O/N. 3
C29
43204

Autopista de la información

PROGRAMA:

" POLÍTICAS DE SOPORTE PARA LA INTRANET DE GOBIERNO – AUTOPISTA DE LA INFORMACIÓN "

PROYECTO 5:

"SOPORTE DE SISTEMAS PARA GESTION INTERNA"

TOMO II: "RENDICIONES DE CUENTAS"

CONSEJO FEDERAL DE INVERSIONES
BIBLIOTECA
Dr. Manuel Belgrano

EN LINEA GOBIERNO DE LA PROVINCIA DE SAN LUIS

TOMO II

PROGRAMA: "RENDICIONES DE CUENTAS"

Índice **Página**

ACTIVIDAD 4: Relevamiento y Definición de Requerimientos del Negocio y Documentación 3
4.1 OBJETIVO 3
4.2 DESARROLLO 3
4.3 DESCRIPCIÓN DE LA DOCUMENTACIÓN INVOLUCRADA 7
4.4 DESARROLLO DEL CIRCUITO FÍSICO 15
ACTIVIDAD 5: Definición, Análisis y Diseño del Modelo de Datos 18
5.1 OBJETIVO 18
5.2 DESARROLLO DEL MODELO ENTIDAD RELACIÓN 18
5.3 MODELO DE ENTIDAD RELACIÓN 21
ACTIVIDAD 6: Definición de una arquitectura técnica de Hardware y Software 22
6.1 OBJETIVO 22
6.2 INTRODUCCIÓN 22
6.3 ARQUITECTURA DE BASE PARA LA SOLUCIÓN 23
6.3.1 ARQUITECTURA HARDWARE DE BASE 23
6.3.2 ARQUITECTURA DEL SOFTWARE DE BASE 26
6.4 INGENIERÍA DE RENDICIÓN DE CUENTAS 31
ACTIVIDAD 7: Desarrollo de Prototipos Funcionales 32
7.1 OBJETIVO 32
7.2 INTRODUCCIÓN 32
7.3 DESCRIPCIÓN DE MÓDULOS 32
7.3.1 FORMS 32
7.3.2 REPORTES 83
ACTIVIDAD 8: Prueba de integración del sistema con otros sistemas externos y corrección de fallas funcionales 105

21/03 20 11 01
Liliana

8.1 OBJETIVO 105
8.2 INTRODUCCIÓN 105
8.3 DESCRIPCIÓN DE MODULOS 105

momento crucial de cada entrega de dinero realizada bajo las supervisiones obligatorias e imprescindibles en la rendición de cuenta, del fin o destino designado a cada monto que forma parte de un total de dinero entregado, así poder controlar que no haya irregularidades en las entidades, a las cuales se les brinda una ayuda económica.

Debido a que cada movimiento en una salida de dinero debe ser totalmente transparente y registrada bajo dicha rendición, será el motivo por el cual estos controles darán seguridad en cada uno de los movimientos propiamente dicho, siendo esta la causa principal a la que los módulos apuntan a supervisar.

Durante esta etapa de análisis, nos enfrentamos con una escasa participación de utilidades tecnológicas en los pasos para la realización de rendiciones de cuentas del estado, habiendo comprobado que las actividades son poco visibles y los procesos por demás ineficaces y lentos en su desarrollo, que hacen casi imposible controlar si existe irregularidad alguna en las rendiciones de cuenta.

Existen diversas normas regidas por decretos, leyes, etc. que dan normativas a las entidades con respecto a como debe realizarse una rendición de cuenta, dentro de estas normativas figuran, las documentaciones que deben presentar, los tiempos en los cuales deben realizarse las rendiciones y otras prioridades que se deben respetar y cumplimentar, cabe destacar que a las entidades que falten a las normas vigentes, se las puede intimar, multar o sancionar dejándoles así inactivos los próximos pagos.

Respecto a los puntos críticos y debilitados en el manejo manual actual, especificamos e identificamos que existen personas o entidades tanto públicas como privadas que reciben subsidios o aportes del Tesoro Provincial, estas

personas o entidades son las que se ven en la obligación vigente a realizar las rendiciones de cuenta, siendo así las mismas las que serán supervisadas con un circuito que se rige por los módulos de desarrollo, y que no podrán faltar a ningún paso que deje cierta incertidumbre de la operación realizada.

Estos procesos deben ser transparentes, concretos, rápidos y eficaces. Con la incorporación de aplicaciones informáticas, se podrían realizar diversos controles, como por ejemplo que la entidad a la que se le dio un aporte realice la respectiva rendición de cuentas en un plazo no mayor a treinta días de la entrega del dinero.

Las características del manejo actual que están siendo estudiadas para ser reemplazadas por la aplicación de manejos informáticos, son aquellas transiciones y caminos conexos en las distintas áreas siendo efectuadas desde la entrega de dinero a las entidades hasta la recepción con conformidad de la rendición de cuenta de la misma.

Además en esta actividad disgregamos la ley (y su decreto correspondiente) que trata sobre Contabilidad, Administración y Control Público, desarrollando y describiendo información que apunta fundamentalmente a **Rendición de Cuentas.**

Hemos analizado dicha ley y estudiado como informatizar el sistema para obtener una efectiva Rendición de Cuentas de todos los agentes y personas que administren fondos que pertenezcan al estado. Estos responderán directamente por los daños que por su culpa o negligencia sufran las finanzas públicas y estarán sometidos al control de la Contaduría General y a la jurisdicción del Tribunal de Cuentas en los términos de la Presente Ley. Además la mencionada responsabilidad alcanza en general a todo agente o persona a quien se haya

confiado el cometido de recaudar, percibir, invertir, pagar o custodiar fondos, valores u otros bienes pertenecientes o administrados por el Estado alcanzando asimismo a aquellos que tomen injerencia en la citadas funciones sin tener autorización legal para hacerlo.

La responsabilidad se extiende a la irregular gestión de los créditos del Estado, a la entrega o utilización indebida de bienes, a los recursos que no se perciban en tiempo y forma, a los intereses o multas que deban pagarse y a la falta de un sistema contable adecuado y estructurado de acuerdo con los lineamientos dados por al contaduría general.

Cuando la responsabilidad alcance al Gobernador, Ministros, Secretarios, Subsecretarios y otros financieros designados con el acuerdo de la Legislatura, el Tribunal de Cuentas una vez determinada tal responsabilidad deberá comunicar a esta última, tal circunstancia.

Es oportuno aclarar que el ámbito cuya aplicación surtirá efecto va más allá de la Administración Central ya que involucra también a la Administración Descentralizada integrada por los entes autárquicos y de regulación de servicios, Empresas y Sociedades del Estado, el Poder Legislativo, el Poder Judicial, el Tribunal de Cuentas y respecto a las Administraciones Municipales regirá esta ley para las Comisiones y Comisionados Municipales y supletoriamente para las Intendencias con Consejo Deliberante, en los casos de silencio u oscuridad de sus regímenes legales.

Quedan asimismo sujetas al alcance de la presente Ley y a la competencia de sus órganos de control, las personas físicas y jurídicas, tanto públicas como privadas, incluyendo las de carácter público no estatal, a las que se les hubiere asignado

recursos para un objeto determinado en cuanto a la responsabilidad de rendir cuentas en tiempo y forma, de la aplicación de los mismos y del cumplimiento de las condiciones establecidas en el acto de asignación. También quedan comprendidos, los programas asistidos financieramente por organismos nacionales e internacionales. En el caso en que la provincia hubiera intervenido para su asistencia, regirán las normas específicas nacionales o internacionales que pudieran ser prevalentes a esta ley.

Cabe destacar que las reparticiones involucrados en el relevamiento son de un perfil técnico contable-económico ya que los sectores: Presupuestario, Contaduría, Tesorería y Crédito Público nos han brindado información de cómo estarán interrelacionados en el control del sistema, para poder armar las pautas de como cierra el circuito contable a través de "**Rendiciones de Cuentas**".

4.3. DESCRIPCIÓN DE LA DOCUMENTACIÓN INVOLUCRADA

Detallamos la Ley Nº 5.172 que incluye el decreto de reglamento 819/00 propiamente dicho destacando exclusivamente los artículos que están relacionados con "Rendición de Cuentas" y como la implementación del sistema tendrá incidencia directa sobre los involucrados.

Los primeros artículos de la Ley son introductorios, pero están involucrados con Rendición de Cuentas. Los artículos son los siguientes:

Art. 1º.-***Materia***. *Es materia de la presente Ley de Contabilidad, Administración y Control Público el régimen de los sistemas relativos a la obtención y administración de los diversos recursos y su aplicación para el cumplimiento de las funciones y objetivos de los organismos públicos de la Provincia.*

Establece asimismo, pautas para el diseño del sistema de control y del régimen del sistema de responsabilidad administrativa.

Art.2º.-***Objeto.*** *Es de objeto de esta Ley la aplicación y desarrollo de un modelo administrativo basado en las pautas siguientes, las que deberán tenerse presentes para su reglamentación e implementación:*

a)*Aplicación integral de los principios generales señalados en el Art. 3º para la gestión de los recursos públicos.*

b) *Sistematización de la gestión integral de los recursos públicos abarcando las etapas de programación de acciones, procesos de ejecución y evaluación de resultados.*

c) *Implementación del sistema de control en los términos del Título VII, de la presente Ley.*

d) *Instrumentación de planes y políticas públicas mediante las acciones y las asignaciones previstas en los presupuestos públicos.*

e) *Inclusión en el régimen de responsabilidad administrativa de la evaluación de los resultados de la gestión de los recursos y del cumplimiento de objetivos.*

f) *Implementación de un sistema de contabilidad que integre la información de los sistemas operantes en el sector público provincial, incluyendo procedimientos de información gerencial adecuados a las características de cada organismo.*

Del análisis de estos dos primeros artículos podemos observar que se encuentran **eslabonados** los distintos puntos que se detallan, concluyendo en la necesidad de la implementación de un sistema contable para la integración general de la

información, en el último eslabón de esta **cadena** se destaca la participación de la Rendición de Cuentas.

En los siguientes articulos se explicita puntualmente los conceptos de la Rendición de Cuentas en la ley propiamente dicha.

Art.58°.- ***Rendición.*** *Los agentes de la Administración Provincial a quienes se les haya confiado el cometido de reanudar, invertir, percibir, pagar y en general custodiar fondos o valores del Tesoro, están obligados a rendir cuenta documentada de su gestión y llevar el sistema contable estructurado en base a los lineamientos dado por la Contaduría General.*

Dicha obligación alcanza asimismo a las personas o entidades tanto públicas como privadas que reciban subsidios o aportes del Tesoro Provincial. Los gastos reservados previstos en la Ley de Presupuestos no están sujetos a rendición.

Art.59°.- ***Fianza.*** *El Poder Ejecutivo determinará los agentes que deban rendir fianza y la forma de prestación de la misma.*

Art.60°.- ***Reglamento para rendiciones.*** *El Poder Ejecutivo deberá dictar el reglamento de rendiciones de cuentas en el que fijará los plazos y procedimientos para las mismas. Las normas complementarias y los modelos de documentación a utilizar serán establecidas por el Tribunal de Cuentas, previa opinión de la Contaduría General.-*

El plazo máximo que podrán establecerse para presentar una rendición de cuentas es de TREINTA (30) días. No podrá efectuarse una nueva entrega de fondos a un responsable que tenga rendiciones pendientes

por el mismo concepto, salvo expresa ampliación del plazo hasta TREINTA (30) días por parte del Poder Ejecutivo.

Art.61°.- ***Presentación de rendiciones.*** *Las rendiciones de cuenta deberán presentarse ante la Contaduría General de la Provincia. En caso de retardo, el citado organismo exigirá su presentación empleando gradualmente las siguientes medidas de apremio:*

a) Requerimiento conminatorio con fijación de plazo.

b) Comunicación al titular del respectivo organismo y al Tribunal de Cuenta.

c) Suspensión de todo nuevo pago a la dependencia incluyendo los sueldos del responsable remiso.

d) *Formación de oficio de la cuenta atrasada y su remisión con los antecedentes necesarios al Tribunal de Cuentas.*

Art.62°.- ***Análisis de la rendición.*** *La Contaduría General analizará cada rendición de cuentas en sus aspectos legales, formales, numéricos y documentales, remitiéndolas al Tribunal de Cuentas para su fallo definitivo.*

En caso de observarse algún documento o de ser necesario corregir errores numéricos o de ampliarse la información, la Contaduría General lo comunicará al responsable, quien en un plazo de TREINTA (30) días corridos regularizará la situación, pudiendo dicho organismo ampliar el plazo hasta TREINTA (30) días en casos justificados. Si no se cumplieran los casos, serán de aplicaciones las medidas prescriptas en el artículo anterior.

Art.63°.- ***Remisión al Tribunal de Cuentas.*** *La Contaduría General deberá remitir al Tribunal de Cuentas las rendiciones de cuentas realizadas, de acuerdo con los prescriptos en el artículo anterior, en un plazo de TREINTA (30) días corridos de recibidas sin objeciones. El citado plazo podrá ampliarse en QUINCE (15) días corridos, más cuando mediaren razones que los justifiquen.*

El Decreto N°819/2000 del 23 de Marzo del 2000 reglamenta a la Ley en general y particularmente a los artículos relacionados con "Rendiciones de Cuenta".

Art.58°.- **RENDICIÓN**

Las Contadurías de los Poderes Legislativo, Judicial, Tribunal de Cuentas y Entidades Autárquicas, aplicarán para sus respectivas jurisdicciones, un procedimiento similar al descrito en el artículo 58 de la Ley.

Para el caso de tratarse de algunas de las entidades del segundo párrafo del artículo 58 de la Ley, se procederá a la suspensión de todo subsidio o aporte del Tesoro Provincial.

Art.59°.-.**FIANZA**

Todos aquellos agentes que determine el Poder Ejecutivo Provincial mediante decreto, están obligados a rendir fianza a través de un depósito en un fondo de garantías por el monto correspondiente a cinco (5) salarios del agente, el cual será cubierto por un descuento mensual del 20% hasta cubrir dicha suma.

La garantía mencionada en el párrafo anterior no libera a tales agentes de

la correspondiente responsabilidad por los daños y perjuicios ocasionados al fisco.

Cuando existan razones fundadas y se demuestre la conveniencia del procedimiento, la fianza podrá cubrirse por otro medio de procedimiento idóneo, en lugar del descrito en el apartado anterior.-

Art.60°.- **REGLAMENTO PARA RENDICIONES**

El reglamento de rendición de cuentas, será elevado ante el Poder Ejecutivo para su aprobación, por la Dirección Provincial de Contaduría General, quien previamente solicitará al Tribunal de Cuentas el dictado de las normas complementarias y los modelos de documentación a utilizar y emitirá su opinión al respecto.-

Art. 61°.- **PRESENTACIÓN DE RENDICIONES**

Las Contadurías de los Poderes Legislativo, Judicial, Tribunal de Cuentas y Entidades Autárquicas, aplicarán para sus respectivas jurisdicciones, un procedimiento similar al descrito en el artículo 61 de la Ley.

El plazo fijado en el inciso a) de la Ley, será de cinco (5) días.

Para el caso de tratarse de alguna de las entidades a que se refiere el artículo 58, segundo párrafo de la Ley y a los efectos del inc. c) del artículo 61 de la Ley, se suspenderán los subsidios y/o aportes del Tesoro.

En virtud de lo establecido en el inciso c) del artículo de la Ley se entiende por "dependencias", a las unidades ejecutoras.-

Art. 62°.- **ANÁLISIS DE LA RENDICIÓN**

1)La Dirección Provincial de Contaduría General, no formulará cargos provisorios cuando el incumplimiento surgiera de una observación formal, es decir, cuando no haya un perjuicio patrimonial al Estado.

Son observaciones formales las que surgen por no haber rendido cuentas conforme a disposiciones legales, reglamentarias o a modelos establecidos no obstante existir documentación que justifica la aplicación de partidas presupuestarias, la prestación de servicios comprobados y/o ingresos de bienes al patrimonio del Fisco, siempre y cuando las operaciones aritméticas estén hechas con exactitud.

Entre las observaciones formales se considerarán:

a)Falta de firma del funcionario responsable al pie del comprobante siempre que exista en la documentación constancia de su intervención y autorización.

b)Falta de orden de provisión u orden de trabajo o servicio.

c)Foliatura incorrecta.

d)Falta de formulario de reintegro, estando demostrado el ingreso.

e)Falta de autorización (firma del superior) en la orden de salida.

f)En los recibos y/o facturas, falta del precio unitario.

g)En los recibos y/o facturas emanadas del proveedor, falta del importe en letras.

h)En los recibos y/o facturas emanadas del proveedor, falta de firma del mismo con leyenda alusiva de haber recibido el importe, siempre que exista sello pagado o constancia que el comprobante es de contado.

i)Falta, en el expediente de rendición, copia del instrumento legal específico que autorice el o los gastos en su caso (resoluciones, decretos, otros).

En los casos de gastos por publicaciones, falta de la copia de la publicación.

Es observación sustancial haber efectuado en la rendición de cuentas un uso ilegítimo de partidas presupuestarias, por no poder comprobarse el uso de las mismas y cualquier otra de la cual deriva daño para la hacienda pública o incumplimiento de la Ley.

2)La Dirección Provincial de Contaduría General realizará, por intermedio de su área de responsables, la determinación provisoria de responsabilidad, llamada también cargo provisorio, observando el siguiente orden:

A)Enunciado:

1)Nombre y apellido del o de los responsables.

2)Domicilio real.

3)Matricula individual y CUIL.

4)Número de la orden de pago o hechos que origine la responsabilidad del agente o persona involucrada.

5)Monto y destino que correspondiera a los fondos.

6)Fecha de su recepción en la Tesorería General.

B)Considerando:

1)Normas a las que debió ajustarse la inversión de los fondos.

2)Métodos que en cambio siguió el responsable en violación de

aquellas normas.

3)Discrepancias existentes entre las normas correctas y la aplicación real dada a los fondos.

4)Daño causado al Fisco.

5)Análisis de los comprobantes observados y /o rechazados.

6)En general explicar todos los fundamentos que dan lugar a la formulación del cargo provisorio.

C)Conclusiones:

1)Monto neto del Cargo Provisorio.

2)Intereses.

3)Total

D)Dicho Cargo Provisorio con toda la documentación correspondiente será elevado al Honorable Tribunal de Cuentas.-

Art. 63°.-.**REMISIÓN AL TRIBUNAL DE CUENTAS**

Sin reglamentación.-

4.4. DESARROLLO DEL CIRCUITO FÍSICO

El aplicativo debe realizar una auditoría contable a todos aquellos Beneficiarios que hubieren solicitado subsidios o aportes de fondos al estado provincial. El sistema debe controlar que los beneficiarios estén en condiciones de poder realizar estas solicitudes. Luego de que se realice el chequeo correspondiente y se proceda a la entrega del dinero, cheque o bien correspondiente, se debe volcar en la base de datos la información que sea relevante, esta información se procesa a través de un módulo llamado "Módulo de Entrega de Fondos a Instituciones"

que deberá operarlo el sector "Tesorería", cuyos datos de fundamental importancia serían: Código del Beneficiario, Fecha de Entrega, Número de Orden de Pago, Número del Expediente iniciado, Monto Entregado, Motivo del gasto y Fecha de Vencimiento, luego existen datos de menor jerarquía no siendo necesarios de describir.

Posteriormente de haberse efectuado la entrega correspondiente, existirá un proceso diario encargado de chequear los vencimientos de la entrega de los fondos (cheques) correspondientes a los subsidios o aportes de los beneficiarios, llamado "Proceso de Control de Vencimientos" que retornará a todos aquellos beneficiarios a los cuales se les haya vencido el plazo para la "Rendición de Cuenta" (dicho plazo se estima 30 días corridos luego de la fecha de la asignación de los fondos correspondiente a subsidios o aportes, este plazo puede estar sujeto a posibles modificaciones). Tal proceso deberá chequear la Fecha de Vencimiento para poder verificar si la presentación está fuera del término de la "Rendición de Cuenta".

El sistema deberá controlar también la existencia de posibles excepciones, como por ejemplo la ampliación del plazo de Rendición de Cuenta, este beneficio es otorgado únicamente por el Poder Ejecutivo de la Provincia. Para esto se deberá modificar la fecha de Vencimiento de la "Rendición de Cuenta" del monto correspondiente a los fondos del estado del subsidio o aporte realizado, operación a cargo del sector de la "Contaduría General".

En caso de haberse excedido del plazo correspondiente a la presentación de la documentación, deberá procederse a la aplicación de sanciones por parte de la Dirección de Contaduría General. Y no podrá la Institución (o Beneficiario)

solicitar ningún nuevo subsidio o aporte hasta que haya efectuado la rendición de cuenta de fondos correspondiente al subsidio o aporte adeudado. Tampoco puede solicitar ningún nuevo tipo de subsidio o aporte en el transcurso de tiempo que tenga pendiente alguna rendición aunque no haya vencido el plazo de presentación.

Se debe contemplar también ciertas particularidades en el manejo de la operatoria de la Rendición, por ejemplo existen conceptos en los cuales se debe chequear si el monto del subsidio o aporte es equivalente al monto Rendido, en este caso se cierra la operatoria, una segunda opción que puede suceder es que el monto rendido sea mayor al monto del subsidio o aporte solicitado, en este caso el beneficiario del subsidio o del aporte debe presentar la documentación avalando un gasto mayor, la repartición por su parte deberá librar una nueva orden de pago por el monto equivalente a la diferencia. Tesorería realizará luego el pago correspondiente. En una tercera opción puede suceder que el monto rendido sea menor al monto del cheque entregado, en este caso el responsable del subsidio o aporte debe presentar una boleta de depósito y los comprobantes del gasto, por su parte la repartición libra una orden de reintegro al estado y por último Tesorería cumplirá con el trámite final.

El circuito culmina cuando se realiza la "Rendición de Cuentas" por parte del beneficiario presentando éste toda la documentación requerida.

ACTIVIDAD 5

5.1. OBJETIVO

Actividad correspondiente a la generación del modelo de datos y del diseño lógico y físico detallado de la Base de Datos e identificación de los requisitos de auditoría y de información de control e integración de aplicaciones correspondiente al programa "Rendiciones de Cuentas".

5.2.DESARROLLO DEL MODELO ENTIDAD RELACIÓN

En esta actividad mostramos el modelo de Datos propio de "Rendición de Cuentas", y además publicamos en el Anexo II el MER (Modelo de Entidad Relación) correspondiente. Este módulo mantiene una conexión directa con el sistema contable actualmente en producción, por este motivo debemos presentar en el modelo algunas entidades previamente definidas para mantener una consistencia en la relación.

El circuito comienza en la repartición dando de alta al expediente (la entidad que lo soporta es CO_EXPEDIENTE relacionando a todas las demás entidades satélites, estas entidades tienen ya tablas asociadas utilizadas para el sistema contable actual en producción) relativo al subsidio o aporte solicitado por la institución (o por algún tercero, beneficiario referente a una institución, este beneficiario está soportado en la entidad CO_BENEFICIARIO la cual acepta todos los datos referente al beneficiario), esta tarea la desarrolla la Repartición Gubernamental con la cual está involucrado tal beneficiario, en esa operativa se da de alta la orden de pago correspondiente, soportado por la entidad

CO_REGISTRO_ORDEN asociado al ejercicio actual cuya entidad es NGR_EJERCICIO.

Teniendo Tesorería General la orden de pago a su disposición, efectúa la entrega de los fondos a través de un cheque, monto correspondiente al subsidio o aporte, (cuyos datos se darán de alta sobre la entidad "CO_REGISTRO_ORDEN" dentro de los campos CARGO, FECHA_CARGO y FECHA_VTO_CARGO), cuando se realiza la Rendición de Cuentas, módulo utilizado en Contaduría, se le asocia a "CO_REGISTRO_ORDEN" la entidad "CO_RC_RENDICION", cuyos datos dejarán constancia de la Rendición, luego suponiendo la posibilidad que dicho beneficiario solicite un nuevo subsidio o aporte sin haber efectuado la rendición, el sistema no lo aceptará, ya que no ha sido creado aún en la base un Registro que pertenezca a la tabla "CO_RC_RENDICIONES" (en nuestra diagrama representada por la entidad "CO_RC_RENDICION"), puede suceder el caso que el campo "Aprobada" de la tabla "CO_RC_RENDICIONES" (en el diagrama se representa a través del atributo "Aprobada") contenga el valor "NO", ocurre que por alguna circunstancia fue presentada la rendición pero no aprobada por la Contaduría General de la Provincia. En caso de que el beneficiario solicite una ampliación de tiempo para la Rendición de Cuenta, este proceso se registrará sobre la entidad "CO_RC_AMPLIACION", cuya relación con "CO_REGISTRO_ORDEN" puede soportar "n" pedidos de Ampliaciones diferentes.

Como hemos descrito anteriormente, el sistema de "Rendición de Cuentas" está inmerso en el Sistema Contable actualmente en producción.

Todos los expedientes, soportados por la tabla CO_EXPEDIENTE, están asociados a distintos tipos de contrataciones, muchos de estos tipos de contrataciones soportados por la tabla NGR_TIPO_CONTR son factibles a rendiciones de cuentas, pero no todos los expedientes asociados a un tipo de contratación factible a rendición de cuenta la deben realizar, por ejemplo, un expediente asociado al tipo de contratación "viáticos" que lleva planilla provisoria, debe realizar la rendición; pero los expedientes de "viáticos" con planilla definitiva no deben realizar la rendición. Para que el sistema determine cuáles son los expedientes que definitivamente deben realizar la rendición de cuenta se le deberá agregar lógica en los controles previos a las órdenes de pago que se realizan en el sistema contable.

La tabla a la que se le agregará un campo para soportar lo mencionado anteriormente es NGR_CONTROL_EXP_DET, tabla en la cual se guarda el nomenclador de detalles de los controles que hay que realizar en algún expediente para los tipos de contrataciones determinadas, el campo a agregar en esta tabla es "Rendición" que es un campo lógico, el cual, en el caso en el que el campo "Opción" posea el valor "TIT", que significa que es un título, se activará el campo nuevo para poder determinar cuales son los títulos dentro de un tipo de contratación que deben llevar rendición.

Por otro lado los tipos de contrataciones asociadas a un expediente tienen diferentes etapas de control por las cuales debe pasar el expediente para poder llegar a generar una orden de pago, estas etapas están almacenadas en la tabla NGR_ETAPA_CONTOL_EXP, y por ser una relación n a m con la tabla NGR_CONTROL_EXP_DEP el nexo es a través de una tercera tabla

NGR_ETAPA_CONTROL_EXP_CTL_DEP, con ésta se completa la relación entre los controles que se le aplican a los expedientes de un tipo de contratación determinada y cada una de las etapas por las que debe pasar el expediente. En la etapa que se realiza previa a la emisión de la orden de pago se agregará la lógica que prohibirá la emisión de una orden de pago a un beneficiario que tiene un cargo de rendición de cuenta sin cerrar.

Además fue necesario modificar otras tablas propias del sistema contable como es el caso de la tabla "NGR_TIPOS_CONTR" que es la que mantiene el nomenclador de tipos de contrataciones a los cuales son asociados los expedientes. En el caso de esta tabla se agregaron dos campos que sirven para relacionar el tipo de contratación que pertenece a un expediente de pago con el tipo de contratación que pertenece a un expediente de rendición.

Por otro lado otras de las tablas del sistema contable que sufrió modificaciones fue la tabla que mantiene el nomenclador de etapas por la que tiene pasar un expediente, esta tabla es llamada "NGR_ETAPA_CONTROL_EXP" y se le agrego un campo llamado "SUBTIPO", este cambio fue necesario para indicar que etapa es la que pertenece a un control de rendición.

5.3. MODELO DE ENTIDAD RELACIÓN

El modelo gráfico de Entidad / Relación junto con la descripción de los atributos de cada entidad se muestra en el Anexo II.

ACTIVIDAD 6

6.1. OBJETIVO

Definición de una arquitectura técnica de Hardware y Software en la que deba funcionar el sistema de aplicación.

6.2. INTRODUCCIÓN

Considerando que el desarrollo y producción establecido para "Rendición de Cuenta" está basado sobre un ambiente "Cliente / Servidor" cuyo soporte de Hardware y Software será la Intranet de Gobierno (cabe recordar que los módulos se encuentran inmersos en el Sistema Contable), se destacarán en el informe características similares a la plataforma de hardware y Software de Base utilizado para el sistema de "Compras y Contrataciones".

Teniendo en cuenta que en la actualidad la Intranet de Gobierno mantiene un alto rendimiento y considerando que es una base sólida y suficiente para soportar nuevos aplicativos que se ejecutan sobre un ambiente "Cliente / Servidor", no será necesario agregar nuevas Herramientas de Base ni componentes de Hardware además de las ya detalladas en el informe anterior. En la etapa del "Testing" se probará definitivamente el servicio que presta actualmente la Intranet (constituida por Servidores, Red y Bases de Datos).

Cabe destacar que no incluiremos el Software y Hardware de Base que se vinculan con aplicativos orientados a web mostrados en el informe correspondiente a "Compras y Contrataciones" ya que no serán de utilidad para este sistema.

6.3. ARQUITECTURA DE BASE PARA LA SOLUCIÓN

Se detallarán concretamente aquellas Herramientas utilizadas actualmente en producción, pero otro software y hardware de base con características y compatibilidad similar pueden ponerse en uso aunque no tengan el mismo rendimiento. Destacando por ejemplo que la mayoría de los Sistemas Operativos utilizados en la actualidad son productos de Unix sin embargo pueden reemplazarse por otros productos que cumplan con los requisitos necesarios, como así también las bases de datos utilizadas que son producto Oracle pueden ser sustituidas por otras semejantes.

6.3.1. ARQUITECTURA HARDWARE DE BASE

Los datos están soportados en servidores ubicados en un Centro de Cómputos los cuales brindan información a todos los usuarios habilitados para el manejo de algún aplicativo en particular.

La red tiene un amplio tendido de fibra óptica por la cual acceden casi la totalidad de los usuarios, el resto tienen acceso vía módem (Existen accesos desde otras localidades de la provincia, en un futuro cercano se ha planificado una conexión total por medio de fibra o en forma satelital, o sea que se eliminaran todas las conexiones a través de modem´s). En cada organismo se encuentran centros de cableado que facilitan la administración y la expansión de la red. En la actualidad, las PC´s independientes y/o estaciones de RED, están utilizando Sistemas Operativos con ambiente Windows.

- *Server de Base de Datos Oracle8i Enterprise.*
    - ✓ Marca IBM

- ✓ Sistema Operativo AIX 4.3
- ✓ Procesadores 4 Risk 340 MHz
- ✓ Memoria RAM 3GB
- ✓ 2 Discos de 9 GB

- *Server de Aplicaciones.*
  - ✓ Marca HP
  - ✓ Sistema Operativo HP-UX 11
  - ✓ Procesadores 4 Risk 166 MHz
  - ✓ Memoria 1GB
  - ✓ 8 Discos de 9GB en Array en alta disponibilidad.
- *Switch*
  - ✓ La conexión de los usuarios se enmarca en la capa de acceso, lugar en que se ubica el switch P550, con la necesidad de desplegar un conmutador Gigabit Ethernet de alto volumen con ruteo integrado.
  - ✓ El beneficio es la conmutación de datos rápida y transparente a medida que se migra del procesador central a la red intranet.
  - ✓ Marca Lucent Technologies.
  - ✓ Modelo Cajun P550.
- *Server para Router y Aplicación Firewall.*
  - ✓ Marca HP.
  - ✓ Sistema Operativo Linux Red Hat 7.
  - ✓ Procesador Pentium III 500 MHz.
  - ✓ Memoria RAM 128MB.
  - ✓ 1 Disco de 8GB.

- *Cliente*

  - ✓ Procesador Pentium 233 MMX o superior.
  - ✓ 64 MB de memoria RAM.
  - ✓ Windows 95 o Windows NT Client 4.0 o superior.
  - ✓ 4 GB de disco rígido.
  - ✓ Monitor color de 15" o superior.
  - ✓ Adaptador de red Ethernet 10/100 Mbps.

- *Conexión de Redes*

  Toda la red interna será Ethernet 100 Mbps para los equipos servidores y 10/100Mbps para las estaciones de trabajo.

  La siguiente tabla describe las redes que serán usadas para la nueva aplicación.

	Network Protocol	Propósito
Producción Intranet (PI)	TCP/IP	Ambiente de Producción
Desarrollo Testing Intranet(DTI)	TCP/IP	Ambiente de desarrollo/Test

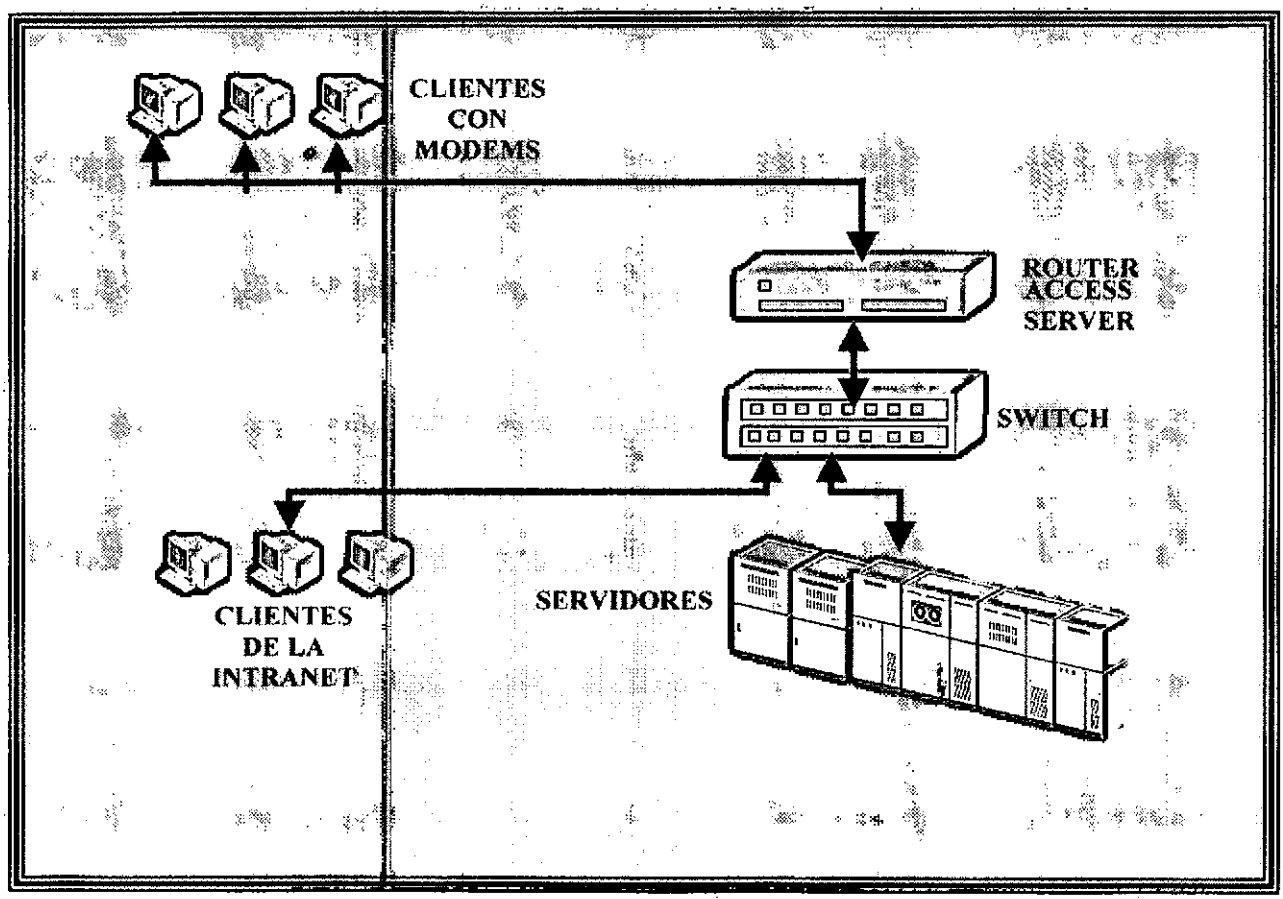
Los cálculos y la recomendación de hardware están basados en el tráfico actual que tiene la Intranet del Gobierno, tomando como referencia que el 62% del volumen es un potencial de usuarios de sistemas de aplicaciones de producción interna.

Bajando el nivel de detalle, se recomienda la separación en distintas máquinas los componentes del servidor de aplicaciones y el servidor de base de datos.

Multiplexando los servidores de aplicaciones en varias máquinas darán mayor escalabilidad y performance en el funcionamiento, siempre que el servidor de base de datos pueda soportar la carga.

En el siguiente diagrama se muestra la topología de la arquitectura. Muestra donde se ubica cada componente de hardware y como se intercomunican. El diagrama es funcional y no físico. Varios servidores "lógicos" que aparecen en el diagrama pueden llegar a estar ubicados finalmente en un único servidor físico.

Gráfico 6.1 Modelo Hardware Intranet de Gobierno



6.3.2. ARQUITECTURA DEL SOFTWARE DE BASE

En este punto describiremos el software de Base que se está utilizando hasta el momento en la intranet, el cual se encarga de mantener los datos que intervienen en distintos sistemas en forma segura y organizada.

Además del software de base que describiremos a continuación sin duda se podría utilizar otro de características similares, el cual puede no igualar la performance del software descrito, pero manteniendo su compatibilidad no generaría inconvenientes.

- *El sistema operativo AIX de Unix.*

  Éste es un sistema operativo abierto de IBM basado en una versión de UNIX, el mismo como tal es un sistema multiusuario y multitarea, lo cual quiere decir que los recursos del sistema pueden ser compartidos simultáneamente por varios usuarios, cada uno de los mismos puede tener en ejecución más de un proceso en un momento dado.

  Cuando un usuario inicia una sesión de trabajo, AIX le obliga a identificarse con su código de cuenta y contraseña, de esta forma el sistema es capaz de identificar quién es el autor de cualquier operación (acceso al sistema, creación de archivos, etc.) esto es importante ya que nos permite restringir las posibles acciones del usuario. La seguridad es posible en el momento de tomar las decisiones referentes a Sistemas Operativos, se tuvo que tener en cuenta algunas características, que de no ser así podrían afectar la performance de nuestros aplicativos.

  Debido a que se debió tener una visión futura del funcionamiento de los aplicativos, mencionaremos algunas características:

  ✓ Ambiente Cliente/Servidor.

  ✓ Manejo de espacio en grandes Bases de Datos.

  ✓ Acceso concurrente de muchos usuarios a la misma información.

  ✓ Performance en altas transacciones.

- ✓ Alta disponibilidad.
- ✓ Seguridad en accesos no autorizados.
- ✓ Conectividad a través de la red.

- *Samba.*

Es una herramienta útil para cualquiera que necesite integrar máquinas que ejecuten múltiples Sistemas Operativos. Aunque compartir una banda ancha entre múltiples Sistemas Operativos es relativamente sencillo cuando todos utilizan TCP/IP como protocolo de red, conseguir que cada una de estas máquinas en realidad lea los archivos de cada una de las otras y compartan impresoras no es tan fácil. A no ser que se utilicen este tipo de herramientas.

Samba es la ejecución de Unix del protocolo Server Message Block (SMB). Este protocolo se utiliza para permitir la interacción del sistema de archivos a través de redes de múltiples sistemas operativos y permite realizar las siguiente tareas:

- ✓ Montar carpetas pertenecientes a Windows en sus ordenadores Unix.
- ✓ Montar segmentos correspondientes al sistema de archivos de Unix en ordenadores Windows.
- ✓ Montar segmentos del sistema de archivos entre un ordenador Unix y cualquier otro Sistema Operativo que use SMB, (otra variedad de Unix o incluso un Linux).
- ✓ Imprimir en una Impresora conectada a Windows desde un Ordenador Unix.

- ✓ Imprimir en una impresora conectada a una máquina Unix desde un Ordenador Windows.
- ✓ Utilizar impresoras entre Sistemas Operativos que interpreten SMB.

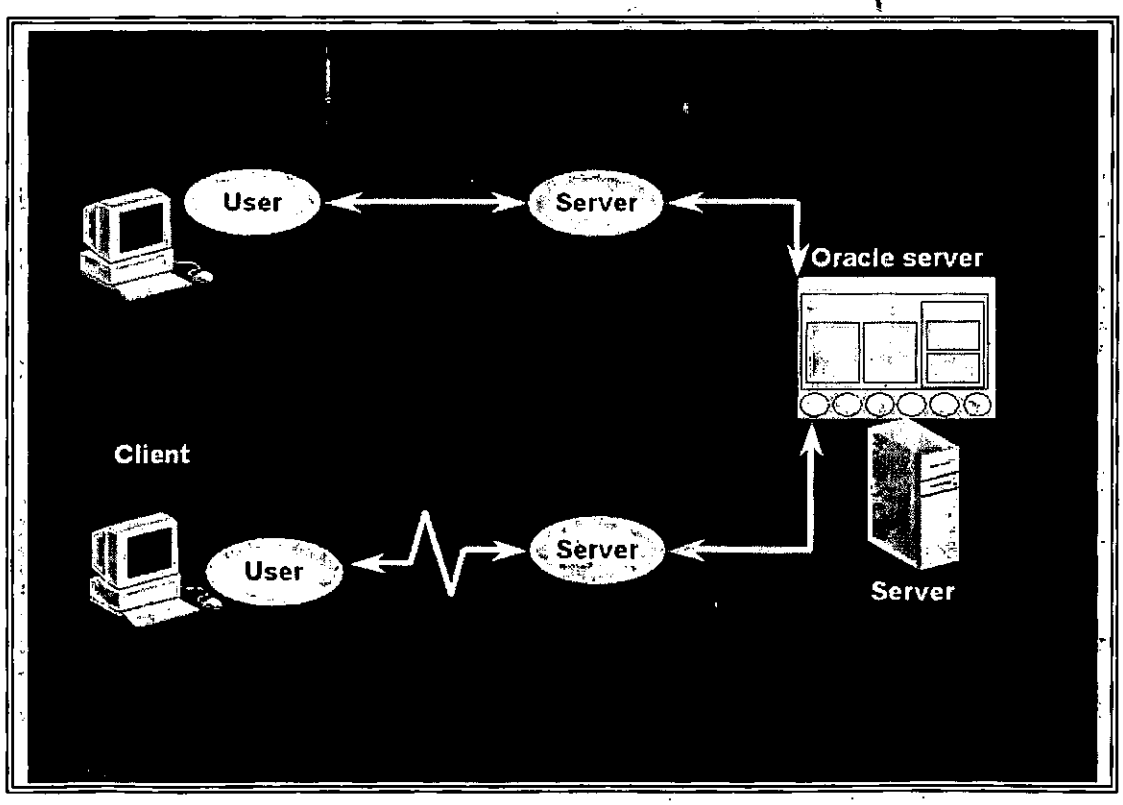
- *El Database Server*:
  - ✓ El Oracle8i Database es la base de datos que contiene el modelo relacional de la solución, posee los datos.
  - ✓ Este sistema es de administración de Base de Datos Relacionales que provee un aprovechamiento integrado, comprensivo y abierto. Oracle Server está compuesto por una Base de Datos Oracle y una Instancia de Oracle Server.
  - ✓ Un servidor de Base de Datos es la clave para resolver los problemas del manejo de información. En general, un servidor debe manejar una gran cantidad de datos en ambientes multiusuario, de esta manera muchos usuarios pueden acceder a los mismos datos concurrentemente, además debe prevenir accesos no autorizados y proveer eficientes soluciones para la recuperación de información cuando ocurran anomalías por fallas.
- *Sistema Operativo en Clientes.*
  - ✓ Los sistemas operativos en los clientes pueden ser compatibles con Windows NT 4.0 workstation, Windows 95/98 o versiones superiores. Los distintos Aplicativos también podrían operar sobre otro tipo de sistema operativo en el cliente, como Linux de la línea UNÍX, pero los resultados de performance están abiertos.
- *Runtime para los Aplicativos Oracle.*

- ✓ Los RunTime son herramientas instaladas en los clientes de sistemas con una tecnologías "Cliente/Servidor" los cuales permiten poder ejecutar aplicativos almacenados en los Servidores. Estos aplicativos no podrían correr por si solos ya que su código objeto no es ejecutable por la máquina hardware, los RunTime cumplen la función de Máquina anfitriona realizando la simulación correspondiente interpretando dichos módulos.

  En Oracle las herramientas son Forms Runtime, Reports Runtime y Graphics Runtime.

  Cabe destacar que el cliente para estos casos no debe ser una máquina "boba", sino que por el contrario necesita una implementación hardware con suficientemente buenos recursos para el soporte de dichas herramientas.

Gráfico 6.2 Database Server.



6.4. INGENIERÍA DE RENDICIÓN DE CUENTAS

El diseño del sistema contendrá operaciones con permanentes actualizaciones, debido a que el sistema está dirigido en gran parte a consultas y reportes. Supongamos la emisión de un cheque a un beneficiario, se deberá corroborar que el mismo no tenga pendiente alguna rendición anterior, además en forma diaria deberán reportarse todas las Rendiciones cuyo vencimiento sean hasta el día de la fecha.

El Software proporcionará las siguientes particularidades:

- Consulta sobre rendiciones pendientes.
- Consulta sobre rendiciones vencidas.
- Consulta sobre rendiciones aprobadas.
    - Generación de cargos a beneficiarios que deban presentar rendiciones.
- Impresión de notas tipo para la notificación a los beneficiarios en infracción.

ACTIVIDAD 7

7.1. OBJETIVO

Desarrollo de prototipos funcionales de módulos principales correspondientes al sistema "Rendiciones de Cuentas".

7.2. INTRODUCCIÓN

En esta actividad mostramos los módulos desarrollados que forman parte del sistema "Rendiciones de Cuentas".

Cabe destacar que estos módulos fueron desarrollados en Oracle Developer/2000 Forms Designer for Windows 95 / NT y los reportes que también formarán parte de este sistema fueron desarrollados en Developer/2000 Reports Designer for Windows 95 / NT, ya que forman parte del Sistema Contable de control y ejecución presupuestaria que actualmente esta en producción en el Gobierno de la Provincia de San Luis, el cual ha sido desarrollado con las mismas herramientas.

También mostramos pantallas de consultas sobre las rendiciones y una serie de menús de opciones por los cuales se accede para llegar a estas pantallas.

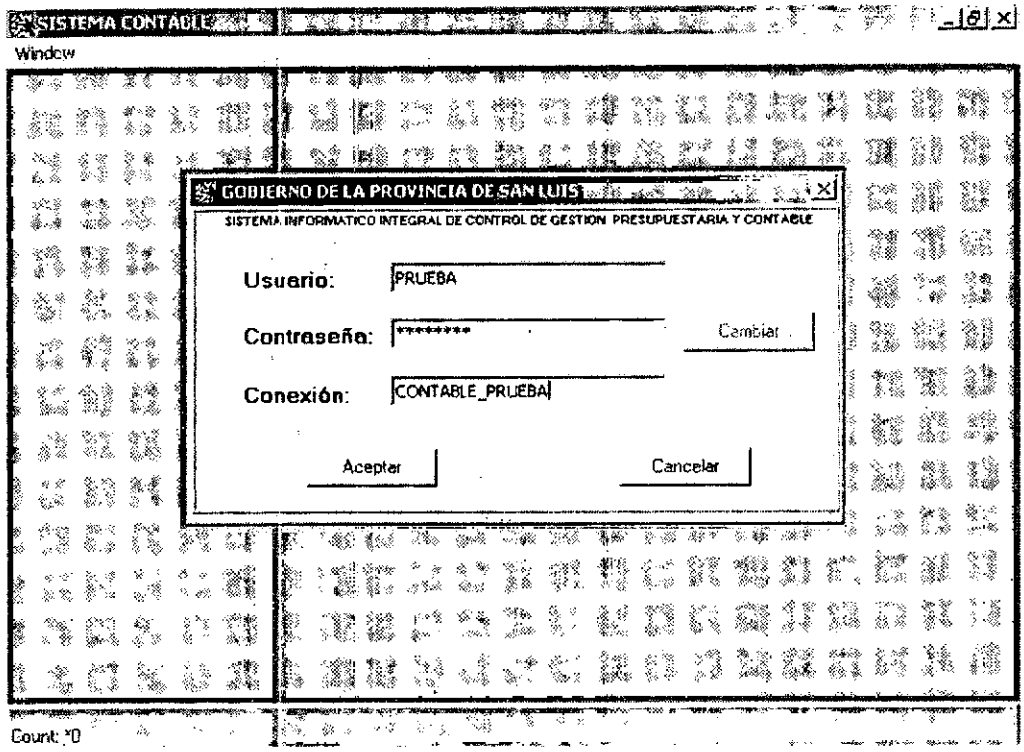
7.3. DESCRIPCIÓN DE MÓDULOS

7.3.1 FORMS

En este punto se presentan los módulos vistos desde el ambiente de desarrollo y también mostramos como serán los form's (pantallas) que interactuan como interfaces con el usuario final en el momento de la ejecución del sistema.

