

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades

Departamento de Ciencias de la Computación

**ANALIZADOR DE ESTRUCTURA DE CÓDIGO
APLICADO AL LENGUAJE DE PROGRAMACIÓN
PL/SQL**



PAULO VLADIMIR MEJÍA CASTILLO

Trabajo de graduación presentado para optar al grado académico de


Maestría en Tecnología y Ciencias de la Computación

Guatemala

2003

Vo Bo:

(f)



Asesor: Ingeniero Jorge Arturo Rivera Perezgil, MSE


Tribunal:

(f)



Ingeniero Patricia Castillo Avila, MSC

(f)



Ingeniero Luis Eduardo Masaya Arias, MSC

(f)



Ingeniero Jorge Arturo Rivera Perezgil, MSE

Fecha de aprobación 19 de mayo de 2003

RESUMEN EJECUTIVO

La mayoría de empresas que cuenta con un equipo dedicado al mantenimiento de *software* invierten mucho tiempo en el desarrollo de los cambios y los resultados pueden no ser los esperados. Para atacar este inconveniente se plantea contar con una herramienta que sea capaz de integrar el análisis de dependencias y permita visualizar las dependencias adecuadamente, para determinar el impacto que tiene la realización de los cambios, y poder establecer los recursos y el tiempo necesario para llevar su implementación.

En el presente trabajo se plasma la arquitectura de esta herramienta, se escogen elementos de análisis seleccionados del lenguaje de programación PL/SQL y se propone un procedimiento que sirve de complemento a las metodologías que las empresas ya tienen. Se documenta el uso de esta herramienta con una lista de requerimientos de una aplicación real. Para utilizar esta herramienta se requiere que se tenga el licenciamiento correspondiente.

Aunque el lenguaje de programación que se utilizó es PL/SQL, la arquitectura es flexible para la incorporación de nuevos lenguajes o el nivel de detalle de información que se necesite analizar, para ello hay que elaborar un componente de esta arquitectura específico a esta tarea.

TABLA DE CONTENIDOS

RESUMEN EJECUTIVO	IX
I. INTRODUCCIÓN	1
II. OBJETIVOS	3
A. GENERAL	3
B. ESPECÍFICOS	3
III. MARCO TEÓRICO	5
A. CONCEPTO DE REPOSITORIO	5
B. CONCEPTO DE METADATA	5
C. LENGUAJE DE CONSULTA ESTRUCTURADO (SQL)	5
1. <i>Lenguaje de definición de datos</i>	5
2. <i>Lenguaje de manipulación de datos</i>	7
3. <i>Lenguaje de control de datos</i>	8
D. ANALIZADORES DE ESTRUCTURAS DE CÓDIGO	9
1. <i>Dependencias de datos</i>	10
2. <i>Dependencias de control</i>	10
3. <i>Dependencias de componente</i>	10
E. ANÁLISIS DE IMPACTO	11
F. LENGUAJE DE PROGRAMACIÓN PL/SQL	12
1. <i>Características principales:</i>	14
2. <i>Arquitectura:</i>	16
IV. DESARROLLO DE LA PROPUESTA	17
A. ARQUITECTURA DE LA HERRAMIENTA DESARROLLADA	17
1. <i>Componentes</i>	17
2. <i>Interacción entre los componentes</i>	18
B. ELEMENTOS DE ANÁLISIS SELECCIONADOS	19
1. <i>Tablas</i>	20
2. <i>Columnas</i>	21
3. <i>Vistas</i>	21
4. <i>Procedimientos</i>	22
5. <i>Funciones</i>	23
6. <i>Triggers</i>	24
7. <i>Paquetes</i>	25
8. <i>Tipos de relaciones entre los elementos de análisis</i>	26
C. IMPLEMENTACIÓN DEL CLIENTE DE CARGA	29
1. <i>Repositorio de carga</i>	29
2. <i>Repositorio de consulta</i>	30

3. <i>Implementación del cliente de carga</i>	35
D. IMPLEMENTACIÓN DEL CLIENTE DE ANÁLISIS	40
1. <i>Tipos de consultas de los elementos de análisis seleccionados</i>	40
2. <i>Implementación del cliente de análisis</i>	42
3. <i>Tipos de análisis de dependencias que se pueden realizar con base en las consultas</i>	54
E. CASO DE ESTUDIO	61
1. <i>Ambiente</i>	61
2. <i>Requerimientos</i>	62
3. <i>Aplicación de la herramienta</i>	63
V. RESULTADOS Y DISCUSIÓN	65
A. ELABORACIÓN DE CLIENTES DE CARGA	65
B. IMPORTANCIA DE LA METADATA	66
C. CLIENTE DE ANÁLISIS GENÉRICO	67
D. APLICACIÓN DE LA HERRAMIENTA CON BASE EN LOS REQUERIMIENTOS	68
1. <i>Requerimiento uno</i>	69
2. <i>Requerimiento dos</i>	71
3. <i>Requerimiento tres</i>	72
4. <i>Requerimiento cuatro</i>	73
5. <i>Requerimiento cinco</i>	74
6. <i>Requerimiento seis</i>	76
E. REDUCCIÓN DE TIEMPOS POR EL USO DE LA HERRAMIENTA	78
F. PRODUCTIVIDAD EN EL USO DE LA HERRAMIENTA	79
1. <i>Procedimiento propuesto</i>	79
2. <i>Consejos para la realización de consultas</i>	82
3. <i>Complementar con otras herramientas</i>	83
4. <i>Apoyo por parte de la administración</i>	83
VI. RECOMENDACIONES	85
VII. RECONOCIMIENTOS	89
VIII. REFERENCIAS BIBLIOGRÁFICAS	91
IX. APÉNDICES	93
A. INSTALACIÓN DE LA HERRAMIENTA	93
B. LISTADO DE ABREVIATURAS	95

I. INTRODUCCIÓN

El proyecto consiste en la elaboración de una herramienta y la aplicación de la misma a un caso real. La herramienta es un analizador de estructuras de código del lenguaje de programación PL/SQL. PL/SQL es un lenguaje propietario de la Corporación Oracle. Para el desarrollo de la herramienta propuesta se seleccionó un conjunto de elementos de análisis, en el PL/SQL, y se registraron sus propiedades y dependencias.

Los elementos se obtienen de un repositorio de archivos texto de documentación. Este repositorio es generado por herramientas externas. Por ejemplo, en el caso de los elementos de la base de datos se utilizó una herramienta que genere archivos que contengan las instrucciones de creación de estos elementos.

Con base en los elementos de análisis registrados, la herramienta elaborada permitirá generar vistas relacionales, textuales y gráficas. Estas vistas pueden ser utilizadas para realizar un análisis de dependencias de los elementos seleccionados. Las vistas gráficas se implementaron por medio de la herramienta de modelación Visio 2002. Tanto los elementos de análisis con sus relaciones, como sus diferentes vistas fueron documentados en este informe.

Se tomó una aplicación real para construir el repositorio. Este repositorio se aplicó en la herramienta desarrollada, para documentarse como un caso de estudio en este informe, con base en los requerimientos propuestos para dicha aplicación.

La problemática que se pretende abordar con el desarrollo de la propuesta anterior es la siguiente: En la mayoría de las empresas que cuenta con un equipo a cargo de actividades de desarrollo y mantenimiento de software, la metodología utilizada para realizar los cambios a una aplicación es la de intuición o fuerza bruta, se hace una revisión exhaustiva de todos los archivos fuentes en el repositorio, o bien se revisan algunos archivos fuentes, elegidos de manera subjetiva. Esto se debe a que con frecuencia los documentos de diseño no están disponibles, o bien han quedado desactualizados. En otros casos, se cuenta con herramientas que permiten reconstruir la información de sistema hasta cierto nivel, pero los detalles útiles son ignorados o la forma de visualización no es la adecuada. Por lo tanto, se invierte mucho tiempo en el desarrollo de los cambios y los resultados pueden no ser los esperados. Además, resulta difícil predecir el impacto de los cambios.

La propuesta resuelve la problemática anterior de la siguiente manera: El equipo de desarrollo y mantenimiento de software debe conocer adecuadamente la estructura del sistema sobre el cual se está trabajando para poder realizar los cambios adecuadamente y en un tiempo razonable. Se plantea, por lo tanto, la conveniencia de contar con una herramienta con la capacidad de analizar el código del sistema y presentar información acerca de las relaciones entre elementos seleccionados de forma tal que pueda ser útil para las personas que necesitan trabajar en ello. Se pretende, por lo tanto, incorporar el análisis de dependencias en las prácticas de la empresa al momento de realizar los cambios de una aplicación. Utilizando diferentes vistas de las dependencias se puede obtener una mejor comprensión de la aplicación a la hora de realizar los cambios a la misma y poder evaluar, de manera objetiva, si se pueden realizar los cambios en los tiempos estipulados.

II. OBJETIVOS

A. General

- Incorporar el análisis de dependencias entre elementos referidos en el código de PL/SQL de una aplicación antes de la realización de los cambios en la misma.

B. Específicos

- Mejorar la visualización y comprensión de los cambios que se realizarán en una aplicación.
- Desarrollar una herramienta para uso de la comunidad.
- Integrar la herramienta a desarrollar con la herramienta de modelación Visio 2002.
- Documentar la aplicación.
- Mejorar los tiempos de respuesta y la calidad de los resultados después de realizados los cambios.

III. MARCO TEÓRICO

A. Concepto de repositorio

El repositorio es una colección de archivos. Los cuales están centralizados y pueden ser compartidos por numerosos usuarios para diversas aplicaciones.

B. Concepto de metadata

La metadata es la información descriptiva de los datos con el fin de comprender la forma y la estructura de los datos mismos (Kendal y Kendal, 1991:667).

C. Lenguaje de consulta estructurado (SQL)

SQL es el lenguaje comúnmente usado para consultas en los sistemas de base de datos relacionales y modificación de la base de datos. Este lenguaje es equivalente al álgebra relacional; sin embargo, contiene características que van más allá de lo encontrado en el álgebra relacional, por ejemplo funciones agregadas y actualizaciones de la base de datos.

Hay varios dialectos de SQL. Primero, hay dos estándares: ANSI (*American National Standards Institute*) SQL y un estándar actualizado adoptado en 1992, llamado SQL-92 o SQL2. Además hay un tercer estándar emergiendo llamado SQL3, que extiende SQL2 con muchas nuevas características tales como recursión, *triggers*, y objetos. Por lo tanto, hay versiones de SQL producidos por los principales vendedores de sistemas de administración de base de datos.

El lenguaje SQL puede ser dividido en las siguientes áreas por la funcionalidad de los enunciados y serán descritos en las próximas secciones:

- Lenguaje de definición de datos (DDL).
- Lenguaje de manipulación de datos (DML).
- Lenguaje de control de datos (DCL).

1. Lenguaje de definición de datos. Los enunciados de SQL que conforman esta área intervienen en el diseño físico y mantenimiento de los objetos de la base de datos. Por ejemplo, las instrucciones para la creación de objetos son *CREATE*, la instrucción

para la modificación de los objetos o establecimiento de restricciones es *ALTER* y la instrucción para la eliminación de objetos es *DROP*.

El DBMS de SQL2 crea un ambiente de SQL donde la información puede existir y operaciones de SQL sobre la información pueden ser ejecutadas. Dentro del ambiente, los elementos de la base de datos tales como relaciones son agrupadas en esquemas, catálogos y arreglos. Un esquema es una colección de objetos, un catálogo es una colección de esquemas, y un arreglo es la colección más grande de elementos que un usuario puede ver. Además, una sesión es una conexión activa donde operaciones de SQL pueden ser ejecutadas. Cada sesión tiene un catálogo actual y un esquema actual dentro de aquel catálogo. Hay también usuarios autorizados para cada sesión.

SQL2 provee una variedad de técnicas para expresar restricciones de integridad como parte de los esquemas de base de datos. Entre las restricciones que se pueden definir están: llaves primarias, integridad referencial (llaves foráneas), chequeo de reglas a nivel de atributos, tuplas, relaciones, o base de datos.

El estándar propuesto de SQL3 incluye *triggers* que especifican ciertos eventos (por ejemplo: inserción, eliminación, o actualización de una particular relación) que los llaman. Una vez disparado, una condición puede ser chequeada, y si es verdadera, una especificada secuencia de acciones (enunciados SQL tales como consultas y modificaciones de la base de datos) serán ejecutadas.

En SQL2 se pueden declarar una aserción como un elemento de un esquema de base de datos con una condición a ser chequeada. Esta condición puede involucrar una o más relaciones del esquema de base de datos o partes de las relaciones. En SQL3 se incluye una noción de aserción diferente de la aserción de SQL2, ya que las aserciones son disparadas por eventos especificados, en vez de la violación de alguna restricción decidida por el sistema.

Hay dos clases de objetos en SQL3: tipos fila y tipos de datos abstractos. Los tipos fila son tipos para tuplas, y los tipos de datos abstractos son tipos para componentes de tuplas. También hay un tipo de referencia para cada tipo de fila, y un valor de su tipo de referencia es el identificador del objeto para una tupla.

