUES(" &
    Trim(TXTPROVEEDOR) & ", " & Trim(TXTNOMBRE) & ", " &
    Trim(TXTDIRECCION) & ", " & Trim(TXTTELEFONO) & ", " &
    Trim(TXTEMAIL) & ", " & Trim(TXTCONTACTO) & ", " & Trim(TEMP) & "
)")
```

```
MsgBox "Proveedor ingresado correctamente", vbInformation, "AVISO"
```

'INGRESA DATOS EN LA TABLA AUDITORIA

```
MENU.CERRAR
```

```
MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('INsercion','PROVEEDORES'," & Trim(Format(Date, "YYYY-MM-DD")) &
", " & Trim(Format(Time, "hh:mm:ss")) & ", " & UCase(Trim(US_NOMBRE)) &
"")")
```

```

TXTPROVEEDOR = ""
TXTNOMBRE = ""
TXTDIRECCION = ""
TXTTELEFONO = ""
TXTEMAIL = ""
TXTCONTACTO = ""
ACTIVO.Enabled = True
TXTPROVEEDOR.Enabled = True
INGRESAR.Enabled = False
TXTPROVEEDOR.SetFocus
End Sub

Private Sub MODIFICAR_Click()
    If ACTIVO.Value = True Then
        TEMP = 1
    Else
        If INACTIVO.Value = True Then
            TEMP = 2
        End If
    End If
    If TXTPROVEEDOR <> "" Then
'ACTUALIZA LOS DATOS DE LA TABLA PROVEEDORES SEGUN EL
CRITERIO
        MENU.CERRAR
        MENU.CONEXION.Execute ("UPDATE PROVEEDORES SET
        PV_NOMPROVEEDOR = " & Trim(TXTNOMBRE) & ", PV_DIRECCION = "
        & Trim(TXTDIRECCION) & " , PV_TELEFONO = " & Trim(TXTTELEFONO)
        & " , PV_EMAIL = " & Trim(TXTEMAIL) & " , PV_CONTACTO = " &
        Trim(TXTCONTACTO) & " , PV_ESTADO = " & Trim(TEMP) & " WHERE
        PV_CODPROVEEDOR = " & Trim(TXTPROVEEDOR))
        MsgBox "Datos modificados correctamente", vbInformation, "AVISO"
'INGRESA DATOS EN LA TABLA AUDITORIA
        MENU.CERRAR
        MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
        ('MODIFICACION','PROVEEDORES'," & Trim(Format(Date, "YYYY-MM-
        DD")) & "," & Trim(Format(Time, "hh:mm:ss")) & "," &
        UCase(Trim(US_NOMBRE)) & ")")
        TXTPROVEEDOR = ""
        TXTNOMBRE = ""
        TXTDIRECCION = ""
        TXTTELEFONO = ""
        TXTEMAIL = ""
        TXTCONTACTO = ""
        ACTIVO.Value = True
        MODIFICAR.Enabled = False
        TXTPROVEEDOR.Enabled = True
        TXTPROVEEDOR.SetFocus
    End If
End Sub

```

End If
End Sub

Private Sub TXTPROVEEDOR_LostFocus()

If TXTPROVEEDOR <> "" Then

'CONSULTA DEL PROVEEDOR

MENU.CERRAR

QUERY = "SELECT * FROM PROVEEDORES WHERE PV_CODPROVEEDOR
= " & Trim(TXTPROVEEDOR)

MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
adLockBatchOptimistic

If Not MENU.REGISTRO.EOF Then

TEMP = MENU.REGISTRO.Fields("PV_ESTADO")

TXTNOMBRE = MENU.REGISTRO.Fields("PV_NOMPROVEEDOR")

TXTDIRECCION = MENU.REGISTRO.Fields("PV_DIRECCION")

TXTTELEFONO = MENU.REGISTRO.Fields("PV_TELEFONO")

TXTEMAIL = MENU.REGISTRO.Fields("PV_EMAIL")

TXTNOMBRE = MENU.REGISTRO.Fields("PV_NOMPROVEEDOR")

TXTCONTACTO = MENU.REGISTRO.Fields("PV_CONTACTO")

If TEMP = 1 Then

ACTIVO.Value = True

Else

INACTIVO.Value = True

End If

TXTPROVEEDOR.Enabled = False

TXTNOMBRE.SetFocus

INGRESAR.Enabled = False

MODIFICAR.Enabled = True

Else

INGRESAR.Enabled = True

MODIFICAR.Enabled = False

TXTNOMBRE = ""

TXTDIRECCION = ""

TXTTELEFONO = ""

TXTEMAIL = ""

TXTCONTACTO = ""

End If

Else

TXTPROVEEDOR = ""

TXTNOMBRE = ""

TXTDIRECCION = ""

TXTTELEFONO = ""

TXTEMAIL = ""

TXTCONTACTO = ""

INGRESAR.Enabled = False

MODIFICAR.Enabled = False

ACTIVO●.Value = True

```
End If
End Sub
```

'LIMPIAR LOS DATOS DEL FORMULARIO

```
Private Sub LIMPIAR_Click()
    TXTPROVEEDOR = ""
    TXTNOMBRE = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    TXTCONTACTO = ""
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
    ACTIVO.Value = True
    TXTPROVEEDOR.Enabled = True
    TXTPROVEEDOR.SetFocus
End Sub
```

'AL MOMENTO DE CARGAR EL FORMULARIO CREAR TIPO DE USUARIOS

```
Private Sub Form_Load()
    TIPO_USUARIOS.KeyPreview = True
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
End Sub
```

```
Private Sub CONSULTAR_Click()
    CONSULTAR_USUARIOS.Show
    CONSULTAR_USUARIOS.ZOrder
End Sub
```

```
Private Sub INGRESAR_Click()
    If TXTTIPO_USUARIO <> "" And TXTDESCRIPCION <> "" Then
        MENU.CERRAR
        MENU.CONEXION.Execute ("INSERT INTO TIPO_USUARIOS VALUES(" &
            Trim(TXTTIPO_USUARIO) & ", " & Trim(TXTDESCRIPCION) & " ")")
        MsgBox "Dato ingresado correctamente", vbInformation, "AVISO"
        TXTTIPO_USUARIO = ""
        TXTDESCRIPCION = ""
        TXTTIPO_USUARIO.SetFocus
        INGRESAR.Enabled = False
    Else
        If TXTTIPO_USUARIO = "" Then
            MsgBox "Digite el codigo", vbInformation, "AVISO"
            TXTTIPO_USUARIO.SetFocus
        Else
            MsgBox "Digite el codigo", vbInformation, "AVISO"
            TXTDESCRIPCION.SetFocus
        End If
    End If
End Sub
```

```

    End If
  End If
End Sub

```

```

Private Sub LIMPIAR_Click()
  TXTTIPO_USUARIO.Enabled = True
  TXTTIPO_USUARIO = ""
  TXTDESCRIPCION = ""
  TXTTIPO_USUARIO.SetFocus
End Sub

```

```

Private Sub MODIFICAR_Click()
  If TXTTIPO_USUARIO <> "" And TXTDESCRIPCION <> "" Then
    MENU.CERRAR
    MENU.CONEXION.Execute ("UPDATE TIPO_USUARIOS SET
      TU_DESCRIPCION = " & Trim(TXTDESCRIPCION) & " ")
    MsgBox "Datos actualizados correctamente", vbInformation, "AVISO"
    TXTTIPO_USUARIO = ""
    TXTDESCRIPCION = ""
    TXTTIPO_USUARIO.Enabled = True
    TXTTIPO_USUARIO.SetFocus
    MODIFICAR.Enabled = False
  Else
    If TXTTIPO_USUARIO = "" Then
      MsgBox "Digite el codigo del tipo de usuario", vbInformation, "AVISO"
      TXTTIPO_USUARIO.SetFocus
    Else
      MsgBox "Digite la descripcion del tipo de usuario", vbInformation, "AVISO"
      TXTDESCRIPCION.SetFocus
    End If
  End If
End Sub

```

```

Private Sub TXTTIPO_USUARIO_LostFocus()
  If Not IsNumeric(TXTTIPO_USUARIO) And TXTTIPO_USUARIO <> "" Then
    MsgBox "Codigo Invalido , este es un campo numerico", vbInformation, "AVISO"
    TXTTIPO_USUARIO = ""
    TXTTIPO_USUARIO.SetFocus
    INGRESAR.Enabled = False
    Exit Sub
  End If
  MENU.CERRAR
  QUERY = "SELECT * FROM TIPO_USUARIOS WHERE TU_CODIGO = " &
    Trim(TXTTIPO_USUARIO)
  If TXTTIPO_USUARIO <> "" Then
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
      adLockBatchOptimistic
  End If

```

```

    If Not MENU.REGISTRO.EOF Then
        TXTTIPO_USUARIO.Enabled = False
        TXTDESCRIPCION = MENU.REGISTRO.Fields("TU_DESCRIPCION")
        MODIFICAR.Enabled = True
        INGRESAR.Enabled = False
    Else
        INGRESAR.Enabled = True
        MODIFICAR.Enabled = False
        Exit Sub
    End If
Else
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
End If
End Sub

```

'AL MOMENTO DE CARGAR EL FORMULARIO CREAR USUARIOS

```

Private Sub Form_Load()
    Me.Left = 0
    Me.Top = 0
    DATO = Date
    LABELFECHAREG.Caption = DATO 'pa la fecha
    USUARIOS.KeyPreview = True
    If TXTCEDULA = "" Then
        INGRESAR.Enabled = False
        MODIFICAR.Enabled = False
    Else
        INGRESAR.Enabled = True
    End If
    MENU.CERRAR
    QUERY = "SELECT * FROM TIPO_USUARIOS ORDER BY TU_DESCRIPCION"
    MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
    adLockBatchOptimistic
    MENU.REGISTRO.MoveFirst
    If MENU.REGISTRO.EOF Then
        MsgBox ("La tabla no tiene datos")
    Else
        Do While MENU.REGISTRO.EOF = False
            COMBOTIPO_USUARIO.AddItem MENU.REGISTRO(1)
            COMBOTIPO_USUARIO.ItemData(COMBOTIPO_USUARIO.ListCount - 1) =
            MENU.REGISTRO(0)
            MENU.REGISTRO.MoveNext
        Loop
    End If
    COMBOTIPO_USUARIO.ListIndex = 0
End Sub

Private Sub CONSULTAR_Click()

```

```

CONSULTAR_USUARIOS.Show
CONSULTAR_USUARIOS.ZOrder
End Sub

```

```

Private Sub INGRESAR_Click()

```

```

'VALIDA LOS CAMPOS VACIOS PARA LA TABLA EMPLEADOS

```

```

If TXTCEDULA = "" Then

```

```

    MsgBox "Digite la identificacion del usuario", vbInformation, "Aviso"

```

```

    TXTCEDULA.SetFocus

```

```

    Exit Sub

```

```

Else

```

```

    If TXTNOMBRES = "" Then

```

```

        MsgBox "Digite el(los) nombre(s) del empleado", vbInformation, "Aviso"

```

```

        TXTNOMBRES.SetFocus

```

```

        Exit Sub

```

```

    Else

```

```

        If TXTAPELLIDOS = "" Then

```

```

            MsgBox "Digite los apellido del usuario", vbInformation, "Aviso"

```

```

            TXTAPELLIDOS.SetFocus

```

```

            Exit Sub

```

```

        Else

```

```

            If TXTDIRECCION = "" Then

```

```

                MsgBox "Digite la direccion del usuario", vbInformation, "Aviso"

```

```

                TXTDIRECCION.SetFocus

```

```

                Exit Sub

```

```

            Else

```

```

                If ACTIVO.Value = False And INACTIVO.Value = False Then

```

```

                    MsgBox "Asigne un estado al usuario", vbInformation, "Aviso"

```

```

                    Exit Sub

```

```

                End If

```

```

            End If

```

```

        End If

```

```

    End If

```

```

End If

```

```

If ACTIVO.Value = True Then

```

```

    TEMP = 1

```

```

Else

```

```

    If INACTIVO.Value = True Then

```

```

        TEMP = 2

```

```

    End If

```

```

End If

```

```

'VALIDA LOS CAMPOS VACIOS PARA LA TABLA DE USUARIOS

```

```

If TXTUSUARIO = "" Then

```

```

    MsgBox "Digite el nombre de usuario", vbInformation, "Aviso"

```

```

    TXTUSUARIO.SetFocus

```

```

    Exit Sub

```

```

Else

```

```

If TXTCONTRASENA = "" Then
    MsgBox "Dígite la contraseña", vbInformation, "Aviso"
    TXTCONTRASENA.SetFocus
    Exit Sub
Else
    IF TXTCONTRASENA <> TXTCONFIRMACION Then
        MsgBox "La confirmacion no coincide con la contraseña disgitada",
            vbInformation, "Aviso"
        TXTCONFIRMACION.SetFocus
        Exit Sub
    End If
End If
End If
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO EMPLEADOS VALUES(" &
    Trim(TXTCEDULA) & ", " & Trim(TXTNOMBRES) & ", " &
    Trim(TXTAPELLIDOS) & ", " & Trim(TXTDIRECCION) & ", " &
    Trim(TXTTELEFONO) & ", " & Trim(TXTEMAIL) & ", " &
    Trim(LABELFECHAREG) & " )")
//CODIFICA LA CONTRASEÑA
TEXTOCODIFICADO = EncryptString(TXTCONTRASENA, "UJUJU")
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO USUARIOS VALUES(" &
    Trim(TXTUSUARIO) & ", " & Trim(TextoCodificado) & ", " &
    Trim(COMBOTIPO_USUARIO.ItemData(COMBOTIPO_USUARIO.ListIndex))
    & ", " & Trim(TXTCEDULA) & ", " & Trim(TEMP) & ")")
'SE INGRESA EN AUDITORIA
MENU.CERRAR
MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('INSERCIÓN',USUARIOS'," & Trim(Format(Date, "YYYY-MM-DD")) & ", " &
    Trim(Format(Time, "hh:mm:ss")) & ", " & UCase(Trim(US_NOMBRE)) & ")")
MsgBox "Usuario creado correctamente", vbInformation, "AVISO"
TXTCEDULA = ""
TXTNOMBRES = ""
TXTAPELLIDOS = ""
TXTDIRECCION = ""
TXTTELEFONO = ""
TXTEMAIL = ""
TXTUSUARIO = ""
TXTCONTRASENA = ""
TXTCONFIRMACION = ""
MODIFICAR.Enabled = False
INGRESAR.Enabled = False
ACTIVO.Enabled = True
TXTCEDULA.SetFocus
End Sub

```

```

Private Sub MODIFICAR_Click()
    If ACTIVO.Value = True Then
        TEMP = 1
    Else
        If INACTIVO.Value = True Then
            TEMP = 2
        End If
    End If
    If TXTCEDULA <> "" Then
        MENU.CERRAR
        MENU.CONEXION.Execute ("UPDATE EMPLEADOS SET EM_NOMBRE = " &
            & Trim(TXTNOMBRES) & ", EM_APELLIDO = " & Trim(TXTAPELLIDOS) &
            " , EM_DIRECCION = " & Trim(TXTDIRECCION) & " , EM_TELEFONO = " &
            & Trim(TXTTELEFONO) & " , EM_EMAIL = " & Trim(TXTEMAIL) & " ,
            EM_FECHAREG = " & Trim(LABELFECHAREG) & " WHERE EM_CEDULA
            = " & Trim(TXTCEDULA))
    '//CODIFICA
    TEXTOCODIFICADO = EncryptString(TXTCONTRASENA, "UJUJU")
    MENU.CERRAR
    MENU.CONEXION.Execute ("UPDATE USUARIOS SET US_CLAVE = " &
        Trim(TextoCodificado) & " , TU_CODIGO = " &
        Trim(COMBOTIPO_USUARIO.ItemData(COMBOTIPO_USUARIO.ListIndex))
        & " , EM_CEDULA = " & Trim(TXTCEDULA) & " , US_ESTADO = " &
        Trim(TEMP) & " WHERE US_NOMBRE = " & Trim(TXTUSUARIO) & " ")
    MsgBox "Datos modificados correctamente", vbInformation, "AVISO"
    'SE INGRESA EN AUDITORIA
    MENU.CERRAR
    MENU.CONEXION.Execute ("INSERT INTO AUDITORIA VALUES
('MODIFICACION','USUARIOS'," & Trim(Format(Date, "YYYY-MM-DD")) &
    "," & Trim(Format(Time, "hh:mm:ss")) & ",'" & UCase(Trim(US_NOMBRE)) &
    "')")
    TXTCEDULA.Enabled = True
    TXTCEDULA = ""
    TXTNOMBRES = ""
    TXTAPELLIDOS = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    ACTIVO.Enabled = True
    MODIFICAR.Enabled = False
    TXTCEDULA.SetFocus
    TXTUSUARIO = ""
    TXTCONTRASENA = ""
    TXTCONFIRMACION = ""
    COMBOTIPO_USUARIO.ListIndex = 0
    MODIFICAR.Enabled = False
    INGRESAR.Enabled = False

```

```

        ACTIVO.Enabled = True
        TXTCEDULA.SetFocus
    End If
End Sub

Private Sub TXTCONFIRMACION_LostFocus()
    If TXTCONTRASENA <> TXTCONFIRMACION Then
        MsgBox "La confirmacion no coincide con la contraseña digitada", vbInformation,
            "AVISO"
        TXTCONFIRMACION.SetFocus
    Exit Sub
    End If
End Sub

Private Sub TXTCEDULA_LostFocus()
    If TXTCEDULA <> "" Then
        MENU.CERRAR
        QUERY = "SELECT * FROM EMPLEADOS WHERE EM_CEDULA = " &
            Trim(TXTCEDULA)
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
            adLockBatchOptimistic
        If Not MENU.REGISTRO.EOF Then
            TXTCEDULA.Enabled = False
            TXTNOMBRES = MENU.REGISTRO.Fields("EM_NOMBRE")
            TXTAPELLIDOS = MENU.REGISTRO.Fields("EM_APELLIDO")
            TXTDIRECCION = MENU.REGISTRO.Fields("EM_DIRECCION")
            TXTTELEFONO = MENU.REGISTRO.Fields("EM_TELEFONO")
            TXTEMAIL = MENU.REGISTRO.Fields("EM_EMAIL")
            MENU.CERRAR
            QUERY = "SELECT * FROM USUARIOS U, TIPO_USUARIOS T WHERE
            U.EM_CEDULA = " & Trim(TXTCEDULA) & " AND U.TU_CODIGO =
            T.TU_CODIGO "
            MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,
            adLockBatchOptimistic
            If Not MENU.REGISTRO.EOF Then
                TEMP = MENU.REGISTRO.Fields("US_ESTADO")
                TXTUSUARIO = MENU.REGISTRO.Fields("US_NOMBRE")
                TXTCONTRASENA = MENU.REGISTRO.Fields("US_CLAVE")
                TEMPTIPOUSU = MENU.REGISTRO.Fields("TU_CODIGO")
                TXTCONFIRMACION = TXTCONTRASENA
                For i = 0 To COMBOTIPO_USUARIO.ListCount - 1
                    If COMBOTIPO_USUARIO.ItemData(i) = TEMPTIPOUSU Then
                        COMBOTIPO_USUARIO.ListIndex = i
                    End If
                Next
            Else
                MsgBox "El usuario no tiene un estado asignado", vbInformation, "AVISO"
            End If
        End If
    End Sub

```

