

UNIVERSIDAD DE CIENCIAS COMERCIALES
Carrera : Ingeniería en Sistemas



TRABAJO MONOGRAFICO

**"PROPUESTA DEL SISTEMA DE CONTROL DE
INVENTARIO PARA LA EMPRESA DIFERCAL S.A."**

AUTOR:

BR. MAYCOLL JOSE MEDINA MORENO

TUTOR

Ing. FAUSTO QUIÑONEZ

ASESOR METODOLOGICO

Lic. JORGE PRADO DELGADILLO

DICIEMBRE DEL 2003

UNIVERSIDAD DE CIENCIAS COMERCIALES
Carrera : Ingeniería en Sistemas



TRABAJO MONOGRAFICO

**“ PROPUESTA DEL SISTEMA DE CONTROL DE
INVENTARIO PARA LA EMPRESA DIFERCAL S.A.”**

AUTOR:

BR. MAYCOLL JOSE MEDINA MORENO

TUTOR

Ing. FAUSTO QUIÑONEZ

ASESOR METODOLOGICO

Lic. JORGE PRADO DELGADILLO

DICIEMBRE DEL 2003



INDICE

I. INTRODUCCION.....	1
II. ANTECEDENTES.....	4
III. JUSTIFICACION.....	5
IV. OBJETIVOS.....	7
OBJETIVO GENERAL:	7
OBJETIVOS ESPECÍFICOS:.....	7
VI. MARCO METODOLOGICO.....	8
1 FASES DE DEFINICIÓN.....	8
1 1 ANALISIS DEL SISTEMA (REQUERIMIENTOS)	8
1 1 1 PLANIFICACION ESTRATEGICA DE LA INFORMACION	11
1 1 2 Análisis del Area del Negocio	12
1 1 3 DISEÑO DEL SISTEMA DEL NEGOCIO	24
1 2 PLANIFICACIÓN DEL PROYECTO	31
1 2 1 AMBITO DEL SOFTWARE.	31
1 2.2 CALCULO DE LA VIABILIDAD (ESTUDIO DE FACTIBILIDAD)	34
1 2 2 1 FACTIBILIDAD OPERATIVA	34
1 2 2 2 FACTIBILIDAD TÉCNICA	35
1 2 2 3 FACTIBILIDAD FINANCIERA	36
1 2 3 GESTION DEL PROYECTO	37
1 2 3 1 PERSONAL	37
1 2.3 2 PROBLEMA	38
1 2 3 3 PROCESO	42
1 2.3 4 PROYECTO	42
1 2 4 ESTIMACIONES	43
1 2 5 ANALISIS DE RIESGOS	44
1 2 6 PLANIFICACION TEMPORAL	47
1 2 7 GESTION DE LA CONFIGURACION	47
2 FASES DE DESARROLLO	48
2 1 DISEÑO	48
2 1 1 DISEÑO CONCEPTUAL	48
2 1 2 DISEÑO LOGICO	49
2 1 3 DISEÑO FÍSICO (COMPONENTES)	51
2 1 4 CODIFICACION	52
3. CONCLUSIONES.....	190
4. RECOMENDACIONES.....	191
5. BIBLIOGRAFIA.....	192
6. ANEXOS.....	194



Dedicatoria

Δ A JESUS Δ

A nuestra fe e inspiracizn de todos los días

A Licé Jorge Jesus Prado

A nuestros padres Æ eternos compaxeros

Mi Familia.

Agradecimientos

A Jorge Jesús Prado Delgadillo por su incondicional apoyo y paciencia, mi madre, María Auxiliadora Moreno Amaya, por creer en mí, todos estos años, la paciencia que han demostrado tener y el apoyo que me han brindado de manera incondicional.

A Dios, creador nuestro



I. INTRODUCCION

Actualmente las empresas han reconocido la importancia que tiene el poseer una organización adecuada de los recursos básicos, tales como Producción, Comercialización y Sistemas de información que pueden llegar a ser elementos importantes que determinen en un momento dado el éxito o fracaso de una empresa o negocio.

Con la disponibilidad actual de las computadoras se ha generado un incremento y diversificación de la información. La información que se genera por computadora difiere de diversas formas de aquella que se obtiene manualmente; a menudo se tiene una mayor Confiabilidad de las primeras. La información que genera la computadora puede llegar a multiplicar a grandes velocidades y con frecuencia, se trata con menos escepticismo que la obtenida por otros medios.

Con el fin de lograr la máxima utilidad de la información, ésta se debe administrar de manera correcta. Los directivos deben comprender que existen costos que se asocian a la producción, distribución, seguridad, almacenamiento y recuperación de la información.

A lo largo de la historia el hombre ha necesitado manejar y transmitir información, para ello ha estado creando máquinas y métodos.

La informática surge como la ciencia encargada del estudio y desarrollo de estas máquinas y métodos que constituyen la tecnología de información.





La tecnología de información abarca desde artefactos como papel, plumas, libros, cámaras, videograbadoras, computadoras, etcétera, hasta herramientas simbólicas como el lenguaje escrito, símbolos matemáticos, modelos químicos, etcétera.

Sin las computadoras y sus métodos no seríamos capaces de visualizar nuestro ambiente, entenderlo ni controlarlo en forma creativa en el nivel que lo vivimos actualmente.

Los continuos avances en tecnologías de información tienen un fuerte impacto sobre la forma en que las personas viven, pues facilita el trabajo en equipo y la comunicación a distancia

La tecnología de información ha hecho que las personas sean más eficientes Facilita el trabajo en equipo y la comunicación a distancia

Es por ello que las empresas buscan como tener sus registros en forma Automatizada

El término Sistemas se aplica generalmente al grupo de acciones, personal y procedimientos empleados para dar apoyo al proceso de datos En una organización, el conjunto de procedimientos, formas y equipos empleados para manejar los datos de la nómina se le denomina **Sistema de Nomina**. De manera similar, un **Sistema de Inventario** representa a todas las personas, formas, máquinas y procedimientos utilizado para procesar los datos de inventarios. En general, un Sistema es un conjunto de actividades relacionadas Puede no involucrar computadoras





Los Sistemas de Información (SI) definen como deben manejarse los datos. Los datos de entrada, que llegan al Sistema de varias maneras, serán procesados metódicamente a través de una secuencia de pasos organizados.

Los procedimientos para el proceso de datos y la emisión de la información resultante serán dictados por el analista, atendiendo peticiones de los directores de la organización.

Los Sistemas pueden diseñarse, por ejemplo, para que automáticamente se hagan pedidos para un inventario, se monitoreen las futuras ganancias de la corporación o se carguen las ventas realizadas con tarjetas de crédito o archivos en línea.

Al escrutinio cuidadoso de todos los aspectos de un Sistema, empleando una investigación metódica y paso por paso se le denomina **Análisis y Diseño de Sistemas**. El analista debe señalar el problema y luego sugerir una solución. Como a menudo existen muchas soluciones alternativas, el analista debe recomendar la más justificable económicamente.





II. ANTECEDENTES

La Distribuidora Fernández Caldera S.A (DIFERCAL) creada en el año de 1999, es una empresa que comercializa productos farmacéuticos a las farmacias, centro de salud, hospitales, etcétera, con el propósito de obtener divisas. Es una empresa que cuenta con todos los requerimientos modernos, incluyendo equipos de computación, aparentemente un Sistema Automatizado de Computación ayudaría a agilizar el trabajo del inventario y mantener organizada toda la información referente a ellos.

La Empresa ha crecido rápidamente en un plazo muy corto, lo que significa gran influencia de público.

La Empresa viene experimentando pérdidas económicas debido al descontrol del y manejo de los productos, una mala organización y sobre todo, reportes no actualizados y de muy poca credibilidad.

Actualmente debido a la diversificación de sus productos se hace necesario un control más eficiente sobre los mismos, aunque cuenta con tres equipos de cómputos estos no están siendo utilizados de forma óptima como herramienta que ayude en esta área.

La dirección de la empresa considera necesario actualizar el control de los productos de inventario, para lograr mejores informes y control sobre los mismos.





III. JUSTIFICACION

Automatizar los servicios operativos

La Empresa desea implementar un Sistema Automatizado que les permita llevar un control correcto del inventario de sus productos con el propósito de mejorar la atención e imagen con respecto a su clientela, mejorando la exactitud y consistencia de la información y aumentando la seguridad de los datos más importantes

Con el Sistema Automatizado que pretenden implementar, permitirá mejorar la organización de sus productos y de minimizar las perdidas económicas que afecten en este rubro a la empresa.

Además la implementación del Sistema Automatizado del inventario de productos médicos lograra hacer más ágil la información sobre el estado actual de cada uno de los medicamentos, liberando al personal encargado de los mismos dedicar más tiempo al análisis y comercialización de los mismos, puesto que el control logrará diversificar nuevas tareas en ese campo, ampliando la comunicación de cliente como empleado y facilitar la integración de funciones individuales

Proporcionar información de apoyo al servicio de toma de decisiones





Facilitar el logro de ventajas competitivas

La Empresa desea mejorar las capacidades, aumentando la velocidad de procesamiento, el manejo de un volumen de transacciones cada vez mayor y recuperar con rapidez la información

Disminuir los costos de operación por medio del control de los mismos mediante la vigilancia de los costos

Se pretende asegurar su clientela, formar bases para nuevos productos, dejar fuera a los competidores y mejorar acuerdos con sus proveedores mediante el Sistema Automatizado.





IV. OBJETIVOS

OBJETIVO GENERAL:

Implementación de un Sistema de Inventario que ayude al control de medicamentos para la solución de problemas

OBJETIVOS ESPECÍFICOS:

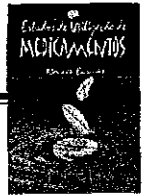
Identificar los elementos de información de inventario por medio de las diferentes técnicas de recolección de datos que permitan elaborar el conjunto de elementos del sistema

Resolver los distintos problemas que enfrenta la Empresa con el Sistema de Inventario

Diseñar formas y pantallas de datos del sistema

Presentar informes de inventario de acuerdo a las necesidades específicas





VI. MARCO METODOLOGICO

1. FASES DE DEFINICIÓN

1.1 ANÁLISIS DEL SISTEMA (REQUERIMIENTOS)

Para realizar el Análisis del Sistema primero realizamos una investigación preliminar

Realizando una investigación preliminar a la Distribuidora "Fernández Caldera S.A. (DIFERCAL) se encontró problema con el control de inventario de medicamentos, por lo que se procedió a realizar un análisis de requerimiento

Para realizarlo por medio de una recopilación de datos, se utilizaron las siguientes técnicas

- Entrevista
- Observación
- Revisión de Documentos

Para realizar la recolección de datos nos planteamos las siguientes preguntas

¿Qué se investigo?

Se investigo la forma en que llevan los servicios de ingreso, control, verificación y manejo de los Productos, así como su despacho.





¿Qué procesos se analizaron?

Los servicios realizados por los usuarios responsables del inventario, que utilizan para el control de los medicamentos

¿Qué técnicas utilizamos para estudiar estos procesos?

Para estudiar estos procesos utilizamos las entrevistas, observaciones y una serie de revisión de documentos de la empresa

Se realizó entrevistas a los usuarios responsables del inventario y se elaboro un cuestionario donde se pregunto lo siguiente

Analista: ¿Cómo es la forma de ingreso y retiro del producto?

Usuario: Para la Entrada (recepción) de productos se ingresa el código del producto, la descripción del producto, la unidad de medida, la cantidad, el tipo de Entrada si es por bonificaciones o por devoluciones, la fecha de Entrada, el precio unitario y el número de documento Para la salida de productos es lo mismo

Analista: ¿Cómo controlan la existencia del producto?

Usuario: Se controla manualmente, llevando un conteo diario de los productos que Ingresan y salen de bodega.





Observamos que cuando el cliente llega a solicitar un determinado producto el usuario tarda en dar una respuesta a la solicitud del cliente muchas veces por falta de tiempo y por un mal control del producto. A veces pierden tiempo en buscar un producto y muchas veces sin éxito. El usuario muchas veces introduce el mismo producto y muchas veces no manejan un control correcto en cuanto a las existencias exactas de los productos.

Cuando realizan un reporte se encuentra muchas veces que no está correcto ya que no presenta la información adecuada y real de la empresa.

Realizamos una revisión de los siguientes documentos:

Reportes quincenales o mensuales de Inventario

Reportes quincenales o mensuales de Inventario Físico.

Reportes de recepción de medicamentos

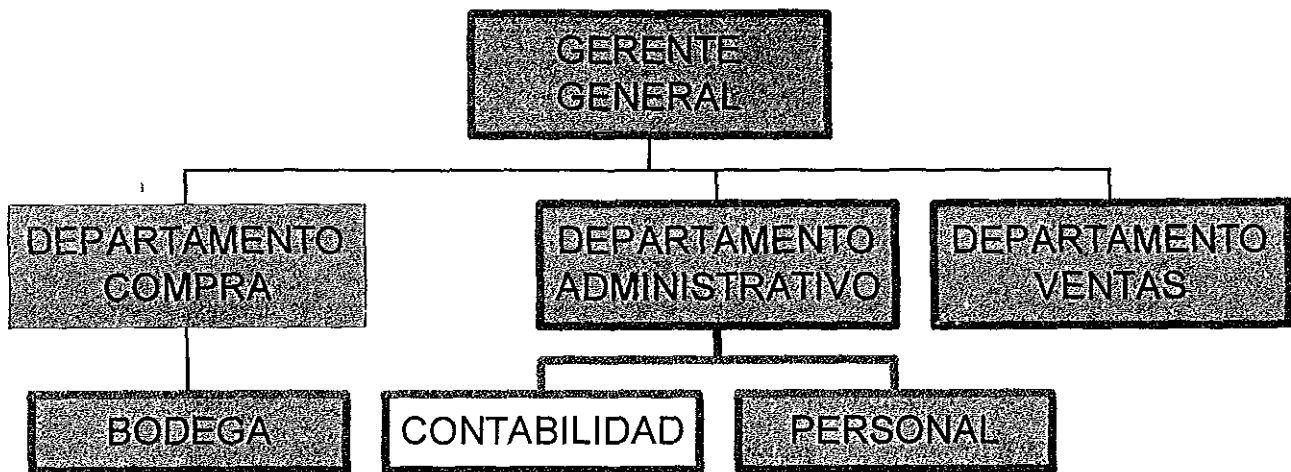
Reportes de salidas de medicamentos





1.1.1 PLANIFICACION ESTRATEGICA DE LA INFORMACION

ORGANIGRAMA DE LA EMPRESA DIFERCAL S.A.

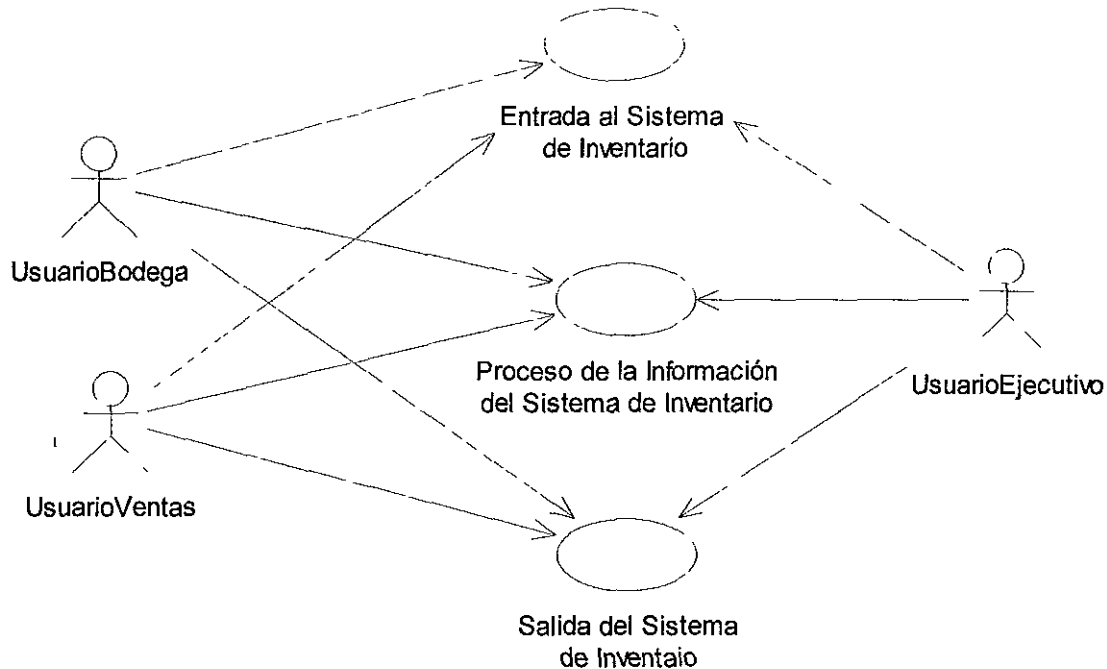


- ✓ **Departamento de Compras:** Responsable de Compra de Medicamentos, Control de los Proveedores
- ✓ **Departamento Administrativo:** Responsable del Control Contable y Financiero
- ✓ **Departamento de Ventas:** Responsable de promoción, Distribución y Ventas de los Medicamentos



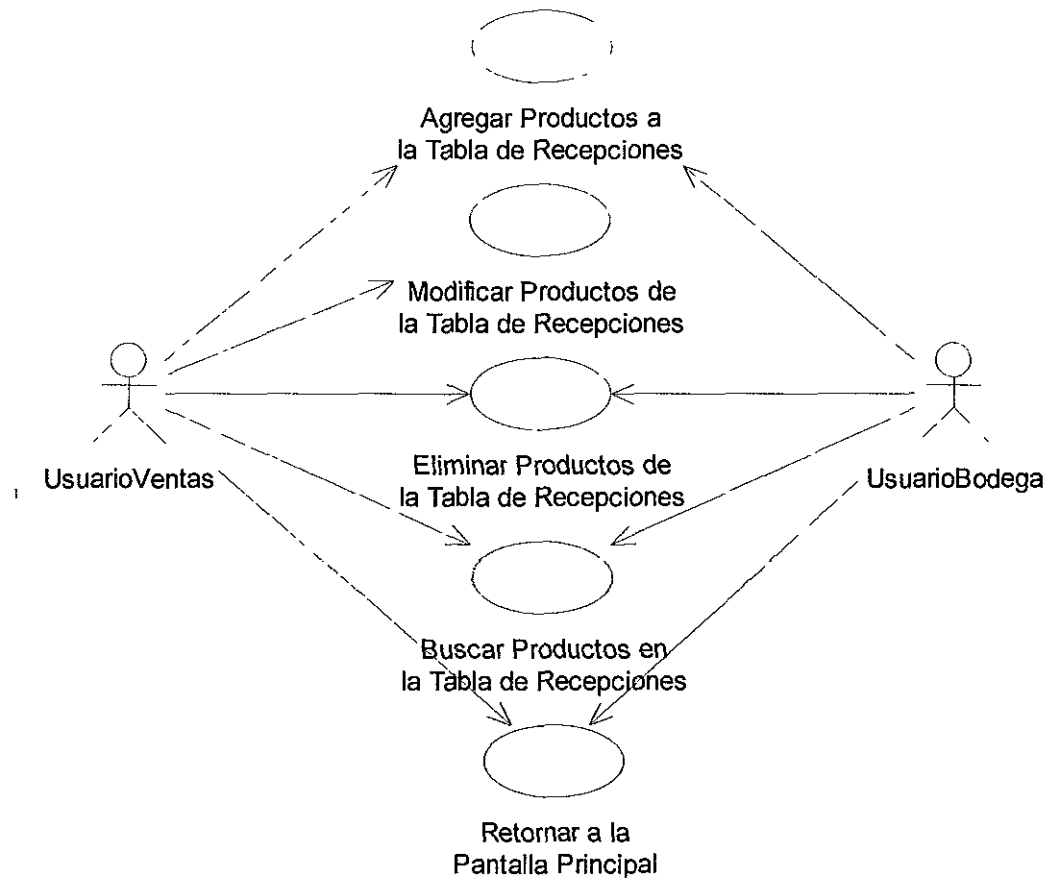


1.1.2 Análisis del Area del Negocio.



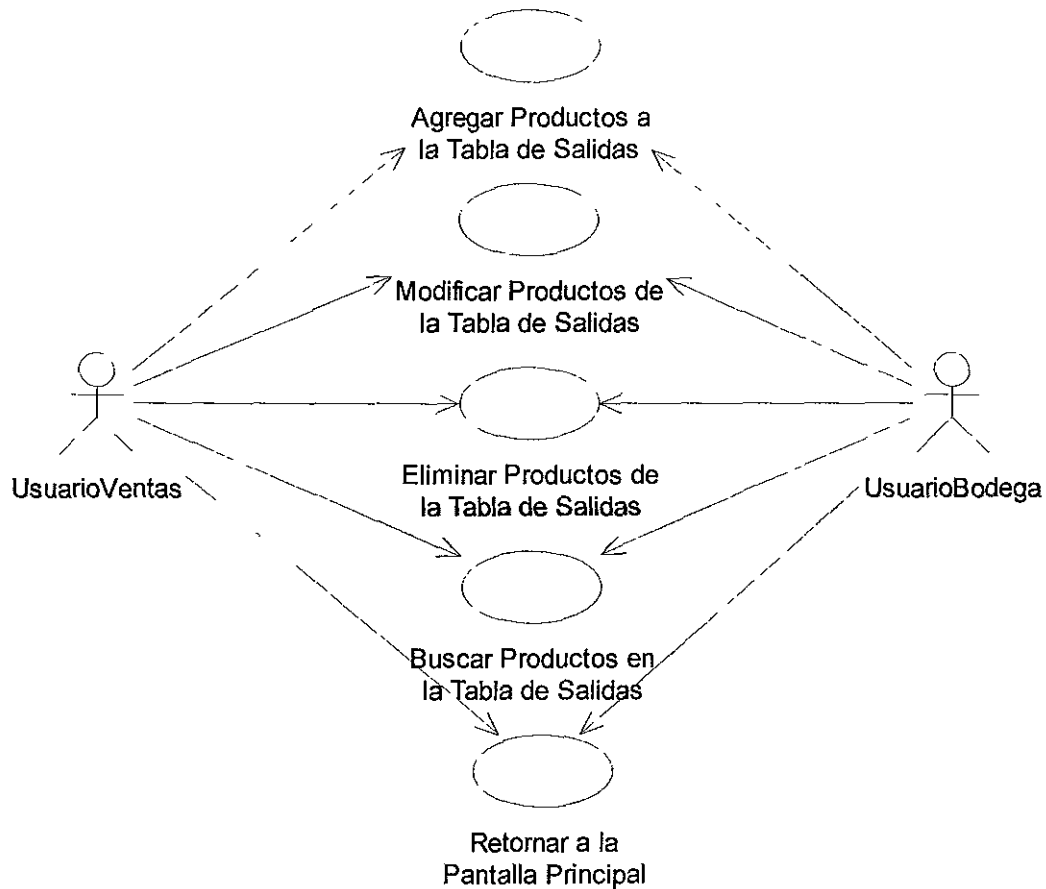
Nombre del Caso de Uso	Ingreso al Sistema
Actor Participante	UsuarioBodega, UsuarioVentas y UsuarioEjecutivo
Condición Inicial	El usuario activa el icono del Sistema de Inventario en su Pantalla, el Sistema responde
Flujo de Evento	1- El usuario escribe la contraseña y acciona continuar 2- El programa valida la contraseña presentando dos situaciones excluyentes 2 1- Contraseña valida y se accesa a otra interfaz. 2 2- Contraseña no valida, el Sistema no avanza y emite mensaje de Error en Contraseña 3- Salir del Sistema
Condición de Salida	El usuario activa el control de salida.
Requerimiento Especial	Después de tres intentos fallidos, el Sistema se bloquea





Nombre del Caso de Uso	Recepciones
Actor Participante	UsuarioBodega, UsuarioVentas.
Condición Inicial	El usuario activa el menú de Recepciones en su pantalla principal, el Sistema responde y accesa a otra interfaz.
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes: Agregar, Eliminar, Buscar, Modificar, Retornar 2- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar





Nombre del Caso de Uso

Salidas

Actor Participante

UsuarioBodega, UsuarioVentas

Condición Inicial

El usuario activa el menú de Salidas en su pantalla principal, el Sistema responde y accesa a la pantalla de Salidas

Flujo de Evento

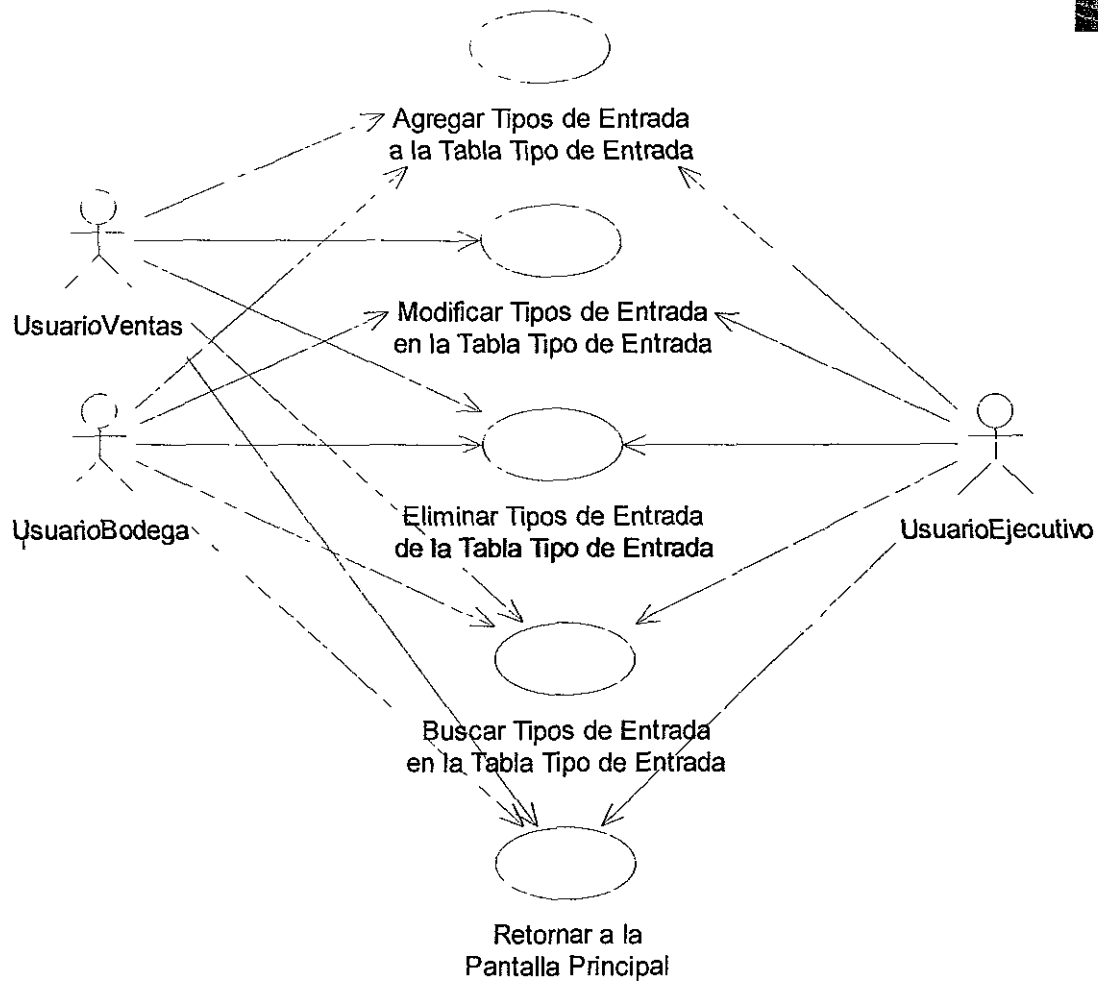
1- El usuario acciona cualquiera de las opciones siguientes: Agregar, Eliminar, Buscar, Modificar, Retornar.

2- El usuario acciona retornar a la pantalla principal

Condición de Salida

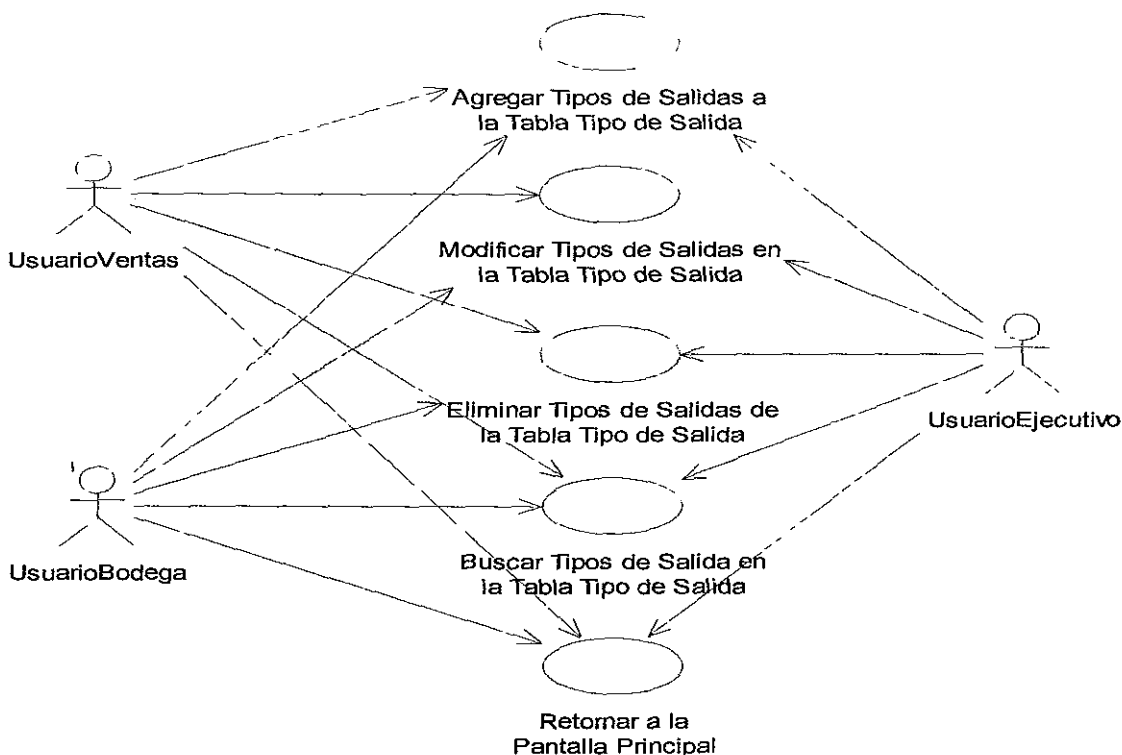
El usuario activa el control de retornar





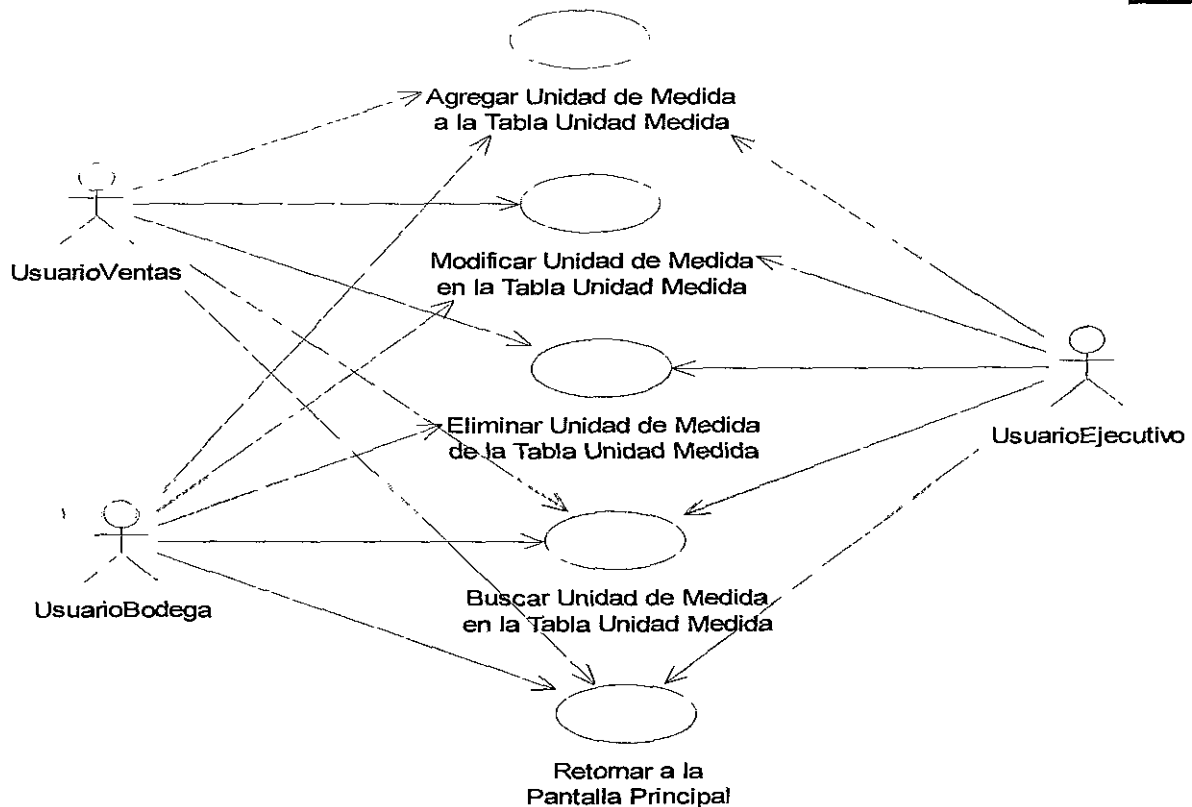
Nombre del Caso de Uso	Tipo de Entradas
Actor Participante	UsuarioVentas, UsuarioBodega y UsuarioEjecutivo
Condición Inicial	El usuario activa el menú de Tipo de Recepciones en su pantalla, el Sistema responde y accesa a la interfaz de Tipo de Recepciones.
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes Agregar, Eliminar, Buscar, Modificar, Retornar 2- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar





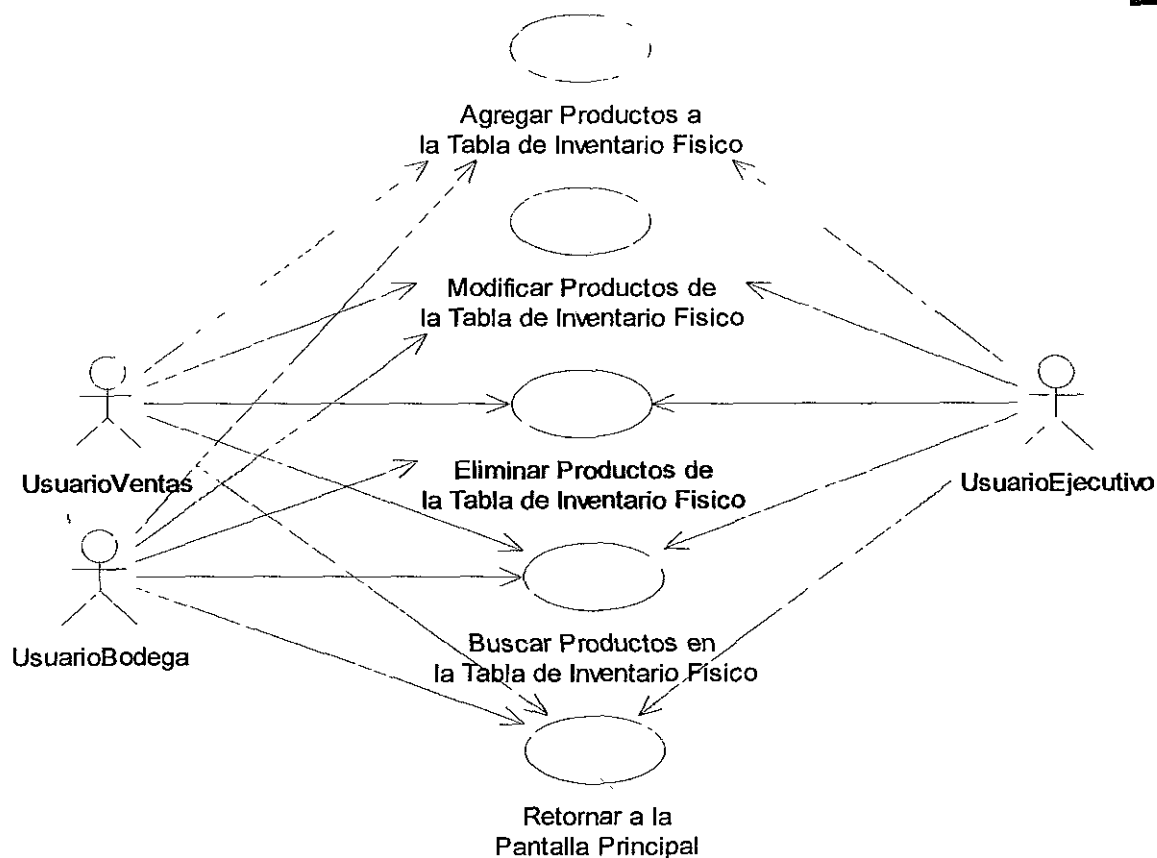
Nombre del Caso de Uso	Tipo de Salidas
Actor Participante	UsuarioVentas, UsuarioBodega y UsuarioEjecutivo.
Condición Inicial	El usuario activa el menú de Tipo de Salidas en su pantalla, el Sistema responde y accesa a la interfaz de Tipo de Salidas
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes Agregar, Eliminar, Buscar, Modificar, Retornar 3- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar.





