

Escuela Superior Politecnica del Litoral

FACULTAD DE INGENIERIA ELECTRICA
Y COMPUTACION

Sistema de Inventario Cliente / Servidor utilizando la
tecnología de Java y JDBC para la conexión
a bases de datos

Tópico de Graduación:

Arquitectura Cliente / Servidor

Presentado por:

Raúl Cordero Carrasco

Lia Naranjo Mena

Raynaud Quiroz Congrains

GUAYAQUIL - ECUADOR

AGOSTO 1997

DEDICATORIA

A NUESTROS PADRES

A NUESTROS HERMANOS

A NUESTROS PROFESORES

AGRADECIMIENTO

ING. CARLOS VALERO

Profesor de Tópico de graduación, por su gran ayuda, paciencia y colaboración en la realización de este proyecto.

INDICE

	Pag
INDICE GENERAL	IV
INDICE DE FIGURAS	VII
1. INTRODUCCION	8
2. OBJETIVOS	8
3. PLATAFORMA.....	10
3.1 REQUISITOS DEL SERVIDOR.....	11
3.2 REQUISITOS DEL CLIENTE.....	11
3.3 ELEMENTOS DE SOFTWARE UTILIZADOS.....	11
4. LENGUAJE JAVA.....	12
4.1 VENTAJAS DEL LENGUAJE JAVA.....	13
5. EL SERVIDOR IDS.....	15
5.1 CASOS DE CONFIGURACIÓN.....	17
5.1.1 <i>Caso 1</i>	17
5.1.2 <i>Caso 2</i>	18
5.1.3 <i>Caso 3</i>	19
5.2 INSTALACIÓN DEL SERVIDOR IDS.....	20

5.3	INSTALACIÓN DEL CONTROLADOR ODBC DE LA BASE DE DATOS	22
5.4	CONFIGURACIÓN DEL SERVIDOR IDS	22
6.	CONECTIVIDAD A BASE DE DATOS: JDBC	23
7.	DISEÑO DE LA BASE DE DATOS	26
7.1	DIAGRAMA ENTIDAD RELACIÓN	27
7.2	DESCRIPCIÓN DE LAS TABLAS	29
8.	MANUAL DEL USUARIO	34
8.1	INTRODUCCIÓN AL SISTEMA DE CONTROL DE INVENTARIO	34
8.2	USO DEL SISTEMA DE CONTROL DE INVENTARIO	35
8.2.1	<i>Opciones que maneja el sistema de inventario</i>	35
8.2.2	<i>Ingresar una bodega o sucursal</i>	35
8.2.3	<i>Actualizar datos de una sucursal o bodega</i>	36
8.2.4	<i>Ingresar productos, marcas, grupos, unidades o características</i>	36
8.2.5	<i>Actualizar valores de productos</i>	37
8.2.6	<i>Consultar productos en las diversas bodegas</i>	37
8.2.7	<i>Consultar el Kardex</i>	38
8.2.8	<i>Realizar comprobantes de ingreso y egreso</i>	39
8.3.	RESPUESTAS A PREGUNTAS FRECUENTES	40
8.3.1.	<i>¿Qué son los grupos?</i>	40
8.3.2.	<i>¿Como me puedo cambiar a otra bodega?</i>	40

9. CONCLUSIONES.....	40
APENDICE	45
BIBLIOGRAFIA	69

INDICE DE FIGURAS

	Pag.
Figura # 1 Arquitectura de la aplicación.....	9
Figura # 2 Caso 1 de configuración del Servidor IDS.....	18
Figura # 3 Caso 2 de configuración del servidor IDS.....	19
Figura # 4 Caso 3 de configuración del servidor IDS.....	20
Figura # 5 Directorio del Servidor IDS.....	21
Figura # 5 Directorio del Servidor IDS.....	21
Figura # 6 ODBC.....	25

1. INTRODUCCION

El rápido desarrollo de la tecnología de **Internet e Intranet** ha mostrado un nuevo desafío y oportunidades para las aplicaciones de base de datos. La flexibilidad de los documentos **HTML**, también conocidos como el enlace del **World Wide Web**, ha evolucionado desde las publicaciones de texto hasta la conexión hacia base de datos.

La gran aceptación de Java y la introducción de APIs para conexión a base datos (**JDBC**) han provisto de un mapa para las aplicaciones de base de datos en Internet. Usando estas dos tecnologías, es ahora posible construir muy poderosos programas de base de datos para Internet.

La interfaz gráfica de los programas hacen ver y sentir a las aplicaciones Java con interacciones mas sofisticadas y atractivas con ayuda de **JDBC** que las páginas Web estáticas.

Realizar un **SISTEMA DE INVENTARIO**, utilizando la nueva tecnología de la que dispone JAVA para el enlace con bases de datos, conocido como **JDBC**, que permita un control de productos en diversas sucursales así como en sus respectivas bodegas - sucursal.

**SERVIDOR IDS /
SERVIDOR WEB**

CLIENTE WEB

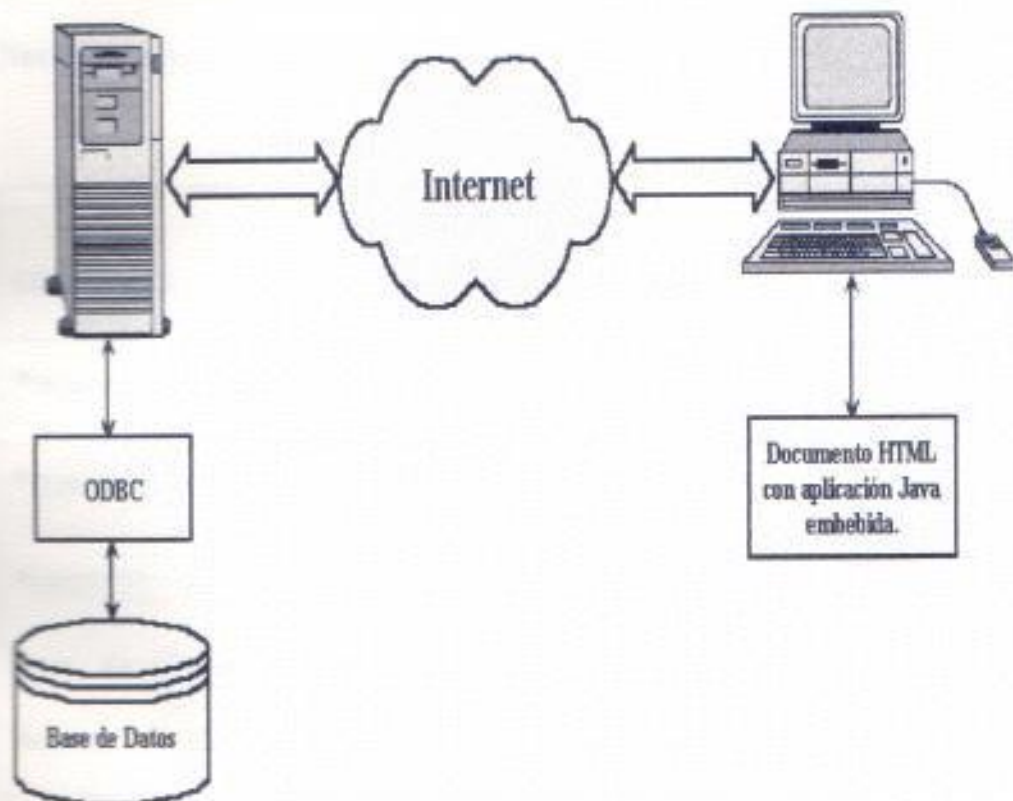


Figura # 1 Arquitectura de la aplicación

3. PLATAFORMA

La aplicación se desarrollo en lenguaje **JAVA** sobre la plataforma **IDS SERVER** para Windows NT/95 , en el lado servidor debido a su fácil disponibilidad, uso, instalación y administración.

Para ejecutar el lado cliente de la aplicación se requirió del **Browser Netscape Navigator** debido a requisitos de software de IDS SERVER.

SERVIDOR	CLIENTE
<ul style="list-style-type: none">• Windows NT/95.• IDS SERVER versión 1.2 Beta.• ODBC32. Base de Datos Cinventario32.mdb Access	<ul style="list-style-type: none">• Windows NT/95.• Netscape Navigator versión 3.0.

3.1 Requisitos del servidor

- Sistema Operativo Windows 95 /NT.
- 16MB de memoria RAM.
- 10MB de espacio libre en disco duro.
- ODBC 32 bit.
- JDBC IDSSERVER.
- Conexión a Internet o a una Intranet.

3.2 Requisitos del cliente

- 16 MB de memoria RAM.
- Browser Netscape Navigator 3.0 o mayor.
- Conexión a la red del servidor.
- Conexión a Internet o a una Intranet donde se provea el servicio.

3.3 Elementos de software utilizados

- Kit de desarrollo: **JDK 1.0.2**

- Web Server: **IDS SERVER versión 1.2 Beta.**
- Editor: **Symantec Café Lite**
- Utilitario: **VBtoJava** que permite que formas en Visual Basic sean pasadas a lenguaje Java.
- Librerías de clases:
 - Estándar de Java
 - SQL (Conexión a base de datos)
 - Objetos Tea-set encontrados en Internet.
 - VE.java grid para el objeto cuadrícula.