7.3.1.1 Menús de Acceso

En este punto mostramos el acceso al sistema contable y los diferentes menús por los que se accede para llegar a los módulos desarrollados hasta el momento.



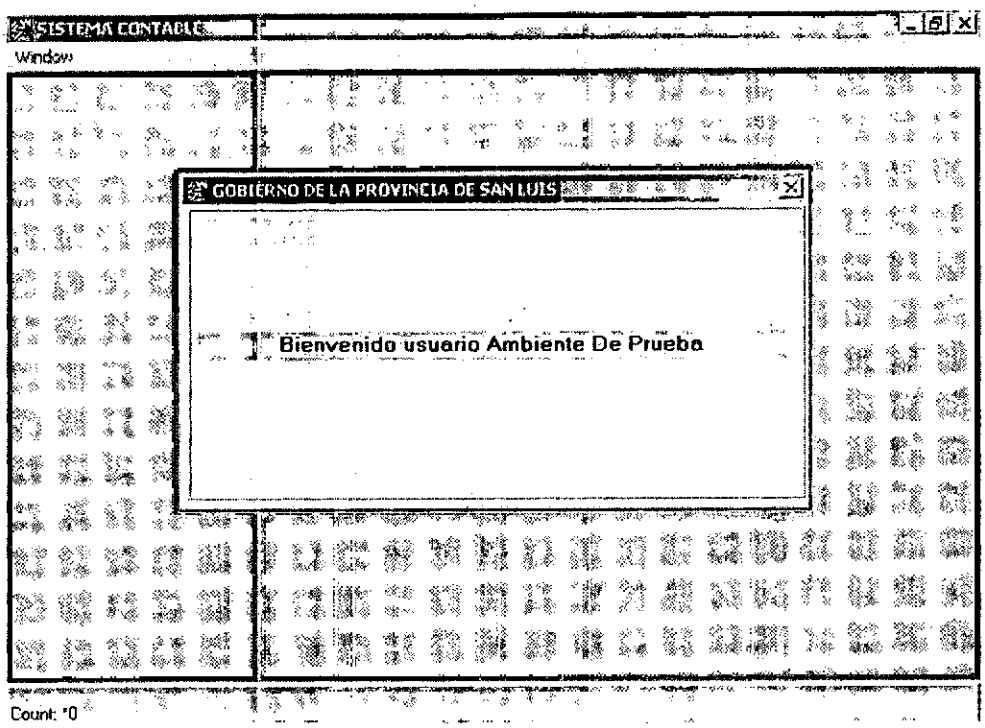
(Figura 7.1)

En la figura 7.1 se muestra el ingreso al sistema contable donde el usuario debe especificar su código de usuario y contraseña y en la parte de conexión se especifica a que base de datos requiere conectarse, para el caso del sistema contable existen las siguientes bases:

CONTABLE_PRUEBA: Esta es la base sobre la cual se desarrolla y se hacen las pruebas en tiempo de desarrollo de módulos.

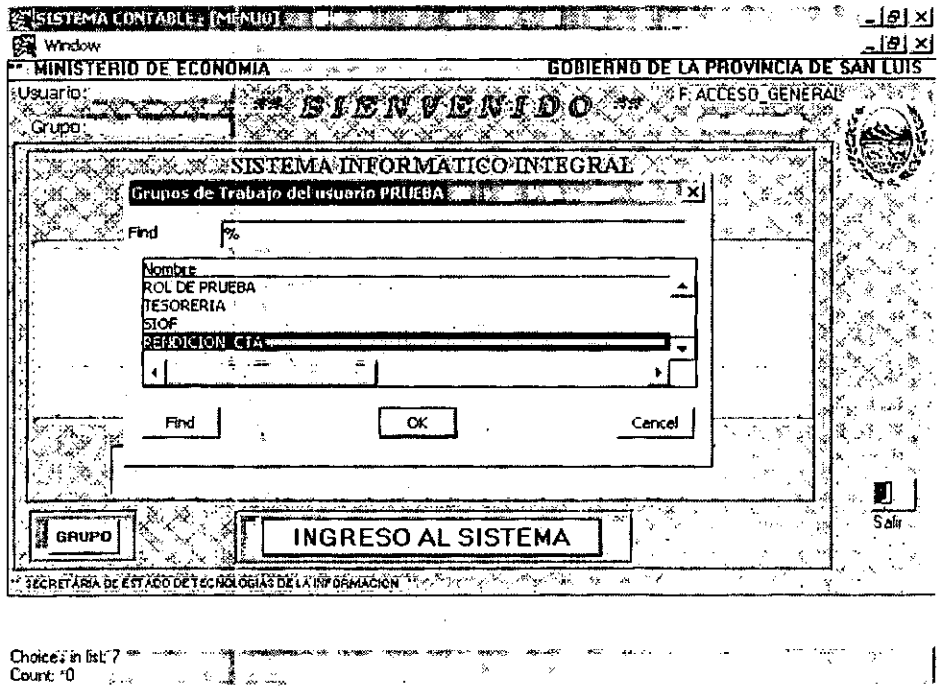
CONTABLE_TEST: Esta es la base sobre la cual se realizan los testing definitivos a los módulos, los datos que existen en esta base se acercan mucho a la realidad.

CONTABLE: Esta es la base de producción.



(Figura 7.2)

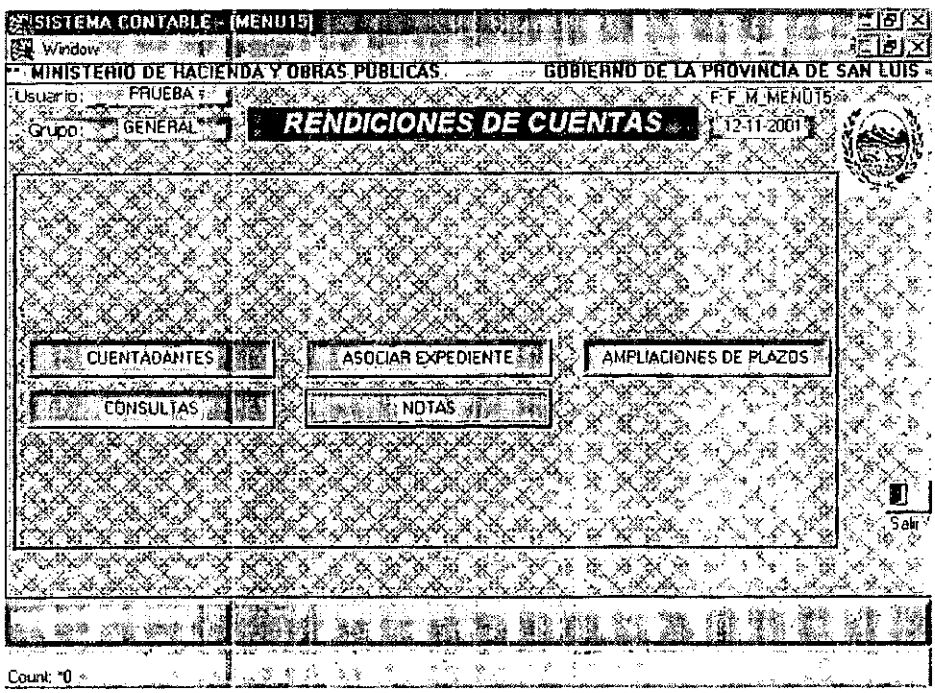
En la figura 7.2 se muestra la pantalla que presenta el sistema contable luego que se autenticó al usuario, en este caso es el usuario PRUEBA ya que estamos conectados a la base de prueba.



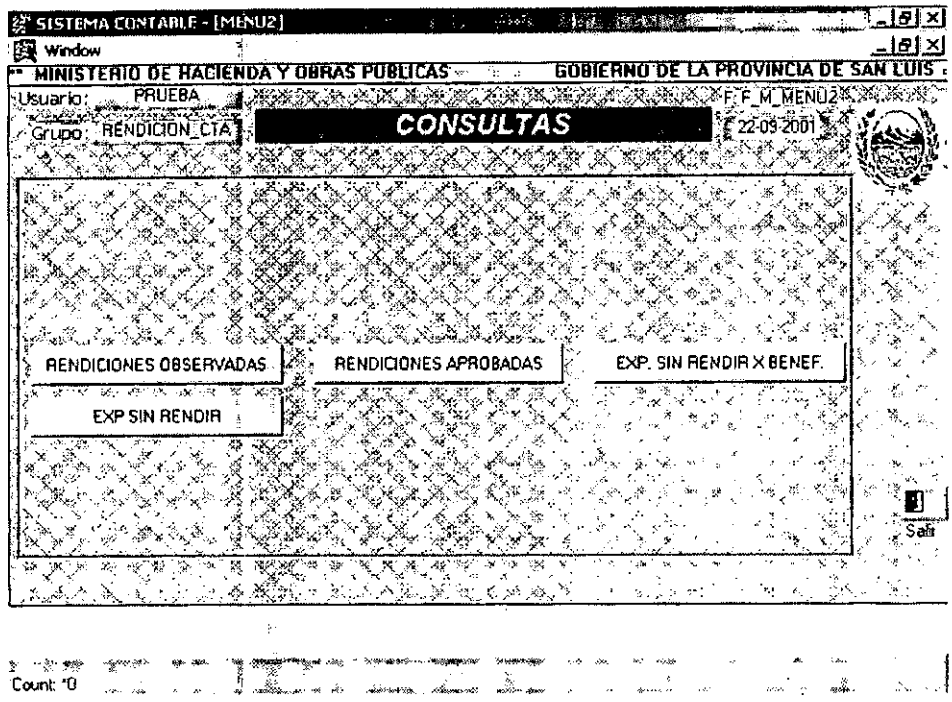
(Figura 7.3)

En la figura 7.3 se muestra una pantalla donde el usuario autorizado para controlar una rendición de cuenta constará con la posibilidad de elegir, entre otros, el grupo de trabajo RENDICION_CTA. Este grupo de Trabajo habilitará al usuario que acaba de ingresar a operar en el sistema de rendiciones de cuentas.

La Figura 7.4 muestra el primer menú que consta de las siguientes opciones: acceder al listado cuentadantes, consultas varias, relacionar expedientes de pago con expedientes de rendición, generar ampliaciones de plazos para los vencimientos de las rendiciones y un acceso a notas comunes. En el caso de las opciones cuentadantes, ampliaciones de plazos y asociar expedientes, el menú accede directamente a estos módulos, mientras que en el caso de consultas y notas se ingresa a un submenú, el cual muestra nuevas opciones que podrá optar el usuario.



(Figura 7.4)

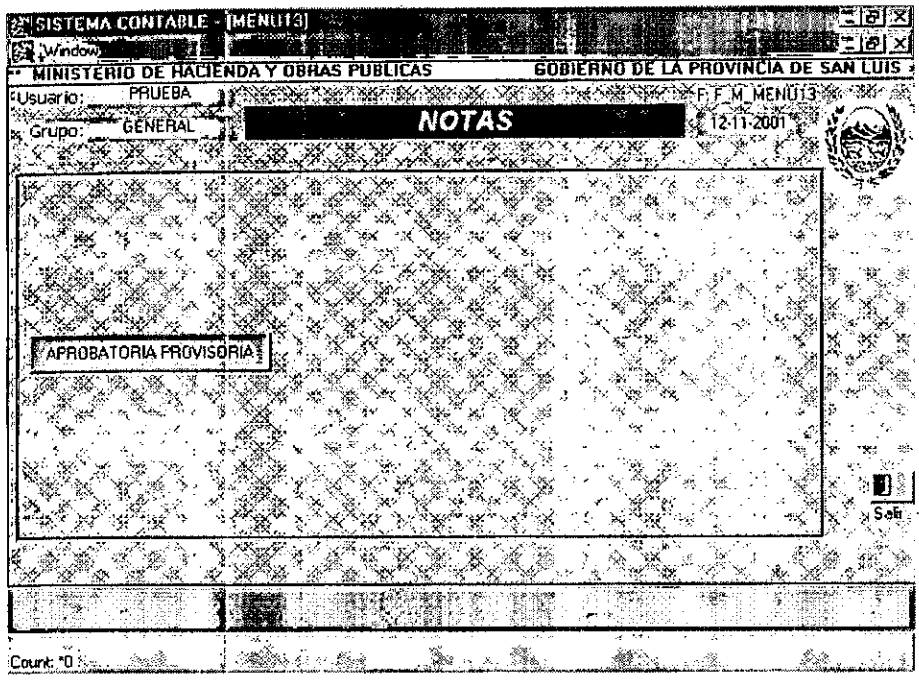


(Figura 7.5)

La figura 7.5 muestra el submenú mencionado anteriormente, donde se puede optar por las siguientes opciones de consultas: Rendiciones de cuentas observadas, Rendiciones de cuentas aprobadas, Expedientes pendientes de rendir por beneficiarios y Expedientes sin rendir por tipo de contrataciones. Al acceder por cualquiera de estas opciones se llega a los módulos finales.

Cabe destacar que la modalidad que tiene el Sistema Contable para generar estos menús es dinámica, de esta manera se puede agregar y quitar opciones a los menús sin perder la performance del sistema.

En la figura 7.6 se muestra el submenú de notas tipos. En este "form" se agregarán los accesos a otras notas que sean definidas en el futuro. El requerimiento y la definición de estas notas tipos vienen dado por el personal técnico de la Dirección Provincial de Contaduría del Gobierno de la Provincia de San Luis.



(Figura 7.6)

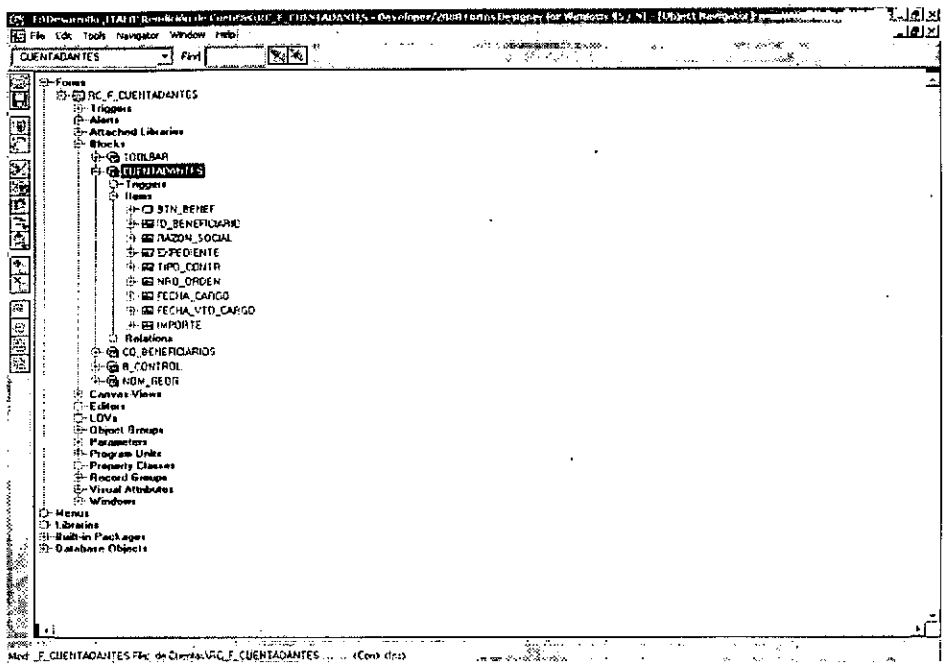
7.3.1.2 Módulo de Cuentadantes

Los cuentadantes son aquellos beneficiarios que tienen rendiciones de cuentas pendientes, entiéndase por rendiciones pendientes a aquellos expedientes que aun no han sido rendido o aquellos que si han sido rendido pero no han sido aprobados.

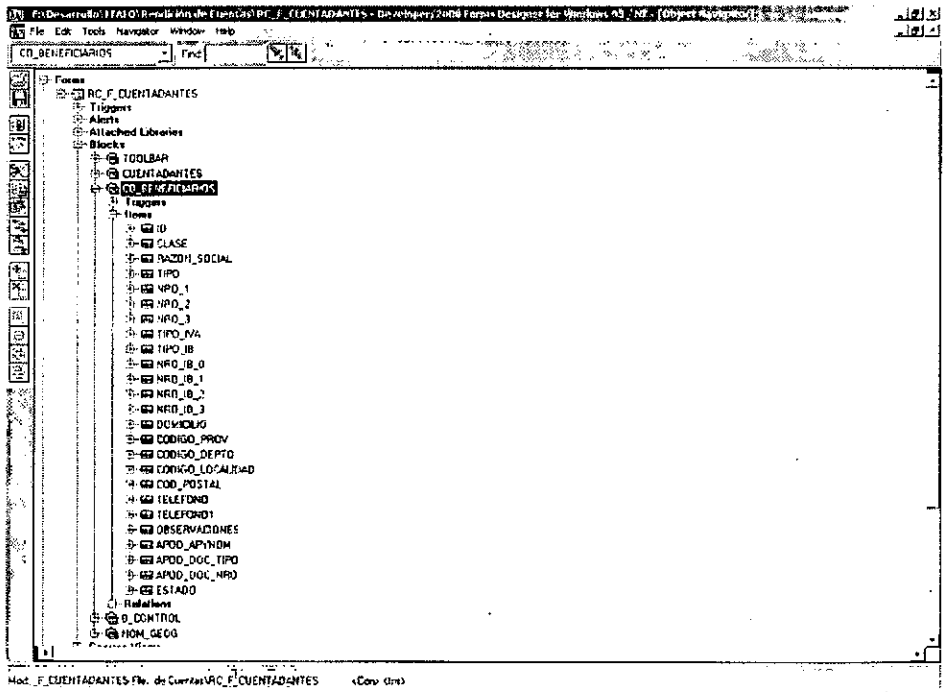
7.3.1.2.1 Desarrollo del Módulo

En la Figura 7.7 se muestra en el ambiente de desarrollo los items relacionados con el bloque de cuentadantes que forma parte de este módulo, estos ítems serán mostrados al usuario del sistema cuando el módulo este en ejecución.

Entre estos ítems de texto que se muestran al usuario hay un ítem que es un botón que es el que tiene la funcionalidad de mostrar en pantalla los datos propios del beneficiario. Estos datos están definidos en el bloque co_beneficiarios como ítems, este bloque esta relacionado directamente con la tabla co_beneficiarios que es la tabla donde se almacenan los datos propios de cada beneficiario.

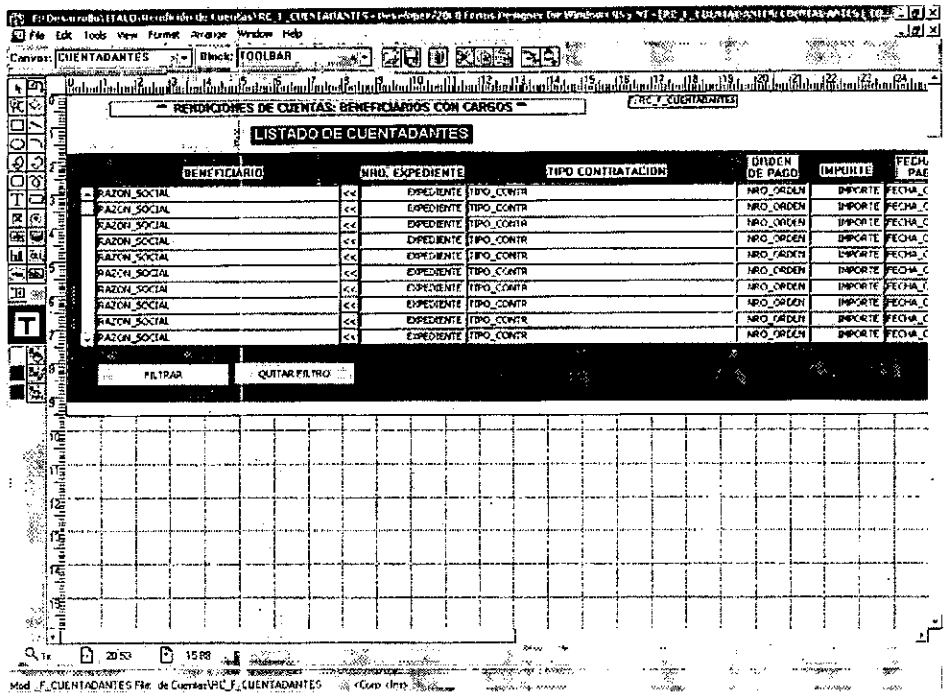


(Figura 7.7)

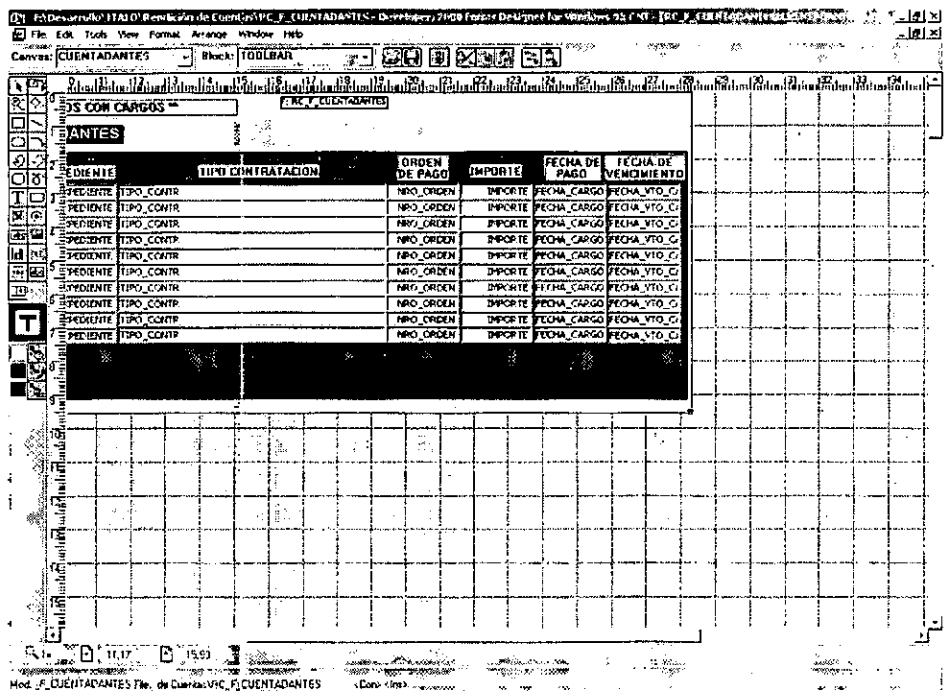


(Figura 7.8)

En la figura 7.8 se muestra la definición del bloque co_beneficiarios mencionado anteriormente, cabe destacar que estos no son todos los datos de la tabla que le corresponde, solo se encuentran los datos relevantes a este módulo.



(Figura 7.9)

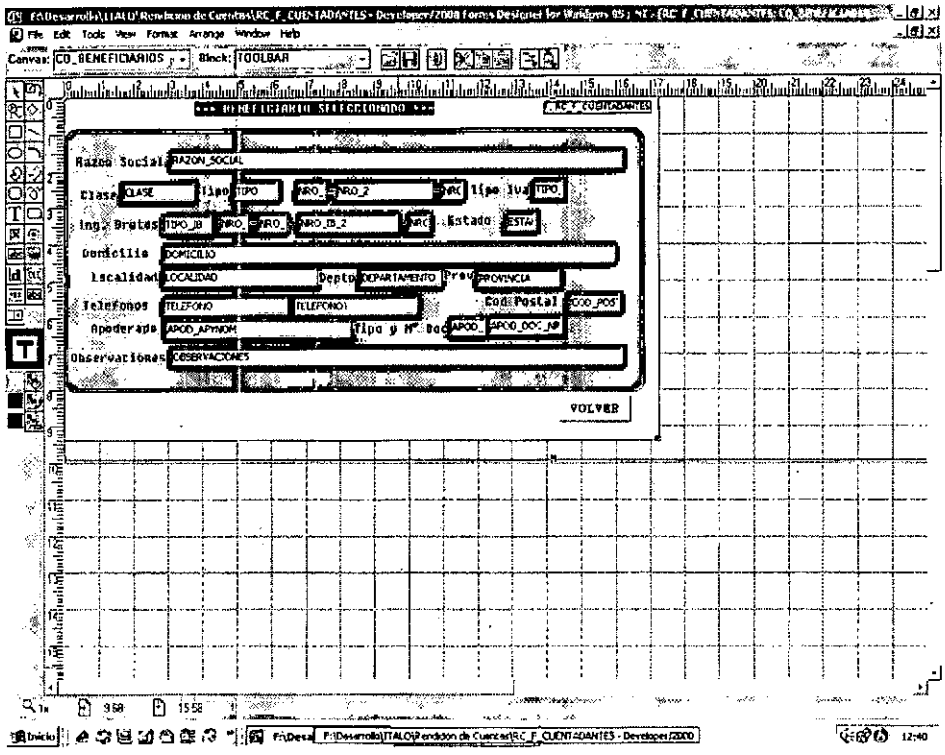


(Figura 7.10)

En las figuras 7.9 y 7.10 se muestra el desarrollo de la parte visible del módulo, donde se muestran los ítems del bloque cuentadantes.

En estas figura 7.9 se puede apreciar los botones FILTRAR y QUITAR FILTRO que dan la posibilidad de filtrar la lista por la razón social del beneficiario.

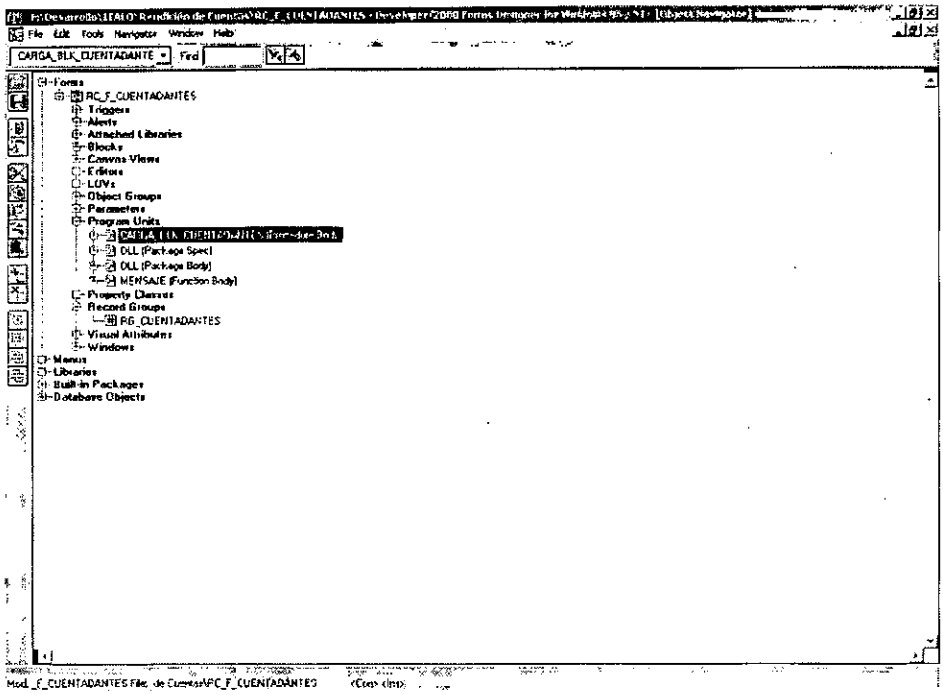
También se puede ver el botón ⊠ que es el ítem que forma parte del bloque cuentadantes.



(Figura 7.11)

En la Figura 7.11 se muestra el desarrollo de cómo se verán los datos propios del beneficiario, a esta pantalla se llega presionando el botón ⊠ que se ve en la figura 7.9.

En el desarrollo de un módulo también hay otros puntos importantes a tener en cuenta, dichos puntos son funcionalidades que hacen que el módulo funcione como debe, estas funciones son: la de carga el bloque de cuentadantes y la función que recupera los datos desde las tablas en la base de datos.



(Figura 7.12)

En la figura 7.12 mostramos en que parte del módulo se almacenan las funciones, como se puede apreciar dentro de los *Program Units* hay un procedimiento llamado CARGA_BLK_CUENTADANTES que es el encargado de cargar los datos en el bloque cuentadantes.

Dentro de los *Record Group* se encuentra un registro llamado RG_CUENTADANTES que es el encargado de recuperar los datos de los cuentadantes desde las tablas.

A continuación se muestra el código fuente que forma parte del Procedimiento CARGA_BLK_CUENTADANTES.

```
PROCEDURE CARGA_BLK_CUENTADANTES IS
        rg_id                RecordGroup;
        razon_social_id   GroupColumn;
        expediente_id       GroupColumn;
        tipo_contr_id       GroupColumn;
        nro_orden_id        GroupColumn;
        importe_id          GroupColumn;
        fecha_cargo_id      GroupColumn;
        fecha_vto_id        GroupColumn;
 id_beneficiario  GroupColumn;

        the_Rowcount     Number;
        sel_row            Number;
        ret_populate     Number;
        cod_error        Number;
BEGIN
        rg_id := Find_Group('RG_CUENTADANTES');
        ret_populate := POPULATE_GROUP(rg_id);
```

```
   IF ret_populate = 0 THEN
                     razon_social_id := Find_Column('RG_CUENTADANTES.RAZON_SOCIAL');
                     expediente_id   := Find_Column('RG_CUENTADANTES.EXPEDIENTE');
                     tipo_contr_id   := Find_Column('RG_CUENTADANTES.DESCRIPCION');
                     nro_orden_id    := Find_Column('RG_CUENTADANTES.NRO_ORDEN_PAGO');
                     importe_id      := Find_Column('RG_CUENTADANTES.IMPORTE');
                     fecha_cargo_id  := Find_Column('RG_CUENTADANTES.FECHA_CARGO');
                     fecha_vto_id    := Find_Column('RG_CUENTADANTES.FECHA_VTO_CARGO');
                     id_beneficiario := Find_Column('RG_CUENTADANTES.ID');


                     the_Rowcount := Get_Group_Row_Count(rg_id);
                     go_block('CUENTADANTES');
                     clear_block;

           FOR j IN 1..the_Rowcount LOOP
             Set_Group_Selection(rg_id, j);
             sel_row := Get_Group_Selection(rg_id, j);
      CREATE_RECORD;
                     :CUENTADANTES.RAZON_SOCIAL := Get_Group_Char_Cell(razon_social_id,sel_row);
                     :CUENTADANTES.EXPEDIENTE   := Get_Group_Char_Cell(expediente_id, sel_row);
                     :CUENTADANTES.TIPO_CONTR   := Get_Group_Char_Cell(tipo_contr_id, sel_row);
                     :CUENTADANTES.NRO_ORDEN    := Get_Group_Char_Cell(nro_orden_id, sel_row);
                     :CUENTADANTES.IMPORTE       := Get_Group_Number_Cell(importe_id, sel_row);
                     :CUENTADANTES.FECHA_CARGO   := Get_Group_Date_Cell(fecha_cargo_id, sel_row);
                     :CUENTADANTES.FECHA_VTO_CARGO:= Get_Group_Date_Cell(fecha_vto_id, sel_row);
                     :CUENTADANTES.ID_BENEFICIARIO:=Get_group_Number_cell(id_beneficiario,sel_row);

     END LOOP;
--    DELETE_RECORD;
         FIRST_RECORD;
       ELSIF ret_populate = 1403 THEN
          cod_error := MENSAJE('ACEPTAR','ERROR EN PARAMETROS','PARAMETROS: No se han
                       encontrado registros para los parametros brindados...');
         go_item('b_control.filtrar_benef');
  ELSE
    cod_error := MENSAJE('ACEPTAR','ERROR EN RECORD GROUP','ERROR:'||
                       TO_CHAR(ret_populate)||': No se han encontrado datos para
                       el RecordGroup CUENTADANTES...');
       END IF;
END;
```

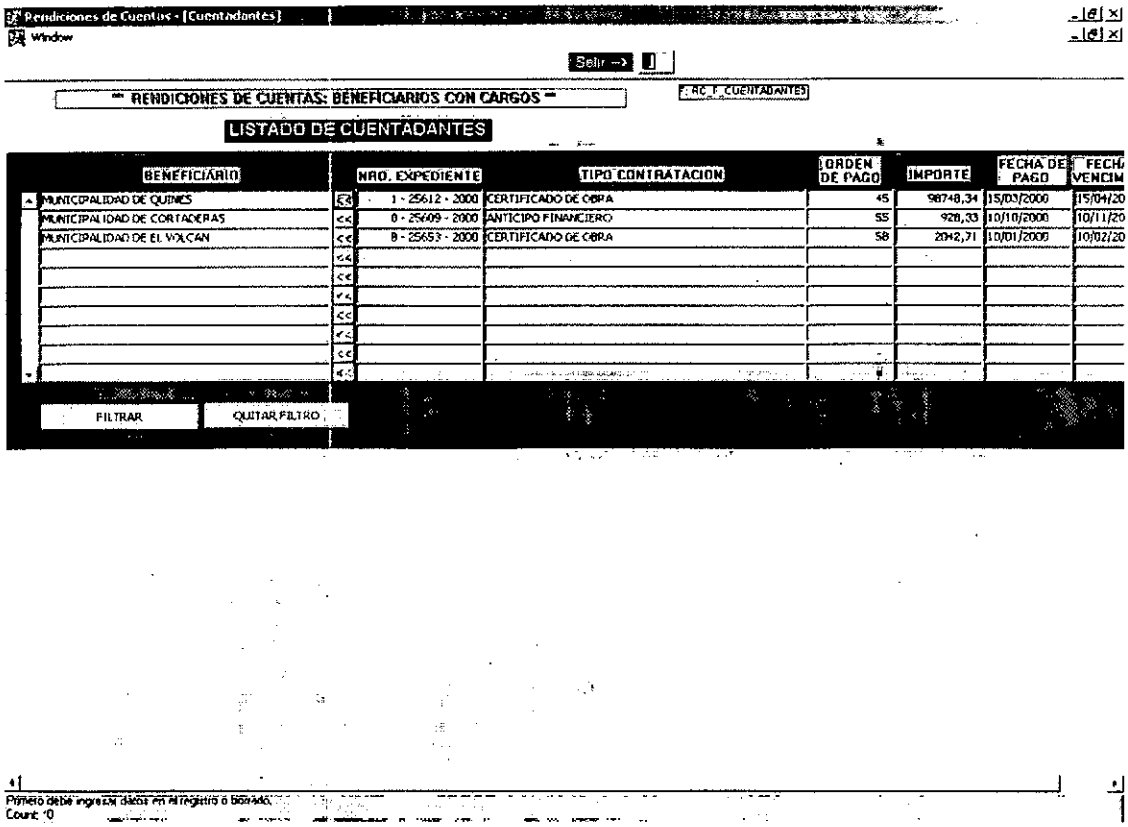
Este procedimiento recupera datos desde el record group y los almacena en los registros del bloque. A continuación se muestra el SQL que forma parte del record group RG_CUENTADANTES.

```
SELECT
  b.razon_social,
  b.id,
  e.inic ||' - '|| e.numero ||' - '|| e.ejercicio expediente,
  tc.descripcion,
  ror.numero nro_orden_pago,
  rod.importe,
  ror.fecha_cargo,
  ror.fecha_vto_cargo
FROM
  co_registro_orden_detalles rod,
  co_beneficiarios b,
  co_expedientes e,
  ngr_tipos_contr tc,
  (SELECT o.numero,
          o.fecha_cargo,
          o.fecha_vto_cargo,
          o.nro_secuencia_exp,
          o.id_beneficiario,
          o.nro_secuencia
         FROM co_registro_ordenes o
   WHERE o.cargo IS NOT NULL
           AND o.tipo = 'OP'
           AND (NOT EXISTS (SELECT 1
                              FROM co_rc_rendiciones rr
                             WHERE rr.co_r_or_nro_secuencia = o.nro_secuencia))
  UNION
   SELECT o.numero,
          o.fecha_cargo,
                o.fecha_vto_cargo,
                o.nro_secuencia_exp,
                o.id_beneficiario,
                o.nro_secuencia
         FROM co_registro_ordenes o,co_rc_rendiciones r
        WHERE o.cargo IS NOT NULL
```

```
        AND o.tipo = 'OP'
        AND (r.aprobada = 'NO' AND  r.co_r_or_nro_secuencia = o.nro_secuencia )
  ) ror
WHERE b.id = ror.id_beneficiario
  AND tc.id_tipo_contr = e.id_tipo_contr
  AND rod.nro_secuencia_op = ror.nro_secuencia
  AND ror.nro_secuencia_exp = e.nro_secuencia
  AND rod.id_retencion = 99
  AND b.razon_social LIKE :p_benef
```

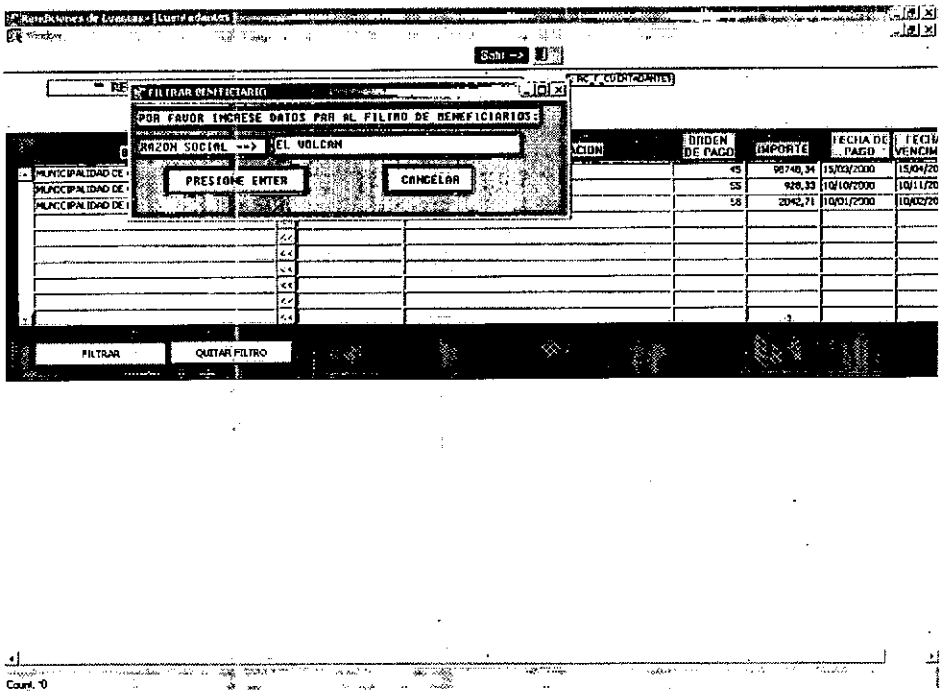
7.3.1.2.2 Ejecución del Módulo

En este punto se muestra al módulo en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.



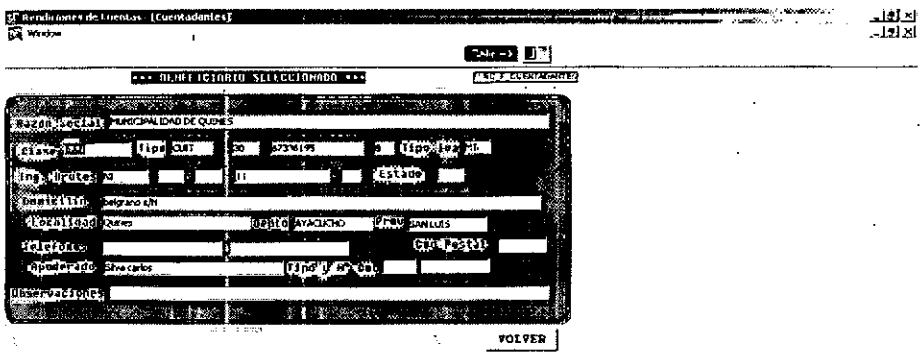
(Figura 7.13)

En la figura 7.13 se muestra al módulo de cuentadantes ya en ejecución, esta pantalla es la que se muestra en las figuras 7.9 y 7.10 en la etapa de desarrollo.



(Figura 7.14)

Presionando el botón FILTRAR se aparece una ventana donde se le da al usuario la posibilidad de ingresar un criterio para poder filtrar la lista de cuentadantes. Luego si el usuario desea quitar el filtro lo puede hacer presionando el otro botón QUITAR FILTRO.



(Figura 7.15)

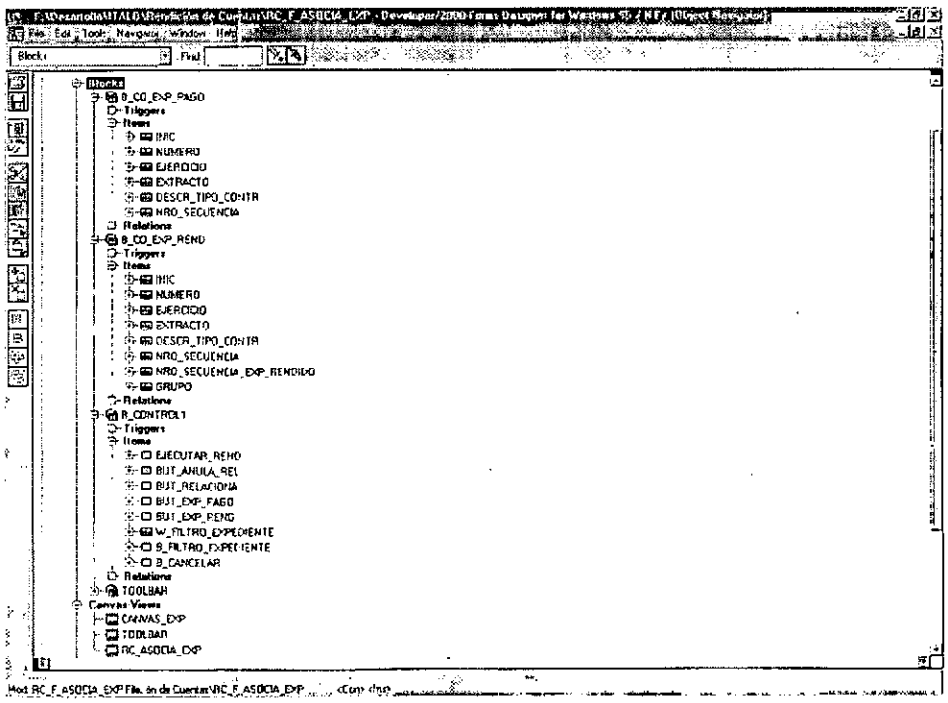
La figura 7.15 observa la pantalla (en ejecución) que muestra los datos del beneficiario seleccionado. Esta figura es la que se corresponde a la figura 7.11 en el desarrollo del módulo.

7.3.1.3 Módulo de Relación de Expedientes

Debido a que la rendición de cuenta de un pago se realiza en un expediente distinto al expediente con que se pago es necesario relacionarlos para que exista la correspondencia entre los dos expedientes y así se hace más ágil la localización de los dos expedientes, teniendo el número de cualquiera de los dos se puede conocer el número del otro.

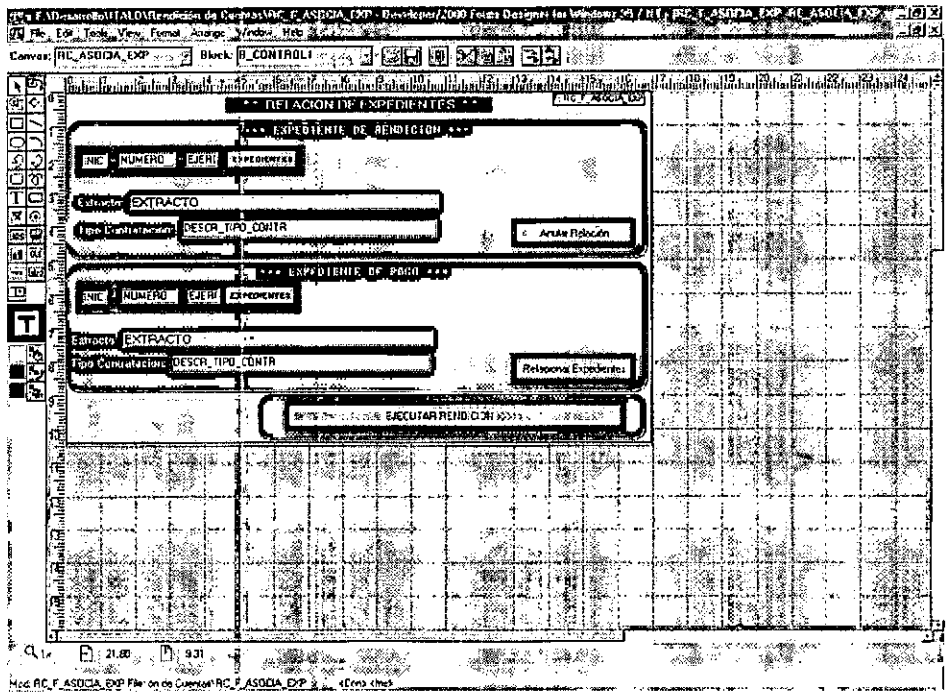
7.3.1.3.1 Desarrollo del Módulo

En la Figura 7.16 se muestra en el ambiente de desarrollo los ítems relacionados a los principales bloques de este módulo.



(Figura 7.16)

Los bloques que se aprecian en la figura 7.16 son B_CO_EXP_PAGO y B_CO_EXP_REND que son los que se corresponden al expediente de pago y expediente de rendición respectivamente. En la misma figura también se observa el nombre del *Canvas-view* RC_ASOCIA_EXP que es la parte visible del módulo.



(Figura 7.17)

En la figura 7.17 se observa exactamente la canvas-view mencionada anteriormente donde se muestra los datos mínimos necesarios de cada expediente, dichos datos son: iniciador, número, ejercicio, extracto y tipo de contratación. En la misma figura se observa dos botones "Anula Relación", "Relacionar Expedientes" y "EJECUTAR RENDICION", a continuación se muestra el código de estos botones.

"Relacionar Expedientes"

```
DECLARE
        alerta     NUMBER;
BEGIN
   alerta := mensaje('SINO','Confirmación', 'Relaciona los Expedientes?');
   IF alerta = ALERT_BUTTON1 THEN
      UPDATE CO_EXPEDIENTES E
         SET E.NRO_SECUENCIA_EXP_RENDIDO = :B_CO_EXP_PAGO.NRO_SECUENCIA
       WHERE E.NRO_SECUENCIA = :B_CO_EXP_REND.NRO_SECUENCIA;
```

```
        COMMIT;
        GO_BLOCK('B_CO_EXP_PAGO');
   END IF;
END;
```

"Anula Relación"

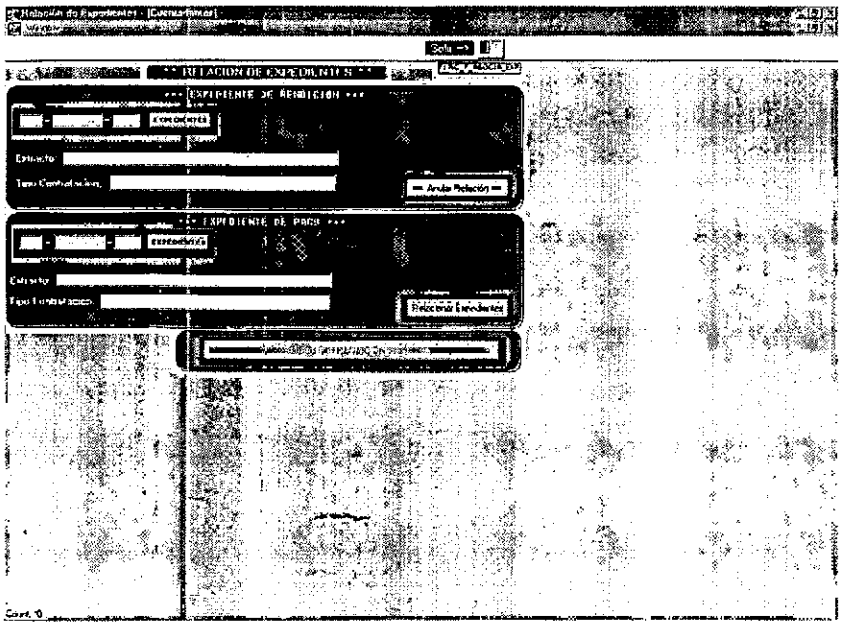
```
DECLARE
        alerta    NUMBER;
BEGIN
   alerta := mensaje('SINO','Confirmación','Esta seguro de quitar la relación de los expedientes');
   IF alerta = ALERT_BUTTON1 THEN
      UPDATE CO_EXPEDIENTES E
         SET E.NRO_SECUENCIA_EXP_RENDIDO = NULL
       WHERE E.NRO_SECUENCIA = :B_CO_EXP_REND.NRO_SECUENCIA;
        COMMIT;
        GO_BLOCK('B_CO_EXP_PAGO');
        CLEAR_BLOCK;
   END IF;
END;
```

"EJECUTAR RENDICION"

```
declare
list_id PARAMLIST;
programa_a_ejecutar  varchar2;
begin
  list_id := GET_PARAMETER_LIST('PARAM_ENTRADA');
  IF NOT ID_NULL(list_id) THEN
    Destroy_Parameter_List(list_id);
  END IF;
  list_id := Create_Parameter_List('PARAM_ENTRADA');
  Add_Parameter(list_id,'P_PATH',TEXT_PARAMETER,:parameter.P_PATH);
  Add_Parameter(list_id,'P_EJERCICIO',TEXT_PARAMETER,TO_CHAR(:parameter.P_EJERCICIO));
  programa_a_ejecutar := 'GRAL\GRAL_F_PEE';
  Call_Form(:parameter.P_PATH || programa_a_ejecutar, NO_HIDE, NO_REPLACE, NO_QUERY_ONLY, list_id);
end;
```

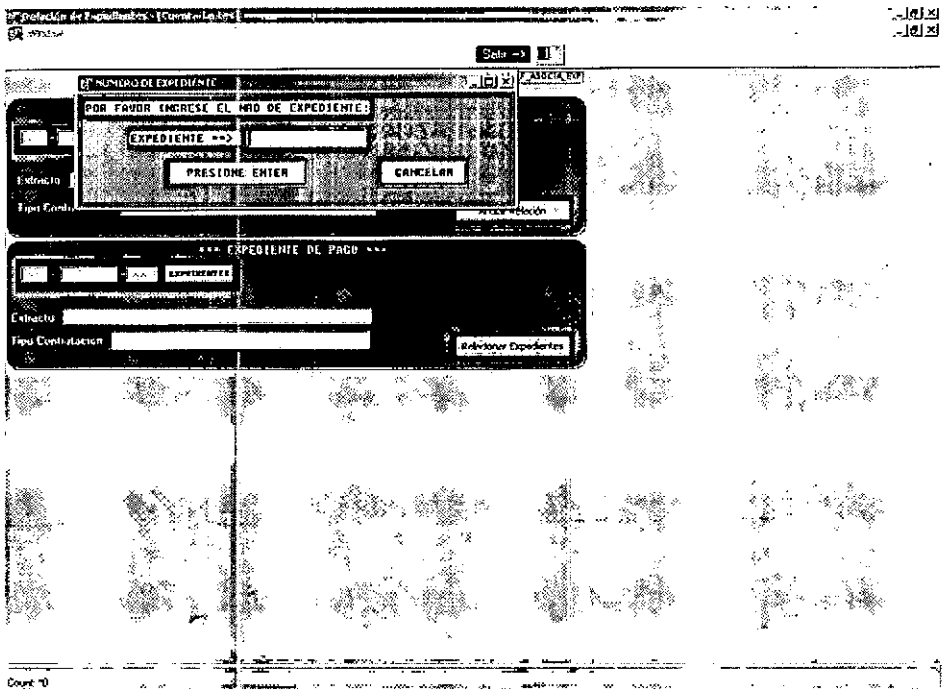
7.3.1.3.2 Ejecución del Módulo

En este punto se muestra al módulo en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.