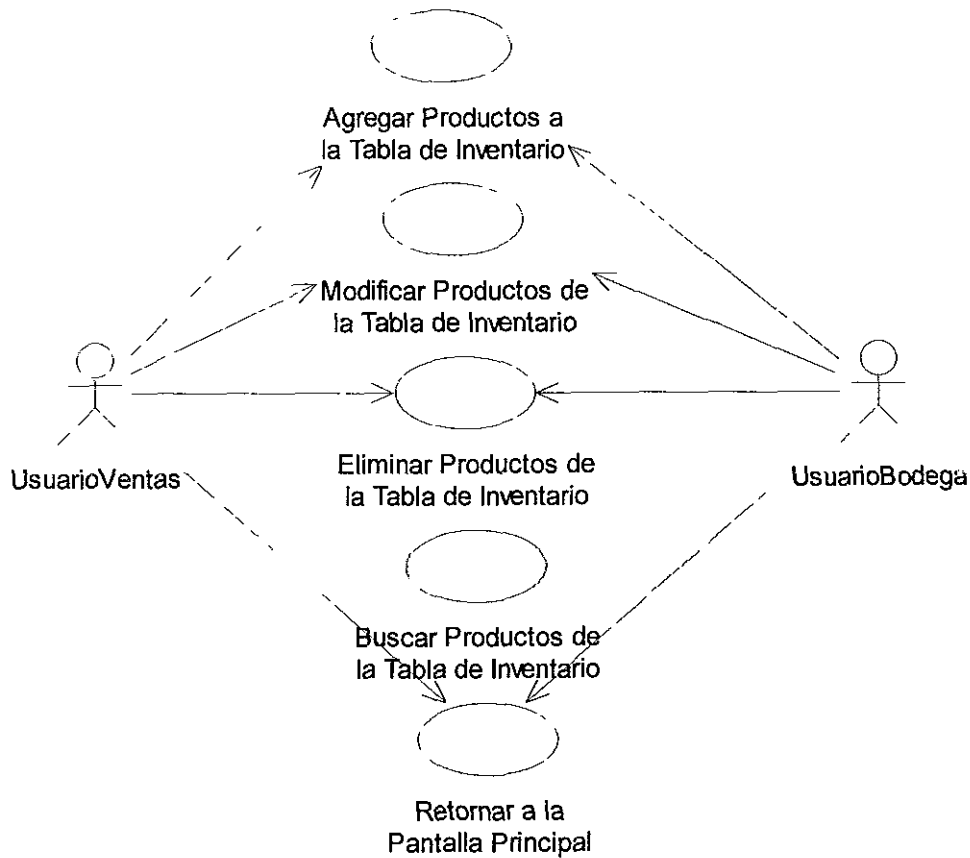
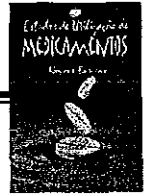
Nombre del Caso de Uso	Unidad de Medida
Actor Participante	UsuarioVentas, UsuarioBodega y UsuarioEjecutivo
Condición Inicial	El usuario activa el menú de Unidad de Medida en su pantalla, el Sistema responde y accesa a la interfaz de Unidad de Medida
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes. Agregar, Eliminar, Buscar, Modificar, Retornar 4- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar





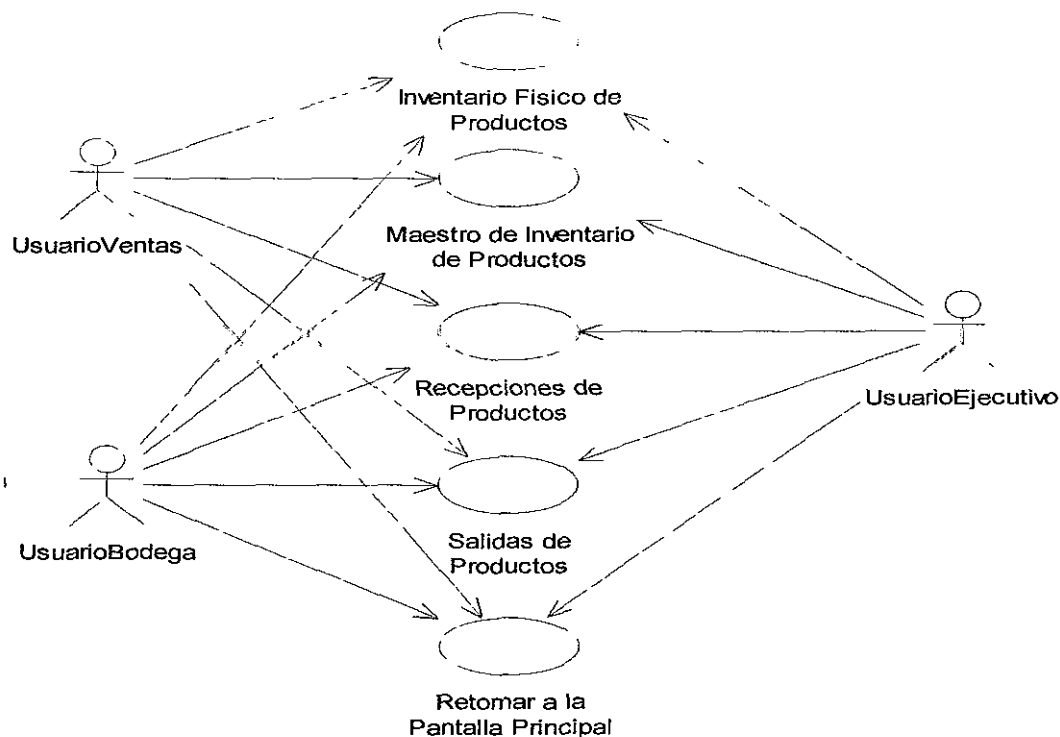
Nombre del Caso de Uso	Inventario Físico
Actor Participante	UsuarioVentas, UsuarioBodega y UsuarioEjecutivo
Condición Inicial	El usuario activa el menú de Inventario Físico en su pantalla, el Sistema responde y accesa a otra interfaz
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes: Agregar, Eliminar, Buscar, Modificar, Retornar 2- El usuario acciona retornar a la pantalla principal.
Condición de Salida	El usuario activa el control de retornar





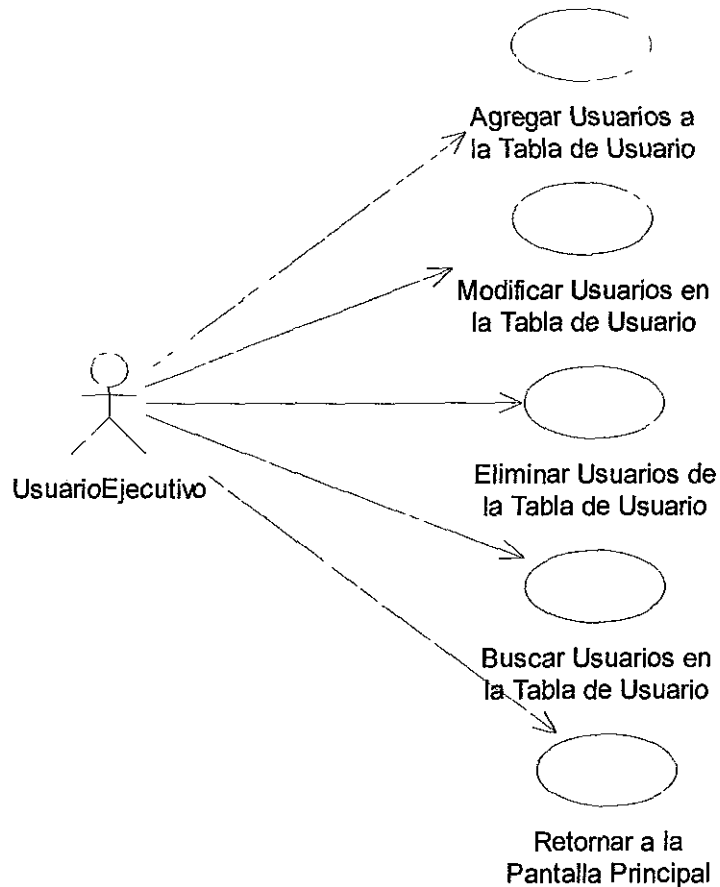
Nombre del Caso de Uso	Inventario
Actor Participante	UsuarioVentas, UsuarioBodega
Condición Inicial	El usuario activa el menú de Inventario en su pantalla, el Sistema responde y accesa a la interfaz de Maestro de Inventario
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes: Agregar, Eliminar, Buscar, Modificar, Retornar. 2- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar





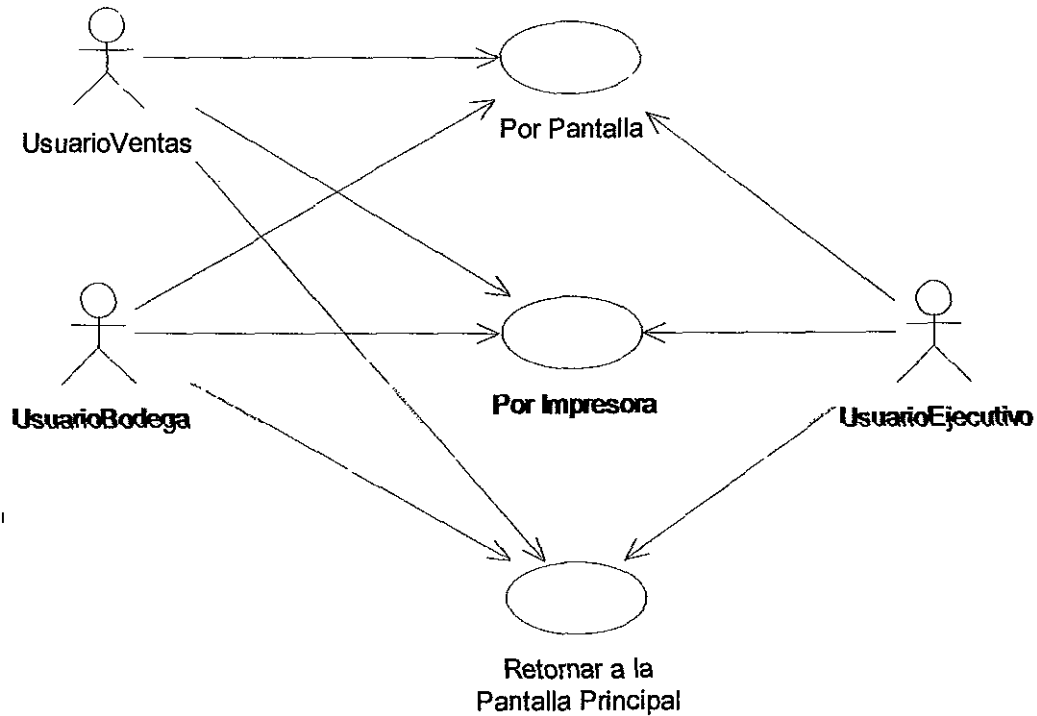
Nombre del Caso de Uso	Reportes.
Actor Participante	UsuarioVentas, UsuarioBodega, UsuarioEjecutivo
Condición Inicial	El usuario activa el menú de Reportes en su pantalla, el Sistema responde y accesa a otra interfaz
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes Inventario Físico, Maestro de Inventario, Recepciones, Salidas, Retomar, y accesa a la interfaz de Reportes 2- El usuario digita la fecha inicial y la fecha final y acciona aceptar. 3- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar





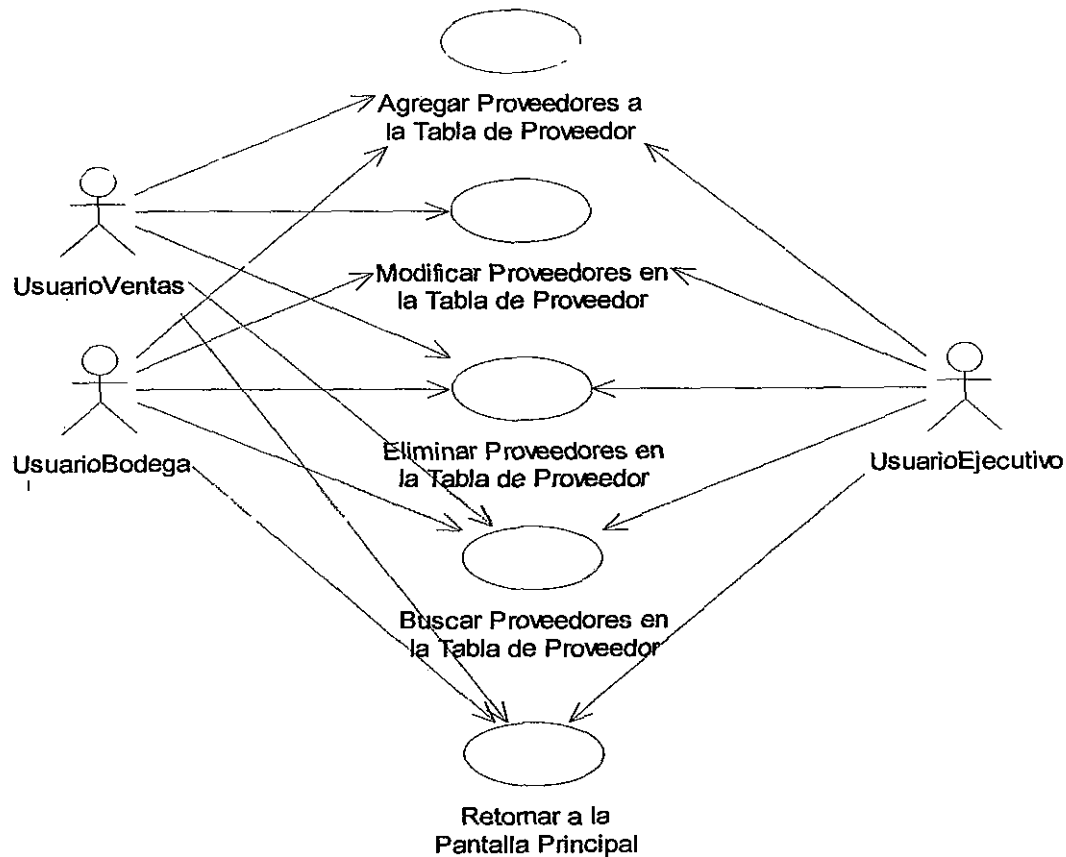
Nombre del Caso de Uso	Usuarios
Actor Participante	UsuarioEjecutivo
Condición Inicial	El usuario activa el menú de Usuarios en su pantalla, el Sistema responde y accesa a la interfaz de Usuarios
Flujo de Evento	1- El usuario acciona cualquiera de las opciones siguientes: Agregar, Eliminar, Buscar, Modificar, Retornar 5- El usuario acciona retornar a la pantalla principal
Condición de Salida	El usuario activa el control de retornar





Nombre del Caso de Uso :	DocumentoSistema
Actor Participante :	UsuarioBodega, UsuarioVentas, UsuarioEjecutivo.
Condición Inicial :	El usuario activa el menú de DocumentoSistema en su pantalla, el Sistema responde y accesa a la interfaz de Documento del Sistema.
Flujo de Evento :	1- El usuario acciona cualquiera de las opciones siguientes: Por Pantalla y Por Impresora. 2- El usuario acciona retornar a la pantalla principal.
Condición de Salida :	El usuario activa el control de retornar.





Nombre del Caso de Uso	Proveedor.
Actor Participante	UsuarioVentas, UsuarioBodega y UsuarioEjecutivo.
Condición Inicial	El usuario activa el menú de Usuarios en su pantalla, el Sistema responde y accesa a la interfaz de Proveedor
Flujo de Evento	: 1- El usuario acciona cualquiera de las opciones siguientes: Agregar, Eliminar, Buscar, Modificar, Retornar 6- El usuario acciona retornar a la pantalla principal
Condición de Salida	: El usuario activa el control de retornar.





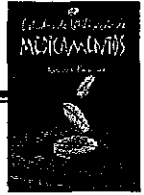
1.1.3 DISEÑO DEL SISTEMA DEL NEGOCIO

MODELO DEL SISTEMA DE INVENTARIO DE LA DISTRIBUIDORA FERNANDEZ CALDERA S.A.

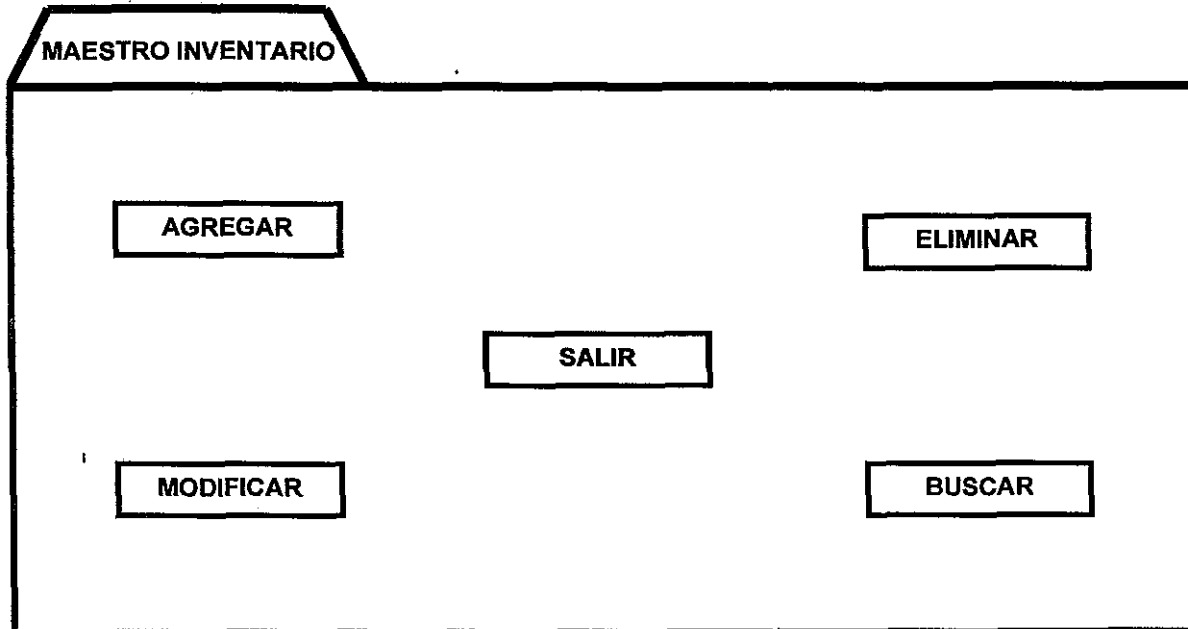
PANTALLA PRINCIPAL

ARCHIVO	MOVIMIENTOS	REPORTES	AYUDA
MAESTRO DE INVENTARIO	RECEPCIONES	MAESTRO DE INVENTARIO	DOCUMENTO DEL SISTEMA
INVENTARIO FISICO	SALIDAS	INVENTARIO FISICO	ACERCA DE..
PROVEEDOR	CIERRE	RECEPCIONES	
TIPOS DE SALIDAS		SALIDAS	
TIPOS DE RECEPCIONES		PROVEEDOR	
UNIDAD DE MEDIDA			
USUARIOS			
SALIR			

