



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

SISTEMA INFORMÁTICO DE CONTABILIDAD.

Trabajo de Titulación para optar al
Título de Técnico Universitario en
INFORMÁTICA

Integrantes:

Valentina Contreras V.

Felipe González C.

Profesor Guía:

Gabriel Jara

Resumen

KEYWORDS: CONTABILIDAD – JAVA – MYSQL – JASPERREPORT.

Este trabajo de título tiene como objetivo presentar la implementación de un sistema de contabilidad para la empresa “Central Asesorías SPA”.

Esta aplicación tiene como fin facilitar y acortar los tiempos en el proceso de llevar la contabilidad a sus distintos clientes.

Este sistema contable está desarrollado en Java utilizando el IDE Netbeans 8.2, la base de datos utiliza el motor MySQL.

Este trabajo se divide en 3 capítulos, en donde el:

Capítulo 1: Se describe brevemente la organización o empresa, la situación actual de esta, los problemas detectados y finalmente una descripción del sistema propuesto.

Capítulo 2: Descripción del medio ambiente computacional, como el software y hardware utilizado para el desarrollo del sistema, como también los requisitos mínimos de hardware y software para que el sistema funcione de manera óptima. También se describen las tablas y sus respectivos campos utilizados en la base de datos.

Capítulo 3: Descripción de la arquitectura del sistema, se presentan diagramas y listados para dar la visión general del funcionamiento del sistema.

Índice

Resumen

Introducción.....	1
CAPÍTULO 1:	2
ASPECTOS RELEVANTES DEL DISEÑO LÓGICO.	2
1. Aspectos relevantes del diseño lógico.	3
1.1 Descripción de la organización.....	3
1.2 Descripción de la situación actual.....	4
1.3 Los problemas detectados.	6
1.4 Descripción del sistema propuesto.....	6
CAPÍTULO 2:	20
MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS.....	20
2. Medio ambiente computacional y descripción de archivos.....	21
2.1 Características del recurso computacional.	21
2.1.1 Configuración del Sistema.....	21
2.1.2 Software Utilizado.	22
2.2 Descripción de Archivos.	23
CAPÍTULO 3:	33
DESCRIPCIÓN DE PROGRAMAS.	33
3. Descripción de programas.....	34
3.1 Diagrama de Menús.....	34
3.2 Diagrama modular.....	34
3.3 Programas del Sistema.	35
3.4 Estructura general del sistema.....	36
3.4.1 Login.....	36
3.4.2 Menú Principal.....	37
3.4.3 Mantenedor de Clientes.....	38
3.4.4 Mantenedor de Cuentas.....	39
3.4.5 Gestión de Asientos.....	41
3.4.6 Generar Libro Diario.	43
3.4.7 Generar Libro Mayor.	44
3.4.8 Generar Balance.	46

3.4.9 Ingreso de Periodo..... 48

3.5 Otros elementos. 49

Conclusiones 50

Bibliografía..... 51

Anexo 52

Introducción

La contabilidad es una disciplina económica que ha sido útil para el hombre desde tiempos inmemoriales, la cual permite tener un orden claro y preciso de las actividades, recursos, gastos y dinero. Por ello, la utilización de ésta es vital para administrar de la mejor manera posible la vida financiera y económica de una empresa, del trabajo o hasta del emprendimiento familiar, para así, facilitar la toma de decisiones con precisión y con conocimiento previo de lo que podría suceder.

Es por esto, que el presente Trabajo de Título consiste en un sistema informático capaz de llevar la contabilidad de los clientes de la empresa “Central Asesorías SPA” de manera rápida, fácil y efectiva.

Este sistema contará con funcionalidades que harán llevar la contabilidad de manera que los datos financieros de los clientes se mantengan consistentes en el tiempo, además de entregar los correspondientes reportes a dichos clientes con información verídica que les permita saber su situación financiera.

CAPÍTULO 1:

ASPECTOS RELEVANTES DEL DISEÑO LÓGICO.

1. Aspectos relevantes del diseño lógico.

1.1 Descripción de la organización.

Esta es una empresa nueva que lleva 1 año en funcionamiento, nacida de la iniciativa de 2 docentes de la Universidad Central con 20 años de experiencia en el rubro de la contabilidad. Esta empresa se enfoca en satisfacer las necesidades de las pymes, como también el asesoramiento:

- Contable
 - Administración de Documentos Tributarios Electrónicos.
 - Contabilidad completa y simplificada.
 - Balance General.
 - Libros correspondientes.
 - Asesoría Contable.
- Tributario
 - Planificación tributaria.
 - Formulario 22, 29 y 50.
 - Declaraciones juradas.
 - Asesoría Tributaria.
- Jurídico
 - Asesoría Jurídica.
 - Juicios Civiles.
 - Juicios Penales.
 - Juicios Familia.
 - Juicios Laborales.
 - Herencias.
 - Empresas.

También cuenta con un servicio de factoring, para solucionar los problemas financieros de las pymes.

Cerca del 70% de los clientes de esta empresa se mueven dentro del rubro de los servicios, y la otra parte en el rubro del retail.

Dentro de las empresas a las cuales les presta servicios se pueden nombrar:

- Insoin (Soluciones de ingeniería para Cencosud).
- La Empanada Mia (Producción y venta de empanadas).
- Grupo Zaror (Empresa de Inversión).
- Sercotec (Asesoramiento a los centros de desarrollo de trabajo).

Objetivos de la empresa

Central Asesoría SPA es una empresa creada para ayudar a formar y desarrollar emprendimientos, asesorando de manera transparente y profesional, siempre buscando lo mejor para la empresa.

1.2 Descripción de la situación actual.

Esta empresa lleva los registros de cuenta en una planilla de Excel y se realizan los balances de manera manual provocando que se requiera de un tiempo considerable para realizar esta tarea.

Los registros de las diferentes empresas para la que trabajan se encuentran almacenados en un archivo Excel para cada empresa (ver Figura 1-1, Figura 1-2 y Figura 1-3).

Asiento	Fecha	Definición	Código	Desglose	Debe	Haber
1	30-ene-16	Activo +	1	CAJA	\$ 200.000,00	
1	30-ene-16	Pasivo +	2	MERCADERIAS	\$ 50.000,00	
1	30-ene-16	R+	3	Vehículos	\$ 900.000,00	
1	30-ene-16	R-	4	LXC	\$ 50.000,00	
1	30-ene-16	ACTIVO	5	CAPITAL		\$ 1.200.000,00
2	30-ene-16	Activo +	6	BANCO	\$ 120.000,00	
2	30-ene-16	Activo +	1	CAJA		\$ 120.000,00
2	30-ene-16	Pasivo +	2	MERCADERIAS	\$ 400.000,00	
2	30-ene-16	Activo +	7	IVA CF	\$ 76.000,00	
2	30-ene-16	Pasivo +	8	PROVEEDORES		\$ 476.000,00
3	12-feb-16	activo+	9	MUEBLES Y UTILES	\$ 25.000,00	
3	12-feb-16	pasivo	7	IVA CF	\$ 4.750,00	
3	12-feb-16	activo	1	CAJA		\$ 29.750,00
4	12-feb-16	Activo -	8	PROVEEDORES	\$ 476.000,00	
4	30-feb-16	Activo +	10	LXP		\$ 476.000,00
5	30-feb-16	Pasivo +	1	CAJA	\$ 42.840,00	
5	30-feb-16	R+	11	DXC	\$ 42.840,00	
5	30-feb-16	R-	12	CLIENTE	\$ 57.120,00	
5	30-feb-16	ACTIVO	13	VENTA		\$ 120.000,00

Figura 1-1. Libro diario otorgado por el cliente.

Autoguardado

PLANILLA EXCEL LIBROS CONTABLES - Vista protegida - Excel

Archivo

Inicio

Insertar

Diseño de página

Fórmulas

Datos

Revisar

Vista

Complementos

Equipo

¿Qué desea hacer?

D162

25000

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Codigo	Fecha	Desglose	Debe	Haber	Saldo							
2	2	30-ene-16	MERCADERIAS	\$50000,00		\$50000,00							
3	2	30-ene-16	MERCADERIAS	\$400000,00		\$450000,00							
4	2	30-feb-16	MERCADERIAS		\$100000,00	\$350000,00							
5	2	30-mar-16	MERCADERIAS		\$15000,00	\$335000,00							
6													
7			TOTAL	\$450000,00	\$115000,00	\$335000,00							
8													
9													
10													
11	Codigo	Fecha	Desglose	Debe	Haber	Saldo							
12	3	30-ene-16	Vehiculos	\$900000,00		\$900000,00							
13	3	30-abr-16	Vehiculos		\$900000,00	\$0,00							
14													
15			TOTAL	\$900000,00	\$900000,00	\$0,00							
16													
17													
18													
19	Codigo	Fecha	Desglose	Debe	Haber	Saldo							
20	4	30-ene-16	LXC	\$50000,00		\$50000,00							
21	4	30-abr-16	LXC		\$50000,00	\$0,00							
22	4	30-abr-16	LXC	\$390000,00		\$390000,00							
23	4	30-may-16	LXC		\$390000,00	\$0,00							
24													
25			TOTAL	\$440000,00	\$440000,00	\$0,00							

Figura 1-2. Libro mayor otorgado por el cliente.

PLANILLA EXCEL LIBROS CONTABLES - Vista protegida - Excel										
¿Qué desea hacer?										
I14										
	A	B	C	D	E	F	G	H	I	J
1										
2										
3	cuenta	debe	haber	deudor	acreedor	activo	pasivo	perdida	ganancia	
4	CAJA	\$ 892.840	\$ 854.810	\$ 38.030	\$ -	\$ 38.030				
5	BANCO	\$ 1.352.800	\$ 700	\$ 1.352.100	\$ -	\$ 1.352.100				
6	LXC	\$ 440.000	\$ 440.000	\$ -	\$ -	\$ -				
7	DXC	\$ 409.960	\$ 149.960	\$ 260.000	\$ -	\$ 260.000				
8	MERCADERIAS	\$ 450.000	\$ 115.000	\$ 335.000	\$ -	\$ 335.000				
9	MUEBLES Y UTILES	\$ 71.540	\$ -	\$ 71.540	\$ -	\$ 71.540				
10	CTA CTE SOCIO	\$ 21.420	\$ -	\$ 21.420	\$ -	\$ 21.420				
11	VEHICULOS	\$ 900.000	\$ 900.000	\$ -	\$ -	\$ -				
12	IVA CF	\$ 89.593	\$ -	\$ 89.593	\$ -	\$ 89.593				
13	PPM			\$ -	\$ -	\$ -				
14	IVA DF	\$ -	\$ 26.220	\$ -	\$ 26.220		\$ 26.220			
15	PROVEEDORES	\$ 476.000	\$ 496.552	\$ -	\$ 20.552		\$ 20.552			
16	LXP	\$ -	\$ 476.000	\$ -	\$ 476.000		\$ 476.000			
17	DXP	\$ -	\$ 51.611	\$ -	\$ 51.611		\$ 51.611			
18	CAPITAL		\$ 1.200.000	\$ -	\$ 1.200.000		\$ 1.200.000			
19	COSTO DE VENTA	\$ 115.000		\$ 115.000	\$ -			\$ 115.000		
20	GASTOS BANCARIOS	\$ 700		\$ 700	\$ -			\$ 700		
21	GASTOS	\$ 4.000		\$ 4.000	\$ -			\$ 4.000		
22	ARRIENDO	\$ 25.000		\$ 25.000	\$ -			\$ 25.000		
23	VENTA		\$ 138.000	\$ -	\$ 138.000				\$ 138.000	
24	UTILIDAD VTA ACTIVO FIJO		\$ 400.000	\$ -	\$ 400.000				\$ 400.000	
25	TOTALES	\$ 5.248.853	\$ 5.248.853	\$ 2.312.383	\$ 2.312.383	\$ 2.167.683	\$ 1.774.383	\$ 144.700	\$ 538.000	
26	UTILIDAD O PERDIDA						\$ 393.300	\$ 393.300		
27	TOTALES FINALES	\$ 5.248.853	\$ 5.248.853	\$ 2.312.383	\$ 2.312.383	\$ 2.167.683	\$ 2.167.683	\$ 538.000	\$ 538.000	

Figura 1-3. Balance otorgado por el cliente.

1.3 Los problemas detectados.

- Requerir de mucho tiempo para realizar los balances en tiempos de alta demanda, esto provoca que tenga que declinar a posibles clientes entrantes ya que no daría abasto debido al tiempo que toma realizar estas tareas.
- Al realizar los procedimientos de forma manual, incrementa la probabilidad de presentar descuadratura en la contabilización.
- A la hora de modificar un registro, tiene que rehacer los cálculos en los cuales dicho registro estaba involucrado, tomando una gran cantidad de tiempo en realizar esta tarea.
- Problemas a la hora de generar informes, ya que se tiene que elegir el formato adecuado con el cual se presentará el informe impreso, generando una gran carga de tiempo en realizar esta tarea.
- Generar los detalles del libro mayor de forma manual, esto incrementa el riesgo a cometer errores en los cálculos y también toma una gran cantidad de tiempo
- Gran cantidad de archivos de empresas a las que se les lleva la contabilidad, que, al no tener una base de datos o respaldo, se corre el riesgo de perder un archivo al ser manipulado y no poder recuperarlo.
- Al entregar reportes en archivos Excel a los clientes no se podría autenticar la validez de este si fue editado o no.

1.4 Descripción del sistema propuesto.

Objetivo General

Realizar un Sistema Informático contable para que la empresa Central Asesoría SPA pueda proveer un mejor servicio a sus clientes de manera rápida y eficiente para así poder satisfacer las necesidades de estos.

Objetivos específicos

- Facilitar el ingreso de detalles a los asientos para la posterior generación del Libro Diario, Libro Mayor y el balance financiero, para así optimizar el tiempo en realizarlos.
- Acortar el tiempo en que el usuario demora en entregar el trabajo finalizado al cliente.
- Proporcionar comodidad al usuario (Central Asesoría SPA) para realizar su trabajo.
- Mantener la integridad de la información de los clientes.

Diagrama de flujo administrativo

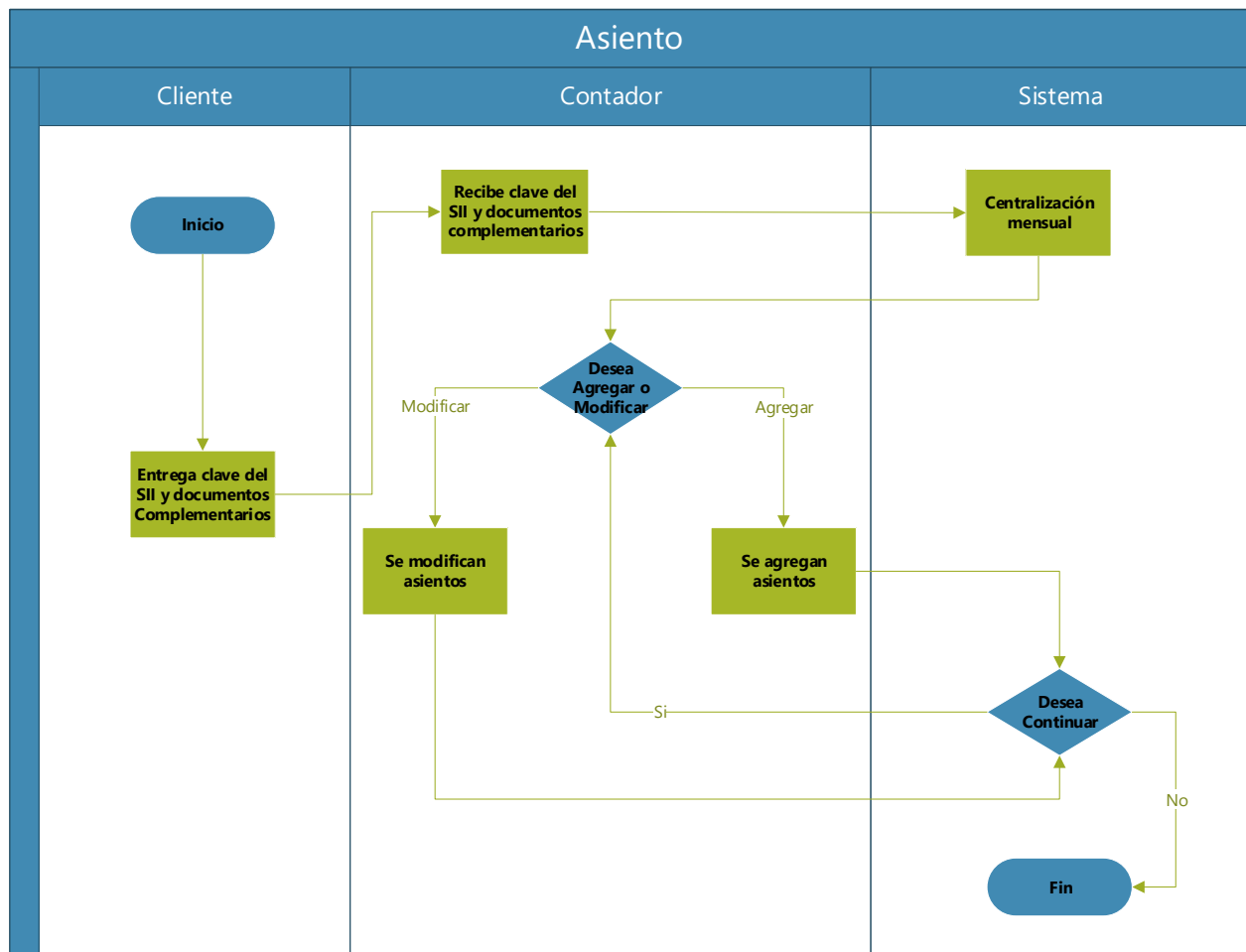


Figura 1-4. Diagrama de flujo administrativo del proceso de ingreso de asientos.

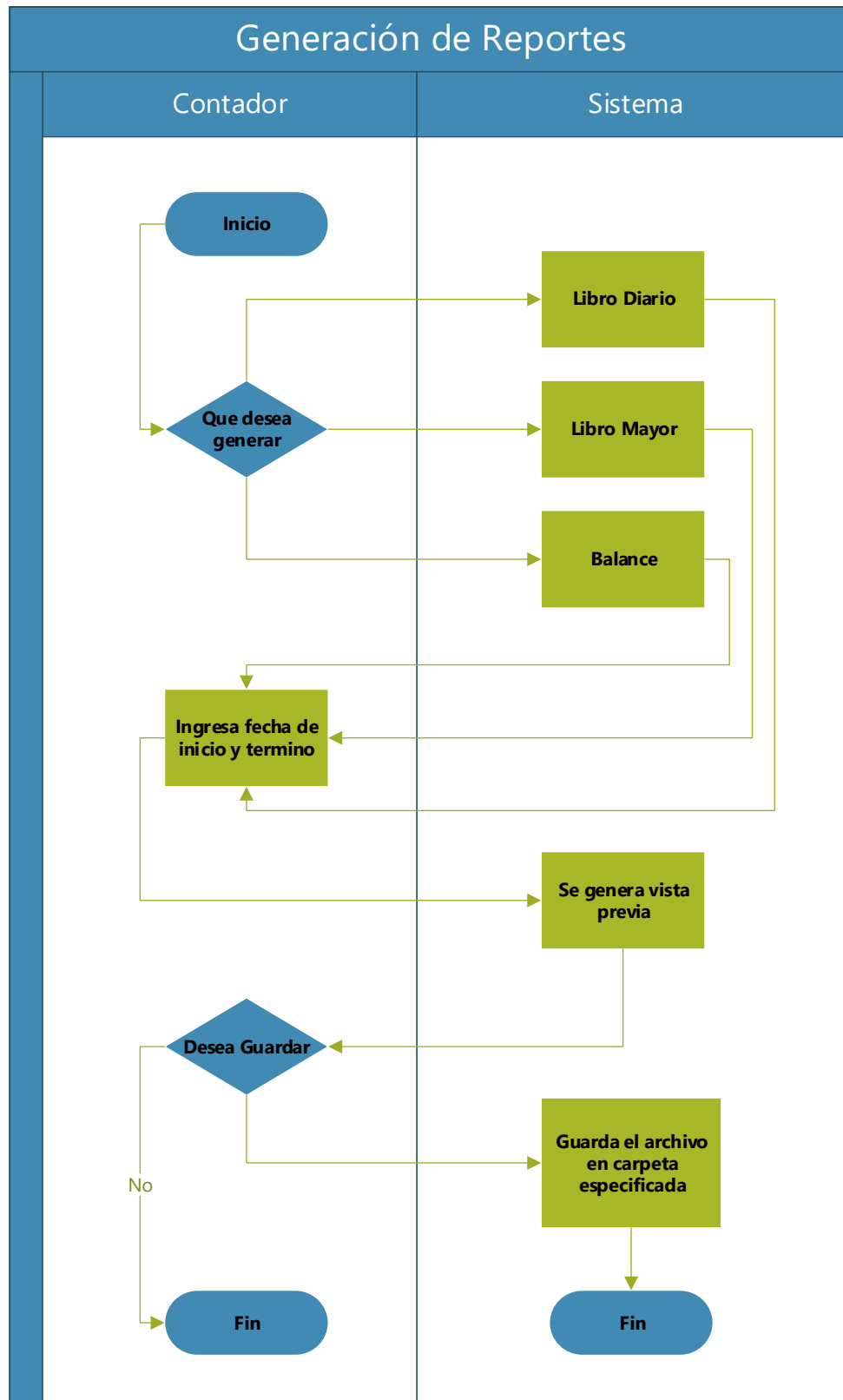


Figura 1-5. Diagrama de flujo administrativo del proceso de generación de reportes.

Funcionalidad del sistema

- Administrador de Clientes:
 - Agregar empresa: Registrará los datos de una empresa específica y de su representante legal.
 - Modificar empresa: Modificará los datos de una empresa específica y de su representante legal.
 - Eliminar empresa: Eliminará los datos de una empresa específica y de su representante legal.
 - Buscar empresa: Buscará una empresa en específico por su nombre, posteriormente se podrá seleccionar la empresa con la cual se va a trabajar, de esta manera se cargarán los datos de dicha empresa.

- Administrador de Cuentas:

Se cargan predeterminadamente una cantidad de cuentas que el común de las empresas podría utilizar, de manera que se encuentran en 2 grandes categorías: Activos y Pasivos (Cuentas Mayores), luego, a raíz de esto se irán agregando cuentas como subcategorías a estas 2 cuentas principales.

- Agregar Cuenta: Agregará una cuenta de acuerdo con su clasificación (Cuentas de Títulos, Sub Cuentas de Títulos, Cuentas Imputables, Sub Cuentas Imputables).
- Modificar Cuenta: Modificará una cuenta ya ingresada.
- Eliminar Cuenta: Eliminará una cuenta.
- Buscar Cuenta: Buscará una cuenta específica por nombre.

Obs: Por el momento, a petición de usuario, no se utilizará depreciación o depreciación acumulada en el sistema.

- Administrador de Asientos:

Se ingresa el tipo de asiento (Ingreso, Egreso o Traspaso), número de asiento y día del asiento, posteriormente se procesa a ingresar el detalle del comprobante contable (número

correlativo, tipo de documento, número de documento, glosa, debe, haber). Una vez finalizado el detalle de el o los comprobantes contables estos se agregan al asiento.

- Ingreso de Asiento: Registrará los datos de un determinado asiento, ingresando el detalle de el o los comprobantes contables para luego ser registrados en el asiento, el cual se ingresará posteriormente en el Libro Diario.
 - Agregar Comprobante Contable: Registra un comprobante contable, que a su vez es el detalle de un asiento específico.
 - Modificar Comprobante Contable: Modifica un comprobante contable.
 - Eliminar Comprobante Contable: Elimina un comprobante contable.
- Eliminar Asiento: Eliminará el asiento y todos los comprobantes asociados a ese asiento.
- Consulta Libro Diario: Imprime el Libro Diario según el periodo que se requiera (día, mes, año).
- Consulta Libro Mayor: Imprime el Libro Mayor según el periodo que se requiera (día, mes, año).
- Consulta Balance: Imprime el Balance según el periodo que se requiera (mensual o anual) y nivel (1, 2, 3 o 4 que corresponde a la cantidad de columnas, con sus respectivos detalles al debe y al haber).

Descripción de fórmulas y cálculos empleados

- Ingreso de Asiento:

D_i : Debe en línea de comprobante contable i .

H_i : Haber en línea de comprobante contable i .

Al finalizar el asiento:

Validación: $\sum D_{it} = \sum H_{it}$
- Libro Diario:

Para un Cliente en un Periodo t :

D_i : Debe en línea de asiento i .

H_i : Haber en línea de asiento i .

Al finalizar el registro de un asiento:

Validación: $\sum D_{it} = \sum H_{it}$

- Libro Mayor:

Para un Cliente en un Periodo t y una Cuenta c :

D_c = Debe en cuenta c .

H_c = Haber en cuenta c .

$$total_debe_cuenta_{ct} = \sum D_{ct}$$

$$total_haber_cuenta_{ct} = \sum H_{ct}$$

$$saldo_{ct} = total_debe_cuenta_{ct} - total_haber_cuenta_{ct}$$

$$total_mayor_t = \sum saldo_{ct} = 0$$

- Balance:

Para un Cliente en un Periodo t :

$Debe_c$ = Debe para una Cuenta c .

$Haber_c$ = Haber para una Cuenta c .

$Deudor_c$ = Deudor para una Cuenta c .

$Acreedor_c$ = Acreedor para una Cuenta c .

$Activo_c$ = Activo para una Cuenta c .

$Pasivo_c$ = Pasivo para una Cuenta c .

$$Debe_c = total_debe_cuenta_{ct}$$

$$Haber_c = total_haber_cuenta_{ct}$$

$$Deudor_c = Debe_c - Haber_c ; \text{ Si } Debe_c > Haber_c, \text{ si no, } Deudor_c = 0.$$

$$Acreedor_c = Haber_c - Debe_c ; \text{ Si } Haber_c > Debe_c, \text{ si no, } Acreedor_c = 0.$$

$$Activo_c = Deudor_c$$

$$Pasivo_c = Acreedor_c$$

$$total_debe = \sum Debe_c$$

$$total_haber = \sum Haber_c$$

$$total_dudor = \sum Deudor_c$$

$$total_acreedor = \sum Acreedor_c$$

$$total_Activo = \sum Activo_c$$

$$total_pasivo = \sum Pasivo_c$$

*Estas ecuaciones fueron validadas con un set de datos de prueba reales proporcionados por el cliente.

