

Universidad Nacional Autónoma de Nicaragua

UNAN-LEON



Facultad de Ciencias.

Departamento de Computación.

Tesis para optar al Título de Ingeniero en Sistema de Información.

Tema: ***“Control Automatizado de Inventario Cliente-Servidor”***
(CAICS)

Autores:

Br. Donald Ismael Chévez Guevara.

Br. Henry José Ibarra Vega.

Br. José Adrián Lindo Guzmán.

Tutor:

M.Sc. Ernesto Espinoza Montenegro.

León, Nicaragua 2005



AGRADECIMIENTO

Agradecemos a DIOS por permitirnos la existencia en esta tierra y vivir la vida que nos ha encomendado, y que a pesar de todo, siempre esta dándonos otra oportunidad para vivir en paz y armonía al igual que la posibilidad de vivir una vida más honorable.

A nuestros padres, que tuvieron que limitarse muchas veces, y muchas otras por sobre esforzarse para que nosotros lográramos lo que ahora le estamos agradeciendo con todo nuestro corazón, ya que sus sueños y nuestra meta se están cumpliendo después de haber concluido un largo tiempo de esfuerzo, dedicación y empeño.

Gracias padres, por todo ese apoyo incondicional que hizo posible este resultado, por ser padres más que responsables, por legarnos lo que no nos pueden quitar, y por gozar y sufrir con nuestros logros y fracasos.



DEDICATORIA

Primeramente al **Dios** supremo, por iluminar mi camino hasta alcanzar el éxito, por haberme dado la oportunidad de realizar uno de los sueños mas grandes de mi vida, por estar siempre a mi lado en las buenas o en las malas, por traerme a este mundo y ser un ejemplo para todo aquel que desea ser un profesional y por haberme dado paciencia en todos los problemas y conflictos que fui enfrentando día a día. Gracias Dios, por ser mi mejor amigo, y más que amigo, por abrigarme en todos estos años de mi carrera.

A mi madre que tanto amo con todas las fuerzas de mi corazón como es **Amelia Guevara**, por haberme formado, por ser la persona a quien le debo mas que mi propia vida, y por ser ella la que me ha ayudado en los momentos mas difíciles. Le doy gracias ya que sin ella no hubiese sido posible culminar mi carrera universitaria.

A mi padre **Borge Chévez**, por ser mi amigo, por su apoyo, comprensión, su confianza depositada en mi, y por haberme ayudado a realizar este sueño que tuve desde niño.

A mi esposa **Guissel Cortez**, por su amor, por ser ella la que me ha animado siempre que lo he necesitado, por estar a mi lado en los peores momentos de mi vida. Gracias mi vida, eres lo mejor que mi Dios me ha dado.

A mis **hermanos y demás familiares** que me han brindado todo su amor y su apoyo. Gracias a todos ellos.

A mi **universidad** por ser yo fruto de ella, por haber formado en mi un profesional, por ser ella la que me enorgullece, por haberme brindado todos los conocimientos adquiridos y por darme la oportunidad de ser alguien que forme parte de la vida profesional.

Br. Donald Ismael Chévez Guevara.



DEDICATORIA

En primer lugar le dedico este trabajo a **Dios**, ya que es el que me ha mantenido con vida hasta estos momentos y me ha permitido lograr el deseo de concluir mi carrera, iluminándome durante todo el tiempo de vida universitaria.

A mi madre **Onésima Olimpia Velásquez** que me brindó su apoyo en todo momento desde el principio hasta el final. Tengo el honor de tener una madre que me proporcionado lo necesario con mucho esfuerzo y de todo corazón.

A mis hermanos que me han ayudado económicamente así como también brindado apoyo emocional.

José Adrián Lindo Guzmán.



DEDICATORIA

Dedico este trabajo primeramente a **Dios**, luz de mi conciencia y faro en mi camino, ya que ha sido más que un semáforo en mi niñez y adolescencia, gracias Dios por darme una madre tan humilde y luchadora por vendecirla y darle las fuerzas para sacar adelante a la familia y sobre todo regalarme la dicha de la preparación profesional, gracias por permitirme estar escalando un peldaño en mi vida el cual considero que me servirá de mucho para integrarme en la sociedad.

También dedico muy especialmente este trabajo a lo mejor que tengo y de lo que presumo, a mi madre que amo con cada parte de mi ser como es **Felicita Vega** por darme sus mayores esfuerzos, por animarme a continuar y a quien le reconozco lo que soy; gracias madre por estar siempre apoyándome en los momentos difíciles y de lucha, gracias por darme más de lo que debías, por elegir que la educación era mi mejor opción cuando aun no sabia su importancia ni tenia el deseo de hacerlo, por todo eso te agradezco.

A mis hermanos **Patricio, Marisela, Darwin, Erica** por su apoyo y creer en mi y especialmente a mi hermano menor **Lesther** quien me ha puesto el hombro como los grandes. Gracias a todos ellos.

Br. Henry José Ibarra Vega.



INDICE DE CONTENIDO.

FASE EXPLORATIVA

I. INTRODUCCIÓN	1
II. ANTECEDENTE	3
III. JUSTIFICACIÓN	4
IV. OBJETIVOS	5
V. METODOLOGÍA	6
5.1. DISEÑO METODOLÓGICO (Material y Método).	6
5.2. RECURSOS DISPONIBLES Y NECESARIOS	11
5.3. CREACIÓN DE LA BASE DE DATOS EN SQL-SERVER 7.0	12
VI. MARCO TEÓRICO	17

FASE DE ANÁLISIS

VII. ANÁLISIS	23
7.1. ANÁLISIS DEL SISTEMA	23
7.2. ESPECIFICACIÓN DE REQUISITOS SOFTWARE	24
7.3. DIAGRAMA DE ENTIDAD RELACIÓN	71
7.3.1. RELACIONES DE LAS TABLAS DE LA BBDD'S	72
7.4. DIAGRAMA DE FLUJO DE DATOS	73
7.5. DICCIONARIO DE DATOS	81

FASE DE DISEÑO

VIII. DISEÑO	84
8.1. DISEÑO DE DATOS	84
8.2. DISEÑO PROCEDIMENTAL	89
8.3. DISEÑO ARQUITECTÓNICO	90
8.4. DISEÑO DE INTERFAZ	91
IX. CONCLUSIONES	120
X. RECOMENDACIONES	121
XI. ANEXOS	122
11.1. SCRIPT DE LA CREACIÓN DE LAS TABLAS DE LA BASE DE DATOS	122
11.2. ANEXO DE CÓDIGOS DE LAS INTERFACES	128
11.3. REPORTES	212
XII. REFERENCIAS BIBLIOGRÁFICAS	221



INDICE DE TABLAS E ILUSTRACIONES.

TABLAS

Tabla 1: Empresa.....	84
Tabla 2: Cliente.....	84
Tabla 3: Proveedor.....	84
Tabla 4: Bodega.....	84
Tabla 5: Producto.....	85
Tabla 6: Entrada_Almacen.....	85
Tabla 7: Detalle_Entrada.....	86
Tabla 8: Lote Producto.....	86
Tabla 9: Salida_Almacen.....	86
Tabla 10: Detalle_Salida.....	86
Tabla 11: Ajustes_Almacen.....	87
Tabla 12: Detalle_Ajuste.....	87
Tabla 13: Factura.....	87
Tabla 14: Detalle_Factura.....	87
Tabla 15: Cotización.....	88
Tabla 16: Detalle_Cotizacion.....	88
Tabla 17: Contraseña.....	88
Tabla 18: FechaProceso.....	88

ILUSTRACIONES

Ilustración 1: Datos de la Empresa.....	91
Ilustración 2: Creación de Nueva Bodega.....	91
Ilustración 3: Creación de los Usuarios.....	92
Ilustración 4: Carga del Sistema.....	92
Ilustración 5: Contraseña de acceso al sistema.....	93
Ilustración 6: Bodega de trabajo en el sistema.....	93
Ilustración 7: Establecer fecha de proceso.....	93
Ilustración 8: Ventana principal del sistema.....	94
Ilustración 9: Respaldo de la base de datos.....	95
Ilustración 10: Restauración de la base de datos.....	95
Ilustración 11: Entrada de productos al almacén.....	96
Ilustración 12: Salida de productos del almacén.....	97
Ilustración 13: Ajuste de productos en almacén.....	98
Ilustración 14: Factura del cliente.....	99
Ilustración 15: Cotización del cliente.....	100
Ilustración 16: Cierre del mes en proceso.....	101
Ilustración 17: Registrar nuevo producto.....	101
Ilustración 18: Registrar un nuevo cliente.....	102
Ilustración 19: Registrar un nuevo proveedor.....	102
Ilustración 20: Reporte de entrada.....	103
Ilustración 21: Reporte de salida.....	103
Ilustración 22: Reporte de ajuste.....	104
Ilustración 23: Kardex por producto.....	105
Ilustración 24: Reporte de factura.....	106
Ilustración 25: Búsqueda de producto.....	107
Ilustración 26: Muestra del producto.....	107



Ilustración 27: Búsqueda de cliente.....	108
Ilustración 28: Muestra del cliente.....	108
Ilustración 29: Búsqueda de proveedor.....	109
Ilustración 30: Muestra del proveedor.	109
Ilustración 31: Búsqueda de factura.	110
Ilustración 32: Muestra de la factura.....	110
Ilustración 33: Contraseña de administrador.	111
Ilustración 34: Búsqueda de cotización.....	111
Ilustración 35: Muestra de la cotización.	112
Ilustración 36: Lista de productos.....	113
Ilustración 37: Modificar producto.	113
Ilustración 38: Modificar cliente.....	114
Ilustración 39: Lista de clientes.	114
Ilustración 40: Modificar proveedor.	115
Ilustración 41: Lista de proveedores.....	115
Ilustración 42: Modificar fecha de proceso.	116
Ilustración 43: Modificar datos de la empresa.	116
Ilustración 44: Modificar contraseña.....	117
Ilustración 45: Acerca de.....	117
Ilustración 46: Manual de usuario.	118
Ilustración 47: Productos cercanos a caducar.....	119
Ilustración 48: Productos vencidos.....	119



I. INTRODUCCIÓN

La dificultad de cuantificar el tiempo, coste y esfuerzo necesario para el desarrollo de un producto software, y la falta de calidad del software producido, son la causa de la aparición de la Ingeniería del software como una disciplina científica.

Inicialmente la programación de las computadoras era un arte que no disponía de métodos sistemáticos en los que se pueda basar para la realización de productos software. Se realizaban sin ninguna planificación. Posteriormente surge la evolución y perspectivas de la Ingeniería del Software. Desde mediados de los 60 hasta finales de los 70 se caracterizó por el establecimiento del software como un producto que se desarrollaba para una distribución general. En esta época nació lo que se conoce como el mantenimiento del software que surge al incrementar los requisitos de los usuarios y se hace necesaria la modificación del software o implementación del proceso de **reingeniería**. El esfuerzo requerido para este mantenimiento era en la mayoría de los casos tan elevado que se hacía imposible su mantenimiento. A continuación, surge una etapa que se caracteriza por la aparición de una serie de técnicas como la programación estructurada y las metodologías de diseño, las cuales solucionan los problemas anteriores. A finales de esta etapa aparecen las herramientas CASE, como podemos imaginar eran muy rudimentarias. La siguiente etapa da inicio en la década de los 90`s, la cual se caracteriza por la aparición de la programación orientada a objetos. Esta ha producido un impacto en el desarrollo de grandes productos software.

En el presente el producto software se ha vuelto muy importante y ha adquirido grandes demandas, pero a pesar de esto existen muchas necesidades para poder controlar de una forma eficiente, rápida y segura el flujo de información de una compañía o negocio, así, evitando la pérdida o probabilidad de que una empresa pueda fracasar. Para impedir este tipo de exposiciones nosotros como estudiantes de la carrera de ingeniero. en sistemas de información nos hemos propuesto desarrollar una estrategia que pueda beneficiar de una forma positiva a una empresa a través de un sistema de inventario automatizado el cual formara gran parte para los sistemas de contabilidad de mercancías ya que este es el corazón de todo negocio.

Los inventarios representan uno de los principales recursos de que dispone una entidad comercial o industrial. Es importante tener un adecuado manejo de inventarios, pues de ellos dependen las actividades primarias de todo negocio; es decir, las operaciones de entrada-salida que concluirán en utilidades y proporcionaran flujos de efectivo. Las empresas dedicadas a la



compra y venta de mercancías, por ser ésta su principal función y la que dará origen a todas las restantes operaciones, necesitarán de una constante información resumida y analizada sobre sus inventarios.

Las compañías exitosas tienen gran cuidado de proteger sus inventarios para poder aumentar sus ingresos, y así analizar y tomar decisiones correctas que la lleven al éxito.

En base a las necesidades descritas anteriormente, aplicaremos el proceso de reingeniería a un sistema automatizado ya existente que corre bajo la plataforma MS-DOS que tiene entre otras características la de ser monousuario, el cual fue desarrollado en el lenguaje de programación clipper. La función principal de dicho sistema es llevar control de las entradas y salidas de mercancías (Inventario), para tal efecto es necesario establecer los requerimientos que se quieren que el sistema cumpla. Dicho sistema se diseñara con el objetivo de que pequeñas y medianas empresas lo puedan utilizar para el control de su negocio.

Nuestro software se desarrollara con interfaces gráficas diseñadas en el lenguaje de programación **visual basic**. Este software nos permitirá la manipulación y control de la información que se maneja en los inventarios de una manera más sencilla y rápida, y que éste pueda ser capaz de almacenar información (base de datos) en una estación servidora la que será accedida mediante técnicas de transacciones desde otras estaciones clientes.



II. ANTECEDENTE

En la vida cotidiana el software ha sido orientado de forma específica sin tomar en cuenta la solución a los problemas de una forma más generalizada, dicho de otra forma, este no puede ser usado por varios usuarios finales o empresas en las que su forma de operar sea de una manera diferente.

Muchas empresas no cuentan con un sistema de inventario automatizado para llevar un control de todas sus operaciones internas, y de esta forma les permita un acceso rápido a la información que se quiere procesar, además, les facilite tomar decisiones concretas para beneficiarse así misma obteniendo mayores ingresos. En la actualidad llevar a cabo este proceso es muy difícil y lento ya que es realizado manualmente.



III. JUSTIFICACIÓN

Nuestra justificación se basa en los siguientes aspectos:

- La situación real de todas las transacciones y movimientos de mercancía en las empresas las cuales van en crecimiento.
- Los fracasos de las instituciones por el mal manejo de sus recursos al igual que sus bienes y no tener información exacta en el momento que la necesita.

Es importante saber que los inventarios son la principal fuente de ingreso de una compañía, pero como hemos mencionado antes, el manejo manual de éste es demasiado complicado y riesgoso, factores que permiten posibles alteraciones de la información que se maneja en una empresa, llevándola de esta manera a una pérdida o fracaso de la misma. Es cierto que existen aplicaciones de contabilidad que contienen diversos campos, pero estos manipulan un alto volumen de información, lo que provoca un manejo complicado de la misma; debido a esta problemática desarrollaremos un sistema únicamente del control y manejo de inventarios con un funcionamiento independiente a los otros sistemas de contabilidad, el cual le permitirá a la empresa proteger su información y aumentar sus ingresos con el propósito de llevarla al éxito.



IV. OBJETIVOS

General:

1 Desarrollar un sistema cliente-servidor que permita el manejo y control de inventario mediante una interfaz de usuario sencilla aplicando técnicas de reingeniería y que sea capaz de ser utilizada por cualquier negocio o pequeña empresa.

Específicos:

- 1 Aplicando el proceso de reingeniería a un sistema de inventario existente que corre bajo la plataforma MS-DOS, pretendemos realizar un cambio al mismo, con el fin que pueda ser ejecutado en un entorno visual bajo la plataforma windows.
- 2 Diseñar un sistema para múltiples usuarios (Cliente-Servidor) el cual permita el acceso a una base de datos por mas de una estación cliente, con el fin de interactuar con la información contenida en la base de datos.
- 3 Ofrecer a las medianas empresas la facilidad de tomar decisiones concretas, con información precisa y actualizada en pos de su desarrollo, crecimiento y competitividad.
- 4 Afianzar las bases de conocimientos adquiridos durante nuestro desarrollo y formación profesional.
- 5 Llevar un mejor control de las entradas y salidas de inventarios que son generadas en las operaciones de las entidades comerciales.



V. METODOLOGÍA

5.1. DISEÑO METODOLÓGICO (Material y Método).

Para el desarrollo de nuestro proyecto utilizaremos un conjunto integrado de técnicas y métodos que nos permitan abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida del diseño de nuestro sistema de información.

Haciendo uso de todos estos mecanismos, procedimientos y recursos, nuestra metodología se implementara en etapas o pasos precisos, tanto en el proceso de desarrollo en general (ciclo de vida) como cada una de las fases generales en las que se divide ésta. Estas fases estarán implicadas en el desarrollo de nuestro software, desde su concepción inicial, pasando por su desarrollo, implantación, funcionamiento y mantenimiento, hasta que el sistema deja de utilizarse o se transforma en otro utilizando una reingeniería.

La utilización de métodos facilitará la comunicación entre el sistema que queremos crear y los usuarios para los que se realiza el sistema, ya que se generarán una serie de interfaces gráficas entre los usuarios y el sistema.

Nuestro sistema de información tendrá un conjunto de elementos que van a interactuar entre sí con el fin de apoyar las actividades de una empresa o negocio.

Este sistema logra la automatización de procesos operativos dentro de una organización, son llamados frecuentemente sistemas transaccionales, ya que su función primordial consiste en procesar transacciones tales como entradas, salidas, etc. De acuerdo con su uso u objetivos que cumplen, uno de estos sistemas es el de los sistemas estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología computacional.

Para llevar a cabo el desarrollo de un sistema de información estratégico basado en tecnología computacional existen tres métodos o enfoques, uno de los cuales es el que implementaremos para el diseño de nuestro software:

Método del ciclo de vida clásico para el desarrollo de sistema.

Método del desarrollo del análisis estructurado.

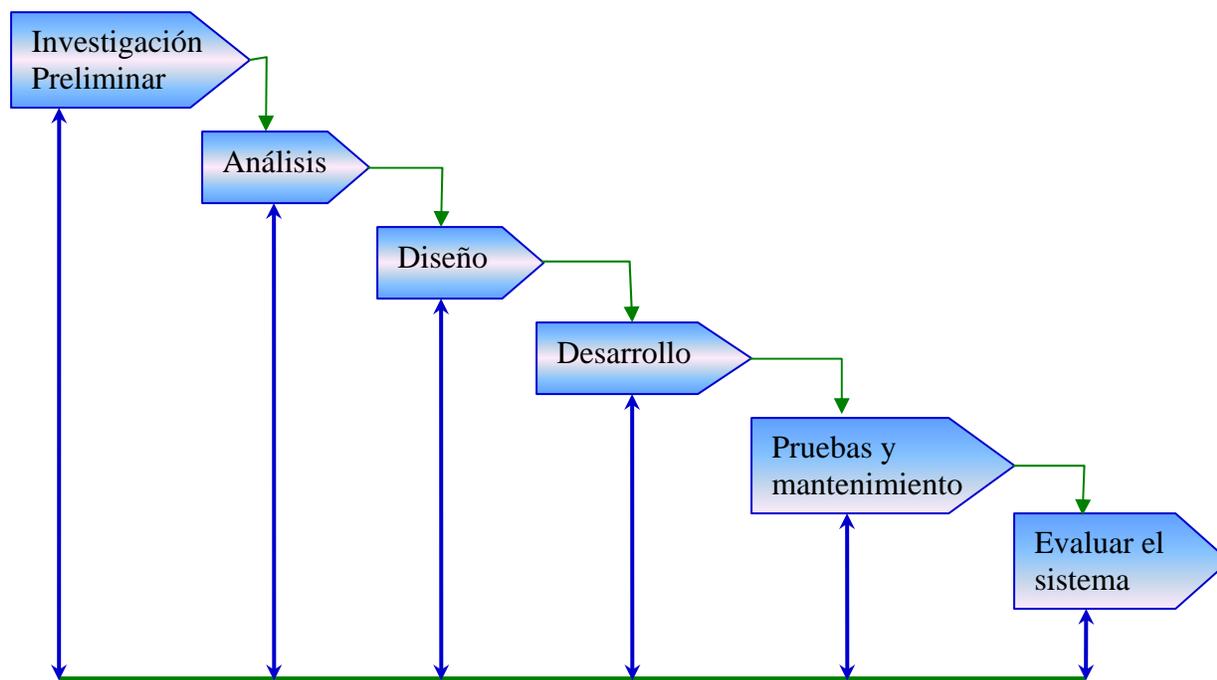
Método del prototipo de sistemas.



Método o estrategia utilizada para el desarrollo de nuestro Sistema de Información.

Para desarrollar e implementar nuestro sistema de información haremos uso del método del ciclo de vida clásico o en cascada ya que este establece una serie de fases, las cuales al finalizar se obtienen una serie de productos (documentos, diagramas, programas) que permite evaluar lo realizado hasta ese momento y continuar con la fase siguiente o modificar algunos aspectos de las fases anteriores.

Implementando este Modelo (ciclo de vida clásico o en cascada), el cual es adecuado para el desarrollo de un sistema, garantizaremos que nuestro software sea de calidad, eficiente, fiable y no erróneo, y que este sea de mucha utilidad para la mejora de la efectividad financiera de una empresa y su competitividad con las demás.





Esta representación gráfica es en la cual se basará el desarrollo de nuestro software. Conforma un conjunto de fases las cuales nunca se llevan como un paso aparte, estas se podrán detallar a continuación:

➤ **INVESTIGACIÓN PRELIMINAR.**

Esta fase comprende el momento en que empezamos a investigar sobre nuestro tema de monografía para desarrollar un sistema confiable con información y proyección de futuro.

Esta es la primera actividad para el desarrollo de nuestro sistema, es de existencia opcional. En esta fase nuestra meta es adecuar los objetivos estratégicos de los usuarios finales y la información necesaria para soportar dichos objetivos. En esta etapa podemos determinar si al sistema se le puede aplicar el proceso de reingeniería y si puede ser factible, incluyendo algunas especificaciones básicas acerca de coste y tiempo necesarios para construir nuestro sistema, así como los beneficios que se obtendrán del mismo.

➤ **ANÁLISIS DE LOS REQUERIMIENTOS DEL SISTEMA.**

Luego de un previo análisis sobre el tema a desarrollar, nos dedicaremos a identificar los requisitos del sistema, los cuales determinaran los elementos y recursos que este necesita para su correcto funcionamiento. En esta fase también estudiaremos las necesidades de información que debe satisfacer el sistema a desarrollar, elaborando una serie de especificaciones formales que describan la funcionalidad del mismo y que permitan abordar con garantía la siguiente fase.

A medida que se van recopilando todos los detalles que debe tener nuestro sistema, se profundiza sobre los requerimientos del mismo, identificando las características que debe tener éste. Todo esto se lleva a cabo haciendo uso de herramientas como el diagrama de flujo de datos (DFD), en el cual representaremos el flujo de la información o los elementos del software y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. Otra herramienta que utilizaremos es el diccionario de datos (DD) ya que nos servirá para integrar una lista de todos los elementos incluido en el conjunto de los diagramas de flujo de datos que describe nuestro sistema, es decir, almacenara los detalles y las descripciones del flujo de datos, el almacenamiento de datos y los procesos que en este se llevaran a cabo.

Además de las herramientas antes mencionadas, haremos uso del modelo conceptual de datos (MCD) en el cual representaremos de forma abstracta los datos que utilizará nuestro sistema.



Este modelo lo representaremos gráficamente mediante el modelo de entidad relación (ER), en el que reflejaremos los elementos descritos en el (MCD).

Todas estas herramientas o elementos esenciales serán la base de nuestro sistema para determinar los requerimientos del mismo.

➤ **DISEÑO DEL SISTEMA RECOMENDADO.**

Una vez obtenidos los requerimientos del sistema nos dispondremos a diseñar el sistema. Cabe mencionar que la investigación preliminar es vital ya que no se debe diseñar un sistema que no tenga todos los elementos que permitirán su desarrollo, ya que al momento de su uso en el futuro puede generar errores.

Especificaremos los aspectos físicos del sistema, considerando las características tecnológicas del entorno específico en el que se implantará, que constituirá el punto de partida para la construcción de nuestro sistema. Equivale a la creación de una jerarquía apropiada de módulos de programas y de interfaces para implantar la especificación creada en el análisis de los requerimientos del sistema. Además, transformaremos el modelo de datos de entidad relación en un diseño de base de datos.

➤ **DESARROLLO Y DOCUMENTACIÓN DEL SOFTWARE.**

En esta parte mostraremos la manera de como desarrollamos este software y la forma de usarlo con facilidad. Luego de haber hecho todas las especificaciones del diseño de nuestro software, llevaremos a cabo la obtención del sistema completamente construido y probado, listo para ser implantado. También durante esta fase desarrollaremos el conjunto de procedimientos y llevaremos a cabo la información necesaria que permitirá al personal del área de usuario final la utilización óptima del sistema. Al final de esta fase obtendremos el software correspondiente y los siguientes documentos: "documentación técnica de programación", "manual de usuario", "documento de pruebas del sistema", etc.

➤ **PRUEBAS Y MANTENIMIENTO DEL SISTEMA.**

Realizaremos pruebas con datos imaginarios y algunas sugerencias para que el sistema funcione sin problemas durante su uso, es decir, daremos la puesta en marcha del sistema construido y conseguir su adaptación final por parte de los usuarios del mismo, para lo cual trataremos de hacerle ver a éstos, mediante demostraciones formales (pruebas de aceptación)



que el sistema cumpla todos los objetivos y requisitos para lo que fue diseñado. Es aquí donde llevaremos a cabo la ejecución y el mantenimiento del sistema, con lo que su duración se prolongará hasta que el sistema deje de utilizarse o sea sustituido por otro.

➤ **IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA.**

Habiendo concluido con las pruebas realizadas de manera cuidadosa al sistema se haya ejecutado este sin error alguno y ha cumplido todos los requerimientos de los usuarios, proseguimos a la verificación e instalación en el equipo, capacitaremos a los usuarios, instalamos la aplicación y construimos todos los archivos de datos que son necesarios para utilizarla.



5.2. RECURSOS DISPONIBLES Y NECESARIOS

Entre los recursos disponibles para la elaboración de nuestro software fueron necesarios los siguientes:

Hardware:

Dos computadoras con las siguientes características:

- ✓ Procesador: Pentium IV
- ✓ Memoria RAM: 256Mb
- ✓ Disco duro: 40 GB
- ✓ Monitor: hp 54

Software:

- ✓ Microsoft Windows XP.
- ✓ Microsoft Word
- ✓ Visual Studio 6.0
- ✓ SQL-Server 7.0
- ✓ FrontPage.
- ✓ Macromedia firewors.
- ✓ Easy Case.
- ✓ Visio.

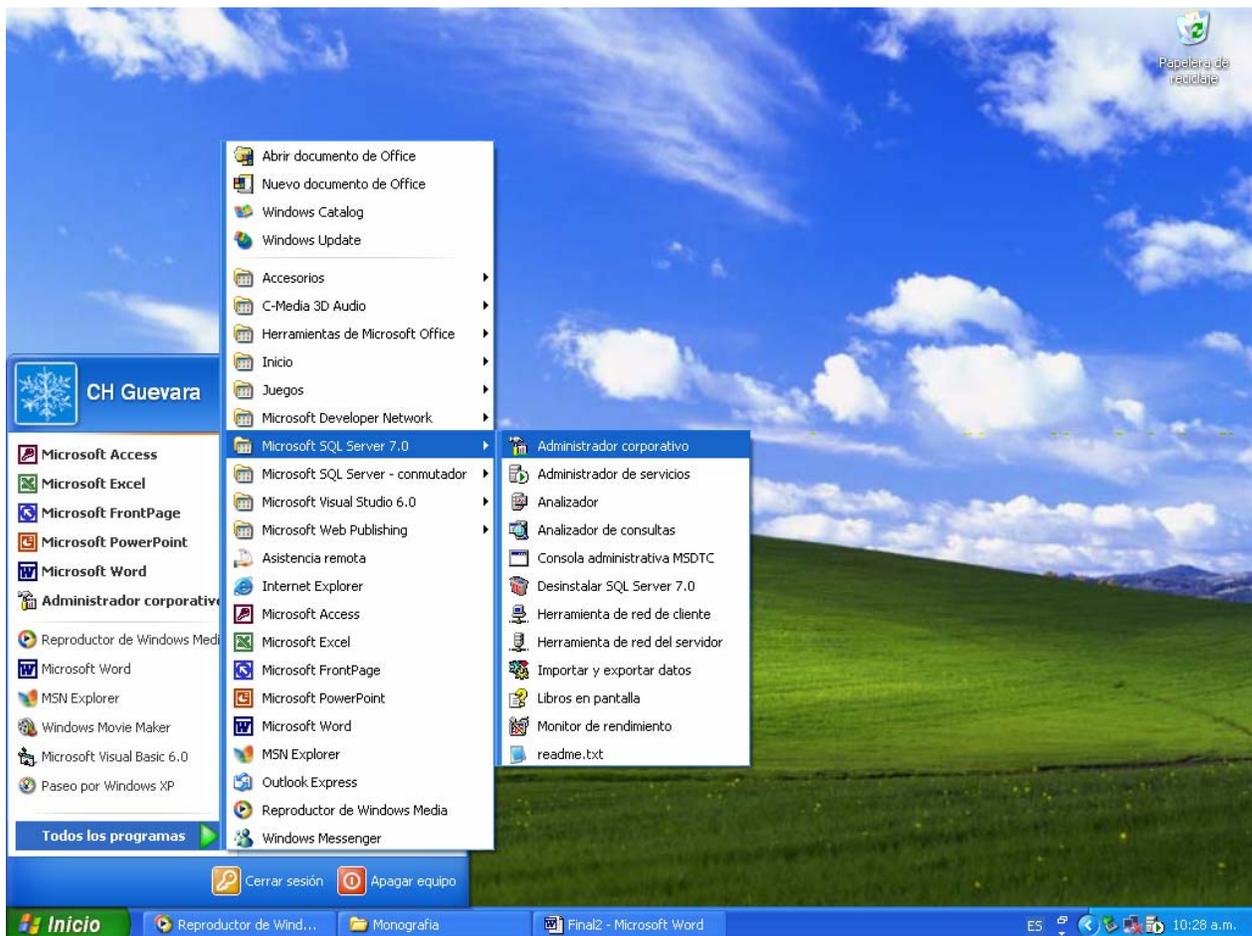


5.3. CREACIÓN DE LA BASE DE DATOS EN SQL-SERVER 7.0

Para crear una base de datos en sql server debe tener instalado sql server.

La creación de la base de datos consiste en los pasos siguientes:

1. Dirijase a **inicio** de la pantalla principal de windows y seleccione **todos los programas**, luego seleccione **Microsoft SQL-Server 7.0**, luego **administrador corporativo** como se indica en el siguiente gráfico.



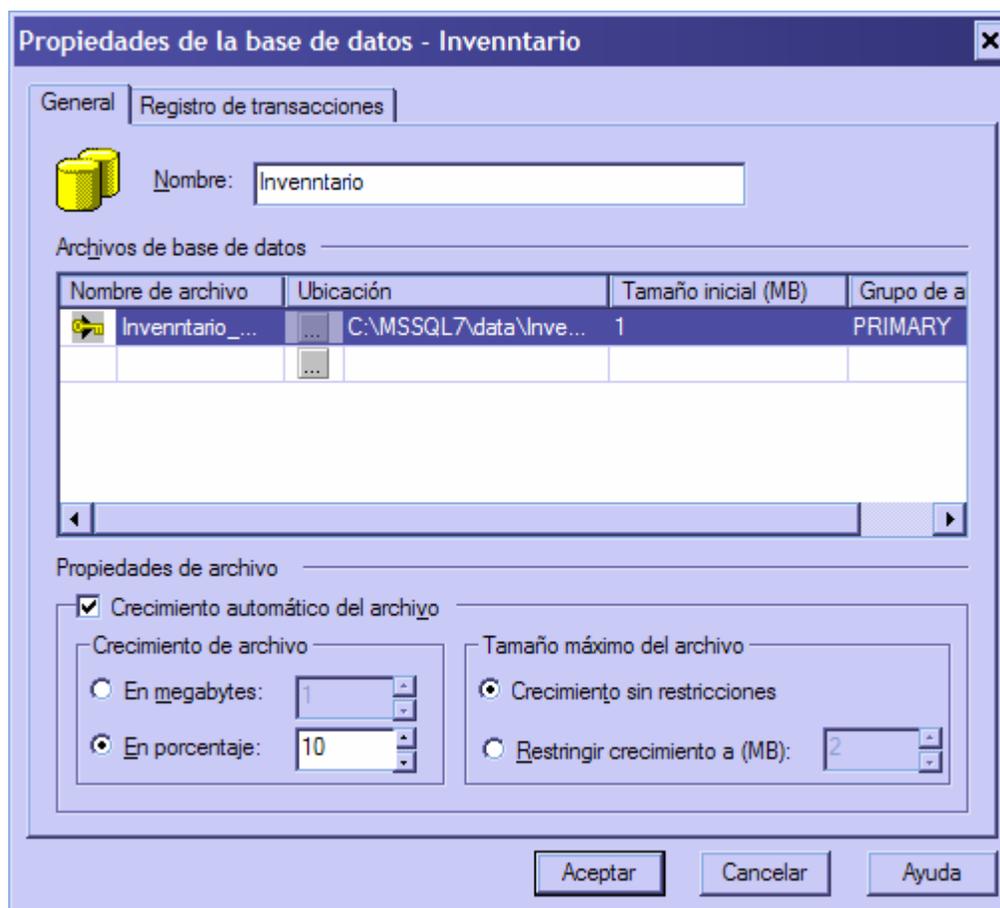


2. Por defecto aparece un grupo de servidores de sql-server, elimine el servidor que esta dentro de ese grupo y agregue un nuevo servidor de la siguiente forma: de click derecho en la opción Grupo de SQL Server, seleccione la opción Nuevo registro de servidor SQL Server, luego en la ventana de click en el botón Siguiente, seleccione y agregue el nombre del servidor que desea, de click en siguiente, siguiente, siguiente y finalizar. En el directorio Seguridad, de click derecho en Inicios de sesión y seleccione la opción Nuevo inicio de sesión, en nombre escriba el nombre con que iniciara sesión en el servidor, active la opción Autenticación de SQL Server y escriba la contraseña de inicio de sesión, en la ficha Funciones de servidor active la casilla System Administrators, de click en Aceptar y a continuación confirme la contraseña de inicio de sesión y de click en Aceptar.
3. En el nombre del servidor de click derecho y seleccione la opción Eliminar registro de servidor SQL Server, en Grupo de SQL Server de click derecho y seleccione la opción Nuevo registro de servidor SQL Server, Siguiente, agregar el servidor, Siguiente, seleccione la segunda opción, Siguiente, seleccione la segunda opción, Siguiente, Siguiente y de click en Finalizar.
4. En el nombre del servidor de click derecho y seleccione la opción Conectar del menú flotante. Al hacer este procedimiento aparecerá el siguiente gráfico.





Hasta este momento usted tiene creado su servidor y su nombre de inicio de sesión; ahora procederemos a crear nuestra base de datos llamada **Inventario** dando clic secundario del ratón sobre Bases de Datos y seleccionar **Nueva base de datos**, escriba el nombre de la base de datos y de click en Aceptar.



Ya se ha creado la base de datos Inventario.



Ahora se crearan las tablas de la bd's, para ello, en menos (-) del Grupo de SQL Server, seleccione herramientas, analizador de consultas de SQL-Server, seleccione el servidor, seleccione la opción Utilizar autenticación SQL Server, escriba el nombre de sesión y la contraseña, en la pestaña BD seleccionamos el nombre de nuestra BD Inventario; ahora pegamos el script y analizamos la consulta para verificar errores, luego ejecutamos la consulta y salimos del analizador sin guardar.

```
Analizador de consultas de SQL Server - [Consulta - SERVER.Inventario.SERVERICH Guevara - (sin título) - #1 Nueva consul...*]
Archivo  Modificar  Ver  Consulta  Ventana  Ayuda
BD: Inventario

/***** TABLA EMPRESA *****/

Create Table Empresa
(
nombreE varchar(50) NOT NULL,
direccionE varchar(150) NOT NULL,
ciudad varchar(20) NOT NULL,
pais varchar(20) NOT NULL,
telefono_o_celular char(8) NOT NULL,          /* 898-2647 */
        CONSTRAINT REGLATELEFONOCELULAR CHECK
        (telefono_o_celular LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]'),
correo_elect varchar(50)
)

/***** TABLA CLIENTE *****/

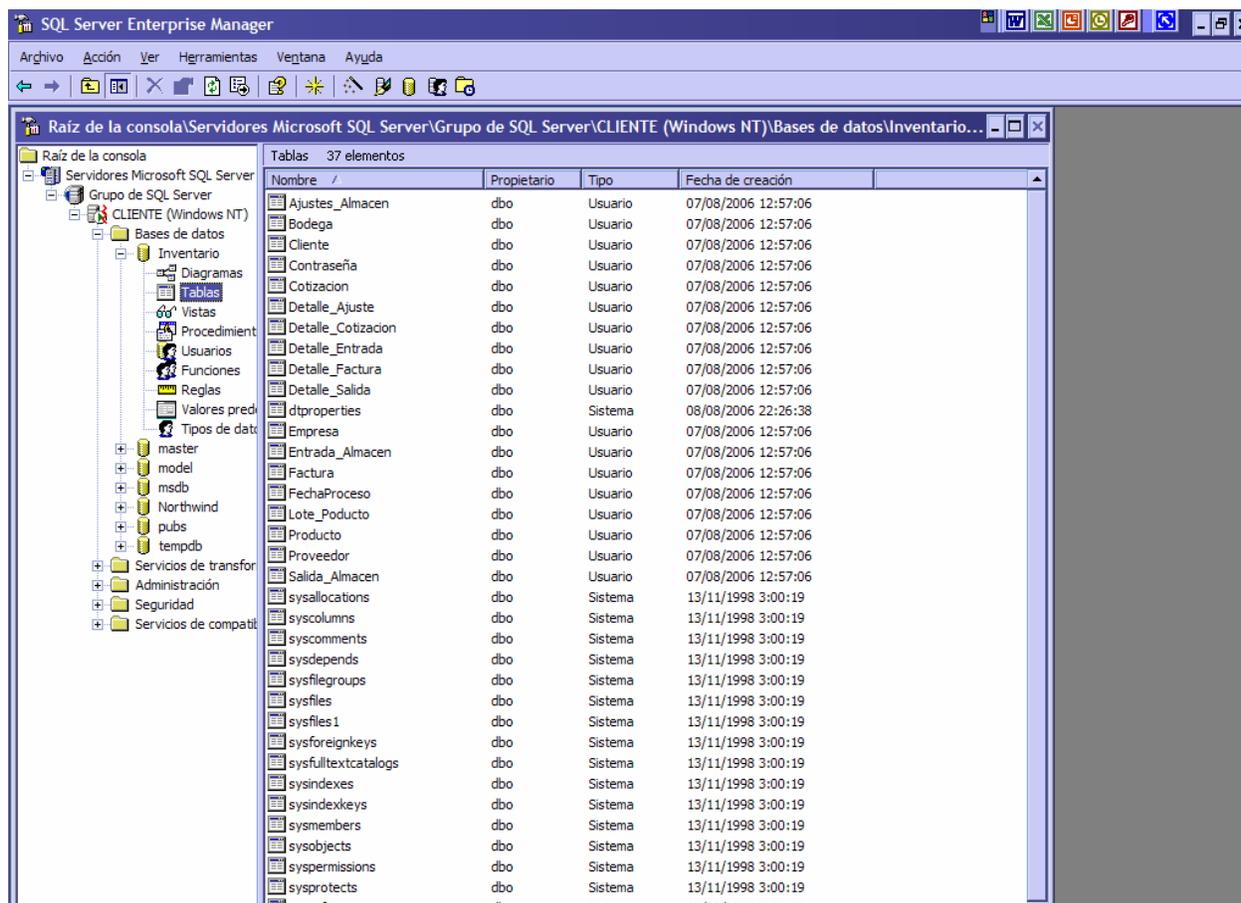
Create Table Cliente
(
codigo_cli char(6) NOT NULL,                  /* C-0000 */
        CONSTRAINT REGLACODIGOCLIENTE CHECK
        (codigo_cli LIKE '[A-Z][-][0-9][0-9][0-9][0-9]'),
nombre_cli varchar(50) NOT NULL,
cedula char(16),                             /* 084-300882-0000H */
        CONSTRAINT REGLACEDULA CHECK
        (cedula LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'),
direccion varchar(150),
telefono char(8),                             /* 898-2647 */
        CONSTRAINT REGLATELEFONO CHECK
        (telefono LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]'),
PRIMARY KEY (codigo_cli)
)

/***** TABLA PROVEEDOR *****/

Create Table Proveedor
(
```



Ahora podrá verificar la creación de las tablas de la BD abriendo la base de datos y seleccionando tablas.





VI. MARCO TEÓRICO

En el desarrollo de nuestro proyecto “Control Automatizado de Inventario Cliente-Servidor”, es necesario especificar los conceptos de algunos términos que emplearemos a lo largo del mismo, y que serán de gran utilidad para los usuarios lectores de la documentación de este proyecto.

Inventario: Los inventarios son recursos patrimoniales destinados a la transformación, consumo y ventas en función de la actividad económica que ejerce la empresa. Es el control de las entradas y las salidas (materia prima, producto, etc.) de la empresa. El control de un inventario es proceder al recuento de todas las existencias poseídas en un momento señalado. Los inventarios constituyen los principales activos, es por eso que tienen gran importancia.

Tipos de inventarios:

- Inventario físico.
- Inventario por documento.
- Inventario de personal.

El tipo de inventario que se adapta a nuestro software es el inventario físico, el cual influirá mucho en el desarrollo de la administración de todas las existencias de productos de la empresa.

Inventario físico: Se da el nombre de inventario de mercancía a la verificación o confirmación de la existencia de los materiales o bienes patrimoniales de la empresa. En realidad, el inventario es una estadística física o conteo de los materiales existentes, para confrontarla con la existencia anotadas en los ficheros de existencias o en el banco de datos sobre materiales. Algunas empresas le dan el nombre de inventario físico porque se trata de una estadística física o palpable de aquellos que hay en existencias en la empresa y para diferenciarlos de la existencia registradas en el sistema de la empresa.

Almacén: Son los espacios designados para que se guarden de manera ordenada los diferentes tipos de mercancía con que opera el negocio o empresa.



Cliente: Son las personas que interactúan con el negocio en venta y compra de mercancía ya sea al crédito o de contado.

Ventas: El valor total de las mercancías entregadas a los clientes, vendidas al contado o al crédito.

Cuenta: Representa los elementos patrimoniales agrupados por clases expresando los aumentos y disminuciones de valor que experimentan en un periodo determinado.

Compras: Son el valor de las mercancías adquiridas ya sea al contado o al crédito.

Control: Es llevar un registro de cada una de las operaciones realizadas en el ejercicio del negocio para controlar las existencias con que cuenta y tomar decisiones acordes a él.

Sistema: Es un conjunto o arreglo de elementos que están organizados para realizar un objetivo predefinido procesando información para conseguir el objetivo, un sistema basado en computadora hace uso de varios elementos del sistema: software, hardware, personas, base de datos, documentación y procedimientos.

Ingeniería del Software: Es una disciplina de la ingeniería que concierne a todos los aspectos de la producción de software. Los ingenieros de software adoptan un enfoque sistemático para llevar a cabo su trabajo y utilizan las herramientas y técnicas necesarias para resolver el problema planteado de acuerdo a las restricciones de desarrollo y recursos disponibles.

Reingeniería: La reingeniería del software, también conocida como “renovación” (renewal) o “recuperación” (recovery), tiene como principal objetivo la reconstrucción de sistemas heredados para incrementar su calidad y mantenibilidad. La reingeniería implica la revisión detenida y consciente de los procesos para lograr un rediseño radical de los mismos y así alcanzar mejoras considerables en aspectos críticos, tales como costos, calidad, servicio y rapidez. Esto implica rehacer el software o parte de él, desde cero, olvidándonos de lo que se hacía y proponer un nuevo sistema de operación.



Visual Basic: Es un lenguaje de programación basado en objeto; porque el trabajo del diseñador consiste en leer y modificar las propiedades de los objetos, llamar a sus métodos y responder a sucesos. Además visual basic se puede considerar como un entorno de desarrollo de programación visual en el que el aspecto de estos objetos se puede definir por medio de herramientas interactivas en tiempo de diseño sin tener que escribir una sola línea de código.

Interfaz Gráfica: Un objeto gráfico que se coloca en un formulario para mostrar datos, realizar una acción o facilitar la lectura de los datos de la base de datos mediante la interfaz. Entre los controles que conforman la interfaz cabe citar casillas de verificación, cuadros de edición, etiquetas, botón de comando, líneas, imágenes, formas, etc.

Proyecto: Un archivo que hace un seguimiento de todos los programas, formularios, menús, bibliotecas, informes, etiquetas, consultas y otros tipos de archivos necesarios para crear una aplicación y tiene una extensión de archivo “.vbp”.

Existencias: Comprenden aquellos productos o mercancías adquiridas por la empresa y destinadas a la venta o para procesarlas.

Facturación: Consiste en dar de alta de manera fiscal a todos los productos que se están vendiendo con todos sus detalles.

Base de Datos: Es una colección de datos clasificados y estructurados con un asunto o finalidad. Los datos de una base de datos están clasificados en campos y registros.

Tablas: Es una colección de datos sobre un tema específico presentada en forma de una matriz bidimensional, donde las filas son los registros y las columnas son los campos.

Consultas: Las consultas consisten en peticiones que hace el usuario del sistema cuando desea verificar la existencia de algún producto, la salida de producto o las entradas de mercancía que se han realizado en el periodo de proceso.

Informes: Los informes muestran de manera detallada todas las entradas y salidas desde una fecha determinada que desee ver el usuario del sistema.

Formulario: Es el contenedor de todos los elementos que conforman la interfaz de usuario; es decir, es un tipo de objeto que se utiliza fundamentalmente para introducir o mostrar datos de



una base de datos, exponen las propiedades que definen su apariencia, los métodos que muestran su comportamiento y los eventos que manifiestan la forma en que interactúan con el usuario.

Botón de Comando: Un control que está asociado a un comando. Cuando hace click en el botón de comando en tiempo de ejecución, el comando asociado al botón se ejecuta.

Aplicación: Una colección de programas, formularios, menús y otros archivos de visual basic que se compilan en un único programa y tienen una extensión ".app".

Diagrama Entidad-Relación (ER): Representa las relaciones entre los objetos de datos, es la representación que se usa para realizar la actividad de modelado de datos, se identifica un conjunto de componentes primarios para el DER: objetos de datos, atributos, relaciones y varios indicadores tipo.

Diccionario de Datos (DD): Es un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión de la funcionalidad del sistema, de las entradas, las salidas y de los componentes de los almacenes.

Diagrama de Flujo de Datos (DFD): Sirve para dos propósitos: proporcionar una indicación de cómo se transforman los datos a medida que se avanza en el sistema, y representar las funciones (y subfunciones) que transforman el flujo de datos.



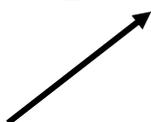
Notación Básica del Diagrama de Flujo de Datos:



Entidad Externa: Un productor o consumidor de información que reside fuera de los límites del sistema a ser modelado.

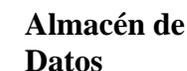


Proceso: Un transformador de información (una función) que reside dentro de los límites del sistema a ser modelado.



Flujo de datos

Flujo de Datos: Un flujo de datos se indica con una flecha, la cabeza de la flecha indica la dirección del flujo de datos.



Almacén de Datos: Un depósito de datos que se va a almacenar para el uso de uno o varios procesos; puede ser tan simple como una memoria intermedia o cola, o tan sofisticado como una base de datos relacional.

Easy Case: Es lo que se conoce como "ingeniería del software asistido por computadoras", proporciona al ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería.

SQL-Server: Es un lenguaje de consulta estructurado inventado por E.F. Codd en la década de los setenta. SQL gestiona grupos de registros lo que lo convierte en un lenguaje adecuado para consultas de cierta complejidad este lenguaje ha sido estandarizado y en la actualidad la mayoría de los servidores de base de datos aceptan su dialecto.

Cliente-Servidor: Cliente servidor es la forma en que funcionara este sistema, abra una fuente servidora que proporcionara todos los datos al cliente dando respuesta a una petición de él.

Visual Basic: La palabra "visual" hace referencia al método que se utiliza para crear la interfaz gráfica del usuario. La palabra "basic" hace referencia al language basic (Beginners All-Purpose Symbolic Instruction Code). Visual Basic es un sistema de desarrollo diseñado especialmente para crear aplicaciones con interfaz gráfica, de una forma rápida y sencilla. Para soportar este



tipo de desarrollo, visual basic utiliza fundamentalmente dos herramientas, una que permite realizar los diseños gráficos y un lenguaje de alto nivel.

Transacciones: Es una unidad de trabajo que se realiza como una operación atómica; es decir, las operaciones de la transacción se realizan todas completamente o no se ejecuta ninguna de ellas.