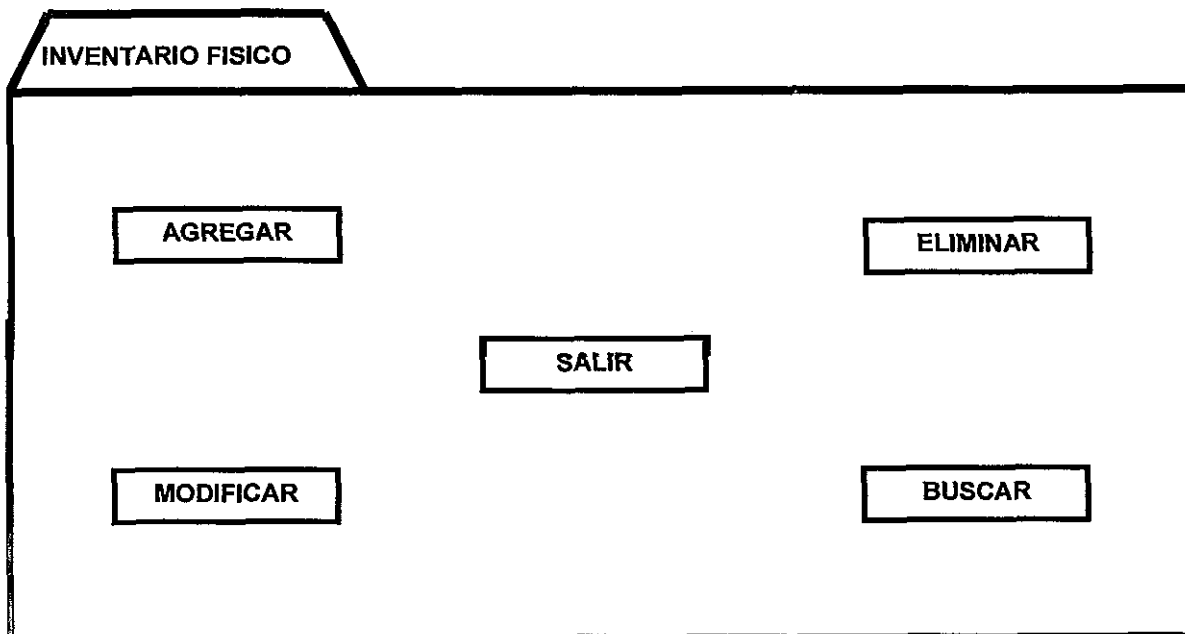




MODELO DEL DOMINIO DE APLICACION

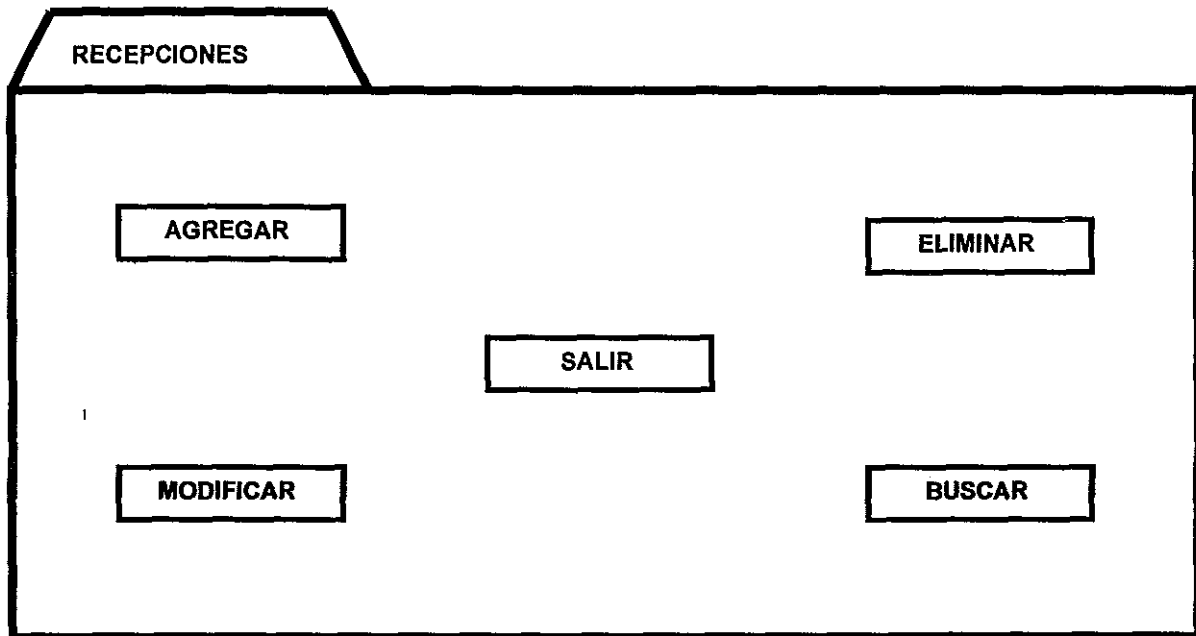


MODELO DEL DOMINIO DE APLICACION

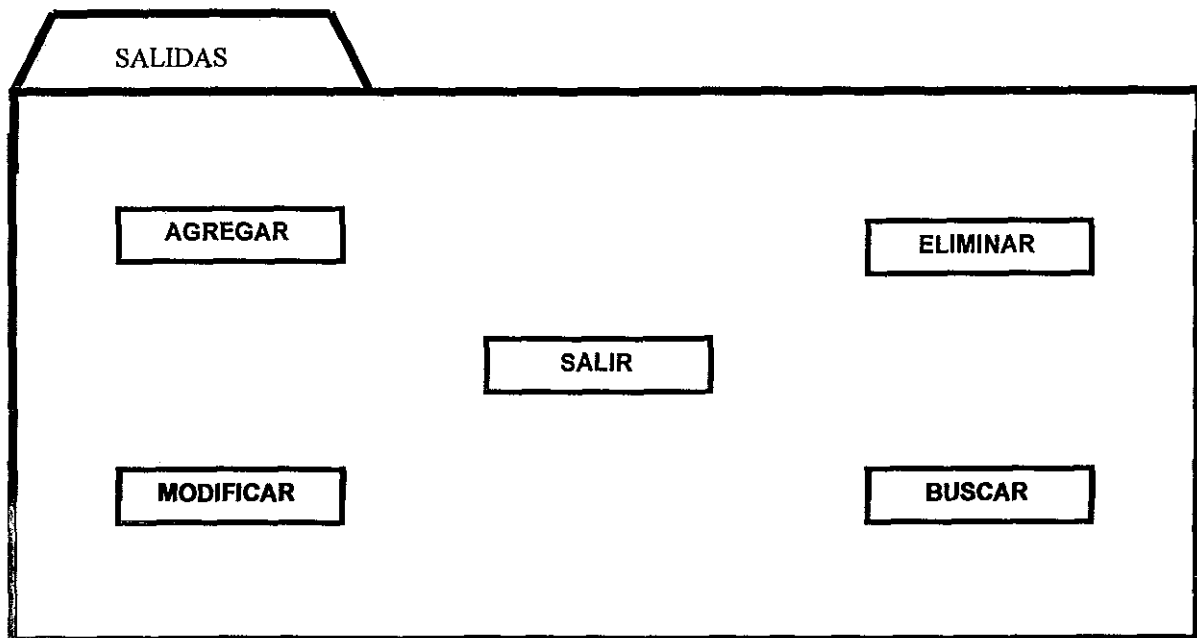




MODELO DEL DOMINIO DE APLICACION

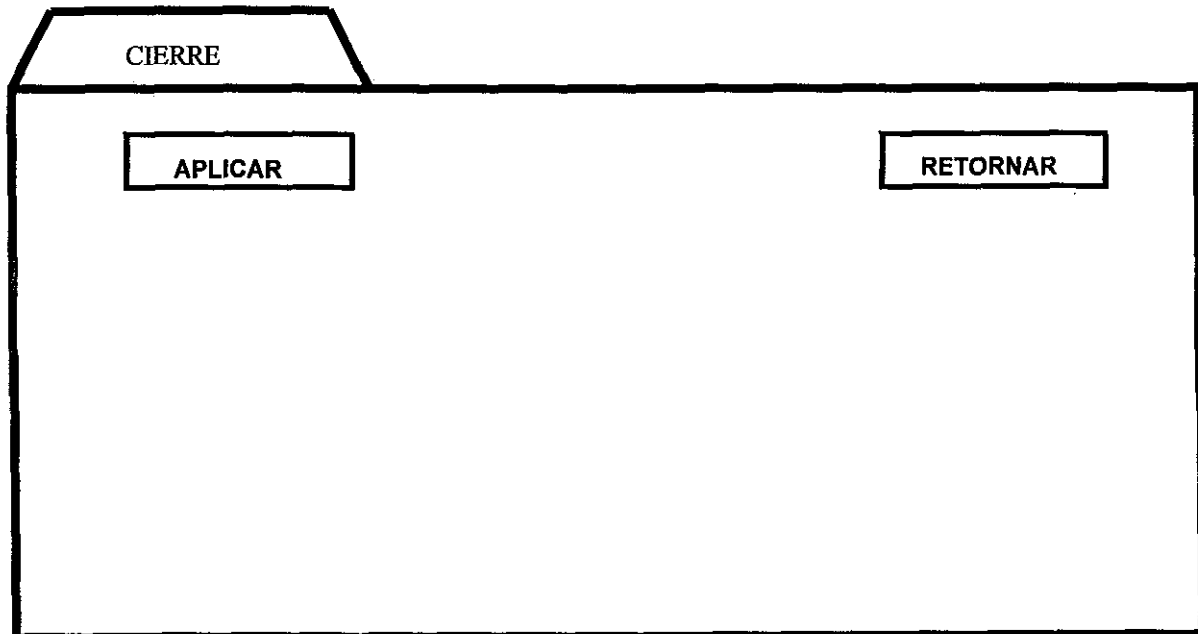


MODELO DEL DOMINIO DE APLICACION

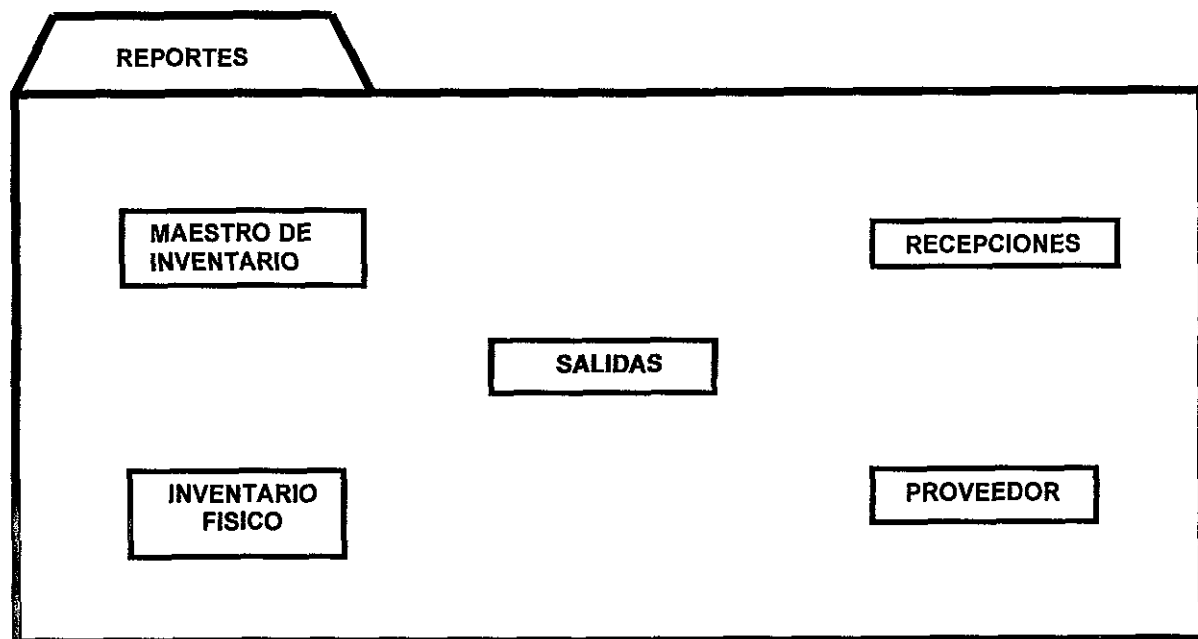




MODELO DEL DOMINIO DE APLICACION

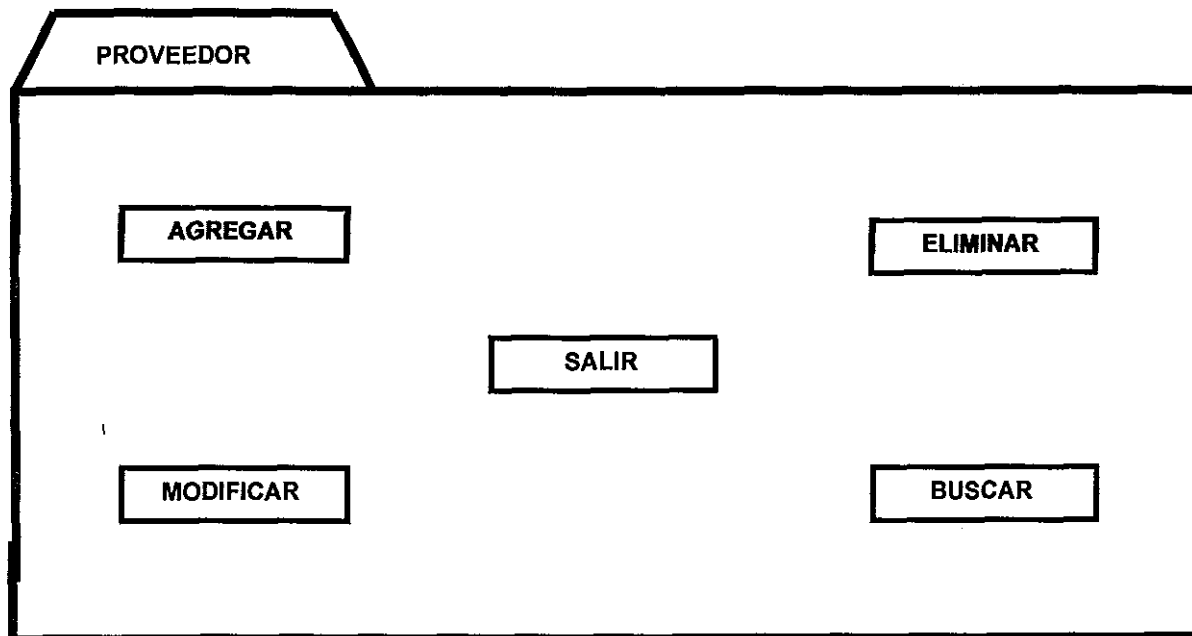


MODELO DEL DOMINIO DE APLICACION

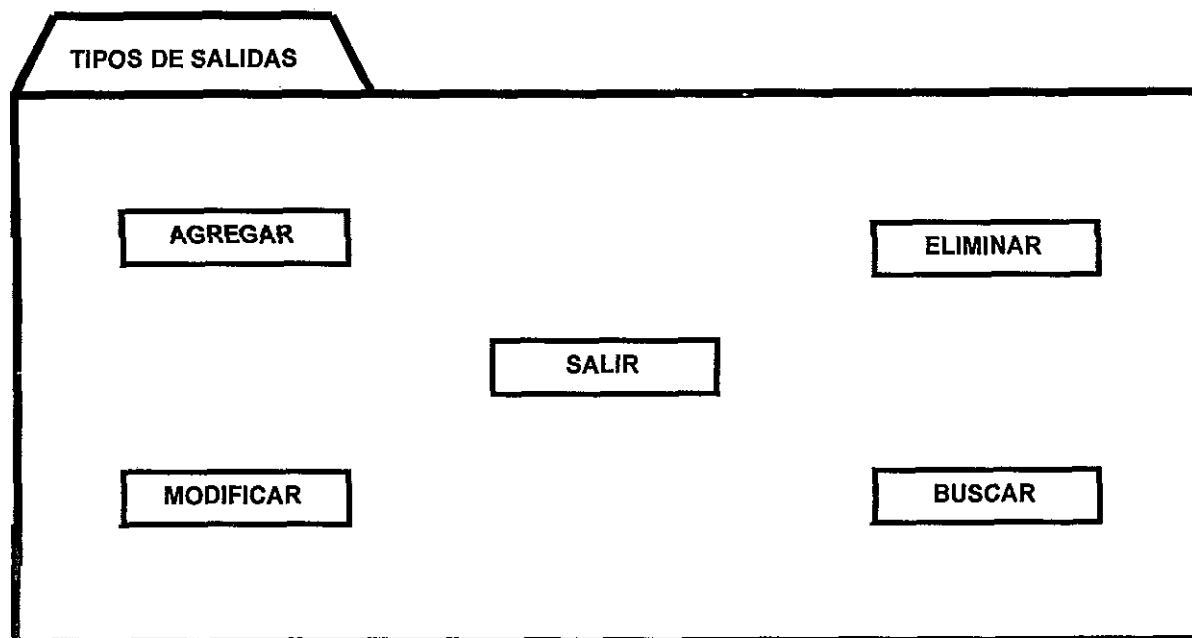




MODELO DEL DOMINIO DE APLICACION

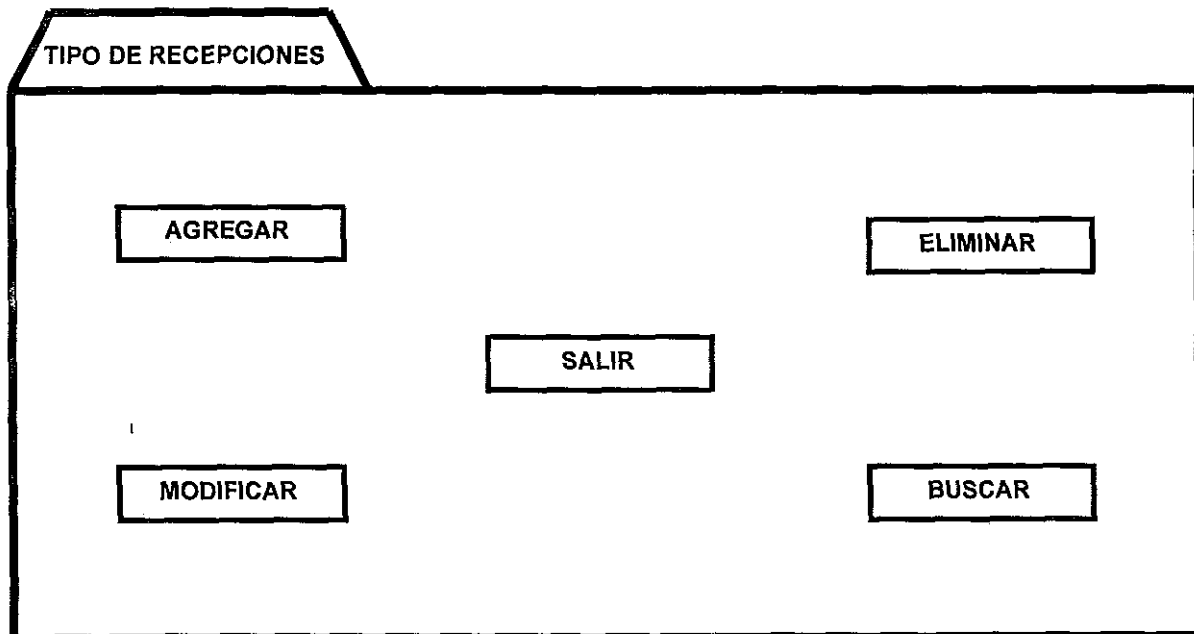


MODELO DEL DOMINIO DE APLICACION

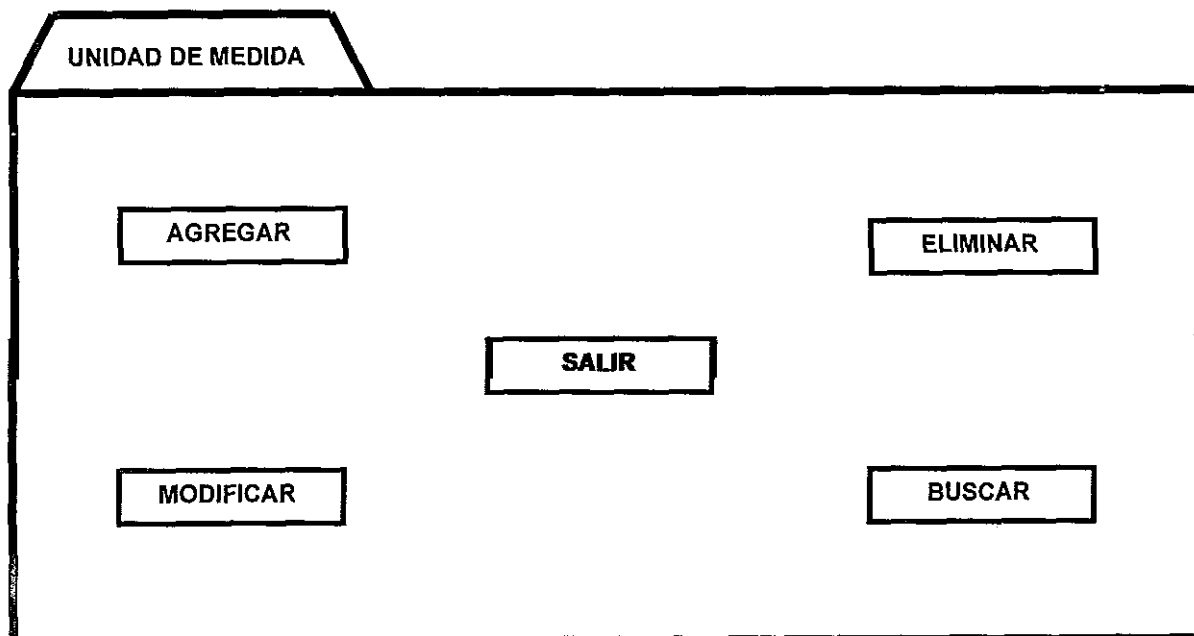




MODELO DEL DOMINIO DE APLICACION

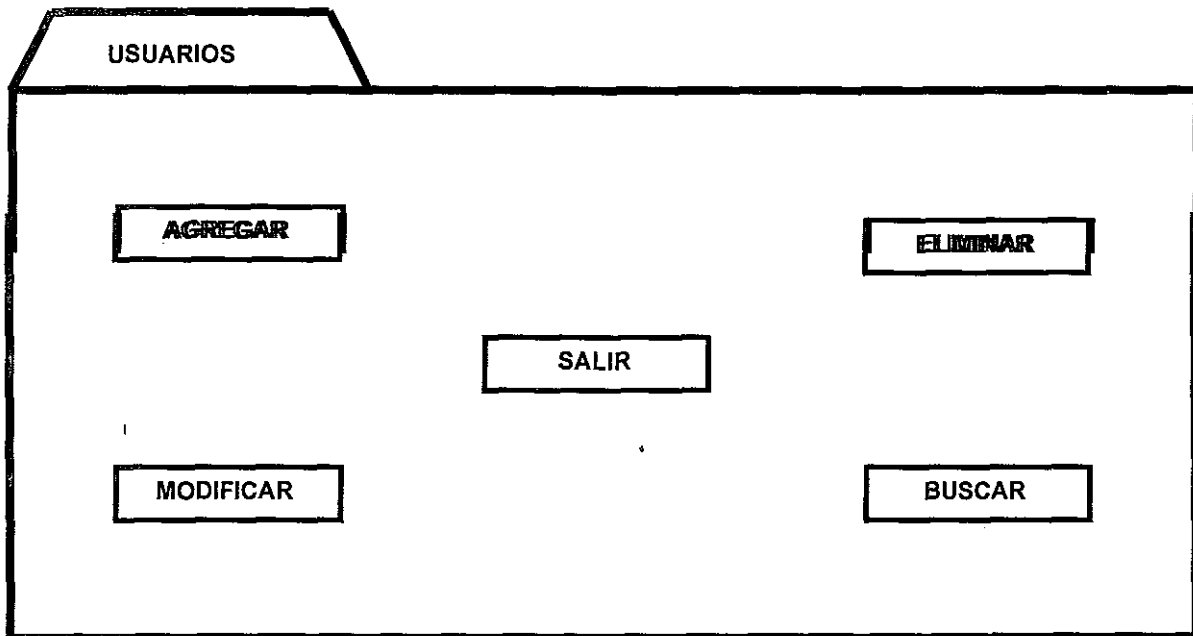


MODELO DEL DOMINIO DE APLICACION

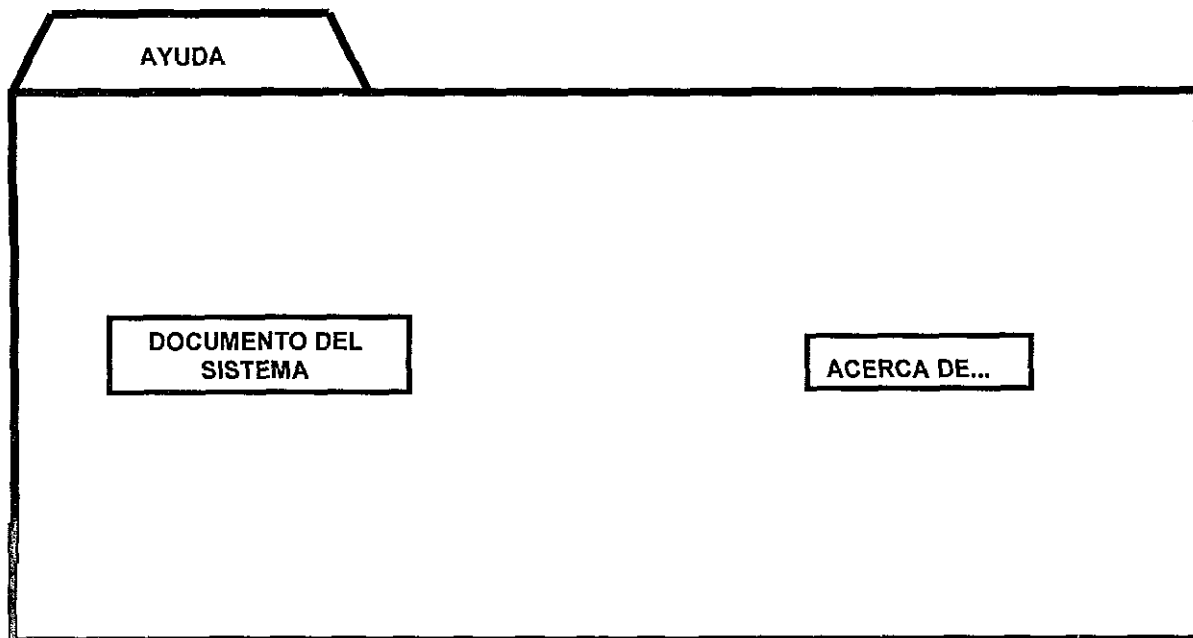




MODELO DEL DOMINIO DE APLICACION



MODELO DEL DOMINIO DE APLICACION





1.2 PLANIFICACIÓN DEL PROYECTO.

1.2.1 AMBITO DEL SOFTWARE:

Los clientes se presentan personalmente en las instalaciones o solicitan los productos telefónicamente. En aquellos casos en que el producto solicitado esta agotado, el Sistema indicara que la existencia esta en cero, es decir, rechaza la venta de ese producto y los empleados interactuan con el cliente para buscar algún producto sustituto. Con este fin, se consulta la tabla de productos

Las consultas típicas son

¿Qué tipo de medicamento tiene?

¿Qué tipo de presentación?

¿A que precio?

La tabla de productos tiene el detalle del inventario de medicamentos existentes en bodega y su descripción.

Cuando el producto esta en nivel critico se realiza una orden de compra. Con este fin, se consulta la tabla de proveedores.

Las dos tablas (Productos y Proveedores) se actualizan en línea, conforme ocurren las transacciones. A medida que se reciben las solicitudes de pedido, se verifican las existencias, si hay disponible se procede al despacho del producto y se actualiza la tabla de





producto Si la existencia no es suficiente, el cliente puede seleccionar otro producto o simplemente omitir el producto agotado. La tabla de proveedores se actualiza con nuevos proveedores

La tabla de productos es modificada con entradas a la bodega proveniente de mercancías recibida, ya sea por compra, por devoluciones o bien por bonificaciones. Las salidas por pedidos, cambia en ciertas características de los productos

Las principales salidas del Sistema serán.

- ◆ Reportes de Recepción de Productos
- ◆ Reportes de Inventario Físico
- ◆ Reporte de Inventario
- ◆ Reportes de Salidas de Productos
- ◆ Reportes de Proveedores

El Sistema controlara aquellos productos agotados o en el limite de seguridad, para las cuales generara ordenes de compra que se enviaran al proveedor

Cada vez que se recibe un despacho de mercadería de algún proveedor, se realiza el ingreso a bodega y se actualiza el inventario, verificando previamente contra la orden de compra Si todo esta correcto, se genera una notificación de recibo de mercadería que se envía al proveedor.





Se extraen todas las funciones importantes del software:

- ❖ Solicitud del pedido por parte del cliente
- ❖ Consultar pedido en la Base de Datos Distribuida.
- ❖ Responder a las consultas
- ❖ Seleccionar los productos
- ❖ Aplicar las salidas de los productos
- ❖ Realizar orden de compra cuando el producto esta agotado
- ❖ Consultar el archivo proveedores en la Base de Datos Distribuida.
- ❖ Recepción de pedidos de compra en la bodega
- ❖ Actualizar la tabla de productos de la Base de Datos
- ❖ Reportes de control de ubicaciones del producto
- ❖ Reportes de recepción de mercadería.
- ❖ Reportes de Inventario.
- ❖ Reportes de Inventario Físico

El rendimiento del Software estará determinado por la velocidad de acceso a la Base de Datos y por el Hardware donde se va ha implementar

El Software del "SIM" estará limitado por el hardware al que tiene que acceder

Una misma función puede producir una diferencia de un orden de magnitud en el esfuerzo de desarrollo cuando se considera en un contexto con diferentes limites de rendimiento. El esfuerzo y los costos requeridos para desarrollar el Software SIM serian drásticamente diferentes si la función fuera la misma (por ejemplo. el usuario del software del área de bodega y el usuario del área de ventas quieren al mismo tiempo acceder a un mismo campo)





el software podría ser considerablemente mas complejo requiriendo, por tanto, un mayor esfuerzo de desarrollo

La función, el rendimiento y las restricciones están íntimamente relacionadas

El software que se va a desarrollar será una aplicación distribuida (cliente/servidor)

1.2.2. CALCULO DE LA VIABILIDAD (ESTUDIO DE FACTIBILIDAD).

1.2.2.1. FACTIBILIDAD OPERATIVA.

De acuerdo a las entrevistas y pláticas con los empleados a utilizar el Sistema, consideran necesario una efectividad del control de Inventario de medicamentos para la Empresa.

El 90% de los empleados consideran trabajar con el Sistema una vez implementado en la empresa. Para el desarrollo del Sistema se tomó en consideración las opiniones y recomendaciones de los trabajadores, sobre todo en los que los usaran a menudo

El Sistema permitirá un ahorro de compra y mantenimiento de Productos hasta un 30 % del gasto actual, lo que justifica en un gran volumen la implementación del mismo

El Sistema coadyudara a trabajar mas eficientemente a los usuarios pues contaran con el Inventario actualizado para su distribución.

El Sistema estará trabajando en un 90% a los 45 días de empezar su implementación y se establecerá totalmente en un lapso de 90 días a partir de su instalación





1.2.2.2. FACTIBILIDAD TÉCNICA

La empresa ya cuenta con el Equipo necesario para la implementación del Software que se describe a continuación

Dos computadora Pentium III de 800 MHz de velocidad, 256Mb de RAM, disco duro de 40 Gigabyte con Sistema Operativo Windows 98 para el área de Ventas y Bodega

Una computadora Pentium IV de 2 4 GHz de velocidad , 500Mb de RAM, disco duro de 80 Gigabyte con Sistema Operativo Windows 2000 (esta computadora será el servidor). Ubicada en el área de Administración

Tres Impresoras HP DeskJet 850c para las áreas de Venta, Bodega y Administración.

La Empresa cuenta con una Red, conectadas con todas las áreas del negocio

La Empresa incurrirá en el gasto del Software de SQL SERVER y Microsoft Visual Estudio✓

Este Equipo permitirá el crecimiento y desarrollo del Sistema a la par del crecimiento y desarrollo de la Empresa DIFERCAL SA.

El Equipo permite el soporte técnico y de capacitación de procesos estándares Es confiable, de fácil acceso, manejo y seguridad de los datos



SISTEMA DE INVENTARIO "SIM"



Se realizarán soportes técnicos y de capacitación antes y durante el desarrollo, después del desarrollo una vez instalado el Software se realizaran soportes técnicos y de capacitación durante el periodo de 3 meses

1.2.2.3. FACTIBILIDAD FINANCIERA

CANT.	DESCRIPCION	PRECIO UNIT	PRECIO TOTAL	VALOR INVERTIDO	VALOR x INVERTIR
2	Comp Pentium III	U\$750 00	U\$1,500 00	U\$1,500 00	0 00
1	Comp. Pentium IV	U\$1,200.00	U\$1,200 00	U\$1,200 00	0 00
3	Impresoras Hp850c	U\$120 00	U\$360 00	U\$360 00	0 00
3	Licencias Microsoft Visual Studio	U\$135 75	U\$407 25	0 00	U\$407 25
1	Licencia de SQL SERVER	U\$149 50	U\$149.50	0 00	U\$149 50
1	Analista de Sistema	U\$400 00	U\$400 00	0 00	U\$400 00
SUB TOTAL DEL PROYECTO					956 75

NOTA: Los U\$400 00 que recibirá el Analista de Sistema es por la realización del Análisis del Software





1.2.3. GESTION DEL PROYECTO

1.2.3.1. PERSONAL

En este caso la empresa donde vamos a desarrollar el Software es una organización pequeña, lo cual no hay necesidad de contratar a diseñadores y programadores de sistemas ya que el analista será. Diseñador y Programador del Sistema.

El analista cuenta con la cualidad de ser hábil, automotivado, cordial, sabe transmitir ideas y sobre todo autodisciplinado

En La gestión del proyecto se tiene que ver además del analista, también los usuarios. Los gerentes y empleados tienen buenas ideas con respecto a qué es lo que sí funciona y qué es lo que no, qué causa problemas y qué no, donde son necesarios los cambios y dónde no, y especialmente en qué partes será aceptado el cambio y en cuáles partes no.

Aun con toda la tecnología, las personas son las piezas más importantes para que la empresa trabaje en forma mas eficiente

Comunicarse y tratar con las personas es uno de los aspectos más importantes del trabajo que se consideró como analista de sistemas.

Actualmente los usuarios participan más en el desarrollo del Sistema.

- Los usuarios de la Distribuidora DIFERCAL tienen una mejor idea de lo que significa la ayuda que puedan brindarles el Sistema de inventario y la forma de cómo obtenerla





- Saben cuáles son las fallas de los sistemas actuales y como evitarlos
- Muchos de los usuarios ya han recibido capacitación en el uso de las computadoras y algún tipo de Software ofimático.
- Para desarrollar la aplicación, se necesita la participación continua de los usuarios para comprender las funciones de la empresa que se encuentra bajo estudio

1.2.3.2 PROBLEMA

El problema es una actividad que se asienta en el núcleo del Análisis de requisitos del software. Durante la exposición que se realizó del ámbito no se intenta descomponer el problema totalmente. Mas bien, la descomposición se aplica en dos áreas principales:

(1) la funcionalidad que debe entregarse y (2) el Proceso que se empleará para entregarlo.

➤ **Descomponiendo la funcionalidad:**

- ❖ Solicitud del pedido por parte del cliente
- ❖ Recepción de pedidos.
- ❖ Consultar pedido en la Base de Datos Distribuida
- ❖ Responder a las consultas
- ❖ Seleccionar los productos





- ❖ Aplicar las salidas de los productos.
- ❖ Realizar orden de compra cuando el producto esta agotado
- ❖ Consultar el archivo proveedores en la Base de Datos Distribuida
- ❖ Recepción de pedidos de compra en la bodega
- ❖ Actualizar la tabla de productos de la Base de Datos
- ❖ Reportes de Inventario.
- ❖ Reportes de Inventario Físico.
- ❖ Reportes de recepción de mercadería
- ❖ Reportes de salida de mercadería.

Una vez hecha la descomposición de la funcionalidad se debe indicar el Proceso que se empleará

El Proceso a utilizar es el Proceso del ciclo de vida para el Desarrollo de Sistemas

Una vez indicado el Proceso a utilizar pasamos a la descomposición del mismo.





➤ **Descomponiendo el Proceso:**

◆ Investigación preliminar.

- ✓ Entrevistas.
- ✓ Observación
- ✓ Revisión de Documentos.

◆ Determinación de requerimientos:

- ✓ Planificación Estratégica de la Información.
- ✓ Análisis del Area del Negocio
- ✓ Diseño del Sistema del Negocio.
- ✓ Ambito del software
- ✓ Calculo de la viabilidad.
- ✓ Gestión del proyecto.
 - Personal
 - Selección del grupo de trabajo
 - Problema.
 - Descomposición de Problema
 - Proceso.
 - Selección del Modelo de Proceso
 - Maduración del Problema y del Proceso.





□ Proyecto

- ✓ Estimaciones
- ✓ Análisis de Riesgos
- ✓ Planificación Temporal
- ✓ Gestión de la Configuración (Seguimiento).

◆ Diseño del Sistema:

- ✓ Diseño Conceptual
- ✓ Diseño Lógico
- ✓ Diseño Físico y Despliegue

◆ Desarrollo del Sistema

- ✓ Codificación.

◆ Prueba del Sistema.

- ✓ Pruebas

◆ implementación y Evaluación.

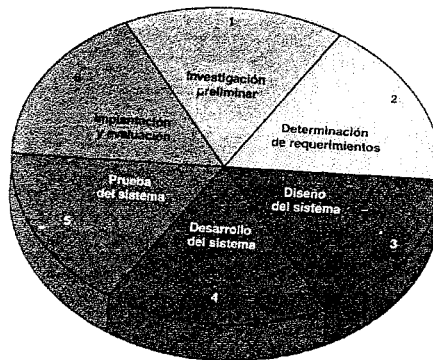
- ✓ Entrenamiento a los Usuarios
- ✓ Instalación de la Aplicación.





1.2.3.3 PROCESO

A como indicamos anteriormente el Proceso a utilizar es el Proceso del ciclo de vida para el Desarrollo de Sistemas como se muestra en la siguiente figura.



✓ **Maduración del problema y del Proceso:**

Ver anexo: Figura de Maduración del problema y del Proceso

1.2.3.4 PROYECTO

Para realizar este proyecto se utilizan diez señales que indicaran que el Sistema podría estar en peligro descrito en el marco conceptual del inciso 1.2.3.4 de la pagina #16.

Durante el desarrollo del proyecto y por medio de la comunicación que se ha mantenido con los clientes se ha logrado comprender las necesidades de los clientes.

Se considera que se ha definido el ámbito del software completamente tratando de abarcar todo los procesos que va ha realizar el software.





Se ha venido evitando que las necesidades del negocio cambien realizando un buen Análisis del área del negocio.

Como gestor de proyecto se realizaron incentivos para conseguir una rotación del personal mínima para evitar atrasos.

1.2.4. ESTIMACIONES

Cocomo [Capta Ratio de Productividad]

Archivo Edición Ver Ingreso de datos Calcular Datos Ventana Ayuda

Proceso del Cálculo de Ratio de Productividad

	Descripción	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
	Exp.Capacidad Des	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
	ICASE madurez y c	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
▶	PROD	4	7	13	25	50

Usted Seleccionó el Ratio de: **4**

Valor Meses-Personas es: **0.75**

Introduzca Costo del Software: **6.75**

Costo del Software

717.56

Retornar

Calcular Software

Retornar Pantalla

Como se mira el costo del proyecto es de U\$956.75, a esto se le suma el costo de U\$717.56 del Software que se realizó lo que hace un total neto del proyecto de U\$1,674.31. El costo





del Software se realizo haciendo una estimación con el COCOMOII como se muestra en la Figura

La estimaciones se realizaron por el número total de pantallas, el número de tablas en el Servidor, el Número de Reportes y por la Capacidad del Desarrollador

1.2.5. ANALISIS DE RIESGOS

Para realizar el Análisis de riesgos se plantearon las siguientes preguntas que provienen de los datos del riesgo obtenidos mediante las encuestas realizadas a gestores de proyectos de software expertos de diferentes partes del mundo Las preguntas están ordenadas por su importancia relativa para el éxito de nuestro proyecto

1. ¿Se han entregado los gestores del software y clientes formalmente para dar soporte al proyecto?
2. ¿Están completamente entusiasmados los usuarios finales con el proyecto y con el Sistema/producto a construir
3. ¿Han comprendido el Equipo de ingenieros de software y los clientes todos los requisitos?
4. Han estado los clientes involucrados por completo en la definición de los requisitos?
- 5 ¿Tienen los usuarios finales expectativas realistas?
- 6 ¿Es estable el ámbito del proyecto?
- 7 ¿Tiene el ingeniero del software el conjunto adecuado de habilidades?
- 8 ¿Son estables los requisitos del proyecto?
9. ¿Tiene experiencia el Equipo del proyecto con la tecnología a implementar?
10. ¿Es adecuado el número de personas del Equipo del proyecto para realizar el trabajo?





11 ¿Están de acuerdo todos los clientes/usuarios en la importancia del proyecto y en los requisitos del Sistema/producto a construir?