```

        Exit Sub
    End If
    If TEMP = 1 Then
        ACTIVO.Value = True
    Else
        INACTIVO.Value = True
    End If
    TXTNOMBRES.SetFocus
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = True
Else
    INGRESAR.Enabled = True
    MODIFICAR.Enabled = False
End If
Else
    INGRESAR.Enabled = False
    MODIFICAR.Enabled = False
End If
End Sub

```

```

Private Sub LIMPIAR_Click()
    TXTCEDULA = ""
    TXTNOMBRES = ""
    TXTAPELLIDOS = ""
    TXTDIRECCION = ""
    TXTTELEFONO = ""
    TXTEMAIL = ""
    TXTUSUARIO = ""
    TXTCONTRASENA = ""
    TXTCONFIRMACION = ""
    MODIFICAR.Enabled = False
    INGRESAR.Enabled = False
    ACTIVO.Value = True
    TXTCEDULA.Enabled = True
    TXTCEDULA.SetFocus
    COMBOTIPO_USUARIO.ListIndex = 0
End Sub

```

```

Private Sub TXTTELEFONO_LostFocus()
    If Not IsNumeric(TXTTELEFONO) And TXTTELEFONO <> "" Then
        MsgBox "Codigo Invalido , este es un campo numerico", vbInformation, "AVISO"
        TXTTELEFONO = ""
        TXTTELEFONO.SetFocus
    Exit Sub
    End If
End Sub

```

```
Private Sub TXTUSUARIO_LostFocus()  
    MENU.CERRAR  
    QUERY = "SELECT US_NOMBRE FROM USUARIOS WHERE US_NOMBRE  
    = "" & Trim(TXTUSUARIO) & ""  
    If TXTUSUARIO <> "" Then  
        MENU.REGISTRO.Open QUERY, MENU.CONEXION, adOpenDynamic,  
        adLockBatchOptimistic  
        If Not MENU.REGISTRO.EOF Then  
            MsgBox "El nombre de usuario ya existe, digite otro nombre de usuario",  
            vbInformation, "AVISO"  
            TXTUSUARIO.SetFocus  
        End If  
    End If  
End Sub
```

LOS SIGUIENTES PROCEDIMIENTOS SE UTILIZAN EN LA MAYOR PARTE DE LOS FORMULARIOS POR LO QUE SE PODRIAN LLAMAR ESTANDARES SOLO CAMBIARIA EL NOMBRE DEL OBJETO PARA EL QUE SEA EL CASO.

'LE ASIGNA LA FUNCION DE TABULACION A LA TECLA ENTER

```
Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        SendKeys Chr(9)
    End If
End Sub
```

'PERMITE SOMBREAR LO QUE ESTE DIGITADO EN LAS CAJAS DE TEXTO AL MOMENTO EN QUE SE REGRESA A ELLAS

```
Private Sub TXTUSUARIO_GotFocus()
    Me.TXTUSUARIO.SelStart = 0
    Me.TXTUSUARIO.SelLength = Len(TXTUSUARIO)
End Sub
```

'CAMBIA LAS LETRAS DIGITADAS EN LA CAJA DE TEXTO A MAYUSCULAS

```
Private Sub TXTUSUARIO_KeyPress(KeyAscii As Integer)
    Char = Chr(KeyAscii)
    KeyAscii = Asc(UCase(Char))
End Sub
```

'SALIR DEL FORMULARIO INICIO DE SESION

```
Private Sub SALIR_Click()
    Unload Me
End Sub
```

'VALIDA QUE SOLO DIGITEN NUMEROS