Entradas y salidas

- Entradas:
 - Agregar empresa:
 - RUT empresa.
 - Nombre empresa.
 - Dirección.
 - Ciudad.
 - Giro.
 - RUT representante legal.
 - Nombre representante legal.
 - Agregar Cuenta:
 - Código Cuenta.
 - Nombre Cuenta.
 - Subcategoría.

- Agregar Asiento:
 - Código de asiento.
 - Fecha.
 - Tipo (ingreso, egreso, traspaso).
 - Comprobante Contable:
 - Código de Comprobante Contable.
 - Cuenta
 - Tipo de Documento
 - Número de Documento
 - Glosa
 - Debe.
 - Haber.
- Salidas:
 - Libro Diario: A través de la consulta del Libro Diario se generará un informe que contendrá los datos de la empresa seleccionada, el periodo seleccionado previamente y el detalle del Libro Diario ordenado por el número de los asientos y la fecha, previamente ingresados.
 - Libro Mayor: A través de la consulta del Libro Mayor se generará un informe que contendrá un listado de los asientos previamente ingresados ordenados por el código de la cuenta.
 - Balance: A través de la consulta del Balance se generará un informe que contendrá los datos de la empresa seleccionada, el periodo seleccionado previamente, el detalle que contiene todas las cuentas y el total y/o subtotal de sus transacciones.
- Entidades: Se genera una base de datos con las siguientes tablas:
 - Cuenta.
 - Cliente.
 - Libro_Diario.

- Libro_Mayor.
- Balance.
- Asiento.
- Comprobante_Contable.
- Categoría.
- Subcategoría.

Estructura de Códigos.

- Usuario.

Descripción: Contiene el usuario y contraseña.

- USER: Contiene el ID del usuario
- PASS: Contiene la contraseña del usuario.

- Asiento.

Descripción: Contiene la información de los asientos contables.

Campos:

- NUM_ASIENTO: Contiene el número de un asiento específico.
- RUT_CLIENTE: Contiene la identificación del cliente.
- FECHA: Fecha del asiento.
- TIPO: Indica si es ingreso(I), egreso(E) o traspaso(T).

- Comprobante Contable.

Descripción: Son las líneas de detalle de un asiento.

Campos:

- COD_COMP_CONT: Contiene el código del comprobante contable
- COD_ASIENTO: Contiene el código del asiento al que pertenece el comprobante.
- COD_CUENTA: Contiene el código de una cuenta.
- TIPO: Contiene el tipo de documento.
- NUM_DOC: Contiene el número de documento

- GLOSA: Contiene una descripción del Comprobante Contable.
- DEBE: Contiene un monto al debe.
- HABER: Contiene un monto al haber.

- Cuenta.

Descripción: Contiene los nombres y códigos de todas las cuentas con el formato proporcionado por el SII.

Campos:

- COD_CUENTA: Contiene el código de una cuenta.
- COD_SUBCATEGORIA: Contiene la subcategoría a la que pertenece una cuenta.
- NOM_CUENTA: Contiene el nombre de una cuenta.

- Subcategoría.

Descripción: Contiene las subcategorías a las que pertenecen las cuentas.

Campos:

- COD_SUBCATEGORIA: Contiene el código de una subcategoría, largo 9.
- COD_CATEGORIA: Contiene el código de la categoría a la que pertenece una Subcategoría.
- NOM_SUBCATEGORIA: Contiene el nombre de una subcategoría.

- Categoría.

Descripción: Contiene las categorías a las que las subcategorías pertenecen.

Campos:

- COD_CATEGORIA: Contiene el código de una categoría, largo 3.
- NOM_CATEGORIA: Contiene el nombre de una categoría.

*Los códigos de las categorías y subcategorías responden al formato del SII.

- Cliente.

Descripción: Contiene los datos de los clientes.

Campos:

- RUT_CLIENTE: Contiene el RUT del cliente (empresa).
- NOM_CLIENTE: Nombre del cliente (empresa).
- CIUDAD: Ciudad en que se ubica.
- DIRECCION: Lugar en que se ubica.
- GIRO: Rubro al que se dedica la empresa.
- RUT_REP_LEGAL: RUT del representante legal de la empresa.
- NOM_REP_LEGAL: Nombre del representante legal de la empresa.
- Libro diario.

Descripción: Contiene un registro de los libros diarios emitidos.

Campos:

- NUM_DIARIO: Contiene la identificación del libro diario.
- RUT_CLIENTE: Contiene el RUT del cliente al que pertenece el libro diario.
- FECHA_INICIO: Fecha de inicio del libro diario.
- FECHA_TERMINO: Fecha de término del libro diario.
- FECHA_EMISION: Contiene la fecha en la que se emitió un libro diario.
- TIPO_ARCHIVO: Contiene el tipo de archivo con que se emitió el libro diario.
- FIRMA: Firma para comprobar si el documento ha sido modificado posterior a la entrega.
- Libro mayor.

Descripción: Contiene un registro de los libros mayores emitidos.

Campos:

- NUM_MAYOR: Contiene la identificación del libro mayor.
- RUT_CLIENTE: Contiene el RUT del cliente al que pertenece el libro mayor.
- FECHA_INICIO: Fecha de inicio del libro mayor.

- FECHA_TERMINO: Fecha de término del libro mayor.
 - FECHA_EMISION: Contiene la fecha en la que se emitió un libro mayor.
 - TIPO_ARCHIVO: Contiene el tipo de archivo con que se emitió el libro mayor.
 - FIRMA: Firma para comprobar si el documento ha sido modificado posterior a la entrega.
- Balance.

Descripción: Contiene un registro de los balances emitidos.

Campos:

- NUM_BALANCE: Contiene la identificación del balance.
- RUT_CLIENTE: Contiene el RUT del cliente al que pertenece el balance.
- FECHA_INICIO: Fecha de inicio del balance.
- FECHA_TERMINO: Fecha de término del balance.
- FECHA_EMISION: Contiene la fecha en la que se emitió un balance.
- TIPO_ARCHIVO: Contiene el tipo de archivo con que se emitió el balance.
- FIRMA: Firma para comprobar si el documento ha sido modificado posterior a la entrega.

Condiciones de diseño

El lenguaje de programación que se utilizará será Java SE, ya que se sitúa en un entorno de estaciones de trabajo (PC de escritorio).

Como Sistema Operativo se utilizará Windows 10 y motor de base de datos se utilizará la plataforma phpMyAdmin debido a que el sistema va dirigido a un usuario que cuenta con clientes de micro y pequeña empresa. Esta base de datos contará con un método de respaldo que creará una copia de seguridad semanalmente, la cual se guardará en una nube (DropBox). Adicionalmente, como 2do método de respaldo se irá guardando una copia de seguridad mensualmente en un dispositivo de almacenamiento externo (Disco duro extraíble), ya que la

cantidad de información diaria que se ingresa a la base de datos es muy importante tanto para la empresa como para el cliente.

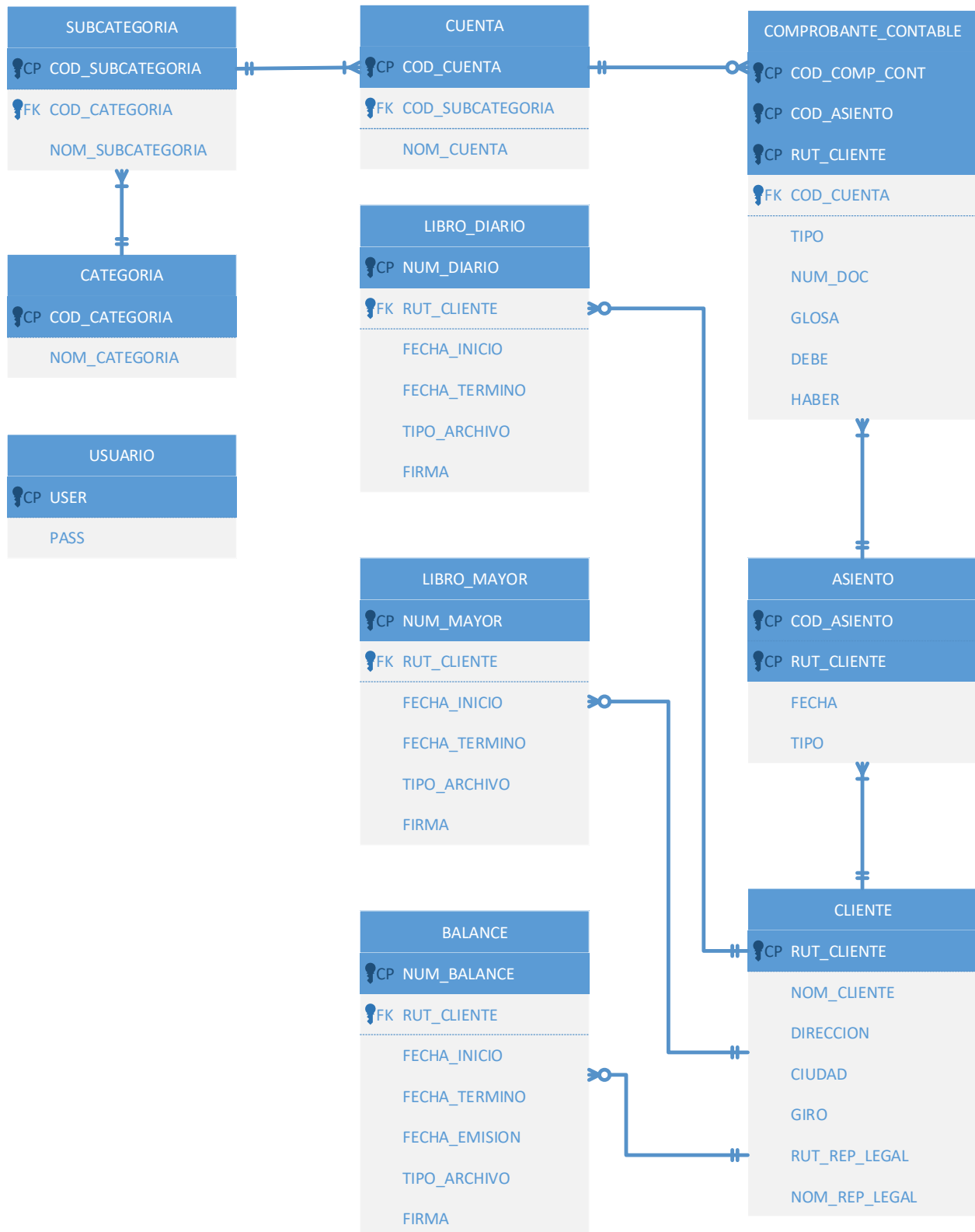
Modelo lógico de datos.

Figura 1-6. Modelo lógico de datos.

CAPÍTULO 2:

MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS.

2. Medio ambiente computacional y descripción de archivos.

2.1 Características del recurso computacional.

A continuación, se especificará el recurso computacional en el que se desarrollará el sistema tanto hardware como software.

2.1.1 Configuración del Sistema.

Hardware empleado en el desarrollo.

- Procesador Intel Core i7-7700k de 4.2 Ghz.
- Almacenamiento primario, 2 memorias RAM de 8GB (capacidad máxima 4 memorias de 8GB).
- Almacenamiento secundario, SSD Crucial (Solid-State Drive o Dispositivo de estado sólido) de 240GB.
- Disco duro Western Digital de 1TB a 7200 rpm (capacidad máxima: 6 unidades con puerto SATA).
- Monitor LED Samsung de 24 pulgadas con resolución 1920 x 1080 (capacidad máxima 3 monitores).

Requisitos mínimos para el sistema.

- Procesador Intel Pentium de 2.0 Ghz.
- Almacenamiento primario, memoria RAM 2 GB.
- Almacenamiento secundario, Disco duro con 2 GB de espacio libre.

Hardware empleado en el Cliente.

- Procesador Intel Core i3-3110M de 2.4 Ghz.
- Almacenamiento primario, memoria RAM de 8GB (capacidad máxima 2 memorias de 8GB).

- Almacenamiento secundario, Disco duro Western Digital de 500GB a 5400 rpm (capacidad máxima: 1 unidad con puerto SATA de 5TB).
- Monitor LED de 14 pulgadas integrado en el notebook, además soporta una pantalla externa conectada vía HDMI. Además, cuenta con una capacidad máxima de 4 puertos USB.
- Impresora láser Samsung ML-1865 conectada por cable USB (capacidad máxima 4 impresoras).

2.1.2 Software Utilizado.

Software utilizado para el desarrollo.

- Sistema operativo Windows 10 Home de 64 bits.
- Herramientas de desarrollo a usar son NetBeans 8.2
- Administrador de base de datos MySQL.
- Adobe Reader (PDF).

Software utilizado por el cliente.

- Sistema operativo Windows 10 Home de 64 bits.
- Administrador de base de datos MySQL.
- Adobe Reader (PDF).

2.2 Descripción de Archivos.

Tabla 2-1. Campos de tabla Asiento.

ASIENTO			
Nombre Lógico	ASIENTO		
Nombre Físico	ASIENTO		
Descripción	Contiene todos los asientos.		
Clave primaria	COD_ASIENTO + RUT_CLIENTE		
Clave foránea	RUT_CLIENTE (ref. tabla CLIENTE).		
Campos			
Nombre	Descripción	Tipo	Longitud
COD_ASIENTO	Contiene el código del Asiento.	int	6
RUT_CLIENTE	Contiene el RUT del cliente al que pertenece el Asiento.	varchar	13
FECHA	Fecha del Asiento.	date	10 (AAAA/MM/DD)
TIPO	Indica si es Ingreso (I), Egreso E o Traspaso (T).	char	1

Tabla 2-2. Campos de tabla Comprobante_Contable.

COMPROBANTE_CONTABLE			
Nombre Lógico	COMPROBANTE_CONTABLE		
Nombre Físico	COMPROBANTE_CONTABLE		
Descripción	Contiene la información de las distintas líneas de detalle de los asientos.		
Clave primaria	COD_COMP_CONT + COD_ASIENTO ASIENTO + RUT_CLIENTE.		
Clave foránea	COD_ASIENTO (ref. tabla ASIENTO) + RUT_CLIENTE (ref. tabla CLIENTE) + COD_CUENTA (ref. tabla CUENTA).		
Campos			
Nombre	Descripción	Tipo	Longitud
COD_COMP_CONT	Contiene el código del comprobante contable.	int	6
COD_ASIENTO	Contiene el código del Asiento al que pertenece el comprobante contable.	int	6
RUT_CLIENTE	Contiene el RUT del cliente al que pertenece el Asiento.	varchar	13
COD_CUENTA	Contiene el código de una cuenta.	varchar	10
TIPO	Contiene el tipo de comprobante contable (I – Ingreso, E – egreso, T - Traspaso)	char	1
NUM_DOC	Contiene el número del documento ingresado	varchar	10
GLOSA	Contiene una breve descripción del comprobante contable	varchar	255

DEBE	Contiene un monto al debe.	int	9
HABER	Contiene un monto al haber.	int	9

Tabla 2-3. Campos de tabla Cuenta.

CUENTA			
Nombre Lógico	CUENTA		
Nombre Físico	CUENTA		
Descripción	Contiene los nombres y códigos de todas las cuentas.		
Clave primaria	COD_CUENTA.		
Clave foránea	COD_SUBCATEGORIA (ref. tabla SUBCATEGORIA).		
Campos			
Nombre	Descripción	Tipo	Longitud
COD_CUENTA	Contiene el código de una cuenta.	varchar	10
COD_SUBCATEGORIA	Contiene la subcategoría a la que pertenece una cuenta.	varchar	5
NOM_CUENTA	Contiene el nombre de una cuenta.	varchar	255

Tabla 2-4. Campos de tabla Subcategoria.

SUBCATEGORIA			
Nombre Lógico	SUBCATEGORIA		
Nombre Físico	SUBCATEGORIA		
Descripción	Contiene las subcategorías a las que pertenecen las cuentas.		
Clave primaria	COD_SUBCATEGORIA.		
Clave foránea	COD_CATEGORIA (ref. tabla CATEGORIA).		
Campos			
Nombre	Descripción	Tipo	Longitud
COD_SUBCATEGORIA	Contiene el código de una subcategoría.	varchar	5
COD_CATEGORIA	Contiene el código de la categoría a la que pertenece una subcategoría.	varchar	3
NOM_SUBCATEGORIA	Contiene el nombre de una subcategoría.	varchar	50

Tabla 2-5. Campos de tabla Categoría.

CATEGORIA			
Nombre Lógico	CATEGORIA		
Nombre Físico	CATEGORIA		
Descripción	Contiene las categorías a las que pertenecen las subcategorías de las cuentas.		
Clave primaria	COD_CATEGORIA		
Campos			
Nombre	Descripción	Tipo	Longitud
COD_CATEGORIA	Contiene el código de una categoría.	varchar	3
NOM_CATEGORIA	Contiene el nombre de una categoría.	varchar	50

Tabla 2-6. Campos de tabla Cliente.

CLIENTE			
Nombre Lógico	CLIENTE		
Nombre Físico	CLIENTE		
Descripción	Contiene los datos de los clientes.		
Clave primaria	RUT_CLIENTE		
Campos			
Nombre	Descripción	Tipo	Longitud
RUT_CLIENTE	Identificación Empresa.	varchar	13
NOM_CLIENTE	Nombre del Cliente (Empresa).	varchar	70
CIUDAD	Ciudad en que se ubica.	varchar	70
DIRECCION	Lugar en que se ubica.	varchar	70
GIRO	Rubro al que se dedica la empresa.	varchar	70
RUT_REP_LEGAL	RUT del representante legal de la empresa.	varchar	13
NOM_REP_LEGAL	Nombre del representante legal de la empresa.	varchar	70
FONO	Número telefónico de la empresa	varchar	14
CORREO	Correo electrónico de la empresa	varchar	70

Tabla 2-7. Campos de tabla Libro_Diario.

LIBRO_DIARIO			
Nombre Lógico	LIBRO_DIARIO		
Nombre Físico	LIBRO_DIARIO		
Descripción	Mantiene un registro de los Libros Diarios emitidos.		
Clave Primaria	NUM_DIARIO		
Clave Foránea	RUT_CLIENTE (ref. tabla CLIENTE).		
Campos			
Nombre	Descripción	Tipo	Longitud
NUM_DIARIO	Contiene la identificación del Libro Diario.	int	9
RUT_CLIENTE	Contiene el RUT del cliente al que pertenece el Libro Diario.	varchar	13
FECHA_INICIO	Fecha inicio del Libro Diario.	date	10 (AAAA/MM/DD)
FECHA_TERMINO	Fecha término del Libro Diario.	date	10 (AAAA/MM/DD)
FECHA_EMISIÓN	Contiene la fecha en la que se emitió un Libro Diario.	date	10 (AAAA/MM/DD)
TIPO_ARCHIVO	Contiene el tipo de archivo con que se emitió el libro diario. (PDF o Excel)	varchar	10
FIRMA	Firma para comprobar si el documento ha sido modificado posterior a la entrega.	bigint	

Tabla 2-8. Campos de tabla Libro_Mayor.

LIBRO_MAYOR			
Nombre Lógico	LIBRO_MAYOR		
Nombre Físico	LIBRO_MAYOR		
Descripción	Mantiene un registro de los Libros Mayores emitidos.		
Clave Primaria	NUM_MAYOR		
Clave Foránea	RUT_CLIENTE (ref. tabla CLIENTE).		
Campos			
Nombre	Descripción	Tipo	Longitud
NUM_MAYOR	Contiene la identificación del Libro Mayor.	int	9
RUT_CLIENTE	Contiene el RUT del cliente al que pertenece el Libro Mayor.	varchar	13
FECHA_INICIO	Fecha inicio del Libro Mayor.	date	10 (AAAA/MM/DD)
FECHA_TERMINO	Fecha Termino del Libro Mayor.	date	10 (AAAA/MM/DD)
FECHA_EMISIÓN	Contiene la fecha en la que se emitió el Libro Mayor.	date	10 (AAAA/MM/DD)
TIPO_ARCHIVO	Contiene el tipo de archivo con que se emitió el libro mayor. (PDF o Excel)	varchar	10
FIRMA	Firma para comprobar si el documento ha sido modificado posterior a la entrega	bigint	

Tabla 2-9. Campos de tabla Balance.

BALANCE			
Nombre Lógico	BALANCE		
Nombre Físico	BALANCE		
Descripción	Mantiene un registro de los Balances emitidos.		
Clave Primaria	NUM_BALANCE		
Clave Foránea	RUT_CLIENTE (ref. tabla CLIENTE).		
Campos			
Nombre	Descripción	Tipo	Longitud
NUM_BALANCE	Contiene la identificación del Balance.	int	9
RUT_CLIENTE	Contiene el RUT del cliente al que pertenece el Balance.	varchar	13
FECHA_INICIO	Fecha inicio del Balance.	date	10 (DD/MM/AAAA)
FECHA_TERMINO	Fecha Termino del Balance.	date	10 (DD/MM/AAAA)
FECHA_EMISIÓN	Contiene la fecha en la que se emitió el Balance.	date	10 (DD/MM/AAAA)
TIPO_ARCHIVO	Contiene el tipo de archivo con que se emitió el balance. (PDF o Excel)	varchar	10
FIRMA	Firma para comprobar si el documento ha sido modificado.	bigint	

Tabla 2-10. Campos de tabla Usuario.

USUARIO			
Nombre Lógico	USUARIO		
Nombre Físico	USUARIO		
Descripción	Contiene el usuario y contraseña.		
Clave primaria	USER		
Campos			
Nombre	Descripción	Tipo	Longitud
USER	Contiene el ID del usuario.	varchar	15
PASS	Contiene la contraseña del usuario.	Varchar	15

CAPÍTULO 3:

DESCRIPCIÓN DE PROGRAMAS.

3. Descripción de programas.

3.1 Diagrama de Menús.

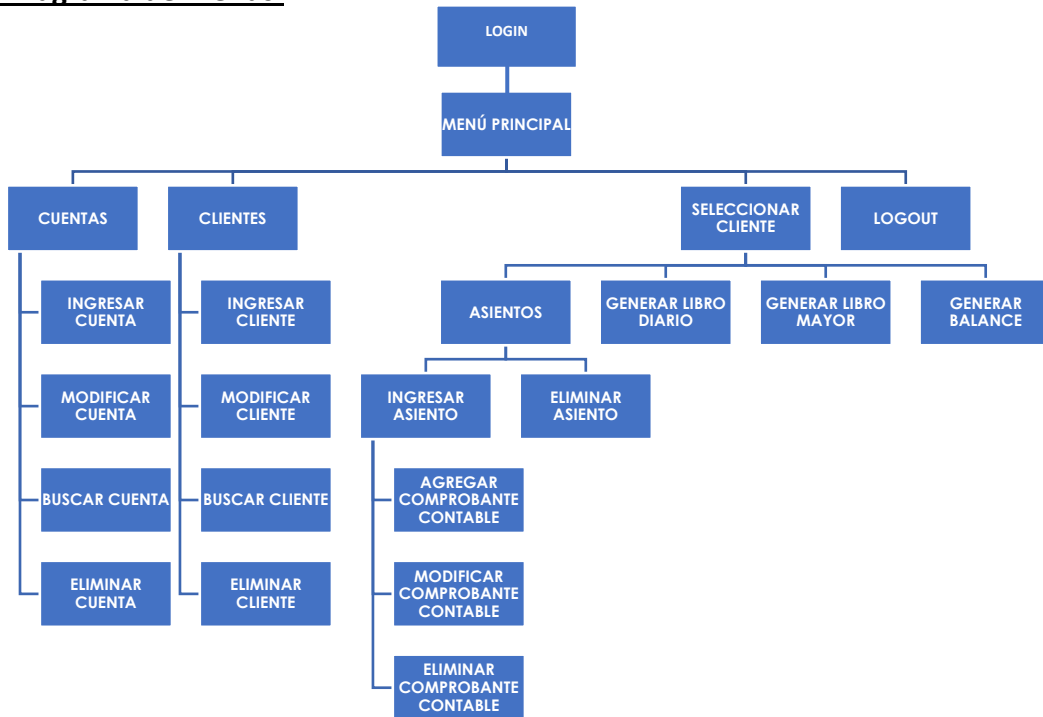


Figura 3-1. Diagrama de menús.

3.2 Diagrama modular.

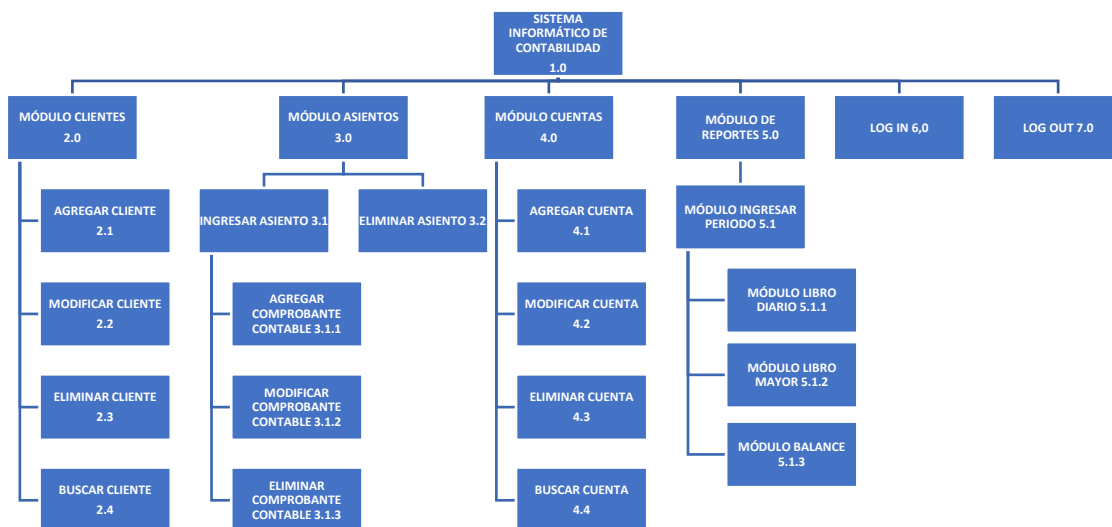


Figura 3-2. Diagrama modular.

3.3 Programas del Sistema.

A continuación, se presentará un listado con la totalidad de los programas con los que interactúa el usuario, de los cuales se destacarán con un “*” aquellos programas que serán descritos detalladamente en el siguiente punto.