De acuerdo a estas 11 preguntas se realiza una evaluación del impacto del riesgo como se muestra en el siguiente cuadro





Componentes		RENDIMIENTO	SOPORTE	COSTE	PLANIFICACIÓN TEMPORAL
Categoría					
CATASTROFICA	1	Dejar de cumplir los requisitos provocaría el fallo de la misión		Malos resultados en un aumento de costes y retrasos de la planificación temporal con gastos	
	2	Degradación significativa para no alcanzar el rendimiento técnico	El software no responde o no admite soporte	Recortes financieros significativos, presupuestos excedidos	Fecha de entrega inalcanzable
CRITICA	1	Dejar de cumplir los requisitos degradaría el rendimiento del Sistema hasta donde el éxito de la misión es cuestionable		Malos resultados en retrasos operativos y/o aumento de coste con un valor esperado	
	2	Alguna reducción en el rendimiento técnico	Pequeños retrasos en modificaciones de software	Algunos recortes de los recursos financiero posibles excesos del presupuesto	Posibles retrasos en la fecha de entrega
MARGINAL	1	Dejar de cumplir los requisitos provocaría la degradación de la misión secundaria		Los costes, impactos y/o retrasos recuperables de la planificación temporal con un valor estimado	
	2	De mínima a pequeña reducción en el rendimiento técnico	El soporte del software responde	Recursos financieros suficientes	Planificación temporal realista, alcanzable
DESPRECIABLE	1	Dejar de cumplir los requisitos provocaría inconvenientes o impactos no operativos		Los errores provocan impactos mínimos en el coste y/o planificación temporal con valor esperado menos	
	2	No hay reducción en el rendimiento técnico	Software fácil de dar soporte	Posible superávit de presupuesto	Fecha de entrega fácilmente alcanzable

(1) Posibles consecuencias de errores o defectos del software no detectados.

(2) Posibles consecuencias si el resultado deseado no se consigue





1.2.6. PLANIFICACION TEMPORAL

Al realizar nuestra planificación temporal utilizamos el gráfico de Gantt el cual nos va a permitir listar todas nuestras tareas y hitos. El gráfico de Gantt se observa en los anexos de este proyecto.

1.2.7. GESTION DE LA CONFIGURACION

Se realizará la Gestión de la Configuración (seguimiento de la planificación temporal) con el objetivo de ver si se ha desarrollado apropiadamente, define las tareas e hitos que deben seguirse y controlarse a medida que progresa el proyecto.

Este seguimiento se realizará efectuando reuniones periódicas del estado del proyecto donde se informará del progreso y de los problemas, evaluando los resultados de todas las revisiones realizadas a lo largo del Proceso de ingeniería del software, determinando si se han conseguido los hitos formales del proyecto.

Se realizará una tabla de proyecto para ver sus fechas previstas y reales de inicio y finalización, así como, las personas asignadas el esfuerzo asignado. Esta tabla se encuentra en los anexos.

Empleando las tablas junto con los gráficos de tiempo, permitirá hacer un seguimiento del progreso.



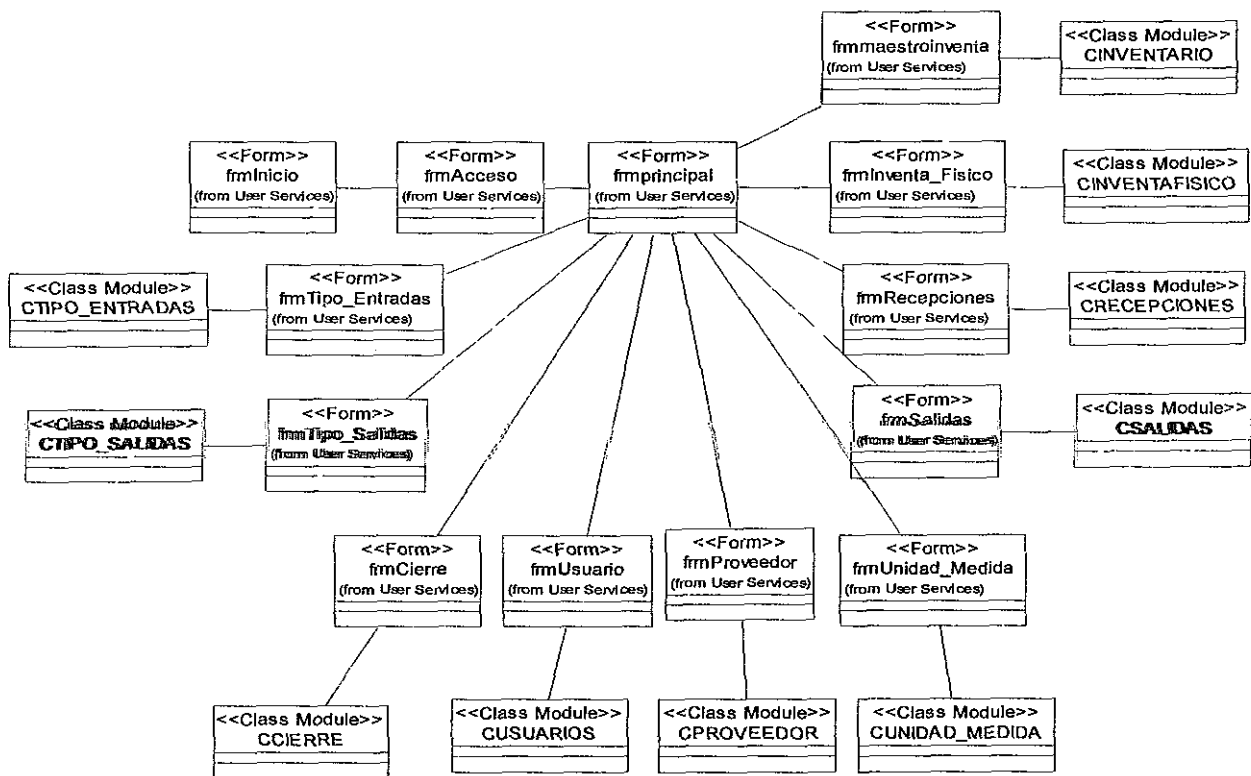


2. FASES DE DESARROLLO

2.1. DISEÑO

2.1.1 DISEÑO CONCEPTUAL

El Diseño Conceptual es una estrategia para desarrollar aplicaciones distribuidas debe comenzar por la comprensión de las necesidades de los usuarios y el negocio, es decir, identificando los requisitos de la aplicación, como se realizó anteriormente. El Diseño Conceptual consiste en las clases a como se observa a continuación





2.1.2 DISEÑO LOGICO

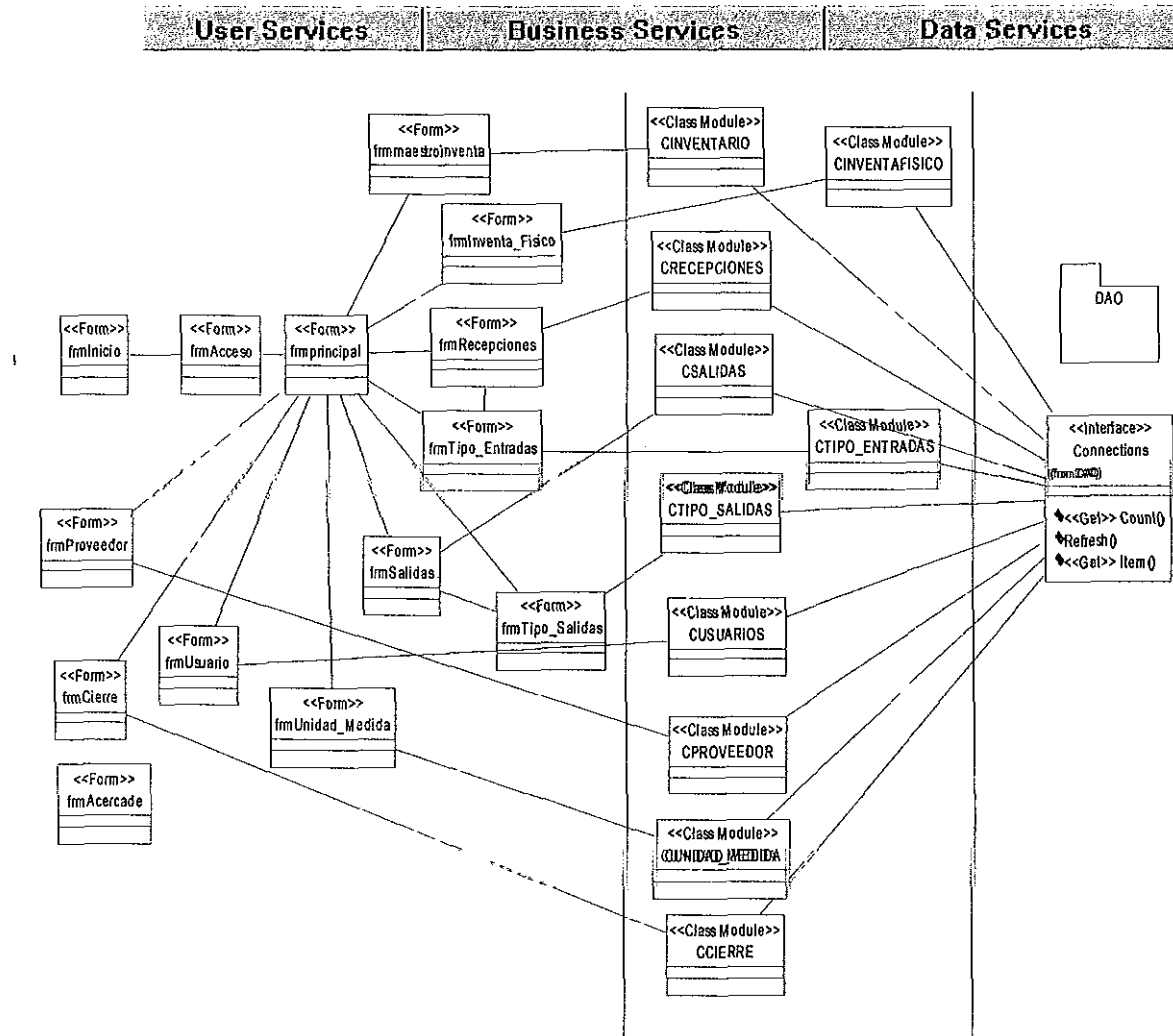
En el diseño lógico es muy importante ya que descompone el modelo de servicios en tres partes principales como son los Servicios de Usuarios, Los Servicios del Negocio y los Servicios de los Datos

En la capa de Servicios del Usuario suele utilizarse para introducir, editar y presentar información y también puede pensarse en ellos como en servicios de interfaz. Estas actividades suponen una gran parte del desarrollo tradicional del Software en las implementaciones Cliente/Servidor

En la capa de Servicios del Negocio suelen ser objetos que realizan una cierta acción, como agregar un producto o procesar el pedido de un cliente. Estos servicios se llaman desde el nivel de servicios de interfaz (Servicios del Usuario)

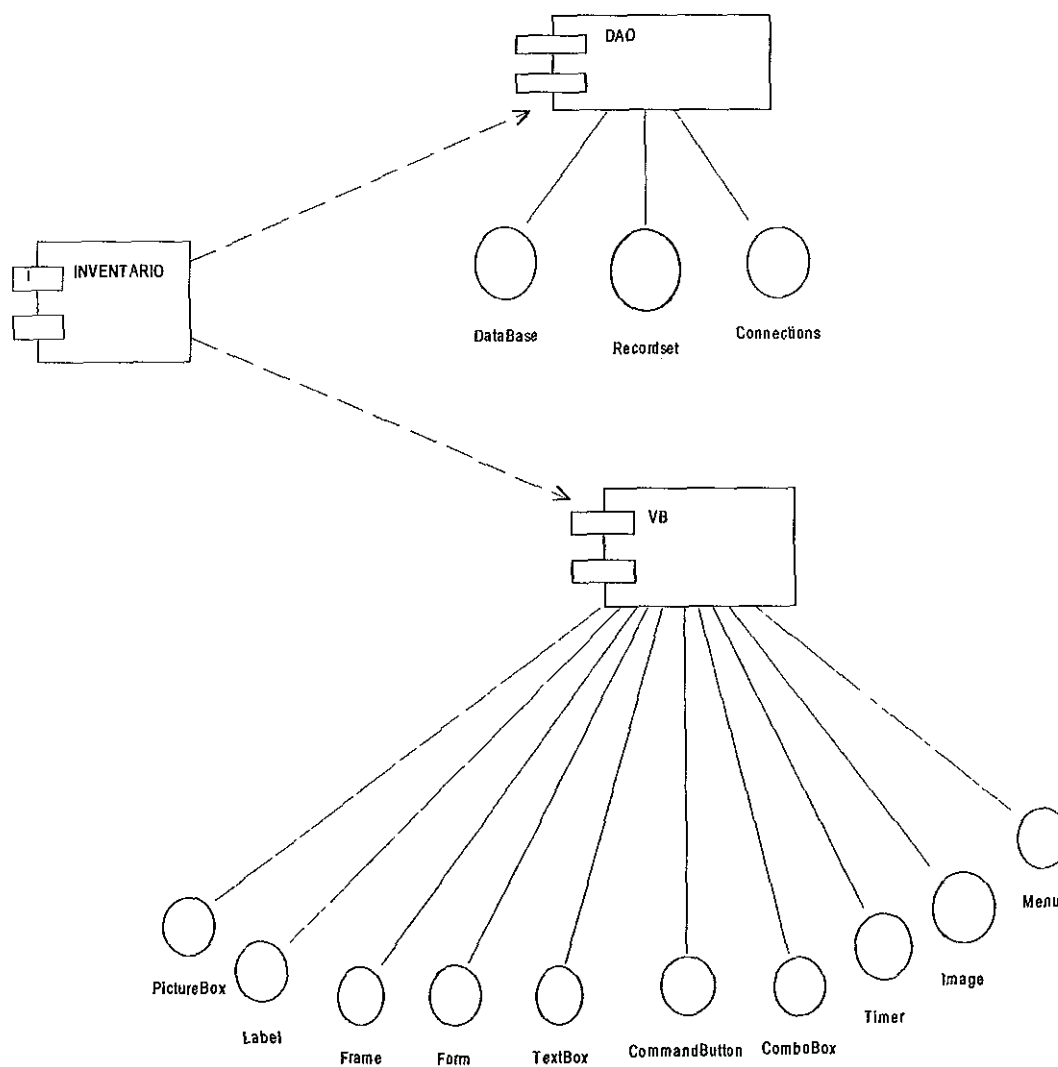
En la capa de Servicios de Datos se encarga de la interfaz con la Base de Datos en un Servidor (SQL SERVER).







2.1.3 DISEÑO FÍSICO (COMPONENTES)





2.1.4 CODIFICACION

CÓDIGO DEL FORMULARIO DE INICIO:

```
'##ModelId=3FD6A7D80032
```

```
Private Sub Timer1_Timer()
```

```
    If Timer1.Interval = 2000 Then
```

```
        frmAcceso.Show
```

```
        Unload Me
```

```
    End If
```

```
End Sub'
```





CÓDIGO DEL FORMULARIO DE ACCESO:

```

'##ModelId=3FD6A7DD017C
Private Sub cmdAceptar_Click()
    If Adodc1 Recordset.RecordCount >= 1 Then
        Adodc1 Recordset MoveFirst
    End If
    Rem BUSQUEDA DE LA CLAVE EN LA TABLA USUARIO
    Do While Not Adodc1 Recordset EOF
        If Adodc1 Recordset EOF Then
            Exit Do
        End If
        If Trim(Adodc1 Recordset Fields("CONTRASEÑA")) = Trim(txtclave Text) Then
            txtUsuario Text = Adodc1 Recordset.Fields("NOMBRE_USUA")
            MsgBox ("CLAVE AUTORIZADA")
            frmprincipal Show
            Exit Sub
        End If
        Adodc1.Recordset MoveNext
    Loop
    MsgBox ("CLAVE NO AUTORIZADA, INVESTIGUE ")
    txtUsuario.Text = ""
    txtclave Text = ""
    txtclave.SetFocus
End Sub

```





```
'##ModelId=3FD6A7DD01B8
```

```
Private Sub cmdCancelar_Click()
```

```
    MsgBox ("HASTA LUEGO")
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub txtclave_KeyPress(KeyAscii As Integer)
```

```
    If (KeyAscii = vbKeyReturn) Then
```

```
        cmdAceptar.SetFocus
```

```
    End If
```

```
    Do While Not Adodc1.Recordset.EOF
```

```
        If Adodc1.Recordset.EOF Then
```

```
            Exit Do
```

```
        End If
```

```
        If Trim(Adodc1.Recordset.Fields("CONTRASEÑA")) = Trim(txtclave.Text) Then
```

```
            txtUsuario.Text = Adodc1.Recordset.Fields("NOMBRE_USUA")
```

```
        Exit Sub
```

```
    End If
```

```
    Adodc1.Recordset.MoveNext
```

```
Loop
```

```
End Sub
```





CÓDIGO DEL FORMULARIO PRINCIPAL:

```
'##ModelId=3FD6A7DF02B2
```

```
Private Sub mnuAcerca_Click()
```

```
    frmprincipal.Enabled = False
```

```
    frmAcercade.Show
```

```
End Sub
```

```
'##ModelId=3FD6A7DF02E4
```

```
Private Sub mnuCierre_Click()
```

```
    frmCierre.Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7DF0352
```

```
Private Sub mnuInventario_Click()
```

```
    frmmaestroinventa.Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7DF038E
```

```
Private Sub mnuInventFisico_Click()
```

```
    frmInventa_Fisico.Show
```

```
    Unload Me
```

```
End Sub
```





```
'##ModelId=3FD6A7E00014
```

```
Private Sub mnuProveedor_Click()
```

```
    frmProveedor Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7E00082
```

```
Private Sub mnuRecepciones_Click()
```

```
    frmRecepciones Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7E000B4
```

```
Private Sub mnuSalidas_Click()
```

```
    frmSalidas Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7E00122
```

```
Private Sub mnusalir_Click()
```

```
    End
```

```
End Sub
```

```
'##ModelId=3FD6A7E00190
```

```
Private Sub mnuTipoRecep_Click()
```

```
    frmTipo_Entradas.Show
```

```
    Unload Me
```

```
End Sub
```





```
'##ModelId=3FD6A7E001FE
```

```
Private Sub mnuTipoSali_Click()
```

```
    frmTipo_Salidas Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7E0026C
```

```
Private Sub mnuUnidadMedida_Click()
```

```
    frmUnidad_Medida Show
```

```
    Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A7E002DA
```

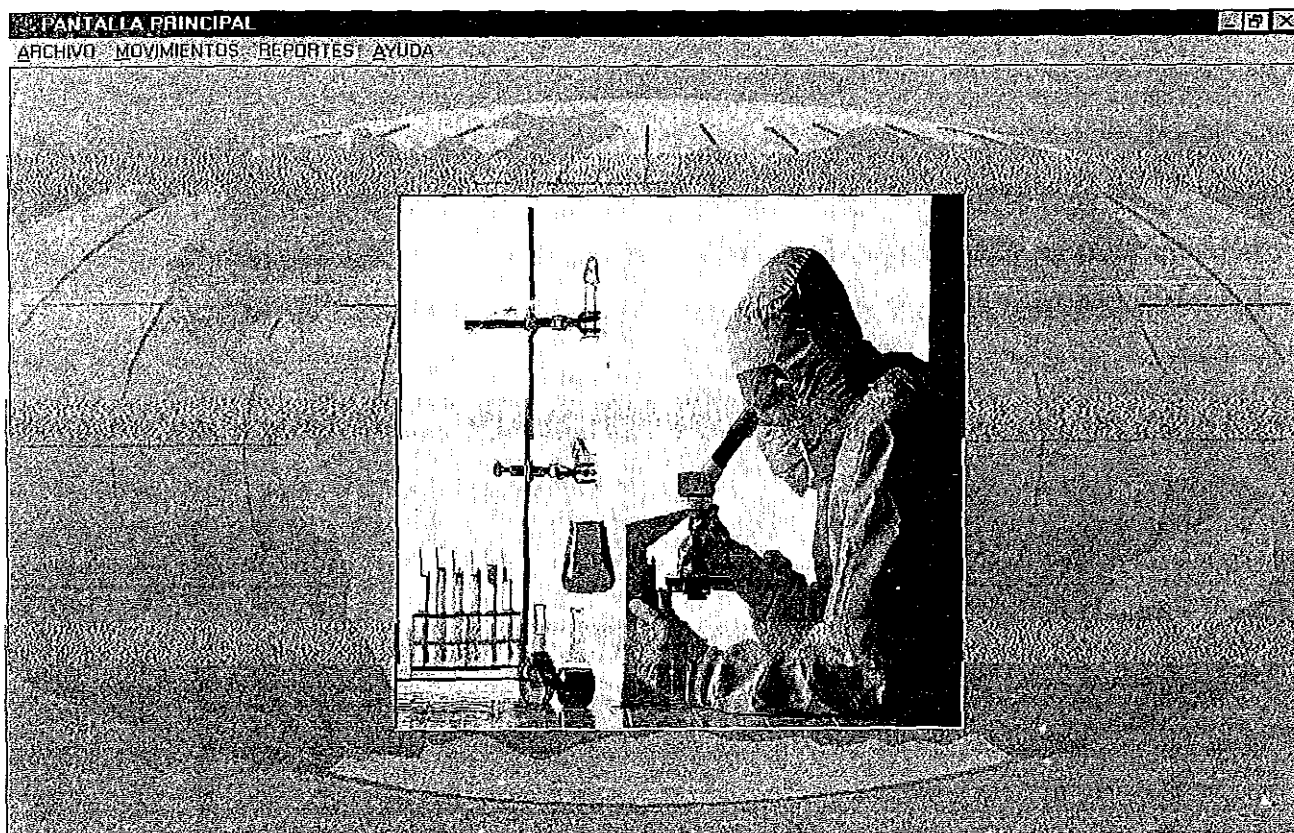
```
Private Sub mnuUsuario_Click()
```

```
    frmUsuario Show
```

```
    Unload Me
```

```
End Sub
```





CÓDIGO DEL FORMULARIO MAESTRO DE INVENTARIO:

Option Explicit

'##ModelId=3FD6A7E403C2

Private CN As Connection

'##ModelId=3FD6A7E50014

Private RS As Recordset

'##ModelId=3FD6A7E50015

Private X As Integer





'##ModelId=3FD6A7E500B4

Private Sub cmdActualizar_Click()

Dim PRODUCTO2 As CINVENTARIO

Set PRODUCTO2 = New CINVENTARIO

Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE

PRODUCTO2 COD_PROD = RTrim(txtCodigo Text)

PRODUCTO2 DESCRIPCION = txtDescripcion Text

PRODUCTO2.UM = txtUm.Text

PRODUCTO2 PRECIO = Val(txtPrecio Text)

PRODUCTO2.COD_PROV = txtCod_Prov Text

PRODUCTO2 EXISTENCIA = txtExistencia Text

PRODUCTO2 NIVEL_MINIMO = txtNivelMinimo Text

PRODUCTO2.MODIFICAR

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1 Visible = False

txtCodigo Visible = False

txtDescripcion Visible = False

txtUm Visible = False

txtPrecio Visible = False

txtCod_Prov Visible = False

txtExistencia.Visible = False

txtNivelMinimo Visible = False

lblCodigo.Visible = False

lblDescripcion Visible = False

lblUm.Visible = False

lblPrecio.Visible = False

lblProveedor.Visible = False





```

lblExistencia.Visible = False
lblNivelMini.Visible = False
lblModificar.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False

```

```
RS.MoveLast
```

```
DataGrid1.Refresh
```

```
MsgBox ("REGISTRO ACTUALIZADO EN MAESTRO DE INVENTARIO")
```

```
End Sub
```

```
'##ModelId=3FD6A7E50122
```

```
Private Sub cmdAgregar_Click()
```

```
Rem BOTON AGREGAR
```

```

txtCodigo.Visible = True
txtCodigo.Enabled = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtPrecio.Visible = True
txtCod_Prov.Visible = True
txtNivelMinimo.Visible = True
lblCodigo.Visible = True
lblDescripcion.Visible = True
lblUm.Visible = True
lblPrecio.Visible = True
lblProveedor.Visible = True
lblNivelMini.Visible = True

```





```

IbLetiqueta Visible = True
IbEncontrado Visible = False
cmdGuardar Visible = True
cmdCancelar Visible = True
cmdModificar Enabled = False
cmdEliminar Enabled = False
Frame1 Visible = True
txtCodigo SetFocus
Rem INICIALIZAR TEXTOS
txtCodigo Text = " "
txtDescripcion Text = " "
txtUm Text = " "
txtPrecio Text = " "
txtCod_Prov.Text = " "
txtNivelMinimo Text = " "
End Sub

```

```

'##ModelId=3FD6A7E50190
Private Sub cmdAnterior_Click()
    Rem ESTA EN EL INICIO
    Adodc1.Recordset MovePrevious
    DataGrid1 Refresh
    cmdSiguiente Enabled = True
    cmdFinal.Enabled = True
    If Adodc1.Recordset BOF Then
        Adodc1.Recordset MoveFirst
        cmdAnterior Enabled = False
    End If
End Sub

```





cmdInicio Enabled = False

End If

End Sub

'##ModelId=3FD6A7E502DA

Private Sub cmdBuscar_Click()

Rem BUSCAR UN PRODUCTO POR MEDIO DEL CODIGO

Text1.Visible = True

cmdEjecutar.Visible = True

Text1.SetFocus

End Sub

'##ModelId=3FD6A7E50384

Private Sub cmdCancelar_Click()

Rem BOTON CANCELAR

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1.Visible = False

txtCodigo.Visible = False

txtDescripcion.Visible = False

txtUm.Visible = False

txtPrecio.Visible = False

txtCod_Prov.Visible = False

txtNivelMinimo.Visible = False

lblCodigo.Visible = False

lblDescripcion.Visible = False

lblUm.Visible = False

lblPrecio.Visible = False





```

lblProveedor.Visible = False
lblNivelMini.Visible = False
lblEtiqueta.Visible = False
lblModificar.Visible = False
lblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
End Sub

'##ModelId=3FD6A7E6003C
Private Sub cmdEliminar_Click()
    Dim PRODUCTO3 As CINVENTARIO
    Set PRODUCTO3 = New CINVENTARIO
    Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE
    PRODUCTO3.COD_PROD = RTrim(txtCodigo.Text)
    PRODUCTO3.DESCRIPCION = txtDescripcion.Text
    PRODUCTO3.UM = txtUm.Text
    PRODUCTO3.PRECIO = Val(txtPrecio.Text)
    PRODUCTO3.COD_PROV = txtCod_Prov.Text
    PRODUCTO3.NIVEL_MINIMO = txtNivelMinimo.Text
    PRODUCTO3.ELIMINAR
    lblEliminar.Visible = True
    Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
    Frame1.Visible = False
    txtCodigo.Visible = False

```





txtDescripcion.Visible = False

txtUm.Visible = False

txtPrecio.Visible = False

txtCod_Prov.Visible = False

txtNivelMinimo.Visible = False

lblCodigo.Visible = False

lblDescripcion.Visible = False

lblUm.Visible = False

lblPrecio.Visible = False

lblProveedor.Visible = False

lblNivelMini.Visible = False

lblEliminar.Visible = False

lblEncontrado.Visible = False

cmdGuardar.Visible = False

cmdCancelar.Visible = False

cmdActualizar.Visible = False

If Adodc1.Recordset.RecordCount = -1 Then

Else

Rem Adodc1.Recordset.Update

Rem RS.MoveLast

DataGrid1.Refresh

End If

MsgBox ("REGISTRO ELIMINADO EN MAESTRO DE INVENTARIO")

End Sub





```
'##ModelId=3FD6A7E600AA
```

```
Private Sub cmdFinal_Click()
```

```
    Rem ESTA EN EL INICIO
```

```
    Adodc1 Recordset MoveLast
```

```
    DataGrid1.Refresh
```

```
    cmdSiguiente.Enabled = False
```

```
    cmdFinal.Enabled = False
```

```
    cmdAnterior.Enabled = True
```

```
    cmdInicio.Enabled = True
```

```
End Sub
```

```
'##ModelId=3FD6A7E60118
```

```
Private Sub cmdGuardar_Click()
```

```
    Dim PRODUCTO As CINVENTARIO
```

```
    Set PRODUCTO = New CINVENTARIO
```

```
    PRODUCTO.COD_PROD = LTrim(txtCodigo.Text)
```

```
    PRODUCTO.DESCRIPCION = txtDescripcion.Text
```

```
    PRODUCTO.UM = txtUm.Text
```

```
    PRODUCTO.PRECIO = Val(txtPrecio.Text)
```

```
    PRODUCTO.COD_PROV = txtCod_Prov.Text
```

```
    PRODUCTO.NIVEL_MINIMO = Val(txtNivelMinimo.Text)
```

```
    PRODUCTO.GUARDAR
```

```
    Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
```

```
    Frame1.Visible = False
```

```
    txtCodigo.Visible = False
```

```
    txtDescripcion.Visible = False
```

```
    txtUm.Visible = False
```





```

txtPrecio.Visible = False
txtCod_Prov.Visible = False
txtNivelMinimo.Visible = False
lblEtiqueta.Visible = False
lblCodigo.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblPrecio.Visible = False
lblProveedor.Visible = False
lblNivelMini.Visible = False
lblEtiqueta.Visible = False
cmdGuardar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
    Else
        Rem Adodc1.Recordset.Update
        DataGrid1.Refresh
    End If

```

MsgBox ("REGISTRO ALMACENADO EN MAESTRO DE INVENTARIO")

End Sub

'##ModelId=3FD6A7E601B8

Private Sub cmdInicio_Click()

```

    Rem ESTA EN EL INICIO
    Adodc1.Recordset.MoveFirst
    DataGrid1.Refresh
    cmdAnterior.Enabled = False
    cmdInicio.Enabled = False

```





```
cmdSiguiente.Enabled = True
End Sub
```

```
'''ModelId=3FD6A7E60262
Private Sub cmdModificar_Click()
    Rem MODIFICAR UN REGISTRO
    cmdActualizar.Visible = True
    lblModificar.Visible = True
    lblEncontrado.Visible = False
    lblEtiqueta.Visible = False
    txtDescripcion.SetFocus
End Sub
```

```
'''ModelId=3FD6A7E602D0
Private Sub cmdRetornar_Click()
    frmprincipal.Show
    Unload Me
End Sub
```

```
'''ModelId=3FD6A7E60370
Private Sub cmdEjecutar_Click()
    Dim PRODUCTO1 As CINVENTARIO
    Set PRODUCTO1 = New CINVENTARIO
    Dim MBUS As String
    MBUS = RTrim(Text1.Text)
    PRODUCTO1.BUSCAR (MBUS)
    Rem If Trim(PRODUCTO1.COD_PROD) = Trim(Text1.Text) Then
```





```

MsgBox ("CODIGO NO EXISTE")
txtCodigo.Visible = True
txtCodigo.Text = PRODUCTO1.COD_PROD
txtDescripcion.Text = PRODUCTO1.DESCRIPCION
txtUm.Text = PRODUCTO1.UM
txtPrecio.Text = PRODUCTO1.PRECIO
txtCod_Prov.Text = PRODUCTO1.COD_PROV
txtNivelMinimo.Text = PRODUCTO1.NIVEL_MINIMO
Rem HACER VISIBLE LOS CAMPOS
lblEncontrado.Visible = True
txtCodigo.Visible = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtPrecio.Visible = True
txtCod_Prov.Visible = True
Rem txtExistencia.Visible = True
txtNivelMinimo.Visible = True
lblCodigo.Visible = True
lblDescripcion.Visible = True
lblUm.Visible = True
lblPrecio.Visible = True
lblProveedor.Visible = True
lblNivelMini.Visible = True
cmdCancelar.Visible = True
Frame1.Visible = True
cmdModificar.Enabled = True
cmdEliminar.Enabled = True
    
```





```

Text1.Visible = False
cmdEjecutar.Visible = False
Text1.Text = " "
Frame1.Visible = True
Rem End If
End Sub

'##ModelId=3FD6A7E603DE
Private Sub cmdSiguiente_Click()
    Rem SE CORRE AL SIGUIENTE REGISTRO
    Adodc1.Recordset.MoveNext
    DataGrid1.Refresh
    cmdAnterior.Enabled = True
    cmdInicio.Enabled = True
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
        cmdFinal.Enabled = False
        cmdSiguiente.Enabled = False
    End If
End Sub

'##ModelId=3FD6A7E700A0
Private Sub Form_Load()
    Set CN = New Connection
    Set RS = New Recordset
    CN.CursorLocation = adUseClient