(Figura 7.18)

En la figura 7.18 se muestra en ejecución el módulo que asocia los expedientes de paco con los expedientes de rendición, esta figura esta relacionada con la figura 7.17 en el punto de desarrollo del módulo, como se puede observar el botón de ejecución de rendiciones se encuentra deshabilitado, el mismo se habilita una vez asociados ambos expedientes.



(figura 7.19)

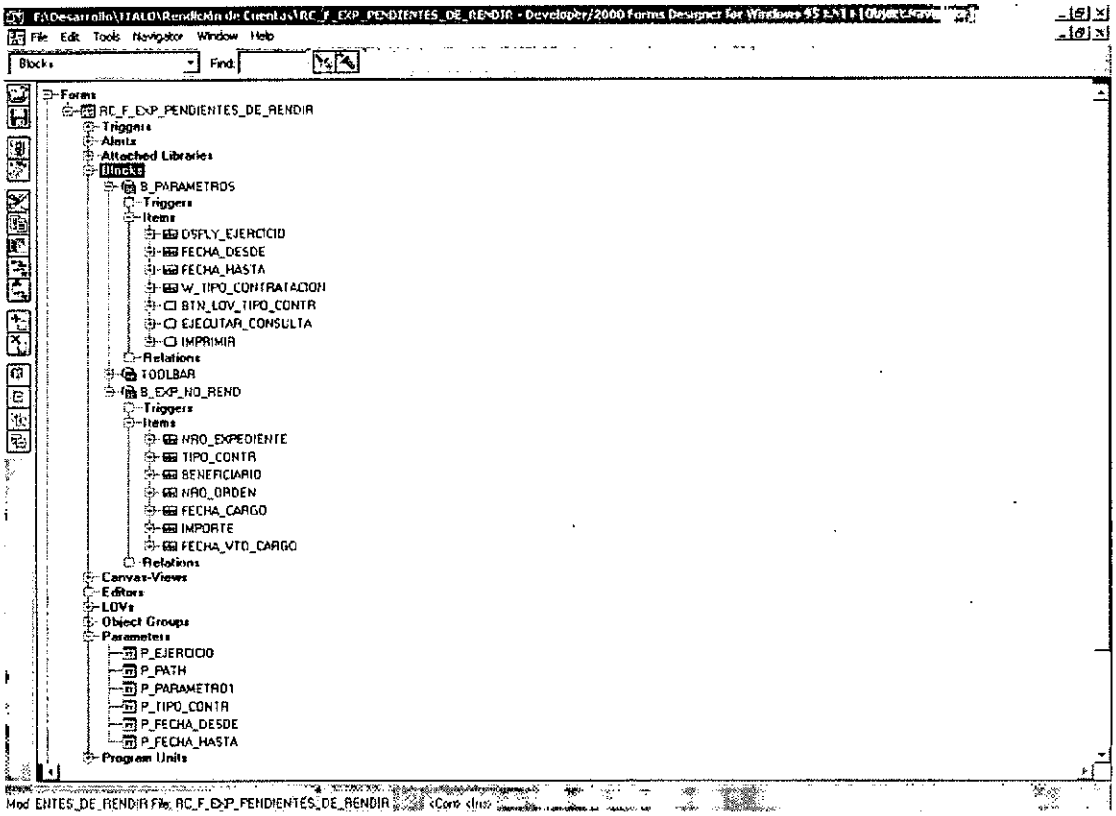
En la figura 7.19 se muestra la ventana que aparece al presionar cualquiera de botones "EXPEDIENTES" y que es donde se debe colocar el número de expediente que desea relacionar.

7.3.1.4 Módulo de Expedientes Pendientes de Rendición

Los expedientes pendientes de rendición son aquellos que están sujetos a rendición de cuenta y que se han pagado pero aún no ha sido aprobada la rendición correspondiente a este expedienteo que hasta el momento no se haya presentado.

7.3.1.4.1 Desarrollo del Módulo

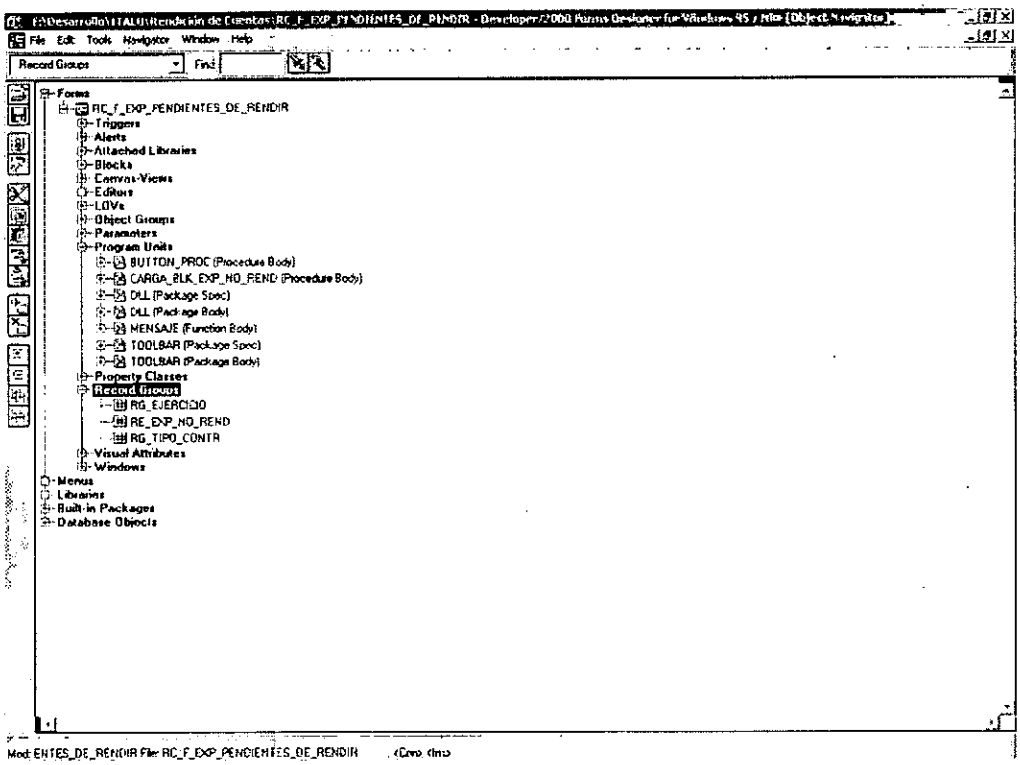
En la Figura 7.20 se muestra en el ambiente de desarrollo los ítems relacionados a los principales bloques de este módulo.



(Figura 7.20)

Los bloques observados en la figura 7.20 son:

B_PARAMETROS que representa los parámetros que se le pasarán al SQL para que recupere los datos requeridos, los valores de estos ítems son cargados en los parámetros definidos en *Parameters*.



(Figura 7.21)

B_EXP_NO_REND que representa los valores que se mostrarán en el módulo cuando se esté ejecutando. Este bloque a es cargado a través de una rutina que encuentra en el punto *Program Units* que toma los datos del *Record Group*.

En la figura 7.21 se muestra los *Program Units* y los *Record Group* que se mencionaron anteriormente, los de mayor importancia para este punto son:

CARGA_BLK_EXP_NO_REND que es un program units que toma los datos de un record group y los almacena en el bloque B_EXP_NO_REND mostrado en la figura 7.20.

A continuación se muestra el código fuente de estas dos funcionalidades que hacen funcionar al módulo.

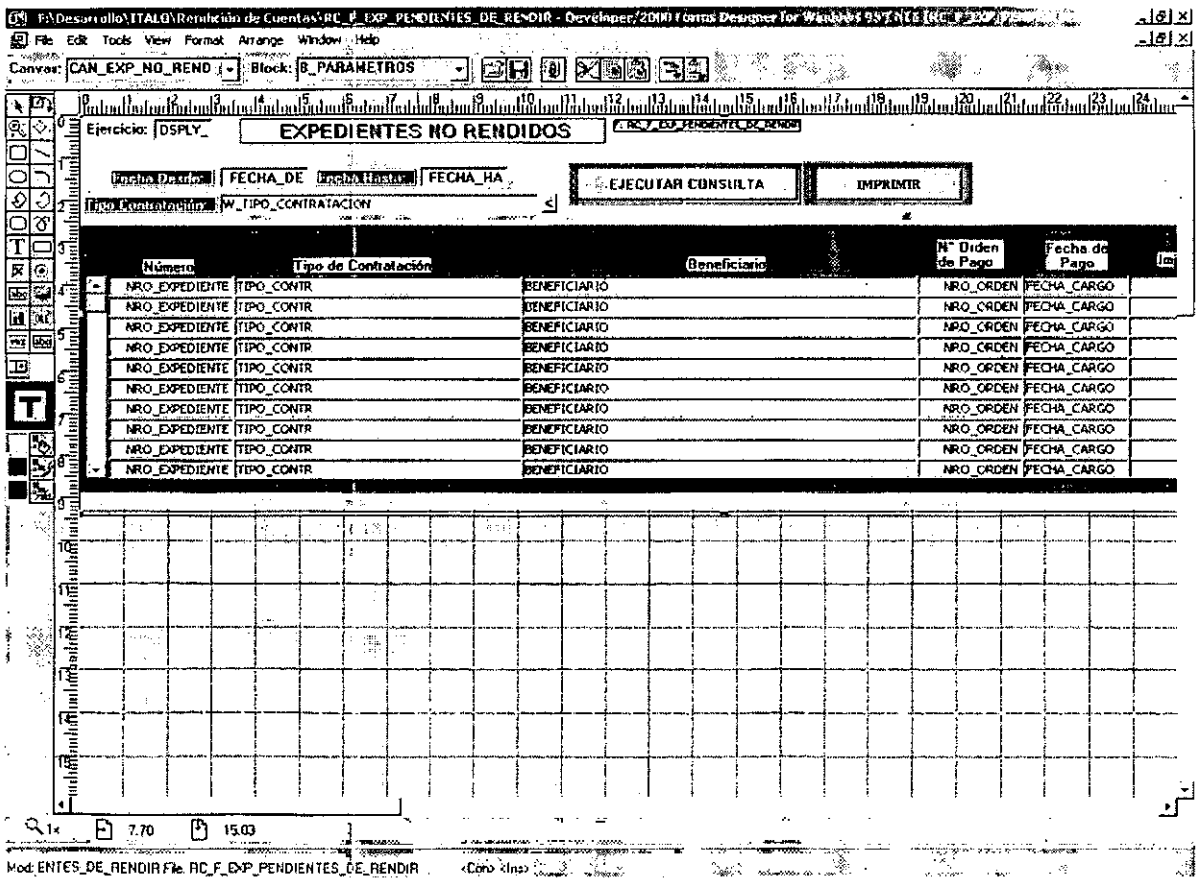
```
PROCEDURE CARGA_BLK_EXP_NO_REND IS
-- -------------------------------------------------------------------------
-- Declaración de variables
-- -------------------------------------------------------------------------
        rg_id         RecordGroup; -- identificación del record group
        expediente    GroupColumn; -- identificación de la columna del expediente
        tipo_contr    GroupColumn;  -- identificación de la columna del tipo de contratación del
expediente
```

```
        razon_social    GroupColumn;  -- identificación de la columna de la razón social del
beneficiario
        nro_orden   GroupColumn; -- identificación de la columna del número de orden de pago
        fecha_cargo GroupColumn; -- identificación de la columna de la fecha del pago
        importe     GroupColumn; -- identificación de la columna del importe
        fecha_vto    GroupColumn;  -- identificación de la columna de la fecha de vencimiento del
cargo

        the_Rowcount    Number;
        sel_row         Number;
        ret_populate    Number;
        cod_error       Number;
BEGIN
  rg_id := Find_Group('RE_EXP_NO_REND');
        ret_populate := POPULATE_GROUP(rg_id);
-- ------------------------------------------------------------------------------
-- Si el record group encontro valores, estos se cargan en el bloque de datos
-- ------------------------------------------------------------------------------
 IF ret_populate = 0 THEN
           expediente      := Find_Column('RE_EXP_NO_REND.EXPEDIENTE');
           tipo_contr      := Find_Column('RE_EXP_NO_REND.DESCRIPCION');
           razon_social    := Find_Column('RE_EXP_NO_REND.RAZON_SOCIAL');
           nro_orden       := Find_Column('RE_EXP_NO_REND.NRO_ORDEN_PAGO');
           fecha_cargo     := Find_Column('RE_EXP_NO_REND.FECHA_CARGO');
           importe         := Find_Column('RE_EXP_NO_REND.IMPORTE');
           fecha_vto       := Find_Column('RE_EXP_NO_REND.FECHA_VTO_CARGO');


           the_Rowcount := Get_Group_Row_Count(rg_id);

           go_block('B_EXP_NO_REND');
           clear_block;

           FOR j IN 1..the_Rowcount LOOP
              Set_Group_Selection(rg_id, j);
              sel_row := Get_Group_Selection(rg_id, j);
              CREATE_RECORD;

              :B_EXP_NO_REND.NRO_EXPEDIENTE  := Get_Group_Char_Cell(expediente, sel_row);
              :B_EXP_NO_REND.TIPO_CONTR      := Get_Group_Char_Cell(tipo_contr, sel_row);
              :B_EXP_NO_REND.BENEFICIARIO    := Get_Group_Char_Cell(razon_social, sel_row);
              :B_EXP_NO_REND.NRO_ORDEN       := Get_Group_Char_Cell(nro_orden, sel_row);
              :B_EXP_NO_REND.FECHA_CARGO     := Get_Group_Date_Cell(fecha_cargo, sel_row);
              :B_EXP_NO_REND.IMPORTE         := Get_Group_Number_Cell(importe, sel_row);
              :B_EXP_NO_REND.FECHA_VTO_CARGO := Get_Group_Date_Cell(fecha_vto, sel_row);

    END LOOP;
          FIRST_RECORD;
       ELSIF ret_populate = 1403 THEN
          cod_error := MENSAJE('ACEPTAR','ERROR EN PARAMETROS','PARAMETROS: No se han
                       encontrado registros para los parametros brindados...');
          :B_PARAMETROS.FECHA_DESDE          := NULL;
          :B_PARAMETROS.FECHA_HASTA          := NULL;
          :B_PARAMETROS.W_TIPO_CONTRATACION := NULL;
          go_item('b_parametros.fecha_desde');
  ELSE
     cod_error := MENSAJE('ACEPTAR','ERROR EN RECORD GROUP', 'ERROR:' ||
                          TO_CHAR(ret_populate)||': No se han encontrado datos para
                          el RecordGroup RE_EXP_NO_REND...');
       END IF;
END;
```

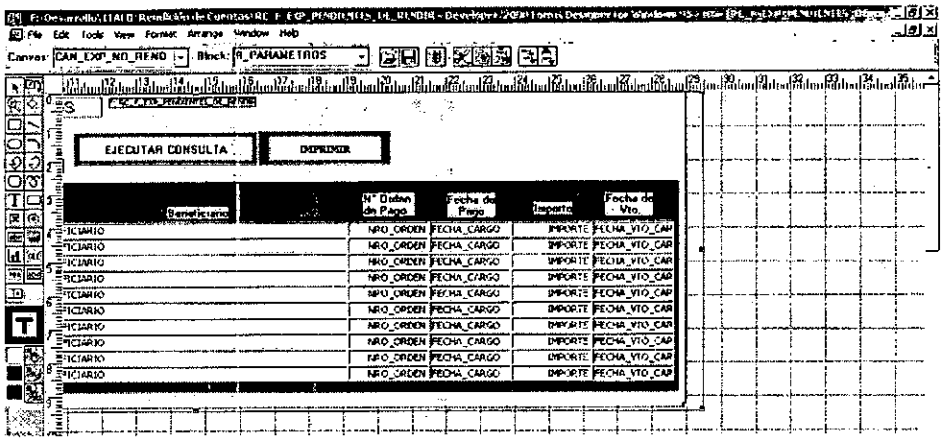
Este procedimiento recupera datos desde el record group y los almacena en los registros del bloque. A continuación se muestra el SQL que forma parte del record group RG_CUENTADANTES.

```
SELECT
  e.inic ||' - '|| e.numero ||' - '|| e.ejercicio expediente,
  tc.descripcion,
  b.razon_social,
  ror.numero nro_orden_pago,
  ror.fecha_cargo,
  rod.importe,
  ror.fecha_vto_cargo
FROM
  co_registro_orden_detalles rod,
  co_beneficiarios b,
  co_expedientes e,
  ngr_tipos_contr tc,
  (SELECT o.numero,
```

```
          o.fecha_cargo,
          o.fecha_vto_cargo,
          o.nro_secuencia_exp,
          o.id_beneficiario,
          o.nro_secuencia
         FROM co_registro_ordenes o
    WHERE o.cargo IS NOT NULL
          AND o.fecha_cargo >= :FECHA_DESDE
          AND o.fecha_cargo <= :FECHA_HASTA
          AND o.tipo = 'OP'
          AND o.EJERCICIO = :EJERCICIO
          AND (NOT EXISTS (SELECT 1
                             FROM co_rc_rendiciones rr
                            WHERE rr.co_r_or_nro_secuencia = o.nro_secuencia))
  UNION
   SELECT o.numero,
          o.fecha_cargo,
                o.fecha_vto_cargo,
                o.nro_secuencia_exp,
                o.id_beneficiario,
                o.nro_secuencia
         FROM co_registro_ordenes o,co_rc_rendiciones r
        WHERE o.cargo IS NOT NULL
      AND o.fecha_cargo >= :FECHA_DESDE
          AND o.fecha_cargo <= :FECHA_HASTA
          AND o.tipo = 'OP'
          AND o.EJERCICIO = :EJERCICIO
          AND (r.aprobada = 'NO' AND  r.co_r_or_nro_secuencia = o.nro_secuencia )
  ) ror
WHERE b.id = ror.id_beneficiario
  AND tc.id_tipo_contr = e.id_tipo_contr
  AND rod.nro_secuencia_op = ror.nro_secuencia
  AND ror.nro_secuencia_exp = e.nro_secuencia
  AND rod.id_retencion = 99
  AND e.ejercicio = :EJERCICIO
  AND tc.descripcion LIKE :TIPO_CONTR
```



(Figura 7.22)



(Figura 7.23)

En las figuras 7.22 y 7.23 se muestra el diseño visible del módulo, en estas figuras se puede apreciar los ítems de los bloques mencionados en el anteriormente.

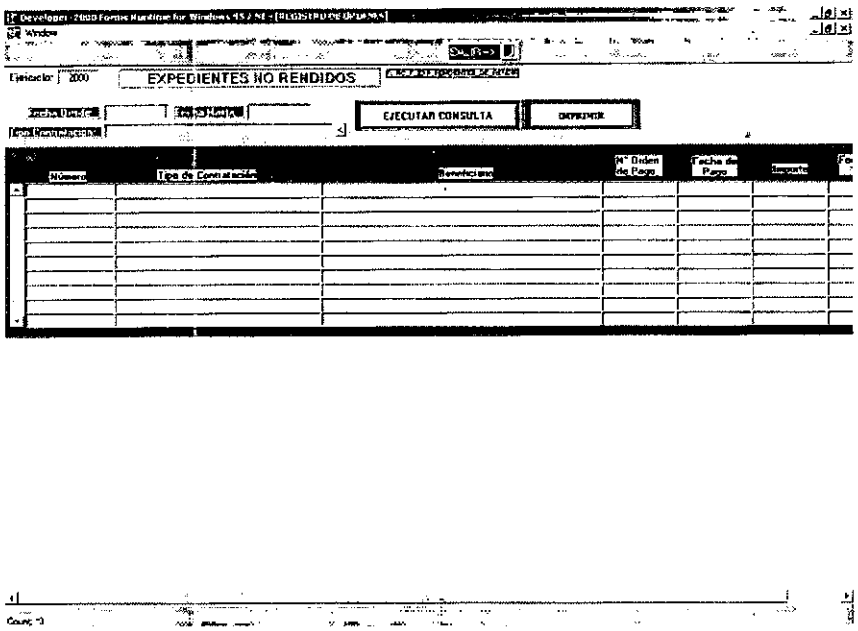
El botón que "EJECUTAR CONSULTA" que se observa en la figura 7.22 tiene la funcionalidad de cargar los parámetros del módulo con los valores que posean los ítems del bloque B_PARAMETROS. A continuación se muestra el código fuente de dicho botón.

```
DECLARE
  v_mensaje  number;
BEGIN
                    IF :B_PARAMETROS.FECHA_DESDE IS NULL THEN
                       v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha desde...');
     ELSIF :B_PARAMETROS.FECHA_HASTA IS NULL THEN
                       v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha hasta...');
     ELSIF :B_PARAMETROS.FECHA_DESDE > :B_PARAMETROS.FECHA_HASTA THEN
                       v_mensaje := MENSAJE('ACEPTAR','ERROR','La "fecha desde" debe ser menor o
                             igual a la "fecha hasta"...');
     ELSE
     IF :B_PARAMETROS.W_TIPO_CONTRATACION IS NULL THEN
                       :B_PARAMETROS.W_TIPO_CONTRATACION := '%%'; END IF;
                       :PARAMETER.P_FECHA_DESDE := TO_CHAR(:B_PARAMETROS.FECHA_DESDE,'DD/MM/YYYY');
                       :PARAMETER.P_FECHA_HASTA := TO_CHAR(:B_PARAMETROS.FECHA_HASTA,'DD/MM/YYYY');
                       :PARAMETER.P_TIPO_CONTR  := :B_PARAMETROS.W_TIPO_CONTRATACION;
                       :PARAMETER.P_EJERCICIO   := :B_PARAMETROS.DSPLY_EJERCICIO;
        CARGA_BLK_EXP_NO_REND;
     END IF;
END;
```

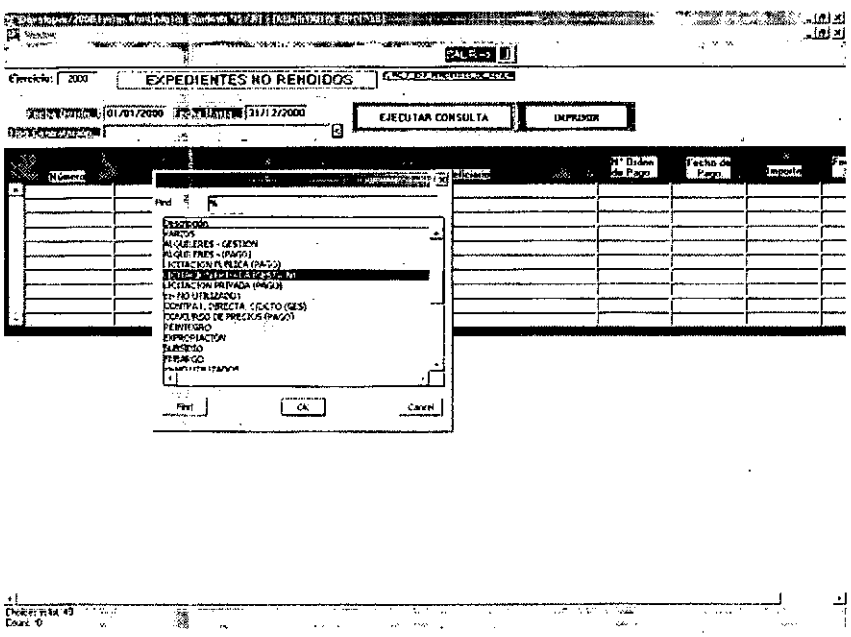
7.3.1.4.2 Ejecución del Módulo

En este punto se muestra al módulo "Expedientes Pendientes de Rendición" en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.

En la figura 7.24 se muestra el módulo de expedientes pendientes de rendición en este caso la consulta que recupera los datos aún no ha sido ejecutada por eso es que se ven todos los campos. En esta figura se puede observar los campos correspondientes a los ítems mostrados en la figura 3.1.20.



(Figura 7.24)



(Figura 7.25)

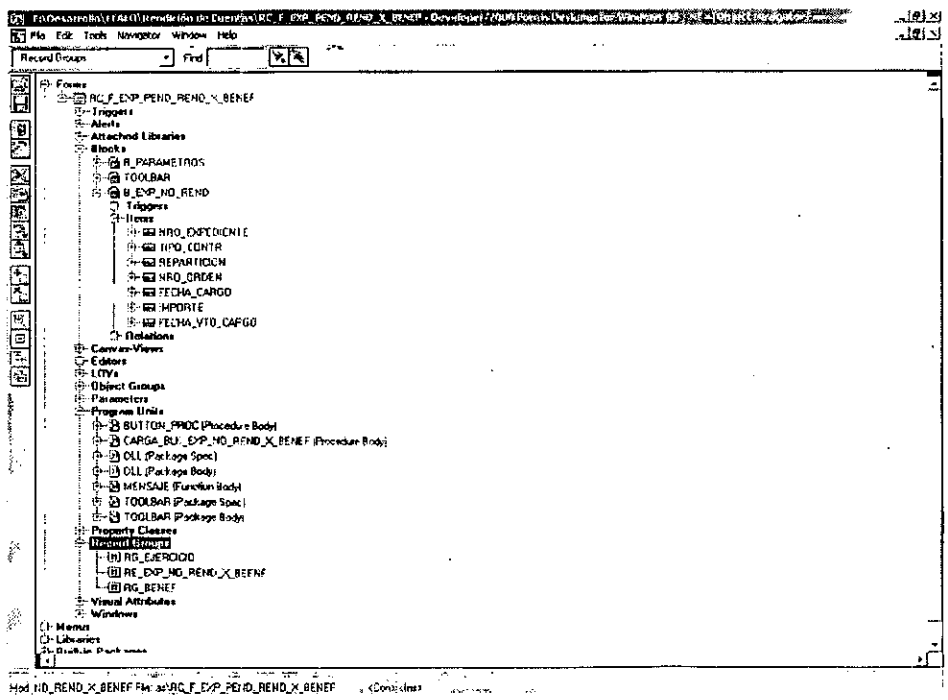
En la figura 7.25 se muestra la lista de valores desplegada que posibilita al usuario poder filtrar la consulta también por tipo de contratación.

7.3.1.5 Módulo de Expedientes sin Rendir por Beneficiarios

Los expedientes pendientes de rendición son aquellos que están sujetos a rendición de cuenta y que se han pagado pero aún no ha sido aprobada la rendición correspondiente a este expedienteo que hasta el momento no se haya presentado. Este módulo difiere del que se presento anteriormente en que en este el listado se puede filtrar la consulta por beneficiarios y también en que en este se muestra la repartición que emitió la orden de pago.

7.3.1.5.1 Desarrollo del Módulo

En la Figura 7.26 se muestra en el ambiente de desarrollo los ítems relacionados al bloque B_EXP_NO_REND.



(Figura 7.26)

En la figura 7.26 también se muestran los *Program Units* y los *Record Groups* entre los cuales se encuentran las funcionalidades que permiten recuperar datos desde la base de datos y almacenarlos en el bloque B_EXP_NO_REND. A continuación se muestra el código fuente de estas funcionalidades.

CARGA_BLK_EXP_NO_REND_X_BENEF: Procedimiento que tiene la funcionalidad de cargar los datos en el bloque B_EXP_NO_REND.

```
PROCEDURE CARGA_BLK_EXP_NO_REND_X_BENEF IS
-- ------------------------------------------------------------------------
-- Declaración de variables
-- ------------------------------------------------------------------------
        rg_id             RecordGroup; -- identificación del record group
        expediente        GroupColumn; -- identificación de la columna del expediente
        tipo_contr        GroupColumn; -- identificación de la columna del tipo de
                                -- contratación del expediente
        reparticion       GroupColumn; -- identificación de la columna de la repartición
        nro_orden         GroupColumn; -- identificación de la columna del número de orden
                                -- de pago
        fecha_cargo       GroupColumn; -- identificación de la columna de la fecha del pago
        importe           GroupColumn; -- identificación de la columna del importe
        fecha_vto         GroupColumn; -- identificación de la columna de la fecha de
                                -- vencimiento del cargo

        the_Rowcount    Number;
        sel_row          Number;
        ret_populate    Number;
        cod_error       Number;
BEGIN
        rg_id := Find_Group('RE_EXP_NO_REND_X_BEENF');
        ret_populate := POPULATE_GROUP(rg_id);
-- ------------------------------------------------------------------------
-- Si el record group encontro valores, estos se cargan en el bloque de datos
-- ------------------------------------------------------------------------
 IF ret_populate = 0 THEN
           expediente      := Find_Column('RE_EXP_NO_REND_X_BEENF.EXPEDIENTE');
           tipo_contr      := Find_Column('RE_EXP_NO_REND_X_BEENF.DESCRIPCION');
           reparticion     := Find_Column('RE_EXP_NO_REND_X_BEENF.REPARTICION');
           nro_orden       := Find_Column('RE_EXP_NO_REND_X_BEENF.NRO_ORDEN_PAGO');
           fecha_cargo     := Find_Column('RE_EXP_NO_REND_X_BEENF.FECHA_CARGO');
           importe         := Find_Column('RE_EXP_NO_REND_X_BEENF.IMPORTE');
           fecha_vto       := Find_Column('RE_EXP_NO_REND_X_BEENF.FECHA_VTO_CARGO');


           the_Rowcount := Get_Group_Row_Count(rg_id);

           go_block('B_EXP_NO_REND');
           clear_block;

           FOR j IN 1..the_Rowcount LOOP
              Set_Group_Selection(rg_id, j);
              sel_row := Get_Group_Selection(rg_id, j);
              CREATE_RECORD;

              :B_EXP_NO_REND.NRO_EXPEDIENTE  := Get_Group_Char_Cell(expediente, sel_row);
              :B_EXP_NO_REND.TIPO_CONTR      := Get_Group_Char_Cell(tipo_contr, sel_row);
              :B_EXP_NO_REND.REPARTICION     := Get_Group_Char_Cell(reparticion, sel_row);
              :B_EXP_NO_REND.NRO_ORDEN       := Get_Group_Char_Cell(nro_orden, sel_row);
              :B_EXP_NO_REND.FECHA_CARGO     := Get_Group_Date_Cell(fecha_cargo, sel_row);
              :B_EXP_NO_REND.IMPORTE         := Get_Group_Number_Cell(importe, sel_row);
              :B_EXP_NO_REND.FECHA_VTO_CARGO := Get_Group_Date_Cell(fecha_vto, sel_row);

    END LOOP;
          FIRST_RECORD;
       ELSIF ret_populate = 1403 THEN
          cod_error := MENSAJE('ACEPTAR','ERROR EN PARAMETROS','PARAMETROS: No se han
                       encontrado registros para los parametros brindados...');
    :B_PARAMETROS.ID_BENEFICIARIO :=NULL;
    :B_PARAMETROS.FECHA_DESDE  := NULL;
          :B_PARAMETROS.FECHA_HASTA  := NULL;
          :B_PARAMETROS.RAZON_SOCIAL := NULL;
          go_item('b_parametros.fecha_desde');
 ELSE
    cod_error := MENSAJE('ACEPTAR','ERROR EN RECORD GROUP' , 'ERROR:'
```

```
                        ||TO_CHAR(ret_populate)||': No se han encontrado datos para
                         el RecordGroup RE_EXP_NO_REND...');
        END IF;
END;
```

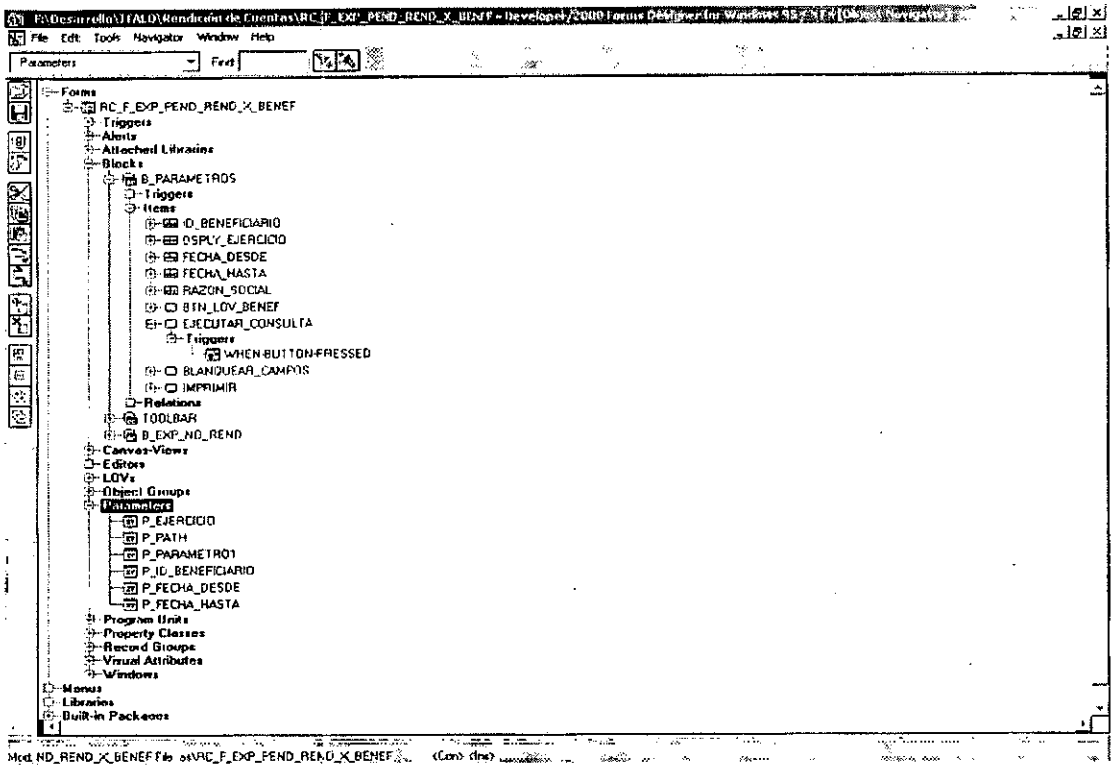
RE_EXP_NO_REND: Tiene la funcionalidad de recuperar los datos desde las tablas.

```
SELECT
  e.inic ||' - '|| e.numero ||' - '|| e.ejercicio expediente,
  tc.descripcion,
  i.instituc||'-'||i.jur||'-'||i.u_ejec||' '||i.descripcion repartición,
  ror.numero nro_orden_pago,
  ror.fecha_cargo,
  rod.importe,
  ror.fecha_vto_cargo
FROM
  co_registro_orden_detalles rod,
  co_beneficiarios b,
  ngr_institucional i,
  co_ctas_presup_gasto_anual cga,
  co_expedientes e,
  ngr_tipos_contr tc,
  (SELECT o.numero,
          o.fecha_cargo,
          o.fecha_vto_cargo,
          o.nro_secuencia_exp,
          o.id_beneficiario,
          o.nro_secuencia,
                  o.secuencia_cta_gasto_anual
         FROM co_registro_ordenes o
    WHERE o.cargo IS NOT NULL
          AND o.fecha_cargo >= :FECHA_DESDE
          AND o.fecha_cargo <= :FECHA_HASTA
          AND o.tipo = 'OP'
          AND o.id_beneficiario = NVL(:idbenf, o.id_beneficiario)
          AND o.EJERCICIO = :EJERCICIO
          AND (NOT EXISTS (SELECT 1
                             FROM co_rc_rendiciones rr
                            WHERE rr.co_r_or_nro_secuencia = o.nro_secuencia))
  UNION
   SELECT o.numero,
          o.fecha_cargo,
                o.fecha_vto_cargo,
                 o.nro_secuencia_exp,
                 o.id_beneficiario,
                 o.nro_secuencia,
                 o.secuencia_cta_gasto_anual
         FROM co_registro_ordenes o,co_rc_rendiciones r
        WHERE o.cargo IS NOT NULL
    AND o.fecha_cargo >= :FECHA_DESDE
          AND o.fecha_cargo <= :FECHA_HASTA
          AND o.tipo = 'OP'
          AND o.id_beneficiario = NVL(:idbenf, o.id_beneficiario)
          AND o.EJERCICIO = :EJERCICIO
          AND (r.aprobada = 'NO' AND  r.co_r_or_nro_secuencia = o.nro_secuencia )
  ) ror
WHERE b.id = ror.id_beneficiario
  AND tc.id_tipo_contr = e.id_tipo_contr
  AND rod.nro_secuencia_op = ror.nro_secuencia
  AND ror.nro_secuencia_exp = e.nro_secuencia
  AND rod.id_retencion = 99
  AND cga.secuencia = ror.secuencia_cta_gasto_anual
  AND i.instituc = cga.instituc
  AND i.jur = cga.jur
  AND i.jur_aux = cga.jur_aux
  AND i.u_ejec = cga.u_ejec
```

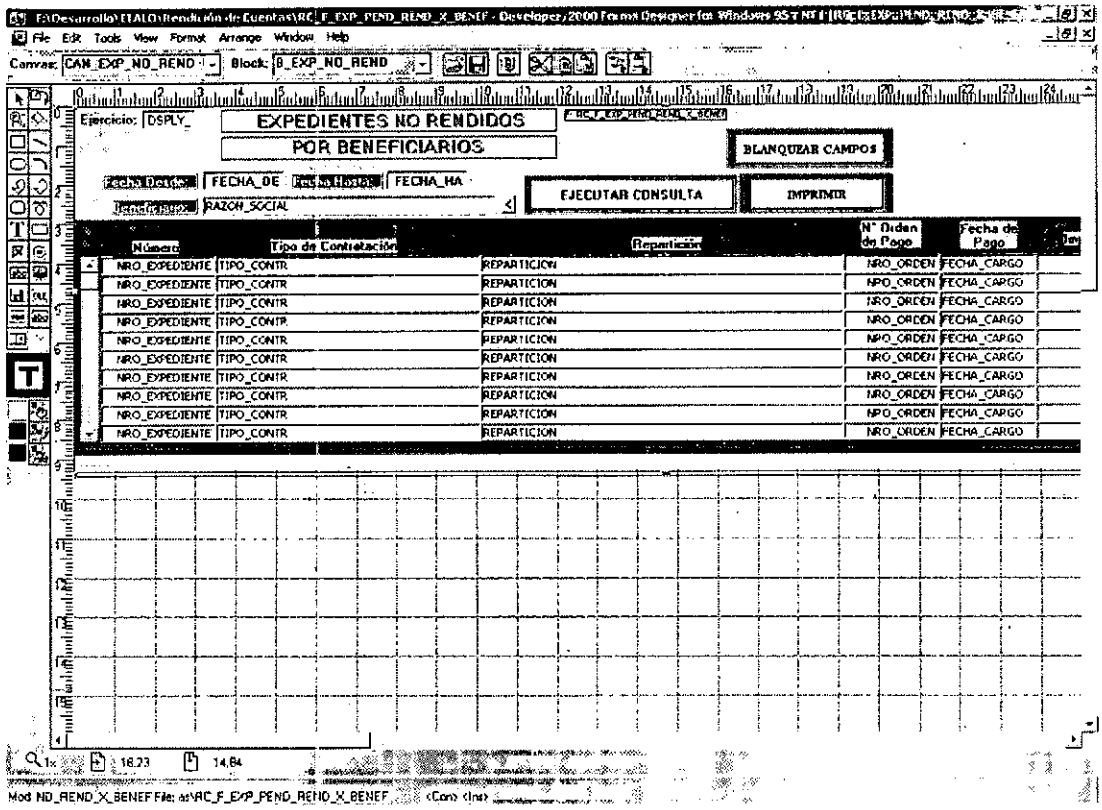
La definición de los parámetros la podemos ver en la figura 7.27 dichos parámetros son cargados al ejecutarse la funcionalidad relacionada con el botón "EJECUTAR CONSULTA" que se muestra a continuación.

```
DECLARE
  v_mensaje  number;
BEGIN
        IF :B_PARAMETROS.FECHA_DESDE IS NULL THEN
```

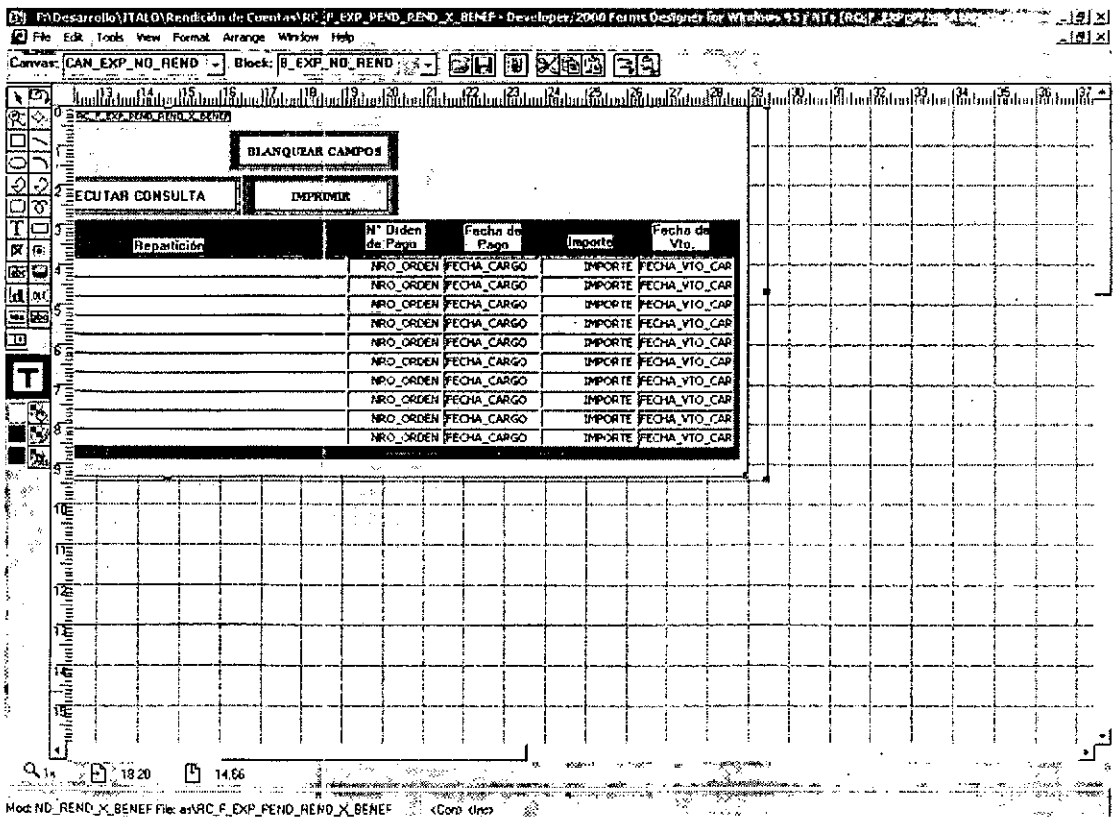
```
    v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha desde. . .');
  ELSIF :B_PARAMETROS.FECHA_HASTA IS NULL THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha hasta. . .');
  ELSIF :B_PARAMETROS.FECHA_DESDE > :B_PARAMETROS.FECHA_HASTA THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','La "fecha desde" debe ser menor o igual a
                         la "fecha hasta". . .');
  ELSE
   :PARAMETER.P_FECHA_DESDE      := TO_CHAR(:B_PARAMETROS.FECHA_DESDE,'DD/MM/YYYY');
   :PARAMETER.P_FECHA_HASTA      := TO_CHAR(:B_PARAMETROS.FECHA_HASTA,'DD/MM/YYYY');
   IF :B_PARAMETROS.ID_BENEFICIARIO IS NULL THEN :PARAMETER.P_ID_BENEFICIARIO := NULL;
   ELSE :PARAMETER.P_ID_BENEFICIARIO := :B_PARAMETROS.ID_BENEFICIARIO;
   END IF;
   :PARAMETER.P_EJERCICIO        := :B_PARAMETROS.DSPLY_EJERCICIO;
    CARGA_BLK_EXP_NO_REND_X_BENEF;
  END IF;
END;
```



(Figura 7.27)



(Figura 7.28)



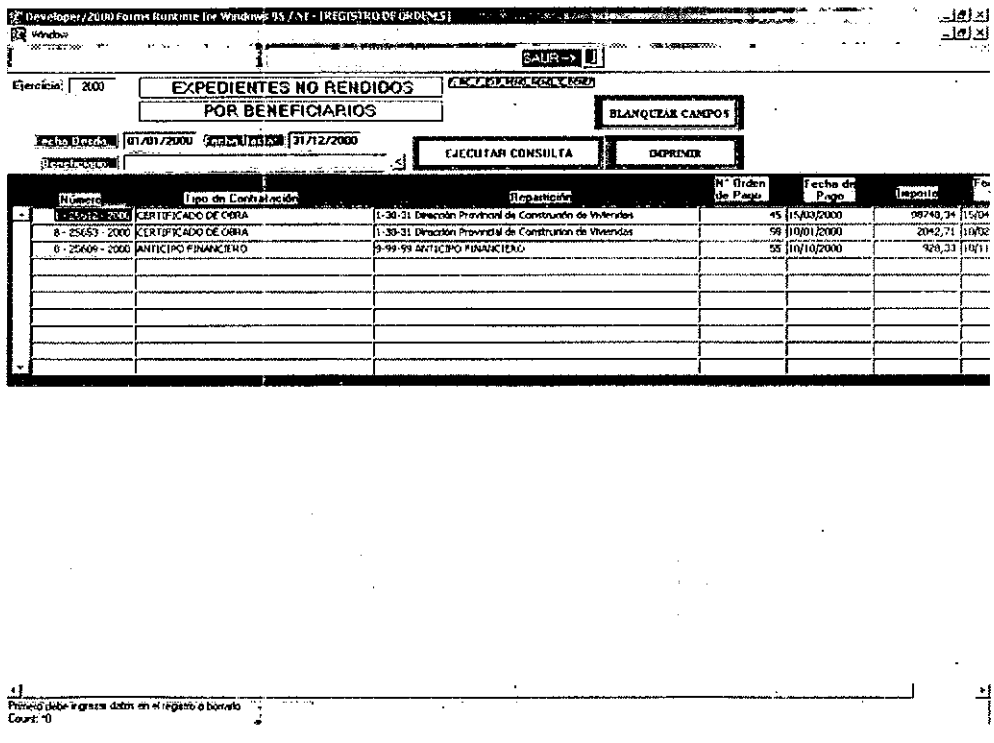
(Figura 7.29)

En las figuras 7.28 y 7.29 se muestran el diseño visible del módulo, en estas figuras se puede apreciar los ítems del bloque mencionado en el anteriormente.