4. LENGUAJE JAVA

La definición del lenguaje JAVA dada por : **Sun Microsystems** lo describe como:

- Simple, parecido de muchas formas al C++.
- Orientado a objetos como el C++ excepto que solo soporta herencia simple.
- Interpretado porque no se ejecuta directamente sobre el procesador, sino sobre una máquina virtual.
- Seguro porque existen restricciones de diseño para los applets implementados en JAVA.
- Portable por su característica de ser multiplataforma.

- Multithread ya que soporta mas de un thread ejecutándose al mismo tiempo

4.1 Ventajas del lenguaje Java

Java cuenta con una serie de **ventajas** tales como:

- Provee de plataforma independiente, necesaria para la programación en Internet esto es, puede usar el mismo código para Windows 95, Unix, Macintosh etc.
- Es simple, puesto que no necesita archivos de cabecera, punteros aritméticos, elimina la colocación manual de memoria, etc. Es un lenguaje orientado a objetos que permite la construcción de software que pueda correr por el solo en pequeñas máquinas.
- La técnica orientada a objetos se enfoca en el diseño de los datos (=objetos) y de las interfaces de estos.

- JAVA cuenta con capacidades de red que lo convierten en fuerte y de fácil uso. Las aplicaciones pueden abrir y acceder a través de la red vía URLs con igual facilidad que cuando accesan a un sistema de archivo.
- Es robusto, puesto permite la escritura de programas en una gran variedad de formas, pone énfasis en el chequeo temprano de posibles problemas y elimina situaciones de error.
- Su compilador genera formatos de archivos objetos de arquitectura neutra, esto quiere decir que el código compilado puede ser ejecutado en muchos procesadores. El compilador genera instrucciones las cuales han sido diseñadas para ser de fácil interpretación en cualquier máquina y fácilmente traducida en código nativo de maquina en vuelo.
- Es un lenguaje interpretado, esto es que puede ser ejecutado en cualquier máquina la cual contenga el interpretador. Sin embargo esto le da características mas lento que los lenguajes nativos pero la rapidez de las computadoras cada día disminuye esta desventaja.

- **JAVA** promete convertirse en el enlace universal que conecte a los usuarios con información desde servidores web, base de datos, información de sus proveedores y algunas otras fuentes inimaginables.

5. EL SERVIDOR IDS

El servidor IDS provee de **acceso a base de datos** para Internet. Este permite que ambos documentos HTML y programas JAVA desarrollen páginas Web interactivas con acceso a base de datos.

El servidor IDS provee de una solución mediante HTML y mediante JDBC para al acceso a aplicaciones de base de datos. Actualmente, el servidor IDS soporta acceso a todos los sistemas ODBC de base de datos.

El servidor IDS introduce un pequeño conjunto de nuevas etiquetas HTML llamadas extensiones IDS HTML. Dentro del documento HTML, estas extensiones definen como acceder a base de datos y como incorporar el resultado en el documento HTML. Un documento que contiene estas extensiones es llamado archivo HTX.

Estas extensiones pueden usarse con características avanzadas HTML como: tablas, frames, JavaScript y Applet Java para construir **deslumbrantes Websites**. El servidor IDS viene además con el **driver IDS JDBC**. Las aplicaciones Java pueden usar el **API JDBC** para acceder a la base de datos y conectarse con el servidor IDS.

El driver IDS JDBC tiene un alto desempeño y es extremadamente compacto y eficiente. Esto se debe a que la mayor parte de los accesos a la base de datos, son realizados por el servidor IDS el cual es 100% código binario nativo el cual es mucho más rápido que su contraparte Java.

El servidor IDS no está diseñado para ser un servidor Web con todas sus características; sin embargo puede servir como un servidor Web simple.

El servidor IDS elimina el peso en el diseño del servidor Web, evitando la competencia por tiempo del CPU y otros recursos, además de no requiere de que el servidor procese todos los datos. Puesto que este a diferencia de CGI, ISAPI y NSAPI corre como un programa TCP/IP utilizando una dirección IP y un puerto determinados.

Este utiliza el protocolo HTTP, el mismo protocolo usado por los servidores Web para todas las comunicaciones. Los clientes se conectan directamente al servidor

IDS durante el acceso a base de datos, relevando al web server de cualquier actividad relacionada con bases de datos. Esta característica hace posible para el servidor IDS corra en ambientes centralizados y distribuidos de Internet e Intranet.

El servidor IDS envuelve tres entidades de software :

- El servidor Web
- El servidor de Base de datos
- El Web browser.

Estas entidades pueden estar en una sola computadora o hasta en tres máquinas separadas.

5.1 Casos de configuración

5.1.1 Caso 1

A continuación se muestra la configuración en donde todas las entidades se encuentran en la misma computadora:

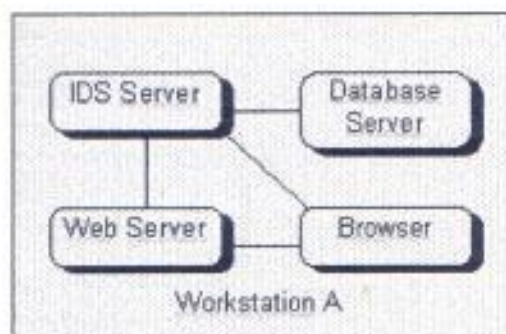


Figura # 2 Caso 1 de configuración del Servidor IDS

5.1.2 Caso 2.

Representa el modelo distribuido, en este modelo el servidor IDS es instalado en la misma máquina que el servidor de base de datos.

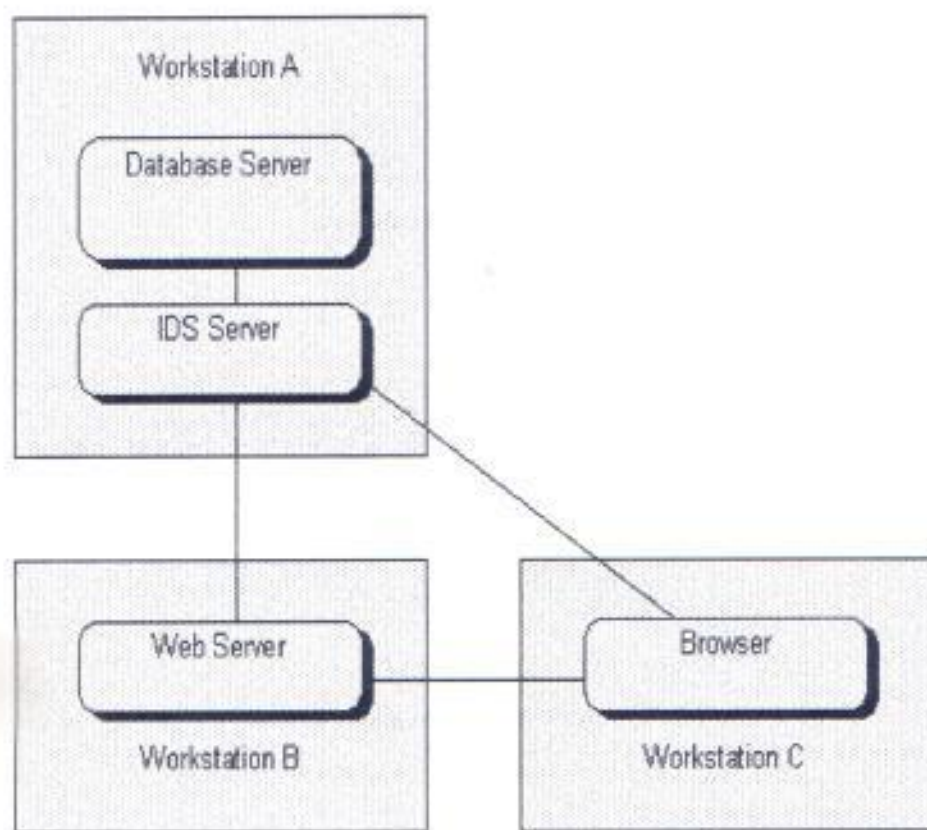


Figura # 3 Caso 2 de configuración del servidor IDS

5.1.3 Caso 3.

Provee conexiones a base de datos y de un servidor Web para browser.

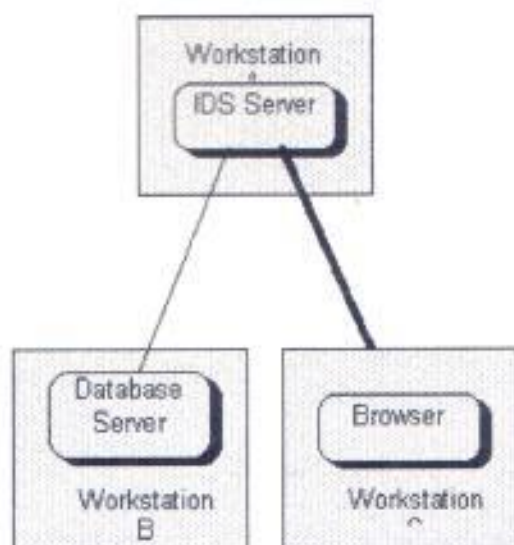


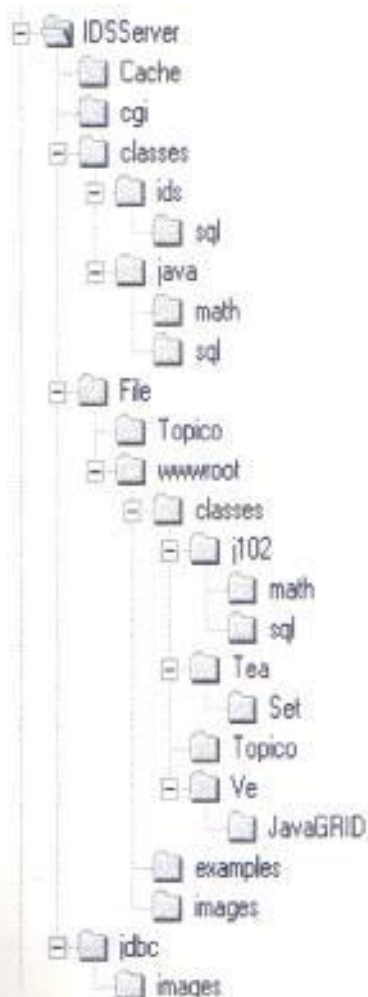
Figura # 4 Caso 3 de configuración del servidor IDS

5.2 Instalación del servidor IDS

El servidor IDS solo lo conforma un archivo ejecutable. Llamado **IDSS.EXE**.

