# Escuela Superior Politécnica del Litoral

## Programa de Tecnología en Computación y Diseño Gráfico

# TOPICO DE GRADUACION

**Previo a la Obtención del Título de:**
**Programación de Sistemas**

**"SCI"**

**SISTEMA DE COMPRAS**

MANUAL DE DISEÑO

AUTORES:

**Sandra Araujo Villacís**

**Ulbio Morales Labre**

DIRECTOR:

**ANL. Miguel Salazar**

***Año 2004***

# AGRADECIMIENTO

Queremos dar constancia de nuestros más sinceros agradecimientos a DIOS, ya que es nuestra fuerza espiritual; a nuestros padres que de alguna manera son el soporte de cada uno de nosotros en todos estos años de sacrificio estudiantil, y a cada una de las personas que de una u otra forma colaboraron para la culminación de este Tópico.

# DEDICATORIA

Este trabajo va dedicado con mucho amor a DIOS y a la persona que siempre me enseñó que en el cielo hay un Ser Supremo, que nunca me falló cuando más la necesité, que siempre insistió para que culminara mi carrera, y de la cual me siento sumamente orgullosa.....mi madre. Además a todas aquellas personas que necesiten conocimientos sobre la elaboración de un Sistema de Compras.

SANDRA ARAUJO V.

# DEDICATORIA

Este trabajo va dedicado con mucho amor a DIOS, a mis padres quienes no teniendo me dieron todo, amor al trabajo, respeto a mis semejantes, dedicación al estudio, por todo lo que me dieron y por lo que me quisieron dar y no pudieron por no estar a su alcance. Además a todas aquellas personas que necesiten conocimientos sobre la elaboración de un Sistema de Compras.

ULBIO MORALES L.

# DECLARACIÓN EXPRESA

La responsabilidad por los hechos, ideas y doctrinas expuestas en este Tópico de Graduación nos corresponden exclusivamente; y el patrimonio intelectual de la misma a ***PROTCOM (Programa de Tecnología en Computación y Diseño Gráfico***) de la Escuela Superior Politécnica del Litoral.

(Reglamento de exámenes y títulos profesionales de la ESPOL).

# FIRMA DE LOS AUTORES DEL PROYECTO

SANDRA ARAUJO

ULBIO MORALES

## FIRMA DEL DIRECTOR DEL TÓPICO

ANL. MIGUEL SALAZAR

# TABLA DE CONTENIDO


*1. Generalidades ..... 1*
*1.1. Introducción ..... 1*
*1.2. Antecedentes ..... 1*
*1.3. Situación Actual de la Empresa ..... 2*
*1.4. Solución Propuesta ..... 3*
*1.4.1. Características del Nuevo Sistema ..... 3*
*1.5. Hardware requerido ..... 3*
*1.6. Capacitación del Personal ..... 4*
*1.7. Tiempo de Desarrollo e Implementación del Sistema ..... 4*
*1.8. Ubicación de Personal y Equipos ..... 4*
*1.9. Organización de este Manual ..... 4*

*2. Diagrama General del Sistema ..... 1*

*3. Diagramas de Casos de Uso ..... 1*

*3.1. Casos de Uso del Sistema de Compras ..... 1*
*3.2. Diagramas de Secuencias ..... 2*
*3.2.1. Proceso de Solicitud de Compra Bajo Petición ..... 2*
*3.2.2. Proceso de Solicitud de Compra por Reposición ..... 4*
*3.2.3. Proceso de Aprobar la Solicitud de Compra ..... 6*
*3.2.4. Proceso de Definir Proveedor ..... 8*
*3.2.5. Proceso de Emisión de Ordenes y Forma de Pago ..... 10*
*3.2.6. Proceso de Llegada de Mercadería ..... 12*
*3.2.7. Proceso de Devolución de Mercadería ..... 14*
*3.2.8. Proceso de Llegada Mercadería Devuelta ..... 16*

*4. Diagrama Jerárquico del Sistema ..... 1*

*4.1. Sistema de Compras ..... 2*
*4.1.1. Sistema de Compras (00000) ..... 2*
*4.1.2. Mantenimiento (10000) ..... 2*
*4.1.3. Movimientos (20000) ..... 2*
*4.1.4. Reportes / Consultas (30000) ..... 2*
*4.1.5. Acerca de... (40000) ..... 2*
*4.2. Menú Mantenimiento ..... 3*
*4.2.1. Ingresar Línea ..... 3*
*4.2.2. Salir ..... 3*
*4.3. Menú Movimientos ..... 4*
*4.3.1. Solicitud Compra ..... 4*
*4.3.1.1. Solicitud Bajo Pedido ..... 5*
*4.3.1.2. Solicitud por Reposición ..... 5*
*4.3.1.3. Aprobar Solicitudes de Compra ..... 5*

*4.3.2. Orden de Compra* ..... 5
*4.3.2.1. Definir Proveedor* ..... 5
*4.3.2.2. Emitir Orden y Forma de Pago* ..... 5
*4.2.3. Compras* ..... 5
*4.2.3.1. Llegada de Mercadería* ..... 5
*4.2.4. Devoluciones* ..... 6
*4.2.4.1. Devolución de Mercadería* ..... 6
*4.2.4.2. Llegada de Mercadería Devuelta* ..... 6
*4.4. Menú Reportes / Consultas* ..... 7
*4.4.1. Solicitud de Compra* ..... 7
*4.4.2. Orden de Compra* ..... 7
*4.4.3. Llegada de Mercadería* ..... 7

*5. Estandarizaciones de Formatos* ..... 1

*5.1. Formatos de Almacenamiento de Información* ..... 1
*5.1.1. Formato para el Nombre del Proyecto* ..... 1
*5.1.2. Formato para el Nombre de las Clases* ..... 2
*5.1.3. Formato para el Nombre de la Base de Datos* ..... 2
*5.1.4. Formato para el Nombre de Formas y Reportes* ..... 3
*5.1.5. Formato para el Nombre de las Tablas* ..... 3
*5.2. Formatos de Estándares de Programación* ..... 5
*5.2.1. Formato para el Nombre de los Controles* ..... 5
*5.2.2. Formato para el Nombre de los Campos de la Tabla* ..... 6
*5.3. Formatos de Estándares para Presentaciones de Pantallas* ..... 7
*5.3.1. Formatos de Colores* ..... 7
*5.3.2. Formatos de Fonts* ..... 7
*5.3.3. Formatos de Capitalización* ..... 7
*5.3.4. Formatos de Localización de Botones* ..... 7
*5.3.5. Formatos de Iconos* ..... 8

*6. Plan de Códigos* ..... 1

*6.1. Solicitud de Compra* ..... 1
*6.2. Orden de Compra* ..... 2
*6.3. Devolución de Mercadería* ..... 3
*6.4. Línea por Proveedor* ..... 4

*7. Modelo Entidad - Relación* ..... 1

*7.1. Modelo Lógico* ..... 1
*7.2. Modelo Físico* ..... 2

*8. Diccionario de Datos* ..... *1*

*8.1. Tabla COM_M_CABSC* ..... *1*
*8.2. Tabla COM_T_DETSC* ..... *2*
*8.3. Tabla COM_M_CABOC* ..... *3*
*8.4. Tabla COM_T_DETCO* ..... *4*
*8.5. Tabla COM_T_CDMER* ..... *5*
*8.6. Tabla COM_H_DDMER* ..... *6*
*8.7. Tabla COM_M_LXPROV* ..... *7*

*9. Diseño de Entradas y Salidas* ..... *1*

*9.1. Diseño de Entradas* ..... *1*
*9.1.1. Menú Principal del Sistema de Compras* ..... *2*
*9.1.2. Menú Mantenimiento* ..... *4*
*9.1.3. Menú Movimientos* ..... *6*
*9.1.4. Menú Reportes / Consultas* ..... *8*
*9.1.5. Menú Acerca de...* ..... *10*
*9.1.6. Mantenimiento en Línea* ..... *12*
*9.1.7. Solicitud de Compra Bajo Pedido* ..... *14*
*9.1.8. Solicitud de Compra por Reposición de Mercadería* ..... *16*
*9.1.9. Aprobación de Solicitud de Compra* ..... *18*
*9.1.10. Definición de Proveedor* ..... *20*
*9.1.11. Emisión de Ordenes y Forma de Pago* ..... *22*
*9.1.12. Registro de Llegada de Mercadería* ..... *24*
*9.1.13. Devolución de Mercaderías* ..... *26*
*9.1.14. Llegada de Mercadería Pendiente* ..... *28*
*9.1.15. Llegada de Mercadería por Devolución* ..... *30*
*9.1.16. Reporte de Solicitud de Compra* ..... *32*
*9.1.17. Reporte de Ordenes de Compra* ..... *34*
*9.1.18. Reporte de Llegada de Mercadería* ..... *36*

*9.2. Diseño de Salidas* ..... *1*
*9.2.1. Reporte de Solicitud de Compra General* ..... *2*
*9.2.2. Rreporte de Solicitud de Compra Detallado* ..... *4*
*9.2.3. Reporte de Orden de Compra General* ..... *6*
*9.2.4. Reporte de Orden de Compra Detallado* ..... *8*
*9.2.5. Reporte de Llegada de Mercadería General* ..... *10*
*9.2.6. Reporte de Llegada de Mercadería Detallado* ..... *12*

*10. Diagramas de Casos de Uso* ..... *1*

*10. Casos de Uso del Sistema de Compras* ..... *1*

*10.1. Diagramas de Estado* ..... *2*

*10.1.1. Estado de Tipo de Movimiento* ..... *2*
*10.1.2. Estado de Orden de Compra* ..... *3*

*11. Diagrama de Clases* ..... *1*

*11. Diagrama de Clase Detallado* ..... *2*

*11.1. Clase COMSOLPEDIDO* ..... *2*
*11.2. Clase COMCONPROD* ..... *3*
*11.3. Clase COMSOLREP* ..... *4*
*11.4. Clase COMBUSQUEDA* ..... *5*
*11.5. Clase COMAPROVSOLCOM* ..... *6*
*11.6. Clase COMPRINCIPAL* ..... *7*
*11.7. Clase COMDEFINIRPROV* ..... *8*
*11.8. Clase COMLLEGMERCADERIA* ..... *9*
*11.9. Clase COMEMIPAGOORDCOM* ..... *10*
*11.10. Clase COMCONSULTAPROV* ..... *11*
*11.11. Clase COMDATEFIELDDOCUMENT* ..... *12*
*11.12. Clase COMACERCADE* ..... *13*
*11.13. Clase COMDEFPAGO* ..... *14*
*11.14. Clase COMFORMATSWING* ..... *15*
*11.15. Clase COMJDATEFIELD* ..... *16*
*11.16. Clase COMJTEXTFIELDFILTER* ..... *17*
*11.17. Clase COMMANTLINEAXPROV* ..... *18*
*11.18. Clase COMLLEGDEVMERCADERÍA* ..... *19*
*11.19. Clase COMLLEGADAMERCADERIA* ..... *20*
*11.20. Clase COMLLEGMERCADERIA* ..... *21*
*11.21. Clase COMDNUMBER* ..... *22*
*11.22. Clase COMDEVOLUCIONMERC* ..... *23*
*11.23. Clase COMCONSSOLCOMPRA* ..... *24*
*11.24. Clase COMCONLLEGMERC* ..... *25*
*11.25. Clase COMMANTSOLPED* ..... *26*
*11.26. Clase COMMANTPRODEXT* ..... *27*
*11.27. Clase COMMANTSOLREP* ..... *28*
*11.28. Clase COMMANTPROVEEDORES* ..... *29*
*11.29. Clase COMMANTORDENCOMPRA* ..... *30*
*11.30. Clase COMMANTDEVOLUCIONMERC* ..... *31*
*11.31. Clase COMMANTLINEAXPROVSQL* ..... *32*
*11.32. Clase COMMANTLLEGDEVMERCADERÍA* ..... *33*
*11.33. Clase COMMANTLLEGMERCADERÍA* ..... *34*
*11.34. Clase COMMANTEMILPAGOORDCOM* ..... *35*
*11.35. Clase COMMANTDEVOLUCION* ..... *36*
*11.36. Clase COMEJECUTASQL* ..... *37*

*12. Definición de Clases* ..... *1*

*12.1. Clase COMSOLPEDIDO* ..... *2*
*12.2. Clase COMCONPROD* ..... *6*
*12.3. Clase COMSOLREP* ..... *8*
*12.4. Clase COMBUSQUEDA* ..... *11*
*12.5. Clase COMAPROVSOLCOM* ..... *13*

*12.6. Clase COMCONLLEGMERC* ..... *16*
*12.7. Clase COMCONORDCOMPRA* ..... *18*
*12.8. Clase COMCONSULTAPROV* ..... *20*
*12.9. Clase COMCONSSOLCOMPRA* ..... *21*
*12.10. Clase COMDNUMBER* ..... *23*
*12.11. Clase COMDATEFIELDDOCUMENT* ..... *25*
*12.12. Clase COMDEFPAGO* ..... *27*
*12.13. Clase COMDEFINIRPROV* ..... *29*
*12.14. Clase COMDEVOLUCIONMERC* ..... *32*
*12.15. Clase COMEMIPAGOORDCOM* ..... *35*
*12.16. Clase COMFORMATSWING* ..... *39*
*12.17. Clase COMJDATEFIELD* ..... *45*
*12.18. Clase COMLLEGDEVMERCADERIA* ..... *48*
*12.19. Clase COMLLEGMERCADERIA* ..... *51*
*12.20. Clase COMPRINCIPAL* ..... *54*
*12.21. Clase COMTABLSORT* ..... *57*
*12.22. Clase COMTABLMAP* ..... *59*
*12.23. Clase COMVISTPREV* ..... *61*
*12.24. Clase COMACERCADE* ..... *63*
*12.25. Clase COMJTEXTFIELDFILTER* ..... *65*
*12.26. Clase COMMANTLINEAXPROV* ..... *67*
*12.27. Clase COMMANTDEVOLUCION* ..... *69*
*12.28. Clase COMMANTDEVOLUCIONMERC* ..... *70*
*12.29. Clase COMMANTEMIPAGOORDCOM* ..... *71*
*12.30. Clase COMMANTLINEAXPROVSQL* ..... *72*
*12.31. Clase COMMANTLLEGDEVMERCADERIA* ..... *73*
*12.32. Clase COMMANTLLEGMERCADERIA* ..... *75*
*12.33. Clase COMMANTORDENCOMPRA* ..... *76*
*12.34. Clase COMMANTPRODEXT* ..... *79*
*12.35. Clase COMMANTPROVEEDORES* ..... *81*
*12.36. Clase COMMANTSOLPEDIDO* ..... *82*
*12.37. Clase COMMANTSOLREP* ..... *85*
*12.38. Clase COMEJECUTARSQL* ..... *87*

*13. Seguridades del Sistema* ..... *1*

*13.1. Niveles de Seguridad de Acceso y Uso* ..... *1*
*13.2. Seguridades de Datos* ..... *1*

# Capítulo 1

## Generalidades

# 1. GENERALIDADES

## 1.1. INTRODUCCIÓN

El objetivo principal de toda Empresa es lograr mayores utilidades y a los más bajos costos posibles, una de las formas que hacen factibles lograr estos principios es ahorrar tiempo y aumentar la eficacia en cada una de las actividades a realizarse.

Actualmente contamos con un avance tecnológico que nos permite alcanzar estos objetivos, lo cual sirve para automatizar nuestras actividades.

Una vez realizado un estudio minucioso de la Empresa, hemos procedido a crear un manual que le presente una solución eficaz a bajo costo y con asesoramiento de profesionales calificados por la Escuela Superior Politécnica del Litoral, los mismos que darán la capacitación adecuada al personal que lo requiera para el rápido manejo de sus actividades.

## 1.2. ANTECEDENTES

ECONOMARKET fue creado el 30 de Mayo de 1986, según la resolución No. 1234 de la Superintendencia de Compañía que autorizó su creación.

Esta empresa se encuentra ubicada en Aguirre 610 y Boyacá Esquina.

La idea de formar el ECONOMARKET surgió cuando hombres visionarios como el Dr. Alfredo Adum Ziade y el Econ. Mario Ribadeneira Auad, se reunieron en esta ciudad para proyectar la creación de esta obra. Sus fundadores depositaron todo el entusiasmo y vigor necesario para hacer realidad este proyecto, la gran obra que es hoy, radicada en la ciudad de Guayaquil.

Es una empresa dedicada a la venta de productos de primera necesidad al por mayor y menor.

Esta empresa durante el tiempo que lleva funcionando ha tenido la característica de familiar.

## 1.3. SITUACIÓN ACTUAL DE LA EMPRESA

La empresa cuenta al presente con un servidor y tres terminales, el servidor consta con un procesador Pentiun III de 700Mhz, un disco de 60 GB, memoria de 256MR. Los terminales tienen un procesador Pentiun III de 700Mhz, un disco de 20 GB y memoria de 128MR.

En cuanto a software, el servidor posee el sistema operativo Windows 2003 Server y las terminales, Windows XP Profesional. El software que controla las operaciones de la empresa es un sistema basado en FOXPRO bajo MS-DOS.

Una vez realizado el estudio correspondiente para conocer y determinar el estado actual de la empresa ECONOMARKET, se ha logrado enfocar diversos inconvenientes en el departamento administrativo, los mismos que impiden gozar de una mayor agilidad al ejecutar controles y procesos, trayendo consigo diversos problemas en el ámbito financiero.

Considerando los problemas, presentamos a continuación un detalle, de cada una de las situaciones que fueron establecidas mediante investigación:

| PROBLEMA | CAUSA | EFECTO |
|---|---|---|
| ✓ Búsqueda de datos de proveedores | ✓ Pérdida de tiempo al consultar la base de datos en Excel en donde enlistan todos sus proveedores | ✓ Si la información no existe se perdió tiempo valioso. |
| ✓ Elaboración de órdenes de compras | ✓ Debido a que se realiza manualmente las órdenes, no existe el sistema de control informático adecuado para el manejo de esta información | ✓ Ocasiona que exista posibles pérdidas de datos por errores humanos |
| ✓ Seguridades en Acceso a los datos. | ✓ No existe un control de acceso de usuario. | ✓ Posible alteración de Información. |
| ✓ Elaboración de reportes | ✓ No se generan reportes rápidos y actuales | ✓ Pérdida de tiempo en el flujo de información ocasionando tomas de decisiones inadecuadas |

## 1.4. SOLUCIÓN PROPUESTA

Una vez detectado los problemas que consideramos que retrasa el desarrollo de su Empresa, ponemos a su consideración una propuesta que representa una solución a los mismos, ya que le permitirá cubrir sus necesidades en menos tiempo y a bajos costos, de manera funcional y controlada, disminuyendo así considerablemente sus gastos.

La solución Propuesta consiste en implementar un Sistema Automatizado de Compras, el mismo que contendrá las opciones necesarias para el Módulo de Compras propuesto, atiende todas las necesidades de adquisición de productos en general. Controla el registro personalizado de requerimientos, genera órdenes de compra, realiza el ingreso y/o egreso de la mercadería recibida, los que detallaremos a continuación:

El Sistema propuesto se desarrollará con el Sistema Operativo Windows 2003 Server, bajo el Manejador de Datos ACCESS y se utilizará el LENGUAJE DE MODELAMIENTO UNIFICADO (UML), que se utiliza para el Diseño Orientado a Objetos, los programas estarán implementados en JAVA 1.4.0.

### 1.4.1. CARACTERÍSTICAS DEL NUEVO SISTEMA

Después de haber efectuado un análisis a fondo de las actividades que se desarrollan en la empresa ECONOMARKET y definido sus problemas principales, se ha determinado que el nuevo sistema tendrá las siguientes características en lo que se refiere a:

**Sistema de Compras**

- Permitirá la búsqueda rápida y exacta de los Proveedores.
- Registrará las diferentes Órdenes de Compra.
- Brindará las seguridades necesarias al momento de acceder a los datos.
- Permitirá la elaboración de los reportes actuales de una manera rápida.
- Dado que el sistema debe ser supervisado en forma general y en todos los puntos, es necesaria la contratación de un Analista de Sistemas para desempeñar dicha función.
- Dentro de cada punto se requiere de un Analista de Sistemas que brinde el asesoramiento necesario a los usuarios.
- Para la fase de desarrollo y la fase de implementación se contratará un Programador de Sistemas, el mismo que brindará la capacitación adecuada para el perfecto desarrollo del sistema.

## 1.5. HARDWARE REQUERIDO

Considerando que la empresa posee los equipos necesarios para el desarrollo de este sistema no deberá incurrir en gastos de nuevas adquisiciones.

## 1.6. CAPACITACIÓN DEL PERSONAL

El personal operativo recibirá la capacitación necesaria por parte del Analista de Sistemas.

## 1.7. TIEMPO DE DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA

La fase del desarrollo del sistema durará 8 semanas a partir del 27 de Septiembre del presente año, por consiguiente deberá concluir el 22 de Noviembre.

La fase de implementación durará 2 semanas y comenzará el 23 de Noviembre del año en curso y concluirá el 6 de Diciembre.

## 1.8. UBICACIÓN DE PERSONAL Y EQUIPOS

La ubicación del personal es la misma en la que realiza actualmente sus actividades, así como el uso de los equipos que posee la empresa.

Debido a que la empresa no cuenta con un departamento de Sistemas propiamente dicho, se deberá crear dicho departamento en donde se colocará el correspondiente servidor, observando las seguridades del caso.

## 1.9. ORGANIZACIÓN DE ESTE MANUAL

Este manual se encuentra estructurado de la siguiente manera:

| | |
|---|---|
| Capítulo 1 | Generalidades |
| Capítulo 2 | Diagrama de General del Sistema |
| Capítulo 3 | Diagrama de Casos de Uso |
| Capítulo 4 | Diagrama Jerárquico del Sistemas |
| Capítulo 5 | Estandarización de Formatos |
| Capítulo 6 | Plan de Código |
| Capítulo 7 | Modelo Entidad – Relación |
| Capítulo 8 | Diseño de Archivos |
| Capítulo 9 | Diseño de Entradas y Salidas |
| Capítulo 10 | Diagramas de Estado |
| Capítulo 11 | Diagrama de Clases |
| Capítulo 12 | Definición de Clases |
| Capítulo 13 | Seguridades del Sistema |

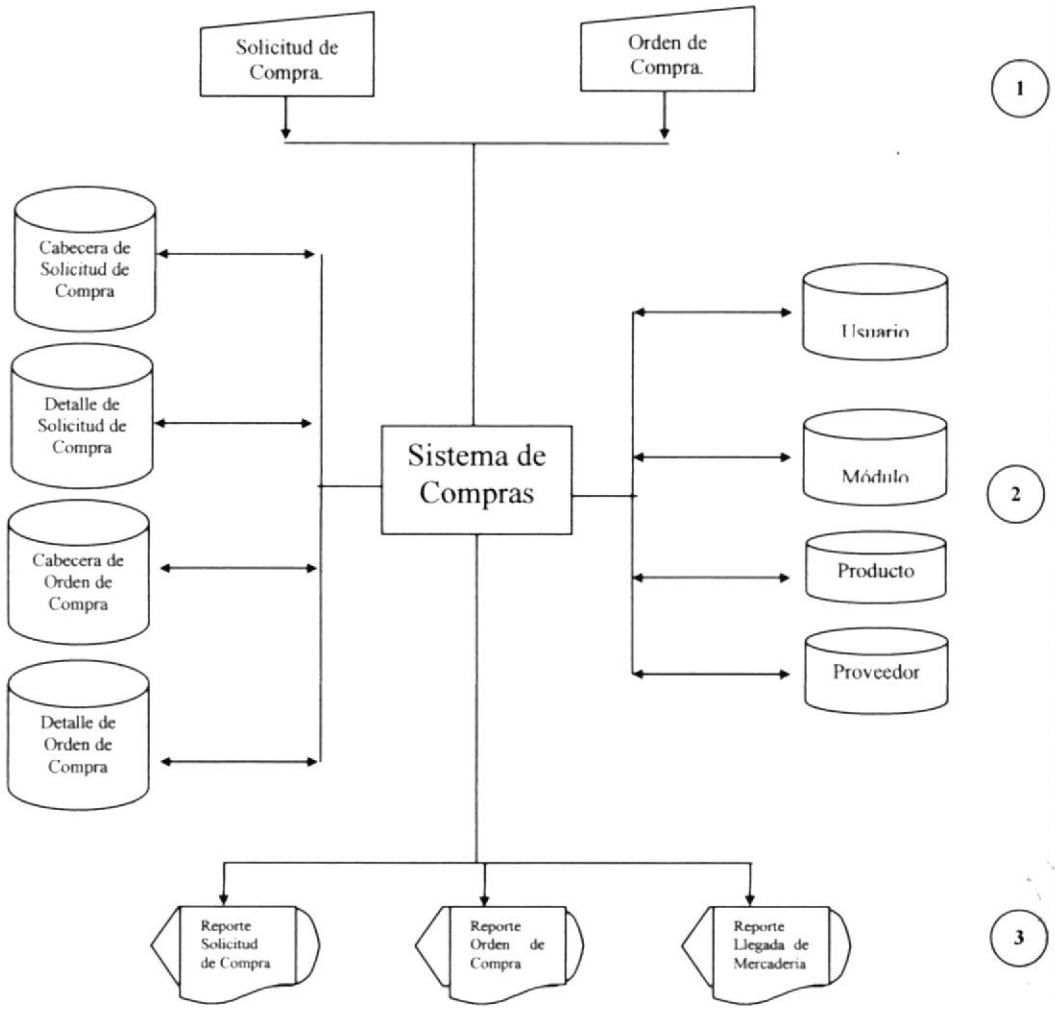
# Capítulo 2

## Diagrama General del Sistema

# 2. DIAGRAMA GENERAL DEL SISTEMA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 de 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DIAGRAMA GENERAL DEL SISTEMA**




Observación:

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA GENERAL DEL SISTEMA

| | |
|---|---|
| **1. -** | Ingreso de los datos para las Tablas maestras del Sistema de Compras. |
| **2. -** | El Sistema de Compras toma información de las diferentes Tablas y actualiza la información. |
| **3. -** | El Sistema genera como salidas Reportes y Formas Impresas necesarias para conocer los precios y costos de los materiales. |

Observación:

# Capítulo 3

# Diagramas de Casos de Uso

ECA
PUS
EÑAS

# 3. DIAGRAMAS DE CASOS DE USO

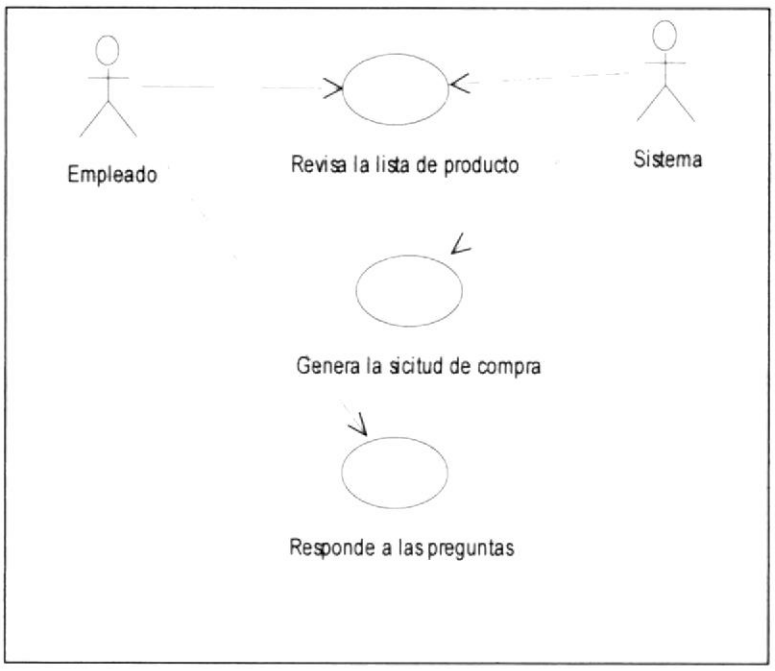
## 3.1. CASOS DE USO DEL SISTEMA DE COMPRAS

| | | |
|---|---|---|
| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 de 1 | |
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| **LISTADO DE CASOS DE USO** | | |
| 1. Proceso de Solicitud de Compra Bajo Petición.<br>2. Proceso de Solicitud de Compra Por Reposición.<br>3. Proceso de Aprobar Solicitud de Compra.<br>4. Proceso de Definir Proveedor.<br>5. Proceso de Emisión de Órdenes y Forma de Pago.<br>6. Proceso de Llegada de Mercadería.<br>7. Proceso de Devolución de Mercadería.<br>8. Proceso de Llegada de Mercadería Devuelta. | | |
| **Observación:** | | |

## 3.2. DIAGRAMAS DE SECUENCIAS

### 3.2.1. PROCESO DE SOLICITUD DE COMPRA BAJO PETICIÓN

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| **DISEÑO DEL DIAGRAMA DE SECUENCIA** | | |




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

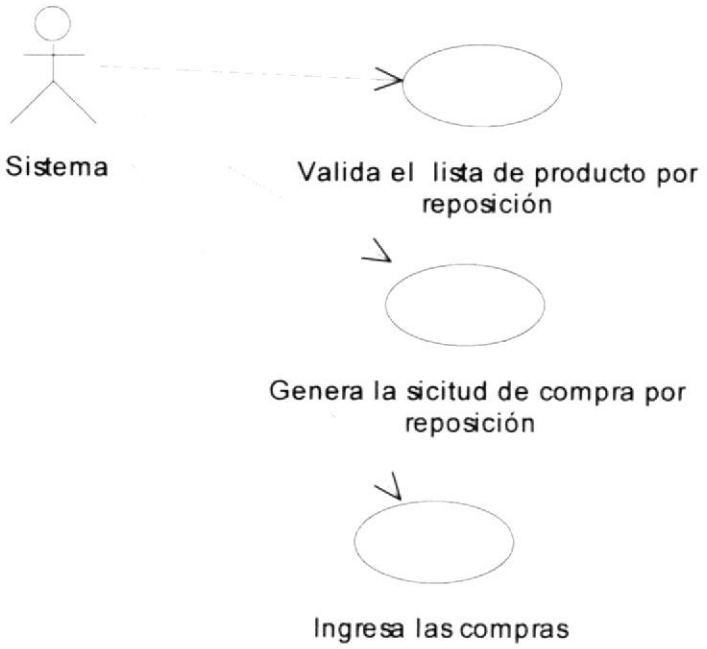
1. El Empleado revisa el Listado de Productos necesarios.
2. El Sistema de Compras revisa el Listado de Productos Disponibles para generar las Solicitudes de Compras Bajo Petición.
3. El Sistema de Compras junto con el Listado de Productos solicitados por el Empleado generan las Solicitudes de Compra Bajo Petición.
4. El Sistema Compras genera un mensaje para preguntar al Empleado si desea grabar.
5. El Empleado responde que si desea grabar los datos y el Sistema de Compras los graba.
6. El Sistema de Compras genera un mensaje para preguntar al Empleado si desea ingresar otra Solicitud de Compra Bajo Petición.
7. El Empleado si desea ingresar otra Solicitud de Compra Bajo Petición vuelve al paso 1.

**Observaciones:**

## 3.2.2. PROCESO DE SOLICITUD DE COMPRA POR REPOSICIÓN

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

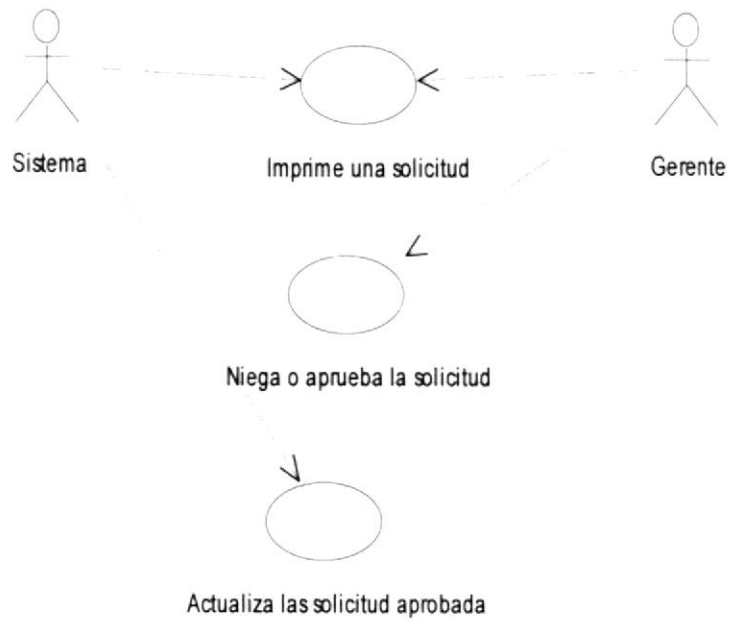
1. El Sistema de Inventario valida un Listado de Productos por Reposición.
2. El Sistema de Inventario genera una Solicitud de Compra Por Reposición.
3. El Sistema de Compras obtiene la Solicitud de Compra Por Reposición.
4. El Sistema de Compras genera el Ingreso del Listado de la Solicitud de Compra Por Reposición.

**Observaciones:**

## 3.2.3. PROCESO DE APROBAR LA SOLICITUD DE COMPRA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

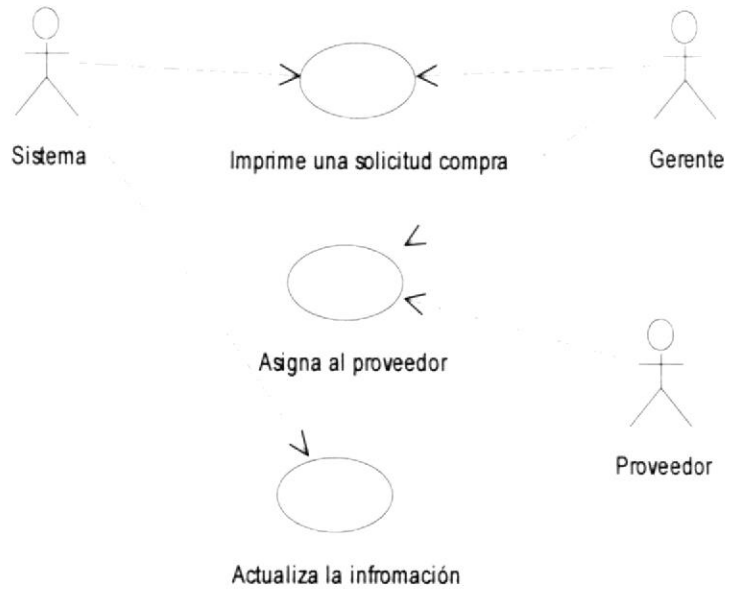
1. El Sistema de Compras presenta el Listado de la Solicitud de Compra al Gerente.
2. El Gerente procede a aprobar o anular la Solicitud de Compra.
3. Si la Solicitud de Compra ha sido aprobada entonces se genera el ingreso del Listado de Solicitudes de Compras Aprobadas .
4. El Sistema de Compras actualiza las Solicitudes de Compras Aprobadas.
5. El Sistema de Compras genera un mensaje para preguntar al Gerente si desea realizar otra Aprobación de Solicitud de Compra.