NOMBRE PROGRAMA	OBJETIVO
Log in (*)	Permite autenticar el usuario y la contraseña.
Menú Principal (*)	Permite acceder al Administrador de Clientes, al Administrador de cuentas y seleccionar un cliente al cual se le llevará la contabilidad.
Mantenedor de Clientes (*)	Permite agregar, modificar, eliminar y buscar un cliente en particular.
Agregar Cliente	Permite agregar nuevo un cliente.
Modificar Cliente	Permite modificar un cliente existente.
Eliminar Cliente	Permite eliminar un cliente existente.
Mantenedor de Cuentas (*)	Permite agregar, modificar, eliminar y buscar una cuenta en particular.
Agregar Cuenta	Permite agregar una nueva cuenta.
Modificar Cuenta	Permite modificar una cuenta existente.
Eliminar Cuenta	Permite eliminar una cuenta existente
Mantenedor de Asientos (*)	Permite ingresar o eliminar asientos, además de ingresar sus respectivos comprobantes contables (detalle de asientos), los cuales también se pueden modificar y eliminar.
Agregar Asiento	Permite agregar asientos a un cliente en particular.
Eliminar Asiento	Permite eliminar un asiento junto con sus comprobantes asociados.
Agregar Comprobante Contable	Permite agregar un comprobante contable a un asiento en particular.
Modificar Comprobante Contable	Permite modificar un comprobante contable de un asiento en particular.
Eliminar Comprobante Contable	Permite eliminar un comprobante contable de un asiento en particular.
Ingresar Periodo (*)	Permite seleccionar la fecha de inicio y de término para la cual se generarán los distintos reportes.
Generar Libro Diario (*)	Permite generar una vista del Libro Diario, que posteriormente será exportado en PDF, o impreso directamente.

Generar Libro Mayor (*)	Permite generar una vista del Libro Mayor, que posteriormente será exportado en PDF, o impreso directamente.
Generar Balance (*)	Permite generar una vista del Balance, que posteriormente será exportado en PDF, o impreso directamente.
Log out	Permite cerrar la sesión del usuario.

3.4 Estructura general del sistema.

3.4.1 Login.

- Nombre: LOGIN.
- Objetivo: Permite autenticar el usuario y la contraseña.
- Referencia de código fuente: Página 49.
- Diagrama de Bloque:



Figura 3-3. Diagrama de bloque "Login".

- Reglas de proceso:
 - El sistema despliega un formulario en donde el usuario debe iniciar sesión con su ID y contraseña. El usuario ingresa sus datos y presiona el botón ACEPTAR. El sistema valida si los datos son correctos, si es así, se redirigirá al formulario del MENÚ PRINCIPAL. Si no son correctos, el sistema desplegará un mensaje de error indicando que el usuario y/o contraseña son incorrectos.

- Diseño de pantalla:

Figura 3-4. Diseño de pantalla “Login”.

3.4.2 Menú Principal.

- Nombre: MENÚ PRINCIPAL.
- Objetivo: Permite acceder al Administrador de Clientes, el Administrador de cuentas y seleccionar un cliente al cual se le llevará la contabilidad.
- Referencia de código fuente: Página 55.
- Diagrama de bloque:

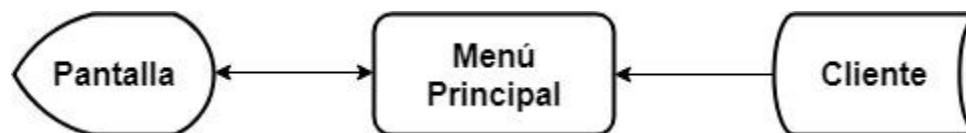


Figura 3-5. Diagrama de bloque “Menú Principal”.

- Reglas de proceso:
- El sistema despliega un menú que contiene el botón CLIENTES, el botón CUENTAS, un Buscador de cliente por su Rut, un buscador de cliente por su nombre, el botón CERRAR SESIÓN y una lista seleccionable de los clientes que tiene la empresa que permitirá llevar la contabilidad del cliente seleccionado. Si se presiona el botón CERRAR SESIÓN, se

redireccionará a la pantalla LOGIN en donde se deberá volver a ingresar el usuario y contraseña.

- Diseño de pantalla:

RUT CLIENTE	NOMBRE CLIENTE
6512684-2	WAs Corporation
25607016-2	Nunc Mauris Associates
33779883-7	Molestie Inc.
45099489-8	Cubilia Curae; Donec Industries
19936544-4	At Velit Incorporated
21253383-1	Nunc Interdum Feugiat Institute
50806605-8	Aliquet Odio Limited
22975474-2	Ornare Tortor At Limited
26239443-3	Ipsum Corporation
24907607-4	Auctor Odio Ltd
25154343-7	Aliquam Erat Volutpat LLP
47597766-1	Orci Consectetur PC
68549335-8	Vel Feugiat Limited

Figura 3-6. Diseño de pantalla “Menú Principal”.

3.4.3 Mantenedor de Clientes.

- Nombre: MANTENEDOR DE CLIENTES
- Objetivo: Permite agregar, modificar, eliminar y buscar un cliente en particular.
- Referencia de código fuente: Página 70.
- Diagrama de Bloque:



Figura 3-7. Diagrama de bloque “Mantenedor de Clientes”.

- Relas de Proceso:
 - El sistema despliega un formulario que contiene un buscador de clientes por su Rut y un buscador de clientes por su nombre. Para agregar un cliente, se presiona el botón AGREGAR, lo cual habilitará los campos del formulario para ser llenados y

posteriormente guardarlos presionando el botón ACEPTAR, en caso contrario el botón CANCELAR. También contiene un listado seleccionable de todos los clientes de la empresa que al clickear sus datos se desplegarán en los campos del formulario, lo cual habilitará el botón MODIFICAR y el botón ELIMINAR. Si se modifica un cliente, para guardar los datos se presiona el botón ACEPTAR, en caso contrario se presiona el botón CANCELAR. Si se desea eliminar se desplegará un cuadro de texto que validará si está seguro de eliminar, al cual se presiona sí o no.

- Diseño de pantalla:

Figura 3-8. Diseño de pantalla “Administrador de clientes”.

3.4.4 Mantenedor de Cuentas.

- Nombre: MANTENEDOR DE CUENTAS.
- Objetivo: Permite agregar, modificar, eliminar y buscar una cuenta en particular.
- Referencia de código fuente: Página 106.

- Diagrama de bloque:

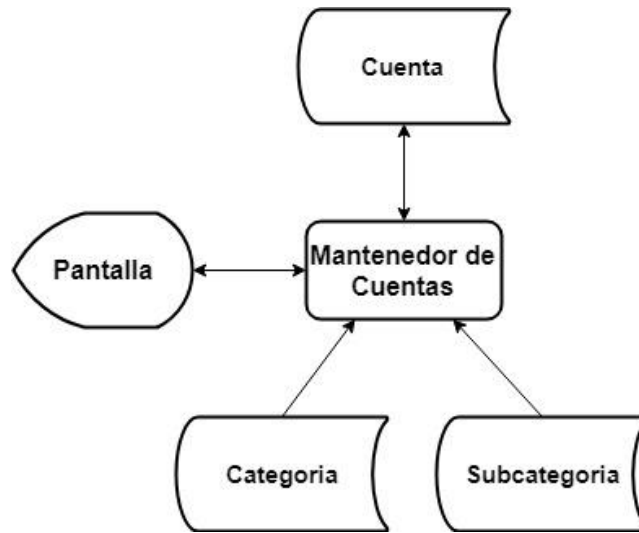


Figura 3-9. Diagrama de bloque "Mantenedor de Cuentas".

- Reglas de proceso:
 - El sistema despliega un formulario que contiene un buscador de cuentas por su código y un buscador de cuentas por su nombre. Para agregar una cuenta, se presiona el botón AGREGAR, lo cual habilitará los campos del formulario para ser llenados y posteriormente guardar presionando el botón ACEPTAR, en caso contrario el botón CANCELAR. También contiene un listado seleccionable de todas las cuentas que al clickear sus datos se desplegarán en los campos del formulario, lo cual habilitará el botón MODIFICAR y el botón ELIMINAR. Si se modifica una cuenta, para guardar los datos se presiona el botón ACEPTAR, en caso contrario se presiona el botón CANCELAR. Si se desea eliminar se desplegará un cuadro de texto que validará si está seguro de eliminar, al cual se presiona sí o no.

- Diseño de pantalla:

MANTENEDOR DE CUENTAS
ADMINISTRADOR DE CUENTAS

CÓDIGO: NOMBRE:

CODIGO	NOMBRE	SUBCATEGORIA	CATEGORIA
1.1.10	CAJA	ACTIVO CIRCULANTE	ACTIVO
1.1.100	DOCUMENTOS POR CO...	ACTIVO CIRCULANTE	ACTIVO
1.1.110	DOCUMENTOS Y CUEN...	ACTIVO CIRCULANTE	ACTIVO
1.1.120	DOCUMENTOS Y CUEN...	ACTIVO CIRCULANTE	ACTIVO
1.1.130	ESTIMACION DEUDORE...	ACTIVO CIRCULANTE	ACTIVO
1.1.140	DEUDORES VARIOS	ACTIVO CIRCULANTE	ACTIVO
1.1.150	ANTICIPO REMUNERAC...	ACTIVO CIRCULANTE	ACTIVO
1.1.160	PRESTAMOS A TRABAJA...	ACTIVO CIRCULANTE	ACTIVO
1.1.170	OTROS DESCUENTOS ...	ACTIVO CIRCULANTE	ACTIVO
1.1.180	PRESTAMOS A SOCIO (...)	ACTIVO CIRCULANTE	ACTIVO
1.1.190	CTA CTE CONSIGNATO...	ACTIVO CIRCULANTE	ACTIVO
1.1.20	BANCO	ACTIVO CIRCULANTE	ACTIVO
1.1.200	IMPUESTOS POR RECU...	ACTIVO CIRCULANTE	ACTIVO
1.1.210	IMPUESTO ESPECIFICO...	ACTIVO CIRCULANTE	ACTIVO
1.1.220	IVA CREDITO	ACTIVO CIRCULANTE	ACTIVO
1.1.230	CREDITO IMPUESTO LE...	ACTIVO CIRCULANTE	ACTIVO
1.1.240	CREDITO IMPUESTO ES...	ACTIVO CIRCULANTE	ACTIVO
1.1.250	CREDITO IMPUESTO AD...	ACTIVO CIRCULANTE	ACTIVO
1.1.260	IMPUESTOS DIFERIDOS	ACTIVO CIRCULANTE	ACTIVO

CÓDIGO: NOMBRE:

CATEGORÍA: ACTIVO

SUBCATEGORÍA: ACTIVOS FIJOS

AGREGAR MODIFICAR ELIMINAR ACEPTAR CANCELAR VOLVER

Figura 3-10. Diseño de pantalla “Mantenedor de Cuentas”.

3.4.5 Gestión de Asientos.

- Nombre: GESTIÓN DE ASIENTOS.
- Objetivo: Permite ingresar asientos y sus respectivos comprobantes contables (detalle de asientos), los cuales también se pueden modificar y eliminar.
- Referencia de código fuente: Página 133.

- Diagrama de bloques:

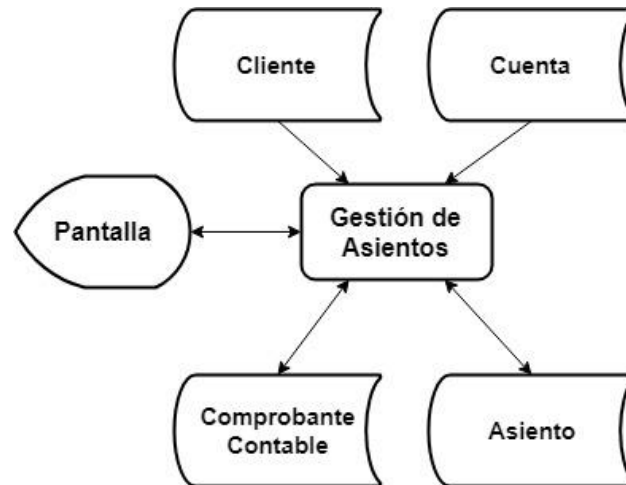


Figura 3-11. Diagrama de bloque "Gestión de Asientos".

- Reglas de proceso:
 - Se despliega un formulario que contiene la información del cliente al cual se le llevará la contabilidad, los botones de Libro Diario, Libro Mayor y Balance que generan reportes según un rango de fecha determinada. También contiene el ingreso de asientos en el que se agregan asientos y además se agregan sus respectivos comprobantes contables, los cuales se pueden modificar editando los campos correspondientes en la sección de comprobantes contables, y eliminar. Al eliminar un asiento se eliminarán también todos los comprobantes contables pertenecientes a ese asiento.

- Diseño de pantalla:

Figura 3-12. Diseño de pantalla “Gestión de asientos”.

3.4.6 Generar Libro Diario.

- Nombre: GENERAR LIBRO DIARIO.
- Objetivo: Permite generar una vista del Libro Diario, que posteriormente será exportado en PDF, o impreso directamente.
- Referencia de código fuente: Página 183.
- Diagrama de bloque:

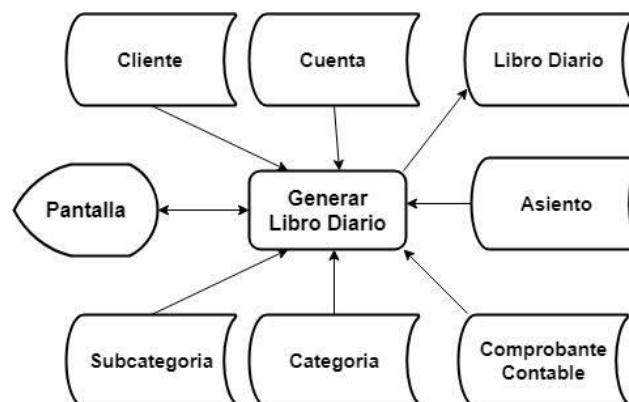


Figura 3-13. Diagrama de bloque “Generar Libro Diario”.

- Reglas de proceso:
 - Este proceso recibe como parámetros de entrada el RUT del cliente con el que se está trabajando, la fecha de inicio y la fecha de término, para obtener los datos de los asientos que tengan movimientos entre dichas fechas, para lo cual, se dará un formato al reporte que contendrá los datos del cliente, la información filtrada por asientos según los parámetros recibidos y un valor total, que permitirá evaluar la situación financiera del cliente. Finalmente se guardará el registro de la reneración del reporte en la tabla Libro_Diario.
- Diseño de pantalla:

EMPRESA SA.				FECHA 2019/04/28	
R.U.T 19489883-5				HORA 7.49 PM	
GIRO MIAU					
DIRECCIÓN PRAT					
CIUDAD VILLA ALEMANA					
LIBRO DIARIO					
NRO. CTA	NOMBRE	TIPO	GLOSA	DÉBITO	CRÉDITO
ASIENTO : 1 TIPO : I DIA : 6					
1.1.10	CAJA		CAJA	200000	0
1.1.50	MERCADERIAS		MERCADERIAS	50000	0
4.3.40	REPARACIONES		VEHICULOS	900000	0
1.1.90	DOCUMENTOS POR		LXC	50000	0
3.1.10	CAPITAL PAGADO		CAPITAL	0	1200000
TOTAL ASIENTO :				1200000	1200000
ASIENTO : 2 TIPO : I DIA : 6					
1.1.20	BANCO		BANCO	120000	0
1.1.10	CAJA		CAJA	0	120000
1.1.50	MERCADERIAS		MERCADERIAS	400000	0
1.1.220	IVA CREDITO		IVA CF	76000	0
1.1.10	CAJA		PROVEEDORES	0	476000
TOTAL ASIENTO :				596000	596000
ASIENTO : 3 TIPO : E DIA : 5					

Figura 3-14. Diseño de pantalla “Generar Libro Diario”.

3.4.7 Generar Libro Mayor.

- Nombre: GENERAR LIBRO MAYOR.
- Objetivo: Permite generar una vista del Libro Mayor, que posteriormente será exportado en PDF, o impreso directamente.

- Referencia de código fuente: Página 183.
- Diagrama de bloque:

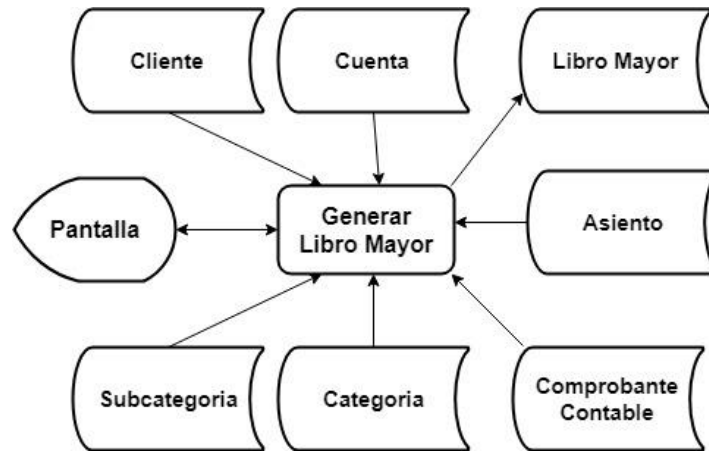


Figura 3-15. Diagrama de bloque “Generar Libro Mayor”.

- Reglas de proceso:
 - Este proceso recibe como parámetros de entrada el RUT del cliente con el que se está trabajando, la fecha de inicio y la fecha de término, para obtener los datos de los asientos que tengan movimientos entre dichas fechas, para lo cual, se dará un formato al reporte que contendrá los datos del cliente, la información filtrada por cuentas según los parámetros recibidos y un valor total, que permitirá evaluar la situación financiera del cliente. Finalmente se guardará el registro de la reneración del reporte en la tabla Libro_Mayor.

- Diseño de pantalla:

FECHA	TIPO	GLOSA	DÉBITO	CRÉDITO	SALDOS
30-01-16	I	CAJA	200000	0	200000
30-01-16	I	CAJA	0	120000	80000
30-01-16	I	PROVEEDORES	0	476000	-396000
12-02-16	E	CAJA	0	29750	-425750
12-02-16	I	PROVEEDORES	476000	0	50250
28-02-16	I	CAJA	42840	0	93090
30-03-16	I	CAJA	0	4000	89090
28-02-16	I	CAJA	0	42840	46250
30-03-16	I	ARRIENDO	25000	0	71250
30-03-16	I	VENTA	0	18000	53250
12-04-16	E	CAJA	0	8220	45030
12-04-16	E	PROVEEDORES	0	20552	24478
30-04-16	E	CAJA	650000	0	674478
12-05-16	E	CAJA	0	650000	24478

Figura 3-16. Diseño de pantalla “Generar Libro Mayor”.

3.4.8 Generar Balance.

- Nombre: GENERAR BALANCE
- Objetivo: Permite generar una vista del Balance, que posteriormente será exportado en PDF, o impreso directamente.
- Referencia de código fuente: Página 183.

- Diagrama de bloque:

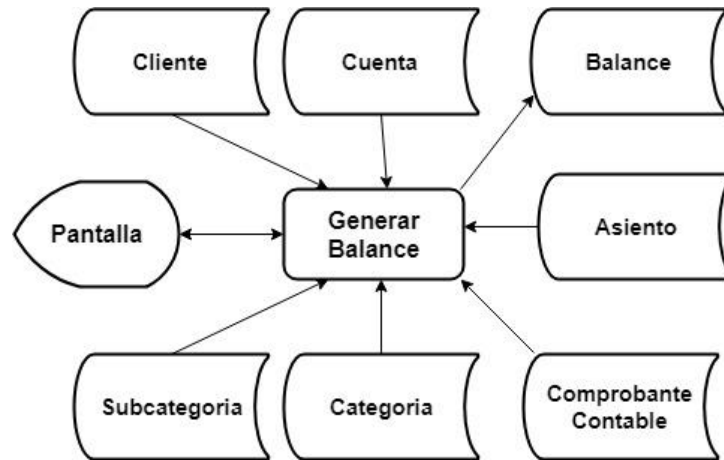


Figura 3-17. Diagrama de bloque "Generar Balance".

- Reglas de proceso:
 - Este proceso recibe como parámetros de entrada el RUT del cliente con el que se está trabajando, la fecha de inicio y la fecha de término, para obtener los datos de los asientos que tengan movimientos entre dichas fechas, para lo cual, se dará un formato al reporte que contendrá los datos del cliente, la información filtrada por el total de las cuentas según los parámetros recibidos y un valor total, que permitirá evaluar la situación financiera del cliente. Finalmente se guardará el registro de la reneración del reporte en la tabla Balance.

- Diseño de pantalla:

cuenta	debe	haber	deudor	acreedor	activo	pasivo	perdida	ganancia
XXXXX	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
XXXXX	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
XXXXX	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
XXXXX	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
XXXXX	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
UTILIDAD VTA ACTIVO FUD	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
TOTALES	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -

Figura 3-18. Diseño de pantalla “Generar Balance”.

3.4.9 Ingreso de Periodo.

- Nombre: INGRESO DE PERIODO
- Objetivo: Permite seleccionar la fecha de inicio y de término para la cual se generarán los distintos reportes.
- Referencia de código fuente: Página 183.
- Diagrama de bloque:

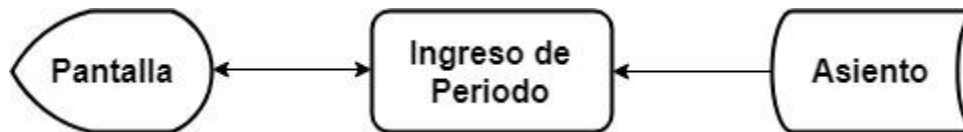


Figura 3-19. Diagrama de bloque “Ingreso de Periodo”.

- Reglas de proceso:
 - Al presionar el botón asociado con alguno de los 3 reportes (Libro diario, Libro Mayor o Balance), el sistema desplegará una ventana, la cual contiene 6 combo box, en donde los primeros 3 son los encargados de determinar la fecha de inicio y los otros 3 son los encargados de determinar la fecha de término para dicho

reporte (Formato: DD/MM/AAAA). Al presionar el botón “ACEPTAR” se envían como parámetros para generar el respectivo informe la fecha de inicio, la fecha de término y el RUT del cliente con el cual se esta trabajando, para así filtrar los asientos según el rango de fechas establecido.

- Diseño de pantalla:



Figura 3-20. Diseño de pantalla “Ingreso Periodo” (para Libro Diario, Libro Mayor y Balance).

3.5 Otros elementos.

Los reportes emitidos contarán con una firma virtual la cual permitirá autenticar la originalidad de dichos reportes. Así se verificará que estos no hayan sido modificados después de su emisión.

Conclusiones

El tiempo que tomó realizar este trabajo fue de aproximadamente 5 meses, en el cual, durante el proceso, fue necesario investigar el uso y funcionamiento de un plugin para el IDE que se utilizó (NetBeans), que tiene por nombre “JasperReports”, en donde esta herramienta permitió generar los reportes necesarios para el sistema y para el cliente.

Un factor importante para poder realizar este sistema contable fue el constante contacto con el cliente, ya que éste proporcionaba detalles relevantes sobre contabilidad durante el desarrollo del sistema que ayudaron a obtener los resultados esperados al momento de realizar las pruebas y así, la información fuera coherente al generar los reportes.

Otra parte importante de este sistema es la conexión con la base de datos y que los datos de ésta sean estrictamente consistentes, sobre todo al momento de generar un reporte, por lo que el conocimiento de SQL y el manejo de base de datos fue esencial para que la información se mantuviera consistente y el sistema funcionara de manera óptima.

Una de las proyecciones de este sistema es que se pueda utilizar como herramienta de enseñanza para aquellas personas que requieran entender cómo es el proceso de la contabilidad estándar y hacer más fácil su comprensión.

Actualmente se ha pensado en realizar una interfaz más interactiva en el caso de que el sistema sea utilizado para la enseñanza.

Para finalizar, realizar este trabajo fue una ardua tarea, donde hubo muchos contratiempos, pero gracias a la paciencia, constancia y perseverancia se pudo llevar a cabo este gran proyecto.

Bibliografía

- **Jaspersoft**, Página principal de los creadores del software que permite generar reportes, sección aprender.

<https://www.jaspersoft.com/quick-start>

- **W3Schools**, Página web para el aprendizaje, las pruebas y la capacitación, Sección SQL.

<https://www.w3schools.com/sql/>

Anexo

A continuación, se adjunta el código fuente de los archivos mencionados en la sección 2 del capítulo 3, a excepción del código de los reportes que será incluido en el CD.