```
Private Sub TXTBODEGA_KeyPress(KeyAscii As Integer)
    If Not (IsNumeric(Chr(KeyAscii))) And KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End Sub
```

MODULOS DE INVENTARIO Y FACTURACION EN DISTRIFAGAR LTDA.

MANUAL DE USUARIO

MINFA

ALEXANDER JOSE FARIAS PEREZ

JAVIER SANCHEZ MUÑOZ

CARLOS HUMBERTO GAMBOA HERNANDEZ

GEOVANI FABIAN VAZQUEZ RADA

UNIVERSIDAD SIMON BOLIVAR

FACULTAD DE INGENIERIA DE SISTEMAS

AREA DE FORMATIVA IV

UNDECIMO SEMESTRE A-NOCTURNO

BARRANQUILLA

2007

INTRODUCCION

El siguiente manual de usuario, fue creado con el objetivo de soportar de manera escrita los pasos para el aprendizaje que el administrador debe tener acerca del manejo de la base de datos de la aplicación.

Esta recopilación de datos se ha realizado de una manera sencilla y fácil para que cualquier usuario, no solamente el administrador, pueda entender y aplicar los conceptos expuestos.

La metodología consiste en la explicación de las interfaces del sistema con su respectiva imagen y las respectivas funcionalidades concernientes al manejo de sus opciones.

OBJETIVOS

OBJETIVO GENERAL

Facilitar la comprensión del usuario sobre todas las funciones y operaciones que se pueden llevar a cabo con el aplicativo, a través de una breve descripción de las mismas.

OBJETIVOS ESPECÍFICOS

- Facilitar la comprensión del usuario acerca de la aplicación.
 - Brindar al usuario un soporte escrito de las diferentes funciones.
 - Apoyar el proceso de capacitación.
-

TIPOS DE USUARIO.

En el aplicativo del Sistema de información Modulo de Inventario y facturación (MINFA) se cuenta con tres tipos de usuario que se describirán a continuación:

- **Administrador:**

Es el encargado de la parte contable de la empresa (Generalmente) que requiere del libre acceso a los datos, con el propósito de tomar decisiones relacionadas con los reportes y con la información suministrada por la aplicación.

El Administrador tendrá opción de:

- Consultar información relacionada con el inventario, entradas y salidas de los productos como facturas de compra y pedidos, así como la locación de productos en bodega.
- Ingresar, modificar y digitar datos en el aplicativo más no podrá eliminar un registro de la aplicación.
- Generar reportes dinámicos cuando así lo requiera.
- Puede ingresar, modificar y administrar usuarios de la aplicación
- Tiene acceso absoluto sobre todos los aplicativos de MINFA.
- Cambiar de contraseña cada vez que así lo requiera

- **Operador**

Es el encargado de alimentar las entradas y salidas de productos.

El digitador tendrá la opción de:

- Generar Facturas de venta de productos y pedidos
 - Consultar información de productos mas no modificarlos.
-

- Solo acceso a Facturación, Pedidos y Contraseñas
- Cambiar de contraseña cada vez que así lo requiera

PERFILES DE USUARIO

INICIO DE SESIÓN

Al ejecutar el modulo de inventario y facturación (MINFA) aparecerá el formulario de inicio de sesión, como se ve en la grafica 3.

Grafica 3.



Grafica 3. Inicio de sesión MINFA.

Luego de haber digitado el usuario y su respectiva contraseña de acuerdo al perfil del usuario antes mencionado aparecerá el inicio de la aplicación. Recordemos que para nuestros perfiles de usuario existen diferentes niveles de acceso al aplicativo y por ende a los datos que esta manipula, a continuación se muestra el inicio de MINFA para cada uno de los perfiles.

Figura 4.

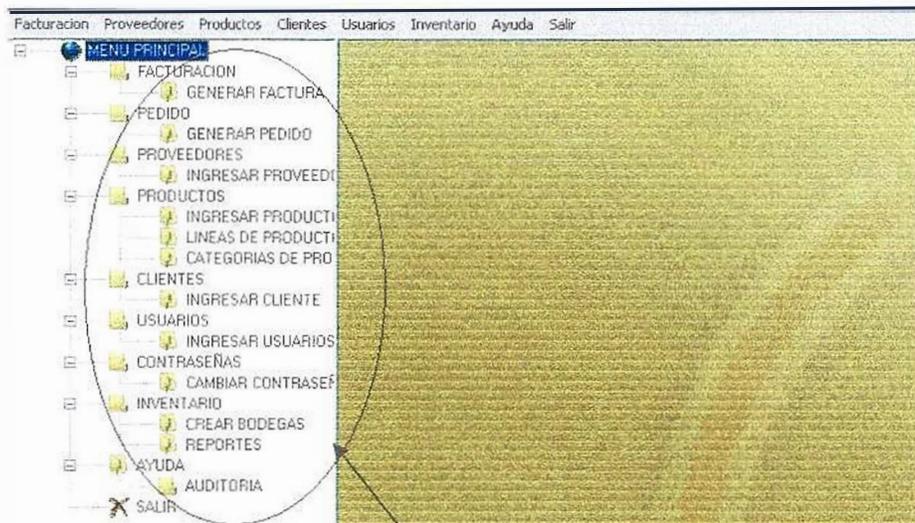


Figura 4. Inicio del aplicativo MINFA

PERFIL DE USUARIO ADMINISTRADOR

Para un usuario con perfil Administrador, se desplegara todo el paquete aplicativo y sus respectivas opciones, ya que este perfil tiene acceso absoluto a todos los datos que la aplicación maneja. En la parte posterior de la aplicación aparecerá el nombre de usuario que actualmente esta utilizando la aplicación.

Grafica 4.1

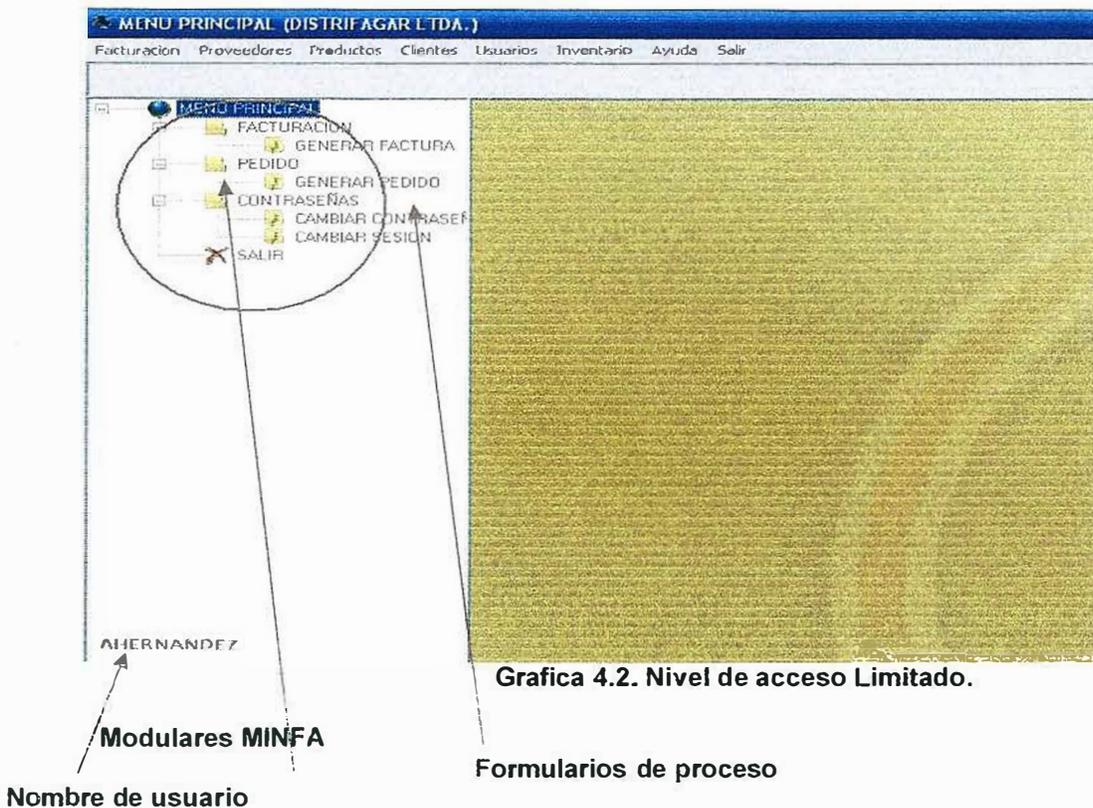


Grafica 4.1. Nivel de acceso absoluto Para perfil Administrador

PERFIL DE USUARIO DIGITADOR

Un usuario con perfil digitador, tendrá un acceso limitado a la aplicación, es decir, no se desplegarán en las opciones todas las funciones con las que trabaja la aplicación como lo muestra la Grafica 4.2.

Grafica 4.2



La aplicación cuenta con una barra de herramientas como apoyo a la localización de algunas funciones como facturación, proveedores y productos entre otras, las cuales cuentan con el método abreviado de acceso que se activa con la tecla **ALT**



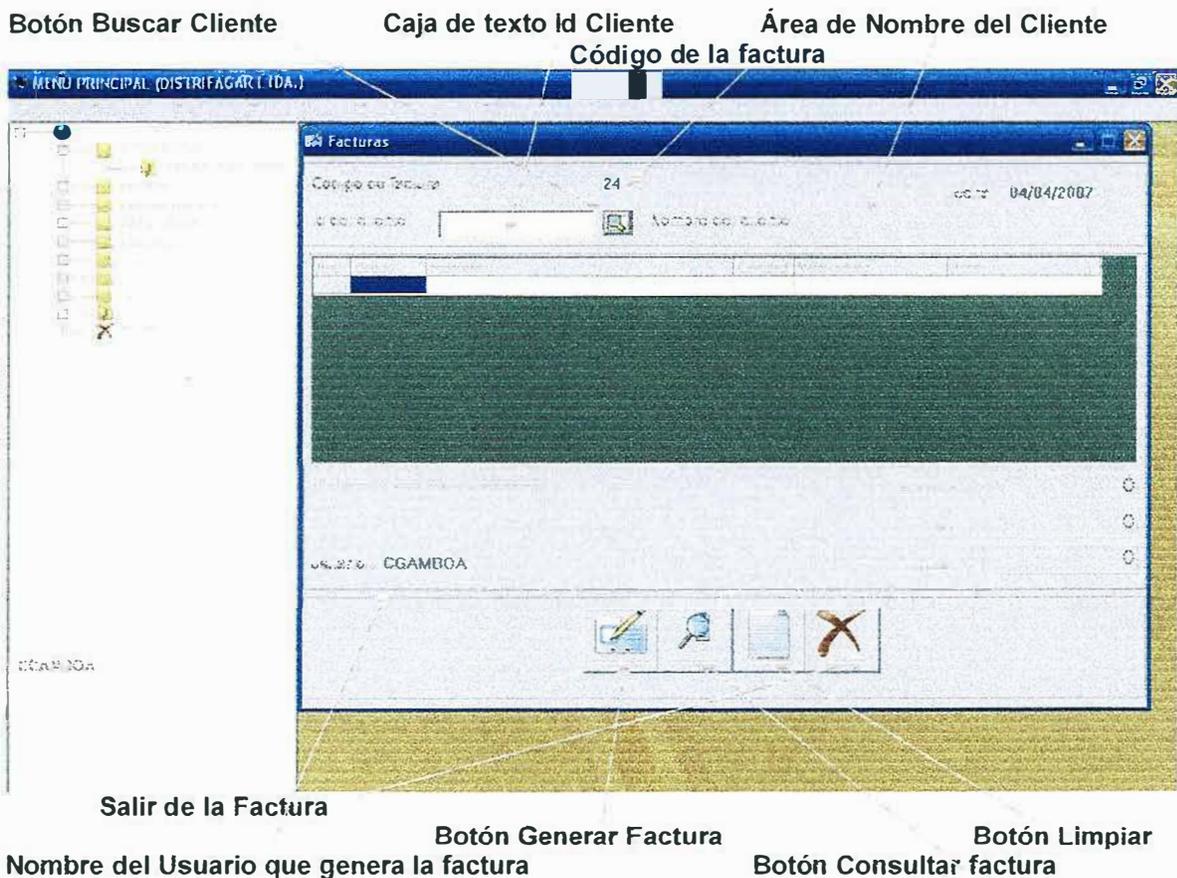
PROCESOS Y FORMULARIOS

Los procesos y formularios del aplicativo MINFA, son muy amistosos y fáciles de administrar, para ejecutar un proceso (En este caso abrir un formulario), se hace doble clic sobre la carpeta de procedimientos, y luego se hace clic sobre el campo en el que vamos a trabajar

FACTURACION

La facturación es el proceso en el que se registra la venta de uno o varios productos que ese encuentran en estado activo dentro de la aplicación, este proceso repercute sobre inventario de donde se descarga el total de cada uno de los productos de la cantidad.

A continuación, una pequeña explicación acerca de los del formulario Facturación.

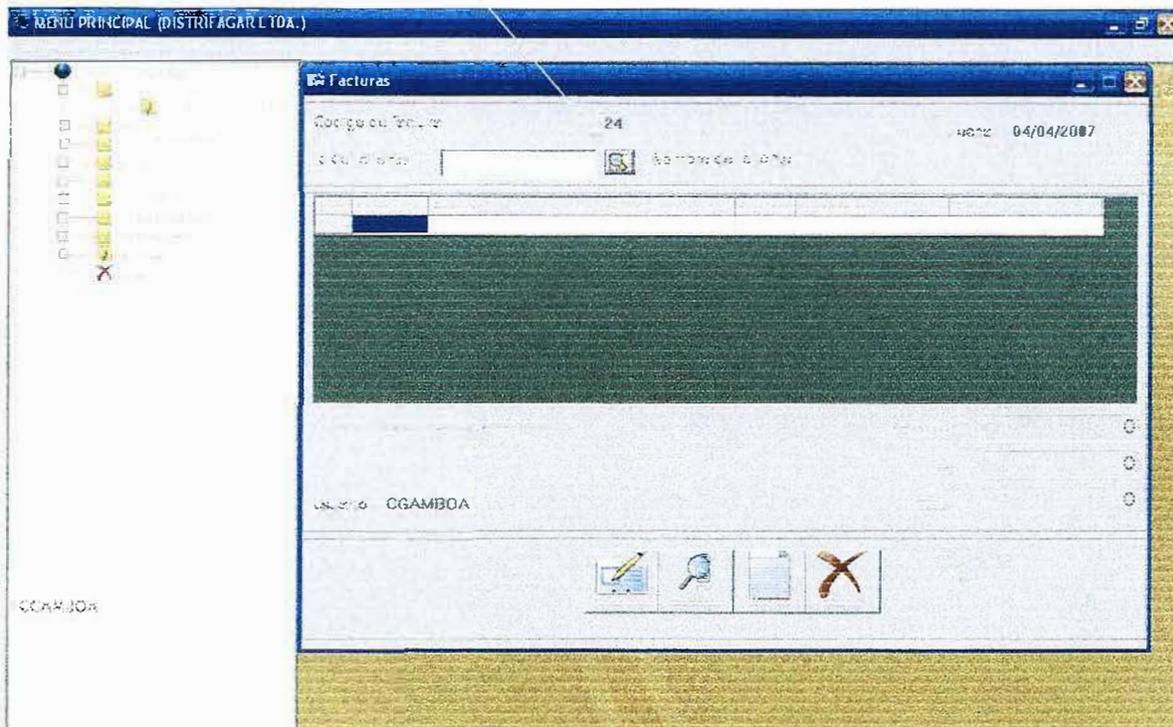


MODULAR GENERAR FACTURA

Para generar una factura, se selecciona primero el Id del cliente al que se le venderán los productos, lo podemos seleccionar de una lista haciendo clic en el botón buscar encuentra en la parte derecha de la caja de texto en donde va el Id del cliente.

Grafica 5.

Botón buscar Cliente



Grafica 5. Formulario generar factura

Al hacer clic sobre el botón buscar se despliega una lista en donde podemos ver los clientes que se encuentran registrados, hacemos doble clic sobre el cliente que vamos a escoger y el código de este, se carga en la caja de texto donde va el Id del cliente, el nombre se carga al lado.

Grafica 5.1

Código Cliente

Nombre del Cliente

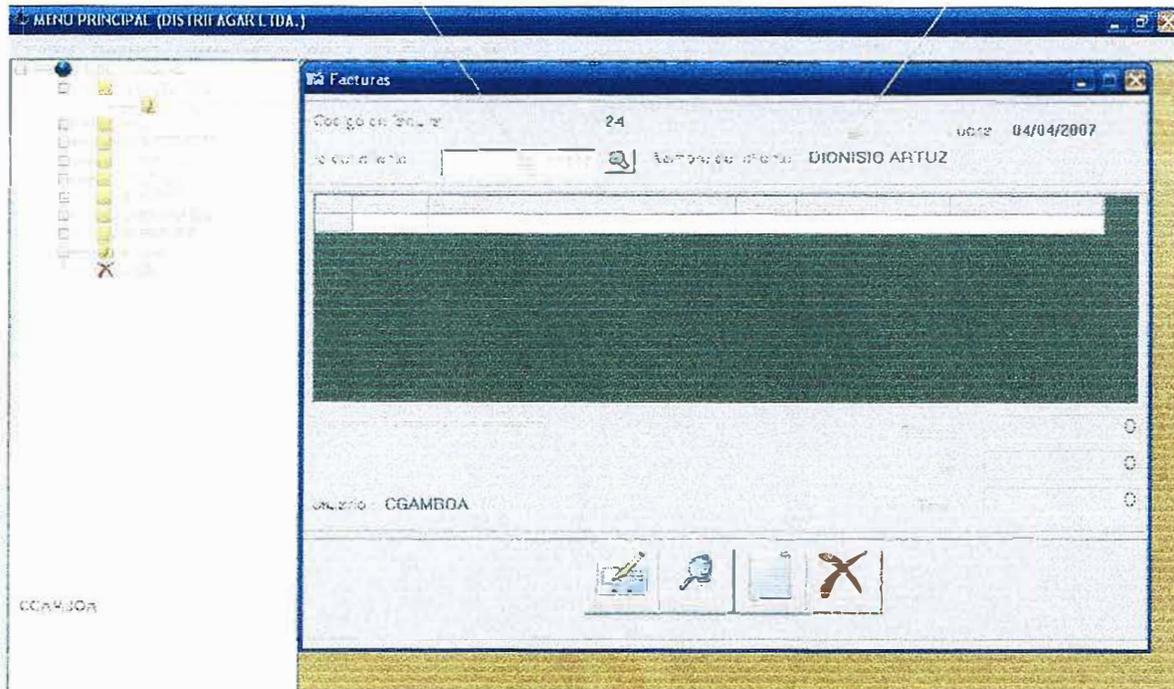
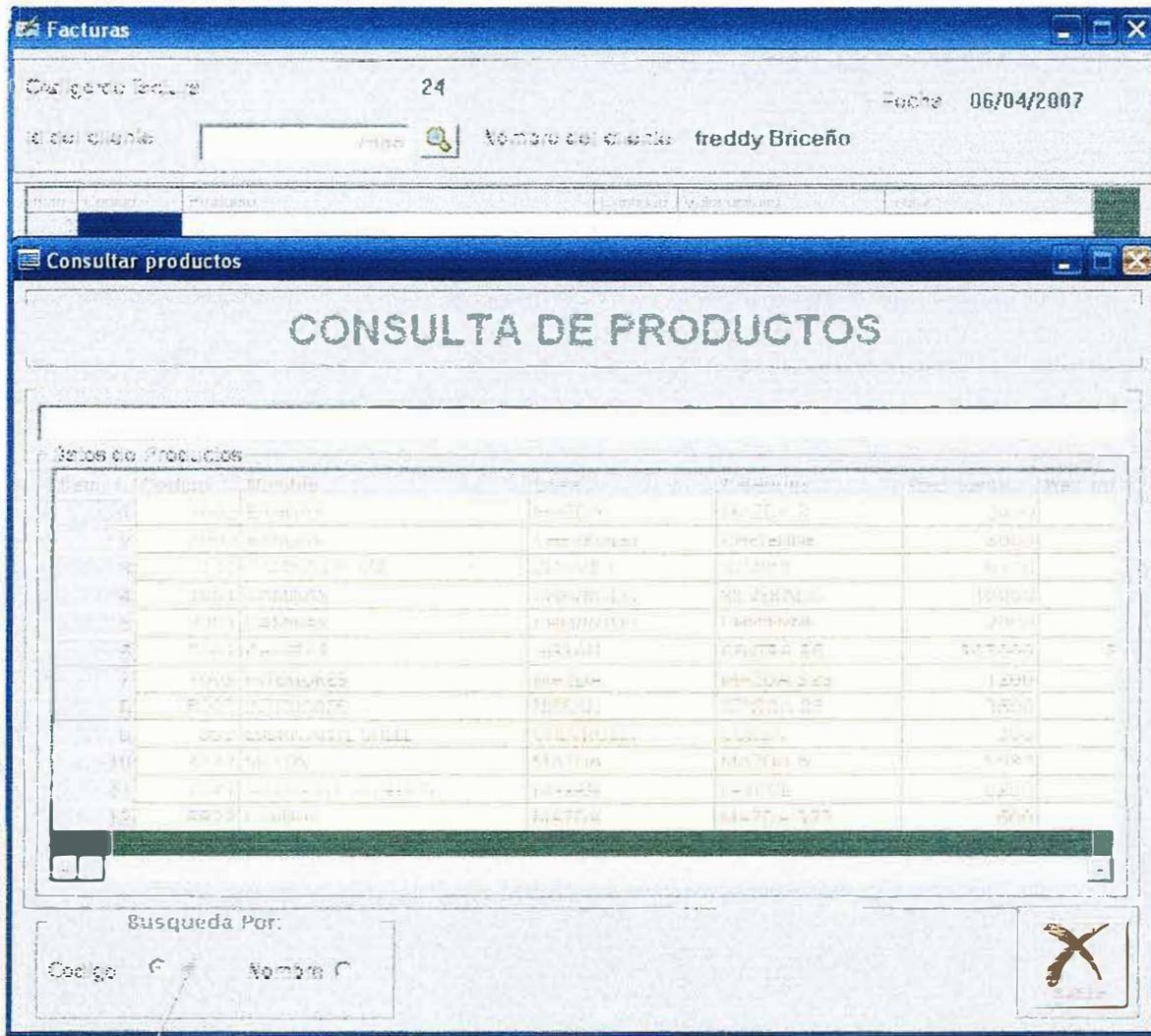


Figura 5.1. Selección del cliente desde una lista

Si el cliente al que le vamos a generar la factura, no se encuentra en la lista de clientes, se digita el Id del cliente, el aplicativo nos informara que el cliente no se encuentra en la base de datos y nos preguntara si queremos ingresarlo o si queremos continuar (Ver mas en Clientes). Luego se escogen los productos de una lista que aparecerá al presionar la tecla de ayuda F1, en esta lista se encuentran relacionados todos los productos que han sido ingresados al sistema pero que tienen en su campo estado el valor ACTIVO. Esto es porque un producto que se encuentre en estado inactivo, no podrá ser despachado en una factura. La consulta de productos como la de cliente se puede hacer por Nombre o por Código.

Grafica 5.2



Grafica 5.2. Formulario de consulta de Productos al presionar F1

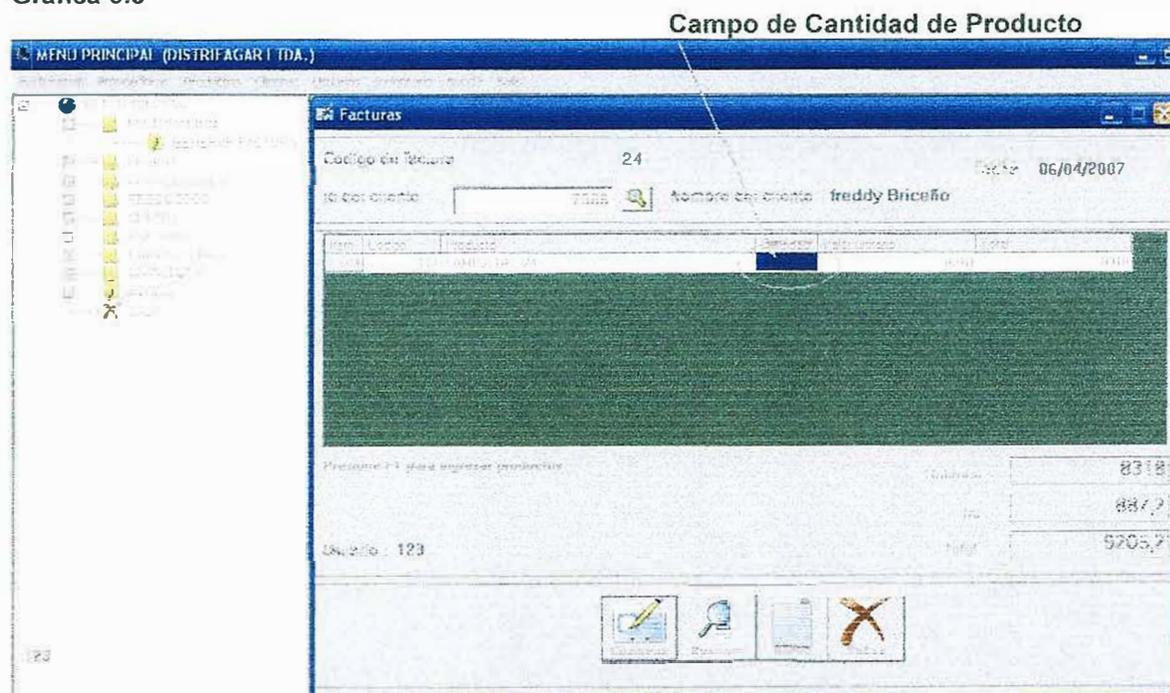
Modo de búsqueda por nombre o por Código

Después de elegir el producto que quiere que aparezca en la factura, puede seguir eligiendo de la misma forma, con la tecla F1. En la factura aparecerá el código, nombre del producto, como mínima cantidad de despacho el numero 1 y su respectivo precio unitario y precio total.

NOTA:

- Lo único que se puede digitar en el detalle de la factura es la cantidad de productos que se venderán como muestra la figura siguiente, los demás campos son de solo lectura y si se quiere modificarlos, se tendrá que hacer desde el formulario de Productos.

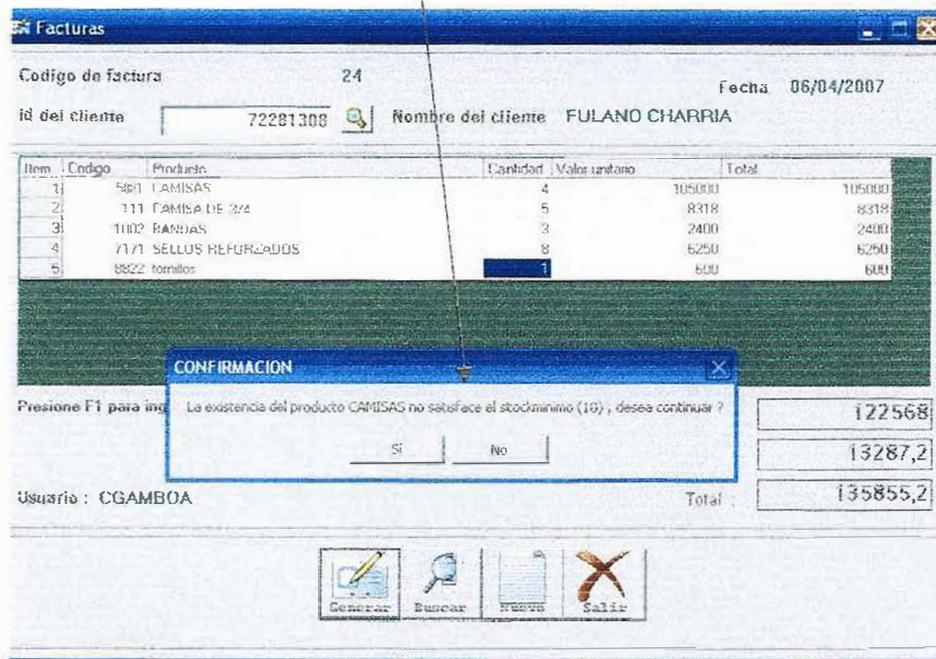
Grafica 5.3



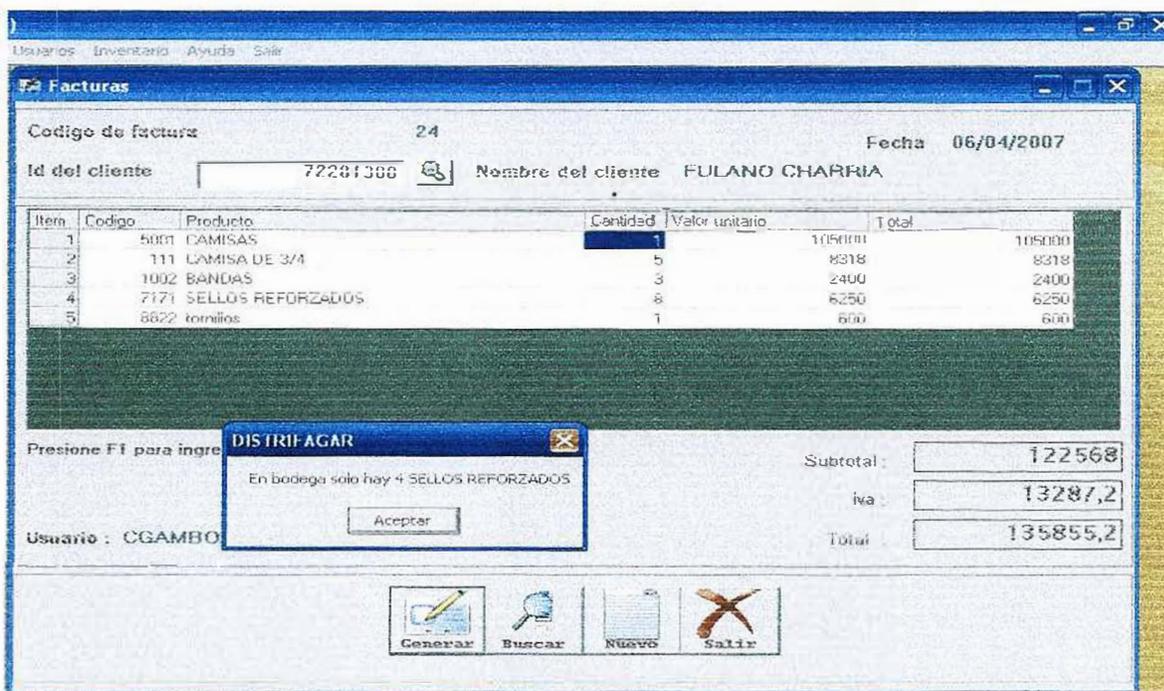
Grafica 5.3. Formulario de Factura con un producto ingresado

En caso que un producto no tenga la cantidad de existencia en bodega que requiere el Stock Mínimo asignado a ese producto, aparecerá el siguiente mensaje.

Grafica 5.4. Mensaje de Stock Minimo Bajo



Si la cantidad de producto digitada en la factura, es mayor a la existencia real en bodega, es decir, que no haya tantos productos como se requiere en la factura, el aplicativo mostrara un aviso advirtiendonos.



En caso que todo este en orden, y se cumplan todos los requerimientos básicos anteriormente mencionados, se generara el reporte de la factura. Cabe anotar que el procedimiento de ingreso de la factura a la base de datos, se hace por debajo de la operación de generar, cuando aparece el reporte de la factura presto a ser impreso, ya el aplicativo ha ingresado en la base de datos los datos pertinentes.

A demás de esto, que si intentamos ingresar dos veces un mismo producto en la factura, el aplicativo nos avisa que ese producto ya se encuentra en la lista.

Grafica 5.6

tos	Nombre	Linea	Categoria	Prec. venta	Prec. co
02	BANDAS		MAZDA 3	2400	
02	BANDAS		CHEYENNE	3000	
11	CAMISA DE 3/4		SYMBOL	8318	
01	CAMISAS		SILVERADO	10000	
01	CAMISAS		CHEYENNE	2000	
01	CAMISAS		CENTRA 86	105000	7
03	INTERIORES	MAZDA	MAZDA 323	1200	
03	INTERIORES	NISSAN	CENTRA 86	3600	

DISTRIFAGAR

Este producto ya esta en la factura

Aceptar

DISTRIFAGAR LTDA.

NIT. 802.010.702-4 Régimen Común
Res. N° 020800072093 de 2006/07/07
Facturación Autorizada 16001 - 18000

Factura 24

Fecha : 06/04/2007
Cliente : 72.281.308 Nombre : EULANO CHARRIA
Dirección : calle 43
Teléfono : 3345158

Item	Cantidad	Código	Descripción	Línea	Categoría	Valor Unitario	Total
1	16	5353	SELLOS	MAZDA	MAZDA 6	5937	201.838
2	1	5001	CAMISAS	NISSAN	CENTRA B6	105.000	105.000
3	5	111	CAMISA DE 3/4	RENAULT	SYMBOL	8.318	3.318
4	2	1002	BANDAS	MAZDA	MAZDA 3	2.400	2.400
5	1	7171	SELLOS REFORZADOS	NISSAN	PATROL	6.250	25.000
6	1	8822	tomillos	MAZDA	MAZDA 323	600	600

Usuario : CGAMBOA

Subtotal : 343.176,00
Iva : 54.906,16
Total : 398.082,16

1 of 1 | Cancel | Close | 6 of 6 | Total:€ | 100%

Grafica 5.6. Factura generada por el aplicativo

Imprimir Factura

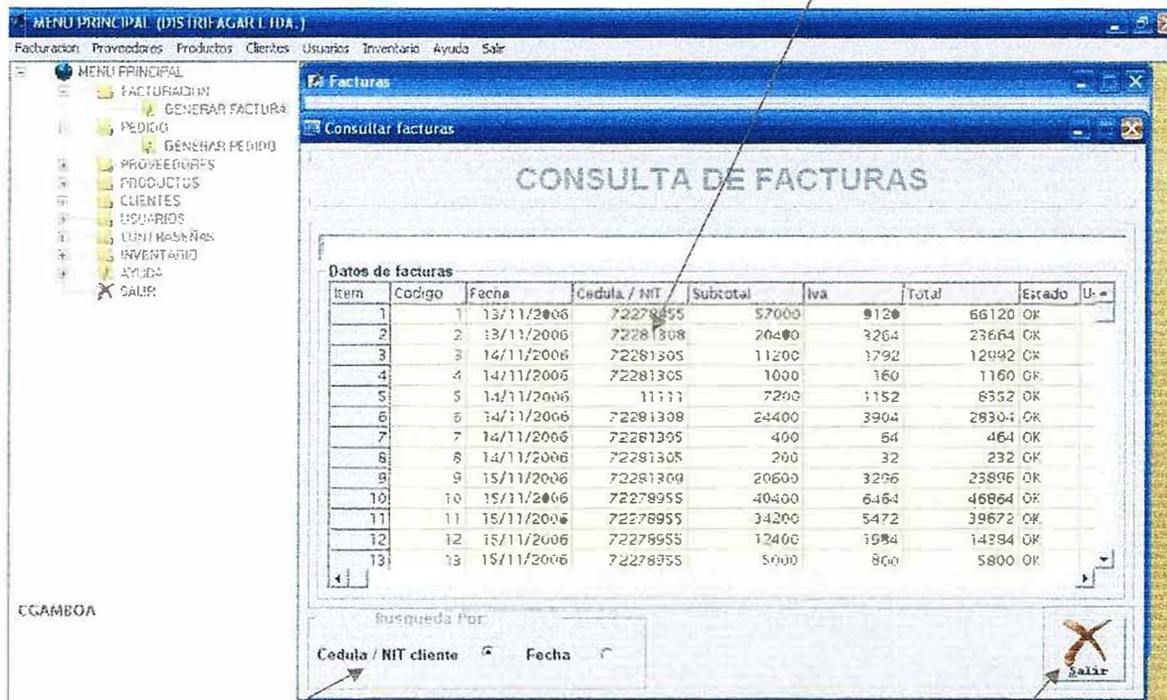
CONSULTAR FACTURA

Si queremos consultar una factura que se haya hecho, hacemos clic en el botón **Buscar**, y aparecerá una lista con todas las facturas que se han generado, con su respectivo código, fecha de ejecución, Nit/Id proveedor, subtotal, IVA, total, el estado y el nombre del usuario que ejecuto la factura, esto ultimo por cuestión de seguridad. La consulta de facturas, se puede hacer por dos modos de búsqueda, por cedula-Nit del cliente o por Fecha de la Factura.

Para cargar estos datos al formulario inicial de Facturas, hacemos doble clic a la factura que queremos ver y listo.

Grafica 5.7

Para cargar una de estas facturas hacemos doble click sobre esta



Grafica 5.7. Consultar Factura

Consulta de Facturas por Cedula o por fecha

Salir de Consulta

Por integridad referencial, los datos consultados no podrán ser modificados, simplemente están en modo de Solo lectura, para regresar otra vez al modo de ingresar factura, le damos clic al botón limpiar.

MODULAR PEDIDOS

El proceso de pedidos es muy parecido al de factura, lo único que los diferencia realmente es el procedimiento que se ejecuta en la base de datos, mientras que en la Facturación decrementa la existencia de productos, en Pedidos aumenta.

También el método de ingreso es igual, solo se digita la cantidad del producto y los demás campos se cargan como solo lectura.

GUARDAR PEDIDO

Grafica 6.

Código de factura del proveedor Consultar Proveedor Área de Nombre Proveedor

Id del Proveedor

MENU PRINCIPAL (DISTRIBUIDORA LTDA.)

Facturacion Proveedores Productos Clientes Usuarios Inventario Ayuda Salir

MENU PRINCIPAL
FACTURACION
PEDIDO
PROVEEDORES
PRODUCTOS
CLIENTES
USUARIOS
CONTRASEÑAS
INVENTARIO
AYUDA
SALIR

Generar Pedido

Código factura: Fecha: 06/04/2007

Id del Proveedor: Nombre del Proveedor:

Item	Codigo	Producto	Cantidad	Costo unitario	Total
1					

Presione F1 para ingresar productos

Subtotal: 0

Iva: 0

Total: 0

Usuario: CCAMBOA

Guardar Borrar Nuevo Salir

CCAMBOA Nombre del Usuario Guardar Pedido Limpiar detalles Consultar Pedido Salir

Figura 6. Formulario de Pedidos

Para generar un pedido, es lo mismo que en una factura, solo que ahora es el código del proveedor y el código del pedido es el de la factura de nuestro proveedor.

Digitamos el código de la factura que viene de nuestro proveedor, asignamos el proveedor de quien viene la factura buscando lo en la lista que se despliega al dar

clic en el botón **Consultar proveedor**. Luego presionamos **F1** y veremos la lista de los productos, los escogemos haciendo doble clic sobre estos y listo.

A diferencia de la factura, el Pedido no nos genera un reporte, al darle clic al botón **Guardar**, si los datos están llenos correctamente, el aplicativo guarda en la base de datos y nos da el aviso que ha hecho la operación satisfactoriamente **como lo muestra la siguiente figura**

Figura 6.1

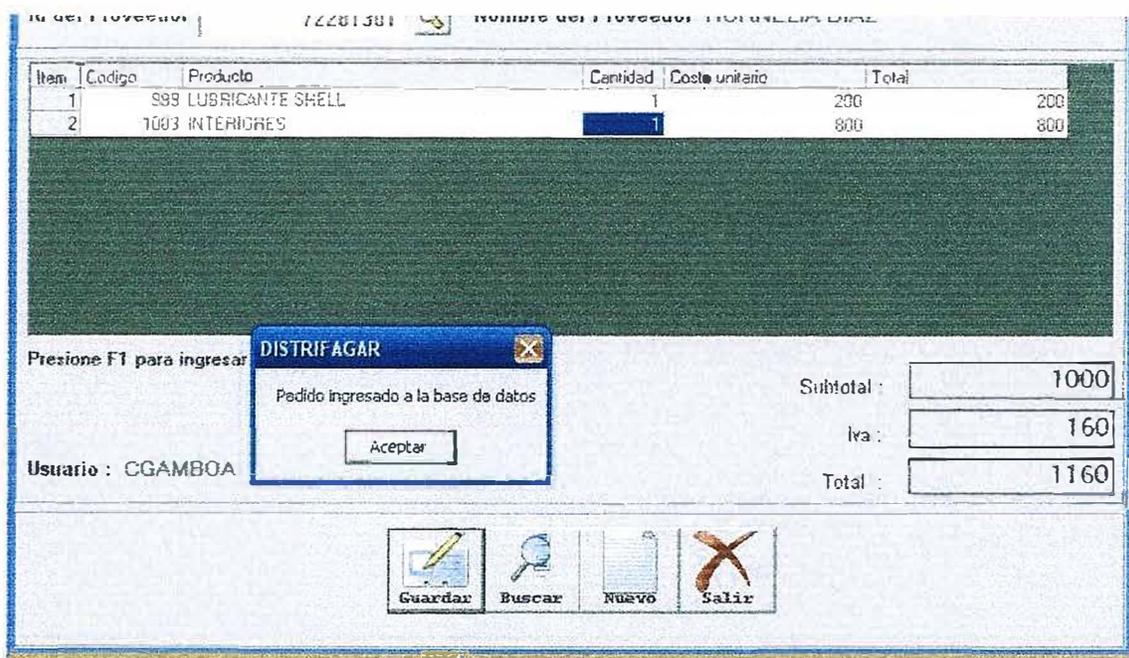


Figura 6.1. Aviso de ingreso satisfactoria a base de datos

De no ser así, el aplicativo se encargara de avisarnos.

Figura 6.2

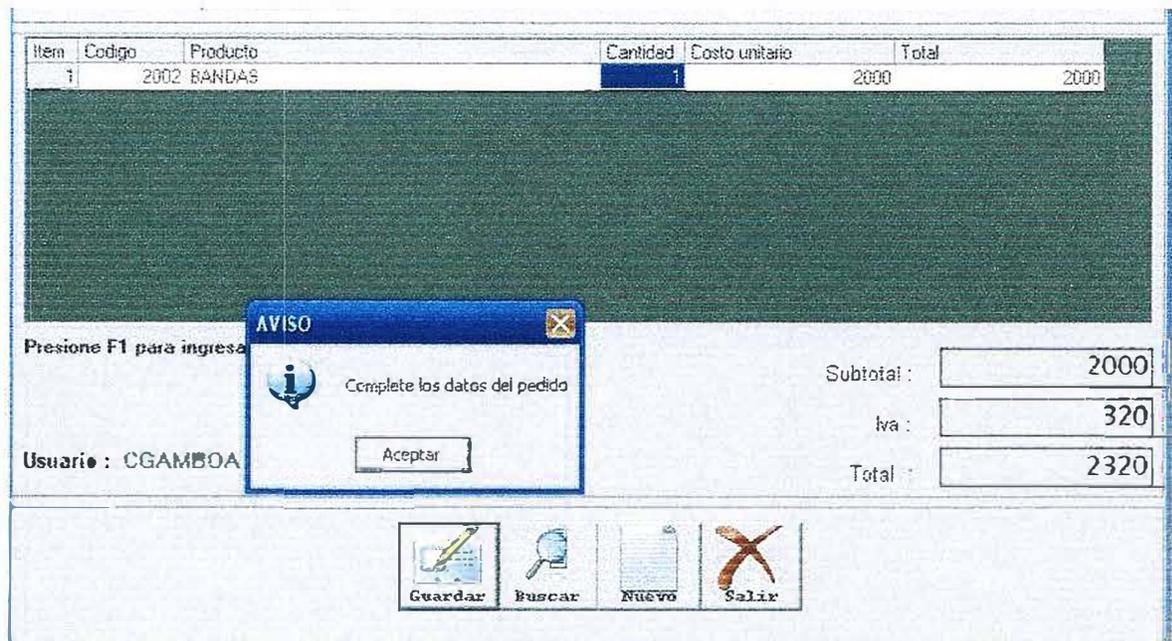


Figura 6.2.Aviso de datos incompletos

CONSULTA DE PEDIDOS

La consulta de pedidos, muestra un formulario de solo lectura en el que aparecen los pedidos que han sido realizados, cada uno con su respectivo código valor, usuario, entre otros. Este formulario se despliega al hacer clic sobre el botón **Buscar** y también tiene el mismo método de búsqueda por ID proveedor o por fecha.

Figura 6.3

Caja de texto para escribir el Id Proveedor o Fecha dependiendo del Modo de Búsqueda

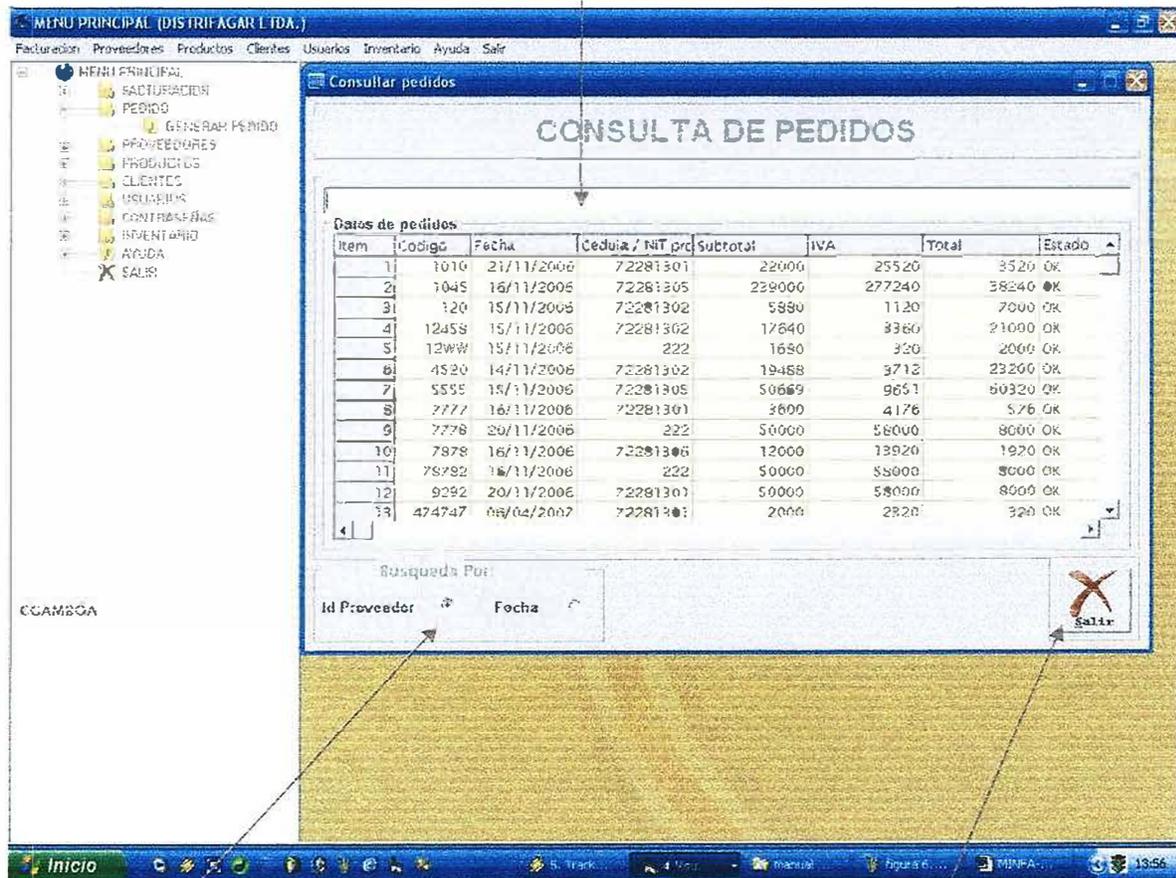


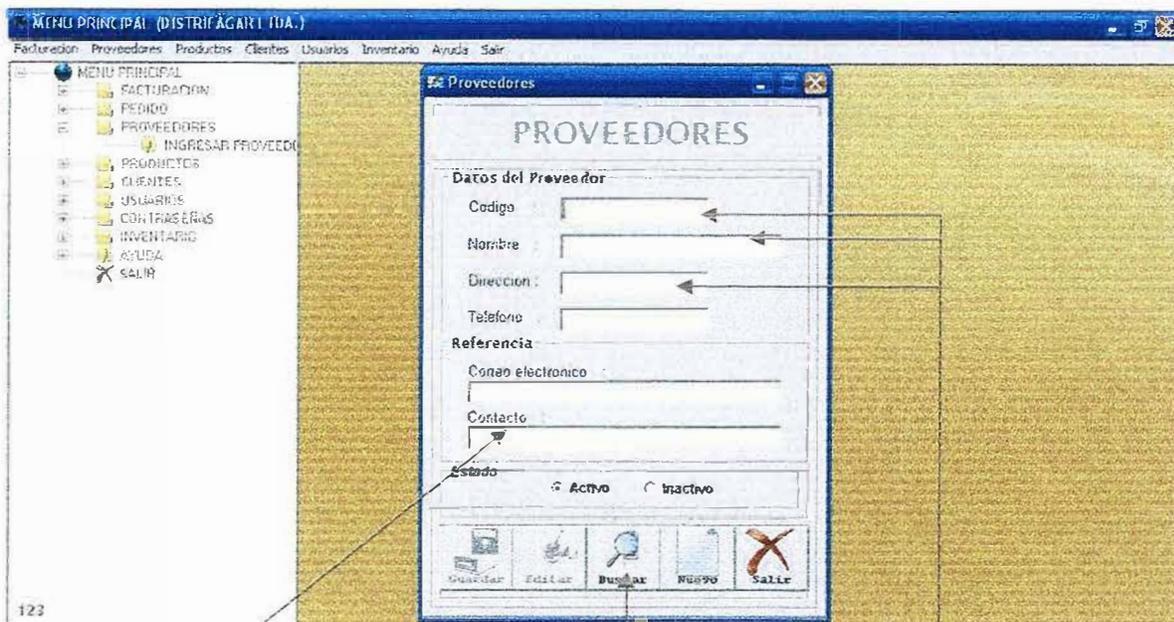
Figura 6.3. Consulta de Pedidos realizados.

Modo de Búsqueda

Salir de la Consulta

MODULAR PROVEEDORES

Grafica 7.



Grafica 7. Formulario de Proveedores

Nombre del asesor de ventas

Campos de Prerrequisitos

Botón Buscar

Este formulario ingresa en la base de datos los respectivos datos de los proveedores que surten la empresa. Los únicos campos que son prerrequisitos para el sistema son **Código, Nombre y Dirección**, esto quiere decir que si vamos a ingresar un proveedor, lo mínimo que podemos ingresar en la base de datos a cerca de nuestro proveedor son los campos anteriormente mencionados.

El área de **Estado** es prerrequisito, pero como mínimo ya esta cargado el estado activo, lo que quiere decir que jamás este valor será vacío.

Los demás campos, tales como **Teléfono**, y el Área de Referencia como **Correo electrónico y Contacto** no son necesarios para el sistema así que se podrá ingresar a la base de datos a un proveedor que no posea ninguno de estos datos.