**Observaciones:**

## 3.2.4. PROCESO DE DEFINIR PROVEEDOR

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 de 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

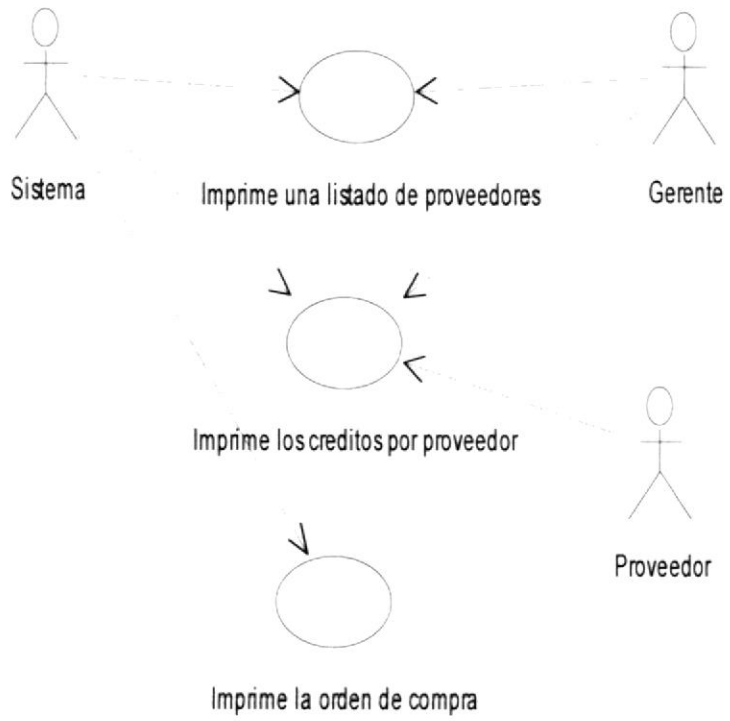
1. El Sistema de Compras presenta las Solicitudes de Compras Aprobadas al Gerente.
2. El Sistema de Cuentas por Pagar presenta un Listado de los Proveedores por Línea.
3. El Gerente con los Listados de las Solicitudes de Compras Aprobadas y de los Proveedores por Línea define el Proveedor.
4. El Sistema de Compras con la información de la definición del Proveedor guarda en la Orden de Compra.
5. El Sistema de Compras genera un mensaje para preguntar al Gerente si desea realizar otra Definición de Proveedor.

**Observaciones:**

## 3.2.5. PROCESO DE EMISIÓN DE ÓRDENES Y FORMA DE PAGO

| **SCI - COM**<br>**Sistema de Compras**<br>VERSIÓN 1.0 | | Pág. 1 de 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

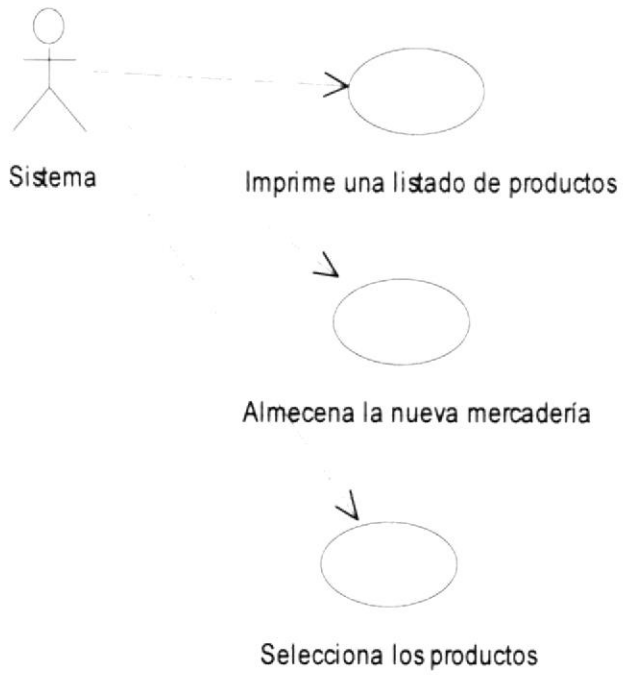
1. El Sistema de Compras presenta un Listado de Productos por Proveedor al Gerente.
2. El Sistema de Cuentas por Pagar presenta un Listado de los Créditos por Proveedor al Gerente.
3. El Gerente se vale de los Listados de Productos por Proveedor y de los Créditos por Proveedor para Definir su forma de Pago.
4. Una vez que el Gerente ha definido su Forma de Pago se lo guarda para luego poder imprimir la Orden de Compra.

**Observaciones:**

## 3.2.6. PROCESO DE LLEGADA DE MERCADERÍA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 de 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

1. El Sistema de Compras presenta el Listado de Productos por Orden de Compra.

2. El Sistema de Compras con el Listado de Productos por Orden de Compra procede a registrar la Llegada de Mercadería.

3. Una vez Llegada la Mercadería, el Sistema de Compras procede a guardar los productos recibidos.

4. Si los productos recibidos están completos, entonces el Sistema de Compras genera dos Listados: de Productos Recibidos que se lo da a Inventario y de Valora Pagar por Producto Recibido que se lo da a Cuentas Por Pagar.

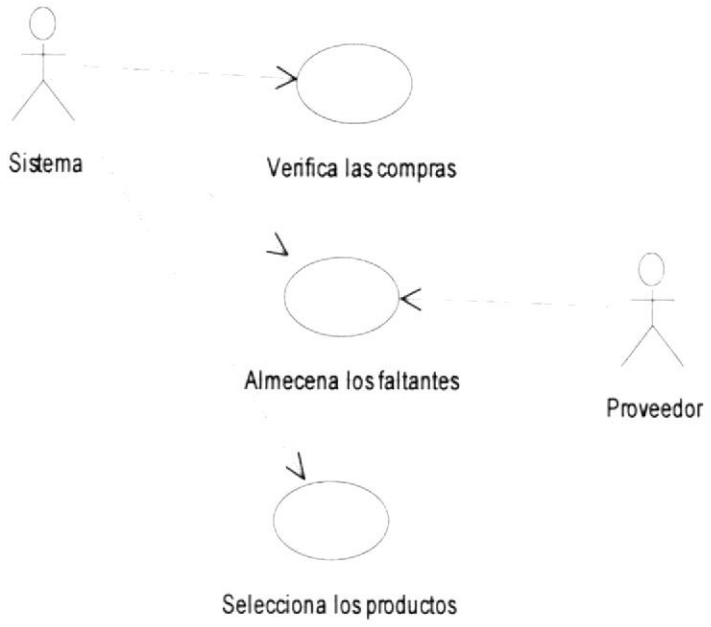
5. Si los productos recibidos no se recibieron completos o llegaron en mal estado, entonces el Sistema de Compras registra ese faltante o devolución para su respectivo trámite.

**Observaciones:**

## 3.2.7. PROCESO DE DEVOLUCIÓN DE MERCADERÍA

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 de 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

1. Compras verifica si es que la mercadería ha llegado en buen estado y/o completo; si no es así, entonces procede a registrar faltante de producto en la Orden de Compra.

2. El Sistema de Compras registra el faltante de producto en la Orden de Compra y así actualizar en el Sistema de Cuentas por Pagar e Inventario.

3. Compras procede a llegar a un acuerdo con el Proveedor que ha llevado el producto en mal estado o incompleto, el día y la hora en que va a ser devuelto el producto en mención.

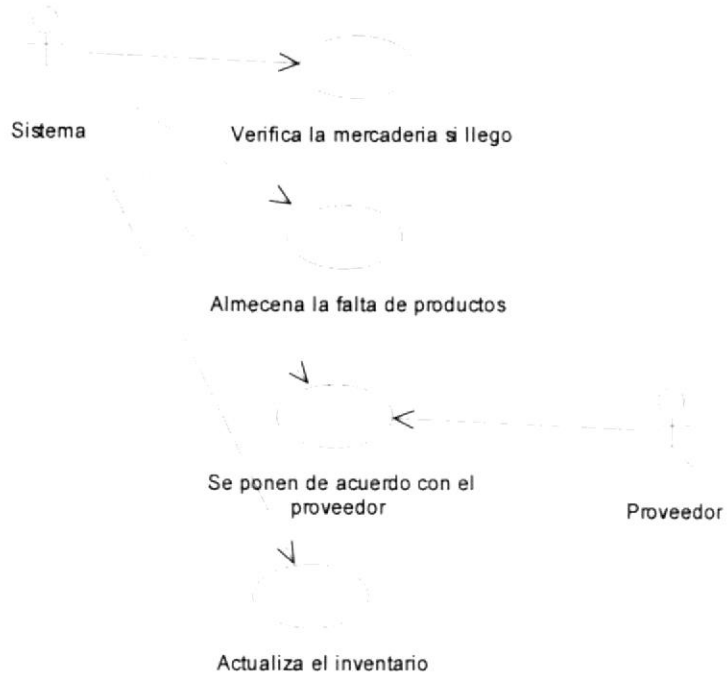
4. El Sistema de Compras una vez llegado el producto faltante, procede a actualizar la Orden de Compra y así se actualice en el Sistema de Cuentas por Pagar e Inventario.

**Observaciones:**

## 3.2.8. PROCESO DE LLEGADA MERCADERÍA DEVUELTA

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE SECUENCIA**




**Observaciones:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 de 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

## NARRATIVA DEL DIAGRAMA DE SECUENCIA

5. Compras verifica si es que la mercadería ha llegado en buen estado y/o completo; si no es así, entonces procede a registrar faltante de producto en la Orden de Compra.

6. El Sistema de Compras registra el faltante de producto en la Orden de Compra y así actualizar en el Sistema de Cuentas por Pagar e Inventario.

7. Compras procede a llegar a un acuerdo con el Proveedor que ha llevado el producto en mal estado o incompleto, el día y la hora en que va a ser devuelto el producto en mención.

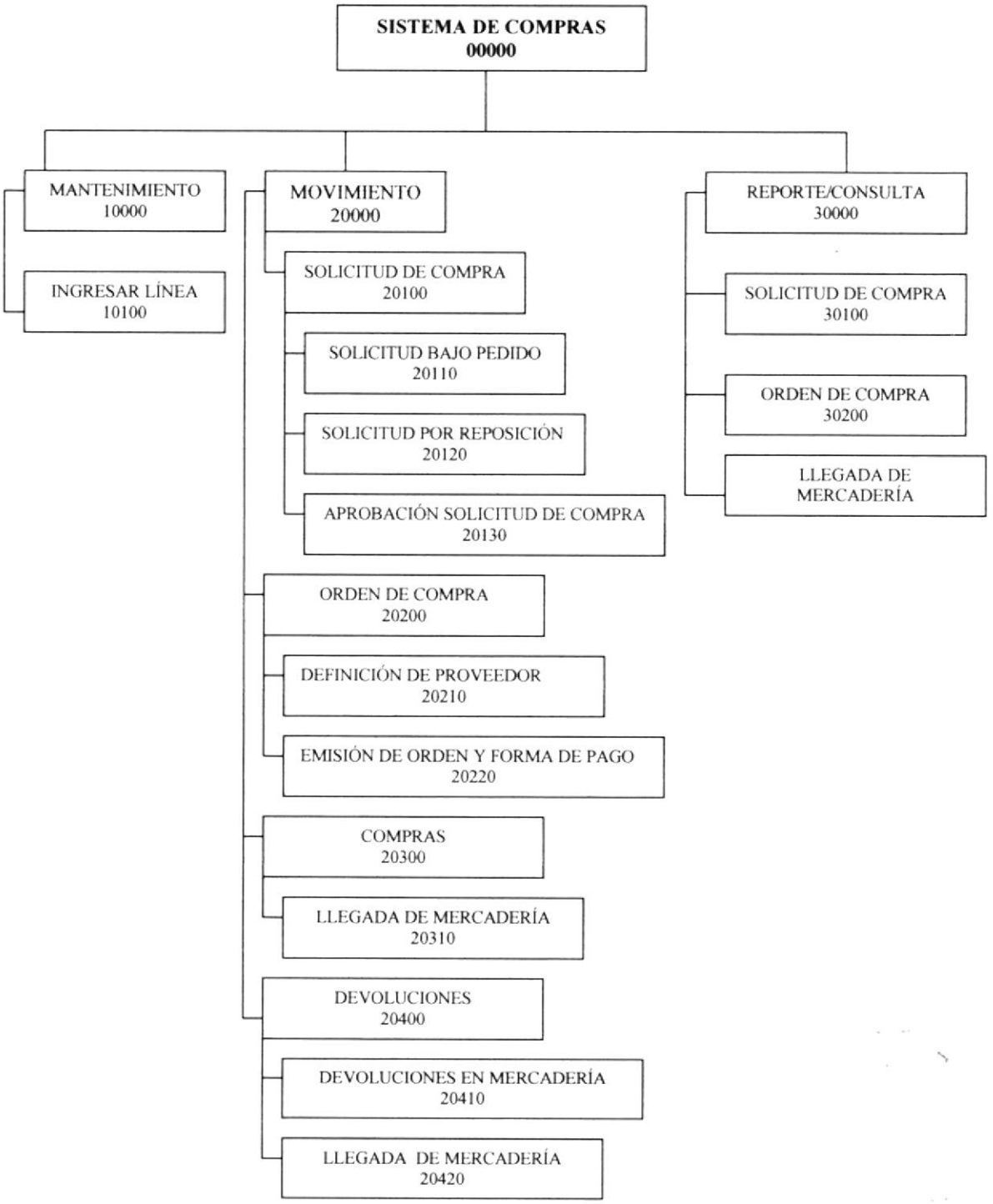
8. El Sistema de Compras una vez llegado el producto faltante, procede a actualizar la Orden de Compra y así se actualice en el Sistema de Cuentas por Pagar e Inventario.

**Observaciones:**

# Capítulo 4

## Diagrama Jerárquico del Sistema

# 4. DIAGRAMA JERÁRQUICO DEL SISTEMA



## 4.1. SISTEMA DE COMPRAS

### 4.1.1. SISTEMA DE COMPRAS (00000)

Permite el enlace con todas las opciones del Sistema de Compras para los diferentes procesos y elaboración de consultas y reportes necesarios para el negocio.

### 4.1.2. MANTENIMIENTO (10000)

Permite realizar el mantenimiento de los proveedores.

### 4.1.3. MOVIMIENTOS (20000)

Permite realizar los diferentes movimientos, para realizar una solicitud de compra, orden de compra, compras y devoluciones en el sistema.

### 4.1.4. REPORTES / CONSULTAS (30000)

Permite realizar las diferentes consultas para el control de la llegada de la mercadería.

### 4.1.5. ACERCA DE... (40000)

Esta opción proporciona información sobre el Sistema de Compras.

## 4.2. MENÚ MANTENIMIENTO

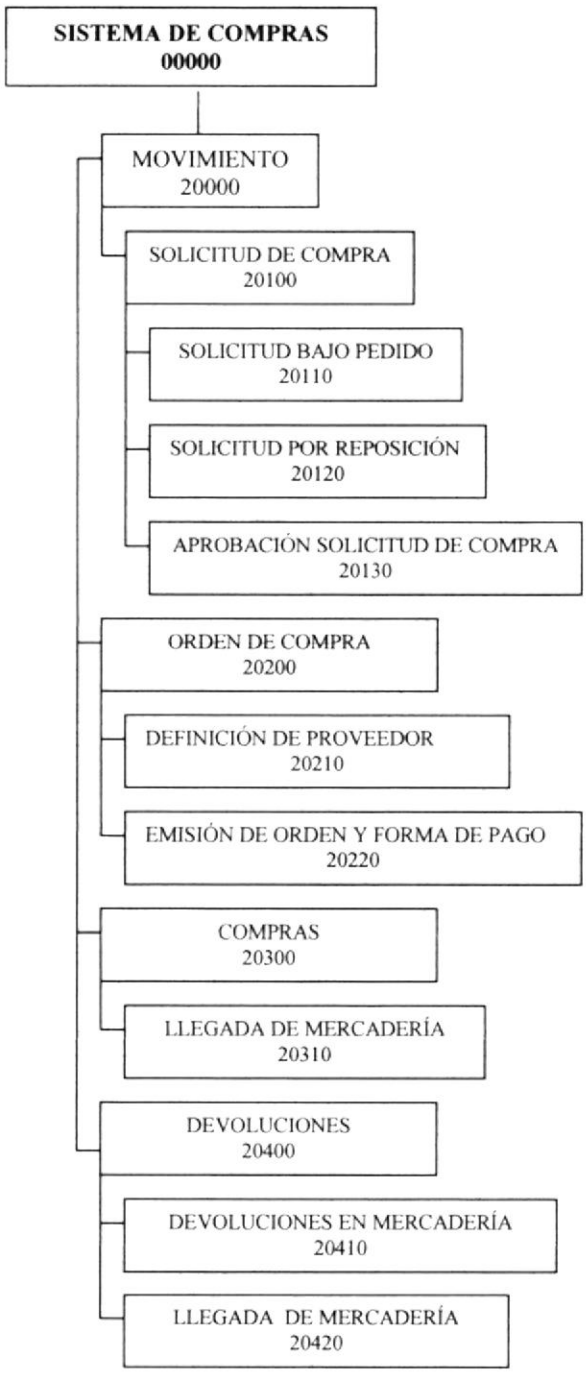
### 4.2.1. INGRESAR LÍNEA

En esta opción permite realizar el mantenimiento de los proveedores con sus productos.

### 4.2.2. SALIR

Esta opción permitirá salir del Sistema.

## 4.3. MENÚ MOVIMIENTOS




### 4.3.1. SOLICITUD COMPRA

Permite registrar todos los pedidos de productos que se necesiten para generar una orden de compra.

### 4.3.1.1. SOLICITUD BAJO PEDIDO

Permite registrar todos los pedidos de materiales que se necesiten para generar una orden de compra.

### 4.3.1.2. SOLICITUD POR REPOSICIÓN

Permite registrar todos los pedidos de materiales que se necesiten cuando el stock llega a su estado mínimo.

### 4.3.1.3. APROBAR SOLICITUDES DE COMPRA

Permite registrar la aprobación de todas las solicitudes de compra, sea bajo pedido o por reposición.

## 4.3.2. ORDEN DE COMPRA

Permite registrar las órdenes de compra que se generan en la Empresa.

### 4.3.2.1. DEFINIR PROVEEDOR

Permite definir el proveedor al cual se va a realizar la compra.

BIBLIOTECA CAMPUS PEÑAS

### 4.3.2.2. EMITIR ORDEN Y FORMA DE PAGO

Permite registrar las órdenes de compra que se generan en la Empresa, y define su forma de pago dependiendo del Proveedor.

## 4.2.3. COMPRAS

Permite registrar las compras que se generan en la Empresa

### 4.2.3.1. LLEGADA DE MERCADERÍA

Permite registrar la llegada de mercadería de compras que se generan en la Empresa.

## 4.2.4. DEVOLUCIONES

Permite registrar la devolución de mercadería.

### 4.2.4.1. DEVOLUCIÓN DE MERCADERÍA

Permite registrar la devolución de mercadería de compras que se generan en la Empresa.

### 4.2.4.2. LLEGADA DE MERCADERÍA DEVUELTA

Permite registrar la llegada de mercadería devueltas por fallas en las compras que ingresan a la en la Empresa.

## 4.4. MENÚ REPORTES / CONSULTAS

### 4.4.1. SOLICITUD DE COMPRA

Permite consultar todos los pedidos de compra que se hayan registrado en la empresa.

### 4.4.2. ORDEN DE COMPRA

Permite visualizar las órdenes de compra que se hayan generado, a manera de consulta.

BIBLIOTECA CAMPUS PEÑAS

### 4.4.3. LLEGADA DE MERCADERÍA

Permite visualizar las llegadas de mercadería que se hayan registrado, a manera de consulta.

# Capítulo 5

# Estandarizaciones de Formatos

# 5. ESTANDARIZACIONES DE FORMATOS

## 5.1. FORMATOS DE ALMACENAMIENTO DE INFORMACIÓN

Para facilitar la identificación de los elementos del sistema, como son:

- ✓ Nombre del sistema.
- ✓ Nombre de la base de datos.
- ✓ Nombres de programas.
- ✓ Menúes.
- ✓ Formatos de pantallas y reportes.

Se han establecido los siguientes estándares:

### 5.1.1. FORMATO PARA EL NOMBRE DEL PROYECTO

El formato del nombre del proyecto tanto físico como lógico consta de hasta 20 caracteres, el cual se describe a continuación:

**XXXXXX . XXX**

→ Extensión del proyecto jcw.

→ Nombre mnemónico.

Ejemplo:

**Compras . jcw**

→ Extensión del proyecto jcw.

→ Nombre mnemónico.

Tipos de archivos de JAVA y sus extensiones más comunes.

| ARCHIVO | EXTENSIÓN |
|---|---|
| PROYECTO | jcw |
| BASE DE DATOS | mdb |
| FORMULARIO | frm |
| ETIQUETA | lbl |
| MENU | mnu |
| PROGRAMA | java |
| REPORTE | cls |
| CLASE | class |

## 5.1.2. FORMATO PARA EL NOMBRE DE LAS CLASES

El formato de nombres para clases constan de hasta 20 caracteres, el cual se describe a continuación:

**XXX XXXXXXXXX . XXXX**

→ Extensión java.

→ Nombre mnemónico.

→ Abreviación del Programa

Ejemplo:

**COMPrincipal . java**

→ Extensión JAVA.

→ Nombre mnemónico.

→ Abreviación del Programa

BIBLIOTECA CAMPUS

## 5.1.3. FORMATO PARA EL NOMBRE DE LA BASE DE DATOS

El formato para el nombre de la base de datos constan de hasta 20 caracteres, el cual se describe a continuación:

**XXXXXXX.XXX**

→ Extensión de la Base de Datos

→ Nombre de la Base de Datos.

Ejemplo:

**Compras.MDB**

→ Extensión de la Base de Datos

→ Nombre de la Base de Datos.

## 5.1.4. FORMATO PARA EL NOMBRE DE FORMAS Y REPORTES

El formato del nombre de formas y reportes tanto físico como lógico consta de hasta 20 caracteres, el cual se describe a continuación:

**XXX _ XXXXXXXX . XXX**

→ Extensión java,cls.

→ Nombre mnemónico.

→ Abreviación de Forma

Ejemplo:

**COMConOrdCompra.java**

→ Extensión java, cls.

→ Nombre mnemónico.

→ Abreviación de Forma

## 5.1.5. FORMATO PARA EL NOMBRE DE LAS TABLAS

El formato del nombre para las tablas de la base de datos constan de hasta 20 caracteres, el cual se describe a continuación:

**XXX _ X _ XXXXXX . XXX**

→ Extensión de la Tabla.

→ Nombre descriptivo de la Tabla.

→ Tipo de Tabla.

→ Identificación del Módulo

**IDENTIFICACIÓN DE TABLAS POR TIPO**

| *ID* | *TABLA* |
|---|---|
| M | Maestro |
| T | Transacción |
| H | Histórico |
| E | Eventual |

Ejemplo:

**COM _ M _ CABOC . DBF**

→ Extensión de la Tabla.

→ Nombre descriptivo de la Tabla.

→ Tipo de Tabla.

→ Identificación del Módulo.

## DICCIONARIO DE TABLAS

| *CÓDIGO* | *DESCRIPCIÓN* |
|---|---|
| COM_M_CABSC | Maestra de la Cabecera de la Solicitud de Compra |
| COM_M-CABOC | Maestra de Cabecera de la Orden de Compra |
| COM_M-LXPROV | Maestra de Línea Por Proveedor |
| COM_T_DETSC | Transaccional de Detalle de Solicitud de Compra |
| COM_T_DETOC | Transaccional de Detalle de Orden de Compra |
| COM_T_CDMER | Transaccional de Devolución de Mercadería |
| COM_H_DDMER | Histórica de Devolución de Mercadería |

## 5.2. FORMATOS DE ESTÁNDARES DE PROGRAMACIÓN

### 5.2.1. FORMATO PARA EL NOMBRE DE LOS CONTROLES

Tipos de controles de JAVA y sus iniciales más comunes:

**XXX XXXXXXXX**

→ Representa el nombre del control.

→ Iniciales propias de cada control.

LAS INICIALES DE CADA CONTROL SON LAS SIGUIENTES:

| *CONTROLES* | *INICIALES* |
|---|---|
| JLABEL | lbl |
| JBUTTON | btn |
| JPANEL | pnl |
| JTEXTFIELD | txt |
| JDESKTOPPANE | jdp |
| JSCROLLPANEL | scr |
| JCOMBOBOX | Cmb |
| JTABLE | tbl |
| JCHECKBOX | chk |
| JRADIOBUTTON | opt |
| JMENU | mnu |
| JMENUITEM | mnu |
| JDATEFIELD | txt |
| JTOOLBAR | tbl |

Ejemplos:

Un Combo para editar la bodega elegida : CmbBodega
Un TextField para presentar un número de solicitud : txtNoSolicitud

## 5.2.2. FORMATO PARA EL NOMBRE DE LOS CAMPOS DE LA TABLA

El formato del nombre para los campos de las tablas que forman parte de una base de datos constan de hasta 20 letras:

**XXX XXXXX**

→ Nombre descriptivo del Campo en la Tabla.

→ Abreviación del nombre de la Tabla.

Ejemplo:

En este campo se guardará el número de la Orden de Compra: CONMERO.
En este campo se guardará el tipo de Solicitud de Compra : CSTIPO

***Para las variables se debe utilizar los códigos definidos para los campos de la base de datos antes del cuerpo.***

## 5.3. FORMATOS DE ESTÁNDARES PARA PRESENTACIONES DE PANTALLAS

### 5.3.1. FORMATOS DE COLORES

Los colores son una propiedad muy importante en la interfase visual, por lo cual lo más recomendable es utilizar los sets de colores que trae Windows y usarlos como default.

### 5.3.2. FORMATOS DE FONTS

El hecho de variar el tamaño y ancho de los fonts, pueda darle o quitarle importancia al texto, evite utilizar italic y serif fonts, ya que usualmente son difíciles de leer.

- ✓ Limite el número de fonts y estilos en su interfase, usar demasiados es confuso.
- ✓ Utilice negrita para el texto en los botones y cuando quiera resaltar alguna información.

### 5.3.3. FORMATOS DE CAPITALIZACIÓN

Para mostrar texto en el menú, utilice el método .setMnemonic(char) y ponga la letra inicial en mayúscula.

Para mostrar texto en botones, ponga en mayúsculas la primera letra de cada palabra, menos en los artículos y preposiciones intermedios.

Las etiquetas de campos, option buttons, check boxes, text boxes, etc., deben de ir solamente la primera letra en mayúscula.

### 5.3.4. FORMATOS DE LOCALIZACIÓN DE BOTONES

Ubique los botones principales en la parte inferior de izquierda a derecha. Si necesita botones específicos para cada grupo, ubíquelos internamente.

## 5.3.5. FORMATOS DE ÍCONOS

Utilice íconos en su interface para representar objetos o tareas. Cuando diseñe un ícono tenga en cuenta que debe representar su función.

| *ICONOS* | *DESCRIPCIÓN* |
|---|---|
| | Guardar Nuevo Registro |
| | Modificar registro |
| | Imprimir registro |
| | Eliminar Registro |
| | Cancelar |
| | Salir de la opción |

# Capítulo 6

## Plan de Códigos

# 6. PLAN DE CÓDIGOS

A continuación detallamos el Plan de Código que utilizaremos en el Sistema:

## 6.1. SOLICITUD DE COMPRA

| TABLA DE CABECERA DE SOLICITUD DE COMPRA(COM_M_CABSC) | | | |
|---|---|---|---|
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Solicitud de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CSNMERO |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

| TABLA DE DETALLE DE SOLICITUD DE COMPRA (COM_T_DETSC) | | | |
|---|---|---|---|
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Detalle Solicitud de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | DSNMERO |
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Solicitud de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CSNMERO |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

## 6.2. ORDEN DE COMPRA

| TABLA DE CABECERA DE ORDEN DE COMPRA (COM_M_CABOC) | | | |
|---|---|---|---|
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Orden de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CONMERO |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

| TABLA DE DETALLE ORDEN DE COMPRA (COM_T_DETOC) | | | |
|---|---|---|---|
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Detalle Orden de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | DCNMERO |
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Orden de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CONMERO |
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Solicitud de Compra |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CSNMERO |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

## 6.3. DEVOLUCIÓN DE MERCADERÍA

| TABLA DE CABECERA DEVOLUCIÓN DE MERCADERÍA (COM_T_CDMER) | | | |
|---|---|---|---|
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Devolución de Mercadería |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CDNMERO |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

| TABLA DE HISTORIAL DETALLE DEVOLUCION MERCADERÍA (COM_H_DDMER) | | | |
|---|---|---|---|
| **TIPO** | NUMERICO | **NOMBRE LÓGICO** | Número Historial de Detalle Devolución de Mercadería |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | DDNMERO |
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Número de Cabecera Devolución de Mercadería |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | CDNMERO |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

## 6.4. LÍNEA POR PROVEEDOR

| TABLA DE LÍNEA POR PROVEEDOR (COM_M_LXPROV) | | | |
|---|---|---|---|
| **TIPO** | NUMERICO | **NOMBRE LÓGICO** | Código Proveedor |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | Pro_codigo |
| **TIPO** | NUMÉRICO | **NOMBRE LÓGICO** | Código de la Línea |
| **LONGITUD** | 5 | **NOMBRE FÍSICO** | Pro_linea |
| **FORMATO** | | | |
| **99999**<br>Ejemplo:<br>**00001**<br>→ Número Secuencial de 5 dígitos. | | | |

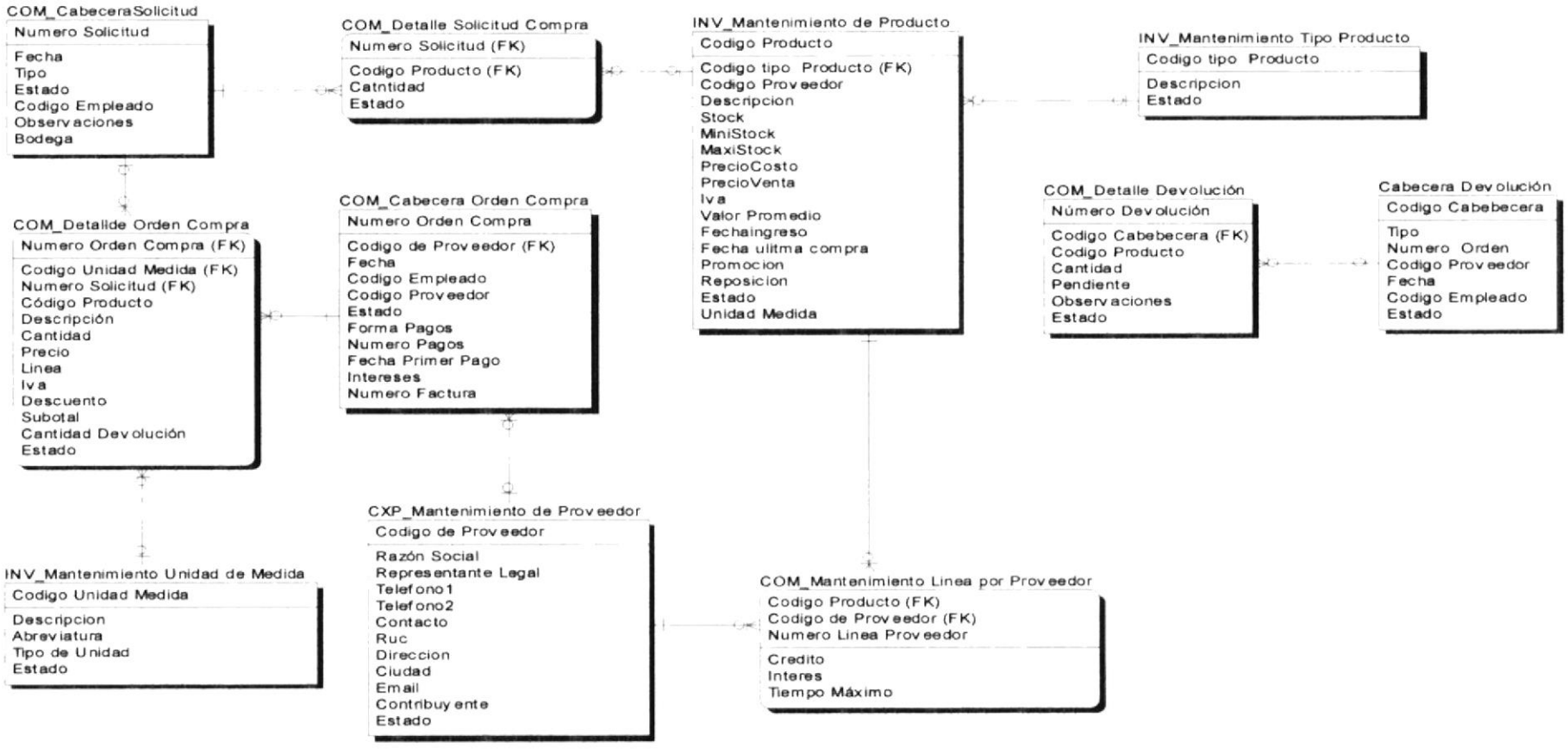
# Capítulo 7

# Modelo Entidad Relación

# 7. MODELO ENTIDAD - RELACIÓN

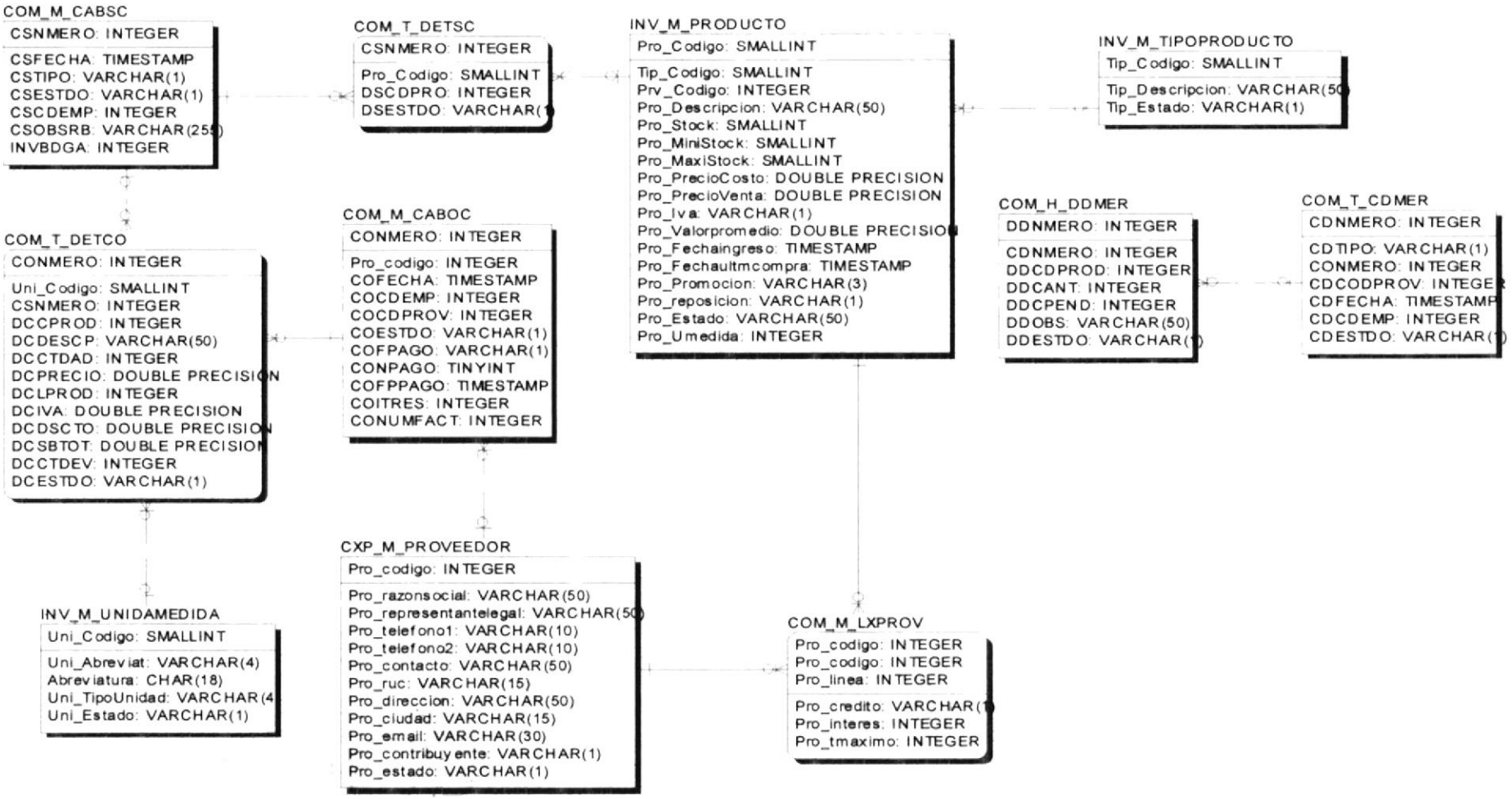
## 7.1. MODELO LÓGICO

**Modelo Lógico**



# 7.2. MODELO FÍSICO

**Modelo Físico**



# Capítulo 8

## Diccionario de Datos

# 8. DICCIONARIO DE DATOS

## 8.1. TABLA COM_M_CABSC

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1 | |
|---|---|---|---|
| | | Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_M_CABSC** | Tipo:<br>**Maestra** | Long. Reg.:<br>**282 bytes** |
| Descripción: *Tabla Maestra de Cabecera de Solicitud de Compras.* | | | |