```





```
CN.ConnectionString = "Provider=MSDASQL 1,Persist Security Info=False,Data
Source=BD_INVENTARIO"
```

```
CN.Open
```

```
Set RS.ActiveConnection = CN
```

```
RS.CursorType = adOpenForwardOnly
```

```
RS.Open "select * from inventario"
```

```
End Sub
```

```
###ModelId=3FD6A7E70140
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    If (KeyAscii = vbKeyReturn) Then
```

```
        cmdEjecutar.SetFocus
```

```
    End If
```

```
End Sub
```

```
###ModelId=3FD6A7E70258
```

```
Private Sub txtCod_Prov_KeyPress(KeyAscii As Integer)
```

```
    If (KeyAscii = vbKeyReturn) Then
```

```
        txtNivelMinimo.SetFocus
```

```
    End If
```

```
End Sub
```

```
###ModelId=3FD6A7E703A2
```

```
Private Sub txtCodigo_KeyPress(KeyAscii As Integer)
```

```
    If (KeyAscii = vbKeyReturn) Then
```

```
        If Adodc1.Recordset.RecordCount >= 1 Then
```

```
            Adodc1.Recordset.MoveFirst
```

```
        End If
```





```

Do While Not Adodc1.Recordset.EOF
    If Adodc1.Recordset.EOF Then
        Exit Do ' si no lo encuentra continua
    End If
    If Trim(Adodc1.Recordset.Fields("COD_PROD")) = Trim(txtCodigo.Text) Then
        MsgBox ("Registro existe ")
        txtCodigo = ""
        txtCodigo.SetFocus
    End If
    Exit Sub
End If
Adodc1.Recordset.MoveNext
Loop
txtDescripcion.SetFocus
End If
End Sub

'##ModelId=3FD6A7E80104
Private Sub txtDescripcion_KeyPress(KeyAscii As Integer)
    If (KeyAscii = vbKeyReturn) Then
        txtUm.SetFocus
    End If
End Sub

'##ModelId=3FD6A7E80212
Private Sub txtPrecio_KeyPress(KeyAscii As Integer)
    If (KeyAscii = vbKeyReturn) Then
        txtCod_Prov.SetFocus
    End If
End Sub

```





```

End If
End Sub

'##ModelId=3FD6A7E90208
Private Sub txtUm_KeyPress(KeyAscii As Integer)
    If (KeyAscii = vbKeyReturn) Then
        txtPrecio.SetFocus
    End If
End Sub

```

Este formulario manda a llamar a una clase llamada CINVENTARIO y su código es el siguiente

```

'variables locales para almacenar los valores de las propiedades
'##ModelId=3FD6A7EC02F8
Private mvarCOD_PROD As String 'copia local
'##ModelId=3FD6A7EC0366
Private mvarDESCRIPCION As String 'copia local
'##ModelId=3FD6A7ED0028
Private mvarUM As String 'copia local
'##ModelId=3FD6A7ED00C8
Private mvarCOD_PROV As String 'copia local
'##ModelId=3FD6A7ED01A4
Private mvarENTRADAS As Double 'copia local
'##ModelId=3FD6A7ED0244
Private mvarSALIDAS As Double 'copia local
'##ModelId=3FD6A7ED02EE

```





```

Private mvarEXISTENCIA As Double 'copia local
'###ModelId=3FD6A7ED03CA
Private mvarNIVEL_MINIMO As Double 'copia local
'###ModelId=3FD6A7EE0082
Private mvarINVEN_FISICO As Double 'copia local
'###ModelId=3FD6A7EE015E
Private mvarPRECIO As Double 'copia local
'###ModelId=3FD6A7EE023C
Private CN As Connection
'###ModelId=3FD6A7EE0276
Private RS As Recordset
'variables locales para almacenar los valores de las propiedades

'###ModelId=3FD6A7EE0277
Private mvarINVEN_INI As Double 'copia local
'###ModelId=3FD6A7EE0352
Private mvarFECHA_FISI As Date 'copia local
'###ModelId=3FD6A7EF0046
Private mvarMES As String 'copia local
'###ModelId=3FD6A7EF0154
Public Property Let MES(ByVal vData As String)
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
'Syntax. X MES = 5
    mvarMES = vData
End Property

'###ModelId=3FD6A7EF02DA
    
```





Public Property Get MES() As String

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax Debug Print X MES

MES = mvarMES

End Property

'##ModelId=3FD6A7F00000

Public Property Let FECHA_FISI(ByVal vData As Date)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X.FECHA_FISI = 5

mvarFECHA_FISI = vData

End Property

'##ModelId=3FD6A7F001B8

Public Property Get FECHA_FISI() As Date

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax Debug Print X FECHA_FISI

FECHA_FISI = mvarFECHA_FISI

End Property

'##ModelId=3FD6A7F00294

Public Property Let INVEN_INI(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X.INVEN_INI = 5

mvarINVEN_INI = vData

End Property





```
'##ModelId=3FD6A7F101E0
```

```
Public Property Get INVEN_INI() As Double
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax. Debug Print X INVEN_INI
```

```
    INVEN_INI = mvarINVEN_INI
```

```
End Property
```

```
'##ModelId=3FD6A7F1032A
```

```
Public Sub MODIFICAR()
```

```
Set RS.ActiveConnection = CN
```

```
RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
```

```
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
```

```
RS.Open "SELECT * FROM INVENTARIO"
```

```
RS.MoveFirst
```

```
Do While Not RS.EOF
```

```
If RS.EOF Then
```

```
    Exit Sub
```

```
Else
```

```
    If RTrim(RS.Fields("COD_PROD")) = COD_PROD Then
```

```
        Rem MsgBox ("LO ENCONTRE")
```

```
        RS.Fields("COD_PROD") = COD_PROD
```

```
        RS.Fields("DESCRIPCION") = DESCRIPCION
```

```
        RS.Fields("UM") = UM
```

```
        RS.Fields("PRECIO") = PRECIO
```

```
        RS.Fields("COD_PROV") = COD_PROV
```

```
        RS.Fields("NIVEL_MINIMO") = NIVEL_MINIMO
```

```
        RS.Update
```





```

Exit Sub
Rem Else
Rem   MsgBox ("CODIGO = ") & RS Fields("COD_PROD")
End If
RS MoveNext
End If
Loop
End Sub

'
'##ModelId=3FD6A7F103D4
Public Sub GUARDAR()
Set RS ActiveConnection = CN
RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS.Open "SELECT * FROM INVENTARIO"
RS.AddNew
RS.Fields("COD_PROD") = COD_PROD
RS.Fields("DESCRIPCION") = DESCRIPCION
RS.Fields("UM") = UM
RS.Fields("PRECIO") = PRECIO
RS.Fields("COD_PROV") = COD_PROV
RS.Fields("EXISTENCIA") = 0
RS.Fields("NIVEL_MINIMO") = NIVEL_MINIMO
RS.Fields("ENTRADAS") = 0
RS.Fields("SALIDAS") = 0
RS.Fields("INVE_INI") = 0
RS.Fields("INVEN_FISICO") = 0

```





RS Fields("FECHA_FISI") = #1/1/2001#

RS Fields("MES") = " "

RS Update

End Sub

'##ModelId=3FD6A7F2008C

Public Sub CLASS_INITIALIZE()

Set CN = New Connection

Set RS = New Recordset

CN.ConnectionString = "Provider=MSDASQL 1,Persist Security Info=False;Data Source=BD_INVENTARIO"

CN.Open

End Sub

'##ModelId=3FD6A7F20168

Public Sub ELIMINAR()

Set RS.ActiveConnection = CN

RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR

RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR

RS.Open "SELECT * FROM INVENTARIO"

RS.MoveFirst

Do While Not RS.EOF

If RS.EOF Then

Exit Sub

Else

If RTrim(RS.Fields("COD_PROD")) = COD_PROD Then

Rem MsgBox ("LO ENCONTRE")





```

    RS Delete
    Exit Sub
    Rem Else
    Rem   MsgBox ("CODIGO = ") & RS Fields("COD_PROD")
    End If
    RS MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A7F20212
Public Sub BUSCAR(MBUSCA As String)
    Dim MBUSCAR As String
    MBUSCAR = MBUSCA
    Set RS.ActiveConnection = CN
    RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
    RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
    RS.Open "SELECT * FROM INVENTARIO order by cod_prod "
    Rem MsgBox (" 1 MBUSCAR = ") & MBUSCAR
    RS.MoveFirst
    Do While Not RS.EOF
        If RS.EOF Then
            MsgBox ("REGISTRO NO EXISTE")
            Exit Sub
        Else
            If RTrim(RS.Fields("COD_PROD")) = MBUSCAR Then
                Rem MsgBox ("LO ENCONTRE")
            End If
        End If
    Loop
End Sub

```





```

COD_PROD = RS Fields("COD_PROD")
DESCRIPCION = RS Fields("DESCRIPCION")
UM = RS Fields("UM")
PRECIO = RS Fields("PRECIO")
Rem EXISTENCIA = RS Fields("EXISTENCIA")
COD_PROV = RS.Fields("COD_PROV")
NIVEL_MINIMO = RS Fields("NIVEL_MINIMO")
Exit Sub
Else
    Rem MsgBox ("CODIGO = ") & RS.Fields("COD_PROD")
End If
RS.MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A7F2038E
Public Property Let PRECIO(ByVal vData As Double)
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
'Syntax. X PRECIO = 5
    mvarPRECIO = vData
End Property

'##ModelId=3FD6A7F30208
Public Property Get PRECIO() As Double
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
'Syntax. Debug.Print X PRECIO

```





PRECIO = mvarPRECIO

End Property

'##ModelId=3FD6A7F30352

Public Property Let INVEN_FISICO(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X INVEN_FISICO = 5

mvarINVEN_FISICO = vData

End Property

'##ModelId=3FD6A7F40190

Public Property Get INVEN_FISICO() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax: Debug Print X INVEN_FISICO

INVEN_FISICO = mvarINVEN_FISICO

End Property

'##ModelId=3FD6A7F402DA

Public Property Let NIVEL_MINIMO(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X NIVEL_MINIMO = 5

mvarNIVEL_MINIMO = vData

End Property





```
'##ModelId=3FD6A7F5014A
```

```
Public Property Get NIVEL_MINIMO() As Double
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax Debug.Print X NIVEL_MINIMO
```

```
    NIVEL_MINIMO = mvarNIVEL_MINIMO
```

```
End Property
```

```
'##ModelId=3FD6A7F50294
```

```
Public Property Let EXISTENCIA(ByVal vData As Double)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax. X EXISTENCIA = 5
```

```
    mvarEXISTENCIA = vData
```

```
End Property
```

```
'##ModelId=3FD6A7F60366
```

```
Public Property Get EXISTENCIA() As Double
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax Debug.Print X.EXISTENCIA
```

```
    EXISTENCIA = mvarEXISTENCIA
```

```
End Property
```

```
'##ModelId=3FD6A7F700C8
```

```
Public Property Let SALIDAS(ByVal vData As Double)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax X SALIDAS = 5
```

```
    mvarSALIDAS = vData
```

```
End Property
```





```
###ModelId=3FD6A7F7035C
```

```
Public Property Get SALIDAS() As Double
```

```
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
```

```
'Syntax Debug Print X.SALIDAS
```

```
    SALIDAS = mvarSALIDAS
```

```
End Property
```

```
###ModelId=3FD6A7F800BE
```

```
Public Property Let ENTRADAS(ByVal vData As Double)
```

```
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
```

```
'Syntax. X ENTRADAS = 5
```

```
    mvarENTRADAS = vData
```

```
End Property
```

```
###ModelId=3FD6A7F80348
```

```
Public Property Get ENTRADAS() As Double
```

```
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.
```

```
'Syntax Debug Print X ENTRADAS
```

```
    ENTRADAS = mvarENTRADAS
```

```
End Property
```

```
###ModelId=3FD6A7F900E6
```

```
Public Property Let COD_PROV(ByVal vData As String)
```

```
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.
```

```
'Syntax: X.COD_PROV = 5
```

```
    mvarCOD_PROV = vData
```

```
End Property
```





'##ModelId=3FD6A7F9037A

Public Property Get COD_PROV() As String

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax Debug Print X COD_PROV

COD_PROV = mvarCOD_PROV

End Property

'##ModelId=3FD6A7FA010E

Public Property Let UM(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.

'Syntax X UM = 5

mvarUM = vData

End Property

'##ModelId=3FD6A7FA03DE

Public Property Get UM() As String

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax Debug Print X UM

UM = mvarUM

End Property

'##ModelId=3FD6A7FB0172

Public Property Let DESCRIPCION(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X DESCRIPCION = 5

mvarDESCRIPCION = vData

End Property





```
'##ModelId=3FD6A7FC005A
```

```
Public Property Get DESCRIPCION() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

```
'Syntax. Debug Print X DESCRIPCION
```

```
    DESCRIPCION = mvarDESCRIPCION
```

```
End Property
```

```
'##ModelId=3FD6A7FC0208
```

```
Public Property Let COD_PROD(ByVal vData As String)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax X COD_PROD = 5
```

```
    mvarCOD_PROD = vData
```

```
End Property
```

```
'##ModelId=3FD6A7FD0122
```

```
Public Property Get COD_PROD() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax Debug Print X COD_PROD
```

```
    COD_PROD = mvarCOD_PROD
```

```
End Property
```





MAESTRO DE INVENTARIO

MAESTRO DE INVENTARIO					
COD PROD	DESCRIPCION	UM	PRECIO	EXISTENCIA	ENTRA
01	Aspirina	Cap	12	268	300
02	Acetaminofen	Tab	5.5	0	0

AGREGAR
MODIFICAR
ELIMINAR
INICIO
SIGUIENTE
ANTERIOR
FINAL
BUSCAR
RETORNAR

AGREGAR PRODUCTOS

CODIGO DEL PRODUCTO:
DESCRIPCION:
UNIDAD DE MEDIDA:

PRECIO:
CODIGO DEL PROVEEDOR:
NIVEL MINIMO:

GUARDAR
CANCELAR





CÓDIGO DEL FORMULARIO INVENTARIO FISICO:

Option Explicit

'##ModelId=3FD6A82A014C

Private CN As Connection

'##ModelId=3FD6A82A014F

Private RS As Recordset

'##ModelId=3FD6A82A0186

Private X As Integer

'##ModelId=3FD6A82A01F4

Private Sub cmdActualizar_Click()

Dim PRODUCTO2 As CINVENTAFISICO

Set PRODUCTO2 = New CINVENTAFISICO

PRODUCTO2 COD_PROD = RTrim(txtCodigo.Text)

PRODUCTO2 INVEN_FISICO = txtInven_Fisico.Text

PRODUCTO2.FECHA_FISI = txtFecha_Fisica.Text

PRODUCTO2 MES = txtMes.Text

PRODUCTO2 MODIFICAR

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1.Visible = False

txtCodigo.Visible = False

txtDescripcion.Visible = False

txtUm.Visible = False

txtInven_Fisico.Visible = False

txtFecha_Fisica.Visible = False

lblCodigo.Visible = False





```
lblDescripcion.Visible = False
lblUm.Visible = False
lblInven_Fisico.Visible = False
lblFecha_Fisica.Visible = False
lblModificar.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
RS.MoveLast
DataGrid1.Refresh
MsgBox ("REGISTRO ACTUALIZADO EN MAESTRO DE INVENTARIO")
End Sub
```

```
'##ModelId=3FD6A82A0262
Private Sub cmdAgregar_Click()
Rem BOTON AGREGAR
txtCodigo.Visible = True
txtCodigo.Enabled = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtInven_Fisico.Visible = True
txtFecha_Fisica.Visible = True
txtMes.Visible = True
lblCodigo.Visible = True
lblDescripción.Visible = True
lblUm.Visible = True
lblInven_Fisico.Visible = True
```





```

lblFecha_Fisica.Visible = True
lblMes.Visible = True
lblEtiqueta.Visible = True
lblEncontrado.Visible = False
cmdGuardar.Visible = True
cmdCancelar.Visible = True
cmdModificar.Enabled = False
cmdEliminar.Enabled = False
Frame1.Visible = True
txtCodigo.SetFocus
Rem INICIALIZAR TEXTOS
txtCodigo.Text = " "
txtDescripcion.Text = " "
txtUm.Text = " "
txtInven_Fisico.Text = " "
txtFecha_Fisica.Text = " "
txtMes.Text = " "
End Sub

```

```

'##ModelId=3FD6A82A02D0
Private Sub cmdAnterior_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MovePrevious
DataGrid1.Refresh
cmdSiguiente.Enabled = True
cmdFinal.Enabled = True

```





```
If Adodc1 Recordset BOF Then
    Adodc1 Recordset MoveFirst
    cmdAnterior.Enabled = False
    cmdInicio.Enabled = False
End If
End Sub
```

```
###ModelId=3FD6A82A033E
Private Sub cmdBuscar_Click()
    Rem BUSCAR UN PRODUCTO POR MEDIO DEL CODIGO
    Text1.Visible = True
    cmdEjecutar.Visible = True
    Text1.SetFocus
End Sub
```

```
###ModelId=3FD6A82A03AC
Private Sub cmdCancelar_Click()
    Rem BOTON CANCELAR
    Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
    Frame1.Visible = False
    txtCodigo.Visible = False
    txtDescripcion.Visible = False
    txtUm.Visible = False
    txtInven_Fisico.Visible = False
    txtFecha_Fisica.Visible = False
    txtMes.Visible = False
    lblCodigo.Visible = False
```





```
lblDescripcion.Visible = False
lblUm Visible = False
lblInven_Fisico Visible = False
lblFecha_Fisica Visible = False
lblMes Visible = False
lblEtiqueta Visible = False
lblModificar Visible = False
lblEncontrado Visible = False
cmdGuardar Visible = False
cmdCancelar Visible = False
cmdActualizar Visible = False
cmdAgregar Enabled = True
cmdInicio Enabled = True
cmdSiguiente Enabled = True
cmdAnterior Enabled = True
cmdFinal Enabled = True
cmdBuscar Enabled = True
cmdModificar Enabled = False
End Sub
```

```
'##ModelId=3FD6A82B0064
Private Sub cmdEjecutar_Click()
Dim PRODUCTO1 As CINVENTAFISICO
Set PRODUCTO1 = New CINVENTAFISICO
Dim MBUS As String
MBUS = RTrim(Text1.Text)
PRODUCTO1.BUSCAR (MBUS)
```





txtCodigo.Visible = True
txtCodigo.Text = PRODUCTO1.COD_PROD
txtDescripcion.Text = PRODUCTO1.DESCRIPCION
txtUm.Text = PRODUCTO1.UM
txtInven_Fisico.Text = PRODUCTO1.INVEN_FISICO
txtFecha_Fisica.Text = PRODUCTO1.FECHA_FISI
txtMes.Text = PRODUCTO1.MES

Rem HACER VISIBLE LOS CAMPOS

lblEncontrado.Visible = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtInven_Fisico.Visible = True
txtFecha_Fisica.Visible = True
txtMes.Visible = True

lblCodigo.Visible = True
lblDescripcion.Visible = True
lblUm.Visible = True
lblInven_Fisico.Visible = True
lblFecha_Fisica.Visible = True
lblMes.Visible = True

cmdCancelar.Visible = True
Rem cmdActualizar.Visible = True

Frame1.Visible = True
cmdModificar.Enabled = True

Text1.Visible = False
cmdEjecutar.Visible = False
cmdAgregar.Enabled = False





```
cmdInicio Enabled = False
cmdSiguiente Enabled = False
cmdAnterior Enabled = False
cmdFinal Enabled = False
cmdBuscar.Enabled = False
Text1.Text = " "
Rem txtDescripcion SetFocus
Frame1 Visible = True
End Sub
```

```
'##ModelId=3FD6A82B00D2
Private Sub cmdEliminar_Click()
Dim PRODUCTO3 As CINVENTAFISICO
Set PRODUCTO3 = New CINVENTAFISICO
Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE
PRODUCTO3 COD_PROD = RTrim(txtCodigo Text)
PRODUCTO3 DESCRIPCION = txtDescripcion Text
PRODUCTO3 UM = txtUm.Text
PRODUCTO3 INVEN_FISICO = Val(txtInven_Fisico Text)
PRODUCTO3 FECHA_FISI = txtFecha_Fisica.Text
PRODUCTO3.MES = txtMes Text
PRODUCTO3 ELIMINAR
lblEliminar Visible = True
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1 Visible = False
txtCodigo Visible = False
txtDescripcion.Visible = False
```





```

txtUm.Visible = False
txtInven_Fisico.Visible = False
txtFecha_Fisica.Visible = False
lblCodigo.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblInven_Fisico.Visible = False
lblFecha_Fisica.Visible = False
lblEliminar.Visible = False
lblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
    Else
DataGrid1.Refresh
    End If
    MsgBox ("REGISTRO ELIMINADO EN MAESTRO DE INVENTARIO")
End Sub
'##ModelId=3FD6A82B0140
Private Sub cmdFinal_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MoveLast
DataGrid1.Refresh
cmdSiguiente.Enabled = False
cmdFinal.Enabled = False
cmdAnterior.Enabled = True

```





cmdInicio Enabled = True

End Sub

'##ModelId=3FD6A82B01AE

Private Sub cmdGuardar_Click()

Dim PRODUCTO As CINVENTAFISICO

Set PRODUCTO = New CINVENTAFISICO

PRODUCTO COD_PROD = RTrim(txtCodigo Text)

PRODUCTO INVEN_FISICO = Val(txtInven_Fisico.Text)

PRODUCTO FECHA_FISI = txtFecha_Fisica Text

PRODUCTO MES = txtMes Text

PRODUCTO GUARDAR

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1 Visible = False

txtCodigo Visible = False

txtDescripcion Visible = False

txtUm.Visible = False

txtInven_Fisico Visible = False

txtFecha_Fisica Visible = False

txtMes Visible = False

lblEtiqueta Visible = False

lblCodigo Visible = False

lblDescripcion Visible = False

lblUm Visible = False

lblInven_Fisico.Visible = False

lblFecha_Fisica.Visible = False

lblMes Visible = False





```

Ibletiqueta.Visible = False
cmdGuardar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
DataGrid1.Refresh
End If
MsgBox ("CANTIDAD GRABADO EN INVENTARIO FISICO")
End Sub

```

```

'##ModelId=3FD6A82B0258
Private Sub cmdInicio_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MoveFirst
DataGrid1.Refresh
cmdAnterior.Enabled = False
cmdInicio.Enabled = False
cmdSiguiente.Enabled = True
End Sub

```

```

'##ModelId=3FD6A82B02C6
Private Sub cmdModificar_Click()
Rem MODIFICAR UN REGISTRO
cmdActualizar.Visible = True
lblModificar.Visible = True
lblEncontrado.Visible = False
Ibletiqueta.Visible = False
txtDescripcion.SetFocus

```





End Sub

'##ModelId=3FD6A82B0334

Private Sub cmdRetornar_Click()

frmprincipal Show

Unload Me

End Sub

'##ModelId=3FD6A82B03D4

Private Sub cmdSiguiente_Click()

Rem SE CORRE AL SIGUIENTE REGISTRO

Adodc1.Recordset MoveNext

DataGrid1.Refresh

cmdAnterior.Enabled = True

cmdInicio.Enabled = True

If Adodc1.Recordset.EOF Then

Adodc1.Recordset MoveLast

cmdFinal.Enabled = False

cmdSiguiente.Enabled = False

End If

End Sub

'##ModelId=3FD6A82C008C

Private Sub Form_Load()

Set CN = New Connection

Set RS = New Recordset

CN.CursorLocation = adUseClient





```
CN.ConnectionString = "Provider=MSDASQL 1,Persist Security Info=False,Data
Source=BD_INVENTARIO"
```

```
CN.Open
```

```
Set RS.ActiveConnection = CN
```

```
RS.CursorType = adOpenForwardOnly
```

```
RS.Open "select * from INVENTARIO"
```

```
End Sub
```

```
'##ModelId=3FD6A82C0136
```

```
Private Sub txtCantidad_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    If txtCantidad.Text = "" Then
```

```
        MsgBox "CANTIDAD DE PRODUCTO EN BLANCO FAVOR REVISE", vbCritical,
"CAMPO EN BLANCO"
```

```
        Exit Sub
```

```
    Else
```

```
        txtFecha.SetFocus
```

```
    End If
```

```
End If
```

```
End Sub
```

```
'##ModelId=3FD6A82C0244
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    cmdEjecutar.SetFocus
```

```
End If
```

```
End Sub
```





```
'##ModelId=3FD6A82C035C
```

```
Private Sub txtCodigo_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
If Adodc1.Recordset RecordCount >= 1 Then
```

```
    Adodc1 Recordset MoveFirst
```

```
End If
```

```
If Trim(Adodc1.Recordset Fields("COD_PROD")) = Trim(txtCodigo Text) Then
```

```
    txtCodigo Enabled = False
```

```
    txtDescripcion Enabled = False
```

```
    txtUm.Enabled = False
```

```
    txtDescripcion.Text = Adodc1 Recordset Fields("DESCRIPCION")
```

```
    txtUm Text = Adodc1.Recordset Fields("UM")
```

```
Else
```

```
    Adodc1 Recordset MoveNext
```

```
    If Adodc1 Recordset EOF Then
```

```
        MsgBox ("REGISTRO NO EXISTE, INVESTIGUE POR FAVOR")
```

```
        Exit Sub
```

```
    End If
```

```
End If
```

```
txtInven_Fisico SetFocus
```

```
End If
```

```
End Sub
```

```
'##ModelId=3FD6A82D00BE
```

```
Private Sub txtFecha_FISICA_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    txtMes.SetFocus
```





End If
End Sub

```

'##ModelId=3FD6A82D0208
Private Sub txtInven_Fisico_KeyPress(KeyAscii As Integer)
If (KeyAscii = vbKeyReturn) Then
    txtFecha_Fisica.SetFocus
End If
End Sub
    
```

Este formulario manda a llamar a una clase llamada CINVENTAFISICO y su código es el siguiente

```

'variables locales para almacenar los valores de las propiedades
'##ModelId=3FD6A830024E
Private mvarCOD_PROD As String 'copia local
'##ModelId=3FD6A83002F8
Private mvarDESCRIPCION As String 'copia local
'##ModelId=3FD6A8300398
Private mvarUM As String 'copia local
'##ModelId=3FD6A831001E
Private mvarCANTIDAD As Double 'copia local
'##ModelId=3FD6A83100C8
Private mvarFECHA_FISI As Date 'copia local
'##ModelId=3FD6A8310168
Private mvarMES As String 'copia local
'##ModelId=3FD6A8310244
    
```





Private mvarINVEN_FISICO As Double 'copia local

'##ModelId=3FD6A83102F0

Private CN As Connection

'##ModelId=3FD6A8310320

Private RS As Recordset

'##ModelId=3FD6A8310321

Public Property Let INVEN_FISICO(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X INVEN_FISICO = 5

mvarINVEN_FISICO = vData

End Property

'##ModelId=3FD6A8320082

Public Property Get INVEN_FISICO() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax. Debug Print X.INVEN_FISICO

INVEN_FISICO = mvarINVEN_FISICO

End Property

'##ModelId=3FD6A832015E

Public Property Let MES(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax. X MES = 5

mvarMES = vData

End Property





```
'###ModelId=3FD6A8320316
```

```
Public Property Get MES() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax Debug Print X MES
```

```
    MES = mvarMES
```

```
End Property
```

```
'###ModelId=3FD6A83203C0
```

```
Public Sub MODIFICAR()
```

```
    Set RS ActiveConnection = CN
```

```
    RS LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
```

```
    RS CursorType = adOpenKeyset ' ACTIVA EL CURSOR
```

```
    RS Open "SELECT * FROM INVENTARIO"
```

```
    RS.MoveFirst
```

```
    Do While Not RS.EOF
```

```
        If RS.EOF Then
```

```
            Exit Sub
```

```
        Else
```

```
            If RTrim(RS.Fields("COD_PROD")) = COD_PROD Then
```

```
                Rem MsgBox ("LO ENCONTRE")
```

```
                RS.Fields("INVEN_FISICO") = INVEN_FISICO
```

```
                RS.Fields("FECHA_FISI") = FECHA_FISI
```

```
                RS.Fields("MES") = MES
```

```
                RS.Update
```

```
                Exit Sub
```

```
            Rem Else
```

```
            Rem MsgBox ("CODIGO = ") & RS.Fields("COD_PROD")
```





```

End If
RS MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A8330046
Public Sub GUARDAR()
Set RS ActiveConnection = CN
RS LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS.Open "SELECT * FROM INVENTARIO ORDER BY COD_PROD"
RS.MoveFirst
Do While Not RS.EOF
If RS.EOF Then
Exit Sub
Else
If RTrim(RS.Fields("COD_PROD")) = COD_PROD Then
Rem MsgBox ("LO ENCONTRE")
RS.Fields("INVEN_FISICO") = INVEN_FISICO
RS.Fields("FECHA_FISI") = FECHA_FISI
RS.Fields("MES") = MES
RS.Update
Exit Sub
Rem Else
Rem MsgBox ("CODIGO = ") & RS.Fields("COD_PROD")
End If

```





```

    RS MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A83300B4
Public Sub CLASS_INITIALIZE()
Set CN = New Connection
Set RS = New Recordset
CN.ConnectionString = "Provider=MSDASQL 1,Persist Security Info=False,Data
Source=BD_INVENTARIO"
CN Open
End Sub

'##ModelId=3FD6A8330122
Public Sub ELIMINAR()
Set RS ActiveConnection = CN
RS LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS Open "SELECT * FROM INVENTARIO"
RS MoveFirst
Do While Not RS EOF
If RS EOF Then
Exit Sub
Else
If RTrim(RS.Fields("COD_PROD")) = COD_PROD Then
Rem MsgBox ("LO ENCONTRE")

```





```

    RS.Fields("INVEN_FISICO") = 0#
    RS.Update
    Exit Sub
End If
    RS.MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A8330190
Public Sub BUSCAR(MBUSCA As String)
    Dim MBUSCAR As String
    MBUSCAR = MBUSCA
    Set RS.ActiveConnection = CN
    RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
    RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
    RS.Open "SELECT * FROM INVENTARIO order by cod_prod "
    Rem MsgBox (" 1 MBUSCAR = ") & MBUSCAR
    RS.MoveFirst
    Do While Not RS.EOF
        If RS.EOF Then
            Exit Sub
        Else
            If RTrim(RS.Fields("COD_PROD")) = MBUSCAR Then
                Rem MsgBox ("LO ENCONTRE")
                COD_PROD = RS.Fields("COD_PROD")
                DESCRIPCION = RS.Fields("DESCRIPCION")
            End If
        End If
    Loop
End Sub

```





```

    UM = RS.Fields("UM")
    If RS.Fields("inven_fisico") = "" Then
        RS.Fields("inven_fisico") = 0#
    End If
    INVEN_FISICO = RS.Fields("INVEN_FISICO")
    FECHA_FISI = RS.Fields("FECHA_FISI")
    MES = RS.Fields("MES")
    Exit Sub
Else
    Rem MsgBox ("CODIGO = ") & RS.Fields("COD_PROD")
End If
RS.MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A833029E
Public Property Let FECHA_FISI(ByVal vData As Date)
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
'Syntax X FECHA = 5
    mvarFECHA_FISI = vData
End Property

'##ModelId=3FD6A834006E
Public Property Get FECHA_FISI() As Date
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
'Syntax: Debug.Print X FECHA

```





```
FECHA_FISI = mvarFECHA_FISI
```

```
End Property
```

```
###ModelId=3FD6A834014A
```

```
Public Property Let CANTIDAD(ByVal vData As Double)
```

```
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
```

```
'Syntax X CANTIDAD = 5
```

```
    mvarCANTIDAD = vData
```

```
End Property
```

```
###ModelId=3FD6A83402C6
```

```
Public Property Get CANTIDAD() As Double
```

```
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
```

```
'Syntax. Debug Print X CANTIDAD
```

```
    CANTIDAD = mvarCANTIDAD
```

```
End Property
```

```
###ModelId=3FD6A83403DE
```

```
Public Property Let UM(ByVal vData As String)
```

```
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
```

```
'Syntax. X.UM = 5
```

```
    mvarUM = vData
```

```
End Property
```

```
###ModelId=3FD6A8350172
```

```
Public Property Get UM() As String
```

```
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
```





'Syntax. Debug Print X.UM

UM = mvarUM

End Property

'##ModelId=3FD6A835028A

Public Property Let DESCRIPCION(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X DESCRIPCION = 5

mvarDESCRIPCION = vData

End Property

'##ModelId=3FD6A836008C

Public Property Get DESCRIPCION() As String

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax: Debug Print X.DESCRIPCION

DESCRIPCION = mvarDESCRIPCION

End Property

'##ModelId=3FD6A8360168

Public Property Let COD_PROD(ByVal vData As String)

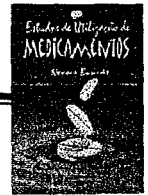
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X COD_PROD = 5

mvarCOD_PROD = vData

End Property