Si se intenta ingresar un proveedor sin llenar, al menos uno de los campos de prerequisites, el aplicativo informara que falta llenar un campo y hasta que no se cumpla que los campos de prerequisites estén completos, no permitirá ingresar a la base de datos.

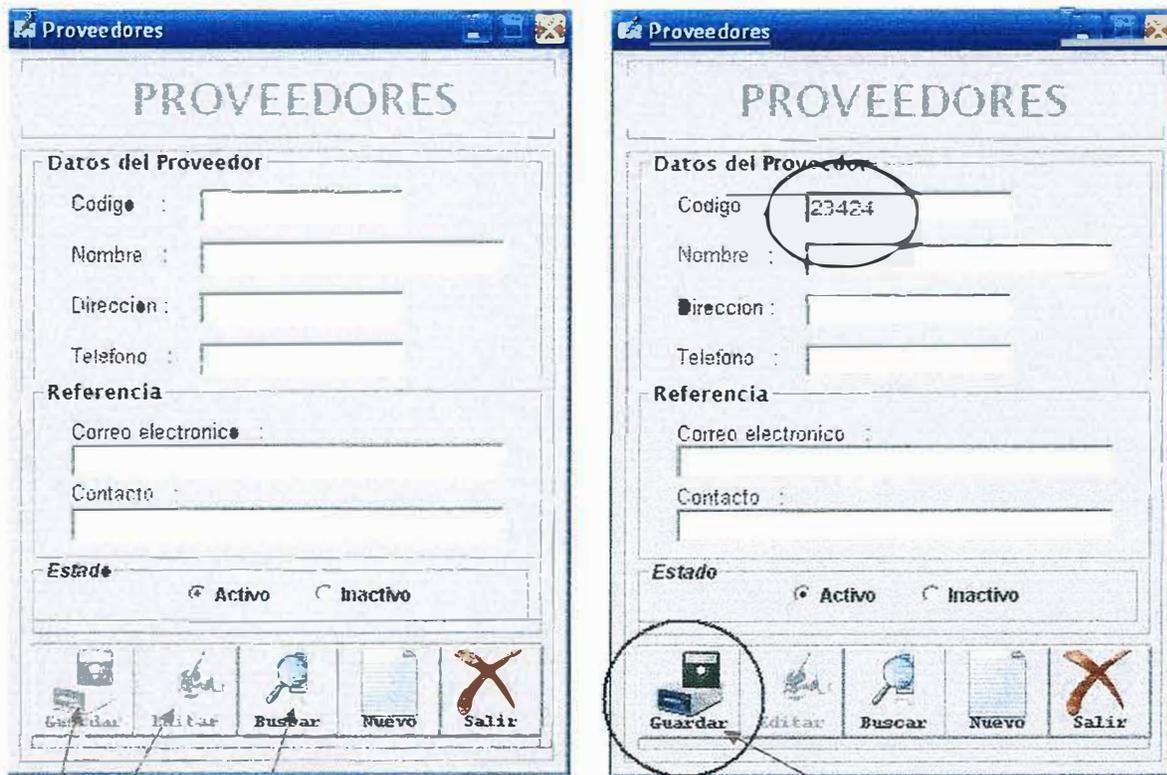
Los datos del proveedor almacenado, se recopilan para ser mostrados en el aplicativo de Facturación y de Pedidos, y para hacer otras consultas,

NOTA:

- Solo aparecerán en las listas de consulta, tales como consultar proveedor (Para ingresar el código del proveedor en Facturas y Pedidos) aquellos que en su Área de **Estado** este **activo**.
- Esta regla rige para la mayoría consultas.

Cuando abrimos el formulario, los Botone **Guardar** y **Editar**, aparecen deshabilitados, una vez se digita el código del proveedor y se dispone a escribir el nombre, se activa el botón **Guardar**, permitiendo así que cuando se llenen todos los campos, se pueda guardar.

Grafica 7.1



Grafica 7.1. Activación del Botón Guardar

Botón Buscar

Botón Guardar Activado

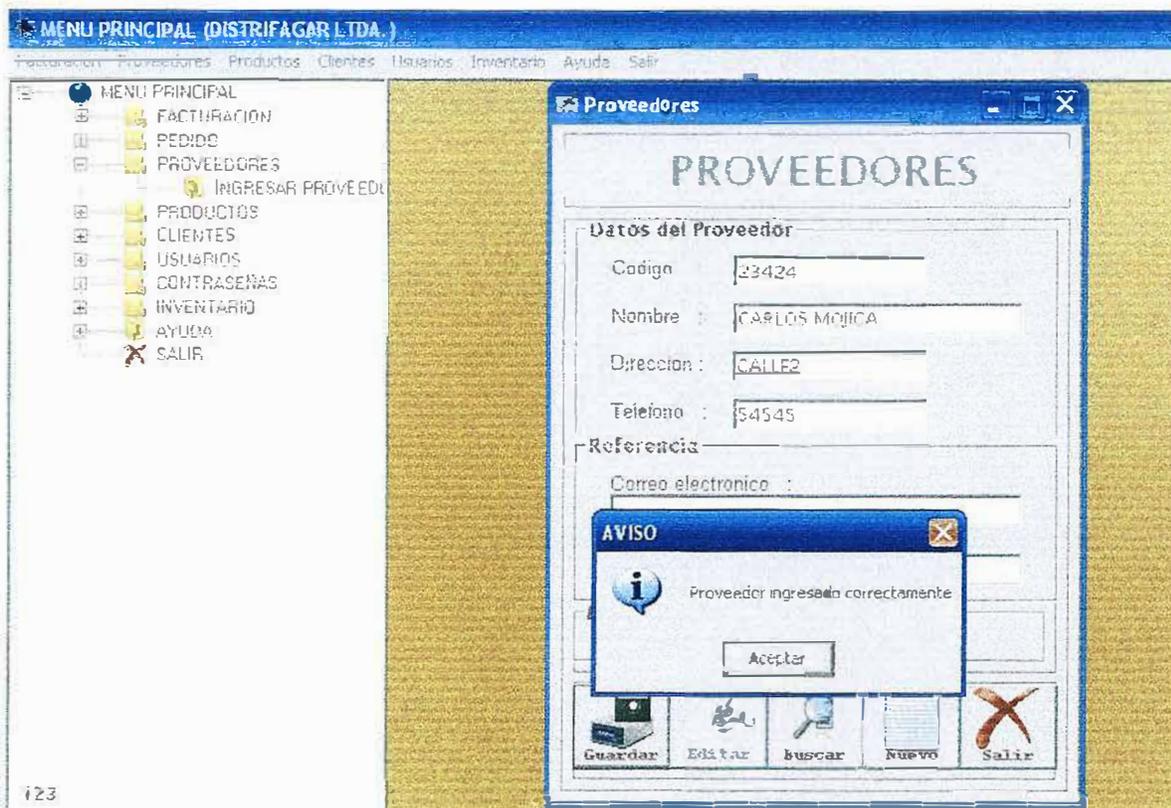
Botón Desactivado

Una vez digitados todos los campos necesarios, al hacer clic en el Botón **Guardar**, si todo esta en orden los datos se guardan en la base de datos.

Si consultamos un proveedor, al cargar sus datos en el formulario, la caja de texto Código, aparecerá como Solo lectura.

Esto ultimo vale para todos los formularios que manejen Código o Id, esoto es por conservar principios de programación y de integridad referencial.

Grafica 7.2



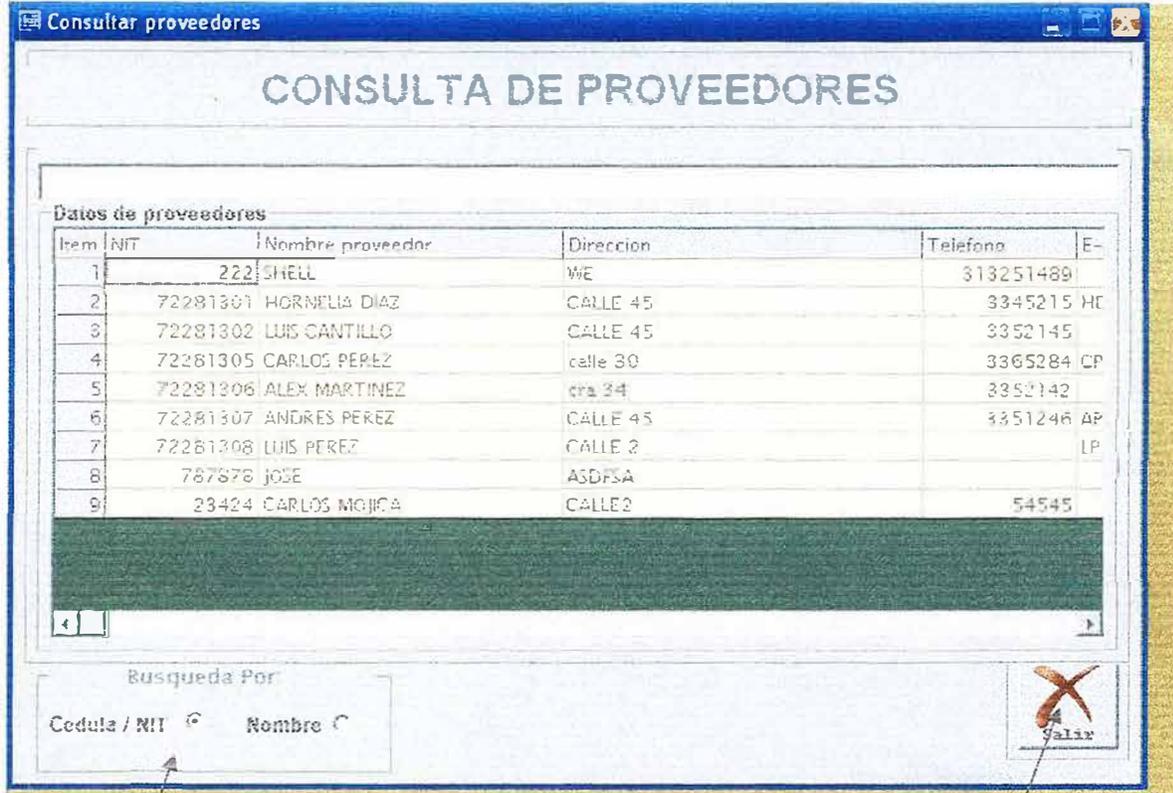
Grafica 7.2. Proveedor ingresado a la base de datos

CONSULTAR PROVEEDOR

Para saber un dato de un proveedor o consultarlo para modificarlo, hacemos click en el botón **Buscar**.

Aparecerá un formulario donde se listan todos los proveedores (Que tengan en su área de estado el valor Activo), con un área de modo de búsqueda que puede ser por Id o por Nombre. Se puede hacer doble click en un proveedor para cargar los datos al formulario de proveedores.

Grafica 7.3



Grafica 7.3 Formulario de Consulta de Proveedores

Modo de búsqueda

Salir de la consulta

Al hacer doble clic sobre uno de los proveedores este se carga en el formulario de proveedores, inmediatamente el botón **Editar** se activa y se deshabilita el botón **Guardar**.



Botón Editar activado

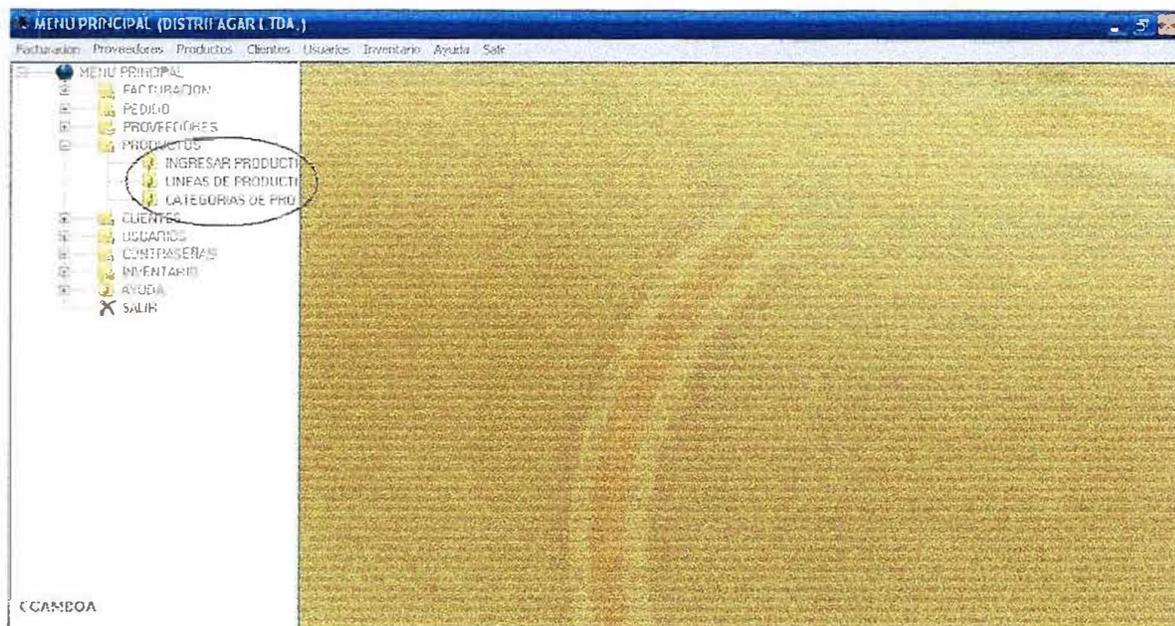
Al hacer click sobre editar, si hemos hecho cambios en los datos del proveedor, estos son modificados y guardados en la base de datos.

Cuando hacemos click sobre el botón limpiar, los campos del formulario quedan vacíos y los botones **Editar** y **Guardar**, nuevamente quedan desactivados.

MODULAR PRODUCTOS

En este punto encontramos tres posibles formularios, para ingresar una línea de productos, para ingresar una categoría y otro para ingresar productos a la base de datos.

Grafica 8



Grafica 8. Productos.

INGRESAR LINEAS DE PRODUCTOS

En este formulario podemos ingresar o modificar una línea para asignarle a los productos, consta de un código y un nombre descriptivo.

Grafica 8.1



The image shows a screenshot of a software application window titled "Ingresar Lineas". The window has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. The main content area is white and features the title "LINEA DE PRODUCTO" in large, bold, black capital letters. Below the title, there are two input fields: "Codigo de linea :" and "Nombre :". At the bottom of the window, there is a toolbar with five buttons: "Guardar" (with a floppy disk icon), "Editar" (with a pencil icon), "Buscar" (with a magnifying glass icon), "Nuevo" (with a document icon), and "Salir" (with a red X icon).

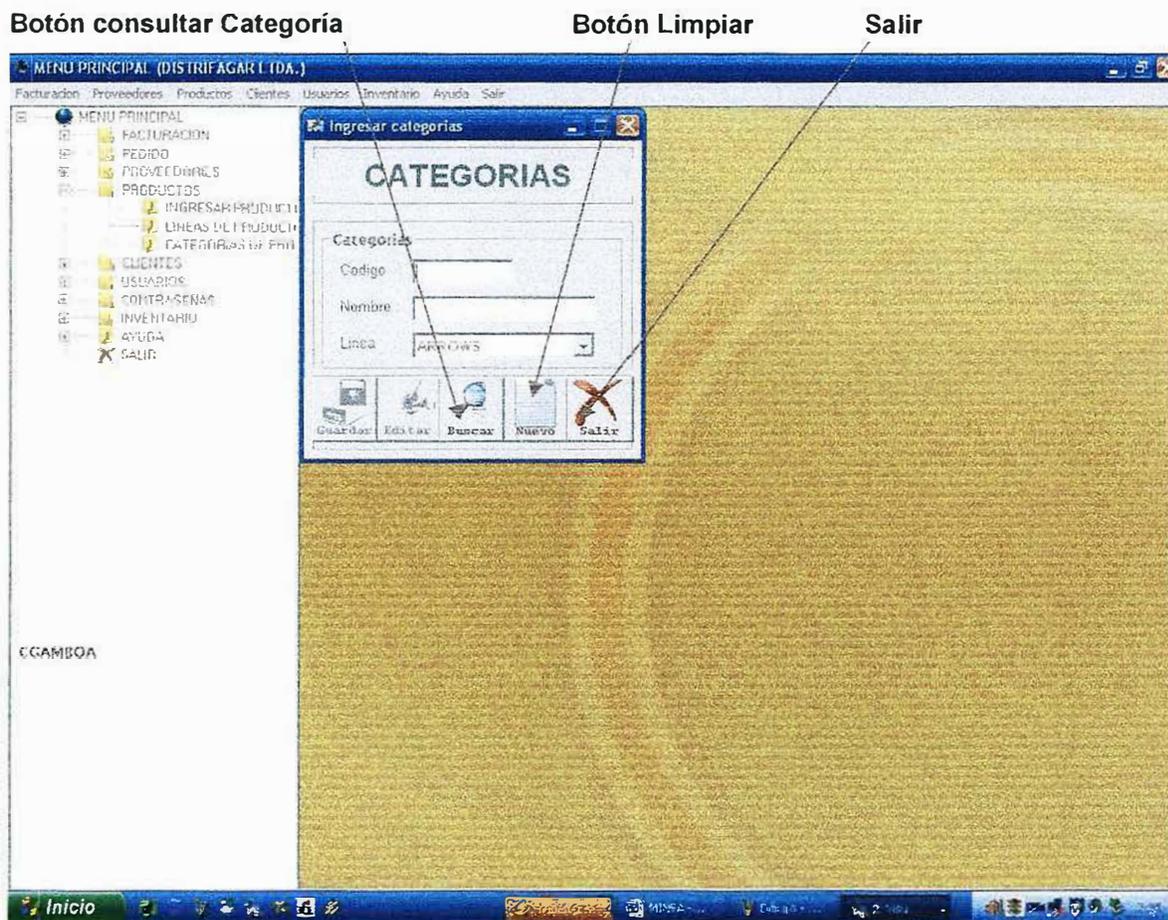
Grafica 8.1 Formulario de ingreso de líneas

En este formulario como en la mayoría de los demás, el botón **Guardar** solo se activan cuando se digita un valor en el código y el botón **Editar** cuando se consulta una línea con el botón **Buscar**.

INGRESAR CATERORIAS

En este formulario se digita el código y el nombre de la categoría, además de esto, se le asigna una de las líneas que anteriormente se digitaron en el formulario de líneas.

Grafica 9



Grafica 9. Formulario para ingreso de categorías

Para consultar una categoría esta el botón **Buscar**, al hacer click se despliega una lista con todas las categorías

FORMULARIO INGRESAR PRODUCTOS

Este tal vez pueda que sea uno de los formularios más importantes para el aplicativo. Este formulario carga varias listas en las que tenemos que escoger, como IVA, categoría, línea, entre otros.

Este formulario posee un área de estado, en el que podemos asignarle activo o inactivo. Si el valor asignado es **Inactivo**, debemos recordar que no aparecerá en las consultas que se realicen desde pedidos o desde Facturación .

Grafica 10.

Área de datos del producto

Área de referencias

The screenshot shows a software window titled 'Productos' with a form titled 'REGISTRO DE PRODUCTOS'. The form is divided into two main sections: 'Datos del producto' and 'Referencias'. The 'Datos del producto' section includes fields for 'Codigo producto', 'Nombre producto', 'Precio de compra', 'Precio de venta', and 'Utilidad' (set to 20). The 'Referencias' section includes dropdown menus for 'IVA' (16), 'Linea' (CHEVROLET), 'Categoría' (AVEO), and 'Bodega' (BODEGA AUX). Below these is an 'Estado' section with radio buttons for 'ACTIVO' and 'Inactivo'. At the bottom of the form are buttons for 'Guardar', 'Editar', 'Buscar', 'Nuevo', and 'Salir'.

Grafica 10. Formulario de ingreso de productos

Este formulario posee dos Áreas:

ÁREA DE DATOS DEL PRODUCTO

Esta área consta:

- Código de producto, Caja de texto numerico
- Nombre del productos, Caja de texto letras
- Precio de compra, Caja de texto numerico
- Precio de venta, Caja de texto numerico
- Utilidad, multilista que devuelve la ganancia de la venta de un producto

En este formulario a diferencia de otros, todos los campos son prerequisites, es decir, para poder ingresar un producto al sistema, deben estar llenos todos los datos que este formulario exige.

ÁREA DE REFERENCIA DEL PRODUCTO

Esta área consta de:

- IVA, una multilista en la que se escoge un valor
- Stock Mínimo, Caja de texto numérico
- Stock Máximo , Caja de texto numérico
- Línea, multilista donde se escoge un valor

- Categoría, multilista
- Bodega, es una multilista que contiene las bodegas posibles en las que podemos almacenar los productos, solo se mostraran las bodegas que tengan en su campo el valor activo.

Como se puede observar, los botones **Guardar y Editar** se encuentran deshabilitados, recordemos que para todos los formularios estos botones no se cargan sino después de unos eventos, digitar un valor en el caso del botón **Guardar** y después de cargar los datos en el formulario desde una consulta para el botón **Editar**.

Grafica 10.1

Grafica 10.1. Botón Guardar Activado

Para hacer una consulta, haga click en el botón buscar; se desplegara una lista con los productos que están guardados en la base de datos, recuerde que solo

aquellos que tengan en su estado el valor Activo. Si se presiona doble click sobre uno de los productos citados en la lista de consulta, este se cargara en el formulario y entonces el botón **Editar** se activara para poder guardar los cambios, en caso de que se haya modificado uno de los datos del producto.