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | CSNMERO | Numérico | 5 | Número Cabecera Solicitud Compra | ✓ | | |
| 2 | CSFECHA | Fecha/Hora | 10 | Fecha de la Solicitud de Compra | | | ✓ |
| 3 | CSTIPO | Texto | 1 | Tipo de Solicitud de Compra | | | ✓ |
| 4 | CSESTDO | Texto | 1 | Estado de la Solicitud de Compra | | | ✓ |
| 5 | CSCDEMP | Numérico | 5 | Código del Empleado | | | ✓ |
| 6 | CSOBSRB | Texto | 255 | Observación en Solicitud Compra | | | ✓ |
| 7 | INVBDGA | Numérico | 5 | Número de Bodega | | | ✓ |

**Definición:**

1. PK: CSNMERO.
2. FK:

## 8.2. TABLA COM_T_DETSC

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 | |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_T_DETSC** | Tipo:<br>**Transaccional** | Long. Reg.:<br>**21 bytes** | |
| Descripción: *Tabla Transaccional de Detalle Solicitud de Compra* | | | | |

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | DSNMERO | Autonumérico | 5 | Número Detalle Solicitud Compra | ✓ | | |
| 2 | CSNMERO | Numérico | 5 | Número Cabecera Solicitud Compra | | ✓ | |
| 3 | DSCDPRO | Numérico | 5 | Código del Producto | | | ✓ |
| 4 | DSCTDAD | Numérico | 5 | Cantidad del Producto Solicitado | | | ✓ |
| 5 | DSESTDO | Texto | 1 | Estado del Producto. | | | ✓ |

**Definición:**

1. PK: DSNMERO.
2. FK: CSNMERO.

## 8.3. TABLA COM_M_CABOC

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_M_CABOC** | Tipo:<br>**Maestra** | Long. Reg.:<br>**45 bytes** |
| Descripción: *Tabla Maestra de Cabecera de Orden de Compra.* | | | |

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | CONMERO | Numérico | 5 | Número Cabecera Orden de Compra | ✓ | | |
| 2 | COFECHA | Fecha/Hora | 10 | Fecha de Orden de Compra | | | ✓ |
| 3 | COCDEMP | Numérico | 5 | Código del Empleado | | | ✓ |
| 4 | COCDPROV | Numérico | 5 | Código del Proveedor | | | ✓ |
| 5 | COESTDO | Texto | 1 | Estado del a Orden de Compra | | | ✓ |
| 6 | COFPAGO | Texto | 1 | Forma de Pago | | | ✓ |
| 7 | COPNAGO | Numérico | 3 | Número de Pagos | | | ✓ |
| 8 | COFPPAGO | Fecha/ Hora | 10 | Fecha del Primer Pago | | | ✓ |
| 9 | COITRES | Numérico | 5 | Intereses | | | ✓ |

**Definición:**

**1.** PK: CONMERO

**2.** FK:

## 8.4. TABLA COM_T_DETCO

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_T_DETCO** | Tipo:<br>**Transaccional** | Long. Reg.:<br>**116 bytes** |
| Descripción: *Tabla Transaccional de Detalle de Orden de Compra.* | | | |

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | DCNMERO | Autonumérico | 5 | Número de Detalle Orden de Compra | ✓ | | |
| 2 | CONMERO | Numérico | 5 | Número de Cabecera Orden de Compra | | ✓ | |
| 3 | CSNMERO | Numérico | 5 | Número Cabecera Solicitud Compra | | ✓ | |
| 4 | DCCPROD | Numérico | 5 | Código del Producto | | | ✓ |
| 5 | DCDESCP | Texto | 50 | Descripción del Producto | | | ✓ |
| 6 | DCCTDAD | Numérico | 5 | Cantidad de Producto Pedido | | | ✓ |
| 7 | DCPRECIO | Numérico | 5 | Precio Unitario del Producto | | | ✓ |
| 8 | DCLPROD | Numérico | 5 | Línea del Producto | | | ✓ |
| 9 | DCUMEDA | Numérico | 5 | Unidad de Medida del Producto | | | ✓ |
| 10 | DCIVA | Numérico | 5 | Impuesto al Valor Agregado (IVA) | | | ✓ |
| 11 | DCDSCTO | Numérico | 5 | Descuento aplicado al Producto | | | ✓ |
| 12 | DCSBTOT | Numérico | 5 | Subtotal | | | ✓ |
| 13 | DCCTDEV | Numérico | 5 | Cantidad Devuelta | | | ✓ |
| 14 | DCCTPED | Numérico | 5 | Cantidad de Producto Ingresado | | | ✓ |
| 15 | DCESTDO | Texto | 1 | Estado del Producto | | | ✓ |

**Definición:**

1. PK: DCNMERO
2. FK: CONMERO, CSNMERO.

## 8.5. TABLA COM_T_CDMER

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_T_CDMER** | Tipo:<br>**Transaccional** | Long. Reg.:<br>**31 bytes** |
| Descripción: *Tabla Transaccional de Cabecera de Devolución de Mercadería.* | | | |

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | CDNMERO | Numérico | 5 | Número de Cabecera Devolución Mercadería | ✓ | | |
| 2 | CONMERO | Numérico | 5 | Número de Cabecera Orden de Compra | | ✓ | |
| 3 | CDCODPROV | Numérico | 5 | Código del Proveedor | | | ✓ |
| 4 | CDFECHA | Fecha/ Hora | 10 | Fecha de la Devolución de Mercadería | | | ✓ |
| 5 | CDCDEMP | Numérico | 5 | Código del Empleado | | | ✓ |
| 6 | CDESTD | Texto | 1 | Estado de la Mercadería Devuelta | | | ✓ |

**Definición:**

1. PK: CDNMERO.
2. FK: CONMERO.

## 8.6. TABLA COM_H_DDMER

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_H_DDMER** | Tipo:<br>**Histórica** | Long. Reg.:<br>**71 bytes** |
| Descripción: *Tabla Histórica de Detalle de Devolución de Mercadería.* | | | |

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | DDNMERO | Autoumérico | 5 | Número de Detalle Devolución Mercadería | ✓ | | |
| 2 | CDNMERO | Numérico | 5 | Número de Cabecera Devolución Mercadería | | ✓ | |
| 3 | DDCDPROD | Numérico | 5 | Descripción del Producto | | | ✓ |
| 4 | DDCANT | Numérico | 5 | Cantidad del Producto | | | ✓ |
| 5 | DDOBS | Texto | 50 | Observación | | | ✓ |
| 6 | DDESTDO | Texto | 1 | Estado del Producto | | | ✓ |

**Definición:**

1. PK: DDNMERO
2. FK: CDNMERO

## 8.7. TABLA COM_M_LXPROV

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Base de Datos:<br>**COMPRAS** | Nombre de Tabla:<br>**COM_M_LXPROV** | Tipo:<br>**Maestra** | Long. Reg.:<br>**21 bytes** |

Descripción: *Tabla Maestra de Línea por Proveedor.*

**DESCRIPCIÓN DEL REGISTRO**

| No. | NOMBRE | TIPO | LONG. | DESCRIPCION | PK | FK | NL |
|---|---|---|---|---|---|---|---|
| 1 | Pro_codigo | Numérico | 5 | Código del Proveedor | ✓ | | |
| 2 | Pro_linea | Numérico | 5 | Código de la Línea | | ✓ | |
| 3 | Pro_credito | Texto | 1 | Crédito | | | ✓ |
| 4 | Pro_interes | Numérico | 5 | Interés | | | ✓ |
| 5 | Pro_tmaximo | Numérico | 5 | Tiempo Máximo | | | ✓ |

**Definición:**

1. PK: Pro_codigo
2. FK: Pro_linea

# Capítulo 9

# Diseño de Entradas y Salidas

ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

# 9. DISEÑO DE ENTRADAS Y SALIDAS

## 9.1. DISEÑO DE ENTRADAS

**Significado de simbología a utilizarse.**

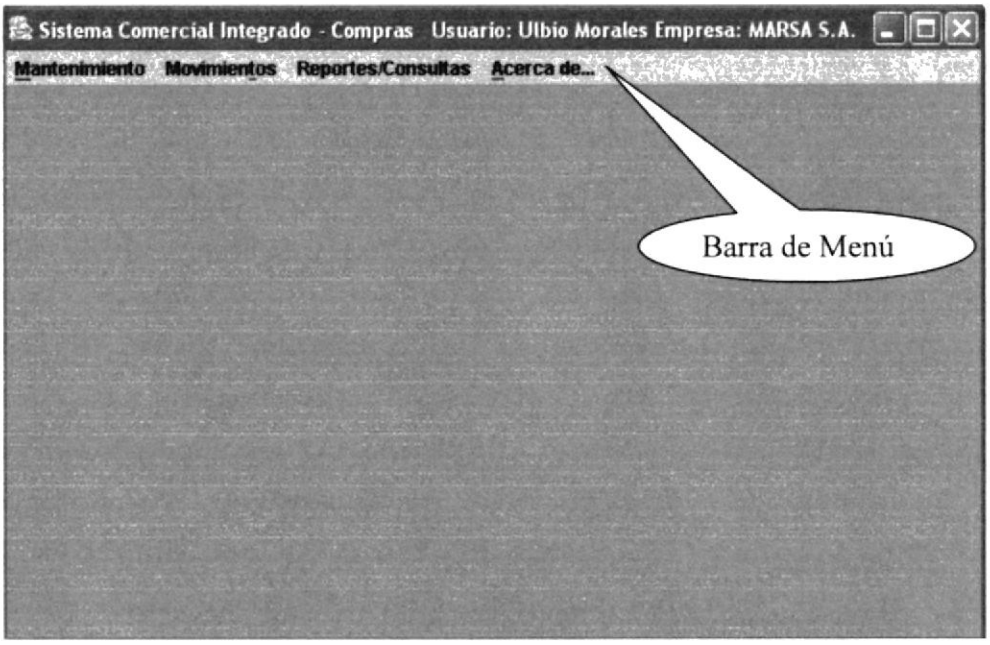
Todos los controles deben estar numerados.

| SÍMBOLOS | | DESCRIPCIÓN |
|---|---|---|
| ID | | Número que identifica el control en la pantalla . |
| NOMBRE DEL CONTROL | | Nombre que se le dio al control en tiempo de diseño. |
| OB | | Indica si el campo es o no obligatorio, debe ser marcado con un visto si es obligatorio. |
| NOMBRE DEL CAMPO | | Nombre del campo que representa ese control en la pantalla. |
| TIPO: | C | De Consulta. |
| | I | De Ingreso. |
| FUNCIÓN: | A | Si el control tiene asociada una acción y la función que tiene (primera columna). |
| | P | Si el control tiene asociada una propiedad y la función que tiene. |

## 9.1.1. MENÚ PRINCIPAL DEL SISTEMA DE COMPRAS

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú del Sistema de Compras* | | |

**DISEÑO DE PANTALLA**




**Función:**
Este formulario contiene el Menú General del Sistema de Compras, desde el cual se accede a todas las aplicaciones de compras.

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú del Sistema de Compras* | | |

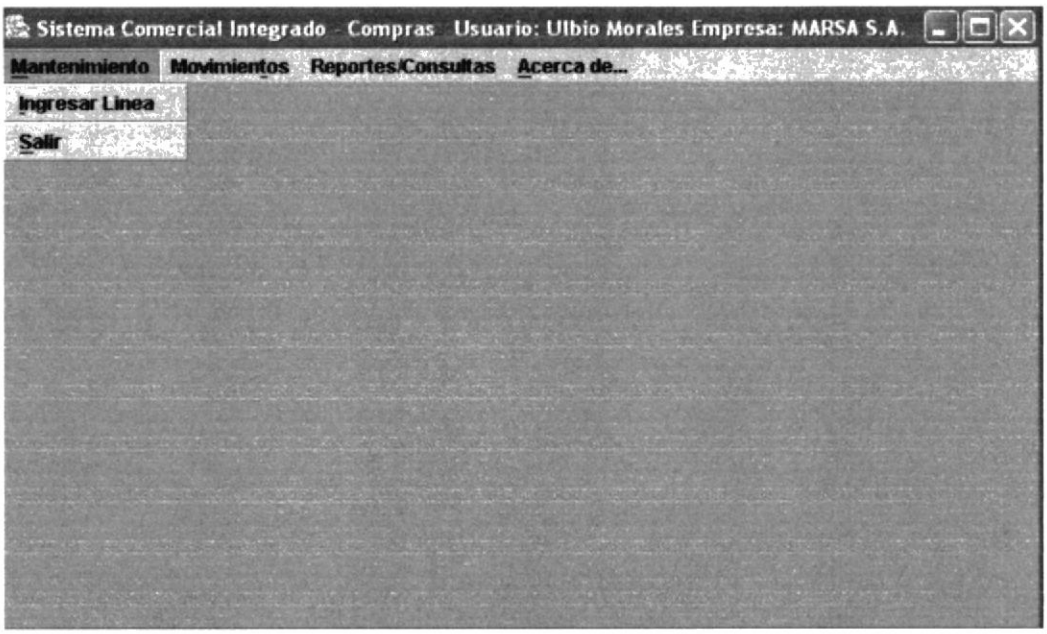
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | mnuMantenimiento | | | | A | Al dar clic se despliega el submenú de mantenimiento de proveedores y la opción Salir. |
| 2 | mnuOperaciones | | | | A | Al dar clic se despliega los submenús Solicitud Bajo Pedido, Solicitud Por Reposición, Aprobar Solicitud de Compra, Definir Proveedor, Emisión Ordenes y Forma de Pago, Llegada de Mercadería, Devolución de Mercadería. |
| 3 | mnuReportes | | | | A | Al dar clic se despliega los submenús Solicitud de Compra, Ordenes de Compra, Llegada de Mercadería. |
| 4 | mnuAcercade | | | | A | Al dar clic se despliega el submenú Acerca de. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.2. MENÚ MANTENIMIENTO

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú del Sistema de Compras, Menú Mantenimiento.* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**
Este formulario contiene el Menú General del Sistema de Compras, en este caso se observa el despliegue del submenú del menú Mantenimiento.

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú del Sistema de Compras, Menú Mantenimiento.* | | |

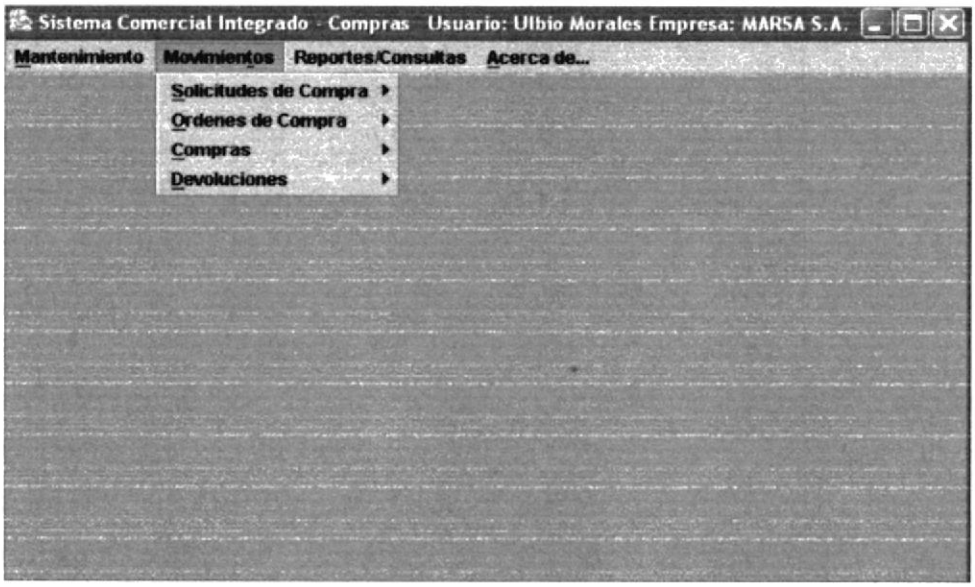
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | mnuIngresoLineas | | | | A | Al dar click se despliega el submenú de mantenimiento de ingreso de líneas. |
| 2 | mnuSalir | | | | A | Al dar clic cierra la aplicación del Sistema de Compras. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.3. MENÚ MOVIMIENTOS

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú del Sistema de Compras, Menú Movimientos.* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**
Este formulario contiene el Menú General del Sistema de Compras, en este caso se observa el despliegue del submenú del menú Movimientos.

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú del Sistema de Compras, Menú Movimientos.* | | |

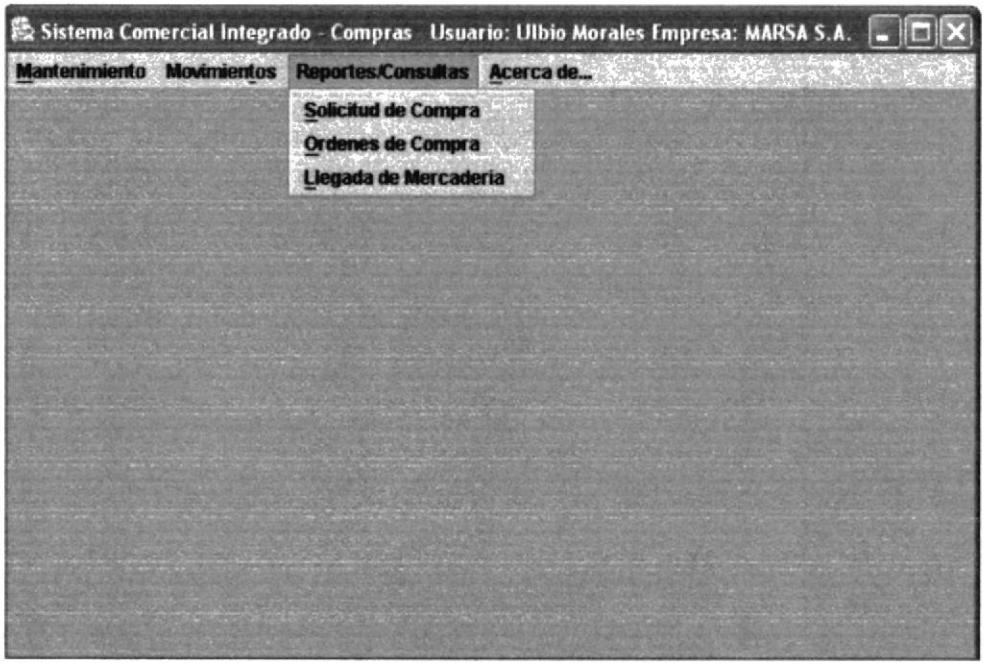
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | mnuSolicitud | | | | A | Al dar clic se ejecuta el formulario de la Solicitud de Compra Bajo Petición y por Reposición y la aprobación de las Solicitudes de Compras. |
| 2 | mnuOrdenes | | | | A<br>A | Al dar clic se ejecuta el formulario que Define al Proveedor al momento de aprobarse la Orden de Compra y de la Emisión de las Ordenes de Compras y su forma de pago. |
| 3 | mnuCompras | | | | A | Al dar clic se ejecuta el formulario de la Llegada de la Mercadería. |
| 4 | mnuDevoluciones | | | | | Al dar clic se ejecuta el formulario de la Devolución de la Mercadería y Llegada de Mercadería Devuelta. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.4. MENÚ REPORTES / CONSULTAS

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú Reportes / Consultas* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**
Este formulario contiene el Menú General del Sistema de Compras, en este caso se observa el despliegue del submenú del menú Reportes/Consultas.

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú Reportes / Consultas.* | | |

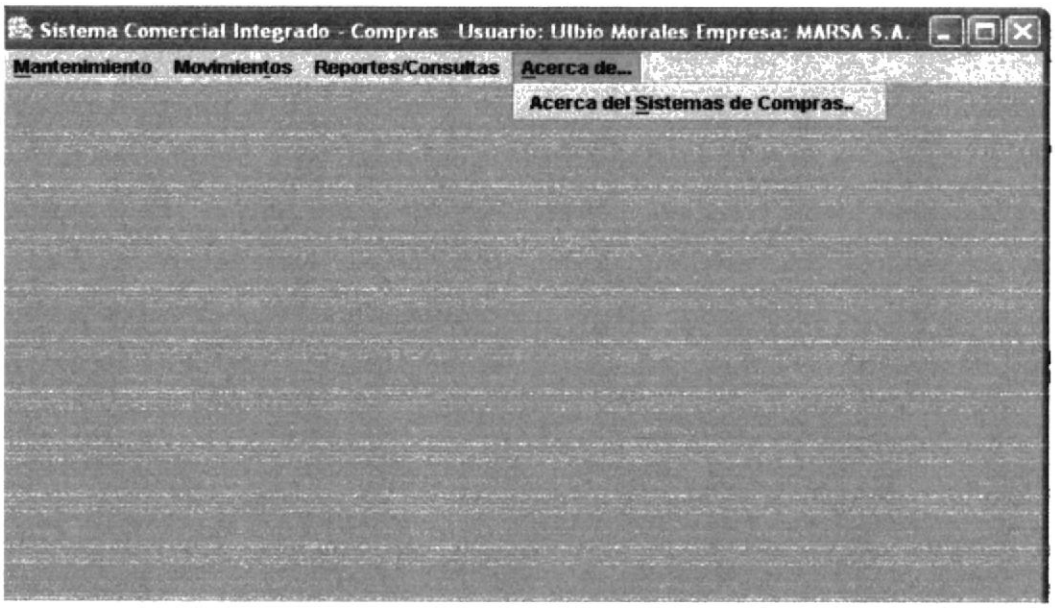
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | mnuSolCompraConsultas | | | | A | Al dar clic se ejecuta el formulario de Solicitud de Compra. |
| 2 | mnuOrdCompraConsultas | | | | A | Al dar clic se visualiza el formulario de Ordenes de Compra. |
| 3 | mnuLlegMercConsultas | | | | A | Al dar clic se visualiza el formulario de Llegada de Mercadería. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.5. MENÚ ACERCA DE...

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú Acerca de...* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**
Este formulario contiene el Menú General del Sistema de Compras, en este caso se observa el despliegue del submenú del menú Acerca de...

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMPrincipal.java** | Nombre Lógico:<br>**frmPrincipal** | |
| Descripción: *Menú Acerca de....* | | |

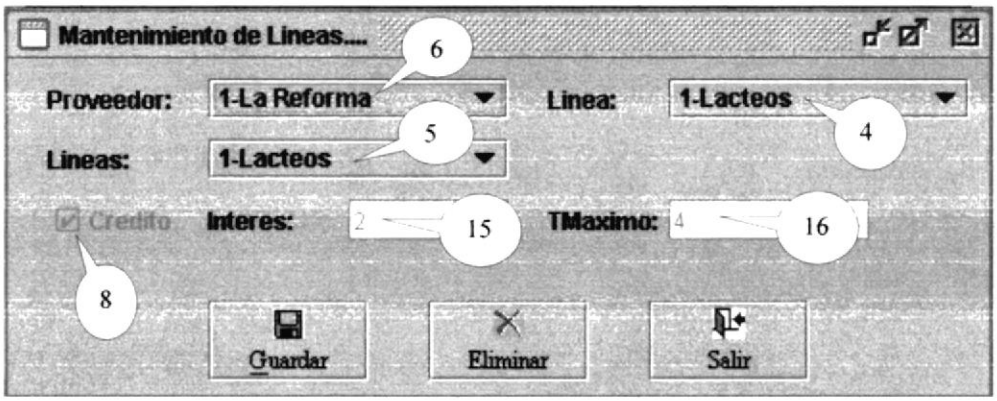
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | mnuAcercadePry | | | | A | Al dar clic se visualiza el formulario de Acerca del Sistema de Compras. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.6. MANTENIMIENTO EN LÍNEA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMMantLineaxProv.java** | Nombre Lógico:<br>**frmMantLineaxProv** | |
| Descripción: *Mantenimiento de Líneas* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMMantLineaxProv.java** | Nombre Lógico:<br>**frmMantLineaxProv** | |
| Descripción: : *Mantenimiento de Líneas* | | |

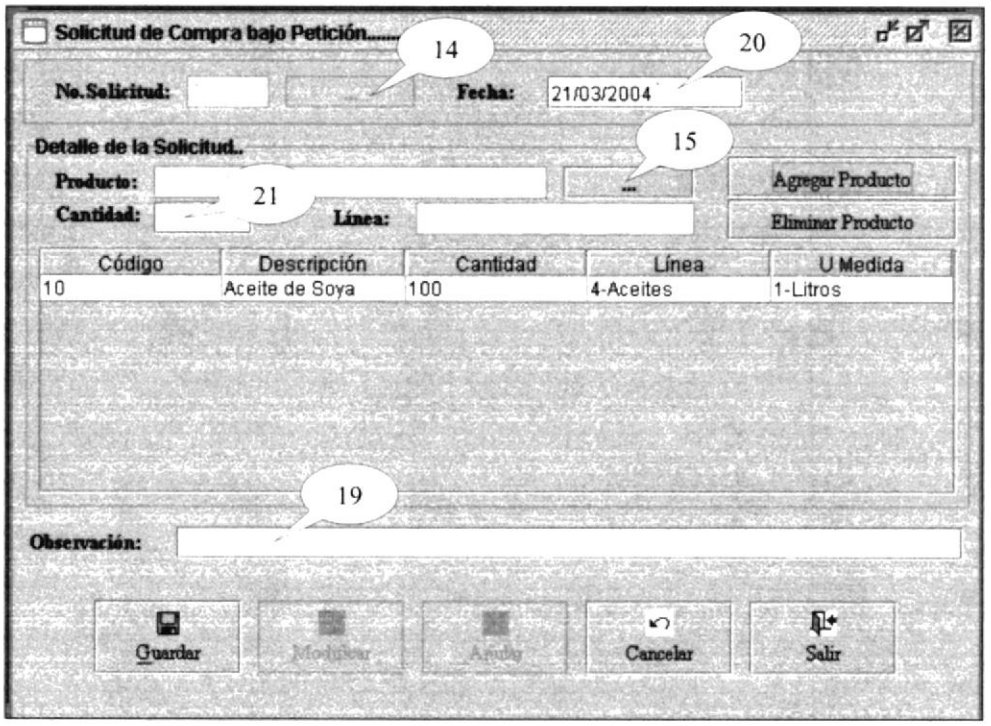
**LISTA DE CONTROLES**
**COM_M_LXPROV**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | btnBorrar | | | | A | Elimina la línea. |
| 2 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 3 | btnSalir | | | | A | Sale del Sistema |
| 4 | cmbLinea | ✓ | Pro_linea | I | P | ComboBox, registra la línea escogida |
| 5 | cmbLineaGen | ✓ | | C | P | ComboBox muestra las líneas |
| 6 | cmbProv | ✓ | Pro_codigo | C | P | ComboBox, registra los proveedores. |
| 7 | p | | | | P | Frame, agrupar varios controles. |
| 8 | chkContado | ✓ | | I | P | Checkbox, registra si es a contado. |
| 9 | chkCredito | ✓ | Pro_credito | I | P | Checkbox, registra si es a crédito. |
| 10 | lblInteres | | | | P | Label, titula una caja de texto. |
| 11 | lblLinea | | | | P | Label, titula una caja de texto. |
| 12 | lblLineaGen | | | | P | Label, titula una caja de texto. |
| 13 | lblProv | | | | P | Label, titula una caja de texto. |
| 14 | lblTmaximo | | | | P | Label, titula una caja de texto. |
| 15 | txtInteres | ✓ | Pro_interes | I | P | Label, titula una caja de texto. |
| 16 | txtTmaximo | ✓ | Pro_tmaximo | I | P | Label, titula una caja de texto. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.7. SOLICITUD DE COMPRA BAJO PEDIDO

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMSolPedido.java** | Nombre Lógico:<br>**frmSolPedido** | |
| Descripción: *Solicitud De Compra Bajo Pedido* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMSolPedido.java** | Nombre Lógico:<br>**frmSolPedido** | |
| Descripción: *Solicitud De Compra Bajo Pedido.* | | |

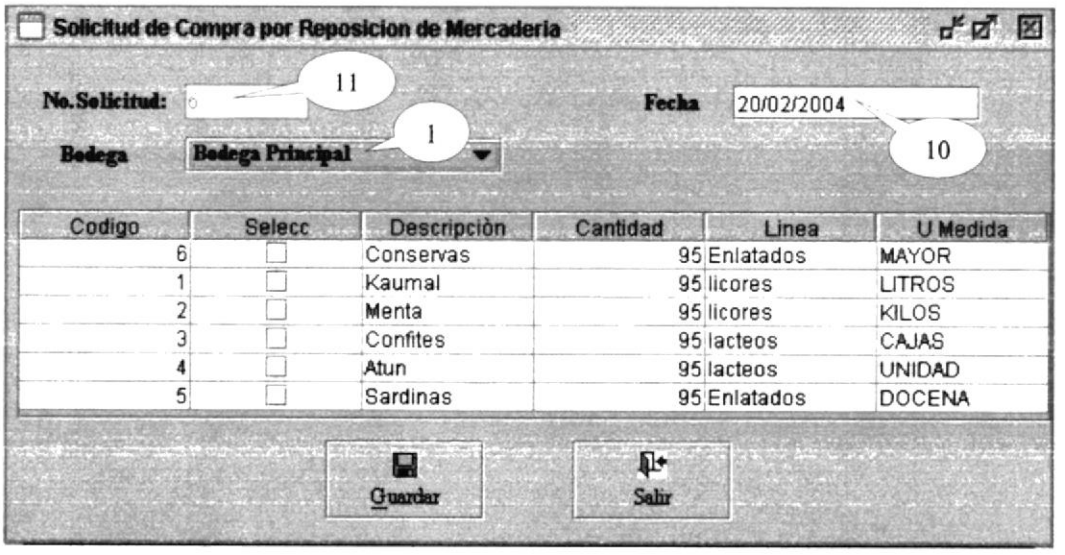
**LISTA DE CONTROLES**
**COM_M_CABSC Y COM_T_DETSC**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | p0 | | | | P | Frame, agrupar varios controles. |
| 2 | p | | | | P | Frame, agrupar varios controles. |
| 3 | p1 | | | | P | Frame, agrupar varios controles. |
| 4 | p2 | | | | P | Frame, agrupar varios controles. |
| 5 | lblNoSolicitud | | | | P | Label, titula una caja de texto. |
| 6 | lblFecha | | | | P | Label, titula una caja de texto. |
| 7 | lblProducto | | | | P | Label, titula una caja de texto. |
| 8 | lblCantidad | | | | P | Label, titula una caja de texto. |
| 9 | lblLinea | | | | P | Label, titula una caja de texto. |
| 10 | lblObservacion | | | | P | Label, titula una caja de texto. |
| 11 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 12 | btnAnular | | | | A | Elimina la línea. |
| 13 | btnSalir | | | | A | Sale del Sistema |
| 14 | btnBuscar | ✓ | | C | A | Consultar números solicitud compra |
| 15 | btnBuscar1 | ✓ | | C | A | Consultar productos disponibles |
| 16 | btnAgrProd | ✓ | | | A | Agregar producto |
| 17 | btnCancelar | | | | A | Cancelar la acción |
| 18 | btnElimProd | ✓ | | | A | Elimina producto en solicitud |
| 19 | txtObservacion | ✓ | | I | A | TextBox, permite ingresar observación. |
| 20 | txtFecha | ✓ | CSFECHA | I | A | TextBox, permite ingresar fecha |
| 21 | txtCantidad | ✓ | DSCTDAD | I | A | TextBox, permite ingresar cantidad |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.8. SOLICITUD DE COMPRA POR REPOSICIÓN DE MERCADERÍA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMSolRep.java** | Nombre Lógico:<br>**frmSolRep** | |
| Descripción: *Solicitud De Compra Por Reposición De Mercadería* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMSolRep.java** | Nombre Lógico:<br>**frmSolRep** | |
| Descripción: *Solicitud De Compra Por Reposición De Mercadería.* | | |