```
##ModelId=3FD6A836035C
```

```
Public Property Get COD_PROD() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

```
'Syntax: Debug.Print X.COD_PROD
```

```
    COD_PROD = mvarCOD_PROD
```

```
End Property
```

COD_PROD	DESCRIPCION	UM	INVEN FISICO	FECHA FISI
01	Aspirina	Cap	0	1/1/01
02	Acetaminofen	Tab	0	1/1/01

AGREGAR PRODUCTOS

CODIGO DEL PRODUCTO:

DESCRIPCION:

UNIDAD DE MEDIDA:

INVENTARIO FISICO:

FECHA FISICA:

MES:

GUARDAR CANCELAR

CÓDIGO DEL FORMULARIO DE PROVEEDORES:

Option Explicit

```
##ModelId=3FD6A84D021E
```

```
Private CN As Connection
```

```
##ModelId=3FD6A84D0250
```

```
Private RS As Recordset
```

```
##ModelId=3FD6A84D0251
```





Private X As Integer

##ModelId=3FD6A84D02BC

```

Private Sub cmdActualizar_Click()
Dim PRODUCTO2 As CPROVEEDOR
Set PRODUCTO2 = New CPROVEEDOR
Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE
PRODUCTO2 COD_PROV = RTrim(txtCod_Prov Text)
PRODUCTO2 NOMBRE_PROVE = txtNombre_Prove Text
PRODUCTO2 DIRECCION = txtDireccion Text
PRODUCTO2 TELEFONO = Val(txtTelefono.Text)
PRODUCTO2 CELULAR = txtCelular.Text
PRODUCTO2 EMAIL = txtEmail Text
PRODUCTO2.MODIFICAR
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1.Visible = False
txtCod_Prov.Visible = False
txtNombre_Prove.Visible = False
txtDireccion.Visible = False
txtTelefono.Visible = False
txtCelular.Visible = False
txtEmail.Visible = False
lblCod_Prov.Visible = False
lblNombre_Prove.Visible = False
lblDireccion.Visible = False
lblTelefono.Visible = False
lblCelular.Visible = False
    
```





```

lblEmail.Visible = False
lblModificar.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
Rem Adodc1.Recordset.Update
Rem RS.MoveLast
Rem Adodc1.Recordset.MoveLast
DataGrid1.Refresh
End If
MsgBox ("REGISTRO ACTUALIZADO EN PROVEEDORES")
End Sub

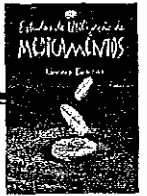
```

```

'##ModelId=3FD6A84D032A
Private Sub cmdAgregar_Click()
Rem BOTON AGREGAR
txtCod_Prov.Visible = True
txtCod_Prov.Enabled = True
txtNombre_Prove.Visible = True
txtDireccion.Visible = True
txtTelefono.Visible = True
txtCelular.Visible = True
txtEmail.Visible = True
lblCod_Prov.Visible = True
lblNombre_Prove.Visible = True

```





```

lblDireccion.Visible = True
lblTelefono.Visible = True
lblCelular.Visible = True
lblEmail.Visible = True
lblEtiqueta.Visible = True
lblEncontrado.Visible = False
cmdGuardar.Visible = True
cmdCancelar.Visible = True
cmdModificar.Enabled = False
cmdEliminar.Enabled = False
Frame1.Visible = True
txtCod_Prov.SetFocus
Rem INICIALIZAR TEXTOS
txtCod_Prov.Text = " "
txtNombre_Prove.Text = " "
txtDireccion.Text = " "
txtTelefono.Text = " "
txtCelular.Text = " "
txtEmail.Text = " "
End Sub

```

```

'##ModelId=3FD6A84D0398
Private Sub cmdAnterior_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MovePrevious
DataGrid1.Refresh
cmdSiguiente.Enabled = True

```





```
cmdFinal Enabled = True
If Adodc1 Recordset BOF Then
    Adodc1 Recordset MoveFirst
    cmdAnterior Enabled = False
    cmdInicio Enabled = False
End If
End Sub
```

```
'##ModelId=3FD6A84E001E
Private Sub cmdBuscar_Click()
Rem BUSCAR UN PRODUCTO POR MEDIO DEL CODIGO
Text1 Visible = True
cmdEjecutar Visible = True
Text1 SetFocus
End Sub
```

```
'##ModelId=3FD6A84E008C
Private Sub cmdCancelar_Click()
Rem BOTON CANCELAR
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1 Visible = False
txtCod_Prov Visible = False
txtNombre_Prove Visible = False
txtDireccion.Visible = False
txtTelefono Visible = False
txtCelular.Visible = False
txtEmail Visible = False
```





```

IblCod_Prov Visible = False
IblNombre_Prove Visible = False
IblDireccion Visible = False
IblTelefono.Visible = False
IblCelular.Visible = False
IblEmail Visible = False
Ibletiqueta Visible = False
IblModificar Visible = False
IblEncontrado.Visible = False
cmdGuardar Visible = False
cmdCancelar Visible = False
cmdActualizar Visible = False
End Sub

```

```

'##ModelId=3FD6A84E0136
Private Sub cmdEjecutar_Click()
Dim PRODUCTO1 As CPROVEEDOR
Set PRODUCTO1 = New CPROVEEDOR
Dim MBUS As String
MBUS = RTrim(Text1.Text)
PRODUCTO1.BUSCAR (MBUS)
txtCod_Prov Visible = True
txtCod_Prov.Text = PRODUCTO1.COD_PROV
txtNombre_Prove.Text = PRODUCTO1.NOMBRE_PROVE
txtDireccion.Text = PRODUCTO1.DIRECCION
txtTelefono.Text = PRODUCTO1.TELEFONO
txtCelular.Text = PRODUCTO1.CELULAR

```





```

txtEmail Text = PRODUCTO1 EMAIL
Rem HACER VISIBLE LOS CAMPOS
lblEncontrado Visible = True
txtCod_Prov Visible = True
txtNombre_Prove Visible = True
txtDireccion Visible = True
txtTelefono Visible = True
txtCelular Visible = True
txtEmail Visible = True
lblCod_Prov Visible = True
lblNombre_Prove Visible = True
lblDireccion Visible = True
lblTelefono.Visible = True
lblCelular Visible = True
lblEmail Visible = True
cmdCancelar Visible = True
Rem cmdActualizar Visible = True
Frame1 Visible = True
cmdModificar Enabled = True
cmdEliminar Enabled = True
Text1.Visible = False
cmdEjecutar.Visible = False
Text1 Text = " "
Rem txtDescripcion SetFocus
Frame1 Visible = True
End Sub
    
```





```
'##ModelId=3FD6A84E01A4
Private Sub cmdEliminar_Click()
Dim PRODUCTO3 As CPROVEEDOR
Set PRODUCTO3 = New CPROVEEDOR
Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE
PRODUCTO3 COD_PROV = RTrim(txtCod_Prov.Text)
PRODUCTO3.NOMBRE_PROVE = txtNombre_Prove.Text
PRODUCTO3 DIRECCION = txtDireccion.Text
PRODUCTO3 TELEFONO = Val(txtTelefono.Text)
PRODUCTO3 CELULAR = txtCelular.Text
PRODUCTO3 EMAIL = txtEmail.Text
PRODUCTO3.ELIMINAR
IblEliminar.Visible = True
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1.Visible = False
txtCod_Prov.Visible = False
txtNombre_Prove.Visible = False
txtDireccion.Visible = False
txtTelefono.Visible = False
txtCelular.Visible = False
txtEmail.Visible = False
IblCod_Prov.Visible = False
IblNombre_Prove.Visible = False
IblDireccion.Visible = False
IblTelefono.Visible = False
IblCelular.Visible = False
IblEmail.Visible = False
```





```

lblEliminar.Visible = False
lblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
Rem Adodc1.Recordset.Update
Rem RS.MoveLast
Rem Adodc1.Recordset.MoveLast
DataGrid1.Refresh
End If
MsgBox ("REGISTRO ELIMINADO EN PROVEEDOR")
End Sub

```

```

'##ModelId=3FD6A84E0212
Private Sub cmdFinal_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MoveLast
DataGrid1.Refresh
cmdSiguiente.Enabled = False
cmdFinal.Enabled = False
cmdAnterior.Enabled = True
cmdInicio.Enabled = True
End Sub

```

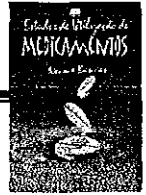




```

'##ModelId=3FD6A84E0280
Private Sub cmdGuardar_Click()
Dim PRODUCTO As CPROVEEDOR
Set PRODUCTO = New CPROVEEDOR
PRODUCTO COD_PROV = LTrim(txtCod_Prov.Text)
PRODUCTO NOMBRE_PROVE = txtNombre_Prove.Text
PRODUCTO DIRECCION = txtDireccion.Text
PRODUCTO TELEFONO = Val(txtTelefono.Text)
PRODUCTO CELULAR = txtCelular.Text
PRODUCTO EMAIL = txtEmail.Text
PRODUCTO GUARDAR
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1.Visible = False
txtCod_Prov.Visible = False
txtNombre_Prove.Visible = False
txtDireccion.Visible = False
txtTelefono.Visible = False
txtCelular.Visible = False
txtEmail.Visible = False
lblEtiqueta.Visible = False
lblCod_Prov.Visible = False
lblNombre_Prove.Visible = False
lblDireccion.Visible = False
lblTelefono.Visible = False
lblCelular.Visible = False
lblEmail.Visible = False
lblEtiqueta.Visible = False
    
```





```
cmdGuardar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
Rem Adodc1.Recordset.Update
Rem RS.MoveLast
Rem Adodc1.Recordset.MoveLast
DataGrid1.Refresh
End If
MsgBox ("REGISTRO ALMACENADO EN PROVEEDOR")
End Sub
```

```
'##ModelId=3FD6A84E02EE
Private Sub cmdInicio_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MoveFirst
DataGrid1.Refresh
cmdAnterior.Enabled = False
cmdInicio.Enabled = False
cmdSiguiente.Enabled = True
End Sub
```

```
'##ModelId=3FD6A84E035C
Private Sub cmdModificar_Click()
Rem MODIFICAR UN REGISTRO
cmdActualizar.Visible = True
lblModificar.Visible = True
lblEncontrado.Visible = False
```





```

Ibletiqueta Visible = False
txtNombre_Prove SetFocus
End Sub

```

```

'##ModelId=3FD6A84F0014
Private Sub cmdRetornar_Click()
frmprincipal Show
Unload Me
End Sub

```

```

'##ModelId=3FD6A84F0082
Private Sub cmdSiguiente_Click()
Rem SE CORRE AL SIGUIENTE REGISTRO
Adodc1.Recordset MoveNext
DataGrid1.Refresh
cmdAnterior.Enabled = True
cmdInicio.Enabled = True
If Adodc1.Recordset EOF Then
    Adodc1.Recordset MoveLast
    cmdFinal.Enabled = False
    cmdSiguiente.Enabled = False
End If
End Sub

```

```

'##ModelId=3FD6A84F00F0
Private Sub Form_Load()
Set CN = New Connection

```

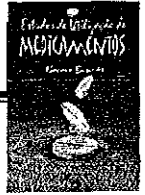




```
Set RS = New Recordset
CN.CursorLocation = adUseClient
CN.ConnectionString = "Provider=MSDASQL.1;Persist Security Info=False;Data
Source=BD_INVENTARIO"
CN.Open
Set RS.ActiveConnection = CN
RS.CursorType = adOpenForwardOnly
RS.Open "select * from PROVEEDOR"
End Sub
```

```
'##ModelId=3FD6A84F015E
Private Sub txtCod_Prov_KeyPress(KeyAscii As Integer)
If (KeyAscii = vbKeyReturn) Then
If Adodc1.Recordset.RecordCount >= 1 Then
    Adodc1.Recordset.MoveFirst
End If
Do While Not Adodc1.Recordset.EOF
    If Adodc1.Recordset.EOF Then
        Exit Do ' si no lo encuentra continua
    End If
    If Trim(Adodc1.Recordset.Fields("COD_PROV")) = Trim(txtCod_Prov.Text) Then
        MsgBox ("Registro existe ")
        txtCod_Prov.Text = ""
        txtCod_Prov.SetFocus
    End Sub
End If
Adodc1.Recordset.MoveNext
```





Loop

txtNombre_Prove SetFocus

End If

End Sub

Este formulario manda a llamar a una clase llamada CPROVEEDOR y su código es el siguiente

'variables locales para almacenar los valores de las propiedades

'##ModelId=3FD6A8520028

Private mvarCOD_PROD As String 'copia local

'##ModelId=3FD6A8520096

Private mvarNOMBRE_PROVE As String 'copia local

'##ModelId=3FD6A8520140

Private mvarDIRECCION As String 'copia local

'##ModelId=3FD6A85201E0

Private mvarTELEFONO As String 'copia local

'##ModelId=3FD6A852028A

Private mvarCELULAR As String 'copia local

'##ModelId=3FD6A852032A

Private mvarEMAIL As String 'copia local

'##ModelId=3FD6A85203D6

Private CN As Connection

'##ModelId=3FD6A85203D9

Private RS As Recordset

'##ModelId=3FD6A853001E





Public Property Let EMAIL(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X EMAIL = 5

mvarEMAIL = vData

End Property

##ModelId=3FD6A8530168

Public Property Get EMAIL() As String

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

'Syntax Debug Print X EMAIL

EMAIL = mvarEMAIL

End Property

'##ModelId=3FD6A8530212

Public Sub MODIFICAR()

Set RS ActiveConnection = CN

RS LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR

RS CursorType = adOpenKeyset ' ACTIVA EL CURSOR

RS.Open "SELECT * FROM PROVEEDOR"

RS MoveFirst

Do While Not RS.EOF

If RS.EOF Then

Exit Sub

Else

If RTrim(RS.Fields("COD_PROV")) = COD_PROV Then

Rem MsgBox ("LO ENCONTRE")

RS.Fields("COD_PROV") = COD_PROV





```

RS Fields("NOMBRE_PROVE") = NOMBRE_PROVE
RS Fields("DIRECCION") = DIRECCION
RS Fields("TELEFONO") = TELEFONO
RS Fields("CELULAR") = CELULAR
RS Fields("EMAIL") = EMAIL
RS Update
Exit Sub
Rem Else
Rem , MsgBox ("CODIGO = ") & RS.Fields("COD_PROD")
End If
RS MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A8530280
Public Sub GUARDAR()
Set RS.ActiveConnection = CN
RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS.Open "SELECT * FROM PROVEEDOR"
RS.AddNew
RS.Fields("COD_PROV") = COD_PROV
RS.Fields("NOMBRE_PROVE") = NOMBRE_PROVE
RS.Fields("DIRECCION") = DIRECCION
RS.Fields("TELEFONO") = TELEFONO
RS.Fields("CELULAR") = CELULAR

```





RS.Fields("EMAIL") = EMAIL

RS.Update

End Sub

'##ModelId=3FD6A85302EE

Public Sub CLASS_INITIALIZE()

Set CN = New Connection

Set RS = New Recordset

CN.ConnectionString = "Provider=MSDASQL 1,Persist Security Info=False;Data Source=BD_INVENTARIO"

CN.Open

End Sub

'##ModelId=3FD6A8530320

Public Sub ELIMINAR()

Set RS.ActiveConnection = CN

RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR

RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR

RS.Open "SELECT * FROM PROVEEDOR"

RS.MoveFirst

Do While Not RS.EOF

If RS.EOF Then

Exit Sub

Else

If RTrim(RS.Fields("COD_PROV")) = COD_PROV Then

Rem MsgBox ("LO ENCONTRE")

RS.Delete





```

Exit Sub
Rem Else
Rem  MsgBox ("CODIGO = ") & RS Fields("COD_PROD")
End If
RS MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A853038E
Public Sub BUSCAR(MBUSCA As String)
Dim MBUSCAR As String
MBUSCAR = MBUSCA
Set RS ActiveConnection = CN
RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS.Open "SELECT * FROM PROVEEDOR order by cod_prov "
Rem MsgBox (" 1 MBUSCAR = ") & MBUSCAR
RS MoveFirst
Do While Not RS EOF
If RS EOF Then
Exit Sub
Else
If RTrim(RS Fields("COD_PROV")) = MBUSCAR Then
Rem  MsgBox ("LO ENCONTRE")
COD_PROV = RS Fields("COD_PROV")
NOMBRE_PROVE = RS Fields("NOMBRE_PROVE")

```





```

DIRECCION = RS Fields("DIRECCION")
TELEFONO = RS.Fields("TELEFONO")
CELULAR = RS Fields("CELULAR")
EMAIL = RS Fields("EMAIL")
Exit Sub

Else
    Rem  MsgBox ("CODIGO = ") & RS Fields("COD_PROD")
End If
    RS MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A85400BE
Public Property Let CELULAR(ByVal vData As String)
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
'Syntax X TELEFONO2 = 5
    mvarCELULAR = vData
End Property

'##ModelId=3FD6A854023A
Public Property Get CELULAR() As String
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
'Syntax. Debug Print X TELEFONO2
    CELULAR = mvarCELULAR
End Property
    
```





```
'''ModelId=3FD6A85402DA
```

```
Public Property Let TELEFONO(ByVal vData As String)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax X TELEFONO1 = 5
```

```
    mvarTELEFONO = vData
```

```
End Property
```

```
'''ModelId=3FD6A8550078
```

```
Public Property Get TELEFONO() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax Debug.Print X.TELEFONO1
```

```
    TELEFONO = mvarTELEFONO
```

```
End Property
```

```
'''ModelId=3FD6A8550154
```

```
Public Property Let DIRECCION(ByVal vData As String)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax X DIRECCION = 5
```

```
    mvarDIRECCION = vData
```

```
End Property
```

```
'''ModelId=3FD6A855030C
```

```
Public Property Get DIRECCION() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

```
'Syntax Debug Print X DIRECCION
```

```
    DIRECCION = mvarDIRECCION
```

```
End Property
```





```
'''ModelId=3FD6A8560000
```

```
Public Property Let NOMBRE_PROVE(ByVal vData As String)
```

```
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
```

```
'Syntax X.NOMBRE_PROVE = 5
```

```
    mvarNOMBRE_PROVE = vData
```

```
End Property
```

```
'''ModelId=3FD6A85601B8
```

```
Public Property Get NOMBRE_PROVE() As String
```

```
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.
```

```
'Syntax: Debug.Print X.NOMBRE_PROVE
```

```
    NOMBRE_PROVE = mvarNOMBRE_PROVE
```

```
End Property
```

```
'''ModelId=3FD6A8560294
```

```
Public Property Let COD_PROV(ByVal vData As String)
```

```
'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación
```

```
'Syntax X.COD_PROD = 5
```

```
    mvarCOD_PROD = vData
```

```
End Property
```

```
'''ModelId=3FD6A8570064
```

```
Public Property Get COD_PROV() As String
```

```
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación
```

```
'Syntax. Debug.Print X.COD_PROD
```

```
    COD_PROV = mvarCOD_PROD
```

```
End Property
```





PROVEEDORES

PROVEEDOR		
COD. PROV.	NOMBRE PROVE.	DIRECCION
01	DISTRIBUIDORA SOLKA	MANAGUA

AGREGAR
MODIFICAR
ELIMINAR
INICIO
SIGUIENTE
ANTERIOR
FINAL
BUSCAR
RETORNAR

AGREGAR PRODUCTOS

CODIGO DEL PROVEEDOR:

TELEFONO:

NOMBRE PROVEEDOR:

CELULAR:

DIRECCION:

CORREO ELECTRONICO:

GUARDAR

CANCELAR





CÓDIGO DEL FORMULARIO DE RECEPCIONES:

Option Explicit

'##ModelId=3FD6A800002A

Private CN As Connection

'##ModelId=3FD6A800002D

Private RS As Recordset

'##ModelId=3FD6A800005A

Private X As Integer

'##ModelId=3FD6A8000104

Private Sub cmdActualizar_Click()

Dim PRODUCTO2 As CRECEPCIONES

Set PRODUCTO2 = New CRECEPCIONES

Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE

PRODUCTO2.COD_PROD = RTrim(txtCodigo.Text)

PRODUCTO2.COD_RECEP = RTrim(txtCod_Recep.Text)

PRODUCTO2.DESCRIPCION = txtDescripcion.Text

PRODUCTO2.UM = txtUm.Text

PRODUCTO2.ENTRADAS = txtEntradas.Text

PRODUCTO2.FECHA_ENTRADA = txtFecha_Entrada.Text

PRODUCTO2.PRECIO_UNIT = Val(txtPrecio_Unit.Text)

PRODUCTO2.PRECIO_TOTAL = Val(txtPrecio_Total.Text)

PRODUCTO2.COD_PROV = txtCod_Prov.Text

PRODUCTO2.NUM_DOC = txtNum_Doc.Text

PRODUCTO2.WEN1 = Val(Text2.Text)





PRODUCTO2 MODIFICAR

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1.Visible = False
txtCodigo.Visible = False
txtCod_Recep.Visible = False
txtDescripcion.Visible = False
txtUm.Visible = False
txtEntradas.Visible = False
txtFecha_Entrada.Visible = False
txtPrecio_Unit.Visible = False
txtPrecio_Total.Visible = False
txtCod_Prov.Visible = False
txtNum_Doc.Visible = False
lblCodigo.Visible = False
lblCod_Recep.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblEntradas.Visible = False
lblFecha_Entrada.Visible = False
lblPrecio_Unit.Visible = False
lblPrecio_Total.Visible = False
lblCod_Prov.Visible = False
lblNum_Doc.Visible = False
lblModificar.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False





```

If Adodc1 Recordset.RecordCount = -1 Then
Else
Rem Adodc1 Recordset.Update
Rem RS MoveLast
Rem Adodc1.Recordset MoveLast
DataGrid1 Refresh
End If
MsgBox ("REGISTRO ACTUALIZADO EN RECEPCIONES")
End Sub

```

```

'##ModelId=3FD6A8000136
Private Sub cmdAgregar_Click()
Rem BOTON AGREGAR
txtCodigo.Visible = True
txtCodigo.Enabled = True
txtCod_Recep.Enabled = True
txtCod_Recep.Visible = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtEntradas.Visible = True
txtFecha_Entrada.Visible = True
txtPrecio_Unit.Visible = True
txtPrecio_Total.Visible = True
txtCod_Prov.Visible = True
txtNum_Doc.Visible = True
lblCodigo.Visible = True

```





```

lblDescripcion.Visible = True
lblUm.Visible = True
lblEntradas.Visible = True
lblFecha_Entrada.Visible = True
lblPrecio_Unit.Visible = True
lblPrecio_Total.Visible = True
lblCod_Prov.Visible = True
lblNum_Doc.Visible = True
lblEtiqueta.Visible = True
Adodc3.Visible = True
lblEncontrado.Visible = False
cmdGuardar.Visible = True
cmdCancelar.Visible = True
cmdModificar.Enabled = False
cmdEliminar.Enabled = False
Frame1.Visible = True
txtCodigo.SetFocus
Rem INICIALIZAR TEXTOS
txtCodigo.Text = " "
txtDescripcion.Text = " "
txtUm.Text = " "
txtEntradas.Text = " "
txtFecha_Entrada.Text = " "
txtPrecio_Unit.Text = " "
txtPrecio_Total.Text = " "
txtCod_Prov.Text = " "
txtNum_Doc.Text = " "
    
```





DataGrid2 Refresh

End Sub

'##ModelId=3FD6A80001A4

Private Sub cmdAnterior_Click()

Rem REGRESA AL REGISTRO ANTERIOR

Adodc1.Recordset.MovePrevious

cmdSiguiente.Enabled = True

cmdFinal.Enabled = True

If Adodc1.Recordset.BOF Then

Adodc1.Recordset.MoveFirst

cmdAnterior.Enabled = False

cmdInicio.Enabled = False

End If

DataGrid2 Refresh

End Sub

'##ModelId=3FD6A8000212

Private Sub cmdBuscar_Click()

Rem BUSCAR UN PRODUCTO POR MEDIO DEL CODIGO

Text1.Visible = True

txtDocumento.Visible = True

cmdCancela.Visible = True

cmdEjecutar.Visible = True

Text1.SetFocus

End Sub





```
'##ModelId=3FD6A80002BC
```

```
Private Sub cmdCancela_Click()
```

```
Text1.Visible = False
```

```
txtDocumento.Visible = False
```

```
cmdCancela.Visible = False
```

```
cmdEjecutar.Visible = False
```

```
End Sub
```

```
'##ModelId=3FD6A800032A
```

```
Private Sub cmdCancelar_Click()
```

```
Rem BOTON CANCELAR
```

```
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
```

```
Frame1.Visible = False
```

```
txtCodigo.Visible = False
```

```
txtCod_Recep.Visible = False
```

```
txtDescripcion.Visible = False
```

```
txtUm.Visible = False
```

```
txtEntradas.Visible = False
```

```
txtFecha_Entrada.Visible = False
```

```
txtPrecio_Unit.Visible = False
```

```
txtPrecio_Total.Visible = False
```

```
txtCod_Prov.Visible = False
```

```
txtNum_Doc.Visible = False
```

```
lblCodigo.Visible = False
```

```
lblCod_Recep.Visible = False
```

```
lblDescripcion.Visible = False
```





```

IblUm Visible = False
IblEntradas Visible = False
IblFecha_Entrada Visible = False
IblPrecio_Unit.Visible = False
IblPrecio_Total Visible = False
IblCod_Prov Visible = False
IblNum_Doc.Visible = False
Ibletiqueta Visible = False
IblModificar.Visible = False
IblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar Visible = False
cmdActualizar Visible = False
cmdModificar Enabled = False
cmdEliminar.Enabled = False
Adodc3 Visible = False
End Sub

```

```

'##ModelId=3FD6A8000398
Private Sub cmdEjecutar_Click()
Dim PRODUCTO1 As CRECEPCIONES
Set PRODUCTO1 = New CRECEPCIONES
Dim MBUS As String
Dim mdocu As String
PRODUCTO1 NUM_DOC = RTrim(txtDocumento Text)
MBUS = RTrim(Text1.Text)

```





PRODUCTO1 BUSCAR (MBUS)

txtCodigo Text = PRODUCTO1.COD_PROD

txtCod_Recep Text = PRODUCTO1 COD_RECEP

txtDescripcion Text = PRODUCTO1 DESCRIPCION

txtUm.Text = PRODUCTO1.UM

txtEntradas.Text = PRODUCTO1 ENTRADAS

Text2 Text = PRODUCTO1.ENTRADAS

txtFecha_Entrada.Text = PRODUCTO1 FECHA_ENTRADA

txtPrecio_Unit Text = PRODUCTO1 PRECIO_UNIT

txtPrecio_Total Text = PRODUCTO1 PRECIO_TOTAL

txtCod_Prov.Text = PRODUCTO1 COD_PROV

txtNum_Doc.Text = PRODUCTO1.NUM_DOC

Rem HACER VISIBLE LOS CAMPOS

txtCodigo Visible = True

txtCod_Recep Visible = True

txtDescripcion Visible = True

txtUm Visible = True

txtEntradas Visible = True

txtFecha_Entrada.Visible = True

txtPrecio_Unit Visible = True

txtPrecio_Total Visible = True

txtCod_Prov.Visible = True

txtNum_Doc Visible = True

lblEncontrado Visible = True

lblCodigo Visible = True

lblCod_Recep Visible = True





```

lblDescripcion.Visible = True
lblUm.Visible = True
lblEntradas.Visible = True
lblFecha_Entrada.Visible = True
lblPrecio_Unit.Visible = True
lblPrecio_Total.Visible = True
lblCod_Prov.Visible = True
lblNum_Doc.Visible = True
cmdCancelar.Visible = True
cmdActualizar.Visible = False
Frame1.Visible = True
cmdModificar.Enabled = True
cmdEliminar.Enabled = True
Text1.Visible = False
txtDocumento.Visible = False
cmdCancela.Visible = False
cmdEjecutar.Visible = False
Text1.Text = " "
txtDocumento.Text = " "
Rem txtDescripcion.SetFocus
Frame1.Visible = True
End Sub

```

```

'##ModelId=3FD6A801001E
Private Sub cmdEliminar_Click()
Dim PRODUCTO3 As CRECEPCIONES
Set PRODUCTO3 = New CRECEPCIONES

```





Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE

PRODUCTO3 COD_PROD = RTrim(txtCodigo Text)

PRODUCTO3 COD_RECEP = RTrim(txtCod_Recep Text)

PRODUCTO3.DESCRIPCION = txtDescripcion.Text

PRODUCTO3.UM = txtUm Text

PRODUCTO3 ENTRADAS = Val(txtEntradas Text)

PRODUCTO3.FECHA_ENTRADA = txtFecha_Entrada Text

PRODUCTO3 PRECIO_UNIT = Val(txtPrecio_Unit.Text)

PRODUCTO3 PRECIO_TOTAL = Val(txtPrecio_Total Text)

PRODUCTO3 COD_PROV = txtCod_Prov Text

PRODUCTO3.NUM_DOC = txtNum_Doc Text

PRODUCTO3.ELIMINAR

lblEliminar Visible = True

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1 Visible = False

txtCodigo Visible = False

txtCod_Recep Visible = False

txtDescripcion Visible = False

txtUm Visible = False

txtEntradas Visible = False

txtFecha_Entrada Visible = False

txtPrecio_Unit Visible = False

txtPrecio_Total Visible = False

txtCod_Prov.Visible = False

txtNum_Doc Visible = False

lblCodigo.Visible = False

lblCod_Recep.Visible = False





```

lblDescripcion.Visible = False
lblUm Visible = False
lblEntradas Visible = False
lblFecha_Entrada Visible = False
lblPrecio_Unit.Visible = False
lblPrecio_Total.Visible = False
lblCod_Prov.Visible = False
lblNum_Doc.Visible = False
lblEliminar.Visible = False
lblEncontrado Visible = False
cmdGuardar.Visible = False
cmdCancelar Visible = False
cmdActualizar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
Rem Adodc1.Recordset Update
Rem RS.MoveLast
Rem Adodc1.Recordset.MoveLast
DataGrid1 Refresh
End If
MsgBox ("REGISTRO ELIMINADO EN RECEPCIONES")
End Sub

'##ModelId=3FD6A801008C
Private Sub cmdFinal_Click()
Rem ESTA EN EL INICIO
Adodc1 Recordset MoveLast
    
```





cmdSiguiente Enabled = False

cmdFinal Enabled = False

cmdAnterior Enabled = True

cmdInicio Enabled = True

DataGrid2.Refresh

End Sub

'##ModelId=3FD6A80100FA

Private Sub cmdGuardar_Click()

Dim PRODUCTO As CRECEPCIONES

Set PRODUCTO = New CRECEPCIONES

PRODUCTO.COD_PROD = LTrim(txtCodigo Text)

PRODUCTO COD_RECEP = txtCod_Recep Text

PRODUCTO.DESCRIPCION = txtDescripcion Text

PRODUCTO UM = txtUm.Text

PRODUCTO ENTRADAS = Val(txtEntradas Text)

PRODUCTO FECHA_ENTRADA = txtFecha_Entrada Text

PRODUCTO PRECIO_UNIT = Val(txtPrecio_Unit Text)

PRODUCTO PRECIO_TOTAL = Val(txtPrecio_Total.Text)

PRODUCTO COD_PROV = Val(txtCod_Prov Text)

PRODUCTO NUM_DOC = Val(txtNum_Doc Text)

PRODUCTO GUARDAR

Rem ESCONDEMOS LOS CAMPOS UTILIZADOS

Frame1 Visible = False

txtCodigo.Visible = False

txtCodigo.Enabled = False

txtDescripcion.Visible = False





```
txtCod_Recep Visible = False
txtUm Visible = False
txtEntradas Visible = False
txtFecha_Entrada Visible = False
txtPrecio_Unit Visible = False
txtPrecio_Total Visible = False
txtCod_Prov Visible = False
txtNum_Doc.Visible = False
lblCodigo Visible = False
lblDescripcion Visible = False
lblUm Visible = False
lblEntradas Visible = False
lblFecha_Entrada Visible = False
lblPrecio_Unit Visible = False
lblPrecio_Total Visible = False
lblCod_Prov Visible = False
lblNum_Doc Visible = False
lblEtiqueta Visible = False
lblEncontrado Visible = False
cmdGuardar Visible = False
cmdCancelar.Visible = False
cmdModificar.Enabled = False
cmdEliminar Enabled = False
Adodc3.Visible = False
If Adodc1 Recordset.RecordCount = -1 Then
Else
Rem Adodc1 Recordset Update
```





```

Rem RS MoveLast
Rem Adodc1 Recordset MoveLast
DataGrid2 Refresh
End If
MsgBox ("REGISTRO ALMACENADO EN RECEPCIONES")
End Sub

```