Grafica 10.2

The screenshot shows a software application window titled "Productos" with a "REGISTRO DE PRODUCTOS" form. The form is divided into two main sections: "Datos de producto" and "Referencias".

Datos de producto:

- Código producto: 998
- Nombre producto: LUBRICANTE SHEL
- Precio de compra: 200
- Precio de venta: 300
- Unidad: L
- Cantidad: 9

Referencias:

- IVA: 18
- Stock minimo: 1
- Stock maximo: 10
- Linea: LUBRICANTE
- Categoria: TIPOSA
- Marca: SHELL

At the bottom of the form, there are several buttons: "Guardar", "Cancelar", "Eliminar", "Actualizar", and "Editar". The "Editar" button is highlighted with a red 'X' over it, indicating it is active.

Grafica 10.2. Botón Editar activado después de una consulta

Una vez digitados todos los datos correctamente, hacemos click en el botón **Guardar** y listo, se ha ingresado un nuevo producto a la base de datos.

MODULAR CLIENTES

Los clientes son todas aquellas personas o entidades que solicitan los productos. Para que un cliente pueda ser incluido en una factura, primero debe ser guardado en la base de datos.

Si guardamos el cliente en la base de datos directamente desde el formula, este se guarda en la base de datos y podrá ser consultado desde facturación para generar una factura(Esto siempre y cuando su Área de Estado este Activa).

En caso contrario, al generar la factura con un cliente nuevo, es decir, que no se encuentre almacenado en la base de datos, el formulario facturación pedirá que sea ingresado a la base de datos como lo muestra la siguiente figura.

Figura 10.3

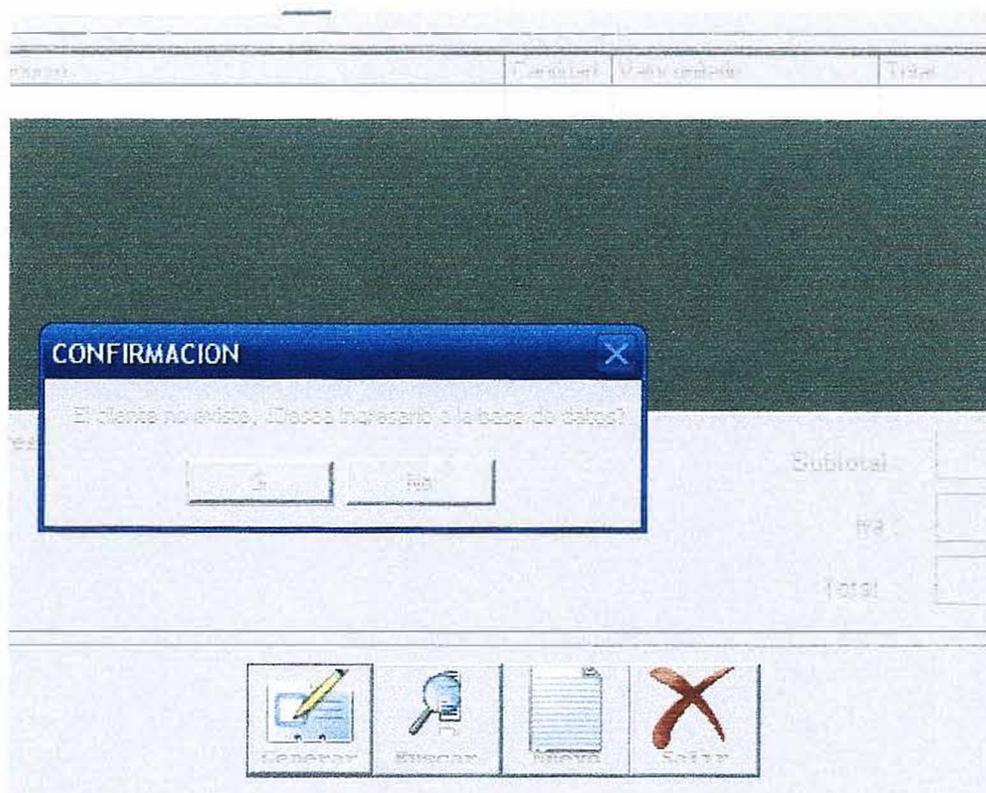


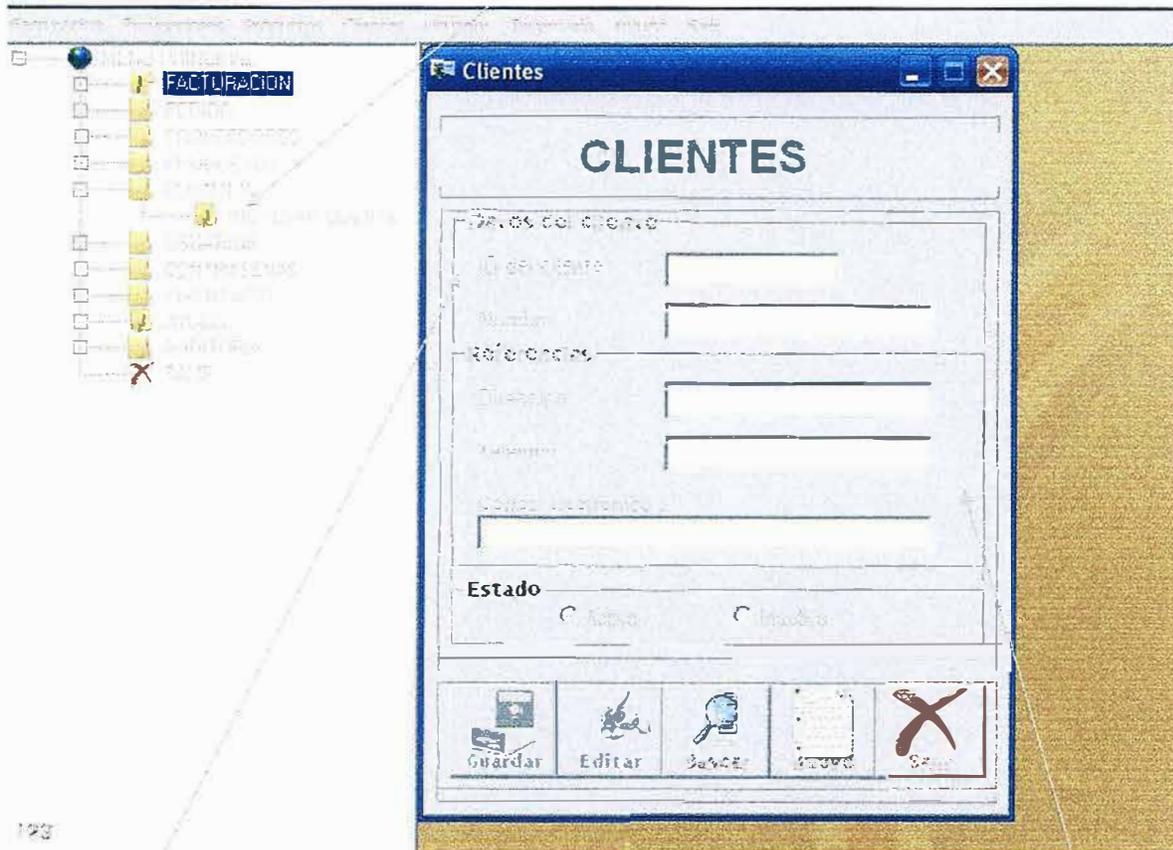
Figura 10.3. Cuando el cliente no existe, se pedirá que sea ingresado

FORMULARIO INGRESAR CLIENTE

Para llegar a este formulario nos dirigimos al modular CLIENTES y hacemos click en INGRESAR CLIENTE

Figura 11.

Modular Clientes



Área datos del cliente

Figura 11. Formulario de clientes

Área Referencias

El formulario de clientes cuenta con tres áreas de información

ÁREA DATOS DEL CLIENTE

Esta área de información cuenta con:

- Id De Cliente, Caja de texto numérico
- Nombre, Caja de texto de letras

ÁREA DE REFERENCIAS

Esta área guarda los datos de ubicación del cliente tales como:

- Dirección, Caja de texto numérico
- Teléfono, Caja de texto letra-numérico
- Correo electrónico, Caja de texto letra-numerico

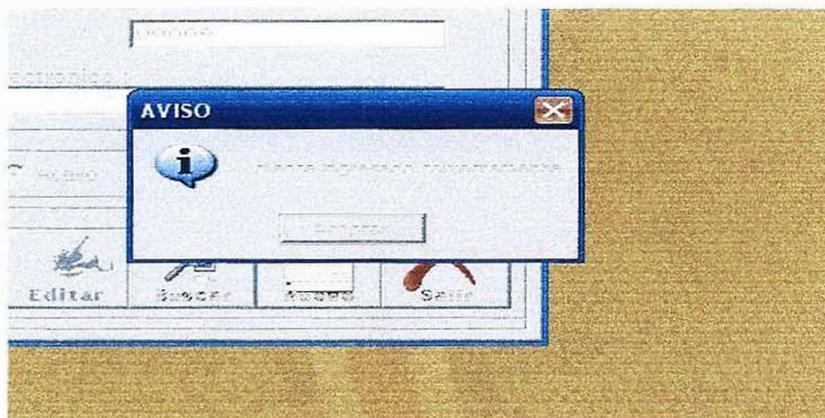
ÁREA DE ESTADO

Esta área guarda el estado del cliente actualmente, puede ser:

- Activo (Puede ser consultado por otros formularios), Opción para escoger
- Inactivo (Fuera de consultas), Opción para escoger

El botón **Guardar** y **Editar**, siguen con las mismas reglas establecidas para lo formularios anteriormente explicados, y de igual manera los botones **Guardar**, **Nuevo** y **Salir**

Una vez digitados los datos correctamente todos los datos, al hacer click en el botón **Guardar** y estos son guardados en la base de datos y listo, hemos ingresado a un nuevo cliente.



Si hacemos una consulta de un cliente, al cargar los datos del cliente en el formulario, la caja de texto **Id Cliente** permanecerá de Solo lectura, esto quiere decir que no podrá ser modificada, esto es por conservar el principio de integridad referencial y otros principios de programación.

MODULAR USUARIOS

Los usuarios son aquellas personas que pueden acceder a la aplicación MINFA por medio de un **Nombre de Usuario y contraseña**. Dependiendo de este usuario se cargara la aplicación con un perfil determinado por el administrador de usuario a la hora de crear el usuario.

Recordemos que por existen unos perfiles de usuarios predeterminados y que cada usuario de acuerdo a su perfil puede ejecutar ciertas funciones.

FORMULARIO INGRESAR USUARIOS

Grafica 12 Modular usuarios

The screenshot shows a software interface for user management. The main window is titled 'Ingresar Usuarios' and contains a form titled 'ADMINISTRACION DE USUARIOS'. The form is divided into two main sections: 'DATOS DEL USUARIO' and 'DATOS DE LA SESION'. The 'DATOS DEL USUARIO' section includes fields for 'Código', 'Nombre', 'Apellido', 'Referencias', 'Telefono', 'Direccion', and 'Correo'. The 'DATOS DE LA SESION' section includes fields for 'Nombre de usuario', 'Contraseña', 'Perfil', 'Fecha de inicio', and 'Fecha de fin'. At the bottom of the form, there is a toolbar with icons for 'Guardar', 'Editar', 'Buscar', 'Imprimir', and 'Salir'. The date '17/04/2007' is displayed in the top right corner of the form area.

El formulario de usuarios está compuesto por cuatro áreas de datos los cuales son:

ÁREA DE DATOS DEL USUARIO

Esta área consta de los datos personales del Usuario.

- Cedula, Caja de texto numérico
- Nombres, Caja de texto letras
- Apellidos, Caja de texto letras

ÁREA DE REFERENCIAS

En esta área se guardan los datos de ubicación del Usuario.

- Teléfono, Caja de texto numérico
- Dirección, Caja de texto letra-Numérico
- E-Mail, Caja de texto letra-Numérico

PERFIL DE USUARIO

Esta área es una de las más importantes del formulario, en esta se le asigna un perfil al usuario y se decide que nivel de acceso podrá tener sobre la aplicación dicho usuario. Los datos que se guardan son:

- Nombre de Usuario, Caja de texto letras
- Contraseña, Caja de texto numérico, no descifrable
- Confirmar Contraseña, Caja de texto numérico, no descifrable
- Tipo de Usuario, Multilista

ÁREA DE ESTADO

En esta área se escoge un estado para el usuario, si se escoge inactivo, el usuario no podrá entrar al sistema, tampoco podrá ser consultado.

- Activo
- Inactivo

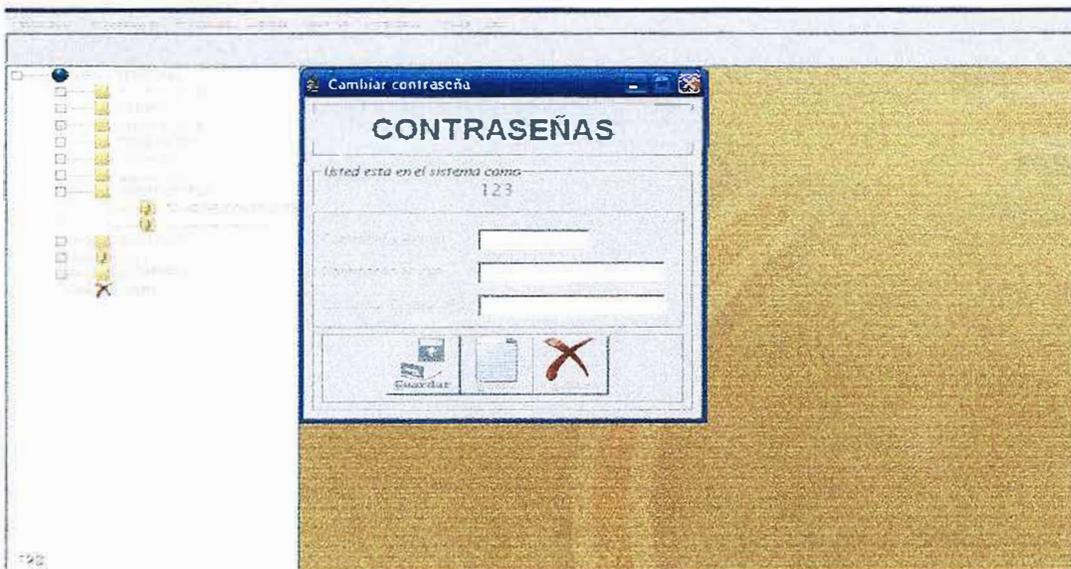
Una vez ingresado todos los datos en el formulario en su respectiva área, se presiona click para guardar los datos en la base de datos.

MODULAR CONTRASEÑAS

En este modulo se le da la opción al usuario de poder cambiar su contraseña las veces que sea necesario.

FORMULARIO CAMBIAR CONTRASEÑAS

Este formulario consta de tres cajas de texto. Contraseña actual, Nueva y Confirmar contraseña. Todas son campos letra-numérico



Para cambiar la contraseña en tiempo de ejecución del aplicativo MINFA, hay que seguir tres pasos:

1. Digite la contraseña actual en la caja de texto respectiva
2. Digite la contraseña nueva.
3. Confirme la nueva contraseña.

En caso que digite erróneamente la contraseña actual, no importa si llena los otros datos correctamente, el sistema no permitirá que usted cambie su contraseña. En

Figura 13.2

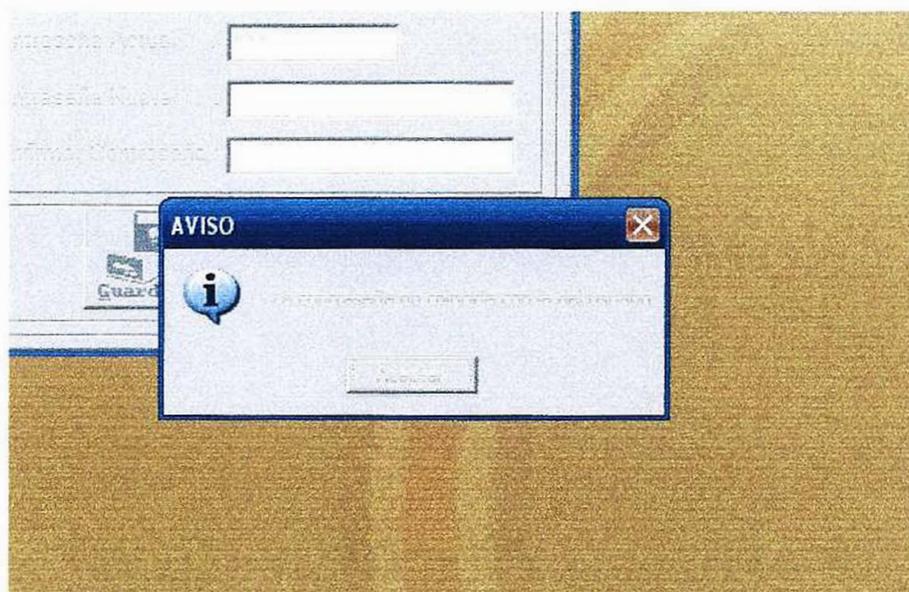


figura 13.2 Actual digitada erróneamente

Entonces saldrá un aviso diciendo que la contraseña y el usuario no coinciden.

Lo mismo ocurrirá si se digita la nueva contraseña bien pero al confirmarla la digita erróneamente.

Figura 13.3

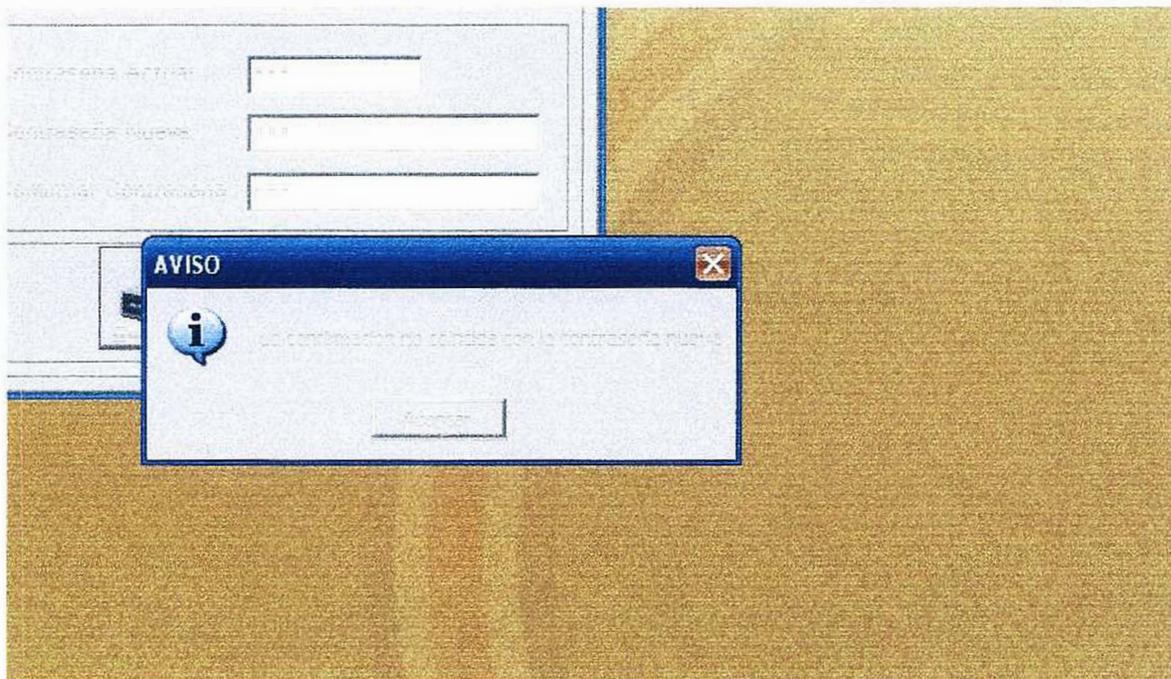


Figura 13.3 Confirmación de contraseña no valida

Por favor, asegúrese de llenar correctamente estos datos o no será posible cambiar su contraseña por una nueva.

Si todos los datos están correctamente digitados, su contraseña se cambiara por la nueva y seguirá trabajando en la aplicación.

FORMULARIO CAMBIAR SESIÓN