**LISTA DE CONTROLES**
**COM_M_CABSC Y COM_T_DETSC**

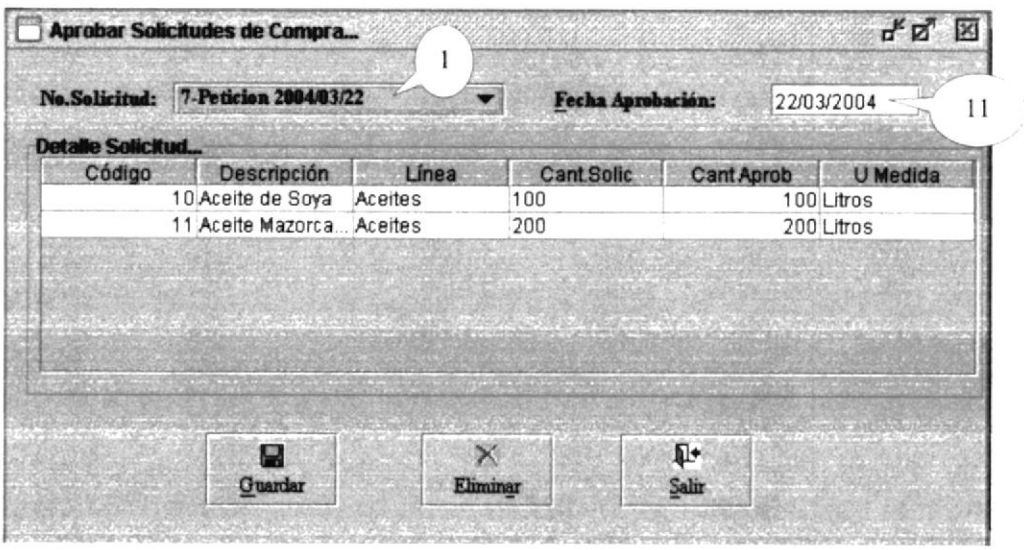
| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | CmbBodega | ✓ | INVBDGA | C | A | ComboBox, registra la línea escogida |
| 2 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 3 | btnSalir | | | | A | Sale del Sistema |
| 4 | lblBodega | | | | P | Label, titula una caja de texto. |
| 5 | lblFecha | | | | P | Label, titula una caja de texto. |
| 6 | lblNo.Solicitud | | | | P | Label, titula una caja de texto. |
| 7 | p | | | | P | Frame, agrupar varios controles. |
| 8 | p1 | | | | P | Frame, agrupar varios controles. |
| 9 | scrollPane | | | | P | ScrollPane contiene a la tabla. |
| 10 | txtFecha | ✓ | CSFECHA | I | A | TextBox, permite ingresar fecha |
| 11 | txtNoSolicitud | ✓ | CSNMERO | I | A | TextBox, permite ingresar # solicitud |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.9. APROBACIÓN DE SOLICITUD DE COMPRA

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMAprovSolCom.java** | Nombre Lógico:<br>**frmAprovSolCom** | |
| Descripción: *Aprobación de Solicitud de Compra.* | | |

**DISEÑO DE PANTALLA**




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMComAprovSolCom.java** | Nombre Lógico:<br>**frmAprovSolCom** | |
| Descripción: : *Aprobación De Solicitud De Compra.* | | |

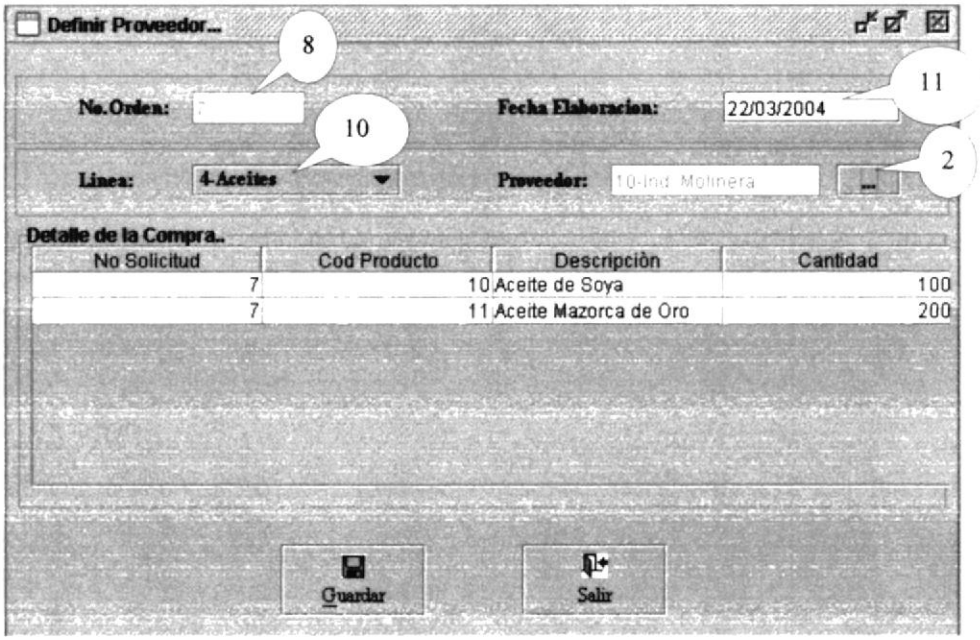
**LISTA DE CONTROLES**
**COM_M_CABSC Y COM_T_DETSC**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | CmbSolicitudes | ✓ | | C | A | ComboBox, registra la línea escogida |
| 2 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 3 | btnSalir | | | | A | Sale del Sistema |
| 4 | btnAnular | | | | A | Elimina la solicitud. |
| 5 | lblFecha | | | | P | Label, titula una caja de texto. |
| 6 | lblNo.Solicitud | | | | P | Label, titula una caja de texto. |
| 7 | p | | | | P | Frame, agrupar varios controles. |
| 8 | p1 | | | | P | Frame, agrupar varios controles. |
| 9 | p2 | | | | P | Frame, agrupar varios controles. |
| 10 | scrollPane | | | | P | ScrollPane contiene a l atabla. |
| 11 | txtFecha | ✓ | CSFECHA | C | A | TextBox, permite ingresar fecha |
| 12 | txtNoSolicitud | ✓ | CSNMERO | C | A | TextBox, permite ingresar # solicitud |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.10. DEFINICIÓN DE PROVEEDOR

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMDefinirProv.java** | Nombre Lógico:<br>**frmDefinirProv** | |
| Descripción: *Definición De Proveedor* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMDefinirProv.java** | Nombre Lógico:<br>**frmDefinirProv** | |
| Descripción: *Definición De Proveedor* | | |

**LISTA DE CONTROLES**

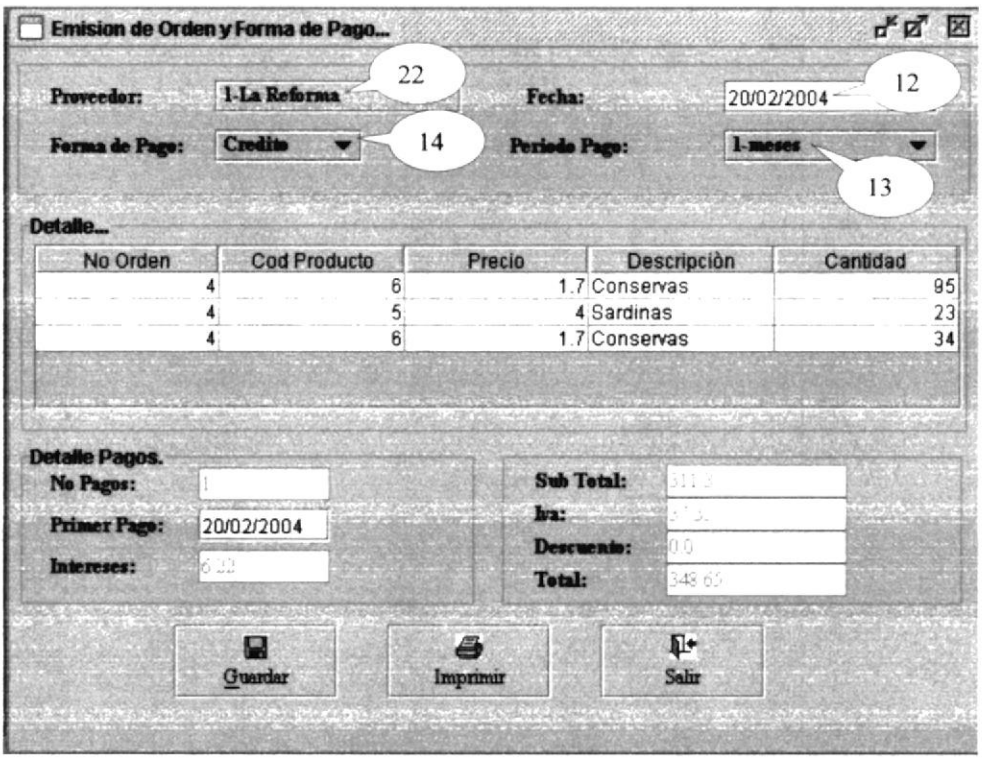
| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 2 | btnAsigProv | ✓ | | C | A | Asigna el proveedor. |
| 3 | btnSalir | | | | A | Sale del Sistema. |
| 4 | lblNoOrden | | | | P | Label, titula una caja de texto. |
| 5 | lblFecha | | | | P | Label, titula una caja de texto. |
| 6 | lblLinea | | | | P | Label, titula una caja de texto. |
| 7 | lblProveedor | | | | P | Label, titula una caja de texto. |
| 8 | txtNoOrden | ✓ | | C | A | TextBox, permite ver # orden. |
| 9 | TxtProveedor | ✓ | | 1 | A | TextBox, permite ingresar proveedor |
| 10 | CmbLineas | ✓ | | C | A | ComboBox de ingreso de líneas. |
| 11 | txtFecha | | CSFECHA | | A | TextBox, permite ingresar fecha |
| 12 | p | | | | | Frame, agrupar varios controles. |
| 13 | p1 | | | | | Frame, agrupar varios controles. |
| 14 | p2 | | | | | Frame, agrupar varios controles. |
| 15 | p3 | | | | | Frame, agrupar varios controles. |
| 16 | p4 | | | | | Frame, agrupar varios controles. |
| 17 | scrollPane | | | | | ScrollPane contiene a la tabla. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

ESPOL 1958
BIBLIOTECA CAMPUS PEÑAS

## 9.1.11. EMISIÓN DE ORDENES Y FORMA DE PAGO

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMEmiPagoOrdCom.java** | Nombre Lógico:<br>**frmEmiPagoOrdCom** | |
| Descripción: *Emisión De Ordenes Y Forma De Pago* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMEmiPagoOrdCom.java** | Nombre Lógico:<br>**FrmEmiPagoOrdCom** | |
| Descripción: *Emisión De Ordenes Y Forma De Pago.* | | |

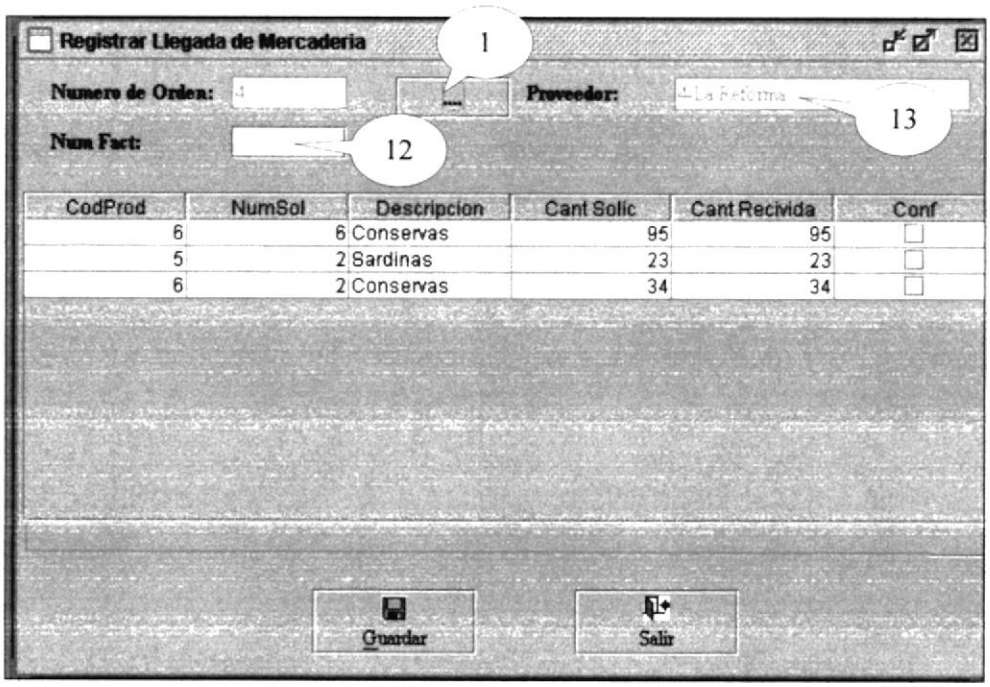
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 2 | btnImprimir | | | | A | Imprime la orden de compra. |
| 3 | btnSalir | | | | A | Sale del sistema. |
| 4 | lblDescuento | | | | P | Label, titula una caja de texto. |
| 5 | lblFecha | | | | P | Label, titula una caja de texto. |
| 6 | lblFormaPago | | | | P | Label, titula una caja de texto. |
| 7 | lblProveedor | | | | P | Label, titula una caja de texto. |
| 8 | lblIntereses | | | | P | Label, titula una caja de texto. |
| 9 | lblIva | | | | P | Label, titula una caja de texto. |
| 10 | lblNPagos | | | | P | Label, titula una caja de texto. |
| 11 | lblPeriodoPago | | | | P | Label, titula una caja de texto. |
| 12 | txtFecha | | | | A | ComboBox que ingresa la fecha. |
| 13 | txtPeriodoPago | | | | A | ComboBox que ingresa periodo pago. |
| 14 | txtFormaPago | | | | A | ComboBox que ingresa forma pago. |
| 15 | txtDescuento | | | C | A | TextBox, permite ingresar descuento |
| 16 | txtIntereses | | | C | A | TextBox, permite ingresar interes |
| 17 | txtIva | | | C | A | TextBox, permite ingresar el IVA |
| 18 | txtNPagos | | | C | A | TextBox, permite ingresar # pagos |
| 19 | txtSubTotal | | | C | A | TextBox, permite ingresar subtotal |
| 20 | txtTotal | | | C | A | TextBox, permite ingresar el total. |
| 21 | txtProveedor | ✓ | | I | A | TextBox, permite ingresar proveedor |
| 22 | CmbProveedores | ✓ | | I | A | ComboBox que ingresa el proveedor |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.12. REGISTRO DE LLEGADA DE MERCADERÍA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMLlegMercaderia.java** | Nombre Lógico:<br>**frmLlegMercaderia** | |
| Descripción: *Registro De Llegada De Mercadería* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2 |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMLlegMercaderia.java** | Nombre Lógico:<br>**frmLlegMercaderia** | |
| Descripción: *Registro De Llegada De Mercadería* | | |

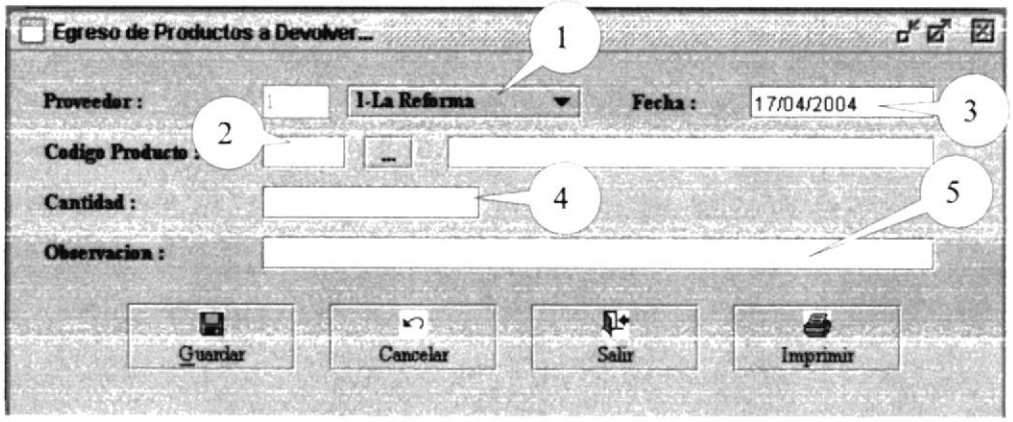
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | btnBuscar | | | | A | Búsqueda de la orden de compra. |
| 2 | btnGuardar | | | | A | Ejecuta el código ingresado. |
| 3 | btnSalir | | | | A | Sale del sistema. |
| 4 | codEmp | | | | A | Código del empleado. |
| 5 | datamodel | | | | P | Tabla Abstracta. |
| 6 | iBand | | | | P | Bandera que me controla validación. |
| 7 | lblNoOrdenCom | | | | P | Label, titula una caja de texto. |
| 8 | lblNumFact | | | | P | Label, titula una caja de texto. |
| 9 | lblProveedor | | | | P | Label, titula una caja de texto. |
| 10 | txtFecha | | CSFECHA | | A | TextBox, permite ingresar fecha. |
| 11 | txtNoOrdenCom | ✓ | | C | A | TextBox, permite ingresar orden. |
| 12 | txtNumFact | ✓ | | I | A | TextBox, permite ingresar factura. |
| 13 | txtProveedor | ✓ | | C | A | TextBox, permite ingresar proveedor. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.13. DEVOLUCIÓN DE MERCADERÍAS

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMDevolucionMerc.java** | Nombre Lógico:<br>**frmDevolucionMerc** | |
| Descripción: *Egreso de Productos a devolver.* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMDevolucionMerc.java** | Nombre Lógico:<br>**frmDevolucionMerc** | |
| Descripción: *Registro de Egreso de Productos a devolver.* | | |

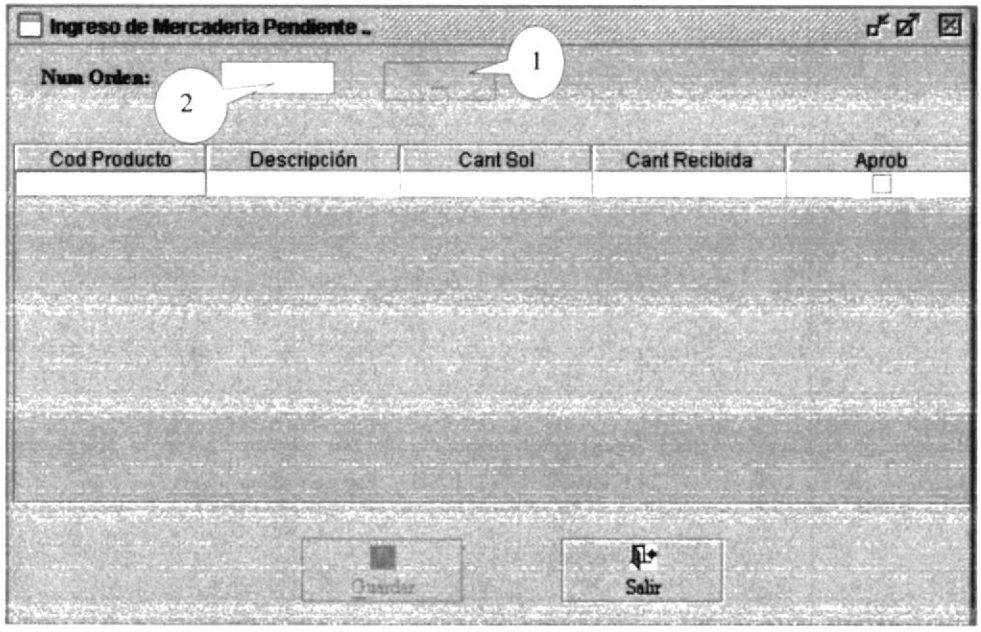
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | cmbProveedores | ✓ | | I | A | ComboBox que ingresa el proveedor. |
| 2 | txtCodProd | ✓ | | C | A | TextBox, permite ingresar código del producto. |
| 3 | txtFecha | | | | P | TextBox, permite ingresar la fecha. |
| 4 | txtCantidad | ✓ | | | A | TextBox, permite ingresar cantidad. |
| 5 | txtObservacion | | | I | P | TextBox, permite ingresar observación. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.14. LLEGADA DE MERCADERÍA PENDIENTE

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMLlegDevMercaderia.java** | Nombre Lógico:<br>**frmLlegDevMercaderia** | |
| Descripción: *Registro del Ingreso de la Mercadería Pendiente.* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMLlegDevMercaderia.java** | Nombre Lógico:<br>**frmLlegDevMercaderia** | |
| Descripción: *Registro del Ingreso de la Mercadería Pendiente..* | | |

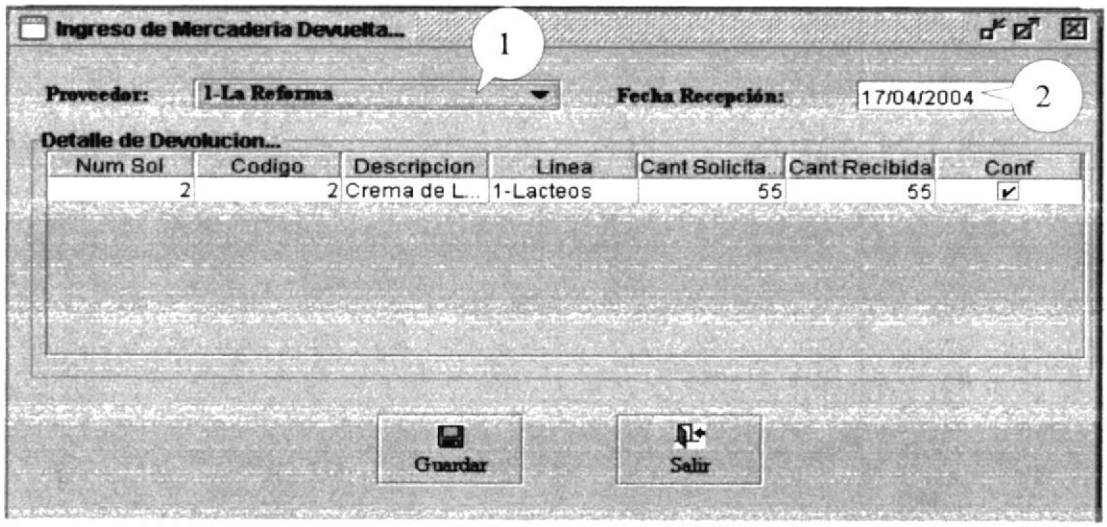
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | btnBuscar | ✓ | | C | A | Búsqueda de la orden de compra. |
| 2 | txtNoOrdenCom | ✓ | | I | A | TextBox, permite ingresar orden. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

# 9.1.15. LLEGADA DE MERCADERÍA POR DEVOLUCIÓN

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMLlegMercaderiaDevolucion.java** | Nombre Lógico:<br>**FrmLlegMercaderiaDevolucion** | |
| Descripción: Ingreso de la Llegada de Mercadería por Devolución. | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMLlegMercaderiaDevolucion.java** | Nombre Lógico:<br>**frmLlegMercaderiaDevolucion** | |
| Descripción: Ingreso de la Llegada de Mercadería por Devolución. | | |

**LISTA DE CONTROLES**

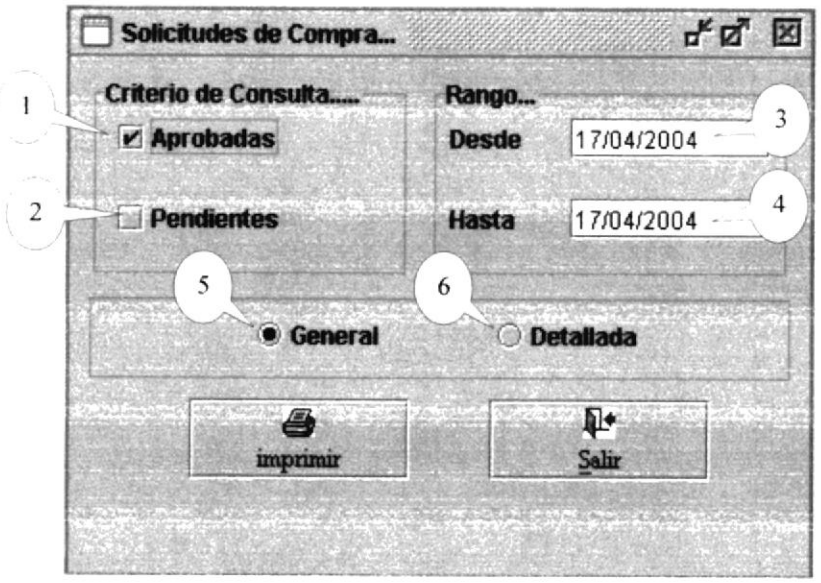
| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | CmbProveedores | ✓ | | C | A | ComboBox que ingresa el proveedor. |
| 2 | txtFecha | ✓ | | C | P | TextBox, permite ingresar la fecha. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

BIBL... C
CAMPUS
PEÑAS

## 9.1.16. REPORTE DE SOLICITUD DE COMPRA

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMConsSolCompra.java** | Nombre Lógico:<br>**frmConsSolCompra** | |
| Descripción: *Reporte de Solicitud De Compra* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

ESPOL BIBLIOTECA CAMPUS PEÑAS

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMConsSolCompra.java** | Nombre Lógico:<br>**frmConsSolCompra** | |
| Descripción: *Reporte de Solicitud De Compra.* | | |

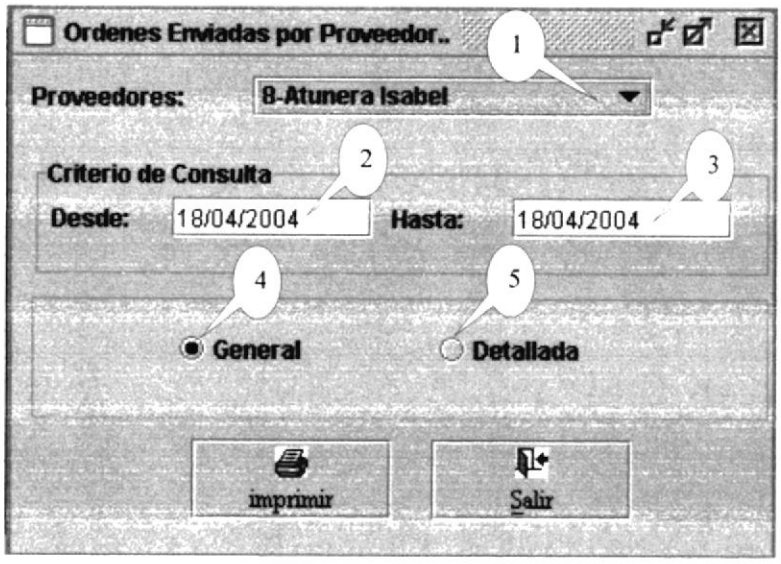
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | Aprobadas | ✓ | | C | A | ComboBox que marca las aprobadas. |
| 2 | Pendientes | ✓ | | C | A | ComboBox que marca las pendientes. |
| 3 | txtFechaIni | ✓ | | I | A | TextBox, permite ingresar la fecha inicial. |
| 4 | txtFechaFin | ✓ | | I | A | TextBox, permite ingresar la fecha final. |
| 5 | General | ✓ | | C | A | Botón para elegir las aprobadas. |
| 6 | Detallada | ✓ | | C | A | Botón para elegir las detalladas. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.17. REPORTE DE ÓRDENES DE COMPRA

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMConOrdCompra.java** | Nombre Lógico:<br>**frmConOrdCompra** | |
| Descripción: *Reporte de las Órdenes de Compra* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMConOrdCompra.java** | Nombre Lógico:<br>**frmConOrdCompra** | |
| Descripción: *Reporte de las Órdenes de Compra* | | |

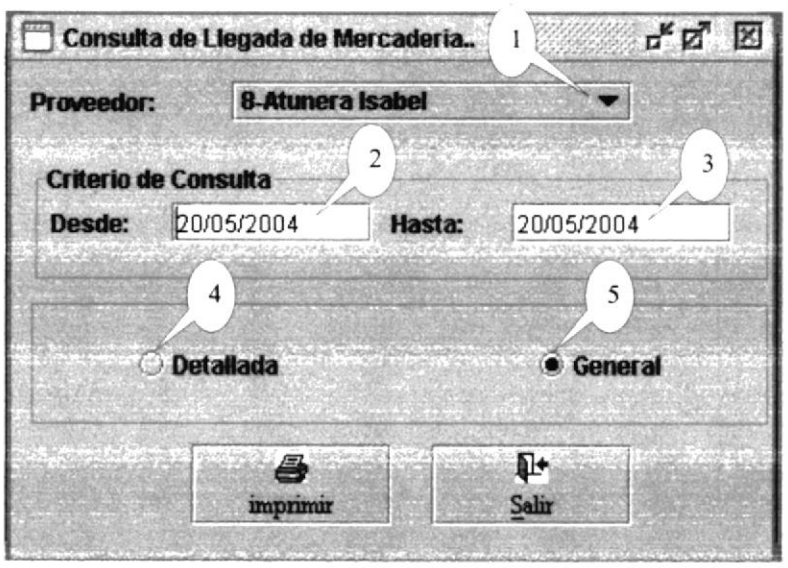
**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | cmbProveedores | ✓ | | C | A | ComboBox marca los proveedores. |
| 2 | txtFechaIni | ✓ | | I | A | TextBox, permite ingresar la fecha inicial. |
| 3 | txtFechaFin | ✓ | | I | A | TextBox, permite ingresar la fecha final. |
| 4 | General | ✓ | | C | A | Botón para elegir las generales. |
| 5 | Detallada | ✓ | | C | A | Botón para elegir las detalladas. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.1.18. REPORTE DE LLEGADA DE MERCADERÍA

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMConLlegMerc.java** | Nombre Lógico:<br>**frmConLlegMerc** | |
| Descripción: *Reporte de Llegada de Mercadería.* | | |
| **DISEÑO DE PANTALLA** | | |




**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Nombre Físico:<br>**COMConLlegMerca.java** | Nombre Lógico:<br>**frmConLlegMerc** | |
| Descripción: *Reporte de Llegada de Mercadería.* | | |

**LISTA DE CONTROLES**

| ID. | NOMBRE DEL CONTROL | OB. | NOMBRE DEL CAMPO | TIPO | FUNCIÓN | |
|---|---|---|---|---|---|---|
| 1 | cmbProveedores | ✓ | | C | A | ComboBox marca los proveedores. |
| 2 | txtFechaIni | ✓ | | I | A | TextBox, permite ingresar la fecha inicial. |
| 3 | txtFechaFin | ✓ | | I | A | TextBox, permite ingresar la fecha final. |
| 4 | Detallada | ✓ | | C | A | Botón para elegir las detalladas. |
| 5 | General | ✓ | | C | A | Botón para elegir las generales. |

**Observaciones:**
En la esquina superior izquierda se encuentra un icono de presentación y en la esquina superior derecha los botones de maximizar, minimizar y cerrar.