SQL3 permite a un atributo tener un tipo que sea una referencia al tipo fila de su propia relación y estos valores de atributos formarán el identificador objeto para la tupla

en la cual éste reside, entonces permite a este identificador de objeto servir como un atributo llave para su relación. Se pueden declarar un ADT en SQL3 con un enunciado *CREATE TYPE*. Los valores de un ADT son registrados en estructuras con uno o más componentes y pueden tener asociados métodos. Los métodos pueden ser declarados para un ADT y estos métodos se pueden escribir en un lenguaje de programación parecido a SQL o ser declarados como funciones externas en un lenguaje anfitrión.

2. Lenguaje de manipulación de datos. Los enunciados de SQL que conforman esta área se utilizan para consultar, modificar, agregar y eliminar información de las relaciones base (llamadas tablas). Por ejemplo, la instrucción para consulta de información es *SELECT FROM WHERE* o *CURSOR*, la instrucción para agregar información es *INSERT INTO VALUE* o *INSERT INTO SELECT*, la instrucción para actualizar información es *UPDATE SET WHERE* y la instrucción para eliminar información es *DELETE FROM WHERE*.

El enunciado *SELECT* tiene una variedad de características. Por ejemplo, se pueden realizar ordenamientos, calcular funciones agregadas, puede ser combinado con operadores de conjuntos, eliminar duplicados, hacer diferentes tipos de *joins*, y puede ser utilizado como subconsulta (consulta que puede ser utilizada en otra consulta).

SQL provee un *outer join* que junta las relaciones que cumplen ciertas condiciones pero también incluye en el resultado las tuplas que no cumplan la condición de una o ambas relaciones. Las columnas de estas tuplas no encontradas se le colocan valores NULL en la relación resultante. Hay variaciones sobre los *outerjoins* disponibles en SQL2. La primera variación es que además del *full outer join* se pueden tener *left outerjoin* y *right outerjoin*, y la segunda variación es la forma como se especifica la condición de comparación que las tuplas deben cumplir.

En SQL2 un módulo es un programa para una aplicación. Se tienen tres clases de módulos: Interfase de SQL genérica donde el usuario escribe los enunciados que son ejecutados por un servidor de SQL, SQL sumergido donde los enunciados de SQL aparecen dentro de un programa de un lenguaje anfitrión y son ingresados en el comando *EXEC SQL*, y el tercer módulo donde hay una colección de funciones o procedimientos que están en el lenguaje anfitrión o son enunciados de SQL, los cuales se comunican

pasándose parámetros y tal vez vía variables compartidas. La ejecución de un módulo se llama un agente SQL.

En vez de usar una interfase de consulta genérica para expresar consultas y modificaciones de SQL, es frecuentemente más efectivo escribir programas que sumerjan consultas de SQL en un lenguaje anfitrión convencional. En el SQL sumergido se tiene una sección de declaración de variables compartidas. También se pueden declarar cursores donde pueden tener la opción de protección contra actualizaciones. En implementaciones de SQL2 es requerido el soporte de, al menos, los siguientes siete lenguajes anfitriones: Ada, C, Cobol, Fortran, Mumps, Pascal y PL/I.

Para no sumergir enunciados particulares de SQL en un lenguaje anfitrión de un programa, se puede utilizar SQL dinámico donde se crea una cadena de caracteres que contenga el enunciado SQL para que lo interprete y ejecute el sistema SQL.

El estándar de SQL3 incluye una manera de definir temporalmente relaciones recursivas y usar estas relaciones en consultas. El estándar propuesto para relaciones recursivas requiere que la relación tenga recursión lineal y cumpla con cierto tipo de restricciones de negación y agregación.

3. Lenguaje de control de datos. Los enunciados de SQL que conforman esta área se utilizan para la administración de privilegios, transaccionalidad y manejo de niveles de aislamiento. Por ejemplo, las instrucciones para administrar privilegios son *GRANT TO* y *REVOKE FROM*, las instrucciones para manejar transaccionalidad son *COMMIT* y *ROLLBACK*, y la instrucción para manejar niveles de aislamiento es *SET TRANSACTION ISOLATION LEVEL*.

SQL2 provee dos mecanismos para prevenir la interferencia de operaciones concurrentes: transacciones y restricciones sobre cursores. Las restricciones sobre cursores incluyen la capacidad de declarar un cursor a ser no sensitivo, en este caso ningún cambio de su relación será visto por el cursor. Con respecto a las transacciones, SQL permite a los programadores agrupar enunciados SQL en transacciones, las cuales pueden ser comprometidas o abortadas. En el caso de ser comprometida, todos los cambios de la transacción serán hechos en la base de datos y en el caso de abortada la

transacción , cualquier cambio hecho por la transacción en la base de datos será cancelado.

SQL2 permite a las transacciones ejecutarse en cuatro niveles de aislamiento llamados, del más severo al menos severo: serializable (las transacciones deben ejecutarse una de trás de otra), lectura repetible (requiere que entre dos lecturas de una tupla en una transacción no se permita la modificación de esta tupla por otra transacción y sólo se permite leer tuplas comprometidas), lectura comprometida (sólo tuplas escritas por transacciones que se han comprometido pueden ser vistas por las otras transacciones), y lectura no comprometida (ninguna restricción puede verse en cualquier transacción).

Para propósitos de seguridad, el sistema SQL2 permite diferentes clases de privilegios a ser obtenidas sobre elementos de base de datos. Estos privilegios incluyen derechos para seleccionar (lectura), insertar, eliminar, o actualizar relaciones y los permisos para referenciar relaciones (referenciar a ellos en una restricción) o hacer una conexión. Los privilegios de inserción, actualización y referencia también pueden ser obtenidos en particular sobre columnas de la relación. Los privilegios pueden ser concedidos por los dueños a otros usuarios o al usuario general *PUBLIC*. Si concede con la opción de permitir dar accesos, entonces estos privilegios pueden ser pasados a otros. Los privilegios también pueden ser revocados.

Los beneficios de la estandarización de SQL son:

- Reducción en costos de entrenamiento.
- Productividad.
- Portabilidad de las aplicaciones.
- Longevidad de las aplicaciones.
- Reducción de la dependencia de los vendedores.
- Comunicación cruzada de sistemas.

D. Analizadores de estructuras de código

Los analizadores de estructuras de código son herramientas de *software* que detectan y capturan información de dependencias, evaluando dependencias de estructura de datos, flujo de control, y componentes o módulos. Por lo general, el analizador está orientado específicamente para un lenguaje de programación (Bohner y Arnold,1996:13).

Entre los tipos de dependencias que puede evaluar un analizador se encuentran:

1. Dependencias de datos. Son relaciones entre enunciados del programa que definen o usan datos. Una dependencia de datos existe cuando un enunciado provee un valor usado directamente o indirectamente por otro enunciado en un programa. La definición de datos y el uso de grafos son representaciones típicas de estas dependencias (Bohner y Arnold,1996;14).

El análisis de flujo de datos produce información de dependencias sobre los datos que van en el sistema de *software*. Esto identifica agregados de datos y los subelementos que ellos representan. Las técnicas de flujo de datos analizan el código fuente para determinar caminos y transformaciones que la información puede tomar a través del sistema. Estos caminos son típicamente representados como grafos o tablas que ayudan a los ingenieros de *software* para entender los flujos claves de información (Bohner y Arnold,1996:14).

2. Dependencias de control. Son relaciones entre enunciados de programa que controlan la ejecución del programa. La idea principal es de qué manera los enunciados relacionados pueden afectar la ejecución del programa. Los grafos de dependencias del programa son frecuentemente usados para explicar las dependencias de control (Bohner y Arnold,1996:14).

El análisis del flujo de control provee información sobre puntos de decisiones lógicas en el sistema de *software* y la complejidad de sus estructuras. Las técnicas del flujo de control identifican dependencias de llamadas a procedimientos, decisiones lógicas (tales como enunciados *IF THEN ELSE*, *LOOPS*, *CASE*), y bajo qué condiciones cumplen su veracidad. Los ingenieros de software pueden utilizar estas técnicas para analizar el código fuente y determinar los caminos de control a través del sistema. Estos caminos son típicamente representados como grafos o tablas para ayudar a entender el flujo de control del *software* (Bohner y Arnold,1996:14).

3. Dependencias de componente. Son relaciones generales entre componentes de código fuente tales como módulos, archivos y evaluaciones de corrida. Muchas técnicas para detectar dependencia de componente son apoyadas por herramientas de

ingeniería de *software* tales como referencia cruzada, analizadores de evaluación de cobertura, y comparación de código fuente (Bohner y Arnold,1996:15).

La referencia cruzada ayuda a encontrar impactos por consultar la información de *software* capturada del código fuente. Como la información trae todos los objetos referenciados, ayudará al ingeniero de *software* a entender las relaciones entre los artefactos de *software*. La referencia cruzada dice qué enunciados del código fuente aparecen ser afectados por las variables del programa. El ingeniero de *software* puede ir a las áreas y continuar revisando si hay impactos potenciales. La referencia cruzada usada en conjunción con herramientas visuales de código fuente es efectiva para identificar dependencias de componentes (Bohner y Arnold,1996:15).

Los analizadores de evaluación de cobertura identifican las partes del sistema que fueron ejecutadas durante pruebas e indican en dónde fue gastado el tiempo de procesamiento. Estos analizadores pueden obtener estadísticas en tiempo de corrida durante la ejecución de un programa. Entonces, ellos son altamente efectivos para ayudar a los encargados de hacer las pruebas para determinar las áreas que hay que cubrir con los casos de pruebas. Éste provee una base para el análisis de impacto (Bohner y Arnold,1996:15).

La comparación de código fuente ayuda a determinar cambios en el código fuente después de que estos han sido hechos. Por ejemplo, dos personas pueden tener que cambiar el mismo programa al mismo tiempo por diferentes razones. Para determinar en dónde los cambios, se pueden traslapar, los ingenieros de *software* pueden usar comparaciones de código fuente para identificar los enunciados cambiados en ambos casos (Bohner y Arnold,1996:15).

E. Análisis de impacto

El análisis de impacto es la identificación de consecuencias potenciales de un cambio, o estimación de lo que necesita ser modificado para lograr un cambio. Un impacto es una parte determinada que se ve afectada, y por lo tanto de importancia para ser inspeccionada. Esta definición enfatiza la estimación del impacto (Bohner y Arnold,1996:3).

El análisis de impacto precede, o es usado en conjunción con la realización de cambios. El análisis de impacto estático analiza el código fuente sin ejecutar el código. El análisis de impacto dinámico ejecuta el *software* como un asistente para determinar los impactos dentro de éste (Bohner y Arnold,1996:3).

Las mayores áreas para análisis de impacto son: análisis de dependencias y análisis de rastreabilidad. El análisis de dependencias involucra la revisión detallada de las relaciones de dependencia entre entidades del programa (variables, módulos, lógica, etc). Éste provee una evaluación detallada de dependencias a bajo nivel en el código. El análisis de rastreabilidad involucra la revisión de relaciones de dependencia entre la lista de todos los objetos del ciclo de vida del software. Como tal, se dirige al análisis de impacto desde una perspectiva más amplia. Por ejemplo, ésta puede relacionar artefactos de software (tales como requerimientos) con componentes de diseño asociados. Aunque la rastreabilidad cubre muchas de las relaciones entre artefactos que una librería de proyecto de *software* o repositorio puede almacenar, estas relaciones típicamente no son muy detalladas (Bohner y Arnold,1996:3).

En resumen, se puede determinar impactos del análisis de dependencia, ya que si existen dependencias entre los elementos, la alteración o modificación de un elemento impactará directa o indirectamente a otros elementos.

F. Lenguaje de programación PL/SQL

PL/SQL es un lenguaje de programación estructurado de la base de datos, extensión de SQL, propiedad de Oracle, disponible en todas las herramientas de desarrollo de Oracle. Basado en el lenguaje de programación Ada, es fácil de aprender, bien estructurado, y extensible. Los módulos de ejecución son escritos en C y los procedimientos son almacenados en una forma compilada, esto hace un lenguaje muy rápido y optimizado para tareas orientadas a bases de datos.

PL/SQL combina el poder de manipulación de SQL con el poder de procesamiento de un lenguaje de programación procedural. Sirve de puente entre la tecnología de base de datos y lenguajes de programación procedural.

PL/SQL se ha desarrollado en un lenguaje de programación poderoso y robusto. Este contiene muchos de los elementos más avanzados de un lenguaje de programación

procedural (estructuras de control, tipos de datos definidos por el usuario, etc...) y, tal vez lo más importante, éste tiene un registro de rastreo mejorado en la programación de la base de datos de Oracle. Esto se debe porque por un largo tiempo, los lenguajes tradicionales como C ó C++ tenían muchas deficiencias de funcionalidad. PL/SQL ha limitado sus capacidades a interactuar con el sistema operativo sobre el cual la base de datos está corriendo. Además, éste no ofrece capacidades de red o capacidades para comunicarse con cualquier clase de aplicación no Oracle que resida sobre otra máquina. Sin embargo, versiones recientes del lenguaje han empezado a dirigirse sobre alguno de estos puntos:

PL/SQL 8 tiene muchas nuevas características del lenguaje y construcciones, tales como tablas anidadas y arreglos de tamaño variables, y PL/SQL ha ido evolucionando hasta ser totalmente orientado a objetos. Éste ha progresado más allá del soporte de objetos tradicional de base de datos tales como funciones, *triggers*, procedimientos, y *packages* para contener la capacidad de definición de tipos de objetos.