```

'##ModelId=3FD6A801019A
Private Sub cmdInicio_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset MoveFirst
cmdAnterior.Enabled = False
cmdInicio.Enabled = False
cmdSiguiente.Enabled = True
DataGrid2 Refresh
End Sub

```

```

'##ModelId=3FD6A8010208
Private Sub cmdModificar_Click()
Rem MODIFICAR UN REGISTRO
cmdActualizar.Visible = True
lblModificar.Visible = True
lblEncontrado.Visible = False
lblEtiqueta.Visible = False
txtDescripcion.SetFocus
End Sub

```





```
'##ModelId=3FD6A80102B2
```

```
Private Sub cmdRetornar_Click()
```

```
frmprincipal Show
```

```
Unload Me
```

```
End Sub
```

```
'##ModelId=3FD6A8010320
```

```
Private Sub cmdSiguiente_Click()
```

```
Rem SE CORRE AL SIGUIENTE REGISTRO
```

```
Adodc1 Recordset MoveNext
```

```
cmdAnterior Enabled = True
```

```
cmdInicio Enabled = True
```

```
If Adodc1.Recordset EOF Then
```

```
    Adodc1 Recordset.MoveLast
```

```
    cmdFinal Enabled = False
```

```
    cmdSiguiente Enabled = False
```

```
End If
```

```
DataGrid2 Refresh
```

```
End Sub
```

```
'##ModelId=3FD6A80103C0
```

```
Private Sub Form_Load()
```

```
Set CN = New Connection
```

```
Set RS = New Recordset
```

```
CN.CursorLocation = adUseClient
```

```
CN.ConnectionString = "Provider=MSDASQL.1,Persist Security Info=False;Data  
Source=BD_INVENTARIO"
```





```
CN Open
Set RS ActiveConnection = CN
RS CursorType = adOpenForwardOnly
RS.Open "select * from RECEPCIONES"
End Sub
```

```
'##ModelId=3FD6A8020046
Private Sub Text1_KeyPress(KeyAscii As Integer)
If (KeyAscii = vbKeyReturn) Then
    txtDocumento SetFocus
End If
End Sub
```

```
'##ModelId=3FD6A8020190
Private Sub txtCod_Recep_KeyPress(KeyAscii As Integer)
If (KeyAscii = vbKeyReturn) Then
    txtCod_Recep Text = Adodc3.Recordset Fields("COD_RECEP")
    txtEntradas.SetFocus
End If
End Sub
```

```
'##ModelId=3FD6A80202A8
Private Sub txtCodigo_KeyPress(KeyAscii As Integer)
If (KeyAscii = vbKeyReturn) Then
    Adodc2.Recordset.MoveFirst
Do While Not Adodc2 Recordset EOF
    If Adodc2 Recordset EOF Then
```





```

Exit Do
End If
If Trim(Adodc2.Recordset.Fields("COD_PROD")) = Trim(txtCodigo.Text) Then
Rem   MsgBox ("pase")
    txtCodigo.Enabled = False
    txtDescripcion.Enabled = False
    txtUm.Enabled = False
    txtCod_Prov.Enabled = False
    txtDescripcion.Text = Adodc2.Recordset.Fields("DESCRIPCION")
    txtUm.Text = Adodc2.Recordset.Fields("UM")
    txtCod_Prov.Text = Adodc2.Recordset.Fields("COD_PROV")
    txtNum_Doc.SetFocus
    Exit Do
Else
    Adodc2.Recordset.MoveNext
    If Adodc2.Recordset.EOF Then
        MsgBox ("REGISTRO NO EXISTE, INVESTIGUE POR FAVOR")
        txtCodigo = ""
        Exit Sub
    End If
End If
Loop
txtNum_Doc.SetFocus
End If
End Sub

```





```
'##ModelId=3FD6A803000A
```

```
Private Sub txtDocumento_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    cmdEjecutar SetFocus
```

```
End If
```

```
    cmdEjecutar Enabled = True
```

```
End Sub
```

```
'##ModelId=3FD6A8030118
```

```
Private Sub txtEntradas_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    txtCod_Recep.Text = Adodc3.Recordset.Fields("COD_RECEP")
```

```
    txtFecha_Entrada SetFocus
```

```
End If
```

```
End Sub
```

```
'##ModelId=3FD6A8030262
```

```
Private Sub txtFecha_Entrada_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    txtPrecio_Unit SetFocus
```

```
End If
```

```
End Sub
```

```
'##ModelId=3FD6A80303AC
```

```
Private Sub txtNum_Doc_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = vbKeyReturn) Then
```

```
    txtCod_Recep SetFocus
```





End If

End Sub

'###ModelId=3FD6A804010E

Private Sub txtPrecio_Total_KeyPress(KeyAscii As Integer)

If (KeyAscii = vbKeyReturn) Then

txtNum_Doc.SetFocus

End If

End Sub

'###ModelId=3FD6A8040258

Private Sub txtPrecio_Unit_KeyPress(KeyAscii As Integer)

If (KeyAscii = vbKeyReturn) Then

txtPrecio_Total.Text = txtEntradas * txtPrecio_Unit

txtNum_Doc.SetFocus

End If

End Sub





RECEPCIONES DE PRODUCTOS

RECEPCIONES						
COD. PROD.	NUM. DOC.	COD. RECEP.	DESCRIPCION	UM	ENTRADAS	FE
01	1099	03	Aspirina	Cap	300	11/

AGREGAR MODIFICAR ELIMINAR INICIO SIGUIENTE ANTERIOR FINAL BUSCAR RETORNAR

AGREGAR PRODUCTOS

CODIGO DEL PRODUCTO:		ENTRADAS:	
NUMERO DOCUMENTO:		FECHA DE ENTRADA:	
CODIGO RECEPCION:	Bonificaciones	PRECIO UNIT:	
DESCRIPCION:		PRECIO TOTAL:	
UNIDAD DE MEDIDA:		CODIGO PROVEEDOR:	

GUARDAR CANCELAR

Inicio Micros INVEN REC 11:02 PM

CÓDIGO DEL FORMULARIO DE SALIDAS:

Option Explicit

'##ModelId=3FD6A822002A

Private CN As Connection

'##ModelId=3FD6A822002D

Private RS As Recordset

'##ModelId=3FD6A822005A

Private X As Integer





```
'##ModelId=3FD6A82200C8
Private Sub cmdActualizar_Click()
Dim PRODUCTO2 As CSALIDAS
Set PRODUCTO2 = New CSALIDAS
Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE
PRODUCTO2.COD_PROD = RTrim(txtCodigo Text)
PRODUCTO2.COD_SALIDAS = RTrim(txtCod_Sali Text)
PRODUCTO2 DESCRIPCION = txtDescripcion Text
PRODUCTO2 UM = txtUm Text
PRODUCTO2 SALIDAS = txtSalidas Text
PRODUCTO2 FECHA_SALIDAS = txtFecha_Salidas.Text
PRODUCTO2 PRECIO_UNIT = Val(txtPrecio_Unit Text)
PRODUCTO2.PRECIO_TOTAL = Val(txtPrecio_Total Text)
PRODUCTO2 NUM_DOC = txtNum_Doc.Text
PRODUCTO2 WSA1 = Val(Text2 Text)
PRODUCTO2.MODIFICAR
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1 Visible = False
txtCodigo Visible = False
txtCod_Sali Visible = False
txtDescripcion Visible = False
txtUm Visible = False
txtSalidas.Visible = False
txtFecha_Salidas.Visible = False
txtPrecio_Unit Visible = False
txtPrecio_Total.Visible = False
txtNum_Doc.Visible = False
```





```

lblCodigo.Visible = False
lblCod_Sali.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblSalidas.Visible = False
lblFecha_Salidas.Visible = False
lblPrecio_Unit.Visible = False
lblPrecio_Total.Visible = False
lblNum_Doc.Visible = False
lblModificar.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
Rem Adodc1.Recordset.Update
Rem RS.MoveLast
Rem Adodc1.Recordset.MoveLast
DataGrid1.Refresh
End If
MsgBox ("REGISTRO ACTUALIZADO EN SALIDAS")
End Sub

'##ModelId=3FD6A8220136
Private Sub cmdAgregar_Click()
Rem BOTON AGREGAR
    lblCodigo.Visible = True

```





```

txtCodigo.Enabled = True
txtCod_Sali.Enabled = True
txtCod_Sali.Visible = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtSalidas.Visible = True
txtFecha_Salidas.Visible = True
txtPrecio_Unit.Visible = True
txtPrecio_Total.Visible = True
txtNum_Doc.Visible = True
lblCodigo.Visible = True
lblDescripcion.Visible = True
lblUm.Visible = True
lblSalidas.Visible = True
lblFecha_Salidas.Visible = True
lblPrecio_Unit.Visible = True
lblPrecio_Total.Visible = True
lblCod_Sali.Visible = True
lblNum_Doc.Visible = True
lblEtiqueta.Visible = True
lblEncontrado.Visible = False
cmdGuardar.Visible = True
cmdCancelar.Visible = True
cmdModificar.Enabled = False
cmdEliminar.Enabled = False
Frame1.Visible = True
txtCodigo.SetFocus
    
```





Rem INICIALIZAR TEXTOS

txtCodigo.Text = " "

txtDescripcion.Text = " "

txtUm.Text = " "

txtSalidas.Text = " "

txtFecha_Salidas.Text = " "

txtPrecio_Unit.Text = " "

txtPrecio_Total.Text = " "

txtNum_Doc.Text = " "

DataGrid1.Refresh

End Sub

'##ModelId=3FD6A8220172

Private Sub cmdAnterior_Click()

Rem REGRESA AL REGISTRO ANTERIOR

Adodc1.Recordset.MovePrevious

cmdSiguiente.Enabled = True

cmdFinal.Enabled = True

If Adodc1.Recordset.BOF Then

Adodc1.Recordset.MoveFirst

cmdAnterior.Enabled = False

cmdInicio.Enabled = False

End If

DataGrid1.Refresh

End Sub





```

'##ModelId=3FD6A82201E0
Private Sub cmdBuscar_Click()
Rem BUSCAR UN PRODUCTO POR MEDIO DEL CODIGO
Text1.Visible = True
cmdEjecutar.Visible = True
cmdCancela.Visible = True
txtDocumento.Visible = True
Text1.SetFocus
End Sub

```

```

'##ModelId=3FD6A822024E
Private Sub cmdCancela_Click()
Text1.Visible = False
txtDocumento.Visible = False
cmdCancela.Visible = False
cmdEjecutar.Visible = False
End Sub

```

```

'##ModelId=3FD6A82202BC
Private Sub cmdCancelar_Click()
Rem BOTON CANCELAR
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1.Visible = False
txtCodigo.Visible = False
txtCod_Sali.Visible = False
txtDescripcion.Visible = False
txtUm.Visible = False

```





```

txtSalidas.Visible = False
txtFecha_Salidas.Visible = False
txtPrecio_Unit.Visible = False
txtPrecio_Total.Visible = False
txtNum_Doc.Visible = False
lblCodigo.Visible = False
lblCod_Sali.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblSalidas.Visible = False
lblFecha_Salidas.Visible = False
lblPrecio_Unit.Visible = False
lblPrecio_Total.Visible = False
lblNum_Doc.Visible = False
lblEtiqueta.Visible = False
lblModificar.Visible = False
lblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
cmdModificar.Enabled = False
cmdEliminar.Enabled = False
End Sub

```

```

'##ModelId=3FD6A822032A
Private Sub cmdEjecutar_Click()
Dim PRODUCTO1 As CSALIDAS

```





```

Set PRODUCTO1 = New CSALIDAS
Dim MBUS As String
Dim mdocu As String
PRODUCTO1 NUM_DOC = RTrim(txtDocumento.Text)
MBUS = RTrim(Text1.Text)
PRODUCTO1 BUSCAR (MBUS)
txtCodigo.Text = PRODUCTO1.COD_PROD
txtCod_Sali.Text = PRODUCTO1.COD_SALIDAS
txtDescripcion.Text = PRODUCTO1.DESCRIPCION
txtUm.Text = PRODUCTO1.UM
txtSalidas.Text = PRODUCTO1.SALIDAS
txtFecha_Salidas.Text = PRODUCTO1.FECHA_SALIDAS
txtPrecio_Unit.Text = PRODUCTO1.PRECIO_UNIT
txtPrecio_Total.Text = PRODUCTO1.PRECIO_TOTAL
txtNum_Doc.Text = PRODUCTO1.NUM_DOC
Text2.Text = PRODUCTO1.SALIDAS
Rem HACER VISIBLE LOS CAMPOS
txtCodigo.Visible = True
txtCod_Sali.Visible = True
txtDescripcion.Visible = True
txtUm.Visible = True
txtSalidas.Visible = True
txtFecha_Salidas.Visible = True
txtPrecio_Unit.Visible = True
txtPrecio_Total.Visible = True
txtNum_Doc.Visible = True
lblEncontrado.Visible = True
    
```





```

IblCodigo.Visible = True
IblCod_Sali.Visible = True
IblDescripcion.Visible = True
IblUm.Visible = True
IblSalidas.Visible = True
IblFecha_Salidas.Visible = True
IblPrecio_Unit.Visible = True
IblPrecio_Total.Visible = True
IblNum_Doc.Visible = True
cmdCancelar.Visible = True
cmdActualizar.Visible = False
Frame1.Visible = True
cmdModificar.Enabled = True
cmdEliminar.Enabled = True
Text1.Visible = False
cmdEjecutar.Visible = False
txtDocumento.Visible = False
Text1.Text = " "
Rem txtDescripcion.SetFocus
Frame1.Visible = True
End Sub

```

```

'##ModelId=3FD6A8220398
Private Sub cmdEliminar_Click()
Dim PRODUCTO3 As CSALIDAS
Set PRODUCTO3 = New CSALIDAS
Rem TRASLADAR LOS CAMBIOS A LOS DATOS DE LA CLASE

```





```
PRODUCTO3 COD_PROD = RTrim(txtCodigo Text)
PRODUCTO3 COD_SALIDAS = RTrim(txtCod_Sali Text)
PRODUCTO3 DESCRIPCION = txtDescripcion Text
PRODUCTO3 UM = txtUm.Text
PRODUCTO3 SALIDAS = Val(txtSalidas Text)
PRODUCTO3.FECHA_SALIDAS = txtFecha_Salidas Text
PRODUCTO3 PRECIO_UNIT = Val(txtPrecio_Unit Text)
PRODUCTO3.PRECIO_TOTAL = Val(txtPrecio_Total Text)
PRODUCTO3 NUM_DOC = txtNum_Doc Text
PRODUCTO3 ELIMINAR
lblEliminar.Visible = True
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1.Visible = False
txtCodigo.Visible = False
txtCod_Sali.Visible = False
txtDescripcion.Visible = False
txtUm.Visible = False
txtSalidas.Visible = False
txtFecha_Salidas.Visible = False
txtPrecio_Unit.Visible = False
txtPrecio_Total.Visible = False
txtNum_Doc.Visible = False
lblCodigo.Visible = False
lblCod_Sali.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblSalidas.Visible = False
```





```

IblFecha_Salidas.Visible = False
IblPrecio_Unit.Visible = False
IblPrecio_Total.Visible = False
IblNum_Doc.Visible = False
IblEliminar.Visible = False
IblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdActualizar.Visible = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
Rem Adodc1.Recordset.Update
Rem RS.MoveLast
Rem Adodc1.Recordset.MoveLast
DataGrid1.Refresh
End If
MsgBox ("REGISTRO ELIMINADO EN SALIDAS")
End Sub

```

```

'##ModelId=3FD6A823001E
Private Sub cmdFinal_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset.MoveLast
cmdSiguiente.Enabled = False
cmdFinal.Enabled = False
cmdAnterior.Enabled = True
cmdInicio.Enabled = True

```





DataGrid1 Refresh

End Sub

```

'##ModelId=3FD6A823008C
Private Sub cmdGuardar_Click()
Dim PRODUCTO As CSALIDAS
Set PRODUCTO = New CSALIDAS
PRODUCTO.COD_PROD = LTrim(txtCodigo.Text)
PRODUCTO.COD_SALIDAS = txtCod_Sali.Text
PRODUCTO.DESCRIPCION = txtDescripcion.Text
PRODUCTO.UM = txtUm.Text
PRODUCTO.SALIDAS = Val(txtSalidas.Text)
PRODUCTO.FECHA_SALIDAS = txtFecha_Salidas.Text
PRODUCTO.PRECIO_UNIT = Val(txtPrecio_Unit.Text)
PRODUCTO.PRECIO_TOTAL = Val(txtPrecio_Total.Text)
PRODUCTO.NUM_DOC = Val(txtNum_Doc.Text)
PRODUCTO.GUARDAR
Rem ESCONDEMOS LOS CAMPOS UTILIZADOS
Frame1.Visible = False
txtCodigo.Visible = False
txtCodigo.Enabled = False
txtDescripcion.Visible = False
txtCod_Sali.Visible = False
txtUm.Visible = False
txtSalidas.Visible = False
txtFecha_Salidas.Visible = False
txtPrecio_Unit.Visible = False
    
```





```
txtPrecio_Total.Visible = False
txtNum_Doc.Visible = False
lblCodigo.Visible = False
lblCod_Sali.Visible = False
lblDescripcion.Visible = False
lblUm.Visible = False
lblSalidas.Visible = False
lblFecha_Salidas.Visible = False
lblPrecio_Unit.Visible = False
lblPrecio_Total.Visible = False
lblNum_Doc.Visible = False
lblEtiqueta.Visible = False
lblEncontrado.Visible = False
cmdGuardar.Visible = False
cmdCancelar.Visible = False
cmdModificar.Enabled = False
cmdEliminar.Enabled = False
If Adodc1.Recordset.RecordCount = -1 Then
Else
    DataGrid1.Refresh
End If
Rem RS.MoveLast
Rem Adodc1.Recordset.Update
Rem DataGrid1.Refresh
MsgBox ("REGISTRO ALMACENADO EN SALIDAS")
End Sub
```





```
##ModelId=3FD6A82300FA
Private Sub cmdInicio_Click()
Rem ESTA EN EL INICIO
Adodc1.Recordset MoveFirst
cmdAnterior.Enabled = False
cmdInicio.Enabled = False
cmdSiguiente.Enabled = True
DataGrid1.Refresh
End Sub
```

```
'##ModelId=3FD6A8230168
Private Sub cmdModificar_Click()
Rem MODIFICAR UN REGISTRO
cmdActualizar.Visible = True
lblModificar.Visible = True
lblEncontrado.Visible = False
lblEtiqueta.Visible = False
txtDescripcion.SetFocus
End Sub
```

```
'##ModelId=3FD6A82301D6
Private Sub cmdRetornar_Click()
frmprincipal.Show
Unload Me
End Sub
```





```
'##ModelId=3FD6A8230244
Private Sub cmdSiguiente_Click()
Rem SE CORRE AL SIGUIENTE REGISTRO
Adodc1.Recordset.MoveNext
cmdAnterior.Enabled = True
cmdInicio.Enabled = True
If Adodc1.Recordset.EOF Then
    Adodc1.Recordset.MoveLast
    cmdFinal.Enabled = False
    cmdSiguiente.Enabled = False
End If
DataGrid1.Refresh
End Sub
```

```
'##ModelId=3FD6A82302E4
Private Sub Form_Load()
Set CN = New Connection
Set RS = New Recordset
CN.CursorLocation = adUseClient
CN.ConnectionString = "Provider=MSDASQL 1,Persist Security Info=False;Data
Source=BD_INVENTARIO"
CN.Open
Set RS.ActiveConnection = CN
RS.CursorType = adOpenForwardOnly
RS.Open "select * from SALIDAS"
End Sub
```





'###ModelId=3FD6A8230352

Private Sub Text1_KeyPress(KeyAscii As Integer)

If (KeyAscii = vbKeyReturn) Then

txtDocumento.SetFocus

End If

End Sub

'###ModelId=3FD6A8240082

Private Sub txtCod_Recep_Change()

End Sub

'###ModelId=3FD6A82400F0

Private Sub txtCod_Sali_KeyPress(KeyAscii As Integer)

If (KeyAscii = vbKeyReturn) Then

txtCod_Sali.Text = Adodc3.Recordset.Fields("COD_SALI")

txtSalidas.SetFocus

End If

End Sub

'###ModelId=3FD6A82401FE

Private Sub txtCodigo_KeyPress(KeyAscii As Integer)

If (KeyAscii = vbKeyReturn) Then

Adodc2.Recordset.MoveFirst

Do While Not Adodc2.Recordset.EOF

If Adodc2.Recordset.EOF Then

Exit Do

End If





```

If Trim(Adodc2.Recordset.Fields("COD_PROD")) = Trim(txtCodigo.Text) Then
    Rem MsgBox ("pase")
    txtCodigo.Enabled = False
    txtDescripcion.Enabled = False
    txtUm.Enabled = False
    txtDescripcion.Text = Adodc2.Recordset.Fields("DESCRIPCION")
    txtUm.Text = Adodc2.Recordset.Fields("UM")
    txtNum_Doc.SetFocus
    Exit Do
Else
    Adodc2.Recordset.MoveNext
    If Adodc2.Recordset.EOF Then
        MsgBox ("REGISTRO NO EXISTE, INVESTIGUE POR FAVOR")
        txtCodigo = ""
        Exit Sub
    End If
End If
Loop
txtNum_Doc.SetFocus
End If
End Sub

```

```

'##ModelId=3FD6A824030C
Private Sub txtDocumento_KeyPress(KeyAscii As Integer)
    If (KeyAscii = vbKeyReturn) Then
        cmdEjecutar.SetFocus
    End If

```





```
cmdEjecutar Enabled = True  
End Sub
```

```
'##ModelId=3FD6A825003C  
Private Sub txtFecha_Salidas_KeyPress(KeyAscii As Integer)  
If (KeyAscii = vbKeyReturn) Then  
    txtPrecio_Unit SetFocus  
End If  
End Sub
```

```
'##ModelId=3FD6A825014A  
Private Sub txtNum_Doc_KeyPress(KeyAscii As Integer)  
If (KeyAscii = vbKeyReturn) Then  
    txtCod_Sali SetFocus  
    If (KeyAscii = vbKeyReturn) Then  
        End If  
End Sub
```

```
'##ModelId=3FD6A8250294  
Private Sub txtPrecio_Unit_KeyPress(KeyAscii As Integer)  
If (KeyAscii = vbKeyReturn) Then  
    txtPrecio_Total Text = txtSalidas * txtPrecio_Unit  
    txtNum_Doc SetFocus  
End If  
End Sub
```



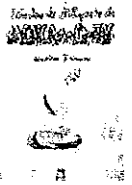


```
###ModelId=3FD6A82503AC
Private Sub txtSalidas_KeyPress(KeyAscii As Integer)
If (KeyAscii = vbKeyReturn) Then
    txtCod_Sali.Text = Adodc3.Recordset.Fields("COD_SALI")
    txtFecha_Salidas.SetFocus
End If
End Sub
```

Este formulario manda a llamar a una clase llamada CSALIDAS y su código es el siguiente

```
'variables locales para almacenar los valores de las propiedades
###ModelId=3FD6A8160154
Private mvarCOD_PROD As String 'copia local
###ModelId=3FD6A81601C2
Private mvarCOD_SALIDAS As String 'copia local
###ModelId=3FD6A8160230
Private mvarDESCRIPCION As String 'copia local
###ModelId=3FD6A81602D0
Private mvarUM As String 'copia local
###ModelId=3FD6A816033E
Private mvarSALIDAS As Double 'copia local
###ModelId=3FD6A8170000
Private mvarFECHA_SALIDAS As Date 'copia local
###ModelId=3FD6A81700A0
Private mvarPRECIO_UNIT As Double 'copia local
###ModelId=3FD6A817010E
Private mvarPRECIO_TOTAL As Double 'copia local
```





'##ModelId=3FD6A81701B8

Private mvarNUM_DOC As Double 'copia local

'##ModelId=3FD6A8170258

Private mvarWEN1 As Double 'copia local

'##ModelId=3FD6A8170302

Private mvarWSA1 As Double 'copia local

'##ModelId=3FD6A81703E0

Private CN As Connection

'##ModelId=3FD6A81703E3

Private RS As Recordset

'##ModelId=3FD6A8180028

Public Property Let WSA1(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax: X WSA1 = 5

 mvarWSA1 = vData

End Property

'##ModelId=3FD6A8180172

Public Property Get WSA1() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

'Syntax. Debug Print X.WSA1

 WSA1 = mvarWSA1

End Property

'##ModelId=3FD6A818021C





Public Property Let WEN1(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X WEN1 = 5

mvarWEN1 = vData

End Property

###ModelId=3FD6A819001E

Public Property Get WEN1() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax. Debug Print X.WEN1

WEN1 = mvarWEN1

End Property

###ModelId=3FD6A81900C8

Public Property Let UM(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X.UM = 5

mvarUM = vData

End Property

###ModelId=3FD6A8190244

Public Property Get UM() As String

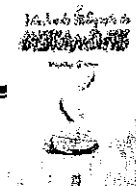
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

'Syntax' Debug Print X UM

UM = mvarUM

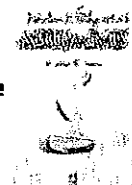
End Property





```
'###ModelId=3FD6A81902E4
Public Sub MODIFICAR()
Set RS.ActiveConnection = CN
Dim WSA2 As Double
Dim WSALIDA As Double
Dim WCODIGO As String
Dim WII As Double
RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS.Open "SELECT * FROM SALIDAS"
RS.MoveFirst
Do While Not RS.EOF
If RS.EOF Then
Exit Sub
Else
If RTrim(RS.Fields("COD_PROD")) = COD_PROD Then
Rem MsgBox ("LO ENCONTRE")
RS.Fields("COD_PROD") = COD_PROD
RS.Fields("COD_SALI") = COD_SALIDAS
RS.Fields("DESCRIPCION") = DESCRIPCION
RS.Fields("UM") = UM
RS.Fields("SALIDAS") = SALIDAS
WSA2 = RS.Fields("SALIDAS")
RS.Fields("FECHA_SALIDA") = FECHA_SALIDAS
RS.Fields("PRECIO_UNIT") = PRECIO_UNIT
RS.Fields("PRECIO_TOTAL") = PRECIO_TOTAL
RS.Fields("NUM_DOC") = NUM_DOC
```





```
RS Update
Exit Do
Rem Else
Rem  MsgBox ("CODIGO = ") & RS Fields("COD_PROD")
End If
RS.MoveNext
End If
Loop
Rem  If (KeyAscii = vbKeyReturn) Then MsgBox ("VALOR WSA1 ") & Str(WSA1)
RS Close
Rem ACTUALIZAR ENTRADAS EN LA TABLA INVENTARIO
Dim WEXISTE As Double
RS Open "SELECT * FROM INVENTARIO ORDER BY COD_PROD"
RS MoveFirst
Rem  MsgBox ("A BUSCAR ") & COD_PROD
Do While Not RS EOF
If RS EOF Then
Exit Sub
Else
If RTrim(RS Fields("COD_PROD")) = RTrim(COD_PROD) Then
Rem  MsgBox ("LO ENCONTRE")
Rem  RS.Fields("COD_PROD") = COD_PROD
WII = RS.Fields("INVE_INI")
WSALIDA = RS.Fields("SALIDAS")
WENTRADA = RS.Fields("ENTRADAS")
WSALIDA = WSALIDA - WSA1 + WSA2
RS.Fields("SALIDAS") = WSALIDA
```





```

RS Fields("EXISTENCIA") = WENTRADA - WSALIDA
If RS.Fields("NIVEL_MINIMO") >= RS.Fields("EXISTENCIA") Then
    MsgBox ("EXISTENCIAS EN NIVEL CRITICO ")
End If
RS Update
Exit Sub
End If
RS MoveNext
End If
Loop
End Sub

```

```

'##ModelId=3FD6A8190352
Public Sub GUARDAR()
Set RS.ActiveConnection = CN
RS LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS Open "SELECT * FROM SALIDAS"
RS.AddNew
    RS.Fields("COD_PROD") = COD_PROD
    RS.Fields("COD_SALI") = COD_SALIDAS
    RS.Fields("DESCRIPCION") = DESCRIPCION
    RS Fields("UM") = UM
    RS Fields("SALIDAS") = SALIDAS
    RS Fields("FECHA_SALIDA") = FECHA_SALIDAS
    RS.Fields("PRECIO_UNIT") = PRECIO_UNIT
    RS Fields("PRECIO_TOTAL") = PRECIO_TOTAL

```





```

RS Fields("NUM_DOC") = NUM_DOC
RS Update
RS Close

Rem ACTUALIZAR SALIDAS EN LA TABLA INVENTARIO
Dim WENTRADA As Double
Dim WSALIDA As Double
Dim WII As Double
RS.Open "SELECT * FROM INVENTARIO ORDER BY COD_PROD"
RS MoveFirst
Rem MsgBox ("A BUSCAR ") & COD_PROD
Do While Not RS.EOF
If RS.EOF Then
Exit Sub
Else
If RTrim(RS Fields("COD_PROD")) = RTrim(COD_PROD) Then
Rem MsgBox ("LO ENCONTRE")
Rem RS Fields("COD_PROD") = COD_PROD
WII = RS.Fields("INVE_INI")
WSALIDA = RS.Fields("SALIDAS")
WENTRADA = RS.Fields("ENTRADAS")
WSALIDA = WSALIDA + SALIDAS
RS Fields("SALIDAS") = WSALIDA
RS Fields("EXISTENCIA") = WII + WENTRADA - WSALIDA
If RS Fields("NIVEL_MINIMO") >= RS.Fields("EXISTENCIA") Then
MsgBox ("EXISTENCIAS EN NIVEL CRITICO ")
End If
RS Update

```





```
Exit Sub
End If
RS MoveNext
End If
Loop
End Sub
```

```
'##ModelId=3FD6A81903C0
```

```
Public Sub CLASS_INITIALIZE()
```

```
Set CN = New Connection
```

```
Set RS = New Recordset
```

```
CN.ConnectionString = "Provider=MSDASQL.1,Persist Security Info=False;Data  
Source=BD_INVENTARIO"
```

```
CN.Open
```

```
End Sub
```

```
'##ModelId=3FD6A81A0046
```

```
Public Sub ELIMINAR()
```

```
Set RS ActiveConnection = CN
```

```
Dim WENTRADA As Double
```

```
Dim WSALIDA As Double
```

```
Dim WII As Double
```

```
Dim WSALI2 As Double
```

```
Dim WCODIGO As String
```

```
RS.LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
```

```
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
```

```
RS.Open "SELECT * FROM SALIDAS"
```





```
RS MoveFirst
Do While Not RS EOF
If RS EOF Then
    Exit Sub
Else
    If RTrim(RS Fields("COD_PROD")) = COD_PROD Then
        Rem MsgBox ("LO ENCONTRE")
        WCODIGO = RS Fields("COD_PROD")
        WSAI2 = RS Fields("SALIDAS")
        RS.Delete
        Exit Do
    Rem Else
        Rem MsgBox ("CODIGO = ") & RS Fields("COD_PROD")
    End If
    RS MoveNext
End If
Loop
RS Close
Rem ACTUALIZAR SALIDAS EN LA TABLA INVENTARIO
RS Open "SELECT * FROM INVENTARIO ORDER BY COD_PROD"
RS.MoveFirst
Rem MsgBox ("A BUSCAR ") & WCODIGO
Do While Not RS EOF
If RS EOF Then
    Exit Sub
Else
    If RTrim(RS.Fields("COD_PROD")) = RTrim(WCODIGO) Then
```

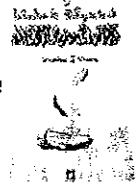




```
Rem MsgBox ("LO ENCONTRE")
Rem RS Fields("COD_PROD") = COD_PROD
WII = RS Fields("INVE_INI")
WSALIDA = RS Fields("SALIDAS")
WENTRADA = RS Fields("ENTRADAS")
WSALIDA = WSALIDA - WSALI2
RS Fields("SALIDAS") = WSALIDA
RS Fields("EXISTENCIA") = WII + WENTRADA - WSALIDA
RS Update
Exit Sub
End If
RS.MoveNext
End If
Loop
End Sub

'##ModelId=3FD6A81A00B4
Public Sub BUSCAR(MBUSCA As String)
Dim MBUSCAR As String
MBUSCAR = MBUSCA
Set RS ActiveConnection = CN
RS LockType = adLockOptimistic ' NECESARIO PARA LEER Y ESCRIBIR
RS.CursorType = adOpenKeyset ' ACTIVA EL CURSOR
RS.Open "SELECT * FROM SALIDAS order by COD_PROD "
Rem MsgBox (" 1 MBUSCAR = ") & MBUSCAR
RS.MoveFirst
Do While Not RS.EOF
```





If RS EOF Then

Exit Sub

Else

If RTrim(RS Fields("COD_PROD")) = MBUSCAR And RTrim(RS Fields("num_doc")) =
RTrim(NUM_DOC) Then

Rem MsgBox ("LO ENCONTRE")

COD_PROD = RS.Fields("COD_PROD")

COD_SALIDAS = RS Fields("COD_SALI")

DESCRIPCION = RS Fields("DESCRIPCION")

UM = RS.Fields("UM")

SALIDAS = RS Fields("SALIDAS")

WSA1 = RS.Fields("SALIDAS") ' ESTO ES PARA ACTUALIZAR INVENTARIO

FECHA_SALIDAS = RS Fields("FECHA_SALIDA")

PRECIO_UNIT = RS Fields("PRECIO_UNIT")

PRECIO_TOTAL = RS.Fields("PRECIO_TOTAL")

NUM_DOC = RS.Fields("NUM_DOC")

Exit Sub

Else

Rem MsgBox ("CODIGO = ") & RS Fields("COD_PROD")

End If

RS MoveNext

End If

Loop

End Sub

'##ModelId=3FD6A81A01FE

Public Property Let NUM_DOC(ByVal vData As Double)





'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X NUM_DOC = 5

mvarNUM_DOC = vData

End Property

###ModelId=3FD6A81A0384

Public Property Get NUM_DOC() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax Debug Print X NUM_DOC

NUM_DOC = mvarNUM_DOC

End Property

###ModelId=3FD6A81B0078

Public Property Let PRECIO_TOTAL(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.

'Syntax: X.PRECIO_TOTAL = 5

mvarPRECIO_TOTAL = vData

End Property

###ModelId=3FD6A81B0230

Public Property Get PRECIO_TOTAL() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

'Syntax. Debug.Print X.PRECIO_TOTAL

PRECIO_TOTAL = mvarPRECIO_TOTAL

End Property

###ModelId=3FD6A81B030C

Public Property Let PRECIO_UNIT(ByVal vData As Double)





'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax X PRECIO_UNIT = 5  
    mvarPRECIO_UNIT = vData  
End Property
```

```
###ModelId=3FD6A81C00DC
```

```
Public Property Get PRECIO_UNIT() As Double
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

```
'Syntax Debug Print X PRECIO_UNIT  
    PRECIO_UNIT = mvarPRECIO_UNIT  
End Property
```

```
###ModelId=3FD6A81C01B8
```

```
Public Property Let FECHA_SALIDAS(ByVal vData As Date)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

```
'Syntax X FECHA_SALIDAS = 5  
    mvarFECHA_SALIDAS = vData  
End Property
```

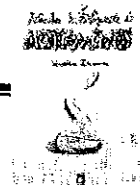
```
###ModelId=3FD6A81C0370
```

```
Public Property Get FECHA_SALIDAS() As Date
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax: Debug Print X.FECHA_SALIDAS  
    FECHA_SALIDAS = mvarFECHA_SALIDAS  
End Property
```





'##ModelId=3FD6A81D0096

Public Property Let SALIDAS(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X.SALIDAS = 5

mvarSALIDAS = vData

End Property

'##ModelId=3FD6A81D0280

Public Property Get SALIDAS() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

'Syntax. Debug.Print X SALIDAS

SALIDAS = mvarSALIDAS

End Property

'##ModelId=3FD6A81D0398

Public Property Let DESCRIPCION(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax X DESCRIPCION = 5

mvarDESCRIPCION = vData

End Property

'##ModelId=3FD6A81E019A

Public Property Get DESCRIPCION() As String

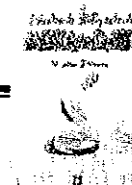
'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax Debug Print X.DESCRIPCION

DESCRIPCION = mvarDESCRIPCION

End Property





'##ModelId=3FD6A81D0096

Public Property Let SALIDAS(ByVal vData As Double)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax. X.SALIDAS = 5

 mvarSALIDAS = vData

End Property

'##ModelId=3FD6A81D0280

Public Property Get SALIDAS() As Double

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

'Syntax: Debug Print X.SALIDAS

 SALIDAS = mvarSALIDAS

End Property

'##ModelId=3FD6A81D0398

Public Property Let DESCRIPCION(ByVal vData As String)

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación

'Syntax. X DESCRIPCION = 5

 mvarDESCRIPCION = vData

End Property

'##ModelId=3FD6A81E019A

Public Property Get DESCRIPCION() As String

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

'Syntax. Debug Print X DESCRIPCION





```
DESCRIPCION = mvarDESCRIPCION
End Property
```

```
###ModelId=3FD6A81E02B2
```

```
Public Property Let COD_SALIDAS(ByVal vData As String)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.