## 9.2. DISEÑO DE SALIDAS

**Significado de simbología a utilizarse.**

| *SÍMBOLOS* | | *DESCRIPCIÓN* |
|---|---|---|
| **ID** | | Es el número que identifica cada uno de los datos presentados en el reporte. |
| **NOMBRE** | | Es el nombre que se utiliza para los datos del reporte en tiempo de diseño. |
| **FORMATO** | | Especifica el formato que tiene el dato que se va a presentar. Por ejemplo X(30) y 9(10,2). |
| **TIPO:** | **C** | Es un campo. |
| | **F** | Es una fórmula. |
| **FÓRMULA:** | **E** | Si la fórmula se la realizó desde la aplicación. |
| | **I** | Si la fórmula se la realizó desde JReport. |
| | ***Nota**: En la columna más ancha se describirá la fórmula.* | |

BIBLIOTECA
CAMPUS
PEÑAS

## 9.2.1. REPORTE DE SOLICITUD DE COMPRA GENERAL

| | | |
|---|---|---|
| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**SolCompraxProveedor1General1.cls** | Descripción:<br>**Reporte General de Solicitud de Compra** | |

**DISEÑO DEL REPORTE**

**Encabezado de Página**

(2) Nombre de la Empresa — (1) **TITULO DEL REPORTE** — (3) Logotipo de la Empresa

**Cabecera**

(4) Número de Solicitud: ##### (5) — (6) Fecha de Elaboración: MM/dd/yy (7) — (8) Tipo de Solicitud: XXXXXXXXXX (9)

(10) Estado: XXXXXXXXXX (11) — (12) Observación: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX (13)

**Totales**

**Pie de Página**

(14) **Página** (15) **#####**

**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**SolCompraxProveedor1General1.cls** | Descripción:<br>**Reporte General de Solicitud de Compra** | |

**INFORMACIÓN DEL REPORTE**

**ENCABEZADO DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 1 | Nombre del reporte | C | X(25) | | |
| 2 | Nombre de la Empresa | C | X(25) | | |
| 3 | Logotipo de la empresa | C | | | |

**CABECERA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 4 | Número de Solicitud | C | X(15) | | |
| 5 | CSNMERO | C | 9(5) | | |
| 6 | Fecha de Elaboración | C | X(15) | | |
| 7 | CSFECHA | C | Date | | |
| 8 | Tipo de Solicitud | C | X(15) | | |
| 9 | Tip_Descripcion | C | X(25) | | |
| 10 | Estado | C | X(15) | | |
| 11 | Descripcion | C | X(15) | | |
| 12 | Observación | C | X(15) | | |
| 13 | CSOBSRB | C | X(25) | | |

**PIE DE PAGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 14 | Página | C | X(15) | | |
| 15 | Page Number | C | 9(3) | | |

**Función:**

ESPOL BIBLIOTECA CAMPUS GUAYAQUIL

## 9.2.2. REPORTE DE SOLICITUD DE COMPRA DETALLADO

| **SCI – COM**<br>**Sistema de Compras**<br>**VERSION 1.0** | | Pág. 1 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**SolComxProveedor1.cls** | Descripción:<br>**Reporte Detallado de Solicitud de Compra** | | |

**DISEÑO DEL REPORTE**

| | |
|---|---|
| Encabezado de Página | (2) Nombre de la Empresa; (1) **TITULO DEL REPORTE**; (3) Logotipo de la Empresa |
| Cabecera | (4) Num de Sol: ##### (5); (6) Fecha de Elab: MM/dd/yy (7); (8) Tipo de Sol: XXXXXXXXXX (9); (10) Estado: XXXXXXXX (11)<br>(12) Código; (13) Descripción; (14) Cantidad; (15) U.Medida; (16) Línea |
| Detalle | (17) ##### ; (18) XXXXXXXXXXXXXXX ; (19) ##### ; (20) XXXXXXXXXX ; (21) XXXXXXXXXX |
| Totales | |
| Pie de Página | (22) **Página** (23) **#####** |

**Función:**

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**SolComxProveedor1.cls** | Descripción:<br>**Reporte Detallado de Solicitud de Compra** | |

**INFORMACIÓN DEL REPORTE**

**ENCABEZADO DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 1 | Nombre del reporte | C | X(25) | | |
| 2 | Nombre de la Empresa | C | X(25) | | |
| 3 | Logotipo de la empresa | C | | | |

**CABECERA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 4 | Num Sol | C | X(15) | | |
| 5 | CSNMERO | C | 9(5) | | |
| 6 | Fecha de Elab | C | X(15) | | |
| 7 | CSFECHA | C | Date | | |
| 8 | Tipo de Sol | C | X(15) | | |
| 9 | Tip_Descripcion | C | X(15) | | |
| 10 | Estado | C | X(15) | | |
| 11 | Descripcion | C | X(15) | | |
| 12 | Código | C | X(15) | | |
| 13 | Descripción | C | X(15) | | |
| 14 | Cantidad | C | X(15) | | |
| 15 | U.Medida | C | X(15) | | |
| 16 | Línea | C | X(15) | | |

**DETALLE**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 17 | DSCDPRO | C | 9(5) | | |
| 18 | Pro_Descripcion | C | X(25) | | |
| 19 | DSCTDAD | C | 9(5) | | |
| 20 | Uni_Descripcion | C | X(15) | | |
| 21 | Tip_Descripcion | C | X(15) | | |

**PIE DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 22 | Página | C | X(15) | | |
| 23 | Page Number | C | 9(3) | | |

**Función:**

## 9.2.3. REPORTE DE ORDEN DE COMPRA GENERAL

| **SCI – COM**<br>**Sistema de Compras**<br>**VERSION 1.0** | Pág. 1 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**OdenCompraxProveedor2General2.cls** | Descripción:<br>**Reporte General de Orden de Compra** | |

**DISEÑO DEL REPORTE**

**Encabezado de Página**

2 → Nombre de la Empresa

1 → **TITULO DEL REPORTE**

3 → Logotipo de la Empresa

**Cabecera**

4 → No. Orden: #####

5 → Proveedor: XXXXXXXXXXXX

6 → Fecha Elaboración: MM/dd/yy

7 → Estado: XXXXXXXXXXXXXXX

8 → Tipo de Pago: XXXXXXXXXXXXX

9 → **Página** #####

10 → **Función:**

11 → **Totales**

12 → **Pie de Página**

13 → **Función:**

14 → **Página**

15 → **#####**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**OdenCompraxProveedor2General2.cls** | Descripción:<br>**Reporte General de Orden de Compra** | |

**INFORMACIÓN DEL REPORTE**

**ENCABEZADO DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 1 | Nombre del reporte | C | X(25) | | |
| 2 | Nombre de la Empresa | C | X(25) | | |
| 3 | Logotipo de la empresa | C | | | |

**CABECERA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 4 | Num de Orden | C | X(15) | | |
| 5 | CONMERO | C | 9(5) | | |
| 6 | Proveedor | C | X(15) | | |
| 7 | Pro_razonsocial | C | X(15) | | |
| 8 | Fecha Elaboración | C | X(15) | | |
| 9 | COFECHA | C | Date | | |
| 10 | Estado | C | X(15) | | |
| 11 | Descripcion | C | X(15) | | |
| 12 | Tipo de Pago | C | X(15) | | |
| 13 | Descripcion | C | X(15) | | |

**PIE DE PAGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 14 | Página | C | X(15) | | |
| 15 | Page Number | C | 9(3) | | |

**Función:**

## 9.2.4. REPORTE DE ORDEN DE COMPRA DETALLADO

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**OrdCompraxProveedor2.cls** | Descripción:<br>**Reporte de Orden de Compra Detallado** | |

**DISEÑO DEL REPORTE**

| | |
|---|---|
| **Encabezado de Página** | (2) Nombre de la Empresa; (1) **TITULO DEL REPORTE**; (3) Logotipo de la Empresa |
| **Cabecera** | (4) No. Orden: #####; (5) Proveedor: XXXXXX; (6) Fecha Elab: MM/dd/yy; (7) F.Pago: XXXX; (8) Estado: XXXXXX; (9) Cod Producto; (10) Descripción; (11) Cantidad; (12) Precio; (13) SubTotal; (14) U.Medida; (15) Línea |
| **Det** | (16) #####; (17) XXXXXXXXXXX; (18) #####; (19) #####,00; (20) #####.00; (21) XXXXXX; (22) XXXXXX |
| **Totales** | |
| **Pie de Página** | (23) **Página** (24) **#####** |

**Función:**

ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**OrdCompraxProveedor2.cls** | Descripción:<br>**Reporte de Orden de Compra Detallado** | |

**INFORMACIÓN DEL REPORTE**

**ENCABEZADO DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 1 | Nombre del reporte | C | X(25) | | |
| 2 | Nombre de la Empresa | C | X(25) | | |
| 3 | Logotipo de la empresa | C | | | |

**CABECERA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 4 | Num Orden | C | X(15) | | |
| 5 | CONMERO | C | 9(5) | | |
| 6 | Proveedor | C | X(15) | | |
| 7 | Pro_razonsocial | C | X(15) | | |
| 8 | Fecha Elab | C | X(15) | | |
| 9 | COFECHA | C | Date | | |
| 10 | F.Pago | C | X(15) | | |
| 11 | Descripcion | C | X(15) | | |
| 12 | Estado | C | X(15) | | |
| 13 | Descripcion | C | X(15) | | |
| 14 | Cod Producto | C | X(15) | | |
| 15 | Descripción | C | X(15) | | |
| 16 | Cantidad | C | X(15) | | |
| 17 | Precio | C | X(15) | | |
| 18 | Subtotal | C | X(15) | | |
| 19 | U.Medida | C | X(15) | | |
| 20 | Línea | C | X(15) | | |

**DETALLE**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 21 | DCCPROD | C | 9(5) | | |
| 22 | DCDESCP | C | X(25) | | |
| 23 | DCCTDAD | C | 9(5) | | |
| 24 | DCPRECIO | C | 9(5,2) | | |
| 25 | DCSBTOT | C | 9(5,2) | | |
| 26 | Uni_Descripcion | C | X(15) | | |
| 27 | Tip_Descripcion | C | X(15) | | |

**PIE DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 28 | Página | C | X(15) | | |
| 29 | Page Number | C | 9(3) | | |

**Función:**

## 9.2.5. REPORTE DE LLEGADA DE MERCADERÍA GENERAL

| **SCI – COM**<br>**Sistema de Compras**<br>**VERSION 1.0** | Pág. 1 / 2 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**LlegMercaderiaxProveedor1General2.cls** | Descripción:<br>**Reporte General Llegada Mercadería** | |

**DISEÑO DEL REPORTE**

**Encabezado de Página**

2 → Nombre de la Empresa

1 → **TITULO DEL REPORTE**

3 → Logotipo de la Empresa

**Cabecera**

4 → Proveedor: XXXXXXXXXXXXXXXX

5 →

6 → Fecha Emisión :

7 → MM/dd/yy

8 → Numero Orden :

9 → #####

10 → Tipo de Pago: XXXXXXXXXXXXX

11 →

**Totales**

**Pie de Página**

12 → **Página**

13 → **#####**

**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**LlegMercaderiaxProveedor1General2.cls** | Descripción:<br>**Reporte General Llegada Mercadería** | |

**INFORMACIÓN DEL REPORTE**

**ENCABEZADO DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 1 | Nombre del reporte | C | X(25) | | |
| 2 | Nombre de la Empresa | C | X(25) | | |
| 3 | Logotipo de la empresa | C | | | |

**CABECERA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 4 | Proveedor | C | X(15) | | |
| 5 | Pro_razonsocial | C | X(25) | | |
| 6 | Fecha Emisión | C | X(15) | | |
| 7 | COFECHA | C | Date | | |
| 8 | Numero Orden | C | X(15) | | |
| 9 | CONMERO | C | 9(5) | | |
| 10 | Tipo de Pago | C | X(15) | | |
| 11 | Descripcion | C | X(15) | | |

**PIE DE PAGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 12 | Página | C | X(15) | | |
| 13 | Page Number | C | 9(3) | | |

**Función:**

## 9.2.6. REPORTE DE LLEGADA DE MERCADERÍA DETALLADO

| | | |
|---|---|---|
| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**LlegMercaderiaxProveedor5.cls** | Descripción:<br>**Reporte Detallado de Llegada de Mercadería** | |

**DISEÑO DEL REPORTE**

| | |
|---|---|
| Encabezado de Página | (2) Nombre de la Empresa; (1) **TITULO DEL REPORTE**; (3) Logotipo de la Empresa |
| Cabecera | (4) Proveedor: (5) XXXXXXXXXXXXXX; (6) Fecha de Emisión : (7) MM/dd/yy<br>(8) Cod Producto; (9) Descripción; (10) Cantidad; (11) U.Medida |
| Det | (12) #####; (13) XXXXXXXXXXXXXXXXXXXX; (14) #####; (15) XXXXXXXX |
| Totales | |
| Pie de Página | (16) **Página** (17) **#####** |

**Función:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Nombre del Reporte:<br>**LlegMercaderiaxProveedor5.cls** | Descripción:<br>**Reporte Detallado de Llegada de Mercadería** | |

**INFORMACIÓN DEL REPORTE**

**ENCABEZADO DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 1 | Nombre del reporte | C | X(25) | | |
| 2 | Nombre de la Empresa | C | X(25) | | |
| 3 | Logotipo de la empresa | C | | | |

**CABECERA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 4 | Proveedor | C | X(15) | | |
| 5 | Pro_razonsocial | C | X(25) | | |
| 6 | Fecha de Emisión | C | X(15) | | |
| 7 | COFECHA | C | Date | | |
| 8 | Cod Producto | C | X(15) | | |
| 9 | Descripción | C | X(15) | | |
| 10 | Cantidad | C | X(15) | | |
| 11 | U.Medida | C | X(15) | | |

**DETALLE**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 12 | DCCPROD | C | 9(5) | | |
| 13 | DCDESCP | C | X(25) | | |
| 14 | DCCTDAD | C | 9(5) | | |
| 15 | Uni_Descripcion | C | X(15) | | |

**PIE DE PÁGINA**

| I.D. | NOMBRE | TIPO | FORMATO | FÓRMULA | |
|---|---|---|---|---|---|
| 16 | Página | C | X(15) | | |
| 17 | Page Number | C | 9(3) | | |

**Función:**

# Capítulo 10

## Diagramas de Estado

# 10. DIAGRAMAS DE CASOS DE USO

## 10. CASOS DE USO DEL SISTEMA DE COMPRAS

| | | |
|---|---|---|
| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 1 | |
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| **LISTADO DE CASOS DE USO** | | |

1. Estado de Tipo Solicitud de Compra
2. Estado de Orden de Compra

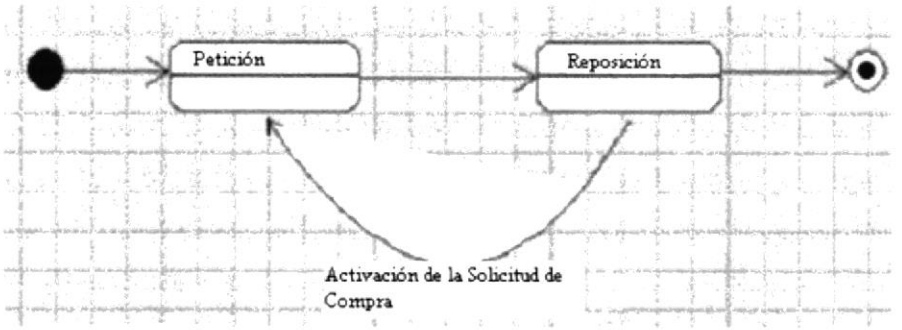
Observación:

# 10.1. DIAGRAMAS DE ESTADO

## 10.1.1. ESTADO DE TIPO DE MOVIMIENTO

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | 1/1 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE ESTADO**




**Narrativa:**

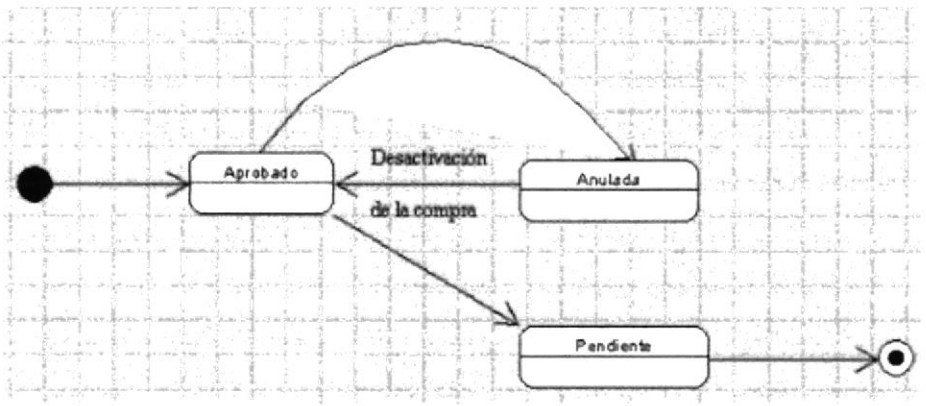
1. El Estado de Tipo de Solicitud de Compra es por petición dependiendo de la Solicitud de Compra que se realicen en el sistema.

2. El Estado de Tipo de Solicitud de Compra es por reposición sino se realiza ninguna petición.

## 10.1.2. ESTADO DE ORDEN DE COMPRA

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0.0 | Pág. 1 / 1 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DISEÑO DEL DIAGRAMA DE ESTADO**




**Narrativa:**

1. La Orden de Compra pasa a estado aprobada cuando una de las compras es aprobada.
2. La Orden de Compra pasa a estado anulado cuando una de las compras es anulada.
3. La Orden de Compra pasa a estado pendiente cuando la compra no a sido procesada.

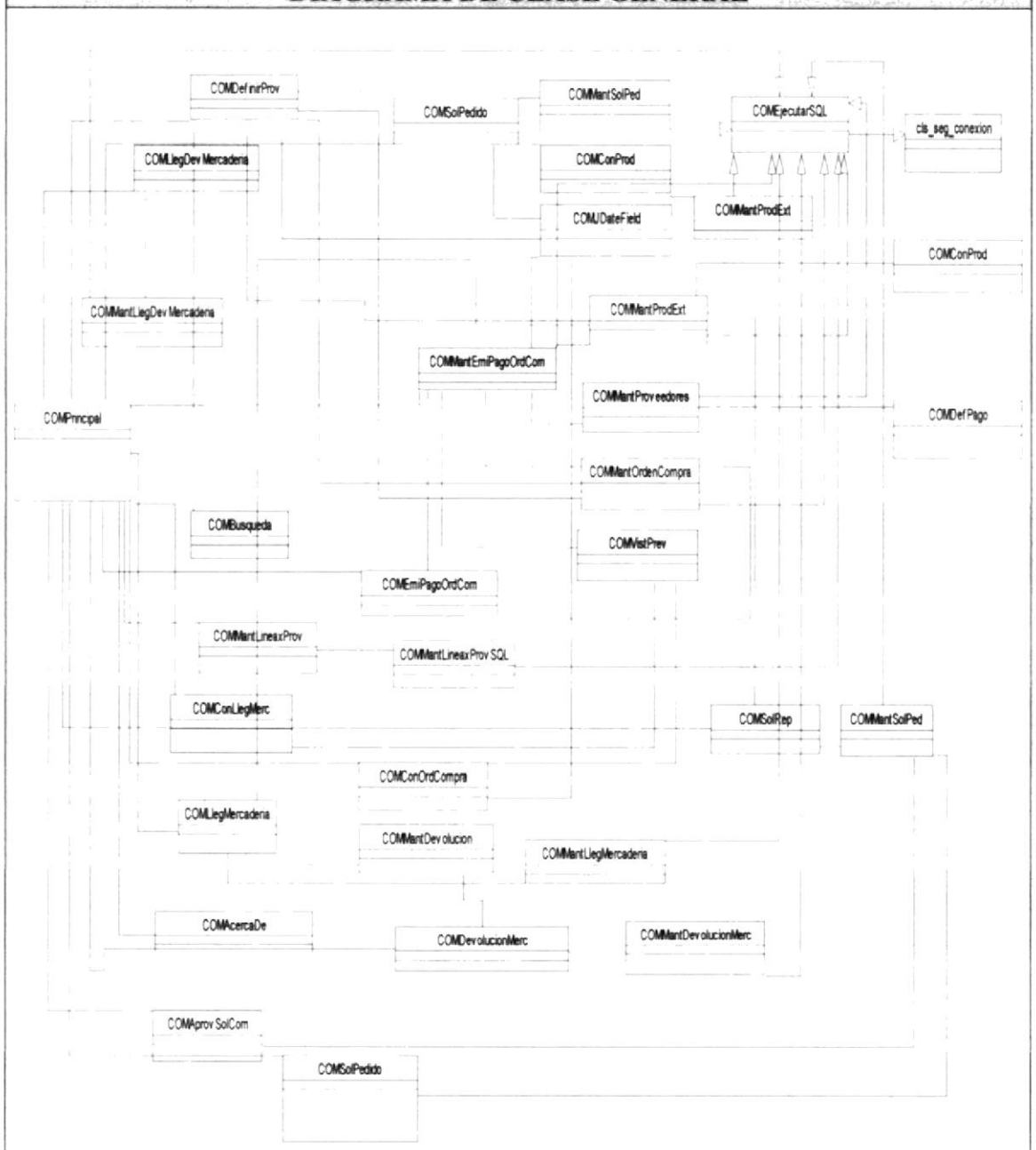
# Capítulo 1

## Diagrama de Clases

# 11. DIAGRAMA DE CLASES

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | Pág. 1 / 1 | |
|---|---|---|
| | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |

**DIAGRAMA DE CLASE GENERAL**




**Observación:**

# Capítulo 1

## Capa Usuario

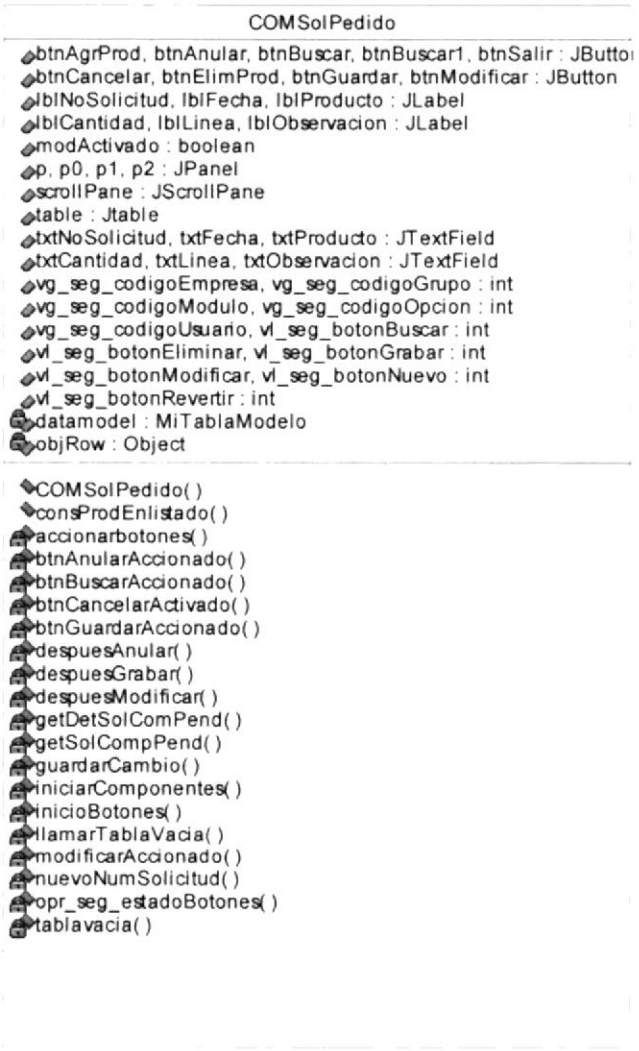
ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

# 11. DIAGRAMA DE CLASE DETALLADO

## 11.1. CLASE COMSolPedido

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMSolPedido** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |

**DIAGRAMA DE CLASE**




**Narrativa:**
La Clase COMSolPedido es un formulario del Menú Operaciones, en dónde se muestra las Solicitudes de Compra Bajo Pedido.

## 11.2. CLASE COMConProd

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMConProd** | Alcance:<br>**Public** | Hereda de:<br>**JDialog** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La Clase COMConProd es un formulario del Menú Reportes/Consultas.

## 11.3. CLASE COMSolRep

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMSolRep** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |

**DIAGRAMA DE CLASE**

**Narrativa:**
La Clase COMSolRep es un formulario del Menú Operaciones.

## 11.4. CLASE COMBusqueda

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMBusqueda** | Alcance:<br>**Public** | Hereda de:<br>**JDialog** | |

**DIAGRAMA DE CLASE**

**Narrativa:**
La Clase COMBusqueda es un formulario del Menú Consultas / Reportes.

## 11.5. CLASE COMAprovSolCom

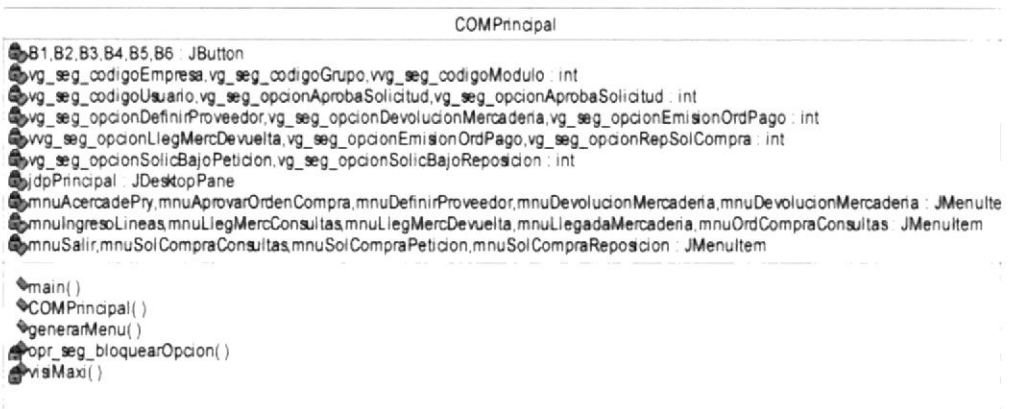
| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMAprovSolCom** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |

**DIAGRAMA DE CLASE**

**Narrativa:**
La Clase COMAprovSolCom es un formulario del Menú Operaciones.

## 11.6. CLASE COMPrincipal

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMPrincipal** | Alcance:<br>**Public** | Hereda de:<br>**JFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

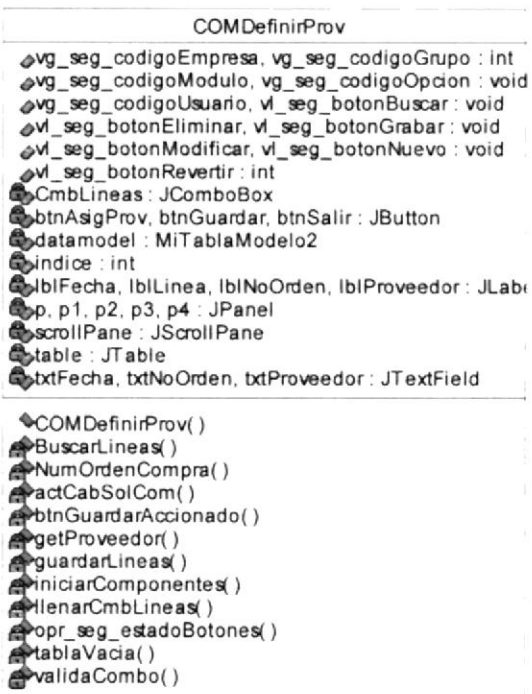



ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

**Narrativa:**
La Clase COMPrincipal es un formulario que se encarga de llamar a todos los menús.

## 11.7. CLASE COMDefinirProv

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMDefinirProv** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMDefinirProv es un formulario del Menú Operaciones.

## 11.8. CLASE COMLlegMercaderia

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMLlegMercaderia** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La Clase COMLlegMercaderia es un formulario del Menú Operaciones.

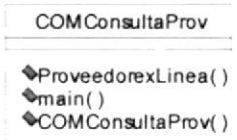
# 11.9. CLASE COMEmiPagoOrdCom

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMEmiPagoOrdCom** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La Clase COMEmiPagoOrdCom es un formulario del Menú Operaciones.

## 11.10. CLASE COMConsultaProv

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMConsultaProv** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMConsultaProv es un formulario me permite realizar la consultad el proveedor.

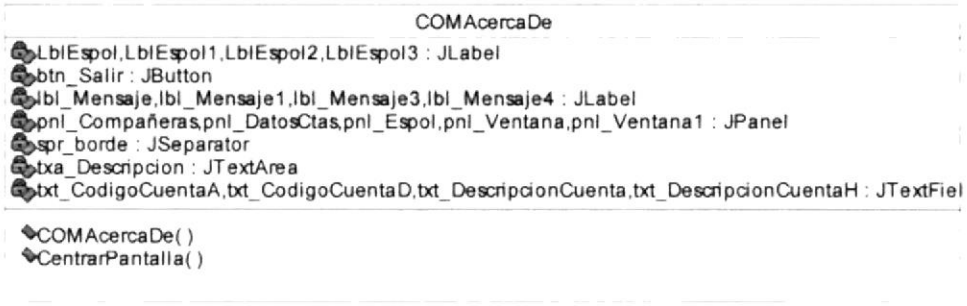
## 11.11. CLASE COMDateFieldDocument

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMDateFieldDocument** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La Clase COMDateFieldDocument es un formulario me permite actualizar las Aprobaciones de las Solicitudes de Compras.

## 11.12. CLASE COMAcercaDe

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMAcercaDe** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La clase COMAcercaDe genera el formulario de información del sistema, que va a contener el nombre de los Integrantes y Derechos del Autor.

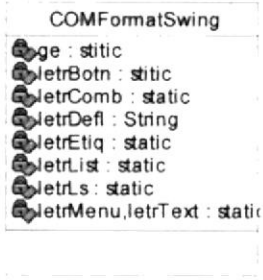
## 11.13. CLASE COMDefPago

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMDefPago** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La Clase COMDefPago *a la base de datos mediante las sentencias SQL.*

## 11.14. CLASE COMFormatSwing

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMFormatSwing** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
*La clase* COMFormatSwing *permite la creaciòn de los principales componentes gráficos (Swing).*

## 11.15. CLASE COMJDateField

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMJDateField** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
*La clase* COMJDateField es una clase propia de Java que ayuda a generar la fecha.

## 11.16. CLASE COMJTextFieldFilter

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMJTextFieldFilter** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La clase COMJTextFieldFilter es una clase propia de Java que ayuda a generar el formato numérico.

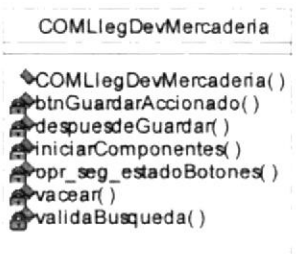
## 11.17. CLASE COMMantLineaxProv

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMMantLineaxProv** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La clase COMMantLineaxProvSQL consulta el tiempo de crédito por proveedor, característica de la forma de pago, consulta de proveedor generalizado, consulta la línea y el proveedor, guarda los detalles del mantenimiento de las líneas por proveedor, elimina una línea

## 11.18. CLASE COMLlegDevMercadería

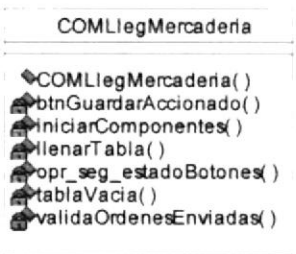
| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMLlegDevMercadería** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMLlegDevMercadería es un formulario me permite actualizar las devolución de las mercadería.

## 11.19. CLASE COMLlegadaMercaderia

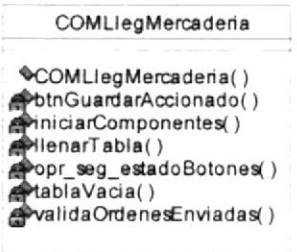
| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMLlegadaMercaderia** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMLlegadaMercaderia es un formulario me permite actualizar l allegada del a mercaderia.

## 11.20. CLASE COMLlegMercaderia

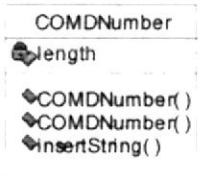
| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMLlegMercaderia** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMAprovSolCom es un formulario me permite actualizar las Aprobaciones de las Solicitudes de Compras.

## 11.21. CLASE COMDNumber

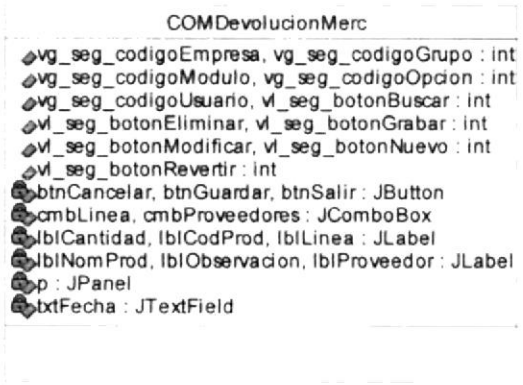
| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMDNumber** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
*La clase* COMDNumber realiza la validación de números.

## 11.22. CLASE COMDevolucionMerc

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMDevolucionMerc** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

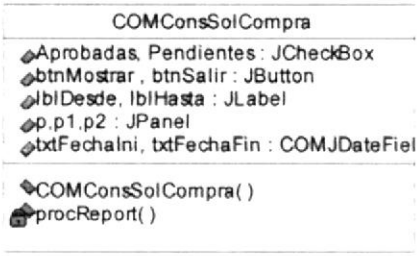



**Narrativa:**
*La clase* COMDevolucionMerc realiza la devolución de mercadería.

## 11.23. CLASE COMConsSolCompra

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMConsSolCompra** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |

**DIAGRAMA DE CLASE**

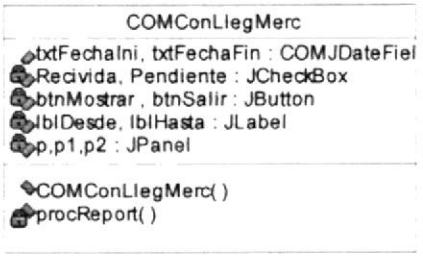



**Narrativa:**
*La clase* COMConsSolCompra genera la pantalla de la consulta de la Solicitud de Compra .

## 11.24. CLASE COMConLlegMerc

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre:<br>**COMConLlegMerc** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |

**DIAGRAMA DE CLASE**



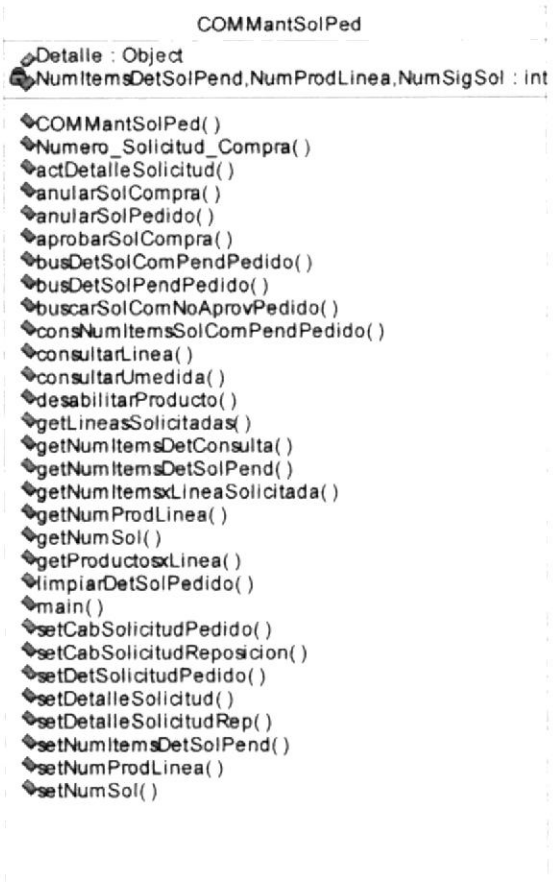

**Narrativa:**
*La clase COMConLlegMerc* realiza la consulta de la Llegada de Mercadería.

# Capítulo 15

## Capa Negocios

## 11.25. CLASE COMMantSolPed

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantSolPed** | Alcance:<br>**Public** | Hereda de:<br>**COMEjecutarSQL** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMMantSolPed es un formulario que permite dar mantenimiento a las Solicitudes de Compras Bajo Pedido.

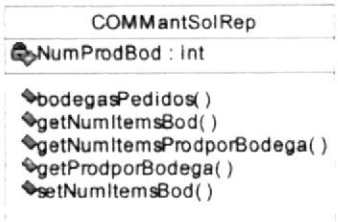
## 11.26. CLASE COMMantProdExt

| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantProdExt** | Alcance:<br>**Public** | Hereda de:<br>**COMEjecutarSQL** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La Clase COMMantProdExt es una clase que permite dar mantenimiento a los Productos.

## 11.27. CLASE COMMantSolRep

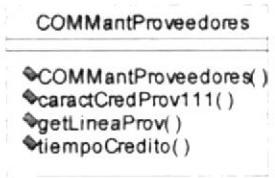
| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantSolRep** | Alcance:<br>**Public** | Hereda de:<br>**COMEjecutarSQL** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMMantSolRep es un formulario que me permite dar mantenimiento a las Solicitudes de Compras por Reposición.

## 11.28. CLASE COMMantProveedores

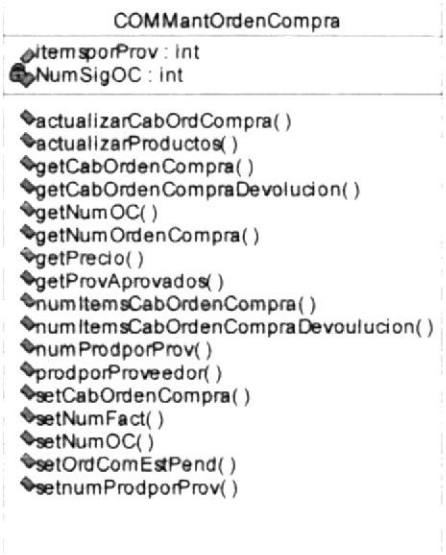
| SCI - COM<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantProveedores** | Alcance:<br>**Public** | Hereda de:<br>**COMEjecutarSQL** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMMantProveedores es un formulario que me permite dar mantenimiento a los Proveedores.

## 11.29. CLASE COMMantOrdenCompra

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantOrdenCompra** | Alcance:<br>**Public** | Hereda de:<br>**COMEjecutarSQL** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMMantOrdenCompra es un formulario que me permite dar mantenimiento a las Ordenes de Compras.

## 11.30. CLASE COMMantDevolucionMerc

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantDevolucionMerc** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
*La clase* COMMantDevolucionMerc *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## 11.31. CLASE COMMantLineaxProvSQL

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantLineaxProvSQL** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

**Narrativa:**
La clase COMMantLineaxProvSQL consulta el tiempo de crédito por proveedor, característica de la forma de pago, consulta de proveedor generalizado, consulta la línea y el proveedor, guarda los detalles del mantenimiento de las líneas por proveedor, elimina una línea

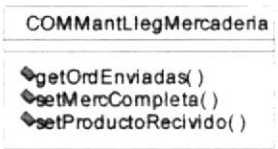
## 11.32. CLASE COMMantLlegDevMercadería

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 1 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantLlegDevMercadería** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| **DIAGRAMA DE CLASE** | | | |

**Narrativa:**
La Clase COMMantLlegDevMercadería es un formulario me permite controlar la devolución de mercadería.

## 11.33. CLASE COMMantLlegMercadería

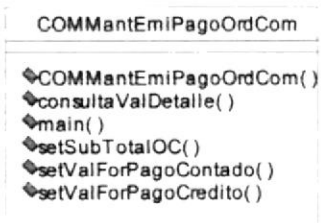
| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantLlegMercadería** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMMantLlegMercadería es un formulario me permite controlar l allegada de mercadería.