El servidor IDS instala los siguientes directorios en jerarquía bajo el directorio escogido por el usuario .

El directorio **cache** es utilizado para almacenar temporalmente los archivos HTX.



El directorio **CGI** almacena el programa gateway IDS CGI llamado idsc.exe y todos los archivos necesarios para crear programas gateway .

El directorio **classes** contiene el driver IDS JDBC y el programa de muestra IDSTest.java

El directorio **File** contiene dos subdirectorios:

- **TOPICO** almacena todos los archivos HTX
- **WWWROOT** es el directorio raiz, cuando el servidor IDS da servicio a requerimientos web como servidor web, dentro de el :

- **TEA** directorio del paquete de clases adicionales. (ImageButton)
- **VE** directorio del paquete de clases adicionales. (JavaGrid)

El directorio **JDBC** contiene la documentación del API JDBC.

Figura # 5 Directorio del Servidor IDS

Además de los directorios provisto por el servidor IDS es necesario incluir en el directorio "File / Wwwroot / Classes" los paquetes de clases que no sean estándares de Java. Incluimos entonces los paquetes de clases

VE.JavaGRID (cuadrícula), **Tea.Set** (botón con imagen), además del desarrollado para fines de este proyecto llamado "Topico".

5.3 Instalación del controlador ODBC de la base de datos

1. Abrir el panel de control
2. Haga un click en botón SYSTEM DNS y presione OK
3. Haga un click en el botón para adicionar
4. Resalte la base de datos a utilizar: **CINVENTARIO32.MDB**
5. Haga un click en el botón Seleccionar y luego utilice el dialogo para encontrar el directorio IDS Server.
6. En el espacio Data Source Name edítele y coloque el nombre de su base de datos.
7. Haga un click sobre OK

5.4 Configuración del servidor IDS

La configuración del servidor IDS es realizada vía el archivo texto **idss.ini** localizado en el directorio de instalación. El contenido de este archivo es cargado durante el levantamiento del servidor IDS. Las modificaciones a

este archivo no hacen ningún efecto antes de que el servidor sea apagado y luego levantado.

En **Windows 95** el servidor IDS se ejecuta como un programa de consola. La instalación crea en el menú de inicio una carpeta llamada IDS Server, y un enlace directo de llamada a IDS Server. Se debe seleccionar este enlace directo

Si se decide ejecutar al servidor en la línea de comando debe hacerlo como se indica a continuación:

```
C:\>IDSServer\idss -win95 ↵
```

Para terminar el servidor en Windows 95, presione <Ctrl>+C en la ventana o simplemente cierre la ventana.

☛ CONECTIVIDAD A BASE DE DATOS: JDBC

JDBC (Java DataBase Connectivity), es una extensión al API de Java estándar que facilita la conectividad de los programas realizados en Java con bases de datos.

JDBC consiste de dos capas:

1. API JDBC

2. API administrador de controladores

El **API JDBC** se comunica con el **API administrador de controladores JDBC**, enviándole a este varias sentencias SQL, el **administrador de controladores** puede comunicarse con varias terceras partes que actualmente son las que se conectan a la base de datos y retornan la información desde el query o logran la acción específica hacia el query, todo esto transparentemente del programado.

JDBC al igual que ODBC están basados en X/Open SQL llamada especificación de nivel de interfaz. Las reglas para escribir estos administradores son encapsuladas en los administradores de **CONTROLADORES API JDBC**.