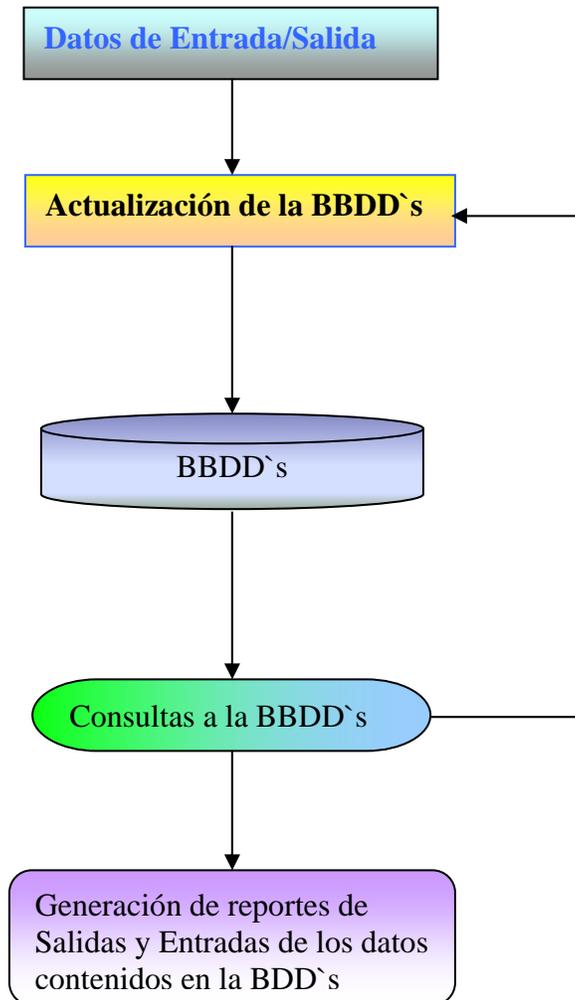
Diseño Arquitectónico: El objetivo primario es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos.



VII. ANÁLISIS

7.1. ANÁLISIS DEL SISTEMA

En si, la especificación conceptual de nuestro sistema se muestra de manera grafica en el diagrama siguiente:





7.2. ESPECIFICACIÓN DE REQUISITOS SOFTWARE

1. Introducción.

1.1. Propósito.

EL propósito de nuestro proyecto es definir un conjunto de especificaciones de requisitos software los cuales se determinaron mediante el análisis del sistema anterior, al igual que con los usuarios finales que lo utilizaban, el cual debe de cumplir el sistema “Control de Inventario”, que podrá ser utilizado por cualquier empresa para el control de sus productos.

Este documento se dirige al Lic. Manlio Reyes, administrador de su negocio, que deberá estudiarlo para su aprobación o desacuerdo antes de abordar la siguiente fase.

1.2. Alcance.

EL nombre con el cual se identificará nuestra aplicación será: Control de Inventario. El producto realizará las siguientes funciones:

- Captura de los datos de la Empresa
- Captura de los datos de la Nueva Bodega.
- Captura de los datos de Contraseña.
- Captura de la Fecha de inicio y fin del mes en proceso del Inventario.
- Captura de los datos de nuevo Producto.
- Captura de los datos de un nuevo Proveedor.
- Captura de las Entradas de Productos al Almacén.
- Captura de las Salidas de Productos del Almacén.
- Captura de los datos de un Cliente.
- Generación de Reportes.
 - Reporte de las Entradas a Almacén.
 - Reporte de las Salidas de Almacén.
 - Reporte de los Ajustes en Almacén.
 - Reporte de las Existencias por Producto.
 - Ver Kardex por Producto.
 - Reporte de los Clientes de la EMPRESA.
 - Reporte de los Proveedores de la EMPESA.
 - Reporte de las Facturas emitidas.



- Generación de las siguientes Búsquedas.
 - Búsqueda de Productos por Descripción.
 - Búsqueda de Productos por Código.
 - Búsqueda de un Cliente por Nombre.
 - Búsqueda de un Proveedor por Nombre.
 - Búsqueda de una Factura emitida a un Cliente por código de Factura.
 - Búsqueda de una Cotización emitida a un Cliente por Nombre.
- Captura de la venta de productos realizada a un Cliente y emisión de la Factura al Cliente por la mercancía servida.
- Captura de Pedidos de Clientes y emisión de la Cotización.
- Captura de los Ajustes en el Almacén.
- Modificación de los datos de la EMPRESA.
- Modificación de la Contraseña.
- Modificación de los datos del Producto.
- Modificación de los datos del Cliente.
- Modificación de los datos del Proveedor.
- Modificación de la fecha de Proceso.
- Establecer el Cierre de Mes de Proceso.
- Realizar copia de seguridad de la base de datos del sistema en unidades de almacenamiento.

1.3. Definiciones, Acrónimos y abreviaturas.

- **Proveedor:** Son las personas o Casas comerciales que suministran los productos con el que se comercia. En el ERS se denotará PROVEEDOR.
- **Producto:** Bien material existente en el negocio. En el ERS se denotará PRODUCTO.
- **Factura al Cliente:** Documento dado al Cliente en el cual se especifican los productos comprados y el importe correspondiente a cada uno de ellos, así como el total de la compra. En el ERS se denotará FACTURA_CLI.
- **Cliente:** Persona que realiza la compra de productos a la EMPRESA o negocio. En el ERS se denotará CLIENTE.
- **Cotización:** Documento emitido al CLIENTE en el cual se especifican los productos con sus precios de los cuales se han solicitado información, esto no es una venta, pero se almacenara para posteriormente convertirla en venta si el CLIENTE al cual se le emitió



la COTIZACIÓN_CLI solicita facturarla, En el ERS se denotará COTIZACIÓN_CLI.

- **Pedido del Cliente:** Son todos los Productos solicitados por el CLIENTE y el importe correspondiente a cada uno de ellos, con el cual se lleva a cabo una COTIZACIÓN_CLI o una Facturación de venta. En el ERS se denotará PEDIDO_CLI.
- **Empresa:** Lugar donde se fabrica o se vende alguna mercancía, o también, un lugar donde se presta algún tipo de servicio. En el ERS se denotará EMPRESA.
- **Existencias:** Artículos que existen en el negocio o EMPRESA. En el ERS se denotará EXISTENCIAS.

1.4. Referencia.

Informe obtenido de la entrevista realizada al Lic. Manlio Reyes, los días 13/12/2005, 06/01/2006 y el 13/01/2006.

1.5. Visión General.

Primeramente se realizará una descripción general del producto que se desea desarrollar para pasar posteriormente a estudiar individualmente cada uno de los Requisitos Específicos.

2. Descripción General.

2.1. Relaciones del producto

Este producto software no estará interactuando con ninguna aplicación de ningún tipo que pudiera existir dentro de la EMPRESA o negocio.

Los equipos en los que se desarrollará el producto software serán:

Una máquina servidora con las siguientes características:

- HP Pavilion.
- Procesador Pentium 4, 1.4 GHz.
- 256 MB de RAM.
- 40 GB de Disco Duro.

También utilizaremos 2 máquinas CLIENTE con las mismas características antes mencionadas para la prueba del funcionamiento de la aplicación.



2.2. Funciones del producto.

El producto software debe contener todas las funciones que realizaba el sistema de control inventario anterior, así como también otras nuevas funciones que harán más fiable el sistema. Estas se mencionarán a continuación:

1. Cuando el usuario ejecute la aplicación por primera vez, se presentará una ventana donde podrá introducir los DATOS DE LA EMPRESA.
2. Cuando el usuario ejecute la aplicación por primera vez, se presentará una ventana donde podrá crear LAS BODEGAS con las cuales podrá entrar al sistema y que obligatoriamente deberá crear al menos una bodega y una vez que la aplicación este en funcionamiento se podrán crear Nuevas Bodegas.
3. Cuando el usuario ejecute la aplicación por primera vez, se presentará una ventana donde podrá crear las cuentas de **Administrador** y **Usuario** e introducir los DATOS DE LAS CONTRASEÑAS con las que accederán al sistema.
4. Una vez instalada la aplicación y puesta en marcha, el Administrador del sistema podrá establecer la FECHA DE INICIO Y FIN del mes en proceso del INVENTARIO
5. Cuando aparezca un nuevo PRODUCTO por parte de un PROVEEDOR, el usuario o Administrador del sistema debe registrarlo en el ordenador y, a continuación almacenarlo en la base de datos.
6. Cuando en la EMPRESA aparezca un nuevo PROVEEDOR de nuestros productos, el usuario lo debe registrar en el ordenador y a continuación almacenarlo en la base de datos.
7. Una vez que esté creada la información de un nuevo PRODUCTO y se obtengan estos de algún PROVEEDOR, el usuario deberá seleccionar de una lista los productos de los cuales se está haciendo una entrada, deberá digitar la cantidad a recibir de cada uno de ellos. Si el PRODUCTO fuese nuevo, el Usuario lo registrará de la forma adecuada en la base de datos.
8. . El sistema calculara automáticamente el subtotal por cada PRODUCTO y la entrada total y actualizara las EXISTENCIAS, además, el Usuario deberá registrar en el ordenador la ENTRADA DE PRODUCTOS AL ALMACEN y, a continuación almacenarla en la base de datos.
9. Una vez que esté creada la información de los PRODUCTOS el usuario deberá seleccionar de una lista los productos a los cuales se le dará de baja, deberá digitar la cantidad a sacar de cada uno de ellos. El sistema calculara automáticamente el



- subtotal por cada PRODUCTO y la salida total y actualizara las EXISTENCIAS, además, el usuario deberá registrar en el ordenador la SALIDAS DE PRODUCTOS DEL ALMACÉN y, a continuación almacenarla en la base de datos.
10. Cuando se presente un CLIENTE nuevo, el usuario debe registrarlo y almacenarlo en la base de datos del sistema.
 11. El administrador o los usuarios que utilicen el sistema podrán disponer de los siguientes Listados o Reportes:
 - Reporte de las ENTRADAS a Almacén.
 - Reporte de las SALIDAS de Almacén.
 - Reporte de los AJUSTES en Almacén.
 - Reporte de las Existencias por PRODUCTO.
 - Ver Kardex por PRODUCTO.
 - Reporte de los CLIENTES.
 - Reporte de los PROVEEDORES.
 - Reporte de las FACTURAS EMITIDAS.
 12. . El usuario que esté utilizando el sistema podrá realizar las siguientes búsquedas para visualizar información detallada de los registros del sistema:
 - Búsqueda de Productos por Descripción.
 - Búsqueda de Productos por Código.
 - Búsqueda de un Cliente por nombre.
 - Búsqueda de un Proveedor por Nombre.
 - Búsqueda de una Factura emitida a un Cliente por código de Factura.
 - Búsqueda de una Cotización emitida a un Cliente
 13. . Cuando se realiza una venta de productos a un CLIENTE, el usuario deberá seleccionar los productos a vender de una lista e indicar la cantidad a llevar por cada PRODUCTO. El sistema calculara automáticamente el sub-total por PRODUCTO y el total de la venta y actualizará las EXISTENCIAS. A continuación el usuario imprimirá la FACTURA_CLI.
 14. . Cuando un CLIENTE llega a la EMPRESA a cotizar productos, este se buscará en la tabla de los clientes de la EMPRESA, en caso de que no esté, el usuario del sistema debe registrarlo en el ordenador y almacenarlo en la Base de Datos. Se seleccionará de una lista los productos a cotizar con sus respectivos precios y luego se agregarán a la COTIZACIÓN_CLI, la cual se guardará en la Base de Datos, para luego ser impresa y entregada al CLIENTE que la solicitó.



15. En algunos negocios existe la problemática que algunas veces no coinciden las cantidades de los productos que se encuentran en los almacenes con los registros existentes en el sistema, lo cual es necesario realizar un ajuste a los mismos con las cantidades físicas reales existentes en el almacén, seleccionando así de una lista de productos ya almacenados en al base de datos al cual se le realizará el ajuste, que puede ser de dos formas: ajuste al físico (disminuyendo las EXISTENCIAS) y ajuste al costo (aumentando o disminuyendo el valor (poco útil)) o ambos. Estos ajustes se realizan con el fin de autenticar las EXISTENCIAS físicas de los productos con las registradas en el sistema.
16. Una vez instalada la aplicación y puesta la misma en marcha, el administrador del sistema podrá realizar MODIFICACION DE LOS DATOS DE LA EMPRESA si este así lo desea.
17. Una vez instalada la aplicación y puesta la misma en marcha, el administrador del sistema podrá realizar MODIFICACION DE LA CONTRASEÑA para poder acceder al sistema si este así lo desea.
18. Una vez instalada la aplicación y haber introducido datos, el administrador del sistema podrá realizar MODIFICACIÓN DE LOS DATOS DEL PRODUCTO si la introducción fue incorrecta o desea cambiar cualquier dato.
19. . Una vez que la aplicación este en funcionamiento y se haya introducido CLIENTE, el administrador del sistema podrá realizar MODIFICACIÓN DE LOS DATOS DEL CLIENTE si la introducción fue incorrecta o desea cambiar cualquier dato.
20. Una vez que la aplicación este en funcionamiento y se haya introducido PROVEEDOR, el administrador del sistema podrá realizar MODIFICACIÓN DE LOS DATOS DEL PROVEEDOR si la introducción fue incorrecta o desea cambiar cualquier dato.
21. Cuando la aplicación se instala se establece la fecha de proceso, el administrador del sistema podrá realizar MODIFICACIÓN DE LA FECHA DE PROCESO únicamente si desea que la fecha de fin de proceso sea menor o mayor a la actual.
22. Cuando el administrador del sistema decida Cerrar el Mes en proceso, el sistema actualizará todas las tablas de la Base de Datos y borrará la Fecha de proceso de inventario. Esto implica que para que el sistema siga operando debe de establecerse una nueva Fecha de proceso de inventario.
23. . El administrador podrá realizar una Copia de Seguridad de la base de datos del sistema en unidades de almacenamiento como: disquetes, CD, Disco Duro, etc.



2.3. Características de los usuarios.

Los usuarios finales que utilicen la aplicación serán personas que no necesariamente tengan un alto nivel de conocimiento de informática, ya que el sistema será sencillo de manejar y además tendrá ayuda en el producto final.

2.4. Restricciones generales.

- El lenguaje de programación utilizado para la elaboración de este producto software es visual basic.
- El gestor de base de datos a utilizar será SQL-Server.
- El sistema operativo será Windows XP.

2.5. Suposiciones y dependencias.

En el futuro se indicara la posibilidad de una actualización del sistema si el cliente lo desea por causas de muerte del software o agregación de otras funcionalidades.

3. Requisitos específicos.

3.1. Requisitos funcionales.

3.1.1. Captura de los datos de la EMPRESA.

3.1.1.1. Especificación.

3.1.1.1.1. Introducción.

Este proceso deberá capturar los datos de la EMPRESA y almacenarlo en la base de datos del sistema, además, se realizara únicamente cuando se inicie el sistema por primera vez.

3.1.1.1.2. Entradas.

El usuario deberá proporcionar los siguientes datos:

- Nombre de la EMPRESA.
- Dirección.
- Ciudad.
- País.
- Teléfono o celular.
- Correo electrónico.



Datos proporcionados por el sistema:

Ninguno.

3.1.1.1.3. Proceso.

En el momento de la instalación del software, se presentará en la pantalla un formulario de captura de información en el cual el usuario del sistema podrá introducir los datos de la EMPRESA.

Los datos necesarios a introducir serán:

- Nombre de la EMPRESA: es un dato obligatorio. Describe el nombre de la EMPRESA.
- Dirección: es un dato obligatorio. Describe la dirección de la EMPRESA.
- Ciudad: es un dato obligatorio. Describe la ciudad de la EMPRESA.
- País: es un dato obligatorio. Describe el país al que pertenece la EMPRESA.
- Teléfono o celular: es un dato obligatorio. Describe el número de Teléfono o Celular de la EMPRESA.
- Correo electrónico: es un dato opcional. Describe la dirección de correo electrónico de la EMPRESA.

3.1.1.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Empresa** de la Base de Datos.

3.1.2. Captura de los datos de la Nueva Bodega.

3.1.2.1. Especificación.

3.1.2.1.1. Introducción.

Este proceso deberá capturar los datos de la nueva bodega y almacenarlo en la base de datos del sistema.

3.1.2.1.2. Entradas.

Por pantalla: Los datos de la nueva BODEGA .

- Número de la bodega .
- Descripción de la Bodega.

Datos proporcionados por el sistema:

Ninguno.



3.1.2.1.3. Proceso.

En el momento de la instalación del software, se presentará en la pantalla un formulario de captura de información en el cual el usuario del sistema podrá introducir los datos de la nueva BODEGA.

Los datos necesarios a introducir serán:

- Número de la Bodega: es un dato obligatorio. Describe el Número de la Bodega que se creara en el sistema para el orden y el almacenamiento de las mercancías.
- Descripción de la Bodega: es un dato obligatorio. Describe los tipos de PRODUCTO contenidos en esa bodega (productos en materia prima, semiprocesados o productos terminados).

3.1.2.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Bodega** de la Base de Datos

3.1.3. Captura de los datos de Contraseña.

3.1.3.1. Especificación.

3.1.3.1.1. Introducción.

Este proceso deberá capturar los datos de la CONTRASEÑA y almacenarlo en la Base de Datos del sistema.

3.1.3.1.2. Entradas.

Por pantalla: Los datos de la CONTRASEÑA.

- Nombre de Usuario
- Contraseña de Usuario.

Datos proporcionados por el sistema:

Ninguno.

3.1.3.1.3. Proceso.

En el momento de la instalación del software, se presentará en al pantalla un formulario de captura de información en el cual el usuario del sistema podrá introducir los datos de la CONTRASEÑA.

Los datos necesarios a introducir serán:

- Nombre de Usuario: es un dato obligatorio. Describe el nombre del usuario que usará el sistema, para ello solamente existen dos tipos de usuarios



(ADMINISTRADOR y USUARIO).

- Contraseña de Usuario: es un dato obligatorio. Describe la CONTRASEÑA de ese usuario.

3.1.3.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Contraseña** de la Base de Datos.

3.1.4. Captura de la Fecha de inicio y fin del mes en proceso del Inventario.

3.1.4.1. Especificación.

3.1.4.1.1. Introducción

Este proceso permite capturar la Fecha de inicio y fin del mes en proceso de Inventario y almacenarlo en la Base de Datos.

3.1.4.1.2. Entradas.

Por pantalla: Los datos de la Fecha.

- Fecha de Inicio de Proceso.
- Fecha de Fin de Proceso

Datos proporcionados por el sistema:

Ninguno

3.1.4.1.3. Proceso.

El Administrador del sistema puede establecer las Fechas de Inicio y Fin de Proceso, necesarias para que el sistema pueda operar, esto se realiza cuando el sistema se ejecuta por primera vez o cuando se realiza el proceso de Cierre de Mes de proceso. Se presentará un formulario de captura de información en el cual el Administrador del sistema podrá introducir dichas Fechas.

Los datos necesarios a introducir serán:

- Fecha de Inicio de Proceso: es un dato obligatorio. Describe la fecha de comienzo del mes en proceso.
- Fecha de Fin de Proceso: es un dato obligatorio. Describe la fecha de fin del mes en proceso.

3.1.4.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **FechaProceso** de la Base de Datos.



3.1.5. Captura de los datos de nuevo Producto.

3.1.5.1. Especificación.

3.1.5.1.1. Introducción

Este proceso deberá capturar los datos de un PRODUCTO nuevo que ingrese al sistema, el cual se almacenará en la Base de Datos.

3.1.5.1.2. Entradas.

Por pantalla: Los datos de un nuevo PRODUCTO.

- Descripción del PRODUCTO.
- Unidad.
- Grupo.
- sub.-Grupo.
- Ubicación.

Datos proporcionados por el sistema:

Referente a la Bodega:

- Código de bodega.

Referente al PRODUCTO

- Código del PRODUCTO.
- Moneda.

3.1.5.1.3. Proceso.

Antes de la llegada de un PRODUCTO a la EMPRESA, los usuarios finales pueden ingresar los datos generales del PRODUCTO con el fin de agilizar el ingreso del mismo cuando llegue. Se presentará un formulario para capturar los datos del mismo, estos se almacenarán en la Base de Datos.

Los datos necesarios a introducir serán:

- Código del PRODUCTO: es un dato obligatorio. Describe el código del PRODUCTO en almacén este código lo proporciona el sistema cada vez que se introduzca un nuevo PRODUCTO.
- Descripción del PRODUCTO: es un dato obligatorio. Describe el nombre del PRODUCTO.
- Unidad: es un dato obligatorio. Describe la unidad de medida del PRODUCTO.
- Moneda: es un dato obligatorio. Describe el tipo de moneda que se manejará el PRODUCTO; este dato será proporcionado por el sistema, entre



ellos se mostrara Córdoba y dólar (C, \$).

- Grupo: es un dato obligatorio. Describe el grupo al que pertenece el PRODUCTO.
- sub.-Grupo: es un dato no obligatorio. Describe un sub-grupo dentro del grupo al que pertenece.
- Ubicación: es un dato obligatorio. Describe la ubicación donde estará el PRODUCTO dentro del almacén o bodega.

3.1.5.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Producto** de la Base de Datos.

3.1.6. Captura de los datos de un nuevo Proveedor.

3.1.6.1. Especificación.

3.1.6.1.1. Introducción

Este proceso deberá capturar los datos de un PROVEEDOR nuevo que ingrese al sistema, el cual se almacenará en la Base de Datos.

3.1.6.1.2. Entradas.

Por pantalla: Los datos del PROVEEDOR.

- Nombre del PROVEEDOR.
- Dirección.
- Teléfono.

Datos proporcionados por el sistema:

- Código del PROVEEDOR.

3.1.6.1.3. Proceso.

Cuando la EMPRESA realiza una compra se agregará un registro del PROVEEDOR del PRODUCTO, se presentará un formulario donde el usuario podrá introducir los datos del PROVEEDOR y luego almacenarlo en la Base de Datos.

Los datos necesarios a introducir serán:

- Código del PROVEEDOR: es un dato obligatorio: Identifica al PROVEEDOR dentro del sistema, este dato es proporcionado por el sistema de manera automática.
- Nombre del PROVEEDOR: es un dato obligatorio: Identifica al



PROVEEDOR de la EMPRESA.

- Dirección: es un dato obligatorio: proporciona el lugar donde reside el PROVEEDOR.
- Teléfono: es un dato obligatorio: Describe el teléfono del PROVEEDOR.

3.1.6.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Proveedor** de la Base de Datos.

3.1.7. Captura de las Entradas de Productos al Almacén.

3.1.7.1. Especificación.

3.1.7.1.1. Introducción.

Este proceso deberá realizar la captura de todos los datos concernientes a la entrada de productos al almacén a través de una compra realizada a un PROVEEDOR y, realizar automáticamente la actualización de las EXISTENCIAS.

3.1.7.1.2. Entradas.

Por pantalla: Datos para capturar la Entrada de PRODUCTOS.

- Documento.
- Fecha de caducidad del PRODUCTO.
- Costo unitario.
- Cantidad de cada PRODUCTO.

Datos proporcionados por el sistema:

Referente a la Bodega:

- Código de la Bodega.

Referente a la Entrada:

- Código de la Entrada.
- Tipo de documento.
- Fecha de la Entrada.
- Movimiento.
- Entrada total.
- Numero de lote.
- Subtotal.

Referente al PROVEEDOR:

- Código del PROVEEDOR
- Nombre del PROVEEDOR



Referente al PRODUCTO:

- Código del PRODUCTO
- Descripción
- Unidad

3.1.7.1.3. Proceso.

Se mostrará al usuario la pantalla de introducción de datos para la nueva entrada de productos al Almacén.

Los datos necesarios a introducir serán:

- Documento de Entrada: es un dato obligatorio. Describe un respaldo tangible de los productos que se van a ingresar en el sistema. Como número de la Factura que recibimos de los PROVEEDORES.
- Fecha de caducidad del PRODUCTO: un dato no obligatorio. Describe la fecha de vencimiento del PRODUCTO.
- Costo unitario: es un dato obligatorio. Describe el costo de adquisición del PRODUCTO.
- Cantidad de cada PRODUCTO: es un dato obligatorio. Describe la cantidad de PRODUCTO que se adquirió.
- Código de la Bodega: es un dato obligatorio. Describe el código de la Bodega en la cual se almacenara el PRODUCTO.
- Código de la Entrada: es un dato obligatorio: Describe el código de la Entrada de los productos al almacén.
- Tipo de documento: es un dato obligatorio. Describe si el tipo de Entrada de los Productos es por una recepción o de Traslado de PRODUCTOS.
- Fecha de la Entrada: es un dato obligatorio. Describe la fecha en que los productos se ingresaron en la Bodega.
- Movimiento: es un dato obligatorio. Describe el tipo de operación que se esta realizando (en este caso una ENTRADA).
- Entrada total: es un dato obligatorio. Describe la sumatoria de la cantidad por el costo de los productos que se están ingresando a la Bodega.
- Numero del lote: es un dato obligatorio. Describe un nuevo lote que se creara para cada PRODUCTO agregado a la Bodega.
- Subtotal: es un dato obligatorio. Describe el subtotal de cada PRODUCTO, es decir, el costo del PRODUCTO por la cantidad que se esta ingresando a



bodega.

- Código del PROVEEDOR: es un dato obligatorio. Describe el código con el cual se identificara el PROVEEDOR dentro del sistema.
- Nombre del PROVEEDOR: es un dato obligatorio. Describe el nombre del PROVEEDOR de la EMPRESA.
- Código del PRODUCTO: es un dato obligatorio. Describe el código con el cual se identifica el PRODUCTO.
- Descripción: es un dato obligatorio. Describe el nombre que recibe el PRODUCTO.
- Unidad: es un dato obligatorio. Describe la unidad de medida del PRODUCTO (libra, quintal, etc.).

A partir de estos datos se calculará el total por cada PRODUCTO y el total de la entrada de productos al Almacén automáticamente. La cantidad de productos digitada por el usuario, actualizará de forma automática las EXISTENCIAS para cada PRODUCTO.

3.1.7.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en las tablas **Producto**, **Entrada_Almacen** y **Detalle Entrada** de la Base de Datos.

3.1.8. Captura de las Salidas de Productos del Almacén.

3.1.8.1. Especificación.

3.1.8.1.1. Introducción.

Este proceso deberá realizar la captura de todos los datos de la Salida de productos y realizar automáticamente la actualización de las EXISTENCIAS.

3.1.8.1.2. Entradas.

Por pantalla: Datos para capturar la Salida de PRODUCTOS.

- Documento.
- Cantidad de PRODUCTO.

Datos proporcionados por el sistema:

Referente a la Bodega:

- Código de la Bodega.

Referente a la Salida:

- Código de la Salida.



- Tipo del documento.
- Fecha de la Salida de los Productos.
- Movimiento.
- Salida total de Productos.
- Numero de lote.
- Subtotal.

Referente al PRODUCTO:

- Código del PRODUCTO.
- Descripción del PRODUCTO.
- Unidad.

3.1.8.1.3. Proceso.

Se mostrará al usuario la pantalla de salida de datos para la nueva salida de productos de la Bodega.

Los datos necesarios para la Salidas de Productos del Almacén serán:

- Documento de Salida: es un dato obligatorio. Describe un respaldo tangible de los productos que se van a dar de baja en el sistema (Puede ser el código de la factura si la Salida se hace a través de una venta, puede ser "Prod_Vencidos", si la salida fue por vencimiento de Productos o por algún detalle que referencie la salida).
- Cantidad de PRODUCTO: un dato obligatorio. Describe la cantidad de PRODUCTO que se sacara de la Bodega.
- Código de la Bodega es un dato obligatorio. Describe el código de la Bodega de la cual se extraerá el PRODUCTO.
- Código de la Salida: es un dato obligatorio: Describe el código de la Salida de los productos de la Bodega.
- Cantidad de PRODUCTO: es un dato obligatorio. Describe la cantidad de PRODUCTO que se adquirió.
- Tipo de documento: es un dato obligatorio. Indica que el tipo de Salida es a través de un Traslado de PRODUCTOS o Requisa de PRODUCTOS.
- Fecha de Salida de los Productos: es un dato obligatorio. Describe la fecha en que los Productos se sacaran de la Bodega.
- Movimiento: es un dato obligatorio. Describe el tipo de operación que se esta realizando (en este caso una SALIDA).



- Salida total de Productos: es un dato obligatorio. Describe la sumatoria de la cantidad por el costo de los productos que se están sacando de la Bodega.
- Numero del lote: es un dato obligatorio. Describe el número de lote del cual se sacaran los Productos contenidos en la Bodega.
- Subtotal: es un dato obligatorio. Describe el subtotal de cada PRODUCTO, es decir, el costo del PRODUCTO por la cantidad que se esta dando de baja.
- Código del PRODUCTO: es un dato obligatorio. Describe el código con el cual se identifica el PRODUCTO que se sacara de Bodega.
- Descripción del PRODUCTO: es un dato obligatorio. Describe el nombre que recibe el PRODUCTO.
- Unidad: es un dato obligatorio. Describe la unidad de medida del PRODUCTO (libra, quintal, etc.).

A partir de estos datos se calculará el total por cada PRODUCTO y el total de la Salida de productos al Almacén automáticamente. La cantidad de productos digitada por el usuario, actualizará de forma automática las EXISTENCIAS para cada PRODUCTO.

3.1.8.1.4. Salidas.

Con todos los datos mencionados anteriormente se almacenarán los registros en las tablas **Producto**, **Detalle_Salida** y **Salida_Almacen** de la Base de Datos.

3.1.9. Captura de los datos de un Cliente.

3.1.9.1. Especificación.

3.1.9.1.1. Introducción

Este proceso deberá realizar la captura de todos los datos de un nuevo CLIENTE y almacenarlo en la Base de Datos.

3.1.9.1.2. Entradas.

Por pantalla: Datos de un CLIENTE.

- Nombre del CLIENTE.
- Cédula.
- Dirección.
- Teléfono.

Datos proporcionados por el sistema:

- Código del CLIENTE.



3.1.9.1.3. Proceso.

Se mostrará al usuario la pantalla de introducción de datos del CLIENTE.

Los datos necesarios a introducir serán:

- Código del CLIENTE: es un dato obligatorio. Describe el código con el cual se identifica el CLIENTE en el sistema.
- Nombre del CLIENTE: es un dato obligatorio. Describe el nombre del CLIENTE de la EMPRESA.
- Cédula: opcional. Describe la identificación del CLIENTE
- Dirección: es un dato obligatorio. Describe el domicilio del CLIENTE.
- Teléfono: opcional. Describe el número de teléfono o de celular del CLIENTE

3.1.9.1.4. Salidas.

Con todos los datos mencionados anteriormente se almacenará el registro del CLIENTE en la Base de Datos.

3.1.10. Reporte de las Entradas a Almacén.

3.1.10.1. Especificación.

3.1.10.1.1. Introducción.

Este proceso permitirá generar el reporte de todas las Entradas al Almacén que se han generado en el rango de fecha del mes en proceso.

3.1.10.1.2. Entradas.

Por pantalla: Datos proporcionados por el sistema para generar el listado:

- Fecha de Inicio del Reporte.
- Fecha de Fin del Reporte.
- Tipo de Documento.

3.1.10.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual se presentara la información necesaria para la generación de dicho Reporte, la cual podrá ser seleccionada por el usuario. Los datos necesarios serán:

- Fecha de Inicio del Reporte: es un dato obligatorio. Describe la fecha a partir de la cual se va a generar el reporte de todas las Entradas que se registraron en una fecha mayor o igual que la fecha en la cual se inicia el mes en proceso del inventario.



- Fecha de Fin del Reporte: es un dato obligatorio. Describe la fecha hasta donde se va a realizar el reporte (todas las Entradas que se generaron en esta fecha o antes de la misma)
- Tipo de Documento: opcional. Describe el tipo de documento de la Entrada (el tipo de documento puede ser un reporte de recepción o un reporte de traslado).

A partir de estos datos que se han extraído del sistema, el usuario podrá modificar los rangos de fechas para generar la consulta o seleccionar el tipo de documento. Para el caso de la fecha inicial se puede consultar de las Entradas que se han generado antes de la fecha inicial del mes en proceso que se extrajo del sistema y para el caso de la fecha de fin, esta debe ser menor o igual que la fecha actual del sistema ya que no se puede extraer información de una Entrada que no se ha registrado en el sistema.

3.1.10.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:

- Nombre de la EMPRESA.
- Documento de Entrada.
- Tipo de Documento.
- Código del PROVEEDOR
- Nombre del PROVEEDOR.
- Código de Bodega.
- Código de la Entrada.
- Fecha.
- Código del PRODUCTO.
- Descripción.
- Unidad.
- Numero de lote.
- Costo Unitario.
- Cantidad por cada PRODUCTO.
- Subtotal por PRODUCTO.
- Total por cada Entrada de PRODUCTO al Almacén.



3.1.11. Reporte de las Salidas de Almacén

3.1.11.1. Especificación.

3.1.11.1.1. Introducción.

Este proceso permitirá generar el reporte de todas las Salidas de Almacén que se han generado en el rango de fecha del mes en proceso.

3.1.11.1.2. Entradas.

Por pantalla: Datos proporcionados por el sistema para generar el listado:

- Fecha de Inicio del Reporte.
- Fecha de Fin del Reporte.
- Tipo de Documento.

3.1.11.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual se presentara la información necesaria para la generación de dicho Reporte, la cual podrá ser seleccionada por el usuario. Los datos necesarios serán:

- Fecha de Inicio del Reporte: es un dato obligatorio. Describe la fecha a partir de la cual se va a generar el reporte de todas las Salidas que se registraron en una fecha mayor o igual que la fecha en la cual se inicia el mes en proceso del inventario.
- Fecha de Fin del Reporte: es un dato obligatorio. Describe la fecha hasta donde se va a realizar el reporte (todas las Salidas que se generaron en esta fecha o antes de la misma)
- Tipo de Documento: opcional. Describe el tipo de documento de la Salida (el tipo de documento puede ser un reporte de una requisita o un reporte de traslado).

A partir de estos datos que se han extraído del sistema, el usuario podrá modificar los rangos de fechas para generar la consulta o seleccionar el tipo de documento. Para el caso de la fecha inicial se puede consultar de las Salidas que se han generado antes de la fecha inicial del mes en proceso que se extrajo del sistema y para el caso de la fecha de fin, esta debe ser menor o igual que la fecha actual del sistema ya que no se puede extraer información de una Salida que no se ha registrado en el sistema.



3.1.11.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:

- Nombre de la EMPRESA.
- Documento de Salida.
- Tipo de Documento.
- Código de la Bodega.
- Código de la Salida.
- Fecha.
- Código del PRODUCTO.
- Descripción.
- Unidad.
- Numero de lote.
- Cantidad por cada PRODUCTO.
- Subtotal por PRODUCTO.
- Total por cada Salida del Almacén.

3.1.12. Reporte de los Ajustes en Almacén.

3.1.12.1. Especificación.

3.1.12.1.1. Introducción.

Este proceso permitirá generar el reporte de todos los Ajustes en Almacén que se han generado en el rango de fecha del mes en proceso.

3.1.12.1.2. Entradas.

Por pantalla: Datos proporcionados por el sistema para generar el listado:

- Fecha de Inicio del Reporte.
- Fecha de Fin del Reporte.
- Tipo de Documento.

3.1.12.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual se presentara la información necesaria para la generación de dicho Reporte, la cual podrá ser seleccionada por el usuario. Los datos necesarios serán:

- Fecha de Inicio del Reporte: es un dato obligatorio. Describe la fecha a partir de la cual se va a generar el reporte de todos los Ajustes que se



registraron en una fecha mayor o igual que la fecha en la cual se inicia el mes en proceso del inventario.

- Fecha de Fin del Reporte: es un dato obligatorio. Describe la fecha hasta donde se va a realizar el reporte (todos los Ajustes que se generaron en esta fecha o antes de la misma)
- Tipo de Documento: opcional. Describe el tipo de documento del Ajuste (el tipo de documento puede ser un reporte de los Ajustes al físico, al costo o de ambos).

A partir de estos datos que se han extraído del sistema, el usuario podrá modificar los rangos de fechas para generar la consulta o seleccionar el tipo de documento. Para el caso de la fecha inicial se puede consultar de los Ajustes que se han generado antes de la fecha inicial del mes en proceso que se extrajo del sistema y para el caso de la fecha de fin, esta debe ser menor o igual que la fecha actual del sistema ya que no se puede extraer información de un Ajustes que no se ha registrado en el sistema.

3.1.12.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:

- Nombre de la EMPRESA.
- Tipo de Documento.
- Fecha.
- Código de la Bodega.
- Código del Ajuste.
- Código del PRODUCTO.
- Descripción.
- Unidad.
- Cantidad en la cual se Ajusto el PRODUCTO.
- Costo en el cual se Ajusto el PRODUCTO.

3.1.13. Reporte de las Existencias por Producto.

3.1.13.1. Especificación.

3.1.13.1.1. Introducción.

Este proceso permitirá generar el reporte de las EXISTENCIAS de todos los PRODUCTOS que se encuentran en Almacén.



3.1.13.1.2. Entradas.

Por pantalla: datos para generar el listado.

Ninguno.

3.1.13.1.3. Proceso.

Se mostrará por pantalla una ventana en la cual contendrá el resultado de la consulta.

3.1.13.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:

- Nombre de la EMPRESA.
- Código de la Bodega
- Descripción de la Bodega.
- Código del PRODUCTO.
- Descripción.
- Unidad de Medida.
- Existencia del PRODUCTO.
- Costo Unitario.
- Saldo.

3.1.14. Ver Kardex por Producto.

3.1.14.1. Especificación.

3.1.14.1.1. Introducción.

Este proceso permitirá generar una vista mostrando todos los movimiento tanto de entradas, salidas y ajustes que se han llevado acabo sobre un producto de una bodega en un rango de fecha determinado.

3.1.14.1.2. Entradas.

Datos proporcionados por el sistema:

- Descripción del PRODUCTO
- Código de la bodega.
- Fecha inicial de proceso del inventario.
- Fecha final de proceso del inventario.

3.1.14.1.3. Proceso.

Se mostrará por pantalla una ventana en la cual permitirá generar una



consulta SQL con los siguientes datos proporcionados por el sistema:

- Descripción del PRODUCTO: Describe el nombre del producto.
- Código de la bodega: Describe el código de la bodega donde se encuentra el producto.
- Fecha inicial de proceso del inventario: Describe la fecha inicial de donde se mostraran los movimientos que ha tenido el PRODUCTO.
- Fecha final de proceso del inventario. Describe la fecha final hasta donde se mostraran los movimientos que ha tenido el PRODUCTO.

3.1.14.1.4. Salidas.

Una vez hecho el procedimiento anterior se visualizarán en la misma ventana todos los datos generados por la consulta que se hizo en SQL, estos se mostrarán a continuación:

Con respecto al PRODUCTO:

- Código del Producto
- Unidad.
- Existencia final (Cantidad).
- Costo Unitario.
- Saldo total.

Con respecto a las Entradas a Almacén:

- Código Entrada.
- Fecha Entrada.
- Cantidad.
- Costo Unitario.
- Sub Total.

Con respecto a las Salidas de Almacén:

- Código Salida.
- Fecha Salida.
- Cantidad.
- Costo Unitario.
- Sub Total.

Con respecto a los Ajustes en Almacén:

- Código Ajuste.



- Fecha Ajuste.
- Cantidad Ajustada.
- Costo Ajusto.
- Costo unitario.
- Sub Total.

3.1.15. Reporte de los Clientes de la Empresa.

3.1.15.1. Especificación.

3.1.15.1.1. Introducción.

Este proceso permitirá generar el reporte de todos los Clientes existentes en la EMPRESA.

3.1.15.1.2. Entradas.

Por pantalla: datos para generar el listado.

- ninguno

3.1.15.1.3. Proceso.

Se mostrará por pantalla una ventana en la cual contendrá el resultado de la consulta.

3.1.15.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:

- Código del CLIENTE.
- Nombre del CLIENTE.
- Cédula.
- Dirección.
- Teléfono.

3.1.16. Reporte de los Proveedores de la Empresa.

3.1.16.1. Especificación.

3.1.16.1.1. Introducción.

Este proceso permitirá generar el reporte de todos los Proveedores que abastecen de PRODUCTO a la EMPRESA.

3.1.16.1.2. Entradas.

Por pantalla: datos para generar el listado.

- ninguno



3.1.16.1.3. Proceso.

Se mostrará por pantalla una ventana en la cual contendrá el resultado de la consulta.

3.1.16.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:

- Código del PROVEEDOR.
- Nombre del PROVEEDOR.
- Dirección.
- Teléfono.

3.1.17. Reporte de las Facturas Emitidas.

3.1.17.1. Especificación.

3.1.17.1.1. Introducción.

Este proceso permitirá generar el reporte de todas las facturas emitidas a los clientes dentro de un rango de fecha indicado.

3.1.17.1.2. Entradas.

Por pantalla: datos proporcionados por el sistema para generar el listado.

- Fecha de Inicio del Reporte.
- Fecha de Fin del Reporte.

3.1.17.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá modificar la información que se extrajo del sistema, la cual será necesaria para la generación de dicho Reporte.

- Fecha de Inicio del Reporte: es un dato obligatorio. Describe la fecha a partir de donde se van a extraer todas las FACTURA_CLI que se le emitió a un CLIENTE.
- Fecha de Fin del Reporte: es un dato obligatorio. Describe la fecha hasta donde se van a extraer todas las FACTURA_CLI que se hayan emitido antes de la misma o en la misma fecha.

3.1.17.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera el reporte con los siguientes datos:



- Nombre de la EMPRESA.
- Código del CLIENTE.
- Nombre del CLIENTE
- Dirección del CLIENTE.
- Teléfono del CLIENTE.
- Estado de la factura.
- Bodega.
- N° de la factura.
- Fecha.
- Descripción de los productos.
- Unidad de Medida.
- Cantidad.
- Precio.
- Subtotal.
- Importe.
- IGV.
- Monto total.

3.1.18. Búsqueda de Productos por Descripción.

3.1.18.1. Especificación.

3.1.18.1.1. Introducción

Este proceso permitirá generar una búsqueda de un PRODUCTO que se encuentra ubicado en una Bodega determinada mostrando información detallada del mismo.

3.1.18.1.2. Entradas.

Por pantalla: datos para generar la búsqueda.

- Descripción del PRODUCTO.

3.1.18.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá seleccionar la descripción del PRODUCTO a buscar.

- Descripción del PRODUCTO: es un dato obligatorio. Describe el nombre del PRODUCTO.



3.1.18.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera la búsqueda del PRODUCTO en la Base de Datos, mostrando la siguiente información:

- Código de la Bodega en donde se encuentra el PRODUCTO.
- Código de PRODUCTO.
- Descripción.
- Unidad de Medida.
- Existencia final.
- Precio al CLIENTE al por menor.
- Precio al CLIENTE al por mayor.
- Grupo al que pertenece el PRODUCTO.
- Sub-Grupo al que pertenece el PRODUCTO.
- Ubicación del PRODUCTO.
- Tipo de Moneda.
- Costo Unitario.
- Saldo.
- Numero de lote en el cual se encuentra el PRODUCTO.
- Fecha de caducidad del PRODUCTO (si tiene fecha de vencimiento).
- Cantidad de PRODUCTO por cada lote existente en la Bodega.

3.1.19. Búsqueda de Productos por Código.

3.1.19.1. Especificación.

3.1.19.1.1. Introducción

Este proceso permitirá generar una búsqueda de un PRODUCTO que se encuentra ubicado en una Bodega determinada mostrando información detallada del mismo.

3.1.19.1.2. Entradas.

Por pantalla: datos para generar la búsqueda.

- Código del PRODUCTO.

3.1.19.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá seleccionar el código del PRODUCTO a buscar.

- Código del PRODUCTO: es un dato obligatorio. Describe el código con el



cual se identifica el PRODUCTO.

3.1.19.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera la búsqueda del PRODUCTO en la Base de Datos, mostrando la siguiente información:

- Código de la Bodega en donde se encuentra el PRODUCTO.
- Código de PRODUCTO.
- Descripción.
- Unidad de Medida.
- Existencia final.
- Precio al CLIENTE al por menor.
- Precio al CLIENTE al por mayor.
- Grupo al que pertenece el PRODUCTO.
- Sub-Grupo al que pertenece el PRODUCTO.
- Ubicación del PRODUCTO.
- Tipo de Moneda.
- Costo Unitario.
- Saldo.
- Numero de lote en el cual se encuentra el PRODUCTO.
- Fecha de caducidad del PRODUCTO (si tiene fecha de vencimiento).
- Cantidad de PRODUCTO por cada lote existente en la Bodega.

3.1.20. Búsqueda de un Cliente por Nombre.

3.1.20.1. Especificación.

3.1.20.1.1. Introducción

Este proceso permitirá generar una búsqueda de un CLIENTE en específico, con el fin de utilizar los datos del mismo para ver toda su información.

3.1.20.1.2. Entradas.

Por pantalla: datos del CLIENTE proporcionados por el sistema para generar la búsqueda del mismo.

- Nombre del CLIENTE.

3.1.20.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá seleccionar el Nombre del CLIENTE del cual se desea ver todos sus datos.



- Nombre del CLIENTE: es un dato obligatorio. Describe el nombre del CLIENTE.

3.1.20.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera la búsqueda del CLIENTE en la Base de Datos, se mostraran los siguientes datos del CLIENTE:

- Código del CLIENTE.
- Nombre del CLIENTE.
- Dirección.
- Cédula.
- Teléfono.

3.1.21. Búsqueda de un Proveedor por Nombre.

3.1.21.1. Especificación.

3.1.21.1.1. Introducción

Este proceso permitirá generar una búsqueda de un PROVEEDOR en específico, con el fin de utilizar los datos del mismo para ver toda su información.

3.1.21.1.2. Entradas.

Por pantalla: datos del PROVEEDOR proporcionados por el sistema para generar la búsqueda del mismo.

- Nombre del PROVEEDOR.

3.1.21.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá seleccionar el Nombre del PROVEEDOR del cual se desea ver todos sus datos.

- Nombre del PROVEEDOR: es un dato obligatorio. Describe el nombre del PROVEEDOR.

3.1.21.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera la búsqueda del PROVEEDOR en la Base de Datos, se mostraran los siguientes datos del PROVEEDOR:

- Código de PROVEEDOR.
- Nombre.
- Dirección.
- Teléfono.



3.1.22. Búsqueda de una Factura emitida a un Cliente por código de Factura.

3.1.22.1. Especificación.

3.1.22.1.1. Introducción

Este proceso permitirá generar una búsqueda de una Factura específica, que puede servir para la verificación de los datos de la misma para efectos contables.

3.1.22.1.2. Entradas.

Por pantalla: datos proporcionados por el sistema para generar la búsqueda de la FACTURA_CLI.

- Código de Factura.

3.1.22.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá seleccionar el Código de Factura que se desea buscar o ver toda su información.

- Código de Factura: es un dato obligatorio. Describe la FACTURA_CLI

El usuario, además de ver la información de una Factura emitida a una CLIENTE, la podrá anular en el caso de que el CLIENTE haga un reclamo de los productos adquiridos, siempre y cuando el usuario del sistema tenga los permisos de Administrador.

3.1.22.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera la búsqueda de la Factura en la Base de Datos, se mostrara en un formulario los datos Generales de la Factura encontrada, así como también los datos referentes al Detalle de la Factura. A continuación se mostraran los datos de la Factura:

- Código del CLIENTE.
- Nombre del CLIENTE
- Dirección del CLIENTE.
- Teléfono del CLIENTE.
- Código de la Bodega de donde se extrajeron los PRODUCTOS.
- Código de la factura.
- Fecha.
- Descripción de los productos.
- Unidad de Medida.
- Cantidad.
- Precio.



- Subtotal.
- Importe.
- IGV.
- Monto Factura.

3.1.23. Búsqueda de una Cotización emitida a un Cliente por Nombre.

3.1.23.1. Especificación.

3.1.23.1.1. Introducción

Este proceso permitirá generar una búsqueda de una COTIZACIÓN_CLI específica, que puede servir para llevar a cabo la Facturación de la misma.

3.1.23.1.2. Entradas.

Por pantalla: datos para generar la búsqueda.

- Nombre del CLIENTE.
- Fecha de emisión.

3.1.23.1.3. Proceso.

Se mostrará por pantalla un formulario en el cual el usuario podrá seleccionar el Nombre del CLIENTE que hizo la COTIZACIÓN_CLI que se desea busca junto con su fecha de emisión r.

- Nombre del CLIENTE: es un dato obligatorio. Describe el Nombre del CLIENTE que realizó la COTIZACIÓN_CLI.
- Fecha de emisión: es un dato obligatorio. Describe la fecha en que se le emitió la FACTURA_CLI.

La información antes mencionada (nombre del CLIENTE y la fecha de emisión) aparecerán como un solo dato. Cabe destacar que también se podrá realizar la Facturación de la COTIZACIÓN_CLI hecha por el CLIENTE

3.1.23.1.4. Salidas.

Una vez hecho el procedimiento anterior se genera la búsqueda de la COTIZACIÓN_CLI en la Base de Datos y se mostraran en un formulario los datos Generales de la COTIZACIÓN_CLI encontrada, así como también los datos referentes al Detalle de la COTIZACIÓN_CLI realizada por dicho CLIENTE. A continuación se mostraran los datos de la COTIZACIÓN_CLI:

- Código del CLIENTE.
- Nombre del CLIENTE



- Dirección del CLIENTE.
- Teléfono del CLIENTE.
- Código de la Bodega de donde se extrajeron los PRODUCTOS.
- Código de la COTIZACIÓN_CLI.
- Fecha.
- Descripción de los productos.
- Unidad de Medida.
- Cantidad.
- Precio.
- Subtotal.
- Importe.
- IGV.
- Monto COTIZACIÓN_CLI.

3.1.24. Captura de la venta de productos realizada a un Cliente y emisión de la Factura al Cliente por la mercancía servida.

3.1.24.1. Especificación.

3.1.24.1.1. Introducción

Este proceso deberá realizar la captura de todos los datos de una venta de productos realizada a un CLIENTE y, realizar automáticamente las actualizaciones de las EXISTENCIAS.

3.1.24.1.2. Entradas.

Por pantalla: Datos para capturar la Factura del CLIENTE.

- Cantidad.

Datos proporcionados por el sistema:

Referente a la Bodega:

- Código de la Bodega.

Referente a la Factura:

- Código de Factura.
- Fecha de la Venta.
- Subtotal.
- Importe de la Factura.
- Igv de la Factura.
- Monto de la Factura.



Referente a CLIENTE:

- Código del CLIENTE
- Nombre.
- Dirección.
- Teléfono.

Referente al PRODUCTO:

- Código del PRODUCTO.
- Descripción.
- Unidad.
- Precio.

3.1.24.1.3. Proceso.

Se mostrará al usuario la pantalla de introducción de datos para la nueva venta.