MINFA permite cambiar de sesión después de haber iniciado una con un perfil de usuario.

Al hacer click en **Cambiar Sesión**, el aplicativo MINFA se cerrará y dará paso al formulario de inicio de sesión, asegúrese de haber guardado los cambios para evitar contratiempos por pérdida de información.



The image shows a screenshot of a web application window titled "Inicio de Sesión". The main heading is "INICIO". Below the heading, there is a prompt: "Digite su nombre de usuario y contraseña para ingresar al sistema". There are two input fields: "USUARIO" and "CONTRASEÑA". At the bottom of the form, there are two buttons: one with a blue globe icon and another with a red 'X' icon.

Figura 13.4 inicio de sesión

MODULAR INVENTARIO

Es uno de los modulares base de MINFA, en este se puede generar reportes lineales a cerca del comportamiento de los productos en bodega, entradas y salidas, entre otros.

FORMULARIO CREAR BODEGAS

Este formulario consta de tres cajas de texto. Código, Cupo y Observaciones.

Modular Creación de Bodegas

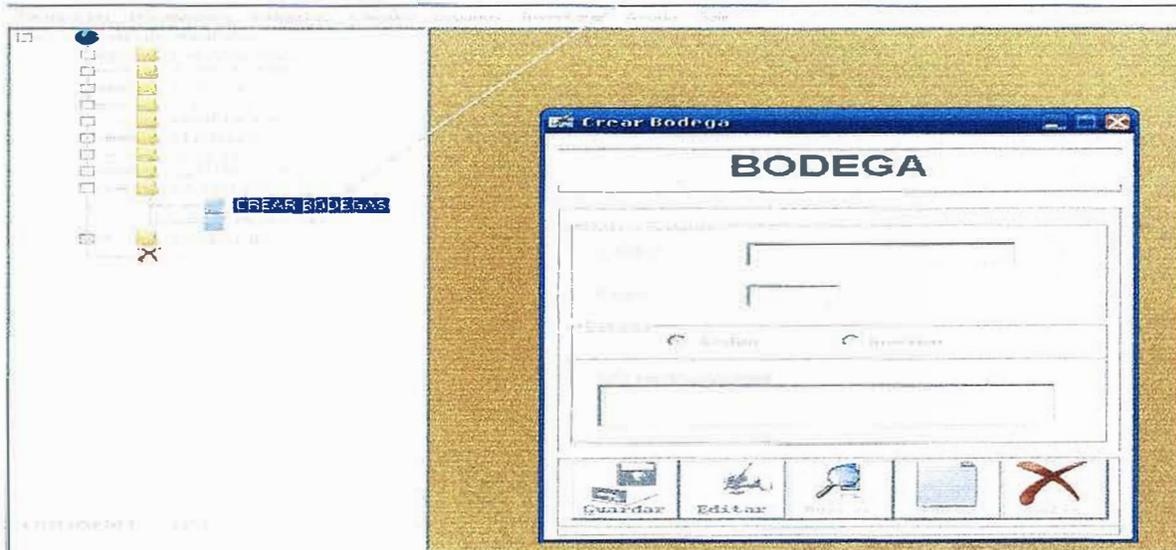


Figura 14 Figura de creación de bodegas

Una vez digitados correctamente todos los datos, al hacer click en el botón **Guardar** y estos son guardados en la base de datos, se ha ingresado una nueva bodega debe salir un mensaje de confirmación.

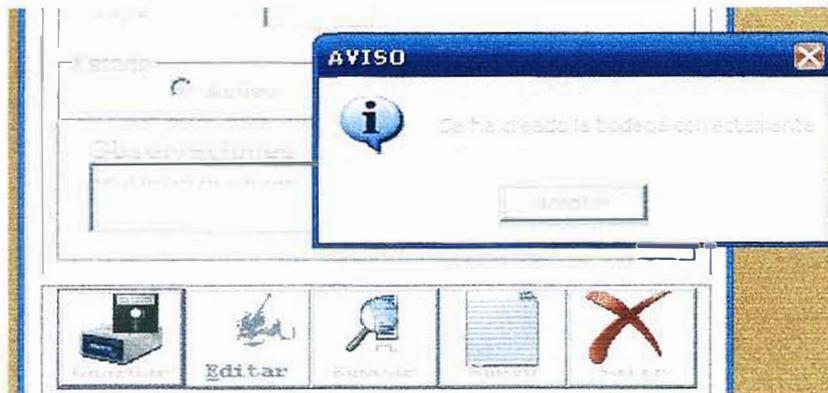


Figura 14.1 Confirmación de transacción

FORMULARIO INVENTARIO

Este formulario consta de tres listas desplegables. Fecha de inicio, Fecha final y Tipo de movimientos.

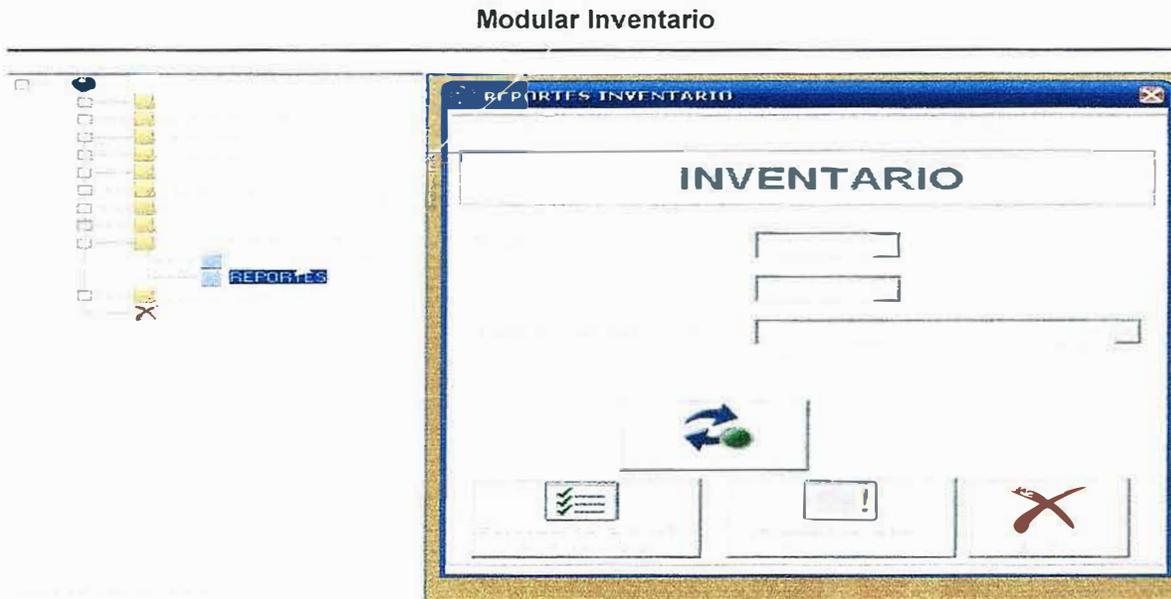


Figura 15 Figura de Consulta de inventario

Este formulario facilita al usuario verificar el estado del inventario de los productos en el almacén por medio de un reporte. En el reporte arrojado se puede depurar la información por fechas y por el tipo de movimiento que se haya generado en la aplicación.



Figura 15.1 Seleccionar fecha

Se selecciona en el criterio de la consulta el tipo de movimiento para el cual se desea el reporte.

INVENTARIO

Fecha de inicio

Fecha final

Tipo de movimiento

ENTRADA POR FACTURA DEL PROV

ENTRADA POR FACTURA DEL PROV

ENTRADA POR DEVOLUCION DEL CL

SALIDA POR FACTURACION AL CLIE

SALIDA POR DEVOLUCION AL PROV

Figura 15.2 Seleccionar tipo de movimiento

REPORTE DE MOVIMIENTOS DE PRODUCTOS - INVENTARIO

El reporte de movimientos es el resultado de la consulta del formulario inventario en el cual se dan como parámetros los criterios de la consulta y el resultado es el siguiente reporte.

Reporte de movimientos de Productos					
Fecha de Impresión: 15/03/2007					
Codigo	Nombre	Fecha	Movimiento	Cantidad	Precio
111	CAMISA DE 34	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	8318,00
111	CAMISA DE 34	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	8318,00
111	CAMISA DE 34	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	8318,00
111	CAMISA DE 34	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	8318,00
111	CAMISA DE 34	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	8318,00
999	LUBRICANTE SHELL	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	300,00
999	LUBRICANTE SHELL	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	300,00
999	LUBRICANTE SHELL	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	300,00
999	LUBRICANTE SHELL	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	300,00
1.001	CAMISAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	10.000,00
1.001	CAMISAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	10.000,00
1.001	CAMISAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	10.000,00
1.001	CAMISAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	10.000,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00
1.002	BANDAS	24/04/2007	SALIDA POR FACTURACION AL CLIENTE	1	2.400,00

Figura 15.3 Figura de Reporte de Movimientos de Productos

REPORTE DE EXISTENCIA DE PRODUCTOS - INVENTARIO

El reporte de existencias es el resultado de la consulta del formulario inventario en el cual se dan como parámetros los criterios de la consulta y el resultado es el siguiente reporte.

Reporte de Existencia de Productos							
							Fecha de Impresión: 15/03/2007
Codigo	Nombre	L ínea	Categoría	Precio Compra	Precio Venta	Existencia	Sub Total
111	CAMISA DE 34	RENAULT	SYMBOL	5,543,00	8,318,00	97	806,894,00
999	LUBRICANTE SHELL	CHEVROLET	CORSA	200,00	300,00	97	29,100,00
1001	CAMISAS	CHEVROLET	SILVERADO	5,000,00	10,000,00	98	900,000,00
1002	BANDAS	MAZDA	MAZDA 3	2,000,00	2,400,00	95	228,000,00
1003	INTERIORES	MAZDA	MAZDA 323	800,00	1,200,00	97	114,400,00
2001	CAMISAS	CHEVROLET	CHEVYENNE	1,000,00	1,000,00	98	194,000,00
2002	BANDAS	CHEVROLET	CHEVYENNE	2,000,00	3,000,00	98	294,000,00
3333	SELLOS	MAZDA	MAZDA 4	4,750,00	1,997,00	98	581,824,00
5001	CAMISAS	NISSAN	CENTRA B6	70,000,00	105,000,00	98	10,290,000,00
5002	BANDAS	NISSAN	CENTRA B6	400,00	400,00	100	40,000,00
5003	INTERIORES	NISSAN	CENTRA B6	1,800,00	3,400,00	98	352,800,00
7171	SELLOS REINCRUZADOS	NISSAN	PATROL	5,000,00	4,250,00	97	406,250,00
8822	TORNILLOS	MAZDA	MAZDA 323	500,00	400,00	97	38,200,00
1268							14,599,422,00

Figura 15.4 Figura de Reporte de Existencia de productos

REPORTE DE PRODUCTOS POR DEBAJO DEL STOCK MINIMO - INVENTARIO

El reporte de productos por debajo del Stock Mínimo es el resultado de la consulta del formulario inventario en el cual se dan como parámetros los criterios de la consulta y el resultado es el siguiente reporte.

Reporte de Productos por Debajo del Stock Mínimo				
				Fecha de Impresión: 15/03/2007
Item	Codigo	Nombre Producto	Existencia	Stock Mínimo
1	1001	CAMISAS	8	10
2	1002	BANDAS	5	10
3	2002	BANDAS	8	10

Figura 15.5 Figura de Reporte de productos por debajo del Stock Mínimo

A continuación se muestra la figura que representa la barra de funciones del reporte que se halla ejecutado, entre esas funciones se encuentran las siguientes: Imprimir, pasar de una pagina a otra, zoom, cerrar, etc.



Figura 15.6 Figura de Barra de Opciones para los reportes ejecutados

FORMULARIO DE REPORTES DINAMICOS

Como su nombre lo indica este formulario permite al usuario final realizar una consulta dinámica con respecto a las tablas de la base de datos, deben escoger las tablas y los campos que van incluido en la elaboración de la consulta y si lo requiere el usuario una cláusula o condición la cual va a depurar la información con el dato exacto que se requiera.

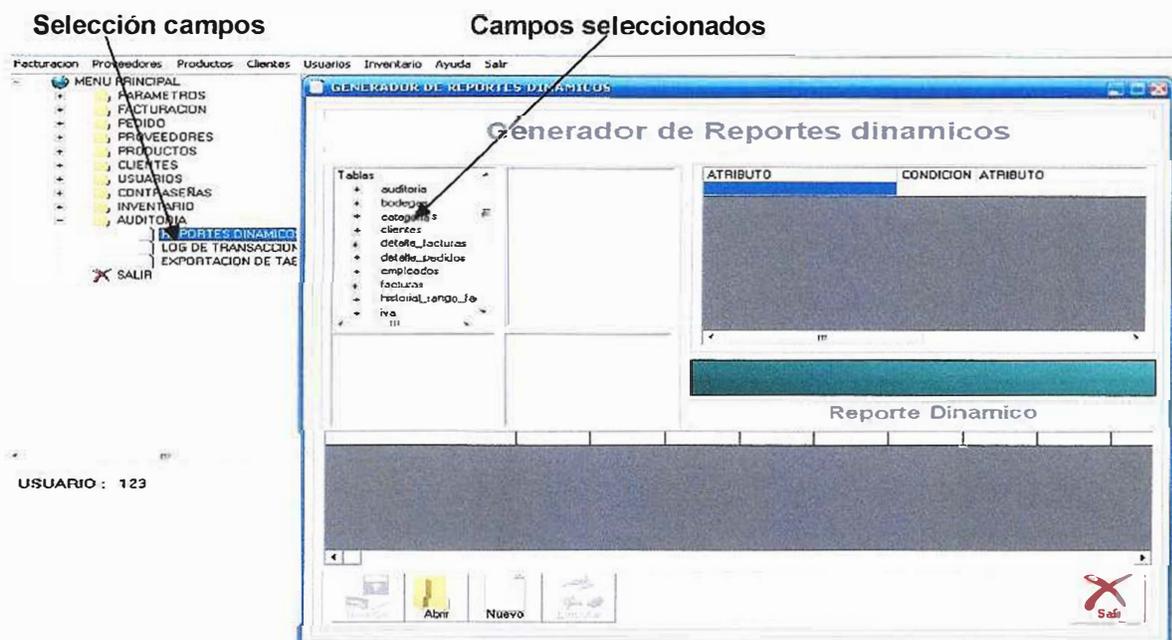


Figura 15.7 Figura de Formulario de reportes Dinámicos

Generador de Reportes dinamicos

The screenshot shows the 'Generador de Reportes dinamicos' interface. On the left, there are two tree views: 'facturas' and 'clientes'. The 'facturas' tree is expanded to show attributes like 'fa_codfac', 'fa_fecha', 'fa_subtot', 'fa_iva', 'fa_total', 'fa_descut', 'fa_estado', 'cl_idclient', and 'us_nombr'. The 'clientes' tree is also expanded to show attributes like 'clientes.cl_idcliente', 'clientes.cl_nomcliente', 'clientes.cl_estado', 'facturas.fa_codfactura', 'facturas.fa_fecha', 'facturas.fa_total', and 'facturas.us_nombre'. Below these trees, the 'Nombre de la Tabla' is set to 'FACTURAS' and its attributes are listed. On the right, a table with columns 'ATRIBUTO', 'CONDICION', and 'ATRIBUTO' is shown. The first row contains 'clientes.cl_idcliente', '=', and '72278955'. Below this table is a large empty area labeled 'Reporte Dinamico'.