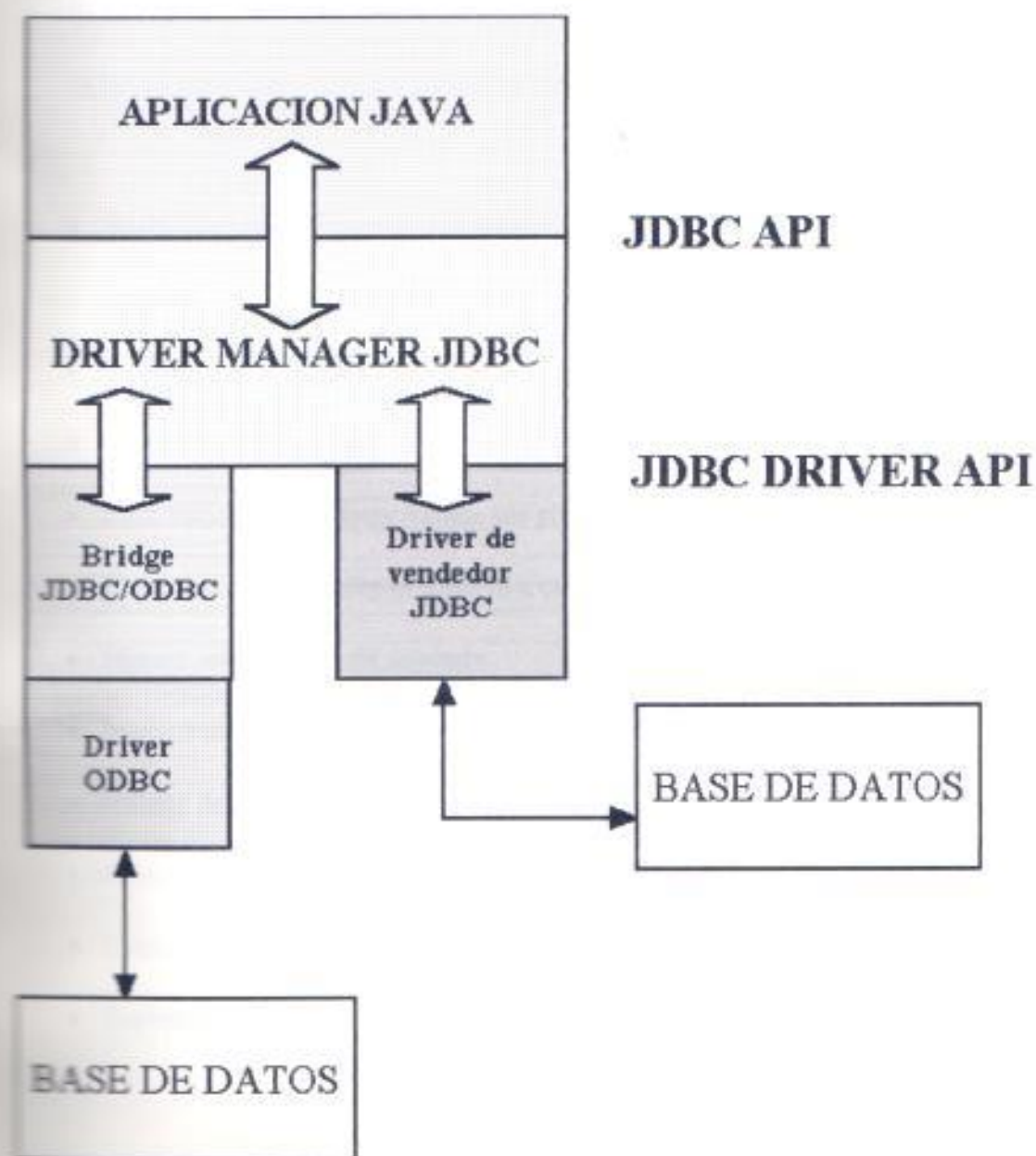


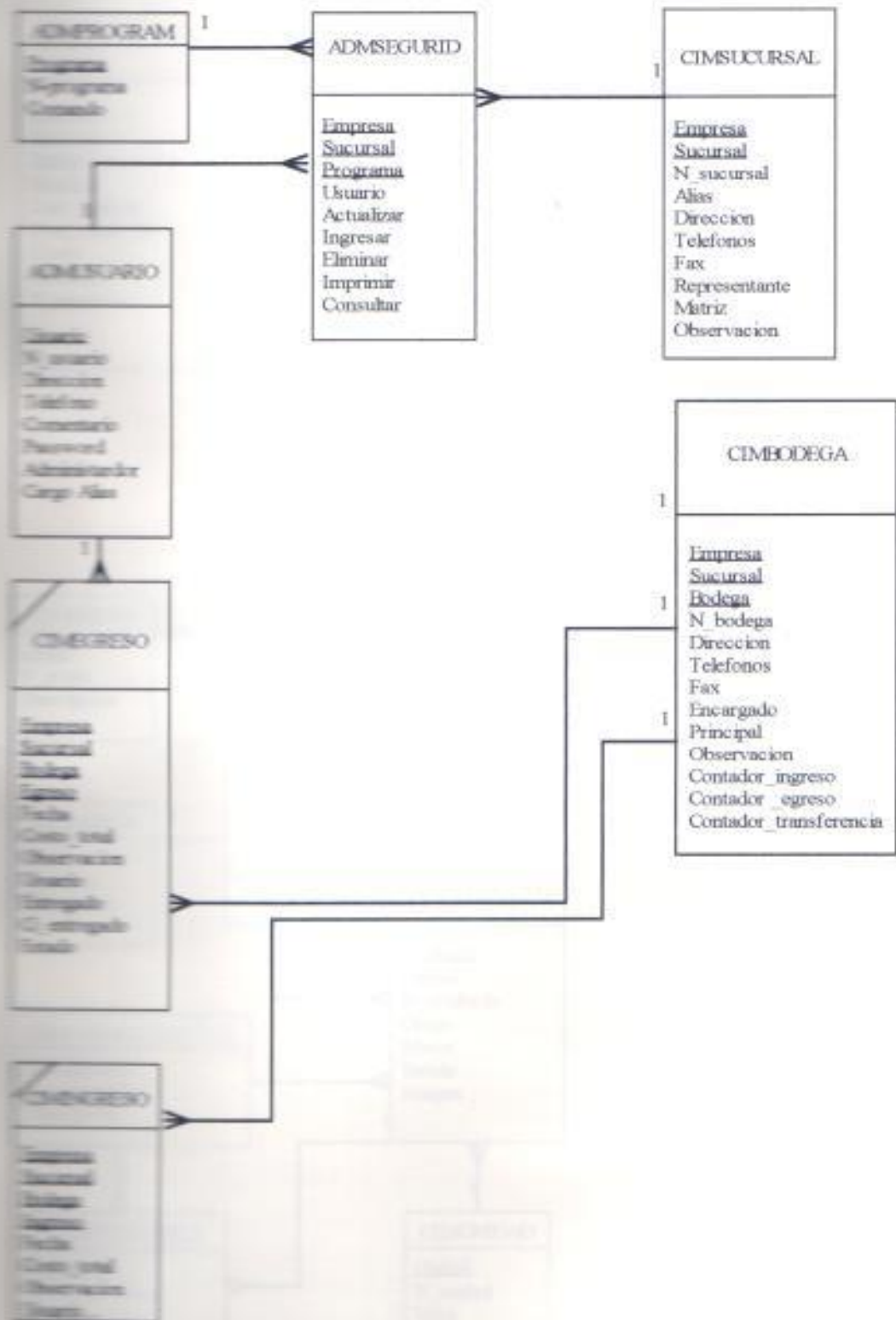
Figura # 6 ODBC

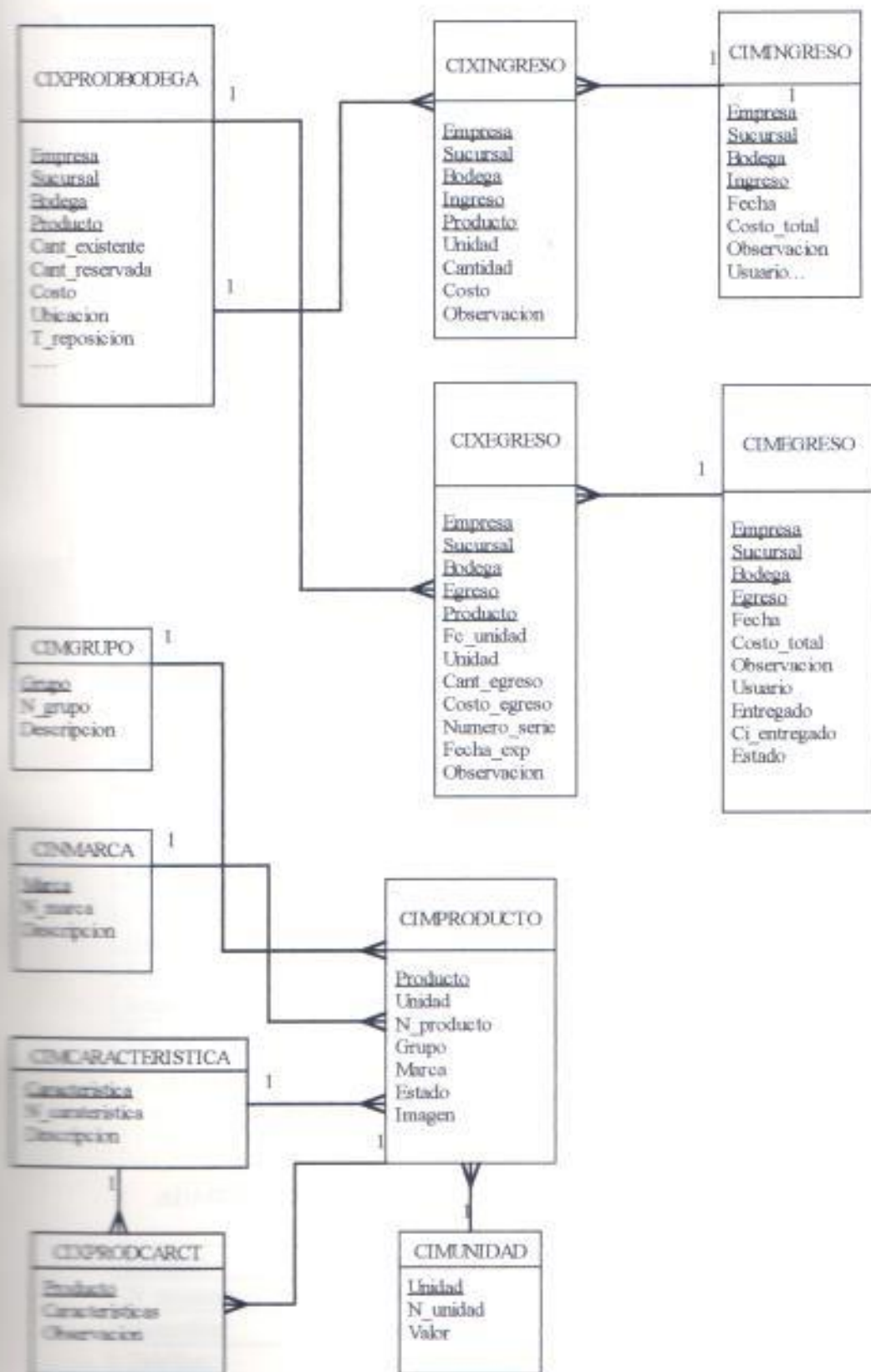
7. DISEÑO DE LA BASE DE DATOS

El **SISTEMA DE INVENTARIO** utiliza una base de datos cuyo diseño se basa en los requerimientos de una compañía real. Sus características principales son:

- **Multiempresarial** puesto que está concebido para el manejo de más de una empresa.
- Maneja **varias sucursales**
- Maneja **varias bodegas**
- Cada **sucursal y bodega** maneja sus productos
- El ingreso de productos **es abierto** a cualquier variedad
- Maneja **solo un tipo de moneda**
- Implementado con un **pequeño sistema de seguridad**
- Manejo de Kardex
- Ingreso y egreso de productos por bodega
- Consultas
- Reportes

7.1 Diagrama entidad relación





7.2 Descripción de las tablas

1. ADMEMPRESA

Empresa	Entero Largo	Auto Numérico
N empresa	Texto	30
RUC	Texto	13
Representante	Texto	35

2. ADMPROGRAM

Programa	Texto	10
N programa	Texto	50
Comando	Texto	20

3. ADMSEGURID

Empresa	Entero Largo	Auto Numérico
Sucursal	Entero Largo	Auto Numérico
Programa	Texto	10
Usuario	Texto	15
Actualizar	Texto	1
Ingresar	Texto	1
Eliminar	Texto	1
Imprimir	Texto	1
Consultar	Texto	1

4. ADMUSUARIO

Usuario	Texto	15
N usuario	Texto	40
Dirección	Texto	40
Teléfono	Texto	15

Comentario	Texto	40
Password	Texto	15
Administrador	Booleano	
Cargo	Texto	15
Alias	Texto	15
Empresa	Entero Largo	
Sucursal	Entero Largo	
Bodega	Entero Largo	

5. CIMBODEGA

Empresa	Entero Largo	
Sucursal	Entero Largo	
Bodega	Entero Largo	Auto Numérico
N_bodega	Texto	30
Dirección	Texto	40
Telefonos	Texto	15
Fax	Texto	15
Encargado	Texto	30
Principal	Booleano	
Observación	Texto	50
Contador_ingreso	Entero Largo	
Contador_egreso	Entero Largo	
Contador_transferencia	Entero Largo	

6. CIMCARACTERISTICA

Características	Entero Largo	Auto Numérico
N_características	Texto	25
Descripción	Texto	50

7. CIMEGRESO

<u>Empresa</u>	Entero Largo	
<u>Sucursal</u>	Entero Largo	
<u>Bodega</u>	Entero Largo	
<u>Egreso</u>	Entero Largo	Auto Numérico
<u>Fecha</u>	Fecha y Hora	
<u>Costo_total</u>	Numérico	Simple
<u>Observación</u>	Texto	100
<u>Usuario</u>	Texto	15
<u>Entregado</u>	Texto	40
<u>Ci_entregado</u>	Texto	13
<u>Estado</u>	Texto	1

8. CIMGRUPO

<u>Grupo</u>	Entero Largo	Auto Numérico
<u>N_grupo</u>	Texto	25
<u>Descripción</u>	Texto	50