Los datos necesarios para la generación de la Factura serán:

- Cantidad: es un dato obligatorio. Describe la cantidad del PRODUCTO que el CLIENTE le ha solicitado al negocio. Si la cantidad no excede a las EXISTENCIAS se podrá efectuar la venta de ese PRODUCTO en específico.
- Código de la Bodega: es un dato obligatorio. Describe el código de la Bodega de la cual se extraerá el PRODUCTO.
- Código de Factura: es un dato obligatorio. Describe el código con el cual se identifica la Factura del CLIENTE.
- Fecha de la Venta: es un dato obligatorio. Describe la fecha en que se le genero la FACTURA_CLI.
- . Subtotal: es un dato obligatorio. Describe el resultado de multiplicar la cantidad de PRODUCTOS por el precio de venta del mismo.
- Importe de la Factura: es un dato obligatorio. Describe la sumatoria de todos los subtotales de los PRODUCTOS solicitados por los CLIENTES
- Igv de la Factura: Es un dato opcional. Describe el impuesto por la venta realizada al CLIENTE, si el negocio desea aplicárselo, para ello tendrá por defecto el valor del quince por ciento.
- Monto de la Factura: es un dato obligatorio. Describe el monto a pagar, en el cual se incluye la sumatoria del importe por los PRODUCTOS vendidos y el IVA, si este fue aplicado a la Factura.
- Código del CLIENT: es un dato obligatorio. Describe el código con el cual



se identifica el CLIENTE al cual se le emitirá la Factura.

- Nombre: es un dato obligatorio. Describe el nombre del CLIENTE.
- Dirección: es un dato opcional. Describe el lugar de residencia del CLIENTE.
- Teléfono: es un dato opcional. Describe el número de teléfono o celular del CLIENTE.
- Código del PRODUCTO: es un dato obligatorio. Describe el código con el cual se identifica el PRODUCTO que se dará de baja del almacén a través de la Factura.
- Descripción: es un dato obligatorio. Describe el nombre que recibe el PRODUCTO.
- Unidad: es un dato obligatorio. Describe la unidad de medida del PRODUCTO (libra, quintal, etc.).
- Precio: es un dato obligatorio. Describe el precio con que el PRODUCTO se deberá vender a los CLIENTES

A partir de los datos antes mencionados se calculará el subtotal por cada PRODUCTO y el total de la venta automáticamente. La cantidad de productos digitado por el usuario, actualizará de forma automática las EXISTENCIAS para cada PRODUCTO.

Si el CLIENTE al cual se le está facturando una venta no se encuentra registrado en la Base de Datos, este deberá registrarse como CLIENTE de la EMPRESA, llamando así de esta forma, al formulario para la captura de los datos de un nuevo CLIENTE.

3.1.24.1.4. Salidas.

Con todos los datos mencionados se mandará a imprimir la FACTURA_CLI a la impresora y almacenarla en las tablas **Factura y Detalle_Factura** de la Base de Datos.

3.1.25. Captura de Pedidos de Clientes y emisión de la Cotización.

3.1.25.1. Especificación.

3.1.25.1.1. Introducción.

Este proceso captura los datos de los PRODUCTOS solicitados por un CLIENTE con el objetivo de conocer los precios de los mismos, así también se efectúan la captura de los datos de un CLIENTE en el caso de que el mismo no esté



registrado en la Base de Datos.

3.1.25.1.2. Entradas.

Por pantalla: Datos para capturar la Cotización del CLIENTE.

- Cantidad.

Datos proporcionados por el sistema:

Referente a la Bodega:

- Código de la Bodega.

Referente a la Factura:

- Código de la COTIZACIÓN_CLI.
- Fecha en que se realizó la COTIZACIÓN_CLI.
- Subtotal.
- Importe de la COTIZACIÓN_CLI.
- Igv de la COTIZACIÓN_CLI.
- Monto de la COTIZACIÓN_CLI.

Referente a CLIENTE:

- Código del CLIENTE
- Nombre.
- Dirección.
- Teléfono.

Referente al PRODUCTO:

- Código del PRODUCTO.
- Descripción.
- Unidad.
- Precio.

3.1.25.1.3. Proceso.

Se mostrará al usuario la pantalla de introducción de datos para la nueva COTIZACIÓN_CLI.

Los datos necesarios para generar la COTIZACIÓN_CLI serán:

- Cantidad: es un dato obligatorio. Describe la cantidad del PRODUCTO que el CLIENTE le ha solicitado al negocio. Si la cantidad no excede a las EXISTENCIAS se podrá efectuar la COTIZACIÓN_CLI de ese PRODUCTO en específico.
- Código de la Bodega: es un dato obligatorio. Describe el código de la



Bodega de la cual se esta Cotizando el PRODUCTO.

- Código de la COTIZACIÓN_CLI: es un dato obligatorio. Describe el código con el cual se identifica la COTIZACIÓN_CLI que el CLIENTE ha solicitado.
- Fecha en que se realizo la COTIZACIÓN_CLI: es un dato obligatorio. Describe la fecha en que se le genero la COTIZACIÓN_CLI al CLIENTE.
- Subtotal: es un dato obligatorio. Describe el resultado de multiplicar la cantidad de PRODUCTOS por el precio de venta del mismo.
- Importe de la COTIZACIÓN_CLI: es un dato obligatorio. Describe la sumatoria de todos los subtotales de los PRODUCTOS solicitados por los CLIENTES
- Igv de la COTIZACIÓN_CLI: Es un dato opcional. Describe el impuesto por la COTIZACIÓN_CLI realizada por el CLIENTE, si el negocio desea aplicárselo, para ello tendrá por defecto el valor del quince por ciento.
- Monto de la COTIZACIÓN_CLI: es un dato obligatorio. Describe el monto por la COTIZACIÓN_CLI, en el cual se incluye la sumatoria del importe de los PRODUCTOS Cotizados y el IVA, si este fue aplicado a la COTIZACIÓN_CLI.
- Código del CLIENTE: es un dato obligatorio. Describe el código con el cual se identifica el CLIENTE al cual se le emitirá la COTIZACIÓN_CLI.
- Nombre: es un dato obligatorio. Describe el nombre del CLIENTE.
- Dirección: es un dato opcional. Describe el lugar de residencia del CLIENTE.
- Teléfono: es un dato opcional. Describe el número de teléfono o celular del CLIENTE.
- Código del PRODUCTO: es un dato obligatorio. Describe el código con el cual se identifica el PRODUCTO que se esta COTIZACIÓN_CLI.
- Descripción: es un dato obligatorio. Describe el nombre que recibe el PRODUCTO.
- Unidad: es un dato obligatorio. Describe la unidad de medida del PRODUCTO (libra, quintal, etc.).
- Precio: es un dato obligatorio. Describe el precio con que el PRODUCTO se ha Cotizado por el CLIENTE.

A partir de esto datos se calculará el subtotal por cada PRODUCTO y el total de la COTIZACIÓN_CLI.

Si el CLIENTE al cual se le está generando la COTIZACIÓN_CLI no se encuentra



registrado en la Base de Datos, se procederá a llamar al formulario para capturar los datos de un nuevo CLIENTE para que en caso de que este deseara realizar una compra se agilizaría el proceso de la venta.

3.1.25.1.4. Salidas.

Con todos los datos mencionados se mandará a imprimir la COTIZACIÓN_CLI hecha por el CLIENTE a la impresora y almacenarla en la Base de Datos.

3.1.26. Captura de los Ajustes en el Almacén.

3.1.26.1. Especificación.

3.1.26.1.1. Introducción.

Este proceso realiza un ajuste tanto del costo, así como también de las EXISTENCIAS físicas de los productos que se encuentran registrados en la Base de Datos, con el fin de llevar un control equilibrado entre costos y EXISTENCIAS.

3.1.26.1.2. Entradas.

Por pantalla: datos para la captura del Ajuste en Almacén.

- Ajuste al Costo del PRODUCTO.
- Ajustes a las EXISTENCIAS de PRODUCTOS.

Datos proporcionados por el sistema:

Referente a la Bodega:

- Código de la Bodega.

Referente al Ajuste.

- Código del Ajuste.
- Fecha del Ajuste
- Tipo de documento.
- Movimiento.
- Número de lote.

Referente al PRODUCTO:

- Código del PRODUCTO a ajustar.
- Descripción.
- Unidad.

3.1.26.1.3. Proceso.

Se mostrará en un menú la opción ajuste en almacén, la cual mostrara al usuario una Ventana en la que se seleccionara el tipo de ajuste a realizar. Este



podrá ser un ajuste al costo o ajuste al físico, que en dependencia del tipo de ajuste seleccionado se llenaran los campos necesarios del formulario para poder llevar acabo el ajuste.

Los datos necesarios para la generación del Ajuste serán:

- Ajuste al Costo del PRODUCTO: es un dato opcional. Describe el nuevo valor del costo que va a adquirir el PRODUCTO Este ajuste se realizará únicamente si el usuario desea realizar ajuste al costo.
- Ajustes a las Existencias de PRODUCTOS: es un dato opcional. Describe la nueva existencia en Almacén. El valor de la nueva existencia debe ser menor que el valor actual. Este ajuste se realizará únicamente si el usuario desea realizar ajuste al físico.
- Código de la Bodega es un dato obligatorio. Describe el código de la Bodega donde se encuentra el PRODUCTO que se va a ajustar.
- Código del Ajuste: es un dato obligatorio: Describe el código del Ajuste de los PRODUCTOS de la Bodega.
- Fecha del Ajuste: es un dato obligatorio. Describe la fecha en que se ajustaron los PRODUCTOS de una Bodega determinada.
- Tipo de Documento: es un dato obligatorio. Describe si el tipo de ajuste es al costo, al físico o a ambos.
- Movimiento: es un dato obligatorio. Describe el tipo de operación que se esta realizando (en este caso una AJUSTE).
- Numero del lote: es un dato obligatorio. Describe el número de lote en el cual se le aplicara el Ajuste al PRODUCTO contenido en la Bodega.
- Código del PRODUCTO: es un dato obligatorio. Describe el código con el cual se identifica el PRODUCTO de la Bodega que se le hará el Ajuste.
- Descripción del PRODUCTO: es un dato obligatorio. Describe el nombre que recibe el PRODUCTO.
- Unidad: es un dato obligatorio. Describe la unidad de medida del PRODUCTO (libra, quintal, etc.)

3.1.26.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en las tablas **Ajuste_Almacen** y **Detalle_Ajuste** de la Base de Datos.



3.1.27. Modificación de los datos de la EMPRESA.

3.1.27.1. Especificación.

3.1.27.1.1. Introducción.

Este proceso deberá capturar los datos nuevos de la EMPRESA y almacenarlo en la Base de Datos del sistema.

3.1.27.1.2. Entradas.

El sistema proporcionara los siguientes datos de la EMPRESA.

- Nombre de la EMPRESA
- Dirección
- Ciudad
- País
- Teléfono o celular
- Correo electrónico

3.1.27.1.3. Proceso.

Cuando el sistema este en funcionamiento el Administrador podrá Modificar los datos de la EMPRESA, se presentará en al pantalla un formulario para capturar los datos nuevos de la EMPRESA, en el cual el Administrador del sistema podrá introducirlos.

Los datos necesarios a introducir serán:

- Nombre de la EMPRESA: es un dato obligatorio. Describe el nombre de la EMPRESA.
- Dirección: es un dato obligatorio. Describe la dirección de la EMPRESA.
- Ciudad: es un dato obligatorio. Describe la ciudad de la EMPRESA.
- País: es un dato obligatorio. Describe el país al que pertenece la EMPRESA.
- Teléfono o celular: es un dato obligatorio. Describe el número de Teléfono o Celular de la EMPRESA.
- Correo electrónico: es un dato opcional. Describe la dirección de correo electrónico de la EMPRESA.

3.1.27.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **EMPRESA** de la Base de Datos.



3.1.28. Modificación de la Contraseña.

3.1.28.1. Especificación.

3.1.28.1.1. Introducción.

Este proceso deberá capturar los datos de la nueva CONTRASEÑA, realizar la modificación y luego almacenarla en la Base de Datos del sistema.

3.1.28.1.2. Entradas.

Por pantalla: Los datos de la CONTRASEÑA.

- Nueva Contraseña del Usuario.

Datos proporcionados por el sistema:

Ninguno.

3.1.28.1.3. Proceso.

En cualquier momento durante se esté usando el sistema, el Administrador del sistema puede modificar los datos de la CONTRASEÑA, se presentará en la pantalla un formulario de captura de información, en el cual el usuario del sistema podrá introducir los datos de la CONTRASEÑA. Con estos datos el sistema realizará dicha modificación y la actualización de la Base de Datos.

Los datos necesarios a introducir serán:

- Nueva Contraseña del Usuario: es un dato obligatorio. Describe la nueva CONTRASEÑA del usuario al cual se le esta haciendo la modificación de la misma.

3.1.28.1.4. Salidas.

El dato mencionado anteriormente se almacenará como un registro en la tabla **Contraseña** de la Base de Datos.

3.1.29. Modificación de los datos del Producto.

3.1.29.1. Especificación.

3.1.29.1.1. Introducción.

Este proceso deberá seleccionar un PRODUCTO de la lista mostrada en el formulario, realizar la modificación y luego almacenarlos en la Base de Datos del sistema.

3.1.29.1.2. Entradas.

Por pantalla: Los datos del PRODUCTO.

- Descripción.



- Precio por menor.
- Precio por mayor.
- Grupo.
- Sub grupo.
- Ubicación.
- Tiempo de aviso antes que caduque un PRODUCTO.

Datos proporcionados por el sistema:

- Código de la bodega.
- Código del PRODUCTO.

3.1.29.1.3. Proceso.

En cualquier momento durante se esté usando el sistema, el Administrador del sistema puede modificar los datos de los PRODUCTOS, se presentará en la pantalla un formulario de captura de información, en el cual el usuario del sistema podrá introducir los datos del PRODUCTO. Con estos datos el sistema realizará la modificación y la actualización de la Base de Datos.

Los datos necesarios a introducir serán:

- Descripción: es un dato obligatorio. Indica el nombre del PRODUCTO.
- Precio por menor: es un dato opcional. Indica el precio del PRODUCTO vendido al por menor.
- Precio por mayor: es un dato opcional. Indica el precio del PRODUCTO vendido al por mayor.
- Grupo: es un dato obligatorio. Indica el grupo al cual pertenece el PRODUCTO.
- Subgrupo: es un dato opcional. Indica el subgrupo al cual pertenece el PRODUCTO.
- Ubicación: es un dato obligatorio. Indica la posición donde esta el PRODUCTO dentro de la Bodega.
- Tiempo de aviso antes que caduque un PRODUCTO: es un dato opcional en dependencia si el PRODUCTO tiene fecha de caducidad. Indica cuantos días antes de que caduque el PRODUCTO nos va a avisar el sistema.

3.1.29.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Producto** de la Base de Datos.



3.1.30. Modificación de los datos del Cliente.

3.1.30.1. Especificación.

3.1.30.1.1. Introducción.

Este proceso consiste en buscar de una lista de clientes al CLIENTE que desea modificar sus datos, realizar la modificación y luego almacenarlos en la Base de Datos del sistema.

3.1.30.1.2. Entradas.

Por pantalla: Los datos del CLIENTE.

- Nombre.
- Dirección.
- Cédula.
- Teléfono.

Datos proporcionados por el sistema:

- Código del CLIENTE.

3.1.30.1.3. Proceso.

En cualquier momento durante se esté usando el sistema, el Administrador del sistema puede modificar los datos de los CLIENTES, se presentará en la pantalla un formulario de captura de información, en el cual el usuario del sistema podrá introducir los datos del CLIENTE. Con estos datos el sistema realizará dicha modificación y la actualización de la Base de Datos.

Los datos necesarios a Modificar serán:

- Nombre: es un dato obligatorio. Describe el Nombre del CLIENTE a modificar.
- Dirección: es un dato obligatorio. Describe la dirección del CLIENTE a modificar.
- Cédula: es un dato opcional. Describe la cedula del CLIENTE, este puede tener o no.
- Teléfono: es un dato opcional. Describe el teléfono del CLIENTE o ya sea celular.

3.1.30.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Cliente** de la Base de Datos.



3.1.31. Modificación de los datos del Proveedor.

3.1.31.1. Especificación.

3.1.31.1.1. Introducción.

Este proceso consiste en buscar en una lista los PROVEEDORES de todos los PRODUCTOS del negocio para llevar a cabo el proceso de modificación y luego almacenar la información en la Base de Datos del sistema.

3.1.31.1.2. Entradas.

Por pantalla: Los datos del PROVEEDOR.

- Nombre del PROVEEDOR.
- Dirección.
- Teléfono.

Datos proporcionados por el sistema:

- Código del PROVEEDOR.

3.1.31.1.3. Proceso.

Durante se esté usando el sistema, el Administrador puede modificar los datos de los PROVEEDORES, se presentará un formulario con los datos del PROVEEDOR en el cual el Administrador del sistema realizará dicha modificación y la actualización de la Base de Datos.

Los datos necesarios a Modificar serán:

- Nombre: es un dato obligatorio. Describe el Nombre del PROVEEDOR a modificar.
- Dirección: es un dato obligatorio: Describe la dirección del PROVEEDOR a modificar.
- Teléfono: es un dato obligatorio. Describe el número de teléfono o celular del PROVEEDOR.

3.1.31.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Proveedor** de la Base de Datos.



3.1.32. Modificación de la fecha de Proceso.

3.1.32.1. Especificación.

3.1.32.1.1. Introducción.

Este proceso captura la nueva fecha de fin del proceso del inventario, realiza la modificación y luego almacena la nueva fecha en la Base de Datos del sistema.

3.1.32.1.2. Entradas.

Por pantalla: La FECHA DE PROCESO.

- Fecha final.

Datos proporcionados por el sistema:

- Fecha inicial.

3.1.32.1.3. Proceso.

Durante se esté usando el sistema, el Administrador puede modificar la fecha de fin de proceso del inventario, se presentará un formulario con los datos de la fecha de proceso en el cual el Administrador del sistema realizará dicha modificación y la actualización de la Base de Datos.

Los datos necesarios a Modificar serán:

- Fecha final: es un dato obligatorio. Describe la Fecha de finalización del proceso del inventario.

3.1.32.1.4. Salidas.

Todos los datos mencionados anteriormente se almacenarán como un registro en la tabla **Proveedor** de la Base de Datos.

3.1.33. Establecer el Cierre de Mes de Proceso.

3.1.33.1. Especificación.

3.1.33.1.1. Introducción

Este proceso realiza el Cierre de Mes de Proceso del Inventario de una EMPRESA, esta acción actualizará la Base de Datos.

3.1.33.1.2. Entradas.

Por pantalla: Fecha de Cierre de proceso

Ninguno

Datos proporcionados por el sistema:

Ninguno



3.1.33.1.3. Proceso.

Se mostrará un menú al usuario en la cual puede realizar el Cierre de Mes de Proceso de Inventario en cualquier momento.

A partir de esto datos se actualizaran las tablas de la Base de Datos de la EMPRESA y se borrará la Fecha de proceso, por tanto seta necesario introducir otra Fecha de proceso si se quiere seguir trabajando con el sistema.

3.1.33.1.4. Salidas.

Con la acción que realizará este proceso en al Base de Datos es recomendable que se realice una copia de seguridad de la misma.

3.1.34. Realizar copia de seguridad de la Base de Datos del Sistema en unidades de almacenamiento.

3.1.34.1. Especificación.

3.1.34.1.1. Introducción.

Este proceso realiza un respaldo de la Base de Datos de la EMPRESA guardándola en unidades de almacenamiento.

3.1.34.1.2. Entradas.

Por pantalla.

Nombre del archivo de la copia se seguridad.

Datos proporcionados por el sistema:

Ninguno.

Proceso.

Se mostrará en un menú la opción de Realizar la copia de seguridad de la Base de Datos el cual se guardara la Base de Datos con todas la tablas y registros correspondientes en la unidad respectiva, puede ser: Disquetes, Discos duros, etc. Además el Administrador del sistema podrá restaurar la base de datos.

3.1.34.1.3. Salidas.

Con toda la acción realizada se obtiene la copia de seguridad de la Base de Datos del sistema.

3.1.34.2. Interfaces externas.

3.1.34.2.1. Interfaces de usuarios.

La captura de los datos se realizara de forma interactiva por pantalla.



3.1.34.2.2. Interfaces hardware.

Se podrá utilizar cualquier terminal conectado al ordenador central y que soporte la aplicación.

3.1.34.2.3. Interfaces software.

Esta aplicación interactúa únicamente con el gestor de base de datos SQL Server, no existe otra para poder ser ejecutada.

3.1.34.2.4. Interfaces de comunicación.

No se nos ha dicho que se incluirá una interfaz de comunicación en la aplicación.

3.2. Requisitos de funcionamiento.

No existe ninguna restricción en cuanto a la ejecución simultánea del sistema.

3.3. Restricciones de diseño.

El formato de las pantallas y listados de la aplicación deberá contener los controles necesarios para el buen funcionamiento y todas deberán regresar a la interfaz principal.

3.4. Atributos.

3.4.1. Seguridad.

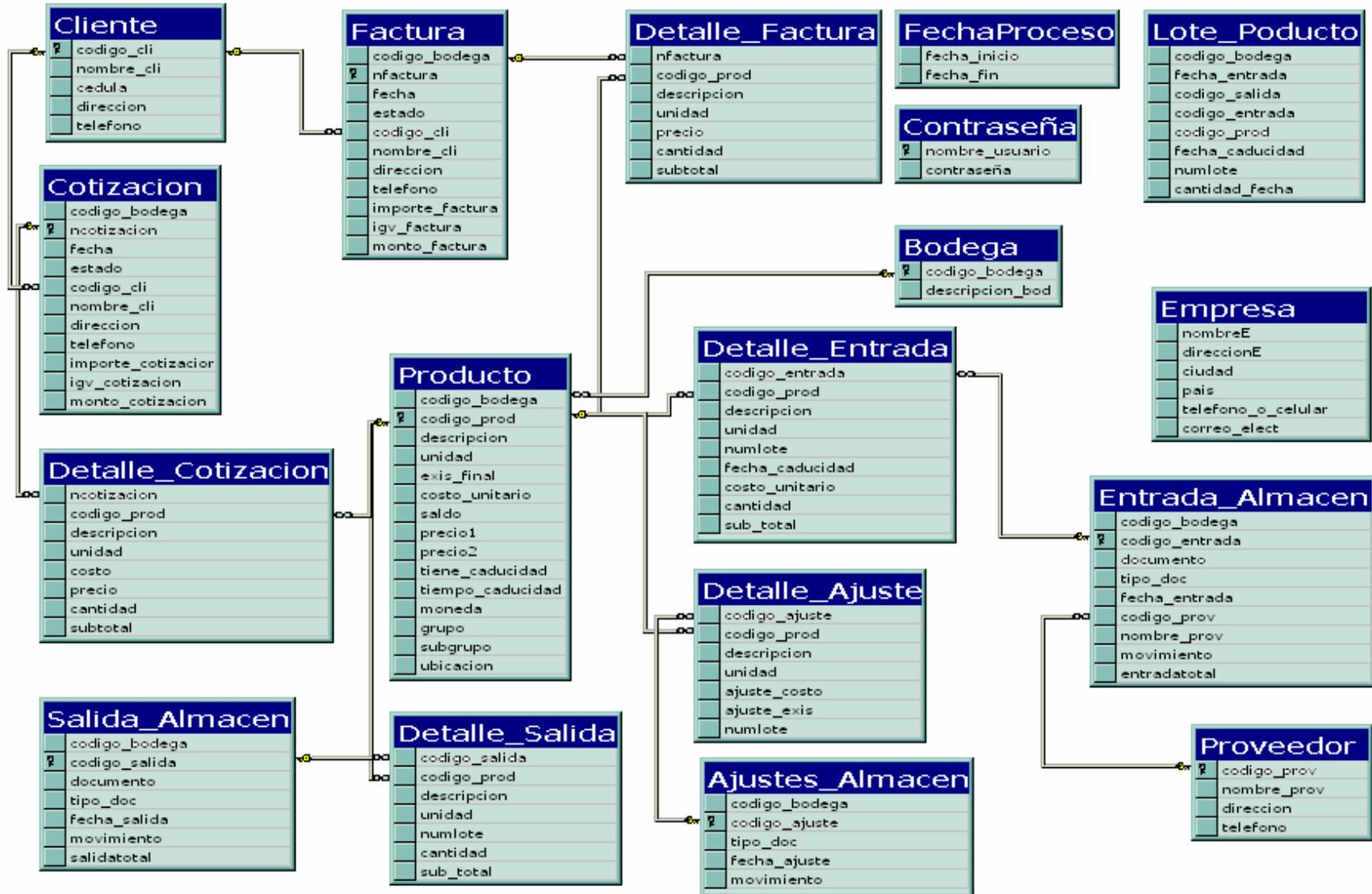
Para poder realizar alguna operación riesgosa en nuestra aplicación el usuario del sistema deberá tener los permisos de administrador.

3.4.2. Mantenimiento.

Cualquier modificación que afecte a los requisitos mencionados en este documento deberá ser reflejado en el mismo y en las otras fases.



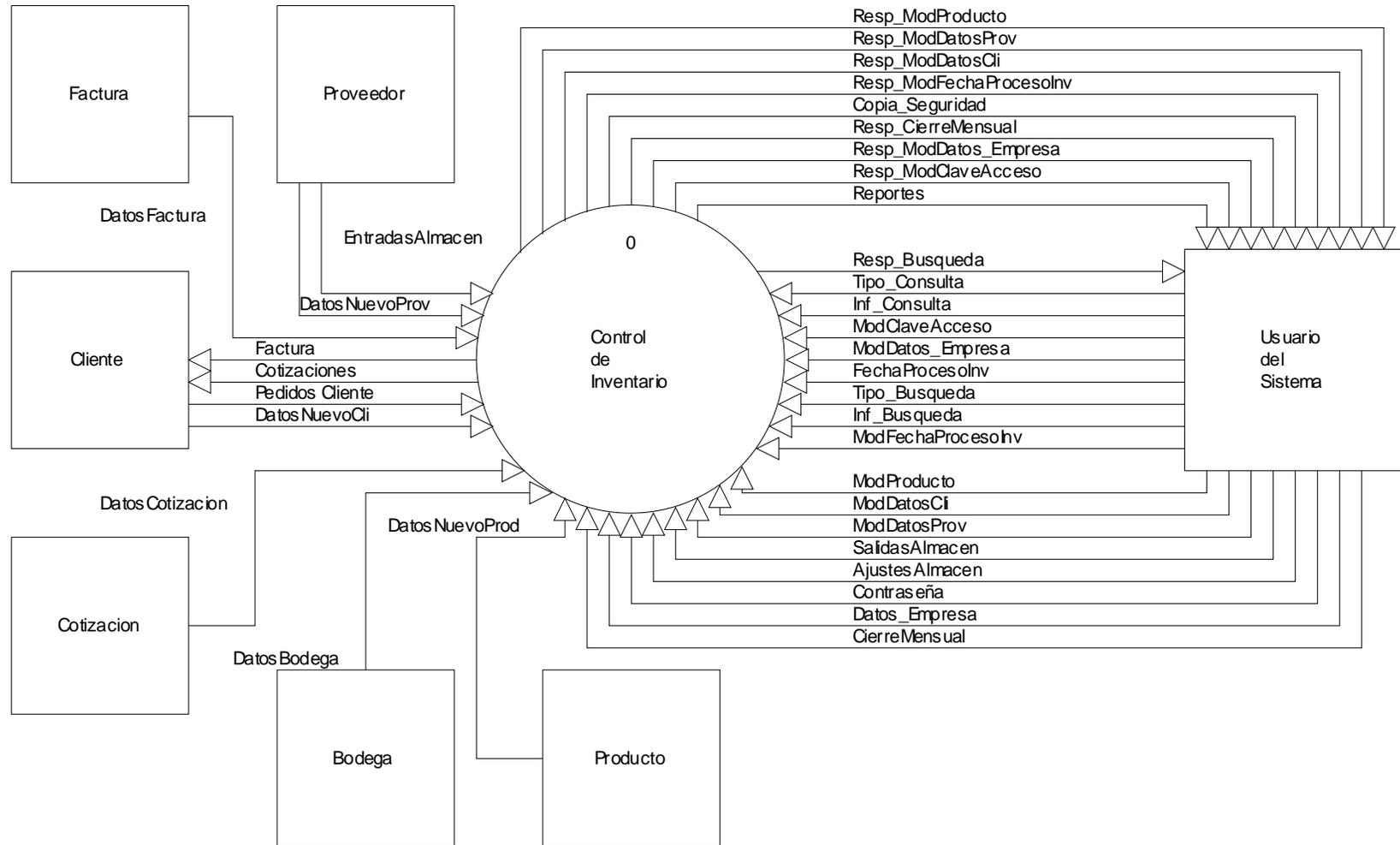
7.3.1. RELACIONES DE LAS TABLAS DE LA BBDD'S





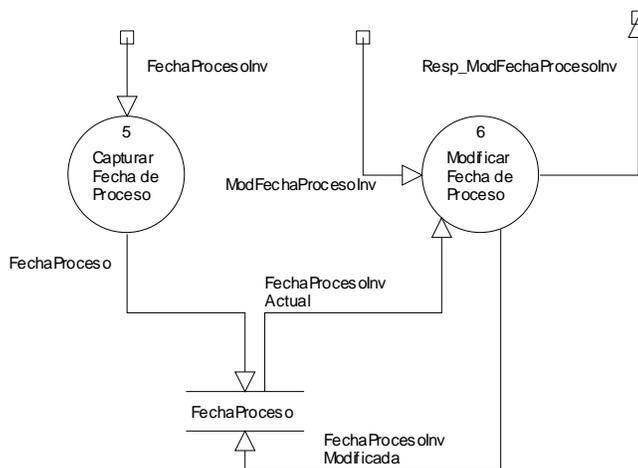
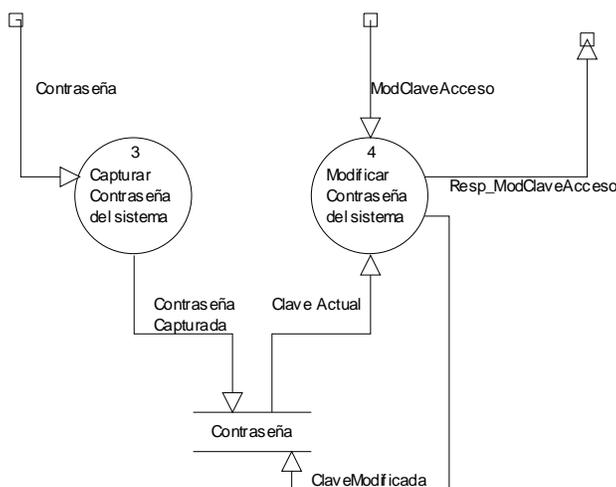
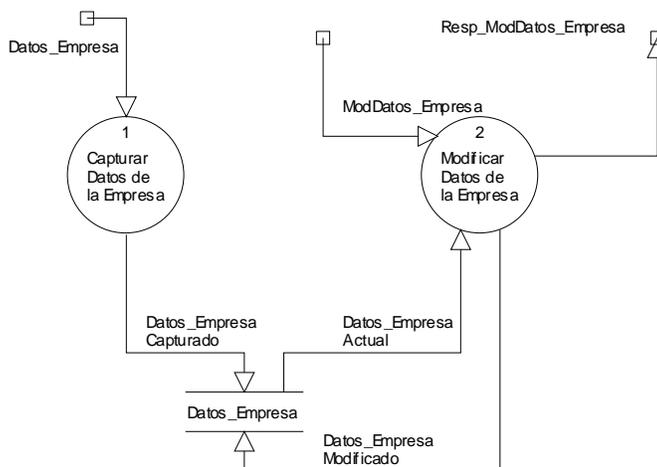
7.4. DIAGRAMA DE FLUJO DE DATOS

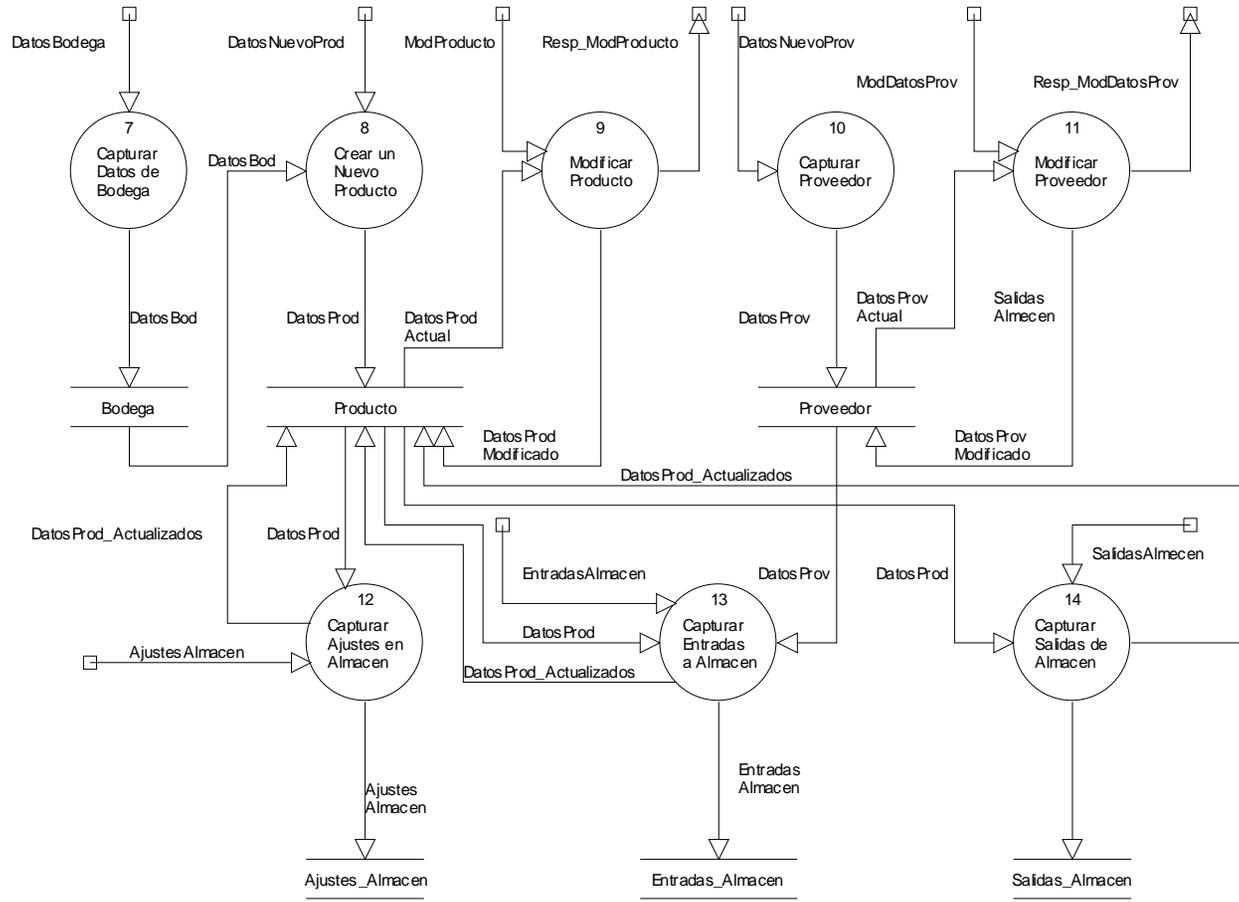
DIAGRAMA DE COTEXTO

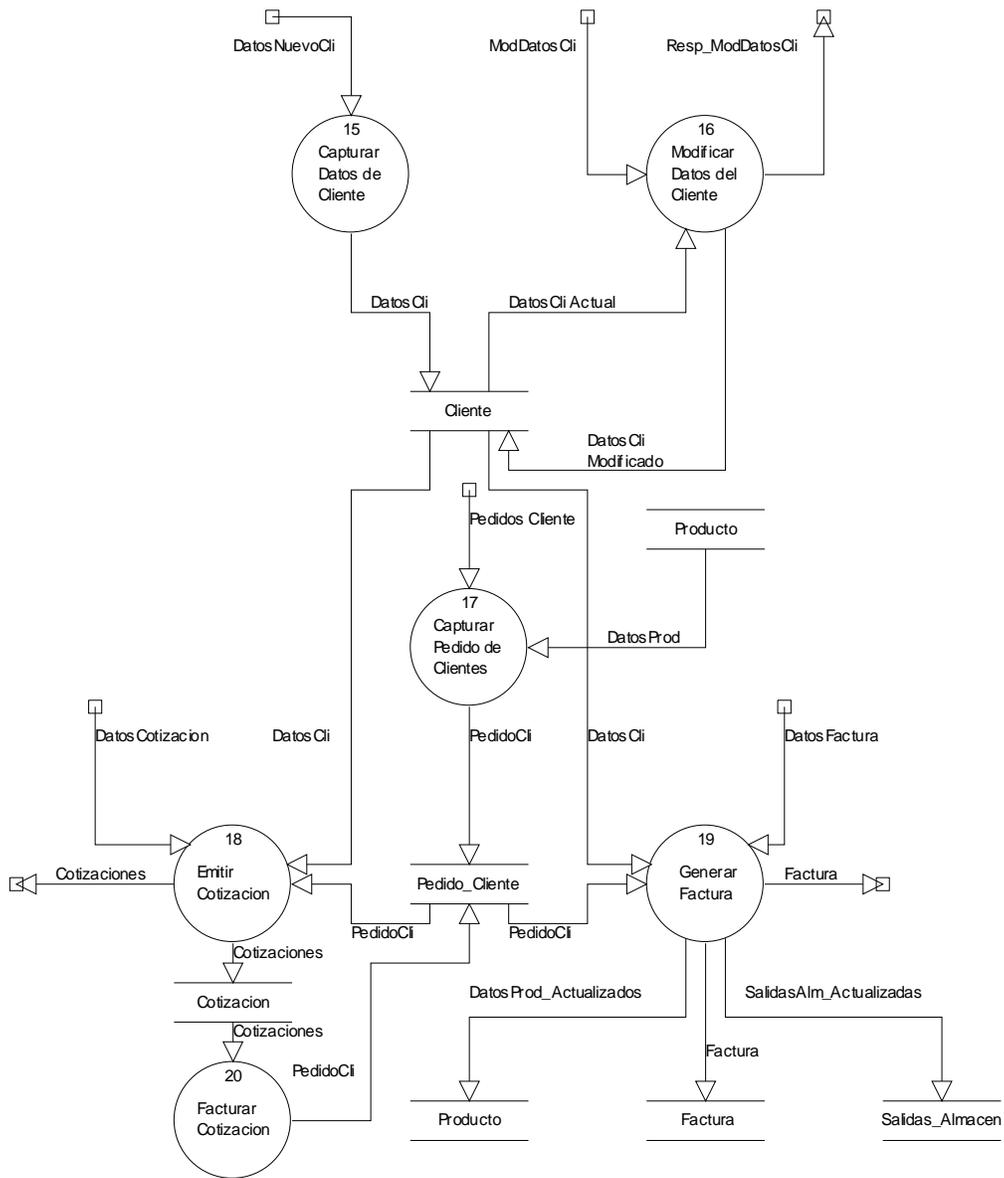


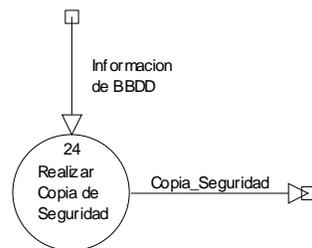
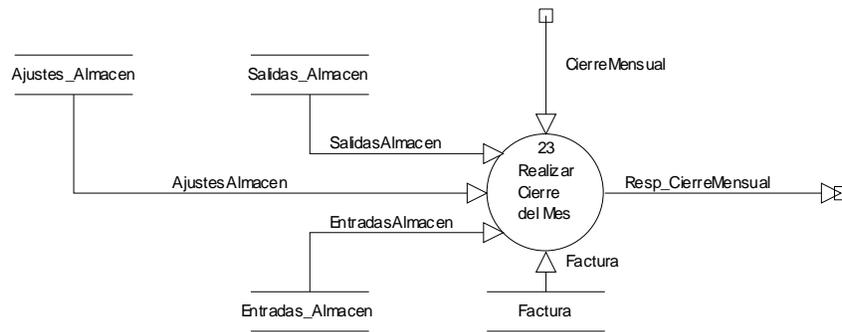
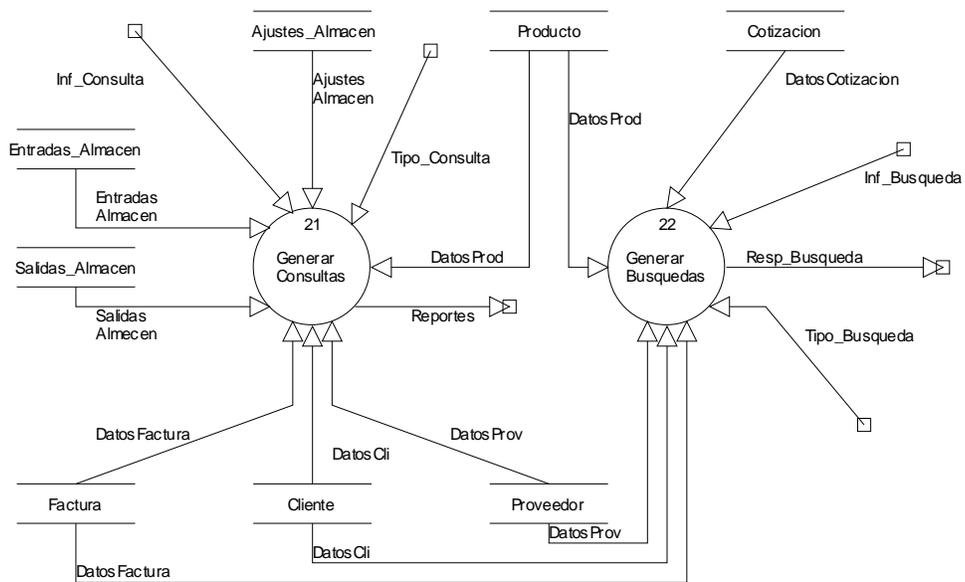


NIVEL 1



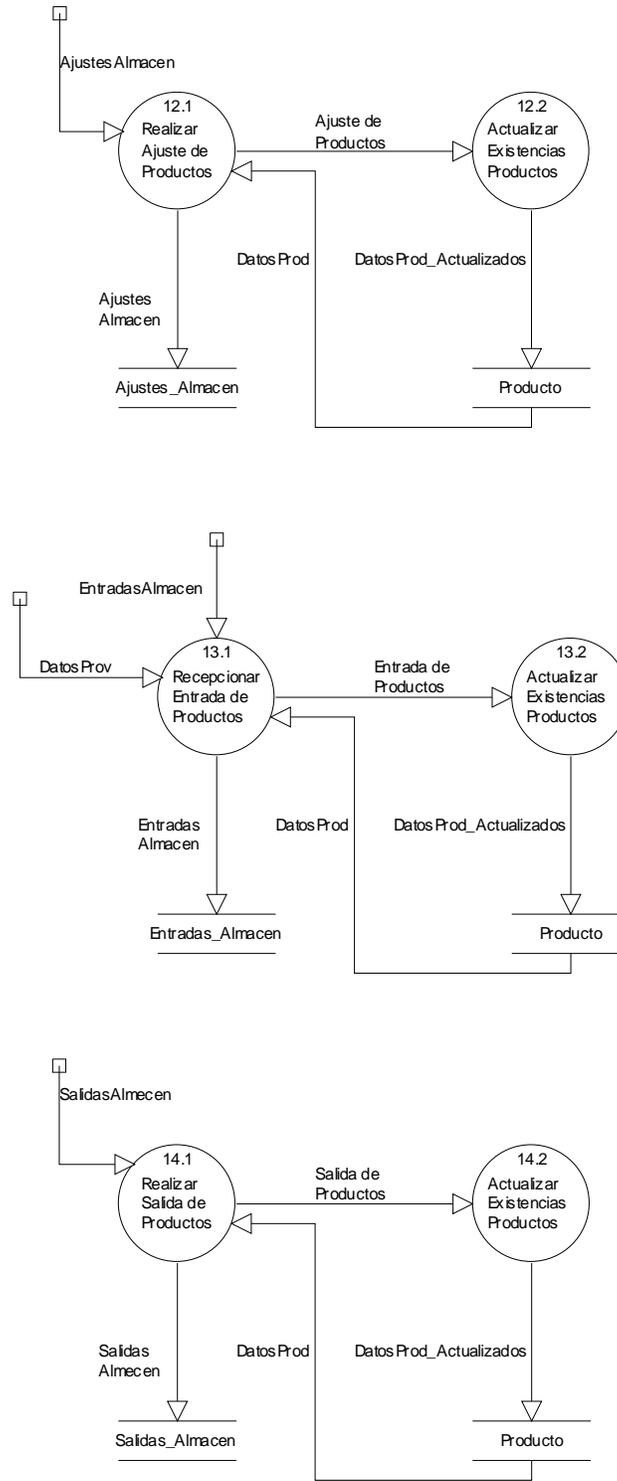


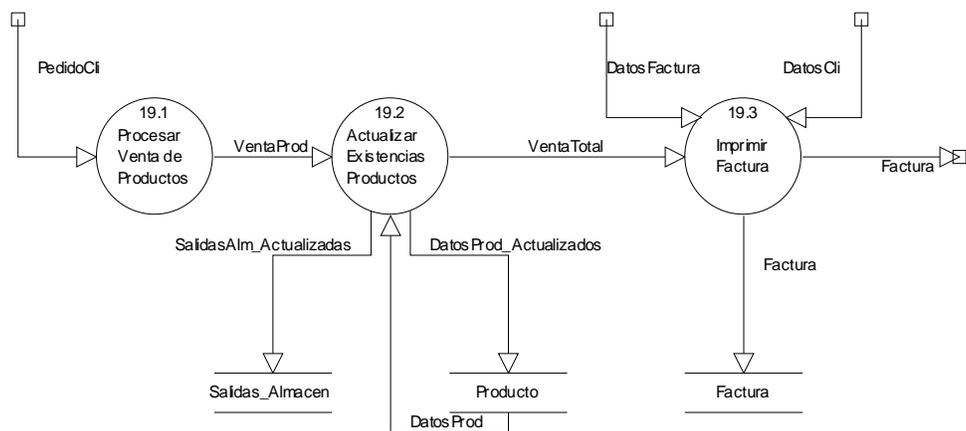
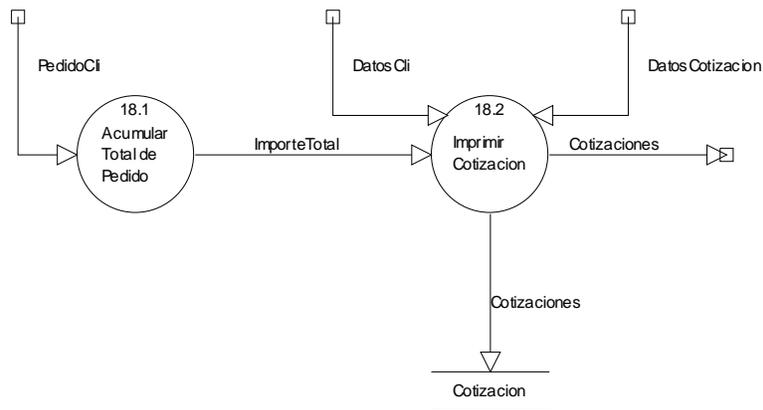


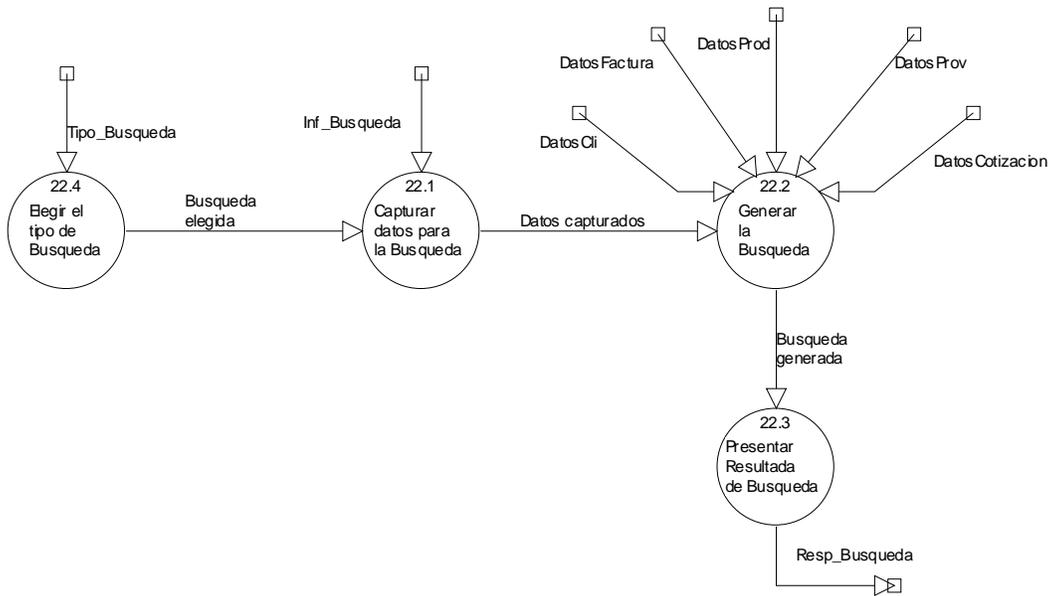
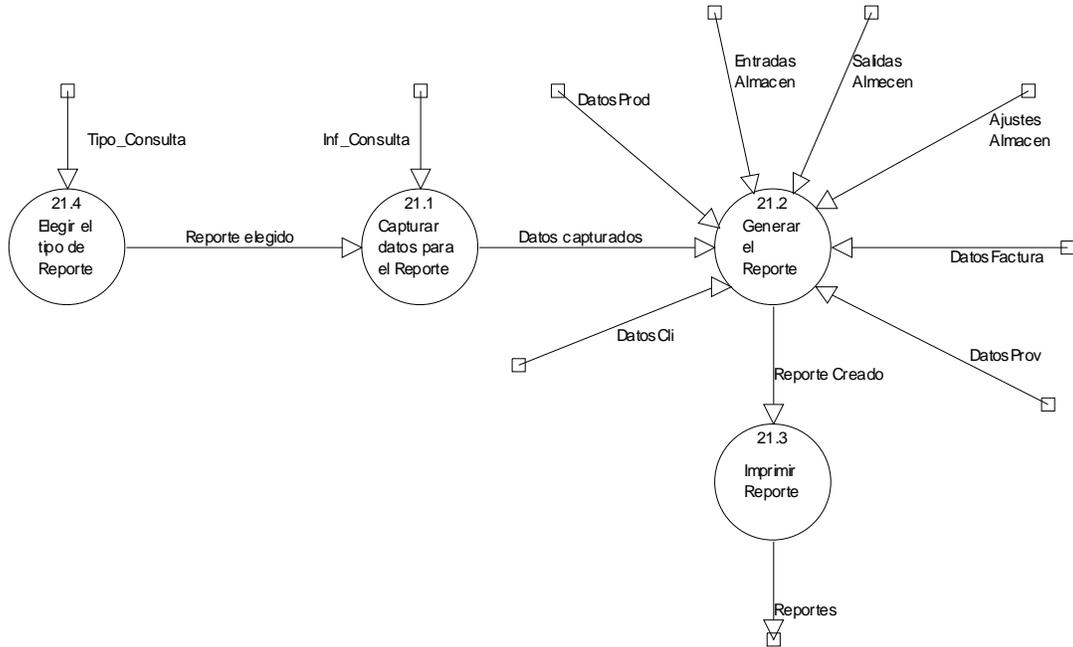




NIVEL 2









7.5. DICCIONARIO DE DATOS

FLUJO DE DATOS.