PL/SQL tiene varias librerías de funciones tales como *utl_file*, esta librería facilita el manejo de archivos que residen el sistema operativo.

Oracle ha soportado desarrollos de *web* con PL/SQL siempre desde la primera liberación de su servidor de *web* y surgimiento de las librerías de *web* de PL/SQL. Estas librerías son un conjunto de paquetes que contienen funciones y procedimientos que mapean a comandos de HTML. Además, sirve de puente entre los programas de base de datos y los programas de *internet*. Los procedimientos son mantenidos en la base de datos, en forma compilada, y se obtiene vía un URL que referencia a un camino donde se encuentra un agente de PL/SQL. Estos procedimientos pueden acceder directamente la información de base de datos a través de cursores y formatear la información para desplegarse en el *web* usando los comandos de la librería.

Las ventajas de utilizar el lenguaje de programación PL/SQL son:

- Soporte para SQL.
- Soporte para programación orientada a objetos.
- Mejor rendimiento.
- Alta productividad.
- Totalmente portable.

- Estrecha integración con Oracle.
- Estrecha seguridad.

Las nuevas características incorporadas al lenguaje de programación PL/SQL son:

- SQL dinámico nativo.
- Manejo de HTML.
- Manejo de Java.
- Manejo de XML.
- Manejo de archivos en el sistema operativo.
- Encriptación.

Las herramientas que utilizan el lenguaje de programación PL/SQL son:

- Forms (herramienta para el desarrollo de mantenimientos y consultas).
- Reports (herramienta para el desarrollo de reportes).
- Plus (herramienta para el desarrollo de consultas interactivas y administración).
- Portal (herramienta para la implementación de portales).
- TOAD (herramienta de administración de base de datos Oracle).
- Navigator (herramienta para el desarrollo de administración y consultas).
- Builder (herramienta para el desarrollo y mantenimiento de objetos PL/SQL).

1. Características principales.

a. Uso de instrucciones de SQL. Las instrucciones de SQL se utilizan en la manipulación de la información de la base de datos. En PL/SQL la integración de las instrucciones es de forma nativa.

b. Uso de instrucciones de flujo de control. Las instrucciones de flujo de control pueden manipular información de la base de datos y procesar la información utilizando enunciados de flujo de control condicionales, iterativos y secuenciales tales como *IF-THEN-ELSE*, *FOR-LOOP*, *WHILE-LOOP*, *EXIT-WHEN*, y *GOTO*.

c. Lenguaje estructurado por bloques. Un programa de PL/SQL está conformado por unidades básicas. Estas unidades son los bloques lógicos. Los bloques lógicos pueden contener cualquier número de subbloques anidados. Típicamente, cada bloque lógico

corresponde a un problema o subproblema que resolver. Por lo tanto, PL/SQL soporta el paradigma de divide y conquista para resolución de problemas.

d. Definición de constantes, variable, funciones, procedimientos y cursores.

PL/SQL permite declarar constantes y variables que posteriormente se utilicen en instrucciones SQL, instrucciones procedurales o formen expresiones.

Oracle utiliza áreas de trabajo para ejecutar instrucciones SQL y mantener guardada la información. En PL/SQL, un cursor nombra el área de trabajo y permite acceder la información guardada. Hay cursores implícitos y explícitos. Los explícitos son los declarados y los implícitos son los cursores que se forman al ejecutar una instrucción SQL. En los cursores explícitos también hay cursores variables. Estos son cursores que en tiempo de ejecución pueden cambiar su declaración.

Se pueden declarar funciones y procedimientos. Estos procedimientos y funciones pueden ser anidados. Se les puede pasar parámetros por valor y regresar parámetros por referencia. También se puede utilizar el mecanismo de recursión.

e. Modularidad. La modularidad divide la aplicación en pedazos más manejables, módulos bien definidos. PL/SQL provee unidades de programa para poder modular. Entre estas unidades se encuentran los subprogramas y los paquetes. Los subprogramas son los procedimientos y funciones. Los paquetes permiten agrupar lógicamente funciones y procedimientos.

f. Abstracción de datos. La abstracción de datos permite extraer las propiedades esenciales de la información mientras ignora detalles innecesarios. Una vez que se diseña la estructura de datos, se olvidan los detalles y se centra en los algoritmos para la manipulación de la estructura de datos. Por ejemplo colecciones, registros, y tipos de objetos.

g. Ocultar información. Al ocultar la información, se ven sólo los detalles que son relevantes a un nivel dado de diseño de algoritmos y estructura de datos. Por ocultar la información se mantienen decisiones de diseño de alto nivel separadas de detalles de diseño de bajo nivel, lo que facilita los cambios. En el caso de los algoritmos una vez definida la interfase se olvida de los detalles de implementación, y en el caso de las

estructuras de datos utilizan el mecanismo de encapsulación de datos, con el cual se desconoce la representación interna.

h. Manejo de errores. PL/SQL utiliza condiciones de error llamados excepciones para el manejo de los errores. Estas excepciones pueden ser predefinidas o definidas por el usuario. Cuando un error ocurre, una excepción es levantada, se para la ejecución normal y se ejecuta un manejador de la excepción.

2. Arquitectura. En la arquitectura se tiene un motor que ejecuta y compila bloques de PL/SQL. Este motor puede ser instalado en el servidor de Oracle o en una herramienta de desarrollo de aplicaciones como Oracle Forms o Oracle Reports. En cualquiera de estos dos ambientes, el motor acepta como entrada cualquier bloque o subprograma en PL/SQL válido. El motor ejecuta instrucciones procedurales pero envía instrucciones SQL a un ejecutor de instrucciones SQL en el servidor de Oracle.

IV. DESARROLLO DE LA PROPUESTA

Para lograr los objetivos expuestos anteriormente. Se desarrolló una herramienta para hacer consultas sobre los elementos de análisis. Además, los resultados de las consultas tienen diferentes vistas, y la herramienta se utilizará en un caso de estudio para ejemplificar su uso.

La herramienta se construyó con base en una arquitectura. Los componentes que conforman esta arquitectura y sus relaciones son descritos en la primera sección, y serán ampliados en las siguientes secciones.

En la segunda sección se especifican los elementos de análisis, sus propiedades y los tipos de relaciones que serán registrados por la herramienta. En la próxima sección se describe la forma en que se implementó la herramienta para registrar los elementos de análisis y sus relaciones.

Los tipos de consultas que se pueden realizar sobre los elementos de análisis son descritos en la cuarta sección. También, en esta sección se describen los tipos de análisis de dependencias que se pueden realizar con base a los tipos de consultas expuestos en esta sección, y como se implementó el cliente de análisis que utiliza estos tipos de consulta.

La última sección describe un caso de estudio. Este caso de estudio está conformado por una lista de requerimientos y la forma en que se pueden resolver estos requerimientos utilizando la herramienta. Los resultados obtenidos son descritos en la parte de RESULTADOS Y DISCUSIÓN.

A. Arquitectura de la herramienta desarrollada

En esta sección se describen los componentes y las interacciones entre los mismos. Estos componentes forman la arquitectura que sirvió de base para implementar la herramienta. En la Figura 1. Arquitectura de la herramienta, se visualizan los componentes y sus interacciones.

1. Componentes.

a. Repositorio de carga. Este repositorio contiene un conjunto de archivos de texto generados por una aplicación externa. Los archivos anteriores tienen formatos

específicos, por lo cual hay que agruparlos en un directorio. Se puede utilizar también como repositorio de carga el diccionario de datos de la base de datos de la aplicación de estudio.

b. Repositorio de consulta. Este repositorio es en donde se centraliza toda la información de los repositorios de carga. Además, este repositorio debe tener facilidades de consulta y su estructura debe ser flexible para poder almacenar cualquier información del repositorio de carga.

c. Cliente específico de carga. Componente de *software* que se desarrolla específicamente para hacer el análisis léxico y sintáctico de los archivos de texto que se encuentran en el repositorio de carga. Además, es el encargado de cargar la información al repositorio de consulta. Este componente es donde se requiere la mayor inversión de recursos para su desarrollo.

d. Cliente de análisis. Componente de *software* que se desarrolla para poder realizar los diferentes tipos de consulta. La información que utiliza está en el repositorio de consulta. Este componente es genérico en el sentido que es independiente de la carga de información y sólo se desarrolla una vez para posteriores usos.

e. Elementos de análisis y sus relaciones. Los elementos de análisis son entidades de las cuales se desea registrar información para poder realizar posteriormente diferentes tipos de consultas sobre ellos. Cada uno de estos elementos tiene propiedades y relaciones hacia otros elementos. Este componente es la metadata en sí.

Las relaciones son elementos que tiene un nombre y cuya naturaleza se originó por la interacción o dependencia de dos elementos de análisis. Entre dos mismos elementos pueden existir diferentes tipos de relaciones. Además, se identifican dos tipos de elementos, un elemento padre y un elemento hijo para conformar la relación.

2. Interacción entre los componentes. La interacción de los componentes se observa al desarrollar las siguientes etapas:

a. Identificación de la metadata. La identificación de la metadata es importante para poder realizar un buen análisis. En esta etapa se identifican los elementos de análisis,

sus propiedades y sus relaciones. Pueden existir diferentes tipos de relaciones entre los mismos elementos de análisis.

b. Carga de la metadata. Con base a la identificación de la metadata, hay que determinar que herramienta externa puede genera los archivos de texto necesarios. Estos archivos de texto conforman el repositorio de carga y serán agrupados en un directorio. Luego, hay que desarrollar el cliente de carga específicamente para que almacene la metadata, que se encuentra en este repositorio de carga, en el repositorio de consulta. Esta etapa se termina en el momento que ya no haya que cargar más archivos de texto.

c. Análisis o consulta de la metadata. Una vez consolidada toda la información en el repositorio de consulta se utilizará el cliente de análisis para poder realizar los diferentes tipos de consultas.

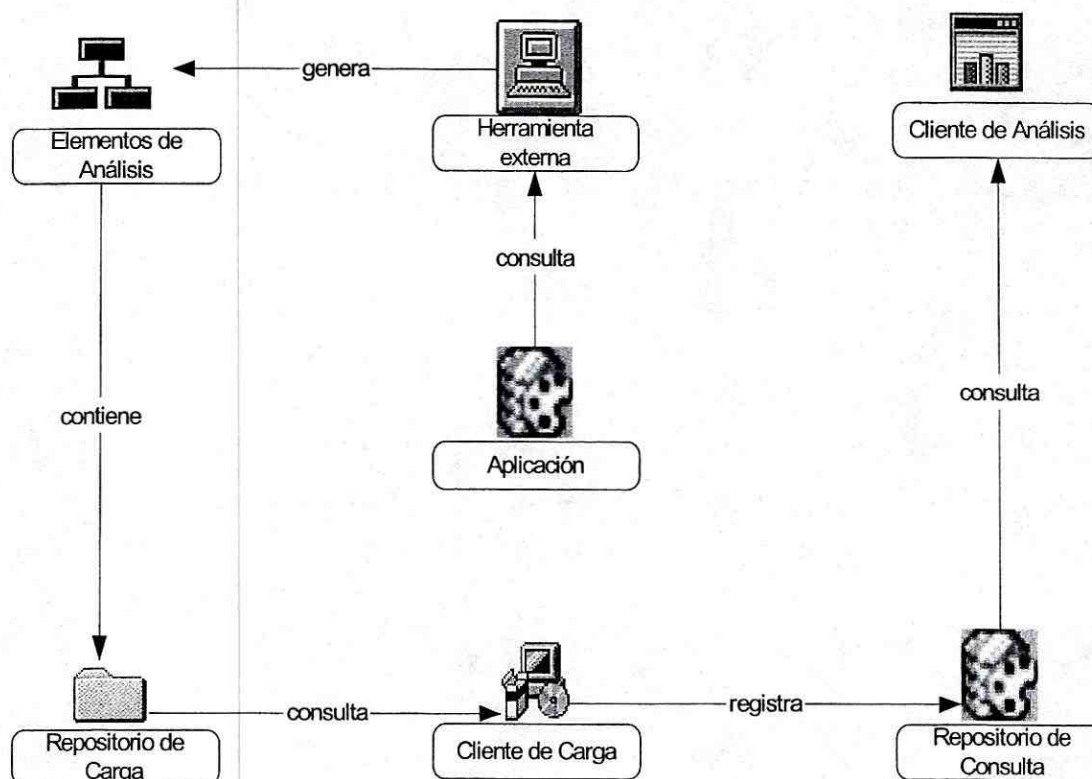


Figura 1. Arquitectura de la herramienta

B. Elementos de análisis seleccionados

El componente de elementos de análisis y sus relaciones se ejemplifica en esta sección. Para la elaboración del cliente de carga se identificaron ciertos elementos del

PL/SQL. Se escogió el lenguaje de programación de PL/SQL porque la mayoría de sistemas orientados a base de datos están desarrollados con herramientas de Oracle en Guatemala. Además, estos elementos se seleccionaron porque son los elementos que mayormente se utilizan en cualquier aplicación de base de datos. Pero, la arquitectura queda abierta para la incorporación de nuevos elementos.