9. CIMINGRESO

<u>Empresa</u>	Entero Largo	
<u>Sucursal</u>	Entero Largo	
<u>Bodega</u>	Entero Largo	
<u>Ingreso</u>	Entero Largo	Auto Numérico
<u>Fecha</u>	Fecha y Hora	
<u>Costo_total</u>	Numérico	Simple
<u>Observación</u>	Texto	100
<u>Usuario</u>	Texto	15
<u>Entregado</u>	Texto	40
<u>Ci_entregado</u>	Texto	13
<u>Estado</u>	Texto	1

10. CIMMARCA

<u>Marca</u>	Entero Largo	Auto Numérico
N_marca	Texto	25
Descripción	Texto	50

11. CIMPEDIDO

<u>Empresa</u>	Entero Largo	
<u>Sucursal</u>	Entero Largo	
<u>Bodega</u>	Entero Largo	
<u>Pedido</u>	Entero Largo	
<u>Fecha</u>	Fecha y Hora	
Observación	Texto	200
Usuario	Texto	15
Entregado	Texto	40
Ci entregado	Texto	13
Estado	Texto	1

12. CIMPRODUCTO

<u>Producto</u>	Texto	20
Unidad	Entero Largo	
N_producto	Texto	30
Grupo	Entero Largo	
Marca	Entero Largo	
Estado	Booleano	
Imagen	Texto	50

13. CIMSUCURSAL

<u>Empresa</u>	Entero Largo	
<u>Sucursal</u>	Entero Largo	Auto Numérico
N_sucursal	Texto	30

Alias	Texto	30
Dirección	Texto	40
Telefonos	Texto	15
Fax	Texto	15
Representante	Texto	40
Matriz	Booleano	
Observación	Texto	50

14. CIMUNIDAD

Unidad	Entero Largo	Auto Numérico
N unidad	Texto	25
Valor	Entero Largo	

15. CIXEGRESO

Empresa	Entero Largo	
Sucursal	Entero Largo	
Bodega	Entero Largo	
Egreso	Entero Largo	
Producto	Texto	20
Unidad	Entero Largo	
Cantidad	Entero Largo	
Costo	Numérico	Simple
Observacion	Texto	100

16. CIXPRODBODEGA

Empresa	Entero Largo	
Sucursal	Entero Largo	
Bodega	Entero Largo	
Producto	Texto	20
Cant existente	Entero Largo	

Cant reservada	Entero Largo	
Costo ubicacion	Entero Largo	
Ubicacion	Texto	15
T_reposicion	Entero Largo	
F_ult compra	Texto	18
Ult_costo	Texto	18
Ult_cantidad	Texto	18
Cant_max	Entero Largo	
Cant_min	Entero Largo	
Fecha	Fecha y Hora	
Tipo	Texto	2
Documento	Entero Largo	
Cant total ant	Entero Largo	
Costo_total_ant	Númeroico	Simple

17. CIXPRODCARACT

Producto	Texto	20
Característica	Entero Largo	
Observación	Texto	100

8. MANUAL DEL USUARIO

8.1 Introducción al Sistema de Control de Inventario

El sistema de control de inventario accesa a una base de datos que contiene la información de características y ubicación de los productos de los que dispone la compañía, así como también de las bodegas o sucursales donde se encuentra almacenado.

Debido a que el sistema de control de inventario fue desarrollado en Java, es una aplicación que se ejecuta en un browser de internet, y tiene la capacidad de ejecutarse en cualquier plataforma que posea una máquina virtual Java

8.2 Uso del Sistema de Control de Inventario

8.2.1 Opciones que maneja el sistema de inventario

El menú de inventario permite:

- El mantenimiento de sucursales de su compañía así como de las bodegas existentes.
- El mantenimiento e ingreso de sus grupos, marcas, unidades, productos y sus características.
- Manejar comprobantes de ingreso y egreso de productos.
- Consulta de productos en las diferentes bodegas existentes.
- Reportes de productos por bodega, cantidad, etc.

8.2.2 Ingresar una bodega o sucursal.

1. Seleccione la opción del menú Sucursal o Bodega.
2. Presione el botón Nuevo.
3. Introduzca información en cada uno de los campos.
4. Presione actualizar.

8.2.3 Actualizar datos de una sucursal o bodega.

1. Seleccione la opción del menú Sucursal o Bodega.
2. Presione el botón Buscar.
3. Ingrese si lo desea un criterio de búsqueda y a continuación presione el botón buscar
4. Seleccione una sucursal o bodega de la lista que se presenta y presione el botón Seleccionar.
5. Realice los cambios en los campos que crea conveniente.
6. Presione actualizar

8.2.4 Ingresar productos, marcas, grupos, unidades o características.

1. En el menú principal seleccione la opción correspondiente.
2. Presione el botón Nuevo.
3. Introduzca información en cada uno de los campos.

4. Presione el botón actualizar

8.2.5 *Actualizar valores de productos*

1. En el menú principal escoja la opción Productos
2. Presione el botón Buscar.
3. Ingrese si lo desea un criterio de búsqueda y a continuación presione el botón Buscar
4. Seleccione una producto de la lista que se presenta y presione el botón Seleccionar.
5. Realice los cambios en los campos que crea conveniente.
6. Presione actualizar

8.2.6 *Consultar productos en las diversas bodegas*

1. En el menú principal escoja la opción Inventario
2. Presione el botón Bodega
3. Seleccione la Sucursal
4. Presione el botón Buscar
5. Presione Seleccionar
6. En la nueva ventana, seleccione el nombre de la Bodega (Se resalta la fila en rosado)

7. Escoja seleccionar
8. Ingrese el grupo y la marca
9. Presione el botón Buscar
10. Le retorna una ventana con los productos

8.2.7 Consultar el Kardex

1. En el menú principal escoja la opción Inventario
2. Presione el botón Bodega para seleccionar la Bodega a consultar
3. Seleccione la Sucursal
4. Presione el botón Buscar
5. Presione Seleccionar
6. En la nueva ventana, seleccione el nombre de la Bodega (Se resalta la fila en rosado)
7. Seleccione el botón Seleccionar
8. Ingrese el grupo y la marca
9. Presione Buscar
10. Le retorna una ventana con los productos
11. Escojo el producto deseado (Se resalta la fila en rosado)
12. Le retorna una ventana de la información del Kardex

8.2.8 Realizar comprobantes de ingreso y egreso

1. En el menú principal escoja la opción Inventario
2. Presione el botón Bodega para seleccionar la Bodega a consultar
3. Seleccione la empresa y Sucursal
4. Presione el botón Buscar
5. Presione el botón Seleccionar
6. En la nueva ventana, seleccione el nombre de la Bodega (Se resalta la fila en rosado)
7. Presione el botón Seleccionar
8. Ingrese el grupo y la marca
9. Presione el botón Buscar
10. Le retorna una ventana con los productos
11. Escoja el producto deseado (Se resalta la fila en rosado)
12. Retorna una ventana con la información del Kardex
13. Presione el botón Buscar
14. En la pantalla de movimientos, ingrese las fechas de consulta

8.3. Respuestas a preguntas frecuentes

8.3.1. *¿Qué son los grupos?*

Se llaman grupos a los nombres generales de las diversas áreas de productos de una compañía. Esto es : Accesorios, Monitores, Impresoras, etc.

8.3.2. *¿Como me puedo cambiar a otra bodega?*

1. En el menú principal escoja la opción Inventario
2. Escoja opción Consulta de productos
3. Presione Buscar
4. En la nueva pantalla, seleccione la sucursal en la tabla (la fila se pinta de rosado)
5. Presione Seleccionar

9. CONCLUSIONES

Para grandes y pequeñas compañías cuyas sucursales y puntos de venta se encuentran distribuidas geográficamente es de vital importancia la eficacia en el

control de existencias de mercadería. Convirtiendo el control de inventario distribuido en una necesidad real.

Estas sucursales y puntos de venta deben tener a su disposición la información acerca de la distribución de sus productos, con el objeto de disponer de ellos en el menor tiempo posible, justo a tiempo.

La solución al problema consistió en implementar un APPLLET JAVA, el cual es ejecutado dentro de la máquina virtual java provista por el browser de internet. El applet una vez cargado, establece una conexión con el servidor IDS y una vez conectado solicita al usuario mediante una caja de dialogo que ingrese el nombre de usuario y el password. Al realizarse el requerimiento (SQL), el servidor IDS lo recibe y lo envia al controlador ODBC correspondiente, ya establecido por la conexión, quien realiza la consulta a la base de datos y retorna la respuesta al servidor IDS.

Una vez que el servidor IDS tiene la respuesta, esta es enviada mediante la conexión establecida al APPLLET DE JAVA, quien finalmente la muestra en una interfaz gráfica de usuario.

Este procedimiento lo lleva a cabo para cada uno de los requerimientos que el cliente solicita.

El sistema distribuido de inventario necesariamente maneja seguridad verificando al usuario, y de acuerdo al nivel habilita las opciones del menú en las cuales tenga acceso.

En relación a las herramientas de desarrollo podemos decir : que al contar únicamente con herramientas de desarrollo en versiones **shareware y freeware** disponibles en Internet las tareas de programación no son las optimas y en algunos casos limitan al programador.