Datos de Entrada:

DatosNuevoProv: codigo_prov+nombre_prov+direccion+telefono.

EntradasAlmacen: codigo_bodega+codigo_entrada+documento+tipo_doc+fecha_entrada+codigo_prov+nombre_prov+movimiento+entradatotal+codigo_prod+descripcion+unidad+numlote+fecha_caducidad+costo_unitario+cantidad+sub_total.

DatosBodega: codigo_bodega+descripción_bod.

DatosNuevoProd: codigo_bodega+codigo_prod+descripcion+unidad+exis_final+costo_unitario+saldo+precio1+precio2+tiene_caducidad+tiempo_caducidad+moneda+grupo+subgrupo+ubicacion.

Pedidos Cliente: codigo_prod+ descripcion+unidad+cantidad+precio+subtotal.

DatosNuevoCli: codigo_cli+nombre_cli +cedula+direccion+telefono.

FechaProcesoInv: fecha_inicio+fecha_fin.

ModClaveAcceso: contraseña.

ModDatos_Empresa: nombreE+direccionE+ciudad+pais+telefono_o_celular+correo_elect.

ModProducto: descripcion+precio1+precio2+grupo+subgrupo+ubicacion+ tiempo_caducidad.

ModDatosCli: nombre_cli+cedula+direccion+telefono.

ModDatosProv: nombre_prov+direccion+telefono.

ModFechaProcesoInv: fecha_fin.

DatosFactura: codigo_bodega+nfactura+fecha+estado+codigo_cli+nombre_cli+direccion+telefono+importe_factura+igv_factura+monto_factura.

DatosCotizacion: codigo_bodega+ncotizacion+fecha+estado+codigo_cli+nombre_cli+direccion+telefono+importe_cotizacion+igv_cotizacion+monto_cotizacion.

Tipo_Busqueda: permite establecer o elegir el tipo de búsqueda, la cual puede ser de Factura por código, Clientes por nombres, etc.

Inf_Busqueda: Este proporciona la información necesaria para la generación de una búsqueda,



ya sea de productos, a través del código o de la descripción del mismo, también puede ser el nombre de un Cliente o un Proveedor para realizar la búsqueda de los mismos. Además puede referirse al código de una Factura o al nombre de un cliente que ha realizado una cotización en una determinada fecha.

Tipo_Consulta: Este nos permite elegir si el tipo de consulta será de los Clientes de la Empresa, Proveedores, etc.

Inf_Consulta: Este proporciona la información necesaria para la generación de una Consulta.

SalidasAlmacen: codigo_bodega+codigo_salida+documento+tipo_doc+fecha_salida+movimiento+salidatotal+codigo_prod+descripcion+unidad+numlote+cantidad+sub_total

AjustesAlmacen: codigo_bodega+codigo_ajuste+tipo_doc+fecha_ajuste+movimiento+codigo_prod+descripcion+unidad+ajuste_costo+ajuste_exis+costo_unitario+numlote.

Contraseña: nombre_usuario+contraseña.

Datos_Empresa: nombreE+direccionE+ciudad+pais+telefono_o_celular+correo_elect.

CierreMensual: Fecha de fin de mes.

Información de BBDD: Todos los almacenes (Proveedor+Entradas_Almacen+DatosBodega +Salidas_Almacen+Ajustes_Almacen+Producto+Cliente+Pedido_Cliente+Cotizacion+Factura+NombreEmpresa+Contraseña+FechaProceso).

Datos de Salida:

Factura: codigo_bodega+nfactura+fecha+codigo_cli+nombre_cli+direccion+telefono +importe_factura+igv_factura+monto_factura+descripcion+unidad+cantidad+precio+subtotal.

Cotizaciones: codigo_bodega ncotizacion+fecha+codigo_cli+nombre_cli+direccion+telefono +importe_cotizacion+igv_cotizacion+monto_cotizacion+descripcion+unidad+costo+cantidad+ precio +subtotal.

Copia_Seguridad: Unidades de almacenamiento (Disquete+Disco Duro).

Resp_CierreMensual: Mensaje de aviso del cierre del mes en proceso del inventario.

Resp_ModDatos_Empresa: Datos de la empresa modificados.

Resp_ModProducto: El Producto X se ha modificado con éxito.

Resp_ModDatosProv: El Proveedor X se ha modificado con éxito.



Resp_ModDatosCli: El Cliente X se ha modificado con éxito.

Resp_ModFechaProcesoInv: La Fecha de proceso se ha modificado con éxito.

Resp_ModClaveAcceso: Clave de acceso modificada.

Reportes: Entradas a Almacén+Salidas de Almacén+Ajustes en Almacén+Clientes+ Proveedores+Existencias por Producto+Kárdex por Producto+Facturas emitidas.

Resp_Búsqueda: Este flujo como resultado nos muestra la información correspondiente a la búsqueda en específico que se ha realizado, ya sea de: Productos, Factura, Cliente o Proveedor y de una Cotización hecha por un determinado Cliente.

Almacenes:

Proveedor: Datos de los Proveedores a los cuales se les compran productos.

Entradas_Almacén: Contiene los datos referente a los Productos que se han suministrado al almacén con sus cantidades específicas.

Bodega: Contiene los datos de las bodegas que pertenecen a la Empresa.

Salidas_Almacén: Contiene los datos de los productos a los cuales se les dan de baja en el almacén con sus cantidades especificadas.

Ajustes_Almacén: Es un contenedor de las conciliaciones de las Entradas, las Existencias y las Salidas de productos y así también de sus costos.

Producto: Contiene los datos de todos los Productos de la empresa.

Cliente: Contenedor de todos los datos referente a un Cliente.

Pedido_Cliente: Datos de los Productos (mas que todo hace referencia al almacén de Producto) que un Cliente le ha solicitado al negocio para realizar una compra de los mismos a través de una Factura o realizar una Cotización.

Cotizacion: Datos de las Cotizaciones de Productos que han realizado los Clientes al negocio.

Factura: Contenedor de los Datos generales de las Facturas que se les han emitido a los Clientes.

Datos_Empresa: Datos generales de la Empresa.

Contraseña: Datos de la Contraseña para poder acceder al sistema.

FechaProceso: Datos de las fechas del mes en proceso del Inventario.



VIII. DISEÑO

8.1. DISEÑO DE DATOS

Tabla 1: Empresa

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
nombreE	varchar	50	Nombre de la Empresa.
direccionE	Varchar	150	Dirección de la Empresa.
ciudad	varchar	20	Ciudad en el País.
pais	varchar	20	País al que pertenece.
teléfono_o_celular	char	8	Número de Teléfono.
correo_elect	varchar	50	Correo Electrónico.

Tabla 2: Cliente

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_cli	char	6	Código del Cliente.
nombre_cli	varchar	50	Nombre del Cliente.
cedula	char	16	Cédula del Cliente.
direccion	varchar	150	Dirección del Cliente.
telefono	char	8	Teléfono del Cliente.

Tabla 3: Proveedor

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_prov	char	6	Código del Proveedor.
nombre_prov	varchar	50	Nombre del Proveedor.
direccion	varchar	150	Dirección de este Proveedor.
telefono	char	8	Teléfono del Proveedor.

Tabla 4: Bodega

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_bodega	char	4	Código de una Bodega.
descripción_bod	varchar	50	Estado de los Productos (Materia prima, Productos en procesos o Productos terminados, etc.).

**Tabla 5: Producto**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
codigo_bodega	char	4	Código de la Bodega.
codigo_prod	char	8	Código del Producto Registrado.
descripcion	varchar	50	Nombre del producto.
unidad	varchar	15	Unidad de medida del Producto.
exis_final	real	4	Existencias total del Producto.
costo_unitario	real	4	Costo unitario del Producto.
saldo	real	4	Valor de la existencia total de un Producto.
precio1	real	4	Precio del Producto por menor.
precio2	real	4	Precio del Producto por mayor.
tiene_caducidad	bit	1	Verifica si el Producto caduca.
tiempo_caducidad	integer	4	Verifica fecha de caducidad.
moneda	char	1	Tipo de Moneda (C o \$).
grupo	varchar	15	Grupo al que pertenece el Producto.
subgrupo	varchar	15	Sub-Grupo al que pertenece.
ubicacion	varchar	50	Lugar de Ubicación del Producto.

Tabla 6: Entrada Almacen

Nombre del Campo	Tipo de Datos	Longitud	Descripción
codigo_bodega	char	4	Código de la Bodega.
codigo_entrada	char	7	Código de la Entrada a Bodega.
documento	varchar	15	Documento de respaldo de la Entrada.
tipo_doc	char	2	Tipo de Documento (Reporte de Recepción "RR", Traslado de Bodega "TD").
fecha_entrada	smalldatetime	4	Fecha de Entrada.
codigo_prov	char	6	Código del Proveedor.
nombre_prov	varchar	50	Nombre del Proveedor.
movimiento	varchar	10	Tipo de movimiento (ENTRADA, SALIDA O AJUSTE).
entradatotal	real	4	Representa la Entrada total de los Productos.

**Tabla 7: Detalle_Entrada**

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_entrada	char	7	Código de la Entrada a Bodega.
codigo_prod	char	8	Código del Producto.
descripción	varchar	50	Nombre del producto.
unidad	varchar	15	Unidad de medida del Producto.
numlote	integer	4	Lote del Producto ingresado.
fecha_caducidad	smalldatetime	4	Fecha de caducidad del Prod.
costo_unitario	real	4	Costo unitario del Producto.
cantidad	real	4	Cantidad del Producto ingresado.
sub_total	real	4	Subtotal del costo del Producto.

Tabla 8: Lote Producto

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_bodega	char	4	Codigo de la bodega.
fecha_entrada	smalldatetime	4	Fecha que se ingreso el Producto.
codigo_salida	char	7	Codigo de salida del Producto.
codigo_entrada	char	7	Codigo de la entrada de Producto.
codigo_prod	char	8	Codigo del producto.
fecha_caducida	smalldatetime	4	Fecha en que caduca el Producto.
numlote	integer	4	Lote del Producto ingresado.

Tabla 9: Salida_Almacen

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_bodega	char	4	Código de la Bodega.
codigo_salida	char	7	Código de la Salida de Bodega.
documento	varchar	15	Documento de apoyo de la Salida.
tipo_doc	char	2	Tipo de Documento (Requisita/Salida de Almacén "RQ", Traslado de Bodega "TA").
fecha_salida	smalldatetime	4	Fecha de Salida.
movimiento	varchar	10	Tipo de movimiento (ENTRADA, SALIDA O AJUSTE).
salidatotal	Numérico	Real	Total del costo de la Salida.

Tabla 10: Detalle_Salida

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_salida	char	7	Código de Salida.
codigo_prod	char	8	Código del Producto.
descripción	varchar	50	Nombre del producto.
unidad	varchar	15	Unidad de medida del Producto.
numlote	integer	4	Lote de donde se extrae el Producto
cantidad	real	4	Cantidad de Producto a dar de baja.
sub_total	real	4	Subtotal por Producto.

**Tabla 11: Ajustes Almacén**

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_bodega	char	4	Código de la Bodega.
codigo_ajuste	char	7	Código de Ajuste.
tipo_doc	char	2	Tipo de Documento (Ajuste al Físico "AF", Ajuste al Costo "AC", Ajuste a Ambos "AJ").
fecha_ajuste	smalldatetime	4	Fecha de Ajuste.
movimiento	varchar	10	Tipo de movimiento.

Tabla 12: Detalle Ajuste

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_Ajuste	char	7	Código de Ajuste.
codigo_prod	char	8	Código del Producto.
descripción	varchar	50	Nombre del producto.
unidad	varchar	15	Unidad de medida del Producto.
ajuste_costo	real	4	Valor del costo a ajustar.
ajuste_exis	real	4	Cantidad física a ajustar.
costo-unitario	real	4	Costo del Producto.
numlote	integer	4	Lote que se está ajustando.

Tabla 13: Factura

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_bodega	char	4	Código de la Bodega.
nfactura	char	8	Número de Factura.
fecha	smalldatetime	4	Fecha en que se emite la Factura.
estado	varchar	10	Factura cancelada o anulada.
codigo_cli	char.	6	Código del Cliente.
nombre_cli	varchar	50	Nombre del Cliente.
direccion	varchar	150	Dirección del Cliente.

Tabla 14: Detalle Factura

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
nfactura	char	9	Número de Factura.
codigo_prod	char	8	Código del Producto.
descripción	varchar	50	Nombre del producto.
unidad	varchar	15	Unidad de medida del Producto.
precio	real	4	Precio unitario del Producto.
cantidad	real	4	Cantidad de Producto a vender.
subtotal	real	4	Sub-total por Producto.

**Tabla 15: Cotización**

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
codigo_bodega	char	4	Código de la Bodega.
ncotizacion	char	9	Número de Cotización.
fecha	smalldatetime	4.	Fecha en que se emite la Cotización.
estado	varchar	10	Cotización válida o inválida.
codigo_cli	char.	6	Código del Cliente.
nombre_cli	varchar	50	Nombre del Cliente.
direccion	varchar	150	Dirección del Cliente.
telefono	char	8	Teléfono del Cliente.
importe_cotizacion	real	4	Suma de los Sub-totales de los Productos cotizados.
igv_cotizacion	real	4	Impuesto por los Productos Cotizados.
monto_cotizacion	real	4	Suma del Importe mas el IGV de la Cotización.

Tabla 16: Detalle_Cotizacion

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
ncotizacion	char	98	Número de Cotización.
codigo_prod	char	8	Código del Producto.
descripcion	varchar	50	Nombre del Producto.
unidad	varchar	15	Unidad de medida del Producto.
costo	real	4	Costo unitario del Producto.
precio	real	4	Precio del Producto cotizado.
cantidad	real	4	Cantidad del Producto Cotizado.
subtotal	real	4	Sub-total por Producto Cotizado.

Tabla 17: Contraseña

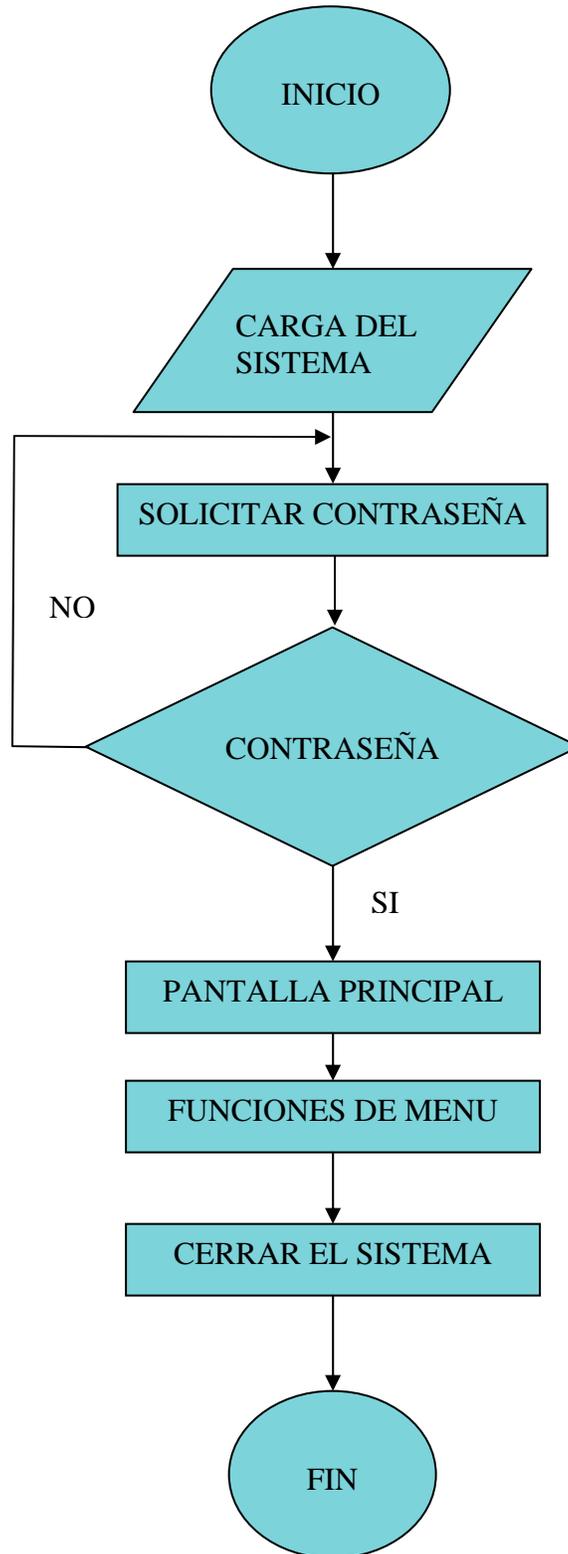
<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
nombre_usuario	varchar	30	Nombre de Usuario del Sistema.
contraseña	varchar	50	Contraseña del Usuario.

Tabla 18: FechaProceso

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Longitud</i>	<i>Descripción</i>
fecha_inicio	smalldatetime	4	Fecha de Inicio de Proceso del Control de Inventario.
fecha_fin	smalldatetime	4.	Fecha de Fin de Proceso del Control de Inventario.

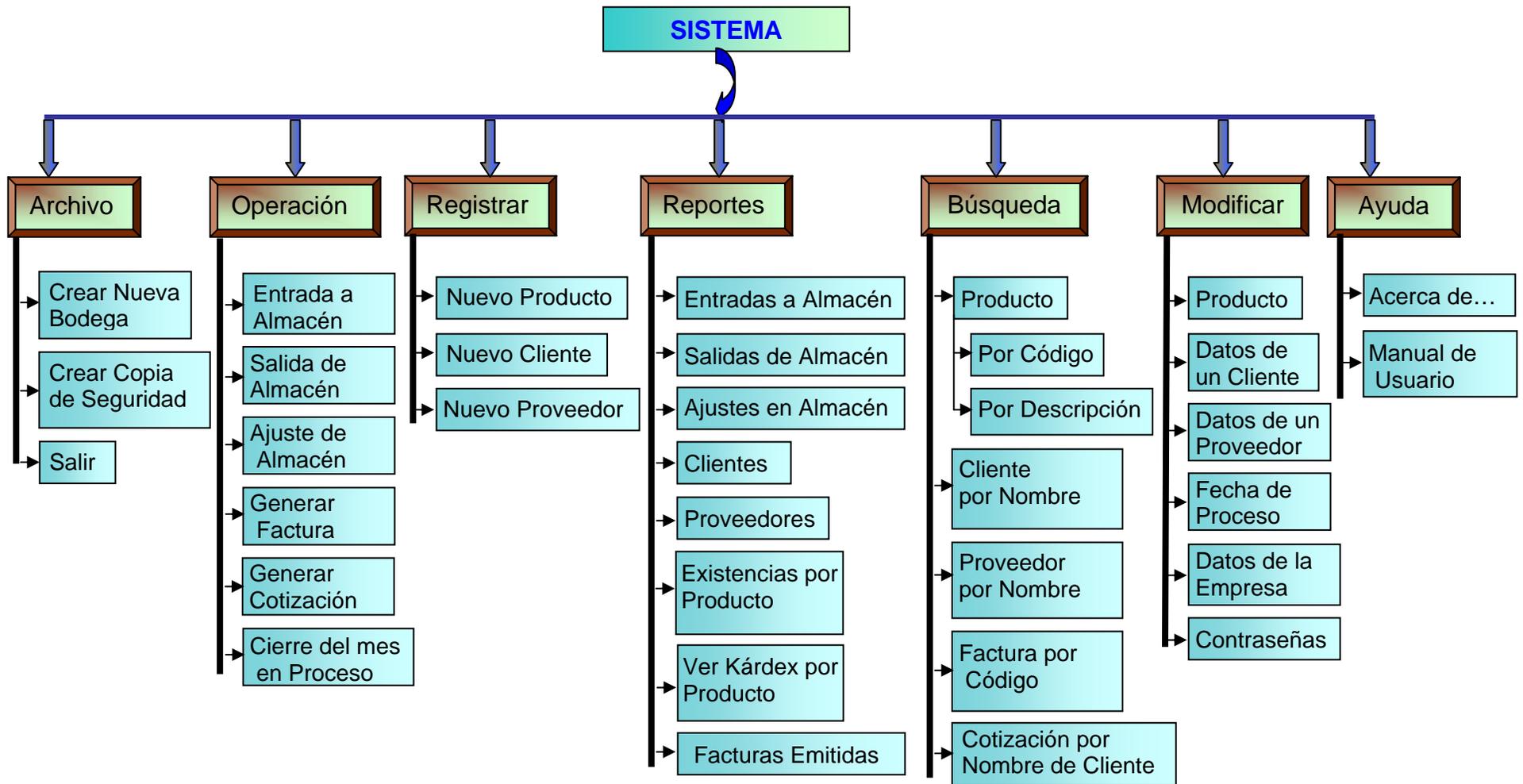


8.2. DISEÑO PROCEDIMENTAL





8.3. DISEÑO ARQUITECTÓNICO





8.4. DISEÑO DE INTERFAZ

Esta interfaz se utiliza para capturar la información general de la Empresa la cual se introducirá al ejecutar el sistema por primera vez y se almacenará en la base de datos la cual puede ser modificable. (**frmDatosEmpresa**)

Datos de la Empresa.

Datos de la Empresa.

Nombre de la Empresa Empresa Farmacéutica Grupo CAICS S.A.

Direccion De la alcaldía, 1/2 C al sur.

Ciudad Chinandega

Pais Nicaragua

Telefono 341-2350

Correo_Elec

Realizar Operaciones

Siguiete>> **Cancelar**

Ilustración 1: Datos de la Empresa.

Luego de introducir los datos de la empresa es necesario agregar las Bodegas al sistema con las cuales se trabajará, es obligatorio agregar al menos una Bodega, para ello basta con seleccionar **Nuevo** e introducir los datos de la bodega. (**frmBodega**)

Ejemplo

Numero de la Bodega: **B-02**.

Tipos de Productos contenidos: **PRODUCTOS LACTEOS**.

Agregar Bodega al Sistema...

Num. Bodega B-01

Tipos de Productos contenidos PRODUCTOS FARMACÉUTICOS

Cancelar **Nuevo** **Aceptar** **Siguiete>>**

Ilustración 2: Creación de Nueva Bodega.



La siguiente interfaz se utiliza para crear los usuarios del sistema, aquí se crearan dos tipos de usuarios de manera obligatoria con permisos diferentes, el **Administrador del Sistema** y la cuenta de **Usuario** con sus contraseñas correspondientes, para ello basta seleccionar **Nuevo**. (**frmUsuarios**)

Ilustración 3: Creación de los Usuarios.

Hasta este momento ya se ha completado la instalación del sistema y se procede a ejecutarlo. La siguiente interfaz muestra la carga del sistema gradualmente. (**frmCarga**)



Ilustración 4: Carga del Sistema.



A continuación se debe seleccionar en la interfaz el Usuario con el cual desea entrar al sistema, el boton cancelar del usuario y la bodega tienen la misma función y se explica una vez.
(frmContraseña)

The screenshot shows a dialog box with a blue title bar. Inside, there's a section for user selection with a dropdown menu showing 'ADMINISTRADOR'. Below it is a password field with a key icon and a masked password '*****'. At the bottom are two buttons: 'Aceptar' and 'Cancelar'.

Ilustración 5: Contraseña de acceso al sistema.

Esta interfaz le permitirá introducir el número de Bodega con la que desea entrar al sistema para realizar las operaciones relacionadas con la misma. **(frmBodegaintro)**

The screenshot shows a dialog box with a blue title bar. The main text is 'Bodega con la que se quiere Trabajar'. Below it is a label 'Num. Bodega:' followed by a text box containing 'B-01'. At the bottom are two buttons: 'Aceptar' and 'Cancelar'.

Ilustración 6: Bodega de trabajo en el sistema.

Esta interfaz se utilizara para establecer la fecha de proceso del Inventario. **(frmFechaProceso)**

The screenshot shows a dialog box with a blue title bar. The main text is 'Captura de la Fecha del mes en proceso del Inventario.'. Below it are two date input fields: 'Fecha Inicial:' with '17/09/2006' and 'Fecha Final:' with '16/10/2006'. At the bottom right is a button labeled 'Aceptar'.

Ilustración 7: Establecer fecha de proceso.



Esta es la interfaz principal a través de la cual se podrán realizar todas las operaciones competentes al control de inventario soportadas por este software las cuales podrán ejecutarse desplazándose por el menú de dicha interfaz. (**frmPrincipal**)



Ilustración 8: Ventana principal del sistema.



MENU ARCHIVO

Esta interfaz se utilizara para crear una copia de seguridad de la BBDD`s.
(frmCopia_Seguridad)



Ilustración 9: Respaldo de la base de datos.

Esta interfaz se utilizara para restaurar la base de datos que ha sido respaldada.



Ilustración 10: Restauración de la base de datos.



MENU OPERACIONES

Esta interfaz se utiliza para ingresar los Productos a bodega. (**frmEntradasAlmacen**)

Entradas de Productos a Almacen

Datos generales de la Entrada de Productos a Almacén.....

Código Bodega: B-01

Información de la Entrada...

Código: E-10001
Fecha: 17/09/2006
Documento: FAC-654321
Tipo de Documento: RR Rpte. Recepción Rpte. Traslado

Datos del Proveedor...

Código: P-1603
Nombre: DISTRIBUIDORA SAN AGUSTÍN. ...
Nuevo Prov

Detalles de la Entrada de Productos a Almacén.....

Código Producto: B01-1008 Cantidad: 100
Descripción: AMIKACINA 100MG **Nuevo** Costo Unitario: 0
Unidad: AMPOLLA Precio/Menor: 0
Precio/Mayor: 0
Sub Total: 0.00
Fecha de Caducidad: //

Operaciones sobre los Productos...

Agregar **Eliminar**

Producto	Unidad	Cantidad	Costo Unit.	Sub-Total
ACETAMINOFEN 500 MG	TABLETA	1000	0.32	320.00

Operaciones sobre la Entrada a Almacén...

Guardar **Cancelar** **Salir** **Total de la Entrada** 320.00

Reporte de la Entrada de Productos al Almacén

Imprimir la Entrada al Almacén para crear la etiqueta con el numero de Lote que el sistema ha asignado al Producto que se ha registrado en la Base de Datos.

Imprimir Entrada
Imprimir

Ilustración 11: Entrada de productos al almacén.



La siguiente interfaz se utilizara para dar de baja a los productos contenidos en una determinada bodega. (frmSalidasAlmacen)

Salida de Productos de Almacén...

Datos generales de la Salida de Productos de Almacén.....

Información de la Salida... Código Bodega: B-01

Código: S-97441 Documento: Ninguno

Fecha: 18/09/2006 Tipo de Documento: RQ

Detalles de la Salida de Productos de Almacén.....

Código Producto: B01-1002 Existencia: 60

Descripción: AZITROMICINA 600 V

Unidad: FRASCO Cantidad: 0

Operaciones sobre los Productos...

Agregar **Eliminar**

Sub Total: 0.00

Producto	Unidad	Cantidad	Sub-Total
ACETAMINOFEN 500 MG	TABLETA	100.00	32.00
AMIKACINA 100MG	AMPOLLA	10.00	269.20
AMBROXOL 7.05 ML	FRASCO GOTER	10.00	336.40

Total de la Salida: 637.60

Operaciones sobre la Salida de Almacén...

Guardar **Cancelar** **Imprimir** **Salir**

Ilustración 12: Salida de productos del almacén.



Esta interfaz se utiliza para realizar ajustes de los productos contenidos en una determinada bodega. (**frmAjusteAlmacen**)

Ajustando el Almacén...

Datos generales del Ajuste en Almacén..... Código Bodega: B-01

Información del Ajuste ...

Código: A-10000 Tipo de Documento: AA Fecha: 18/09/2006

Detalles del Ajuste en Almacén.....

Código Producto: B01-1002 Existencia: 60 Nueva Existencia: 59

Descripción: AZITROMICINA 600 V Costo: 213.5 Nuevo Costo: 0

Unidad: FRASCO Saldo: 12810

Operaciones sobre los Productos...

Agregar **Eliminar**

Producto	Unidad	Cant_Ajustar	Costo_Ajustar
ACETAMINOFEN 500 MG	TABLETA	-20.00	0.03
AMIKACINA 100MG	AMPOLLA	-4.00	0.08

Operaciones el Ajuste de Almacén...

Guardar **Cancelar** **Imprimir** **Salir**

Ilustración 13: Ajuste de productos en almacén.



Esta interfaz se utiliza para emitir una Factura a un cliente al realizar una venta. (frmFactura).

Generación de una Factura para emitirla a un Cliente...

Datos generales de la Factura para emitirla a un Cliente...

Datos del Cliente...

Código: Nombre: ...

Dirección: Teléfono:

Bodega:

Inf. de la Factura...

Código: Fecha:

Gracias por Preferirnos.

Detalles de la Factura.....

Código Producto: Descripción: Precio:

Unidad:

Existencia:

Cantidad:

Sub Total:

Operaciones sobre los Productos...

Tipo de Precio....

Precio al por Menor.
 Precio al por Mayor.

Producto	Unidad	Cantidad	Precio	Sub-Total
ACETAMINOFEN 500 MG	TABLETA	20.00	0.55	11.00
AMBROXOL 7.05 ML	FRASCO GOTER	10.00	39.42	394.20
AMIKACINA 100MG	AMPOLLA	15.00	31.20	468.00

Operaciones sobre la Entrada a Almacén...

Importe:

IGV:

Monto Factura:

Ilustración 14: Factura del cliente.



Esta interfaz se utiliza para generar una Cotización solicitada por un cliente.
(frmCotizacion)

Generación de una Cotización realizada por un Cliente...

Datos generales de la Cotización realizada por un Cliente...

Nº de Bodega: B-01

Datos del Cliente...

Código: C-2756 Nombre: FRANCISCO JOSÉ FLORES MARTIN ... **Nuevo**

Dirección: CHINANDEGA, DE ENITEL, 1C AL SUR Teléfono: 821-1487

Inf. de la Cotización...

Código: COT-31842 Fecha: 19/09/2006

Gracias por Preferirnos.

Detalles de la Cotización.....

Código Producto: B01-1002 Descripción: AZITROMICINA 600 V Precio: 250

Unidad: FRASCO

Existencia: 60

Cantidad: 10

Sub Total: 2,500.00

Operaciones sobre los Productos...

Agregar **Eliminar**

Tipo de Precio...

Precio al por Menor.

Precio al por Mayor.

Producto	Unidad	Cantidad	Precio	Sub-Total
ACETAMINOFEN 500 MG	TABLETA	300.00	0.55	165.00
AMBROXOL 7.05 ML	FRASCO GOTER	10.00	39.42	394.20
AMIKACINA 100MG	AMPOLLA	20.00	31.20	624.00

Operaciones sobre la Entrada a Almacén...

Guardar **Cancelar** **Imprimir** **Salir**

Importe: 1,183.20

ICV: 0 0.00

Monto Cotización: 1,183.20

Ilustración 15: Cotización del cliente.



Esta Interfaz se utiliza para cerrar el mes de Proceso del Inventario. (**frmCierreMes**)

Establecer Cierre del mes en Proceso de Inventario.

Cerrar Mes de Datos del Inventario.

El sistema actualmente lleva el Inventario correspondiente del 31/08/2006 al 02/09/2006

Establecer cierre del mes

Una vez que haya realizado el cierre correspondiente al mes en proceso del Inventario, el programa se cerrará y cuando lo ejecute nuevamente, se mostrará el formulario para establecer la fecha de proceso de Inventario del siguiente mes en proceso.

Cierre del Mes

Establecer

Cancelar

Ilustración 16: Cierre del mes en proceso.

MENU REGISTRAR

Esta interfaz se utiliza para Registrar un nuevo Producto en la bodega. (**frmProducto**)

Nuevo Producto

Datos del Nuevo Producto

Codigo de Bodega: B-01

Código de Producto: B01-4218

Descripcion: AMBROXOL 7.05 ML

Unidad: FRASCO GOTERO

Moneda: C

Grupo: MEDICINA

Sub_Grupo:

Ubicación: ESTANTE # 03

Caducidad del Producto:

Tiene Vencimiento.

Avisar días antes del vencimiento

10

Realizar Operación:

Nuevo Aceptar Cancelar

Ilustración 17: Registrar nuevo producto.



En las dos siguientes interfaces la función del botón cancelar es la misma y se muestra una sola vez.

Esta interfaz se utiliza para agregar un nuevo Cliente al sistema. (**frmCliente**)

Nuevo Cliente

Datos del Nuevo Cliente

Código: C-1613

Nombre: JAVIER ANTONIO ROSTRAN

Dirección: CHICHIGALPA, DE LA ALCALDIA 2 C AL OESTE

Cédula: 084-121079-0000X

Teléfono: 687-2563

Realizar Operación

Aceptar Cancelar

Ilustración 18: Registrar un nuevo cliente.

Esta interfaz es utilizada para ingresar un nuevo Proveedor al sistema. (**frmProveedor**)

Nuevo Proveedor

Datos del Nuevo Proveedor.

Código: P-7826

Nombre: FARMACIA SANGRE DE CRISTO

Dirección: CHINANDEGA, COSTADO NORTE DEL BISNE

Teléfono: 886-0429

Realizar Operación

Aceptar Cancelar

Ilustración 19: Registrar un nuevo proveedor.



MENU REPORTES

El código para el botón **Cancelar** en los REPORTES solo se pondrá una vez ya que tienen la misma funcionalidad (descargar el formulario).

Esta interfaz se utiliza para generar reporte de las entradas de Productos a Bodega en un rango de fecha determinado. (**frmReporte_Entrada**)

Generar reporte de las Entradas a Almacén.

Introduzca un rango de fecha

Inicial *Final*

17/10/2006 19/10/2006

Tipo de Documento: RR

Opciones de Reporte.

Imprimir Visualizar Cancelar

Ilustración 20: Reporte de entrada.

Esta interfaz se utiliza para generar Reporte de todas las Salidas de una Bodega en un rango de fecha determinado. (**frmReporte_Salida**)

Generar reporte de las Salidas de Almacén.

Introduzca un rango de fecha

Inicial *Final*

17/10/2006 19/10/2006

Tipo de Documento: RQ

Opciones de Reporte.

Imprimir Visualizar Cancelar

Ilustración 21: Reporte de salida.



Esta interfaz se utiliza para generar Reporte de todos los Ajustes realizados en una Bodega en un rango de fecha determinado. (**frmReporte_Ajuste**)

Generar reporte de los Ajustes en Almacén.

Introduzca un rango de fecha

Inicial *Final*

17/10/2006 19/10/2006

Tipo de Documento: AF

Opciones de Reporte.

Imprimir Visualizar Cancelar

Ilustración 22: Reporte de ajuste.



Esta interfaz se utiliza para mostrar todos los movimientos que se llevaron a cabo sobre un determinado producto en un rango de fecha dado . (**frmKardex**)

Kardex por Producto

Detalle del Producto

Nº de Bodega: **B-01** Unidad: **TABLETA**

Cod. del Producto: **B01-1003** Rango de fechas del mov. del Producto

Descripción: **ACETAMINOFEN 500 MG** Fecha inicial: **17/10/2006** Fecha final: **17/10/2006**

Entradas a Almacén.

Código Entrada	Fecha Entrada	Cantidad	Costo Unitario	Sub Total
E-10000	17/10/2006	2,000.00	0.23	460.00

Salidas de Almacén.

Código Salida	Fecha Salida	Cantidad	Costo Unitario	Sub Total
S-25384	17/10/2006	500.00	0.23	115.00

Ajustes en Almacén.

Código Ajuste	Fecha Ajuste	Cantidad Ajustada	Costo Ajustado	Costo Unit.	Sub Total
A-10000	17/10/2006	-100.00		0.23	-23.00
A-10001	17/10/2006		0.07	0.30	
A-10002	17/10/2006	-100.00	0.10	0.40	-40.00

Existencia final.

Cantidad: **1,300.00** Costo Unitario: **0.40** Saldo Total: **520.00**

 Salir

Ilustración 23: Kardex por producto.



Esta interfaz se utiliza para generar Reporte de todas las Facturas emitidas a los Cliente en un rango de fecha determinado. (**frmReporte_Factura**).

Generar reporte de las Facturas emitidas.

Introduzca un rango de fecha

Inicial *Final*

17/10/2006 19/10/2006

Opciones de Reporte.

Imprimir Visualizar Cancelar

Ilustración 24: Reporte de factura.



MENU BUSQUEDAS

Para el botón **Salir** de las vistas de Búsqueda se agregó una sola vez ya que tienen la misma funcionalidad. (**frmBusProd**)

Esta interfaz se utiliza para la búsqueda de Productos de dos maneras:

- ❖ Por descripción del Producto.
- ❖ Por código del producto.



Ilustración 25: Búsqueda de producto.

La siguiente interfaz es invocada por el botón mostrar donde se muestra el producto seleccionado. (**frmVerProducto**)



Ilustración 26: Muestra del producto.



Esta interfaz se utiliza para mostrar una lista de Clientes de la Empresa. (**frmBusCliente**)

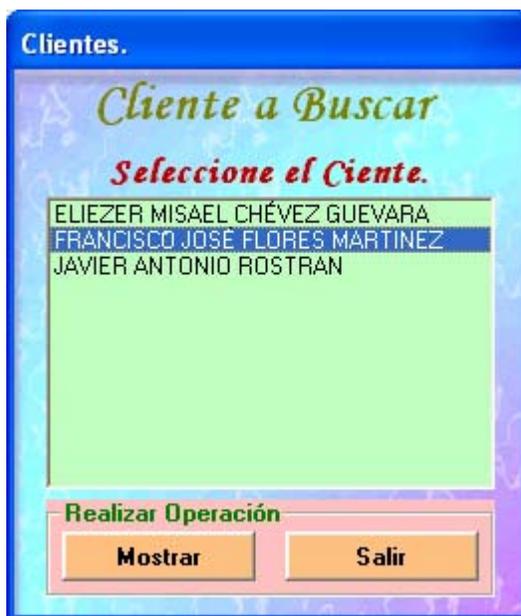


Ilustración 27: Búsqueda de cliente.

Esta interfaz es invocada por el botón mostrar donde se muestra el cliente seleccionado con sus datos. (**frmVerCliente**)



Ilustración 28: Muestra del cliente.



Esta interfaz se utiliza para buscar los Proveedores de la Empresa. (**frmBusProveedor**)



Ilustración 29: Búsqueda de proveedor.

Esta interfaz es invocada por el botón mostrar donde se muestra el proveedor seleccionado con sus datos. (**frmVerProveedor**)



Ilustración 30: Muestra del proveedor.



Esta interfaz se utiliza para buscar las Facturas emitidas durante la fecha del mes en proceso del Inventario. (**frmBusFacturas**)



Ilustración 31: Búsqueda de factura.

Esta interfaz es invocada por el botón mostrar donde se muestra la factura y sus datos correspondiente al código de factura seleccionado. (**frmVerFactura**)



Ilustración 32: Muestra de la factura.



La siguiente interfaz es invocada por el botón anular factura únicamente si se quiere anular una factura con permisos restringidos y se debe autorizar con la clave de administrador, o también para realizar otras operaciones.



Ilustración 33: Contraseña de administrador.

Esta interfaz se utiliza para buscar las cotizaciones validas emitidas a los Clientes de acuerdo a su fecha de .emisión. (**frmBusCotizacion**)



Ilustración 34: Búsqueda de cotización.



Esta interfaz es invocada por el botón mostrar donde se muestra la cotización correspondiente al Cliente que la solicitó en una fecha determinada. (**frmVerCotización**)

Información de Cotización...

Datos generales de la Cotización realizada por un Cliente...

Datos del Cliente...

Código: C-1613 Nombre: JAVIER ANTONIO ROSTRAN Teléfono: 687-2563

Dirección: CHICHIGALPA, DE LA ALCALDIA 2 C AL OESTE

Bodega: B-01

Inf. de la Cotización...

Código: COT-13172 Fecha: 19/09/2006

Detalles de la Cotización.....

Producto	Unidad	Cantidad	Precio	Sub-Total
ACETAMINOFEN 500 MG	TABLETA	250.00	0.55	137.50
AMBROXOL 7.05 ML	FRASCO GOTER	10.00	39.42	394.20
AZITROMICINA 600 V	FRASCO	20.00	250.00	5,000.00

Operaciones...

Facturar **Cerrar**

Importe: 5,531.70

IGV: 829.76

Monto Cotización: 6,361.46

Ilustración 35: Muestra de la cotización.



MENU MODIFICAR

Esta interfaz se utiliza para mostrar los productos existentes en la bodega en que se encuentran. (**frmListaProductos**)



Ilustración 36: Lista de productos.

Esta interfaz se utiliza para la modificación de los Productos existentes en Bodega, es invocada por el boton **Aceptar** de la interfaz anterior. (**frmModProducto**)



Ilustración 37: Modificar producto.



En el menú **Modificar** el código del botón **Cancelar** en todos los formularios de las listas y los otros es similar y se muestra una sola vez.

Esta interfaz se utiliza para modificar la información de un Cliente. (**frmModCliente**)

Cliente

Modificar Datos del Cliente

Código: C-2756

Nombre: FRANCISCO JOSÉ FLORES MART

Dirección: CHINANDEGA, DE ENITEL, 1C AL SUR

Cédula: 084-240880-0000P

Teléfono: 821-1487

Realizar Operación

Ilustración 38: Modificar cliente.

El botón **Buscar** invoca a esta interfaz y se selecciona un cliente para mostrar sus datos y luego modificarlos en el formulario de llamada. (**frmListaCliente**)

Lista de los Clientes de la Empresa...

Clientes

Seleccione un Cliente.

FRANCISCO JOSÉ LEYTON CALDERÓN
JUANA DEL SOCORRO ROSTRAN
MARÍA DEL CARMEN POZO TOVAL.

Realizar Operación

Ilustración 39: Lista de clientes.



Esta interfaz se utiliza para modificar la información de los Proveedores de la empresa.
(frmModProveedor)

Proveedor

Modificar Datos del Proveedor

Código: P-1603

Nombre: DISTRIBUIDORA SAN AGUSTÍN. **Elegir >>**

Dirección: COSTADO OESTE DEL MERCADO CENTRAL,
CHINANDEGA.

Teléfono: 343-2565

Realizar Operación

Modificar **Cancelar**

Ilustración 40: Modificar proveedor.

El botón **Elegir** de la vista anterior invoca a la siguiente interfaz y se selecciona un Proveedor para mostrar sus datos y luego modificarlos en el formulario en el cual se hizo la llamada.
(frmListaProveedores)

Lista de los Proveedores de la Empresa.

Proveedores

Seleccione un Proveedor.

DISTRIBUIDORA SAN AGUSTÍN.
FARMACIA SANGRE DE CRISTO

Realizar Operación

Aceptar **Cancelar**

Ilustración 41: Lista de proveedores.



Esta interfaz se utiliza para modificar la fecha de finalización de Proceso del Inventario.
(frmModFechaProceso)

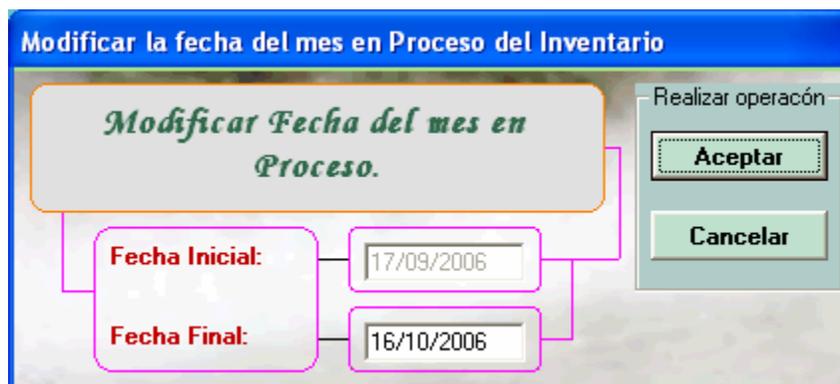


Ilustración 42: Modificar fecha de proceso.

Esta interfaz se utiliza para Modificar los datos de la Empresa existentes en la BBDD`s.
(frmModDatosEmpresa)



Ilustración 43: Modificar datos de la empresa.



Esta interfaz se utiliza para modificar las contraseñas de los Usuarios del Sistema. (frmModUsuarios)

Ilustración 44: Modificar contraseña.

MENU AYUDA

Esta interfaz muestra la versión del software y los autores del mismo. (frmAcercade)

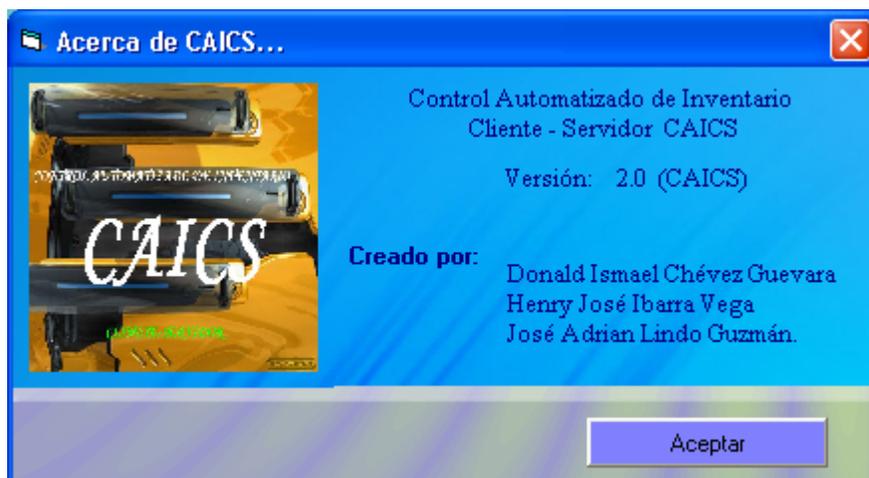


Ilustración 45: Acerca de.



Esta interfaz se utiliza para mostrar la ayuda sobre como funciona el software. Muestra cada uno de los temas de ayuda, y usted podrá acceder solo dando click sobre el que desee, para ver los resultados, no se hizo una codificación solamente un enlace a una pagina creada en front page. (frmManualUsuario)

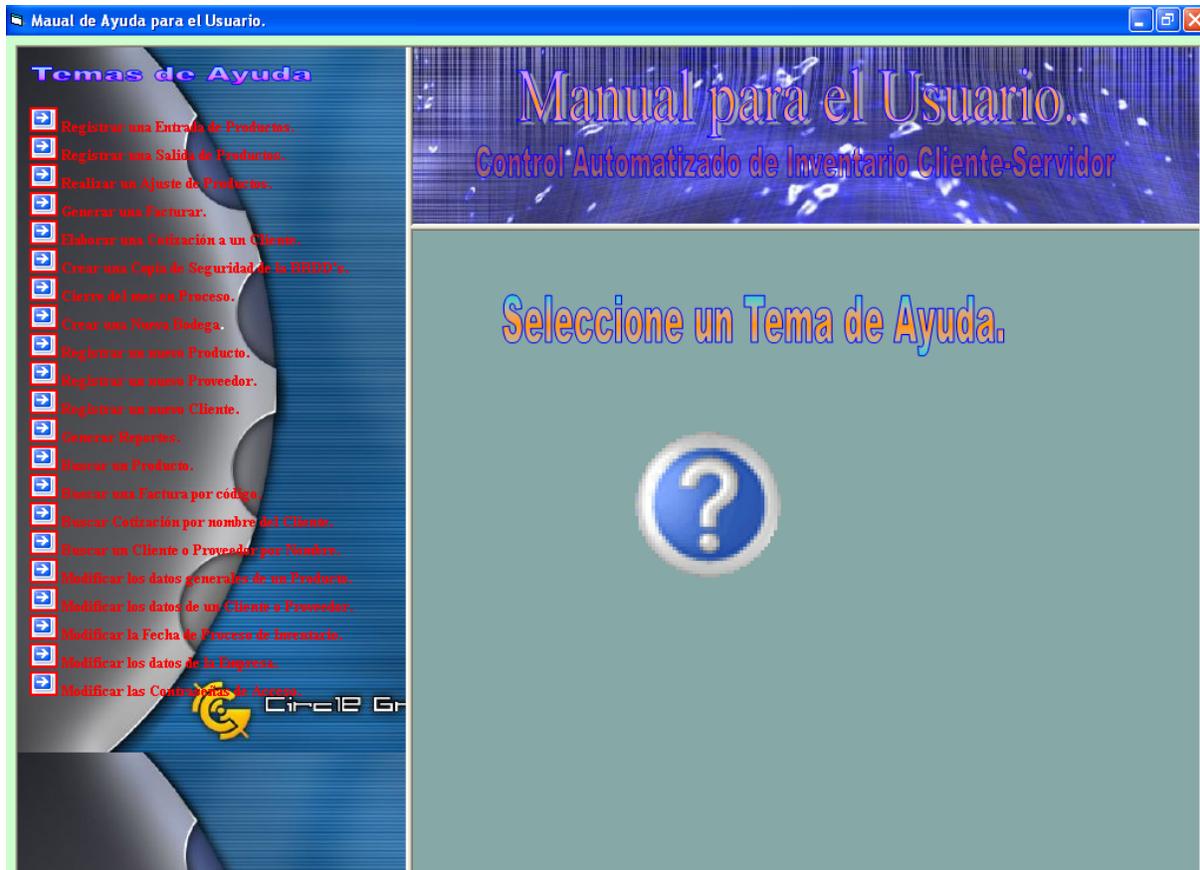


Ilustración 46: Manual de usuario.