También se definió un objeto padre *DATABASE*. De este objeto descienden todos los elementos de análisis seleccionados que se describirán posteriormente.

1. Tablas.

a. Definición del elemento. Las tablas son las unidades básicas de almacenamiento de información en una base de datos Oracle. La información es mantenida en filas y columnas. Se define la tabla con un nombre y un conjunto de columnas.

Después de creada la tabla, se insertan filas de información utilizando instrucciones de SQL. La información de la tabla puede ser consultada, eliminada, y actualizada utilizando instrucciones de SQL.

Los atributos del elemento tabla son:

- Las columnas.
- Llave primaria.
- Llaves foráneas.

b. Elementos de búsqueda. La definición de las columnas de la tabla se encuentra en la instrucción *CREATE TABLE*. Las restricciones de llave primaria y llave foránea se encuentran en las instrucciones *ALTER TABLE*.

c. Suposiciones y restricciones. Se asume que las instrucciones que vienen en el *script* o conjunto de instrucciones de base de datos son válidas y están bien construidas. Se espera que estas instrucciones que se especificaron en los elementos de búsqueda terminen con un delimitador de punto y coma. En caso de que venga más información de la esperada, esta debe ser ignorada. Si se detecta una mala construcción la instrucción restante será ignorada. Los nombres de los elementos tabla no pueden estar encerrados por comillas dobles ni tener espacios en blanco. También, se espera que cada tabla lleve el nombre del esquema al que pertenece, en caso contrario la instrucción será ignorada.

No se harán búsquedas en instrucciones de SQL dinámico para este elemento, ni cuando la instrucción incorpore características orientadas a objetos.

d. Identificación en la herramienta desarrollada. Este elemento de análisis es identificado con el nombre de *TABLE*.

2. Columnas

a. Definición del elemento. Una colección de columnas de información representa una fila. Esta fila corresponde a un simple registro en la tabla. A cada columna se le asigna un nombre, un tipo de dato y un tamaño. El tamaño puede ser predeterminado por el tipo de dato, por ejemplo *DATE*. Si el tipo de dato es *NUMBER*, se define una precisión y una escala en lugar de tamaño. Se pueden especificar reglas para cada columna en la tabla. Estas reglas son llamadas restricciones de integridad. Los atributos del elemento columna son:

- Tipo de dato.
- Anulable.
- Parámetro por omisión.

b. Elementos de búsqueda. Este elemento se encuentra en la instrucción *CREATE TABLE*.

c. Suposiciones y restricciones. Se espera que los nombre de las columnas no vengan entre comillas dobles. En caso de que la instrucción este mal construida se ignorará por completo.

d. Identificación en la herramienta desarrollada. Este elemento de análisis es identificado con el nombre de *COLUMN*.

3. Vistas.

a. Definición del elemento. Una vista es una presentación de la información hecha a la medida de datos contenidos en una o más tabla u otras vistas. Este elemento vista toma la salida de la consulta y es tratada como una tabla. Por lo tanto, se puede pensar a

la vista como una consulta almacena o una tabla virtual. Se puede utilizar las vistas en muchos lugares donde se utilicen las tablas.

El atributo del elemento vista es:

- La consulta.

Las ventajas del uso de las vistas son:

- Esconde la complejidad de la información.
- Provee nivel de seguridad en la tabla por restringir el acceso a columnas y filas.
- Presenta la información en diferentes perspectivas de la tabla base.
- Aísla a la aplicación de cambios en la definición de la tabla base.

b. Elementos de búsqueda. Este elemento se encuentra en la instrucción *CREATE VIEW*.

c. Suposiciones y restricciones. Se asume que la instrucción especificada en el elemento de búsqueda está bien construida. El nombre del elemento vista no puede ir entre comillas dobles y la vista debe ser un objeto válido. Además, se espera que la instrucción termine con un delimitador de punto y coma. En el caso de que la instrucción esté incorrecta se ignorará la parte restante. También, se espera que cada vista lleve el nombre del esquema al que pertenece, en caso contrario la instrucción será ignorada. No se harán búsquedas en instrucciones de SQL dinámico para este elemento, ni cuando la instrucción sea de una vista materializada o incorpore características orientadas a objetos.

d. Identificación en la herramienta desarrollada. Este elemento de análisis es identificado con el nombre de *VIEW*.

4. Procedimientos.

a. Definición del elemento. Un procedimiento es un bloque nombrado de PL/SQL que desarrolla una acción específica. El procedimiento puede tomar parámetros y ser invocado. La única manera que se tiene para que devuelva valores es por el paso por referencia de los parámetros.

Los atributos del elemento procedimiento son:

- Los argumentos del procedimiento .

Las ventajas del uso de los procedimientos son:

- Los procedimientos pueden forzar la seguridad de la información.
- Los procedimientos almacenados mejoran la integridad y la consistencia de la información.
- Los procedimientos almacenados incrementan la productividad de los desarrolladores por la reutilización de código y modularidad.
- Sólo una simple copia del procedimiento necesita ser cargada para que pueda ser utilizada por varios usuarios.

b. Elementos de búsqueda. Este elemento se encuentra en la instrucción *CREATE OR REPLACE PROCEDURE*. También se puede encontrar dentro de un bloque anónimo, otro subprograma y paquete con la instrucción *PROCEDURE*.

c. Suposiciones y restricciones. Se espera que las instrucciones especificadas en los elementos de búsqueda estén bien construidas. El nombre del elemento procedimiento no puede estar encerrado entre comillas dobles, y este elemento debe ser un objeto válido. También, se espera que cada procedimiento lleve el nombre del esquema al que pertenece, en caso contrario la instrucción será ignorada. No se harán búsquedas en instrucciones de SQL dinámico para este elemento.

d. Identificación en la herramienta desarrollada. Este elemento de análisis se identifica con el nombre de *PROCEDURE*.

5. Funciones.

a. Definición del elemento. Una función es un bloque nombrado de PL/SQL que calcula un valor. La función puede tomar parámetros y ser invocado para devolver un valor. Se puede devolver otros valores por el paso por referencia de los parámetros.

Los atributos del elemento función son:

- Los argumentos de la función.
- Valor que devuelve la función.

Las ventajas de las funciones son similares a las ventajas de los procedimientos.

b. Elementos de búsqueda. Este elemento se encuentra en la instrucción *CREATE OR REPLACE FUNCTION*. También se puede encontrar dentro de un bloque anónimo, otro subprograma y paquete con la instrucción *FUNCTION*.

c. Suposiciones y restricciones. Se espera que las instrucciones especificadas en los elementos de búsqueda estén bien construidas. El nombre del elemento función no debe estar encerrado entre comillas dobles, y este elemento debe ser un objeto válido. También, se espera que cada función lleve el nombre del esquema al que pertenece, en caso contrario la instrucción será ignorada. No se harán búsquedas en instrucciones de SQL dinámico para este elemento.

d. Identificación en la herramienta desarrollada. Este elemento de análisis se identifica con el nombre de *FUNCTION*.

6. Triggers.

a. Definición del elemento. El *trigger* es un procedimiento que se ejecuta implícitamente en el momento en el que se emite una instrucción de *UPDATE*, *INSERT* o *DELETE* en contra de una tabla asociada o una vista o cuando ocurre una acción el sistema de la base de datos. Estos procedimientos pueden ser escritos en PL/SQL, Java o C.

Los atributos del elemento *trigger* son:

- Objeto al que aplica el *trigger*.
- Cuando aplica el evento.
- Eventos.
- Referencias.
- Aplica a la fila.

Uso de los *triggers*:

- Automáticamente genera valores derivados en las columnas.
- Previene de transacciones inválidas.
- Fuerza autorizaciones de seguridad complejas.
- Fuerza integridad referencial a través de nodos en base de datos distribuidos.

- Fuerza complejas reglas del negocio.
- Provee transparencia en eventos de bitácoras.
- Provee sofisticada auditoría.
- Mantiene sincronizadas las tablas en la replicación.
- Obtiene estadísticas sobre el acceso a las tablas.
- Modifica información de la tabla cuando instrucciones de actualización son emitidas contra vistas.
- Publica información sobre eventos de base de datos, eventos de usuario, e instrucciones de SQL a aplicaciones de suscripción.

b. Elementos de búsqueda. Este elemento se encuentra en la instrucción *CREATE OR REPLACE TRIGGER*.

c. Suposiciones y restricciones. Se espera que las instrucciones especificadas en los elementos de búsqueda estén bien construidas. El nombre del elemento *trigger* no debe estar encerrado entre comillas dobles y este elemento no debe estar un objeto inválido. Se espera que cada *trigger* lleve el nombre del esquema al que pertenece, en caso contrario la instrucción debe ser ignorada. No se debe buscar en instrucciones de SQL dinámico para este elemento, únicamente se hará la búsqueda para los *triggers* creados para las tablas.

d. Identificación en la herramienta desarrollada. Este elemento de análisis es identificado con el nombre de *TRIGGER*.

7. Paquetes.

a. Definición del elemento. Los paquetes encapsulan procedimientos y funciones relacionadas, y asocia cursores y variables juntos como una unidad en la base de datos. Los paquetes se declaran en dos partes: Especificación y cuerpo. La especificación declara todas las construcciones públicas del paquete, y el cuerpo define todas las construcciones privadas o públicas del paquete. Los atributos del elemento paquete son:

- Las especificaciones y declaraciones de cuerpo de los procedimientos o funciones.

Ventajas de los paquetes:

- Con las especificaciones se declaran las interfases sin necesidad crear la implementación.
- Se puede alterar la implementación de los procedimientos del cuerpo separadamente de la especificación sino se cambia la firma.
- Encapsulación de variables y procedimientos relacionados.
- Mejoramiento de rendimiento, ya que una vez accesado un procedimiento se mantiene en memoria todo el paquete para poder ser accesado por varios usuarios.

b. Elementos de búsqueda. Este elemento se encuentra en las instrucciones *CREATE OR REPLACE PACKAGE* y *CREATE OR REPLACE PACKAGE BODY*. En la primero instrucción se define la especificación y en la segunda se define el cuerpo.

c. Suposiciones y restricciones. Se espera que las instrucciones especificadas en los elementos de búsqueda estén bien construidas. El nombre del elemento paquete no puede estar encerrado entre doble comilla y este elemento debe ser un objeto válido. También, se espera que cada paquete lleve el nombre del esquema al que pertenece, en caso contrario la instrucción debe ser ignorada. No se deben buscar en instrucciones de SQL dinámico para este elemento.

d. Identificación en la herramienta desarrollada. Este elemento de análisis se identifica con el nombre de *PACKAGE*.

8. Tipos de relaciones entre los elementos de análisis. Los tipos de relaciones se obtendrán según los elementos de análisis que se seleccionen y con base en que información se desee obtener.

a. Relaciones entre los elementos de análisis. En el siguiente cuadro se enumeran las relaciones entre los elementos de análisis, donde se observa que las relaciones se obtienen de las instrucciones de actualización de SQL (DML), de las llamadas de procedimientos y de las funciones, y de la forma en que están compuestos los elementos de análisis seleccionados. En la Figura 2 se puede tener una mejor visualización de los elementos de análisis seleccionados con sus relaciones.

Relaciones	Tabla	Columna	Vista	Procedimiento	Función	Trigger	Paquete
Tabla		Contiene	Seleccionada	Seleccionada Actualizada Eliminada Insertada	Seleccionada Actualizada Eliminada Insertada	Contiene Seleccionada Actualizada Eliminada Insertada	
Columna	Pertenece						
Vista	Selecciona			Seleccionada Actualizada Eliminada Insertada	Seleccionada Actualizada Eliminada Insertada	Seleccionada Actualizada Eliminada Insertada	
Procedimiento	Seleccionó Actualizó Eliminó Insertó		Seleccionó Actualizó Eliminó Insertó	Llama	Llama	Llamado	Contenido
Función	Seleccionó Actualizó Eliminó Insertó		Seleccionó Actualizó Eliminó Insertó	Llama	Llama	Llamado	Contenido
Trigger	Contenido Seleccionó Actualizó Eliminó Insertó		Seleccionó Actualizó Eliminó Insertó	Llama	Llama		
Paquete				Contiene	Contiene		

Cuadro 1. Relaciones entre los elementos de análisis seleccionados

b. Tipos de relaciones. Los tipos de relaciones explicados a continuación representan las relaciones que serán registradas en la herramienta desarrollada. Cuando se haga referencia a una relación se utilizarán los mismos nombres de estos tipos de relaciones. Se describen los diferentes tipos de relaciones:

1) **Contiene.** Esta relación se origina del concepto de elementos de estructuras menores que forman parte o pertenecen a elementos de una estructura mayor. Se visualiza una estructura jerárquica del tipo de relación. Por lo general, la instrucción *CREATE* origina este tipo de relación.

2) **Selección.** Esta relación se origina de la instrucción *SELECT* de SQL. Pretende indicar que el elemento de análisis padre obtuvo información del elemento de análisis hijo.