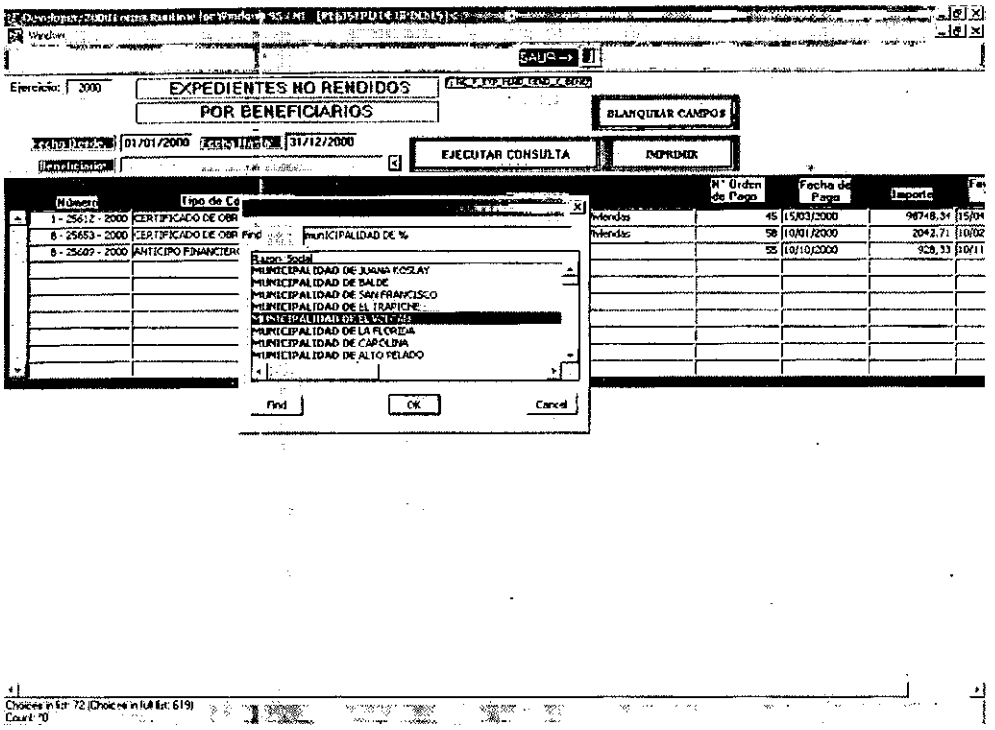
7.3.1.5.2 Ejecución del Módulo

En este punto se muestra al módulo "Expedientes sin Rendir por Beneficiarios" en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.

En la figura 7.30 se muestra el módulo de expedientes pendientes de rendición en este caso si se muestran datos ya que la consulta que recupera los datos ha sido ejecutada. En esta figura se puede observar los campos correspondientes a los ítems mostrados en la figura 7.26.



(Figura 7.30)



(Figura 7.31)

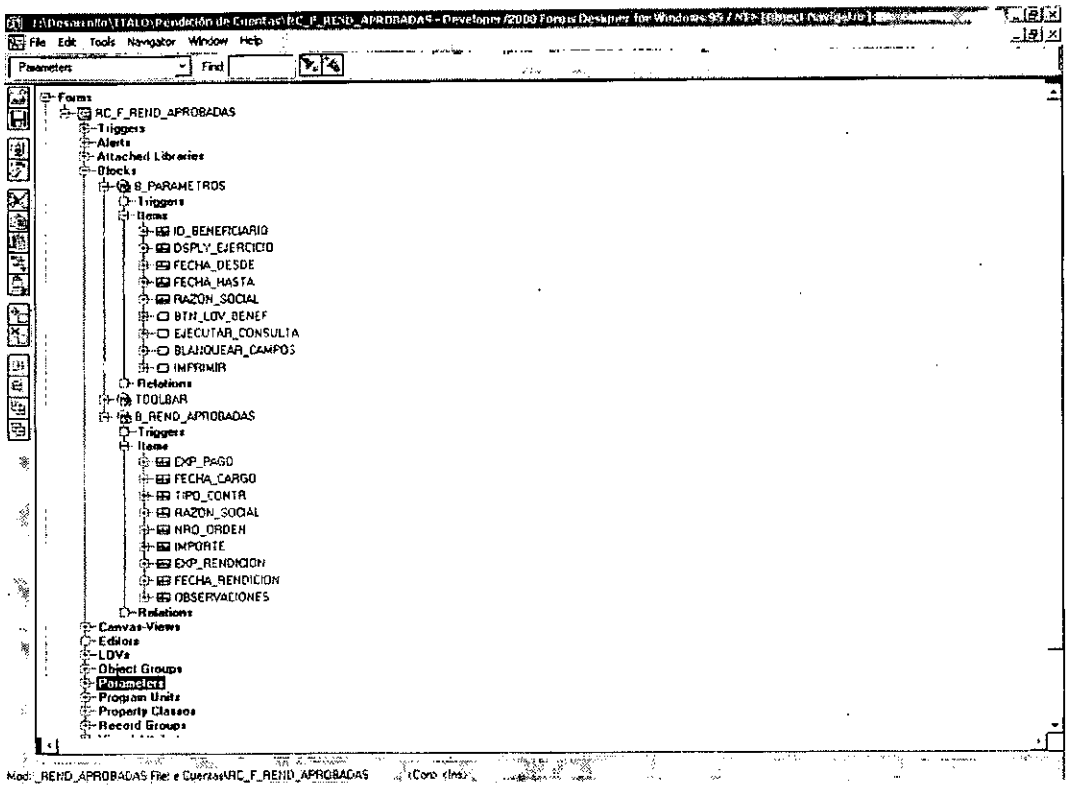
En la figura 7.31 se muestra la lista de valores desplegada que posibilita al usuario poder filtrar la consulta por la razón social del beneficiario.

7.3.1.6 Módulo de Rendiciones Aprobadas

Las Rendiciones Aprobadas son aquellas que fueron presentadas con toda la documentación necesaria y que en la Contaduría General de Provincia interprete que esta bien realizada.

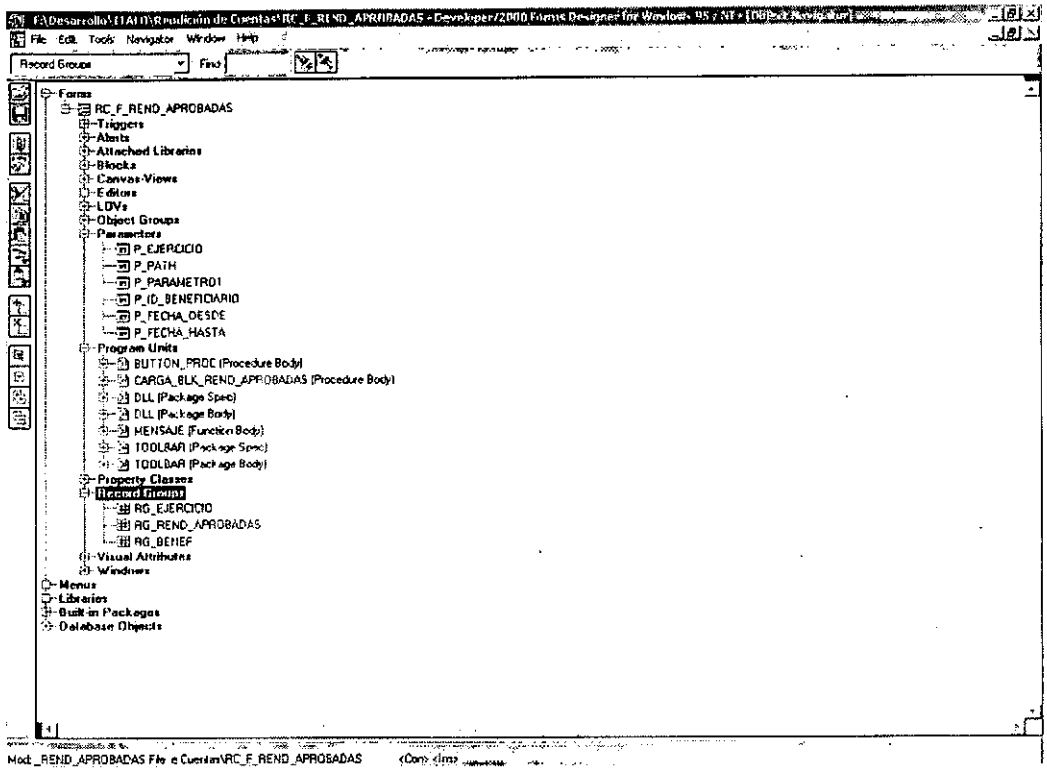
7.3.1.6.1 Desarrollo del Módulo

En la figura 7.32 se muestran lo items de los principales bloques de este módulo. Mientras que en la Figura 7.33 se muestran otros puntos de funcionalidad importante dentro de este módulo. Se muestran los *Program Units*



(Figura 7.32)

donde se encuentra el procedimiento que carga los datos en el bloque B_REND_APROBADAS y en los *Record Groups* se observa el record group RG_REND_APROBADAS que es la funcionalidad que recupera los datos desde la base de datos.



(Figura 7.33)

A continuación se muestra el código fuente de las funcionalidades mencionadas anteriormente.

CARGA_BLK_REND_APROBADAS: Procedimiento que tiene la funcionalidad de cargar los datos en el bloque B_REND_APROBADAS.

```
PROCEDURE CARGA_BLK_REND_APROBADAS IS
-- ------------------------------------------------------------------------
-- Declaración de variables
-- ------------------------------------------------------------------------
        rg_id     RecordGroup; -- identificación del record group
        exp_pago  GroupColumn; -- identificación de la columna del expediente de pago
 fecha_cargo GroupColumn; -- identificación de la columna de la fecha del pago
        tipo_contr GroupColumn; -- identificación de la columna del tipo de contratación del
                                -- expediente
        beneficiario GroupColumn; -- identificación de la columna de la razon social del
                                  --  beneficiario
        nro_orden GroupColumn; -- identificación de la columna del número de orden de pago
        importe   GroupColumn; -- identificación de la columna del importe
        exp_rendicion GroupColumn; -- identificación de la columna del expediente de
                                   -- rendición
        fecha_rend GroupColumn; -- identificación de la columna de la fecha de rendición
        observaciones GroupColumn; -- identificación de la columna de observaciones

        the_Rowcount    Number;
        sel_row          Number;
        ret_populate    Number;
        cod_error       Number;
BEGIN
        rg_id := Find_Group('RG_REND_APROBADAS');
        ret_populate := POPULATE_GROUP(rg_id);
-- ------------------------------------------------------------------------
-- Si el record group encontro valores, estos se cargan en el bloque de datos
-- ------------------------------------------------------------------------
  IF ret_populate = 0 THEN
          exp_pago        := Find_Column('RG_REND_APROBADAS.EXPEDIENTE');
          fecha_cargo     := Find_Column('RG_REND_APROBADAS.FECHA_PAGO');
          tipo_contr      := Find_Column('RG_REND_APROBADAS.TIPO_CONTR');
```

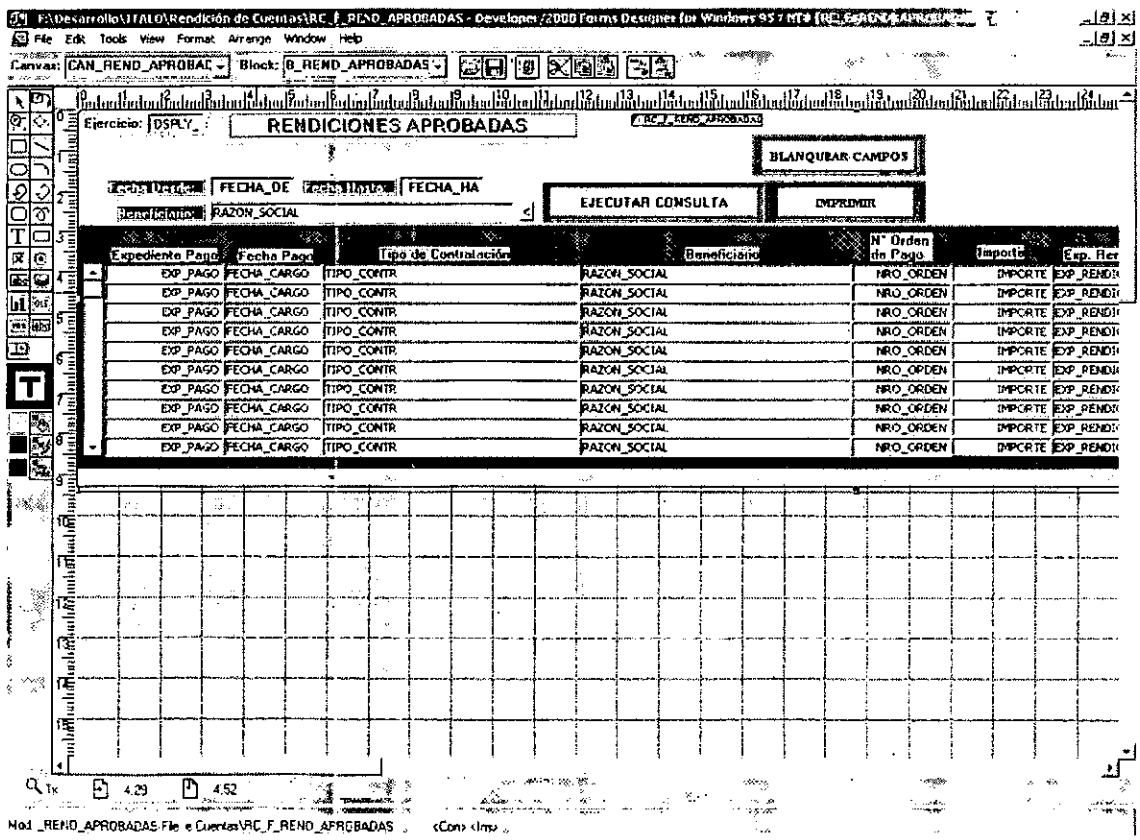
```
        beneficiario    := Find_Column('RG_REND_APROBADAS.RAZON_SOCIAL');
        nro_orden       := Find_Column('RG_REND_APROBADAS.NUMERO');
        importe         := Find_Column('RG_REND_APROBADAS.IMPORTE');
        exp_rendicion   := Find_Column('RG_REND_APROBADAS.EXP_REND');
        fecha_rend      := Find_Column('RG_REND_APROBADAS.FECHA_ENTREGA');
        observaciones   := Find_Column('RG_REND_APROBADAS.OBSERVACIONES');

        the_Rowcount := Get_Group_Row_Count(rg_id);

        go_block('B_REND_APROBADAS');
        clear_block;

        FOR j IN 1..the_Rowcount LOOP
           Set_Group_Selection(rg_id, j);
           sel_row := Get_Group_Selection(rg_id, j);
           CREATE_RECORD;

           :B_REND_APROBADAS.EXP_PAGO       := Get_Group_Char_Cell(exp_pago, sel_row);
           :B_REND_APROBADAS.FECHA_CARGO    := Get_Group_Date_Cell(fecha_cargo, sel_row);
           :B_REND_APROBADAS.TIPO_CONTR     := Get_Group_Char_Cell(tipo_contr, sel_row);
           :B_REND_APROBADAS.RAZON_SOCIAL   := Get_Group_Char_Cell(beneficiario, sel_row);
           :B_REND_APROBADAS.NRO_ORDEN      := Get_Group_Char_Cell(nro_orden, sel_row);
           :B_REND_APROBADAS.IMPORTE        := Get_Group_Number_Cell(importe, sel_row);
           :B_REND_APROBADAS.EXP_RENDICION:= Get_Group_Char_Cell(exp_rendicion, sel_row);
           :B_REND_APROBADAS.FECHA_RENDICION := Get_Group_Date_Cell(fecha_rend, sel_row);
      :B_REND_APROBADAS.OBSERVACIONES := Get_Group_Char_Cell(observaciones, sel_row);

    END LOOP;
         FIRST_RECORD;
       ELSIF ret_populate = 1403 THEN
          cod_error := MENSAJE('ACEPTAR','ERROR EN PARAMETROS','PARAMETROS: No se han
                       encontrado registros para los parametros brindados...');
     :B_PARAMETROS.ID_BENEFICIARIO :=NULL;
          :B_PARAMETROS.FECHA_DESDE  := NULL;
          :B_PARAMETROS.FECHA_HASTA  := NULL;
          :B_PARAMETROS.RAZON_SOCIAL := NULL;
          go_item('b_parametros.fecha_desde');
 ELSE
     cod_error := MENSAJE('ACEPTAR','ERROR EN RECORD GROUP' , 'ERROR: ' ||
                          TO_CHAR(ret_populate)||': No se han encontrado datos para
                          el RecordGroup RE_EXP_NO_REND...');
       END IF;
END;
```

RG_REND_APROBADAS: Tiene la funcionalidad de recuperar los datos desde las tablas.

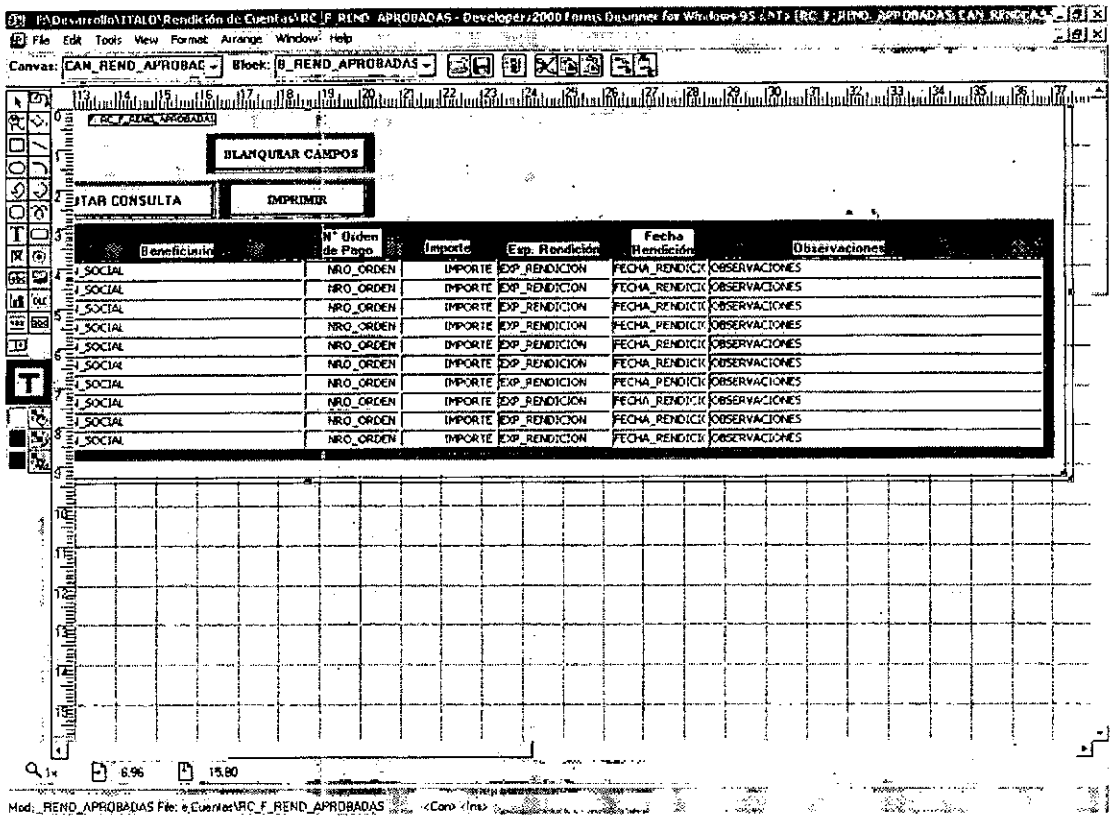
```
-- ------------------------------------------------------------------
-- LISTADO DE RENDICIONES APROBADAS
-- ------------------------------------------------------------------

SELECT
  ep.inic||' - '|| ep.numero ||' - '|| ep.ejercicio expediente,
  o.fecha_cargo fecha_pago,
  tc.descripcion tipo_contr,
  b.razon_social,
  o.numero,
  od.importe,
  er.inic||' - '|| er.numero ||' - '|| er.ejercicio exp_rend,
  r.fecha_entrega,
  r.observaciones
FROM
  co_rc_rendiciones r,
  co_registro_ordenes o,
  co_beneficiarios b,
  co_expedientes ep,
  ngr_tipos_contr tc,
  co_registro_orden_detalles od,
  co_expedientes er
WHERE r.aprobada = 'SI'
  and o.nro_secuencia = r.co_r_or_nro_secuencia
  and o.nro_secuencia_exp = ep.nro_secuencia
  and tc.id_tipo_contr = ep.id_tipo_contr
  and od.nro_secuencia_op = o.nro_secuencia
  and er.nro_secuencia = r.co_exp_nro_secuencia
  and od.id_retencion = 99
  and o.ejercicio = :p_ejercicio
  and r.fecha_entrega >= :p_fecha_desde
  and r.fecha_entrega <= :p_fecha_hasta
  and o.id_beneficiario = nvl(:p_id_beneficiario,o.id_beneficiario)
  and b.id = o.id_beneficiario
```

La definición de los parámetros la podemos ver en la figura 7.32 dichos parámetros son cargados al ejecutarse la funcionalidad relacionada con el botón "EJECUTAR CONSULTA" que se muestra a continuación.

```
DECLARE
  v_mensaje  number;
BEGIN
        IF :B_PARAMETROS.FECHA_DESDE IS NULL THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha desde. . .');
  ELSIF :B_PARAMETROS.FECHA_HASTA IS NULL THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha hasta. . .');
  ELSIF :B_PARAMETROS.FECHA_DESDE > :B_PARAMETROS.FECHA_HASTA THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','La "fecha desde" debe ser menor o igual a
                         la "fecha hasta". . .');
  ELSE
   :PARAMETER.P_FECHA_DESDE      := TO_CHAR(:B_PARAMETROS.FECHA_DESDE,'DD/MM/YYYY');
   :PARAMETER.P_FECHA_HASTA      := TO_CHAR(:B_PARAMETROS.FECHA_HASTA,'DD/MM/YYYY');
   IF :B_PARAMETROS.ID_BENEFICIARIO IS NULL THEN :PARAMETER.P_ID_BENEFICIARIO := NULL;
   ELSE :PARAMETER.P_ID_BENEFICIARIO := :B_PARAMETROS.ID_BENEFICIARIO;
   END IF;
   :PARAMETER.P_EJERCICIO        := :B_PARAMETROS.DSPLY_EJERCICIO;
   CARGA_BLK_REND_APROBADAS;
   END IF;
END;
```



(Figura 7.34)



(Figura 7.35)

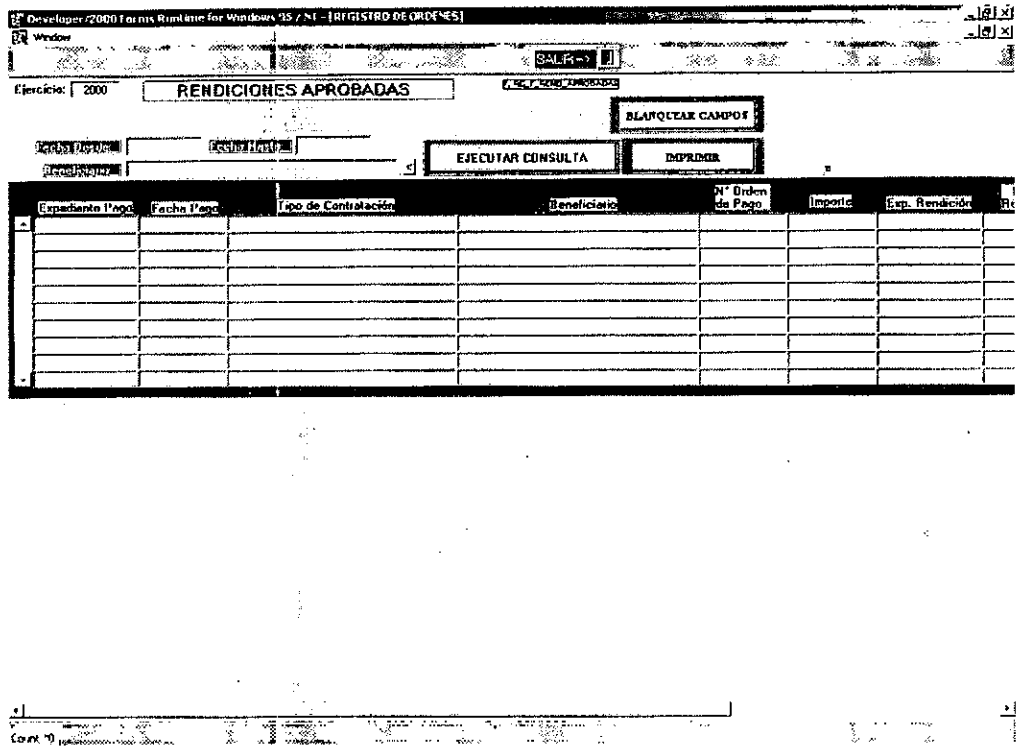
En las figuras 7.34 y 7.35 se muestran el diseño visible del módulo, en estas figuras se puede apreciar los ítems del bloque mencionado en el anteriormente.

7.3.1.6.2 Ejecución del Módulo

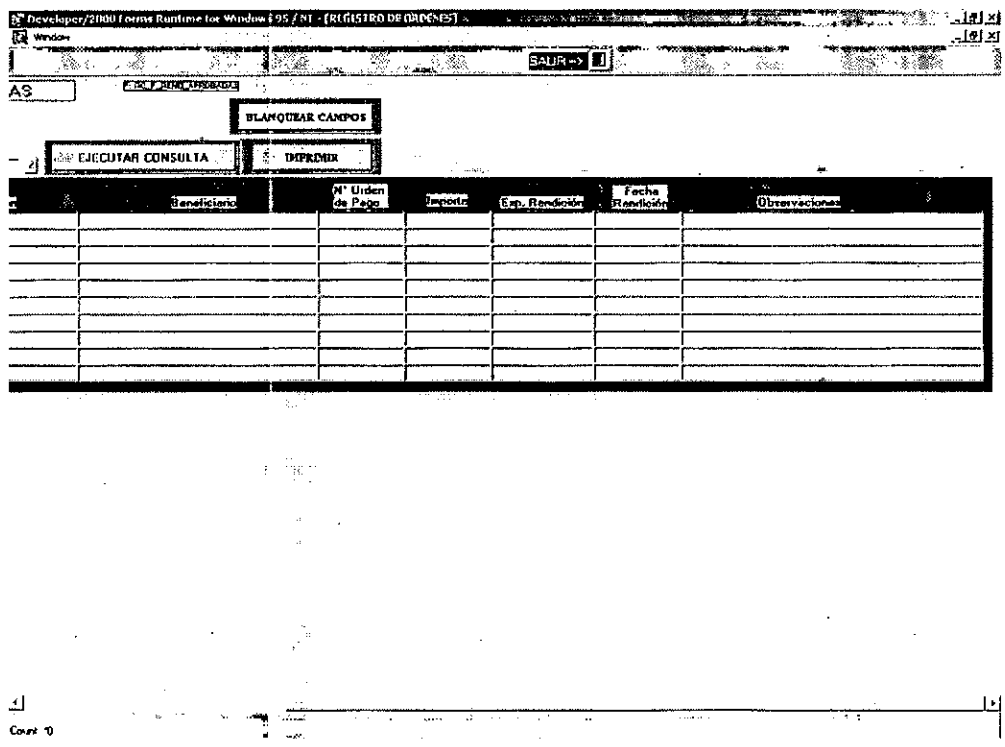
En este punto se muestra al módulo "Rendiciones Aprobadas" en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.

En las figuras 7.36 y 7.37 se muestra al módulo en ejecución, dichas imágenes se corresponden con las figuras 7.34 y 7.35 que muestran el desarrollo.

En las figuras 7.36 y 7.37 se puede observar que figuran todos los datos que corresponden al bloque B_REND_APROBADAS.

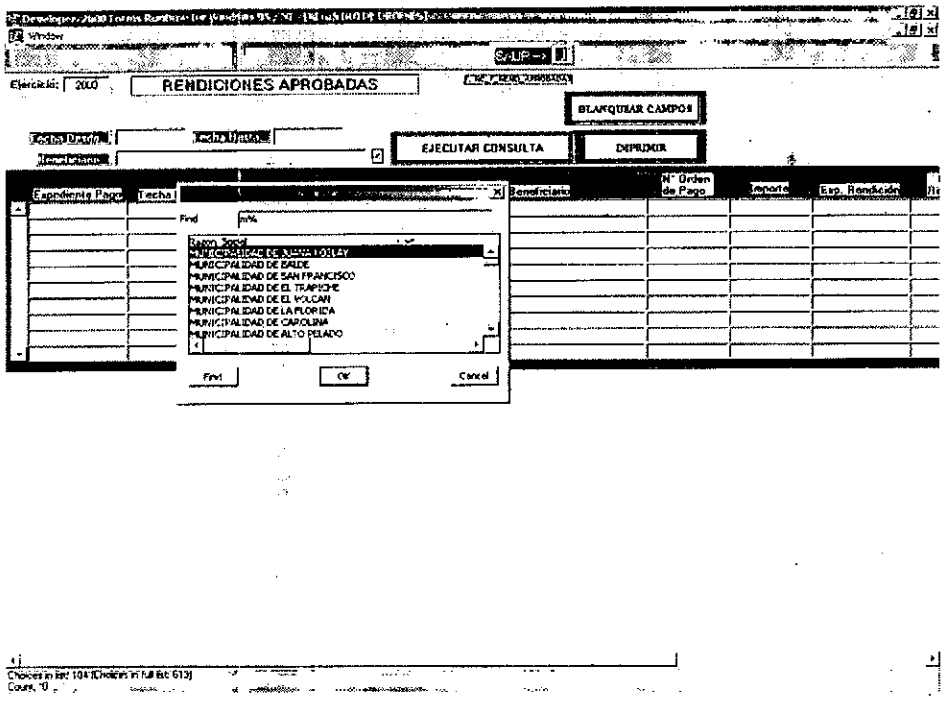


(Figura 7.36)



(Figura 7.37)

En la figura 7.38 se muestra la lista de valores desplegada que posibilita al usuario poder filtrar la consulta por la razón social del beneficiario.



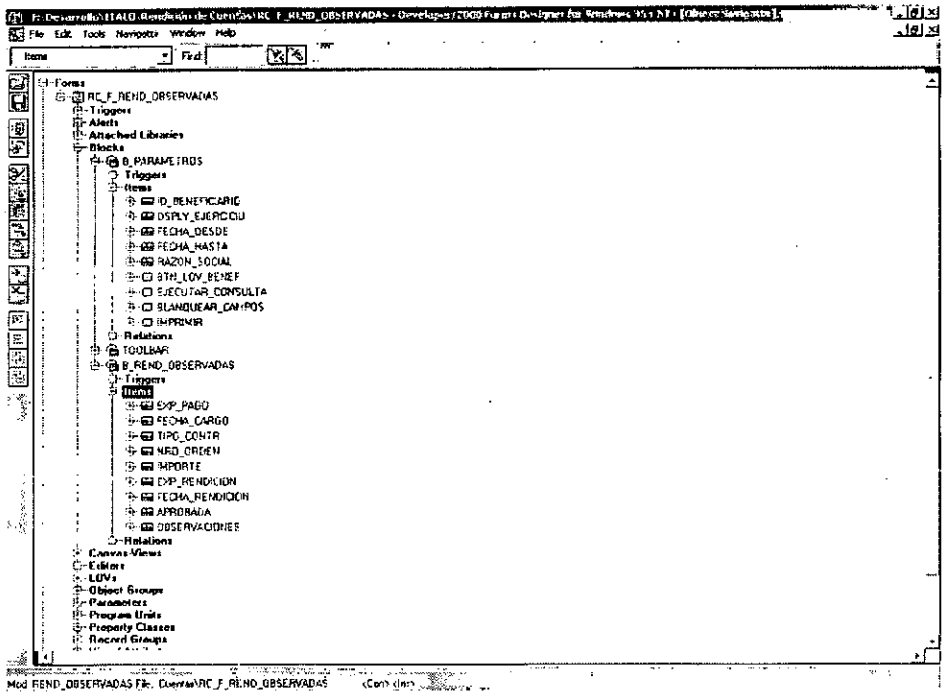
(Figura 7.38)

7.3.1.7 Módulo de Rendiciones Observadas

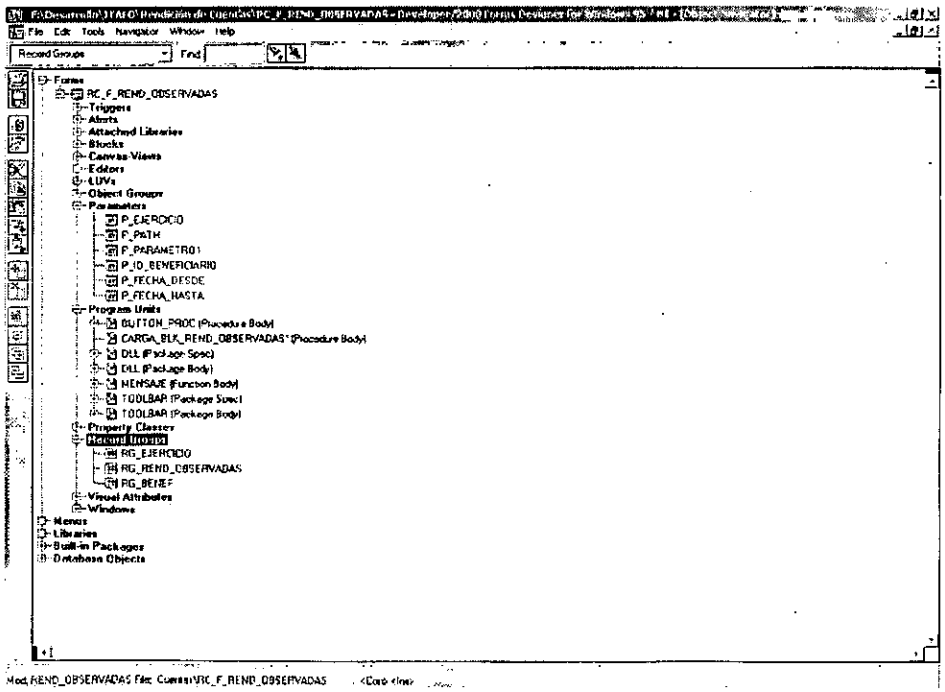
Las Rendiciones Observadas son aquellas que fueron presentadas y hayan sido aprobadas o no, tengan alguna observación realizada por la Contaduría General de Provincia interprete que esta bien realizada.

7.3.1.7.1 Desarrollo del Módulo

En la figura 7.39 se muestran lo ítems de los principales bloques de este módulo. Mientras que en la Figura 7.40 se muestran otros puntos de funcionalidad importante dentro de este módulo. Se muestran los *Program Units* donde se encuentra el procedimiento que carga los datos en el bloque B_REND_OBSERVADAS y en los *Record Groups* se observa el record group RG_REND_OBSERVADAS que es la funcionalidad que recupera los datos desde la base de datos.



(Figura 7.39)



(Figura 7.40)

A continuación se muestra el código fuente de las funcionalidades mencionadas anteriormente.

CARGA_BLK_REND_OBSERVADAS: Procedimiento que tiene la funcionalidad de cargar los datos en el bloque B_REND_OBSERVADAS.

```
PROCEDURE CARGA_BLK_REND_OBSERVADAS IS
-- ------------------------------------------------------------------------
-- Declaración de variables
-- ------------------------------------------------------------------------
        rg_id     RecordGroup; -- identificación del record group
        exp_pago  GroupColumn; -- identificación de la columna del expediente de pago
  fecha_cargo     GroupColumn; -- identificación de la columna de la fecha del pago
        tipo_contr GroupColumn; -- identificación de la columna del tipo de contratación del
                        -- expediente
        nro_orden GroupColumn; -- identificación de la columna del número de orden de pago
        importe   GroupColumn; -- identificación de la columna del importe
        exp_rendicion GroupColumn; -- identificación de la columna del expediente de
                           -- rendición
        fecha_rend GroupColumn; -- identificación de la columna de la fecha de rendición
        aprobada          GroupColumn; -- identificación de la columna de aprobación
        observaciones     GroupColumn; -- identificación de la columna de observaciones

        the_Rowcount     Number;
        sel_row           Number;
        ret_populate     Number;
        cod_error        Number;
BEGIN
        rg_id := Find_Group('RG_REND_OBSERVADAS');
        ret_populate := POPULATE_GROUP(rg_id);
-- ------------------------------------------------------------------------
-- Si el record group encontro valores, estos se cargan en el bloque de datos
-- ------------------------------------------------------------------------
        IF ret_populate = 0 THEN
           exp_pago         := Find_Column('RG_REND_OBSERVADAS.EXPEDIENTE');
           fecha_cargo      := Find_Column('RG_REND_OBSERVADAS.FECHA_PAGO');
     tipo_contr      := Find_Column('RG_REND_OBSERVADAS.TIPO_CONTR');
           nro_orden        := Find_Column('RG_REND_OBSERVADAS.NUMERO');
           importe          := Find_Column('RG_REND_OBSERVADAS.IMPORTE');
           exp_rendicion    := Find_Column('RG_REND_OBSERVADAS.EXP_REND');
           fecha_rend       := Find_Column('RG_REND_OBSERVADAS.FECHA_ENTREGA');
           aprobada         := Find_Column('RG_REND_OBSERVADAS.APROBADA');
           observaciones    := Find_Column('RG_REND_OBSERVADAS.OBSERVACIONES');

           the_Rowcount := Get_Group_Row_Count(rg_id);

           go_block('B_REND_OBSERVADAS');
           clear_block;

           FOR j IN 1..the_Rowcount LOOP
              Set_Group_Selection(rg_id, j);
              sel_row := Get_Group_Selection(rg_id, j);
              CREATE_RECORD;

              :B_REND_OBSERVADAS.EXP_PAGO      := Get_Group_Char_Cell(exp_pago, sel_row);
              :B_REND_OBSERVADAS.FECHA_CARGO   := Get_Group_Date_Cell(fecha_cargo, sel_row);
              :B_REND_OBSERVADAS.TIPO_CONTR    := Get_Group_Char_Cell(tipo_contr, sel_row);
              :B_REND_OBSERVADAS.NRO_ORDEN     := Get_Group_Char_Cell(nro_orden, sel_row);
              :B_REND_OBSERVADAS.IMPORTE       := Get_Group_Number_Cell(importe, sel_row);
              :B_REND_OBSERVADAS.EXP_RENDICION:= Get_Group_Char_Cell(exp_rendicion,sel_row);
              :B_REND_OBSERVADAS.FECHA_RENDICION:= Get_Group_Date_Cell(fecha_rend, sel_row);
              :B_REND_OBSERVADAS.APROBADA         := Get_Group_Char_Cell(aprobada, sel_row);
       :B_REND_OBSERVADAS.OBSERVACIONES:= Get_Group_Char_Cell(observaciones,sel_row);

     END LOOP;
           FIRST_RECORD;
        ELSIF ret_populate = 1403 THEN
           cod_error := MENSAJE('ACEPTAR','ERROR EN PARAMETROS','PARAMETROS: No se han
                        encontrado registros para los parametros brindados...');
     :B_PARAMETROS.ID_BENEFICIARIO :=NULL;
     :B_PARAMETROS.FECHA_DESDE  := NULL;
           :B_PARAMETROS.FECHA_HASTA  := NULL;
           :B_PARAMETROS.RAZON_SOCIAL := NULL;
           go_item('b_parametros.fecha_desde');
  ELSE
     cod_error := MENSAJE('ACEPTAR','ERROR EN RECORD GROUP' , 'ERROR: ' ||
                          TO_CHAR(ret_populate)||': No se han encontrado datos para
                          el RecordGroup RE_EXP_NO_REND...');
        END IF;
END;
```

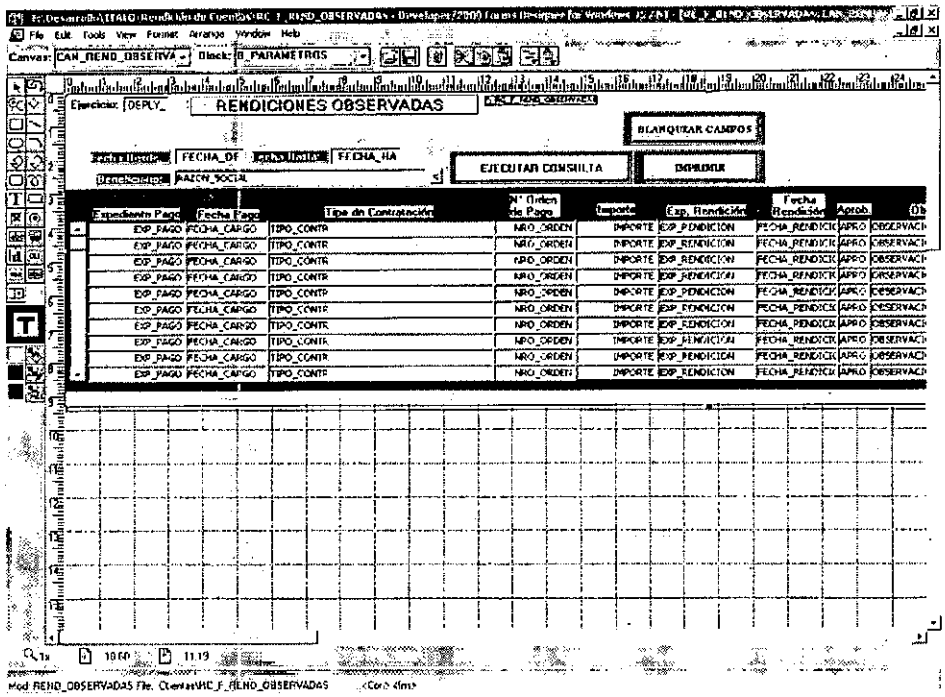
RG_REND_OBSERVADAS: Tiene la funcionalidad de recuperar los datos desde las tablas.

```
-- ------------------------------------------------------------------
-- LISTADO DE RENDICIONES APROBADAS
-- ------------------------------------------------------------------
```

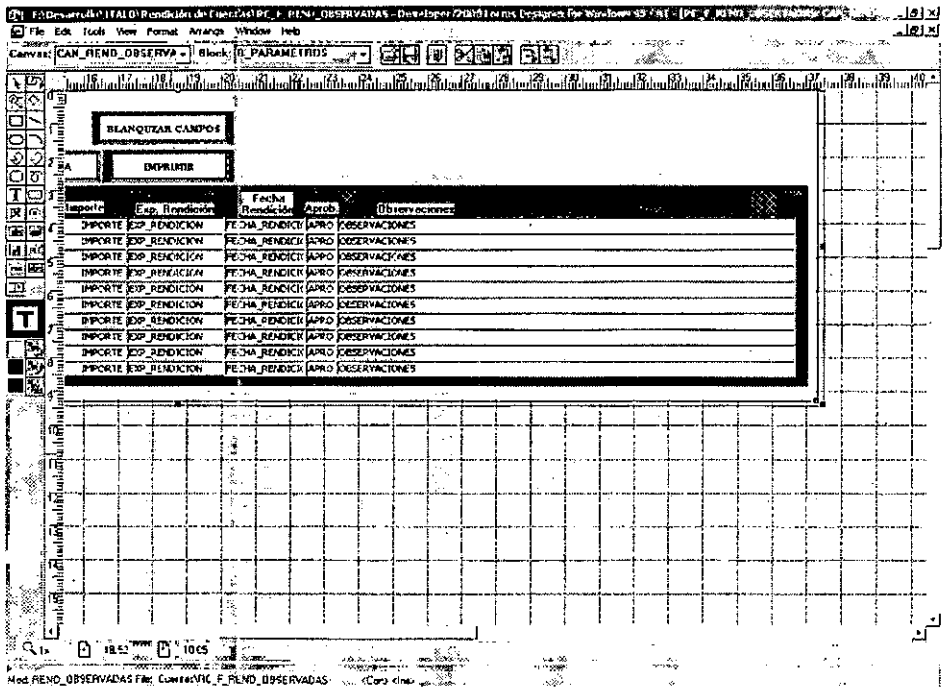
```
SELECT
  ep.inic||' - '|| ep.numero ||' - '|| ep.ejercicio expediente,
  o.fecha_cargo fecha_pago,
  tc.descripcion tipo_contr,
  b.razon_social,
  o.numero,
  od.importe,
  er.inic||' - '|| er.numero ||' - '|| er.ejercicio exp_rend,
  r.fecha_entrega,
  r.observaciones
FROM
  co_rc_rendiciones r,
  co_registro_ordenes o,
  co_beneficiarios b,
  co_expedientes ep,
  ngr_tipos_contr tc,
  co_registro_orden_detalles od,
  co_expedientes er
WHERE r.aprobada = 'SI'
  and o.nro_secuencia = r.co_r_or_nro_secuencia
  and o.nro_secuencia_exp = ep.nro_secuencia
  and tc.id_tipo_contr = ep.id_tipo_contr
  and od.nro_secuencia_op = o.nro_secuencia
  and er.nro_secuencia = r.co_exp_nro_secuencia
  and od.id_retencion = 99
  and o.ejercicio = :p_ejercicio
  and r.fecha_entrega >= :p_fecha_desde
  and r.fecha_entrega <= :p_fecha_hasta
  and o.id_beneficiario = nvl(:p_id_beneficiario,o.id_beneficiario)
  and b.id = o.id_beneficiario
```

La definición de los parámetros la podemos ver en la figura 3.1.39 dichos parámetros son cargados al ejecutarse la funcionalidad relacionada con el botón "EJECUTAR CONSULTA" que se muestra a continuación.

```
DECLARE
  v_mensaje  number;
BEGIN
       IF :B_PARAMETROS.FECHA_DESDE IS NULL THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha desde. . .');
  ELSIF :B_PARAMETROS.FECHA_HASTA IS NULL THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','Debe especificar la fecha hasta. . .');
  ELSIF :B_PARAMETROS.FECHA_DESDE > :B_PARAMETROS.FECHA_HASTA THEN
    v_mensaje := MENSAJE('ACEPTAR','ERROR','La "fecha desde" debe ser menor o igual a
                          la "fecha hasta". . .');
  ELSE
    :PARAMETER.P_FECHA_DESDE      := TO_CHAR(:B_PARAMETROS.FECHA_DESDE,'DD/MM/YYYY');
    :PARAMETER.P_FECHA_HASTA      := TO_CHAR(:B_PARAMETROS.FECHA_HASTA,'DD/MM/YYYY');
    IF :B_PARAMETROS.ID_BENEFICIARIO IS NULL THEN :PARAMETER.P_ID_BENEFICIARIO:= NULL;
    ELSE :PARAMETER.P_ID_BENEFICIARIO := :B_PARAMETROS.ID_BENEFICIARIO;
    END IF;
    :PARAMETER.P_EJERCICIO        := :B_PARAMETROS.DSPLY_EJERCICIO;
    CARGA_BLK_REND_OBSERVADAS;
  END IF;
END;
```



(Figura 7.41)

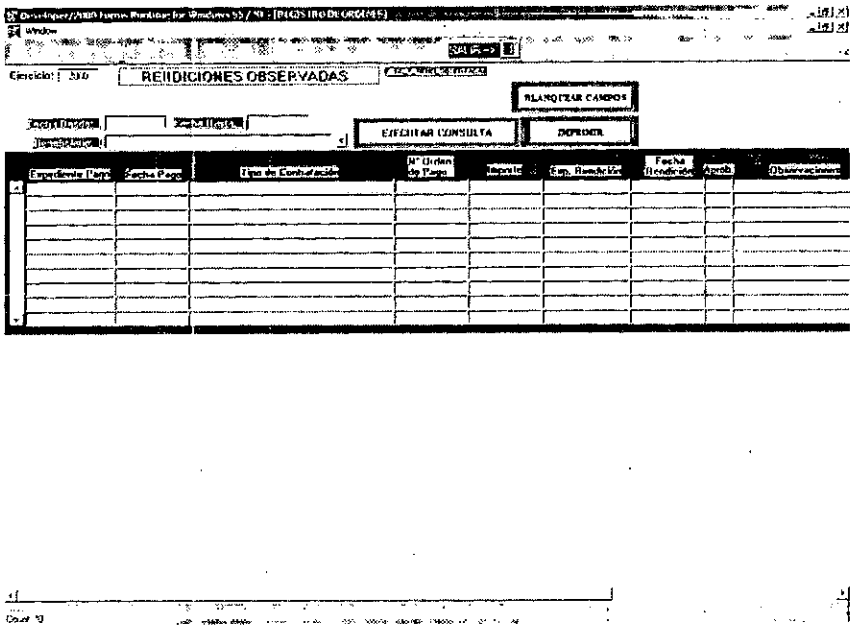


(Figura 7.42)

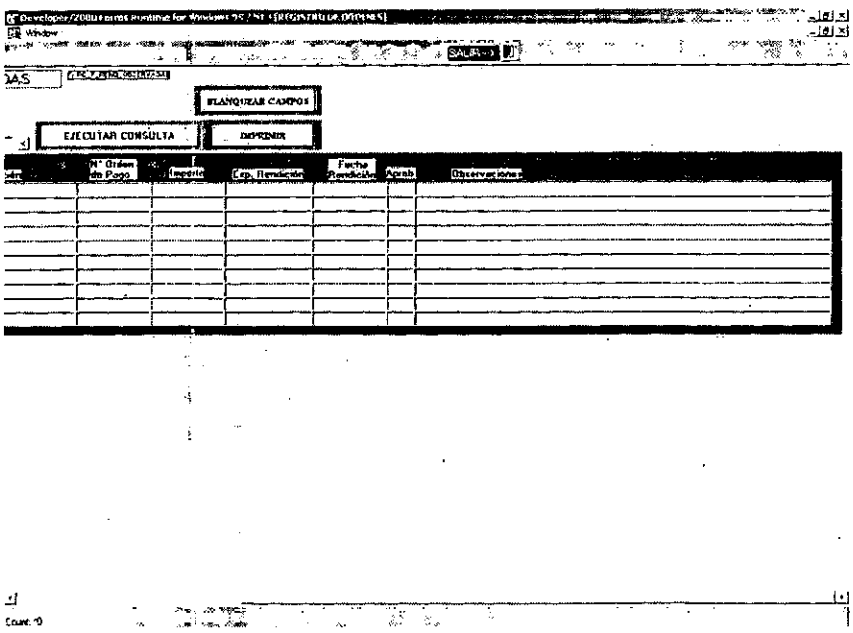
En las figuras 7.41 y 7.42 se muestran el diseño visible del módulo, en estas figuras se puede apreciar los ítems del bloque mencionado en el anteriormente.