## 11.34. CLASE COMMantEmilPagoOrdCom

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantEmilPagoOrdCom** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La clase COMMantEmiPagoOrdCom guarda el número y la forma de pago en la orden de compra, guarda el valor que se va a pagar, consulta el precio, la línea y la unidad de medida, guarda la unidad de medida y la línea en la orden de compra

## 11.35. CLASE COMMantDevolucion

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre:<br>**COMMantDevolucion** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |

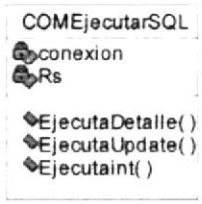
**Narrativa:**
La clase COMMantDevolucion permite obtener las devoluciones pendientes, el nombre de un proveedor, guarda las cabeceras y los detalles de la devolución y obtiene el siguiente número de devolución de los datos de la capa de usuario de COMDevolucionMerc

# Capítulo 1

## Capa de Datos

## 11.36. CLASE COMEjecutaSQL

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ultima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacis**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Datos** | Nombre:<br>**COMEjecutaSQL** | Alcance:<br>**Public** | Hereda de:<br>**JInternalFrame** | |
| **DIAGRAMA DE CLASE** | | | | |




**Narrativa:**
La Clase COMEjecutaSQL es la clase conexión.

# Capítulo 12

## Definición de Clases

# 12. DEFINICIÓN DE CLASES

**Significado de simbología a utilizarse.**

A continuación se muestra una tabla con los símbolos a utilizarse en la definición de las clases:

| SÍMBOLOS | DESCRIPCIÓN |
|---|---|
| **Capa** | Determina si pertenece a la capa Usuario, Intermedia o de Negocio y Dato (Modelo Tres Capas) |
| **Nombre de Clase** | Es el nombre del archivo .java . |
| **Tipo** | Determina el tipo de la clase (public, private, protected, final, static, etc). |
| **Hereda de** | Nombre de la clase de quién hereda. |
| **DETALLE** | |
| **Tipo** | Es el tipo de dato en el atributo o método según sea el caso. Cuando un método no devuelve un dato debe especificarse: *void.* |
| **PB** | Indica si el atributo o método es público. |
| **PV** | Indica si el atributo o método es privado. |
| **PG** | Indica si el atributo o método es protegido. |

# Capítulo 12

## Capa de Usuario

## 12.1. CLASE COMSolPedido

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 4 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolPedido** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMSolPedido genera el formulario de Solicitud de Compra Bajo Pedido.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | p0 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 2 | p | JPanel() | Panel agrupador de controles. | ✓ | | |
| 3 | p1 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 4 | p2 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 5 | lblNoSolicitud | JLable() | Etiqueta de la Solicitud de Pedido. | ✓ | | |
| 6 | lblFecha | JLabel() | Etiqueta de Fecha. | ✓ | | |
| 7 | lblProducto | JLable() | Etiqueta del Producto. | ✓ | | |
| 8 | lblCantidad | JLabel() | Etiqueta de Cantidad. | ✓ | | |
| 9 | lblLinea | JLabel() | Etiqueta de Linea. | ✓ | | |
| 10 | lblObservacion | JLabel() | Etiqueta de Observación. | ✓ | | |
| 11 | btnGuardar | JButton() | Botón de Guardar. | ✓ | | |
| 12 | btnAnular | JButton() | Botón de Anular. | ✓ | | |
| 13 | btnSalir | JButton() | Botón de Salir. | ✓ | | |
| 14 | btnBuscar | JButton() | Botón de Buscar. | ✓ | | |
| 15 | btnBuscar1 | JButton() | Botón de Buscar1. | ✓ | | |
| 16 | btnAgrProd | JButton() | Botón de AgrProd. | ✓ | | |
| 17 | btnElimProd | JButton() | Botón de ElimProd. | ✓ | | |
| 18 | btnCancelar | JButton() | Botón de Cancelar. | ✓ | | |
| 19 | btnModificar | JButton() | Botón de Modificar. | ✓ | | |
| 20 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | ✓ | | |
| 21 | txtCantidad | JTextField() | Caja de Texto Cantidad. | ✓ | | |

**Observación**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 4 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolPedido** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMSolPedido genera el formulario de Solicitud de Compra Bajo Pedido.* | | | |

## DESCRIPCIÓN DE CLASE

### ATRIBUTOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 22 | txtLinea | JTextField() | Caja de Texto Linea. | ✓ | | |
| 23 | txtNoSolicitud | JTextField() | Caja de Texto No. de Solicitud. | ✓ | | |
| 24 | txtObservacion | JTextField() | Caja de Texto Observación. | ✓ | | |
| 25 | txtProducto | JTextField() | Caja de Texto Producto. | ✓ | | |
| 26 | objRow | Object | Arreglo de table. | | ✓ | |
| 27 | table | JTable() | Variable de la clase Tabla. | ✓ | | |
| 28 | datamodel | MiTablaModelo | Variable de clase MiTablaModelo | | ✓ | |
| 29 | scrollPane | JScrollPane() | Barra de Desplazamiento. | ✓ | | |
| 30 | modActivado | Boolean | Variable de tipo lógico. | ✓ | | |
| 31 | Modificando | Boolean | Variable de tipo lógico. | ✓ | | |
| 32 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 33 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 34 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 35 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 36 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 37 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 38 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 39 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 40 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 41 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 42 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 /4 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolPedido** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMSolPedido genera el formulario de Solicitud de Compra Bajo Pedido.* | | | |

## DESCRIPCIÓN DE CLASE

### METODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMSolPedido(int Usuario,int Grupo,int Modulo,int Empresa, int Opcion) | | Construye el formulario de Solicitud Bajo Pedido. | ✓ | | |
| 2 | opr_seg_estadoBotones(boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Activa el estado de los botones y verifica si tiene acceso al botón el grupo usuario. | | ✓ | |
| 3 | btnBuscarAccionado() | void | Realiza la búsqueda de la Solicitud Bajo Pedido pendiente. | | ✓ | |
| 4 | tablavacia() | void | Permite instanciar a todas las Tablas existentes de la Solicitud Bajo Pedido. | | ✓ | |
| 5 | consProdEnlistado() | Integer | Permite realizar la consulta de productos | | ✓ | |
| 6 | modificarAccionado() | void | Permite modificar la Solicitud. | | ✓ | |
| 7 | accionarbotones(boolean bus,boolean bus1, boolean ap, boolean ep, boolean guar, boolean mod, boolean an, boolean canc) | void | Permite habilitar todos los botones. | | ✓ | |
| 8 | btnAnularAccionado(int numSol) | void | Permite anular lo accionado. | | ✓ | |
| 9 | getDetSolComPend() | void | Se obtiene el detalle de la Solicitud. | | ✓ | |
| 10 | btnCancelarActivado() | void | Limpia la pantalla de la Solicitud. | | ✓ | |
| 11 | getSolCompPend() | void | Se obtiene el código de la Solicitud Bajo Pedido pendiente. | | ✓ | |
| 12 | iniciarComponentes() | void | Permite activar todos los botones de la Solicitud Bajo Pedido. | | ✓ | |
| 13 | llamarTablaVacia() | void | Permite instanciar a todas las Tablas existentes de la Solicitud Bajo Pedido. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 4 /4 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolPedido** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMSolPedido genera el formulario de Solicitud de Compra Bajo Pedido.* | | | |

## DESCRIPCIÓN DE CLASE

**METODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 14 | btnGuardarAccionado() | void | Permite guardar lo accionado. | | ✓ | |
| 15 | despuesAnular() | void | Método usado al anular la Solicitud. | | ✓ | |
| 16 | despuesGrabar() | void | Método usado al guardar la Solicitud. | | ✓ | |
| 17 | despuesModificar() | void | Método usado al modificar la Solicitud. | | ✓ | |
| 18 | guardarCambio() | void | Método usado al guardar lo modificado. | | ✓ | |
| 19 | inicioBotones() | void | Método que se usa al accionar los botones. | | ✓ | |
| 20 | nuevoNumSolicitud() | void | Realiza el conteo de la nueva Solicitud. | | ✓ | |

**Observación:**

## 12.2. CLASE COMConProd

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConProd** | Tipo:<br>**Public** | Hereda de:<br>**JDialog** |
| Descripción: *La clase COMConProd permite realizar la consulta de los Productos .* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | lblProducto | JLabel() | Etiqueta de Producto. | ✓ | | |
| 2 | scrollPane | JScrollPane() | Barra de Desplazamiento. | ✓ | | |
| 3 | table | JTable() | Variable de la clase Tabla. | ✓ | | |
| 4 | txtProducto | JTextField() | Caja de texto de Producto. | ✓ | | |
| 5 | datamodel | MiTablaModelo1 | Variable de la clase MiTablaModelo1. | | ✓ | |
| 6 | iCodigo | Integer | Variable de tipo integer. | | ✓ | |
| 7 | iLinea | Integer | Variable de tipo integer. | | ✓ | |
| 8 | iMedida | Integer | Variable de tipo integer. | | ✓ | |
| 9 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 10 | p1 | JPanel () | Panel agrupador de controles. | | ✓ | |
| 11 | strDescripcion | String | Variable de tipo string. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 2/ 2 | |
|---|---|---|---|---|
| | | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConProd** | Tipo:<br>**Public** | Hereda de:<br>**JDialog** | |
| Descripción: *La clase COMConProd permite realizar la consulta de los Productos .* | | | | |

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMConProd(JFrame father) | | Construye la pantalla de la Consulta de Productos. | ✓ | | |
| 2 | getCodigo() | Integer | Trae el código del Producto. | ✓ | | |
| 3 | getDescripcion () | String | Trae la descripción del Producto. | ✓ | | |
| 4 | getLinea () | Integer | Trae la línea a la cual pertenece el Producto. | ✓ | | |
| 5 | getMedida () | Integer | Trae la medida del Producto. | ✓ | | |
| 6 | setCodigo(int cod) | void | Lleva el código del Producto. | ✓ | | |
| 7 | setDescripcion(String des) | void | Lleva el descripción del Producto. | ✓ | | |
| 8 | setLinea(int lin) | void | Lleva el línea a la cual pertenece el Producto. | ✓ | | |
| 9 | setMedida(int med) | void | Lleva la medida del Producto. | ✓ | | |

**Observación:**

## 12.3. CLASE COMSolRep

| SCI - COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolRep** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |

*Descripción: La clase COMSolRep genera el formulario de Solicitud de Compra Por Reposición.*

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | CmbBodega | JCombobox() | Combo de Bodegas. | | ✓ | |
| 13 | btnGuardar | JButton() | Botón Guardar. | | ✓ | |
| 14 | btnSalir | JButton() | Botón Salir. | | ✓ | |
| 15 | codEmp | Integer | Variable de tipo entero. | | ✓ | |
| 16 | datamodel | MiTablaModelo5 | Variable de la clase MiTablaModelo1. | | ✓ | |
| 17 | iBand | Integer | Variable de tipo entero. | | ✓ | |
| 18 | lblBodega | JLabel() | Etiqueta de la Solicitud Por Reposición | | ✓ | |
| 19 | lblFecha | JLabel() | Etiqueta de la Fecha | | ✓ | |

**Observación:**

| **SCI - COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolRep** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |

*Descripción: La clase COMSolRep genera el formulario de Solicitud de Compra Por Reposición.*

## DESCRIPCIÓN DE CLASE

### ATRIBUTOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | lblNo.Solicitud | JLabel() | Etiqueta de la Bodega | | ✓ | |
| 21 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 22 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 23 | scrollPane | JScrollPane() | Barra de Desplazamiento. | | ✓ | |
| 24 | table | JTable() | Variable de la clase Tabla | | ✓ | |
| 25 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 26 | txtNoSolicitud | JTextField() | Caja de Texto de la Solicitud | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 3 / 3 | |
|---|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 | |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMSolRep** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** | |

Descripción: *La clase COMSolRep genera el formulario de Solicitud de Compra Por Reposición.*

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMSolRep(int Usuario,int Grupo,int Modulo,int Empresa, int Opcion) | | Construye la pantalla de la Solicitud por Reposición. | ✓ | | |
| 2 | actualizarNumSolicitud() | void | Actualiza la Solicitud Por Reposición | | ✓ | |
| 3 | btnGuardarAccionado() | void | Permite guardar lo accionado. | | ✓ | |
| 4 | desabilitarProductos() | void | Deshabilita el botón del os Productos. | | ✓ | |
| 5 | guardarCabecera() | void | Permite guardar la cabecera de la Solicitud Por Reposición. | | ✓ | |
| 6 | rellenarCombo() | void | Rellena el combo de la bodega. | | ✓ | |
| 7 | guardarLineas() | void | Guarda la línea a la cual pertenece el Producto. | | ✓ | |
| 8 | opr_seg_estadoBotones(boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Permite activar el estado de los botones de la Solicitud y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 9 | inicioComponetes() | void | Permite activar todos los botones de la Solicitud Por Reposición. | | ✓ | |
| 10 | tablavacia() | void | Permite instanciar a todas las Tablas existentes de Solicitud Por Reposición. | | ✓ | |

**Observación:**

## 12.4. CLASE COMBusqueda

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMBusqueda** | Tipo:<br>**Public** | Hereda de:<br>**JDialog** |
| *Descripción: La clase COMBusqueda realiza la búsqueda en cada formulario.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | DEBUG | Boolean | Variable de tipo boolean. | | ✓ | |
| 2 | cantPk | Integer | Variable de tipo integer. | | ✓ | |
| 3 | dataFils | Boolean | Arreglo de tipo boolean | | ✓ | |
| 4 | erro | String | Variable de tipo string. | | ✓ | |
| 5 | nombColm | String | Variable de tipo string. | | ✓ | |
| 6 | nombPk1 | String | Variable de tipo string. | | ✓ | |
| 7 | nombPk2 | String | Variable de tipo string. | | ✓ | |
| 8 | nombPk3 | String | Variable de tipo string. | | ✓ | |
| 9 | pk1 | String | Variable de tipo string. | | ✓ | |
| 10 | pk2 | String | Variable de tipo string. | | ✓ | |
| 11 | pk3 | String | Variable de tipo string. | | ✓ | |
| 12 | resl | Boolean | Variable de tipo boolean. | | ✓ | |
| 13 | tabla | JTable() | Variable de la clase Tabla. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMBusqueda** | Tipo:<br>**Public** | Hereda de:<br>**JDialog** |

*Descripción: La clase COMBusqueda realiza la búsqueda en cada formulario.*

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMBusqueda(Frame fram, Dimension size, Dimension rootSize, String sentSQL, int cantPk, String titu) | | Constructor lógico del Table. | ✓ | | |
| 2 | getCantPk () | Integer | Trae valor de CantPk. | ✓ | | |
| 3 | getErro () | String | Trae valor de Erro. | ✓ | | |
| 4 | getNombPk1 () | String | Trae valor de NombPk1. | ✓ | | |
| 5 | getNombPk2 () | String | Trae valor de NombPk2. | ✓ | | |
| 6 | getNombPk3 () | String | Trae valor de NombPk3. | ✓ | | |
| 7 | getPk1() | String | Trae valor de Pk1. | ✓ | | |
| 8 | getPk2() | String | Trae valor de Pk2. | ✓ | | |
| 9 | getPk3() | String | Trae valor de Pk3. | ✓ | | |
| 10 | getResl () | Boolean | Trae el valor de Resl. | ✓ | | |
| 11 | asigPk () | Void | Asigna valor de Pk!, Pk2, Pk3. | | | |
| 12 | ejecConsSelect(String sentSQL, int cantPk) | Boolean | Valida sentencia SQL. | | ✓ | |
| 13 | setCantPk (int cantPk) | Void | Lleva valor de canPk. | | ✓ | |
| 14 | setErro (String erro) | Void | Lleva valor de Erro. | | ✓ | |
| 15 | setNombPk1 (String nombPk1) | Void | Lleva valor de NombPk1. | | ✓ | |
| 16 | setNombPk2 (String nombPk2) | Void | Lleva valor de NombPk2. | | ✓ | |
| 17 | setNombPk3 (String nombPk3) | Void | Lleva valor de NombPk3. | | ✓ | |
| 18 | setPk1 (String pk1) | Void | Lleva valor de Pk1. | | ✓ | |
| 19 | setPk2 (String pk2) | Void | Lleva valor de canPk2. | | | |
| 20 | setNombPk3(String nombPk3) | Void | Lleva valor de canPk3. | | ✓ | |
| 21 | setResl (boolean resl) | Void | Lleva valor de Resl. | | | |

**Observación:**

## 12.5. CLASE COMAprovSolCom

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMAprovSolCom** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMAprovSolCom genera el formulario de Aprobación de la Solicitud de Compra.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | SC | ResultSet | Variable de tipo resulset. | ✓ | | |
| 13 | Cad | String | Variable de tipo String. | | ✓ | |
| 14 | CmbSolicitudes | JComboBox() | Combo de Solicitudes | | ✓ | |
| 15 | btnGuardar | JButton() | Botón Guardar. | | ✓ | |
| 16 | btnSalir | JButton() | Botón Salir. | | ✓ | |
| 17 | btnAnular | JButton() | Botón Anular. | | ✓ | |
| 18 | lblFecha | JLabel() | Etiqueta de la Fecha. | | ✓ | |
| 19 | lblNo.Solicitud | JLabel() | Etiqueta de la Fecha. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMAprovSolCom** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMAprovSolCom genera el formulario de Aprobación de la Solicitud de Compra.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 21 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 22 | p2 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 23 | scrollPane | JScrollPane() | Barra de Desplazamiento. | | ✓ | |
| 24 | table | JTable() | Variable de la clase Tabla | | ✓ | |
| 25 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 26 | txtNoSolicitud | JTextField() | Caja de Texto de la Solicitud | | ✓ | |
| 27 | datamodel | MiTablaModelo4 | Mi tabla model. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMAprovSolCom** | Tipo:<br>**Public** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase COMAprovSolCom genera el formulario de Aprobación de la Solicitud de Compra.*

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMAprovSolCom(int Usuario,int Grupo,int Modulo,int Empresa, int Opcion) | | Construye la pantalla de la Solicitud por Compra. | ✓ | | |
| 2 | LlenarCmbSolicitudes ( ) | void | Rellena el combo de las Solicitudes. | | ✓ | |
| 3 | anularSolicitud ( ) | void | Anula la Solicitud. | | ✓ | |
| 4 | internalFrameActivated(Internal FrameEvent e) | void | Permite llamar al Frame Interno. | ✓ | | |
| 5 | btnGuardarAccionado ( ) | void | Guarda la Solicitud de Compra. | | ✓ | |
| 6 | define(String Car) | String | Define cual Solicitud se va a elegir. | | ✓ | |
| 7 | iniciarComponentes ( ) | void | Permite activar todos los botones de la Solicitud Por Reposición. | | ✓ | |
| 8 | opr_seg_estadoBotones () | void | Permite activar el estado de los botones de la Solicitud y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 9 | llenarLineas() | void | Permite Llenar todas las líneas. | | ✓ | |
| 10 | tablavacia() | void | Permite instanciar a todas las Tablas existentes de Solicitud. | | ✓ | |

**Observación:**

## 12.6. CLASE COMConLlegMerc

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConLlegMerc** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMConLlegMerc* realiza la consulta de la Llegada de Mercadería. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | txtFechaFin | JTextField() | Caja de Texto Fecha Final. | | ✓ | |
| 2 | txtFechaIni | JTextField() | Caja de Texto Fecha Inicial. | | ✓ | |
| 3 | Pendiente | JCheckBox | Combo Llegada de Producto Pendiente. | | ✓ | |
| 4 | Recivida | JCheckBox | Combo Llegada de Producto Recibido. | | ✓ | |
| 5 | btnMostrar | JButton() | Botón Mostrar. | | ✓ | |
| 6 | btnSalir | JButton() | Botón Cerrar. | | ✓ | |
| 7 | lblDesde | JLable() | Etiqueta de Desde. | | ✓ | |
| 8 | lblHasta | JLable() | Etiqueta de Hasta. | | ✓ | |
| 9 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 10 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 11 | p2 | JPanel() | Panel agrupador de controles. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConLlegMerc** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase COMConLlegMerc* realiza la consulta de la Llegada de Mercadería.

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMConLlegMerc() | | Construye la pantalla de la Consulta de Llegada de Mercadería. | ✓ | | |
| 2 | procReport() | void | Realiza el reporte de la Consulta de Llegada de Mercadería. | | ✓ | |

**Observación:**

## 12.7. CLASE COMConOrdCompra

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConOrdCompra** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase COMConOrdCompra* permite generar una consulta de la Orden de Compra. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | Aprobadas | JCheckBox() | Combo de Orden de Compra Aprobada. | ✓ | | |
| 2 | Pendiente | JCheckBox() | Combo de Orden de Compra Pendiente. | ✓ | | |
| 3 | btnMostrar | JButton() | Botón Mostrar. | ✓ | | |
| 4 | btnSalir | JButton() | Botón Salir. | ✓ | | |
| 5 | lblDesde | JLable() | Etiqueta de Desde. | ✓ | | |
| 6 | lblHasta | JLable() | Etiqueta de Hasta. | ✓ | | |
| 7 | p | JPanel() | Panel agrupador de controles. | ✓ | | |
| 8 | p1 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 9 | p2 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 10 | txtFechaFin | JTextField() | Caja de Texto FechaFin. | ✓ | | |
| 11 | txtFechaIni | JTextField() | Caja de Texto FechaIni. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConOrdCompra** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMConOrdCompra permite generar una consulta de la Orden de Compra. | | | |

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMConOrdCompra() | | Construye la pantalla de la Consulta de la Orden de Compra. | ✓ | | |
| 2 | AprobadasMouseClicked(java.awt.event.MouseEvent evt) | void | Captura las Ordenes de Compras Aprobadas. | | ✓ | |
| 3 | procReport() | void | Realiza el reporte de la Consulta de Llegada de Mercadería. | | ✓ | |

**Observación:**

## 12.8. CLASE COMConsultaProv

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConsultaProv** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |
| Descripción: *La clase* COMConsultaProv | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | ProveedorexLinea(int Linea) | void | Presenta los Proveedores por Línea. | ✓ | | |
| 2 | main(String arg[]) | void | Programa Principal. | ✓ | | |
| 3 | COMConsultaProv () | | Construye la pantalla de la Consulta de Proveedores por Línea. | ✓ | | |

**Observación:**

## 12.9. CLASE COMConsSolCompra

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConsSolCompra** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMConsSolCompra genera la pantalla de la consulta de la Solicitud de Compra | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | btnMostrar | JButton() | Botón Mostrar. | ✓ | | |
| 2 | btnSalir | JButton() | Botón Salir. | ✓ | | |
| 3 | p | JPanel() | Panel agrupador de controles. | ✓ | | |
| 4 | p1 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 5 | p2 | JPanel() | Panel agrupador de controles. | ✓ | | |
| 6 | lblDesde | JLabel() | Etiqueta de Desde. | ✓ | | |
| 7 | lblHasta | JLabel() | Etiqueta de Hasta. | ✓ | | |
| 8 | Aprobadas | JCheckBox() | Combo de Orden de Compra Aprobada. | ✓ | | |
| 9 | Pendientes | JCheckBox() | Combo de Orden de Compra Pendiente. | ✓ | | |
| 10 | txtFechaIni | COMJDateField() | Caja de Texto FechaFin. | ✓ | | |
| 11 | txtFechaFin | COMJDateField() | Caja de Texto FechaIni. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMConsSolCompra** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMConsSolCompra genera la pantalla de la consulta de la Solicitud de Compra

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMConsSolCompra() | | Construye la pantalla de la Consulta de la Solicitud de Compra. | ✓ | | |
| 2 | procReport() | void | Realiza el reporte de la Consulta de Solicitud de Compra. | | ✓ | |

**Observación:**

## 12.10. CLASE COMDNumber

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDNumber** | Tipo:<br>**Pública** | Hereda de:<br>**PlainDocument** |
| Descripción: *La clase* COMDNumber realiza la validación de números. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | length | Integer | Variable de tipo integer. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDNumber** | Tipo:<br>**Pública** | Hereda de:<br>**PlainDocument** |
| Descripción: *La clase* COMDNumber realiza la validación de números. | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMDNumber() | | Construye la validación de números. | ✓ | | |
| 2 | COMDNumber(int l) | void | Método de validación. | ✓ | | |
| 3 | insertString(int offset, String string, AttributeSet attributes) | void | Inserta un String | ✓ | | |

**Observación:**

## 12.11. CLASE COMDateFieldDocument

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDateFieldDocument** | Tipo:<br>**Pública** | Hereda de:<br>**PlainDocument** |
| Descripción: *La clase* COMDateFieldDocument realiza la validación de las fechas | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | ANIO | String | Variable de tipo String. | | ✓ | |
| 2 | MES | String | Variable de tipo String. | | ✓ | |
| 3 | DIA | String | Variable de tipo String. | | ✓ | |
| 4 | dateFormat | SimpleDateFormat | Variable de tipo Date Simple. | | ✓ | |
| 5 | diasMax | Integer[] | Variable de tipo entero. | | ✓ | |
| 6 | initDate | Calendar | Variable de tipo calendario. | | ✓ | |
| 7 | initString | String | Variable de tipo String. | | ✓ | |
| 8 | newOffset | Integer | Variable de tipo entero. | | ✓ | |
| 9 | primeraVez | Boolean | Variable de tipo lógico. | | ✓ | |
| 10 | textComponent | JTextComponent() | Variable de tipo JTextComponent. | | ✓ | |
| 11 | trenner1 | Integer | Variable de tipo entero. | | ✓ | |
| 12 | trenner2 | Integer | Variable de tipo entero. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDateFieldDocument** | Tipo:<br>**Pública** | Hereda de:<br>**PlainDocument** |
| Descripción: *La clase* COMDateFieldDocument realiza la validación de las fechas | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMDateFieldDocument (JTextComponent textComponent) | | Construye la validación de las fechas. | ✓ | | |
| 2 | COMDateFieldDocument (JTextComponent textComponent, Calendar givenDate) | | Construye la validación de fechas gregorianas. | ✓ | | |
| 3 | insertString (int offset, String digito, AttributeSet attributeSet) | void | Permite generar los días y meses. | ✓ | | |
| 4 | remove (int offset, int length) | void | Lleva valor posicional de offset. | ✓ | | |
| 5 | atSeparator (int offset) | Boolean | Retorna offset | | ✓ | |
| 6 | esBisiesto (int anio) | Boolean | Retorna el año bisiesto. | | ✓ | |
| 7 | siError (int inic, int finl) | void | Lleva la fecha inicial y la final. | | ✓ | |

**Observación:**

## 12.12. CLASE COMDefPago

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDefPago** | Tipo:<br>**Pública** | Hereda de:<br>**JFrame** |
| Descripción: *La clase* COMDefPago realiza la definición de pago de la Solicitud de Compra. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | lblTipoPago | JLabel() | Etiqueta de Tipo Pago. | | ✓ | |
| 2 | lblTiempoPago | JLabel() | Etiqueta de Tiempo Pago. | | ✓ | |
| 3 | CmbTipoPago | JComboBox() | Combo de Tipo Pago. | | ✓ | |
| 4 | CmbTiempoPago | JComboBox() | Combo de Tiempo Pago. | | ✓ | |
| 5 | btnAceptar | JButton() | Botón Aceptar. | | ✓ | |
| 6 | btnCancelar; | JButton() | Botón Cancelar. | | ✓ | |
| 7 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 8 | Cred | String | Variable de tipo String. | | ✓ | |
| 9 | Tmax | String | Variable de tipo String. | | ✓ | |
| 10 | Rs | String[] | Variable de tipo String. | | ✓ | |

BIBLIOTECA CAMPUS PEÑAS

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDefPago** | Tipo:<br>**Pública** | Hereda de:<br>**JFrame** |
| Descripción: *La clase* COMDefPago realiza la definición de pago de la Solicitud de Compra. | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMDefPago (int codProv) | | Construye la pantalla de la Definición de Pago. | ✓ | | |
| 2 | consultaForPagoProveedores (int CodProv) | void | Realiza la consulta por Pago Proveedores. | | ✓ | |
| 3 | emitirPeriodos() | void | Emite los Periodos. | | ✓ | |

**Observación:**

## 12.13. CLASE COMDefinirProv

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDefinirProv** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMDefinirProv realiza la definición del proveedor en la Solicitud de Compra. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | btnGuardar | JButton() | Botón Aceptar. | | ✓ | |
| 13 | btnAsigProv | JButton() | Botón Asignar Proveedor. | | ✓ | |
| 14 | btnSalir | JButton() | Botón Salir. | | ✓ | |
| 15 | lblNoOrden | JLabel() | Etiqueta de No. de Orden. | | ✓ | |
| 16 | lblFecha | JLabel() | Etiqueta de Fecha. | | ✓ | |
| 17 | lblLinea | JLabel() | Etiqueta de Línea. | | ✓ | |
| 18 | lblProveedor | JLabel() | Etiqueta del Proveedor. | | ✓ | |
| 19 | txtNoOrden | JTextField() | Caja de Texto de No. Orden. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDefinirProv** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMDefinirProv realiza la definición del proveedor en la Solicitud de Compra.

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | TxtProveedor | JTextField() | Caja de Texto de Proveedor. | | ✓ | |
| 21 | CmbLineas | JComboBox() | Combo de Líneas. | | ✓ | |
| 22 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 23 | indice | Integer | Variable de tipo Entero. | | ✓ | |
| 24 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 25 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 26 | p2 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 27 | p3 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 28 | p4 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 29 | datamodel | MiTablaModelo2 | Variable de clase MiTablaModelo | | ✓ | |
| 30 | table | JTable() | Variable de la clase Tabla. | | ✓ | |
| 31 | scrollPane | JScrollPane() | Barra de Desplazamiento. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDefinirProv** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMDefinirProv realiza la definición del proveedor en la Solicitud de Compra. | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMDefinirProv (int Usuario, int Grupo, int Modulo, int Empresa, int Opcion) | | Construye la pantalla de la Definición del Proveedor. | ✓ | | |
| 2 | BuscarLineas () | void | Realiza la Búsqueda de Líneas. | | ✓ | |
| 3 | NumOrdenCompra () | void | Obtiene el Número de Orden. | | ✓ | |
| 4 | actCabSolCom () | void | Actualiza la Solicitud de Compra. | | ✓ | |
| 5 | btnGuardarAccionado () | void | Guarda lo accionado. | | ✓ | |
| 6 | getProveedor (int Linea) | void | Obtiene el Proveedor por Línea. | | ✓ | |
| 7 | guardarLineas () | void | Guarda las Líneas. | | ✓ | |
| 8 | iniciarComponentes () | void | Permite activar todos los botones de la Definición del Proveedor. | | ✓ | |
| 9 | llenarCmbLineas (ResultSet B) | void | Llena el combo de las líneas. | | ✓ | |
| 10 | opr_seg_estadoBotones (boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Permite activar el estado de los botones y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 11 | tablaVacia () | void | Permite instanciar a todas las Tablas existentes. | | ✓ | |
| 12 | tablevacia () | void | Permite limpiar la Tabla actual. | | ✓ | |
| 13 | validaCombo () | void | Valida el Combo. | | ✓ | |

**Observación:**

## 12.14. CLASE COMDevolucionMerc

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDevolucionMerc** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMDevolucionMerc realiza la devolución de mercadería. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | btnGuardar | JButton() | Botón Aceptar. | | ✓ | |
| 13 | btnCancelar | JButton() | Botón Cancelar. | | ✓ | |
| 14 | btnSalir | JButton() | Botón Salir. | | ✓ | |
| 15 | lblCantidad | JLabel() | Etiqueta de Cantidad | | ✓ | |
| 16 | lblFecha | JLabel() | Etiqueta de Fecha. | | ✓ | |
| 17 | lblLinea | JLabel() | Etiqueta de Línea. | | ✓ | |
| 18 | lblProveedor | JLabel() | Etiqueta del Proveedor. | | ✓ | |
| 19 | lblCodProd | JLabel() | Etiqueta del Código del Producto. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDevolucionMerc** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMDevolucionMerc realiza la devolución de mercadería.

## DESCRIPCIÓN DE CLASE

### ATRIBUTOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | lblNomProd | JLabel() | Etiqueta del Nombre del Producto. | | ✓ | |
| 21 | lblObservacion | JLabel() | Etiqueta de la Observación. | | ✓ | |
| 22 | lblCantidad | JTextField() | Caja de Texto de Cantidad. | | ✓ | |
| 23 | txtCodProd | JTextField() | Caja de Texto de Código Producto. | | ✓ | |
| 24 | txtNomProd | JTextField() | Caja de Texto Nombre Producto. | | ✓ | |
| 25 | txtObservacion | JTextField() | Caja de Texto de Observación. | | ✓ | |
| 26 | txtProveedor | JTextField() | Caja de Texto de Proveedor. | | ✓ | |
| 27 | CmbLineas | JComboBox() | Combo de Líneas. | | ✓ | |
| 28 | cmbProveedores | JComboBox() | Combo de Proveedores. | | ✓ | |
| 29 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 30 | p | JPanel() | Panel agrupador de controles. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMDevolucionMerc** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMDevolucionMerc realiza la devolución de mercadería.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMDevolucionMerc (int Usuario, int Grupo, int Modulo, int Empresa, int Opcion) | | Construye la pantalla de la Devolución de Mercadería. | ✓ | | |
| 2 | btnGuardarAccionado () | void | Guarda lo accionado. | | ✓ | |
| 3 | iniciarComponentes () | void | Permite activar todos los botones de la Definición del Proveedor. | | ✓ | |
| 4 | llenarLineas (int codProv) | void | Permite Llenar las Líneas. | | ✓ | |
| 5 | llenarProveedores () | void | Permite Llenar los Proveedores. | | ✓ | |
| 6 | opr_seg_estadoBotones (boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Permite activar el estado de los botones y verifica si tiene acceso el grupo de usuario. | | ✓ | |

**Observación:**

BIBLIOTECA CAMPUS PENAS ESPOL

## 12.15. CLASE COMEmiPagoOrdCom

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 4<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMEmiPagoOrdCom** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMEmiPagoOrdCom realiza la Emisión y Forma de Pago de la Orden de Compra. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | btnGuardar | JButton() | Botón Aceptar. | | ✓ | |
| 13 | btnImprimir | JButton() | Botón Imprimir. | | ✓ | |
| 14 | btnSalir | JButton() | Botón Salir. | | ✓ | |
| 15 | lblDescuento | JLabel() | Etiqueta de Descuento. | | ✓ | |
| 16 | lblFecha | JLabel() | Etiqueta de Fecha. | | ✓ | |
| 17 | lblFormaPago | JLabel() | Etiqueta de Forma de Pago | | ✓ | |
| 18 | lblProveedor | JLabel() | Etiqueta del Proveedor. | | ✓ | |
| 19 | lblIntereses | JLabel() | Etiqueta de Intereses | | ✓ | |

**Observación:**

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 4<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMEmiPagoOrdCom** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMEmiPagoOrdCom realiza la Emisión y Forma de Pago de la Orden de Compra.