JAVA es una herramienta que aunque este en desarrollo ha demostrado ser poderosa, útil y flexible, sin embargo con la versión de JDK 1.0.2, que es la utilizada en este proyecto, hemos encontrado los siguientes inconvenientes:

El lenguaje Java no soporta múltiple herencia. Todas las clases cuenta solo con una SUPERCLASE llamada Object. Sin embargo esta no es una limitación importante gracias a que existe en Java el concepto de Interfaces, las cuales se utilizan para reutilizar código heredando métodos.

La librería AWT no es particularmente poderosa, debido a que los objetos no tienen todos los métodos que un programador de Visual Basic, por ejemplo, podría esperar de ellos.

Las interfaces realizadas con la librería AWT lucen poco sofisticadas si se las compara con las nativas de Windows.

No existen en esta versión cajas de herramientas gráficas como las existentes en Visual Basic, u otros lenguajes que permiten generar el diseño visual y luego generan el código, lo que obliga a los programadores a escribir gran cantidad de código.

El diseño de los pantallas sin ayuda de la herramienta VBtoJava es tedioso, porque debe realizarse a base de paneles.

No existen aceleradores (acceso directo mediante teclas), a opciones de menú.

Una vez que ha sido adicionado un item a un objeto choice, este ya no puede ser removido.

El objeto JavaGRID (Cuadrícula) a pesar de prestar una gran ayuda visual, no permite ocultar columnas.

Windows cuenta con muchos eventos sofisticados . Si se compara existen cientos de eventos Windows y aproximadamente una docena en AWT.

Al ser el lenguaje JAVA independiente de plataforma, AWT no distingue entre los botones del ratón porque existen equipos como los Macintosh que cuentan con solo uno.

El lenguaje Java es sensitivo al teclado esto es, no distingue entre mayúsculas y minúsculas.

La tecla TAB no tiene el comportamiento asociado con Windows.

No se puede mezclar tipos de letra en Java sin tener que cuidadosamente posicionar los string manualmente. (pixeles).

BIBLIOGRAFIA

GARY CORNELL Y CAY HORSTMANN, Corejava, Sun Press Publications, 1996, 776p

LAURA LEMAY, Teach yourself Java in 21 days, Sam Net Publications, 1995, 556p

VARIOS, Manuales de Java encontrados en Internet

A P E N D I C E

PARTE DE PROGRAMA EN JAVA DEL SISTEMA DE INVENTARIO

```

package topico;

import java.awt.*;
import java.util.*;
import j102.sql.*;
import java.applet.*;
import tea.set.*;
import VE.JavaGRID.*;

public class Cimegreso extends Frame {
// Declaración de Variables
    boolean banClose=false;

    Connection conn;
    Statement prep1=null;

    ResultSet rsTabla;
    int banLee=0,banNuevo=0,banStament=0;
    Dialog frmBuscar;

    String vUltegreso;

    JavaGRID lst;

    CheckboxGroup group = new CheckboxGroup(); //used or reserved for radio
    buttons

    TextField txtusuario = new TextField( "", 12);
    Checkbox chkeestado = new Checkbox( "Procesado");
    TextField txtcosto = new TextField( "", 11);
    Button cmdbodega = new Button( "Bodega");
    TextField txtempresa = new TextField( "", 22);

    TextField txtestado = new TextField( "", 48);

```

```

ImageButton cmdsalir ;
ImageButton cmdactualizar ;
ImageButton cmdeliminar ;
ImageButton cmdbuscar ;
ImageButton cmdnuevo ;

```

```

Imágenes Cimg;

```

```

TextArea txtobservacion = new TextArea( "", 4, 34);
TextField txtcientregado = new TextField( "", 13);
TextField txtentregado = new TextField( "", 22);
TextField txtfecha = new TextField( "", 7);
TextField txtegreso = new TextField( "", 6);
TextField txtbodega = new TextField( "", 22);
TextField txtsucursal = new TextField( "", 25);
Label label20 = new Label( "Usuario", Label.LEFT );
Label Label4 = new Label( "Costo", Label.LEFT );
Label Label8 = new Label( "Observaciones", Label.LEFT );
Label Label7 = new Label( "CI/RUC Recibido", Label.LEFT );
Label Label6 = new Label( "Recibido", Label.LEFT );
Label Label5 = new Label( "Fecha", Label.LEFT );
Label label = new Label( "egreso", Label.LEFT );
Label Label3 = new Label( "Bodega", Label.LEFT );
Label Label2 = new Label( "Sucursal", Label.LEFT );
Label Label1 = new Label( "Empresa", Label.LEFT );
Menu mnuegreso = new Menu( "&egresos");

```

```

MenuItem mnuproducto = new MenuItem( "&Productos");
MenuItem mnuprocesar = new MenuItem( "&Procesar");
MenuItem mnuraya1 = new MenuItem( "-");
MenuItem mnusalir = new MenuItem( "&Salir");
MenuBar topMenu = new MenuBar();
String varusuario;
float d_x;
float d_y;

```

```

// ADD POSITION
=====
=====

```

```

protected void addPosition( Component c,
    int x,
    int y,

```

```

int width,
int height ){
add(c);
x= (int)(x * d_x);
y= (int)(y * d_y);
width = (int)(width * d_x);
height = (int)(height * d_y);
c.reshape(x,y, width, height);
} //end addPosition

```

```

//CONSTRUCTOR=====

```

```

public Cimegreso( String titulo,Connection c, String parametro[],Imagenes vimg ){

    super(titulo);

    Cimg=vimg;

    lst      =      new      JavaGRID      (JavaGRID.ROW_HEADERS      +
JavaGRID.COL_HEADERS, true);
    lst.setColumns (7);

    cmdsalir = new ImageButton (Cimg.ImgSalir, "Salir");
    cmdactualizar = new ImageButton (Cimg.ImgActualizar, "Actualizar");
    cmdeliminar = new ImageButton (Cimg.ImgEliminar, "Eliminar");
    cmdbuscar = new ImageButton (Cimg.ImgBuscar, "Buscar");
    cmdnuevo = new ImageButton (Cimg.ImgNuevo, "Nuevo");

    vUltegreso="";

    txtcosto.setEditable(false);
    java.util.Date varFecha=new java.util.Date();
    txtfecha.setText(varFecha.toLocaleString());

    varusuario=parametro[0];
    txtempresa.setText(parametro[1]);
    txtsucursal.setText(parametro[2]);
    txtbodega.setText(parametro[3]);

    setLayout(null);
    addNotify();

```



```

setFont(new Font ("Dialog", Font.PLAIN, 8));
Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
d_x = (float)(d.width) / 800;
d_y = (float)(d.height)/ 600;
resize ((int)(521 * d_x), (int)(400 * d_y) );
setResizable(false);
setBackground( new Color ( 192,192,192 ));
this.move(100,100);
//  addPosition( txtestado , 20,335,466,24);
addPosition( lst, 20,135,480,135);

lst.hide();

addPosition( txtusuario , 365,205,121,24);
addPosition( chkeestado , 370,245,101,21);
addPosition( txtcosto , 385,155,106,24);
addPosition( cmdbodega , 250,95,71,31);
addPosition( txtempresa , 20,55,216,24);

addPosition( cmdnuevo , 20,290,80,60);
addPosition( cmdactualizar , 110,290,80,60);
addPosition( cmdeliminar , 200,290,80,60);
addPosition( cmdbuscar , 290,290,80,60);
addPosition( cmdsalir , 380,290,80,60);

addPosition( txtobservacion , 20,205,331,66);
addPosition( txtcientregado , 250,155,126,24);
addPosition( txtentregado , 20,155,216,24);
addPosition( txtfecha , 404,105,87,24);
addPosition( txtegreso , 340,105,50,24);
addPosition( txtbodega , 20,105,216,24);
addPosition( txtsucursal , 250,55,241,24);
addPosition( label20 , 365,185,51,16);
addPosition( Label4 , 385,135,106,21);
addPosition( Label8 , 20,185,121,21);
addPosition( Label7 , 250,135,121,21);
addPosition( Label6 , 20,135,71,21);
addPosition( Label5 , 420,85,66,16);
addPosition( label , 340,85,66,21);
addPosition( Label3 , 20,85,71,21);
addPosition( Label2 , 250,35,81,16);
addPosition( Label1 , 20,35,91,16);

```

```

txtusuario.setFont( new Font ( "Dialog", Font.PLAIN, 8));
chkestado.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtcosto.setFont( new Font ( "Dialog", Font.PLAIN, 8));
cmdbodega.setFont( new Font ( "Dialog", Font.BOLD, 8));
txtempresa.setFont( new Font ( "Dialog", Font.PLAIN, 8));
cmdsalir.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtestado.setFont( new Font ( "Dialog", Font.PLAIN, 8));
cmdactualizar.setFont( new Font ( "Dialog", Font.PLAIN, 8));
cmdeliminar.setFont( new Font ( "Dialog", Font.PLAIN, 8));
cmdbuscar.setFont( new Font ( "Dialog", Font.PLAIN, 8));
cmdnuevo.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtobservacion.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtcientregado.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtentregado.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtfecha.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtegreso.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtbodega.setFont( new Font ( "Dialog", Font.PLAIN, 8));
txtsucursal.setFont( new Font ( "Dialog", Font.PLAIN, 8));
label20.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label4.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label8.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label7.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label6.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label5.setFont( new Font ( "Dialog", Font.BOLD, 8));
label.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label3.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label2.setFont( new Font ( "Dialog", Font.BOLD, 8));
Label1.setFont( new Font ( "Dialog", Font.BOLD, 8));