7.3.1.7.2 Ejecución del Módulo

En este punto se muestra al módulo "Rendiciones Observadas" en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.



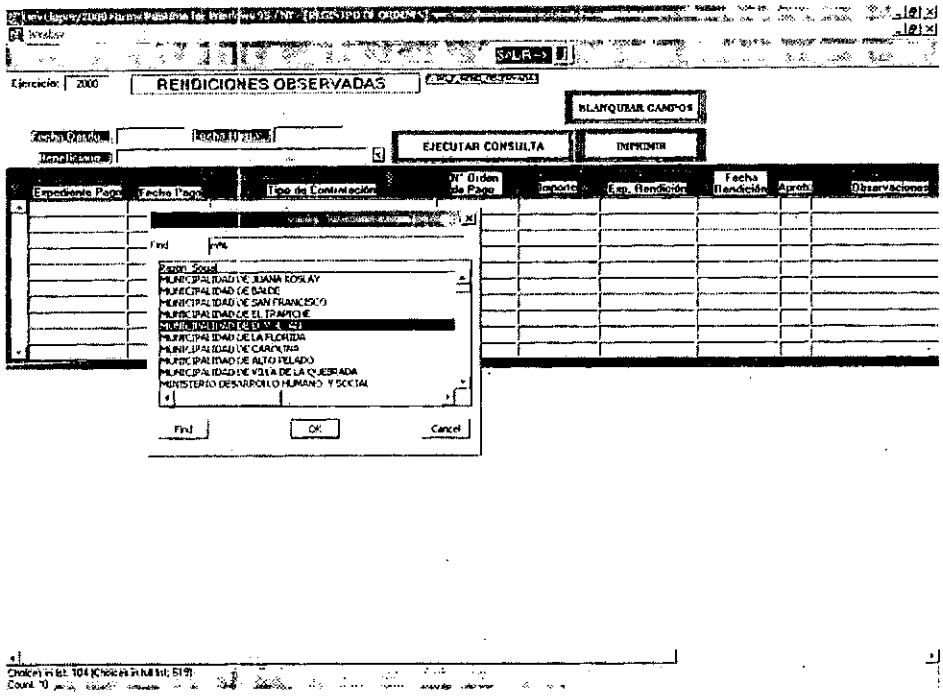
(Figura 7.43)



(Figura 7.44)

En las figuras 7.43 y 7.44 se muestra al módulo en ejecución, dichas imágenes se corresponden con las figuras 7.41 y 7.42 que muestran el desarrollo.

En las figuras 7.43 y 7.44 se puede observar que figuran todos los datos que corresponden al bloque B_REND_OBSERVADAS.



(Figura 7.45)

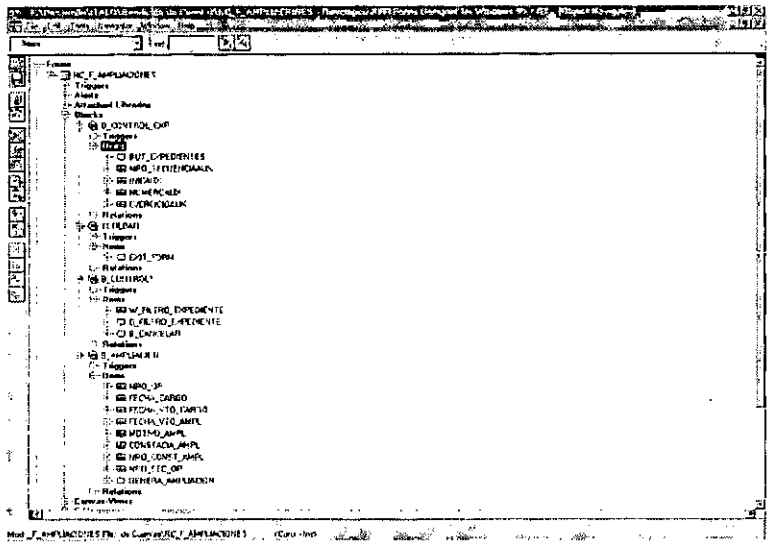
En la figura 7.45 se muestra la lista de valores desplegada que posibilita al usuario poder filtrar la consulta por la razón social del beneficiario.

7.3.1.8 Módulo de Ampliaciones

Este módulo permite generar extensiones en las fechas de vencimiento de las Rendiciones de Cuentas. El módulo interactúa directamente con la tabla CO_RC_AMPLIACIONES donde quedan registradas todas las ampliaciones de plazos que se le hacen a una orden de pago que debe rendirse. Estas ampliaciones suceden en casos excepcionales.

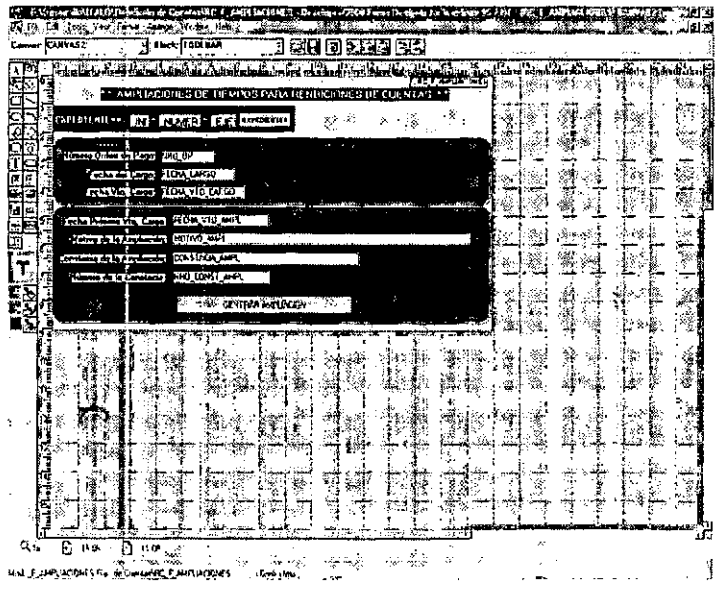
7.3.1.8.1 Desarrollo del Módulo

En la Figura 7.46 se muestra en el ambiente de desarrollo los bloques e ítems que forman parte del módulo, estos ítems estarán visibles por el usuario del sistema cuando el módulo esté en ejecución.



(Figura 7.46)

El Bloque B_AMPLIACION que se observa en la figura 7.46 es el que está directamente relacionado con la tabla CO_RC_AMPLIACIONES que se mencionó anteriormente.



(Figura 7.47)

En la figura 7.47 se muestra el desarrollo de la pantalla principal de este módulo, esta pantalla es la que interactuará con el usuario final, en ella el usuario deberá seleccionar un expediente determinado, al seleccionarlo el sistema recuperará el número de la orden de pago, la fecha y vencimiento del cargo, luego el usuario deberá cargar la fecha del próximo vencimiento, el motivo de la ampliación, cual es la constancia que autoriza la ampliación y el número de esa constancia, estos datos son requeridos por el sistema para poder generar la ampliación.

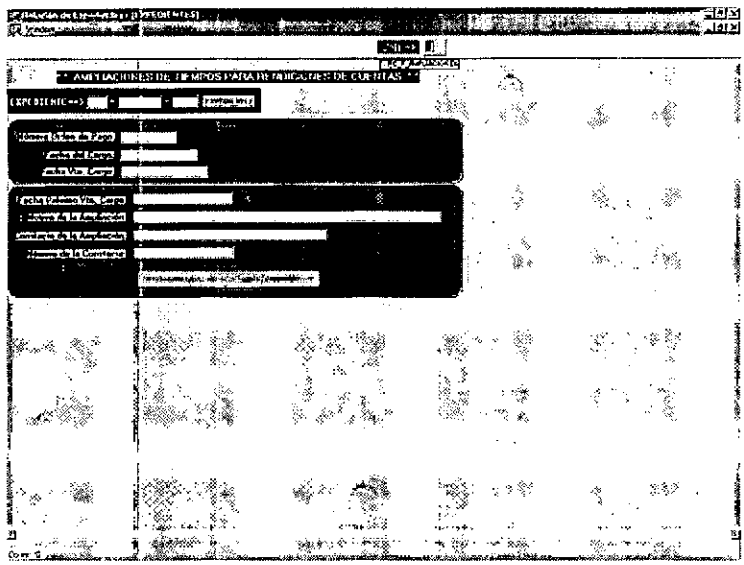
A continuación mostramos el código fuente que esta relacionado al "Trigger" WHEN-BUTTON-PRESSED del botón "GENERAR AMPLIACION", o sea, este código es ejecutado cuando se presiona el botón.

```
DECLARE
  al_id          Alert;
  al_button      Number;
  ERRCODE        NUMBER ;
  err_txt        VARCHAR2(200);
  a_value_chosen BOOLEAN;
  SEC_AMPL       NUMBER;
  FECHA_ACTUAL   DATE;
begin

   if  (:B_AMPLIACION.FECHA_VTO_AMPL IS NULL)  then
     ERR_TXT := '** LA FECHA DEL PROXIMO VENCIMIENTO NO PUEDE SER NULA **';
     al_id := Find_Alert('ALERTA_ACEPTAR');
     Set_Alert_Property(al_id, alert_message_text, err_txt );
     al_button := Show_Alert( al_id );
     raise form_trigger_failure;
   END IF;

   if  (:B_AMPLIACION.MOTIVO_AMPL IS NULL)  then
     ERR_TXT := '** EL MOTIVO DE LA AMPLIACION NO PUEDE SER NULO **';
     al_id := Find_Alert('ALERTA_ACEPTAR');
     Set_Alert_Property(al_id, alert_message_text, err_txt );
     al_button := Show_Alert( al_id );
     raise form_trigger_failure;
   END IF;

   SELECT S_CO_RC_AMPLIACIONES.NEXTVAL
     INTO SEC_AMPL
     FROM dual;

   SELECT TRUNC(SYSDATE) INTO FECHA_ACTUAL FROM DUAL;

   INSERT INTO CO_RC_AMPLIACIONES (CO_R_OR_NRO_SECUENCIA,
                                   ID_AMPLIACION,
                                     FECHA_AMPLIACION,
                                     FECHA_VEN_ANT,
                                     MOTIVO_AMPLIACION,
                                   CONSTANCIA_AMPL,
                                   NRO_CONST_AMPL)
   VALUES (:B_AMPLIACION.NRO_SEC_OP,
           SEC_AMPL,
           FECHA_ACTUAL,
           :B_AMPLIACION.FECHA_VTO_CARGO,
           :B_AMPLIACION.MOTIVO_AMPL,
           :B_AMPLIACION.CONSTACIA_AMPL,
           :B_AMPLIACION.NRO_CONST_AMPL);

  UPDATE CO_REGISTRO_ORDENES REG
    SET REG.FECHA_VTO_CARGO = :B_AMPLIACION.FECHA_VTO_AMPL
```

```
  WHERE REG.NRO_SECUENCIA = :B_AMPLIACION.NRO_SEC_OP;

  COMMIT;
  GO_ITEM('B_CONTROL_EXP.BUT_EXPEDIENTES');
  CLEAR_BLOCK;
  GO_ITEM('B_CONTROL1.W_FILTRO_EXPEDIENTE');
  CLEAR_BLOCK;
  GO_ITEM('B_AMPLIACION.FECHA_VTO_AMPL');
  CLEAR_BLOCK;
  SET_ITEM_PROPERTY('B_AMPLIACION.GENERA_AMPLIACION',ENABLED, PROPERTY_FALSE);
  GO_ITEM('B_CONTROL_EXP.BUT_EXPEDIENTES');
Synchronize;
     ERR_TXT := '** AMPLIACION GENERADA **';
     al_id := Find_Alert('ALERTA_ACEPTAR');
     Set_Alert_Property(al_id, alert_message_text, err_txt );
     al_button := Show_Alert( al_id );

end;
```

7.3.1.8.2 Ejecución del Módulo

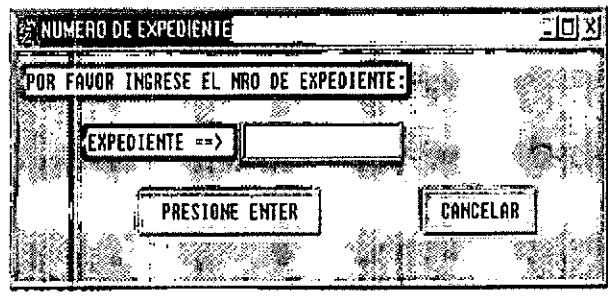
En la figura 7.48 se muestra al módulo en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior.



(Figura 7.48)

En la figura 7.48 se muestra el módulo en ejecución de las ampliaciones de plazos de las rendiciones, como se puede observar, el botón GENERAR AMPLIACION se encuentra deshabilitado, este se habilita después de ser seleccionado el expediente al cual se le quiere generar una ampliación.

Para poder seleccionar hay que hacer click en el botón EXPEDINTE luego se abrirá una ventana para poder ingresar el número del expediente requerido.



(Figura 7.49)

La figura 7.49 muestra la ventada donde se ingresa el número del expediente requerido, luego al presionar el botón PRESIONAR ENTER se ejecuta el código que mostraremos a continuación, este código es el encargado de recuperar los datos de la orden de pago relacionada con el expediente ingresado.

```
DECLARE
  al_id          Alert;
  al_button      Number;
  ERRCODE        NUMBER ;
  err_txt        VARCHAR2(200);
  a_value_chosen BOOLEAN;
  cargos         number;
BEGIN

if  :B_CONTROL1.W_FILTRO_EXPEDIENTE IS NULL  then
     ERR_TXT := '** DEBE INGRESAR EL NUMERO DE EXPEDIENTE **';
     al_id := Find_Alert('ALERTA_ACEPTAR');
     Set_Alert_Property(al_id, alert_message_text, err_txt );
     al_button := Show_Alert( al_id );
     raise form_trigger_failure;
END IF;


   a_value_chosen := Show_Lov('LOV_EXPEDIENTE');
   IF a_value_chosen THEN

      select count(1)
        into cargos
        from co_registro_ordenes ro
       where ro.nro_secuencia_exp = :b_control_exp.nro_secuenciaaux
         and ro.cargo is not null;
      IF cargos > 0 THEN

        SELECT RO.NRO_SECUENCIA,
               RO.NUMERO,
               RO.FECHA_CARGO,
               RO.FECHA_VTO_CARGO
          INTO :B_AMPLIACION.NRO_SEC_OP,
               :B_AMPLIACION.NRO_OP,
               :B_AMPLIACION.FECHA_CARGO,
               :B_AMPLIACION.FECHA_VTO_CARGO
          FROM CO_REGISTRO_ORDENES RO
         WHERE RO.NRO_SECUENCIA_EXP = :b_control_exp.nro_secuenciaaux
           AND RO.CARGO is not null;

        :B_AMPLIACION.NRO_OP := NRO_ORDEN(:B_AMPLIACION.NRO_OP);
        SET_ITEM_PROPERTY('B_AMPLIACION.GENERA_AMPLIACION',ENABLED, PROPERTY_TRUE);
        GO_BLOCK('B_AMPLIACION');
        Synchronize;
      ELSE
          ERR_TXT := '** EL EXPEDIENTE SELECCIONADO NO TIENE CARGOS**';
          al_id := Find_Alert('ALERTA_ACEPTAR');
          Set_Alert_Property(al_id, alert_message_text, err_txt );
          al_button := Show_Alert( al_id );
          :B_CONTROL1.W_FILTRO_EXPEDIENTE := NULL;
          GO_ITEM('B_CONTROL1.W_FILTRO_EXPEDINTE');
```

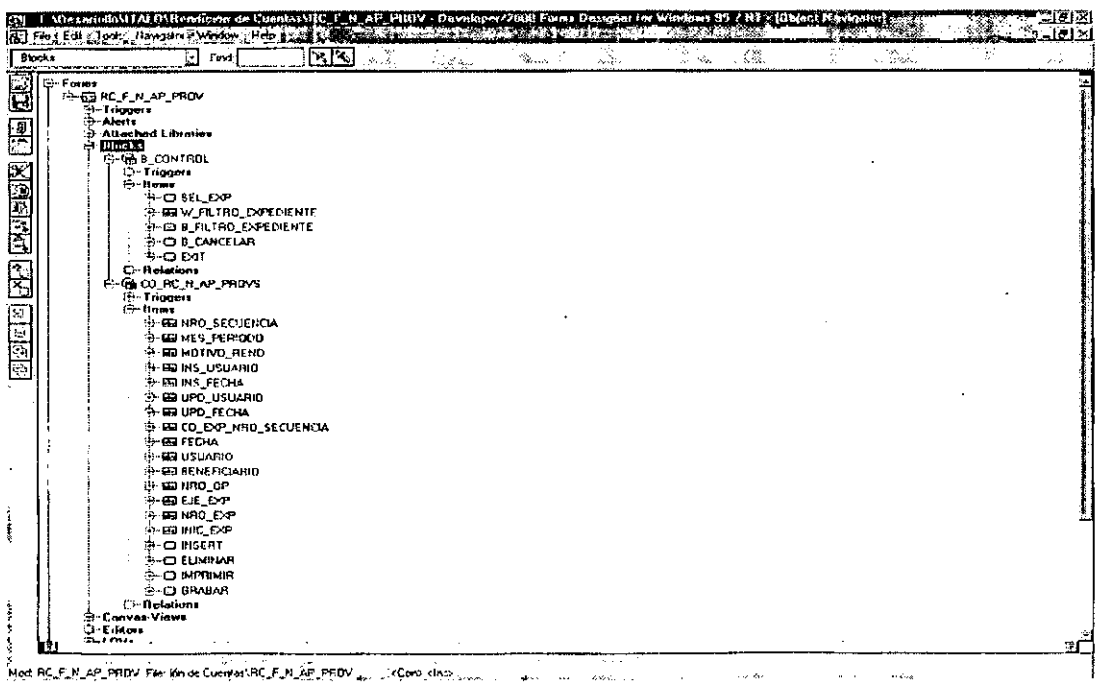
```
      END IF;
   END IF;
END;
```

7.3.1.9 Módulo de Nota de Aprobatoria Provisoria

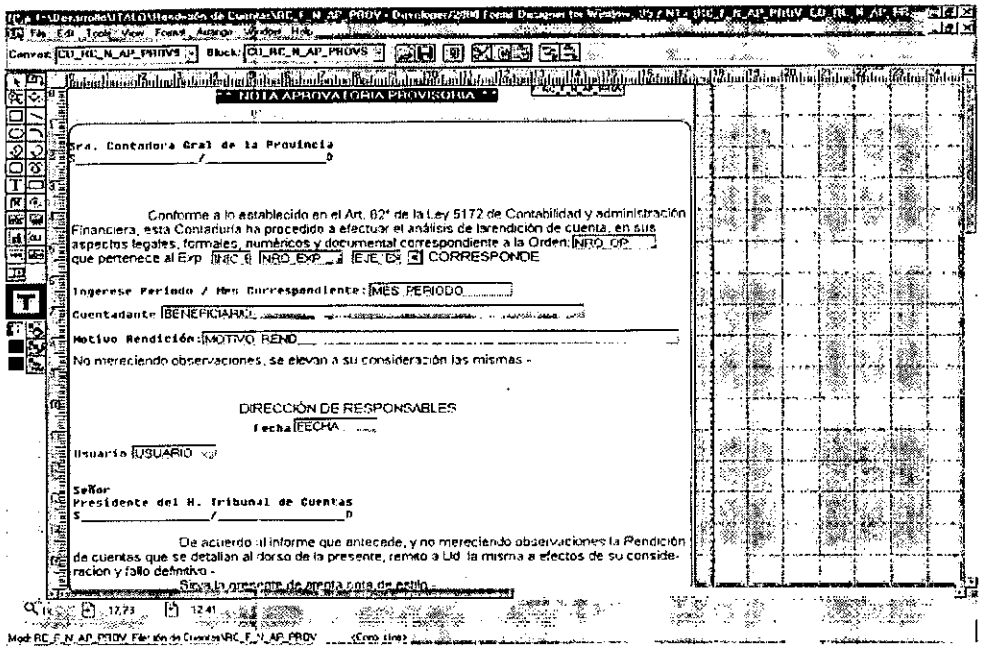
Este módulo es el que administra las notas de aprobatoria provisoria de las rendiciones de cuentas realizadas, se decide incorporar al sistema este tipo de funcionalidad para poder tener registro de todas las notas relacionadas con rendiciones de cuentas y que serán realizadas por el personal especializado de la Dirección Provincial de Contaduría.

7.3.1.9.1 Desarrollo del Módulo

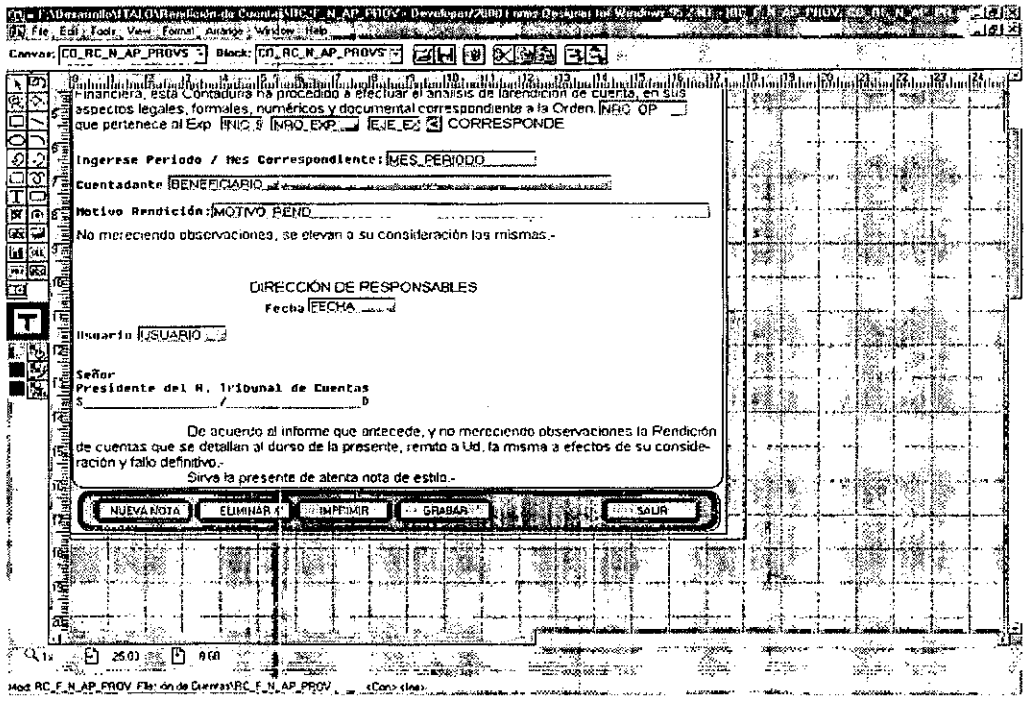
En la Figura 7.50 se muestra en el ambiente de desarrollo los bloques e ítems que forman parte del módulo, estos ítems estarán visibles por los usuarios del sistema cuando el módulo esté en ejecución.



(Figura 7.50)



(Figura 7.51)



(Figura 7.52)

En las figuras 7.51 y 7.52 se puede observar la pantalla del formulario que hace referencia a la nota de aprobatoria provisoria de una rendición de cuentas, los principales botones de este "forms" son "NUEVA NOTA", "ELIMINAR", "IMPRIMIR" y "GRABAR".

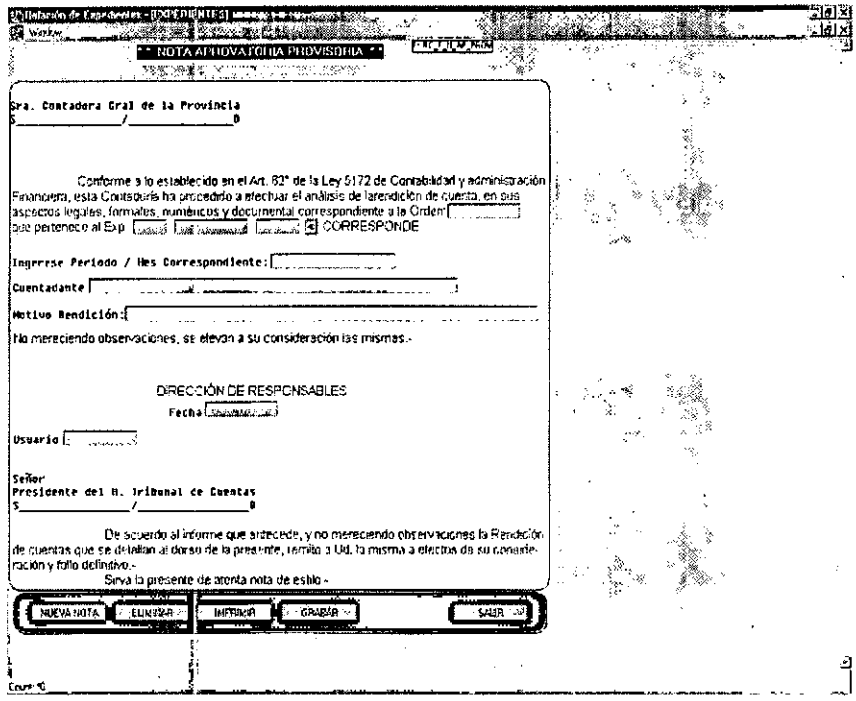
A continuación mostraremos el código fuente que se ejecuta al seleccionar un expediente y que es el encargado de recuperar los datos que no pertenecen a la tabla CO_EXPEDIENTES.

```
DECLARE
  al_id          Alert;
  al_button      Number;
  ERRCODE        NUMBER ;
  err_txt        VARCHAR2(200);
  a_value_chosen BOOLEAN;
  cantidad       number;

BEGIN

if  :B_CONTROL.W_FILTRO_EXPEDIENTE IS NULL  then
     ERR_TXT := '** DEBE INGRESAR EL NUMERO DE EXPEDIENTE **';
     al_id := Find_Alert('ALERTA_ACEPTAR');
     Set_Alert_Property(al_id, alert_message_text, err_txt );
     al_button := Show_Alert( al_id );
     raise form_trigger_failure;
END IF;

   a_value_chosen := Show_Lov('LOV_EXPEDIENTE');

   IF a_value_chosen THEN
      SELECT COUNT(1)
        INTO cantidad
        FROM CO_EXPEDIENTES E
       WHERE E.NRO_SECUENCIA_EXP_RENDIDO = :CO_RC_N_AP_PROVS.CO_EXP_NRO_SECUENCIA;

      IF cantidad = 0 THEN
        ERR_TXT := '** EL EXPEDIENTE SELECCIONADO NO ES UN EXPEDIENTE SUJETO A RENDICIÓN **';
        al_id := Find_Alert('ALERTA_ACEPTAR');
        Set_Alert_Property(al_id, alert_message_text, err_txt );
        al_button := Show_Alert( al_id );
        raise form_trigger_failure;
      ELSE
         SELECT RO.NUMERO
           INTO :CO_RC_N_AP_PROVS.NRO_OP
           FROM CO_REGISTRO_ORDENES RO
          WHERE RO.NRO_SECUENCIA_EXP = :CO_RC_N_AP_PROVS.CO_EXP_NRO_SECUENCIA
            AND RO.TIPO = 'OP'
            AND RO.ESTADO IS NULL;

        SELECT B.RAZON_SOCIAL
          INTO :CO_RC_N_AP_PROVS.BENEFICIARIO
          FROM CO_REGISTRO_ORDENES RO, CO_BENEFICIARIOS B
         WHERE RO.NRO_SECUENCIA_EXP = :CO_RC_N_AP_PROVS.CO_EXP_NRO_SECUENCIA
           AND RO.TIPO = 'OP'
           AND RO.ESTADO IS NULL
           AND B.ID = RO.ID_BENEFICIARIO;
      END IF;
      GO_ITEM('CO_RC_N_AP_PROVS.MOTIVO_REND');
      Synchronize;

   END IF;
END;
```

7.3.1.9.2 Ejecución del Módulo

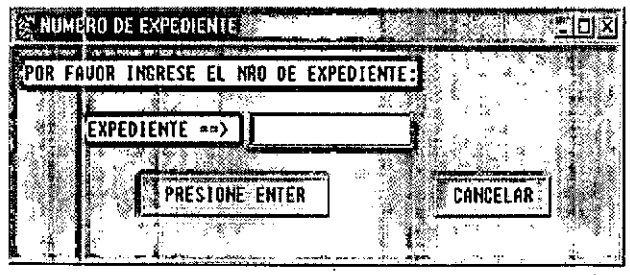
En la figura 7.53 se muestra al módulo en ejecución, donde se podrá apreciar el comportamiento de todo lo mostrado en el punto anterior, en este módulo el usuario solo podra seleccionar el expedinte con el cual se realizo el pago, con esta selección el sistema recuperará el número de la orden de pago con la cual

se hizo efectivo la entrega de dinero y también recuperará la razón social del beneficiario al cual se le hizo entrega del dinero.



(Figura 7.53)

Para poder realizar una nota de aprobatoria provisoria lo primero que tenemos que hacer es seleccionar un expediente mediante el botón ⊡ es el que abre una ventana donde se describe el número del expediente con el cual se generó la orden de pago que debe rendirse.



CONSEJO FEDERAL DE INVERSIONES
BIBLIOTECA
Dr Manuel
Belgrano

(Figura 7.54)

La figura 7.54 es la ventana donde se ingresa el número del expediente que mencionamos anteriormente, una vez seleccionado el expediente, si este está

sujeto a rendición, el "form" recuperará el número de orden de pago y los datos del beneficiario (cuentadante) al que se le pagó.

Por último debemos cargar el periodo correspondiente y el motivo de la rendición y quedaría la nota lista para grabarlo o imprimirla dependiendo de lo que el usuario necesite.

7.3.2 REPORTES

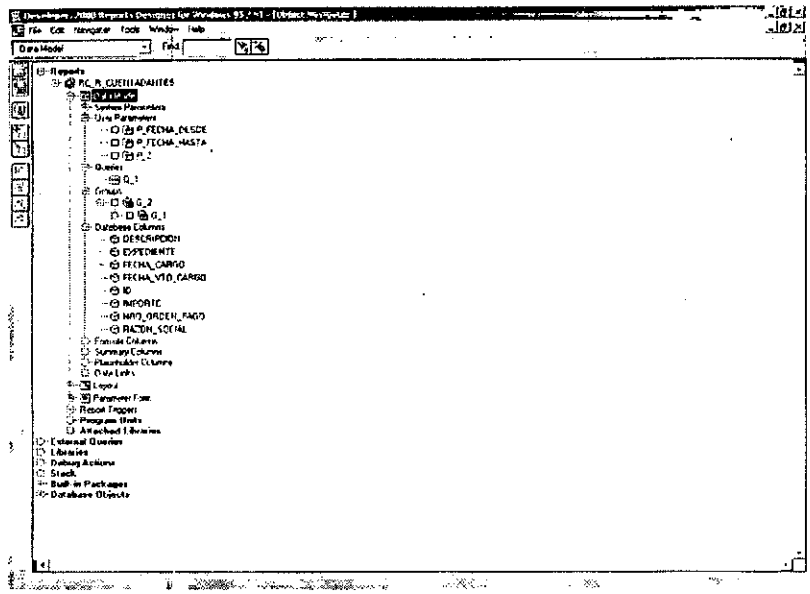
En este punto se analizamos los reportes tanto en desarrollo como en ejecución.

7.3.2.1 Reporte de Cuentadantes

Para acceder a este reporte hay que hacerlo a través del forms mostrado en el punto 7.3.1.2 de este reporte.

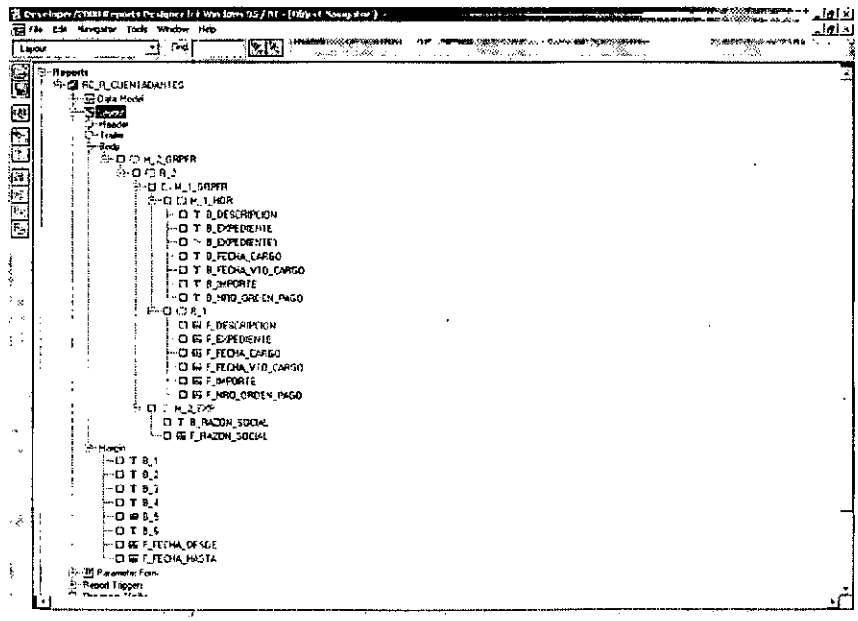
7.3.2.1.1 Desarrollo del Reporte

Los reportes constan de dos puntos fundamentales en el desarrollo, el primer punto es el *Data Model* que es donde se definen los datos que se desean modificar y el segundo punto es el *Layout* que donde se define la forma en que se va a imprimir el Reporte.



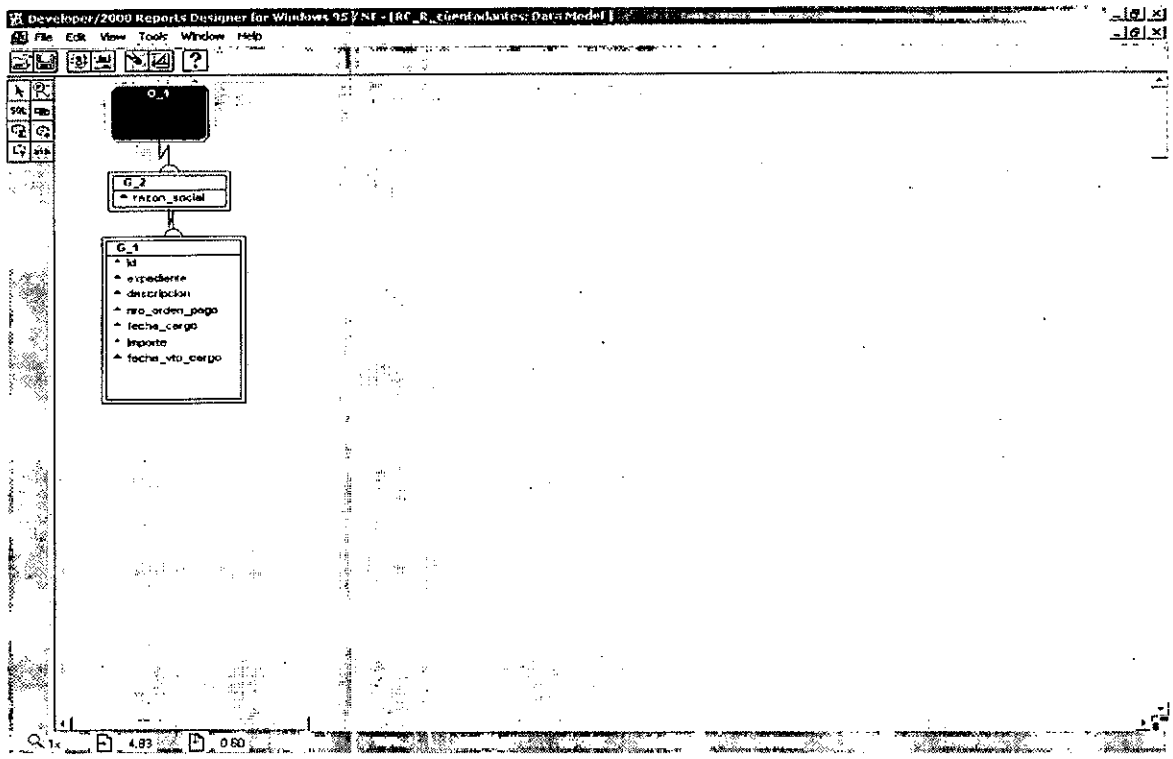
(Figura 7.55)

La figura 7.55 muestra los datos que han sido incorporados en el *Data Model,* dentro de esto se definen los parámetros que serán usados para filtrar en el SQL que recupere los datos desde la base, en *Query* se define el SQL propiamente dicho, y por último en *Database Columns* se definen los ítems a imprimir en el reporte.



(Figura 7.56)

La figura 7.56 muestra los ítems que forman parte de la parte visible del reporte, donde se encuentran los ítems de datos mas los ítems de texto que hacen la compresión del reporte.



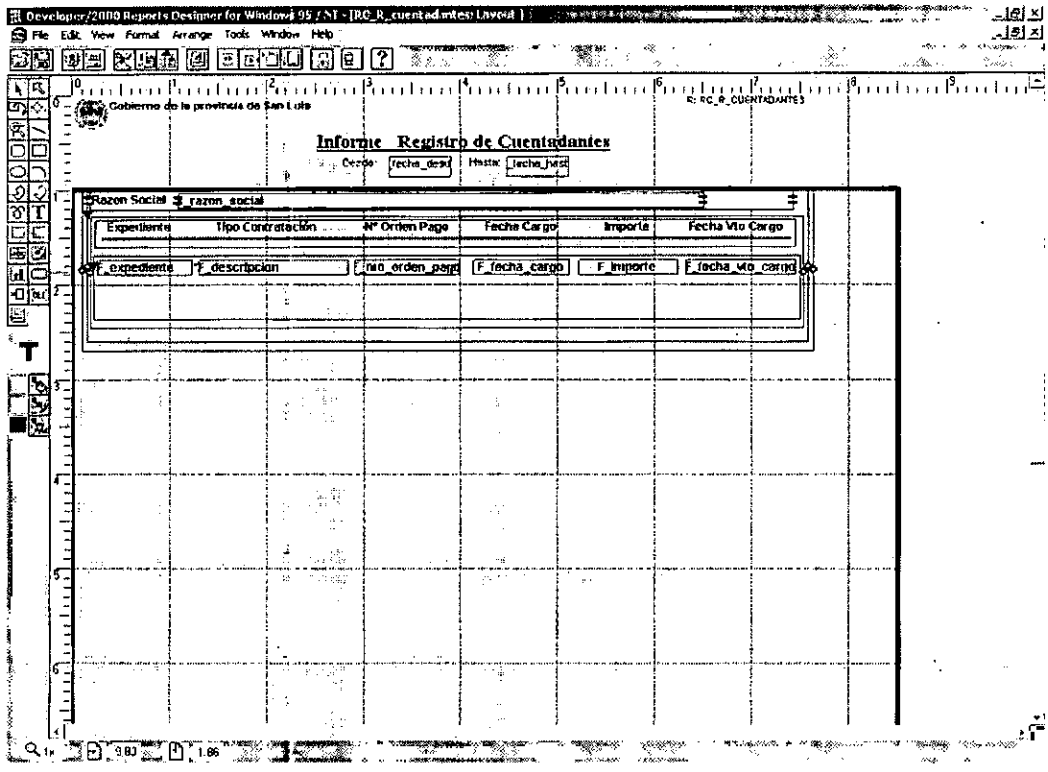
(Figura 7.57)

La figura 7.57 muestra en forma esquemática los ítems que forman parte de la consulta principal. A continuación se muestra el query que de este reporte.

```
SELECT
  b.razon_social,
  b.id,
  e.inic ||' - '|| e.numero ||' - '|| e.ejercicio expediente,
  tc.descripcion,
  ror.numero nro_orden_pago,
  rod.importe,
  ror.fecha_cargo,
  ror.fecha_vto_cargo
FROM
  co_registro_orden_detalles rod,
  co_beneficiarios b,
  co_expedientes e,
  ngr_tipos_contr tc,
  (SELECT o.numero,
          o.fecha_cargo,
          o.fecha_vto_cargo,
          o.nro_secuencia_exp,
          o.id_beneficiario,
          o.nro_secuencia
        FROM co_registro_ordenes o
    WHERE o.cargo IS NOT NULL
          AND o.tipo = 'OP'
          AND (NOT EXISTS (SELECT 1
                             FROM co_rc_rendiciones rr
                            WHERE rr.co_r_or_nro_secuencia = o.nro_secuencia))
```

```
  UNION
   SELECT o.numero,
          o.fecha_cargo,
                 o.fecha_vto_cargo,
                 o.nro_secuencia_exp,
                 o.id_beneficiario,
                 o.nro_secuencia
        FROM co_registro_ordenes o,co_rc_rendiciones r
       WHERE o.cargo IS NOT NULL
         AND o.tipo = 'OP'
         AND (r.aprobada = 'NO' AND  r.co_r_or_nro_secuencia = o.nro_secuencia )
  ) ror
WHERE b.id = ror.id_beneficiario
  AND tc.id_tipo_contr = e.id_tipo_contr
  AND rod.nro_secuencia_op = ror.nro_secuencia
  AND ror.nro_secuencia_exp = e.nro_secuencia
  AND rod.id_retencion = 99
```

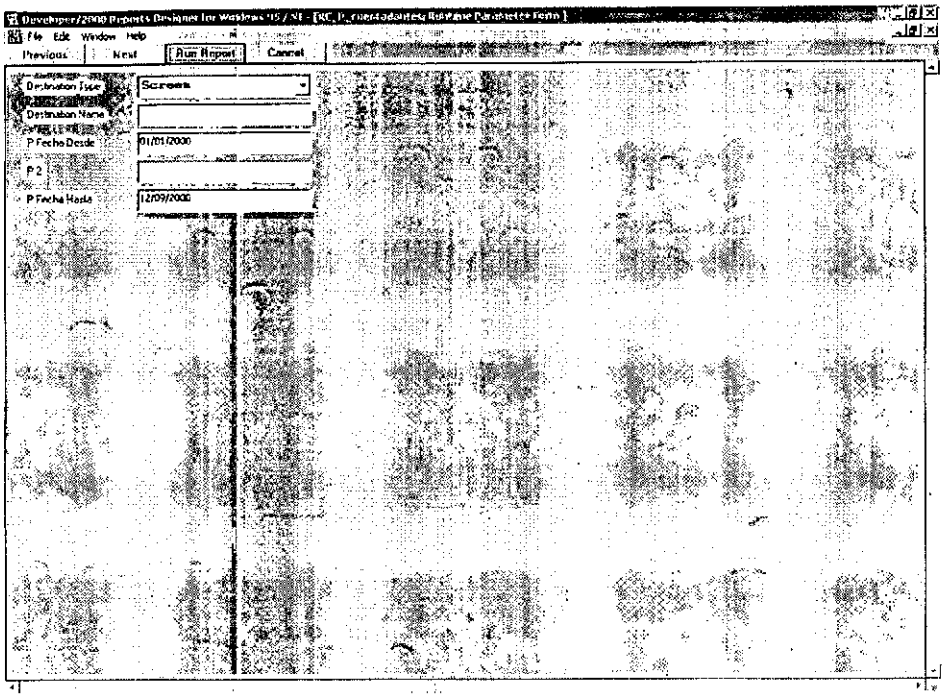
Esta es la consulta que recupera los datos desde las tablas para imprimirlos en el reporte.



(Figura 7.58)

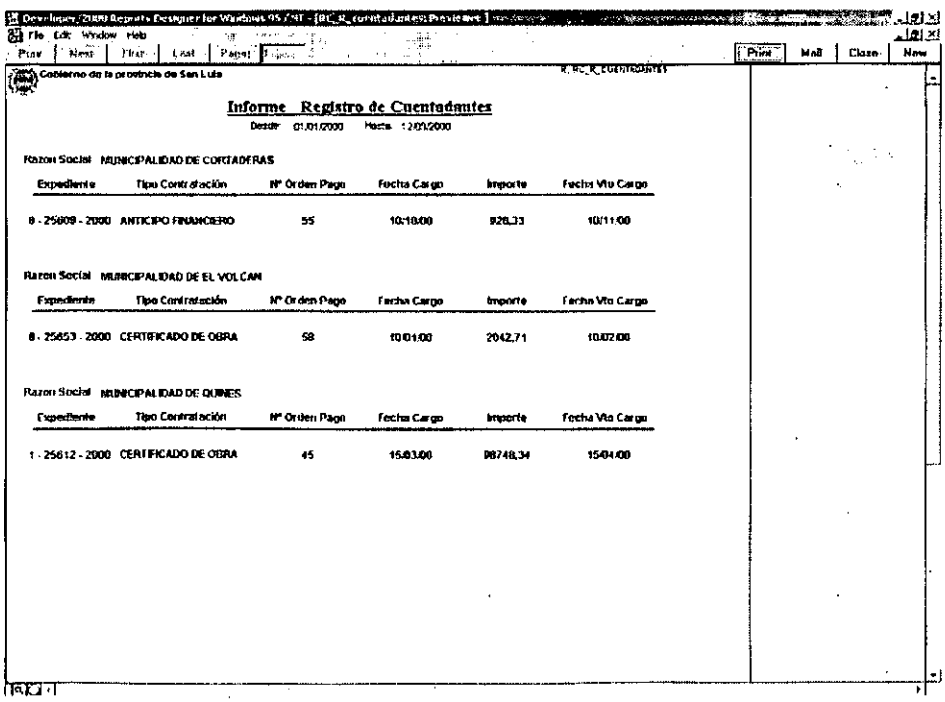
En la figura 7.58 se observa los datos que se vieron anteriormente en la figura 7.56 que muestra el Layout del reporte.