## DESCRIPCIÓN DE CLASE

### ATRIBUTOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | lblIva | JLabel() | Etiqueta del I.V.A. | | ✓ | |
| 21 | lblNPagos | JLabel() | Etiqueta del Número de Pagos. | | ✓ | |
| 22 | lblPeriodoPago | JLabel() | Etiqueta del Período de Pago. | | ✓ | |
| 23 | lblPrimerPago | JLabel() | Etiqueta del Prime Pago | | ✓ | |
| 24 | lblSubTotal | JLabel() | Etiqueta del Subtotal | | ✓ | |
| 25 | lblTotal | JLabel() | Etiqueta del Total. | | ✓ | |
| 26 | txtDescuento | JTextField() | Caja de Texto de Descuento. | | ✓ | |
| 27 | txtIntereses | JTextField() | Caja de Texto de Intereses. | | ✓ | |
| 28 | txtIva | JTextField() | Caja de Texto del I.V.A. | | ✓ | |
| 29 | txtNPagos | JTextField() | Caja de Texto de No. de Pagos. | | ✓ | |
| 30 | txtSubTotal | JTextField() | Caja de Texto de Subtotal. | | ✓ | |
| 31 | txtTotal | JTextField() | Caja de Texto de Total. | | ✓ | |
| 31 | txtProveedor | JTextField() | Caja de Texto de Proveedor. | | ✓ | |
| 32 | CmbProveedores | JComboBox() | Combo de Proveedores. | | ✓ | |
| 33 | txtFormaPago | JComboBox() | Combo de Forma de Pago. | | ✓ | |
| 34 | txtPeriodoPago | JComboBox() | Combo de Periodo de Pago. | | ✓ | |
| 35 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 36 | txtPrimerPago | COMJDateField() | Caja de Texto de Primer Pago. | | ✓ | |
| 37 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 38 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 39 | p2 | JPanel() | Panel agrupador de controles. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 4<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMEmiPagoOrdCom** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMEmiPagoOrdCom realiza la Emisión y Forma de Pago de la Orden de Compra.

## DESCRIPCIÓN DE CLASE

### ATRIBUTOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 40 | p3 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 41 | p4 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 42 | datamodel | MiTablaModelo6 | Variable de clase MiTablaModelo6 | | ✓ | |
| 43 | table | JTable() | Variable de la clase Tabla. | | ✓ | |
| 44 | scrollPane | JScrollPane() | Barra de Desplazamiento. | | ✓ | |
| 45 | Aud | String | Variable de tipo String. | | ✓ | |
| 46 | Cred | String | Variable de tipo String. | | ✓ | |
| 47 | Rs | String[] | Variable de tipo String. | | ✓ | |
| 48 | Tmax | String | Variable de tipo String. | | ✓ | |

**Observación:**

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 4 / 4<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMEmiPagoOrdCom** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMEmiPagoOrdCom realiza la Emisión y Forma de Pago de la Orden de Compra.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMEmiPagoOrdCom (int Usuario, int Grupo, int Modulo, int Empresa, int Opcion) | | Construye la pantalla de la Emisión y Forma de Pago de la Orden de Compra. | ✓ | | |
| 2 | actualizarOrdenCompra () | void | Actualiza la Orden de Compra. | | ✓ | |
| 3 | adicionarDatosDetOC () | void | Adiciona Datos a la Orden de Compra. | | ✓ | |
| 4 | btnGuardarAccionado () | void | Guarda lo accionado. | | ✓ | |
| 5 | consultaForPagoProveedores (int CodProv) | void | Realiza la consulta para Pago Proveedores. | | ✓ | |
| 6 | consultaValorIvaDesc () | void | Realiza consulta de ValorIvaDesc. | | ✓ | |
| 7 | emitirPeriodos (int tiempo) | void | Emite los Periodos de Tiempo. | | ✓ | |
| 8 | iniciarComponentes () | void | Permite activar todos los botones de la Emisión Orden de Compra. | | ✓ | |
| 9 | llenarGrid () | void | Permite llenar el Combo de Proveedores. | | ✓ | |
| 10 | llenarTiepoPago () | void | Permite llenar combo Tipo Pago. | | ✓ | |
| 11 | opr_seg_estadoBotones (boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Permite activar el estado de los botones y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 12 | procReport (String codiProv) | void | Realiza el reporte de la Emisión de Orden de Compra. | | ✓ | |
| 13 | setDatosFormaPago () | void | Indica la Forma de Pago. | | ✓ | |
| 14 | setInteres (String cant) | void | Indica el Interés. | | ✓ | |
| 15 | setIva (String cant) | void | Indica el I.V.A. | | ✓ | |
| 16 | setSubTotal () | void | Indica el Subtotal. | | ✓ | |
| 17 | setTotal (String cant) | void | Indica el Total. | | ✓ | |
| 18 | setsubTotal (String cant) | void | Indica la cantidad del Subtotal. | | ✓ | |

**Observación:**

## 12.16. CLASE COMFormatSwing

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 6<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMFormatSwing** | Tipo:<br>**Pública** | Hereda de: |

Descripción: *La clase* COMFormatSwing *permite la creación de los principales componentes graficos (Swing).*

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | ge | GraphicsEnvironment | Maneja el amnbiente gráfico | | ✓ | |
| 2 | letrBotn | Font | Letra para JButton. | | ✓ | |
| 3 | letrComb | Font | Letra para JComboBox. | | ✓ | |
| 4 | letrDefl | Font | Letra predeterminada. | | ✓ | |
| 5 | letrEtiq | Font | Letra para JLabel. | | ✓ | |
| 6 | letrList | Font | Letra para JList. | | ✓ | |
| 7 | letrLs | String [] | Arreglo de letras disponibles. | | ✓ | |
| 8 | letrMenu | Font | Letra para JMenu. | | ✓ | |
| 9 | letrText | Font | Letra para JTextField. | | ✓ | |

**Observación:**

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 6<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMFormatSwing** | Tipo:<br>**Pública** | Hereda de: |

Descripción: *La clase* COMFormatSwing *permite la creaciòn de los principales componentes graficos (Swing).*

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | adicBagCons (GridBagConstraints objCons, int llenado, int orientado, int minimoX, int minimoY) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 2 | adicBagCons (GridBagConstraints objCons, int llenado, int orientado, int minimoX) | void | Permite el manejo de Layout GridBackLayout | ✓ | | |
| 3 | adicBagCons (GridBagConstraints objCons, int llenado, int orientado) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 4 | adicBagCons (GridBagConstraints objCons, int llenado) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 5 | defiBagCons (int x, int y, int celdsX, int celdsY, double pesoX, double pesoY, int llenado, int orientado, Insets espacio, int minimoX, int minimoY) | GridBagConstraints | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 6 | defiBagCons (GridBagConstraints objCons, int x, int y, int celdsX, int celdsY, double pesoX, double pesoY, Insets espacio) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 7 | defiBagCons (GridBagConstraints objCons, int x, int y, int celdsX, int celdsY, double pesoX, double pesoY) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 8 | defiBagCons (GridBagConstraints objCons, int x, int y, int celdsX, int celdsY) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 6<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMFormatSwing** | Tipo:<br>**Pública** | Hereda de: |

Descripción: *La clase* COMFormatSwing *permite la creaciòn de los principales componentes graficos (Swing).*

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 9 | defiBagCons (GridBagConstraints objCons, int x, int y) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 10 | defiBagCons (GridBagConstraints objCons, int x, int y, int celdsX, int celdsY, Insets espacio) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 11 | defiBagCons (GridBagConstraints objCons, int x, int y, Insets espacio) | void | Permite el manejo de Layout GridBackLayout. | ✓ | | |
| 12 | defiBotn (String nomb, String titu, ImageIcon icon, Font letr, int posiTextVert, int posiTextHorz, char mnem, ActionListener listener) | JButton() | Permite definir las propiedades de un JButton. | ✓ | | |
| 13 | defiBotn (String nomb, String titu, ImageIcon icon, int posiTextVert, int posiTextHorz, char mnem, ActionListener listener) | JButton() | Permite definir las propiedades de un JButton. | ✓ | | |
| 14 | defiBotn (String nomb, String titu, ImageIcon icon, int posiTextVert, int posiTextHorz, char mnem) | JButton() | Permite definir las propiedades de un JButton. | ✓ | | |
| 15 | defiBotn (String nomb, String titu, ImageIcon icon, Font letr, String actnCommd, String toolTip, ActionListener listener, MouseListener mousList) | JButton() | Permite definir las propiedades de un JButton. | ✓ | | |
| 16 | defiBotn (String nomb, String titu, ImageIcon icon, String actnCommd, String toolTip, ActionListener listener, MouseListener mousList) | JButton() | Permite definir las propiedades de un JButton. | ✓ | | |

**Observación:**

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 4 / 6<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMFormatSwing** | Tipo:<br>**Pública** | Hereda de: |

Descripción: *La clase* COMFormatSwing *permite la creación de los principales componentes graficos (Swing).*

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 17 | defiComb (Font letr, float alinVert, int visiFils, ActionListener listener, String nomb) | JComboBox() | Permite definir las características de un JComboBox. | ✓ | | |
| 18 | defiComb (float alinVert, int visiFils, ActionListener listener, String nomb) | JComboBox() | Permite definir las características de un JComboBox. | ✓ | | |
| 19 | defiComb (Font letr, float alinVert, int visiFils) | JComboBox() | Permite definir las características de un JComboBox. | ✓ | | |
| 20 | defiComb (float alinVert, int visiFils) | JComboBox() | Permite definir las características de un JComboBox. | ✓ | | |
| 21 | defiComb (float alinVert, int visiFils, String nomb) | JComboBox() | Permite definir las características de un JComboBox. | ✓ | | |
| 22 | defiEtiq (String titu, Font letr) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 23 | defiEtiq (String titu, Font letr, char mnem) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 24 | defiEtiq (String titu) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 25 | defiEtiq (String titu, char mnem) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 26 | defiEtiq (String titu, Font letr, String toolTip) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 27 | defiEtiq (String titu, Font letr, String toolTip, char mnem) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 28 | defiEtiq (String titu, String toolTip) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |
| 29 | defiEtiq (String titu, String toolTip, char mnem) | JLabel() | Permite definir las características de un JLabel. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 5 / 6<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMFormatSwing** | Tipo:<br>**Pública** | Hereda de: |

Descripción: *La clase* COMFormatSwing *permite la creaciòn de los principales componentes graficos (Swing).*

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 30 | defiText (Document docu, int colu, Font letr, ActionListener listener, char teclAccs, String nomb) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 31 | defiText (Document docu, int colu, ActionListener listener, char teclAccs, String nomb) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 32 | defiText (int colu, ActionListener listener, char teclAccs, String nomb) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 33 | defiText (int colu, char teclAccs, String nomb) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 34 | defiText (Document docu, int colu, Font letr, char teclAccs, String nomb) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 35 | defiText (Document docu, int colu, char teclAccs, String nomb) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 36 | defiText (Document docu, int colu, char teclAccs) | JTextField() | Permite definir las características de un JTextField(). | ✓ | | |
| 37 | estaFramAqui (String framBusc, JDesktopPane desktop) | Boolean | Determina si el InternalFrame a instanciar ya se encuentra en el Escritorio. | ✓ | | |
| 38 | llamFram (JInternalFrame form, JDesktopPane desktop) | void | Agrega, ubica y hace visible un JInternalFrame al JDesktopPane. | ✓ | | |
| 39 | llenComb (JComboBox comb, String[] elem) | void | Llena un JComboBox con arreglo del parámetro. | ✓ | | |
| 40 | llenList (JList list, String[] elem | void | Llena lista con el arreglo del parámetro. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 6 / 6 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMFormatSwing** | Tipo:<br>**Pública** | Hereda de: |

Descripción: *La clase* COMFormatSwing *permite la creación de los principales componentes graficos (Swing).*

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 41 | traeLetr (String letra, int estl, int tama) | Font | Retorna un objeto Font, de acuerdo a los parámetros; sino, envía letrDefl(Font) | ✓ | | |
| 42 | ubicFram (JInternalFrame form, JDesktopPane desktop) | void | Manipula la ubicación de los internalFrames dentro del MDI | ✓ | | |
| 43 | ubicItemCmbo (JComboBox cmbObjt, String elemBusc) | void | Selecciona un item del combo. | ✓ | | |
| 44 | modfCmbo (JComboBox comp, ActionListener listener, Font letr, String nomb) | void | Permite definir las características de un JTextField(). | | | ✓ |
| 45 | modfText (JTextField comp, ActionListener listener, Font letr, String nomb, char teclAccs) | void | Permite modificar el combo. | | | ✓ |
| 46 | busqLetr (String letrBusc) | Boolean | Permite realizar bússqueda de letra, para determinar si existe(true) o no(false). | | ✓ | |
| 47 | modfBotn (JButton comp, ActionListener listener, MouseListener mousList, Font letr, String nomb) | void | Permite realizar modificaciones sobre los botones. | | ✓ | |

**Observación:**

ESPOL BIBL... CAMPUS PEÑAS

## 12.17. CLASE COMJDateField

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMJDateField** | Tipo:<br>**Pública** | Hereda de:<br>**JTextField** |
| Descripción: *La clase* COMJDateField es una clase propia de Java que ayuda a generar la fecha. | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | anioInt | Integer | Variable de tipo Entero. | | ✓ | |
| 2 | anioString | String | Variable de tipo String. | | ✓ | |
| 3 | diaInt | Integer | Variable de tipo Entero. | | ✓ | |
| 4 | diaString | String | Variable de tipo String. | | ✓ | |
| 5 | mesInt | Integer | Variable de tipo Entero. | | ✓ | |
| 6 | mesString | String | Variable de tipo String. | | ✓ | |
| 7 | myDates | Calendar | Variable de tipo Calendar. | | ✓ | |
| 8 | tageMax | static final int[] | Variable de tipo static final entero. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMJDateField** | Tipo:<br>**Pública** | Hereda de:<br>**JTextField** |
| Descripción: *La clase* COMJDateField es una clase propia de Java que ayuda a generar la fecha.. | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMJDateField () | | Construye la fecha del sistema. | ✓ | | |
| 2 | COMJDateField (int columns) | | Construye la fecha del sistema, especifica el número de columnas y la fecha actual (fecha predefinida: Hoy). | ✓ | | |
| 3 | COMJDateField (String text) | | Construye la fecha del sistema, para una fecha-String de la forma mm.dd.yyyy (xx.xx.xxxx ). | ✓ | | |
| 4 | COMJDateField (String text, int columns) | | Construye la fecha del sistema., para una fecha-String de la forma mm.dd.yyyy | ✓ | | |
| 5 | COMJDateField (Calendar myDate) | | Construye la fecha del sistema, especifica el calendario actual. | ✓ | | |
| 6 | COMJDateField (Calendar myDate, int columns) | | Construye la fecha del sistema, especifica el calendario actual. | ✓ | | |
| 7 | COMJDateField (int dia, int mes, int anio) | | Construye la fecha del sistema, especifica día, mes y año. | ✓ | | |
| 8 | COMJDateField (int dia, int mes, int anio, int columns) | | Construye la fecha del sistema, especifica día, mes y año. | ✓ | | |
| 9 | focusGained (FocusEvent focusevent) | void | Este evento indica que el componente es ahora el propio focus | ✓ | | |
| 10 | focusLost (FocusEvent focusevent) | void | Este evento indica que el componente no es más largo que el propio focus. | ✓ | | |
| 11 | getDate () | Calendar | Obtiene-el método para el objeto Calendar. | ✓ | | |
| 12 | getDay () | Integer | Obtiene-el método para el día. | ✓ | | |
| 13 | getDayText () | String | Obtiene-el método para el día. | ✓ | | |
| 14 | getMonth () | Integer | Obtiene-el método para el mes. | ✓ | | |
| 15 | getMonthText () | String | Obtiene-el método para el mes. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMJDateField** | Tipo:<br>**Pública** | Hereda de:<br>**JTextField** |
| Descripción: *La clase* COMJDateField es una clase propia de Java que ayuda a generar la fecha.. | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 16 | getYear () | Integer | Obtiene-el método para el año. | ✓ | | |
| 17 | getYearText () | String | Obtiene-el método para el año | ✓ | | |
| 18 | hasNoZeroField () | Boolean | Método para validar fecha. | ✓ | | |
| 19 | hasZeroField () | Boolean | Método para validar fecha. | ✓ | | |
| 20 | setDate (Calendar myNewDate) | void | Asigna la fecha del calendario. | ✓ | | |
| 21 | setDate (String myDateText) | void | Asigna la fecha de forma String. | ✓ | | |
| 22 | setDate (int day, int month, int year) | void | Asigna la fecha de tres valores enteros. | ✓ | | |
| 23 | setDay (int day) | void | Asigna el día. | ✓ | | |
| 24 | setDayText (String day) | void | Asigna el día de forma String. | ✓ | | |
| 25 | setMonth (int month) | void | Asigna el mes. | ✓ | | |
| 26 | setMonthText (String month) | void | Asigna el mes de forma String. | ✓ | | |
| 27 | setYear (int year) | void | Asigna el año. | ✓ | | |
| 28 | setYearText (String year) | void | Asigna el año de forma String. | ✓ | | |
| 29 | isDate (int dia, int mes, int anio) | Boolean | Método para validar calendario gregoriano. | ✓ | | |
| 30 | isDate (String checkString) | Boolean | Método para validar calendario gregoriano. | ✓ | | |
| 31 | _cls1 () | void | Método que realiza la construcción | | ✓ | |
| 32 | attributeUpdate () | void | Método que actualiza la fecha. | | ✓ | |
| 33 | constructor$0 (COMJDateField datetextfield) | void | Método que realiza la construcción | | ✓ | |

**Observación:**

## 12.18. CLASE COMLlegDevMercaderia

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMLlegDevMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMLlegDevMercaderia *permite la llegada de devolución de mercadería.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | algunoAct | Integer | Variable de tipo entero. | | ✓ | |
| 13 | btnBuscar | JButton() | Botón de Buscar. | | ✓ | |
| 14 | btnBuscarDev | JButton() | Botón de Buscar Devolución. | | ✓ | |
| 15 | btnGuardar | JButton() | Botón de Guardar. | | ✓ | |
| 16 | btnSalir | JButton() | Botón de Salir. | | ✓ | |
| 17 | cmbProveedor | JComboBox() | Combo de Proveedor. | | ✓ | |
| 18 | codEmp | Integer | Variable de tipo entero. | | ✓ | |
| 19 | datamodel | MiTablaModelo8 | Variable de clase MiTablaModelo8 | | ✓ | |

**Observación:**

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMLlegDevMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMLlegDevMercaderia *permite la llegada de devolución de mercadería.*

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | PorDevInv | Boolean | Variable de tipo Boolean. | | ✓ | |
| 21 | elimItem | Boolean | Variable de tipo Boolean. | | ✓ | |
| 22 | IBand | Boolean | Variable de tipo Boolean. | | ✓ | |
| 23 | lblNoOrdenCom | JLabel() | Etiqueta de la Número de Orden de Compra. | | ✓ | |
| 24 | lblProveedor | JLabel() | Etiqueta del Proveedor. | | ✓ | |
| 25 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 26 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 27 | scrollPane | JScrollPane() | Barra de desplazamiento. | | ✓ | |
| 28 | table | JTable | Variable de la clase Tabla. | | ✓ | |
| 29 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 30 | txtNoOrdenCom | JTextField() | Caja de la Número de Orden de Compra. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ultima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMLlegDevMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMLlegDevMercaderia *permite la llegada de devolución de mercadería.* | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMLlegDevMercaderia (int Usuario, int Grupo, int Modulo, int Empresa, int Opcion) | | Construye la Llegada de Devolución de Mercadería. | ✓ | | |
| 2 | btnGuardarAccionado () | void | Guarda lo accionado. | | ✓ | |
| 3 | despuesdeGuardar () | void | Limpia la tabla después de guardar. | | ✓ | |
| 4 | eliminarItem () | void | Elimina el ítem. | | ✓ | |
| 5 | getProveedor (int codProv) | String | Obtiene el Proveedor. | | ✓ | |
| 6 | iniciarComponentes () | void | Permite activar todos los botones de la Llegada de Devolución de Mercadería. | | ✓ | |
| 7 | llenarCombDevoluciones () | void | Llena el Combo de Devoluciones. | | ✓ | |
| 8 | opr_seg_estadoBotones (boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Permite activar el estado de los botones y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 9 | vacear () | void | Vacia la Tabla. | | ✓ | |
| 10 | validaBusqueda () | Boolean | Validación de Búsqueda. | | ✓ | |
| 11 | validaCmbDevoluciones () | void | Valida Combo de Devoluciones. | | ✓ | |
| 12 | validaDevPendientes () | void | Valida las Devoluciones Pendientes. | | ✓ | |

**Observación:**

## 12.19. CLASE COMLlegMercaderia

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMLlegMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMLlegMercaderia *permite la llegada de la mercadería.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoOpcion | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vl_seg_botonBuscar | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vl_seg_botonEliminar | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vl_seg_botonGrabar | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vl_seg_botonModificar | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vl_seg_botonNuevo | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vl_seg_botonRevertir | Integer | Variable de tipo entero. | ✓ | | |
| 12 | algunoAct | Integer | Variable de tipo entero. | | ✓ | |
| 13 | btnBuscar | JButton() | Botón de Buscar. | | ✓ | |
| 14 | btnGuardar | JButton() | Botón de Guardar. | | ✓ | |
| 15 | btnSalir | JButton() | Botón de Salir. | | ✓ | |
| 16 | codEmp | Integer | Variable de tipo entero. | | ✓ | |
| 17 | datamodel | MiTablaModelo7 | Variable de clase MiTablaModelo7 | | ✓ | |
| 18 | iBand | Boolean | Variable de tipo Boolean. | | ✓ | |
| 19 | lblNoOrdenCom | JLabel() | Etiqueta de la Número de Orden de Compra. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMLlegMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMLlegMercaderia *permite la llegada de la mercadería.*

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | lblNumFact | JLabel() | Etiqueta del Número de Factura. | | ✓ | |
| 21 | lblProveedor | JLabel() | Etiqueta del Proveedor. | | ✓ | |
| 22 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 23 | p1 | JPanel() | Panel agrupador de controles. | | ✓ | |
| 24 | scrollPane | JScrollPane() | Barra de desplazamiento. | | ✓ | |
| 25 | table | JTable | Variable de la clase Tabla. | | ✓ | |
| 26 | txtFecha | COMJDateField() | Caja de Texto de Fecha. | | ✓ | |
| 27 | txtNoOrdenCom | JTextField() | Caja de texto del Número de Orden de Compra. | | ✓ | |
| 28 | txtNumFact | JTextField() | Caja de texto del Número de Factura. | | ✓ | |
| 29 | txtProveedor | JTextField() | Caja de texto del Proveedor. | | ✓ | |

ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMLlegMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMLlegMercaderia *permite la llegada de la mercadería.* | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMLlegMercaderia (int Usuario, int Grupo, int Modulo, int Empresa, int Opcion) | | Construye la Llegada de Devolución de Mercadería. | ✓ | | |
| 2 | btnGuardarAccionado () | void | Guarda lo accionado. | | ✓ | |
| 3 | buscarOrdenesCompra () | void | Busca las Ordenes de Compra. | | ✓ | |
| 4 | iniciarComponentes () | void | Permite activar todos los botones de la Llegada de Mercadería. | | ✓ | |
| 5 | llenarTabla (int numOrd) | void | Llena la tabla de la Llegada de Mercadería. | | ✓ | |
| 6 | opr_seg_estadoBotones (boolean nuevo, boolean grabar, boolean modificar, boolean eliminar, boolean revertir, boolean buscar) | void | Permite activar el estado de los botones y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 7 | tablaVacia () | void | Vacia la Tabla. | | ✓ | |
| 8 | validaOrdenesEnviadas () | void | Valida las Ordenes Enviadas. | | ✓ | |

**Observación:**

## 12.20. CLASE COMPrincipal

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMPrincipal** | Tipo:<br>**Pública** | Hereda de:<br>**JFrame** |
| Descripción: *La clase* COMPrincipal *genera un formulario de presentación del sistema,Ingreso a los mantenimientos, ingreso a Movimientos y finalmente a los Reportes..* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | vg_seg_codigoEmpresa | Integer | Variable de tipo entero. | ✓ | | |
| 2 | vg_seg_codigoGrupo | Integer | Variable de tipo entero. | ✓ | | |
| 3 | vg_seg_codigoModulo | Integer | Variable de tipo entero. | ✓ | | |
| 4 | vg_seg_codigoUsuario | Integer | Variable de tipo entero. | ✓ | | |
| 5 | vg_seg_opcionAprobaSolicitud | Integer | Variable de tipo entero. | ✓ | | |
| 6 | vg_seg_opcionDefinirProveedor | Integer | Variable de tipo entero. | ✓ | | |
| 7 | vg_seg_opcionDevolucionMercaderia | Integer | Variable de tipo entero. | ✓ | | |
| 8 | vg_seg_opcionEmisionOrdPago | Integer | Variable de tipo entero. | ✓ | | |
| 9 | vg_seg_opcionLlegMercDevuelta | Integer | Variable de tipo entero. | ✓ | | |
| 10 | vg_seg_opcionLlegadaMercaderia | Integer | Variable de tipo entero. | ✓ | | |
| 11 | vg_seg_opcionRepLleMercaderia | Integer | Variable de tipo entero. | ✓ | | |
| 12 | vg_seg_opcionRepOrdCompra | Integer | Variable de tipo entero. | ✓ | | |
| 13 | vg_seg_opcionRepSolCompra | Integer | Variable de tipo entero. | ✓ | | |
| 14 | vg_seg_opcionSolicBajoPeticion | Integer | Variable de tipo entero. | ✓ | | |
| 15 | vg_seg_opcionSolicBajoReposicion | Integer | Variable de tipo entero. | ✓ | | |
| 16 | B1 | JButton() | Botón de B1. | ✓ | | |
| 17 | B2 | JButton() | Botón de B2. | ✓ | | |
| 18 | B3 | JButton() | Botón de B3. | ✓ | | |
| 19 | B4 | JButton() | Botón de B4. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMPrincipal** | Tipo:<br>**Pública** | Hereda de:<br>**JFrame** |

Descripción: *La clase* COMPrincipal *genera un formulario de presentación del sistema, otro formulario de ingreso al sistema., y finalmente uno de ingreso a módulos.*

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 20 | B5 | JButton() | Botón de B5. | ✓ | | |
| 21 | B6 | JButton() | Botón de B6. | ✓ | | |
| 22 | jdpPrincipal | JDesktopPane() | Menú Principal. | | ✓ | |
| 23 | mnuAcercadePry | JMenuItem() | SubMenú Acerca de... | | ✓ | |
| 24 | mnuAprovarOrdenCompra | JMenuItem() | SubMenú Aprobar Orden de Compra. | | ✓ | |
| 25 | mnuDefinirProveedor | JMenuItem() | SubMenú Definir Proveedor. | | ✓ | |
| 26 | mnuDevolucionMercaderia | JMenuItem() | SubMenú Devolución Mercadería. | | ✓ | |
| 27 | mnuEmisionOrdenPago | JMenuItem() | SubMenú Emisión Orden de Pago. | | ✓ | |
| 28 | mnuIngresoLineas | JMenuItem() | SubMenú Ingreso de Líneas. | | ✓ | |
| 29 | mnuLlegMercConsultas | JMenuItem() | SubMenú Consulta Llegada de Mercadería. | | ✓ | |
| 30 | mnuLlegMercDevuelta | JMenuItem() | SubMenú Llegada de Mercadería Devuelta. | | ✓ | |
| 31 | mnuLlegadaMercaderia | JMenuItem() | SubMenú Llegada de Mercadería. | | ✓ | |
| 32 | mnuOrdCompraConsultas | JMenuItem() | SubMenú Consulta Orden de Compra. | | ✓ | |
| 33 | mnuSalir | JMenuItem() | SubMenú Salir. | | ✓ | |
| 34 | mnuSolCompraConsultas | JMenuItem() | SubMenú Consulta Solicitud Compra. | | ✓ | |
| 35 | mnuSolCompraPeticion | JMenuItem() | SubMenú Solicitud Compra por Petición. | | ✓ | |
| 36 | mnuSolCompraReposicion | JMenuItem() | SubMenú Solicitud Compra por Reposición. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMPrincipal** | Tipo:<br>**Pública** | Hereda de:<br>**JFrame** |

Descripción: *La clase* COMPrincipal *genera un formulario de presentación del sistema, otro formulario de ingreso al sistema., y finalmente uno de ingreso a módulos.*

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMPrincipal (int Usuario, int Grupo, int Modulo, int Empresa, String nombreUsuario, String nombreEmpresa) | | Construye el Menú Principal.. | ✓ | | |
| 2 | main (String[] arg) | void | Construye el main Principal. | ✓ | | |
| 3 | generarMenu () | JMenuBar() | Genera el Menú. | ✓ | | |
| 4 | opr_seg_bloquearOpcion (int vg_seg_codigoGrupo) | void | Permite bloquear el estado de los botones y verifica si tiene acceso el grupo de usuario. | | ✓ | |
| 5 | visiMaxi () | void | Maximiza el JFrame. | | ✓ | |

**Observación:**

## 12.21. CLASE COMTablSort

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMTablSort** | Tipo:<br>**Pública** | Hereda de:<br>**COMTablMap** |
| Descripción: *La clase* COMTablSort *permite ordenar la tabla al dar un click.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | asc | Boolean | Variable de tipo lógico. | | ✓ | |
| 2 | ascending | Boolean | Variable de tipo lógico. | | ✓ | |
| 3 | compares | Integer | Variable de tipo entero. | | ✓ | |
| 4 | indexes | Integer | Variable de tipo entero. | | ✓ | |
| 5 | sortingColumns | Vector | Variable de tipo vector. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMTablSort** | Tipo:<br>**Pública** | Hereda de:<br>**COMTablMap** |
| Descripción: *La clase* COMTablSort *permite ordenar la tabla al dar un click.* | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMTablSort () | | Construye la tabla sort. | ✓ | | |
| 2 | COMTablSort (TableModel model) | | Construye la tabla sort. | ✓ | | |
| 3 | addMouseListenerToHeaderIn Table (JTable table) | void | Evento que sirve para cuando un encabezando de columna es pulsado. | ✓ | | |
| 4 | checkModel () | void | Validación para cambio en el Model. | ✓ | | |
| 5 | compare (int row1, int row2) | Integer | Compara 2 filas. | ✓ | | |
| 6 | compareRowsByColumn (int row1, int row2, int column) | Integer | Compara 2 filas en una columna. | ✓ | | |
| 7 | getValueAt (int aRow, int aColumn) | Object | Obtiene el valor. | ✓ | | |
| 8 | n2sort () | void | Compara 2 índices ordenados. | ✓ | | |
| 9 | reallocateIndexes () | void | Monta un arreglo de índices con el número de elementos al datamodel. | ✓ | | |
| 10 | setModel (TableModel model) | void | Coloca el arreglo de índices en el model. | ✓ | | |
| 11 | setValueAt (Object aValue, int aRow, int aColumn) | void | Coloca el valor. | ✓ | | |
| 12 | shuttlesort (int[] from, int[] to, int low, int high) | void | Realiza un enlace entre 2 arreglos. | ✓ | | |
| 13 | sort (Object sender) | void | Ordena el objeto. | ✓ | | |
| 14 | sortByColumn (int column) | void | Ordena la columna. | ✓ | | |
| 15 | sortByColumn (int column, boolean ascending) | void | Ordena la columna en forma ascendente. | ✓ | | |
| 16 | swap (int i, int j) | void | Pasa todas las demandas a estas filas a través del arreglo de "indexes". | ✓ | | |
| 17 | tableChanged (TableModelEvent e) | void | Es usada para notificar la tabla model cuando tiene cambios. | ✓ | | |

**Observación:**

## 12.22. CLASE COMTablMap

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMTablMap** | Tipo:<br>**Pública** | Hereda de:<br>**AbstractTableModel** |
| Descripción: *La clase* COMTablMap *es usada por la TablaSort, es padre de Tabla Sort.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | model | TableModel | Variable de tipo tabla model. | | | ✓ |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMTablMap** | Tipo:<br>**Pública** | Hereda de:<br>**AbstractTableModel** |
| Descripción: *La clase* COMTablMap *es usada por la TablaSort, es padre de Tabla Sort.* | | | |

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | getColumnClass (int aColumn) | Class | Obtiene la columna. | ✓ | | |
| 2 | getColumnCount () | Integer | Obtiene el número total de columnas. | ✓ | | |
| 3 | getColumnName (int aColumn) | String | Obtiene el nombre de la columna. | ✓ | | |
| 4 | getModel () | TableModel | Obtiene el model. | ✓ | | |
| 5 | getRowCount () | Integer | Obtiene el número total de filas. | ✓ | | |
| 6 | getValueAt (int aRow, int aColumn) | Object | Obtiene el valor. | ✓ | | |
| 7 | isCellEditable (int row, int column) | Boolean | Método que valida si la fila de esa columna puede ser o no editable. | ✓ | | |
| 8 | setModel (TableModel model) | Integer | Coloca el arreglo de índices en el model. | ✓ | | |
| 9 | setValueAt (Object aValue, int aRow, int aColumn) | void | Coloca el valor. | ✓ | | |
| 10 | tableChanged (TableModelEvent e)<br>sort (Object sender) | void | Es usada para notificar la tabla model cuando tiene cambios. | ✓ | | |

**Observación:**

## 12.23. CLASE COMVistPrev

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 2 | |
|---|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 | |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMVistPrev** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** | |
| Descripción: *La clase* COMVistPrev *muestra la pantalla preliminar de los Reportes.* | | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | arrBtns | JButton[] | Variable de tipo tabla model. | | ✓ | |
| 2 | bean | JRResultViewer | Variable de tipo JRResultViewer | | ✓ | |
| 3 | cmbExport | JcomboBox() | Combo de Export. | | ✓ | |
| 4 | cmbZoom | JcomboBox() | Combo de Zoom. | | ✓ | |
| 5 | codiModu | short | Variable de tipo short. | | ✓ | |
| 6 | codiUser | Integer | Variable de tipo entero. | | | |
| 7 | condWhere | String | Variable de tipo String. | | | |
| 8 | jREngine1 | jet.bean | Variable de tipo jet.bean.JREngine. | | | |
| 9 | lblPagnTotal | JLabel() | Caja de texto del total de páginas. | | ✓ | |
| 10 | nombFramPadr | String | Variable de tipo String. | | ✓ | |
| 11 | path | String | Variable de tipo String. | | ✓ | |
| 12 | prmt | String | Variable de tipo String. | | ✓ | |
| 13 | rpte | String | Variable de tipo String. | | ✓ | |
| 14 | tlbInferior | JToolBar() | Barra de herramientas. | | ✓ | |
| 15 | tlbSuperior | JToolBar() | Barra de herramientas. | | ✓ | |
| 16 | txtPagina | JtextField() | Caja de texto del número de página. | | ✓ | |
| 17 | ultmZoom | Integer | Variable de tipo entero. | | ✓ | |