Cuando se hayan ingresado datos y el sistema este en funcionamiento ocurrirán los casos que se muestran a continuación.

Cuando se selecciona un número de Bodega con el cual trabajar, se hace una consulta en dicha bodega para ver si hay productos próximos a vencer y mostrárselos al usuario del Sistema.

Esta interfaz muestra todos los productos contenidos en una Bodega que están con una fecha cercana de vencimiento. (**frmCadcercana**)

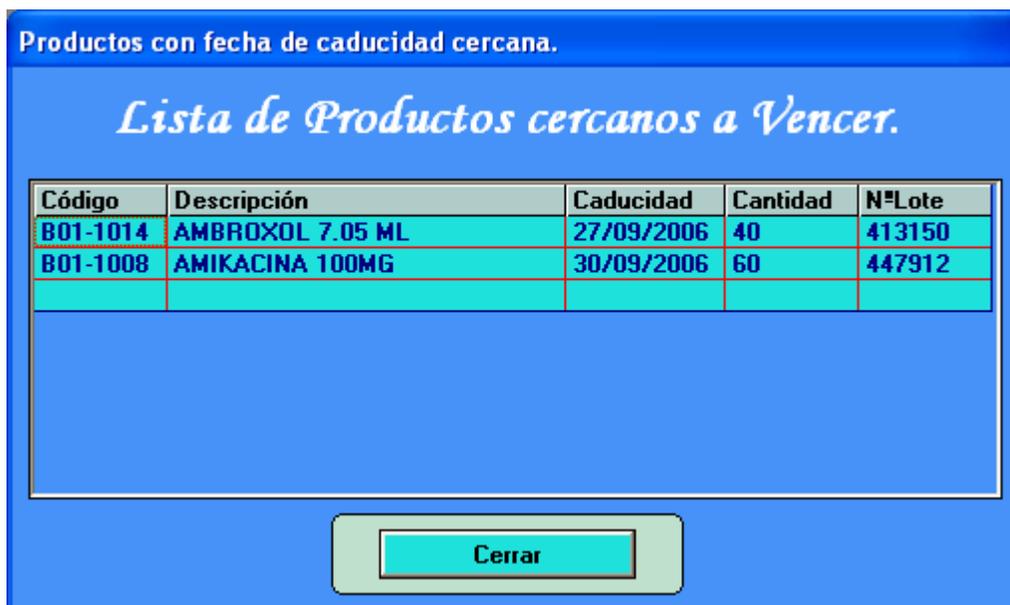


Ilustración 47: Productos cercanos a caducar.

Cuando se ha seleccionado un número de Bodega se hace una consulta en dicha bodega para verificación de la fecha de vencimiento de los productos.

Esta interfaz muestra todos los productos de la Bodega actual que ya se han vencidos hasta la fecha. (**frmProdVencidos**)



Ilustración 48: Productos vencidos.



IX. CONCLUSIONES

Manifestamos de manera placentera que hemos satisfecho nuestros objetivos planteados en la ejecución de este proyecto, al cual llegamos cumpliendo las etapas del ciclo de vida clásico, además, se cumplió con las metas generales y otras mas específicas las cuales nos ayudaron a desarrollar con mas profundidad y coherencia este software tomando en cuenta no solo el sentido lógico de programación, sino la parte contable tanto superficiales como internas.

En la actualidad los sistemas de control de inventario que utilizan las empresas, algunos no poseen interfaz gráfica y muchos de estos son desarrollados y utilizados por una empresa en específico. CAICS, es un sistema con interfaz gráfica, el cual facilitará a los usuarios el manejo del mismo, es decir, es un software sencillo, rápido y eficaz que podrá ser utilizado por cualquier empresa donde las operaciones que se llevan a cabo dentro de la misma se adapten a este.

Este Sistema permitirá mantener una organización integral de la estructura de la que forma parte; controlar el inventario de una empresa mediante este sistema proporcionará información actualizada en el momento que se desee y se tomaran decisiones correctas en cuanto a compra o venta se refiera.

CAICS es una aplicación que funciona en una red. Además, se ha comprobado mediante multiples accesos a la base de datos de manera simultanea, que éste sistema permite agilizar las operaciones que se llevan a cabo dentro de una empresa manteniendo la integridad y consistencia de la base de datos por ser éste un sistema de transacciones



X. RECOMENDACIONES

Es necesario que todos aquellos usuarios que vayan a utilizar este software, adquieran conocimientos previos sobre el sistema, ya sea a través de la ayuda del programa o mediante un facilitador (alguien con conocimiento del mismo), todo esto se debe realizar para el buen funcionamiento o ejecución de la aplicación.

Debido a que este sistema funciona en Red, es necesario que se revisen las conexiones físicas de todos los equipos de la red donde se encuentre instalada la aplicación, además, debe estar instalado el servidor de base de datos SQL-Server en uno de los computadores de la red con todas las configuraciones pertinentes.



XI. ANEXOS

11.1. SCRIPT DE LA CREACIÓN DE LAS TABLAS DE LA BASE DE DATOS

TABLA Empresa

```
Create Table Empresa
(
nombreE varchar(50) NOT NULL,
direccionE varchar(150) NOT NULL,
ciudad varchar(20) NOT NULL,
pais varchar(20) NOT NULL,
telefono_o_celular char(8) NOT NULL,
    CONSTRAINT REGLATELEFONOCELULAR CHECK
    (telefono_o_celular LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]'),
correo_elect varchar(50)
)
```

TABLA Cliente

```
Create Table Cliente
(
codigo_cli char(6) NOT NULL,
    CONSTRAINT REGLACODIGOCLIENTE CHECK
    (codigo_cli LIKE '[A-Z][-][0-9][0-9][0-9][0-9]'),
nombre_cli varchar(50) NOT NULL,
cedula char(16) ,
    CONSTRAINT REGLACEDULA CHECK
    (cedula LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9][0-9][0-9][0-9][-]
[0-9][0-9][0-9][0-9][A-Z]'),
direccion varchar(150),
telefono char(8),
    CONSTRAINT REGLATELEFONO CHECK
    (telefono LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]'),

PRIMARY KEY (codigo_cli)
)
```

TABLA Proveedor

```
Create Table Proveedor
(
codigo_prov char(6) NOT NULL,
    CONSTRAINT REGLACODIGOPROVEEDOR CHECK
    (codigo_prov LIKE '[A-Z][-][0-9][0-9][0-9][0-9]'),
nombre_prov varchar(50) NOT NULL,
direccion varchar(150),
telefono char(8),
    CONSTRAINT REGLATELEFONOPROVEEDOR CHECK
    (telefono LIKE '[0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]'),
PRIMARY KEY (codigo_prov)
)
```



TABLA Bodega

```
Create Table Bodega
(
codigo_bodega char(4) NOT NULL,
CONSTRAINT REGLACODIGOBODEGA CHECK
(codigo_bodega LIKE '[A-Z][-][0-9][0-9]'),
descripcion_bod varchar(50) NOT NULL,

PRIMARY KEY (codigo_bodega)
)
```

TABLA Producto

```
Create Table Producto
(
codigo_bodega char(4) NOT NULL,
codigo_prod char(8) NOT NULL,
CONSTRAINT REGLACODIGOPRODUCTO CHECK
(codigo_prod LIKE '[A-Z][0-9][0-9][-][0-9][0-9][0-9][0-9]'),
descripcion varchar(50) NOT NULL,
unidad varchar(15) NOT NULL,
exis_final real NOT NULL DEFAULT 0,
costo_unitario real NOT NULL DEFAULT 0,
saldo real NOT NULL DEFAULT 0,
precio1 real NOT NULL DEFAULT 0,
precio2 real NOT NULL DEFAULT 0,
tiene_caducidad bit NOT NULL DEFAULT 0,
tiempo_caducidad integer,
moneda char(1) NOT NULL DEFAULT 'C'
CONSTRAINT REGLAMONEDA CHECK (moneda IN ('C','$')),
grupo varchar(15) NOT NULL,
subgrupo varchar(15),
ubicacion varchar(50),

PRIMARY KEY (codigo_prod),
FOREIGN KEY(codigo_bodega) REFERENCES Bodega (codigo_bodega)
)
```



TABLA Entrada_Almacen

```
Create Table Entrada_Almacen
(
codigo_bodega char(4) NOT NULL,
codigo_entrada char(7) NOT NULL,
    CONSTRAINT REGLACODIGOENTRADA CHECK
    (codigo_entrada LIKE '[A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
documento varchar(15) NOT NULL,
tipo_doc char(2) NOT NULL DEFAULT 'RR',
    CONSTRAINT REGLATIPODOCUMENTOENTRADA CHECK (tipo_doc IN ('RR','TD')),
fecha_entrada smalldatetime,
codigo_prov char(6) NOT NULL,
    CONSTRAINT REGLACODIGOPROVEEDORENTRADA CHECK
    (codigo_prov LIKE '[A-Z][-][0-9][0-9][0-9][0-9]'),
nombre_prov varchar(50) NOT NULL,
movimiento varchar(10) NOT NULL,    DEFAULT 'ENTRADA',
    CONSTRAINT REGLAMOVIMIENTOENTRADA CHECK (movimiento IN ('ENTRADA')),
entradatotal real NOT NULL DEFAULT 0,

PRIMARY KEY (codigo_entrada),
FOREIGN KEY (codigo_prov) REFERENCES Proveedor (codigo_prov)
)
```

TABLA Detalle_Entrada

```
Create Table Detalle_Entrada
(
codigo_entrada char(7) NOT NULL,
    CONSTRAINT REGLACODIGOENTRADADETALLE CHECK
    (codigo_entrada LIKE '[A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
codigo_prod char(8) NOT NULL,
descripcion varchar(50) NOT NULL,
unidad varchar(15) NOT NULL,
numlote integer NOT NULL,
fecha_caducidad smalldatetime,
costo_unitario real NOT NULL,
cantidad real NOT NULL,
sub_total real NOT NULL,

FOREIGN KEY (codigo_entrada) REFERENCES Entrada_Almacen (codigo_entrada),
FOREIGN KEY (codigo_prod) REFERENCES Producto (codigo_prod)
)
```

TABLA Lote_Poducto

```
Create Table Lote_Poducto
(
codigo_bodega char(4) NOT NULL,
fecha_entrada smalldatetime,
codigo_salida char(7),
codigo_entrada char(7),
codigo_prod char(8) NOT NULL,
fecha_caducidad smalldatetime,
numlote integer NOT NULL,
cantidad_fecha real NOT NULL
)
```



TABLA Salida_Almacén

```
Create Table Salida_Almacen
(
codigo_bodega char(4) NOT NULL,
codigo_salida char(7) NOT NULL,
    CONSTRAINT REGLACODIGOSALIDA CHECK
    (codigo_salida LIKE '[A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
documento varchar(15) NOT NULL,
tipo_doc char(2) NOT NULL DEFAULT 'RQ',
    CONSTRAINT REGLATIPODOCUMENTOSALIDA CHECK (tipo_doc IN ('RQ','RT')),
fecha_salida smalldatetime,
movimiento varchar(10) NOT NULL DEFAULT 'SALIDA',
    CONSTRAINT REGLAMOVIMIENTOSALIDA CHECK (movimiento IN ('SALIDA')),
salidatotal real NOT NULL,

PRIMARY KEY (codigo_salida)
)
```

TABLA Detalle_Salida

```
Create Table Detalle_Salida
(
codigo_salida char(7) NOT NULL,
    CONSTRAINT REGLACODIGOSALIDADETALLE CHECK
    (codigo_salida LIKE '[A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
codigo_prod char(8) NOT NULL,
descripcion varchar(50) NOT NULL,
unidad varchar(15) NOT NULL,
numlote integer NOT NULL,
cantidad real NOT NULL,
sub_total real NOT NULL,

FOREIGN KEY (codigo_salida) REFERENCES Salida_Almacen (codigo_salida),
FOREIGN KEY (codigo_prod) REFERENCES Producto (codigo_prod)
)
```

TABLA Ajustes_Almacén

```
Create Table Ajustes_Almacen
(
codigo_bodega char(4) NOT NULL,
codigo_ajuste char(7) NOT NULL,
    CONSTRAINT REGLACODIGOAJUSTE CHECK
    (codigo_ajuste LIKE '[A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
tipo_doc char(2) NOT NULL DEFAULT 'AF',
    CONSTRAINT REGLATIPODOCUMENTOAJUSTE CHECK (tipo_doc IN ('AF','AC','AA')),
fecha_ajuste smalldatetime,
movimiento varchar(10) NOT NULL DEFAULT 'AJUSTE',
    CONSTRAINT REGLAMOVIMIENTOAJUSTE CHECK (movimiento IN ('AJUSTE')),

PRIMARY KEY (codigo_ajuste)
)
```



TABLA Detalle_Ajuste

```
Create Table Detalle_Ajuste
(
codigo_ajuste char(7) NOT NULL,
    CONSTRAINT REGLACODIGOAJUSTEDETALLE CHECK
        (codigo_ajuste LIKE '[A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
codigo_prod char(8) NOT NULL,
descripcion varchar(50) NOT NULL,
unidad varchar(15) NOT NULL,
ajuste_costo real,
ajuste_exis real,
costo_unitario real NOT NULL,
numlote integer,

FOREIGN KEY (codigo_ajuste) REFERENCES Ajustes_Almacen (codigo_ajuste),
FOREIGN KEY (codigo_prod) REFERENCES Producto (codigo_prod)
)
```

TABLA Factura

```
Create Table Factura
(
codigo_bodega char(4) NOT NULL,
nfactura char(9) NOT NULL,
    CONSTRAINT REGLANFACTURA CHECK
        (nfactura LIKE '[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
fecha smalldatetime,
estado varchar(10) NOT NULL DEFAULT 'CANCELADA',
    CONSTRAINT REGLAESTADOFACTURA CHECK (estado IN ('CANCELADA','ANULADA')),
codigo_cli char(6) NOT NULL,
nombre_cli varchar(50) NOT NULL,
direccion varchar(150) ,
telefono char(8),
importe_factura real NOT NULL ,
igv_factura real NOT NULL DEFAULT 0,
monto_factura real NOT NULL ,

PRIMARY KEY (nfactura),
FOREIGN KEY (codigo_cli) REFERENCES Cliente (codigo_cli)
)
```

TABLA Detalle_Factura

```
Create Table Detalle_Factura
(
nfactura char(9) NOT NULL,
    CONSTRAINT REGLANFACTURADETALLE CHECK
        (nfactura LIKE '[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
codigo_prod char(8) NOT NULL,
descripcion varchar(50) NOT NULL,
unidad varchar(15) NOT NULL,
precio real NOT NULL,
cantidad real NOT NULL,
subtotal real NOT NULL ,

FOREIGN KEY (nfactura) REFERENCES Factura (nfactura),
FOREIGN KEY (codigo_prod) REFERENCES Producto (codigo_prod)
)
```



TABLA Cotizacion

```
Create Table Cotizacion
(
codigo_bodega char(4) NOT NULL,
ncotizacion char(9) NOT NULL,
    CONSTRAINT REGLANCOTIZACION CHECK
    (ncotizacion LIKE '[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
fecha smalldatetime,
estado varchar(10) NOT NULL DEFAULT 'VALIDA',
    CONSTRAINT REGLAESTADOCOTIZACION CHECK (estado IN ('VALIDA','INVALIDA')),
codigo_cli char(6) NOT NULL,
nombre_cli varchar(50) NOT NULL,
direccion varchar(150),
telefono char(8),
importe_cotizacion real NOT NULL,
igv_cotizacion real NOT NULL DEFAULT 0,
monto_cotizacion real NOT NULL,

PRIMARY KEY (ncotizacion),
FOREIGN KEY (codigo_cli) REFERENCES Cliente (codigo_cli)
)
```

TABLA Detalle_Cotizacion

```
Create Table Detalle_Cotizacion
(
ncotizacion char(9) NOT NULL,
    CONSTRAINT REGLANCOTIZACIONDETALLE CHECK
    (ncotizacion LIKE '[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9][0-9]'),
codigo_prod char(8) NOT NULL,
descripcion varchar(50) NOT NULL,
unidad varchar(15) NOT NULL,
costo real NOT NULL,
precio real NOT NULL,
cantidad real NOT NULL,
subtotal real NOT NULL,

FOREIGN KEY (ncotizacion) REFERENCES Cotizacion (ncotizacion),
FOREIGN KEY (codigo_prod) REFERENCES Producto (codigo_prod)
)
```

TABLA FechaProceso

```
Create Table FechaProceso
(
fecha_inicio smalldatetime,
fecha_fin smalldatetime
)
```

TABLA Contraseña

```
Create Table Contraseña
(
nombre_usuario varchar(30) NOT NULL,
contraseña varchar(50) NOT NULL,

PRIMARY KEY (nombre_usuario)
)
```



11.2. ANEXO DE CÓDIGOS DE LAS INTERFACES

Código de la Ilustración 1: Datos de la Empresa.

Este código permite cancelar la instalación de la aplicación.

```
Private Sub cmdCancelar_Click()  
    If MsgBox("¿Desea realmente salir de la Instalación?", vbYesNo + vbDefaultButton1,  
"Cancelar la Instalación...") = vbYes Then  
        End  
    Else  
        Exit Sub  
    End If  
End Sub
```

Este código permite validar y guardar los datos de la empresa y continuar instalando.

```
Private Sub cmdSiguiente_Click()  
    On Error GoTo Error_Guardar  
    If Len(txtNombre.Text) = 0 Then  
        MsgBox "Falta el nombre de la Empresa.", , "Datos de la empresa."  
        txtNombre.SetFocus  
        Exit Sub  
    ElseIf Len(txtDireccion.Text) = 0 Then  
        MsgBox "Falta la dirección de la Empresa.", , "Datos de la empresa."  
        txtDireccion.SetFocus  
        Exit Sub  
    ElseIf Len(txtCiudad.Text) = 0 Then  
        MsgBox "Falta la ciudad de la Empresa.", , "Datos de la empresa."  
        txtCiudad.SetFocus  
        Exit Sub  
    ElseIf Len(txtPais.Text) = 0 Then  
        MsgBox "Falta el país de la Empresa.", , "Datos de la empresa."  
        txtPais.SetFocus  
        Exit Sub  
    ElseIf mskTelefono.Text = "###-####" Then  
        MsgBox "Falta el número de teléfono de la Empresa.", , "Datos de la empresa."  
        mskTelefono.SetFocus  
        Exit Sub  
    End If  
  
    If Not EstablecerConex Then  
        End  
    End If  
    rstEmpresa.Open "Empresa", cnnControlInv, adOpenStatic, adLockOptimistic,  
adCmdTable  
    With rstEmpresa  
        .AddNew  
        'Guardar el registro en la tabla Empresa.  
        !nombreE = txtNombre.Text  
        !direccionE = txtDireccion.Text  
        !ciudad = txtCiudad.Text  
        !pais = txtPais.Text  
        !telefono_o_celular = mskTelefono.Text
```



```
!correo_elect = txtCorreo.Text
'actualizar.
.Update
End UIT
'Cerrar el recordset y la conexión.
rstEmpresa.Close: Set rstEmpresa = Nothing
Call LiberarConex
Unload Me
frmBodega.Show
Exit Sub
Error_Guardar:
MsgBox "Error al registrar los Datos de la Empresa, verifique que estos esten
correctos.", , "Datos de la empresa."
Call LiberarConex
End Sub
```

Código de la Ilustración 2: Creación de Nueva Bodega.

Este procedimiento habilita y limpia los controles para la creación de bodegas.

```
Private Sub cmdNuevo_Click()
cmdAceptar.Enabled = True
cmdNuevo.Enabled = False
msknumero.Enabled = True
msknumero.Text = "_-_"
txtdescripcion.Enabled = True
txtdescripcion = ""
If cmdSiguiente.Caption = "Finalizar" Then cmdCancelar.Enabled = True
msknumero.SetFocus
End Sub
```

Este procedimiento valida y guarda los datos de la nueva bodega.

```
Option Explicit

Private Sub cmdAceptar_Click()
On Error GoTo Error_Aceptar
If msknumero.Text = "_-_" Then
MsgBox "Ingrese el numero de la bodega.", , "Error al guardar el # de Bodega."
msknumero.SetFocus
Exit Sub
ElseIf Len(Trim(txtdescripcion.Text)) = 0 Then
MsgBox "Ingrese la descripción de la bodega.", , "Error en la creación de la
Bodega."
txtdescripcion.SetFocus
Exit Sub
End If
If Not EstablecerConex Then
End
End If
rstBodega.Open "Bodega", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdTable
If cmdSiguiente.Caption = "Siguiente>>" Then
With rstBodega
.AddNew
'Guardar el registro en la tabla Bodega.
!codigo_bodega = msknumero.Text
```



```
        !descripcion_bod = UCase(Trim(txtdescripcion.Text))
        'intentar actualizar.
        .Update
    End With
    cmdAceptar.Enabled = False
    cmdNuevo.Enabled = True
    cmdSiguiente.Enabled = True
    msknumero.Enabled = False
    txtdescripcion.Enabled = False
    rstBodega.Close: Set rstBodega = Nothing
    cnnControlInv.Close: Set cnnControlInv = Nothing
    Exit Sub
ElseIf cmdSiguiente.Caption = "Finalizar" Then
    'Aqui se guardan los datos de las nuevas Bodegas creadas por el Administrador.
    With rstBodega
        .AddNew
        'Guardar el registro en la tabla Bodega.
        !codigo_bodega = msknumero.Text
        !descripcion_bod = UCase(Trim(txtdescripcion.Text))
        'intentar actualizar.
        .Update
    End With
    cmdAceptar.Enabled = False
    cmdNuevo.Enabled = True
    cmdSiguiente.Enabled = True
    msknumero.Enabled = False
    txtdescripcion.Enabled = False
    cmdCancelar.Enabled = False
    rstBodega.Close: Set rstBodega = Nothing
    Call LiberarConex
    Exit Sub
End If
Error_Aceptar:
    MsgBox "El número de la nueva Bodega que introdujo ya existe o no es correcto. ", ,
"Error al registrar los Datos..."
    msknumero.Text = "_-_"
    msknumero.SetFocus
    Call LiberarConex
End Sub
```

Este procedimiento cancela la creación de la nueva bodega y la instalación del sistema

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```



Este procedimiento permite finalizar la creación de bodegas y continúa instalando la aplicación.

```
Private Sub cmdSiguiete_Click()  
    If cmdSiguiete.Caption = "Finalizar" Then  
        Unload Me  
        Exit Sub  
    End If  
    frmUsuarios.Show  
    Unload Me  
End Sub
```

Este procedimiento continua con el siguiente paso si se esta instalando la aplicación o finaliza la creación de bodega.

```
Private Sub Form_Activate()  
    If cmdSiguiete.Caption = "Finalizar" Then  
        cmdCancelar.Enabled = True  
    ElseIf cmdSiguiete.Caption = "Siguiete>>" Then  
        cmdCancelar.Enabled = False  
    End If  
End Sub
```

Este procedimiento deshabilita los botones que no se necesitan en este momento.

```
Private Sub Form_Load()  
    cmdAceptar.Enabled = False  
    cmdSiguiete.Enabled = False  
    msknumero.Enabled = False  
    txtdescripcion.Enabled = False  
    'Establecer la conexion ala DB  
    Call EstablecerConex  
End Sub
```

Código de la Ilustración 3: Creación de los Usuarios.

Este procedimiento habilita los controles para introducir los datos de los usuarios y sus contraseñas.

```
Option Explicit  
Dim intUsr As Integer  
  
Private Sub cmdNuevo_Click()  
    cmdNuevo.Enabled = False  
    cmdAceptar.Enabled = True  
    cbxUsuario.Enabled = True  
    txtContraseña.Enabled = True  
    txtConfirmar.Enabled = True  
    txtContraseña.Text = ""  
    txtConfirmar.Text = ""  
End Sub
```



Este procedimiento no permite la introducción de cualquier carácter en el control.

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

Este procedimiento valida los datos de los usuario y los guarda en la base de datos.

```
Private Sub cmdAceptar_Click()
    Dim varUsuario, varContraseña, varConfirmar As String
    On Error GoTo Error_Guardar

    varUsuario = cbxUsuario.Text
    varContraseña = txtContraseña.Text
    varConfirmar = txtConfirmar.Text

    If Len(varUsuario) = 0 Then
        MsgBox " Falta el nombre del Usuario. ", vbInformation, "Error al ingresar el
Usuario..."
        Exit Sub
    ElseIf Len(varContraseña) = 0 Then
        MsgBox " Falta la Contraseña. ", vbInformation, "Error al ingresar la
Contraseña..."
        txtContraseña.SetFocus
        Exit Sub
    Else
        If varContraseña = varConfirmar Then
            'Establecer la conexión a la Base de Datos.
            If Not EstablecerConex Then
                End
            End If
            'Abrir la tabla Empresa para grabar la información.
            rstUsuarios.Open "Contraseña", cnnControlInv, adOpenDynamic,
adLockOptimistic
            With rstUsuarios
                .AddNew
                'Guardar el registro en la tabla Contraseña.
                !nombre_usuario = cbxUsuario.Text
                !contraseña = txtContraseña.Text
                'intentar actualizar.
                .Update
            End With
            'Cerrar el recordset y la conexión.
            Call LiberarConex
            'Verificar que en el sistema solo existan 2 cuentas de usuarios
            If intUsr < 2 Then
                intUsr = intUsr + 1
                cmdNuevo.Enabled = True
                cmdAceptar.Enabled = False
                cbxUsuario.Enabled = False
                txtContraseña.Enabled = False
                txtConfirmar.Enabled = False
            Else
                cmdFinalizar.Enabled = True
                cmdAceptar.Enabled = False
                cbxUsuario.Enabled = False
            End If
        End If
    End If
End Sub
```



```
        txtContraseña.Enabled = False
        txtConfirmar.Enabled = False
    End If
    Exit Sub
Else
    MsgBox " Las contraseñas no coinciden. ", vbInformation, "Error al ingresar
la contraseña..."
    txtContraseña.Text = ""
    txtConfirmar.Text = ""
    cmdAceptar.Enabled = True
    Exit Sub
End If
End If
Error_Guardar: 'En caso de generarse un error al guardar el registro
                'saltar a la siguiente linea.
    MsgBox "Ya existe un usuario con ese nombre. ", vbInformation, "Error al registrar
los Datos del usuario..."
    cbxUsuario.Text = "USUARIO"
    txtContraseña.Text = ""
    txtConfirmar.Text = ""
    Call LiberarConex
End Sub
```

Este procedimiento finaliza la instalacion y la creación de los usuarios.

```
Private Sub cmdFinalizar_Click()
    MsgBox " La Instalación ha finalizado satisfactoriamente. ", vbInformation,
"Mensaje de Instalación..."
    Unload Me
End Sub
```

Código de la Ilustración 4: Carga del Sistema.

Este procedimiento inicializa la semilla con la funcion random para la carga del sistema.

```
Option Explicit
Dim RndNum As Byte

Private Sub Form_Load()
    Randomize Timer
End Sub
```

Este procedimiento carga la aplicación si esta instalada, de lo contrario la instala.

```
Private Sub tmrTimer_Timer()
    RndNum = Rnd * 4
    cpbProgress.Value = cpbProgress.Value + RndNum
    lblPercentage = CStr(cpbProgress.Value \ 2) & "%" 'convierte valores a cadena
    If cpbProgress.Value = 200 Then
        'Establecer la conexión a la Base de Datos.
        If Not EstablecerConex Then
            End
        End If
        'Abrir la tabla Empresa.
        Set rstEmpresa = New ADODB.Recordset
```



```
rstEmpresa.Open "Empresa", cnnControlInv, adOpenDynamic, adLockOptimistic,
adCmdTable
'Ver si ya existe información de la Empresa.
If rstEmpresa.RecordCount > 0 Then
    rstEmpresa.Close: Set rstEmpresa = Nothing
    Call LiberarConex
    Unload Me
    frmContraseña.Show
Else
    rstEmpresa.Close: Set rstEmpresa = Nothing
    Call LiberarConex
    Unload Me
    'Registrar la información de la Empresa en la Base de Datos.
    frmDatosEmpresa.Show
End If
End If
End Sub
```

Código de la Ilustración 5: Contraseña de acceso al sistema.

Este procedimiento carga en un control las cuentas creadas para que el usuario seleccione con cual entrar al sistema.

```
Private Sub Form_Load()
    Dim i As Integer
    i = 0
    If Not EstablecerConex Then
        End
    End If
    rstUsuarios.Open "SELECT nombre_usuario FROM Contraseña", cnnControlInv,
adOpenStatic, adLockReadOnly, adCmdText
    With rstUsuarios
        While Not .EOF
            cboUsuario.AddItem !nombre_usuario, i
            i = i + 1
            .MoveNext
        Wend
    End With
    rstUsuarios.Close: Set rstUsuarios = Nothing
    cnnControlInv.Close: Set cnnControlInv = Nothing
End Sub
```

Este procedimiento permite validar al usuario y la contraseña para entrar al sistema.

```
Private Sub cmdAceptar_Click()
    If Not EstablecerConex Then
        End
    End If
    If Len(cboUsuario.Text) = 0 Then
        MsgBox "Falta el nombre de usuario.", vbCritical, "USUARIO."
        cboUsuario.SetFocus
        Exit Sub
    End If
    rstUsuarios.Open "SELECT * FROM Contraseña WHERE nombre_usuario='" &
cboUsuario.Text & "'", cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
```



```
With rstUsuarios
  If .RecordCount > 0 And !contraseña = Trim(txtContraseña.Text) Then
    tipoUsuario = !nombre_usuario
    rstUsuarios.Close: Set rstUsuarios = Nothing
    Call LiberarConex
    Unload Me
    frmBodegaintro.Show
  Else
    MsgBox "Intentelo de nuevo.", vbCritical, "Error al ingresar la
Contraseña."
    rstUsuarios.Close: Set rstUsuarios = Nothing
    Call LiberarConex
    txtContraseña.Text = ""
    txtContraseña.SetFocus
  End If
End With
End Sub
```

Este procedimiento cancela el acceso al sistema.

```
Private Sub cmdCancelar_Click()
  Unload Me
End Sub
```

Código de la Ilustración 6: Bodega de trabajo en el sistema.

En este procedimiento se introduce la bodega con la cual trabajar, además, verifica si hay productos próximos a vencerse o vencidos y mostrárselos al usuario, luego de este procedimiento se carga la ventana principal del sistema.

```
Private Sub cmdAceptar_Click()
  If Not EstablecerConex Then
    End
  End If
  'Abrir la tabla Bodega para entrar al sistema.
  rstBodega.Open "SELECT * FROM Bodega WHERE codigo_bodega='" & msknumerobod.Text &
"';", cnnControlInv, adOpenDynamic, adLockReadOnly, adCmdText
  With rstBodega
    If .RecordCount > 0 Then
      Nbodega = !codigo_bodega
      Tipobod = !descripcion_bod
      rstBodega.Close: Set rstBodega = Nothing
      Call LiberarConex
      Unload Me
      'verificar si hay productos caducados
      If ProdCaducidadCercana > 0 Then
        frmCadcercana.Show vbModal
      End If
      If ProductosCaducados > 0 Then
        frmProdVencidos.Show vbModal
      End If
      frmPrincipal.Show vbModal
    Else
      MsgBox "No se encontró la Bodega: " & msknumerobod.Text
      rstBodega.Close: Set rstBodega = Nothing
      Call LiberarConex
    End If
  End With
End Sub
```



```
msknumerobod.Text = "__-__"
msknumerobod.SetFocus
End If
End With
End Sub
```

Código de la Ilustración 7: Establecer fecha de proceso.

Este procedimiento establece la fecha inicial de proceso.

```
Private Sub Form_Load()
    mskFechainicio.Text = Date
End Sub
```

Este procedimiento valida y guarda las fechas del mes en proceso.

```
Private Sub cmdAceptar_Click()
    Dim ini As Date
    Dim fin As Date
    On Error GoTo Error_Aceptar
    If IsDate(mskFechainicio.Text) And IsDate(mskFechafin.Text) Then
        ini = CDate(mskFechainicio.Text)
        fin = CDate(mskFechafin.Text)
        If ini < Date Then
            MsgBox "Fecha de Inicio de Proceso incorrecta.", vbInformation, "Fecha de
Proceso..."
            mskFechainicio.Text = "__/__/____"
            mskFechainicio.SetFocus
            Exit Sub
        ElseIf fin < ini Then
            MsgBox "Fecha de Fin de Proceso Ya transcurrido", vbInformation, "Fecha de
Proceso..."
            mskFechafin.Text = "__/__/____"
            mskFechafin.SetFocus
            Exit Sub
        End If
        If Not EstablecerConex Then
            End
        End If
        'Abrir la tabla FechaProceso para grabar la información.
        Set rstFechaProceso = New ADODB.Recordset
        rstFechaProceso.Open "FechaProceso", cnnControlInv, adOpenDynamic,
adLockOptimistic
        rstFechaProceso.AddNew
        rstFechaProceso!fecha_inicio = mskFechainicio.Text
        rstFechaProceso!fecha_fin = mskFechafin.Text
        rstFechaProceso.Update
        rstFechaProceso.Close: Set rstFechaProceso = Nothing
        Call LiberarConex
        Unload Me
        Exit Sub
    Else
        MsgBox "Debes de establecer las Fechas correctamente.", vbInformation, "Fecha
de Proceso..."
        Exit Sub
    End If
```



```
Error_Aceptar:
    MsgBox "Se presento un error al registrar la Fecha de Proceso del Inventario. ", ,
    "Ingresar fecha del mes en proceso..."
    Call LiberarConex
End Sub
```

Código de la Ilustración 8: Ventana principal del sistema.

El siguiente procedimiento es multifuncional ya que verifica si una cotización es válida a la fecha y verifica si existe una fecha de proceso del inventario, además, pone en la barra de tareas de la aplicación la fecha, la hora, la cuenta activa y el rango de fecha del control del inventario.

```
Option Explicit
Private Sub Form_Load()
    Dim fecha_anulada As Date
    If Not EstablecerConex Then
        End
    End If
    'Verificar si una Cotización aún es válida.
    rstCotizacion.Open "SELECT estado FROM Cotizacion WHERE codigo_bodega =" &
    Nbodega & "' AND estado='VALIDA' AND fecha <= '" & Date - 10 & "' ;", cnnControlInv,
    adOpenStatic, adLockOptimistic, adCmdText
    With rstCotizacion
        If .RecordCount > 0 Then
            .MoveFirst
            Do Until .EOF
                !estado = "INVALIDA"
                .Update
                .MoveNext
            Loop
        End If
    End With
    rstCotizacion.Close: Set rstCotizacion = Nothing
    rstEmpresa.Open "SELECT nombreE FROM Empresa;", cnnControlInv, adOpenStatic,
    adLockReadOnly, adCmdText
    If rstEmpresa.RecordCount > 0 Then
        lblnomE.Caption = rstEmpresa!nombreE
    End If
    rstEmpresa.Close: Set rstEmpresa = Nothing
    barEstado.Panels("Permiso").Text = "Permisos de " & tipoUsuario
    lblnumbod.Caption = lblnumbod.Caption & " " & Nbodega
    lbltipobod.Caption = Tipobod
    If tipoUsuario = "USUARIO" Then
        InhabilitarMenus
    End If
    Call LiberarConex
    'verificar si ya existe una fecha de proceso
    If Not VerificarFechaProceso Then
        MsgBox "Debes de establecer la Fecha de Proceso de Inventario.", vbInformation,
        "Establecer nueva fecha de proceso."
```



```
    If tipoUsuario = "USUARIO" Then
        frmClaveEliminar.Show vbModal
        'Verificar Clave
        If claveAdmor Then
            frmFechaProceso.Show vbModal
        Else
            End
        End If
    Else
        frmFechaProceso.Show vbModal
    End If
Else
    'Verificar la fecha de proceso de inventario
    If Not EstablecerConex Then
        End
    End If
    rstFechaProceso.Open "SELECT * FROM FechaProceso;", cnnControlInv,
adOpenDynamic, adLockReadOnly, adCmdText
    If rstFechaProceso!fecha_fin < Date Then
        rstFechaProceso.Close: Set rstFechaProceso = Nothing
        Call LiberarConex
        frmCierreMes.cmdCancelar.Enabled = False
        frmCierreMes.Show vbModal
    End If
    rstFechaProceso.Close: Set rstFechaProceso = Nothing
    Call LiberarConex
End If
barEstado.Panels("modulo").Text = "Control de Inventario del mes en proceso del " &
final & " al " & ffinal
End Sub
```

MENU DE CONTEXTO

Este procedimiento muestra la vista de los autores de la aplicación.

```
Private Sub mnuAcerca_Click()
    frmAcercade.Show vbModal
End Sub
```

Este procedimiento cambia el caption del boton siguiente a finalizar cuando crea la primera bodega.

```
Private Sub mnuBodega_Click()
    frmBodega.cmdSiguiente.Caption = "Finalizar"
    frmBodega.Show vbModal
End Sub
```

Este procedimiento muestra los clientes existentes.

```
Private Sub mnuBusCliente_Click()
    frmBusCliente.Show vbModal
End Sub
```



Este procedimiento muestra las cotizaciones emitidas.

```
Private Sub mnuBusCotizacion_Click()  
    frmBusCotizacion.Show vbModal  
End Sub
```

Este procedimiento muestra las facturas emitidas.

```
Private Sub mnuBusFactura_Click()  
    frmBusFacturas.Show vbModal  
End Sub
```

Este procedimiento muestra los productos existentes mediante el código.

```
Private Sub mnuBusProCodigo_Click()  
    frmBusProd.cmdPorCodigo_Click  
    frmBusProd.Show vbModal  
End Sub
```

Este procedimiento muestra los productos existentes mediante la descripción.

```
Private Sub mnuBusProDescripcion_Click()  
    frmBusProd.cmdPorDescripcion_Click  
    frmBusProd.Show vbModal  
End Sub
```

Este procedimiento muestra los proveedores existentes.

```
Private Sub mnuBusProveedor_Click()  
    frmBusProveedor.Show vbModal  
End Sub
```

Este procedimiento muestra el manual para el usuario.

```
Private Sub mnuManual_Click()  
    frmManualUsuario.Show vbModal  
End Sub
```

Este procedimiento verifica si existen clietes y muestra el formulario de modificacion de no haber lo indica al usuario.

```
Private Sub mnuModCli_Click()  
    If Not EstablecerConex Then  
        End  
    End If  
    'Abrir la tabla Cliente para verificar si hay información en la Base de Datos.  
    rstCliente.Open "SELECT * FROM Cliente", cnnControlInv, adOpenDynamic,  
adLockReadOnly, adCmdText  
    If rstCliente.RecordCount > 0 Then  
        rstCliente.Close: Set rstCliente = Nothing  
        Call LiberarConex  
        frmModCliente.Show vbModal  
    Else  
        MsgBox "No se han encontrado Datos de Clientes.", vbInformation, "Error en los  
campos..."
```



```
        rstCliente.Close: Set rstCliente = Nothing
        Call LiberarConex
    End If
End Sub
```

Este procedimiento muestra la vista de usuarios para modificar la contraseña.

```
Private Sub mnuModContraseña_Click()
    frmModUsuarios.Show vbModal
End Sub
```

Este procedimiento muestra los datos de la empresa para modificarlos.

```
Private Sub mnuModDatosEmpresa_Click()
    frmModDatosEmpresa.Show vbModal
End Sub
```

Este procedimiento muestra la vista modificar fecha proceso para modificar la fecha final.

```
Private Sub mnuModFechaProceso_Click()
    frmModFechaProceso.Show vbModal
End Sub
```

Este procedimiento muestra la lista de productos contenidos en bodega.

```
Private Sub mnuModificarProducto_Click()
    frmListaProductos.Show vbModal
End Sub
```

Este procedimiento verifica si existen proveedores y muestra el formulario de modificación de no haber lo indica al usuario.

```
Private Sub mnuModProv_Click()
    If Not EstablecerConex Then
        End
    End If
    'Abrir la tabla Proveedor para verificar si hay información en la Base de Datos.
    rstProveedor.Open "Proveedor", cnnControlInv, adOpenDynamic, adLockReadOnly
    If rstProveedor.RecordCount <> 0 Then
        rstProveedor.Close: Set rstProveedor = Nothing
        Call LiberarConex
        frmModProveedor.Show vbModal
    Else
        MsgBox "No se han encontrado Datos de Proveedores.", vbInformation, "Error en los campos..."
        rstProveedor.Close: Set rstProveedor = Nothing
        Call LiberarConex
    End If
End Sub
```



Este procedimiento muestra la ventana para la creación de un nuevo producto.

```
Private Sub mnuNuevoProducto_Click()  
    frmProducto.Show vbModal  
End Sub
```

Este procedimiento muestra una ventana donde se ajustara el almacén.

```
Private Sub mnuOpAjustes_Click()  
    frmAjusteAlmacen.Show vbModal  
End Sub
```

Este procedimiento muestra la vista donde se establece la fecha de cierre del mes.

```
Private Sub mnuOpCierreMes_Click()  
    frmCierreMes.Show vbModal  
End Sub
```

Este procedimiento muestra una ventana para la creación de la copia de seguridad.

```
Private Sub mnuOpCopiaSeg_Click()  
    frmCopia_Seguridad.Show vbModal  
End Sub
```

Este procedimiento muestra la vista para crear una cotización al cliente.

```
Private Sub mnuOpCotizacion_Click()  
    frmCotizacion.Show vbModal  
End Sub
```

Este procedimiento muestra una ventana donde se registrará la entrada de productos.

```
Private Sub mnuOpEntradas_Click()  
    frmEntradasAlmacen.Show vbModal  
End Sub
```

Este procedimiento muestra la vista donde se agregaran los productos a facturar.

```
Private Sub mnuOpFacturacion_Click()  
    frmFactura.Show vbModal  
End Sub
```

Este procedimiento muestra la vista donde se agregaran los productos que se le darán de baja en almacén.

```
Private Sub mnuOpSalidas_Click()  
    frmSalidasAlmacen.Show vbModal  
End Sub
```

Este procedimiento muestra la vista donde se agregaran clientes nuevos al sistema.

```
Private Sub mnuRegCli_Click()  
    frmCliente.Show vbModal  
End Sub
```



Este procedimiento muestra la vista donde se agregaran proveedores nuevos al sistema

```
Private Sub mnuRegProv_Click()  
    frmProveedor.Show vbModal  
End Sub
```

Este procedimiento muestra la vista donde se seleccionan los datos para mostrar los reportes de los ajustes.

```
Private Sub mnuRptAjustes_Click()  
    frmReporte_Ajuste.Show vbModal  
End Sub
```

Este procedimiento muestra los reportes de todos los clientes existentes.

```
Private Sub mnuRptClientes_Click()  
    DE_ForReport.cnnControlInv.Open  
    DE_ForReport.cmdTodoCliente  
    If DE_ForReport.rscmdTodoCliente.RecordCount > 0 Then  
        Rpt_TodoCliente.Show vbModal  
    Else  
        MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de los  
Clientes."  
    End If  
    DE_ForReport.cnnControlInv.Close  
End Sub
```

Este procedimiento muestra la vista donde se seleccionan los datos para mostrar los reportes de las entradas a almacén.

```
Private Sub mnuRptEntradas_Click()  
    frmReporte_Entrada.Show vbModal  
End Sub
```

Este procedimiento muestra los reportes de las existencias de productos en almacén.

```
Private Sub mnuRptExistencias_Click()  
    DE_ForReport.cnnControlInv.Open  
    DE_ForReport.cmdExistenciasProd Nbodega  
    If DE_ForReport.rscmdExistenciasProd.RecordCount > 0 Then  
        Rpt_ExistenciasProd.Show vbModal  
    Else  
        MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de las  
Existencias de Productos."  
    End If  
    DE_ForReport.cnnControlInv.Close  
End Sub
```

Este procedimiento muestra la vista donde se seleccionan los datos para mostrar los reportes de las facturas emitidas a los clientes.

```
Private Sub mnuRptFacturas_Click()  
    frmReporte_Factura.Show vbModal  
End Sub
```



Este procedimiento muestra un reporte de los proveedores existentes.

```
Private Sub mnuRptProveedores_Click()  
    DE_ForReport.cnnControlInv.Open  
    DE_ForReport.cmdTodoProveedor  
    If DE_ForReport.rscmdTodoProveedor.RecordCount > 0 Then  
        Rpt_TodoProveedor.Show vbModal  
    Else  
        MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de los  
Proveedores."  
    End If  
    DE_ForReport.cnnControlInv.Close  
End Sub
```

Este procedimiento muestra la vista donde se seleccionan los datos para mostrar los reportes de las de salidas de almacén.

```
Private Sub mnuRptSalidas_Click()  
    frmReporte_Salida.Show vbModal  
End Sub
```

Este procedimiento permite salir del programa.

```
Private Sub mnuSalir_Click()  
    Unload Me  
End Sub
```

Esta funcion limita a la cuenta de usuario a hacer funciones que solo las realizará el administrador del sistema.

```
Private Function InhabilitarMenus()  
    mnuBodega.Enabled = False  
    mnuOpCopiaSeg.Enabled = False  
    mnuOpAjustes.Enabled = False  
    mnuOpCierreMes.Enabled = False  
    mnuModificarProducto = False  
    mnuModFechaProceso = False  
    mnuModDatosEmpresa.Enabled = False  
    mnuModContraseña.Enabled = False  
    mnuOpCierreMes.Enabled = False  
End Function
```

**Código de la Ilustración 9: Respaldo de la base de datos.**

Este procedimiento respalda la base de datos indicando su nombre y la ruta.

```
Private Sub cmdRespaldar_Click()  
    On Error GoTo Err_respaldar  
    Dim rutaDestino As String  
    Dim cmdBackup As New ADODB.Command  
  
    If Trim(txtnombreCS.Text) = "" Then  
        MsgBox "Introduzca el nombre con el cual se creará la Copia de Seguridad.",  
vbInformation, "Copias de Seguridad..."  
        Exit Sub  
    End If  
    rutaDestino = dirDestino.Path & "\" & Trim(txtnombreCS.Text) & ".bak"  
    cmdBackup.ActiveConnection = "Provider=SQLOLEDB.1;Password=caics;Persist Security  
Info=True;User ID=caics;" _  
        & "Data Source=SERVER"  
    cmdBackup.CommandText = "BACKUP DATABASE Inventario TO DISK = '" & rutaDestino & "'  
WITH NOINIT, NOSKIP, STATS = 10 "  
    cmdBackup.Execute  
    MsgBox "Base de datos Respaldata.", vbInformation, "Respaldar Base de Datos."  
    flstDestino.Refresh  
    flstOrigen.Refresh  
    txtnombreCS.Text = ""  
    lblRutaCS.Caption = lblRutaCS.Caption & dirDestino.Path  
    Exit Sub  
Err_respaldar:  
    MsgBox "Es posible que el nombre de la copia de seguridad lleve Caracteres no  
permitidos, pongase en contacto con el Programador.", vbInformation, "Copias de  
Seguridad..."  
End Sub
```

Código de la Ilustración 10: Restauración de la base de datos.

Este procedimiento restaura la base de datos buscando la ruta y el nombre con que fue respaldada.

```
Private Sub cmdRestaurar_Click()  
    On Error GoTo Err_restaurar  
    Dim rutaorigen As String  
    Dim cmdRestore As New ADODB.Command  
  
    If flstOrigen.FileName = "" Then  
        MsgBox "Debe Seleccionar un archivo de la lista.", vbInformation, "Copias de  
Seguridad..."  
        Exit Sub  
    End If  
    rutaorigen = dirOrigen.Path & "\" & flstOrigen.FileName  
    cmdRestore.ActiveConnection = "Provider=SQLOLEDB.1;Password=caics;Persist Security  
Info=True;User ID=caics;" _  
        & "Data Source=SERVER"  
    cmdRestore.CommandText = "RESTORE DATABASE Inventario FROM DISK = '" & rutaorigen & "'  
    cmdRestore.Execute  
    MsgBox "Base de datos restaurada.", vbInformation, "Restaurar Base de Datos."
```



```
Exit Sub
Err_restaurar:
    MsgBox "La Base de datos no se puede restaurar, es posible que este siendo
utilizada." & Err.Description, vbInformation, "Copias de Seguridad..."
End Sub
```

Este procedimiento cancela el respaldo o la restauración de la base de datos.

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```

Código de la Ilustración 11: Entrada de productos al almacén.