7.3.2.1.2 Ejecución del Reporte



(Figura 7.59)

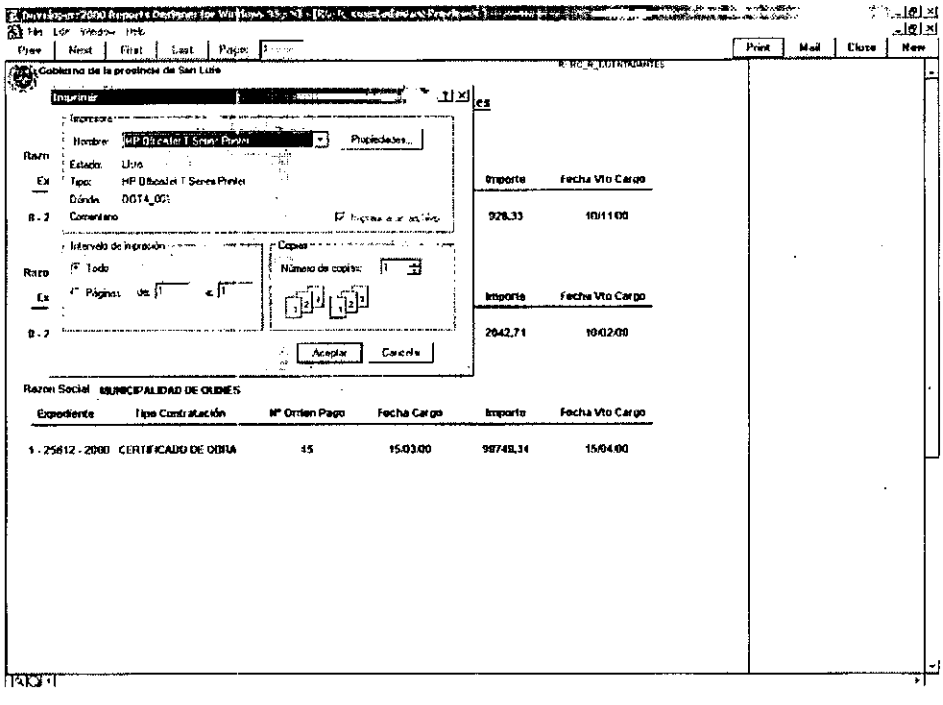
En la figura 7.59 se muestra el reporte ya ejecutándose y al detectar que tiene parámetros de entrada solicita su carga por medio de esta pantalla. Luego al hacer click en el botón "Run Report" se ejecuta el reporte esta ejecución se puede observar en la figura 7.60.



(Figura 7.60)

En la figura 7.60 se muestra la vista previa del reporte, en este punto el usuario puede elegir entre imprimir el reporte o simplemente cerrarlo, en el caso de querer imprimirlo debe hacer click el botón "Print", con esto se le mostrará al usuario una ventana con las propiedades de las impresoras (figura 7.61).

Para poder cerrar el reporte el usuario debe hacer click en el botón "Close".



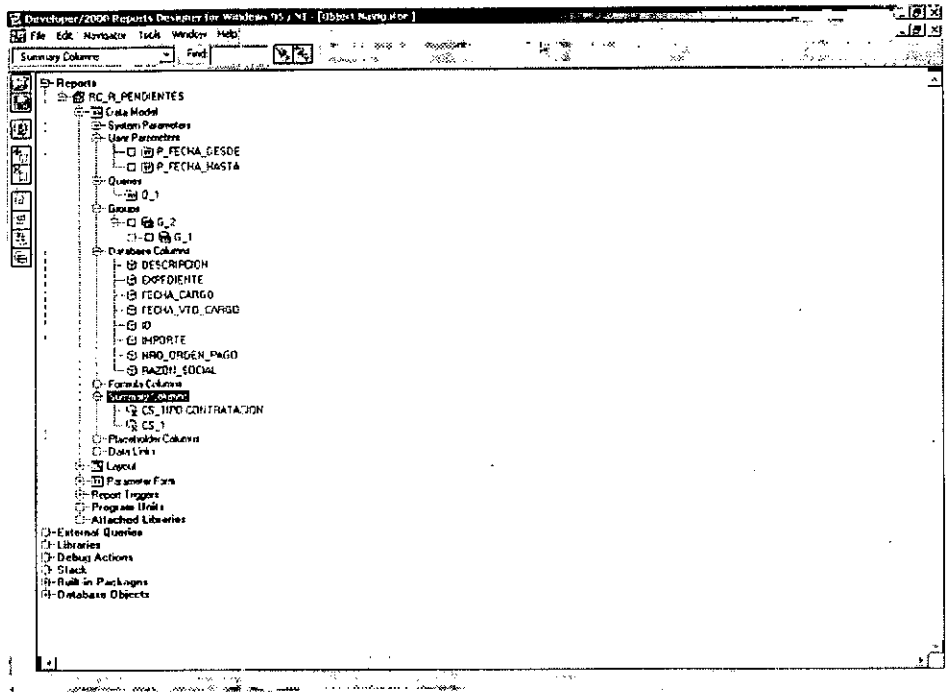
(Figura 7.61)

7.3.2.2 Reporte de Expedientes Pendientes de Rendición

Para acceder a este reporte hay que hacerlo a través del forms mostrado en el punto 7.3.1.4 de este reporte.

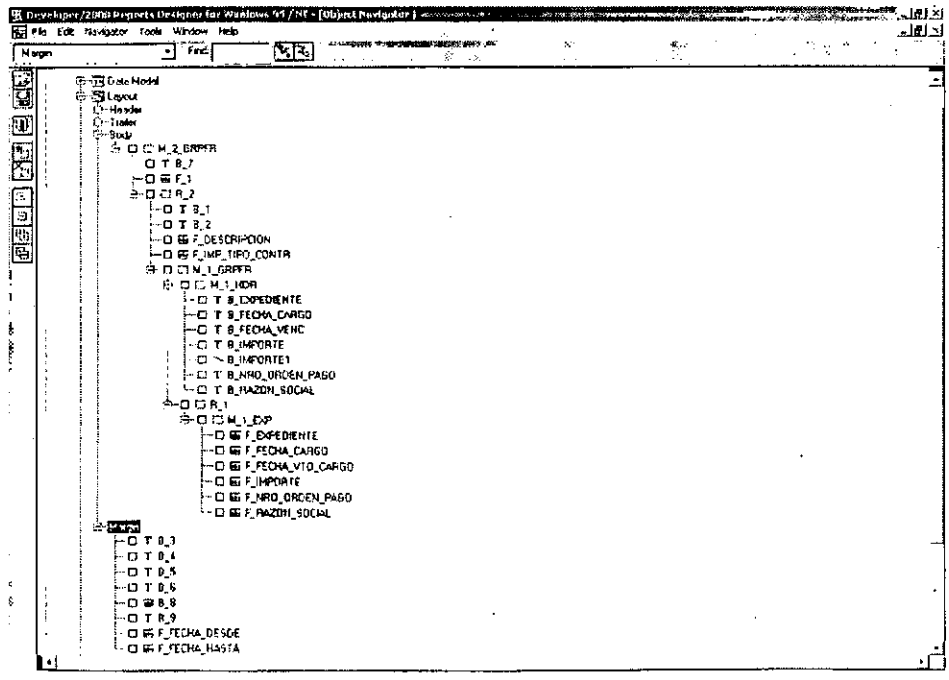
7.3.2.2.1 Desarrollo del Reporte

Los reportes constan de dos puntos fundamentales en el desarrollo, el primer punto es el *Data Model* que es donde se definen los datos que se desean modificar y el segundo punto es el *Layout* que donde se define la forma en que se va a imprimir el Reporte.



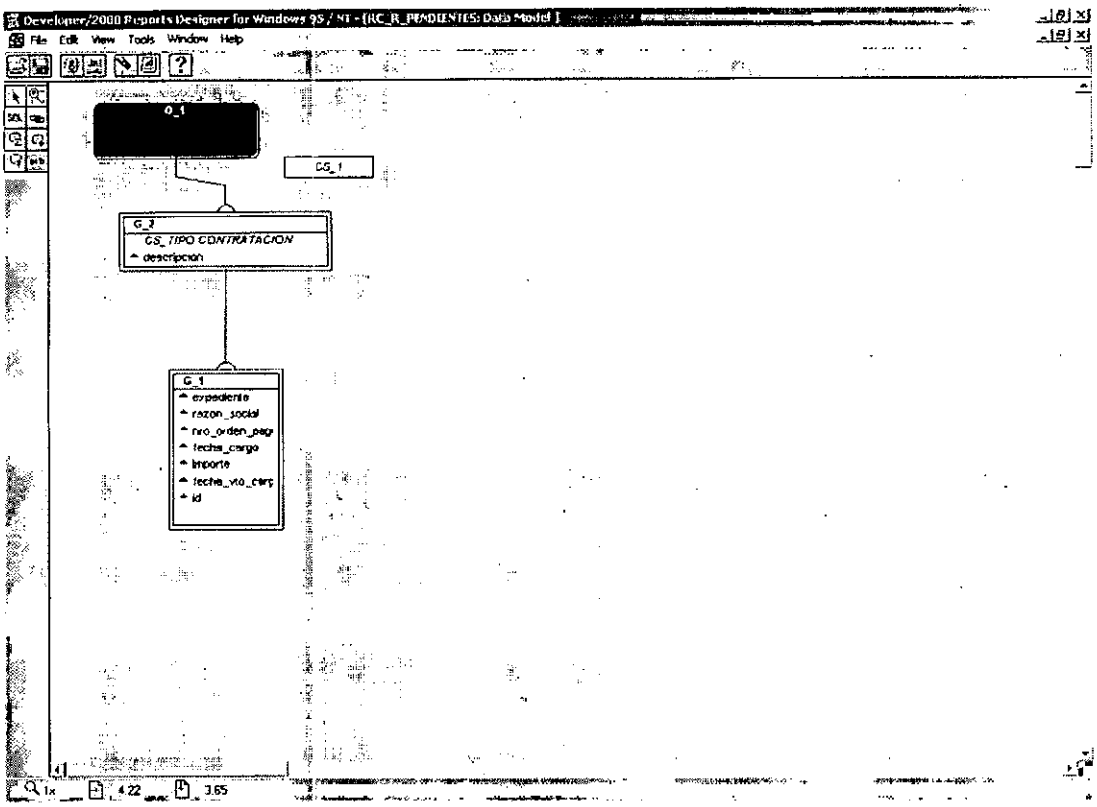
(Figura 7.62)

La figura 7.62 muestra los datos que han sido incorporados dentro del *Data Model,* en *Query* se define el SQL del reporte, y por último en *Database Columns* se definen los ítems a imprimir en el reporte.



(Figura 7.63)

La figura 7.63 muestra los ítems que forman parte de la parte visible del reporte, donde se encuentran los ítems de datos mas los ítems de texto que hacen la compresión del reporte.



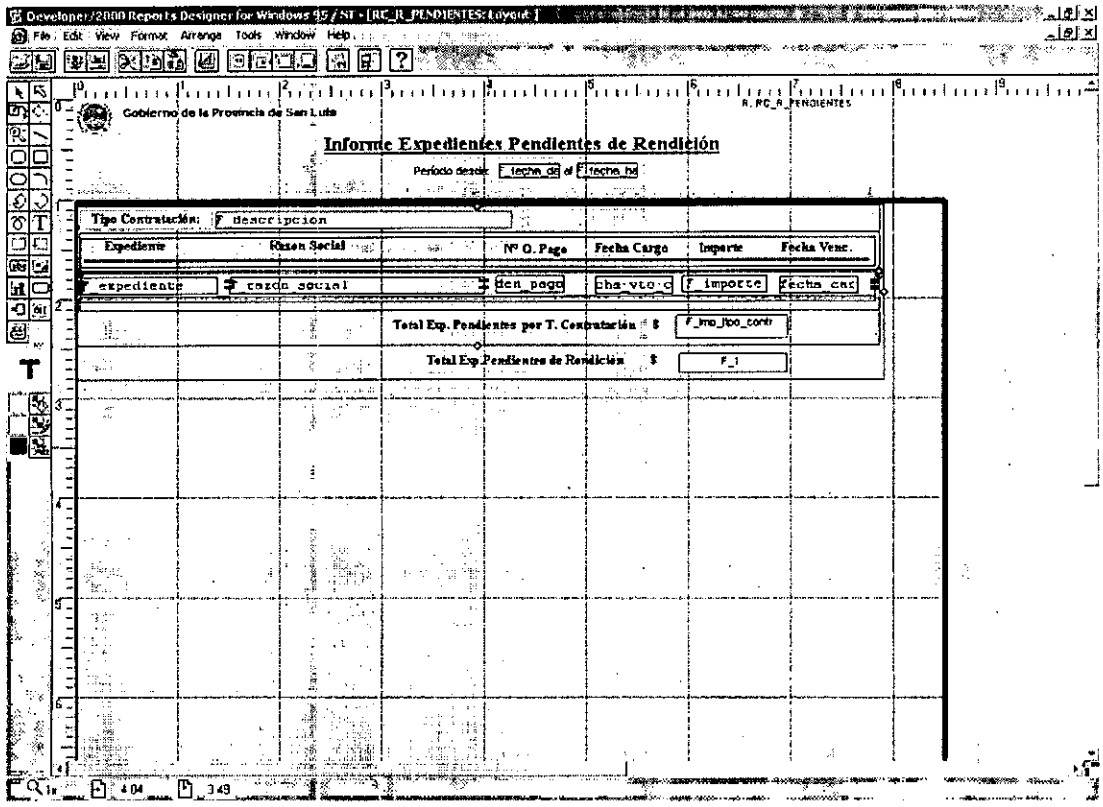
(Figura 7.64)

La figura 7.64 muestra en forma esquemática los ítems que forman parte de la consulta principal. A continuación se muestra el query que de este reporte.

```
  SELECT
  tc.descripcion,
  b.id,
  e.inic ||' - '|| e.numero ||' - '|| e.ejercicio expediente,
  b.razon_social,
  ror.numero nro_orden_pago,
  rod.importe,
  ror.fecha_cargo,
  ror.fecha_vto_cargo
FROM
  co_registro_orden_detalles rod,
  co_beneficiarios b,
  co_expedientes e,
  ngr_tipos_contr tc,
  (SELECT o.numero,
          o.fecha_cargo,
          o.fecha_vto_cargo,
          o.nro_secuencia_exp,
          o.id_beneficiario,
          o.nro_secuencia
         FROM co_registro_ordenes o
    WHERE o.cargo IS NOT NULL
          AND o.tipo = 'OP'
          AND (NOT EXISTS (SELECT 1
                           FROM co_rc_rendiciones rr
                          WHERE rr.co_r_or_nro_secuencia = o.nro_secuencia))
  UNION
```

```
   SELECT o.numero,
          o.fecha_cargo,
                  o.fecha_vto_cargo,
                  o.nro_secuencia_exp,
                  o.id_beneficiario,
                  o.nro_secuencia
            FROM co_registro_ordenes o,co_rc_rendiciones r
           WHERE o.cargo IS NOT NULL
             AND o.tipo = 'OP'
             AND (r.aprobada = 'NO' AND  r.co_r_or_nro_secuencia = o.nro_secuencia )
  ) ror
WHERE b.id = ror.id_beneficiario
  AND tc.id_tipo_contr = e.id_tipo_contr
  AND rod.nro_secuencia_op = ror.nro_secuencia
  AND ror.nro_secuencia_exp = e.nro_secuencia
  AND rod.id_retencion = 99
```

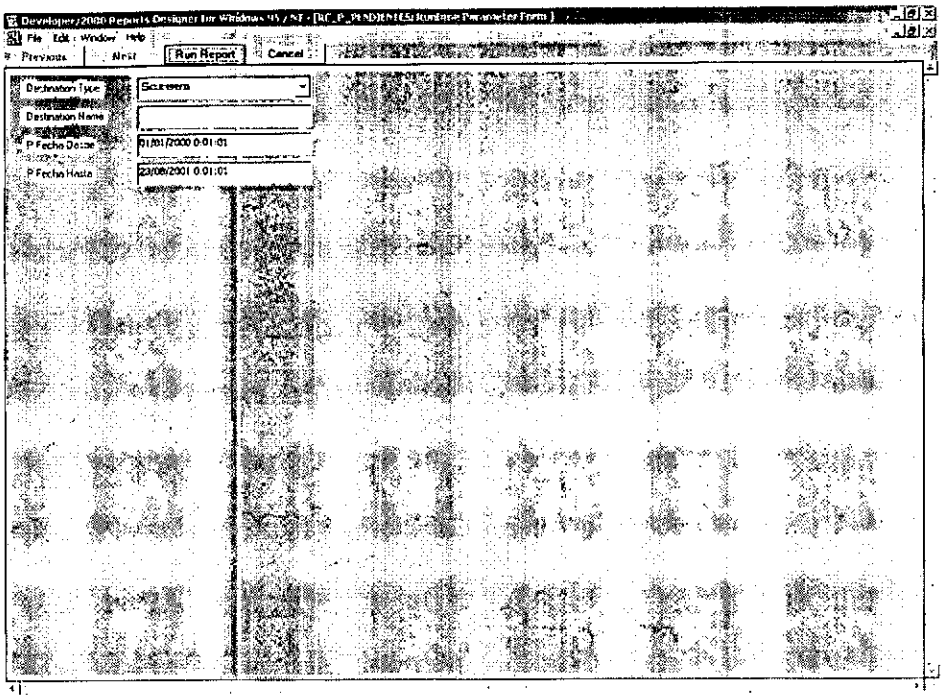
Esta es la consulta que recupera los datos desde las tablas para imprimirlos en el reporte.



(Figura 7.65)

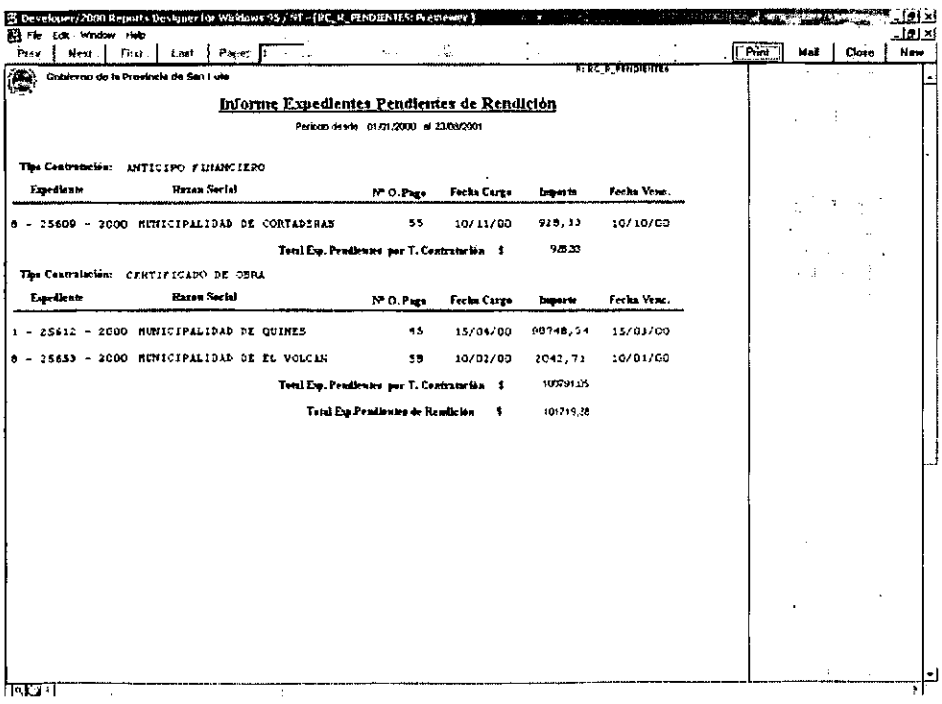
En la figura 7.65 se observa los datos que se vieron anteriormente en la figura 7.63 que muestra el Layout del reporte.

7.3.2.2.2 Ejecución del Reporte



(Figura 7.66)

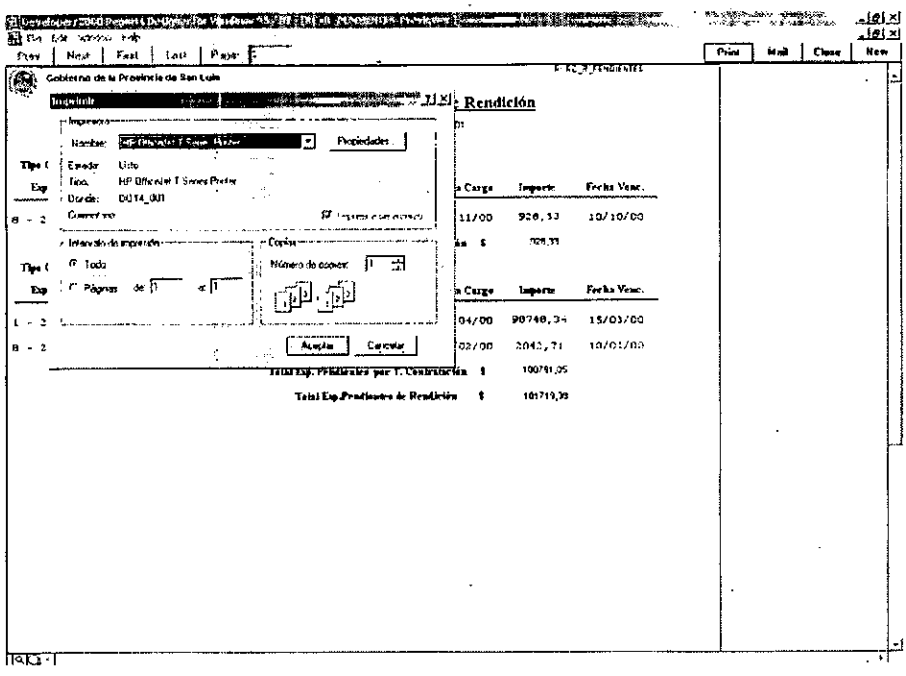
En la figura 7.66 se muestra el reporte ya ejecutándose y al detectar que tiene parámetros de entrada solicita su carga por medio de esta pantalla. Luego al hacer click en el botón "Run Report" se ejecuta el reporte esta ejecución se puede observar en la figura 7.67.



(Figura 7.67)

En la figura 7.67 se muestra la vista previa del reporte, en este punto el usuario puede elegir entre imprimir el reporte o simplemente cerrarlo, en el caso de querer imprimirlo debe hacer click el botón "Print", con esto se le mostrará al usuario una ventana con las propiedades de las impresoras (figura 7.68).

Para poder cerrar el reporte el usuario debe hacer click en el botón "Close".



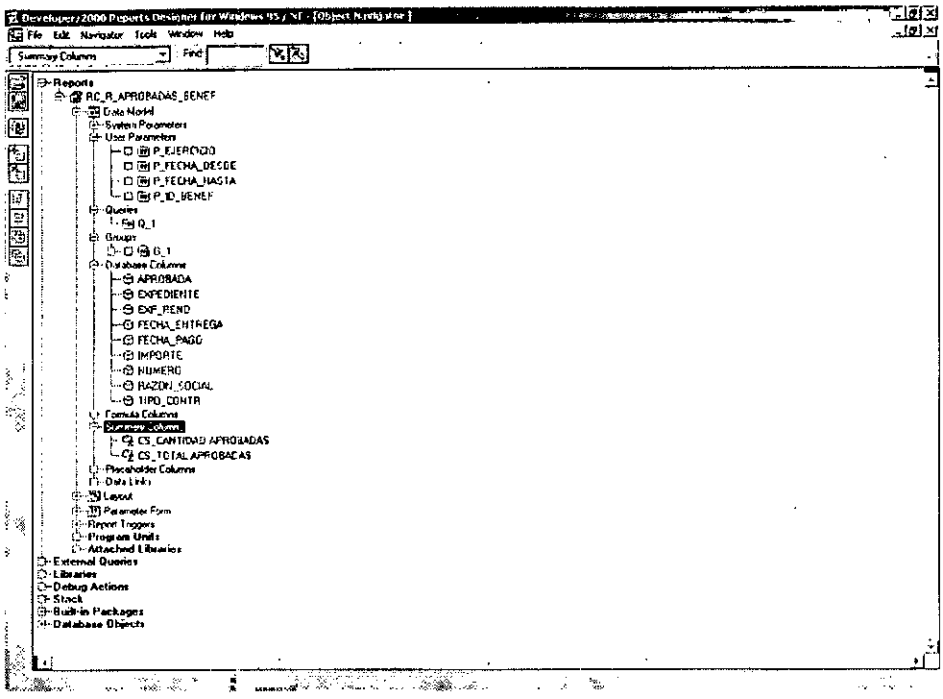
(Figura 7.68)

7.3.2.3 Reporte de Rendiciones Aprobadas

Para acceder a este reporte hay que hacerlo a través del forms mostrado en el punto 7.3.1.6 de este reporte.

7.3.2.3.1 Desarrollo del Reporte

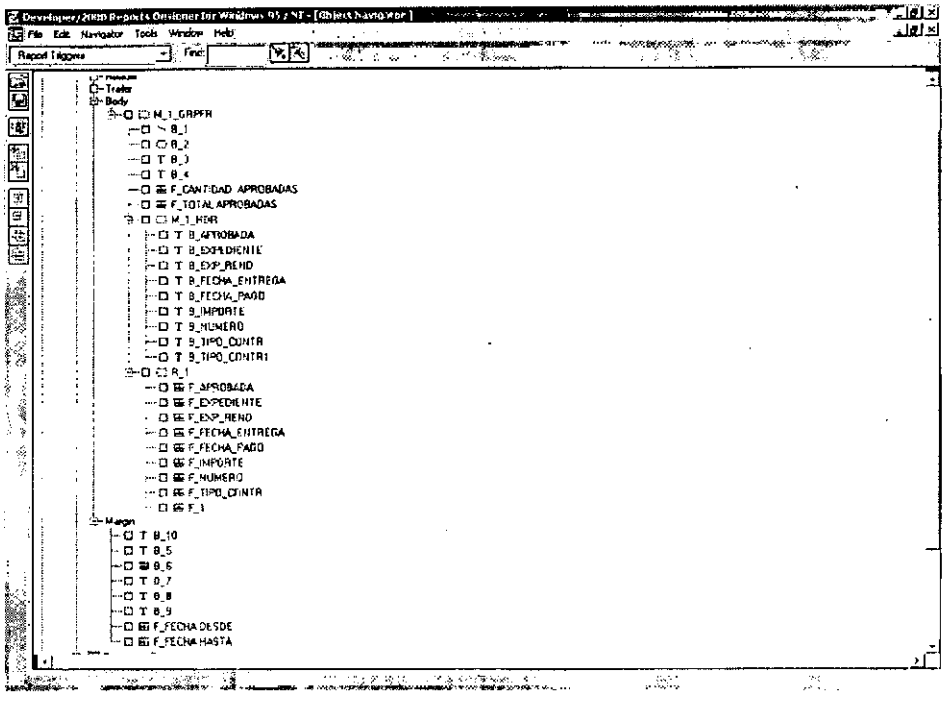
Los reportes constan de dos puntos fundamentales en el desarrollo, el primer punto es el *Data Model* que es donde se definen los datos que se desean modificar y el segundo punto es el *Layout* que donde se define la forma en que se va a imprimir el Reporte.



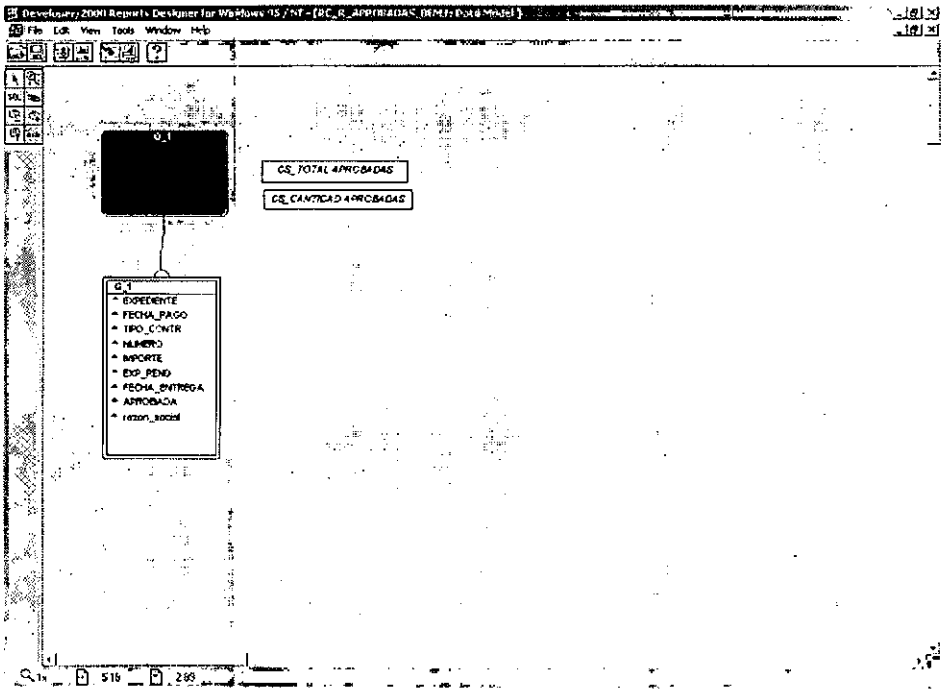
(Figura 7.69)

La figura 7.69 muestra los datos que han sido incorporados en el del *Data Model,* dentro de esto se definen los parámetros que serán usados para filtrar en el SQL que recupere los datos desde la base, en *Query* se define el SQL propiamente dicho, y por último en *Database Columns* se definen los ítems a imprimir en el reporte.

La figura 7.70 muestra los ítems que forman parte de la parte visible del reporte, donde se encuentran los ítems de datos mas los ítems de texto que hacen la compresión del reporte.



(Figura 7.70)



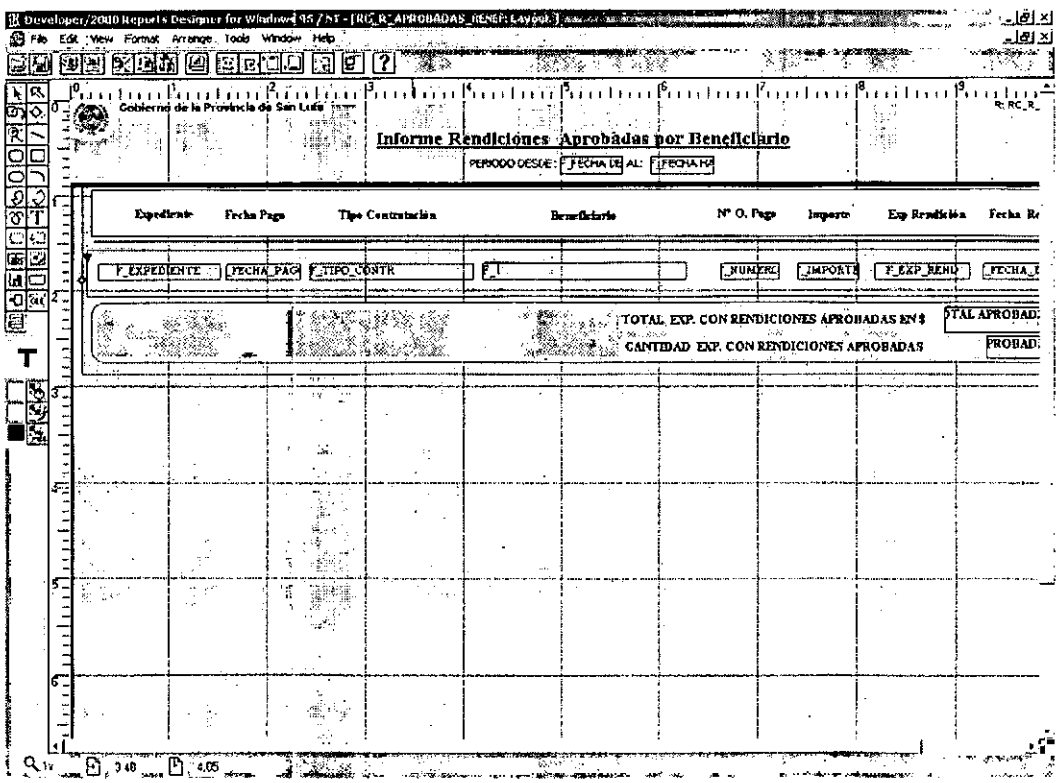
(Figura 7.71)

La figura 7.71 muestra en forma esquemática los ítems que forman parte de la consulta principal. A continuación se muestra el query que de este reporte.

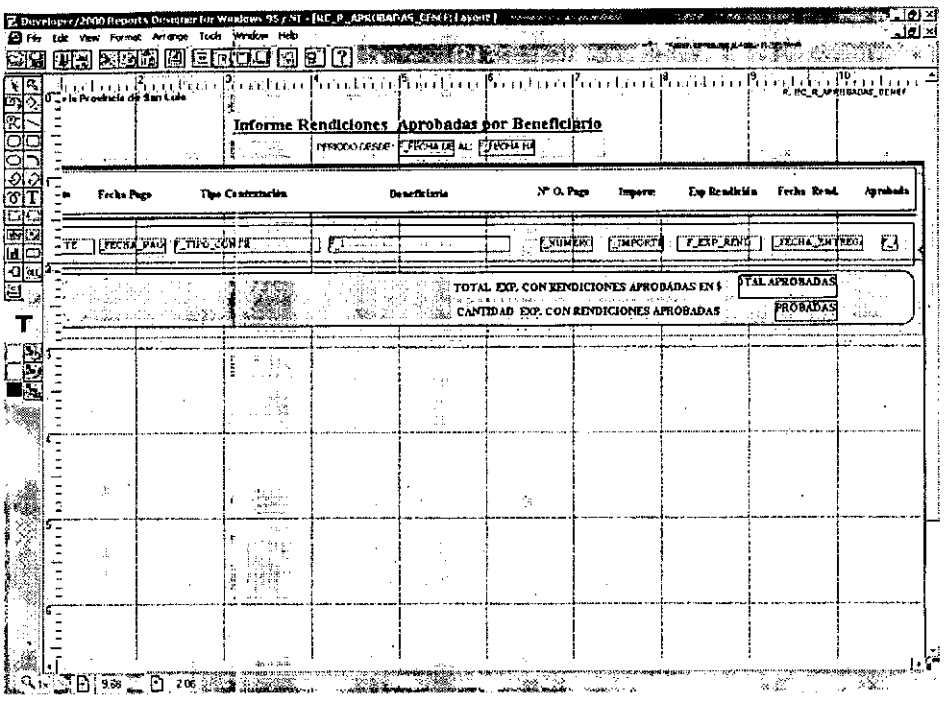
```
SELECT
  ep.inicII' - 'II ep.numero II' - 'II ep.ejercicio expediente,
  o.fecha_cargo fecha_pago,
  tc.descripcion tipo_contr,
```

```
  b.razon_social,
  o.numero,
  od.importe,
  er.inic||' - '|| er.numero ||' - '|| er.ejercicio exp_rend,
  r.fecha_entrega,
  r.aprobada
FROM
  co_rc_rendiciones r,
  co_registro_ordenes o,
  co_beneficiarios b,
  co_expedientes ep,
  ngr_tipos_contr tc,
  co_registro_orden_detalles od,
  co_expedientes er
WHERE r.aprobada = 'SI'
  and o.nro_secuencia = r.co_r_or_nro_secuencia
  and o.nro_secuencia_exp = ep.nro_secuencia
  and tc.id_tipo_contr = ep.id_tipo_contr
  and od.nro_secuencia_op = o.nro_secuencia
  and er.nro_secuencia = r.co_exp_nro_secuencia
  and od.id_retencion = 99
  and o.ejercicio = :p_ejercicio
  and r.fecha_entrega >= :p_fecha_desde
  and r.fecha_entrega <= :p_fecha_hasta
  and o.id_beneficiario = nvl(:p_id_benef,o.id_beneficiario)
  and b.id = o.id_beneficiario
```

Esta es la consulta que recupera los datos desde las tablas para imprimirlos en el reporte.



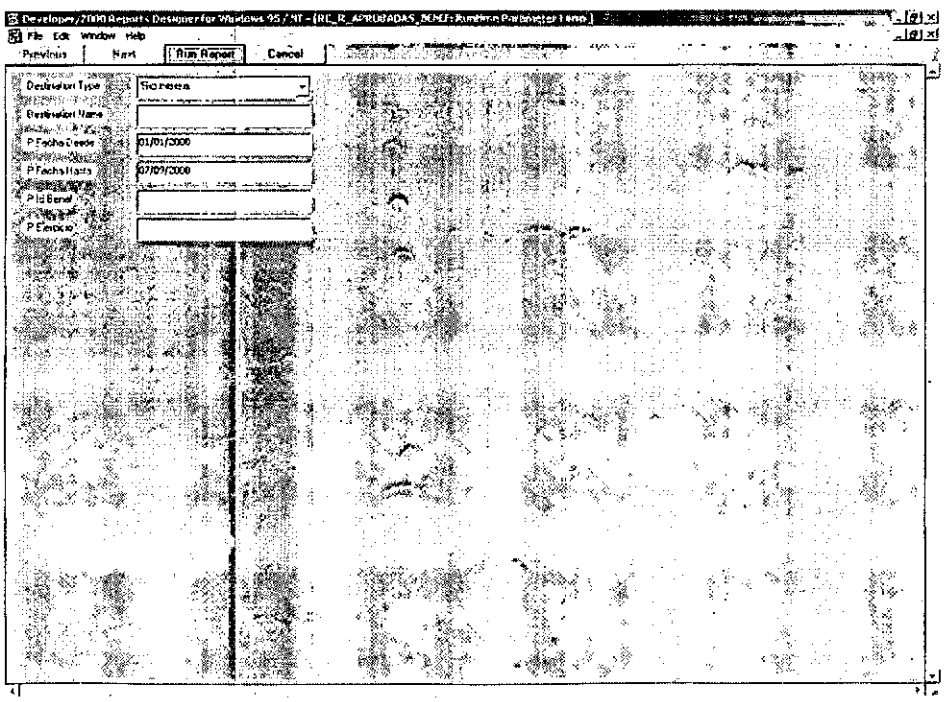
(Figura 7.72)



(Figura 7.73)

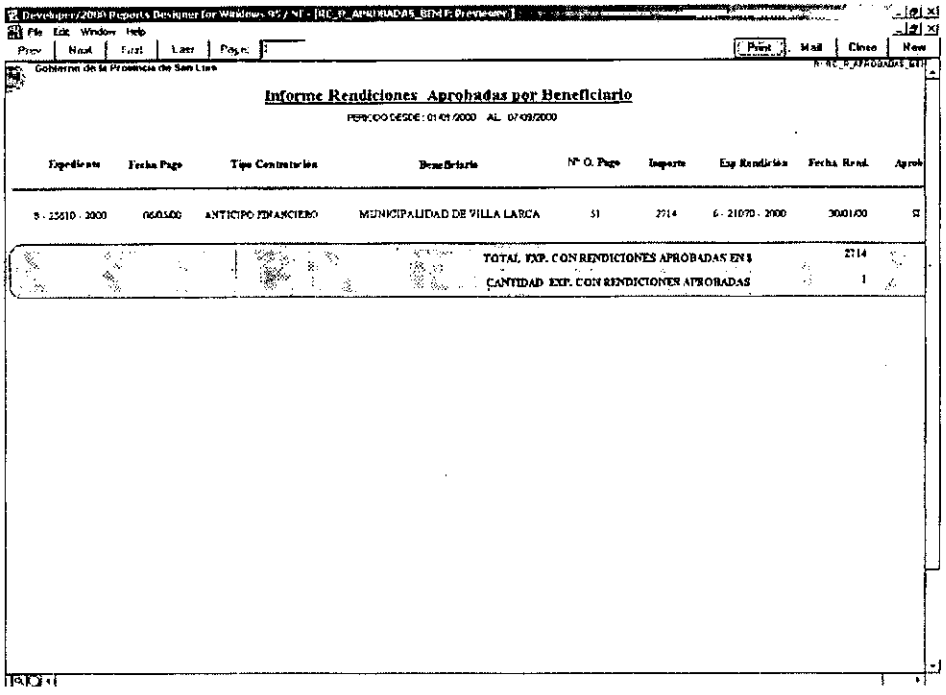
En las figuras 7.72 y 7.73 se observa los datos que se vieron anteriormente en la figura 7.70 que muestra el Layout del reporte.

7.3.2.3.2 Ejecución del Reporte



(Figura 7.74)

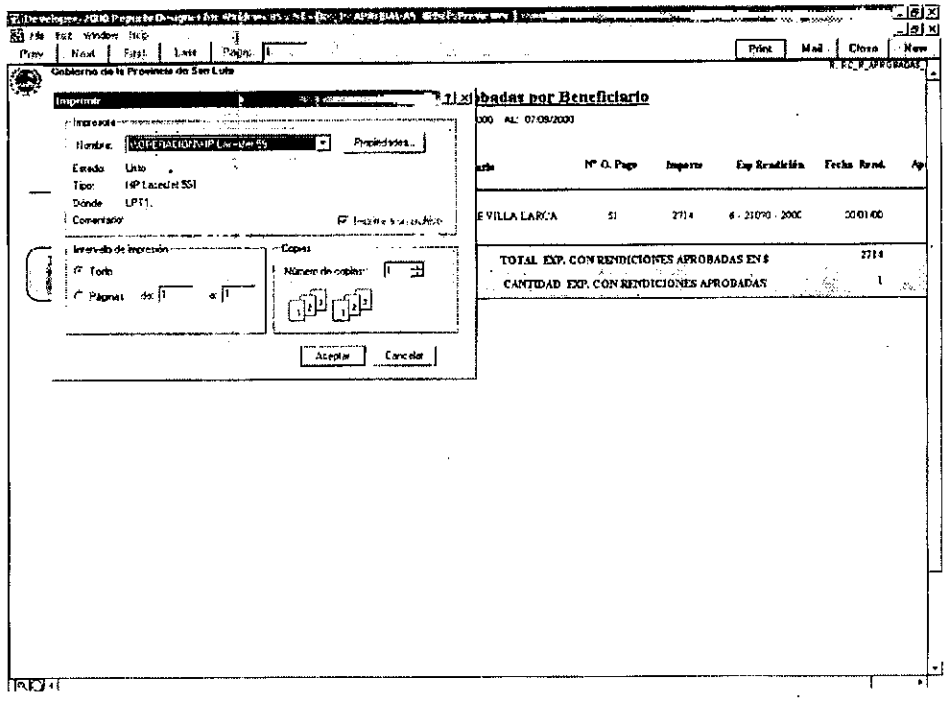
En la figura 7.74 se muestra el reporte ya ejecutándose y al detectar que tiene parámetros de entrada solicita su carga por medio de esta pantalla. Luego al hacer click en el botón "Run Report" se ejecuta el reporte esta ejecución se puede observar en la figura 7.75.



(Figura 7.75)

En la figura 7.75 se muestra la vista previa del reporte, en este punto el usuario puede elegir entre imprimir el reporte o simplemente cerrarlo, en el caso de querer imprimirlo debe hacer click el botón "Print", con esto se le mostrará al usuario una ventana con las propiedades de las impresoras (figura 7.76).

Para poder cerrar el reporte el usuario debe hacer click en el botón "Close".



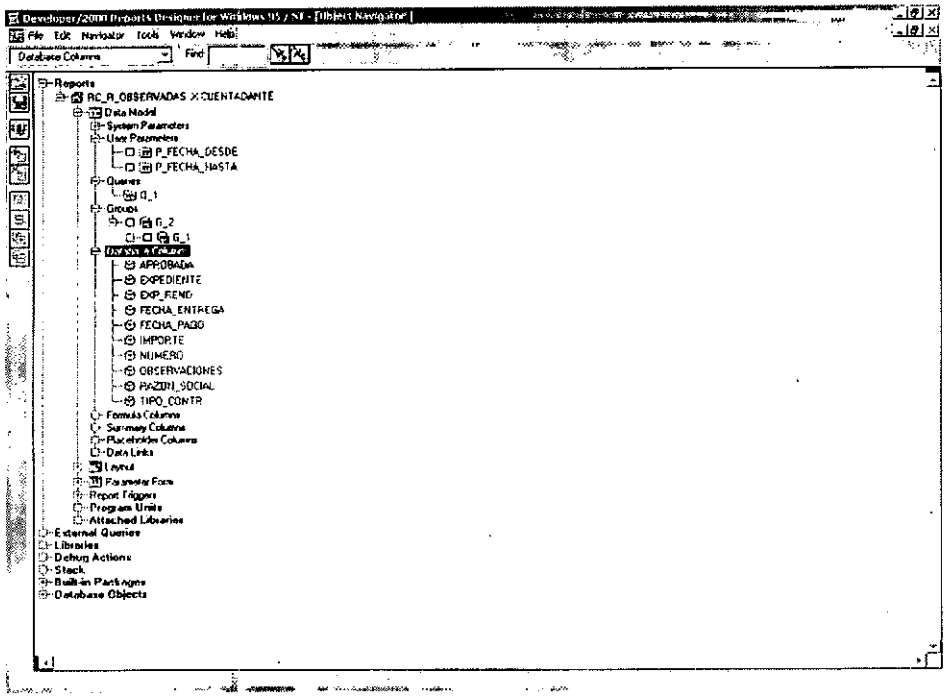
(Figura 7.76)

7.3.2.4 Reporte de Rendiciones Observadas

Para acceder a este reporte hay que hacerlo a través del forms mostrado en el punto 3.1.7 de este reporte.

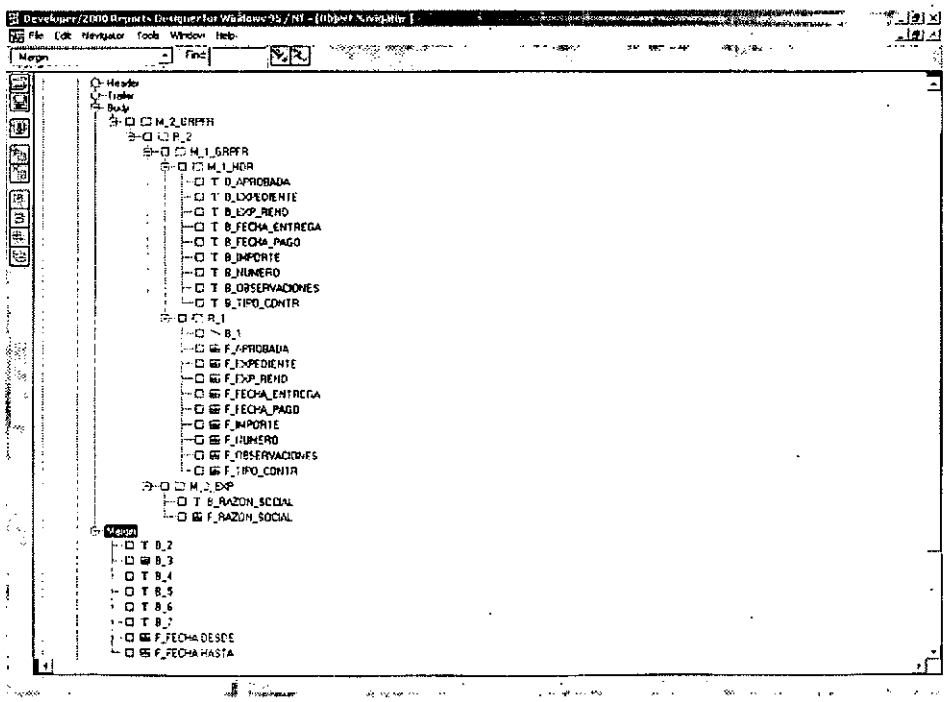
7.3.2.4.1 Desarrollo del Reporte

Los reportes constan de dos puntos fundamentales en el desarrollo, el primer punto es el *Data Model* que es donde se definen los datos que se desean modificar y el segundo punto es el *Layout* que donde se define la forma en que se va a imprimir el Reporte.



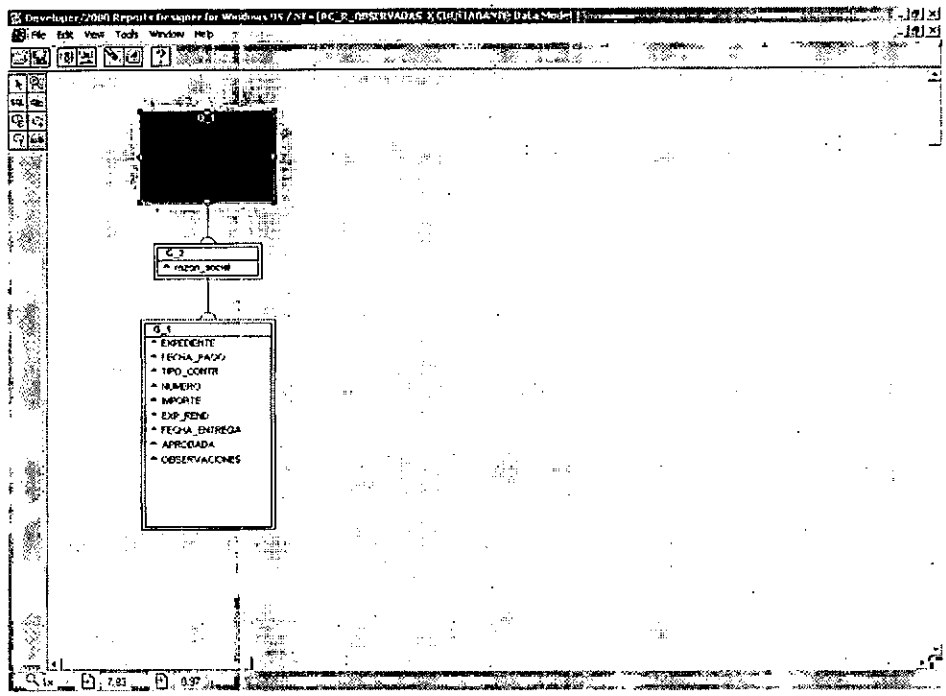
(Figura 7.77)

La figura 7.77 muestra los datos que han sido incorporados en el del *Data Model,* en *Query* se define el SQL del reporte, y por último en *Database Columns* se definen los ítems a imprimir en el reporte.