Detalle de tablas seleccionadas

Tablas seleccionadas

El resultado de la consulta dinámica será mostrado en una grilla (VsFlexGrid) con sus respectivos datos.

Resultado del reporte dinámico

The screenshot shows the 'Resultado del reporte dinámico' interface. It features a table with the following data:

clientes.cl_idcliente	clientes.cl_nomcliente	clientes.cl_estado	facturas.fa_codfactura	facturas.fa_fecha	facturas.f:
72278955	GEOVANI VASQUEZ RAD	1	10000	26/06/2007	
72278955	GEOVANI VASQUEZ RAD	1	10001	26/06/2007	
72278955	GEOVANI VASQUEZ RAD	1	10002	26/06/2007	
72278955	GEOVANI VASQUEZ RAD	1	10003	26/06/2007	
72278955	GEOVANI VASQUEZ RAD	1	10004	26/06/2007	
72278955	GEOVANI VASQUEZ RAD	1	10005	26/06/2007	
72278955	GEOVANI VASQUEZ RAD	1	10006	15/08/2007	

Below the table is a toolbar with icons for home, back, forward, and search, and a red 'X' icon.

FORMULARIO DE AUDITORIA

El formulario auditoria permite verificar los movimientos que se han generado en la base de datos por medio de la aplicación al momento de cargar el formulario se mostraran todos los movimientos.

En manos del usuario queda la opción de depurar la información por fechas de movimientos, usuario que haya realizado las transacciones y el tipo de movimiento generado.

Modular Auditoria

Log de Transacciones

Item	Operacion	Tabla	Fecha	Hora	Usuario
1	INSERCIÓN	CLIENTE	26/08/2007	17/08/2007	123
2	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
3	INSERCIÓN	FACTURA	28/08/2007	17/08/2007	123
4	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
5	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
6	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
7	INSERCIÓN	FACTURA	28/08/2007	17/08/2007	123
8	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
9	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
10	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
11	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
12	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
13	INSERCIÓN	FACTURA	26/08/2007	17/08/2007	123
14	INSERCIÓN	BODEGAS	15/08/2007	17/08/2007	123
15	MODIFICACION	PRODUCTOS	15/08/2007	17/08/2007	123
16	INSERCIÓN	FACTURA	14/08/2007	17/08/2007	123

Figura 16 Figura de Formulario de Auditoria

FORMULARIO DE EXPORTACION E IMPORTACION DE DATOS

Este formulario permite exportar e importar las tablas de la base de datos con sus respectivos registros almacenados.

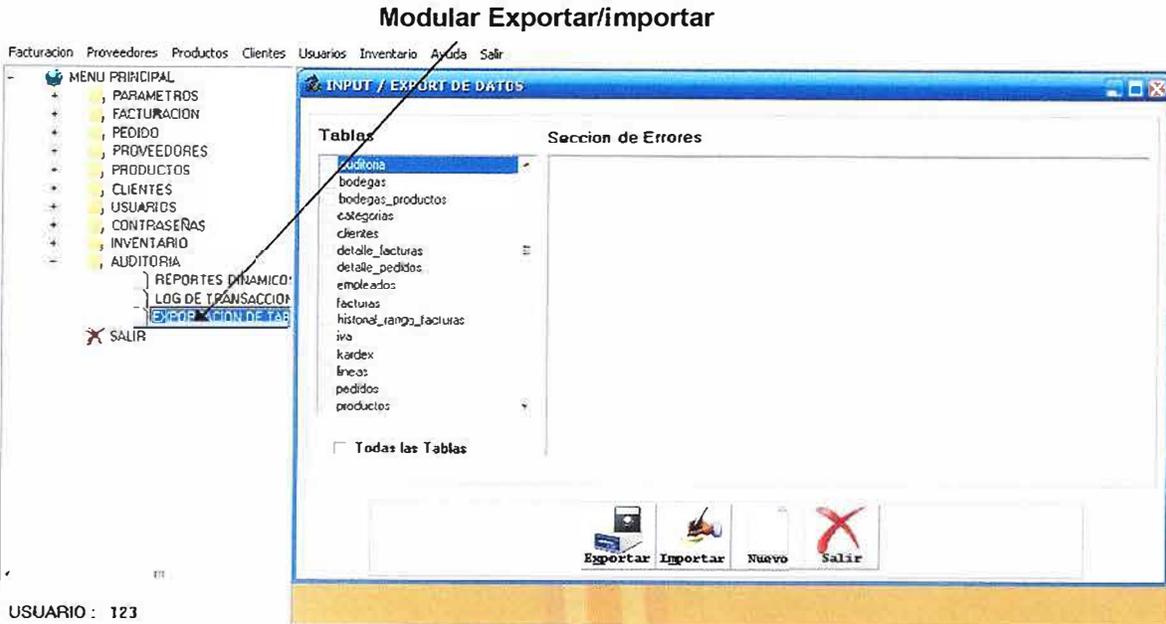


Figura 17 Figura de Formulario de Exportar/Importar

Este formulario permite realizar un back-up de la base de datos, como una medida preventiva ante posible perdida de la información ya sea por casos fortuitos o catástrofes que afecten los equipos o las instalaciones.

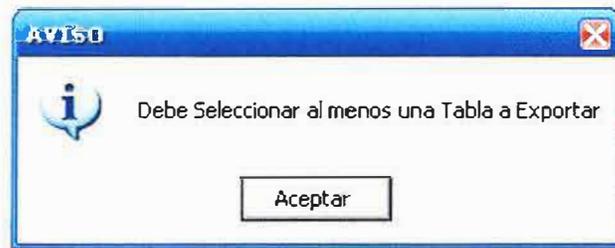
A su vez, se puede también cargar un back-up que hayamos realizado anteriormente con el botón importar, el cual nos muestra un menú examinar para así poder cargar el archivo que contiene el back-up cuya extensión se encuentra en .txt.

Tablas Seleccionadas para Exportar



Con este cuadro que aparece en el formulario se escogen todas las tablas para hacer la exportación o en el caso que sea pertinente tabla por tabla o una sola tabla.

Mensaje de confirmación cuando no se ha seleccionado por lo menos una tabla a exportar.



Mensaje de confirmación

Al momento que se realice la exportación de los datos y no se presente ningún tipo de error durante el proceso al final se mostrara un mensaje de confirmación el cual indicara que el proceso finalizo satisfactoriamente.

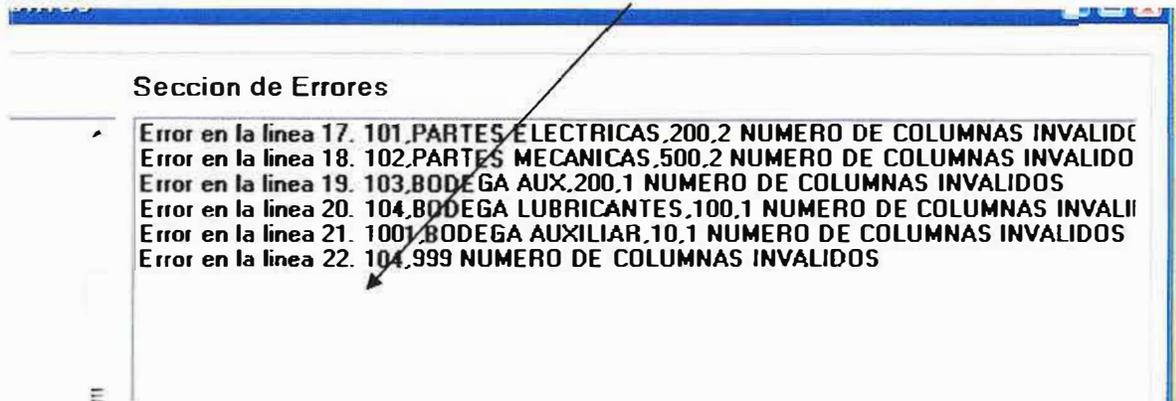


Mensaje de confirmación

Importación de Tablas. Este formulario permite cargar las tablas que se van a importar



Sección de errores



En esta sección se visualizan los posibles errores por los cuales la transacción de importación no se pudo realizar correctamente detallando por línea cada uno de los errores.

Mensaje de confirmación para Salir de la Aplicación



Esta confirmación se muestra en el momento que se disponga salir de la aplicación.

NOTAS IMPORTANTES

- En los campos donde se requieran valores numéricos, solo se podrá digitar valores entre 0-1-2-3-4-5-6-7-8-9, si intenta digitar letras, el sistema no mostrara nada en la caja de texto.
 - Una caja de texto Letra-Numérico es una caja de texto en la que podemos insertar todo tipo de carácter de la a-z , A-Z, 0-9.
 - Los botones **Guardar y Editar** solo se activaran de acuerdo a los eventos ya explicados en este manual en cada uno de los formularios La funcionalidad para los botones **Buscar, Nuevo y salir**, es la misma para todos los formularios que pudiesen tener dichos botones.
 - En todos los formularios donde se maneje el campo Código o Id, si se hace una consulta, al cargar estos datos al formulario aparecerán como Solo Lectura.
 - Al hacer click en Cambio de Sesión, el aplicativo se cerrara automáticamente dando paso al formulario de inicio de sesión. Asegúrese de haber guardado los cambios
-

2.3.1 Instalación de MySQL Servers and Clients 5.0.1. - Alpha.

Busque en el CD el directorio MySQL 5.0.1.\mysql-5.0.1-alpha-snapshot-win, dentro de este subdirectorio hay un archivo llamado SETUP.exe. Ejecútelo y siga los pasos de la instalación.

Después de la instalación de la Base de Datos reinicie el equipo para que Windows inicie el servicio de MySQL y se puedan generar las conexiones a la Base de Datos.

El aplicativo MINFA se comunica con la Base de Datos mediante un intermediario conocido como ODBC, el cual realiza la conexión con MySQL Servers and clients 5.0.1 – alpha, con un origen de base de datos de MySQL ODBC 3.51 Driver DSN como muestra la Figura No 1.

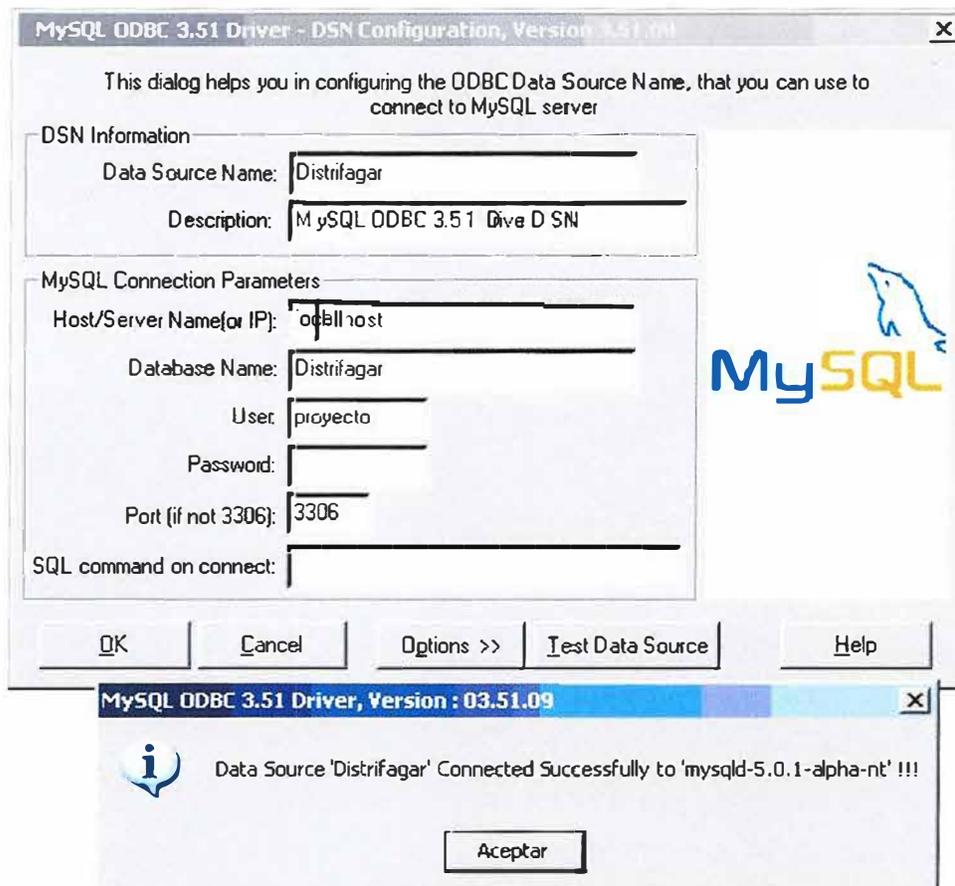


Figura No. 1: Comunicación entre aplicativo MINFA y Base de Datos MySQL Servers and Clients 5.0.1. – alpha, a través del ODBC.

Posteriormente a la instalación de la Base de Datos se instala el controlador ODBC (es el que permite la conexión del aplicativo con la Base de Datos), el cual se encuentra en el CD, en el directorio MySQL 5.0.1., dentro de este subdirectorio hay un archivo llamado MyODBC-

standard-3.51.9-win.exe. Ejecútelo y siga los pasos de Instalación.

Y por ultimo para un mejor manejo de la Base de Datos con una interface grafica se instala la consola de Base de Datos SQLyog51, la cual se encuentra en el CD, en el directorio MySQL 5.0.1., dentro de este subdirectorio hay un archivo llamado SQLyog512.exe. Ejecútelo y siga los pasos de instalación. (Opcional)

Esta última le facilita y agiliza la respectiva creación y modificación de la base de datos pues viene en ambiente Windows, ver Figura No 2.

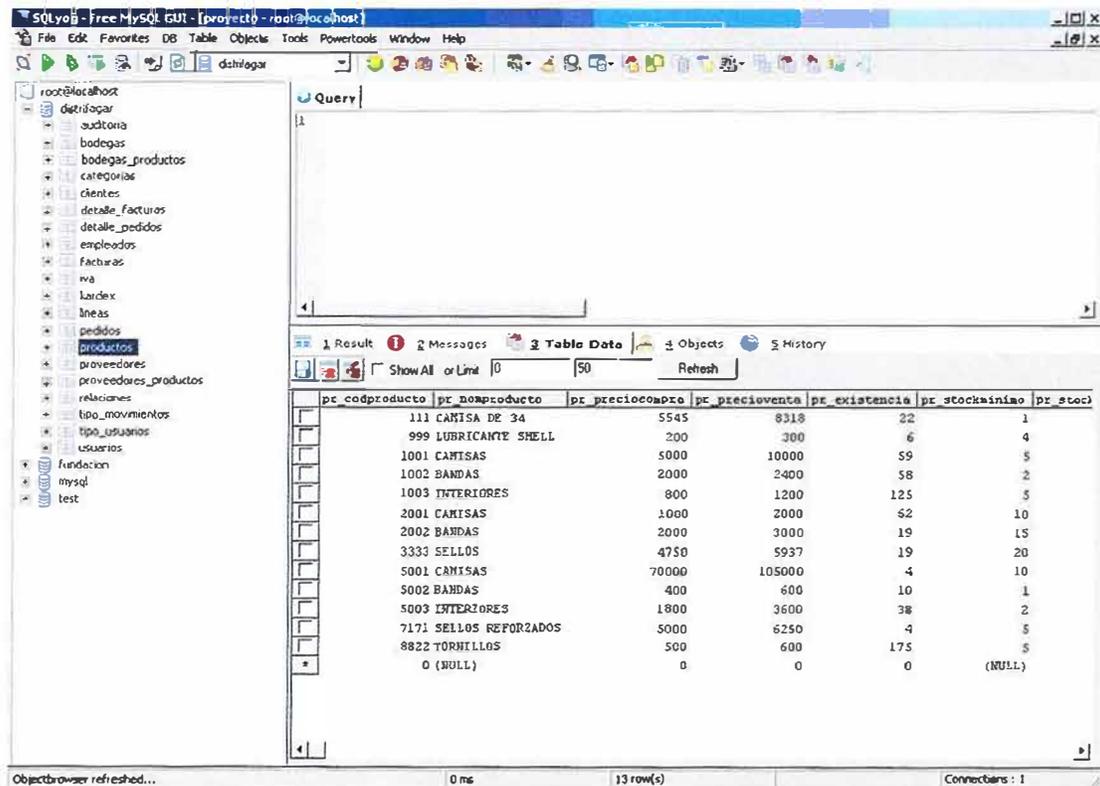


Figura No. 2: Consola grafica de MySQL para mejor manejo de las operaciones en la base de datos.

El instalador MINFA coloca el aplicativo en el equipo cliente; pero en sistemas de red, es frecuente que se prefiera mantener el ejecutable en un servidor de aplicaciones debido a que la actualización del aplicativo sólo se realiza una vez sobre dicho servidor, y no sobre cada equipo cliente. Para esta estrategia administrativa de aplicativos dirigidos a un servidor,