En este procedimiento se validan los datos del producto, se realizan operaciones automáticamente y se agregan de uno en uno al control flexgrid.

```
Private Sub cmdAgregarprod_Click()
    Dim TotalEnt As Single
    If cbxdescripcion_Entrada.Text = "" Then
        MsgBox "Falta descripción del Producto.", vbInformation
        cbxdescripcion_Entrada.SetFocus
        Exit Sub
    ElseIf CCur(txtcantidad.Text) = 0 Then
        MsgBox "Falta la cantidad del Producto.", vbInformation
        txtcantidad.SetFocus
        Exit Sub
    ElseIf CCur(txtcostounitario.Text) = 0 Then
        MsgBox "Falta el costo unitario del Producto.", vbInformation
        txtcostounitario.SetFocus
        Exit Sub
    ElseIf txtpreciomenor.Text = "" Then
        MsgBox "El precio por Menor del Producto no puede ser Null...", vbInformation
        txtpreciomenor.SetFocus
        Exit Sub
    ElseIf txtpreciomayor.Text = "" Then
        MsgBox "El precio por Mayor del Producto no puede ser Null...", vbInformation
        txtpreciomayor.SetFocus
        Exit Sub
    ElseIf ComprobarProducto Then
        Exit Sub
    ElseIf tienefechaCad Then
        'Validar la fecha de caducidad del producto.
        If Not IsDate(mskfechacad.Text) Then
            MsgBox "Este Producto tiene Fecha de Caducidad o la misma es incorrecta.",
, "Error en la Fecha de Caducidad."
            mskfechacad.SetFocus
            Exit Sub
        Else
            If CDate(mskfechacad.Text) <= Date Then
                MsgBox "La Fecha de Caducidad del Producto ya ha sido transcurrida.",
vbInformation
                mskfechacad.SetFocus
                Exit Sub
            End If
            FechaCaducidad = mskfechacad.Text
        End If
    End If
End Sub
```



```
End If
Else
    FechaCaducidad = ""
End If
With flexDetalleEntrada
    Fila = Fila + 1
    .AddItem txtcodigoprod.Text, Fila
    .TextMatrix(Fila, 1) = txtcodigoprod.Text
    .TextMatrix(Fila, 2) = cbxdescripcion_Entrada.Text
    .TextMatrix(Fila, 3) = txtunidad.Text
    .TextMatrix(Fila, 4) = txtcantidad.Text
    .TextMatrix(Fila, 5) = FormatNumber(txtcostounitario.Text, 2)
    .TextMatrix(Fila, 6) = FormatNumber(txtsubtotal.Text, 2)
    .TextMatrix(Fila, 7) = FechaCaducidad
    .TextMatrix(Fila, 8) = FormatNumber(CCur(txtpreciomenor.Text), 2)
    .TextMatrix(Fila, 9) = FormatNumber(CCur(txtpreciomayor.Text), 2)
End With
TotalEnt = FormatNumber(CCur(lblTotalent.Caption), 2)
lblTotalent.Caption = FormatNumber(CCur(TotalEnt) + CCur(txtsubtotal.Text), 2)
Desactivarcajas
cmdAgregarprod.Enabled = False
cmdGuardar.Enabled = True
cmdEliminar.Enabled = True
cmdCancelar.Enabled = True
End Sub
```

Este procedimiento elimina un producto en especifico de los agregados en el flexgrid.

```
Private Sub cmdEliminar_Click()
    With flexDetalleEntrada
        If .Row = .Rows - 1 And .Row < .Rows Then
            MsgBox "No puedes eliminar una fila en blanco", vbInformation, "Error al eliminar..."
            .Row = 1
            Exit Sub
        End If
        'Eliminar de la cuadrícula la fila seleccionada.
        If MsgBox("Desea quitar este Producto del detalle?", vbYesNo + vbQuestion) = vbYes Then
            .RemoveItem .Row
            Fila = Fila - 1
        End If
    End With
    If Fila = 0 Then
        cmdGuardar.Enabled = False
        cmdEliminar.Enabled = False
        cmdCancelar.Enabled = False
    End If
End Sub
```



Este procedimiento actualiza las existencias de productos, valida los datos de la entrada al almacén y los guarda junto con el contenido del control flexgrid en la base de datos.

```
Private Sub cmdGuardar_Click()
    Dim i As Integer
    Dim lote As Long
    If txtdocent.Text = "" Then
        MsgBox "Falta el campo Documento", vbInformation, "Entrada a Almacén..."
        txtdocent.SetFocus
        Exit Sub
    ElseIf txtnombrepurv = "" Then
        MsgBox "Falta el Proveedor", vbInformation, "Entrada a Almacén..."
        Exit Sub
    End If
    If Not EstablecerConex Then
        End
    End If
    cnnControlInv.BeginTrans

    CodificarEntradaAlmacen
    txtcodigoent.Text = CodigoEnt

    rstDetalleEntrada.Open "Detalle_Entrada", cnnControlInv, adOpenDynamic,
adLockOptimistic, adCmdTable
    rstEntradaAlmacen.Open "Entrada_Almacen", cnnControlInv, adOpenStatic,
adLockOptimistic, adCmdTable
    rstLote.Open "Lote_Poducto", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdTable
    'Agregar el registro en la tabla Entrada_Almacen
    With rstEntradaAlmacen
        .AddNew
        !codigo_bodega = txtcodigobod.Text
        !codigo_entrada = Trim(txtcodigoent.Text)
        !documento = txtdocent.Text
        !tipo_doc = cbxtipodocent.Text
        !fecha_entrada = txtfechaent.Text
        !codigo_prov = txtcodigoprov.Text
        !nombre_prov = txtnombrepurv.Text
        !movimiento = "ENTRADA"
        !entradatotal = CCur(lblTotalent.Caption)
        .Update
        EntradaGuardada = True
    End With
    rstEntradaAlmacen.Close: Set rstEntradaAlmacen = Nothing
    For i = 1 To Fila
        lote = CodificarLote          'Lamada a la función para crear la lotificación del
Producto.
        With rstDetalleEntrada
            .AddNew
            !codigo_entrada = Trim(txtcodigoent.Text)
            !codigo_prod = flexDetalleEntrada.TextMatrix(i, 1)
            !descripcion = flexDetalleEntrada.TextMatrix(i, 2)
            !unidad = flexDetalleEntrada.TextMatrix(i, 3)
            !numlote = lote
            !cantidad = flexDetalleEntrada.TextMatrix(i, 4)
            !costo_unitario = CCur(flexDetalleEntrada.TextMatrix(i, 5))
        End With
    Next i
End Sub
```



```
!sub_total = CCur(flexDetalleEntrada.TextMatrix(i, 6))
If flexDetalleEntrada.TextMatrix(i, 7) <> "" Then
    !fecha_caducidad = CDate(flexDetalleEntrada.TextMatrix(i, 7))
End If
.Update
End With
'Actualizar las existencias de los productos en almacén.
rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod ='" &
(flexDetalleEntrada.TextMatrix(i, 1)) & "' And codigo_bodega ='" & (txtcodigobod.Text)
& " '"; cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
With rstProducto
    !exis_final = CCur(!exis_final +
FormatNumber(CCur(flexDetalleEntrada.TextMatrix(i, 4)), 2))
    !costo_unitario = CCur(FormatNumber(!saldo +
CCur(flexDetalleEntrada.TextMatrix(i, 6))) / !exis_final, 2))
    !saldo = CCur(FormatNumber(!exis_final * !costo_unitario, 2))
    !preciol = CSng(flexDetalleEntrada.TextMatrix(i, 8))
    !precio2 = CSng(flexDetalleEntrada.TextMatrix(i, 9))
    If .EditMode = adEditInProgress Then
        .Update
    End If
End With
rstProducto.Close: Set rstProducto = Nothing
'Agregar el registro a la tabla Lote_Producto.
With rstLote
    .AddNew
    !codigo_bodega = Nbodega
    !codigo_entrada = txtcodigoent.Text
    !codigo_prod = flexDetalleEntrada.TextMatrix(i, 1)
    If flexDetalleEntrada.TextMatrix(i, 7) <> "" Then
        !fecha_caducidad = flexDetalleEntrada.TextMatrix(i, 7)
    Else
        !fecha_entrada = Date
    End If
    !numlote = lote
    !cantidad_fecha = CCur(flexDetalleEntrada.TextMatrix(i, 4))
    .Update
End With
Next i
rstDetalleEntrada.Close: Set rstDetalleEntrada = Nothing
rstLote.Close: Set rstLote = Nothing
cnnControlInv.CommitTrans
Call LiberarConex
DesactivarBotones
cmdImprimir.Enabled = True
txtdocent.Enabled = False
cbxtipodocent.Enabled = False
cbxdescripcion_Entrada.Enabled = False
End Sub
```



Este procedimiento cancela la introducción de productos y elimina los que se hayan agregado en el flexgrid.

```
Private Sub cmdCancelar_Click()  
    Dim i As Integer  
    'Eliminar todas las filas agregadas a la cuadrícula.  
    For i = Fila To 1 Step -1  
        flexDetalleEntrada.RemoveItem i  
    Next i  
    Fila = 0  
    DesactivarBotones  
End Sub
```

Este procedimiento se usa para salir de la entrada de productos, pero si no se han guardado en la base de datos se pregunta al usuario si desea guardarlos.

```
Private Sub cmdSalir_Click()  
    If Not EntradaGuardada And Fila > 0 Then  
        If MsgBox("¿Desea guardar la Entrada de los Productos?", vbYesNo) = vbYes Then  
            cmdGuardar_Click  
            Exit Sub  
        Else  
            cmdCancelar_Click  
        End If  
    End If  
    bolEntradavisible = False  
    CodigoEnt = ""  
    Unload Me  
End Sub
```

Este procedimiento imprime la entrada de los productos para la creación de lotes.

```
Private Sub cmdImprimir_Click()  
    DE_ForReport.cnnControlInv.Open  
    DE_ForReport.cmdEntrada Trim(txtcodigoent.Text), Trim(Nbodega)  
    Rpt_Entrada.Show vbModal  
    cmdImprimir.Enabled = False  
    DE_ForReport.cnnControlInv.Close  
End Sub
```

Código de la Ilustración 12: Salida de productos del almacén.

En este procedimiento se validan los datos del producto, se realizan operaciones automáticamente y se agregan de uno en uno al control flexgrid para luego extraerlos del almacén.

```
Private Sub cmdAgregarprod_Click()  
    Dim totalsal As Currency  
    If cbxdescripcion.Text = "" Then  
        MsgBox "Falta la descripción del Producto.", vbInformation  
        cbxdescripcion.SetFocus  
        Exit Sub  
    ElseIf CCur(txtcantidad.Text) = 0 Then  
        MsgBox "Falta la cantidad del Producto.", vbInformation  
        txtcantidad.SetFocus  
        Exit Sub  
    End If
```



```
ElseIf ComprobarProducto Then
    Exit Sub
End If
With flexDetalleSalida
    filadetsal = filadetsal + 1
    .AddItem txtcodigoprod.Text, filadetsal
    .TextMatrix(filadetsal, 1) = txtcodigoprod.Text
    .TextMatrix(filadetsal, 2) = cbxdescripcion.Text
    .TextMatrix(filadetsal, 3) = txtunidad.Text
    .TextMatrix(filadetsal, 4) = FormatNumber(CCur(txtcantidad.Text), 2)
    .TextMatrix(filadetsal, 5) = FormatNumber(CCur(txtsubtotal.Text), 2)
End With
totalsal = CCur(lblTotalsal.Caption)
lblTotalsal.Caption = FormatNumber(CCur(totalsal) + CCur(txtsubtotal.Text), 2)
cmdAgregarpod.Enabled = False
cmdGuardar.Enabled = True
cmdEliminar.Enabled = True
cmdCancelar.Enabled = True
txtcantidad.Enabled = False
LimpiarCajas
End Sub
```

Este procedimiento elimina del control flexgrid el producto que se ha seleccionado.

```
Private Sub cmdEliminar_Click()
With flexDetalleSalida
    If .Row = .Rows - 1 And .Row < .Rows Then
        MsgBox "No puedes eliminar una fila en blanco", vbInformation, "Error al
eliminar..."
        .Row = 1
        Exit Sub
    End If
    If MsgBox("Desea quitar este Producto del detalle?", vbYesNo + vbQuestion) =
vbYes Then
        lblTotalsal.Caption = FormatNumber(CCur(lblTotalsal.Caption) -
CCur(.TextMatrix(.Row, 5)), 2)
        .RemoveItem .Row
        filadetsal = filadetsal - 1
        LimpiarCajas
    End If
End With
If filadetsal = 0 Then
    cmdGuardar.Enabled = False
    cmdEliminar.Enabled = False
    cmdCancelar.Enabled = False
End If
End Sub
```



Este procedimiento actualiza las existencias de productos, valida los datos de la salida de almacén y los guarda junto con el contenido del control flexgrid en la base de datos.

```
Public Sub cmdGuardar_Click()  
    Dim i As Integer, numeroLote As Long  
    On Error GoTo Error_Guardar  
    If txtdocsal.Text = "" Then  
        MsgBox "Falta el Documento de salida", vbInformation, "Salida de Almacén..."  
        txtdocsal.SetFocus  
        Exit Sub  
    End If  
    If Not EstablecerConex Then  
        End  
    End If  
    cnnControlInv.BeginTrans  
  
    CodificarSalidaAlmacen  
    txtcodigosal.Text = CodigoSal  
  
    rstSalidaAlmacen.Open "Salida_Almacen", cnnControlInv, adOpenStatic,  
adLockOptimistic, adCmdTable  
    rstDetalleSalida.Open "Detalle_Salida", cnnControlInv, adOpenStatic,  
adLockOptimistic, adCmdTable  
  
    With rstSalidaAlmacen  
        'Agregar el registro en la tabla Salida_Almacen  
        .AddNew  
        !codigo_bodega = txtcodigobod.Text  
        !codigo_salida = Trim(txtcodigosal.Text)  
        !documento = txtdocsal.Text  
        !tipo_doc = cbxtipodocsal.Text  
        !fecha_salida = txtfechasal.Text  
        !movimiento = "SALIDA"  
        !salidatotal = CCur(lblTotalsal.Caption) : .Update  
        SalidaGuardada = True  
    End With  
    rstSalidaAlmacen.Close: Set rstSalidaAlmacen = Nothing  
    For i = 1 To filadetsal  
        rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod ='" &  
(flexDetalleSalida.TextMatrix(i, 1)) & "' And codigo_bodega ='" & (txtcodigobod.Text) &  
" '";", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText  
        With rstProducto  
            If CCur(flexDetalleSalida.TextMatrix(i, 4)) > !exis_final Then  
                MsgBox "No existen " & flexDetalleSalida.TextMatrix(i, 4) & " " &  
flexDetalleSalida.TextMatrix(i, 3) & " de " & flexDetalleSalida.TextMatrix(i, 2), ,  
"Existencia insuficiente."  
                cnnControlInv.RollbackTrans  
                rstProducto.Close: Set rstProducto = Nothing  
                rstDetalleSalida.Close: Set rstDetalleSalida = Nothing  
                Call LiberarConex  
                Exit Sub  
            Else  
                If !tiene_caducidad Then  
                    numeroLote = SacarProductoDelLote(!codigo_prod,  
CCur(flexDetalleSalida.TextMatrix(i, 4)), True)  
                Else
```



```
                numeroLote = SacarProductoDelLote(!codigo_prod,
CCur(flexDetalleSalida.TextMatrix(i, 4)), False)
                End If
                !exis_final = CCur(!exis_final) - CCur(flexDetalleSalida.TextMatrix(i,
4))

                If !exis_final = 0 Then !costo_unitario = 0
                !saldo = CCur(FormatNumber(!exis_final * !costo_unitario, 2))
                'Sacar el producto previo a vencer
                If .EditMode = adEditInProgress Then
                    .Update
                End If
            End If
        End With
        rstProducto.Close: Set rstProducto = Nothing
        With rstDetalleSalida
            .AddNew
            !codigo_salida = Trim(txtcodigosal.Text)
            !codigo_prod = flexDetalleSalida.TextMatrix(i, 1)
            !descripcion = flexDetalleSalida.TextMatrix(i, 2)
            !unidad = flexDetalleSalida.TextMatrix(i, 3)
            !numlote = numeroLote
            !cantidad = CCur(flexDetalleSalida.TextMatrix(i, 4))
            !sub_total = CCur(flexDetalleSalida.TextMatrix(i, 5))
            .Update
        End With
    Next i
    rstDetalleSalida.Close: Set rstDetalleSalida = Nothing
    cnnControlInv.CommitTrans
    Call LiberarConex
    cmdImprimir.Enabled = True
    cbxdescripcion.Enabled = False
    txtcantidad.Enabled = False
    DesactivarBotones
    txtdocsal.Enabled = False
    cbxtipodocsal.Enabled = False
    Exit Sub
Error_Guardar:
    MsgBox "No se puede guardar la Salida de Almacén. ", vbInformation, "Error al
guardar la Salida de Almacén ..."
    cmdGuardar.Enabled = True
    cmdCancelar.Enabled = True
    cnnControlInv.RollbackTrans
    Call LiberarConex
End Sub
```

Este procedimiento cancela todos los productos que se han agregado al control flexgrid.

```
Private Sub cmdCancelar_Click()
    Dim i As Integer
    For i = filadetsal To 1 Step -1
        flexDetalleSalida.RemoveItem i
    Next i
    lblTotalsal.Caption = "0,00"
    filadetsal = 0
    DesactivarBotones
End Sub
```



Este procedimiento imprime la salida de productos del almacén.

```
Public Sub cmdImprimir_Click()  
    DE_ForReport.cnnControlInv.Open  
    DE_ForReport.cmdSalida Trim(txtcodigosal.Text), Trim(Nbodega)  
    Rpt_Salida.PrintReport True  
    cmdImprimir.Enabled = False  
    DE_ForReport.cnnControlInv.Close  
End Sub
```

Este procedimiento finaliza la salida de productos y pregunta si desea guardar la salidad de productos contenidos en el flexgrid si esta no ha sido guardada.

```
Private Sub cmdSalir_Click()  
    If Not SalidaGuardada And filadetsal > 0 Then  
        If MsgBox("¿Desea guardar la Salida de los Productos de la lista", vbYesNo) =  
vbYes Then  
            cmdGuardar_Click  
            Exit Sub  
        Else  
            cmdCancelar_Click  
        End If  
    End If  
    CodigoSal = ""  
    Unload Me  
End Sub
```

Código de la Ilustración 13: Ajuste de productos en almacén.

Este procedimiento valida y agrega los productos que se han ajustado al control flexgrid.

```
Private Sub cmdAgregarprod_Click()  
    'Validar cajas de texto.  
    If cbxdescripcion.Text = "" Then  
        MsgBox "Falta la descripción del Producto.", vbInformation  
        cbxdescripcion.SetFocus  
        Exit Sub  
    ElseIf txtnuevaexistencia.Text = "" And _  
(cbxtipodocaju.Text = "AF" Or cbxtipodocaju.Text = "AA") Then  
        MsgBox "Falta la nueva existencia del Producto.", vbInformation  
        txtnuevaexistencia.SetFocus  
        Exit Sub  
    ElseIf CCur(txtnuevocosto.Text) = 0 And txtnuevocosto.Enabled = True And _  
(cbxtipodocaju.Text = "AC" Or cbxtipodocaju.Text = "AA") Then  
        MsgBox "Falta el nuevo costo del Producto.", vbInformation  
        txtnuevocosto.SetFocus  
        Exit Sub  
    ElseIf ComprobarProducto Then  
        Exit Sub  
    End If  
    With flexDetalleAjuste  
        filapro = filapro + 1  
        .AddItem txtcodigoprod.Text, filapro  
        .TextMatrix(filapro, 1) = txtcodigoprod.Text  
        .TextMatrix(filapro, 2) = cbxdescripcion.Text
```



```
.TextMatrix(filapro, 3) = txtunidad.Text
If Not IsNull(varCant_Ajustar) Then _
    .TextMatrix(filapro, 4) = varCant_Ajustar
If varCosto_Ajustar <> 0 Then _
    .TextMatrix(filapro, 5) = FormatNumber(varCosto_Ajustar, 2)
End With
cbxtipodocaju.Enabled = False
cmdAgregarprod.Enabled = False
cmdGuardar.Enabled = True
cmdEliminar.Enabled = True
cmdCancelar.Enabled = True
varCant_Ajustar = Null
LimpiarCajas
End Sub
```

Este procedimiento elimina del control flexgrid el producto que ha sido seleccionado.

```
Private Sub cmdEliminar_Click()
    With flexDetalleAjuste
        If .Row = .Rows - 1 And .Row < .Rows Then
            MsgBox "No puedes eliminar una fila en blanco", vbInformation, "Error al
eliminar..."
            .Row = 1
            Exit Sub
        End If
        If MsgBox("Desea quitar este Producto del detalle?", vbYesNo + vbQuestion) =
vbYes Then
            txtexistencia.Text = 0
            .RemoveItem .Row
            filapro = filapro - 1
            LimpiarCajas
        End If
    End With
    If filapro = 0 Then
        cmdGuardar.Enabled = False
        cmdEliminar.Enabled = False
        cmdCancelar.Enabled = False
    End If
End Sub
```

Este procedimiento actualiza los datos de los productos, guarda los datos del ajuste en almacén junto con el contenido del control flexgrid en la base de datos.

```
Private Sub cmdGuardar_Click()
    Dim i As Integer, numeroLote As Long
    Dim costouni As Currency
    costouni = 0
    If Not EstablecerConex Then
        End
    End If
    cnnControlInv.BeginTrans

    CodificarAjuste
    txtcodigoajuste.Text = CodigoAju
```



```
rstAjusteAlmacen.Open "Ajustes_Almacen", cnnControlInv, adOpenStatic,
adLockOptimistic
rstDetalleAjuste.Open "Detalle_Ajuste", cnnControlInv, adOpenStatic,
adLockOptimistic

With rstAjusteAlmacen
  'Agregar el registro en la tabla Salida_Almacen
  .AddNew
  !codigo_bodega = txtcodigobod.Text
  !codigo_ajuste = Trim(txtcodigoajuste.Text)
  !tipo_doc = cbxtipodocaju.Text
  !fecha_ajuste = txtfechaaju.Text
  !movimiento = "AJUSTE"
  .Update
  AjusteGuardado = True
End With
rstAjusteAlmacen.Close: Set rstAjusteAlmacen = Nothing
For i = 1 To filapro
  rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod ='" &
flexDetalleAjuste.TextMatrix(i, 1) & "' And codigo_bodega ='" & (txtcodigobod.Text) & "'";
  cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
  With rstProducto
    If Len(flexDetalleAjuste.TextMatrix(i, 4)) > 0 Then
      !exis_final = CCur(FormatNumber(!exis_final +
CCur(flexDetalleAjuste.TextMatrix(i, 4)), 2))
      If !tiene_caducidad Then
        numeroLote = SacarProductoDelLote(!codigo_prod,
CCur(flexDetalleAjuste.TextMatrix(i, 4)) * -1, True)
      Else
        numeroLote = SacarProductoDelLote(!codigo_prod,
CCur(flexDetalleAjuste.TextMatrix(i, 4)) * -1, False)
      End If
    End If
    If Len(flexDetalleAjuste.TextMatrix(i, 5)) > 0 Then
      !costo_unitario = CCur(FormatNumber(!costo_unitario +
CCur(flexDetalleAjuste.TextMatrix(i, 5)), 2))
      costouni = CCur(flexDetalleAjuste.TextMatrix(i, 5))
    End If
    If !exis_final = 0 Then !costo_unitario = 0
    !saldo = CCur(FormatNumber(!exis_final * !costo_unitario, 2))
    If .EditMode = adEditInProgress Then
      .Update
    End If
  End With
  rstProducto.Close: Set rstProducto = Nothing
  With rstDetalleAjuste
    .AddNew
    !codigo_ajuste = Trim(txtcodigoajuste.Text)
    !codigo_prod = flexDetalleAjuste.TextMatrix(i, 1)
    !descripcion = flexDetalleAjuste.TextMatrix(i, 2)
    !unidad = flexDetalleAjuste.TextMatrix(i, 3)
    If Len(flexDetalleAjuste.TextMatrix(i, 4)) > 0 Then
      !ajuste_exis = CCur(flexDetalleAjuste.TextMatrix(i, 4))
      !numlote = numeroLote
    End If
    If Len(flexDetalleAjuste.TextMatrix(i, 5)) > 0 Then _
      !ajuste_costo = CCur(flexDetalleAjuste.TextMatrix(i, 5))
```



```
        !costo_unitario = CCur(flexDetalleAjuste.TextMatrix(i, 6)) + costouni
        .Update
    End With
Next i
rstDetalleAjuste.Close: Set rstDetalleAjuste = Nothing
cnnControlInv.CommitTrans
Call LiberarConex
cbxDescripcion.Enabled = False
cbxtipodocaju.Enabled = False
txtnuevaexistencia.Enabled = False
txtnuevocosto.Enabled = False
DesactivarBotones
cmdImprimir.Enabled = True
cmdSalir.Enabled = True
Exit Sub
Error_Guardar:
MsgBox "No se puede guardar el Ajuste en Almacén. ", vbInformation, "Error al
guardar el Ajuste en Almacén ..."
cmdGuardar.Enabled = True
cmdCancelar.Enabled = True
cnnControlInv.RollbackTrans
Call LiberarConex
End Sub
```

Este procedimiento cancela los datos de los productos que se han agregado al control flexgrid.

```
Private Sub cmdCancelar_Click()
    Dim i As Integer
    For i = filapro To 1 Step -1
        flexDetalleAjuste.RemoveItem i
    Next i
    filapro = 0
    cbxtipodocaju.Enabled = True
    DesactivarBotones
    LimpiarCajas
End Sub
```

Este procedimiento imprime el ajuste que se realizo a los productos en almacén.

```
Private Sub cmdImprimir_Click()
    DE_ForReport.cnnControlInv.Open
    DE_ForReport.cmdAjuste Trim(txtcodigoajuste.Text), Trim(Nbodega)
    If MsgBox("¿Desea realmente imprimir el ajuste en Almacén? ", vbYesNo) = vbYes Then
        Rpt_Ajuste.PrintReport True
    End If
    cmdImprimir.Enabled = False
    DE_ForReport.cnnControlInv.Close
End Sub
```



Este procedimiento finaliza el ajuste de productos y pregunta si desea guardar el ajuste realizado a los productos contenidos en el flexgrid si estos no se guardaron.

```
Private Sub cmdSalir_Click()  
    If Not AjusteGuardado And filapro > 0 Then  
        If MsgBox("¿Desea guardar el ajuste de los Productos en almacén?", vbYesNo) =  
vbYes Then  
            cmdGuardar_Click  
            Exit Sub  
        Else  
            cmdCancelar_Click  
        End If  
    End If  
    CodigoAju = ""  
    Unload Me  
End Sub
```

Código de la Ilustración 14: Factura del cliente.

Este procedimiento valida y agrega un producto al control flexgrid el cual ha sido pedido por un cliente para luego ser este facturado para el mismo.

```
Public Sub cmdAgregarprod_Click()  
    Dim importefac As Currency  
    If cbxdescripcion.Text = "" Then  
        MsgBox "Falta descripción del Producto.", vbInformation  
        cbxdescripcion.SetFocus  
        Exit Sub  
    ElseIf CCur(txtcantidad.Text) = 0 Then  
        MsgBox "Falta la cantidad del Producto :" & cbxdescripcion.Text, vbInformation  
        txtcantidad.SetFocus  
        Exit Sub  
    ElseIf ComprobarProducto Then  
        LimpiarCajas  
        Exit Sub  
    End If  
    With flexDetalleFactura  
        filasfac = filasfac + 1  
        .AddItem txtcodigoprod.Text, filasfac  
        .TextMatrix(filasfac, 1) = txtcodigoprod.Text  
        .TextMatrix(filasfac, 2) = cbxdescripcion.Text  
        .TextMatrix(filasfac, 3) = txtunidad.Text  
        .TextMatrix(filasfac, 4) = FormatNumber(CCur(txtcantidad.Text), 2)  
        .TextMatrix(filasfac, 5) = FormatNumber(CCur(txtprecio.Text), 2)  
        .TextMatrix(filasfac, 6) = FormatNumber(CCur(txtsubtotal.Text), 2)  
        .TextMatrix(filasfac, 7) = FormatNumber(varcosto, 2)  
        SalidaTot = SalidaTot + (CCur(txtcantidad.Text) * varcosto)  
    End With  
    importefac = CCur(lblTotalfac.Caption)  
    lblTotalfac.Caption = FormatNumber(importefac + CCur(txtsubtotal.Text), 2)  
    cmdAgregarprod.Enabled = False  
    cmdGuardar.Enabled = True  
    cmdCancelar.Enabled = True  
    LimpiarCajas  
End Sub
```



Este procedimiento elimina un producto del control flexgrid que ha sido seleccionado.

```
Private Sub cmdEliminar_Click()
    With flexDetalleFactura
        If .Row = .Rows - 1 And .Row < .Rows Then
            MsgBox "No puedes eliminar una fila en blanco", vbInformation, "Error al
eliminar..."
            .Row = 1
            cmdEliminar.Enabled = False
            Exit Sub
        End If
        If MsgBox("Desea quitar este Producto del detalle?", vbYesNo + vbQuestion) =
vbYes Then
            txtexistencia.Text = 0
            lblTotalfac.Caption = FormatNumber(CCur(lblTotalfac.Caption) -
CCur(.TextMatrix(.Row, 6)), 2)
            SalidaTot = SalidaTot - (CCur(.TextMatrix(.Row, 7)) *
CCur(.TextMatrix(.Row, 4)))
            .RemoveItem .Row
            filasfac = filasfac - 1
            LimpiarCajas
        End If
    End With
    If filasfac = 0 Then
        cmdGuardar.Enabled = False
        cmdEliminar.Enabled = False
        cmdCancelar.Enabled = False
        cmdImprimir.Enabled = False
    End If
End Sub
```

Este procedimiento valida, actualiza la salida del almacén y las existencias de productos y, guarda la factura con todos los productos solicitados por el cliente.

```
Private Sub cmdGuardar_Click()
    Dim i As Integer
    ReDim varnumLotes(filasfac)
    If txtnombrecli.Text = "" Then
        MsgBox "Falta los datos del Cliente.", vbInformation, "Facturando..."
        cmdBuscarcli.Enabled = True
        cmdNuevocli.Enabled = True
        Exit Sub
    End If
    If filasfac = 0 Then
        MsgBox "No hay productos que facturar.", vbInformation, "Facturando..."
        Exit Sub
    End If
    If Not EstablecerConex Then
        End
    End If
    cnnControlInv.BeginTrans

    CodificarFactura
    txtcodigofac.Text = CódigoFac
    rstFactura.Open "Factura", cnnControlInv, adOpenStatic, adLockOptimistic
```



```
rstDetalleFactura.Open "Detalle_Factura", cnnControlInv, adOpenStatic,
adLockOptimistic

With rstFactura
    'Agregar el registro en la tabla Factura
    .AddNew
    !codigo_bodega = txtcodigobod.Text
    !nFactura = Trim(txtcodigofac.Text)
    !fecha = txtfechafac.Text
    !codigo_cli = txtcodigocli.Text
    !nombre_cli = txtnombrecli.Text
    !direccion = txtdireccioncli.Text
    If (Len(txttelefonocli.Text) <> 0) And (txttelefonocli.Text <> "Nulo") Then _
    !telefono = txttelefonocli.Text
    !importe_factura = CCur(lblTotalfac.Caption)
    !igv_factura = CCur(lbltotaligv.Caption)
    !monto_factura = CCur(lblmontofac.Caption)
    .Update
    FacturaGuardada = True
End With
rstFactura.Close: Set rstFactura = Nothing

CodificarSalidaAlmacen

For i = 1 To filasfac
    With rstDetalleFactura
        .AddNew
        !nFactura = Trim(txtcodigofac.Text)
        !codigo_prod = flexDetalleFactura.TextMatrix(i, 1)
        !descripcion = flexDetalleFactura.TextMatrix(i, 2)
        !unidad = flexDetalleFactura.TextMatrix(i, 3)
        !cantidad = CCur(flexDetalleFactura.TextMatrix(i, 4))
        !precio = CCur(flexDetalleFactura.TextMatrix(i, 5))
        !subtotal = CCur(flexDetalleFactura.TextMatrix(i, 6))
        .Update
    End With

    rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod ='" &
(Trim(flexDetalleFactura.TextMatrix(i, 1))) & "' And codigo_bodega = '" &
(txtcodigobod.Text) & " '";", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
    With rstProducto
        If CCur(flexDetalleFactura.TextMatrix(i, 4)) > !exis_final Then
            MsgBox "No existen " & flexDetalleFactura.TextMatrix(i, 4) & " " &
flexDetalleFactura.TextMatrix(i, 3) & " de " & flexDetalleFactura.TextMatrix(i, 2), ,
"Existencia insuficiente."
            cnnControlInv.RollbackTrans
            rstProducto.Close: Set rstProducto = Nothing
            rstDetalleFactura.Close: Set rstDetalleFactura = Nothing
            Call LiberarConex
            Exit Sub
        Else
            If !tiene_caducidad Then
                varnumLotes(i) = SacarProductoDelLote(!codigo_prod,
CCur(flexDetalleFactura.TextMatrix(i, 4)), True)
            Else
                varnumLotes(i) = SacarProductoDelLote(!codigo_prod,
CCur(flexDetalleFactura.TextMatrix(i, 4)), False)
```



```
End If
!exis_final = CCur(!exis_final) - CCur(flexDetalleFactura.TextMatrix(i,
4))
If !exis_final = 0 Then !costo_unitario = 0
!saldo = FormatNumber(!exis_final * !costo_unitario, 2)
If .EditMode = adEditInProgress Then
.Update
End If
End If
End With
rstProducto.Close: Set rstProducto = Nothing
Next i
rstDetalleFactura.Close: Set rstDetalleFactura = Nothing
'Esta función es similar al código del botón guardar del formulario
frmSalidasAlmacen.
ActualizarSalidaAlmacen

cnnControlInv.CommitTrans
Call LiberarConex
DesactivarBotones
cbxdescripcion.Enabled = False
txtcantidad.Enabled = False
cmdImprimir.Enabled = True
cmdBuscarcli.Enabled = False
cmdNuevocli.Enabled = False
End Sub
```

Este procedimiento cancela los registros que se hayan agregado al flexgrid.

```
Public Sub cmdCancelar_Click()
Dim i As Integer
For i = filasfac To 1 Step -1
flexDetalleFactura.RemoveItem i
Next i
DesactivarBotones
cmdBuscarcli.Enabled = True
cmdNuevocli.Enabled = True
lblTotalfac.Caption = "0.00"
filasfac = 0
End Sub
```

Este procedimiento imprime la factura que se entregara al cliente por su compra.

```
Private Sub cmdImprimir_Click()
DE_ForReport.cnnControlInv.Open
DE_ForReport.cmdFactura Trim(txtcodigofac.Text), Trim(Nbodega)
Rpt_Factura.PrintReport True
If MsgBox("¿Desea realmente imprimir la Salida del Almacén? ", vbYesNo) = vbYes
Then
DE_ForReport.cmdSalida CodigoSal, Trim(Nbodega)
Rpt_Salida.PrintReport True
End If
DE_ForReport.cnnControlInv.Close
cmdImprimir.Enabled = False
End Sub
```



Este procedimiento finaliza la creación de factura y pregunta si desea guardar la factura si aun no ha sido guardada esta.

```
Private Sub cmdSalir_Click()  
    If Not FacturaGuardada And filasfac > 0 Then  
        If MsgBox("¿Desea guardar la Factura.", vbYesNo) = vbYes Then  
            cmdGuardar_Click  
            Exit Sub  
        Else  
            cmdCancelar_Click  
        End If  
    End If  
    CodigoFac = ""  
    CodigoSal = ""  
    Unload Me  
End Sub
```

Código de la Ilustración 15: Cotización del cliente.

Este procedimiento valida y agrega un producto al control flexgrid el cual ha sido pedido por un cliente para ver su precio.

```
Private Sub cmdAgregarprod_Click()  
    Dim Totalcot As Currency  
    If cbxdescripcion.Text = "" Then  
        MsgBox "Falta descripción del Producto.", vbInformation  
        cbxdescripcion.SetFocus  
        Exit Sub  
    ElseIf CCur(txtcantidad.Text) = 0 Then  
        MsgBox "Falta la cantidad del Producto.", vbInformation  
        txtcantidad.SetFocus  
        Exit Sub  
    ElseIf ComprobarProducto Then  
        LimpiarCajas  
        Exit Sub  
    End If  
    With flexDetalleCotizacion  
        filascot = filascot + 1  
        .AddItem txtcodigoprod.Text, filascot  
        .TextMatrix(filascot, 1) = txtcodigoprod.Text  
        .TextMatrix(filascot, 2) = cbxdescripcion.Text  
        .TextMatrix(filascot, 3) = txtunidad.Text  
        .TextMatrix(filascot, 4) = FormatNumber(CCur(txtcantidad.Text), 2)  
        .TextMatrix(filascot, 5) = FormatNumber(CCur(txtprecio.Text), 2)  
        .TextMatrix(filascot, 6) = FormatNumber(CCur(txtsubtotal.Text), 2)  
        .TextMatrix(filascot, 7) = FormatNumber(varcosto, 2)  
    End With  
    Totalcot = CCur(lblTotalcot.Caption)  
    lblTotalcot.Caption = FormatNumber(Totalcot + CCur(txtsubtotal.Text), 2)  
    LimpiarCajas  
    cmdAgregarprod.Enabled = False  
    cmdGuardar.Enabled = True  
    cmdCancelar.Enabled = True  
End Sub
```



Este procedimiento elimina el producto seleccionado por el usuario del flexgrid.

```
Private Sub cmdEliminar_Click()  
    With flexDetalleCotizacion  
        If .Row = .Rows - 1 And .Row < .Rows Then  
            MsgBox "No puedes eliminar una fila en blanco", vbInformation, "Error al  
eliminar..."  
            .Row = 1  
            Exit Sub  
        End If  
        If MsgBox("Desea quitar este Producto del detalle?", vbYesNo + vbQuestion) =  
vbYes Then  
            txtexistencia.Text = 0  
            lblTotalcot.Caption = FormatNumber(CCur(lblTotalcot.Caption) -  
CCur(.TextMatrix(.Row, 6)), 2)  
            .RemoveItem .Row  
            filascot = filascot - 1  
            LimpiarCajas  
        End If  
    End With  
    If filascot = 0 Then  
        DesactivarBotones  
        txtigv.Text = 15  
    End If  
End Sub
```

Este procedimiento guarda la cotizacion de los productos con los datos del cliente para facturar si el cliente deside llevarlos mas tarde.

```
Private Sub cmdGuardar_Click()  
    Dim i As Integer  
    If txtnombrecli.Text = "" Then  
        MsgBox "Falta los datos del Cliente", vbInformation, "Cotizando..."  
        cmdBuscarcli.Enabled = True  
        cmdNuevocli.Enabled = True  
        Exit Sub  
    End If  
    'Guardar los datos de la cotizacion  
    If Not EstablecerConex Then  
        End  
    End If  
    cnnControlInv.BeginTrans  
    CodificarCotizacion  
    txtcodigocot.Text = CodigoCot  
  
    rstCotizacion.Open "Cotizacion", cnnControlInv, adOpenStatic, adLockOptimistic  
    rstDetalleCotizacion.Open "Detalle_Cotizacion", cnnControlInv, adOpenStatic,  
adLockOptimistic  
    With rstCotizacion  
        'Agregar el registro en la tabla Cotizacion.  
        .AddNew  
        !codigo_bodega = txtcodigobod.Text  
        !ncotizacion = Trim(txtcodigocot.Text)  
        !fecha = txtfechacot.Text  
        !codigo_cli = txtcodigocli.Text  
    End With
```



```
!nombre_cli = txtnombrecli.Text
!direccion = txtdireccioncli.Text
If Len(txttelefonocli.Text) <> 0 Then _
!telefono = txttelefonocli.Text
!importe_cotizacion = CCur(lblTotalcot.Caption)
!igv_cotizacion = CCur(lbltotaligv.Caption)
!monto_cotizacion = CCur(lblmontocot.Caption)
.Update
End With
rstCotizacion.Close: Set rstCotizacion = Nothing
With rstDetalleCotizacion
  For i = 1 To filascot
    .AddNew
    !ncotizacion = Trim(txtcodigocot.Text)
    !codigo_prod = flexDetalleCotizacion.TextMatrix(i, 1)
    !descripcion = flexDetalleCotizacion.TextMatrix(i, 2)
    !unidad = flexDetalleCotizacion.TextMatrix(i, 3)
    !cantidad = CCur(flexDetalleCotizacion.TextMatrix(i, 4))
    !precio = CCur(flexDetalleCotizacion.TextMatrix(i, 5))
    !subtotal = CCur(flexDetalleCotizacion.TextMatrix(i, 6))
    !costo = CCur(flexDetalleCotizacion.TextMatrix(i, 7))
    .Update
  Next i
End With
bolCotizacioGuardada = True
rstDetalleCotizacion.Close: Set rstDetalleCotizacion = Nothing
cnnControlInv.CommitTrans
Call LiberarConex
DesactivarBotones
cbxdescripcion.Enabled = False
txtcantidad.Enabled = False
cmdImprimir.Enabled = True
End Sub
```

Este procedimiento cancela la cotización y elimina los productos agregados a la misma.

```
Private Sub cmdCancelar_Click()
  Dim i As Integer
  For i = filascot To 1 Step -1
    flexDetalleCotizacion.RemoveItem i
  Next i
  DesactivarBotones
  txtcantidad.Enabled = False
  cmdBuscarcli.Enabled = True
  cmdNuevocli.Enabled = True
  lblTotalcot.Caption = "0.00"
  filascot = 0
End Sub
```



Este procedimiento imprime la cotización que se entregara al cliente.

```
Private Sub cmdImprimir_Click()  
    DE_ForReport.cnnControlInv.Open  
    DE_ForReport.cmdCotizacion Trim(txtcodigocot.Text), Trim(Nbodega)  
    Rpt_Cotizacion.PrintReport True  
    cmdImprimir.Enabled = False  
    DE_ForReport.cnnControlInv.Close  
End Sub
```

Este procedimiento finaliza la creación de la cotización y pregunta si desea guardarla si aun no ha sido guardada.

```
Private Sub cmdSalir_Click()  
    If Not bolCotizacioGuardada And filascot > 0 Then  
        If MsgBox("¿Desea guardar la Cotizacion de la lista de los Productos ",  
vbYesNo) = vbYes Then  
            cmdGuardar_Click  
            Exit Sub  
        Else  
            cmdCancelar_Click  
        End If  
    End If  
    bolCotizacionvisible = False  
    CodigoCot = ""  
    Unload Me  
End Sub
```

Código de la Ilustración 16: Cierre del mes en proceso.

Este procedimiento finaliza el periodo del mes de proceso del inventario y cierra automáticamente el programa.

```
Private Sub cmdEstablecer_Click()  
    Dim cmdBorrar As New ADODB.Command  
    If Not EstablecerConex Then  
        End  
    End If  
    cnnControlInv.BeginTrans  
    cmdBorrar.ActiveConnection = cnnControlInv.ConnectionString  
    cmdBorrar.CommandText = "DELETE FROM FechaProceso"  
    cmdBorrar.Execute  
    cnnControlInv.CommitTrans  
    Call LiberarConex  
    MsgBox "Usted ha cerrado el mes en proceso de Inventario.", , "Cierre del mes."  
    End  
End Sub
```

Este procedimiento cancela el cierre del mes de proceso del inventario.

```
Private Sub cmdCancelar_Click()  
    Unload Me  
End Sub
```

**Código de la Ilustración 17: Registrar nuevo producto.**

Este procedimiento codifica de forma automática el código del nuevo producto, limpia y valida los controles para agregarlo al sistema..

```
Private Sub cmdNuevo_Click()  
    ProductoGuardado = False  
    If Not EstablecerConex Then  
        End  
    End If  
    'Función para el código del nuevo Producto.  
    CodificarProducto (bodega)  
    mskcodigoprod.Text = CódigoProd  
    Call LiberarConex  
    txtdescripcion.Text = ""  
    txtunidad.Text = ""  
    cbxmoneda.Text = "C"  
    txtgrupo.Text = ""  
    txtsubgrupo.Text = ""  
    txtubicacion.Text = ""  
    txtdiasantesven.Text = ""  
    ActivarControles  
    cmdAceptar.Enabled = True  
    cmdNuevo.Enabled = False  
End Sub
```

Este procedimiento verifica que los datos del producto estén completos, se verifica si el producto tiene caducidad y si existe en el sistema y, se guarda en la base de datos.

```
Private Sub cmdAceptar_Click()  
    On Error GoTo Error_Aceptar  
    If Len(Trim(txtdescripcion.Text)) = 0 Then  
        MsgBox "Falta la descripción del Producto.", vbInformation, "Registrar un nuevo  
Producto."  
        txtdescripcion.SetFocus  
        Exit Sub  
    ElseIf Len(txtunidad.Text) = 0 Then  
        MsgBox "Falta la unidad del Producto.", vbInformation, "Registrar un nuevo  
Producto."  
        txtunidad.SetFocus  
        Exit Sub  
    ElseIf Len(txtgrupo.Text) = 0 Then  
        MsgBox "Falta el grupo al cual pertenece el Producto.", vbInformation,  
"Registrar un nuevo Producto."  
        txtgrupo.SetFocus  
        Exit Sub  
    ElseIf Len(txtubicacion.Text) = 0 Then  
        MsgBox "Falta la ubicación del Producto.", vbInformation, "Registrar un nuevo  
Producto."  
        txtubicacion.SetFocus  
        Exit Sub  
    ElseIf chktieneVenc.Value = 1 And Val(txtdiasantesven.Text) = 0 Then  
        MsgBox "Debes de introducir cuantos días antes del vencimiento de un producto  
te va a informar el sistema.", vbInformation, "Registrar un nuevo Producto."  
        txtunidad.SetFocus  
        Exit Sub  
End Sub
```



```
End If
'Establecer la conexión
If Not EstablecerConex Then
    End
End If
CodificarProducto (bodega)
mskcodigoprod.Text = CodigoProd

'verificar si el producto ya existe
If Verificarduplica(UCase(Trim(txtdescripcion.Text))) Then
    Call LiberarConex
    Exit Sub
End If
'Abrir la tabla Producto para guardar el registro.
rstProducto.Open "Producto", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdTable
With rstProducto
    .AddNew
    !codigo_bodega = msknumbodega.Text
    !codigo_prod = mskcodigoprod.Text
    !descripcion = UCase(Trim(txtdescripcion.Text))
    If chktieneVenc.Value = 1 Then
        !tiene_caducidad = 1
        !tiempo_caducidad = CInt(txtdiasantesven.Text)
    End If
    !unidad = UCase(txtunidad.Text)
    !moneda = cbxmoneda.Text
    !grupo = UCase(txtgrupo.Text)
    If Len(txtsubgrupo.Text) > 0 Then _
        !subgrupo = UCase(txtsubgrupo.Text)
    !ubicacion = UCase(txtubicacion.Text)
    'Actualizar la Tabla Producto
    .Update
End With
rstProducto.Close: Set rstProducto = Nothing
Call LiberarConex
ProductoGuardado = True
cmdNuevo.Enabled = True
cmdAceptar.Enabled = False
desactivarControles
'En caso de agregar desde la Entrada
If bolEntradavisible Then
    frmEntradasAlmacen.cbxdescripcion_Entrada.Text = UCase(txtdescripcion.Text)
    frmEntradasAlmacen.cbxdescripcion_Entrada.AddItem UCase(txtdescripcion.Text)
    frmEntradasAlmacen.cbxdescripcion_Entrada_Click
    Unload Me
End If
Exit Sub
Error_Aceptar:
MsgBox "Error al registrar los Datos de un nuevo Producto. " & Err.Description, ,
"Registrar un nuevo Producto..."
Call LiberarConex
cmdNuevo.Enabled = False
cmdAceptar.Enabled = True
txtdescripcion.SetFocus
End Sub
```



Este procedimiento cancela el agregado de productos a la base de datos.

```
Private Sub cmdCancelar_Click()  
    If Not ProductoGuardado Then  
        If (MsgBox("¿Desea guardar los cambios antes de salir?", vbYesNo + vbQuestion)  
= vbYes) Then  
            cmdAceptar_Click  
        Else  
            Unload Me  
        End If  
    Else  
        Unload Me  
    End If  
End Sub
```

Código de la Ilustración 18: Registrar un nuevo cliente.

Este procedimiento captura los datos de un nuevo cliente y los valida, su código lo genera automáticamente y verifica que no haya un cliente repetido en el sistema.

```
Private Sub cmdAceptar_Click()  
    On Error GoTo Error_Aceptar  
    If Len(Trim(txtnombrecli.Text)) = 0 Then  
        MsgBox "Debes de ingresar el Nombre del Cliente.", vbInformation, "Error en los  
Datos del Cliente..."  
        txtnombrecli.SetFocus  
        Exit Sub  
    End If  
    If Not EstablecerConex Then  
        End  
    End If  
    'Verificar si existe este Cliente.  
    If Verificarduplicaccli(UCase(Trim(txtnombrecli.Text))) Then  
        Call LiberarConex  
        Exit Sub  
    End If  
    rstCliente.Open "Cliente", cnnControlInv, adOpenStatic, adLockOptimistic,  
adCmdTable  
    With rstCliente  
        .AddNew  
        'Guardar el registro en la tabla Cliente.  
        !codigo_cli = mskcodigocli.Text  
        !nombre_cli = UCase(Trim(txtnombrecli.Text))  
        If mskcedula.Text <> "___-____-____" Then _  
            !cedula = UCase(Trim(mskcedula.Text))  
        If mskTelefono.Text <> "___-____" Then _  
            !telefono = Trim(mskTelefono.Text)  
        !direccion = UCase(txtdireccioncli.Text)  
        'intentar actualizar.  
        .Update  
        If bolCotizacionvisible Then  
            frmCotizacion.txtcodigocli = !codigo_cli  
            frmCotizacion.txtnombrecli = !nombre_cli  
            frmCotizacion.txtdireccioncli = !direccion  
            If Len(!telefono) <> 0 Then  
                frmCotizacion.txttelefonocli = !telefono
```



```
Else
    frmCotizacion.txttelefonocli = ""
End If
ElseIf bolFacturavisible Then
    frmFactura.txtcodigocli = !codigo_cli
    frmFactura.txtnombrecli = !nombre_cli
    frmFactura.txtdireccioncli = !direccion
    If Len(!telefono) <> 0 Then
        frmFactura.txttelefonocli = !telefono
    Else
        frmFactura.txttelefonocli = ""
    End If
End If
End With
'Cerrar el recordset y la conexión.
rstCliente.Close: Set rstCliente = Nothing
Call LiberarConex
MsgBox "El Cliente " & UCase(txtnombrecli.Text) & " se ha creado con éxito. ",
vbInformation, "Registro del Cliente..."
Unload Me
Exit Sub
Error_Aceptar:
MsgBox "Error al registrar un nuevo Cliente. Verifique que los datos esten
correctos.", vbInformation, "Error al registrar los Datos..."
Call LiberarConex
End Sub
```

Esta función verifica si el cliete ingresado ya existe de lo contrario se agrega al sistema.

```
Function Verificarduplicacli(nombrecli As String) As Boolean
    rstCliente.Open "SELECT nombre_cli FROM Cliente WHERE nombre_cli ='" &
(nombrecli) & "';", cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    If rstCliente.RecordCount Then
        Verificarduplicacli = True
        MsgBox "El Cliente " & nombrecli & " ya existe.", vbInformation, "Clientes."
    Else
        Verificarduplicacli = False
    End If
    rstCliente.Close: Set rstCliente = Nothing
End Function
```

Este procedimiento cancela la introducción de un clientes al sistema.