(Figura 7.78)

La figura 7.78 muestra los ítems que forman parte de la parte visible del reporte, donde se encuentran los ítems de datos mas los ítems de texto que hacen la compresión del reporte.

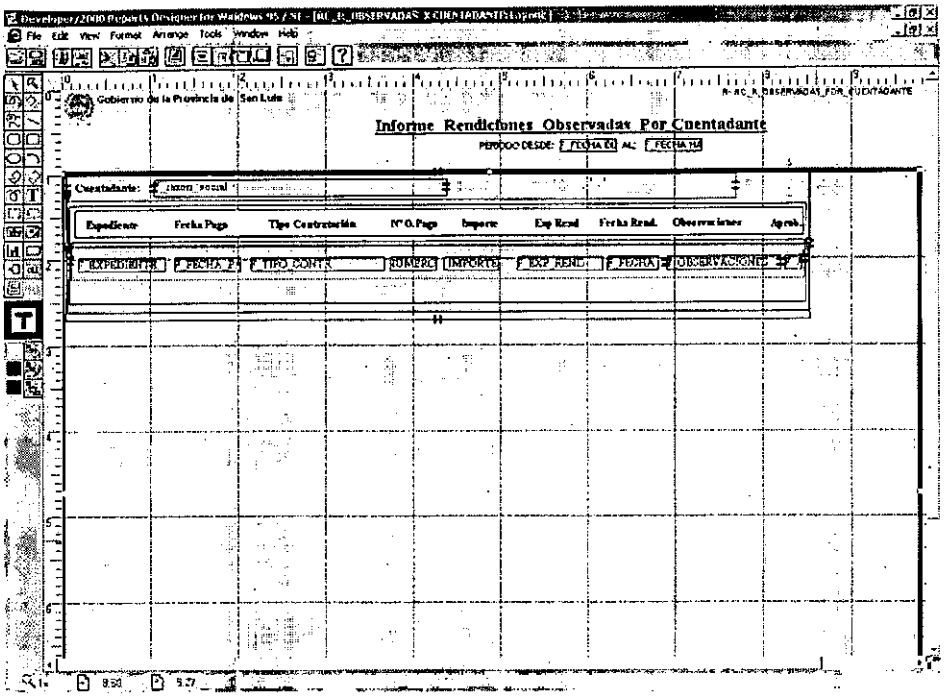


(Figura 7.79)

La figura 7.79 muestra en forma esquemática los ítems que forman parte de la consulta principal. A continuación se muestra el query que de este reporte.

```
SELECT EP.INIC||' - '|| EP.NUMERO ||' - '|| EP.EJERCICIO EXPEDIENTE,
       O.FECHA_CARGO FECHA_PAGO,
            TC.DESCRIPCION TIPO_CONTR,
            o.NUMERO,
            OD.IMPORTE,
       b.razon_social,
            ER.INIC||' - '|| ER.NUMERO ||' - '|| ER.EJERCICIO EXP_REND,
       R.FECHA_ENTREGA, R.APROBADA, R.OBSERVACIONES
from co_rc_rendiciones r,
     co_registro_ordenes o,
          CO_EXPEDIENTES EP,
     NGR_TIPOS_CONTR TC,
          co_beneficiarios b,
          CO_REGISTRO_ORDEN_DETALLES OD,
          CO_EXPEDIENTES ER
where r.OBSERVACIONES is not null
  and o.NRO_SECUENCIA = r.CO_R_OR_NRO_SECUENCIA
  AND O.NRO_SECUENCIA_EXP = EP.NRO_SECUENCIA
  AND TC.ID_TIPO_CONTR = EP.ID_TIPO_CONTR
  AND b.id = o.id_beneficiario
  AND OD.NRO_SECUENCIA_OP = O.NRO_SECUENCIA
  AND ER.NRO_SECUENCIA = R.CO_EXP_NRO_SECUENCIA
  AND OD.ID_RETENCION = 99
```

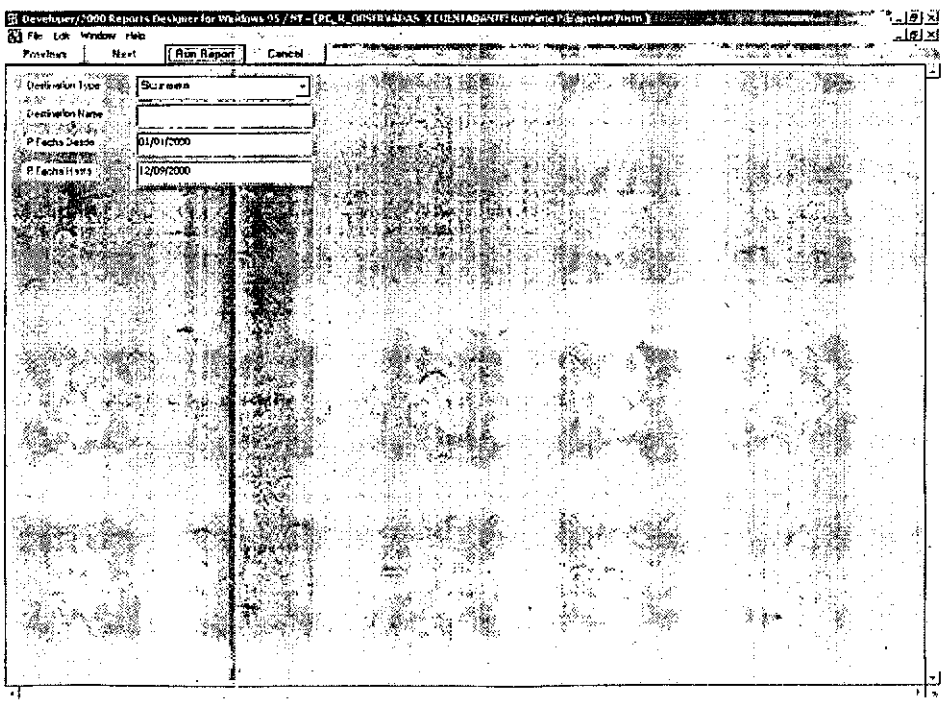
Esta es la consulta que recupera los datos desde las tablas para imprimirlos en el reporte.



(Figura 7.80)

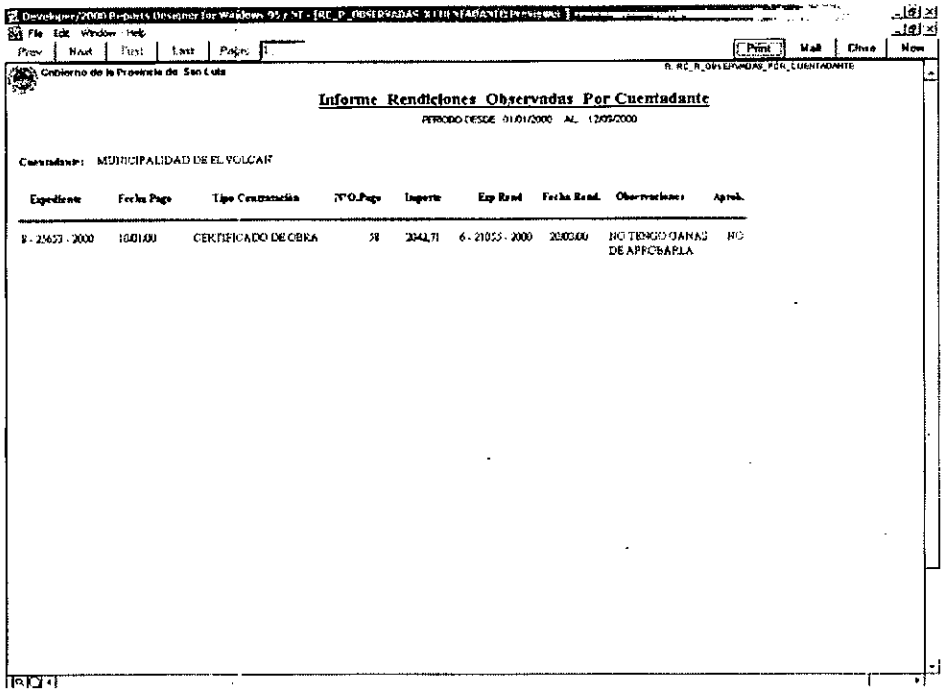
En la figura 7.80 se observa los datos que se vieron anteriormente en la figura 7.78 que muestra el Layout del reporte.

7.3.2.4.2 Ejecución del Reporte



(Figura 7.81)

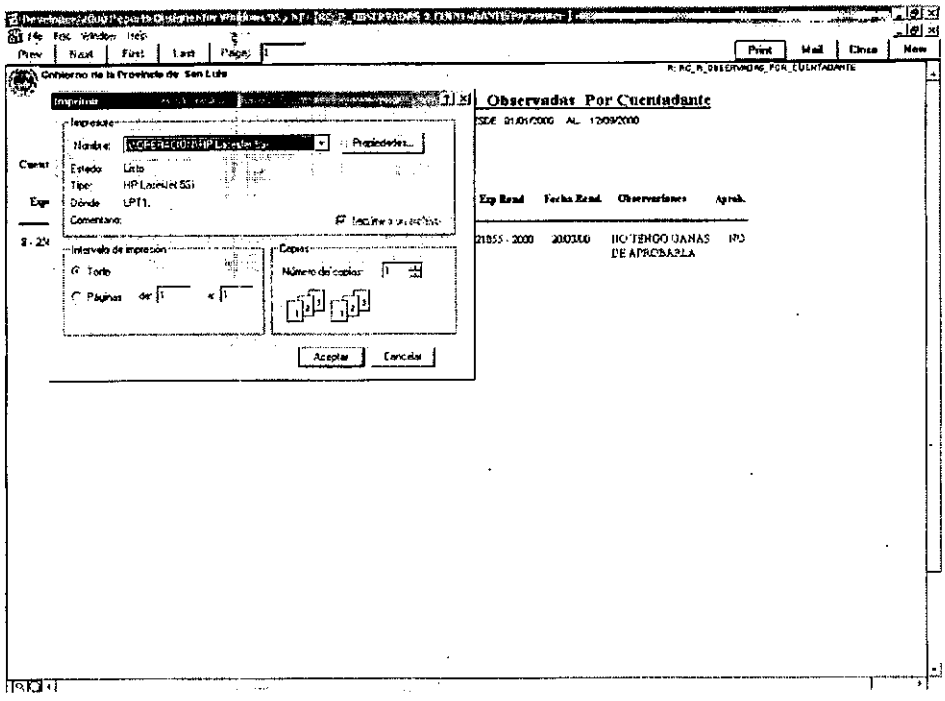
En la figura 7.81 se muestra el reporte ya ejecutándose y al detectar que tiene parámetros de entrada solicita su carga por medio de esta pantalla. Luego al hacer click en el botón "Run Report" se ejecuta el reporte esta ejecución se puede observar en la figura 7.82.



(Figura 7.82)

En la figura 7.82 se muestra la vista previa del reporte, en este punto el usuario puede elegir entre imprimir el reporte o simplemente cerrarlo, en el caso de querer imprimirlo debe hacer click el botón "Print", con esto se le mostrará al usuario una ventana con las propiedades de las impresoras (figura 7.83).

Para poder cerrar el reporte el usuario debe hacer click en el botón "Close".


Observadas Por Cuentadante
Imprimir
Nombre:
Propiedades...
Estado: Lista
Tipo: HP LaserJet 5Si
Dónde: LPT1:
Comentario:
Intervalo de impresión
Todo
Páginas
Copias
Número de copias:
Aceptar
Cancelar
Exp Rend
Fecha Rend.
Observaciones
Aprob.
21855 - 2000
2003300
NO TENGO DANAS DE APROBARLA
NO

(Figura 7.83)

ACTIVIDAD 8

8.1 OBJETIVO

Integración del sistema con otros sistemas externos (Sistema Contable) y corrección de fallas funcionales y de integración.

8.2 INTRODUCCIÓN

En este informe parcial mostramos los módulos que son del sistema contable y que fueron modificados para poder integrarlos con la funcionalidad del sistema "Rendiciones de Cuentas".

Cabe destacar que estos módulos fueron desarrollados en "Oracle Developer/2000 Forms Designer for Windows 95 / NT" y solo se mostrarán códigos fuentes que pertenecen al sistema de rendiciones de cuenta.

En esta actividad describimos el desarrollo de los puntos específicos que forman parte del sistema de rendición de cuentas y que estén embebidos en módulos del sistema contable.

Los módulos principales del sistema contable que fueron modificados, son los módulos que hacen a la etapa final de los pagos dentro del circuito de gestión interna de un pago. El cargo de una rendición de cuenta se genera en el momento en el cual se le entrega el dinero al beneficiario; a partir de allí comienza a contarse 30 días que otorga la ley para poder realizar la presentación de la rendición.

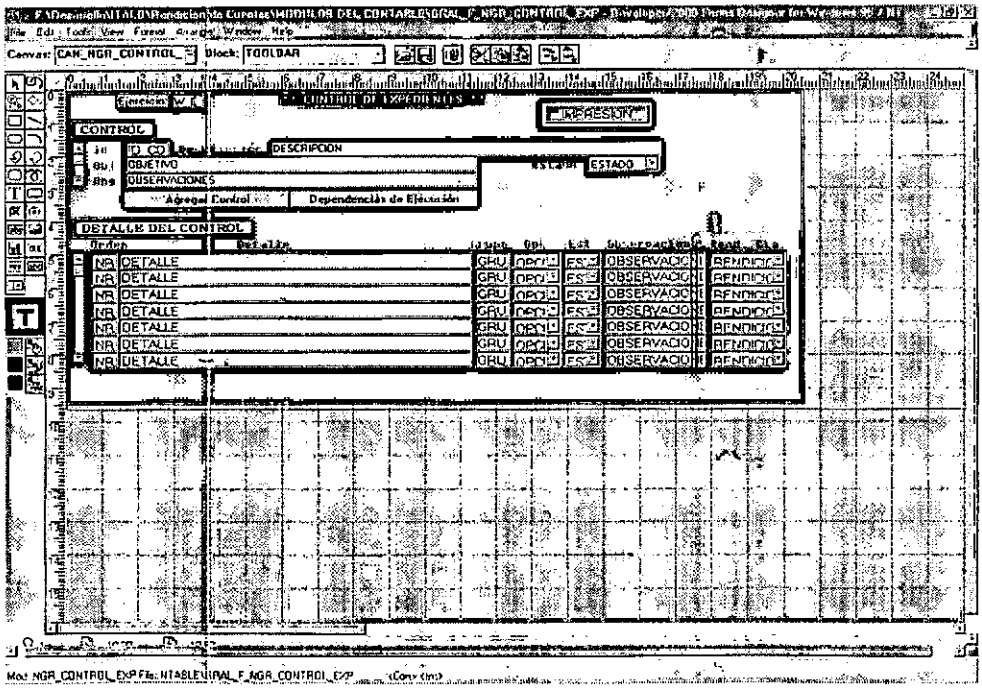
8.3 DESCRIPCIÓN DE MÓDULOS

En este punto se detallan los módulos desde el ambiente de desarrollo "Developer Forms", también se muestran las pantallas que interactúan como interfaces con el usuario final en el momento de la ejecución del sistema.

Estos módulos son propios del sistema contable a los cuales se les ha agregado funcionalidad para que cumplan con los requisitos del sistema de rendición de cuentas.

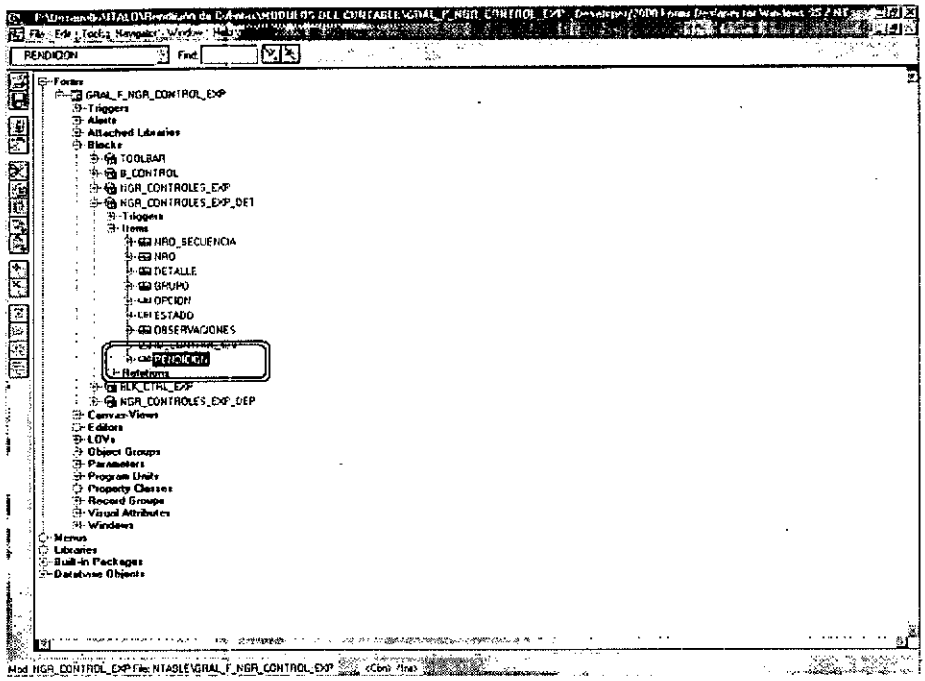
8.3.1 Módulo de control de expedientes

En la figura 8.1 se muestra el módulo donde se cargan los nomencladores de control de expedientes, estos controles indican cuales serán aquellos expedientes que deben realizar rendición de cuenta. Para esto dichos expedientes deben ser seleccionados, estos controles se encuentran almacenados en la tabla NGR_CONTROLES_EXP_DEP a la cual se le agregó una columna 'RENDICIÓN' con este objetivo.



(Figura 8.1)

Como se observa en la figura 8.1, el campo que se agrega a este módulo está enmarcado por un cuadrado rojo.



(Figura 8.2)

En la figura 8.2 se puede observar el ítem que se relacionó con el campo al cual estamos haciendo referencia, este ítem se encuentra dentro del bloque NGR_CONTROLES_EXP_DET, este bloque está asociado con la tabla del mismo nombre.

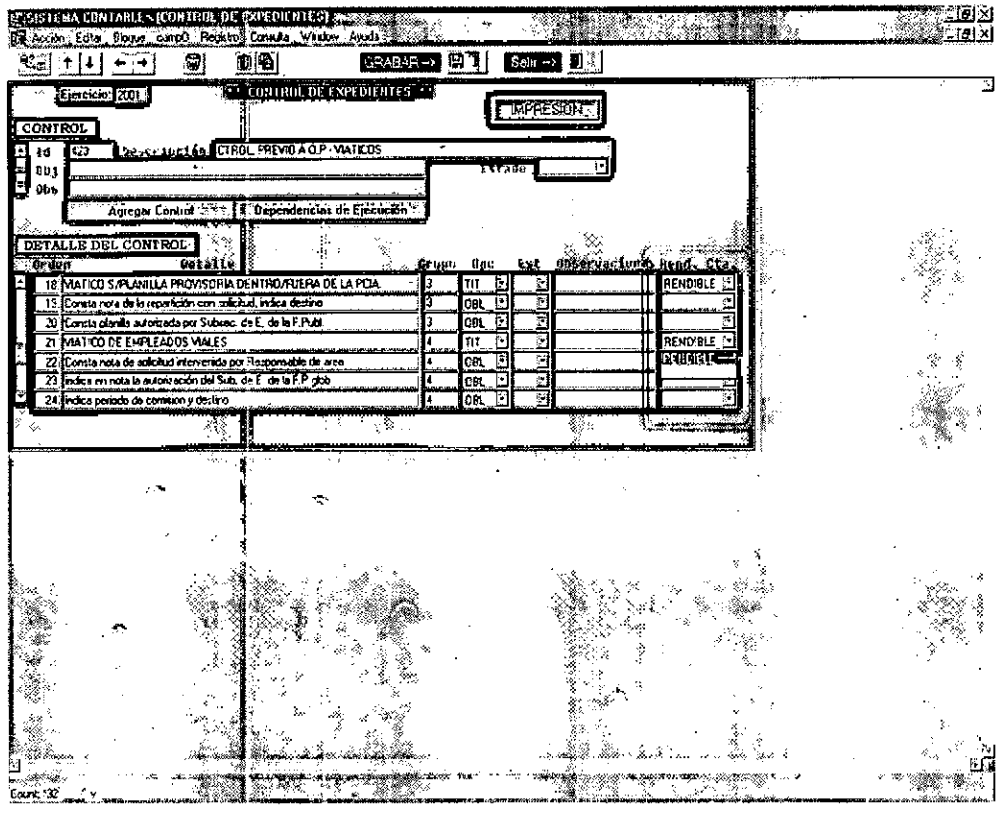
Para controlar que el registro al cual se le agrega el campo de rendición, sea un título que engloba a varios controles, se añadió el siguiente código a este módulo:

```
DECLARE
  titulo_alerta              VARCHAR(30);
  texto_alerta               VARCHAR2(150);
  sigo                       NUMBER;
BEGIN
IF NOT( :NGR_CONTROLES_EXP_DET.NRO IS NOT NULL  ) THEN
      TITULO_ALERTA := 'Nomenclador de controles';
      TEXTO_ALERTA  := 'ERROR: El orden en el detalle en un control no puede omitirse...';
      Set_Alert_Property('ALERTA_ACEPTAR', title, TITULO_ALERTA);
      Set_Alert_Property('ALERTA_ACEPTAR', alert_message_text, TEXTO_ALERTA );
      SIGO := Show_Alert('ALERTA_ACEPTAR');
     raise form_trigger_failure;
END IF;
IF NOT( :NGR_CONTROLES_EXP_DET.detalle IS NOT NULL ) THEN
      TITULO_ALERTA := 'Nomenclador de controles';
      TEXTO_ALERTA  := 'ERROR: El detalle del control no puede omitirse...';
      Set_Alert_Property('ALERTA_ACEPTAR', title, TITULO_ALERTA);
      Set_Alert_Property('ALERTA_ACEPTAR', alert_message_text, TEXTO_ALERTA );
      SIGO := Show_Alert('ALERTA_ACEPTAR');
     raise form_trigger_failure;
end if;
IF (:NGR_CONTROLES_EXP_DET.RENDICION IS NOT NULL AND :NGR_CONTROLES_EXP_DET.OPCION <> 'TIT') THEN
      TITULO_ALERTA := 'Rendición de Cuentas';
      TEXTO_ALERTA  := 'ERROR: La Opción del control no es un "TIT". . . ';
      Set_Alert_Property('ALERTA_ACEPTAR', title, TITULO_ALERTA);
      Set_Alert_Property('ALERTA_ACEPTAR', alert_message_text, TEXTO_ALERTA );
```

```
        SIGO := Show_Alert('ALERTA_ACEPTAR');
      raise form_trigger_failure;
END IF;
END;
```

Este código se ejecuta en el momento en que el usuario que administra estos nomencladores de controles sale del registro que está modificando o agregando.

El texto que se agregó al código original es lo que se observa con letra cursiva negrita.



(Figura 8.3)

En la figura 8.3 se muestra el módulo en ejecución (en el ambiente de prueba), donde podemos observar en el detalle del control "CTROL PREVIO A O.P. – VIATICOS" varios ítems de tipo "TIT" (títulos) a los que se le indica a aquellos expedientes que pasen por estos controles, que deben realizar la rendición de cuenta.

El control "CTROL PREVIO A O.P. – VIATICOS" se encuentra almacenado en la tabla "NGR_CONTROLES_EXP" y cumple la función de '*padre*' en una relación "1 a n" con la tabla "NGR_CONTROLES_EXP_DET" donde se encuentran los detalles de cada control. Estos detalles pueden ser varios grupos de cada control, cada grupo de registro viene especificado por un atributo **OPCION** = 'TIT' y un atributo **GRUPO** cuyo valor indica a que grupo de detalles pertenece.

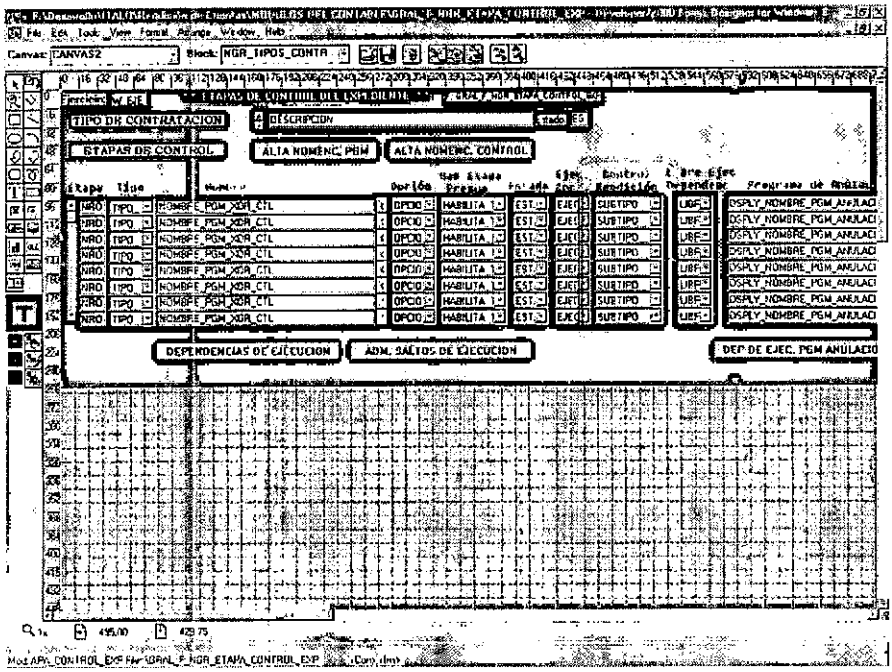
8.3.2 Módulo de Etapas de control de expedientes

Un expediente puede pasar por varias etapas (controles) antes de que se realize un pago. Por lo general todo expediente común pasa por dos etapas, la primera es el control previo a la emisión de la orden de pago (realizada por la repartición imputante), la segunda etapa es el control previo a la firma de la orden de pago (realizada por la contaduría general), una vez superados estos controles la Jefatura de Programas Pagos a Proveedores y Municipios (ex. Dirección Provincial de Tesorería General) está habilitada para realizar los pagos correspondientes.

Los expedientes que sean de rendición se diferencian de los descritos anteriormente porque deberán tener una etapa previa a las mencionadas anteriormente, que es la etapa de control de la rendición. Luego si esta etapa se ejecuta correctamente, lo que hace el sistema es generar un descargo de rendición al beneficiario.

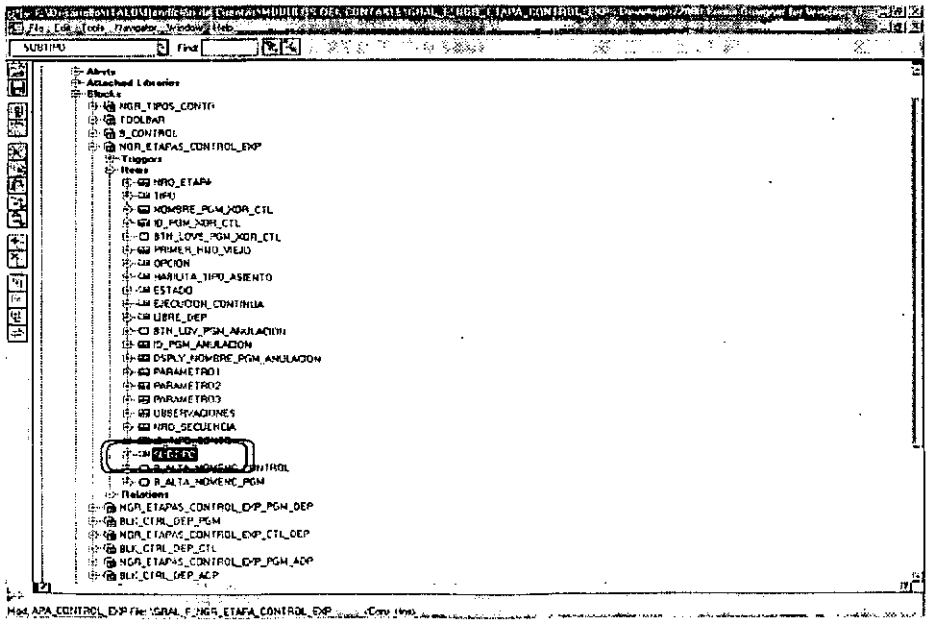
Para que el sistema pueda reconocer que el usuario está ejecutando esta etapa, fue necesario agregar un atributo a la tabla "NGR_ETAPAS_CONTROL_EXP" (mantiene el nomenclador de las etapas que

se asocian a cada tipo de contratación), cuyo nombre es "**SUBTIPO**"; cuando se indique que la etapa es de rendición, este atributo tomará el valor 'CRC' que significa Control de Rendición de Cuentas.



(Figura 8.4)

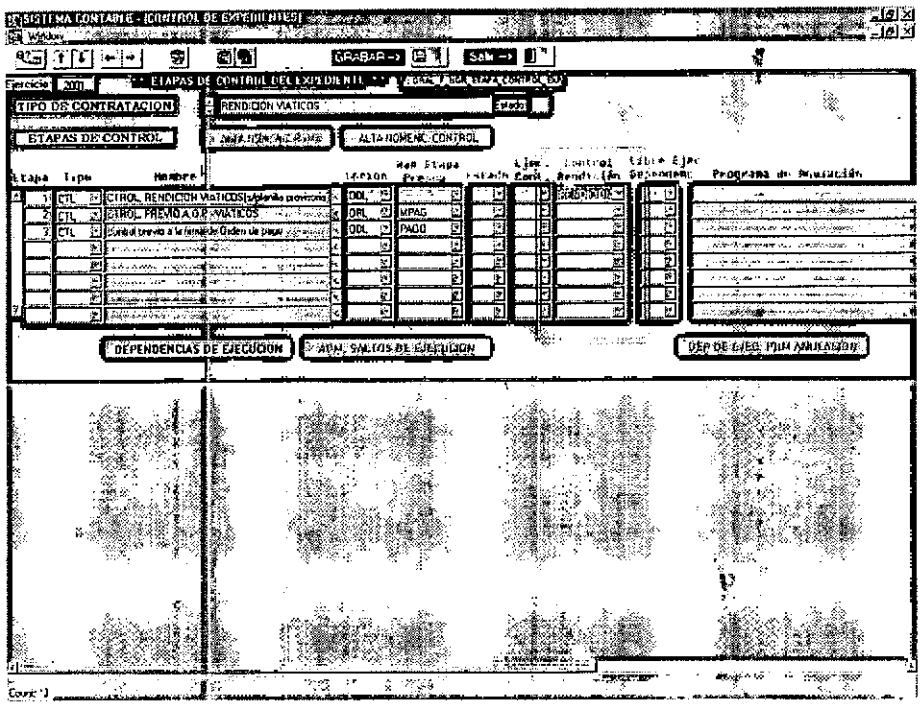
En la figura 8.4 se puede observar el campo que se agrega a este módulo enmarcado por un cuadrado rojo.



(Figura 8.5)

En la figura 8.5 se puede observar el ítem que se relaciona con el campo que hicimos referencia anteriormente, este ítem se encuentra dentro del bloque NGR_ETAPAS_CONTROL_EXP, este bloque está directamente relacionado con la tabla del mismo nombre.

En éste módulo no fue necesario incorporar código fuente, los cambios realizados son suficiente para obtener la funcionalidad necesitada.



(Figura 8.6)

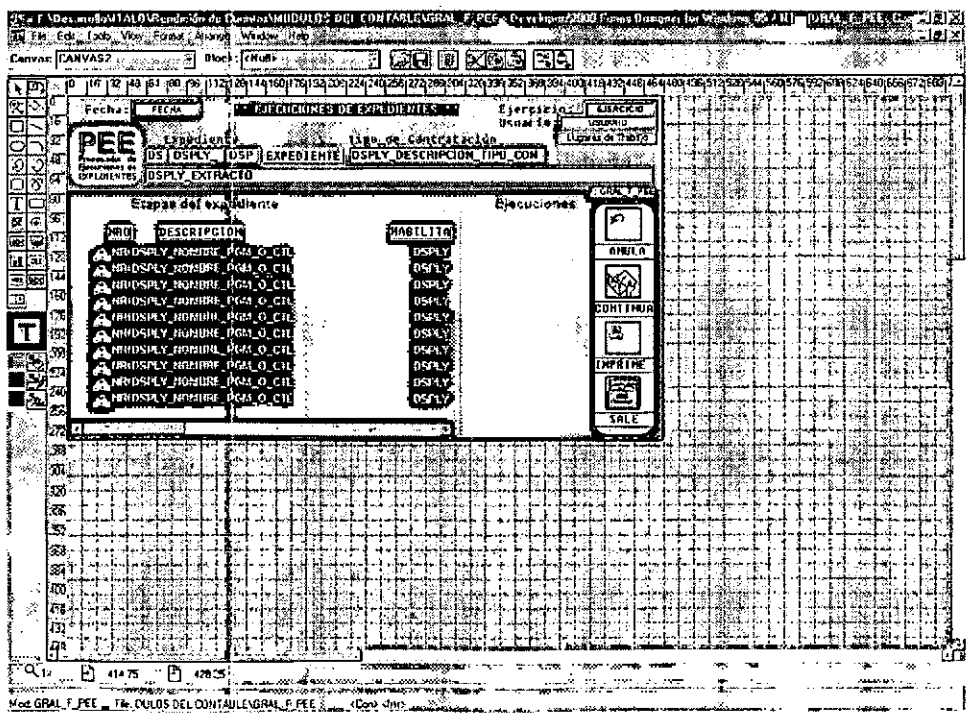
En la figura 8.6 se muestra el módulo "Etapas de Control de Expedientes" en ejecución (en ambiente de prueba), donde podemos observar las distintas etapas por las que debe pasar un expediente asociado a un tipo de contratación en particular. En la figura el tipo de contratación "RENDICION VIATICOS" tiene tres etapas de controles asociado:

- La primera es la que controla la rendición. Para que esta etapa se cumpla correctamente el campo "Control de Rendición" (cuadro rojo) debe contener el valor 'RENDICION'.

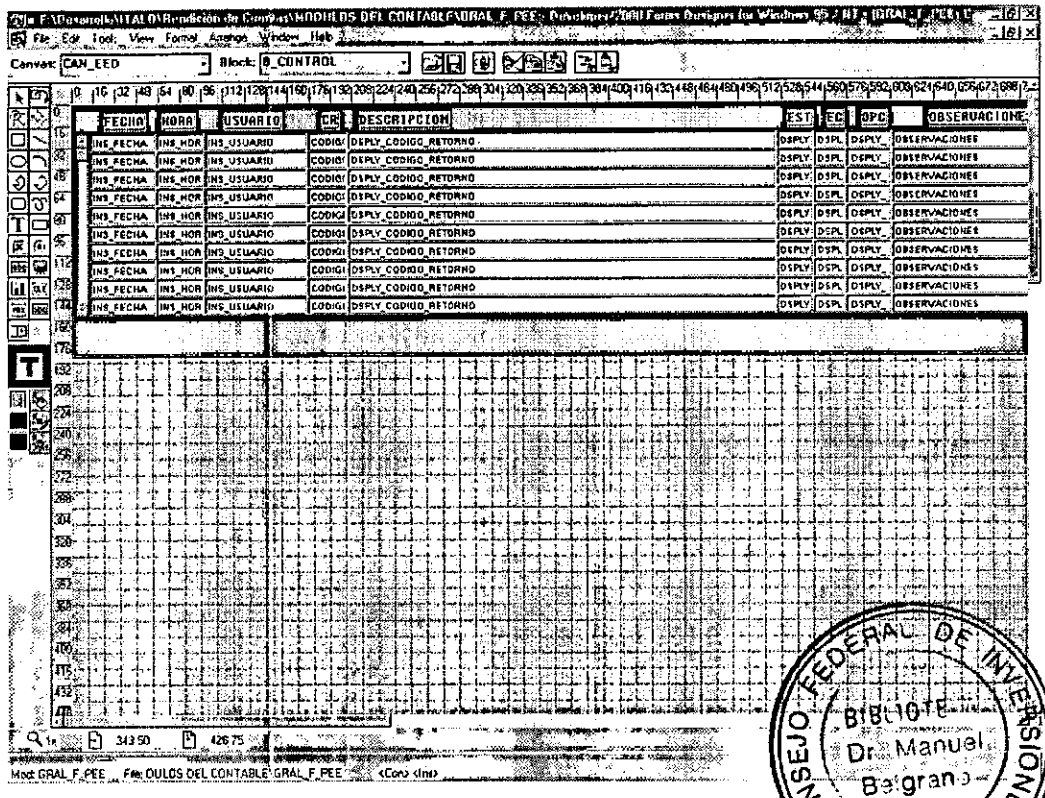
- En la segunda etapa va el control previo a la orden de pago u orden de reintegro.
- Y por último la tercera etapa es el control previo a la firma de la orden de pago/reintegro.

8.3.3 Módulo de Ejecuciones de expedientes

En este módulo se relacionan los dos puntos mencionados anteriormente, (en uno se muestran las etapas a realizar, en otro se muestran que controles se le deben aplicar a un expediente). Aquí se asocian registros generados por ambos módulos, también se registra para cada etapa todas las ejecuciones de controles que se realizaron (quedan registradas varias ejecuciones porque no todas superan los controles requeridos, por esto una etapa queda inconclusa y debe reiterarse el proceso, hasta que se realice una ejecución correcta de tal forma que esta supere todos los controles).



(Figura 8.6)

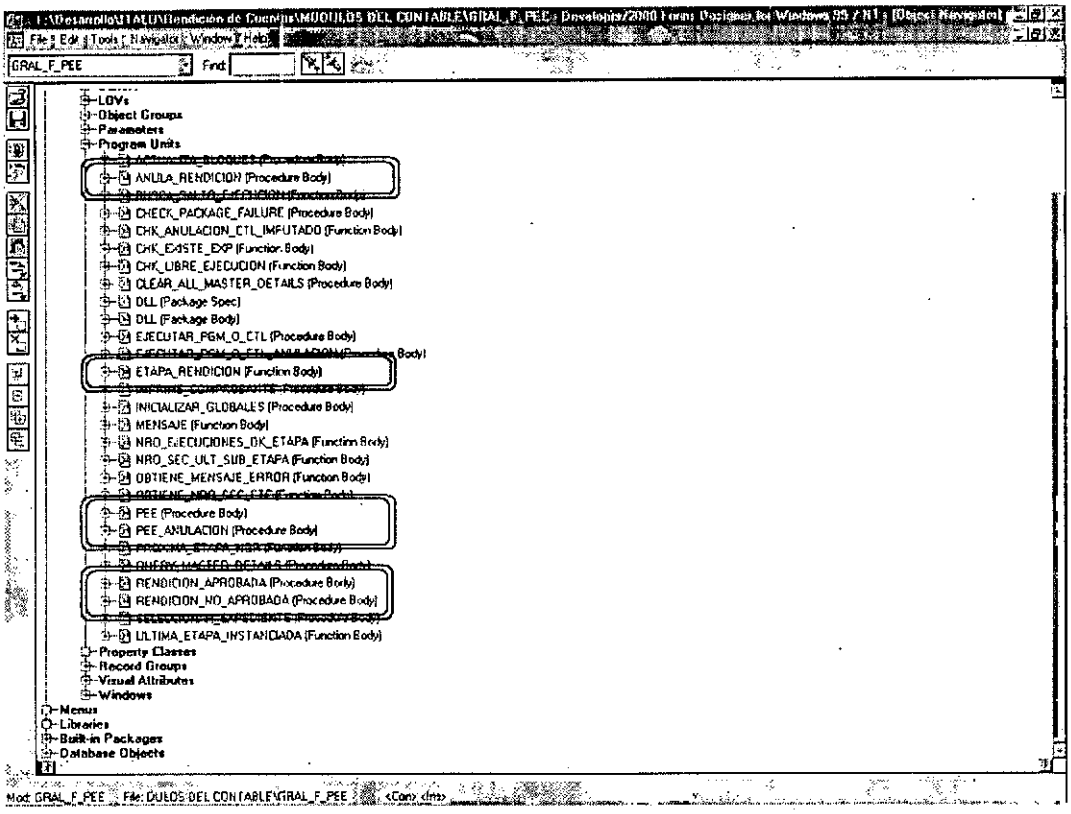


CONSEJO FEDERAL DE INVERSIONES
BIBLIOTECA
Dr. Manuel Belgrano

(Figura 8.7)

En la figura 8.6 se puede observar el form de este módulo en su etapa de desarrollo, donde se puede advertir un lugar vacío debajo de la descripción **Ejecuciones**, este lugar será rellenado en tiempo de ejecución con la "canvas-views" que se muestra en la figura 8.7.

Para poder realizar la ejecución de una etapa hay que hacer click en el botón CONTINÚA; a través de este botón se invoca a un procedimiento llamado PEE, el cual contiene código fuente que fue modificado, cuyo agregado contiene funcionalidad del sistema de rendición de cuentas. Otro de los procedimientos que fue modificado es el PEE_ANULACION que es invocado desde el botón ANULA.



(Figura 8.8)

En la figura 8.8 se puede observar con un recuadro en color rojo los Program Units (Unidades de Programas) PEE y PEE_ANULACION que fueron modificadas; y en recuadro de color azul los Program Units (ETAPA_RENDICION, RENDICION_APROBADA, ANULA_RENDICION y RENDICION_NO_APROBADA) que fueron agregados.

A continuación se muestra el código fuente de las unidades de programa nuevas agregadas a este módulo.

- ETAPA_RENDICION: Es una función que retorna 'SI' en caso de que la etapa que se esta ejecutando sea una etapa de rendición, y retorna 'NO' en otro caso.

```
FUNCTION ETAPA_RENDICION (NRO_SEC_ECE NUMBER)RETURN VARCHAR2 IS
V_SUBTIPO  NGR_ETAPAS_CONTROL_EXP.SUBTIPO%TYPE;
BEGIN
     SELECT ECE.SUBTIPO
       INTO V_SUBTIPO
       FROM NGR_ETAPAS_CONTROL_EXP ECE
      WHERE ECE.NRO_SECUENCIA = NRO_SEC_ECE;

    IF V_SUBTIPO = 'CRC' THEN RETURN 'SI';
    ELSE RETURN 'NO';
```

```
    END IF;
END;
```

- RENDICION_APROBADA: Esta función se encarga de generar un registro en la tabla "CO_RC_RENDICIONES" si los controles de la etapa de rendición fueron superados con éxito.

```
PROCEDURE RENDICION_APROBADA (NRO_SEC_EXP NUMBER) IS

NRO_SEC_EXP_PAGO  CO_EXPEDIENTES.NRO_SECUENCIA%TYPE;
NRO_SEC_OP        CO_REGISTRO_ORDENES.NRO_SECUENCIA%TYPE;
NRO_SEC_REND      NUMBER;
FECHA_ACTUAL      DATE;

BEGIN

-- ------------------------------------------------------------------------
-- SELECCIONO EL EXPEDIENTE DE PAGO
-- ------------------------------------------------------------------------

  SELECT E.NRO_SECUENCIA_EXP_RENDIDO
    INTO NRO_SEC_EXP_PAGO
    FROM CO_EXPEDIENTES E
   WHERE E.NRO_SECUENCIA = NRO_SEC_EXP;

-- ------------------------------------------------------------------------
-- SELECCIONO LA ORDEN DE PAGO
-- ------------------------------------------------------------------------

  SELECT RO.NRO_SECUENCIA
    INTO NRO_SEC_OP
    FROM CO_REGISTRO_ORDENES RO
   WHERE RO.NRO_SECUENCIA_EXP = NRO_SEC_EXP_PAGO
     AND RO.CARGO IS NOT NULL;

-- ------------------------------------------------------------------------
-- GENERO LA RENDICIÓN NO APROBADA
-- ------------------------------------------------------------------------

  SELECT s_co_rc_n_ap_provs.NEXTVAL
    INTO NRO_SEC_REND
    FROM dual;

  SELECT SYSDATE INTO FECHA_ACTUAL FROM DUAL;

   INSERT INTO CO_RC_RENDICIONES (CO_R_OR_NRO_SECUENCIA,
                                  ID_RENDICION,
                                FECHA_ENTREGA,
                                APROBADA,
                                CO_EXP_NRO_SECUENCIA)
   VALUES (NRO_SEC_OP,NRO_SEC_REND,FECHA_ACTUAL,'SI',NRO_SEC_EXP);

END;
```

- RENDICION_NO_APROBADA: En caso de que los controles de la etapa de rendición de cuentas no hayan sido superados con éxito, esta función es la encargada de generar un registro en la tabla "CO_RC_RENDICIONES" con el valor 'NO' en el atributo APROBADA.

```
PROCEDURE RENDICION_NO_APROBADA (NRO_SEC_EXP NUMBER) IS

NRO_SEC_EXP_PAGO  CO_EXPEDIENTES.NRO_SECUENCIA%TYPE;
NRO_SEC_OP        CO_REGISTRO_ORDENES.NRO_SECUENCIA%TYPE;
NRO_SEC_REND      NUMBER;
FECHA_ACTUAL      DATE;

BEGIN
```

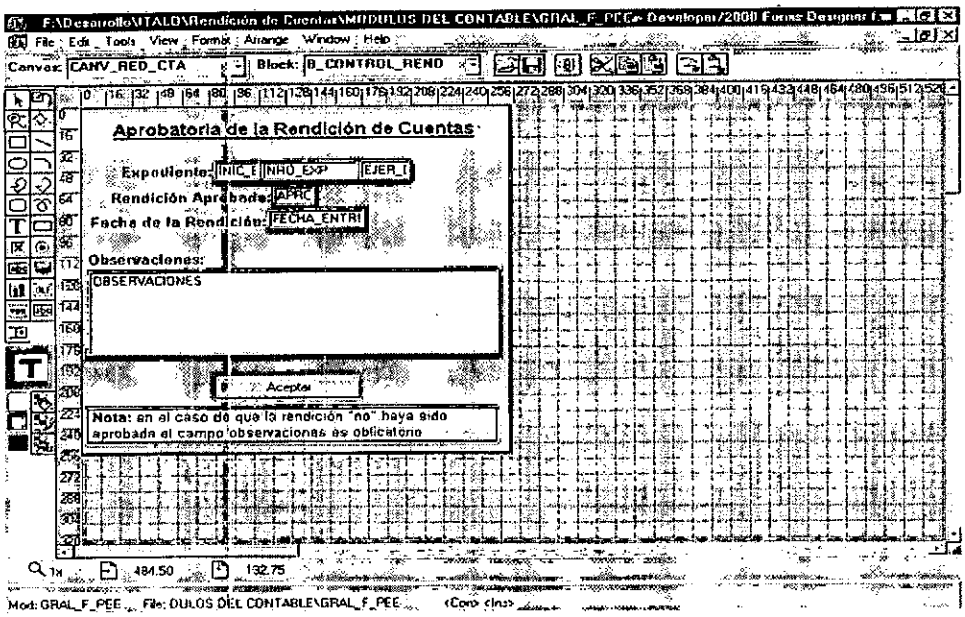
```
-- ------------------------------------------------------------------------
-- SELECCIONO EL EXPEDIENTE DE PAGO
-- ------------------------------------------------------------------------

  SELECT E.NRO_SECUENCIA_EXP_RENDIDO
    INTO NRO_SEC_EXP_PAGO
    FROM CO_EXPEDIENTES E
   WHERE E.NRO_SECUENCIA = NRO_SEC_EXP;

-- ------------------------------------------------------------------------
-- SELECCIONO LA ORDEN DE PAGO
-- ------------------------------------------------------------------------

  SELECT RO.NRO_SECUENCIA
    INTO NRO_SEC_OP
    FROM CO_REGISTRO_ORDENES RO
   WHERE RO.NRO_SECUENCIA_EXP = NRO_SEC_EXP_PAGO
     AND RO.CARGO IS NOT NULL;

-- ------------------------------------------------------------------------
-- GENERO LA RENDICIÓN NO APROBADA
-- ------------------------------------------------------------------------

  SELECT s_co_rc_n_ap_provs.NEXTVAL
    INTO NRO_SEC_REND
    FROM dual;

  SELECT SYSDATE INTO FECHA_ACTUAL FROM DUAL;

   INSERT INTO CO_RC_RENDICIONES (CO_R_OR_NRO_SECUENCIA,
                                  ID_RENDICION,
                                  FECHA_ENTREGA,
                                  APROBADA,
                                  CO_EXP_NRO_SECUENCIA)
   VALUES (NRO_SEC_OP,NRO_SEC_REND,FECHA_ACTUAL,'NO',NRO_SEC_EXP);

END;
```

- PEE: Este procedimiento se encarga de controlar todas las ejecuciones de las etapas que tiene que cumplir cada expediente, cabe destacar que en el código que se muestra a continuación solo se observa código nuevo que cumple con la funcionalidad del sistema de rendición de cuentas, el resto del código (propio del sistema contable) estará enmarcado por puntos suspensivos para reguardar la integridad del sistema contable.

```
PROCEDURE pee(nro_sec_exp IN NUMBER) IS

   .........
   /* Lugar para las declaraciones de variables locales de este procedimiento */

BEGIN

   ...........

LOOP      /* para la ejecucion_continua */

  .........
  /* en este lugar el sistema realiza operaciones propias del sistema contable */

-- ------------------------------------------------------------------------------------------
-- Control de los expedietnes que son factibles de rendición de cuentas
-- ------------------------------------------------------------------------------------------
```

```
  IF etapa_rendicion(:CO_EXPEDIENTES_EJECUCION_DET.NRO_SECUENCIA_ETAPA_CONTROL) = 'SI' THEN
     IF :GLOBAL.codigo_retorno = 0 THEN
        IF :GLOBAL.continua_ejecucion = 'SI' THEN
           rendicion_no_aprobada(:B_CONTROL.NRO_SEC_EXP);
        ELSE
           rendicion_aprobada(:B_CONTROL.NRO_SEC_EXP);
        END IF;
     END IF;
  END IF;

-- ------------------------------------------------------------------------------------------

  .........

-- ------------------------------------------------------------------------------------------
-- cargar las observaciones de las rendiciones
-- ------------------------------------------------------------------------------------------
  IF etapa_rendicion(:CO_EXPEDIENTES_EJECUCION_DET.NRO_SECUENCIA_ETAPA_CONTROL) = 'SI' THEN
     IF :GLOBAL.codigo_retorno = 0 THEN

        select r.fecha_entrega,r.aprobada
          into :b_control_rend.fecha_entrega, :b_control_rend.aprobada
          from co_rc_rendiciones r
         where r.co_exp_nro_secuencia = :B_CONTROL.NRO_SEC_EXP;

        :b_control_rend.nro_sec_exp := :B_CONTROL.NRO_SEC_EXP;
        :b_control_rend.inic_exp    := :B_CONTROL.dsply_inic;
        :b_control_rend.nro_exp     := :B_CONTROL.dsply_numero;
        :b_control_rend.ejer_exp    := :B_CONTROL.dsply_ejercicio;

        go_item(:b_control_rend.observaciones);
     END IF;
  END IF;
-- ------------------------------------------------------------------------------------------

  .........

  EXIT WHEN NOT hay_ejecucion_continua; /* Si ejecucion_continua del (ngr) es NULL(NO), sale */
END LOOP;


EXCEPTION

  .........
  /* Lugar donde el sistema contable captura las excepciones que puedan ocurrir en ejecución */
  .........

END;
```

En este código se puede observar un fragmento, que es el encargado de habilitarle al usuario un campo para que pueda describir en la rendición distintas observaciones, la pantalla que utiliza el usuario se muestra en la figura 8.9.