3) **Actualización.** Esta relación se origina de la instrucción *UPDATE* de SQL. Pretende indicar que el elemento de análisis padre realizó modificaciones a la información del elemento de análisis hijo.

4) **Inserción.** Esta relación se origina de la instrucción *INSERT* de SQL. Pretende indicar que el elemento de análisis padre adicionó información al elemento de análisis hijo.

5) **Eliminación.** Esta relación se origina de la instrucción *DELETE* de SQL. Pretende indicar que el elemento de análisis padre eliminó información del elemento de análisis hijo.

6) **Llama.** Esta relación se origina al ejecutar una función o procedimiento. La ejecución de las funciones se puede realizar dentro de las instrucciones de SQL como en el momento de realizar una asignación o formar parte de una expresión.

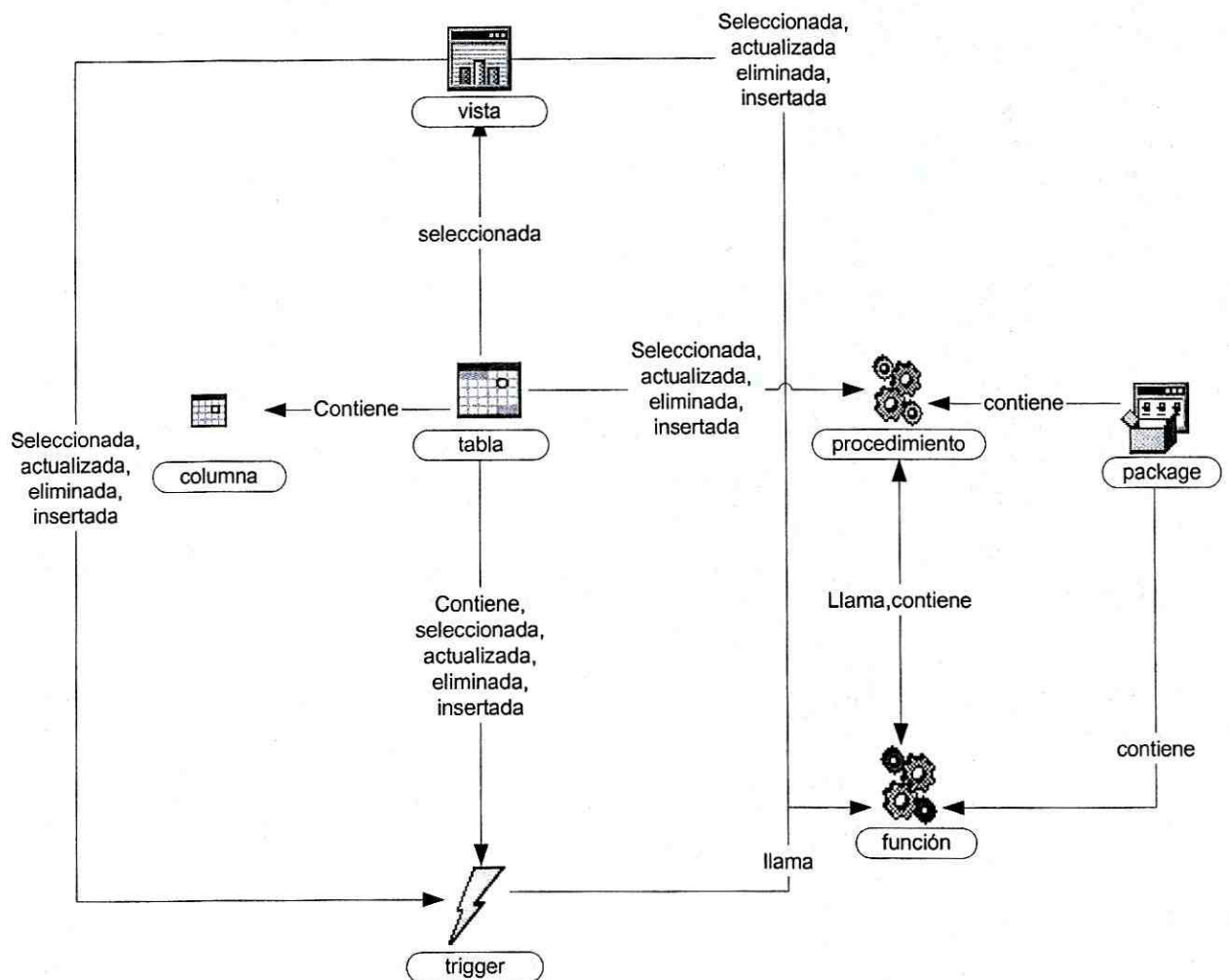


Figura 2. Elementos de análisis seleccionados con sus relaciones

C. Implementación del cliente de carga

En esta sección se describe a detalle el componente del cliente de carga, y los repositorios que interactúan con él mismo. Se especifica cómo construir los repositorios y se describen los objetos del repositorio de consulta para facilitar su uso para posteriores implementaciones de clientes de carga.

1. Repositorio de carga. Los archivos que conforman este repositorio pueden ser generados por una herramienta externa de *scripts* de base de datos. Estos archivos deben cumplir con las restricciones y suposiciones indicadas en los elementos de análisis. En caso que no cumplan alguna de las condiciones deben ser transformados.

Los pasos para la creación del repositorio de carga son:

- Definir el nombre del directorio de la aplicación. El nombre del directorio no debe contener espacios en blanco, ni caracteres especiales, ni ser muy extenso.
- Crear tres directorios debajo del directorio de aplicación con los siguientes nombres: “repositorio”, “template” y “temp”. En el directorio “temp” se almacenan los archivos que la herramienta de modelación Visio interpretará y dibujará (“consulta.txt”), y los archivos que se generan cuando la consulta contiene mucha información (“respuesta.txt”), y el directorio “template” contiene un archivo de los íconos de los elementos de análisis para que la herramienta de modelación Visio lo utilice (“elemento.vss”).
- Crear el directorio “db” dentro del directorio repositorio. En este directorio se almacenan los archivos generados por las herramientas externas. El cliente de carga lee de este directorio los archivos de texto, para registrar los elementos de análisis seleccionados con sus relaciones. No debe almacenar subdirectorios en el directorio “db” porque el cliente de carga no leerá los archivos dentro de estos subdirectorios.
- El directorio de la aplicación debe quedar compartido para que lo puedan acceder las otras máquinas y el directorio “temp” debe tener permisos de escritura. Los otros directorios de la aplicación deben quedar con permiso de lectura, exceptuando el directorio “db” que debe tener permisos de escritura para los usuarios que realizan la carga, ya que ellos tienen que colocar los archivos de texto en este directorio.

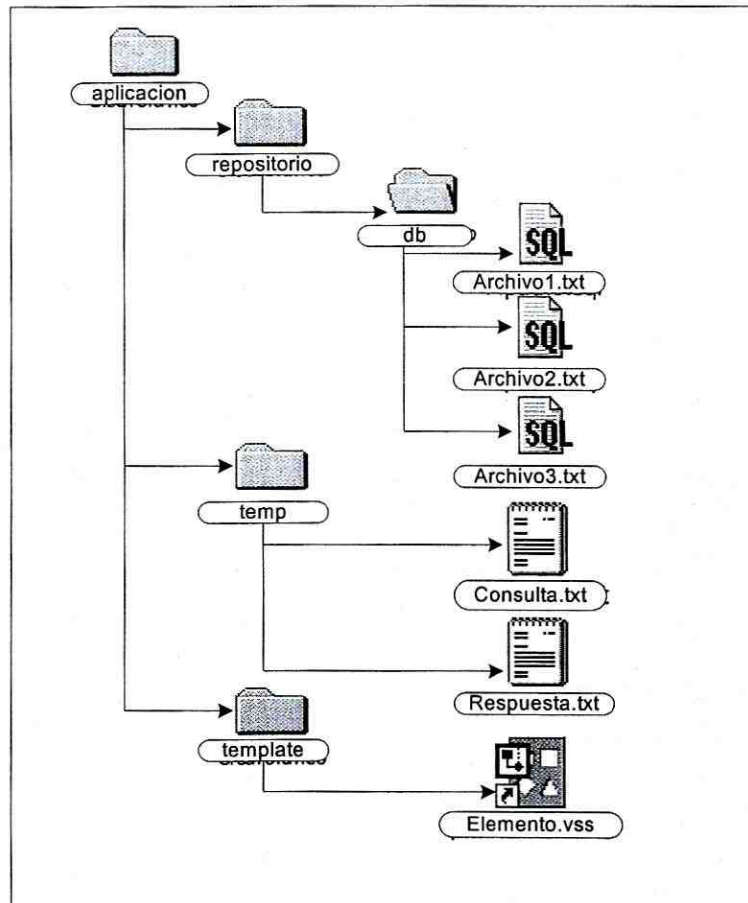


Figura 3. Repositorio de carga

2. Repositorio de consulta. El repositorio de consulta consiste en una base de datos relacional Oracle 8i. Pasos para creación del repositorio:

- Utilizar los discos compactos de instalación de una base de datos relacional 8i para configurar la instancia que utilizará la herramienta.
- Conectarse con el usuario SYS en la instancia configurada en la opción anterior, utilizando la herramienta PLUS 8. Correr el *script* "usuario_aplicación_analisis.sql". Este archivo crea el usuario APLICACION_ANALISIS, este usuario es el dueño del repositorio de consulta. Además, se crea un *tablespace* APLICACION_ANALISIS para la instalación del repositorio de consulta y debe ser exclusivo de este repositorio de consulta. Con el usuario APLICACION_ANALISIS deben realizarse las cargas de información. Se recomienda modificar el camino de la ubicación en el disco de almacenamiento donde residirá el *tablespace*.
- Ingresar con el usuario APLICACION_ANALISIS cuya palabra clave es la

misma, usando la herramienta PLUS 8. Ejecutar el *script* “objetos_aplicación_analisis.sql”. Este archivo crea todos los objetos de base de datos que utiliza la herramienta desarrollada.

- Conectarse con el usuario SYS utilizando la herramienta PLUS 8, y correr el *script* “usuario_consulta_analisis.sql”. Este archivo crea un usuario CONSULTA_ANALISIS, este usuario es el encargado de realizar las consultas a este repositorio.
- Ingresar con el usuario CONSULTA_ANALISIS cuya palabra clave es la misma, utilizando la herramienta PLUS 8. Correr el *script* “sinonimos_usuario_consulta.sql”. Este archivo crea todos los sinónimos privados del usuario de CONSULTA_ANALISIS.

El repositorio de consulta se visualiza como un conjunto de aplicaciones. Cada una de estas aplicaciones está conformada por los objetos y relaciones válidos de los elementos de análisis que se encontraron en los respectivos repositorios de carga.

La lista de tablas que conforman el repositorio de consulta se describen a continuación, para que facilite la comprensión del tipo de información que se puede obtener de este cliente de carga. Además, puede ayudar a nuevas implementaciones de clientes de carga.

En la tabla APLICACION se registra la información de control de la aplicación. La lista de campos se presenta en el siguiente cuadro:

Nombre del campo	Tipo	Información
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
FECHA	NOT NULL DATE	Fecha de carga
NOMBRE	NOT NULL VARCHAR2(30)	Nombre de la aplicación
DESCRIPCION	VARCHAR2(100)	Breve descripción de la aplicación
CORREL_OBJETO	NOT NULL NUMBER(9)	Correlativo de objetos
CORREL_RELACION	NOT NULL NUMBER(9)	Correlativo de relaciones
PATH_REPOSITORIO	NOT NULL VARCHAR2(60)	Camino del directorio del repositorio
CORREL_REPOSITORIO	NOT NULL NUMBER(9)	Correlativo del repositorio
ESQUEMA	NOT NULL VARCHAR2(30)	Dueño del esquema
INSTANCIA	NOT NULL VARCHAR2(30)	Nombre de la instancia

Cuadro 2. Lista de campos de la tabla APLICACION

En la tabla CORREL_APLICACION se lleva el control del número del correlativo de la última aplicación cargada. Si una aplicación nueva se carga se le asignará este correlativo más uno y se actualizará el número del correlativo de la última aplicación a este nuevo correlativo. La lista de campos se describe en el siguiente cuadro:

Nombre del campo	Tipo	Información
CORREL	NOT NULL NUMBER(9)	Correlativo de la última aplicación cargada

Cuadro 3. Lista de campos en la tabla CORREL_APLICACION

En la tabla REPOSITORIO_APLICACION se lleva el control de los archivos que conforman el repositorio de carga. La lista de campos se detalla en el próximo cuadro:

Nombre del campo	Tipo	Información
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
CODARCHIVO	NOT NULL NUMBER(9)	Código del archivo
PATHARCHIVO	NOT NULL VARCHAR2(100)	Camino del archivo con el nombre del archivo
NOMBRE	NOT NULL VARCHAR2(30)	Nombre del archivo

Cuadro 4. Lista de campos en la tabla REPOSITORIO_APLICACION

En la tabla OBJETO_APLICACION se registra la información de los objetos que conforman los elementos de análisis. La lista de campos se describe en el siguiente cuadro:

Nombre del campo	Tipo	Información
CODOBJETO	NOT NULL NUMBER(9)	Código del objeto
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
LOCALIZACION	NOT NULL VARCHAR2(30)	En donde reside el objeto
ESQUEMA	NOT NULL VARCHAR2(30)	Dueño del objeto
TIPO	NOT NULL VARCHAR2(30)	Tipo del objeto
NOMBRE	NOT NULL VARCHAR2(35)	Nombre del objeto
CODOBJETOPADRE	NUMBER(9)	Código del objeto padre
PADRE_TIPO	VARCHAR2(30)	Tipo del objeto padre
PADRE_NOMBRE	VARCHAR2(30)	Nombre del objeto padre
NOMBRE_REFERENCIAL	NOT NULL VARCHAR2(200)	Nombre completo donde reside el objeto

Cuadro 5. Lista de campos de la tabla OBJETO_APLICACION

Nota: La localización es el lugar donde reside el objeto. Por ejemplo, el objeto puede residir dentro de una base de datos o puede residir dentro de un repositorio de formas, reportes o archivos de alguna aplicación.