Login

```
import java.sql.Connection;

import java.sql.Statement;

import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in Project Properties.
 *
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Valentaiguer
 */
public class Login extends javax.swing.JFrame {

    /**
     * Creates new form Login
     */
    public Login() {
        initComponents();
        setTitle("LOGIN");
        this.setLocationRelativeTo(null);
    }
    /**
```

```

* This method is called from within the constructor to initialize the form.

* WARNING: Do NOT modify this code. The content of this method is always

* regenerated by the Form Editor.

*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

    jPanel1 = new javax.swing.JPanel();

    BTN_SALIR = new javax.swing.JButton();

    BTN_ACEPTAR = new javax.swing.JButton();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    TXT_USUARIO = new javax.swing.JTextField();

    TXT_PASS = new javax.swing.JPasswordField();

    jLabel3 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setMinimumSize(new java.awt.Dimension(400, 300));

    setPreferredSize(new java.awt.Dimension(450, 350));

    getContentPane().setLayout(null);

    jPanel1.setMinimumSize(new java.awt.Dimension(400, 300));

    BTN_SALIR.setText("SALIR");

    BTN_SALIR.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            BTN_SALIRActionPerformed(evt);

        }

    });

    BTN_ACEPTAR.setText("ACEPTAR");

    BTN_ACEPTAR.addActionListener(new java.awt.event.ActionListener() {

```



```

        .addGap(12, 12, 12)

        .addComponent(jLabel3))

        .addComponent(TXT_USUARIO, javax.swing.GroupLayout.DEFAULT_SIZE, 173, Short.MAX_VALUE)

        .addComponent(TXT_PASS))

        .addGap(83, 83, 83))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

        .addGap(46, 46, 46)

        .addComponent(BTN_SALIR, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addGap(92, 92, 92)

        .addComponent(BTN_ACEPTAR, javax.swing.GroupLayout.PREFERRED_SIZE, 116, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(22, 22, 22))

    );

    jPanel1Layout.setVerticalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

        .addGap(25, 25, 25)

        .addComponent(jLabel3)

        .addGap(56, 56, 56)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel1)

            .addComponent(TXT_USUARIO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(55, 55, 55)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(TXT_PASS, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66, Short.MAX_VALUE)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(BTN_SALIR)

        .addComponent(BTN_ACEPTAR))

        .addGap(21, 21, 21))

    );

    getContentPane().add(jPanel1);

    jPanel1.setBounds(0, 0, 400, 300);

    pack();

} // </editor-fold>

private void TXT_PASSActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void BTN_ACEPTARActionPerformed(java.awt.event.ActionEvent evt) {

    if (TXT_USUARIO.getText().toUpperCase().equals("ADMIN") && (TXT_PASS.getText().toUpperCase().equals("ADMIN"))) {

        Menu menu = new Menu();

        menu.setVisible(true);

        this.dispose();

    } else {

        JOptionPane.showMessageDialog(null, "USUARIO O CONTRASEÑA INCORRECTOS.", "ERROR", JOptionPane.ERROR_MESSAGE);

        TXT_USUARIO.setText("");

        TXT_PASS.setText("");

        TXT_USUARIO.requestFocus();

    }

    // TODO add your handling code here:

}

private void BTN_SALIRActionPerformed(java.awt.event.ActionEvent evt) {

    System.exit(0);

    // TODO add your handling code here:

}

```

```

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    //</editor-fold>

    /* Create and display the form */

    java.awt.EventQueue.invokeLater(new Runnable() {

```

```

    public void run() {

        new Login().setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JButton BTN_ACEPTAR;

private javax.swing.JButton BTN_SALIR;

private javax.swing.JPasswordField TXT_PASS;

private javax.swing.JTextField TXT_USUARIO;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

// End of variables declaration

}

```

Menú

```

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.JOptionPane;

import javax.swing.RowFilter;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableRowSorter;

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Valentaiguer
 */
public class Menu extends javax.swing.JFrame {

    private Statement SENTENCIA1;

    private Connection CONEXION1;

    private TableRowSorter filtro2;

    DefaultTableModel tabla2 = new DefaultTableModel(){

        @Override

        public boolean isCellEditable(int fil, int col) {

            return false;

        }

    };

    /**
     * Creates new form Menú
     */
    public Menu() {

        initComponents();

        setTitle("MENÚ");

        this.setLocationRelativeTo(null);
    }
}

```

```

    CONECTAR();

    tabla2.addColumn("RUT CLIENTE");

    tabla2.addColumn("NOMBRE CLIENTE");

    this.TABLA_DATOS2.setModel(tabla2);

    TABLA_DATOS2.getColumnModel().getColumn(0).setResizable(false);

    TABLA_DATOS2.getColumnModel().getColumn(1).setResizable(false);

    cargarDatos();

}

/**
 * This method is called from within the constructor to initialize the form.
 *
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 *
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();

    BTN_CLIENTES = new javax.swing.JButton();

    BTN_CUENTAS = new javax.swing.JButton();

    jLabel1 = new javax.swing.JLabel();

    BTN_CERR = new javax.swing.JButton();

    jLabel9 = new javax.swing.JLabel();

    BUS_RUTE = new javax.swing.JTextField();

    jLabel10 = new javax.swing.JLabel();

    BUS_NOME = new javax.swing.JTextField();

    jScrollPane2 = new javax.swing.JScrollPane();

    TABLA_DATOS2 = new javax.swing.JTable();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

jPanel1.setName("frame1"); // NOI18N

BTN_CLIENTES.setText("CLIENTES");

BTN_CLIENTES.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_CLIENTESActionPerformed(evt);

    }

});

BTN_CUENTAS.setText("CUENTAS");

BTN_CUENTAS.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_CUENTASActionPerformed(evt);

    }

});

jLabel1.setText("MENÚ");

BTN_CERR.setText("CERRAR SESIÓN");

BTN_CERR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_CERRActionPerformed(evt);

    }

});

jLabel9.setText("RUT");

BUS_RUTE.setToolTipText("BUSCAR POR RUT");

BUS_RUTE.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BUS_RUTEActionPerformed(evt);

    }

});

BUS_RUTE.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

public void keyPressed(java.awt.event.KeyEvent evt) {

    BUS_RUTEKeyPressed(evt);

}

public void keyTyped(java.awt.event.KeyEvent evt) {

    BUS_RUTEKeyTyped(evt);

}

});

jLabel10.setText("NOMBRE");

BUS_NOME.setToolTipText("BUSCAR POR NOMBRE");

BUS_NOME.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BUS_NOMEActionPerformed(evt);

    }

});

BUS_NOME.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        BUS_NOMEKeyPressed(evt);

    }

    public void keyTyped(java.awt.event.KeyEvent evt) {

        BUS_NOMEKeyTyped(evt);

    }

});

TABLA_DATOS2.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

        {},

        {},

        {},

        {},

```

```
        },  
        },  
        },  
        },  
        },  
        },  
        },  
        },  
    },  
  
    new String [] {  
  
    }  
  
));  
  
TABLA_DATOS2.addMouseListener(new java.awt.event.MouseAdapter() {  
  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
  
        TABLA_DATOS2MouseClicked(evt);  
  
    }  
  
});  
  
jScrollPane2.setViewportViewView(TABLA_DATOS2);  
  
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);  
  
jPanel1.setLayout(jPanel1Layout);  
  
jPanel1Layout.setHorizontalGroup(  
  
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
.addGroup(jPanel1Layout.createSequentialGroup()  
  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
.addComponent(BTN_CERR)
```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(320, 320, 320)

.addComponent(jLabel1)

.addGap(188, 188, 188))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(43, 43, 43)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jScrollPane2)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(BUS_RUTE, javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel9))

.addGap(138, 138, 138)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(BUS_NOME, javax.swing.GroupLayout.PREFERRED_SIZE, 255,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addComponent(jLabel10)

.addGap(205, 205, 205))))

.addGap(46, 46, 46)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(205, 205, 205)

.addComponent(BTN_CLIENTES)

.addGap(89, 89, 89)

.addComponent(BTN_CUENTAS)))

.addContainerGap(57, Short.MAX_VALUE)

);

```

```

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 14, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(BTN_CUENTAS)

.addComponent(BTN_CLIENTES))

.addGap(14, 14, 14)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9)

.addComponent(jLabel10))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(BUS_NOME, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(BUS_RUTE, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 25, Short.MAX_VALUE)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 227, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(16, 16, 16)

.addComponent(BTN_CERR)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold>

private void BTN_CLIENTESActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    MantenedorClientes mant = new MantenedorClientes();

    mant.setVisible(true);

    try {

        CONEXION1.close();

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

    }

    this.dispose();
}

private void BTN_CUENTASActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    MantenedorCuentas mantc = new MantenedorCuentas();

    mantc.setVisible(true);

    try {

        CONEXION1.close();

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

```

```

    }

    this.dispose();
}

private void BUS_RUTEActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void BUS_RUTEKeyTyped(java.awt.event.KeyEvent evt) {

    /* BUS_RUTE.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            // SENTENCIA.executeQuery("SELECT NOM_CIUDDAD FROM CIUDADES");

            String cadena = (BUS_RUTE.getText());

            BUS_RUTE.setText(cadena);

            repaint();

            filtro.setRowFilter(RowFilter.regexFilter(BUS_RUTE.getText(), 0));

        }

    });

    filtro = new TableRowSorter(TABLA_DATOS.getModel());

    TABLA_DATOS.setRowSorter(filtro);*/

    // TODO add your handling code here:

}

private void BUS_NOMEActionPerformed(java.awt.event.ActionEvent evt) {

}

private void BUS_NOMEKeyTyped(java.awt.event.KeyEvent evt) {

    /* BUS_NOME.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            String cadena = (BUS_NOME.getText());

            BUS_NOME.setText(cadena);

            repaint();

```

```

        filtro.setRowFilter(RowFilter.regexFilter(BUS_NOME.getText(), 1));

    }

});

filtro = new TableRowSorter(TABLA_DATOS.getModel());

TABLA_DATOS.setRowSorter(filtro); */

}

private void BTN_CERRActionPerformed(java.awt.event.ActionEvent evt) {

    Login login = new Login();

    login.setVisible(true);

    try {

        CONEXION1.close();

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

    }

    this.dispose();

}

private void TABLA_DATOS2MouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

    try{

        TABLA_DATOS2.setRowSelectionAllowed(true);

        TABLA_DATOS2.setColumnSelectionAllowed(false);

        int fila = TABLA_DATOS2.getSelectedRow();

        String rut=((String)TABLA_DATOS2.getValueAt(fila, 0));

        String nom=((String)TABLA_DATOS2.getValueAt(fila, 1));

        Seleccionar selec = new Seleccionar(rut, nom);

        selec.setVisible(true);

        try {

            CONEXION1.close();

```

```

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

    }

    this.dispose();

} catch (Exception l1){

    JOptionPane.showMessageDialog(null, "Error en alguna parte " + l1, "Lista Clientes",JOptionPane.ERROR_MESSAGE);

}

}

private void BUS_RUTEKeyPressed(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    BUS_RUTE.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            // SENTENCIA.executeQuery("SELECT NOM_CIUADAD FROM CIUDADES");

            String cadena = (BUS_RUTE.getText());

            BUS_RUTE.setText(cadena);

            repaint();

            filtro2.setRowFilter(RowFilter.regexFilter(BUS_RUTE.getText(), 0));

        }

    });

    filtro2 = new TableRowSorter(TABLA_DATOS2.getModel());

    TABLA_DATOS2.setRowSorter(filtro2);

}

private void BUS_NOMEKeyPressed(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    BUS_NOME.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            String cadena = (BUS_NOME.getText());

```

```

        BUS_NOME.setText(cadena);

        repaint();

        filtro2.setRowFilter(RowFilter.regexFilter(BUS_NOME.getText(), 1));

    }

});

filtro2 = new TableRowSorter(TABLA_DATOS2.getModel());

TABLA_DATOS2.setRowSorter(filtro2);

}

public void cargarDatos(){

    try{

        String SQL = "SELECT RUT_CLIENTE, NOM_CLIENTE FROM CLIENTES ORDER BY RUT_CLIENTE";

        ResultSet lista2 = SENTENCIA1.executeQuery(SQL);

        Object datos2[] = new Object[2];

        tabla2.setRowCount(0);

        while (lista2.next()){

            for (int i = 0; i < 2; i++){

                datos2[i] = lista2.getObject(i+1);

            }

            tabla2.addRow(datos2);

        }

    }catch(Exception l1){

        JOptionPane.showMessageDialog(null, "Error en listar " + l1, "Listar Datos",JOptionPane.ERROR_MESSAGE);

    }

}

private void formWindowClosing(java.awt.event.WindowEvent evt) {

    try {

        CONEXION1.close();

    } catch (Exception ex) {

```

```

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

    }

}

public void CONECTAR(){

    String URL = "";

    String USERNAME="root";

    String PASSWORD="12345";

    try {

        //Class.forName("com.mysql.jdbc.Driver");

        //URL = "jdbc:mysql://localhost:3306/" + NOMBRE_BD;

        Class.forName("com.mysql.jdbc.Driver");

        URL = "jdbc:mysql://localhost:3306/contabilidad";

        CONEXION1 = DriverManager.getConnection(URL,USERNAME,PASSWORD);

        SENTENCIA1 = CONEXION1.createStatement();

    }

    catch(Exception e1){

        JOptionPane.showMessageDialog(null,"Error de conexion "+e1, "Conexion", JOptionPane.ERROR_MESSAGE);

    }

}

/**

 * @param args the command line arguments

 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

    */

```

```

try {

    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

        if ("Nimbus".equals(info.getName())) {

            javax.swing.UIManager.setLookAndFeel(info.getClassName());

            break;

        }

    }

} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

    java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

    java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Menu().setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JButton BTN_CERR;

private javax.swing.JButton BTN_CLIENTES;

```

```

private javax.swing.JButton BTN_CUENTAS;

private javax.swing.JTextField BUS_NOME;

private javax.swing.JTextField BUS_RUTE;

private javax.swing.JTable TABLA_DATOS2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane2;

// End of variables declaration
}

```

Mantenedor de Clientes

```

import java.awt.Frame;

import java.awt.event.KeyAdapter;

import javax.swing.RowFilter;

import java.awt.event.KeyEvent;

import java.sql.Statement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.sql.*;

import java.util.logging.*;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.swing.table.TableRowSorter;

```

```

import javax.swing.JFrame;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author labinf10
 */
public class MantenedorClientes extends javax.swing.JFrame {

    private Statement SENTENCIA;

    private Connection CONEXION;

    public String USERNAME="root";

    public String PASSWORD="12345";

    private TableRowSorter filtro;

    int sw =0;

    DefaultTableModel tabla = new DefaultTableModel();

    public String ciudad;

    /**
     * Creates new form MantenedorClientes
     */
    public MantenedorClientes(){

        initComponents();

        this.setLocationRelativeTo(null);

        setTitle("MANTENEDOR DE CLIENTES");

        deshabilitar();

        CONECTAR();
    }

```

```

        tabla.addColumn("RUT");

        tabla.addColumn("NOMBRE");

        tabla.addColumn("DIRECCION");

        tabla.addColumn("CIUDAD");

        tabla.addColumn("GIRO");

        tabla.addColumn("RUT REP. LEGAL");

        tabla.addColumn("NOMBRE REP. LEGAL");

        tabla.addColumn("FONO");

        tabla.addColumn("CORREO");

        this.TABLA_DATOS.setModel(tabla);

        cargarDatos();

        cargarCombo();

        TABLA_DATOS.getColumnModel().getColumn(0).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(1).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(2).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(3).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(4).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(5).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(6).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(7).setResizable(false);

        TABLA_DATOS.getColumnModel().getColumn(8).setResizable(false);

        BTN_MODIFICAR.disable();

        BTN_ELIMINAR.disable();

    }

    public MantenedorClientes(String ciu){

        this.ciudad = ciu;

    }

    public void deshabilitar(){

```

```

    TXT_RUTE.disable();

    TXT_NOME.disable();

    TXT_CIUDAD.disable();

    TXT_DIRECCION.disable();

    TXT_GIRO.disable();

    TXT_RUTREPL.disable();

    TXT_NOMREPL.disable();

    CMB_CIUDAD.disable();

    TXT_FONO.disable();

    TXT_CORREO.disable();
}

public void cargarDatos(){

    try{

        String SQL = "SELECT * FROM CLIENTES";

        ResultSet lista = SENTENCIA.executeQuery(SQL);

        Object datos[] = new Object[9];

        tabla.setRowCount(0);

        while (lista.next()){

            for (int i = 0; i < 9; i++){

                datos[i] = lista.getObject(i+1);

            }

            tabla.addRow(datos);

        }

    }catch(Exception l1){

        JOptionPane.showMessageDialog(null, "Error en listar " + l1, "Listar Datos",JOptionPane.ERROR_MESSAGE);

    }

}

/**

```

```

* This method is called from within the constructor to initialize the form.

* WARNING: Do NOT modify this code. The content of this method is always

* regenerated by the Form Editor.

*/

```

```
@SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```

private void initComponents() {

    jPanel1 = new javax.swing.JPanel();

    jLabel1 = new javax.swing.JLabel();

    BUS_RUTE = new javax.swing.JTextField();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jLabel4 = new javax.swing.JLabel();

    jLabel5 = new javax.swing.JLabel();

    jLabel6 = new javax.swing.JLabel();

    jLabel7 = new javax.swing.JLabel();

    jLabel8 = new javax.swing.JLabel();

    TXT_RUTE = new javax.swing.JTextField();

    TXT_NOME = new javax.swing.JTextField();

    TXT_CIUDAD = new javax.swing.JTextField();

    TXT_DIRECCION = new javax.swing.JTextField();

    TXT_GIRO = new javax.swing.JTextField();

    TXT_RUTREPL = new javax.swing.JTextField();

    TXT_NOMREPL = new javax.swing.JTextField();

    jScrollPane2 = new javax.swing.JScrollPane();

    TABLA_DATOS = new javax.swing.JTable();

    BUS_NOME = new javax.swing.JTextField();

    CMB_CIUDAD = new javax.swing.JComboBox();

```

```

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

TXT_FONO = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

TXT_CORREO = new javax.swing.JTextField();

BTN_ACEPTAR = new javax.swing.JButton();

BTN_ELIMINAR = new javax.swing.JButton();

BTN_AGREGAR = new javax.swing.JButton();

BTN_VOLVER = new javax.swing.JButton();

BTN_CANCELAR = new javax.swing.JButton();

BTN_MODIFICAR = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setMinimumSize(new java.awt.Dimension(900, 700));

setPreferredSize(new java.awt.Dimension(900, 700));

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {

    public void windowClosing(java.awt.event.WindowEvent evt) {

        formWindowClosing(evt);

    }

});

getContentPane().setLayout(null);

jPanel1.setToolTipText("");

jPanel1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

jPanel1.setMinimumSize(new java.awt.Dimension(900, 700));

jPanel1.setName(""); // NOI18N

jPanel1.setPreferredSize(new java.awt.Dimension(900, 700));

jLabel1.setText("ADMINISTRADOR DE CLIENTES");

```

```

BUS_RUTE.setToolTipText("BUSCAR POR RUT");

BUS_RUTE.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BUS_RUTEActionPerformed(evt);

    }

});

BUS_RUTE.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        BUS_RUTEKeyTyped(evt);

    }

});

jLabel2.setText("RUT EMPRESA");

jLabel3.setText("NOMBRE EMPRESA");

jLabel4.setText("CIUDAD");

jLabel5.setText("DIRECCIÓN");

jLabel6.setText("GIRO");

jLabel7.setText("RUT REP. LEGAL");

jLabel8.setText("NOMBRE REP. LEGAL");

TXT_RUTE.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_RUTEActionPerformed(evt);

    }

});

TXT_RUTE.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusLost(java.awt.event.FocusEvent evt) {

        TXT_RUTEFocusLost(evt);

    }

});

```

```

TXT_RUTE.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_RUTEKeyTyped(evt);

    }

});

TXT_NOME.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_NOMEActionPerformed(evt);

    }

});

TXT_CIUADAD.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_CIUADADActionPerformed(evt);

    }

});

TXT_DIRECCION.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_DIRECCIONActionPerformed(evt);

    }

});

TXT_GIRO.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_GIROActionPerformed(evt);

    }

});

TXT_RUTREPL.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_RUTREPLActionPerformed(evt);

```

```

    }

});

TXT_RUTREPL.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusLost(java.awt.event.FocusEvent evt) {

        TXT_RUTREPLFocusLost(evt);

    }

});

TXT_RUTREPL.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_RUTREPLKeyTyped(evt);

    }

});

TXT_NOMREPL.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_NOMREPLActionPerformed(evt);

    }

});

TABLA_DATOS.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

    },

    new String [] {

    }

));

TABLA_DATOS.setColumnSelectionAllowed(true);

TABLA_DATOS.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        TABLA_DATOSMouseClicked(evt);

    }

});

```

```

    }

});

jScrollPane2.setViewportViewView(TABLA_DATOS);

TABLA_DATOS.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelectionModel.SINGLE_INTERVAL_SELECTION);

BUS_NOME.setToolTipText("BUSCAR POR NOMBRE");

BUS_NOME.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BUS_NOMEActionPerformed(evt);

    }

});

BUS_NOME.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        BUS_NOMEKeyTyped(evt);

    }

});

CMB_CIUADAD.setModel(new javax.swing.DefaultComboBoxModel(new String[] { " " }));

jLabel9.setText("RUT");

jLabel10.setText("NOMBRE");

jLabel11.setText("FONO");

TXT_FONO.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_FONOActionPerformed(evt);

    }

});

TXT_FONO.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_FONOKeyTyped(evt);

    }

});

```

```

});

jLabel12.setText("CORREO");

TXT_CORREO.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_CORREOActionPerformed(evt);

    }

});

TXT_CORREO.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusLost(java.awt.event.FocusEvent evt) {

        TXT_CORREOFocusLost(evt);

    }

});

BTN_ACEPTAR.setText("ACEPTAR");

BTN_ACEPTAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_ACEPTARActionPerformed(evt);

    }

});

BTN_ELIMINAR.setText("ELIMINAR");

BTN_ELIMINAR.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        BTN_ELIMINARMouseClicked(evt);

    }

});

BTN_ELIMINAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_ELIMINARActionPerformed(evt);

    }

});

```

```

});

BTN_AGREGAR.setText("AGREGAR");

BTN_AGREGAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_AGREGARActionPerformed(evt);

    }

});

BTN_VOLVER.setText("VOLVER");

BTN_VOLVER.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_VOLVERActionPerformed(evt);

    }

});

BTN_CANCELAR.setText("CANCELAR");

BTN_CANCELAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_CANCELARActionPerformed(evt);

    }

});

BTN_MODIFICAR.setText("MODIFICAR");

BTN_MODIFICAR.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

BTN_MODIFICAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_MODIFICARActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

```

```

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(341, 341, 341)

.addComponent(jLabel1)

.addGap(300, 408, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(33, 33, 33)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel4)

.addComponent(jLabel5)

.addComponent(jLabel6)

.addComponent(jLabel7)

.addComponent(jLabel8)

.addComponent(jLabel11)

.addComponent(jLabel12))

.addGap(21, 21, 21)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(TXT_GIRO)

.addComponent(TXT_RUTREPL)

.addComponent(TXT_NOMREPL)

.addComponent(TXT_NOME)

```

```

        .addComponent(TXT_RUTE)

        .addComponent(TXT_DIRECCION)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

        .addComponent(TXT_CIUADAD, javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(CMB_CIUADAD, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(232, 232, 232))

        .addComponent(TXT_FONO)

        .addComponent(TXT_CORREO)))

        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGap(9, 9, 9)

        .addComponent(jLabel9)

        .addGap(91, 91, 91)

        .addComponent(jLabel10))

        .addComponent(jScrollPane2)))

        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGap(29, 29, 29)

        .addComponent(BUS_RUTE, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(BUS_NOME, javax.swing.GroupLayout.PREFERRED_SIZE, 255, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(BTN_ACEPTAR, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(BTN_CANCELAR, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(BTN_VOLVER, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel1Layout.createSequentialGroup())

```

```

        .addGap(10, 10, 10)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(BTN_AGREGAR, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_MODIFICAR, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_ELIMINAR, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE))))

        .addGap(64, 64, 64))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(4, 4, 4)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel9)

    .addComponent(jLabel10))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(BUS_NOME, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(BUS_RUTE, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(22, 22, 22)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 226, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(35, 35, 35)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(TXT_RUTE, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_NOME, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_CIUDAD, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(CMB_CIUDAD, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_DIRECCION, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_GIRO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_RUTREPL, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_NOMREPL, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_FONO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 23, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_CORREO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addContainerGap(84, Short.MAX_VALUE))

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66, Short.MAX_VALUE)

        .addComponent(BTN_AGREGAR)

        .addGap(17, 17, 17)

        .addComponent(BTN_MODIFICAR)

        .addGap(17, 17, 17)

        .addComponent(BTN_ELIMINAR)

        .addGap(187, 187, 187)

        .addComponent(BTN_ACEPTAR)

        .addGap(7, 7, 7)

        .addComponent(BTN_CANCELAR)

        .addGap(117, 117, 117)

        .addComponent(BTN_VOLVER)

        .addGap(73, 73, 73))))

);

```

```

BUS_RUTE.getAccessibleContext().setAccessibleName("");

BTN_MODIFICAR.getAccessibleContext().setAccessibleDescription("");

getContentPane().add(jPanel1);

jPanel1.setBounds(0, 0, 900, 700);

jPanel1.getAccessibleContext().setAccessibleName("");

pack();

} // </editor-fold>

private void BTN_AGREGARActionPerformed(java.awt.event.ActionEvent evt) {

    TXT_RUTE.setText("");

    TXT_NOME.setText("");

    TXT_CIUADAD.setText("");

    CMB_CIUADAD.setSelectedIndex(0);

    TXT_DIRECCION.setText("");

    TXT_GIRO.setText("");

    TXT_RUTREPL.setText("");

    TXT_NOMREPL.setText("");

    TXT_FONO.setText("");

    TXT_CORREO.setText("");

    BTN_MODIFICAR.disable();

    habilitar();

    focusear();

    // TODO add your handling code here:

}

private void BTN_MODIFICARActionPerformed(java.awt.event.ActionEvent evt) {

    // int fila = TABLA_DATOS.getSelectedRow();

    // String RUT = ((String)TABLA_DATOS.getValueAt(fila, 0)).trim();

    sw = 1;

```

```

    habilitar();

    focusear();

    TXT_NOME.requestFocus();

    TXT_RUTE.disable();

    // TODO add your handling code here:

}

private void BTN_ELIMINARActionPerformed(java.awt.event.ActionEvent evt) {

    int fila = TABLA_DATOS.getSelectedRow();

    String RUT = ((String)TABLA_DATOS.getValueAt(fila, 0)).trim();

    int reply = JOptionPane.showConfirmDialog(null, "Está Seguro/ a de eliminar? ", "ELIMINAR", JOptionPane.YES_NO_OPTION);

    if (reply == JOptionPane.YES_OPTION) {

        String SQL = "DELETE FROM CLIENTES WHERE RUT_CLIENTE = '"+RUT+"'";

        try {

            SENTENCIA.executeUpdate(SQL);

            JOptionPane.showMessageDialog(null, "Empresa eliminada");

        } catch (SQLException ex) {

            JOptionPane.showMessageDialog(null, "Error en ELIMINAR " + ex, "ELIMINAR",JOptionPane.ERROR_MESSAGE);

        }

        cargarDatos();

        TXT_RUTE.setText("");

        TXT_NOME.setText("");

        TXT_CIUADAD.setText("");

        CMB_CIUADAD.setSelectedIndex(0);

        TXT_DIRECCION.setText("");

        TXT_GIRO.setText("");

        TXT_RUTREPL.setText("");

        TXT_NOMREPL.setText("");
    }
}

```

```

        TXT_FONO.setText("");

        TXT_CORREO.setText("");

        habilitar();

        focusear();

        deshabilitar();

    }

    else {

        TXT_RUTE.setText("");

        TXT_NOME.setText("");

        TXT_CIUADAD.setText("");

        CMB_CIUADAD.setSelectedIndex(0);

        TXT_DIRECCION.setText("");