```
'Syntax X.COD_SALIDAS = 5
```

```
    mvarCOD_SALIDAS = vData
```

```
End Property
```

```
###ModelId=3FD6A81F00B4
```

```
Public Property Get COD_SALIDAS() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación

```
'Syntax Debug Print X.COD_SALIDAS
```

```
    COD_SALIDAS = mvarCOD_SALIDAS
```

```
End Property
```

```
###ModelId=3FD6A81F01CC
```

```
Public Property Let COD_PROD(ByVal vData As String)
```

'se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.

```
'Syntax: X COD_PROD = 5
```

```
    mvarCOD_PROD = vData
```

```
End Property
```

```
###ModelId=3FD6A820000A
```

```
Public Property Get COD_PROD() As String
```

'se usa al recuperar un valor de una propiedad, en la parte derecha de una asignación.

```
'Syntax Debug.Print X.COD_PROD
```





COD_PROD = mvarCOD_PROD

End Property

SALIDAS							
COD_PROD	NUM_DOC	COD_SAL	DESCRIPCION	UM	SALIDAS	FECHA S	
01	12121	01	Aspirina	Cap	2	12/12/03	
01	12121	01	Aspirina	Cap	1	12/12/03	

AGREGAR MODIFICAR ELIMINAR INICIO SIGUIENTE ANTERIOR FINAL BUSCAR RETORNAR

AGREGAR PRODUCTOS

NOMBRE DEL PRODUCTO			
CODIGO DE PRODUCTO			
CODIGO SALIDA	Bonifica	PRECEDENTE	
DESCRIPCION		PRECEDENTE	
UNIDAD DE MEDIDA		PRECEDENTE	

GUARDAR CANCELAR

Quiero aclarar que en esta parte de la codificacion solamente tome los formularios que considere más importante con sus respectivas clases y código.





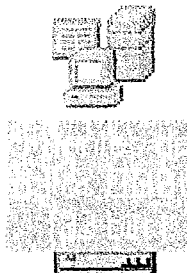
Las características útiles de este Sistema Cliente/Servidor es que las bases de datos suelen proporcionar potentes funciones que no se encuentran en las bases de escritorio como Acces, FoxPro, etc. Estas funciones no sólo automatizan la administración de la base de datos, sino que también proporcionan potentes posibilidades de desarrollo de aplicaciones a como fue mi caso donde utilice un proyecto DLL para incluir todas las clases.

Para realizar la conexión de la base de datos del Servidor con el cliente utilizamos los componentes **ODBC, ADODC**

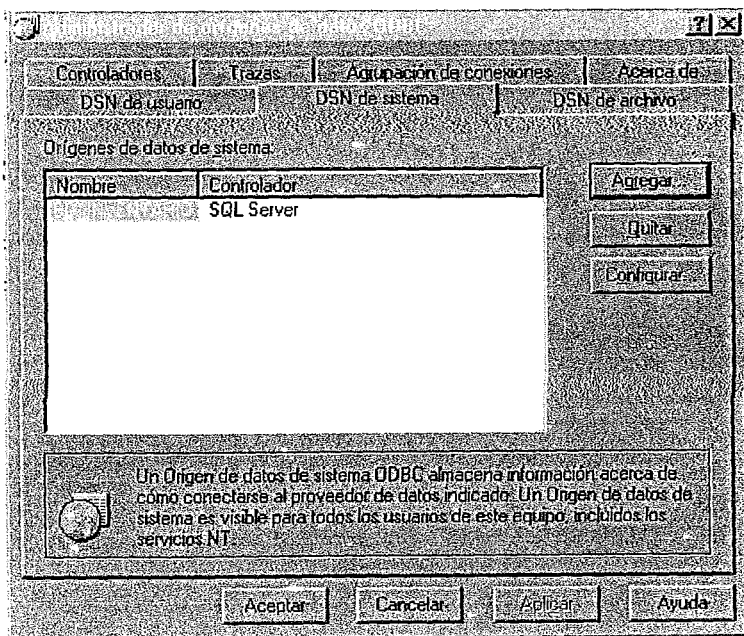
El componente **ODBC** ofrece mediante la definición de una API en el lado del cliente que es independiente de la base de datos que haya en el otro extremo. Gracias a su controlador **ODBC** es posible acceder a la base de datos de forma transparente.

Otro estándar es **SQL Server 6.0**, aunque lleva utilizándose algo más de tiempo, es de vital importancia en el desarrollo de aplicaciones cliente/servidor.

La conexión **ODBC** se hace entrando al panel de control entrando en:

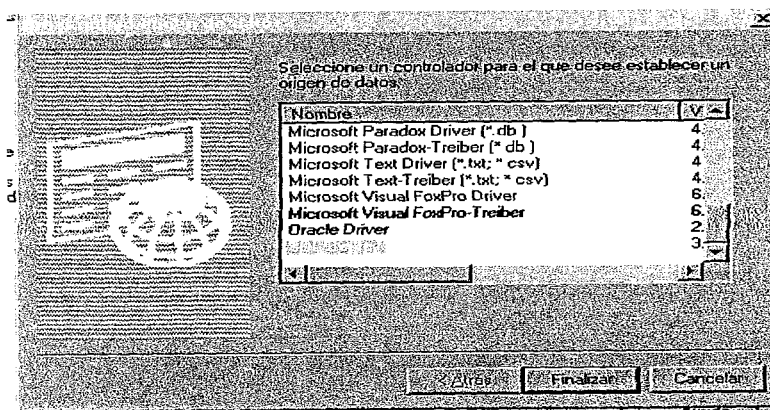


Para comprender un poco mejor la forma en que se conecta la base de datos del servidor con la aplicación del cliente observe la siguiente figura:



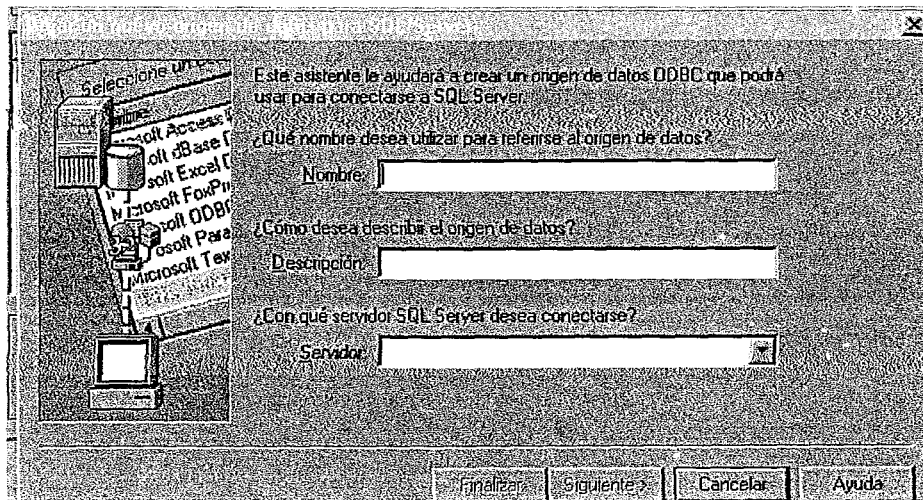
Se le da Agregar:

:

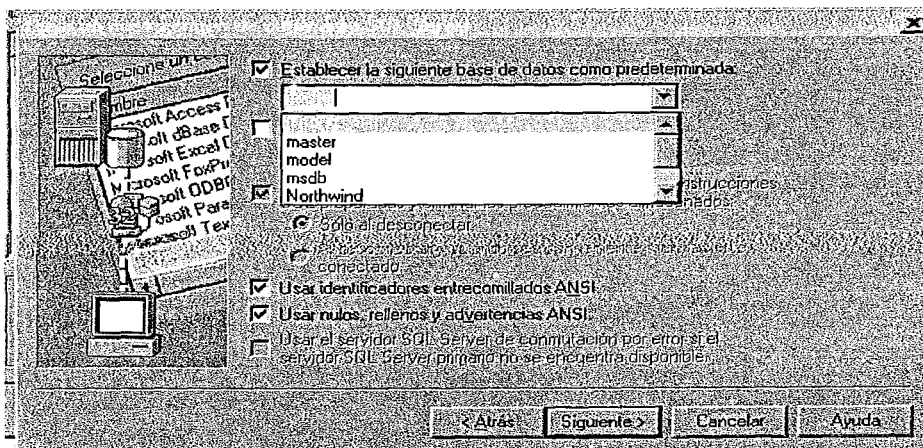




Se le da Finalizar para luego crear un nuevo origen de datos para SQL Server, hay que introducir un nombre para referirse al origen de datos y luego el servidor donde se encuentra la base de datos.

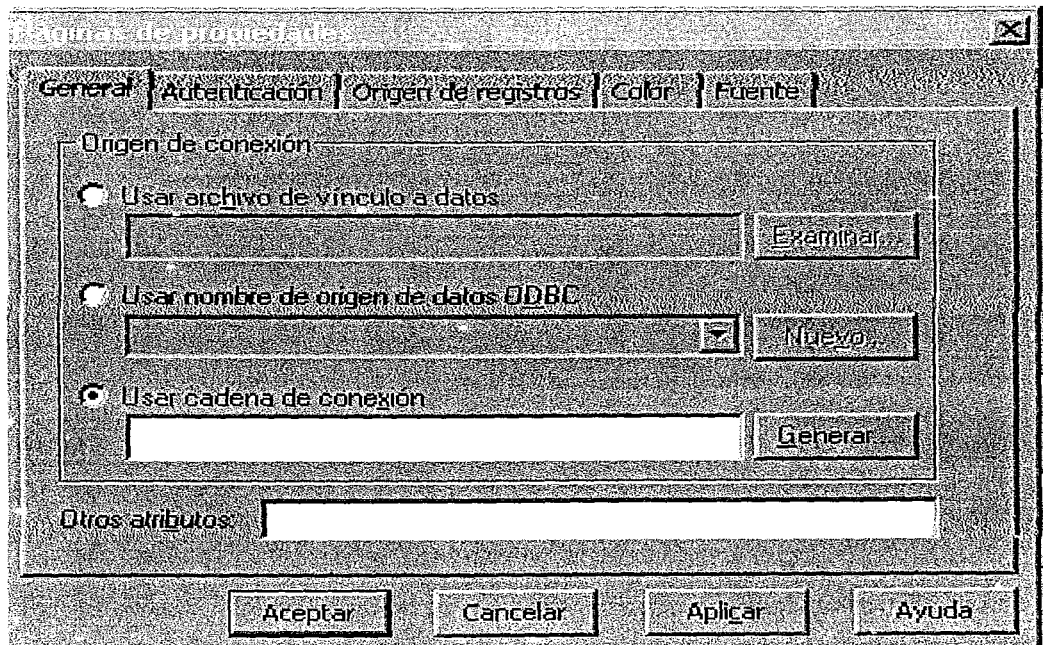


Se le da siguiente y escoge la base de datos como se muestra en la siguiente figura:



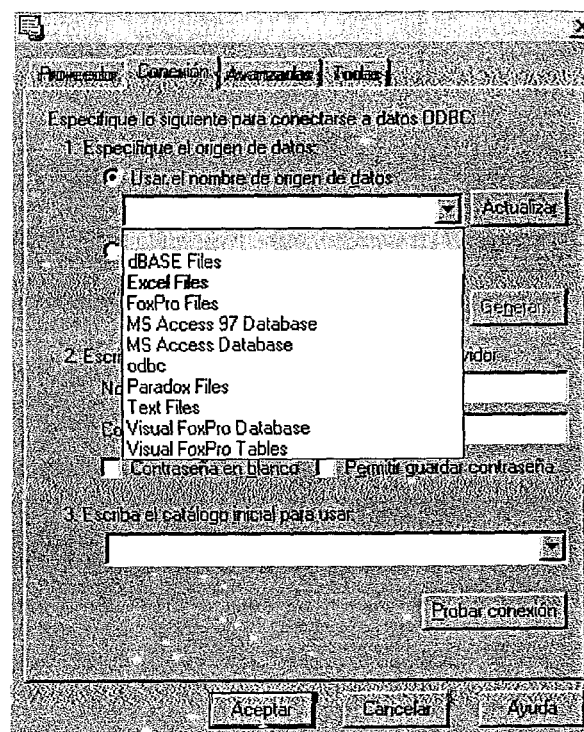
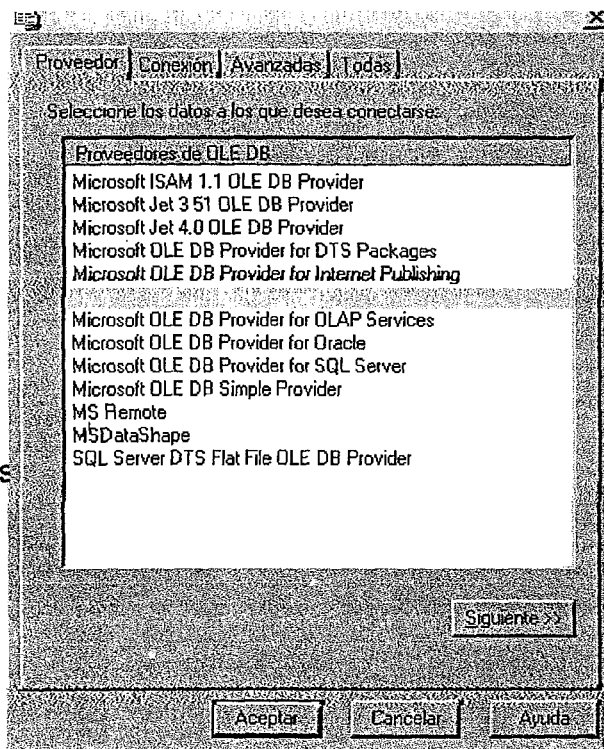


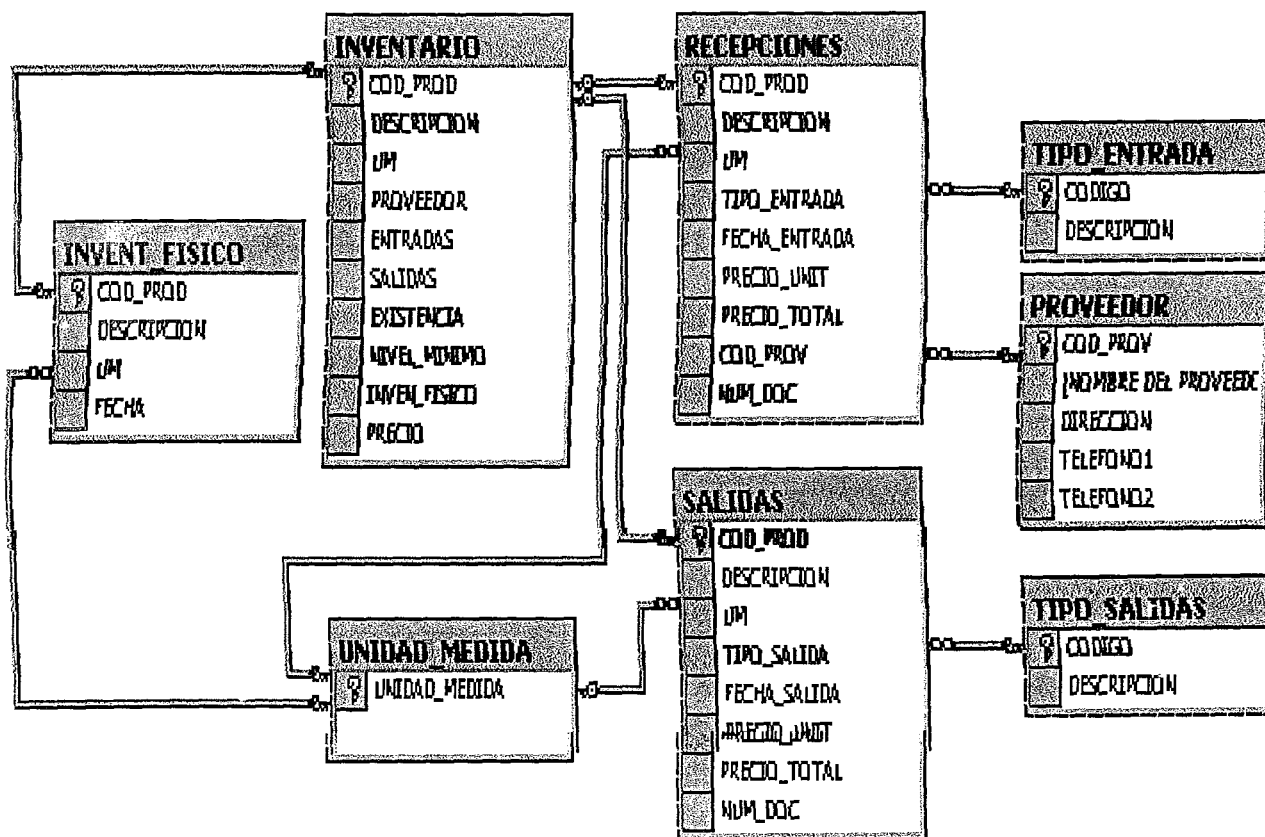
Una vez que se ha realizado la conexión **ODBC** entramos a nuestro proyecto en Visual Basic y conectamos la conexión con el componente **ADODC** dando con el boton derecho del mouse sobre el componente:

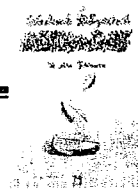


Tenemos que activar: "Usar cadena de conexión" y generar .









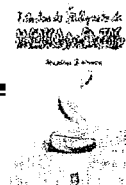
3. CONCLUSIONES

La elaboración de este proyecto para la Empresa **DIFERCAL S.A.** permitira una mejor ayuda al control de Inventario, asi como un medio para la solución de problemas de la Empresa.

Se identificaron gran parte de los elementos de información del Inventario por medio de las diferentes técnicas de recolección de datos que permitieron elaborar el conjunto de elementos del Sistema.

La implementación de este proyecto ayudara a la Empresa a disminuir sus perdidas monetarias y de factor tiempo, ya que es Sistema de Inventario permitira un control exacto de sus productos, lo que permitira ser una herramienta muy útil en la toma de decisiones.





4. RECOMENDACIONES

Se recomienda a la Empresa **DIFERCAL S.A.** aplicar la automatización del Sistema de Inventario de Productos Farmaceuticos par un mejor control de los mismos.

Realizar controles de Inventario Físico en períodos regulares que la empresa considere conveniente para su analisis contra los resultados obtenidos por el Sistema Automatizado de Inventario.

Preparar al personal que maneja el Sistema Automatizado de Inventario para un uso optimo del mismo.

Utilizar el Sistema Automatizado de Inventario par la elaboración de otros tipos de reportes e informes que ayuden para la buena toma de decisiones en la politica de inventario de la empresa.





5. BIBLIOGRAFIA

- ✓ **Aprenda Programación en Microsoft SQL Server 2000 Ya.**
Rebecca M. Riordan.
Mc Graw Hill.
Edición 2001.
- ✓ **El Lenguaje Unificado de Modelado. Manual de Referencia.**
James Rumbaugh.
Ivar Jacobson.
Grady Booch.
Addison Wesley.
Edición 2002.
- ✓ **Ingenieria de Software Orientado a Objetos.**
Bernd Bruegge.
Allen H. Dutoit.
Prentice Hall.
Edición 2002.
- ✓ **Ingenieria del Software un enfoque práctico**
Roger S. Pressman.
5ta Edición.
Mc Graw Hill.





- ✓ Análisis y Diseño de Sistemas.
Mc Graw Hil.

- ✓ Aprendiendo Visual Basic 6 en 21 días.
Pearson.
Prentice Hall.

- ✓ Programación Cliente/Servidor con Microsoft Visual Basic.
Kenneth L. Spencer
Ken Miller.
Mc Graw Hill.

- ✓ Programación de bases de datos con Microsoft Visual Basic.
Alfons González.
CompuTec.

- ✓ Bases de Datos con Visual Basic.
Jeffrey P. McManus.
Sams.





ANEXOS.



MADURACION DEL PROBLEMA Y DEL PROCESO

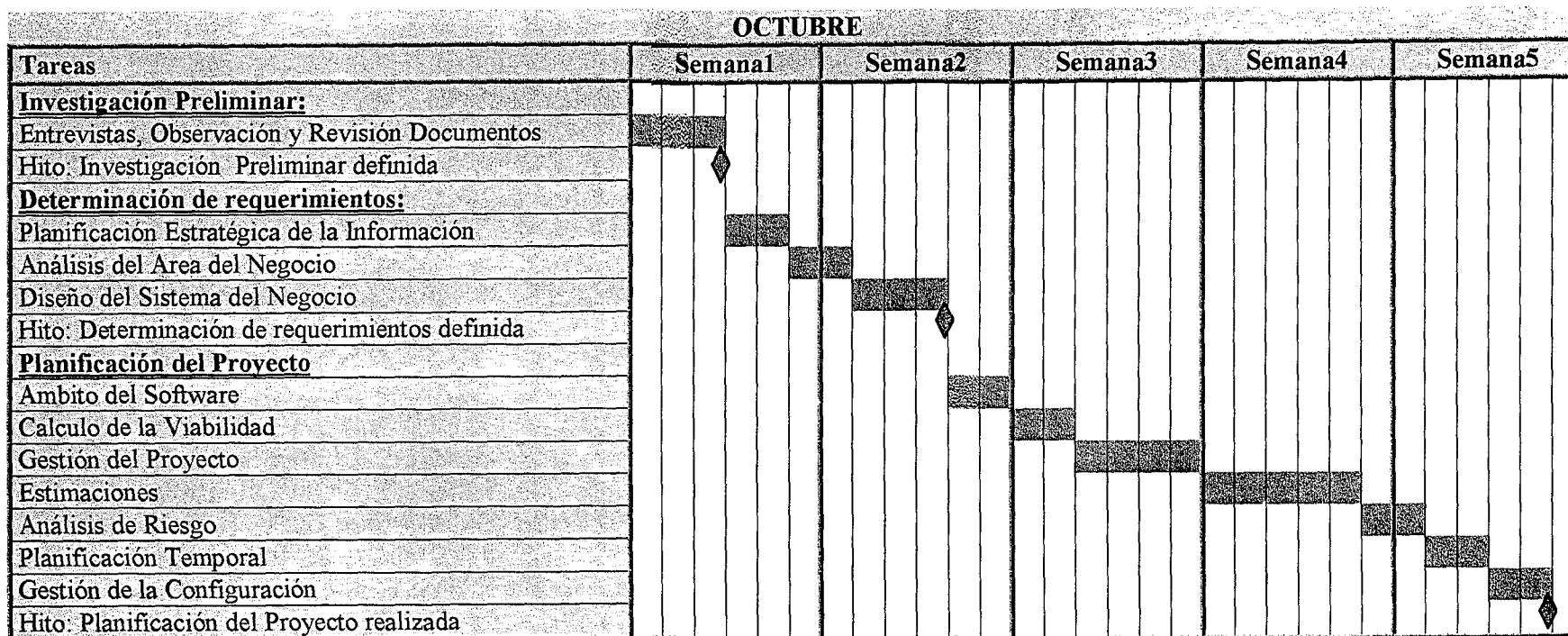
ACTIVIDADES ESTRUCTURALES DE PROCESOS COMUNES	Comunicación con el Cliente		Planificación		Análisis de riesgo		Ingeniería	
	Inicio	Final	Inicio	Final	Inicio	Final	Inicio	Final
Investigación Preliminar:								
Entrevistas, Observación y Revisión Documentos	01-Oct-03	03-Oct-03						
Determinación de requerimientos								
Planificación Estratégica de la Información			04-Oct-03	05-Oct-03				
Análisis del Area del Negocio			06-Oct-03	07-Oct-03				
Diseño del Sistema del Negocio			08-Oct-03	10-Oct-03				
Ambito del Software			11-Oct-03	12-Oct-03				
Calculo de la Viabilidad			13-Oct-03	14-Oct-03				
Gestión del Proyecto								
Personal (Selección del grupo de trabajo)			15-Oct-03	15-Oct-03				
Problema (Descomposición del Problema)			15-Oct-03	16-Oct-03				
Proceso (Selección y Maduración)			17-Oct-03	19-Oct-03				
Proyecto			20-Oct-03	20-Oct-03				
Estimaciones			21-Oct-03	25-Oct-03				
Análisis de Riesgos					25-Oct-03	26-Oct-03		
Planificación Temporal			26-Oct-03	27-Oct-03				
Gestión de la Configuración (Seguimiento)			28-Oct-03	29-Oct-03				
Diseño del Sistema:								
Diseño Conceptual							30-Oct-03	01-Nov-03
Diseño Lógico							02-Nov-03	04-Nov-03
Diseño Físico y Despliegue							05-Nov-03	08-Nov-03
Desarrollo del Sistema								
Codificación							09-Nov-03	28-Nov-03
Prueba del Sistema								
Pruebas							29-Nov-03	01-Dic-03
Implementación y Evaluación								
Entrenamiento a los Usuarios							02-Dic-03	10-Dic-03
Instalación de la Aplicación							11-Dic-03	11-Dic-03

UNIVERSIDAD DE CIENCIAS COMERCIALES





GRAFICA DE TIEMPO (GANTT)



UNIVERSIDAD DE CIENCIAS COMERCIALES





NOVIEMBRE																									
Tareas	Semana1					Semana2					Semana3					Semana4					Semana5				
Diseño del Sistema:																									
Diseño Conceptual																									
Diseño Lógico																									
Diseño Físico y Despliegue																									
Hito: Diseño del Sistema elaborado																									
Desarrollo del Sistema																									
Codificación																									
Hito: Desarrollo del Sistema elaborado																									
Prueba del Sistema																									
Pruebas																									
Hito: Prueba del Sistema realizada																									

DICIEMBRE																									
Tareas	Semana1					Semana2					Semana3					Semana4					Semana5				
Implementación y Evaluación																									
Entrenamiento a los Usuarios																									
Instalación de la Aplicación																									
Hito: Implementación y Evaluación realizada																									

UNIVERSIDAD DE CIENCIAS COMERCIALES



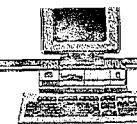


TABLA DE PROYECTO

Tareas de trabajo	Inicio previsto	Inicio real	Terminación prevista	Terminación real	Personas asignadas	Esfuerzo asignado	observaciones
<u>Investigacion Preliminar</u>							
Entrevistas, Observaciones y Revision Documentos	Sem1, d1	Sem1, d1	Sem1, d3	Sem1, d3	MJMM	1 p = d	
Hito: Investigacion Preliminar definida	Sem1, d1	Sem1, d1	Sem1, d3	Sem1, d3			
<u>Determinacion de requerimientos:</u>							
Planificacion Estrategica de la Información	Sem1, d4	Sem1, d4	Sem1, d5	Sem1, d5	MJMM	1 p = d	
Analisis del Area del Negocio	Sem1, d6	Sem1, d6	Sem2, d1	Sem2, d1	MJMM	1 p = d	
Diseño del Sistema del Negocio	Sem2, d2	Sem2, d2	Sem2, d4	Sem2, d4	MJMM	1 p = d	
Hito: Determinacion de requerimientos definida	Sem2, d2	Sem2, d2	Sem2, d4	Sem2, d4			
<u>Planificacion del Provento:</u>							
Ambito del Software	Sem2, d5	Sem2, d5	Sem2, d6	Sem2, d6	MJMM	1 p = d	
Calculo de la Viabilidad	Sem3, d1	Sem3, d1	Sem3, d2	Sem3, d2	MJMM	1 p = d	
Gestion del Proyecto	Sem3, d3	Sem3, d3	Sem3, d6	Sem3, d6	MJMM	1 p = d	Es previsible que requiera de más esfuerzo / tiempo
Estimaciones	Sem4, d1	Sem4, d1	Sem4, d5	Sem4, d5	MJMM	1 p = d	
Analisis de Riesgo	Sem4, d6	Sem4, d6	Sem5, d1	Sem5, d1	MJMM	1 p = d	
Planificacion Temporal	Sem5, d2	Sem5, d2	Sem5, d3	Sem5, d3	MJMM	1 p = d	
Gestion de la Configuracion	Sem5, d4	Sem5, d4	Sem5, d5	Sem5, d5	MJMM	1 p = d	
Hito: Planificacion del Proyecto realizada	Sem5, d4	Sem5, d4	Sem5, d5	Sem5, d5			

UNIVERSIDAD DE CIENCIAS COMERCIALES