En la tabla PROPIEDAD_OBJETO se registran las propiedades de los objetos. La lista de campos se detalla en el próximo cuadro:

Nombre del campo	Tipo	Información
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
CODOBJETO	NOT NULL NUMBER(9)	Código del objeto
NOMBRE	NOT NULL VARCHAR2(50)	Nombre de la propiedad
VALOR	NOT NULL VARCHAR2(250)	Valor de la propiedad

Cuadro 6. Lista de campos de la tabla PROPIEDAD_OBJETO

En la tabla RELACION_OBJETO se registran las relaciones entre los objetos. La lista de campos se describe en el siguiente cuadro:

Nombre del campo	Tipo	Información
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
CODRELACION	NOT NULL NUMBER(9)	Código de la relación
CODOBJETOPADRE	NOT NULL NUMBER(9)	Código del objeto padre
CODOBJETOHIJO	NOT NULL NUMBER(9)	Código del objeto hijo
TIPO_RELACION	NOT NULL VARCHAR2(30)	Tipo de relación

Cuadro 7. Lista de campos de la tabla RELACION_OBJETO

En la tabla TEXTUAL_OBJETO se registra la información para las vistas textuales de los objetos. La lista de campos se detalla en el próximo cuadro:

Nombre del campo	Tipo	Información
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
CODOBJETO	NOT NULL NUMBER(9)	Código del objeto
CODARCHIVO	NOT NULL NUMBER(9)	Código del archivo
POSLINEAINICIAL	NOT NULL NUMBER(13)	Posición inicial en el archivo
POSLINEAFINAL	NOT NULL NUMBER(13)	Posición final en el archivo
NUMLINEA	NOT NULL NUMBER(13)	Número de línea en el archivo

Cuadro 8. Lista de campos de la tabla TEXTUAL_OBJETO

En la tabla TEXTUAL_RELACION_OBJETO se registra la información para las vistas textuales de las relaciones. Lista de campos se presenta en el siguiente cuadro:

Nombre del campo	Tipo	Información
CODAPLICACION	NOT NULL NUMBER(9)	Código de la aplicación
CODRELACION	NOT NULL NUMBER(9)	Código de la relación
CODARCHIVO	NOT NULL NUMBER(9)	Código del archivo
POSLINEAINICIAL	NOT NULL NUMBER(13)	Posición inicial en el archivo
POSLINEAFINAL	NOT NULL NUMBER(13)	Posición final en el archivo
NUMLINEA	NOT NULL NUMBER(13)	Número de línea en el archivo

Cuadro 9. Lista de campos de la tabla TEXTUAL_RELACION_OBJETO

Se construyeron tres procedimientos con los cuales se resuelven las ambigüedades de los objetos y sus relaciones. Estos procedimientos son: res_amb_obj, res_amb_rel, res_amb_rel_obj. Todos estos objetos de base de datos que conforman el repositorio de consulta son visualizados en la Figura 4.

El diagrama entidad relación se muestra en la Figura 5, en donde la llave primaria es conformada por la combinación de campos que tienen como marca el símbolo de asterisco y las llaves foráneas están marcadas por el símbolo de número para cada entidad. Las relaciones son visualizadas por flechas, la etiqueta que tiene la fecha está conformada por dos partes separadas por coma, cada parte indica la cardinalidad y opcionalidad de la relación. La primera parte de esta etiqueta se refiere a la relación entre dos entidades en la dirección a favor de la fecha y la segunda parte se refiere a la relación entre dos entidades en dirección inversa a la fecha.

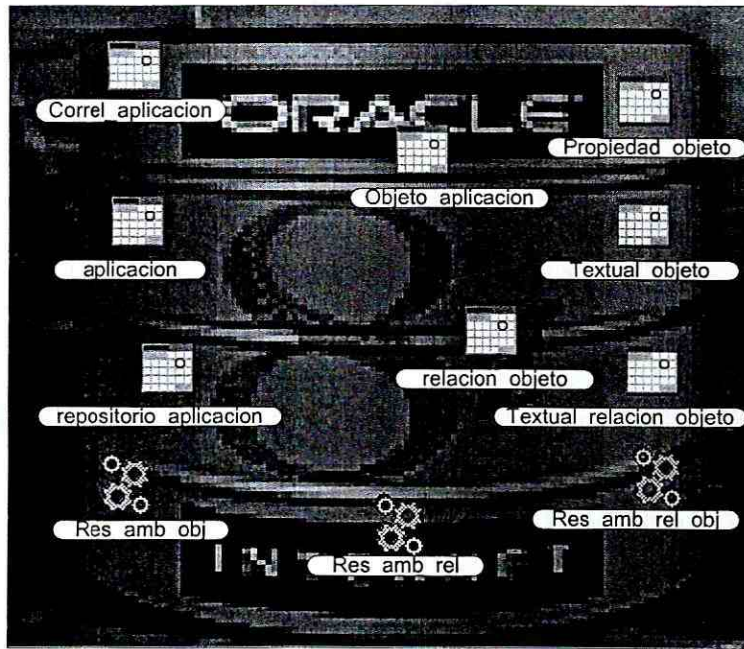


Figura 4. Repositorio de consulta

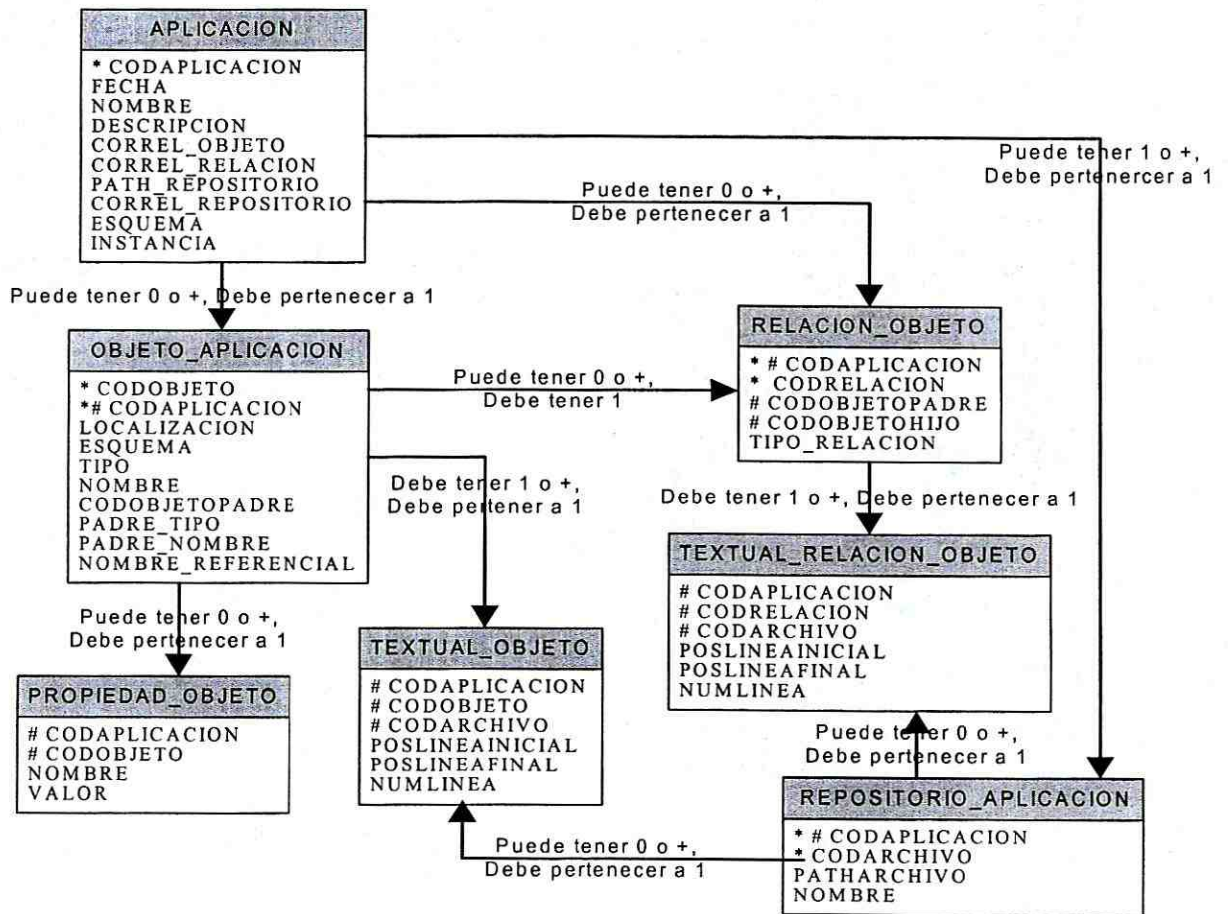


Figura 5. Diagrama objeto relacional del repositorio de consulta

3. Implementación del cliente de carga. El componente de cliente de carga fue desarrollado en la herramienta de Microsoft Visual Basic .Net. El cliente de carga tiene un analizador léxico y sintáctico, de los elementos de análisis seleccionados del lenguaje de programación PL/SQL, de los archivos que conforman el repositorio de carga. El analizador léxico es el encargado de obtener los elementos individuales, pertenecientes al lenguaje de programación PL/SQL, al agrupar secuencias de caracteres de los archivos que conforman el repositorio de carga. El analizador sintáctico es el encargado de verificar las reglas gramaticales del lenguaje de programación PL/SQL, obtener la información de los elementos de análisis seleccionados y sus relaciones, y registrar dicha información en el repositorio de consulta.

Además, el analizador sintáctico en el momento de encontrar algún error utiliza un conjunto de puntos de sincronización para ignorar las instrucciones mal construidas. El conjunto de puntos de sincronización está formado por las palabras reservadas *CREATE* y *ALTER*, y el símbolo de punto y coma.

En el repositorio de consulta se define un número de aplicación. Este número de aplicación identifica de manera única la carga de la información del repositorio de carga. La carga de la información se realiza con el usuario *APLICACION_ANALISIS*. Los primeros elementos de análisis y relaciones que se agregan son objetos pertenecientes al esquema *SYS*. Si se desean agregar más objetos de este esquema, hay que revisar y modificar el código que crea y concede el número de la aplicación que corresponde en el momento de carga en este componente de la herramienta desarrollada.

Se realizan dos revisiones por cada archivo de texto perteneciente al repositorio de carga. La primera revisión sirve para identificar los elementos de análisis seleccionados, y se registra el elemento mismo con sus propiedades. La segunda revisión sirve para registrar la información de las relaciones de los elementos de análisis identificados en la primera pasada. El único elemento de análisis seleccionado que no se registra en la primera pasada son los *trigger*, éstos se registran en la segunda pasada porque se necesita conocer el elemento de análisis de tipo tabla, vista o base de datos al cual pertenecen.

Los nombre de los elementos de análisis son únicos salvo esquema. Excepto cuando son declarados en los paquetes ya que existe sobrecarga, y cuando las funciones o

procedimientos son declarados en bloques anónimos o de manera anidada. Para resolver su referencia o llamada los elementos de análisis deben tener el esquema al que pertenecen. En el momento de tener alguna ambigüedad de resolución de nombres se deja al usuario para que lo resuelva. De igual manera, si existieran ambigüedades de resolución en las relaciones se deja al usuario para que lo resuelva. Se le indicará al usuario cuáles son elementos con problema y él escogerá entre las alternativas dadas cual es la más apropiada.

Cada elemento y relación tiene un identificador único, que es un entero positivo, perteneciente al número de la aplicación que se está cargando en el repositorio de consulta. Para identificar cuáles son los elementos con problema en el momento de encontrarlos se registran con un entero negativo.