        TXT_GIRO.setText("");

        TXT_RUTREPL.setText("");

        TXT_NOMREPL.setText("");

        TXT_FONO.setText("");

        TXT_CORREO.setText("");

        habilitar();

        focusear();

    }

}

private void BUS_RUTEActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void TXT_RUTEActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void TXT_NOMEActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void TXT_DIRECCIONActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void TXT_GIROActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void TXT_RUTREPLActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void TXT_NOMREPLActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void BTN_VOLVERActionPerformed(java.awt.event.ActionEvent evt) {

    Menu menu = new Menu();

    menu.setVisible(true);

    try {

        CONEXION.close();

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, "Error al cerrar bd " + ex, "BASE DE DATOS",JOptionPane.ERROR_MESSAGE);

    }

    // Menu().setVisible(true);

}

this.dispose();

```

```

/*try {

    CONEXION.close();

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

}

System.exit(0);*/
}

private void BTN_ACEPTARActionPerformed(java.awt.event.ActionEvent evt) {

    String RUT, NOM, DIR, CIU, GIRO, RUTR, NOMR, FONO, CORREO, INSERTAR, MODIFICAR;

    RUT = TXT_RUTE.getText();

    NOM = TXT_NOME.getText();

    CIU = (String) CMB_CIUADAD.getSelectedItemAt();

    String ciu2 = (String)CMB_CIUADAD.getItemAt(0);

    if (CIU == ciu2){

        CIU = TXT_CIUADAD.getText();

    }

    DIR = TXT_DIRECCION.getText();

    GIRO = TXT_GIRO.getText();

    RUTR = TXT_RUTREPL.getText();

    NOMR = TXT_NOMREPL.getText();

    FONO = TXT_FONO.getText();

    FONO = "{32}" + FONO;

    CORREO = TXT_CORREO.getText();


    if(RUT.equals("") || NOM.equals("") || CIU.equals("") || DIR.equals("") || GIRO.equals("") || RUTR.equals("") || NOMR.equals("") ||
FONO.equals("") || CORREO.equals("")){

        JOptionPane.showMessageDialog(null, "Debe llenar todos los campos", "Error al intentar guardar.",
JOptionPane.INFORMATION_MESSAGE);

    }

```

```

else{

    if (sw == 0) {

        INSERTAR = "INSERT INTO CLIENTES (RUT_CLIENTE, NOM_CLIENTE, DIRECCION, CIUDAD, GIRO, RUT_REP, NOM_REP, FONO, CORREO)
VALUES"

        + "(" + RUT + "," + NOM + "," + DIR + "," + CIU + "," + GIRO + "," + RUTR + "," + NOMR + "," + FONO + "," + CORREO + ")";

    try {

        SENTENCIA.executeUpdate(INSERTAR);

        JOptionPane.showMessageDialog(null, "Datos Guardados");

    } catch (SQLException e3) {

        JOptionPane.showMessageDialog(null, "Error: Ya existe la empresa.", "Conexion", JOptionPane.ERROR_MESSAGE);

    }

    cargarDatos();

    TXT_RUTE.setText("");

    TXT_NOME.setText("");

    TXT_CIUUDAD.setText("");

    TXT_DIRECCION.setText("");

    TXT_GIRO.setText("");

    TXT_RUTREPL.setText("");

    TXT_NOMREPL.setText("");

    TXT_FONO.setText("");

    TXT_CORREO.setText("");

    CMB_CIUUDAD.setSelectedIndex(0);

}

else {

    MODIFICAR = "UPDATE CLIENTES SET NOM_CLIENTE = " + NOM + ", DIRECCION = " + DIR + ", CIUDAD = " + CIU + ", "

        + "GIRO = " + GIRO + ", RUT_REP = " + RUTR + ", NOM_REP = " + NOMR + ", FONO = " + FONO + ", CORREO = " + CORREO + "
WHERE RUT_CLIENTE = " + RUT + """;

    try {

        SENTENCIA.executeUpdate(MODIFICAR);

```

```

        JOptionPane.showMessageDialog(null, "Datos Guardados");

    } catch (SQLException e3) {

        JOptionPane.showMessageDialog(null, "Error al modificar" + e3, "Conexion", JOptionPane.ERROR_MESSAGE);

    }

    cargarDatos();

    TXT_RUTE.setText("");

    TXT_NOME.setText("");

    TXT_CIUADAD.setText("");

    TXT_DIRECCION.setText("");

    TXT_GIRO.setText("");

    TXT_RUTREPL.setText("");

    TXT_NOMREPL.setText("");

    CMB_CIUADAD.setSelectedIndex(0);

    TXT_FONO.setText("");

    TXT_CORREO.setText("");

    deshabilitar();

}

}

}

private void formWindowClosing(java.awt.event.WindowEvent evt) {

    try {

        CONEXION.close();

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD",JOptionPane.ERROR_MESSAGE);

    }

}

private void TABLA_DATOSMouseClicked(java.awt.event.MouseEvent evt) {

    BTN_AGREGAR.disable();

```

```

BTN_MODIFICAR.enable();

BTN_ELIMINAR.enable();

try{

    TABLA_DATOS.setRowSelectionAllowed(true);

    TABLA_DATOS.setColumnSelectionAllowed(false);

    int fila = TABLA_DATOS.getSelectedRow();

    String RUTE = ((String)TABLA_DATOS.getValueAt(fila, 0)).trim();

    TXT_RUTE.setText((String)TABLA_DATOS.getValueAt(fila, 0));

    TXT_NOME.setText((String)TABLA_DATOS.getValueAt(fila, 1));

    // JOptionPane.showMessageDialog(null, "RUT = " + RUTE, "RUT",JOptionPane.ERROR_MESSAGE);

    // String SQL = "SELECT * FROM CLIENTES WHERE RUT_EMP = '"+RUTE+"'";

    // ResultSet item = SENTENCIA.executeQuery(SQL);

    TXT_DIRECCION.setText((String)TABLA_DATOS.getValueAt(fila, 2));

    TXT_CIUADAD.setText((String)TABLA_DATOS.getValueAt(fila, 3));

    TXT_GIRO.setText((String)TABLA_DATOS.getValueAt(fila, 4));

    TXT_RUTREPL.setText((String)TABLA_DATOS.getValueAt(fila, 5));

    TXT_NOMREPL.setText((String)TABLA_DATOS.getValueAt(fila, 6));

    TXT_FONO.setText((String)TABLA_DATOS.getValueAt(fila, 7));

    TXT_CORREO.setText((String)TABLA_DATOS.getValueAt(fila, 8));

}catch (Exception e3){

    JOptionPane.showMessageDialog(null, "Error en encontrar " + e3, "Buscar tipo",JOptionPane.ERROR_MESSAGE);

}

// TODO add your handling code here:

}

private void BTN_ELIMINARMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

}

private void BTN_CANCELARActionPerformed(java.awt.event.ActionEvent evt) {

```

```

deshabilitar();

TXT_RUTE.setText("");

TXT_NOME.setText("");

TXT_CIUADAD.setText("");

TXT_DIRECCION.setText("");

TXT_GIRO.setText("");

TXT_RUTREPL.setText("");

TXT_NOMREPL.setText("");

TXT_FONO.setText("");

TXT_CORREO.setText("");

// TODO add your handling code here:

}

private void BUS_NOMEActionPerformed(java.awt.event.ActionEvent evt) {

}

private void BUS_RUTEKeyTyped(java.awt.event.KeyEvent evt) {

    BUS_RUTE.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            String cadena = (BUS_RUTE.getText());

            BUS_RUTE.setText(cadena);

            repaint();

            filtro.setRowFilter(RowFilter.regexFilter(BUS_RUTE.getText(), 0));

        }

    });

    filtro = new TableRowSorter(TABLA_DATOS.getModel());

    TABLA_DATOS.setRowSorter(filtro);

    // TODO add your handling code here:

```

```
}
```

```
private void BUS_NOMEKeyTyped(java.awt.event.KeyEvent evt) {

    BUS_NOME.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            String cadena = (BUS_NOME.getText());

            BUS_NOME.setText(cadena);

            repaint();

            filtro.setRowFilter(RowFilter.regexFilter(BUS_NOME.getText(), 1));

        }

    });

    filtro = new TableRowSorter(TABLA_DATOS.getModel());

    TABLA_DATOS.setRowSorter(filtro);

}
```

```
private void TXT_CIUDADAActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}
```

```
private void TXT_FONOActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}
```

```
private void TXT_CORREOActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}
```

```
private void TXT_RUTEKeyTyped(java.awt.event.KeyEvent evt) {
```

```

// TODO add your handling code here:

String Caracteres = TXT_RUTE.getText();

if(Caracteres.length()>=12){

    evt.consume();

}

}

private void TXT_FONOKeyTyped(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    char character = evt.getKeyChar();

    if(((character < '0') ||

        (character > '9')) &&

        (character != '\b'))

    {

        evt.consume();

    }

    String Caracteres = TXT_FONO.getText();

    if(Caracteres.length()>=12){

        evt.consume();

    }

}

private void TXT_RUTREPLKeyTyped(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    String Caracteres = TXT_RUTE.getText();

    if(Caracteres.length()>=12){

        evt.consume(); // ignorar el evento de teclado

    }

}

```

```
}
```

```
private void TXT_RUTEFocusLost(java.awt.event.FocusEvent evt) {

    // TODO add your handling code here:

    boolean validacion = false;

    try {

        String rut = TXT_RUTE.getText();

        rut = rut.toUpperCase();

        rut = rut.replace(".", "");

        rut = rut.replace("-", "");

        int rutAux = Integer.parseInt(rut.substring(0, rut.length() - 1));

        char dv = rut.charAt(rut.length() - 1);

        int m = 0, s = 1;

        for (; rutAux != 0; rutAux /= 10) {

            s = (s + rutAux % 10 * (9 - m++ % 6)) % 11;

        }

        if (dv == (char) (s != 0 ? s + 47 : 75)) {

            validacion = true;

        }

        if (validacion == false){

            JOptionPane.showMessageDialog(null,"Rut no valido.", "ERROR", JOptionPane.ERROR_MESSAGE);

            TXT_RUTE.setText("");

            TXT_RUTE.requestFocus();

        }else{

            rut = rut.substring(0,rut.length()-1);

            rut = rut + "-" + dv;

            TXT_RUTE.setText(rut);

        }

    }

}
```

```

    } catch (java.lang.NumberFormatException e) {

    } catch (Exception e) {

    }

}

private void TXT_RUTREPLFocusLost(java.awt.event.FocusEvent evt) {

    // TODO add your handling code here:

    boolean validacion = false;

    try {

        String rut = TXT_RUTREPL.getText();

        rut = rut.toUpperCase();

        rut = rut.replace(".", "");

        rut = rut.replace("-", "");

        int rutAux = Integer.parseInt(rut.substring(0, rut.length() - 1));

        char dv = rut.charAt(rut.length() - 1);

        int m = 0, s = 1;

        for (; rutAux != 0; rutAux /= 10) {

            s = (s + rutAux % 10 * (9 - m++ % 6)) % 11;

        }

        if (dv == (char) (s != 0 ? s + 47 : 75)) {

            validacion = true;

        }

        if (validacion == false){

            JOptionPane.showMessageDialog(null,"Rut no valido.", "ERROR", JOptionPane.ERROR_MESSAGE);

            TXT_RUTREPL.setText("");

            TXT_RUTREPL.requestFocus();

        }else{

```

```

        rut = rut.substring(0,rut.length()-1);

        rut = rut + "-" + dv;

        TXT_RUTREPL.setText(rut);

    }

} catch (java.lang.NumberFormatException e) {

} catch (Exception e) {

}

}

private void TXT_CORREOFocusLost(java.awt.event.FocusEvent evt) {

    /* // TODO add your handling code here:

    Pattern pattern = Pattern

        .compile("^[_A-Za-z0-9-\\]+(\\.[_A-Za-z0-9-]+)*@"

            + "[A-Za-z0-9-]+(\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$");

    // El email a validar

    Matcher mather = pattern.matcher(TXT_CORREO.getText());

    if (mather.find() == false) {

        JOptionPane.showMessageDialog(null,"Correo no valido.", "ERROR", JOptionPane.ERROR_MESSAGE);

    }*/

}

public void habilitar(){

    TXT_RUTE.enable();

    TXT_NOME.enable();

    TXT_DIRECCION.enable();

    TXT_GIRO.enable();

    TXT_RUTREPL.enable();

```

```

    TXT_NOMREPL.enable();

    CMB_CIUADAD.enable();

    TXT_FONO.enable();

    TXT_CORREO.enable();
}

public void focusear(){

    TXT_NOME.requestFocus();

    TXT_CIUADAD.requestFocus();

    CMB_CIUADAD.requestFocus();

    TXT_DIRECCION.requestFocus();

    TXT_GIRO.requestFocus();

    TXT_RUTREPL.requestFocus();

    TXT_NOMREPL.requestFocus();

    TXT_RUTE.requestFocus();

    TXT_FONO.requestFocus();

    TXT_CORREO.requestFocus();

    TXT_RUTE.requestFocus();
}

public void cargarCombo(){

    ResultSet LISTA = null;

    try{

        LISTA = SENTENCIA.executeQuery("SELECT NOM_CIUADAD FROM CIUDADES");

        while(LISTA.next()){

            CMB_CIUADAD.addItem(LISTA.getString("NOM_CIUADAD"));

        }

    }

    catch(SQLException e2){

        JOptionPane.showMessageDialog(null,"Error en consultar"+e2,"Conexion",JOptionPane.ERROR_MESSAGE);
    }
}

```

```

    }

}

/* public void LLENAR_COMBO(){

    ResultSet LISTA = null;

    try{

        LISTA = SENTENCIA.executeQuery("SELECT NOMBRE FROM Persona");

        while(LISTA.next()){

            CMB_NOMBRE.addItem(LISTA.getString("NOMBRE"));

        }

    }

    catch(SQLException e2){

        JOptionPane.showMessageDialog(null,"Error en consultar"+e2,"Conexion",JOptionPane.ERROR_MESSAGE);

    }

} */

public void CONECTAR(){

    String URL = "";

    String USERNAME="root";

    String PASSWORD="12345";

    try {

        //Class.forName("com.mysql.jdbc.Driver");

        //URL = "jdbc:mysql://localhost:3306/" + NOMBRE_BD;

        Class.forName("com.mysql.jdbc.Driver");

        URL = "jdbc:mysql://localhost:3306/contabilidad";

        CONEXION = DriverManager.getConnection(URL,USERNAME,PASSWORD);

        SENTENCIA = CONEXION.createStatement();

    }

    catch(Exception e1){

        JOptionPane.showMessageDialog(null,"Error de conexion "+e1, "Conexion", JOptionPane.ERROR_MESSAGE);
    }
}

```

```

    }
}

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(MantenedorClientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(MantenedorClientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(MantenedorClientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(MantenedorClientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    //</editor-fold>

    /* Create and display the form */

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

```

```

        new MantenedorClientes().setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JButton BTN_ACEPTAR;

private javax.swing.JButton BTN_AGREGAR;

private javax.swing.JButton BTN_CANCELAR;

private javax.swing.JButton BTN_ELIMINAR;

private javax.swing.JButton BTN_MODIFICAR;

private javax.swing.JButton BTN_VOLVER;

private javax.swing.JTextField BUS_NOME;

private javax.swing.JTextField BUS_RUTE;

private javax.swing.JComboBox CMB_CIUDAD;

private javax.swing.JTable TABLA_DATOS;

private javax.swing.JTextField TXT_CIUDAD;

private javax.swing.JTextField TXT_CORREO;

private javax.swing.JTextField TXT_DIRECCION;

private javax.swing.JTextField TXT_FONO;

private javax.swing.JTextField TXT_GIRO;

private javax.swing.JTextField TXT_NOME;

private javax.swing.JTextField TXT_NOMREPL;

private javax.swing.JTextField TXT_RUTE;

private javax.swing.JTextField TXT_RUTREPL;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

```

```

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane2;

// End of variables declaration
}

```

Mantenedor de Cuentas

```

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.RowFilter;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableRowSorter;

```

```

/*

```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

/**

*

* @author Valentaiguer

*/

```
public class MantenedorCuentas extends javax.swing.JFrame {
```

```
    private Statement SENTENCIA;
```

```
    private Connection CONEXION;
```

```
    private TableRowSorter filtro;
```

```
    int sw = 0;
```

```
    int sw2 = 0;
```

```
    DefaultTableModel tablac = new DefaultTableModel() {
```

```
        @Override
```

```
        public boolean isCellEditable(int fil, int col) {
```

```
            return false;
```

```
        }
```

```
    };
```

```
    /**
```

```
    * Creates new form MantenedorCuentas
```

```
    */
```

```
    public MantenedorCuentas() {
```

```
        initComponents();
```

```
        setTitle("MANTENEDOR DE CUENTAS");
```

```
        deshabilitar();
```

```
        this.setLocationRelativeTo(null);
```

```

    CONECTAR();

    tablac.addColumn("CODIGO");

    tablac.addColumn("NOMBRE");

    tablac.addColumn("SUBCATEGORIA");

    tablac.addColumn("CATEGORIA");

    this.TABLA_CUENTAS.setModel(tablac);

    cargarDatos();

    cargarComboC();

    TABLA_CUENTAS.getColumnModel().getColumn(0).setResizable(false);

    TABLA_CUENTAS.getColumnModel().getColumn(1).setResizable(false);

    TABLA_CUENTAS.getColumnModel().getColumn(2).setResizable(false);

}

public void cargarDatos() {

    try {

        String SQL = "SELECT C.COD_CUENTA, C.NOM_CUENTA, S.NOM_SUBCATEGORIA, CA.NOM_CATEGORIA FROM CUENTAS C,
SUBCATEGORIAS S, CATEGORIAS CA "

        + "WHERE C.COD_SUBCATEGORIA = S.COD_SUBCATEGORIA AND CA.COD_CATEGORIA = S.COD_CATEGORIA ORDER BY
C.COD_CUENTA";

        ResultSet lista = SENTENCIA.executeQuery(SQL);

        Object datos[] = new Object[4];

        tablac.setRowCount(0);

        while (lista.next()) {

            for (int i = 0; i < 4; i++) {

                datos[i] = lista.getObject(i + 1);

            }

            tablac.addRow(datos);

        }

    } catch (Exception l1) {

        JOptionPane.showMessageDialog(null, "Error en listar " + l1, "Listar Datos", JOptionPane.ERROR_MESSAGE);
    }
}

```

```

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 *
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 *
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();

    jLabel1 = new javax.swing.JLabel();

    jScrollPane1 = new javax.swing.JScrollPane();

    TABLA_CUENTAS = new javax.swing.JTable();

    jLabel9 = new javax.swing.JLabel();

    jLabel10 = new javax.swing.JLabel();

    BUS_COD = new javax.swing.JTextField();

    BUS_NOMC = new javax.swing.JTextField();

    BTN_AGREGAR = new javax.swing.JButton();

    BTN_MODIFICAR = new javax.swing.JButton();

    BTN_ELIMINAR = new javax.swing.JButton();

    BTN_ACEPTAR = new javax.swing.JButton();

    BTN_CANCELAR = new javax.swing.JButton();

    BTN_VOLVER = new javax.swing.JButton();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jLabel4 = new javax.swing.JLabel();

    jLabel5 = new javax.swing.JLabel();

```

```

TXT_CODC = new javax.swing.JTextField();

TXT_NOMC = new javax.swing.JTextField();

CMB_CAT = new javax.swing.JComboBox();

CMB_SUBC = new javax.swing.JComboBox();

TXT_CAT = new javax.swing.JTextField();

TXT_SUBC = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setName(""); // NOI18N

jLabel1.setText("ADMINISTRADOR DE CUENTAS");

jLabel1.setName("frame3"); // NOI18N

TABLA_CUENTAS.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

        {},

        {},

        {},

        {},

        {},

        {},

        {},

        {}

    },

    new String [] {

    }

));

TABLA_CUENTAS.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        TABLA_CUENTASMouseClicked(evt);

    }

}

```

```

});

jScrollPane1.setViewportViewView(TABLA_CUENTAS);

jLabel9.setText("CÓDIGO");

jLabel10.setText("NOMBRE");

BUS_COD.setToolTipText("BUSCAR POR RUT");

BUS_COD.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BUS_CODActionPerformed(evt);

    }

});

BUS_COD.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        BUS_CODKeyTyped(evt);

    }

});

BUS_NOMC.setToolTipText("BUSCAR POR NOMBRE");

BUS_NOMC.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BUS_NOMCActionPerformed(evt);

    }

});

BUS_NOMC.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        BUS_NOMCKeyTyped(evt);

    }

});

BTN_AGREGAR.setText("AGREGAR");

BTN_AGREGAR.addActionListener(new java.awt.event.ActionListener() {

```

```

public void actionPerformed(java.awt.event.ActionEvent evt) {

    BTN_AGREGARActionPerformed(evt);

}

});

BTN_MODIFICAR.setText("MODIFICAR");

BTN_MODIFICAR.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

BTN_MODIFICAR.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        BTN_MODIFICARMouseClicked(evt);

    }

});

BTN_MODIFICAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_MODIFICARActionPerformed(evt);

    }

});

BTN_ELIMINAR.setText("ELIMINAR");

BTN_ELIMINAR.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        BTN_ELIMINARMouseClicked(evt);

    }

});

BTN_ELIMINAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_ELIMINARActionPerformed(evt);

    }

});

BTN_ACEPTAR.setText("ACEPTAR");

```

```

BTN_ACEPTAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_ACEPTARActionPerformed(evt);

    }

});

BTN_CANCELAR.setText("CANCELAR");

BTN_CANCELAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_CANCELARActionPerformed(evt);

    }

});

BTN_VOLVER.setText("VOLVER");

BTN_VOLVER.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_VOLVERActionPerformed(evt);

    }

});

jLabel2.setText("CÓDIGO ");

jLabel3.setText("NOMBRE");

jLabel4.setText("CATEGORÍA");

jLabel5.setText("SUBCATEGORÍA");

TXT_CODC.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_CODCAActionPerformed(evt);

    }

});

TXT_NOMC.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        TXT_NOMCActionPerformed(evt);
    }
});

CMB_CAT.setModel(new javax.swing.DefaultComboBoxModel(new String[] { " " }));

CMB_CAT.addItemListener(new java.awt.event.ItemListener() {

    public void itemStateChanged(java.awt.event.ItemEvent evt) {

        CMB_CATItemStateChanged(evt);

    }

});

CMB_CAT.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        CMB_CATActionPerformed(evt);

    }

});

CMB_SUBC.setModel(new javax.swing.DefaultComboBoxModel(new String[] { " " }));

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(240, 240, 240)

            .addComponent(jLabel1)

            .addGap(Short.MAX_VALUE, Short.MAX_VALUE, Short.MAX_VALUE)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(jPanel1Layout.createSequentialGroup()

                    .addGap(19, 19, 19)

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(jPanel1Layout.createSequentialGroup()

                            .addGap(19, 19, 19)

                            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel4)

.addComponent(jLabel5))

.addGap(21, 21, 21)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

    .addGroup(jPanel1Layout.createSequentialGroup())

        .addComponent(TXT_CAT, javax.swing.GroupLayout.PREFERRED_SIZE, 141, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(CMB_CAT, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    .addComponent(TXT_CODC, javax.swing.GroupLayout.PREFERRED_SIZE, 92, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanel1Layout.createSequentialGroup())

        .addComponent(TXT_SUBC, javax.swing.GroupLayout.PREFERRED_SIZE, 192, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(CMB_SUBC, javax.swing.GroupLayout.PREFERRED_SIZE, 162, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addComponent(TXT_NOMC, javax.swing.GroupLayout.PREFERRED_SIZE, 289, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(jPanel1Layout.createSequentialGroup())

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(BUS_COD, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel1Layout.createSequentialGroup())

            .addGap(11, 11, 11)

            .addComponent(jLabel9)))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(BUS_NOMC, javax.swing.GroupLayout.PREFERRED_SIZE, 255, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel1Layout.createSequentialGroup())

            .addGap(9, 9, 9)

            .addComponent(jLabel10))))

```

```

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 644, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 27, Short.MAX_VALUE)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(BTN_AGREGAR, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_MODIFICAR, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_ELIMINAR, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_ACEPTAR, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_CANCELAR, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(BTN_VOLVER, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(36, 36, 36))
    );

    jPanel1Layout.setVerticalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()

                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(jLabel1)

                    .addGap(18, 18, 18)

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                        .addComponent(jLabel9)

                        .addComponent(jLabel10))

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(jPanel1Layout.createSequentialGroup()

                            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                                .addComponent(BUS_COD, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                                .addComponent(BUS_NOMC, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                                .addGap(13, 13, 13)

```

```

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 330, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGroup(jPanel1Layout.createSequentialGroup())

        .addComponent(BTN_AGREGAR)

        .addGap(17, 17, 17)

        .addComponent(BTN_MODIFICAR)

        .addGap(17, 17, 17)

        .addComponent(BTN_ELIMINAR)

        .addGap(187, 187, 187)

        .addComponent(BTN_ACEPTAR)

        .addGap(7, 7, 7)

        .addComponent(BTN_CANCELAR)))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 32, Short.MAX_VALUE)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

            .addComponent(BTN_VOLVER)

            .addGap(28, 28, 28))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel2)

                .addComponent(TXT_CODC, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel3)

                .addComponent(TXT_NOMC, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel4)

```

```

        .addComponent(CMB_CAT, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(TXT_CAT, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(CMB_SUBC, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel5)

                .addComponent(TXT_SUBC, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

            .addGap(42, 42, 42))))

    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()

                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(0, 0, Short.MAX_VALUE)

            );

    pack();

} // </editor-fold>

private void BUS_CODActionPerformed(java.awt.event.ActionEvent evt) {

```

```

// TODO add your handling code here:

}

private void BUS_CODKeyTyped(java.awt.event.KeyEvent evt) {

    BUS_COD.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            // SENTENCIA.executeQuery("SELECT NOM_CIUADAD FROM CIUDADES");

            String cadena = (BUS_COD.getText());

            BUS_COD.setText(cadena);

            repaint();

            filtro.setRowFilter(RowFilter.regexFilter(BUS_COD.getText(), 0));

        }

    });

    filtro = new TableRowSorter(TABLA_CUENTAS.getModel());

    TABLA_CUENTAS.setRowSorter(filtro);

    // TODO add your handling code here:

}

private void BUS_NOMCActionPerformed(java.awt.event.ActionEvent evt) {

}

private void BUS_NOMCKeyTyped(java.awt.event.KeyEvent evt) {

    BUS_NOMC.addKeyListener(new KeyAdapter() {

        public void keyReleased(final KeyEvent e) {

            String cadena = (BUS_NOMC.getText());

            BUS_NOMC.setText(cadena);

            repaint();

            filtro.setRowFilter(RowFilter.regexFilter(BUS_NOMC.getText(), 1));

        }

    });

    filtro = new TableRowSorter(TABLA_CUENTAS.getModel());

```

```

    TABLA_CUENTAS.setRowSorter(filtro);
}

private void BTN_AGREGARActionPerformed(java.awt.event.ActionEvent evt) {

    TXT_CODC.setText("");

    TXT_NOMC.setText("");

    TXT_CAT.setText("");

    TXT_SUBC.setText("");

    CMB_CAT.setSelectedIndex(0);

    CMB_SUBC.setSelectedIndex(0);

    TXT_CAT.disable();

    TXT_SUBC.disable();

    TXT_CODC.disable();

    habilitar();

    focusear();

    // TODO add your handling code here:
}

private void BTN_MODIFICARActionPerformed(java.awt.event.ActionEvent evt) {

    // int fila = TABLA_DATOS.getSelectedRow();

    // String RUT = ((String)TABLA_DATOS.getValueAt(fila, 0)).trim();

    sw = 1;

    TXT_NOMC.enable();

    CMB_CAT.disable();

    CMB_SUBC.disable();

    TXT_NOMC.requestFocus();

    TXT_CODC.disable();

    TXT_CAT.disable();

    TXT_SUBC.disable();

```

```

// TODO add your handling code here:

}

private void BTN_ELIMINARMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

}

private void BTN_ELIMINARActionPerformed(java.awt.event.ActionEvent evt) {

    int fila = TABLA_CUENTAS.getSelectedRow();

    String CODC = ((String) TABLA_CUENTAS.getValueAt(fila, 0)).trim();

    int reply = JOptionPane.showConfirmDialog(null, "Está Seguro/ a de eliminar? ", "ELIMINAR", JOptionPane.YES_NO_OPTION);

    if (reply == JOptionPane.YES_OPTION) {

        String SQL = "DELETE FROM CUENTAS WHERE COD_CUENTA = " + CODC + "";

        try {

            SENTENCIA.executeUpdate(SQL);

            JOptionPane.showMessageDialog(null, "Cuenta eliminada");

        } catch (SQLException ex) {

            JOptionPane.showMessageDialog(null, "Error en ELIMINAR " + ex, "ELIMINAR", JOptionPane.ERROR_MESSAGE);

        }

        cargarDatos();

        TXT_CODC.setText("");

        TXT_NOMC.setText("");

        TXT_CAT.setText("");

        TXT_SUBC.setText("");

        CMB_CAT.setSelectedIndex(0);

        CMB_SUBC.setSelectedIndex(0);

        habilitar();

        focusear();

        deshabilitar();

    } else {

```

```

    TXT_CODC.setText("");

    TXT_NOMC.setText("");

    TXT_CAT.setText("");

    TXT_SUBC.setText("");

    CMB_CAT.setSelectedIndex(0);

    CMB_SUBC.setSelectedIndex(0);

    habilitar();

    focusear();

}

}

private void BTN_ACEPTARActionPerformed(java.awt.event.ActionEvent evt) {

    String CODC, CODCA, CODIGO = null, NOMC, CAT, CATE, SUBC, SUBCA, SUBCATE = null, INSERTAR, MODIFICAR, ULTIMO;

    int CODE;

    CODCA = TXT_CODC.getText().trim();

    NOMC = TXT_NOMC.getText();

    CAT = (String) CMB_CAT.getSelectedItem();

    SUBC = (String) CMB_SUBC.getSelectedItem();

    if (NOMC.equals("") || CAT.equals("") || SUBC.equals("")) {

        JOptionPane.showMessageDialog(null, "Debe llenar todos los campos", "Error al intentar guardar.",
JOptionPane.INFORMATION_MESSAGE);

    } else {

        if (sw == 0) {

            try {

                SUBCA = "SELECT COD_SUBCATEGORIA FROM SUBCATEGORIAS WHERE NOM_SUBCATEGORIA = '" + SUBC + "'";

                ResultSet SUBCAT = SENTENCIA.executeQuery(SUBCA);

                while (SUBCAT.next()) {

                    SUBCATE = SUBCAT.getString("COD_SUBCATEGORIA");

                }

            } catch (SQLException e3) {

```

```

JOptionPane.showMessageDialog(null, "Error al en la sentencia SUBCA" + e3, "Conexion", JOptionPane.ERROR_MESSAGE);

}

String SUBA = SUBCATE;

try {

    ULTIMO = "SELECT COD_CUENTA FROM CUENTAS CU, CATEGORIAS CA, SUBCATEGORIAS S "

        + "WHERE S.COD_SUBCATEGORIA = '" + SUBA + "'" AND CU.COD_SUBCATEGORIA = S.COD_SUBCATEGORIA AND
CA.COD_CATEGORIA = S.COD_CATEGORIA";

    ResultSet NUMERO = SENTENCIA.executeQuery(ULTIMO);

    while (NUMERO.next()) {

        CODIGO = NUMERO.getString("COD_CUENTA");

    }

    if (CODIGO == null) {

        CODIGO = SUBCATE + ".0";

    }

    CODIGO = CODIGO.substring(CODIGO.lastIndexOf(".") + 1, CODIGO.length());

    CODE = Integer.parseInt(CODIGO);

    CODE = CODE + 10;

    CODIGO = Integer.toString(CODE);

} catch (SQLException e3) {

    JOptionPane.showMessageDialog(null, "Error en la sentencia ULTIMO" + e3, "Conexion", JOptionPane.ERROR_MESSAGE);

}

CODC = SUBCATE + "." + CODIGO;

TXT_CODC.setText(CODC);

INSERTAR = "INSERT INTO CUENTAS (COD_CUENTA, COD_SUBCATEGORIA, NOM_CUENTA) VALUES"

    + "(" + CODC + "," + SUBCATE + "," + NOMC + ")";

try {

    SENTENCIA.executeUpdate(INSERTAR);

    JOptionPane.showMessageDialog(null, "Datos Guardados");

```

```

    } catch (SQLException e3) {

        JOptionPane.showMessageDialog(null, "Error: Ya existe la cuenta.", "Conexion", JOptionPane.ERROR_MESSAGE);

    }

    cargarDatos();

    TXT_CODC.setText("");

    TXT_NOMC.setText("");

    TXT_CAT.setText("");

    TXT_SUBC.setText("");

    CMB_CAT.setSelectedIndex(0);

    CMB_SUBC.setSelectedIndex(0);

    deshabilitar();

} else {

    MODIFICAR = "UPDATE CUENTAS SET NOM_CUENTA = '" + NOMC + "' WHERE COD_CUENTA = '" + CODCA + "'";

    try {

        SENTENCIA.executeUpdate(MODIFICAR);

        JOptionPane.showMessageDialog(null, "Datos Guardados");

    } catch (SQLException e3) {

        JOptionPane.showMessageDialog(null, "Error al modificar.", "Conexion", JOptionPane.ERROR_MESSAGE);

    }

    cargarDatos();

    TXT_CODC.setText("");

    TXT_NOMC.setText("");

    TXT_CAT.setText("");

    TXT_SUBC.setText("");

    CMB_CAT.setSelectedIndex(0);

    CMB_SUBC.setSelectedIndex(0);

    deshabilitar();

    sw = 0;

```

```

    }

}

}

private void BTN_CANCELARActionPerformed(java.awt.event.ActionEvent evt) {

    TXT_CODC.setText("");

    TXT_NOMC.setText("");

    TXT_CAT.setText("");

    TXT_SUBC.setText("");

    CMB_CAT.setSelectedIndex(0);

    CMB_SUBC.setSelectedIndex(0);

    deshabilitar();

    // TODO add your handling code here:

}

private void BTN_VOLVERActionPerformed(java.awt.event.ActionEvent evt) {

    Menu menu = new Menu();

    menu.setVisible(true);

    try {

        CONEXION.close();

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD", JOptionPane.ERROR_MESSAGE);

    }

    this.dispose();

}

private void TXT_CODCAActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void TXT_NOMCAActionPerformed(java.awt.event.ActionEvent evt) {

```

```

// TODO add your handling code here:
}

private void TABLA_CUENTASMouseClicked(java.awt.event.MouseEvent evt) {

    BTN_AGREGAR.disable();

    BTN_MODIFICAR.enable();

    BTN_ELIMINAR.enable();

    try {

        TABLA_CUENTAS.setRowSelectionAllowed(true);

        TABLA_CUENTAS.setColumnSelectionAllowed(false);

        int fila = TABLA_CUENTAS.getSelectedRow();

        String CODC = ((String) TABLA_CUENTAS.getValueAt(fila, 0)).trim();

        TXT_CODC.setText((String) TABLA_CUENTAS.getValueAt(fila, 0));

        TXT_NOMC.setText((String) TABLA_CUENTAS.getValueAt(fila, 1));

        // JOptionPane.showMessageDialog(null, "RUT = " + RUTE, "RUT",JOptionPane.ERROR_MESSAGE);

        // String SQL = "SELECT * FROM CLIENTES WHERE RUT_EMP = '"+RUTE+"'";

        // ResultSet item = SENTENCIA.executeQuery(SQL);

        TXT_SUBC.setText((String) TABLA_CUENTAS.getValueAt(fila, 2));

        String SUBC = TXT_SUBC.getText();

        String SQL = "SELECT C.NOM_CATEGORIA FROM CATEGORIAS C, SUBCATEGORIAS S"

            + " WHERE C.COD_CATEGORIA = S.COD_CATEGORIA AND S.COD_SUBCATEGORIA = (SELECT A.COD_SUBCATEGORIA FROM"
SUBCATEGORIAS A WHERE A.NOM_SUBCATEGORIA= '" + SUBC + "'";

        ResultSet lista = SENTENCIA.executeQuery(SQL);

        while (lista.next()) {

            String res = lista.getString("NOM_CATEGORIA");

            TXT_CAT.setText(res);

        }

    } catch (Exception e3) {

        JOptionPane.showMessageDialog(null, "Error en encontrar " + e3, "Buscar tipo", JOptionPane.ERROR_MESSAGE);

    }
}

```

```

        // TODO add your handling code here:

    }

    private void CMB_CATItemStateChanged(java.awt.event.ItemEvent evt) {

        String CAT = (String) CMB_CAT.getSelectedItem();

        if ("ACTIVO".equals(CAT)) {

            CMB_SUBC.removeAllItems();

            cargarComboS(1);

        }

        if ("PASIVO".equals(CAT)) {

            CMB_SUBC.removeAllItems();

            cargarComboS(2);

        }

        if ("PATRIMONIO".equals(CAT)) {

            CMB_SUBC.removeAllItems();

            cargarComboS(3);

        }

        if ("RESULTADO".equals(CAT)) {

            CMB_SUBC.removeAllItems();

            cargarComboS(4);

        }

        // TODO add your handling code here:

    }

    private void BTN_MODIFICARMouseClicked(java.awt.event.MouseEvent evt) {

        sw = 1;

        // TODO add your handling code here:

    }

    private void CMB_CATActionPerformed(java.awt.event.ActionEvent evt) {

```

```

// TODO add your handling code here:

}

public void CONECTAR() {

    String URL = "";

    String USERNAME = "root";

    String PASSWORD = "12345";

    try {

        //Class.forName("com.mysql.jdbc.Driver");

        //URL = "jdbc:mysql://localhost:3306/" + NOMBRE_BD;

        Class.forName("com.mysql.jdbc.Driver");

        URL = "jdbc:mysql://localhost:3306/contabilidad";

        CONEXION = DriverManager.getConnection(URL, USERNAME, PASSWORD);

        SENTENCIA = CONEXION.createStatement();

    } catch (Exception e1) {

        JOptionPane.showMessageDialog(null, "Error de conexion " + e1, "Conexion", JOptionPane.ERROR_MESSAGE);

    }

}

public void cargarComboC() {

    ResultSet LISTA = null;

    try {

        LISTA = SENTENCIA.executeQuery("SELECT NOM_CATEGORIA FROM CATEGORIAS");

        while (LISTA.next()) {

            CMB_CAT.addItem(LISTA.getString("NOM_CATEGORIA"));

        }

    } catch (SQLException e2) {

        JOptionPane.showMessageDialog(null, "Error en consultar" + e2, "Conexion", JOptionPane.ERROR_MESSAGE);

    }

}

```

```

}

public void cargarCombo5(int sw2) {

    if (sw2 == 1) {

        ResultSet LISTA = null;

        try {

            String SQL = "SELECT NOM_SUBCATEGORIA FROM SUBCATEGORIAS WHERE COD_CATEGORIA = '1'";

            LISTA = SENTENCIA.executeQuery(SQL);

            while (LISTA.next()) {

                CMB_SUBC.addItem(LISTA.getString("NOM_SUBCATEGORIA"));

            }

        } catch (SQLException e2) {

            JOptionPane.showMessageDialog(null, "Error en consultar" + e2, "Conexion", JOptionPane.ERROR_MESSAGE);

        }

    }

    if (sw2 == 2) {

        ResultSet LISTA = null;

        try {

            String SQL = "SELECT NOM_SUBCATEGORIA FROM SUBCATEGORIAS WHERE COD_CATEGORIA = '2'";

            LISTA = SENTENCIA.executeQuery(SQL);

            while (LISTA.next()) {

                CMB_SUBC.addItem(LISTA.getString("NOM_SUBCATEGORIA"));

            }

        } catch (SQLException e2) {

            JOptionPane.showMessageDialog(null, "Error en consultar" + e2, "Conexion", JOptionPane.ERROR_MESSAGE);

        }

    }

    if (sw2 == 3) {

        ResultSet LISTA = null;
    }

```

```

try {

    String SQL = "SELECT NOM_SUBCATEGORIA FROM SUBCATEGORIAS WHERE COD_CATEGORIA = '3'";

    LISTA = SENTENCIA.executeQuery(SQL);

    while (LISTA.next()) {

        CMB_SUBC.addItem(LISTA.getString("NOM_SUBCATEGORIA"));

    }

} catch (SQLException e2) {

    JOptionPane.showMessageDialog(null, "Error en consultar" + e2, "Conexion", JOptionPane.ERROR_MESSAGE);

}

}

if (sw2 == 4) {

    ResultSet LISTA = null;

    try {

        String SQL = "SELECT NOM_SUBCATEGORIA FROM SUBCATEGORIAS WHERE COD_CATEGORIA = '4'";

        LISTA = SENTENCIA.executeQuery(SQL);

        while (LISTA.next()) {

            CMB_SUBC.addItem(LISTA.getString("NOM_SUBCATEGORIA"));

        }

    } catch (SQLException e2) {

        JOptionPane.showMessageDialog(null, "Error en consultar" + e2, "Conexion", JOptionPane.ERROR_MESSAGE);

    }

}

}

public void deshabilitar() {

    TXT_CODC.disable();

    TXT_CAT.disable();

    TXT_SUBC.disable();

    TXT_NOMC.disable();

```

```

    CMB_CAT.disable();

    CMB_SUBC.disable();
}

public void habilitar() {

    TXT_NOMC.enable();

    CMB_CAT.enable();

    CMB_SUBC.enable();

}

public void focusear() {

    CMB_CAT.requestFocus();

    CMB_SUBC.requestFocus();

    TXT_NOMC.requestFocus();

}

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

```

```

    }

    }

} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(MantenedorCuentas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(MantenedorCuentas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

    java.util.logging.Logger.getLogger(MantenedorCuentas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

    java.util.logging.Logger.getLogger(MantenedorCuentas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new MantenedorCuentas().setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JButton BTN_ACEPTAR;

private javax.swing.JButton BTN_AGREGAR;

private javax.swing.JButton BTN_CANCELAR;

private javax.swing.JButton BTN_ELIMINAR;

private javax.swing.JButton BTN_MODIFICAR;

private javax.swing.JButton BTN_VOLVER;

private javax.swing.JTextField BUS_COD;

private javax.swing.JTextField BUS_NOMC;

```

```

private javax.swing.JComboBox CMB_CAT;

private javax.swing.JComboBox CMB_SUBC;

private javax.swing.JTable TABLA_CUENTAS;

private javax.swing.JTextField TXT_CAT;

private javax.swing.JTextField TXT_CODC;

private javax.swing.JTextField TXT_NOMC;

private javax.swing.JTextField TXT_SUBC;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

// End of variables declaration

}

```

Administrador de Asientos / Comprobantes Contables

```

import java.awt.Frame;

import java.awt.Point;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

```

```

import java.sql.Statement;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JOptionPane;

import javax.swing.ListSelectionModel;

import javax.swing.RowFilter;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableRowSorter;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author El Sheriff
 */

public class Seleccionar extends javax.swing.JFrame {

    public String rut;

    public String nom;

    public int CODAS;

    public int CODCOM;

    int sw = 0;

    int sw2 = 0;

```

```

int sw3 = 0;

boolean totales = true;

String RUTS;

Calendar c = Calendar.getInstance();

private Statement SENTENCIA;

private Connection CONEXION;

private TableRowSorter filtroC;

DefaultTableModel tablaA = new DefaultTableModel() {

    @Override

    public boolean isCellEditable(int fil, int col) {

        return false;

    }

};

DefaultTableModel tablaCO = new DefaultTableModel() {

    @Override

    public boolean isCellEditable(int fil, int col) {

        return false;

    }

};

DefaultComboBoxModel Combo = new DefaultComboBoxModel();

/**

 * Creates new form Menu

 */

public Seleccionar() {

    initComponents();

}

public Seleccionar(String rut, String nom) {

    initComponents();

```

```

this.rut = rut;

this.nom = nom;

TXT_RUT.setText(rut);

TXT_CLIENTE.setText(nom);

TXT_RUT.disable();

TXT_CLIENTE.disable();

RUTS = TXT_RUT.getText();

setTitle("CONTABILIDAD");

this.setLocationRelativeTo(null);

deshabilitar();

CONECTAR();

tablaA.addColumn("NUMERO");

tablaA.addColumn("TIPO");

tablaA.addColumn("FECHA");

tablaCO.addColumn("DETALLE");

tablaCO.addColumn("CUENTA");

tablaCO.addColumn("TIPO");

tablaCO.addColumn("NUM. DOCUMENTO");

tablaCO.addColumn("GLOSA");

tablaCO.addColumn("DEBE");

tablaCO.addColumn("HABER");

this.TABLA_ASIENTO.setModel(tablaA);

this.TABLA_COMPROBANTE.setModel(tablaCO);

cargarDatos();

cargarDias(3);

}

/**

```

* This method is called from within the constructor to initialize the form.

* WARNING: Do NOT modify this code. The content of this method is always

* regenerated by the Form Editor.

*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

BTN_DIARIO = new javax.swing.JButton();

BTN_MAYOR = new javax.swing.JButton();

BTN_BALANCE = new javax.swing.JButton();

BTN_SALIR = new javax.swing.JButton();

jTabbedPane2 = new javax.swing.JTabbedPane();

PES_ASIENTO = new javax.swing.JDesktopPane();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

TXT_TIPOA = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

TABLA_ASIENTO = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

CMB_MES = new javax.swing.JComboBox();

jLabel6 = new javax.swing.JLabel();

CMB_ANO = new javax.swing.JComboBox();

BTN_ELIMINARA = new javax.swing.JButton();

jLabel9 = new javax.swing.JLabel();

CMB_DIA = new javax.swing.JComboBox();

jPanel4 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

```
TABLA_COMPROBANTE = new javax.swing.JTable();

BTN_MODIFICAR = new javax.swing.JButton();

BTN_ELIMINAR = new javax.swing.JButton();

BTN_GUARDAR = new javax.swing.JButton();

BTN_AGREGAR = new javax.swing.JButton();

jLabel15 = new javax.swing.JLabel();

TXT_CUENTA = new javax.swing.JTextField();

jLabel16 = new javax.swing.JLabel();

TXT_TIPO = new javax.swing.JTextField();

jLabel17 = new javax.swing.JLabel();

jLabel18 = new javax.swing.JLabel();

TXT_DOC = new javax.swing.JTextField();

jLabel19 = new javax.swing.JLabel();

TXT_GLOSA = new javax.swing.JTextField();

jLabel20 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

TXT_DEBE = new javax.swing.JTextField();

TXT_HABER = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

TXT_TOTALD = new javax.swing.JTextField();

TXT_TOTALH = new javax.swing.JTextField();

jLabel10 = new javax.swing.JLabel();

CMB_CUENTA = new javax.swing.JComboBox();

BTN_ACEPTAR = new javax.swing.JButton();

BTN_CANCELAR = new javax.swing.JButton();

PES_LD = new javax.swing.JDesktopPane();

jDesktopPane1 = new javax.swing.JDesktopPane();

jDesktopPane2 = new javax.swing.JDesktopPane();
```

```

TXT_CLIENTE = new javax.swing.JTextField();

TXT_RUT = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("Sistema de Contabilidad");

BTN_DIARIO.setText("Libro Diario");

BTN_DIARIO.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_DIARIOActionPerformed(evt);

    }

});

BTN_MAYOR.setText("Libro Mayor");

BTN_MAYOR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_MAYORActionPerformed(evt);

    }

});

BTN_BALANCE.setText("Balance");

BTN_BALANCE.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_BALANCEActionPerformed(evt);

    }

});

BTN_SALIR.setText("SALIR");

BTN_SALIR.addActionListener(new java.awt.event.ActionListener() {

```

```

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_SALIRActionPerformed(evt);

    }

});

jLabel2.setText("TIPO:");

TXT_TIPOA.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_TIPOAActionPerformed(evt);

    }

});

TXT_TIPOA.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusLost(java.awt.event.FocusEvent evt) {

        TXT_TIPOAFocusLost(evt);

    }

});

TABLA_ASIENTO.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

    },

    new String [] {

    }

));

TABLA_ASIENTO.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        TABLA_ASIENTOMouseClicked(evt);

    }

});

jScrollPane1.setViewportViewView(TABLA_ASIENTO);

jLabel1.setText("DÍA:");

```

```

jLabel5.setText("MES:");

CMB_MES.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11",
"12" }));

CMB_MES.addItemListener(new java.awt.event.ItemListener() {

    public void itemStateChanged(java.awt.event.ItemEvent evt) {

        CMB_MESItemStateChanged(evt);

    }

});

CMB_MES.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        CMB_MESFocusGained(evt);

    }

});

jLabel6.setText("AÑO:");

CMB_ANO.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "2008", "2009", "2010", "2011", "2012", "2013", "2014",
"2015", "2016", "2017" }));

BTN_ELIMINARA.setText("ELIMINAR");

BTN_ELIMINARA.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_ELIMINARAActionPerformed(evt);

    }

});

jLabel9.setText("INGRESO DE ASIENTOS");

CMB_DIA.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11",
"12" }));

CMB_DIA.addItemListener(new java.awt.event.ItemListener() {

    public void itemStateChanged(java.awt.event.ItemEvent evt) {

        CMB_DIAItemStateChanged(evt);

    }

});

```

```

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel2Layout.createSequentialGroup()

            .addGap(158, 158, 158)

            .addComponent(BTN_ELIMINARA, javax.swing.GroupLayout.PREFERRED_SIZE, 147, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(jLabel5)

            .addGap(18, 18, 18)

            .addComponent(CMB_MES, javax.swing.GroupLayout.PREFERRED_SIZE, 83, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 26, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

        )

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel2Layout.createSequentialGroup()

                .addComponent(TXT_TIPOA, javax.swing.GroupLayout.PREFERRED_SIZE, 30, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

                .addComponent(jLabel2)

                .addGap(10, 10, 10)

            )

            .addGroup(jPanel2Layout.createSequentialGroup()

                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 26, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

            )

            .addGroup(jPanel2Layout.createSequentialGroup()

                .addComponent(CMB_MES, javax.swing.GroupLayout.PREFERRED_SIZE, 83, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

            )

            .addGroup(jPanel2Layout.createSequentialGroup()

                .addComponent(jLabel5)

                .addGap(18, 18, 18)

            )

            .addGroup(jPanel2Layout.createSequentialGroup()

                .addComponent(BTN_ELIMINARA, javax.swing.GroupLayout.PREFERRED_SIZE, 147, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(158, 158, 158)

            )

        )

    );

```

```

        .addComponent(CMB_DIA, javax.swing.GroupLayout.PREFERRED_SIZE, 51, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(jLabel6)

        .addGap(18, 18, 18)

        .addComponent(CMB_ANO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(jPanel2Layout.createSequentialGroup())

        .addGap(130, 130, 130)

        .addComponent(jLabel9))))

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 465, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addContainerGap(20, Short.MAX_VALUE))))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup())

.addGap(22, 22, 22)

.addComponent(jLabel9)

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(TXT_TIPOA, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(CMB_MES, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5)

.addComponent(jLabel1)

.addComponent(jLabel6)

.addComponent(CMB_ANO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(CMB_DIA, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 338, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(BTN_ELIMINARA)

        .addContainerGap(27, Short.MAX_VALUE))

    );

    TABLA_COMPROBANTE.setModel(new javax.swing.table.DefaultTableModel(

        new Object [][] {

            },

        new String [] {

            }

    ));

    TABLA_COMPROBANTE.addMouseListener(new java.awt.event.MouseAdapter() {

        public void mouseClicked(java.awt.event.MouseEvent evt) {

            TABLA_COMPROBANTEMouseClicked(evt);

        }

    });

    jScrollPane3.setViewportViewView(TABLA_COMPROBANTE);

    BTN_MODIFICARC.setText("MODIFICAR");

    BTN_MODIFICARC.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            BTN_MODIFICARCActionPerformed(evt);

        }

    });

    BTN_ELIMINARC.setText("ELIMINAR");

    BTN_ELIMINARC.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        BTN_ELIMINARActionPerformed(evt);

    }

});

BTN_GUARDAR.setText("<html><p>GUARDAR</p><p>COMPROBANTE</p></html>"); // NOI18N

BTN_GUARDAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_GUARDARActionPerformed(evt);

    }

});

BTN_AGREGAR.setText("AGREGAR");

BTN_AGREGAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_AGREGARActionPerformed(evt);

    }

});

jLabel15.setText("CUENTA: ");

TXT_CUENTA.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_CUENTAActionPerformed(evt);

    }

});