```

```

Font f = new Font (getFont ().getName (), Font.BOLD, getFont ().getSize ());
lst.setColHeaderFont (f);
lst.setRowHeaderFont (f);
lst.setGridFont (new Font (getFont ().getName (), Font.PLAIN, getFont
().getSize ()));

```

```

mnuegreso.add(mnuproducto);
mnuegreso.add(mnuprocesar);
mnuegreso.add(mnuraya1);
mnuegreso.add(mnusalir);
topMenu.add(mnuegreso);
setMenuBar(topMenu);

```



```

    return(str);
}

//
=====
ACTION
=====

public boolean action (Event evt, Object arg){
    Frame frmIng;
    if(evt.target ==cmdnuevo) {
        if (!lst.isVisible())
            f_nuevo();
        else if (!(chkeestado.getState())){
            frmIng =new Cidegreso_productoing(conn,true,this);
            frmIng.show();
        }
    }
    else if (evt.target ==cmdactualizar) {
        if (!lst.isVisible())
            f_actualizar(true);
        else if (!(chkeestado.getState())){
            frmIng =new Cidegreso_productoing(conn,false,this);
            frmIng.show();
        }
    }
    else if (evt.target ==cmdeliminar) {
        if (!lst.isVisible())
            f_eliminar();
        else if (!(chkeestado.getState()))
            f_eliminar_detalle();
    }
    else if (evt.target ==cmdbuscar){
        if (!lst.isVisible()) f_buscar();
    }
    else if (evt.target ==cmdbodega){
        if (!lst.isVisible()) f_bodega();
    }
    else if (evt.target ==cmdsalir) {
        f_salir();
    }
    else if (evt.target ==mnuproducto) {
        if (banNuevo!=1) mnuproducto_Click();
    }
}

```



```

}
else if (evt.target ==mnuprocesar) {
    f_procesar();
}
else if (evt.target ==mnusalir) {
    f_salir();
}
return true;
}

```

```
//
```

```
ADDITEM=====
```

```

public void additem(String str[],int row){
    String varS[]= {"", "", "", "", "", "", ""};
    varS[0]=str[0];varS[1]=str[1];
    varS[2]=str[2];varS[3]=str[3];
    varS[4]=str[4];varS[5]=str[5];
    varS[6]=str[6];
    //lst.addRow(varS);
    lst.setElement(varS[0],row,0);
    lst.setElement(varS[1],row,1);
    lst.setElement(varS[2],row,2);
    lst.setElement(varS[3],row,3);
    lst.setElement(varS[4],row,4);
    lst.setElement(varS[5],row,5);
    lst.setElement(varS[6],row,6);

}

```

```
//
```

```
MENU
```

```
PRODUCTO
```

```
=====
```

```

void mnuproducto_Click(){
/* if ((banNuevo==0)){
    Dialog w=new Cixegreso(this,conn);
    w.move(40,40);
    w.show();
}*/
if (lst.isVisible())
    lst.hide();

```

```

else{
    lst.show();
    f_siguiente_detalle();
}

}

//BODEGA
=====
public void f_bodega() {
    if ((banNuevo==1)){
        Dialog w=new Cimegresobodega_busca(conn,this);
        w.show();
    }

}

//
=====
STOP

public void stop() {

    try{
        if (banStament==1) rsTabla.close();
        prep1.close() ;
        prep1=null;
        rsTabla=null;
    } catch (SQLException e) { ;}

}

//=====
public boolean handleEvent(Event evt) {
    if (evt.id == Event.WINDOW_DESTROY) {
        f_salir();
        return false;
    };
    return super.handleEvent(evt);
} // end handle event

```



```

//=====
void f_salir(){
    f_nuevo();
    stop();
    this.dispose();
}

//ACTUALIZAR=====
void f_actualizar(boolean banMsg){
    MsgBox msg=null;
    if (!(chkeestado.getState()))
    if (f_valida()==1){
        String qry="";
        int fila=0;
        try{
            if (banNuevo==1){
                qry="Insert into Cimegreso (empresa,sucursal,bodega,fecha,";
                qry+=" costo_total,observacion,usuario,entregado,ci_entregado,estado)";
                qry+=" values("+txtempresa.getText().substring(0,4);
                qry+=" ", "+txtsucursal.getText().substring(0,4);
                qry+=" ", "+txtbodega.getText().substring(0,4);
                qry+=" ", "+txtfecha.getText();
                qry+=" ", "+txtcosto.getText();
                qry+=" ", "+txtobservacion.getText();
                qry+=" ", "+txtusuario.getText();
                qry+=" ", "+txtentregado.getText();
                qry+=" ", "+txtcientregado.getText();
                if (chkeestado.getState())
                    qry+=" ", 'P)";
                else
                    qry+=" ", 'E)";
                } else{
                qry="UPDATE Cimegreso SET observacion="+txtobservacion.getText();
                qry+=" ", entregado="+ txtentregado.getText();
                qry+=" ", ci_entregado="+txtcientregado.getText();
                qry+=" ", usuario="+txtusuario.getText();
                qry+=" ", costo_total="+txtcosto.getText();
                qry+=" " WHERE";
                qry+=" " egreso="+txtegreso.getText();
                }
            txtestado.setText("Prepara Query: "+qry);

```

```

Statement prep = conn.createStatement() ;

ResultSet rs;
txtestado.setText("Ejecutando... ");
fila= prep.executeUpdate(qry) ;

prep.close() ;
prep=null;
txtestado.setText("Ejecutado: "+ qry);
if (fila>0){
    if (banNuevo==1) {
        //f_nuevo();
        banNuevo=1;
        banLee=0;

        qry="Select cimegreso.empresa,admempresa.n_empresa, ";
        qry+=" cimegreso.sucursal,cimsucursal.n_sucursal,cimegreso.bodega, ";
        qry+=" cimbodega.n_bodega,cimegreso.egreso, ";
        qry+=" cimegreso.fecha,cimegreso.costo_total,cimegreso.observacion, ";
        qry+=" cimegreso.usuario,cimegreso.entregado,cimegreso.ci_entregado,
";

        qry+=" cimegreso.estado ";
        qry+=" from Cimegreso,admempresa,cimsucursal,cimbodega ";
        qry+=" where cimegreso.empresa=admempresa.empresa";
        qry+=" and cimegreso.sucursal=cimsucursal.sucursal";
        qry+=" and cimegreso.bodega=cimbodega.bodega";
        qry+=" ORDER BY ";
        qry+=" cimegreso.egreso desc";
        f_siguiente(qry);

        txtestado.setText("Registro Ingresado ");
        msg=new                               MsgBox("Registro
Ingresado","Mensaje",Cimg.ImgMsgboxInfo,this);
        if (banMsg) msg.show();
    }else{
        txtestado.setText("Registro Actualizado" );
        msg=new                               MsgBox("Registro
Actualizado","Mensaje",Cimg.ImgMsgboxInfo,this);
        if (banMsg) msg.show();
    }
}
else{
//      txtestado.setText("No se ejecuto sentencia: "+ qry);

```

```

                                msg=new      MsgBox("No      se      ejecuto
comando", "Mensaje", Cimg.ImgMsgboxError, this);
                                msg.show();

                                }

                                } catch (Throwable e) {
                                txtestado.setText("SQLException: "+ qry);
                                msg=new      MsgBox("No      se      ejecuto
comando", "Mensaje", Cimg.ImgMsgboxError, this);
                                msg.show();

                                }

                                }
                                }
//

```

ELIMINAR

```

void f_eliminar(){
    MsgBox msg=null;
    if (!(chkeestado.getState())){
        String qry="";
        try{
            if (banNuevo==0){
                qry = "DELETE * FROM Cimegreso WHERE";
                qry+= " egreso="+txtegreso.getText();
                txtestado.setText("Prepara Query: "+qry);
                Statement prep = conn.createStatement() ;
                ResultSet rs;
                txtestado.setText("Eliminando Registro, ejecutando...");
                prep.executeUpdate(qry) ;
                txtestado.setText("Registro eliminado");
                                msg=new      MsgBox("Registro
eliminado", "Mensaje", Cimg.ImgMsgboxInfo, this);
                                msg.show();
                prep.close() ;
                prep=null;
                f_nuevo();
            }
        } catch (SQLException e) {
                                msg=new      MsgBox("No      se      ejecuto
comando", "Mensaje", Cimg.ImgMsgboxError, this);

```

```

        msg.show();
        txtestado.setText("Exepcion al eliminar registro en :"+ qry);
    }
}
}
//NUEVO

```

```

=====
void f_nuevo(){
    f_limpiar();
    banNuevo=1;
    banLee=0;
    txtestado.setText("Ingrese un nuevo registro");
}
//BUSCAR

```

```

=====
void f_buscar(){
    frmBuscar = new Cimegreso_busca(conn,this);
    frmBuscar.move(100,100);
    frmBuscar.show();
}
//

```

PROCESAR

```

=====
void f_procesar(){
    MsgBox msg=null;
    if (!(chkeestado.getState()))
    if (!(txtcosto.getText().equals("0"))){
        chkeestado.setState(true);

        String qry=" ";
        int fila=0;
        try{
            //corregir este codigo
            Statement prep = conn.createStatement() ;

            ResultSet rs;
            for(int i=0;i<lst.getRows();i++){
                txtestado.setText("Ejecutando... ");
                qry="UPDATE cixprodbodega SET " ;
                qry+= " cixprodbodega.fecha='"+txtfecha.getText()+"'";
                qry+= " , cixprodbodega.tipo='CE'";
            }
        }
    }
}