Al ejecutar la aplicación desarrollada se requiere una autenticación del usuario APLICACION_ANALISIS, para ello se necesita un archivo de texto donde se encuentre la cadena de conexión. Se recomienda configurar un archivo con extensión udl para probar la cadena de conexión, editar este archivo con Notepad para copiar la cadena y crear un archivo de texto con esta cadena. En la Figura 6 se observa la cadena de conexión que indica la utilización del *provider* de Oracle, el usuario APLICACION_ANALISIS con su palabra clave y la cadena de conexión de Oracle, ORACLEDB.

```
[oledb]
; Everything after this line is an OLE DB initstring
Provider=oraOLEDB.oracle.1;Password=aplicacion_analisis;Persist security Info=True;
User ID=aplicacion_analisis;data source=oracledb
```

Figura 6. Cadena de conexión

En la Figura 7 se observa la pantalla para realizar la autenticación con base al archivo que contiene la cadena de conexión. En esta pantalla hay tres botones, un botón para buscar el archivo, otro botón para probar la conexión y el último botón que llama a otra pantalla para realizar la carga de la información en el repositorio de consulta. La pantalla para realizar la carga se muestra en la Figura 8 donde se llena la información del directorio del repositorio de carga, el nombre del esquema, el nombre de la instancia, el nombre la aplicación y descripción de la aplicación. Se presiona el botón para cargar los

datos de la aplicación en la herramienta desarrollada. La información de la ubicación del directorio del repositorio de carga es importante para el uso de la herramienta desarrollada. En el caso de que este repositorio de carga sea compartido por varios clientes de análisis, es necesario escoger un directorio virtual. Este directorio virtual tiene que formar parte en el nombre del directorio del repositorio de carga.

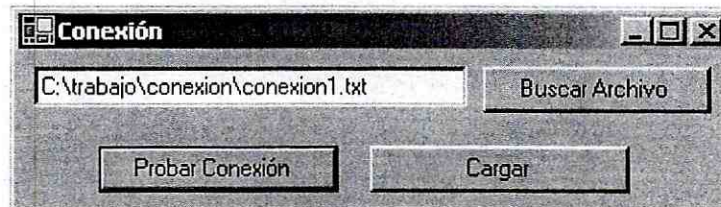


Figura 7. Pantalla de conexión cliente de carga

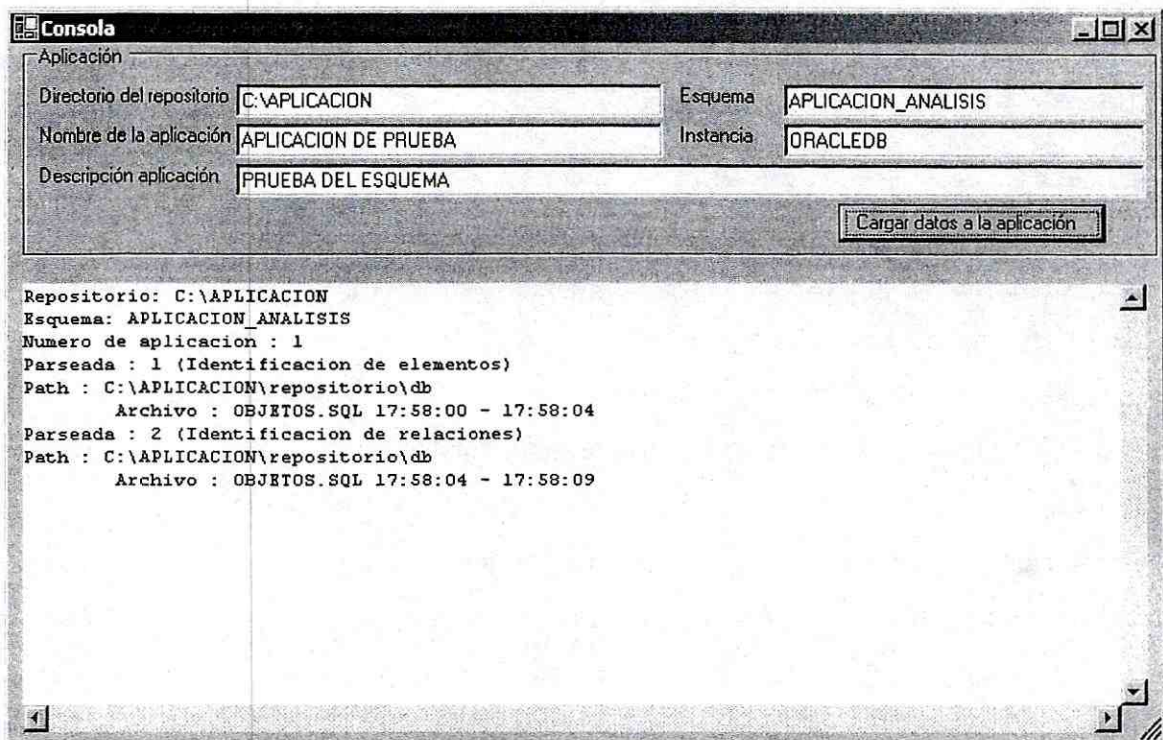


Figura 8. Pantalla de carga

Después de realizada la etapa de identificación de elementos, en el caso de que existan elementos con problema, se levanta una pantalla de resolución de ambigüedades de elementos, ver Figura 9. En esta pantalla se escoge un elemento incorrecto y se dan tres acciones a tomar: el elemento es nuevo, actualizar este elemento por otro elemento seleccionado, o eliminar el elemento. Cada vez que se escoge un elemento incorrecto de la lista se llena el recuadro que se encuentra debajo de él con su definición, propiedades y

relaciones encontradas. De igual manera sucede con el elemento seleccionado. La información que se detalla en los recuadros es de ayuda.

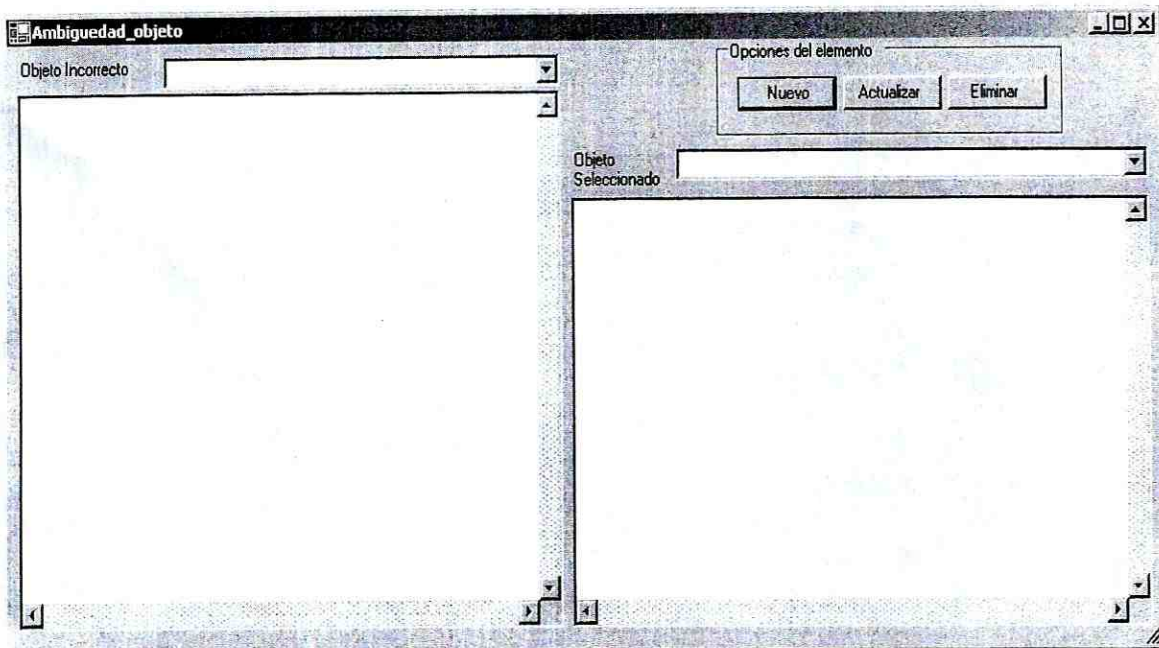


Figura 9. Pantalla de resolución de ambigüedades de elementos

Una vez terminadas todas las ambigüedades de elementos, se ejecuta la etapa de identificación de relaciones. Al terminar la etapa anterior, se procede a resolver las ambigüedades de elementos que se encontraron en las relaciones, en el caso de que existan relaciones con problema se levanta una pantalla de resolución de ambigüedades de elementos pertenecientes a una relación (ver Figura 10). De igual manera que la pantalla anterior, se escoge una relación incorrecta y se tienen tres acciones a tomar: la relación es nueva, actualizar la relación con la escogencia de elemento padre e hijo, o eliminar la relación. Cada vez que se escoge una relación incorrecta de la lista se llena el recuadro que se encuentra debajo de él con el texto original donde se encontró la relación. De igual manera sucede con los elemento seleccionados padre e hijo, en los recuadros de abajo se llena con la información de su definición, propiedades y relaciones. La información que se detalla en los recuadros sirve para ayudar a esclarecer la ambigüedad.

Después de finalizar todas las ambigüedades de los elementos encontrados en las relaciones, se procede a resolver las ambigüedades de las relaciones, en el caso de que existan relaciones con problema, se levanta una pantalla de resolución de ambigüedades de relaciones (ver Figura 11). En esta pantalla se escoge una relación incorrecta y se dan

tres acciones a tomar: la relación es nueva, actualizar esta relación por otra relación seleccionada, o eliminar la relación. Cada vez que se escoge una relación incorrecta de la lista se llena el recuadro que se encuentra debajo de él con el texto original de donde se encontró la relación. De igual manera sucede con la relación seleccionada. La información que se detalla en los recuadros es de ayuda para esclarecer la ambigüedad.

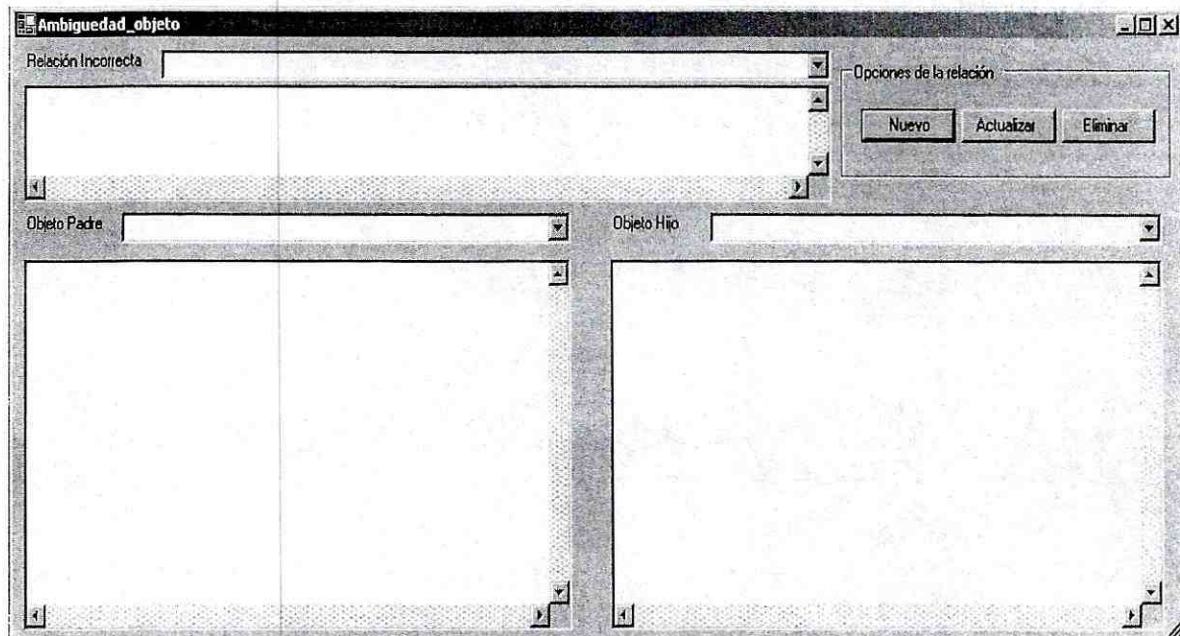


Figura 10. Pantalla de resolución de ambigüedades de elementos en una relación

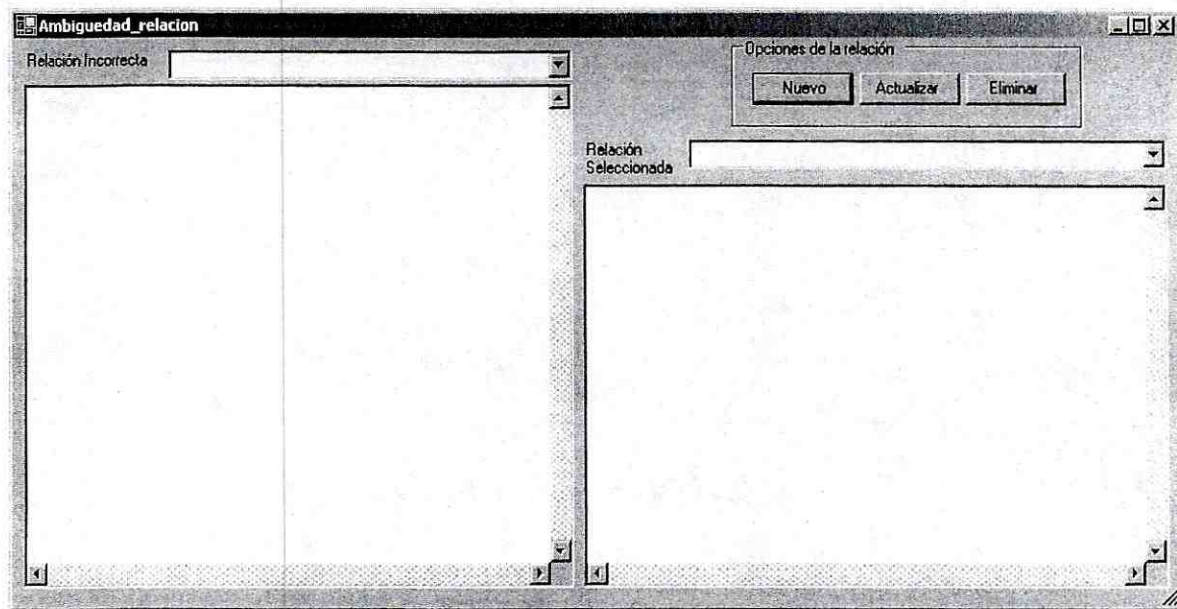


Figura 11. Pantalla de resolución de ambigüedades de una relación

D. Implementación del cliente de análisis

Esta sección describe los diferentes tipos de consultas que puede realizar el componente de cliente de análisis, y de la forma en que se implementó en la herramienta desarrollada. También se dan ejemplos del cliente de análisis de como implementar tipos de análisis de dependencias con base en las consultas.