**Observación:**

BIBLIOTECA CAMPUS PEÑAS ESPOL

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMVistPrev** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMVistPrev *muestra la pantalla preliminar de los Reportes.* | | | |

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMVistPrev (int codiUser, short codiModu, String nomb, String rpte) | | Construye la pantalla preliminar de los Reportes. | ✓ | | |
| 2 | COMVistPrev (int codiUser, short codiModu, String nomb, String rpte, String condWhere) | | Construye la pantalla preliminar de los Reportes. | ✓ | | |
| 3 | COMVistPrev (int codiUser, short codiModu, String nomb, String rpte, String condWhere, String prmt) | | Construye la pantalla preliminar de los Reportes. | ✓ | | |
| 4 | actionPerformed (java.awt.event.ActionEvent e) | void | Implementación de actionListener. | ✓ | | |
| 5 | clickOnField (Record rec) | void | Realiza el click en el campo deseado. | ✓ | | |
| 6 | getCodiModu () | short | Obtiene el código del módulo. | ✓ | | |
| 7 | getCodiUser () | Integer | Obtiene el código del usuario. | ✓ | | |
| 8 | setCodiModu (short codiModu) | void | Coloca el código del módulo. | ✓ | | |
| 9 | setCodiUser (int codiUser) | void | Coloca el código del usuario. | ✓ | | |
| 10 | saveExportAs (String descFile, String tipoFile) | String | Para generar archivo de exportación | | | ✓ |
| 11 | actlPagn () | void | Actualiza información de la página. | | ✓ | |
| 12 | agreEven () | void | Manejo de eventos. | | ✓ | |
| 13 | ejecRpte () | void | Ejecuta reporte, se crea archivo *.rst | | ✓ | |
| 14 | exportar (int cualExport) | void | Permite exportar el reporte al formato especificado. | | ✓ | |
| 15 | inicComp () | void | Este método es llamado por el constructor para inicializar la forma. | | ✓ | |
| 16 | salida () | void | Libera memoria al cerrar el JInternalFrame | | ✓ | |
| 17 | zoom (int cualZoom) | void | Aplica el nuevo zoom. | | ✓ | |

**Observación:**

## 12.24. CLASE COMAcercaDe

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 2 | |
|---|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 | |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMAcercaDe** | Tipo:<br>**Final Pública** | Hereda de:<br>**JInternalFrame** | |
| Descripción: *La clase* COMAcercaDe *genera el formulario de información del sistema, que va a contener el nombre de los Integrantes y Derechos del Autor.* | | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | LblEspol | JLabel() | Etiqueta de Espol. | | ✓ | |
| 2 | LblEspol1 | JLabel() | Etiqueta de Espol1. | | ✓ | |
| 3 | LblEspol2 | JLabel() | Etiqueta de Espol2. | | ✓ | |
| 4 | LblEspol3 | JLabel() | Etiqueta de Espol3. | | ✓ | |
| 5 | btn_Salir | JButton() | Botón de Salir. | | ✓ | |
| 6 | lbl_Mensaje | JLabel() | Etiqueta de Mensaje. | | ✓ | |
| 7 | lbl_Mensaje1 | JLabel() | Etiqueta de Mensaje1. | | ✓ | |
| 8 | lbl_Mensaje3 | JLabel() | Etiqueta de Mensaje2. | | ✓ | |
| 9 | lbl_Mensaje4 | JLabel() | Etiqueta de Mensaje3. | | ✓ | |
| 10 | pnl_Compañeras | JPanel() | Panel agrupador de controles. | | ✓ | |
| 11 | pnl_DatosCtas | JPanel() | Panel agrupador de controles. | | ✓ | |
| 12 | pnl_Espol | JPanel() | Panel agrupador de controles. | | ✓ | |
| 13 | pnl_Ventana | JPanel() | Panel agrupador de controles. | | ✓ | |
| 14 | pnl_VentanaI | JPanel() | Panel agrupador de controles. | | ✓ | |
| 15 | spr_borde | JSeparator() | Separador del Borde. | | ✓ | |
| 16 | txa_Descripcion | JTextArea() | Text área de Descripción. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMAcercaDe** | Tipo:<br>**Final Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMAcercaDe *genera el formulario de información del sistema, que va a contener el nombre de los Integrantes y Derechos del Autor.* | | | |

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMAcercaDe () | | Construye el formulario de información del Sistema. | ✓ | | |
| 2 | CentrarPantalla () | void | Método que sirve para centrar la pantalla. | ✓ | | |
| 3 | | | | | | |

**Observación:**

## 12.25. CLASE COMJTextFieldFilter

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMJTextFieldFilter** | Tipo:<br>**Pública** | Hereda de:<br>**PlainDocument** |
| Descripción: *La clase* COMJTextFieldFilter *es una clase propia de Java que ayuda a generar el formato numérico.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | ALL | String | Variable de tipo String. | ✓ | | |
| 2 | ALPHA | String | Variable de tipo String. | ✓ | | |
| 3 | ALPHA_ESPC | String | Variable de tipo String. | ✓ | | |
| 4 | ALPHA_NUMERIC | String | Variable de tipo String. | ✓ | | |
| 5 | ALPHA_NUMERIC_ESPC | String | Variable de tipo String. | ✓ | | |
| 6 | DIRE | String | Variable de tipo String. | ✓ | | |
| 7 | EMAIL | String | Variable de tipo String. | ✓ | | |
| 8 | FLOAT | String | Variable de tipo String. | ✓ | | |
| 9 | IVA | String | Variable de tipo String. | ✓ | | |
| 10 | LOWERCASE | String | Variable de tipo String. | ✓ | | |
| 11 | NOMB | String | Variable de tipo String. | ✓ | | |
| 12 | NUMERIC | String | Variable de tipo String. | ✓ | | |
| 13 | TELF | String | Variable de tipo String. | ✓ | | |
| 14 | UPPERCASE | String | Variable de tipo String. | ✓ | | |
| 15 | acceptedChars | String | Variable de tipo String. | | | ✓ |
| 16 | negativeAccepted | Boolean | Variable de tipo lógico. | | | ✓ |
| 17 | limit | Integer | Variable de tipo entero. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Ùltima Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMJTextFieldFilter** | Tipo:<br>**Pública** | Hereda de:<br>**PlainDocument** |
| Descripción: *La clase* COMJTextFieldFilter *es una clase propia de Java que ayuda a generar el formato numérico.* | | | |

## DESCRIPCIÓN DE CLASE

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMJTextFieldFilter (String acceptedchars, int limit) | | Constructor que genera el formato numérico. | ✓ | | |
| 2 | COMJTextFieldFilter (int limit) | | Constructor que genera el formato numérico. | ✓ | | |
| 3 | insertString (int offset, String str, AttributeSet attr) | void | Inserta el String. | ✓ | | |
| 4 | setNegativeAccepted (boolean negativeaccepted) | void | Acepta el número negativo | ✓ | | |

**Observación:**

## 12.26. CLASE COMMantLineaxProv

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMMantLineaxProv** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |

Descripción: *La clase* COMMantLineaxProv *permite el mantenimiento de las líneas por proveedor.*

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | btnBorrar | JButton() | Botón de Borrar. | ✓ | | |
| 2 | btnGuardar | JButton() | Botón de Guardar. | ✓ | | |
| 3 | btnSalir | JButton() | Botón de Salir. | ✓ | | |
| 4 | cmbLinea | JComboBox() | Combo de la línea por proveedor. | ✓ | | |
| 5 | cmbLineaGen | JComboBox() | Combo de todas las líneas existentes. | ✓ | | |
| 6 | cmbProv | JComboBox() | Combo del proveedor. | ✓ | | |
| 7 | p | JPanel() | Panel agrupador de controles. | | ✓ | |
| 8 | chkContado | JCheckBox() | Check box de contado. | | ✓ | |
| 9 | chkCredito | JCheckBox() | Check box de crédito. | | ✓ | |
| 10 | lblInteres | JLabel() | Etiqueta de interés. | | ✓ | |
| 11 | lblLinea | JLabel() | Etiqueta de la línea. | | ✓ | |
| 12 | lblLineaGen | JLabel() | Etiqueta de todas las lineas existentes. | | ✓ | |
| 13 | lblProv | JLabel() | Etiqueta del proveedor. | | ✓ | |
| 14 | lblTmaximo | JLabel() | Etiqueta del tiempo máximo. | | ✓ | |
| 15 | txtInteres | JTextField () | Caja de texto del interés. | | ✓ | |
| 16 | txtTmaximo | JTextField () | Caja de texto del tiempo máximo. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Usuario** | Nombre de Clase:<br>**COMMantLineaxProv** | Tipo:<br>**Pública** | Hereda de:<br>**JInternalFrame** |
| Descripción: *La clase* COMMantLineaxProv *permite el mantenimiento de las líneas por proveedor.* | | | |

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCIÓN | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | COMMantLineaxProv () | | Construye el formulario de mantenimiento de las líneas por proveedor. | ✓ | | |
| 2 | borrarLinea () | void | Método que borra la línea.. | ✓ | | |
| 3 | btnGuardarAccionado () | void | Guarda lo accionado. | ✓ | | |
| 4 | iniciarComponentes () | void | Permite activar todos los botones de la Definición del Proveedor. | ✓ | | |
| 5 | llenaLineas () | void | Método que llena la línea. | ✓ | | |
| 6 | llenaProveedores () | void | Método que llena el proveedor. | ✓ | | |
| 7 | llenarContado (int codP, int codL) | void | Método que elije si es a contado. | ✓ | | |
| 8 | llenarCredito (int codP, int codL, int inte, int tiemp) | void | Método que elije si es a crédito. | ✓ | | |
| 9 | llenarFormPago () | void | Método que llena su forma de pago. | ✓ | | |
| 10 | llenarLineasxProv () | void | Método que llena las líneas por proveedor. | ✓ | | |

**Observación:**

# Capítulo 15

## Capa de Negocios

## 12.27. CLASE COMMantDevolucion

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | | Pág. 1 / 1 | |
|---|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 | |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantDevolucion** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** | |

Descripción: *La clase* COMMantDevolucion *permite obtener las devoluciones pendientes, el nombre de un proveedor, guarda las cabeceras y los detalles de la devolución y obtiene el siguiente número de devolución de los datos de la capa de usuario de COMDevolucionMerc y COMLlegMercadería a la base de datos mediante las sentencias SQL.*
*La clase* COMMantDevolucion *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | getDevPendienes () | ResultSet | Obtiene devoluciones pendientes. | ✓ | | |
| 2 | getNmeroCabecera () | Integer | Obtiene el número de la cabecera. | ✓ | | |
| 3 | getnombProv (int codProv) | String | Obtiene el nombre del proveedor. | ✓ | | |
| 4 | getsiguiente () | Integer | Obtiene el siguiente número de cabecera. | ✓ | | |
| 5 | setCabeceraDevolucion (int numOrden, String tipo, int codProv, String fecha, int codEmp, String est) | void | Coloca el número de la cabecera. | ✓ | | |
| 6 | setDetalleDevolucion (int nmero, int codProd, int cant, String obs, String stdo) | void | Obtiene el detalle de la cabecera. | ✓ | | |

ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

**Observación:**

## 12.28. CLASE COMMantDevolucionMerc

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1 | |
|---|---|---|---|
| | | Fecha/Diseño:<br>01/08/2003 | Fecha/Última Actualización:<br>20/12/2003 |
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantDevolucionMerc** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantDevolucionMerc *consulta proveedores, las líneas por proveedor, descripción de la línea y la cabecera de la devolución de los datos de la capa de usuario COMDevolucionMerc a la base de datos mediante las sentencias SQL.*
*La clase* COMMantDevolucionMerc *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | getLinea (int codLinea) | ResultSet | Obtiene la línea del proveedor. | ✓ | | |
| 2 | getNumDevolucion () | Integer | Obtiene el número de devolución. | ✓ | | |
| 3 | getNumProveedores () | Integer | Obtiene el número del proveedor. | ✓ | | |
| 4 | getProveedores () | ResultSet | Obtiene la consulta del proveedor. | ✓ | | |
| 5 | lineasporProveedor (int codProv) | ResultSet | Obtiene la consulta de las líneas por proveedor. | ✓ | | |
| 6 | setCabDevolucion (int codProv, String Fecha, int codEmp, String stdo) | void | Coloca la cabecera de la devolución. | ✓ | | |

**Observación:**

## 12.29. CLASE COMMantEmiPagoOrdCom

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantEmiPagoOrdCom** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantEmiPagoOrdCom *guarda el número y la forma de pago en la orden de compra, guarda el valor que se va a pagar, consulta el precio, la línea y la unidad de medida, guarda la unidad de medida y la línea en la orden de compra de los datos de la capa de usuario* COMEmiPagoOrdCom *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantEmiPagoOrdCom *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | consultaValDetalle (int codProd, int ctdad, int numOrd) | void | Consulta la orden de compra. | ✓ | | |
| 2 | main (String[] arg) | void | Main Principal | ✓ | | |
| 3 | setSubTotalOC (int sbtotal, int numOrd, int codProd) | void | Coloca el subtotal de la orden de compra. | ✓ | | |
| 4 | setValForPagoContado (String fpagos, String fchpripago, int codprov) | void | Coloca el valor de la forma de pago cuando es a contado. | ✓ | | |
| 5 | setValForPagoCredito (String fpagos, int npagos, String fchpripago, int codprov) | void | Coloca el valor de la forma de pago cuando es a crédito. | ✓ | | |

**Observación:**

## 12.30. CLASE COMMantLineaxProvSQL

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantLineaxProvSQL** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantLineaxProvSQL *consulta el tiempo de crédito por proveedor, característica de la forma de pago, consulta de proveedor generalizado, consulta la línea y el proveedor, guarda los detalles del mantenimiento de las líneas por proveedor, elimina una línea de los datos de la capa de usuario* COMMantLineaxProv *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantLineaxProvSQL *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | eliminarLinea (int codProv, int codLin) | void | Elimina una línea. | ✓ | | |
| 2 | getCarCredito (int codProv) | String | Obtiene el crédito por proveedor. | ✓ | | |
| 3 | getLinea (int codLinea) | String | Obtiene la línea. | ✓ | | |
| 4 | getLineas () | ResultSet | Obtiene todas las líneas existentes. | ✓ | | |
| 5 | getProveedores () | ResultSet | Obtiene los proveedores. | ✓ | | |
| 6 | getTiempCred (int codProv) | ResultSet | Obtiene el tiempo de crédito. | ✓ | | |
| 7 | lineaxProveedor (int codProv) | ResultSet | Consulta las líneas por proveedor. | ✓ | | |
| 8 | setLinesconContado (int codP, int codL, String cred, int inte, int tiemp) | Boolean | Coloca las líneas cuando es contado. | ✓ | | |
| 9 | setLinesconCredito (int codP, int codL, String cred, int inte, int tiemp) | Boolean | Coloca las líneas cuando es crédito. | ✓ | | |

**Observación:**

# 12.31. CLASE COMMantLlegDevMercaderia

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantLlegDevMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantLlegDevMercaderia *consulta las devoluciones pendientes, los estados de las devoluciones, las cantidades de los productos a devolver, actualiza las cabeceras y detalles de las devoluciones, los proveedores que me tienen que devolver de los datos de la capa de usuario* COMLlegDevMercadería *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantLlegDevMercaderia *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | actCabDevMercaderia (int numOrden) | void | Actualiza la cabecera devolución. | ✓ | | |
| 2 | actCabDevMercaderiaPend (int numOrden) | void | Actualiza la cabecera de devolución pendiente. | ✓ | | |
| 3 | actCabDevolucion (int numCab) | void | Actualiza la cabecera devolución. | ✓ | | |
| 4 | actCabDevolucionPend (int numCab) | void | Actualiza la cabecera de devolución pendiente. | ✓ | | |
| 5 | actCabOrden (int numCab) | void | Actualiza cabecera orden compra. | ✓ | | |
| 6 | actDetDevMercaderia (int numOrden) | void | Actualiza detalle devolución. | ✓ | | |
| 7 | actProDevTotal (int numOrd, int codProd) | void | Actualiza el total producto devuelto. | ✓ | | |
| 8 | actProdDetDevolucion (int numCab, int codProd) | void | Actualiza el detalle producto devuelto. | ✓ | | |
| 9 | actProdPendDetDevolucion (int numCab, int codProd, int cantPend) | void | Actualiza el detalle producto pendiente devuelto. | ✓ | | |
| 10 | consCantProdDev (int numOrden) | Integer | Consulta cantidad producto devuelto. | ✓ | | |
| 11 | consDevPendientes () | Integer | Consulta devoluciones pendientes. | ✓ | | |
| 12 | consEstDevolucion (int numDev) | String | Consulta estado de devolución. | ✓ | | |
| 13 | getDetDevoluciones (int numOrd) | Object[][] | Obtiene detalle devolución. | ✓ | | |
| 14 | getDevPendientes_0 (int numOrden) | Object[][] | Obtiene detalle devolución pendiente | ✓ | | |
| 15 | getDevPendientes_1 (int numOrd) | Object[]] | Obtiene detalle devolución pendiente | | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantLlegDevMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantLlegDevMercaderia *consulta las devoluciones pendientes, los estados de las devoluciones, las cantidades de los productos a devolver, actualiza las cabeceras y detalles de las devoluciones, los proveedores que me tienen que devolver de los datos de la capa de usuario* COMLlegDevMercadería *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantLlegDevMercaderia *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 16 | getNumCabDevsegunOrden (int numOrd) | Integer | Obtiene el número de devolución según órden. | ✓ | | |
| 17 | getProvDev () | ResultSet | Obtiene el código del proveedor. | ✓ | | |
| 18 | getnombProv (int codProv) | String | Obtiene el nombre del proveedor. | ✓ | | |
| 19 | itemsDetDevoluciones (int numOrd) | Integer | Detalle de items a devolver. | ✓ | | |
| 20 | main (String[] arg) | void | Main Principal. | ✓ | | |
| 21 | numItemsgetDevPendientes_0 (int numOrden) | Integer | Número de ítems pendientes por devolver. | ✓ | | |
| 22 | setCantPendDev (int can, int numOrd) | void | Coloca las cantidades pendientes. | ✓ | | |
| 23 | valNumOrden (int numOrd) | Boolean | Coloca el estado de la orden de compra. | ✓ | | |

**Observación:**

## 12.32. CLASE COMMantLlegMercaderia

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantLlegMercaderia** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantLlegMercaderia *guarda el detalle de los productos recibidos, actualiza las cabeceras como pendientes o recibidos de los datos de la capa de usuario* COMLlegMercadería *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantLlegMercaderia *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | getOrdEnviadas () | Integer | Obtiene la orden de compra enviada. | ✓ | | |
| 2 | setMercCompleta (int numOrd) | void | Coloca la mercadería completa. | ✓ | | |
| 3 | setProductoRecivido (int numOrd) | void | Coloca el producto recibido. | ✓ | | |

**Observación:**

## 12.33. CLASE COMMantOrdenCompra

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantOrdenCompra** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantOrdenCompra *obtiene la cabecera de la orden de compra, actualiza la cabecera y detalle de la orden de compra como pendiente, consulta los proveedores aprobados, el precio de los productos, los productos por proveedor de los datos de la capa de usuario* COMEmiPagoOrdCom y COMLlegMercaderia *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantOrdenCompra *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | itemsporProv | Integer | Variable de tipo entero. | ✓ | | |
| 2 | NumSigOC | Integer | Variable de tipo entero. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantOrdenCompra** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantOrdenCompra *obtiene la cabecera de la orden de compra, actualiza la cabecera y detalle de la orden de compra como pendiente, consulta los proveedores aprobados, el precio de los productos, los productos por proveedor de los datos de la capa de usuario* COMEmiPagoOrdCom y COMLlegMercaderia *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantOrdenCompra *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | actualizarCabOrdCompra (int codProv, int numOrd) | void | Actualiza la cabecera de la orden de compra. | ✓ | | |
| 2 | actualizarProductos (int numSol, int CodProd) | void | Actualiza los productos. | ✓ | | |
| 3 | getCabOrdenCompra (int numOrden) | Object[][] | Obtiene la cabecera orden compra. | ✓ | | |
| 4 | getCabOrdenCompraDevolucion (int numOrden) | Object[][] | Obtiene la cabcera orden compra por devolución. | ✓ | | |
| 5 | getNumOC () | Double | Obtiene número de orden compra. | ✓ | | |
| 6 | getNumOrdenCompra () | Double | Obtiene número de orden compra. | ✓ | | |
| 7 | getPrecio (int codProd) | Integer | Obtiene el precio del producto. | ✓ | | |
| 8 | getPrecioProd (int codProd) | void | Obtiene el precio del producto. | ✓ | | |
| 9 | getProvAprovados () | ResultSet | Obtiene proveedores aprobados. | ✓ | | |
| 10 | numItemsCabOrdenCompra (int numOrden) | Integer | Número de ítems de la cabecera de orden de compra. | ✓ | | |
| 11 | numItemsCabOrdenCompraDevouluci on (int numOrden) | Integer | Número de ítems de la cabecera de orden de compra por devolución. | ✓ | | |
| 12 | numProdporProv (int codProv) | void | Número producto por proveedor. | ✓ | | |
| 13 | prodporProveedor (int codProv) | Object[][] | Producto por proveedor. | ✓ | | |
| 14 | setCabOrdenCompra (int NumOC, String Fecha, int CodProv) | void | Coloca la cabecera de la orden de compra. | ✓ | | |
| 15 | setDetComProdEstPend (int numOrd, int codProd, int cant) | void | Coloca el detalle de la compra del producto pendiente. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantOrdenCompra** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantOrdenCompra *obtiene la cabecera de la orden de compra, actualiza la cabecera y detalle de la orden de compra como pendiente, consulta los proveedores aprobados, el precio de los productos, los productos por proveedor de los datos de la capa de usuario* COMEmiPagoOrdCom y COMLlegMercaderia *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantOrdenCompra *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 16 | setDetalleOrdenCompra (int NumOC, int NumCS, int CodProd, String Desc, int Lin, int Cant) | void | Coloca el detalle de la orden de compra. | ✓ | | |
| 17 | setNumFact (int numOrd, int numFact) | void | Coloca el número de factura. | ✓ | | |
| 18 | setNumOC (int Num) | void | Coloca el número de orden compra | ✓ | | |
| 19 | setOrdComEstPend (int numOrd) | void | Coloca orden compra pendiente. | ✓ | | |
| 20 | setnumProdporProv (int numProd) | void | Coloca el número producto por proveedor. | ✓ | | |

**Observación:**

## 12.34. CLASE COMMantProdExt

| SCI – COM<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantProdExt** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantProdExt *consulta los productos por línea y los pendientes, deshabilita los productos enviados, guarda los recibidos de los datos de la capa de usuario* COMConProd, COMSolRep, COMEmiPagoOrdCom y COMLlegMercaderia *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantProdExt *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | Data | ResultSet | Variable de tipo ResulSet. | | ✓ | |
| 2 | NumItems | Integer | Variable de tipo entero. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantProdExt** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantProdExt *consulta los productos por línea y los pendientes, deshabilita los productos enviados, guarda los recibidos de los datos de la capa de usuario* COMConProd, COMSolRep, COMEmiPagoOrdCom y COMLlegMercaderia *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantProdExt *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | Consulta_Productos_Linea (String Letra) | ResultSet | Consulta de productos por línea. | ✓ | | |
| 2 | Obtener_NumItems (String Letra) | void | Obtiene el número de ítems. | ✓ | | |
| 3 | actSolComPendiente (int numSol) | void | Actualiza la solicitud pendiente. | ✓ | | |
| 4 | actSolComRecibida (int numSol) | void | Actualiza la solicitud recibida. | ✓ | | |
| 5 | desabilitarProducto (int CodProd) | void | Deshabilita el producto. | ✓ | | |
| 6 | getDetPend (int numOrd) | Integer | Obtiene el detalle pendiente. | ✓ | | |
| 7 | getNumItems () | Integer | Obtiene el número de ítems. | ✓ | | |
| 8 | getProdPend (int numOrd) | Integer | Obtiene el producto pendiente. | ✓ | | |
| 9 | getUmedida (int uniCod) | String | Obtiene la unidad de medida. | ✓ | | |
| 10 | getValIvaDes (int codProd) | Double[][] | Obtiene el valor, iva y descuento. | ✓ | | |
| 11 | setNumItems (int Num) | void | Coloca el número de ítems. | ✓ | | |
| 12 | setProductoRecivido (int numOrd, int codProd) | void | Coloca el producto recibido. | ✓ | | |

**Observación:**

ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

## 12.35. CLASE COMMantProveedores

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 1<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantProveedores** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantProveedores *consulta las características de crédito por proveedores y el tiempo máximo de los datos de la capa de usuario* COMEmiPagoOrdCom y COMDefPago *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantProveedores *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**MÉTODOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | caractCredProv (int CodProv, int Linea) | String[] | Características de crédito por proveedor. | ✓ | | |
| 2 | getLineaProv (int CodProv) | Integer | Obtiene la línea por proveedor. | ✓ | | |
| 3 | tiempoCredito (int codProv) | Integer | Obtiene el tiempo máximo de crédito. | ✓ | | |

**Observación:**

## 12.36. CLASE COMMantSolPedido

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantSolPedido** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantSolPedido *consulta el número de la solicitud siguiente, guarda la cabecera y detalle de la solicitud de compra por pedido y por reposición, limpia el detalle de la solicitud, anula, aprueba, actualiza el detalle de la solicitud pendiente, consulta las líneas solicitadas, los productos por línea de los datos de la capa de usuario* COMAprovSolCom, COMSolPedido y COMSolRep *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantSolPedido *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | Detalle | Object[][] | Variable de tipo object. | ✓ | | |
| 2 | NumItemsDetSolPend | Integer | Variable de tipo entero. | | ✓ | |
| 3 | NumProdLinea | Integer | Variable de tipo entero. | | ✓ | |
| 4 | NumSigSol | Integer | Variable de tipo entero. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantSolPedido** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantSolPedido *consulta el número de la solicitud siguiente, guarda la cabecera y detalle de la solicitud de compra por pedido y por reposición, limpia el detalle de la solicitud, anula, aprueba, actualiza el detalle de la solicitud pendiente, consulta las líneas solicitadas, los productos por línea de los datos de la capa de usuario* COMAprovSolCom, COMSolPedido y COMSolRep *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantSolPedido *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | Numero_Solicitud_Compra () | void | Obtiene el número de solicitud. | ✓ | | |
| 2 | actDetalleSolicitud (int codProd, int numSoli) | void | Actualiza el detalle de la solicitud. | ✓ | | |
| 3 | anularSolCompra (int NumSol) | void | Anula la solicitud de compra. | ✓ | | |
| 4 | anularSolPedido (int numSol) | void | Anula la solicitud por pedido. | ✓ | | |
| 5 | aprobarSolCompra (int NumSol) | void | Aprueba la solicitud de compra. | ✓ | | |
| 6 | busDetSolComPendPedido (int Num) | Object[][] | Busca detalle solicitud pendiente por pedido. | ✓ | | |
| 7 | busDetSolPendPedido (int Num) | Object[][] | Busca detalle solicitud pendiente. | ✓ | | |
| 8 | buscarSolComNoAprovPedido () | ResultSet | Busca la solicitud aprobada. | ✓ | | |
| 9 | consNumItemsSolComPendPedido (int NumSol) | void | Consulta el número de ítems de solicitud de compra pendiente | ✓ | | |
| 10 | consultarLinea (int codLinea) | String | Consulta las líneas. | ✓ | | |
| 11 | consultarUmedida (int codUmedida) | String | Consulta la unidad de medida. | ✓ | | |
| 12 | desabilitarProducto (int CodProd) | void | Deshabilitael producto. | ✓ | | |
| 13 | getLineasSolicitadas () | ResultSet | Obtiene las líneas solicitadas. | ✓ | | |
| 14 | getNumItemsDetConsulta (int NumSol) | void | Obtiene el número de ítems del detalle. | ✓ | | |
| 15 | getNumItemsDetSolPend () | Integer | Obtiene el número de ítems del detalle de solicitud pendiente. | ✓ | | |
| 16 | getNumItemsxLineaSolicitada (int Linea) | void | Obtiene el número de ítems por línea solicitada. | | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 3 / 3<br>Fecha/Diseño: 01/08/2003 | Fecha/Ùltima Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantSolPedido** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantSolPedido *consulta el número de la solicitud siguiente, guarda la cabecera y detalle de la solicitud de compra por pedido y por reposición, limpia el detalle de la solicitud, anula, aprueba, actualiza el detalle de la solicitud pendiente, consulta las líneas solicitadas, los productos por línea de los datos de la capa de usuario* COMAprovSolCom, COMSolPedido y COMSolRep *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantSolPedido *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 17 | getNumProdLinea () | Integer | Obtiene número productos línea. | ✓ | | |
| 18 | getNumSol () | Integer | Obtiene el número de solicitud. | | | |
| 19 | getProductosxLinea (int Linea) | Object[][] | Obtiene los productos por línea. | ✓ | | |
| 20 | limpiarDetSolPedido (int NumSol) | void | Limpia el detalle solicitud pedido. | ✓ | | |
| 21 | setCabSolicitudPedido (int NumSol, String Fecha, String Tipo, String Obs) | void | Coloca la cabecera de solicitud por pedido. | ✓ | | |
| 22 | setCabSolicitudReposicion (int NumSol, String Fecha, String Tipo, int codEmp, String Bodega) | void | Coloca la cabecera de solicitud por reposición. | ✓ | | |
| 23 | setDetSolicitudPedido (int codProd, int numSol, int cant, String est) | void | Coloca el detalle de solicitud por pedido. | | | |
| 24 | setDetalleSolicitud (int CodProd, int NumSolCom, int Cant) | void | Coloca el detalle de solicitud. | | | |
| 25 | setDetalleSolicitudRep (int CodProd, int NumSolCom, int Cant) | void | Coloca el detalle de solicitud por reposición. | | | |
| 26 | setNumItemsDetSolPend (int NumItems) | void | Coloca el número de ítems de la solicitud pendiente. | | | |
| 27 | setNumProdLinea (int Num) | void | Coloca el número productos línea. | | | |
| 28 | setNumSol (int Num) | void | Coloca el número de solicitud. | | | |

**Observación:**

## 12.37. CLASE COMMantSolRep

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantSolRep** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantSolRep *consulta las bodegas que hicieron pedido y los productos por bodegas de los datos de la capa de usuario* COMSolRep *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantSolRep *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | NumProdBod | Integer | Características de crédito por proveedor. | | ✓ | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Negocios** | Nombre de Clase:<br>**COMMantSolRep** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |

Descripción: *La clase* COMMantSolRep *consulta las bodegas que hicieron pedido y los productos por bodegas de los datos de la capa de usuario* COMSolRep *a la base de datos mediante las sentencias SQL.*
*La clase* COMMantSolRep *accesa a la base de datos SCI mediante la conexión con la capa de datos* COMEjecutarSQL.

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | bodegasPedidos () | ResulSet | Consulta de Pedidos por Bodegas. | ✓ | | |
| 2 | getNumItemsBod () | Integer | Obtiene número de ítems por bodega | ✓ | | |
| 3 | getNumItemsProdporBodega (String Bod) | void | Obtiene número de ítems por producto en cada bodega. | ✓ | | |
| 4 | getProdporBodega (String Bod) | Object[] | Obtiene el producto por bodega. | ✓ | | |
| 5 | setNumItemsBod (int NumItemBod) | Void | Coloca el número de ítems por bodega. | ✓ | | |

ESPOL BIBLIOTECA CAMPUS PEÑAS

**Observación:**

# Capítulo 12

## Capa de Datos

## 12.38. CLASE COMEjecutarSQL

| SCI – COM<br>Sistema de Compras<br>VERSION 1.0 | | Pág. 1 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Datos** | Nombre de Clase:<br>**COMEjecutarSQL** | Tipo:<br>**Pública** | Hereda de:<br>**cls_seg_conexion** |
| Descripción: *La clase* COMEjecutarSQL *ejecuta los selects, inserts, updates, deletes que realice mi capa de negocios.* | | | |

**DESCRIPCIÓN DE CLASE**

**ATRIBUTOS**

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | conexión | Connection | Variable de conexión. | ✓ | | |
| 2 | Rs | ResultSet | Variable tipo ResulSet. | ✓ | | |

**Observación:**

| **SCI – COM**<br>**Sistema de Compras**<br>VERSION 1.0 | | Pág. 2 / 2<br>Fecha/Diseño: 01/08/2003 | Fecha/Última Actualización: 20/12/2003 |
|---|---|---|---|
| Para:<br>**PROTCOM – ESPOL** | | Autores:<br>**Sandra Araujo Villacís**<br>**Ulbio Morales Labre** | |
| Capa:<br>**Datos** | Nombre de Clase:<br>**COMEjecutarSQL** | Tipo:<br>**Pública** | Hereda de:<br>**COMEjecutarSQL** |
| Descripción: *La clase* COMEjecutarSQL *ejecuta los selects, inserts, updates, deletes que realice mi capa de negocios.* | | | |

## DESCRIPCIÓN DE CLASE

### MÉTODOS

| No. | NOMBRE | TIPO | DESCRIPCION | PB | PV | PG |
|---|---|---|---|---|---|---|
| 1 | EjecutaDetalle (String sql) | ResulSet | Realiza la conección y ejecuta los querys | ✓ | | |
| 2 | EjecutaUpdate (String sql) | Boolean | Ejecuta los updates, inserts y deletes. | ✓ | | |
| 3 | Ejecutaint (String sql) | Integer | Retorna un entero realizado en un select. | ✓ | | |

ESPOL
BIBLIOTECA
CAMPUS
PEÑAS

**Observación:**

# Capítulo 3

## Seguridades del Sistema

# 13. SEGURIDADES DEL SISTEMA

## 13.1. NIVELES DE SEGURIDAD DE ACCESO Y USO

La entrada a SCI se realizará mediante el menú Mantenimiento, al momento de elegir esta opción <Iniciar Sesión> le pedirá la clave y el usuario con el que va a trabajar.

Además, una vez dentro del sistema tendrá acceso a determinadas opciones del Sistema, dependiendo de la función que vaya a cumplir cada usuario del Módulo.

## 13.2. SEGURIDADES DE DATOS

La persona encargada del Sistema se asegurará de respaldar la Base de Datos a diario.

### 13.2.1. RESPALDO DE LA BASE DE DATOS

| Item a Evaluar | Explicación |
|---|---|
| Período en que se realiza el respaldo | Todos los días |
| Períodos en que se borrarán los datos | No se borran |
| Persona que lo realiza | Encargado del Sistema |
| Tipo de respaldo | Toda la Base |
| Tipo de dispositivo | Unidad de ZIP |
| Ubicación de respaldo | Habitación segura a prueba de fuego |

### 13.2.2. RESTAURACIÓN

Este proceso se da cuando se ha perdido información y se desea recuperar información respaldada. El encargado de la Base de Datos es la única persona autorizada para realizar este proceso.