TXT_CUENTA.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_CUENTAKeyTyped(evt);

    }

});

jLabel16.setText("TIPO DOCUMENTO: ");

TXT_TIPO.setText("-");

```

```

jLabel17.setText("EJ: FA, GD, CH, ETC. ");

jLabel18.setText("NÚMERO DOCUMENTO: ");

TXT_DOC.setText("0");

TXT_DOC.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_DOCKeyTyped(evt);

    }

});

jLabel19.setText("GLOSA: ");

TXT_GLOSA.setText("-");

jLabel20.setText("DEBE $ :");

jLabel7.setText("HABER $ :");

TXT_DEBE.setText("0");

TXT_DEBE.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_DEBEKeyTyped(evt);

    }

});

TXT_HABER.setText("0");

TXT_HABER.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        TXT_HABERKeyTyped(evt);

    }

});

jLabel8.setText("TOTALES $ :");

TXT_TOTALH.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_TOTALHActionPerformed(evt);

```

```
}  
});  
  
jLabel10.setText("COMPROBANTES CONTABLES");  
  
CMB_CUENTA.setModel(new javax.swing.DefaultComboBoxModel(new String[] { " " }));  
  
BTN_ACEPTAR.setText("ACEPTAR");  
  
BTN_ACEPTAR.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        BTN_ACEPTARActionPerformed(evt);  
    }  
});  
  
BTN_CANCELAR.setText("CANCELAR");  
  
BTN_CANCELAR.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        BTN_CANCELARActionPerformed(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  
jPanel4.setLayout(jPanel4Layout);  
jPanel4Layout.setHorizontalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
.addGroup(jPanel4Layout.createSequentialGroup()  
    .addGap(89, 89, 89)  
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addContainerGap(67, true)  
        .addGroup(jPanel4Layout.createSequentialGroup().addContainerGap().add(jLabel10).addContainerGap())  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .add(CMB_CUENTA)  
            .add(BTN_ACEPTAR)  
            .add(BTN_CANCELAR)  
            .add(ContainerUtils.createPlaceholder(jPanel4Layout, 3))  
        )  
    .addContainerGap(108, false))  
);
```

```

.addGap(67, 67, 67)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel7)

.addGroup(jPanel4Layout.createSequentialGroup())

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel20)

.addComponent(jLabel19))

.addGap(1, 1, 1)))

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel15, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel18, javax.swing.GroupLayout.Alignment.TRAILING))

.addGroup(jPanel4Layout.createSequentialGroup())

.addGap(23, 23, 23)

.addComponent(jLabel16, javax.swing.GroupLayout.PREFERRED_SIZE, 117, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup())

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(TXT_DOC, javax.swing.GroupLayout.PREFERRED_SIZE, 56, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel4Layout.createSequentialGroup())

.addComponent(TXT_TIPO, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jLabel17))

.addComponent(TXT_GLOSA, javax.swing.GroupLayout.PREFERRED_SIZE, 347,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(TXT_HABER, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE,
110, Short.MAX_VALUE)

```

```

        .addComponent(TXT_DEBE, javax.swing.GroupLayout.Alignment.LEADING)))

        .addGap(0, 63, Short.MAX_VALUE))

        .addGroup(jPanel4Layout.createSequentialGroup())

        .addComponent(TXT_CUENTA)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(CMB_CUENTA, javax.swing.GroupLayout.PREFERRED_SIZE, 227,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(jPanel4Layout.createSequentialGroup())

        .addGap(23, 23, 23)

        .addComponent(jScrollPane3)))

        .addGap(18, 18, 18)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addComponent(BTN_ELIMINARC, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(BTN_MODIFICARC, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(BTN_AGREGARC, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(BTN_GUARDAR)

        .addComponent(BTN_ACEPTAR, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(BTN_CANCELAR, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addGap(37, 37, 37))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel4Layout.createSequentialGroup())

        .addGap(0, 0, Short.MAX_VALUE)

        .addComponent(jLabel8)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(TXT_TOTALD, javax.swing.GroupLayout.PREFERRED_SIZE, 118, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(TXT_TOTALH, javax.swing.GroupLayout.PREFERRED_SIZE, 118, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(120, 120, 120))))

    .addGroup(jPanel4Layout.createSequentialGroup())

    .addGap(295, 295, 295)

    .addComponent(jLabel10)

    .addGap(0, 0, Short.MAX_VALUE)

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup())

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup())

.addContainerGap()

.addComponent(jLabel10)

.addGap(24, 24, 24)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 203, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel15)

.addComponent(TXT_CUENTA, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(CMB_CUENTA, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel16)

.addComponent(TXT_TIPO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel17))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel18)

    .addComponent(TXT_DOC, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(TXT_GLOSA, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jLabel19)

    .addComponent(BTN_ACCEPTAR))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel20)

    .addComponent(TXT_DEBE, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(BTN_CANCELAR))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel7)

    .addComponent(TXT_HABER, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(jPanel4Layout.createSequentialGroup())

.addGap(25, 25, 25)

.addComponent(BTN_AGREGARC, javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(BTN_MODIFICARC, javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(BTN_ELIMINARC, javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(104, 104, 104)

```

```

        .addComponent(BTN_GUARDAR, javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(9, 9, 9)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel8)

            .addComponent(TXT_TOTALD, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(TXT_TOTALH, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addContainerGap(21, Short.MAX_VALUE))

    );

    javax.swing.GroupLayout PES_ASIENTOLayout = new javax.swing.GroupLayout(PES_ASIENTO);

    PES_ASIENTO.setLayout(PES_ASIENTOLayout);

    PES_ASIENTOLayout.setHorizontalGroup(

        PES_ASIENTOLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(PES_ASIENTOLayout.createSequentialGroup()

            .addGap(24, 24, 24)

            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 46, Short.MAX_VALUE)

            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(45, 45, 45))

        );

    PES_ASIENTOLayout.setVerticalGroup(

        PES_ASIENTOLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        );

    PES_ASIENTO.setLayer(jPanel2, javax.swing.JLayeredPane.DEFAULT_LAYER);

    PES_ASIENTO.setLayer(jPanel4, javax.swing.JLayeredPane.DEFAULT_LAYER);

```

```

jTabbedPane2.addTab("ASIENTOS", PES_ASIENTO);

javax.swing.GroupLayout PES_LDLayout = new javax.swing.GroupLayout(PES_LD);

PES_LD.setLayout(PES_LDLayout);

PES_LDLayout.setHorizontalGroup(

    PES_LDLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGap(0, 1338, Short.MAX_VALUE)

);

PES_LDLayout.setVerticalGroup(

    PES_LDLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGap(0, 491, Short.MAX_VALUE)

);

jTabbedPane2.addTab("LIBRO DIARIO", PES_LD);

javax.swing.GroupLayout jDesktopPane1Layout = new javax.swing.GroupLayout(jDesktopPane1);

jDesktopPane1.setLayout(jDesktopPane1Layout);

jDesktopPane1Layout.setHorizontalGroup(

    jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGap(0, 1338, Short.MAX_VALUE)

);

jDesktopPane1Layout.setVerticalGroup(

    jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGap(0, 491, Short.MAX_VALUE)

);

jTabbedPane2.addTab("LIBRO MAYOR", jDesktopPane1);


javax.swing.GroupLayout jDesktopPane2Layout = new javax.swing.GroupLayout(jDesktopPane2);

jDesktopPane2.setLayout(jDesktopPane2Layout);

jDesktopPane2Layout.setHorizontalGroup(

```

```

jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGap(0, 1338, Short.MAX_VALUE)

);

jDesktopPane2Layout.setVerticalGroup(

    jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGap(0, 491, Short.MAX_VALUE)

    );

jTabbedPane2.addTab("BALANCE", jDesktopPane2);

TXT_RUT.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TXT_RUTActionPerformed(evt);

    }

});

jLabel4.setText("RUT:");

jLabel11.setText("NOMBRE:");

jButton1.setText("IMPRIMIR");

jButton1.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton1ActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(28, 28, 28)

            .addComponent(BTN_DIARIO, javax.swing.GroupLayout.PREFERRED_SIZE, 102, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(BTN_MAYOR, javax.swing.GroupLayout.PREFERRED_SIZE, 98, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(10, 10, 10)

.addComponent(BTN_BALANCE, javax.swing.GroupLayout.PREFERRED_SIZE, 104, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(233, 233, 233)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 91, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel4, javax.swing.GroupLayout.Alignment.TRAILING)

    .addComponent(jLabel11, javax.swing.GroupLayout.Alignment.TRAILING))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(TXT_CLIENTE, javax.swing.GroupLayout.PREFERRED_SIZE, 253, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(TXT_RUT, javax.swing.GroupLayout.PREFERRED_SIZE, 180, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(BTN_SALIR, javax.swing.GroupLayout.PREFERRED_SIZE, 90, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(23, 23, 23))

.addGroup(jPanel1Layout.createSequentialGroup()

    .addComponent(jTabbedPane2))

);

jPanel1Layout.setVerticalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(jPanel1Layout.createSequentialGroup()

                    .addGap(32, 32, 32)

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(BTN_DIARIO, javax.swing.GroupLayout.PREFERRED_SIZE, 48, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(BTN_MAYOR, javax.swing.GroupLayout.PREFERRED_SIZE, 48, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(BTN_BALANCE, javax.swing.GroupLayout.PREFERRED_SIZE, 48, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(BTN_SALIR, javax.swing.GroupLayout.PREFERRED_SIZE, 48, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jButton1))

        .addGap(38, 38, 38))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

        .addContainerGap()

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(TXT_RUT, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel4))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(TXT_CLIENTE, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel11))

        .addGap(40, 40, 40)))

        .addComponent(jTabbedPane2)

        .addContainerGap())

);

jTabbedPane2.getAccessibleContext().setAccessibleName("ASIENTOS\n");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addContainerGap())

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

);

pack();

} // </editor-fold>

private void BTN_BALANCEActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void BTN_MAYORActionPerformed(java.awt.event.ActionEvent evt) {

    Imprimir imp = new Imprimir(TXT_RUT.getText(),2);

    imp.setVisible(true);

}

private void BTN_DIARIOActionPerformed(java.awt.event.ActionEvent evt) {

    Imprimir imp = new Imprimir(TXT_RUT.getText(),1);

    imp.setVisible(true);

}

private void BTN_SALIRActionPerformed(java.awt.event.ActionEvent evt) {

    Menu menu = new Menu();

    menu.setVisible(true);

    try {

        CONEXION.close();

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Error en Cerrar la BD. " + ex, "Cerrar BD", JOptionPane.ERROR_MESSAGE);

    }

    this.dispose();

```

```

// TODO add your handling code here:
}

private void TXT_TOTALHActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
}

private void TXT_RUTActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
}

private void TABLA_COMPROBANTEMouseClicked(java.awt.event.MouseEvent evt) {

    try {

        TABLA_COMPROBANTE.setRowSelectionAllowed(true);

        TABLA_COMPROBANTE.setColumnSelectionAllowed(false);

        int fila = TABLA_COMPROBANTE.getSelectedRow();

        int COD_C = ((Integer) TABLA_COMPROBANTE.getValueAt(fila, 0));

        String CODC = Integer.toString(COD_C);

        TXT_CUENTA.setText((String) TABLA_COMPROBANTE.getValueAt(fila, 1));

        TXT_TIPO.setText((String) TABLA_COMPROBANTE.getValueAt(fila, 2));

        // JOptionPane.showMessageDialog(null, "RUT = " + RUTE, "RUT",JOptionPane.ERROR_MESSAGE);

        // String SQL = "SELECT * FROM CLIENTES WHERE RUT_EMP = '"+RUTE+"'";

        // ResultSet item = SENTENCIA.executeQuery(SQL);

        int DOC = (Integer) TABLA_COMPROBANTE.getValueAt(fila, 3);

        TXT_DOC.setText(Integer.toString(DOC));

        TXT_GLOSA.setText((String) TABLA_COMPROBANTE.getValueAt(fila, 4));

        int DEBE = (Integer) TABLA_COMPROBANTE.getValueAt(fila, 5);

```

```

        TXT_DEBE.setText(Integer.toString(DEBE));

        int HABER = (Integer) TABLA_COMPROBANTE.getValueAt(fila, 6);

        TXT_HABER.setText(Integer.toString(HABER));

        deshabilitar();

    } catch (Exception e3) {

        JOptionPane.showMessageDialog(null, "Error en encontrar " + e3, "Buscar tipo", JOptionPane.ERROR_MESSAGE);

    }

    // TODO add your handling code here:

}

private void BTN_AGREGARActionPerformed(java.awt.event.ActionEvent evt) {

    CMB_CUENTA.setSelectedIndex(0);

    BTN_MODIFICAR.disabled();

    BTN_ELIMINAR.disabled();

    habilitar();

    limpiar();

    focusear();

    cargarCombo();

    if (sw3 == 3) {

        sw2 = 1;

    }

    // TODO add your handling code here:

}

private void BTN_MODIFICARActionPerformed(java.awt.event.ActionEvent evt) {

    sw = 1;

    habilitar();

    focusear();

    TXT_CUENTA.requestFocus();

```

```

// TODO add your handling code here:

}

private void BTN_ELIMINARActionPerformed(java.awt.event.ActionEvent evt) {

    int fila = TABLA_COMPROBANTE.getSelectedRow();

    int COD_C = ((Integer) TABLA_COMPROBANTE.getValueAt(fila, 0));

    String CODC = Integer.toString(COD_C);

    int reply = JOptionPane.showConfirmDialog(null, "ESTÁ SEGURO/A DE ELIMINAR? ", "ELIMINAR", JOptionPane.YES_NO_OPTION);

    if (reply == JOptionPane.YES_OPTION) {

        String SQL = "DELETE FROM COMPROBANTES WHERE COD_COMPROBANTE = " + CODC + " AND COD_ASIENTO = " + CODAS + " AND RUT_CLIENTE = '" + RUTS + "'";

        try {

            SENTENCIA.executeUpdate(SQL);

            JOptionPane.showMessageDialog(null, "COMPROBANTE ELIMINADO.");

        } catch (SQLException ex) {

            JOptionPane.showMessageDialog(null, "ERROR EN ELIMINAR " + ex, "ELIMINAR", JOptionPane.ERROR_MESSAGE);

        }

        cargarDatosC();

        if (TABLA_COMPROBANTE.getRowCount() == 0) {

            String SQL2 = "DELETE FROM ASIENTOS WHERE COD_ASIENTO = " + CODAS + " AND RUT_CLIENTE = '" + RUTS + "'";

            try {

                SENTENCIA.executeUpdate(SQL2);

                JOptionPane.showMessageDialog(null, "Asiento eliminado");

                cargarDatos();

                limpiar();

                deshabilitar();

            } catch (SQLException ex) {

                JOptionPane.showMessageDialog(null, "Error en ELIMINAR " + ex, "ELIMINAR", JOptionPane.ERROR_MESSAGE);

            }

        }
    }
}

```

```

    }

    } else {

    }

    // TODO add your handling code here:

}

private void BTN_GUARDARActionPerformed(java.awt.event.ActionEvent evt) {

    String ASI, ASIE = null, CODC = null, CODCO = null, CUE, NOMCU, NCUENTA = null, TIPO, DOC, GLOSA, DEBE, HABER, INSERTAR, INSERTAR_A,
MODIFICAR;

    int ASIAUX = 0;

    NOMCU = (String) CMB_CUENTA.getSelectedItemAt();

    String NOMC = (String) CMB_CUENTA.getItemAt(0);

    if (NOMCU == NOMC) {

        NOMCU = TXT_CUENTA.getText();

    }

    TIPO = TXT_TIPO.getText();

    DOC = TXT_DOC.getText();

    GLOSA = TXT_GLOSA.getText();

    DEBE = TXT_DEBE.getText();

    HABER = TXT_HABER.getText();

    if (NOMCU.equals("") || TIPO.equals("") || DOC.equals("") || GLOSA.equals("") || DEBE.equals("") || HABER.equals("")) {

        JOptionPane.showMessageDialog(null, "Debe llenar todos los campos", "Error al intentar guardar.",
JOptionPane.INFORMATION_MESSAGE);

    } else {

        if (sw == 0) {

            if (sw2 == 0) {

                String TIPOA, FECHAUX, DIA, MES, ANO;

                DIA = (String) CMB_DIA.getSelectedItemAt();

                MES = (String) CMB_MES.getSelectedItemAt();

```

```

ANO = (String) CMB_ANO.getSelectedItem();

TIPOA = TXT_TIPOA.getText();

FECHAUX = ANO + "/" + MES + "/" + DIA;

/*ASI = "SELECT A.COD_ASIENTO FROM ASIENTOS A, CLIENTES CL WHERE A.RUT_CLIENTE = '" + RUTS + "' AND A.RUT_CLIENTE =
CL.RUT_CLIENTE";

ResultSet ASIEN = SENTENCIA.executeQuery(ASI);

if (ASIEN.next() == false) {

    ASIE = "0";

} else {

    while (ASIEN.next()) {

        ASIE = ASIEN.getString("COD_ASIENTO");          }

    }

JOptionPane.showMessageDialog(null, "COD ASIENTO: "+ ASIE);

int COD_A = Integer.parseInt(ASIE);

COD_A = COD_A + 1;

ASIE = Integer.toString(COD_A);*/

int fila = TABLA_ASIENTO.getRowCount() - 1;

if (fila == -1) {

    ASIE = Integer.toString(1);

} else {

    ASI = "SELECT MAX(COD_ASIENTO)+1 FROM ASIENTOS WHERE RUT_CLIENTE = '" + RUTS + "'";

    try {

        ResultSet ASIEN = SENTENCIA.executeQuery(ASI);

        while (ASIEN.next()) {

            ASIE = ASIEN.getString("MAX(COD_ASIENTO)+1");

        }

    } catch (SQLException ex) {

    }

}

```

```

INSERTAR_A = "INSERT INTO ASIENTOS VALUES (" + ASIE + ", " + RUTS + ", " + TIPOA + ", " + FECHAUX + ")";

try {

    SENTENCIA.executeUpdate(INSERTAR_A);

} catch (SQLException e3) {

    //JOptionPane.showMessageDialog(null, "NO SE PUDO AGREGAR ASIENTO" + e3, "Conexion", JOptionPane.ERROR_MESSAGE);

}

try {

    CODAS = Integer.parseInt(ASIE);

    fila = TABLA_COMPROBANTE.getRowCount() - 1;

    if (fila == -1) {

        CODCO = Integer.toString(1);

    } else {

        CODC = "(SELECT MAX(COD_COMPROBANTE)+1 FROM COMPROBANTES WHERE COD_ASIENTO = " + CODAS + " AND
RUT_CLIENTE = " + RUTS + ")";

        ResultSet COD_C = SENTENCIA.executeQuery(CODC);

        while (COD_C.next()) {

            CODCO = COD_C.getString("MAX(COD_COMPROBANTE)+1");

        }

    }

} catch (SQLException e3) {

    JOptionPane.showMessageDialog(null, "Error en nro. comprobante." + e3, "Conexion", JOptionPane.ERROR_MESSAGE);

}

try {

    CUE = "SELECT COD_CUENTA FROM CUENTAS WHERE NOM_CUENTA = " + NOMCU + """;

    ResultSet CUENTA = SENTENCIA.executeQuery(CUE);

    while (CUENTA.next()) {

        NCUENTA = CUENTA.getString("COD_CUENTA");

    }

}

```

```

    } catch (SQLException e3) {

        JOptionPane.showMessageDialog(null, "ERROR AL ENCONTRAR CUENTA" + e3, "INGRESAR COMPROBANTE",
JOptionPane.ERROR_MESSAGE);

    }

    INSERTAR = "INSERT INTO COMPROBANTES VALUES "

        + "(" + CODCO + ", " + ASIE + ", " + NCUENTA + ", " + RUTS + ", " + TIPO + ", " + DOC + ", " + GLOSA + ", " + DEBE + ", " +
HABER + ")";

    try {

        SENTENCIA.executeUpdate(INSERTAR);

        JOptionPane.showMessageDialog(null, "Datos Guardados");

    } catch (SQLException e3) {

        JOptionPane.showMessageDialog(null, "ERROR AL INGRESAR COMPROBANTE" + e3, "INGRESAR COMPROBANTE",
JOptionPane.ERROR_MESSAGE);

    }

    cargarDatos();

    cargarDatosC();

    limpiar();

    TXT_TIPOA.setText("");

    TABLA_ASIENTO.changeSelection(TABLA_ASIENTO.getRowCount() - 1, 0, false, false);

    // int fila3 = TABLA_ASIENTO.getSelectedRow();

    sw3 = 3;

    //sw2 =1;

}

if (sw2 == 1) {

    //CODC = "(SELECT COD_COMPROBANTE FROM COMPROBANTES WHERE COD_ASIENTO = " + CODAS + " AND RUT_CLIENTE = " +
RUTS + ")";

    CODCO = Integer.toString(TABLA_COMPROBANTE.getRowCount());

    int CODCOM = Integer.parseInt(CODCO);

    CODCOM = CODCOM + 1;

    CODCO = Integer.toString(CODCOM);

```

```

try {

    CUE = "SELECT COD_CUENTA FROM CUENTAS WHERE NOM_CUENTA = '" + NOMCU + "'";

    ResultSet CUENTA = SENTENCIA.executeQuery(CUE);

    while (CUENTA.next()) {

        NCUENTA = CUENTA.getString("COD_CUENTA");

    }

} catch (SQLException e3) {

    JOptionPane.showMessageDialog(null, "ERROR AL ENCONTRAR CUENTA. " + e3, "INGRESAR COMROBANTE",
JOptionPane.ERROR_MESSAGE);

}

INSERTAR = "INSERT INTO COMPROBANTES VALUES "

        + "( " + CODCO + ", " + CODAS + ", '" + NCUENTA + "', '" + RUTS + "', '" + TIPO + "', " + DOC + ", '" + GLOSA + "', " + DEBE + ", " +
HABER + ")";

try {

    SENTENCIA.executeUpdate(INSERTAR);

    JOptionPane.showMessageDialog(null, "Datos Guardados");

} catch (SQLException e3) {

    JOptionPane.showMessageDialog(null, "ERROR AL AGREGAR: " + e3, "INSERTAR", JOptionPane.ERROR_MESSAGE);

}

cargarDatosC();

limpiar();

deshabilitar();

TXT_TIPOA.setText("");

}

}

if (sw == 1) {

    TABLA_COMPROBANTE.setRowSelectionAllowed(true);

    TABLA_COMPROBANTE.setColumnSelectionAllowed(false);

    int fila = TABLA_COMPROBANTE.getSelectedRow();

```

```

CODCOM = ((Integer) TABLA_COMPROBANTE.getValueAt(fila, 0));

CUE = "SELECT COD_CUENTA FROM CUENTAS WHERE NOM_CUENTA = '" + NOMCU + "'";

try {

    ResultSet CUENTA = SENTENCIA.executeQuery(CUE);

    while (CUENTA.next()) {

        NCUENTA = CUENTA.getString("COD_CUENTA");

    }

} catch (Exception e) {

}

MODIFICAR = "UPDATE COMPROBANTES SET COD_CUENTA = '" + NCUENTA + "', TIPO = '" + TIPO + "', NUM_DOC = '" + DOC + "', GLOSA = '" + GLOSA + "', DEBE = '" + DEBE + "', HABER = '" + HABER + "' WHERE COD_COMPROBANTE = '" + CODCOM + "' AND COD_ASIENTO = '" + CODAS + "' AND RUT_CLIENTE = '" + RUTS + "'";

try {

    SENTENCIA.executeUpdate(MODIFICAR);

    JOptionPane.showMessageDialog(null, "Datos Guardados");

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "ERROR AL MODIFICAR.", "MODIFICAR COMPROBANTES", JOptionPane.ERROR_MESSAGE);

}

cargarDatosC();

limpiar();

TABLA_ASIENTO.changeSelection(TABLA_ASIENTO.getRowCount() - 1, 0, false, false);

//int fila3 = TABLA_ASIENTO.getSelectedRow();

}

}

sw = 0;

cargarTotales();

// TODO add your handling code here:

}

private void BTN_ACEPTARActionPerformed(java.awt.event.ActionEvent evt) {

```

```

if (!(TXT_TOTALD.getText().equals(TXT_TOTALH.getText())) {

    JOptionPane.showMessageDialog(null, "ERROR EN LOS TOTALES.", "COMPROBANTES", JOptionPane.ERROR_MESSAGE);

    totales = false;

    sw3 = 3;

    habilitar();

    focusear();

} else {

    JOptionPane.showMessageDialog(null, "ASIENTO GUARDADO EXITOSAMENTE.", "GUARDAR", JOptionPane.INFORMATION_MESSAGE);

    TABLA_COMPROBANTE.setColumnSelectionAllowed(false);

    TABLA_COMPROBANTE.setCellSelectionEnabled(false);

    TABLA_ASIENTO.setColumnSelectionAllowed(false);

    TABLA_ASIENTO.setCellSelectionEnabled(false);

    sw=0;

    sw2=0;

    limpiar();

    TXT_TIPOA.setText("");

    deshabilitar();

    eliminarDatosC();

    totales = true;

}

// TODO add your handling code here:

}

private void TXT_CUENTAKeyTyped(java.awt.event.KeyEvent evt) {

    /* try {

        if (TXT_CUENTA.equals("")) {

            CMB_CUENTA.setSelectedIndex(0);

        } else {

            String SQL = "SELECT NOM_CUENTA FROM CUENTAS WHERE NOM_CUENTA LIKE '" + TXT_CUENTA.getText() + "%'";

```

```

ResultSet resultado = SENTENCIA.executeQuery(SQL);

while (resultado.next()) {

    CMB_CUENTA.setSelectedItem(resultado.getString("NOM_CUENTA"));

}

}

} catch (SQLException e4) {

    JOptionPane.showMessageDialog(null, "Error en Filtrar" + e4, "COMBO", JOptionPane.ERROR_MESSAGE);

}

Combo = new TableRowSorter(CMB_CUENTA.getModel());

TABLA_DATOS.setRowSorter(filtroC);*/

// TODO add your handling code here:

}

private void TXT_CUENTAActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void BTN_CANCELARActionPerformed(java.awt.event.ActionEvent evt) {

    TABLA_ASIENTO.setColumnSelectionAllowed(false);

    TABLA_ASIENTO.setCellSelectionEnabled(false);

    limpiar();

    focusear();

    deshabilitar();

    sw = 0;

    sw2 = 0;

    eliminarDatosC();

    sw3 = 1;