```



```

txtestado.setText("Ejecutando Query... "+qry);
fila= prep.executeUpdate(qry) ;
txtestado.setText("Ejecutado: "+ qry);

prep.close() ;
prep=null;

if (fila>0){
    txtestado.setText("Registro Procesado" );
    msg=new                               MsgBox("Registro
Procesado", "Mensaje", Cimg.ImgMsgboxInfo, this);
    msg.show();
}
else{
    txtestado.setText("No se ejecuto sentencia: "+ qry);
}

} catch (Throwable e) {
    txtestado.setText("SQLException: "+ qry);
    msg=new                               MsgBox("Problemas           en
Proceso", "Mensaje", Cimg.ImgMsgboxError, this);
    msg.show();
}

}
}

```

```

//                               SIGUIENTE                               DETALLE

```

```

void f_siguiente_detalle(){
    if (!vUltegreso.equals(txttegreso.getText())){
        vUltegreso=txttegreso.getText();
        String                               qry="Select
Cixegreso.producto,cimproducto.n_producto,Cixegreso.unidad,";
        qry+="cimunidad.n_unidad,Cixegreso.cantidad,Cixegreso.costo,";
        qry+="Cixegreso.cantidad*Cixegreso.costo as total,Cixegreso.observacion ";
        qry+=" from cimproducto,Cimunidad,Cixegreso ";
        qry+=" where Cixegreso.empresa =" +txttempresa.getText().substring(0,4);
        qry+=" and Cixegreso.sucursal =" +txtsucursal.getText().substring(0,4);
        qry+=" and Cixegreso.bodega =" +txtbodega.getText().substring(0,4);
        qry+=" and Cixegreso.egreso =" +txttegreso.getText();
    }
}

```



```

qry+=" and Cixegreso.producto = cimproducto.producto";
qry+=" and Cixegreso.unidad = Cimunidad.unidad";

```

```

try{
    //Prepara Qry
    txtestado.setText("Prepara Query: "+qry);
    Statement prep = conn.createStatement() ;
    txtestado.setText("Ejecutando...+qry");
    ResultSet rsTabla1 = prep.executeQuery(qry) ;
    txtestado.setText("Cargando Lista");
    String s1;
    Double a;
    lst.setRows(0);
    String varFila[]={ "", "", "", "", "", "", "" };
    while (rsTabla1.next())      {
        varFila[0]= rsTabla1.getString(1);
        varFila[1]= rsTabla1.getString(2);
        varFila[2]= f_llenabancos(rsTabla1.getString(3),4)+rsTabla1.getString(4);
        varFila[3]= rsTabla1.getString(5);
        a=new Double(rsTabla1.getString(6));
        varFila[4]= ""+a.longValue();
        a=new Double(rsTabla1.getString(7));
        varFila[5]= ""+a.longValue();
        varFila[6]= rsTabla1.getString(8);

        lst.insertRowAt (lst.getRows());
        lst.setRowHeaderElement (String.valueOf (lst.getRows()), lst.getRows()-
1);
        additem(varFila,lst.getRows()-1);

    }
    rsTabla1.close();
    rsTabla1=null;
    prep.close() ;
    prep=null;
    txtestado.setText("No existen más registros "+ qry);

// f_lee();
} catch (SQLException e) {
    txtestado.setText("Error de exepcion en: "+qry);
}
}
}
}

```

//ACTUALIZAR

DETALLE

```

public void f_actualizar_detalle(String varP[],boolean banNuevo){
    String qry="";
    MsgBox msg=null;

    try{
        if (banNuevo){

            qry="Insert into Cixegreso(empresa,sucursal,bodega,egreso,producto,";
            qry+="unidad,cantidad,costo,observacion)";
            qry+= " values( "+ txtempresa.getText().substring(0,4);
            qry+=     " , "+ txtsucursal.getText().substring(0,4);
            qry+=     " , "+ txtbodega.getText().substring(0,4);
            qry+=     " , "+ txtegreso.getText();
            qry+=     " , "+ varP[0]; //producto
            qry+=     " , "+ varP[1].substring(0,4);//unidad
            qry+=     " , "+ varP[2]; //cantidad
            qry+=     " , "+ varP[3]; //costo
            qry+=     " , "+ varP[4]; //observacion
            qry+=     " )";
        }else{

            qry="UPDATE Cixegreso SET ";
            qry+= " unidad="+varP[1].substring(0,4)+"";
            qry+= ",cantidad="+varP[2]+"";
            qry+= ",costo="+varP[3]+"";
            qry+= ",observacion="+varP[4]+"";
            qry+= " WHERE";
            qry+= " egreso="+ txtegreso.getText();
            qry+= " and producto="+varP[0]+"";

        }

        txtestado.setText("Prepara Query: "+qry);
        Statement prep = conn.createStatement() ;

        txtestado.setText("Ejecutando... ");
        int nFilas=prep.executeUpdate(qry) ;

        prep.close() ;
    }
}

```

```

        prep=null;
        txtestado.setText("Ejecutado: "+ qry);
        if (nFilas>0){
            String varFila[]={ "", "", "", "", "", "", "" };
                varFila[0] = varP[0];
            varFila[1]= varP[5];
            varFila[2]= varP[1];
            varFila[3]= varP[2];
            varFila[4]= varP[3];
            Double a=new Double(varP[2]);
            Double b=new Double(varP[3]);
            varFila[5]=""+(a.doubleValue()*b.doubleValue());
            varFila[6]= varP[4];

        }

        if (banNuevo) {

            lst.insertRowAt (lst.getRows());
            lst.setRowHeaderElement (String.valueOf (lst.getRows()-
1);
            additem(varFila,lst.getRows()-1);

            // txtestado.setText("Registro Ingresado ");
            msg=new                               MsgBox("Registro
Ingresado", "Mensaje", Cimg.ImgMsgboxInfo,this);
            msg.show();

        }else{

            Rectangle pos=lst.getSelection();
            if (pos!=null){

                lst.insertRowAt (pos.y);
                lst.setRowHeaderElement (String.valueOf (pos.y+1), pos.y);
                additem(varFila,pos.y);
                lst.deleteRow(pos.y+1);
            // txtestado.setText("Registro Actualizado" );
            msg=new                               MsgBox("Registro
Actualizado", "Mensaje", Cimg.ImgMsgboxInfo,this);
            msg.show();

        }
    }
}

```

```

double a1=0;
for (int i=0;i<lst.getRows();i++){
    b=new Double(lst.getElement(i,5));
    a1+=b.doubleValue();
}
txtcosto.setText(""+a1);
f_actualizar(false);
}else{
    txtestado.setText("No se puede ejecutar comando"+qry);
}
} catch (Throwable e) {
//    txtestado.setText("SQLException: "+ qry);
    msg=new      MsgBox("No      se      puede      ejecutar
comando", "Mensaje", Cimg.ImgMsgboxInfo, this);
    msg.show();
}
}

```

```

//          ELIMINAR          DETALLE

```

```

void f_eliminar_detalle(){
    Rectangle lstPosicion=lst.getSelection ();
    MsgBox msg=null;
    if (lstPosicion!=null){
        try{
            String qry;

            qry = "DELETE * FROM Cixegreso WHERE";
            qry+= " egreso="+ txtegreso.getText();
            qry+= " and producto="+lst.getElement(lstPosicion.y,0)+"";

            txtestado.setText("Prepara Query: "+qry);
            Statement prep = conn.createStatement() ;
            txtestado.setText("Eliminando Registro, ejecutando...");
            int nFilas=prep.executeUpdate(qry) ;
            if (nFilas>0){
//                txtestado.setText("Registro eliminado");

```



```

        } catch (SQLException e) { ; }
    }

//LEE
=====
=====

void f_lee(){
    try{
        txtempresa.setText(f_llenablanco(rsTabla.getString(1),3)+rsTabla.getString(2));
        txtsucursal.setText(f_llenablanco(rsTabla.getString(3),3)+rsTabla.getString(4));
        txtbodega.setText(f_llenablanco(rsTabla.getString(5),3)+rsTabla.getString(6));
        txtegreso.setText(rsTabla.getString(7));
        txtfecha.setText(rsTabla.getString(8).substring(0,10));
        txtcosto.setText(rsTabla.getString(9));
        txtobservacion.setText(rsTabla.getString(10));
        txtusuario.setText(rsTabla.getString(11));
        txtentregado.setText(rsTabla.getString(12));
        txtcientregado.setText(rsTabla.getString(13));
        if (rsTabla.getString(14).equals("P"))
            chkeestado.setState(true);
        else
            chkeestado.setState(false);
    } catch (SQLException e) { ; }
    banNuevo=0;
}

void f_limpiar(){
    txtegreso.setText("");
    txtfecha.setText("");
    txtcosto.setText("0");
    txtobservacion.setText("");
    chkeestado.setState(false);
    txtusuario.setText(varusuario);
    txtentregado.setText("");
    txtcientregado.setText("");
    java.util.Date varFecha=new java.util.Date();
    String fecha;
    if (varFecha.getYear()>=97)
        fecha= "19" +varFecha.getYear();
    else
        fecha= "20" +varFecha.getYear();
    if (varFecha.getMonth()>8 )
        fecha+="-"+(varFecha.getMonth()+1);
}

```



```
txtestado.setText("Ingrese C.I. recibido");
    return(0);
}
return(1);
}
}
```