(Figura 8.9)

En la figura 8.9 se observan los siguientes datos: el número de expediente, aprobatoria de la rendición, fecha de la rendición y por último un campo donde el usuario deberá ingresar observaciones que sean necesarias.

A continuación se describe el código relacionado con el botón "Aceptar" que se muestra en la figura.

```
declare
 mens integer;
begin
     if (:b_control_rend.aprobada = 'NO') and (:b_control_rend.observaciones is null) then
          mens := mensaje('ACEPTAR','Observaciones','Error ==> El campo observación no puede ser
nulo en una rendicion no aprovada');
          go_item(:b_control_rend.observaciones);
        else
            update co_rc_rendiciones
               set observaciones = :b_control_rend.observaciones
            where co_exp_nro_secuencia = :b_control_rend.nro_sec_exp;
            commit;
        end if;
end;
```

- PEE_ANULACION: Esta función cumple el rol inverso a la anterior, lo que hace es anular una ejecución de una etapa correspondiente a un expediente, por lo tanto también debe ser anulado el registro de la rendición en el caso de que este exista.

```
PROCEDURE pee_anulacion ( nro_sec_exp IN NUMBER) IS

   .........
```

```
   /* Lugar para las declaraciones de variables locales de este procedimiento */
BEGIN


   ..........

LOOP      /* para la ejecucion_continua */


  .........
  /* en este lugar el sistema realiza operaciones propias del sistema contable */

-- ------------------------------------------------------------------------------------------
-- Control de los expedietnes que son factibles de rendición de cuentas para anular la misma
-- ------------------------------------------------------------------------------------------
/*
  IF etapa_rendicion(:CO_EXPEDIENTES_EJECUCION_DET.NRO_SECUENCIA_ETAPA_CONTROL) = 'SI' THEN
     IF :GLOBAL.codigo_retorno = 0 THEN
           anula_rendicion(:B_CONTROL.NRO_SEC_EXP);
    END IF;
  END IF;
*/
-- ------------------------------------------------------------------------------------------


  .........

  EXIT WHEN NOT hay_ejecucion_continua; /* Si ejecucion_continua del (ngr) es NULL(NO), sale */
END LOOP;


EXCEPTION

  .........
  /* Lugar donde el sistema contable captura las excepciones que puedan ocurrir en ejecución */
  .........

END;
```

En este procedimiento se puede observar que la principal funcionalidad de rendiciones de cuenta es hacer la invocación al procedimiento anula_rendición.

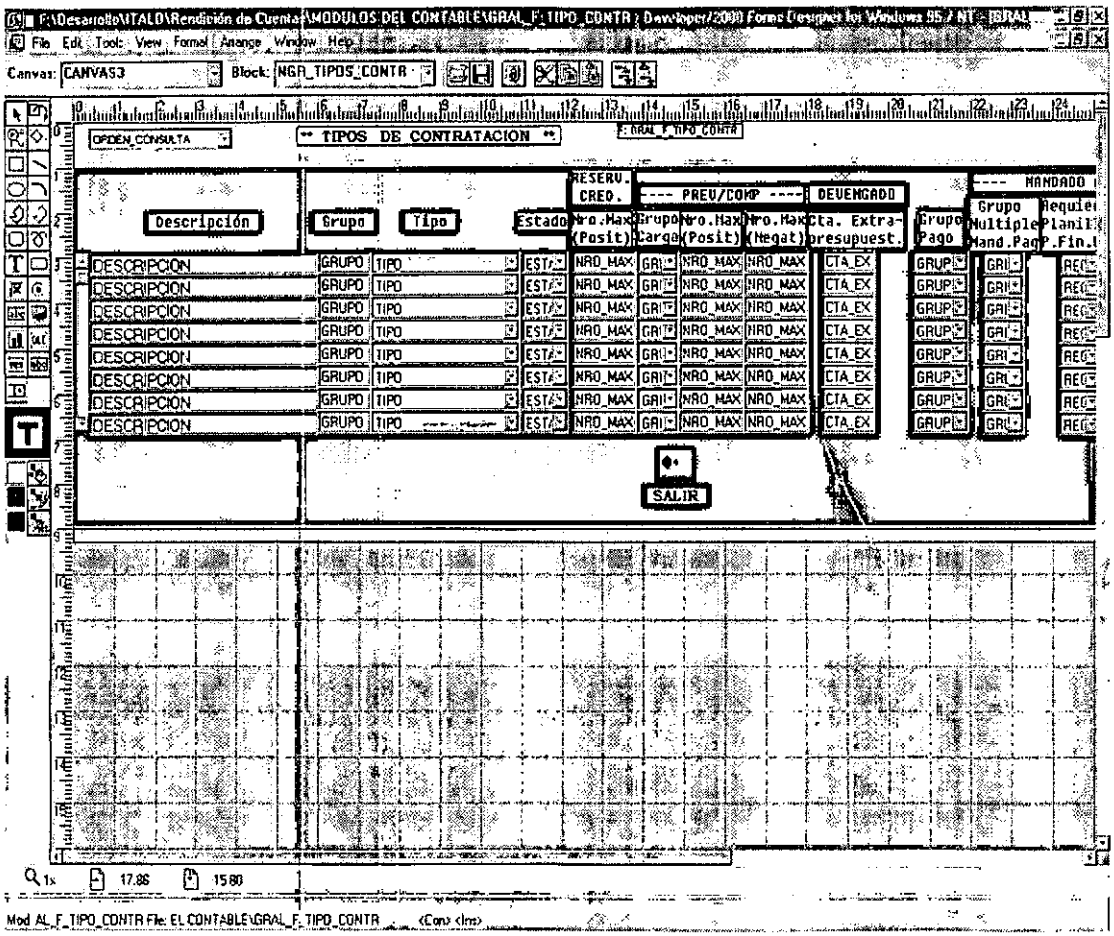
- ANULA_RENDICION: la funcionalidad de este procedimiento es eliminar el registro correspondiente a la rendición del expediente dado, este registro que se elimina se encuentra en la tabla CO_RC_RENDICIONES.

```
PROCEDURE anula_rendicion  (NRO_SEC_EXP NUMBER) IS

BEGIN
-- ------------------------------------------------------------------
-- ELIMINO LA RENDICIÓN
-- ------------------------------------------------------------------
   DELETE CO_RC_RENDICIONES R
    WHERE R.CO_EXP_NRO_SECUENCIA = NRO_SEC_EXP;

END;
```

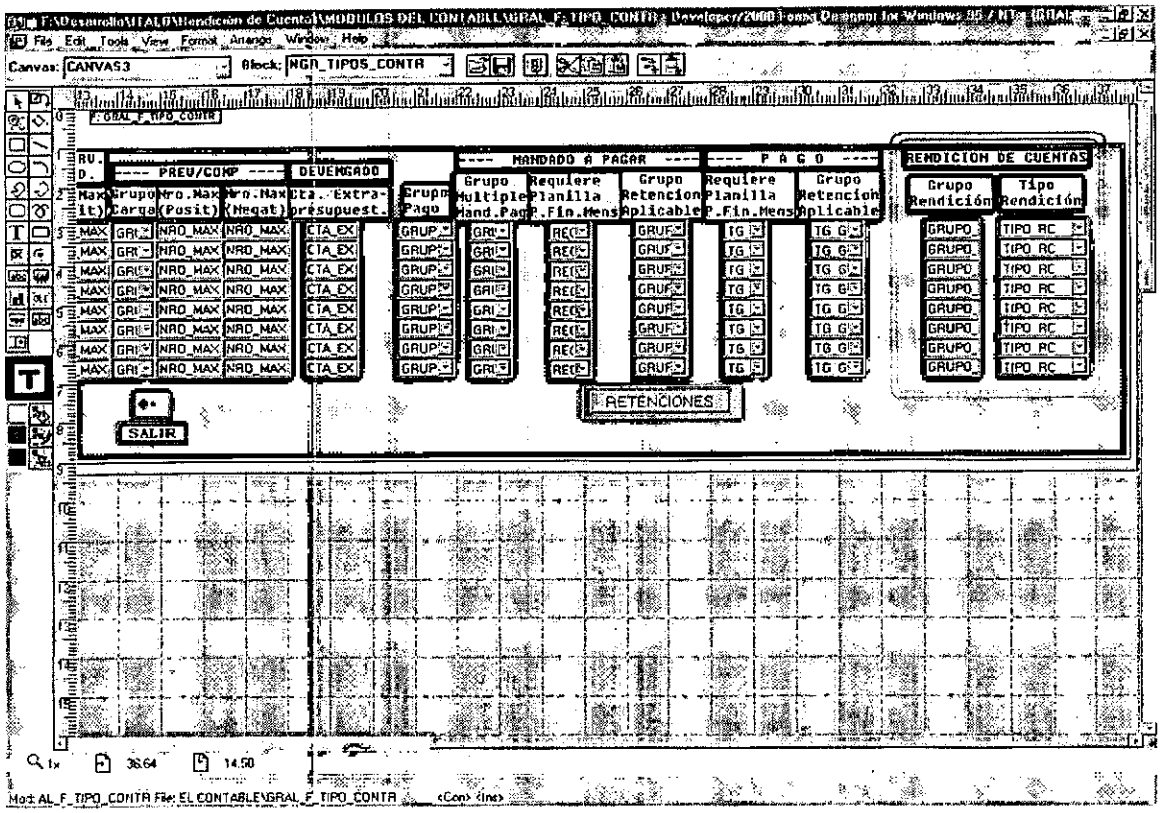
8.3.4 Módulo de Tipos de Contrataciones

Para poder manejar dentro del sistema de rendiciones de cuentas expedientes tanto de pago, como expedientes de rendiciones, fue necesario modificar la

tabla "NGR_TIPOS_CONTR" agregándole dos campos **GRUPO_RC** y **TIPO_RC**, donde 'tipo_rc' indica si el expediente asociado a este tipo de contratación es un expediente de pago o un expediente de rendición, mientras que el campo 'grupo_rc' asocia los expediente de pago con los expedientes de rendición que corresponden al mismo tipo; por ejemplo en tipo de rendición "Viáticos" debe tener el mismo grupo que el tipo de contratación "Viáticos Rendición".



(Figura 8.10)



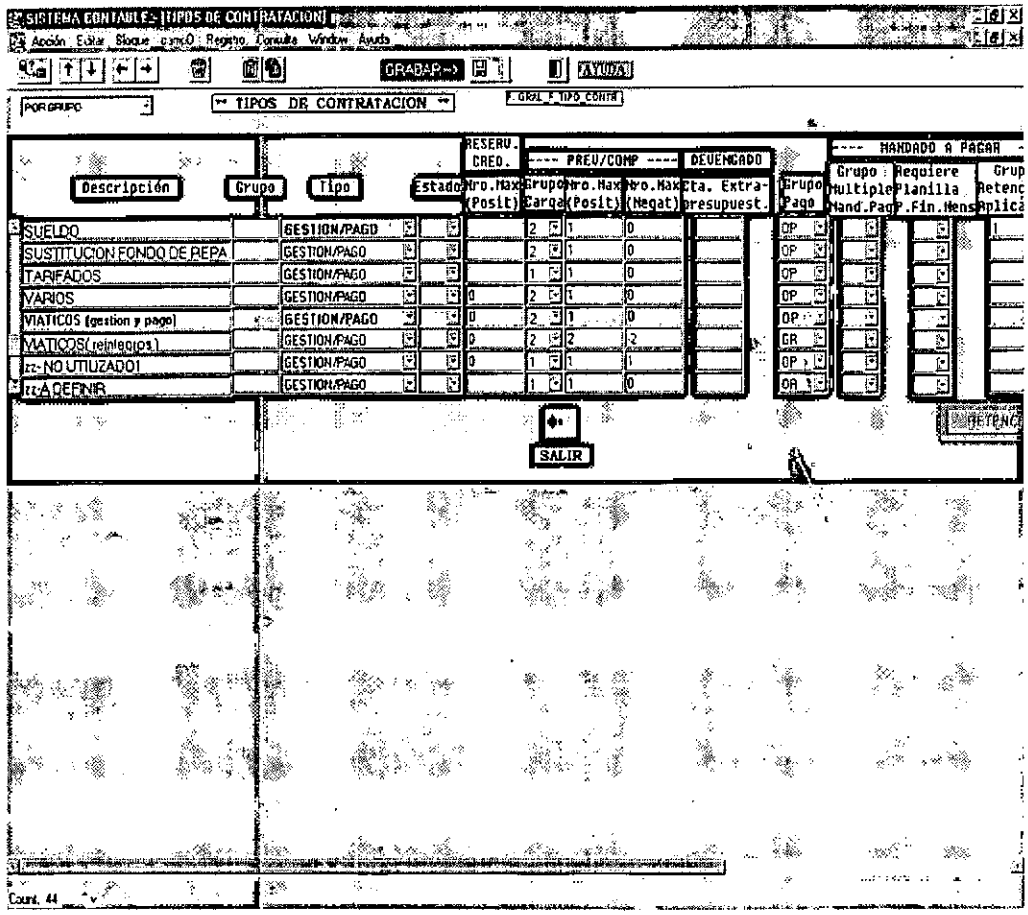
(Figura 8.11)

En las figuras 8.10 y figura 8.11 se observa la pantalla principal de este módulo, que es el encargado de manejar el nomenclador de los tipos de contrataciones, la figura 8.11 muestra los nuevos campos agregados a este módulo, que cumplen con la funcionalidad mencionada antériormente. Dentro del "*trigger*" (procedimiento) que se ejecuta cuando el usuario sale del registro, se agregó funcionalidad, esta controla que los grupos de tipos de contrataciones de rendiciones sean los correctos.

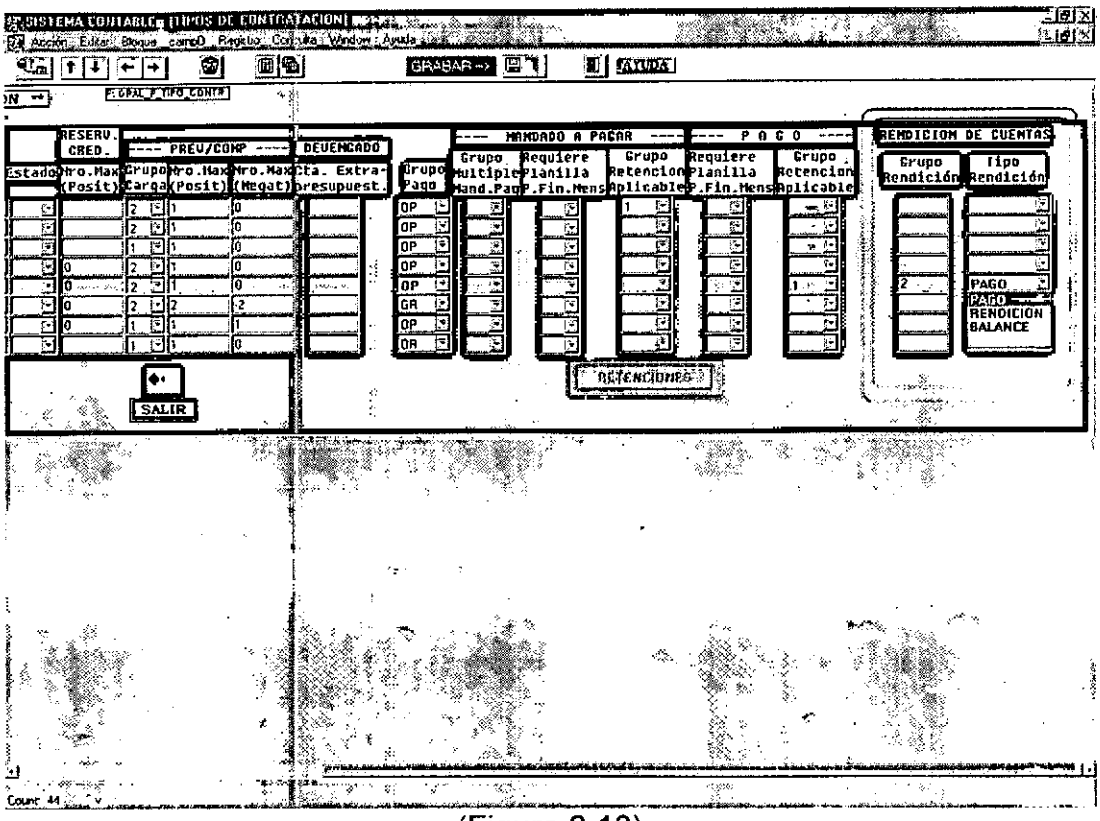
A continuación se muestra el código que cumple con esta funcionalidad.

```
DECLARE
 ...../*otras variables del sistema contable*/
 hay_grupo Number;
BEGIN

 ...../*otras controles de los tipos de contrataciones propios del sistema contable*/


-----------------------------------------
-- GRUPO DE RENDICIONES DE CUENTAS
-----------------------------------------
IF :NGR_TIPOS_CONTR.TIPO_RC IS NOT NULL AND :NGR_TIPOS_CONTR.GRUPO_RC IS NULL THEN
     ERR_TXT := '** EL GRUPO DE LAS RENDICIONES DE CUENTAS NO PUEDE SER NULO **';
     al_id := Find_Alert('ALERTA_ACEPTAR');
```

```
      Set_Alert_Property(al_id, alert_message_text, err_txt );
      al_button := Show_Alert( al_id );
      raise form_trigger_failure;
ELSIF :NGR_TIPOS_CONTR.TIPO_RC = 'R' THEN
      SELECT COUNT(1)
        INTO hay_grupo
        FROM NGR_TIPOS_CONTR TC
       WHERE TC.GRUPO_RC = :ngr_tipos_contr.grupo_rc
         AND TC.TIPO_RC = 'P';
      IF hay_grupo = 0 THEN
         ERR_TXT := '** EN EL GRUPO DE RENDICIÓN DE CUENTAS SELECCIONADO '||
                    ' NO EXISTE UN TIPO DE CONTRATACIÓN QUE SEA (PAGO) **';
         al_id := Find_Alert('ALERTA_ACEPTAR');
         Set_Alert_Property(al_id, alert_message_text, err_txt );
         al_button := Show_Alert( al_id );
         raise form_trigger_failure;
      END IF;
end if;

END IF;
END;
```



(Figura 8.12)



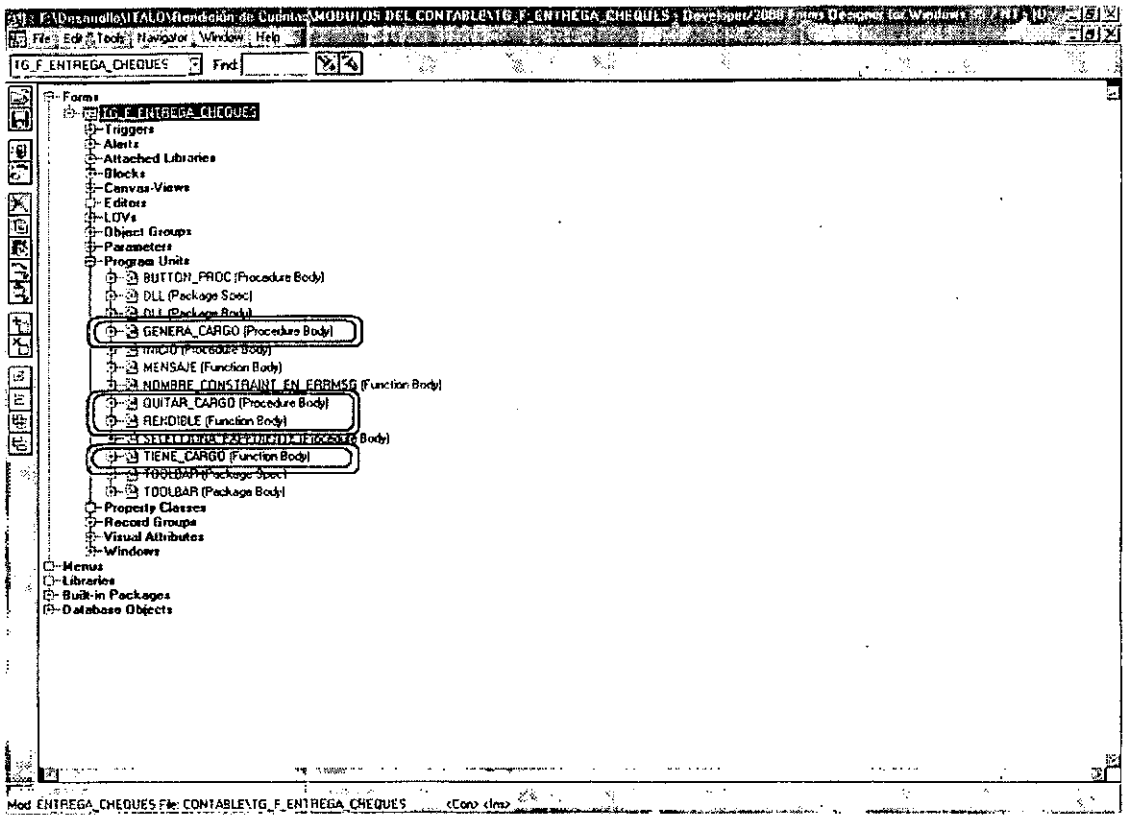
(Figura 8.13)

En las figuras 8.12 y 8.13 se observa el módulo en ejecución, además en la figura 8.13 se muestran los valores posibles que se le puede dar a un tipo de contratación en el grupo de rendiciones, internamente en el campo **'tipo_rc'** se almacenan los siguientes valores: para "PAGO" 'P', para "RENDICION" 'R' y para "BALANCE" se almacena 'B', de esta manera se relacionan los tipos de contrataciones que pertenecen a expedientes de pago y los tipos de contrataciones que pertenecen a expedientes de rendición; el tipo balance es un caso particular de las rendiciones que no lleva un expediente de pago, ya que en este expediente se presentan todos los balances de la Provincia.

8.3.5 Módulo de Entrega de Cheques

Este módulo cumple una de las principales funcionalidades del sistema de rendición de cuentas, es el módulo que registra la entrega de un cheque a un

beneficiario. Si el cheque que se entrega a un beneficiario está gestionado a través de un expediente, este debe ser rendido, entonces hay que generar un cargo de rendición al beneficiario. En el momento que se le hace la entrega correspondiente empiezan a correr 30 (treinta) días de plazos asignados por ley para realizar la presentación de la rendición de cuentas.



(Figura 8.14)

En la figura 8.14 se pueden observar enmarcadas en rojo las funciones que han sido agregadas en este módulo para poder cumplir con la funcionalidad del sistema de rendiciones de cuentas, las funciones nuevas son: GENERA_CARGO, QUITAR_CARGO, RENDIBLE y TIENE_CARGO; a continuación se muestra el código fuente de cada una de ellas.

- GENERA_CARGO: Este procedimiento es el encargado de generar un cargo de rendición al beneficiario que está recibiendo un cheque. La generación de un cargo consiste en asignarle valores a los campos

'cargo', **'fecha_cargo'** y **'fecha_vto_cargo'** dentro de la tabla CO_REGISTRO_ORDENES. Los valores para estos campos son: 'SI' para el caso de **'cargo'** y la fecha actual para el caso de **'fecha_cargo'** y la fecha actual mas treinta días para el caso de **'fecha_vto_cargo'**. A continuación se observa el código fuente de esta rutina.

```
PROCEDURE genera_cargo(SEC_REG_ORD IN INTEGER) IS
  fecha date;
BEGIN
  select trunc(sysdate) into fecha from dual;
  UPDATE CO_REGISTRO_ORDENES REG
    SET REG.CARGO = 'SI',
     REG.FECHA_CARGO = fecha,
     REG.FECHA_VTO_CARGO = fecha + 30
  WHERE REG.NRO_SECUENCIA = SEC_REG_ORD;
END;
```

- QUITAR_CARGO: Este procedimiento tiene la funcionalidad de quitar un cargo en el caso de que se anule la entrega de un cheque, este es el proceso inverso al que tiene el procedimiento explicado anteriormente. A continuación se muestra el código fuente de este procedimiento.

```
PROCEDURE genera_cargo(SEC_REG_ORD IN INTEGER) IS
  fecha date;
BEGIN
  select trunc(sysdate) into fecha from dual;
  UPDATE CO_REGISTRO_ORDENES REG
    SET REG.CARGO = 'SI',
     REG.FECHA_CARGO = fecha,
     REG.FECHA_VTO_CARGO = fecha + 30
  WHERE REG.NRO_SECUENCIA = SEC_REG_ORD;
END;
```

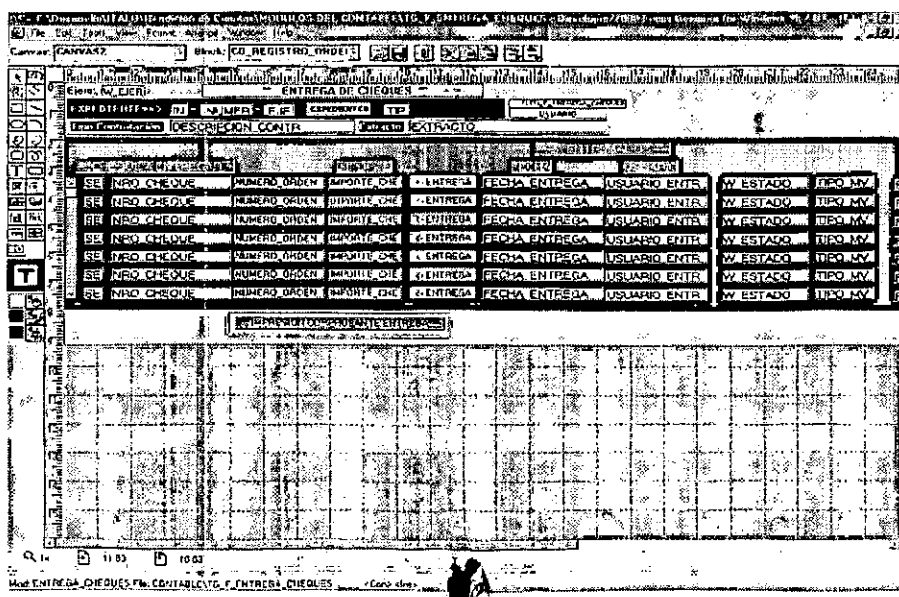
- RENDIBLE: Esta función determina si un expediente es factible a la rendición de cuentas, esta es llamada en el momento que se hace la entrega de un cheque para poder determinar si se genera o no el cargo a un beneficiario. A continuación se puede observar el código fuente de esta rutina.

```
FUNCTION rendible (SEC_OP IN INTEGER) RETURN INTEGER IS
contador integer;
SEC_EXP INTEGER;
BEGIN
       SELECT ro.nro_secuencia_exp
         INTO sec_exp
         FROM co_registro_crdenes ro
        WHERE ro.nro_secuencia = sec_op;

       SELECT count(1)
      INTO contador
      FROM co_expedientes_control ec,
```

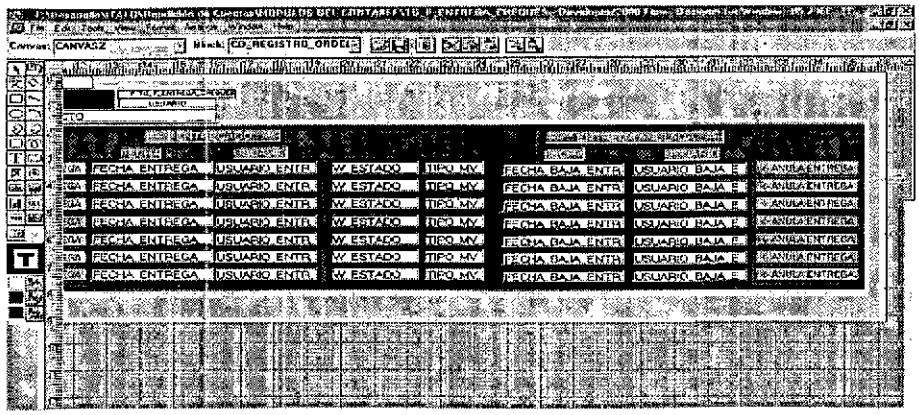
```
            co_expedientes_control_det ecd,
            co_expedientes_ejecucion ee,
            co_expedientes_ejecucion_det eed,
            co_expedientes_tipos_contr etc,
            co_expedientes e,
            ngr_tipos_contr tc,
            ngr_controles_exp ngrc,
            ngr_controles_exp_det ngrcd
     WHERE ngrc.id_control = ec.id_control_exp
       AND ngrcd.nro_secuencia = ecd.nro_secuencia_control_exp_det
       AND eed.nro_secuencia = ec.nro_secuencia_ejecucion_det
       AND ecd.nro_secuencia_exp_control = ec.nro_secuencia
       AND eed.nro_secuencia_ejecucion = ee.nro_secuencia
       AND ec.ok_general = 'SI'
       AND ngrcd.opcion = 'TIT'
       AND ngrcd.rendicion = 'SI'
       AND e.nro_secuencia = SEC_EXP
       AND ee.nro_secuencia_tipo_contratacio = etc.nro_secuencia
       AND etc.nro_secuencia_exp = e.nro_secuencia
       AND etc.id_tipo_contr = tc.id_tipo_contr
       AND e.id_tipo_contr = tc.id_tipo_contr;

    RETURN contador;
END;
```

- TIENE_CARGO: Esta función es la encargada de determinar si el expediente al que pertenece el cheque que se está por anular tiene o no un cargo generado. A continuación se muestra el código fuente perteneciente a esta rutina.

```
FUNCTION TIENE_CARGO (SEC_REG_ORD IN INTEGER) RETURN VARCHAR2 IS
  V_CARGO VARCHAR2(2);
BEGIN
    SELECT REG.CARGO
      INTO V_CARGO
      FROM CO_REGISTRO_ORDENES REG
     WHERE REG.NRO_SECUENCIA = SEC_REG_ORD;

    RETURN V_CARGO;
END;
```



(Figura 15)



(Figura 8.16)

En las figuras 8.15 y 8.16 se observa en el ambiente de desarrollo la pantalla principal del módulo de entregas de cheques. La funcionalidad del sistema de rendición de cuentas incluida en este módulo es totalmente transparente para los usuarios de este módulo.

En la figura 8.15 se observa el botón "<- ENTREGA", dentro de la rutina que se ejecuta cuando se hace click en este botón se incorporó la funcionalidad correspondiente para que se genere un cargo. A continuación vemos el código fuente.

```
DECLARE
  . . . /* otras variables correspondientes al sistema contable */
ENTREGA_CHEQUE          BOOLEAN;

BEGIN

 . . . /* funcionalidad propia del sistema contable que registra la entrega de un cheque */

-- ------------------------------------------------------------------
-- Control de expedientes que tienen que ser rendidos
-- ------------------------------------------------------------------
   IF rendible(:co_registro_orden_detalles.nro_secuencia_op) > 0 then
      genera_cargo(:co_registro_orden_detalles.nro_secuencia_op);
   END IF;
-- ------------------------------------------------------------------

COMMIT_FORM;
END;
```

En esta rutina se puede observar las llamadas a las rutinas RENDIBLE y GENERA_CARGO mencionadas anteriormente.

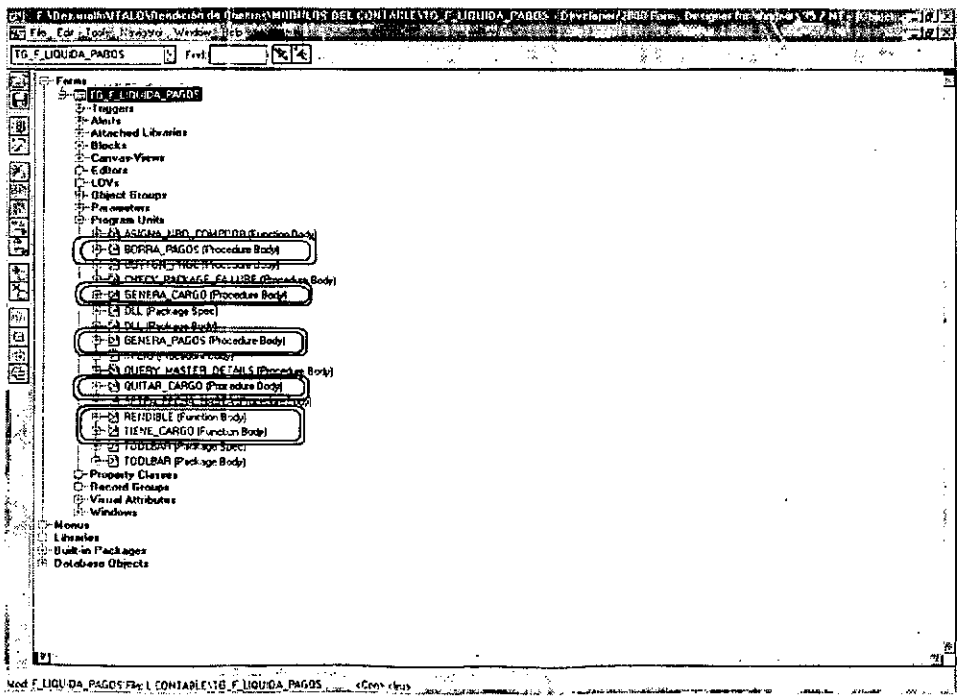
En la figura 8.16 se destaca el botón "<- ANULA ENTREGA". Dentro de la rutina de este botón que se ejecuta cuando se hace click, se incorporó la funcionalidad correspondiente para anular cargos, en el caso de que haya

alguno correspondiente a la orden de pago con la que se hizo entrega del cheque. A continuación veremos este código fuente.

```
DECLARE
 . . . /* Variables propias del sistema contable */
BEGIN

 . . . /* Código fuente propio del sistema contable que anula la entrega de un cheque */

-- ------------------------------------------------------------------
-- Control de expedientes que tienen que ser rendidos
-- hay que quitar el cargo de rendición si se anula la entrega del
-- cheque
-- ------------------------------------------------------------------
   IF tiene_cargo(:co_registro_orden_detalles.nro_secuencia_op) > 0 then
      quitar_cargo(:co_registro_orden_detalles.nro_secuencia_op);
   END IF;
-- ------------------------------------------------------------------
COMMIT_FORM;
END;
```

8.3.6 Módulo de Liquidación de Pagos

Este es el módulo encargado de generar pagos a través de transferencias bancarias. Esta es otra forma que se puede hacer efectivo un pago. Entonces para el caso de expedientes que sean rendidos se deben generar los cargos de rendición correspondientes.



(Figura 8.17)

En la figura 8.17 se muestran con un recuadro en color rojo los nombres de los Program Units (Unidades de Programas) que fueron modificadas (BORRA_PAGOS y GENERA_PAGOS) y los Program Units que fueron agregados en recuadro de color azul (GENERA_CARGO, QUITAR_CARGO, RENDIBLE y TIENE_CARGO).

En el caso de las rutinas nuevas tienen la misma lógica que en el módulo presentado anteriormente, solamente cambia el código fuente ya que no se cuenta con los mismos datos en ambos módulos. Seguidamente se muestra el código fuente perteneciente a cada una de estas rutinas.

GENERA_CARGOS:

```
PROCEDURE genera_cargo(SEC_OP_DET IN INTEGER) IS
  fecha date;
  SEC_REG_ORD  CO_REGISTRO_ORDENES.NRO_SECUENCIA%TYPE;
BEGIN
-- ------------------------------------------------
-- RECUPERO LA FECHA ACTUAL
-- ------------------------------------------------
  select trunc(sysdate) into fecha from dual;

-- ------------------------------------------------
-- RECUPERO EL NUMERO DE SECUENCIA DE LA ORDEN DE PAGO
-- ------------------------------------------------
  SELECT ROD.NRO_SECUENCIA
    INTO SEC_REG_ORD
    FROM CO_REGISTRO_ORDEN_DETALLES ROD
   WHERE ROD.NRO_SECUENCIA = SEC_OP_DET;

-- ------------------------------------------------
-- ACTUALIZO LA ORDEN DE PAGO GENERANDO UN CARGO
-- PARA IDENTIFICAR QUE LA MISMA DEBE SER RENDIDA
-- ------------------------------------------------
  UPDATE CO_REGISTRO_ORDENES REG
    SET REG.CARGO = 'SI',
     REG.FECHA_CARGO = fecha,
     REG.FECHA_VTO_CARGO = fecha + 30
  WHERE REG.NRO_SECUENCIA = SEC_REG_ORD;
END;
```

QUITAR_CARGOS:

```
PROCEDURE QUITAR_CARGO (SEC_OP_DET IN INTEGER) IS
SEC_OP CO_REGISTRO_ORDENES.NRO_SECUENCIA%TYPE;
BEGIN

-- ------------------------------------------------
-- RECUPERO EL NUMERO DE SECUENCIA DE LA ORDEN DE PAGO
-- ------------------------------------------------
      SELECT ROD.NRO_SECUENCIA
        INTO SEC_OP
        FROM CO_REGISTRO_ORDEN_DETALLES ROD
       WHERE ROD.NRO_SECUENCIA = SEC_OP_DET;

-- ------------------------------------------------
-- QUITA EL CARGO DE RENDICION DE LA ORDEN DE
-- ------------------------------------------------

  UPDATE CO_REGISTRO_ORDENES REG
    SET REG.CARGO = NULL,
     REG.FECHA_CARGO = NULL,
     REG.FECHA_VTO_CARGO = NULL
```

```
  WHERE REG.NRO_SECUENCIA = SEC_OP;
END;
```

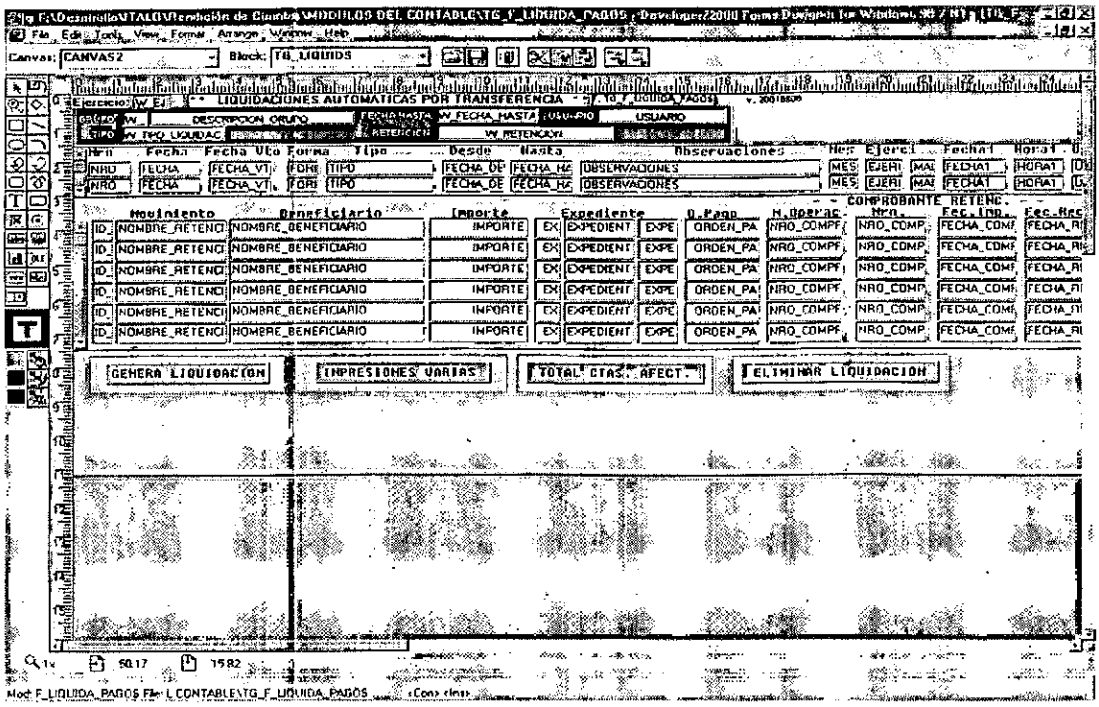
RENDIBLE:

```
FUNCTION rendible (SEC_OP_DET IN INTEGER) RETURN INTEGER IS
contador   integer;
SEC_EXP    CO_EXPEDIENTES.NRO_SECUENCIA%TYPE;
SEC_OP     CO_REGISTRO_ORDENES.NRO_SECUENCIA%TYPE;
BEGIN

-- ------------------------------------------------------
-- RECUPERO EL NUMERO DE SECUENCIA DE LA ORDEN DE PAGO
-- ------------------------------------------------------
      SELECT ROD.NRO_SECUENCIA
        INTO SEC_OP
        FROM CO_REGISTRO_ORDEN_DETALLES ROD
       WHERE ROD.NRO_SECUENCIA = SEC_OP_DET;

-- ------------------------------------------------------
-- RECUPERO EL NUMERO DE SECUENCIA DEL EXPEDIENTE DE PAGO
-- ------------------------------------------------------
       SELECT ro.nro_secuencia_exp
         INTO sec_exp
         FROM co_registro_ordenes ro
        WHERE ro.nro_secuencia = sec_op;

-- ------------------------------------------------------
-- CONTROLO SI EL EXPEDIENTE ES FACTIBLE DE RENDICION
-- ------------------------------------------------------
       SELECT count(1)
      INTO contador
      FROM co_expedientes_control ec,
               co_expedientes_control_det ecd,
               co_expedientes_ejecucion ee,
               co_expedientes_ejecucion_det eed,
               co_expedientes_tipos_contr etc,
               co_expedientes e,
               ngr_tipos_contr tc,
               ngr_controles_exp ngrc,
               ngr_controles_exp_det ngrcd
     WHERE ngrc.id_control = ec.id_control_exp
       AND ngrcd.nro_secuencia = ecd.nro_secuencia_control_exp_det
       AND eed.nro_secuencia = ec.nro_secuencia_ejecucion_det
       AND ecd.nro_secuencia_exp_control = ec.nro_secuencia
       AND eed.nro_secuencia_ejecucion = ee.nro_secuencia
       AND ec.ok_general = 'SI'
       AND ngrcd.opcion = 'TIT'
       AND ngrcd.rendicion = 'SI'
       AND e.nro_secuencia = SEC_EXP
       AND ee.nro_secuencia_tipo_contratacio = etc.nro_secuencia
       AND etc.nro_secuencia_exp = e.nro_secuencia
       AND etc.id_tipo_contr = tc.id_tipo_contr
       AND e.id_tipo_contr = tc.id_tipo_contr;

     RETURN contador;
END;
```

TIENE_CARGO:

```
FUNCTION TIENE_CARGO (SEC_OP_DET IN INTEGER) RETURN VARCHAR2 IS
  V_CARGO       VARCHAR2(2);
  SEC_OP        CO_REGISTRO_ORDENES.NRO_SECUENCIA%TYPE;
BEGIN

-- ------------------------------------------------------
-- RECUPERO EL NUMERO DE SECUENCIA DE LA ORDEN DE PAGO
-- ------------------------------------------------------
      SELECT ROD.NRO_SECUENCIA
        INTO SEC_OP
        FROM CO_REGISTRO_ORDEN_DETALLES ROD
       WHERE ROD.NRO_SECUENCIA = SEC_OP_DET;

-- ------------------------------------------------------
-- RECUPERO SI LA ORDEND DE PAGO TIENE UN CARGO GENERADO
-- ------------------------------------------------------
    SELECT REG.CARGO
      INTO V_CARGO
      FROM CO_REGISTRO_ORDENES REG
     WHERE REG.NRO_SECUENCIA = SEC_OP;

    RETURN V_CARGO;
```

```
END;
```

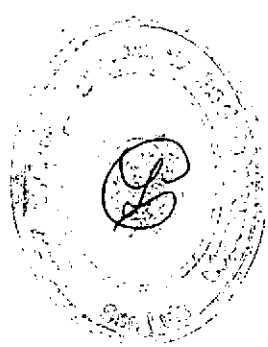


(Figura 8.18)

En la figura 8.18 se pueden observar los botones "GENERA LIQUIDACION" y "ELIMINAR LIQUIDACION", estos son los que llaman a las rutinas que fueron modificadas para incorporarle funcionalidad del sistema de rendiciones de cuentas. A continuación se muestra el código fuente relevante al sistema de rendición de cuentas embebido en estas rutinas.

GENERA_PAGOS:

```
PROCEDURE GENERA_PAGOS IS
 . . . /* Declaraciones propias del sistema contable */
BEGIN
. . . /* Código perteneciente al sistema contable */
LOOP
 . . . /* Código perteneciente al sistema contable */
-- ------------------------------------------------------------------------------------------
-- CONTROL DE EXPEDIENTES QUE TIENEN QUE SER RENDIDOS
-- ------------------------------------------------------------------------------------------
   IF rendible(D_NRO_SECUENCIA) > 0 then
      genera_cargo(D_NRO_SECUENCIA);
   END IF;
-- ------------------------------------------------------------------------------------------
. . . /* Código perteneciente al sistema cont      */
END LOOP;
. . . /* Código perteneciente al sistema cont      */
EXCEPTION
. . . /* Código perteneciente al sistema conta     de maneja las posible excepciones que
         puedan ocurrir*/
END;
```

BORRA_PAGOS:

```
PROCEDURE BORRA_PAGOS IS
. . . /* Declaraciones propias del sistema contable */
BEGIN
. . . /* Código perteneciente al sistema contable */
-- ------------------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------------------
-- QUITAR EL CARGO DE RENDICION DE AQUELLAS OP QUE TIENEN UN CARGO PENDIENTE
-- ------------------------------------------------------------------------------------------
LOOP   -- CICLA POR TODO EL CURSOR

    FETCH C_REG_OP_DET INTO V_REG_OP_DET;--CARGA EL REGISTRO
    EXIT WHEN C_REG_OP_DET%NOTFOUND; -- SALE DEL CICLO CUANDO SE ACABA EL CURSOR

    IF rendible(V_REG_OP_DET.NRO_SECUENCIA) > 0 then
       genera_cargo(V_REG_OP_DET.NRO_SECUENCIA);
    END IF;

END LOOP;
CLOSE C_REG_OP_DET; -- CIERRA EL CURSOR
-- ------------------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------------------
. . . /* Código perteneciente al sistema contable */

EXCEPTION
. . . /* Código perteneciente al sistema contable que maneja las posible excepciones que
         puedan ocurrir*/
END;
```