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```

**Código de la Ilustración 19: Registrar un nuevo proveedor.**

Este procedimiento captura los datos de un nuevo proveedor y los valida, su código lo genera automáticamente y verifica que no haya un proveedor repetido en el sistema.

```
Private Sub cmdAceptar_Click()
    On Error GoTo Error_Aceptar
    If Len(txtnombreprov.Text) = 0 Then
        MsgBox "Debes ingresar el Nombre del Proveedor.", vbInformation, "Error en los
Datos del Proveedor..."
        txtnombreprov.SetFocus
        Exit Sub
    ElseIf Len(txtdireccionprov.Text) = 0 Then
        MsgBox "Debes ingresar la dirección del Proveedor.", vbInformation, "Error en
los Datos del Proveedor..."
        txtdireccionprov.SetFocus
        Exit Sub
    ElseIf msktelefonoprov.Text = "###-####" Then
        MsgBox "Debes ingresar el # de teléfono del Proveedor.", vbInformation, "Error
en los Datos del Proveedor..."
        msktelefonoprov.SetFocus
        Exit Sub
    End If

    If Not EstablecerConex Then
        End
    End If
    'Verificar si el proveedor ya existe.
    If Verificarduplicaprov(UCASE(Trim(txtnombreprov.Text))) Then
        Call LiberarConex
        Exit Sub
    End If

    rstProveedor.Open "Proveedor", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdTable
    With rstProveedor
        .AddNew
        'Guardar el registro en la tabla Proveedor.
        !codigo_prov = mskcodigoprov.Text
        !nombre_prov = UCASE(Trim(txtnombreprov.Text))
        !direccion = UCASE(Trim(txtdireccionprov.Text))
        !telefono = msktelefonoprov.Text
        'intentar actualizar.
        .Update
        If bolEntradavisible Then
            frmEntradasAlmacen.txtcodigoprov = !codigo_prov
            frmEntradasAlmacen.txtnombreprov = !nombre_prov
        End If
    End With
    'Cerrar el recordset y la conexión.
    rstProveedor.Close: Set rstProveedor = Nothing
    Call LiberarConex
    MsgBox "El Proveedor " & UCASE(txtnombreprov.Text) & " se ha creado con éxito. ",
vbInformation, "Registro del Proveedor..."
    Unload Me
    Exit Sub
End Sub
```



```
Error_Aceptar:
    MsgBox "Error al registrar un nuevo Proveedor. Verifique que los datos esten
correctos.", vbInformation, "Error al registrar los Datos..."
    Call LiberarConex
End Sub
```

Código de la Ilustración 20: Reporte de entrada.

Este procedimiento valida que las fechas sean correctas e imprime el reporte de todas las entradas realizadas en el rango de fecha establecido.

```
Private Sub cmdImprimir_Click()
    If Not IsDate(mskFechaini.Text) Then
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la
fecha."
        mskFechaini.SetFocus
        Exit Sub
    ElseIf Not IsDate(mskFechafin.Text) Then
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."
        mskFechafin.SetFocus
        Exit Sub
    Else
        'La fecha inicial y final deben ser menor o igual a la fecha actual
        If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then
            DE_ForReport.cnnControlInv.Open
            DE_ForReport.cmdTodaEntrada Trim(mskFechaini.Text), Trim(mskFechafin.Text),
-
                                Trim(cbxtipo_doc.Text), Trim(Nbodega)
            'Comprobar si generó algún resultado el recordset
            If DE_ForReport.rscmdTodaEntrada.RecordCount > 0 Then
                Rpt_EntradaAlmacen.PrintReport True
                cmdImprimir.Enabled = False
            Else
                MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de
las Entradas."
            End If
            DE_ForReport.cnnControlInv.Close
        Else
            MsgBox "Las fechas iniciales y finales" _
                & " deben ser menores que la actual.", vbCritical, "Error al introducir las
fechas."
        End If
    End If
End Sub
```

Este procedimiento valida que las fechas sean correctas y visualiza el reporte de todas las entradas realizadas en el rango de fecha establecido.

```
Private Sub cmdVisualizar_Click()
    If Not IsDate(mskFechaini.Text) Then
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la
fecha."
        mskFechaini.SetFocus
        Exit Sub
    ElseIf Not IsDate(mskFechafin.Text) Then
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."
```



```
    mskFechafin.SetFocus
    Exit Sub
Else
    'La fecha inicial debe ser menor o igual a la fecha actual del sistema
    'al igual que la fecha final
    If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then
        DE_ForReport.cnnControlInv.Open
        DE_ForReport.cmdTodaEntrada Trim(mskFechaini.Text), Trim(mskFechafin.Text),
-
        Trim(cbxtipo_doc.Text), Trim(Nbodega)
        'Comprobar si generó algún resultado el recordset
        If DE_ForReport.rscmdTodaEntrada.RecordCount > 0 Then
            Rpt_EntradaAlmacen.Show vbModal
            cmdVisualizar.Enabled = False
        Else
            MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de
las Entradas."
        End If
        DE_ForReport.cnnControlInv.Close
    Else
        MsgBox "Las fechas iniciales y finales" _
        & " deben ser menores que la actual.", vbCritical, "Error al introducir las
fechas."
    End If
End If
End Sub
```

Código de la Ilustración 21: Reporte de salida.

Este procedimiento valida que las fechas sean correctas e imprime el reporte de todas las salidas realizadas en el rango de fecha establecido.

```
Private Sub cmdImprimir_Click()
    If Not IsDate(mskFechaini.Text) Then
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la
fecha."
        mskFechaini.SetFocus
        Exit Sub
    ElseIf Not IsDate(mskFechafin.Text) Then
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."
        mskFechafin.SetFocus
        Exit Sub
    Else
        'La fecha inicial debe ser menor o igual a la fecha actual del sistema
        'al igual que la fecha final
        If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then
            DE_ForReport.cnnControlInv.Open
            DE_ForReport.cmdTodaSalida Trim(mskFechaini.Text), Trim(mskFechafin.Text),
-
            Trim(cbxtipo_doc.Text), Trim(Nbodega)
            'Comprobar si generó algún resultado el recordset
            If DE_ForReport.rscmdTodaSalida.RecordCount > 0 Then
                Rpt_TodaSalida.PrintReport True
                cmdImprimir.Enabled = False
            End If
        End If
    End If
End Sub
```



```
        Else
            MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de
las Entradas."
        End If
        DE_ForReport.cnnControlInv.Close
    Else
        MsgBox "Las fechas iniciales y finales" _
            & " deben ser menores que la actual.", vbCritical, "Error al introducir las
fechas."
    End If
End If
End Sub
```

Este procedimiento valida que las fechas sean correctas y visualiza el reporte de todas las salidas realizadas en el rango de fecha establecido.

```
Private Sub cmdVisualizar_Click()
    If Not IsDate(mskFechaini.Text) Then
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la
fecha."
        mskFechaini.SetFocus
        Exit Sub
    ElseIf Not IsDate(mskFechafin.Text) Then
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."
        mskFechafin.SetFocus
        Exit Sub
    Else
        'La fecha inicial debe ser menor o igual a la fecha actual del sistema
        'al igual que la fecha final
        If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then
            DE_ForReport.cnnControlInv.Open
            DE_ForReport.cmdTodaSalida Trim(mskFechaini.Text), Trim(mskFechafin.Text),
            Trim(cbxtipo_doc.Text), Trim(Nbodega)
            'Comprobar si generó algún resultado el recordset
            If DE_ForReport.rscmdTodaSalida.RecordCount > 0 Then
                Rpt_TodaSalida.Show vbModal
                cmdVisualizar.Enabled = False
            Else
                MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de
las Entradas."
            End If
            DE_ForReport.cnnControlInv.Close
        Else
            MsgBox "Las fechas iniciales y finales" _
                & " deben ser menores que la actual.", vbCritical, "Error al introducir las
fechas."
        End If
    End If
End Sub
```

**Código de la Ilustración 22: Reporte de ajuste.**

Este procedimiento valida que las fechas sean correctas e imprime el reporte de todos los ajustes realizados en el rango de fecha establecido.

```
Private Sub cmdImprimir_Click()  
    If Not IsDate(mskFechaini.Text) Then  
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la  
fecha."  
        mskFechaini.SetFocus  
        Exit Sub  
    ElseIf Not IsDate(mskFechafin.Text) Then  
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."  
        mskFechafin.SetFocus  
        Exit Sub  
    Else  
        'La fecha inicial debe ser menor o igual a la fecha actual del sistema  
        'al igual que la fecha final  
        If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then  
            DE_ForReport.cnnControlInv.Open  
            DE_ForReport.cmdTodoAjuste Trim(mskFechaini.Text), Trim(mskFechafin.Text),  
Trim(cbxtipo_doc.Text), Trim(Nbodega)  
            'Comprobar si generó algún resultado el recordset  
            If DE_ForReport.rscmdTodoAjuste.RecordCount > 0 Then  
                Rpt_TodoAjuste.PrintReport True  
                cmdImprimir.Enabled = False  
            Else  
                MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de  
las Entradas."  
                End If  
            DE_ForReport.cnnControlInv.Close  
        Else  
            MsgBox "Las fechas iniciales y finales" _  
            & " deben ser menores que la actual.", vbCritical, "Error al introducir las  
fechas."  
            End If  
        End If  
    End If  
End Sub
```

Este procedimiento valida que las fechas sean correctas y visualiza el reporte de todos los ajustes realizados en el rango de fecha establecido.

```
Private Sub cmdVisualizar_Click()  
    If Not IsDate(mskFechaini.Text) Then  
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la  
fecha."  
        mskFechaini.SetFocus  
        Exit Sub  
    ElseIf Not IsDate(mskFechafin.Text) Then  
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."  
        mskFechafin.SetFocus  
        Exit Sub  
    Else  
        'La fecha inicial debe ser menor o igual a la fecha actual del sistema  
        'al igual que la fecha final  
        If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then
```



```
DE_ForReport.cnnControlInv.Open
DE_ForReport.cmdTodoAjuste Trim(mskFechaini.Text),Trim(mskFechafin.Text),
Trim(cbxtipo_doc.Text), Trim(Nbodega)
'Comprobar si generó algún resultado el recordset
If DE_ForReport.rscmdTodoAjuste.RecordCount > 0 Then
    Rpt_TodoAjuste.Show vbModal
    cmdVisualizar.Enabled = False
Else
    MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de
las Entradas."
End If
DE_ForReport.cnnControlInv.Close
Else
    MsgBox "Las fechas iniciales y finales" _
    & " deben ser menores que la actual.", vbCritical, "Error al introducir las
fechas."
End If
End If
End Sub
```

Código de la Ilustración 23: kardex por producto.

En este procedimiento se validan las fecha apartir de las cuales se presentaran todos los movimientos del producto seleccionado mediante consulta de entradas, salidas y ajustes en almacén.

```
Dim filasEntrada, filasSalida, filasAjuste As Integer
Private Sub cbxdescripcion_Click()
    Dim filEnt, filSal, filAju As Integer
    filEnt = 0: filSal = 0: filAju = 0
    If Not IsDate(mebFechaini.Text) Then
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la
fecha."
        mebFechaini.SetFocus
        Exit Sub
    Else
        If CDate(mebFechaini.Text) > Date Then
            MsgBox "La fecha inicial debe ser menor o igual que la final.",
vbInformation, "Error al ingresar la fecha."
            mebFechaini.SetFocus
            Exit Sub
        End If
    End If

    For filEnt = filasEntrada To 1 Step -1
        FlexGridEntrada.RemoveItem filEnt
    Next filEnt
    For filSal = filasSalida To 1 Step -1
        FlexGridSalida.RemoveItem filSal
    Next filSal
    For filAju = filasAjuste To 1 Step -1
        FlexGridAjuste.RemoveItem filAju
    Next filAju
    filasEntrada = 0: filasSalida = 0: filasAjuste = 0
```



```
If Not EstablecerConex Then
    End
End If
rstProducto.Open "SELECT codigo_prod,unidad,exis_final,costo_unitario,saldo FROM
Producto WHERE codigo_bodega = '" & txtBodega.Text & "' AND descripcion='" &
cbxDescripcion.Text & "' ORDER BY descripcion;", cnnControlInv, adOpenStatic,
adLockReadOnly, adCmdText
With rstProducto
    If .RecordCount > 0 Then
        txtCodigoProducto.Text = !codigo_prod
        txtUnidad.Text = !unidad
        txtCantidad.Text = FormatNumber(CCur(!exis_final), 2)
        txtCosto.Text = FormatNumber(CCur(!costo_unitario), 2)
        txtSaldo.Text = FormatNumber(CCur(!saldo), 2)
    End If
End With
rstProducto.Close: Set rstProducto = Nothing

rstEntradaAlmacen.Open "select
Entrada_Almacen.codigo_entrada,Entrada_Almacen.fecha_entrada,Detalle_Entrada.cantidad,D
etalle_Entrada.costo_unitario,Detalle_Entrada.sub_total from Entrada_Almacen INNER JOIN
Detalle_Entrada ON Entrada_Almacen.codigo_entrada=Detalle_Entrada.codigo_entrada WHERE
Detalle_Entrada.codigo_prod ='" & txtCodigoProducto.Text & "' and
Entrada_Almacen.fecha_entrada>='" & mebFechaini.Text & "' and
Entrada_Almacen.fecha_entrada<='" & Date & "' and Entrada_Almacen.codigo_bodega = '" &
Nbodega & "' order by Entrada_Almacen.fecha_entrada;", cnnControlInv, adOpenStatic,
adLockReadOnly, adCmdText
With rstEntradaAlmacen
    Do Until .EOF
        filasEntrada = filasEntrada + 1
        FlexGridEntrada.AddItem !codigo_entrada, filasEntrada
        FlexGridEntrada.TextMatrix(filasEntrada, 1) = !codigo_entrada
        FlexGridEntrada.TextMatrix(filasEntrada, 2) = !fecha_entrada
        FlexGridEntrada.TextMatrix(filasEntrada, 3) = FormatNumber(CCur(!cantidad),
2)
        FlexGridEntrada.TextMatrix(filasEntrada, 4) =
FormatNumber(CCur(!costo_unitario), 2)
        FlexGridEntrada.TextMatrix(filasEntrada, 5) =
FormatNumber(CCur(!sub_total), 2)
        .MoveNext
    Loop
End With
rstEntradaAlmacen.Close: Set rstEntradaAlmacen = Nothing

rstSalidaAlmacen.Open "select
Salida_Almacen.codigo_salida,Salida_Almacen.fecha_salida,Detalle_Salida.cantidad,Detall
e_Salida.sub_total from Salida_Almacen INNER JOIN Detalle_Salida ON
Salida_Almacen.codigo_salida=Detalle_Salida.codigo_salida WHERE
Detalle_Salida.codigo_prod ='" & txtCodigoProducto.Text & "' and
Salida_Almacen.fecha_salida>='" & mebFechaini.Text & "' and
Salida_Almacen.fecha_salida<='" & Date & "' and Salida_Almacen.codigo_bodega = '" &
Nbodega & "' order by Salida_Almacen.fecha_salida;", cnnControlInv, adOpenStatic,
adLockReadOnly, adCmdText
With rstSalidaAlmacen
    Do Until .EOF
        filasSalida = filasSalida + 1
        FlexGridSalida.AddItem !codigo_salida, filasSalida
```



```
FlexGridSalida.TextMatrix(filasSalida, 1) = !codigo_salida
FlexGridSalida.TextMatrix(filasSalida, 2) = !fecha_salida
FlexGridSalida.TextMatrix(filasSalida, 3) = FormatNumber(CCur(!cantidad),
2)
FlexGridSalida.TextMatrix(filasSalida, 4) = FormatNumber(CCur(!sub_total /
!cantidad), 2)
FlexGridSalida.TextMatrix(filasSalida, 5) = FormatNumber(CCur(!sub_total),
2)
.MoveNext
Loop
End With
rstSalidaAlmacen.Close: Set rstSalidaAlmacen = Nothing

rstAjusteAlmacen.Open "select
Ajustes_Almacen.codigo_ajuste,Ajustes_Almacen.fecha_ajuste,Detalle_Ajuste.ajuste_costo
,Detalle_Ajuste.ajuste_exis,Detalle_Ajuste.costos_unitario from Ajustes_Almacen INNER
JOIN Detalle_Ajuste ON Ajustes_Almacen.codigo_ajuste=Detalle_Ajuste.codigo_ajuste WHERE
Detalle_Ajuste.codigo_prod ='" & txtCodigoProducto.Text & "' and
Ajustes_Almacen.fecha_ajuste>='" & mebFechaini.Text & "' and
Ajustes_Almacen.fecha_ajuste<=''" & Date & "' and Ajustes_Almacen.codigo_bodega = '" &
Nbodega & "' order by Ajustes_Almacen.fecha_ajuste;", cnnControlInv, adOpenStatic,
adLockReadOnly, adCmdText
With rstAjusteAlmacen
Do Until .EOF
filasAjuste = filasAjuste + 1
FlexGridAjuste.AddItem !codigo_ajuste, filasAjuste
FlexGridAjuste.TextMatrix(filasAjuste, 1) = !codigo_ajuste
FlexGridAjuste.TextMatrix(filasAjuste, 2) = !fecha_ajuste
If Len(!ajuste_costo) > 0 Then _
FlexGridAjuste.TextMatrix(filasAjuste, 4) =
FormatNumber(CCur(!ajuste_costo), 2)
If Len(!ajuste_exis) > 0 Then
FlexGridAjuste.TextMatrix(filasAjuste, 3) =
FormatNumber(CCur(!ajuste_exis), 2)
FlexGridAjuste.TextMatrix(filasAjuste, 6) =
FormatNumber(CCur(!ajuste_exis * !costo_unitario), 2)
End If
FlexGridAjuste.TextMatrix(filasAjuste, 5) =
FormatNumber(CCur(!costo_unitario), 2)
.MoveNext
Loop
End With
rstAjusteAlmacen.Close: Set rstAjusteAlmacen = Nothing

Call LiberarConex
End Sub
```

Este procedimiento permite cerrar la ventana del Kardex.

```
Private Sub cmdSalir_Click()
Unload Me
End Sub
```



Este procedimiento hace una consulta de la fecha de inicio del inventario y de todos los productos de la bodega visualizandolo en un combo, ademas, crea los nombres de las columnas de todos los controles flexgrid.

```
Private Sub Form_Load()  
    filasEntrada = 0: filasSalida = 0: filasAjuste = 0  
    txtBodega.Text = Nbodega  
  
    If Not EstablecerConex Then  
        End  
    End If  
  
    rstProducto.Open "SELECT descripcion FROM Producto WHERE  codigo_bodega = '" &  
(Nbodega) & "' ORDER BY descripcion;", cnnControlInv, adOpenStatic, adLockReadOnly,  
adCmdText  
    With rstProducto  
        While Not .EOF  
            cbxDescripcion.AddItem (!descripcion)  
            .MoveNext  
        Wend  
    End With  
    rstProducto.Close: Set rstProducto = Nothing  
  
    'Abrir la tabla FechaProceso .  
    rstFechaProceso.Open "SELECT fecha_inicio FROM FechaProceso", cnnControlInv,  
adOpenDynamic, adLockReadOnly, adCmdText  
    If Not rstFechaProceso.EOF Then  
        mebFechaIni.Text = rstFechaProceso!fecha_inicio  
    End If  
    rstFechaProceso.Close: Set rstFechaProceso = Nothing  
    Call LiberarConex  
  
    mebFechaFin.Text = Date  
  
    With FlexGridEntrada  
        .FormatString = "|<Código Entrada|<Fecha Entrada|>Cantidad|>Costo  
Unitario|>Sub Total"  
        .ColWidth(0) = 0 'esta columna no se visualiza.  
        .ColWidth(1) = 1500  
        .ColWidth(2) = 1500  
        .ColWidth(3) = 1500  
        .ColWidth(4) = 1500  
        .ColWidth(5) = 1500  
    End With  
    With FlexGridSalida  
        .FormatString = "|<Código Salida|<Fecha Salida|>Cantidad|>Costo  
Unitario|>Sub Total"  
        .ColWidth(0) = 0 'esta columna no se visualiza.  
        .ColWidth(1) = 1500  
        .ColWidth(2) = 1500  
        .ColWidth(3) = 1500  
        .ColWidth(4) = 1500  
        .ColWidth(5) = 1500  
    End With  
    With FlexGridAjuste  
        .FormatString = "|<Código Ajuste|>Fecha Ajuste|>Cantidad Ajustada|>Costo  
Ajustado|>Costo Unit.|>Sub Total"
```



```
.ColWidth(0) = 0 'esta columna no se visualiza.  
.ColWidth(1) = 1260  
.ColWidth(2) = 1200  
.ColWidth(3) = 1640  
.ColWidth(4) = 1400  
.ColWidth(5) = 1100  
.ColWidth(6) = 970  
  
End With  
End Sub
```

Código de la Ilustración 24: Reporte de factura.

Este procedimiento valida que las fechas sean correctas e imprime el reporte de todas las facturas emitidas en el rango de fecha establecido.

```
Private Sub cmdImprimir_Click()  
    If Not IsDate(mskFechaini.Text) Then  
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la  
fecha."  
        mskFechaini.SetFocus  
        Exit Sub  
    ElseIf Not IsDate(mskFechafin.Text) Then  
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."  
        mskFechafin.SetFocus  
        Exit Sub  
    Else  
        'La fecha inicial debe ser menor o igual a la fecha actual del sistema  
        'al igual que la fecha final  
        If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then  
            DE_ForReport.cnnControlInv.Open  
            DE_ForReport.cmdTodaFactura Trim(mskFechaini.Text),  
Trim(mskFechafin.Text),Trim(Nbodega)  
            'Comprobar si generó algún resultado el recordset  
            If DE_ForReport.rscmdTodaFactura.RecordCount > 0 Then  
                Rpt_TodaFactura.PrintReport True  
                cmdImprimir.Enabled = False  
            Else  
                MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de  
las Entradas."  
                End If  
            DE_ForReport.cnnControlInv.Close  
        Else  
            MsgBox "Las fechas iniciales y finales" _  
            & " deben ser menores que la actual.", vbCritical, "Error al introducir las  
fechas."  
            End If  
        End If  
    End If  
End Sub
```

Este procedimiento valida que las fechas sean correctas y visualiza el reporte de todas las facturas emitidas en el rango de fecha establecido.

```
Private Sub cmdVisualizar_Click()  
    If Not IsDate(mskFechaini.Text) Then  
        MsgBox "Fecha inicial incorrecta.", vbInformation, "Error al ingresar la  
fecha."
```



```
        mskFechaini.SetFocus
    Exit Sub
ElseIf Not IsDate(mskFechafin.Text) Then
    MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."
    mskFechafin.SetFocus
    Exit Sub
Else
    'La fecha inicial debe ser menor o igual a la fecha actual del sistema
    'al igual que la fecha final
    If CDate(mskFechaini.Text) <= Date And CDate(mskFechafin.Text) <= Date Then
        DE_ForReport.cnnControlInv.Open
        DE_ForReport.cmdTodaFactura Trim(mskFechaini.Text), Trim(mskFechafin.Text),
Trim(Nbodega)
        'Comprobar si generó algún resultado el recordset
        If DE_ForReport.rscmdTodaFactura.RecordCount > 0 Then
            Rpt_TodaFactura.Show vbModal
            cmdVisualizar.Enabled = False
        Else
            MsgBox "No se encontro ningun resultado.", vbInformation, "Consulta de
las Entradas."
        End If
        DE_ForReport.cnnControlInv.Close
    Else
        MsgBox "Las fechas iniciales y finales" _
& " deben ser menores que la actual.", vbCritical, "Error al introducir las
fechas."
    End If
End If
End Sub
```

Código de la Ilustración 25: Búsqueda de producto.

Este procedimiento hace una búsqueda de los productos através del código.

```
Public Sub cmdPorCodigo_Click()
    opcion = False : lblsel.Enabled = True
    lstProductos.Enabled = True
    cmdPorCodigo.Enabled = False
    cmdPorDescripcion.Enabled = True
    cmdMostrar.Enabled = False
    lstProductos.Clear
    If Not EstablecerConex Then
        End
    End If
    rstProducto.Open "SELECT codigo_prod FROM Producto WHERE codigo_bodega ='&
Nbodega & "'ORDER BY codigo_prod;";", cnnControlInv, adOpenStatic, adLockReadOnly,
adCmdText
    With rstProducto
        While Not .EOF
            lstProductos.AddItem (!codigo_prod)
            .MoveNext
        Wend
    End With
    rstProducto.Close: Set rstProducto = Nothing
    Call LiberarConex
End Sub
```



Este procedimiento hace una búsqueda de los productos por la descripción.

```
Public Sub cmdPorDescripcion_Click()  
    opcion = True  
    lblsel.Enabled = True  
    lstProductos.Enabled = True  
    cmdPorCodigo.Enabled = True  
    cmdPorDescripcion.Enabled = False  
    cmdMostrar.Enabled = False  
    lstProductos.Clear  
    If Not EstablecerConex Then  
        End  
    End If  
    rstProducto.Open "SELECT descripcion FROM Producto WHERE codigo_bodega =" &  
Nbodega & "' ORDER BY descripcion;", cnnControlInv, adOpenStatic, adLockReadOnly,  
adCmdText  
    With rstProducto  
        While Not .EOF  
            lstProductos.AddItem (!descripcion)  
            .MoveNext  
        Wend  
    End With  
    rstProducto.Close: Set rstProducto = Nothing  
    Call LiberarConex  
End Sub
```

Este procedimiento permite salir de la zona de búsqueda de los productos.

```
Private Sub cmdSalir_Click()  
    Unload Me  
End Sub
```

Este procedimiento muestra el producto seleccionado de una lista con todos sus datos.

```
Private Sub cmdMostrar_Click()  
    Dim i As Integer, x As Integer  
    x = 0  
    If Not EstablecerConex Then  
        End  
    End If  
    If opcion Then  
        rstProducto.Open "SELECT * FROM Producto WHERE descripcion=" & lstProductos &  
"'" and codigo_bodega =" & Nbodega & "';", cnnControlInv, adOpenStatic,  
adLockReadOnly, adCmdText  
    Else  
        rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod =" & lstProductos &  
"'" and codigo_bodega =" & Nbodega & "';", cnnControlInv, adOpenStatic, adLockReadOnly,  
adCmdText  
    End If  
    With frmVerProducto  
        .msknumbodega.Text = rstProducto!codigo_bodega  
        .mskcodigoprod.Text = rstProducto!codigo_prod  
        .txtdescripcion.Text = rstProducto!descripcion  
        .txtunidad.Text = rstProducto!unidad  
        .txtexistencias.Text = rstProducto!exis_final  
        .txtcosto.Text = rstProducto!costo_unitario
```



```
.txtsaldo.Text = rstProducto!saldo
.txtpreciol.Text = rstProducto!preciol
.txtprecio2.Text = rstProducto!precio2
.cbxmoneda.Text = rstProducto!moneda
.txtgrupo.Text = rstProducto!grupo
If Len(rstProducto!subgrupo) > 0 Then _
    .txtsubgrupo.Text = rstProducto!subgrupo
    .txtubicacion.Text = rstProducto!ubicacion
rstLote.Open "SELECT * FROM Lote_Poducto WHERE codigo_entrada LIKE 'E%' AND
codigo_prod='" & .mskcodigoprod.Text & "' ;", cnnControlInv, adOpenStatic,
adLockReadOnly, adCmdText
ReDim varlistafechacad(rstLote.RecordCount)
ReDim varlistacantlote(rstLote.RecordCount)
For i = 0 To rstLote.RecordCount - 1
    If Len(rstLote!codigo_entrada) > 0 Then
        .cbxlote.AddItem rstLote!numlote
        varlistafechacad(x) = rstLote!fecha_caducidad
        varlistacantlote(x) = rstLote!cantidad_fecha
        x = x + 1
    End If
    rstLote.MoveNext
Next i
'verificar si hay productos
If x > 0 Then
    .cbxlote.Text = .cbxlote.List(0)
    If Len(varlistafechacad(0)) Then
        .txtfechacad.Text = varlistafechacad(0)
    Else
        .txtfechacad.Text = "No tiene."
    End If
    .txtcantlote.Text = varlistacantlote(0)
End If
'Desactivar las cajas
.msknumbodega.Enabled = False
.mskcodigoprod.Enabled = False
.txtdescripcion.Enabled = False
.txtunidad.Enabled = False
.txtexistencias.Enabled = False
.txtcosto.Enabled = False
.txtsaldo.Enabled = False
.cbxmoneda.Enabled = False
.txtgrupo.Enabled = False
.txtsubgrupo.Enabled = False
.txtpreciol.Enabled = False
.txtprecio2.Enabled = False
.txtubicacion.Enabled = False
.txtfechacad.Enabled = False
.txtcantlote.Enabled = False
End With
frmVerProducto.Show vbModal
rstProducto.Close: Set rstProducto = Nothing
Call LiberarConex
End Sub
```



Código de la Ilustración 26: Muestra del producto.

Este procedimiento cierra la ventana donde se muestran todos los datos del producto.

```
Private Sub cmdCerrar_Click()  
    Unload Me  
End Sub
```

Código de la Ilustración 27: Búsqueda de cliente.

Este procedimiento hace una búsqueda de los clientes existentes y los muestra de manera ordenada por su nombre en una lista.

```
Private Sub Form_Load()  
    If Not EstablecerConex Then  
        End  
    End If  
    rstCliente.Open "SELECT nombre_cli FROM Cliente ORDER BY nombre_cli; ",  
cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText  
    With rstCliente  
        While Not .EOF  
            lstClientes.AddItem (!nombre_cli)  
            .MoveNext  
        Wend  
    End With  
    rstCliente.Close: Set rstCliente = Nothing  
    Call LiberarConex  
    cmdMostrar.Enabled = False  
End Sub
```

Este procedimiento muestra todos los datos del cliente seleccionado de una lista.

```
Private Sub cmdMostrar_Click()  
    If Not EstablecerConex Then  
        End  
    End If  
    rstCliente.Open "SELECT * FROM Cliente WHERE nombre_cli='" & lstClientes & "'",  
cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText  
    With frmVerCliente  
        .txtcodigocli = rstCliente!codigo_cli  
        .txtnombrecli = rstCliente!nombre_cli  
        If Len(rstCliente!direccion) > 0 Then _  
            .txtdireccioncli = rstCliente!direccion  
        If Len(rstCliente!cedula) > 0 Then  
            .mskcedulacli = rstCliente!cedula  
        Else  
            .mskcedulacli.Text = "___-____-____"  
        End If  
        If Len(rstCliente!telefono) <> 0 Then  
            .msktelefonocli = rstCliente!telefono  
        Else  
            .msktelefonocli.Text = "___-____"  
        End If  
    End With  
    Desactivarcajas  
    frmVerCliente.Show vbModal
```



```
rstCliente.Close: Set rstCliente = Nothing: Call LiberarConex
End Sub
```

Código de la Ilustración 28: Muestra del cliente.

Este procedimiento cierra la ventana donde se muestran los datos del cliente.

```
Private Sub cmdCerrar_Click()
    Unload Me
End Sub
```

Código de la Ilustración 29: Búsqueda de proveedor.

Este procedimiento hace una búsqueda de los proveedores existentes y los muestra de manera ordenada por su nombre en una lista.

```
Private Sub Form_Load()
    If Not EstablecerConex Then
        End
    End If
    rstProveedor.Open "SELECT nombre_prov FROM Proveedor ORDER BY nombre_prov;",
cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    With rstProveedor
        While Not .EOF
            lstProveedores.AddItem (!nombre_prov)
            .MoveNext
        Wend
    End With
    cmdMostrar.Enabled = False
    rstProveedor.Close: Set rstProveedor = Nothing
    Call LiberarConex
End Sub
```

Este procedimiento muestra los datos del proveedor seleccionado de una lista.

```
Private Sub cmdMostrar_Click()
    If Not EstablecerConex Then
        End
    End If
    rstProveedor.Open "SELECT * FROM Proveedor WHERE nombre_prov ='" & lstProveedores &
"';", cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    With frmVerProveedor
        .txtcodigoprov = rstProveedor!codigo_prov
        .txtnombreprov = rstProveedor!nombre_prov
        .txtdireccionprov = rstProveedor!direccion
        .msktelefonoprov = rstProveedor!telefono
        Desactivarcajas
    End With
    frmVerProveedor.Show vbModal
    rstProveedor.Close: Set rstProveedor = Nothing
    Call LiberarConex
End Sub
```

**Código de la Ilustración 30: Muestra del proveedor.**

Este procedimiento cierra la interfaz donde se muestran los datos del proveedor.

```
Private Sub cmdCerrar_Click()  
    Unload Me  
End Sub
```

Código de la Ilustración 31: Búsqueda de factura.

Este procedimiento hace una búsqueda de las facturas existentes que hayan sido canceladas o saldadas y las muestra ordenadas por el número de factura en una lista.

```
Option Explicit  
Private Sub Form_Load()  
    If Not EstablecerConex Then  
        End  
    End If  
    rstFactura.Open "SELECT nfactura FROM Factura WHERE codigo_bodega ='" & Nbodega &  
&"' AND estado='CANCELADA' ORDER BY nfactura;", cnnControlInv, adOpenStatic,  
adLockReadOnly, adCmdText  
    With rstFactura  
        While Not .EOF  
            lstFacturas.AddItem (!nFactura)  
            .MoveNext  
        Wend  
    End With  
    cmdMostrar.Enabled = False  
    rstFactura.Close: Set rstFactura = Nothing  
    Call LiberarConex  
End Sub
```

Este procedimiento muestra los datos de la factura seleccionada de la lista.

```
Private Sub cmdMostrar_Click()  
    frmVerFactura.Show vbModal  
End Sub
```

Código de la Ilustración 32: Muestra de la factura.

Este procedimiento anula una factura emitida a un cliente, elimina los registros de la salida de almacén que corresponden a esta factura y el estado de la factura la pone como anulada, además, actualiza las existencias de productos en el almacén.

```
Private Sub cmdEliminarFac_Click()  
    On Error GoTo Error_Eliminar  
    Dim i As Integer  
    Dim NumFactura, NumSalida As String, c_producto As String  
    Dim cmdBorrar As New ADODB.Command  
    If tipoUsuario = "USUARIO" Then  
        frmClaveEliminar.Show vbModal  
        'Verificar Clave  
        If Not claveAdmor Then  
            Exit Sub  
        End If
```



```
End If
'Borrar datos
NumFactura = Trim(txtcodigofac.Text)
If Not EstablecerConex Then
    End
End If

cnnControlInv.BeginTrans

rstSalidaAlmacen.Open "SELECT codigo_salida FROM Salida_Almacen WHERE
documento='" & NumFactura & " '";", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdText
If rstSalidaAlmacen.RecordCount > 0 Then _
    NumSalida = rstSalidaAlmacen!codigo_salida
rstSalidaAlmacen.Close: Set rstSalidaAlmacen = Nothing
For i = 1 To filasfac
    'Actualizar las existencias de los productos en almacén.
    rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod ='" &
(flexDetalleFactura.TextMatrix(i, 1)) & "' And codigo_bodega ='" & (txtcodigobod.Text)
& " '";", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
    With rstProducto
        !exis_final = CCur(!exis_final) + CCur(flexDetalleFactura.TextMatrix(i, 4))
        !saldo = CCur(FormatNumber(!exis_final * !costo_unitario, 2))
        If .EditMode = adEditInProgress Then
            .Update
        End If
        ' Eliminar los registros de los Productos almacenado en Salida
correspondiente a un codigo de salida.
        Call DeshacerSalidaAlmacen(NumSalida, !codigo_prod)
    End With
    rstProducto.Close: Set rstProducto = Nothing
Next i
cmdBorrar.ActiveConnection = cnnControlInv.ConnectionString
rstFactura.Open "SELECT estado FROM Factura WHERE nfactura='" & NumFactura & " '";",
cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
If rstFactura.RecordCount > 0 Then
    rstFactura!estado = "ANULADA"
    rstFactura.Update
Else
    rstFactura.Close: Set rstFactura = Nothing
    cnnControlInv.RollbackTrans
    Call LiberarConex
    MsgBox "No se puede anular la Factura " & NumFactura, , "Anular Factura."
    Exit Sub
End If
cmdBorrar.CommandText = "DELETE FROM Detalle_Salida WHERE codigo_salida ='" &
NumSalida & " '"
cmdBorrar.Execute
cmdBorrar.CommandText = "DELETE FROM Salida_Almacen WHERE codigo_salida='" &
NumSalida & " '"
cmdBorrar.Execute
rstFactura.Close: Set rstFactura = Nothing
cnnControlInv.CommitTrans
Call LiberarConex
MsgBox "La factura se ha anulado con éxito.", vbInformation, "Eliminando Factura."
Unload Me
frmBusFacturas.Show vbModal
```



```
Exit Sub
Error_Eliminar: 'En caso de no poder anularse una Factura lanza este error
    MsgBox "No se puede Anular esta Factura. ", vbInformation, "Error al anular la
Factura"
    cnnControlInv.RollbackTrans
    Call LiberarConex
End Sub
```

Este procedimiento cierra la la vista de la factura visible con todos sus datos.

```
Private Sub cmdCerrar_Click()
    Unload Me
    frmBusFacturas.Show vbModal
End Sub
```

Código de la Ilustración 33: Contraseña de administrador.

Este procedimiento captura los datos de la clave del administrador y verifica que sea correcta para llevar a cabo la operción que solo como administrador se puede hacer.

```
Private Sub cmdAceptar_Click()
    If Not EstablecerConex Then
        End
    End If
    rstUsuarios.Open "SELECT contraseña FROM Contraseña WHERE contraseña='" &
Trim(txtclave.Text) & "' and nombre_usuario='ADMINISTRADOR';", cnnControlInv,
adOpenStatic, adLockReadOnly, adCmdText
    With rstUsuarios
        If .RecordCount > 0 Then
            claveAdmor = True
            rstUsuarios.Close: Set rstUsuarios = Nothing
            Call LiberarConex
            Unload Me
        Else
            MsgBox "Clave Incorrecta", vbExclamation, "Clave Administrador"
            rstUsuarios.Close: Set rstUsuarios = Nothing
            Call LiberarConex
            txtclave.Text = ""
            txtclave.SetFocus
        End If
    End With
End Sub
```

Este procedimiento cancela la introducción de la clave del administrador.

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```

**Código de la Ilustración 34: Búsqueda de cotización.**

Este procedimiento hace una búsqueda de las cotizaciones emitidas y las muestra ordenadas por el nombre del cliente y la fecha en que se emitió.

```
Private Sub Form_Load()  
    Dim str As String  
    If Not EstablecerConex Then  
        End  
    End If  
    rstCotizacion.Open "SELECT nombre_cli, fecha FROM Cotizacion WHERE codigo_bodega  
=' ' & Nbodega & ' ' AND estado='VALIDA' ORDER BY nombre_cli;"; cnnControlInv,  
adOpenStatic, adLockReadOnly, adCmdText  
    With rstCotizacion  
        While Not .EOF  
            str = !nombre_cli & " " & !fecha  
            lstCotizacion.AddItem (str)  
            .MoveNext  
        Wend  
    End With  
    cmdMostrar.Enabled = False  
    rstCotizacion.Close: Set rstCotizacion = Nothing  
    Call LiberarConex  
End Sub
```

Código de la Ilustración 35: Muestra de la cotización.

Este procedimiento muestra la cotización seleccionada.

```
Private Sub cmdMostrar_Click()  
    nombrecli = Mid$(lstCotizacion, 1, Len(lstCotizacion) - 12)  
    fechacot = Mid$(lstCotizacion, Len(lstCotizacion) - 9, 10)  
    frmVerCotizacion.Show vbModal  
End Sub
```

Este procedimiento es utilizado para facturar una cotización guardada si la fecha de validez de la cotización no se ha vencido y verifica si existen los productos en almacén. Puede agregar o quitar más producto de la factura si el cliente lo desea.

```
Private Sub cmdFacturar_Click()  
    Dim i As Integer  
    Load frmFactura  
    With frmFactura  
        .cmdAgregarprod.Enabled = False  
        .cmdGuardar.Enabled = True  
        .cmdCancelar.Enabled = True  
        .txtcodigobod.Text = txtcodigobod.Text  
        .txtcodigocli.Text = txtcodigocli.Text  
        .txtnombrecli.Text = txtnombrecli.Text  
        .txtdireccioncli.Text = txtdireccioncli.Text  
        .txttelefonocli.Text = txttelefonocli.Text  
        .lblTotalfac = FormatNumber(CCur(lblTotalcot.Caption), 2)  
        .lbltotaligv = FormatNumber(CCur(lbltotaligv.Caption), 2)  
        .txtigv.Text = (CCur(.lbltotaligv) / CCur(.lblTotalfac)) * 100  
        .lblmontofac = FormatNumber(CCur(lblmontocot.Caption), 2)
```



```
For i = 1 To Fila
  With .flexDetalleFactura
    .AddItem flexDetalleCotizacion.TextMatrix(i, 1), i
    .TextMatrix(i, 1) = flexDetalleCotizacion.TextMatrix(i, 1)
    .TextMatrix(i, 2) = flexDetalleCotizacion.TextMatrix(i, 2)
    .TextMatrix(i, 3) = flexDetalleCotizacion.TextMatrix(i, 3)
    .TextMatrix(i, 4) = FormatNumber(CCur(flexDetalleCotizacion.TextMatrix(i,
4)), 2)
    .TextMatrix(i, 5) = FormatNumber(CCur(flexDetalleCotizacion.TextMatrix(i,
5)), 2)
    .TextMatrix(i, 6) = FormatNumber(CCur(flexDetalleCotizacion.TextMatrix(i,
6)), 2)
    .TextMatrix(i, 7) = FormatNumber(CCur(flexDetalleCotizacion.TextMatrix(i,
7)), 2)
    frmFactura.SalidaTot = CCur(frmFactura.SalidaTot) +
(CCur(flexDetalleCotizacion.TextMatrix(i, 7)) *
CCur(flexDetalleCotizacion.TextMatrix(i, 4)))
  End With
Next i
frmFactura.filasfac = Fila
End With
Unload frmVerCotizacion
Unload frmBusCotizacion
frmFactura.cmdEliminar.Enabled = True
frmFactura.Show vbModal
End Sub
```

Este procedimiento cierra la cotización mostrada.

```
Private Sub cmdCerrar_Click()
  Unload Me
End Sub
```

Código de la Ilustración 36: Lista de productos.

Este procedimiento muestra los datos del producto seleccionado a modificar.

```
Private Sub cmdAceptar_Click()
  If Not EstablecerConex Then
    End
  End If
  rstProducto.Open "SELECT * FROM Producto WHERE descripcion =" & lstProductos & " '
And codigo_bodega =" & Nbodega & "';", cnnControlInv, adOpenStatic, adLockReadOnly,
adCmdText
  With frmModProducto
    Load frmModProducto
    .msknumbodega.Text = rstProducto!codigo_bodega
    .mskcodigoprod.Text = rstProducto!codigo_prod
    .txtdescripcion.Text = rstProducto!descripcion
    .txtunidad.Text = rstProducto!unidad
    .txtexistencias.Text = rstProducto!exis_final
    .txtcosto.Text = rstProducto!costo_unitario
    .txtsaldo.Text = rstProducto!saldo
    .txtprecio1.Text = rstProducto!precio1
    .txtprecio2.Text = rstProducto!precio2
    .cbxmoneda.Text = rstProducto!moneda
```



```
.txtgrupo.Text = rstProducto!grupo
If Len(rstProducto!subgrupo) > 0 Then _
    .txtsubgrupo.Text = rstProducto!subgrupo
.txtubicacion.Text = rstProducto!ubicacion
If rstProducto!tiene_caducidad Then
    .txtdiasantesven.Enabled = True
    .txtdiasantesven.Text = rstProducto!tiempo_caducidad
End If
Desactivarcajas
End With
rstProducto.Close: Set rstProducto = Nothing
Call LiberarConex
Unload frmListaProductos
frmModProducto.Show vbModal
End Sub
```

Este procedimiento cierra la lista de productos a modificar.

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```

Código de la Ilustración 37: Modificar producto.

Este procedimiento modifica los datos del producto el cual se está modificando se realiza con transacciones para optimizar el proceso de modificación.

```
Private Sub cmdModificar_Click()
    On Error GoTo Error_Modificar

    If Verificardatos Then
        Exit Sub
    End If
    If Not EstablecerConex Then
        End
    End If
    cnnControlInv.BeginTrans
    rstProducto.Open "SELECT * FROM Producto WHERE codigo_prod = '" &
(mskcodigoprod.Text) & "' and codigo_bodega = '" & Nbodega & "';", cnnControlInv,
adOpenStatic, adLockOptimistic, adCmdText
    With rstProducto
        'Guardar el registro en la tabla Producto.
        !descripcion = UCase(Trim(txtdescripcion.Text))
        !precio1 = FormatNumber(CCur(txtprecio1.Text), 2)
        !precio2 = FormatNumber(CCur(txtprecio2.Text), 2)
        !grupo = UCase(txtgrupo.Text)
        !subgrupo = UCase(txtsubgrupo.Text)
        !ubicacion = UCase(txtubicacion.Text)
        If Len(txtdiasantesven.Text) > 0 Then
            !tiempo_caducidad = txtdiasantesven.Text
        End If
        .Update
    End With
    MsgBox "El Producto " & txtdescripcion.Text & " se ha modificado con éxito. ",
vbInformation, "Registro del Producto..."
    'Cerrar el recordset y la conexión.
```



```
rstProducto.Close: Set rstProducto = Nothing
cnnControlInv.CommitTrans
Call LiberarConex
Unload Me
frmListaProductos.Show vbModal
Exit Sub
Error_Modificar: 'En caso de generarse un error al guardar el registro
                 'saltar a la siguiente linea.
MsgBox "Se presento un error al modificar los Datos del Producto. " &
Err.Description, vbInformation, "Error al modificar los Datos..."
cnnControlInv.RollbackTrans
Call LiberarConex
End Sub
```

Este procedimiento cancela la modificación del producto y guarda la modificación del registro si el cliente lo desea o seleccionó cancelar por accidente.

```
Private Sub cmdCancelar_Click()
    If MsgBox("¿Desea guardar los cambios?", vbYesNo, "Modificar registro") = vbYes
Then
        cmdModificar_Click
    Else
        Unload Me
        frmListaProductos.Show vbModal
    End If
End Sub
```

Código de la Ilustración 38: Modificar cliente.

Este procedimiento muestra la vista de todos los clientes, donde se busca el cliente a modificar.

```
Private Sub cmdBuscar_Click()
    frmListaCliente.Show vbModal
End Sub
```

Este procedimiento modifica los datos del cliente seleccionado y actualiza la base de datos.

```
Private Sub cmdModificar_Click()
    On Error GoTo Error_Modificar
    If Len(txtnombrecli.Text) = 0 Then
        MsgBox "Debes de ingresar el Nombre del Cliente.", vbInformation, "Error en los
Datos del Cliente..."
        txtnombrecli.SetFocus
        Exit Sub
    End If
    If Not EstablecerConex Then
        End
    End If
    cnnControlInv.BeginTrans
    rstCliente.Open "SELECT * FROM Cliente WHERE codigo_cli='" & (txtcodigocli.Text) &
"';", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
    With rstCliente
        'Guardar el registro en la tabla Cliente.
        !nombre_cli = UCase(txtnombrecli.Text)
    End With
End Sub
```



```
    If Len(Trim(txtDireccioncli.Text)) > 0 Then _
        !direccion = UCase(txtDireccioncli.Text)
    If mskcedulacli.Text <> "___-____-_____" Then
        !cedula = Trim(mskcedulacli.Text)
    End If
    If msktelefonocli.Text <> "___-____" Then
        !telefono = Trim(msktelefonocli.Text)
    End If
End With
If rstCliente.EditMode = adEditInProgress Then
    If MsgBox("¿Desea guardar los cambios?", vbYesNo, "Modificar registro") = vbYes
Then
        'Actualizar.
        rstCliente.Update
        cnnControlInv.CommitTrans
        MsgBox "El Cliente " & rstCliente!nombre_cli & " se ha modificado con
éxito. ", vbInformation, "Registro del Cliente..."
        cmdModificar.Enabled = False
        cmdEliminar.Enabled = False
        Desactivarcajas
    Else
        rstCliente.CancelUpdate
        cnnControlInv.RollbackTrans
        'cmdModificar.Enabled = False
        frmListaCliente.Activarcajas
    End If
End If
'Cerrar el recordset y la conexión.
rstCliente.Close: Set rstCliente = Nothing
Call LiberarConex
Exit Sub
Error_Modificar:
    MsgBox "No se pueden modificar los Datos del Cliente, verifique que estos esten
correctos.", vbInformation, "Error al modificar los Datos..."
    cnnControlInv.RollbackTrans
    Call LiberarConex
End Sub
```

Este procedimiento permite eliminar a un cliente desde el área de modificación.

```
Private Sub cmdEliminar_Click()
On Error GoTo Error_Eliminar
    If Not EstablecerConex Then
        End
    End If
    cnnControlInv.BeginTrans
    If MsgBox("¿Desea Eliminar este Registro?", vbYesNo, "Eliminar Cliente") = vbYes
Then
        rstCliente.Open "DELETE FROM Cliente WHERE codigo_cli='" & (txtcodigocli.Text)
& "'", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
        MsgBox "Se ha eliminado el Cliente:" & txtnombrecli.Text, vbInformation,
"Eliminado Cliente"
        cnnControlInv.CommitTrans
        LimpiarCajas
        Desactivarcajas
        cmdModificar.Enabled = False
    End If
End Sub
```



```
        cmdEliminar.Enabled = False
    Else
        cnnControlInv.RollbackTrans
    End If
    Call LiberarConex
    Exit Sub
Error_Eliminar:
    MsgBox "No se puede eliminar un cliente que se le haya emitido una Factura o una
    Cotización. ", vbInformation, "Error al eliminar un Cliente ..."
    cmdEliminar.Enabled = False
    cnnControlInv.RollbackTrans
    Call LiberarConex
End Sub
```

Este procedimiento cancela la modificación o la eliminación de los clientes.