    // TODO add your handling code here:

}

public void eliminarDatosC() {

```

```

DefaultTableModel tablac = (DefaultTableModel) TABLA_COMPROBANTE.getModel();

int a = TABLA_COMPROBANTE.getRowCount() - 1;

for (int i = a; i >= 0; i--) {

    tablac.removeRow(tablac.getRowCount() - 1);

}

}

private void BTN_ELIMINARAActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        String COMPR = null;

        int fila = TABLA_ASIENTO.getSelectedRow();

        CODAS = ((Integer) TABLA_ASIENTO.getValueAt(fila, 0));

        int reply = JOptionPane.showConfirmDialog(null, "Está Seguro/ a de eliminar? ", "ELIMINAR", JOptionPane.YES_NO_OPTION);

        if (reply == JOptionPane.YES_OPTION) {

            try {

                String SQL = "SELECT COUNT(COD_COMPROBANTE) FROM COMPROBANTES WHERE COD_ASIENTO = " + CODAS + " AND
RUT_CLIENTE = " + RUTS + """;

                ResultSet COMP = SENTENCIA.executeQuery(SQL);

                while (COMP.next()) {

                    COMPR = COMP.getString("COUNT(COD_COMPROBANTE)");

                }

            } catch (Exception e) {

                JOptionPane.showMessageDialog(null, "ERROR EN ELIMINAR " + e, "ELIMINAR", JOptionPane.ERROR_MESSAGE);

            }

            if (COMPR.equals("0")) {

            } else {

                String SQL1 = "DELETE FROM COMPROBANTES WHERE COD_ASIENTO = " + CODAS + " AND RUT_CLIENTE = " + RUTS + """;

                try {

                    SENTENCIA.executeUpdate(SQL1);

```

```

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, "ERROR EN ELIMINAR " + ex, "ELIMINAR", JOptionPane.ERROR_MESSAGE);

    }

}

String SQL2 = "DELETE FROM ASIENTOS WHERE COD_ASIENTO = '" + CODAS + "' AND RUT_CLIENTE = '" + RUTS + "'";

try {

    SENTENCIA.executeUpdate(SQL2);

    JOptionPane.showMessageDialog(null, "ASIENTO EIMINADO");

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(null, "ERROR EN ELIMINAR " + ex, "ELIMINAR", JOptionPane.ERROR_MESSAGE);

}

TABLA_ASIENTO.setColumnSelectionAllowed(false);

TABLA_ASIENTO.setCellSelectionEnabled(false);

limpiar();

focusear();

deshabilitar();

sw = 0;

sw2 = 0;

}

cargarDatos();

cargarDatosC();

// TODO add your handling code here:

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN ASIENTO", "ELIMINAR", JOptionPane.INFORMATION_MESSAGE);

}

}

```

```

private void CMB_MESItemStateChanged(java.awt.event.ItemEvent evt) {

    if (CMB_MES.getSelectedItem().equals("02")) {

        CMB_DIA.removeAllItems();

        cargarDias(1);

    }

    if (CMB_MES.getSelectedItem().equals("01") || CMB_MES.getSelectedItem().equals("03") || CMB_MES.getSelectedItem().equals("05") ||
CMB_MES.getSelectedItem().equals("07") || CMB_MES.getSelectedItem().equals("08") || CMB_MES.getSelectedItem().equals("10") ||
CMB_MES.getSelectedItem().equals("12")) {

        CMB_DIA.removeAllItems();

        cargarDias(2);

    }

    if (CMB_MES.getSelectedItem().equals("04") || CMB_MES.getSelectedItem().equals("06") || CMB_MES.getSelectedItem().equals("09") ||
CMB_MES.getSelectedItem().equals("11") || CMB_MES.getSelectedItem().equals("08") || CMB_MES.getSelectedItem().equals("11")) {

        CMB_DIA.removeAllItems();

        cargarDias(3);

    }

    // TODO add your handling code here:

}

private void TABLA_ASIENTOMouseClicked(java.awt.event.MouseEvent evt) {

    if (totales == false) {

        JOptionPane.showMessageDialog(null, "DEBEN CALZAR LOS TOTALES", "ERROR", JOptionPane.ERROR_MESSAGE);

        cargarDatosC();

    } else {

        sw2 = 1;

        try {

            TABLA_ASIENTO.setRowSelectionAllowed(true);

            TABLA_ASIENTO.setColumnSelectionAllowed(false);

            int fila = TABLA_ASIENTO.getSelectedRow();

            CODAS = ((Integer) TABLA_ASIENTO.getValueAt(fila, 0));


```

```

try {

    String SQL = "SELECT C.COD_COMPROBANTE, CU.NOM_CUENTA, C.TIPO, C.NUM_DOC, C.GLOSA, C.DEBE, C.HABER FROM CLIENTES
CL, COMPROBANTES C, ASIENTOS A, CUENTAS CU "

        + "WHERE C.RUT_CLIENTE = " + RUTS + " AND C.COD_ASIENTO = " + CODAS + " AND C.COD_CUENTA = CU.COD_CUENTA AND
A.RUT_CLIENTE = CL.RUT_CLIENTE AND C.RUT_CLIENTE = CL.RUT_CLIENTE AND C.RUT_CLIENTE = A.RUT_CLIENTE AND C.COD_ASIENTO =
A.COD_ASIENTO ORDER BY C.COD_COMPROBANTE";

    ResultSet lista = SENTENCIA.executeQuery(SQL);

    Object datos[] = new Object[7];

    tablaCO.setRowCount(0);

    while (lista.next()) {

        for (int i = 0; i < 7; i++) {

            datos[i] = lista.getObject(i + 1);

        }

        tablaCO.addRow(datos);

    }

} catch (Exception l1) {

    JOptionPane.showMessageDialog(null, "Error en listar " + l1, "Listar Datos", JOptionPane.ERROR_MESSAGE);

}

} catch (Exception e3) {

    JOptionPane.showMessageDialog(null, "Error en encontrar " + e3, "Buscar tipo", JOptionPane.ERROR_MESSAGE);

}

cargarTotales();

}

// TODO add your handling code here:

}

private void TXT_TIPOAActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void CMB_DIAItemStateChanged(java.awt.event.ItemEvent evt) {

    // TODO add your handling code here:

}

private void TXT_TIPOAFocusLost(java.awt.event.FocusEvent evt) {

    // TODO add your handling code here:

}

private void CMB_MESFocusGained(java.awt.event.FocusEvent evt) {

    if ((TXT_TIPOA.getText().toUpperCase().equals("I")) || (TXT_TIPOA.getText().toUpperCase().equals("E")) ||
(TXT_TIPOA.getText().toUpperCase().equals("T"))) {

        } else {

            TXT_TIPOA.requestFocus();

            JOptionPane.showMessageDialog(null, "DEBE INGRESAR I, E O T. \n I = INGRESO\n E = EGRESO\n T = TRASPASO.", "ERROR",
JOptionPane.ERROR_MESSAGE);

            TXT_TIPOA.setText("");

        }

        // TODO add your handling code here:

    }

private void TXT_DOCKeyTyped(java.awt.event.KeyEvent evt) {

    char character = evt.getKeyChar();

    if (((character < '0')

        || (character > '9'))

        && (character != '\b')) {

        evt.consume();

    }

    String Caracteres = TXT_DOC.getText();

    if (Caracteres.length() >= 5) {

        evt.consume();

    }

    // TODO add your handling code here:

```

```

}

private void TXT_DEBEKeyTyped(java.awt.event.KeyEvent evt) {

    char character = evt.getKeyChar();

    if (((character < '0')

        || (character > '9'))

        && (character != '\b')) {

        evt.consume();

    }

    String Caracteres = TXT_DEBE.getText();

    if (Caracteres.length() >= 12) {

        evt.consume();

    }

    // TODO add your handling code here:

}

private void TXT_HABERKeyTyped(java.awt.event.KeyEvent evt) {

    char character = evt.getKeyChar();

    if (((character < '0')

        || (character > '9'))

        && (character != '\b')) {

        evt.consume();

    }

    String Caracteres = TXT_HABER.getText();

    if (Caracteres.length() >= 12) {

        evt.consume();

    }

    // TODO add your handling code here:

}

```

```

public void cargarDias(int dia) {

    if (dia == 1) {

        for (int num = 1; num <= 28; num++) {

            String num2 = Integer.toString(num);

            CMB_DIA.addItem(num2);

        }

    }

    if (dia == 2) {

        for (int num = 1; num <= 31; num++) {

            String num2 = Integer.toString(num);

            CMB_DIA.addItem(num2);

        }

    }

    if (dia == 3) {

        for (int num = 1; num <= 30; num++) {

            String num2 = Integer.toString(num);

            CMB_DIA.addItem(num2);

        }

    }

}

public void CONECTAR() {

    String URL = "";

    String USERNAME = "root";

    String PASSWORD = "12345";

    try {

        //Class.forName("com.mysql.jdbc.Driver");

        //URL = "jdbc:mysql://localhost:3306/" + NOMBRE_BD;

        Class.forName("com.mysql.jdbc.Driver");
    }

```

```

URL = "jdbc:mysql://localhost:3306/contabilidad";

CONEXION = DriverManager.getConnection(URL, USERNAME, PASSWORD);

SENTENCIA = CONEXION.createStatement();

} catch (Exception e1) {

    JOptionPane.showMessageDialog(null, "Error de conexion " + e1, "Conexion", JOptionPane.ERROR_MESSAGE);

}

}

public void cargarDatos() {

    try {

        String SQL = "SELECT COD_ASIENTO, TIPO, FECHA FROM ASIENTOS A, CLIENTES CL WHERE A.RUT_CLIENTE = '" + RUTS + "' AND A.RUT_CLIENTE = CL.RUT_CLIENTE ORDER BY COD_ASIENTO";

        ResultSet lista = SENTENCIA.executeQuery(SQL);

        Object datos[] = new Object[3];

        tablaA.setRowCount(0);

        while (lista.next()) {

            for (int i = 0; i < 3; i++) {

                datos[i] = lista.getObject(i + 1);

            }

            tablaA.addRow(datos);

        }

    } catch (Exception l1) {

        JOptionPane.showMessageDialog(null, "Error en listar " + l1, "Listar Datos", JOptionPane.ERROR_MESSAGE);

    }

}

public void cargarTotales() {

    try {

        String SQL = "SELECT SUM(DEBE) FROM COMPROBANTES WHERE COD_ASIENTO = " + CODAS + " AND RUT_CLIENTE = '" + RUTS + "'";

        ResultSet lista = SENTENCIA.executeQuery(SQL);

        while (lista.next()) {

```

```

String DEBE = lista.getString("SUM(DEBE)");

TXT_TOTALD.setText(DEBE);

}

String SQL2 = "SELECT SUM(HABER) FROM COMPROBANTES WHERE COD_ASIENTO = " + CODAS + " AND RUT_CLIENTE = '" + RUTS + "'";

ResultSet lista2 = SENTENCIA.executeQuery(SQL2);

while (lista2.next()) {

    String HABER = lista2.getString("SUM(HABER)");

    TXT_TOTALH.setText(HABER);

}

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, "Error en Totales " + ex, "COMPROBANTES", JOptionPane.ERROR_MESSAGE);

}

}

public void cargarDatosC() {

    try {

        String SQL = "SELECT C.COD_COMPROBANTE, CU.NOM_CUENTA, C.TIPO, C.NUM_DOC, C.GLOSA, C.DEBE, C.HABER FROM CLIENTES CL,
COMPROBANTES C, ASIENTOS A, CUENTAS CU "

        + "WHERE C.RUT_CLIENTE = '" + RUTS + "' AND C.COD_ASIENTO = " + CODAS + " AND C.COD_CUENTA = CU.COD_CUENTA AND
A.RUT_CLIENTE = CL.RUT_CLIENTE AND C.RUT_CLIENTE = CL.RUT_CLIENTE AND C.RUT_CLIENTE = A.RUT_CLIENTE AND C.COD_ASIENTO =
A.COD_ASIENTO ORDER BY C.COD_COMPROBANTE";

        ResultSet lista = SENTENCIA.executeQuery(SQL);

        Object datos[] = new Object[7];

        tablaCO.setRowCount(0);

        while (lista.next()) {

            for (int i = 0; i < 7; i++) {

                datos[i] = lista.getObject(i + 1);

            }

            tablaCO.addRow(datos);

        }

    }

}

```

```

    } catch (Exception l1) {

        JOptionPane.showMessageDialog(null, "Error en listar " + l1, "Listar Datos", JOptionPane.ERROR_MESSAGE);

    }

    cargarTotales();
}

public void deshabilitar() {

    TXT_CUENTA.disable();

    TXT_TIPO.disable();

    TXT_DOC.disable();

    TXT_GLOSA.disable();

    TXT_DEBE.disable();

    TXT_HABER.disable();

    TXT_TOTALD.disable();

    TXT_TOTALH.disable();

    CMB_CUENTA.disable();

}

public void focusear() {

    TXT_TIPO.requestFocus();

    TXT_DOC.requestFocus();

    TXT_GLOSA.requestFocus();

    TXT_DEBE.requestFocus();

    TXT_HABER.requestFocus();

    TXT_TOTALD.requestFocus();

    TXT_TOTALH.requestFocus();

    CMB_CUENTA.requestFocus();

}

public void habilitar() {

```

```

CMB_CUENTA.enable();

TXT_TIPO.enable();

TXT_DOC.enable();

TXT_GLOSA.enable();

TXT_DEBE.enable();

TXT_DEBE.enable();

TXT_HABER.enable();

}

public void limpiar() {

    TXT_CUENTA.setText("");

    CMB_CUENTA.setSelectedIndex(0);

    TXT_TIPO.setText("-");

    TXT_DOC.setText("0");

    TXT_GLOSA.setText("-");

    TXT_DEBE.setText("0");

    TXT_HABER.setText("0");

    TXT_TOTALD.setText("0");

    TXT_TOTALH.setText("0");

}

private void cargarCombo() {

    ResultSet LISTA = null;

    try {

        LISTA = SENTENCIA.executeQuery("SELECT NOM_CUENTA FROM CUENTAS ORDER BY NOM_CUENTA");

        while (LISTA.next()) {

            CMB_CUENTA.addItem(LISTA.getString("NOM_CUENTA"));

        }

    } catch (SQLException e2) {

        JOptionPane.showMessageDialog(null, "Error en consultar" + e2, "Conexion", JOptionPane.ERROR_MESSAGE);
    }
}

```

```

    }

}

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Seleccionar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Seleccionar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Seleccionar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Seleccionar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

}

//</editor-fold>

```

```

//</editor-fold>

//</editor-fold>

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Seleccionar().setVisible(true);

    }

});
}

// Variables declaration - do not modify

private javax.swing.JButton BTN_ACEPTAR;

private javax.swing.JButton BTN_AGREGAR;

private javax.swing.JButton BTN_BALANCE;

private javax.swing.JButton BTN_CANCELAR;

private javax.swing.JButton BTN_DIARIO;

private javax.swing.JButton BTN_ELIMINARA;

private javax.swing.JButton BTN_ELIMINARC;

private javax.swing.JButton BTN_GUARDAR;

private javax.swing.JButton BTN_MAYOR;

private javax.swing.JButton BTN_MODIFICARC;

private javax.swing.JButton BTN_SALIR;

private javax.swing.JComboBox CMB_AÑO;

private javax.swing.JComboBox CMB_CUENTA;

private javax.swing.JComboBox CMB_DIA;

private javax.swing.JComboBox CMB_MES;

private javax.swing.JDesktopPane PES_ASIENTO;

private javax.swing.JDesktopPane PES_LD;

```

```
private javax.swing.JTable TABLA_ASIENTO;

private javax.swing.JTable TABLA_COMPROBANTE;

private javax.swing.JTextField TXT_CLIENTE;

private javax.swing.JTextField TXT_CUENTA;

private javax.swing.JTextField TXT_DEBE;

private javax.swing.JTextField TXT_DOC;

private javax.swing.JTextField TXT_GLOSA;

private javax.swing.JTextField TXT_HABER;

private javax.swing.JTextField TXT_RUT;

private javax.swing.JTextField TXT_TIPO;

private javax.swing.JTextField TXT_TIPOA;

private javax.swing.JTextField TXT_TOTALD;

private javax.swing.JTextField TXT_TOTALH;

private javax.swing.JButton jButton1;

private javax.swing.JDesktopPane jDesktopPane1;

private javax.swing.JDesktopPane jDesktopPane2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel17;

private javax.swing.JLabel jLabel18;

private javax.swing.JLabel jLabel19;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel20;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;
```

```

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JTabbedPane jTabbedPane2;

// End of variables declaration
}

```

Seleccionar Fecha Inicio/Termino de Reportes

```

import java.sql.Connection;

import java.util.Date;

import java.sql.DriverManager;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.HashMap;

import java.util.Map;

import javax.swing.JComboBox;

import javax.swing.JOptionPane;

import static javax.swing.WindowConstants.DISPOSE_ON_CLOSE;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.util.JRLoader;

import net.sf.jasperreports.view.JasperViewer;

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 *
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Valentaiguer
 */
public final class Imprimir extends javax.swing.JFrame {

    private Statement SENTENCIA;

    private Connection CONEXION;

    public String rut;

    public int sw;

    /**
     * Creates new form Imprimir
     *
     * @param rut
     * @param sw
     */
    public Imprimir(String rut, int sw) {

        initComponents();

        this.rut = rut;

        this.sw = sw;

        CONECTAR();

        TXT_RUT.setText(rut);

        TXT_RUT.setVisible(false);

        this.setLocationRelativeTo(null);
    }
}

```

```

        setTitle("PERIODO");
    }

    private Imprimir() {

        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel4 = new javax.swing.JLabel();

        jLabel3 = new javax.swing.JLabel();

        jPanel1 = new javax.swing.JPanel();

        jLabel1 = new javax.swing.JLabel();

        jLabel2 = new javax.swing.JLabel();

        CMB_DIA_INICIO = new javax.swing.JComboBox();

        CMB_MES_INICIO = new javax.swing.JComboBox();

        jLabel5 = new javax.swing.JLabel();

        CMB_DIA_FIN = new javax.swing.JComboBox();

        CMB_MES_FIN = new javax.swing.JComboBox();

        CMB_AÑO_FIN = new javax.swing.JComboBox();

        BTN_ACEPTAR = new javax.swing.JButton();

        jButton2 = new javax.swing.JButton();

        CMB_AÑO_INICIO = new javax.swing.JComboBox();

        TXT_RUT = new javax.swing.JTextField();

```

```

jLabel4.setText("jLabel4");

jLabel3.setText("jLabel3");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("INGRESE PERIODO");

jLabel2.setText("DESDE:");

CMB_DIA_INICIO.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09",
"10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" }));

CMB_DIA_INICIO.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        CMB_DIA_INICIOActionPerformed(evt);

    }

});

CMB_MES_INICIO.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09",
"10", "11", "12" }));

jLabel5.setText("HASTA:");

CMB_DIA_FIN.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
"11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" }));

CMB_MES_FIN.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
"11", "12" }));

CMB_AÑO_FIN.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "2008", "2009", "2010", "2011", "2012", "2013", "2014",
"2015", "2016", "2017", "2018" }));

CMB_AÑO_FIN.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        CMB_AÑO_FINActionPerformed(evt);

    }

});

BTN_ACEPTAR.setText("ACEPTAR");

BTN_ACEPTAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BTN_ACEPTARActionPerformed(evt);

    }

});

```



```

        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(jPanel1Layout.createSequentialGroup())

                .addComponent(CMB_DIA_INICIO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

                .addComponent(CMB_MES_INICIO, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(jPanel1Layout.createSequentialGroup())

                .addComponent(CMB_DIA_FIN, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

                .addComponent(CMB_MES_FIN, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(CMB_AÑO_INICIO, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(CMB_AÑO_FIN, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(jPanel1Layout.createSequentialGroup())

            .addComponent(jButton2)

            .addGap(55, 55, 55)

            .addComponent(BTN_ACEPTAR)

            .addGap(20, 20, 20)))

        .addGroup(jPanel1Layout.createSequentialGroup())

            .addGap(70, 70, 70)

            .addComponent(TXT_RUT, javax.swing.GroupLayout.PREFERRED_SIZE, 213, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addContainerGap(45, Short.MAX_VALUE))

    );

```

```

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(TXT_RUT, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(32, 32, 32)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(CMB_DIA_INICIO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(CMB_MES_INICIO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(CMB_AÑO_INICIO, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGap(38, 38, 38)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(CMB_AÑO_FIN, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(CMB_MES_FIN, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(CMB_DIA_FIN, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5))

.addGap(40, 40, 40)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(BTN_ACEPTAR)

```

```

        .addComponent(jButton2))

        .addContainerGap(48, Short.MAX_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

);

pack();

} // </editor-fold>

private void BTN_ACEPTARActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        if (sw == 1) {

            String FECHA_INICIO = "";

            String FECHA_TERMINO = "";

            FECHA_INICIO = CMB_AÑO_INICIO.getSelectedItem() + "-" + CMB_MES_INICIO.getSelectedItem() + "-" +
CMB_DIA_INICIO.getSelectedItem();

            FECHA_TERMINO = CMB_AÑO_FIN.getSelectedItem() + "-" + CMB_MES_FIN.getSelectedItem() + "-" + CMB_DIA_FIN.getSelectedItem();

            SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

            Date parsed_inicio = format.parse(FECHA_INICIO);

            Date parsed_termino = format.parse(FECHA_TERMINO);

            JOptionPane.showMessageDialog(null, parsed_inicio + " " + parsed_termino);

            JasperReport reporte = null;

            String path = "src\\Reportes\\PruebaDiario.jasper";

```

```

Map parametros = new HashMap();

parametros.put("RUT_CLIENTE", TXT_RUT.getText());

parametros.put("FECHA_INICIO", parsed_inicio);

parametros.put("FECHA_TERMINO", parsed_termino);

reporte = (JasperReport) JRLoader.loadObjectFromFile(path);

JasperPrint jprint = JasperFillManager.fillReport(reporte, parametros, CONEXION);

JasperViewer view = new JasperViewer(jprint, false);

view.setDefaultCloseOperation(DISPOSE_ON_CLOSE);

view.setVisible(true);

}

if (sw == 2){

    String FECHA_INICIO = "";

    String FECHA_TERMINO = "";

    FECHA_INICIO = CMB_AÑO_INICIO.getSelectedItem() + "-" + CMB_MES_INICIO.getSelectedItem() + "-" +
CMB_DIA_INICIO.getSelectedItem();

    FECHA_TERMINO = CMB_AÑO_FIN.getSelectedItem() + "-" + CMB_MES_FIN.getSelectedItem() + "-" + CMB_DIA_FIN.getSelectedItem();

    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

    Date parsed_inicio = format.parse(FECHA_INICIO);

    Date parsed_termino = format.parse(FECHA_TERMINO);

    JOptionPane.showMessageDialog(null, parsed_inicio + " " + parsed_termino);

    JasperReport reporte = null;

    String path = "src\\Reportes\\PruebaMayor.jasper";

    Map parametros = new HashMap();

    parametros.put("RUT_CLIENTE", TXT_RUT.getText());

    parametros.put("FECHA_INICIO", parsed_inicio);

    parametros.put("FECHA_TERMINO", parsed_termino);

    reporte = (JasperReport) JRLoader.loadObjectFromFile(path);

    JasperPrint jprint = JasperFillManager.fillReport(reporte, parametros, CONEXION);

    JasperViewer view = new JasperViewer(jprint, false);

```

```

        view.setDefaultCloseOperation(DISPOSE_ON_CLOSE);

        view.setVisible(true);

    }

    CONEXION.close();

} catch (Exception E) {

    JOptionPane.showMessageDialog(null, "ERROR: " + E);

}

}

private void CMB_AÑO_FINActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void CMB_DIA_INICIOActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    dispose();

}

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

```

```

        if ("Nimbus".equals(info.getName())) {

            javax.swing.UIManager.setLookAndFeel(info.getClassName());

            break;

        }

    }

} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(Imprimir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(Imprimir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

    java.util.logging.Logger.getLogger(Imprimir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

    java.util.logging.Logger.getLogger(Imprimir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Imprimir().setVisible(true);

    }

});

}

public void cargarDias(JComboBox combo) {

    int i;

    for (i = 1; i <= 31; i++) {

        String i1 = Integer.toString(i);

        combo.addItem(i1);

    }

}

```

```

}

public void cargarMes(JComboBox combo) {

    int i;

    for (i = 1; i <= 12; i++) {

        String i1 = Integer.toString(i);

        combo.addItem(i1);

    }

}

public void cargarAños(JComboBox combo) {

    int i;

    for (i = 2014; i <= 2018; i++) {

        String i1 = Integer.toString(i);

        combo.addItem(i1);

    }

}

public void CONECTAR() {

    String URL = "";

    String USERNAME = "root";

    String PASSWORD = "12345";

    try {

        //Class.forName("com.mysql.jdbc.Driver");

        //URL = "jdbc:mysql://localhost:3306/" + NOMBRE_BD;

        Class.forName("com.mysql.jdbc.Driver");

        URL = "jdbc:mysql://localhost:3306/contabilidad";

        CONEXION = DriverManager.getConnection(URL, USERNAME, PASSWORD);

        SENTENCIA = CONEXION.createStatement();

    } catch (Exception e1) {

        JOptionPane.showMessageDialog(null, "Error de conexion " + e1, "Conexion", JOptionPane.ERROR_MESSAGE);
    }
}

```

```
    }  
}  
  
// Variables declaration - do not modify  
  
private javax.swing.JButton BTN_ACEPTAR;  
  
private javax.swing.JComboBox CMB_AÑO_FIN;  
  
private javax.swing.JComboBox CMB_AÑO_INICIO;  
  
private javax.swing.JComboBox CMB_DIA_FIN;  
  
private javax.swing.JComboBox CMB_DIA_INICIO;  
  
private javax.swing.JComboBox CMB_MES_FIN;  
  
private javax.swing.JComboBox CMB_MES_INICIO;  
  
private javax.swing.JTextField TXT_RUT;  
  
private javax.swing.JButton jButton2;  
  
private javax.swing.JLabel jLabel1;  
  
private javax.swing.JLabel jLabel2;  
  
private javax.swing.JLabel jLabel3;  
  
private javax.swing.JLabel jLabel4;  
  
private javax.swing.JLabel jLabel5;  
  
private javax.swing.JPanel jPanel1;  
  
// End of variables declaration  
}
```