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```

Código de la Ilustración 39: Lista de clientes.

Este procedimiento envía los datos del cliente seleccionado de una lista a la vista de modificación del cliente.

```
Private Sub cmdAceptar_Click()
    If Not EstablecerConex Then
        End
    End If
    rstCliente.Open "SELECT * FROM Cliente WHERE nombre_cli='" & lstClientes & "'",
    cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    'Cotizacion
    If rstCliente.RecordCount > 0 Then
        If bolCotizacionvisible Then
            frmCotizacion.txtcodigocli = rstCliente!codigo_cli
            frmCotizacion.txtnombrecli = rstCliente!nombre_cli
            If Len(rstCliente!direccion) > 0 Then
                frmCotizacion.txtdireccioncli = rstCliente!direccion
            Else
                frmCotizacion.txtdireccioncli = ""
            End If
            If Len(rstCliente!telefono) > 0 Then
                frmCotizacion.txttelefonocli = rstCliente!telefono
            Else
                frmCotizacion.txttelefonocli = ""
            End If
        ElseIf bolFacturavisible Then
            frmFactura.txtcodigocli = rstCliente!codigo_cli
            frmFactura.txtnombrecli = rstCliente!nombre_cli
            If Len(rstCliente!direccion) > 0 Then
                frmFactura.txtdireccioncli = rstCliente!direccion
            Else
                frmFactura.txtdireccioncli = ""
            End If
            If Len(rstCliente!telefono) <> 0 Then
                frmFactura.txttelefonocli = rstCliente!telefono
            End If
        End If
    End If
End Sub
```



```
Else
    frmFactura.txttelefonocli = ""
End If
Else
    With frmModCliente
        .txtcodigocli = rstCliente!codigo_cli
        .txtnombrecli = rstCliente!nombre_cli
        If Len(rstCliente!direccion) > 0 Then _
            .txtdireccioncli = rstCliente!direccion
        If Len(rstCliente!cedula) > 0 Then
            .mskcedulacli = rstCliente!cedula
            .mskcedulacli.Enabled = False
        Else
            .mskcedulacli.Text = "__-____-____"
            .mskcedulacli.Enabled = True
        End If
        If Len(rstCliente!telefono) > 0 Then
            .msktelefonocli = rstCliente!telefono
        Else
            .msktelefonocli.Text = "__-____"
        End If
        .cmdModificar.Enabled = True
        .cmdEliminar.Enabled = True
        Activarcajas
    End With
End If
End If
rstCliente.Close: Set rstCliente = Nothing
Call LiberarConex
Unload Me
End Sub
```

Código de la Ilustración 40: Modificar proveedor.

Este procedimiento muestra la vista de todos los proveedores, donde se busca al proveedor a modificar.

```
Private Sub cmdBuscar_Click()
    frmListaProveedores.Show vbModal
End Sub
```

Este procedimiento modifica los datos de un proveedor específico.

```
Private Sub cmdModificar_Click()
    On Error GoTo Error_Aceptar
    If Len(txtdireccionprov.Text) = 0 Then
        MsgBox "Debes ingresar la dirección del Proveedor.", vbInformation, "Error en los Datos del Proveedor..."
        txtdireccionprov.SetFocus
        Exit Sub
    ElseIf Len(txtnombreprov.Text) = 0 Then
        MsgBox "Debes ingresar el nombre del Proveedor.", vbInformation, "Error en los Datos del Proveedor..."
        txtnombreprov.SetFocus
        Exit Sub
    End If
End Sub
```



```
ElseIf msktelefonoprov.Text = "###-####" Then
    MsgBox "Debes ingresar el # de teléfono del Proveedor.", vbInformation, "Error
en los Datos del Proveedor..."
    msktelefonoprov.SetFocus
    Exit Sub
End If
If Not EstablecerConex Then
    End
End If
cnnControlInv.BeginTrans
rstProveedor.Open "SELECT * FROM Proveedor WHERE codigo_prov ='" &
(txtcodigoprov.Text) & "';", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
With rstProveedor
    'Guardar el registro en la tabla Proveedor.
    !nombre_prov = UCase(Trim(txtnombreprov.Text))
    !direccion = UCase(Trim(txtdireccionprov.Text))
    !telefono = msktelefonoprov.Text
End With
If rstProveedor.EditMode = adEditInProgress Then
    If MsgBox("¿Desea guardar los cambios?", vbYesNo, "Modificar registro") = vbYes
Then
        'intentar actualizar.
        rstProveedor.Update
        cnnControlInv.CommitTrans
        MsgBox "El Proveedor " & rstProveedor!nombre_prov & " se ha modificado con
éxito. ", vbInformation, "Registro del Cliente..."
        cmdModificar.Enabled = False
        Desactivarcajas
    Else
        rstProveedor.CancelUpdate
        cnnControlInv.RollbackTrans
        cmdModificar.Enabled = True
        frmListaProveedores.Activarcajas
    End If
End If
'Cerrar el recordset y la conexión.
rstProveedor.Close: Set rstProveedor = Nothing
Call LiberarConex
Exit Sub
Error_Aceptar:
MsgBox "No se pueden modificar los Datos del Proveedor, verifique que estos esten
correctos.", vbInformation, "Error al modificar los Datos..."
cnnControlInv.RollbackTrans
Call LiberarConex
End Sub
```

Este procedimiento cancela la modificación o la eliminación de los proveedores.

```
Private Sub cmdCancelar_Click()
    Unload Me
End Sub
```

**Código de la Ilustración 41: Lista de proveedores.**

Este procedimiento envía los datos del proveedor seleccionado de una lista a la vista de modificación del proveedor.

```
Private Sub cmdAceptar_Click()
    If Not EstablecerConex Then
        End
    End If
    rstProveedor.Open "SELECT * FROM Proveedor WHERE nombre_prov = '" & lstProveedores &
    "'";", cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    If rstProveedor.RecordCount > 0 Then
        If bolEntradavisible Then
            With frmEntradasAlmacen
                .txtcodigoprov = rstProveedor!codigo_prov
                .txtnombreprov = rstProveedor!nombre_prov
            End With
        Else
            With frmModProveedor
                .txtcodigoprov = rstProveedor!codigo_prov
                .txtnombreprov = rstProveedor!nombre_prov
                .txtdireccionprov = rstProveedor!direccion
                .msktelefonoprov = rstProveedor!telefono
                .cmdModificar.Enabled = True
                Activarcajas
            End With
        End If
    End If
    rstProveedor.Close: Set rstProveedor = Nothing
    Call LiberarConex
    Unload Me
End Sub
```

Código de la Ilustración 42: Modificar fecha de proceso.

Este procedimiento carga en la vista fecha proceso, la fecha inicial y final del mes de proceso del inventario, de la cual solo se permite modificar la fecha final.

```
Private Sub Form_Load()
    If Not EstablecerConex Then
        End
    End If
    'Abrir la tabla FechaProceso .
    rstFechaProceso.Open "SELECT * FROM FechaProceso", cnnControlInv, adOpenDynamic,
    adLockReadOnly, adCmdText
    If Not rstFechaProceso.EOF Then
        mskFechainicio.Text = rstFechaProceso!fecha_inicio
        mskFechafin.Text = rstFechaProceso!fecha_fin
    End If
    rstFechaProceso.Close: Set rstFechaProceso = Nothing
    Call LiberarConex
End Sub
```



Este procedimiento se usa para modificar la fecha final del mes de proceso y verifica que sea mayor que la fecha inicial.

```
Private Sub cmdAceptar_Click()  
    If Not IsDate(mskFechafin.Text) Then  
        MsgBox "Fecha final incorrecta.", vbInformation, "Error al ingresar la fecha."  
        mskFechafin.SetFocus  
        Exit Sub  
    Else  
        If CDate(mskFechafin.Text) >= Date And CDate(mskFechafin.Text) >=  
CDate(mskFechainicio.Text) Then  
            If Not EstablecerConex Then  
                End  
            End If  
            cnnControlInv.BeginTrans  
            'Abrir la tabla FechaProceso para modificar la información.  
            rstFechaProceso.Open "SELECT * FROM FechaProceso WHERE fecha_inicio='" &  
mskFechainicio.Text & "'", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText  
            rstFechaProceso!fecha_fin = mskFechafin.Text  
            If rstFechaProceso.EditMode = adEditInProgress Then  
                rstFechaProceso.Update  
                cnnControlInv.CommitTrans  
            End If  
            rstFechaProceso.Close: Set rstFechaProceso = Nothing  
            Call LiberarConex  
            MsgBox "La Fecha de Proceso se ha modificado con éxito.", vbInformation,  
"Fecha de Proceso."  
            Unload Me  
        Else  
            MsgBox "La Fecha final debe ser mayor o igual que la actual.",  
vbInformation, "Error al ingresar la fecha."  
        End If  
    End If  
End Sub
```

Código de la Ilustración 43: Modificar datos de la empresa.

Este procedimiento envia a la vista datos de la empresa, todos aquellos datos que pueden ser modificados en el momento de carga.

```
Private Sub Form_Load()  
    If Not EstablecerConex Then  
        End  
    End If  
    'Abrir la tabla Empresa para modificar la información.  
    rstEmpresa.Open "Empresa", cnnControlInv, adOpenDynamic, adLockOptimistic  
    With rstEmpresa  
        txtNombre.Text = !nombreE  
        txtDireccion.Text = !direccionE  
        txtCiudad.Text = !ciudad  
        txtPais.Text = !pais  
        mskTelefono.Text = !telefono_o_celular  
        txtCorreo.Text = !correo_elect  
    End With  
    'inhabilitar las cajas  
    txtNombre.Enabled = False
```



```
txtDireccion.Enabled = False
txtCiudad.Enabled = False
txtPais.Enabled = False
mskTelefono.Enabled = False
txtCorreo.Enabled = False
'inhabilitar los botones
cmdAceptar.Enabled = False
rstEmpresa.Close: Set rstEmpresa = Nothing
Call LiberarConex
```

```
End Sub
```

Este procedimiento activa los campos o cajas de texto para que el usuario pueda modificar los datos de la empresa.

```
Private Sub cmdModificar_Click()
txtNombre.Enabled = True
txtDireccion.Enabled = True
txtCiudad.Enabled = True
txtPais.Enabled = True
mskTelefono.Enabled = True
txtCorreo.Enabled = True
cmdModificar.Enabled = False
cmdAceptar.Enabled = True
cmdCancelar.Enabled = True
```

```
End Sub
```

Este procedimiento verifica que todos los campos de los datos de la empresa se hayan completado, luego se guardan en la base de datos.

```
Private Sub cmdAceptar_Click()
On Error GoTo Error_Aceptar
'Validar las cajas de texto.
If Len(txtNombre.Text) = 0 Then
MsgBox "Falta el nombre de la Empresa.", , "Datos de la empresa."
txtNombre.SetFocus
Exit Sub
ElseIf Len(txtDireccion.Text) = 0 Then
MsgBox "Falta la dirección de la Empresa.", , "Datos de la empresa."
txtDireccion.SetFocus
Exit Sub
ElseIf Len(txtCiudad.Text) = 0 Then
MsgBox "Falta la ciudad de la Empresa.", , "Datos de la empresa."
txtCiudad.SetFocus
Exit Sub
ElseIf Len(txtPais.Text) = 0 Then
MsgBox "Falta el país de la Empresa.", , "Datos de la empresa."
txtPais.SetFocus
Exit Sub
ElseIf mskTelefono.Text = "###-####" Then
MsgBox "Falta el número de teléfono de la Empresa.", , "Datos de la empresa."
mskTelefono.SetFocus
Exit Sub
End If
If Not EstablecerConex Then
End
End If
cnnControlInv.BeginTrans
```



```
rstEmpresa.Open "Empresa", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdTable
With rstEmpresa
    'Guardar el registro en la tabla Empresa.
    !nombreE = txtNombre.Text
    !direccionE = txtDireccion.Text
    !ciudad = txtCiudad.Text
    !pais = txtPais.Text
    !telefono_o_celular = mskTelefono.Text
    !correo_elect = txtCorreo.Text
End With
If rstEmpresa.EditMode = adEditInProgress Then
    If MsgBox("¿Desea guardar los cambios?", vbYesNo, "Modificar Datos de Empresa")
= vbYes Then
        'intentar actualizar.
        rstEmpresa.Update
        cnnControlInv.CommitTrans
        MsgBox "Los datos de la Empresa se han Modificado con éxito.",
vbInformation, "Datos de la Empresa..."
        frmPrincipal.lblnomE.Caption = Trim(txtNombre.Text)
    Else
        rstEmpresa.CancelUpdate
        cnnControlInv.RollbackTrans
        cmdModificar.Enabled = True
    End If
End If
'Cerrar el recordset y la conexión.
rstEmpresa.Close: Set rstEmpresa = Nothing
Call LiberarConex
Unload Me
Exit Sub
Error_Aceptar: 'En caso de generarse un error al guardar el registro
                'saltar a la siguiente línea.
MsgBox "Error al modificar los Datos de la Empresa, verifique que estos esten
correctos.", , "Datos de la empresa."
cnnControlInv.RollbackTrans
Call LiberarConex
End Sub
```

Código de la Ilustración 44: Modificar contraseña.

Este procedimiento envía a la vista modificar contraseña, las cuentas de usuario a los cuales se va a modificar su clave de acceso.

```
Private Sub Form_Load()
    If Not EstablecerConex Then
        End
    End If
    'Abrir la tabla Contraseña para ver la información de los Usuarios.
    Set rstUsuarios = New ADODB.Recordset
    rstUsuarios.Open "SELECT nombre_usuario FROM Contraseña ;", cnnControlInv,
adOpenStatic, adLockReadOnly, adCmdText
    Do Until rstUsuarios.EOF
        'Guardar el registro en la tabla
        cbxUsuario.AddItem rstUsuarios!nombre_usuario
        rstUsuarios.MoveNext
    End Do
End Sub
```



```
Loop
rstUsuarios.Close: Set rstUsuarios = Nothing
Call LiberarConex
End Sub
```

Este procedimiento permite cambiar las contraseñas de las cuentas de los usuarios creadas ingresando la contraseña anterior e ingresando la nueva contraseña y confirmación de las mismas.

```
Private Sub cmdAceptar_Click()
Dim varContraseña As String
On Error GoTo Error_Guardar
varContraseña = Trim(txtContraseña.Text)
'Validar las cajas de texto.
If Len(Trim(txtContraseña.Text)) = 0 Then
MsgBox " Falta la Contraseña actual. ", vbInformation, "Error al ingresar la
Contraseña..."
Limpiarcajascontraseñas
txtContraseña.SetFocus
Exit Sub
End If
If Len(Trim(txtNContraseña.Text)) = 0 Then
MsgBox " Falta la nueva Contraseña. ", vbInformation, "Error nueva
Contraseña..."
Limpiarcajascontraseñas
txtContraseña.SetFocus
Exit Sub
ElseIf Len(Trim(txtNConfirmar.Text)) = 0 Then
MsgBox " Falta confirmar la nueva Contraseña. ", vbInformation, "Error nueva
Contraseña..."
Limpiarcajascontraseñas
txtContraseña.SetFocus
Exit Sub
ElseIf Trim(txtNContraseña.Text) <> Trim(txtNConfirmar.Text) Then
MsgBox " Las nuevas Contraseña no coinciden. ", vbInformation, "Error nueva
Contraseña..."
Limpiarcajascontraseñas
txtContraseña.SetFocus
Exit Sub
End If
'Establecer la conexión a la Base de Datos.
If Not EstablecerConex Then
End
End If
cnnControlInv.BeginTrans
'Abrir la tabla Contraseña para grabar la información.
Set rstUsuarios = New ADODB.Recordset
rstUsuarios.Open "SELECT * FROM Contraseña WHERE nombre_usuario =" &
(cbxUsuario.Text) & "';", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText

If varContraseña = rstUsuarios!contraseña Then
With rstUsuarios
'Guardar el registro en la tabla
!contraseña = txtNContraseña.Text
End With
```



```
        If rstUsuarios.EditMode = adEditInProgress Then
            If MsgBox("¿Desea guardar los cambios?", vbYesNo, "Modificar Datos de
Usuarios") = vbYes Then
                'intentar actualizar.
                rstUsuarios.Update
                cnnControlInv.CommitTrans
                MsgBox "Los datos de los Usuarios se han Modificado con éxito.",
vbInformation, "Datos de Usuarios..."
                cmdAceptar.Enabled = False
            Else
                rstUsuarios.CancelUpdate
                cnnControlInv.RollbackTrans
            End If
        End If
    Else
        MsgBox "Contraseña actual incorrecta", vbInformation
        cnnControlInv.RollbackTrans
        cmdAceptar.Enabled = True
    End If
    rstUsuarios.Close: Set rstUsuarios = Nothing
    Call LiberarConex
    txtContraseña.SetFocus
    Limpiarcajascontraseñas
    Exit Sub
Error_Guardar:
    MsgBox "Error al Modificar los Datos de los Usuarios. ", , "Modificar las
contraseñas..."
    cnnControlInv.RollbackTrans
    Limpiarcajascontraseñas
End Sub
```

Código de la Ilustración 47: Productos cercanos a caducar.

Esta función es invocada desde la vista bodega de trabajo del sistema, pero se ha ubicado aquí para mayor aclaración para el lector y tiene como objetivo vigilar la caducidad de los productos y avisar al usuario.

```
Private Function ProdCaducidadCercana() As Integer
    Dim fecha As Date
    Dim producto As String
    Dim i, k As Integer
    Dim j As Integer
    j = 0
    Load frmCadcercana
    'inicializar el flex
    With frmCadcercana.flexcaducidad
        .FormatString = " |<Código|<Descripción|<Caducidad|<Cantidad|<N°Lote"
        .ColWidth(0) = 0
        .ColWidth(1) = 1000
        .ColWidth(2) = 3000
        .ColWidth(3) = 1200
        .ColWidth(4) = 1000
        .ColWidth(5) = 1000
    End With
```



```
'Generar los productos que estan cerca a caducar.
If Not EstablecerConex Then
    End
End If
rstProducto.Open "SELECT * FROM Producto WHERE codigo_bodega='" & Nbodega & "' AND
exis_final > '0'and tiene_caducidad='1' ";", cnnControlInv, adOpenStatic,
adLockOptimistic, adCmdText
For i = 1 To rstProducto.RecordCount
    fecha = rstProducto!tiempo_caducidad + Date
    producto = rstProducto!codigo_prod
    'todos los productos que estan prox. a vencer.
    rstLoteTemporal.Open "SELECT * FROM Lote_Poducto WHERE codigo_prod='" &
producto & "' AND codigo_bodega='" & Nbodega & "' AND fecha_caducidad <= '" & fecha
& "' AND fecha_caducidad>'" & Date & "' AND codigo_entrada LIKE 'E%' ORDER BY
codigo_prod ";", cnnControlInv, adOpenStatic, adLockOptimistic, adCmdText
    With rstLoteTemporal
        For k = 1 To .RecordCount
            j = j + 1
            frmCadcercana.flexcaducidad.AddItem !codigo_prod, k
            frmCadcercana.flexcaducidad.TextMatrix(k, 1) = !codigo_prod
            frmCadcercana.flexcaducidad.TextMatrix(k, 2) = rstProducto!descripcion
            frmCadcercana.flexcaducidad.TextMatrix(k, 3) = !fecha_caducidad
            frmCadcercana.flexcaducidad.TextMatrix(k, 4) = !cantidad_fecha
            frmCadcercana.flexcaducidad.TextMatrix(k, 5) = !numlote

            .MoveNext
        Next k
    End With
    rstLoteTemporal.Close: Set rstLoteTemporal = Nothing
    rstProducto.MoveNext
Next i
rstProducto.Close: Set rstProducto = Nothing
Call LiberarConex
ProdCaducidadCercana = j
If j = 0 Then Unload frmCadcercana
End Function
```

Este procedimiento cierra la vista despues de haber mostrado los productos proximos a vencer.

```
Private Sub cmdCerrar_Click()
    Unload Me
End Sub
```

Código de la Ilustración 48: Productos vencidos.

Esta función es invocada desde la vista bodega de trabajo del sistema, pero se ha ubicado aquí para mayor aclaración para el lector y tiene como objetivo buscar en la bodega de trabajo productos vencidos y notificar al usuario.

```
Private Function ProductosCaducados() As Integer
    Dim producto As String
    Dim i, k As Integer
    Dim j As Integer
    j = 0
```



```
'inicializar el flex
Load frmProdVencidos
With frmProdVencidos.flexcaducidad
    .FormatString = "|<Código|<Descripción|<FechaCaducidad|<Cantidad|<Nº Lote"
    .ColWidth(0) = 0
    .ColWidth(1) = 1000
    .ColWidth(2) = 3000
    .ColWidth(3) = 1300
    .ColWidth(4) = 1000
    .ColWidth(5) = 1000
End With
'Generar los productos que están vencidos.
If Not EstablecerConex Then
    End
End If
rstProducto.Open "SELECT * FROM Producto WHERE codigo_bodega='" & Nbodega & "' AND
exis_final > '0'and tiene_caducidad='1' ";", cnnControlInv, adOpenStatic,
adLockOptimistic, adCmdText
For i = 1 To rstProducto.RecordCount
    producto = rstProducto!codigo_prod
    'todos los productos que estan vencidos.
    rstLoteTemporal.Open "SELECT * FROM Lote_Poducto WHERE codigo_prod='" &
producto & "' AND codigo_bodega='" & Nbodega & "' AND fecha_caducidad <= '" & Date &
"' AND codigo_entrada LIKE 'E%' ORDER BY codigo_prod ";", cnnControlInv,
adOpenStatic, adLockOptimistic, adCmdText
    With rstLoteTemporal
        For k = .RecordCount To 1 Step -1
            j = j + 1
            frmProdVencidos.flexcaducidad.AddItem !codigo_prod, j
            frmProdVencidos.flexcaducidad.TextMatrix(j, 1) = !codigo_prod
            frmProdVencidos.flexcaducidad.TextMatrix(j, 2) =
rstProducto!descripcion
            frmProdVencidos.flexcaducidad.TextMatrix(j, 3) = !fecha_caducidad
            frmProdVencidos.flexcaducidad.TextMatrix(j, 4) = !cantidad_fecha
            frmProdVencidos.flexcaducidad.TextMatrix(j, 5) = !numlote

            .MoveNext
        Next k
    End With
    rstLoteTemporal.Close: Set rstLoteTemporal = Nothing
    rstProducto.MoveNext
Next i
rstProducto.Close: Set rstProducto = Nothing
Call LiberarConex
ProductosCaducados = j
    If j = 0 Then Unload frmProdVencidos
End Function
```

Este procedimiento muestra al usuario una lista de todos los productos vencidos en una vista y los saca del almacén y actualiza la base de datos.

```
Private Sub cmdSacarProducto_Click()
    Dim i As Integer
    Dim codigoproducto As String
    Dim subtotal, totalsal
    Load frmSalidasAlmacen
```



```
frmSalidasAlmacen.txtdocsal.Text = "Prod_Vencidos"
If Not EstablecerConex Then
    End
End If
cnnControlInv.BeginTrans
rstSalidaAlmacen.Open "SELECT codigo_salida FROM Salida_Almacen WHERE
codigo_bodega = '" & Nbodega & "' AND documento='Prod_Vencidos' AND fecha_salida ='" &
Date & "';", cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
If rstSalidaAlmacen.RecordCount = 0 Then
    rstSalidaAlmacen.Close: Set rstSalidaAlmacen = Nothing
    With flexcaducidad
        For i = 1 To flexcaducidad.Rows - 2
            codigoproducto = .TextMatrix(i, 1)
            rstProducto.Open "SELECT unidad,costo_unitario FROM Producto WHERE
codigo_prod='" & (codigoproducto) & "' and codigo_bodega = '" & (Nbodega) & "' ORDER BY
descripcion;", cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
            frmSalidasAlmacen.flexDetalleSalida.AddItem .TextMatrix(i, 1), i
            frmSalidasAlmacen.flexDetalleSalida.TextMatrix(i, 1) = .TextMatrix(i,
1)
            frmSalidasAlmacen.flexDetalleSalida.TextMatrix(i, 2) = .TextMatrix(i,
2)
            frmSalidasAlmacen.flexDetalleSalida.TextMatrix(i, 3) =
rstProducto!unidad
            frmSalidasAlmacen.flexDetalleSalida.TextMatrix(i, 4) = .TextMatrix(i,
4)
            subtotal = CCur(rstProducto!costo_unitario * .TextMatrix(i, 4))
            totalsal = CCur(totalsal + subtotal)
            frmSalidasAlmacen.flexDetalleSalida.TextMatrix(i, 5) = subtotal
            rstProducto.Close: Set rstProducto = Nothing
        Next i
    End With
    cnnControlInv.CommitTrans
    Call LiberarConex
    frmSalidasAlmacen.lblTotalsal.Caption = FormatNumber(CCur(totalsal), 2)
    frmSalidasAlmacen.filadetsal = flexcaducidad.Rows - 2
    frmSalidasAlmacen.cmdGuardar_Click
    MsgBox "Se imprimirá el reporte de todos esto productos Vencidos que se han
retirado del Almacén.", vbInformation, "Retirando de Almacén"
    frmSalidasAlmacen.cmdImprimir_Click
Else
    MsgBox "Estos productos ya han sido retirados del Almacén.", vbInformation
    rstSalidaAlmacen.Close: Set rstSalidaAlmacen = Nothing
    cnnControlInv.RollbackTrans
    Call LiberarConex
End If
Unload frmSalidasAlmacen
Unload Me
End Sub
```



Código de la Clase global.

Declaracion de variables y funciones a utilizar en toda la aplicación.

```
Option Explicit

Public varlistafechacad, varlistacantlote()
Public varcantidadcad() As Single
Public maxRegistros As Integer
Public varnumLotes() As Long
Public p1() As Single
Public p2() As Single
Public NomEmpresa_DesPro, Nbodega, CodigoProd, CodigoEnt, CodigoSal, CodigoAju,
CodigoFac, CodigoCot, Tipobod As String
Public bolCotizacionvisible, claveAdmor, bolEntradavisible, bolFacturavisible As
Boolean
Public tipoUsuario As String
Public finicial, ffinal As Date
Public cnnControlInv As New ADODB.Connection
Public rstFechaProceso As New ADODB.Recordset
Public rstAjusteAlmacen As New ADODB.Recordset
Public rstDetalleAjuste As New ADODB.Recordset
Public rstProducto As New ADODB.Recordset
Public rstLote As New ADODB.Recordset
Public rstEntradaAlmacen As New ADODB.Recordset
Public rstDetalleEntrada As New ADODB.Recordset
Public rstSalidaAlmacen As New ADODB.Recordset
Public rstDetalleSalida As New ADODB.Recordset
Public rstCotizacion As New ADODB.Recordset
Public rstDetalleCotizacion As New ADODB.Recordset
Public rstFactura As New ADODB.Recordset
Public rstDetalleFactura As New ADODB.Recordset
Public rstUsuarios As New ADODB.Recordset
Public rstProveedor As New ADODB.Recordset
Public rstCliente As New ADODB.Recordset
Public rstEmpresa As New ADODB.Recordset
Public rstBodega As New ADODB.Recordset
Public rstLoteTemporal As New ADODB.Recordset
```

Esta Función conecta la aplicacion con la base de datos, es invocada cada vez que se desea hacer una consulta a las tablas de la base de datos.

```
Public Function EstablecerConex() As Boolean
    On Error GoTo Error_Conexion
    Set cnnControlInv = New ADODB.Connection

    cnnControlInv.ConnectionString = "Provider=SQLOLEDB.1;Password=caics;Persist
Security Info=True;User ID=caics;" _
        & "Initial Catalog=Inventario;Data Source=SERVER"
    cnnControlInv.CursorLocation = adUseClient
    cnnControlInv.Open
    EstablecerConex = True
    Exit Function
Error_Conexion:
MsgBox "No se ha podido establecer la conexión, el programa se cerrara y no podra ser
iniciado hasta que sea puesto en marcha el SERVIDOR. Pongase en contacto con el
administrador del sistema. ", vbCritical, "Error al conectarse al Servidor..."
```



```
Err.Clear
EstablecerConex = False
End Function
```

Esta Funcion libera la conexión de la aplicación con la base de datos, es invocada cada vez que no se utilizaran dichos recursos.

```
Public Function LiberarConex() As Object
    If Not (cnnControlInv Is Nothing) Then
        cnnControlInv.Close: Set cnnControlInv = Nothing
    End If
    Err.Clear
End Function
```

Esta función se usa para verificar si existe una fecha de proceso, normalmente es invocada al cargar la aplicación por primera vez o al finalizar la fecha de proceso del inventario.

```
Public Function VerificarFechaProceso() As Boolean
    If Not EstablecerConex Then
        End
    End If
    'Abrir la tabla FechaProceso para verificar la fecha.
    rstFechaProceso.Open "SELECT * FROM FechaProceso;", cnnControlInv, adOpenDynamic,
adLockReadOnly, adCmdText
    If rstFechaProceso.RecordCount > 0 Then
        finicial = rstFechaProceso!fecha_inicio
        ffinal = rstFechaProceso!fecha_fin
        VerificarFechaProceso = True
    Else
        VerificarFechaProceso = False
    End If
    rstFechaProceso.Close: Set rstFechaProceso = Nothing
    Call LiberarConex
End Function
```

Esta funcion genera números aleatorios utilis para codificar; es invocada al cargar la vista nuevo cliente al igual que nuevo proveedor y codificar producto.

```
Public Function Aleatorio() As Integer
    Aleatorio = Int((9999 - 1000 +1) * Rnd + 1000)
End Function
```

Esta funcion genera números aleatorios utilizados para codificar; es invocada por las funciones codificar cotizacion, codificar salida y codifiar factura.

```
Public Function Aleatorigrande() As Long
    Aleatorigrande = CLng((99999 - 10000 + 1) * Rnd + 10000)
End Function
```



Esta función genera un código para cada producto ingresado y los compara con códigos de todos los registros de la base de datos para no repetirlos.

```
Public Function CodificarProducto(bodega As String)
    Dim codigopro As String
    Dim calcularRND, numero As Integer
    Randomize
    CodigoProd = ""
    rstProducto.Open "SELECT codigo_prod FROM Producto ;", cnnControlInv,
adOpenStatic, adLockReadOnly, adCmdText
    With rstProducto
        calcularRND = Aleatorio
        If .RecordCount <> 0 Then
            .MoveFirst
            Do Until .EOF
                codigopro = !codigo_prod : numero = CInt(Mid(codigopro, 5, 4))
                If numero = calcularRND Then
                    .MoveFirst
                    calcularRND = Aleatorio
                Else
                    .MoveNext
                End If
            Loop
            End If
            CodigoProd = bodega & "-" & CStr(calcularRND)
        End With
        rstProducto.Close: Set rstProducto = Nothing
    End Function
```

Esta función codifica las entradas de los productos invocada por cada entrada.

```
Public Function CodificarEntradaAlmacen()
    Dim codigoentrada As String
    Dim numero As Integer
    rstEntradaAlmacen.Open "SELECT codigo_entrada FROM Entrada_Almacen;",
cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    If rstEntradaAlmacen.RecordCount <> 0 Then
        rstEntradaAlmacen.MoveLast
        codigoentrada = rstEntradaAlmacen!codigo_entrada
        numero = CInt(Mid(codigoentrada, 3, 7))
        CodigoEnt = "E-" & CStr(numero + 1)
    Else
        CodigoEnt = "E-10000"
    End If
    rstEntradaAlmacen.Close: Set rstEntradaAlmacen = Nothing
End Function
```

Esta función genera un código por cada salida de productos del almacén, para un mejor control de las salidas.

```
Public Function CodificarSalidaAlmacen()
    Dim codigoSalida As String
    Dim calcularRND, numero As Long
    Randomize

    rstSalidaAlmacen.Open "SELECT codigo_salida FROM Salida_Almacen ;", cnnControlInv,
```



```
adOpenStatic, adLockReadOnly, adCmdText
  With rstSalidaAlmacen
    calcularRND = Aleatorigrande
    If .RecordCount <> 0 Then
      .MoveFirst
      Do Until .EOF
        codigoSalida = !codigo_salida
        numero = CLng(Mid(codigoSalida, 3, 7))
        If numero = calcularRND Then
          .MoveFirst
          calcularRND = Aleatorigrande
        Else
          .MoveNext
        End If
      Loop
      End If
      CodigoSal = "S-" & CStr(calcularRND)
    End With
    rstSalidaAlmacen.Close: Set rstSalidaAlmacen = Nothing
End Function
Public Function CodificarAjuste()
  Dim codigoAjuste As String
  Dim numero As Integer
  rstAjusteAlmacen.Open "SELECT codigo_ajuste FROM Ajustes_Almacen;", cnnControlInv,
adOpenStatic, adLockOptimistic
  If rstAjusteAlmacen.RecordCount <> 0 Then
    rstAjusteAlmacen.MoveLast
    codigoAjuste = rstAjusteAlmacen!codigo_ajuste
    numero = CInt(Mid(codigoAjuste, 3, 7))
    CodigoAju = "A-" & CStr(numero + 1)
  Else
    CodigoAju = "A-10000"
  End If
  rstAjusteAlmacen.Close: Set rstAjusteAlmacen = Nothing
End Function
```

Esta función genera un código por cada factura emitida al cliente, para un mejor control de las ventas y mejorar las búsquedas.

```
Public Function CodificarFactura()
  Dim codigoFactura As String
  Dim calcularRND, numero As Long
  Randomize
  rstFactura.Open "SELECT nfactura FROM Factura;", cnnControlInv, adOpenStatic,
adLockReadOnly, adCmdText
  With rstFactura
    calcularRND = Aleatorigrande
    If .RecordCount <> 0 Then
      .MoveFirst
      Do Until .EOF
        codigoFactura = !nFactura
        numero = CLng(Mid(codigoFactura, 5, 9))
        If numero = calcularRND Then
          .MoveFirst
          calcularRND = Aleatorigrande
        Else
```



```

        .MoveNext
    End If
    Loop
    End If
    CodigoFac = "FAC-" & CStr(calcularRND)
End With
rstFactura.Close: Set rstFactura = Nothing
End Function

```

Esta función genera un código por cada cotización emitida a un cliente generando números aleatorios con la función random.

```

Public Function CodificarCotizacion()
    Dim codigoCotizacion As String
    Dim calcularRND, numero As Long
    Randomize
    rstCotizacion.Open "SELECT ncotizacion FROM Cotizacion;", cnnControlInv,
adOpenStatic, adLockReadOnly, adCmdText
    With rstCotizacion
        calcularRND = AleatorioGrande
        If .RecordCount <> 0 Then
            .MoveFirst
            Do Until .EOF
                codigoCotizacion = !ncotizacion
                numero = CLng(Mid(codigoCotizacion, 5, 9))
                If numero = calcularRND Then .MoveFirst
                    calcularRND = AleatorioGrande
                Else
                    .MoveNext
                End If
            Loop
        End If
        CodigoCot = "COT-" & CStr(calcularRND)
    End With
    rstCotizacion.Close: Set rstCotizacion = Nothing
End Function

```

Esta función inserta un registro de una salida de un producto por factura o se saca un producto por vencimiento.

```

Public Function InsertarSalidaDelLote(CodigoProd As String, cantidadprod As Currency,
fecha As Date, lote As Long, fechacad As Boolean)
    rstLoteTemporal.Open "Lote_Producto", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdTable
    With rstLoteTemporal
        .AddNew
        !codigo_bodega = Nbodega
        If Len(CodigoAju) > 0 Then
            !codigo_salida = CodigoAju
        Else
            !codigo_salida = CodigoSal
        End If
        !codigo_prod = CodigoProd
        If fechacad Then
            !fecha_caducidad = fecha
        Else
            !fecha_entrada = fecha
        End If
    End With

```



```
End If
!numlote = lote : !cantidad_fecha = CCur(cantidadprod)
.Update
End With
rstLoteTemporal.Close: Set rstLoteTemporal = Nothing
End Function
```

Esta función se utiliza para sacar producto de un lote determinado de una bodega; es invocada cuando se factura, se ajusta o se da de salida a los productos.

```
Public Function SacarProductoDelLote(CodigoProd As String, cantidadprod As Currency,
caducidad As Boolean) As Long
Dim i As Integer
Dim eliminolote As Boolean
eliminolote = False
If caducidad Then
rstLote.Open "SELECT * FROM Lote_Poducto WHERE codigo_bodega='" & Nbodega &
"'and codigo_prod='" & CodigoProd & "' AND fecha_caducidad <> ' ' AND codigo_entrada
LIKE 'E%' ORDER BY fecha_caducidad ;", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdText
Else
rstLote.Open "SELECT * FROM Lote_Poducto WHERE codigo_bodega='" & Nbodega &
"'and codigo_prod='" & CodigoProd & "' AND fecha_entrada <> ' ' AND codigo_entrada
LIKE 'E%' ORDER BY fecha_entrada ;", cnnControlInv, adOpenStatic, adLockOptimistic,
adCmdText
End If
With rstLote
For i = 1 To .RecordCount
If cantidadprod >= !cantidad_fecha Then
cantidadprod = CSng(cantidadprod - !cantidad_fecha)
If caducidad Then
Call InsertarSalidaDelLote(CodigoProd, !cantidad_fecha,
!fecha_caducidad, !numlote, True)
Else
Call InsertarSalidaDelLote(CodigoProd, !cantidad_fecha,
!fecha_entrada, !numlote, False)
End If
If cantidadprod = 0 Then
SacarProductoDelLote = !numlote
.Delete
rstLote.Close: Set rstLote = Nothing
Exit Function
End If
SacarProductoDelLote = !numlote
.Delete
eliminolote = True
Else
!cantidad_fecha = CSng(!cantidad_fecha - cantidadprod)
.Update
If caducidad Then
Call InsertarSalidaDelLote(CodigoProd, cantidadprod,
!fecha_caducidad, !numlote, True)
Else
Call InsertarSalidaDelLote(CodigoProd, cantidadprod,
!fecha_entrada, !numlote, False)
End If
End With
End Function
```



```
        If Not eliminolote Then _
            SacarProductoDelLote = !numlote
            rstLote.Close: Set rstLote = Nothing
            Exit Function
        End If
        .MoveNext
    Next i
End With
rstLote.Close: Set rstLote = Nothing
End Function
```

Esta función se utiliza para codificar la entrada de los productos creando un nuevo lote en el almacén, con el objetivo de controlar las búsquedas de los productos.

```
Public Function CodificarLote() As Long
    Dim RNDgenerado As Long
    Randomize
    RNDgenerado = NumLoteAleatorio
    rstLoteTemporal.Open "SELECT numlote FROM Detalle_Entrada;", cnnControlInv,
adOpenStatic, adLockReadOnly, adCmdText
    With rstLoteTemporal
        If rstLoteTemporal.RecordCount <> 0 Then
            .MoveFirst
            Do Until .EOF
                If RNDgenerado = !numlote Then
                    .MoveFirst
                    RNDgenerado = NumLoteAleatorio
                Else
                    .MoveNext
                End If
            Loop
        End If
        CodificarLote = RNDgenerado
    End With
    rstLoteTemporal.Close: Set rstLoteTemporal = Nothing
End Function
```

Esta función se utiliza para generar números aleatorios y codificar los lotes creados durante la entrada de los productos al almacén.

```
Public Function NumLoteAleatorio() As Long
    NumLoteAleatorio = CLng((999999 - 100000 + 1) * Rnd + 100000)
End Function
```

Esta función se utiliza para deshacer la salida de los productos del almacén manteniendo un registro de los productos que se han sacado del almacén, "anular facturas".

```
Public Function DeshacerSalidaAlmacen(Nsalida As String, codprod As String)
    Dim i As Integer, numeroLote As Long
    Dim f_entrada, f_caducidad As Date
    Dim codentrada As String
    Dim cantidad As Single
    Dim blncaducidad As Boolean
    blncaducidad = False
    rstLoteTemporal.Open "SELECT * FROM Lote_Poducto WHERE codigo_salida =" &
(Nsalida) & "' And codigo_prod =" & (codprod) & "' ;", cnnControlInv, adOpenStatic,
```



```
adLockOptimistic, adCmdText
  For i = 1 To rstLoteTemporal.RecordCount
    With rstLoteTemporal
      If Not IsNull(!fecha_caducidad) Then
        f_caducidad = !fecha_caducidad
        blncaducidad = True
      Else
        f_entrada = !fecha_entrada
      End If
      numeroLote = !numlote
      cantidad = !cantidad_fecha
      .Delete
    End With
    rstDetalleEntrada.Open "SELECT codigo_entrada FROM Detalle_Entrada WHERE
codigo_prod ='" & (codprod) & "' And numlote ='" & (numeroLote) & "';",
cnnControlInv, adOpenStatic, adLockReadOnly, adCmdText
    If rstDetalleEntrada.RecordCount > 0 Then codentrada =
rstDetalleEntrada!codigo_entrada
    rstDetalleEntrada.Close: Set rstDetalleEntrada = Nothing

    rstLote.Open " SELECT * FROM Lote_Poducto WHERE numlote = '" & numeroLote & "'
AND codigo_entrada LIKE 'E%';", cnnControlInv, adOpenDynamic, adLockOptimistic,
adCmdText
    With rstLote
      If .RecordCount > 0 Then
        !cantidad_fecha = CCur(!cantidad_fecha) + CCur(cantidad)
      Else
        .AddNew
        !codigo_bodega = Nbodega
        !codigo_prod = codprod
        !codigo_entrada = codentrada
        If blncaducidad Then
          !fecha_caducidad = f_caducidad
        Else
          !fecha_entrada = f_entrada
        End If

        !numlote = numeroLote
        !cantidad_fecha = cantidad
      End If
      .Update
    End With
    rstLote.Close: Set rstLote = Nothing
    rstLoteTemporal.MoveNext
  Next i
  rstLoteTemporal.Close: Set rstLoteTemporal = Nothing
End Function
```



11.3. REPORTES

11.3.1. Reporte 1:

Reporte de todos los Clientes de la Empresa.

<i>Código</i>	<i>Nombre del cliente:</i>	<i>Cédula</i>	<i>Dirección:</i>	<i>Teléfono:</i>
C-1842	JOSE ADRIAN LINDO	089-040383-0001B		823-1452
C-2512	COMERCIAL LALA. S.A.		CHICHIGALPA	634-5789
C-6688	FRANCISCO MORALES		LEON	
C-6756	CARLOS PEREZ		CHINANDEGA	



11.3.2. Reporte 2:

Reporte de todos los Proveedores de la Empresa.

<i>Código</i>	<i>Nombre del Proveedor.</i>	<i>Dirección:</i>	<i>Teléfono:</i>
P-4740	DISTRIBUIDORA SAN AGUSTIN	CHINANDEGA	341-7023
P-63344	DISTRIBUIDORA UNIVERSAL	MANAGUA	6547854
P-6519	CEMEX	MANAGUA	3-4517
P-6967	DISTRIBUIDORA IMACASA S.A.	MASAYA	665-6468

**11.3.3. Reporte 3:***Reporte para etiquetar los Lotes donde estarán ubicados los
Productos*

Nº Entrada: E-10004	Nº Bodega: B01
----------------------------	-----------------------

<i>Código</i>	<i>Descripción del Producto:</i>	<i>Cantidad:</i>	<i>No Lote:</i>	<i>Ubicación del Producto:</i>
B01-1003	ACETAMINOFEN	500	893371	ESTANTE # 01
B01-1008	AMIKACINA 100MG	250	585650	ESTANTE # 01
B01-1014	AMBROXOL 7.05ML	600	937698	ESTANTE # 02



11.3.4. Reporte 4:

Distribuidora PRODFARM S.A.

Documento: FAC-123456	N° Bodega: B01
Tipo de Documento: RR	N° Entrada: E-10003
Código Proveedor: P-4740	Fecha: 21/01/2007
Nombre: DISTRIBUIDORA SAN AGUSTIN	

<i>Código</i>	<i>Descripción:</i>	<i>Unidad:</i>	<i>No Lote:</i>	<i>Costo:</i>	<i>Cantidad:</i>	<i>Subtotal:</i>
B01-1003	ACETAMINOFEN	TABLETA	893371	0.32	1000	320
B01-1008	AMIKACINA 100MG	AMPOLLA	585650	23	200	4600
B01-1014	AMBROXOL 7.05ML	FRASSCO	937698	59.6	100	5960
Entrada Total:						10880

**11.3.5. Reporte 5:*****Distribuidora PRODFARM S.A.***

Documento: F-456987	Nº Bodega: B-01
Fecha: 21/01/2007	Nº de Salida: S-11029

<i>Código:</i>	<i>Descripción del producto:</i>	<i>Unidad:</i>	<i>Nº Lote</i>	<i>Cantidad</i>	<i>Subtotal</i>
B01-1003	ACETAMINOFEN 500MG	TABLETA	202665	20	5.8
B01-1014	AMBROXOL 7.05MG	FRASCO	937698	100	6851
B01-1008	AMIKACINA 100MG	AMPOLLA	490921	40	1075.6
Salida Total:					7932.4



11.3.6. Reporte 6:

Distribuidora PRODFARM S.A.

N° Bodega: B-01

PRODUCTOS DE FARMACIA.

<i>Código:</i>	<i>Descripción del Producto:</i>	<i>Unidad:</i>	<i>Existencia</i>	<i>Costo</i>	<i>saldo</i>
B01-1003	ACETAMINOFEN 500MG	TABLETA	1480	0.29	429.2
B01-1014	AMBROXOL 7.05MG	FRASCO	600	68.51	41106
B01-1008	AMIKACINA 100MG	AMPOLLA	410	26.89	11024.9
B01-1002	AZITROMICINA 600V	FRASCO	0	0	0
B01-1000	CEFTRIAZONA 1G	FAM	0	0	0
B01 -8805	CLAVOS DE ZINC	LIBRA	70	10	700
B01-1007	DEXTROMETROFANO 15 MG/5ML	AMPOLLAS	0	0	0
B01-1017	DEXTROMETROFANO 30 MG/5ML	FRASCO	0	0	0
B01-1006	DICLOFENAC 75 MG	AMP	0	0	0
B01-1018	DICLOXACILINA 125 MG/5ML	FRASCO	0	0	0
B01-1012	DIMETICONA	FRASCO	0	0	0
B01 -1009	GLUCONATO	FRASCO	0	0	0
B01-1014	HIDROCORTIZONA 500 MG	FAM	0	0	0
B01 -1016	LOPERAMIDA 2 MG	TABLETA	0	0	0
B01 -1005	METRONIDAZOL 100 MG	OVULO	0	0	0
B01 -1011	NISTATINA 10000 V	FRASCO	0	0	0
B01 -1013	OXIMETAZOLINA 0.05%	FRASCO	0	0	0
B01 -1001	PENICILINA PROCAINICA 4MILLONES	FAM	0	0	0
B01-5826	PERLINES 3 PULGADAS	UNIDAD	300	200	60000
B01-1010	SOLUCION SALINA 0.9%	FRASCO	0	0	0
B01-1019	TEOFINLINA 80MG/5ML	FRASCO	0	0	0
B01-1015	VITAMINA C 500 MG	TABLETA	0	0	0
B01-7511	ZINC 12 PIE	UNIDAD	150	133.33	19999.5



11.3.7. Reporte 7:

Distribuidora PRODFARM S.A.

Tipo de Documento: AF	No Bodega: B-01
Fecha: 21/10/2007	N° Ajuste: A-10000

<i>Código:</i>	<i>Descripción del producto:</i>	<i>Unidad:</i>	<i>Cant. Ajuste</i>	<i>Costo Ajuste</i>
B01-1003	ACETAMINOFEN 500MG	TABLETA	-10	
B01-1014	AMBROXOL 7.05MG	FRASCO	-3	
B01-1008	AMIKACINA 100MG	AMPOLLA	-1	

**11.3.8. Reporte 8:*****Distribuidora PRODFARM S.A.******De la alcaldía 1C al sur******Chinandega******Teléfono: 3417010***

Código Cliente: C-6688	Bodega: B-01
Nombre Cliente: FRANCISCO MORALES	No Factura: FAC-93384
Teléfono:	Fecha: 21/01/2007
Dirección: LEON	

<i>Descripción del producto</i>	<i>Unidad</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Subtotal</i>
ACETAMINOFEN 500MG	TABLETA	200	0.46	92
AMIKACINA 100MG	AMPOLLA	20	35	700
AMBROXOL 7.05MG	FRASCO	30	73	2190
IMPORTE:				2982
IGV:				447.3
Monto Total:				3429.3

Para reclamos de cualquier índole, esta factura es válida dentro de un periodo de 48 horas después de ser emitida.

**11.3.9. Reporte 9:*****Distribuidora PRODFARM S.A.******De la alcaldía 1C al sur******Chinandega******Teléfono: 3417010***

Código Cliente: C-6756	N° Bodega: B-01
Nombre Cliente: CARLOS PEREZ	N° Cotización: COT-70181
Dirección: CHINANDEGA	Fecha: 21/10/2007
	Teléfono:

<i>Descripción del Producto.</i>	<i>Unidad</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Subtotal</i>
ACETAMINOFEN 500MG	TABLETA	150	0.46	69
AMIKACINA 100MG	AMPOLLA	70	35	2450
AMBROXOL 7.05MG	FRASCO	80	73	5840
IMPORTE:				8359
IGV:				1253.85
Monto Total:				9612.85

Esta cotización es válida por un periodo de 10 días.



XII. REFERENCIAS BIBLIOGRÁFICAS

- ✓ Contabilidad Intermedia.
McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.
Javier Romero López.
- ✓ Monografias.com/trabajo20/Sistemas-Automatizado-envio/Sistemas-Autorizado-envio.shtml
- ✓ Programación avanzada con visual Basic
Francesco Balena
- ✓ Modulo de contabilidad del sistema de contabilidad completa
Enoc tomas Mendoza
Edwin Daniel padilla
Eder matus flores