1. Tipos de consultas de los elementos de análisis seleccionados. Se describen los diferentes tipos de consultas que se pueden realizar:

a. Textual. Este tipo de consulta se puede realizar sobre los elementos o relaciones. Esta consulta da el texto original de donde se registró la información del elemento o relación. Puede servir de validación del cliente de carga. En la Figura 12, se ejemplifica este tipo de consulta.

```

RAMA
-----
LOCALIZACION_PADRE      ESQUEMA_PADRE      TIPO_PADRE      NOMBRE
DB      APLICACION_ANALISIS  PROCEDURE      POSINI      POSFIN      RES_AMB_OBJ
ARCHIVO
-----
C:\APLICACION\repositorio\db\objetos.sql      12296      12665
1
2 insert into objeto_aplicacion (CODOBJETO,CODAPLICACION,LOCALIZACION,ESQUEMA,TIPO,NOMBRE,CODOBJETOPAI
3 NOMBRE_REFERENCIAL)
4 select -1*CODOBJETO,CODAPLICACION,LOCALIZACION,ESQUEMA,TIPO,NOMBRE,CODOBJETOPADRE,PADRE_TIPO,PADRE_I
5 NOMBRE_REFERENCIAL
6 from objeto_aplicacion
7 where codaplicacion = pcodaplicacion and codobjeto = pcodobjetoinc;
RAMA
-----
LOCALIZACION_PADRE      ESQUEMA_PADRE      TIPO_PADRE      NOMBRE
DB      APLICACION_ANALISIS  PROCEDURE      POSINI      POSFIN      RES_AMB_OBJ
ARCHIVO
-----
C:\APLICACION\repositorio\db\objetos.sql      12296      12603
1
2 insert into objeto_aplicacion (CODOBJETO,CODAPLICACION,LOCALIZACION,ESQUEMA,TIPO,NOMBRE,CODOBJETOPAI
3 NOMBRE_REFERENCIAL)
4 select -1*CODOBJETO,CODAPLICACION,LOCALIZACION,ESQUEMA,TIPO,NOMBRE,CODOBJETOPADRE,PADRE_TIPO,PADRE_I
5 NOMBRE_REFERENCIAL
6 from objeto_aplicacion
7 where
RAMA
-----
LOCALIZACION_PADRE      ESQUEMA_PADRE      TIPO_PADRE      NOMBRE
DB      APLICACION_ANALISIS  PROCEDURE      POSINI      POSFIN      RES_AMB_OBJ
ARCHIVO
-----
C:\APLICACION\repositorio\db\objetos.sql      12120      12791
1
2 CREATE OR REPLACE
3 procedure APLICACION_ANALISIS.res_amb_obj ( pcodaplicacion number , pcodobjetoinc number, pcodobje
4 begin

```

Figura 12. Tipo de consulta textual

b. Relacional. Este tipo de consulta se realiza entre dos elementos de análisis. La consulta nos provee información de los tipos de relaciones existente entre la rama jerárquica que arman los dos elementos de análisis (ver Figura 13).

respuesta.txt - Notepad

	RAMA	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ
DB	LOCALIZACION_PADRE	ESQUEMA_PADRE	TIPO_PADRE	NOMBRE_PADRE
	RAMA	APLICACION_ANALISIS	PROCEDURE	RES_AMB_OBJ

Figura 13. Tipo de consulta relacional

c. **Gráfica.** Este tipo de consulta es una vista gráfica de la información que se obtiene al realizar el tipo de consulta relacional. La estructura de matemática discreta que se forma es un grafo (ver Figura 14).

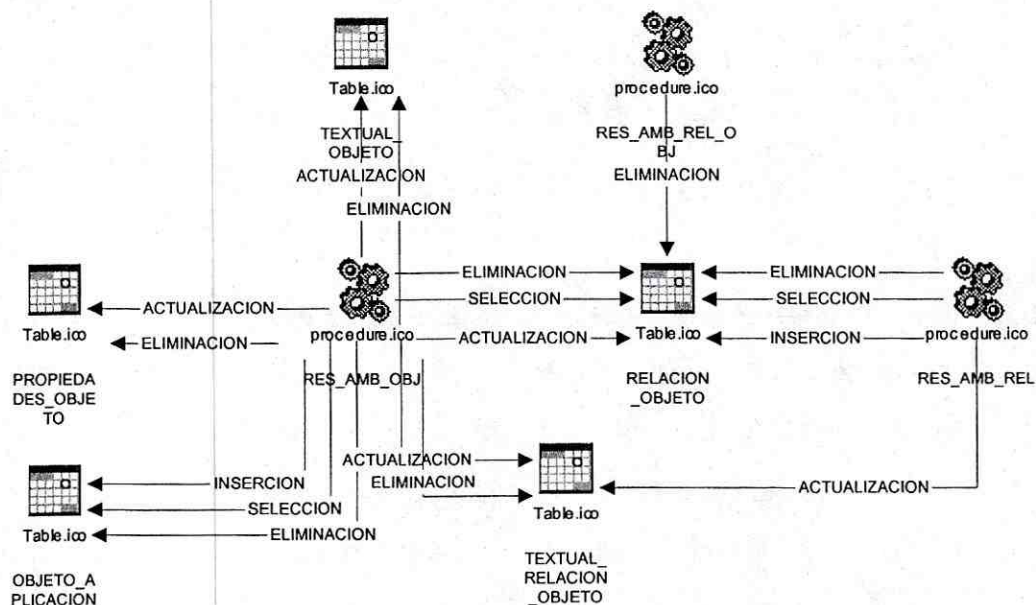


Figura 14. Tipo de consulta gráfica

2. Implementación del cliente de análisis. El componente de cliente de análisis fue desarrollado en la herramienta de Microsoft Visual Basic .Net.

Se elaboró un cliente genérico de consulta que implementa los diferentes tipos de consultas descritos anteriormente. El cliente básicamente implementa dos tipos de pantallas de consultas que son parametrizables.

A continuación se describe el contenido de cada pantalla realizada en la herramienta que forma parte del cliente de análisis.

a. Pantalla de conexión. En la Figura 15 se visualiza la pantalla de conexión del cliente de análisis para poder acceder a todas las demás pantallas de consultas y resultado. En esta pantalla se revisa la conexión al repositorio de consulta y se escoge la aplicación a consultar. Se utiliza de igual manera que en el cliente de carga un archivo con la cadena de conexión. Se recomienda utilizar el usuario CONSULTA_ANALISIS en el momento de hacer la conexión. Al presionar el botón identificado con la leyenda de consultar, aparecerá la pantalla de consultas y resultado(ver Figura 16).

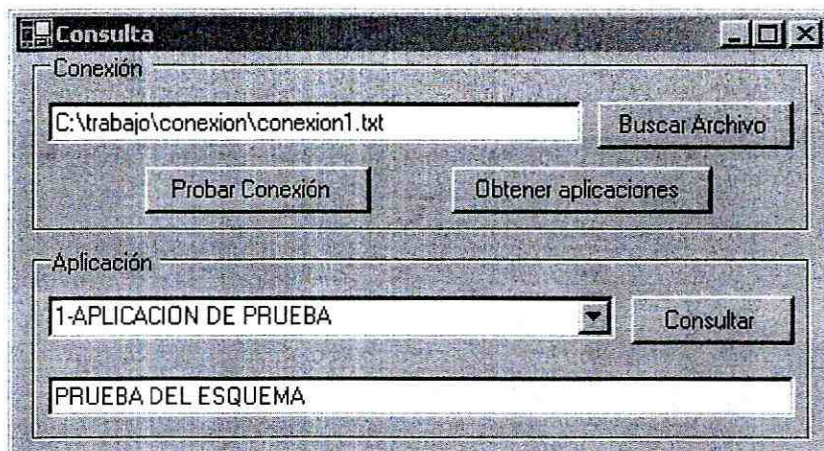


Figura 15. Pantalla de conexión cliente de análisis

b. Pantalla de consultas y resultados. En la Figura 16, Figura 18 y Figura 19 se visualiza la misma pantalla de tres diferentes formas. Las primeras dos formas son para realizar consultas y la tercera es para desplegar el resultado de las consultas.

Para moverse en esta pantalla se presionan las pestañas. Esta pantalla tiene tres pestañas, las primeras dos de izquierda a derecha son para realizar consultas

parametrizables y la tercera muestra el resultado de las consultas, como se mencionó con anterioridad.

1) Primera pantalla de consulta. Ésta se utiliza para obtener información de los elementos de análisis. Las diferentes opciones de filtro de búsqueda que se pueden realizar son:

a) Tipo de elemento. En esta casilla se ingresa el tipo de elemento de análisis por el que se desea hacer el filtro. Si la casilla se deja en blanco indica cualquier tipo de elemento de análisis. En el caso de escribir un valor en la casilla se puede utilizar el símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

También, se puede llenar una lista de valores separados por coma, no deben dejarse espacios en blanco entre los valores y las comas. Estos valores deben corresponder a tipos de elementos de análisis.

Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes tipos de elementos de análisis.

Esta opción se observa en la parte superior izquierda en la pantalla que se encuentra en la Figura 16.

b) Nombre del elemento. En esta casilla se ingresa el nombre del elementos de análisis por el que se desea hacer el filtro. Si la casilla se deja en blanco indica cualquier nombre del elemento de análisis. En el caso de escribir un valor en la casilla se puede utilizar el símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

También, se puede llenar una lista de valores separados por coma, no deben dejarse espacios en blanco entre los valores y las comas. Estos valores deben corresponder a nombres de elementos de análisis.

Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes nombres de elementos de análisis.

Esta opción se observa en la parte superior derecha de la pantalla que se encuentra en la Figura 16.

c) Filtros sobre comparaciones en la propiedades y combinación de estas comparaciones. La comparación de una propiedad se construyen tomando la propiedad y

comparando un valor con un operador relacional o tomando la propiedad y utilizando un operador unitario.

Los operadores unitarios son: existe la propiedad (*EXISTS*) y no existe la propiedad (*NOT EXISTS*), y los operadores relacionales son: igual (=), no igual (!=), se parece (LIKE), no se parece (NOT LIKE).

En los operadores relacionales que verifican si se parece o no se parece el valor de la propiedad a un valor determinado, este valor determinado puede llevar un símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

Para la elaboración de comparaciones entre propiedades se tiene un mantenimiento para agregar, borrar, y actualizar estas comparaciones. También, se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre las propiedades y los operadores. Esta opción de mantenimiento está en medio de la pantalla que se encuentra en la Figura 16.

Además, estas comparaciones de propiedades se pueden combinar con operadores lógicos. La selección del operador lógico depende de la escogencia en la opción de combinación de propiedades. Hay dos opciones en la combinación de propiedades, las cuales son disyunción y conjunción. La opción de disyunción indica que el operador lógico es el “o” lógico, y en el caso de la opción de conjunción indica que el operador lógico es el “y” lógico. Esta opción de combinación de propiedades se encuentra en la parte inferior central de la pantalla (ver Figura 16).

d) Alternativas para obtener diferentes tipos de información. La opción tiene tres diferentes tipos de alternativas: definición, propiedades, y textual. Esta opción indica el tipo de información a ser desplegada de los elemento de análisis filtrados por las opciones de búsquedas expuestos anteriormente. El tipo de información desplegada se describe en el siguiente cuadro.

Valor	Información
Definición	El esquema, tipo del elemento, nombre del elemento, padre del elemento, y camino referencial
Propiedad	Atributos del elemento de análisis y la misma información del valor definición
Textual	Texto original del archivo fuente del elemento de análisis y la misma información del valor definición

Cuadro 10. Tipo de información desplegada por cada alternativa

Esta opción se observa en la parte inferior izquierda de la pantalla en el recuadro cuyo nombre es Opciones que se encuentra en la Figura 16.

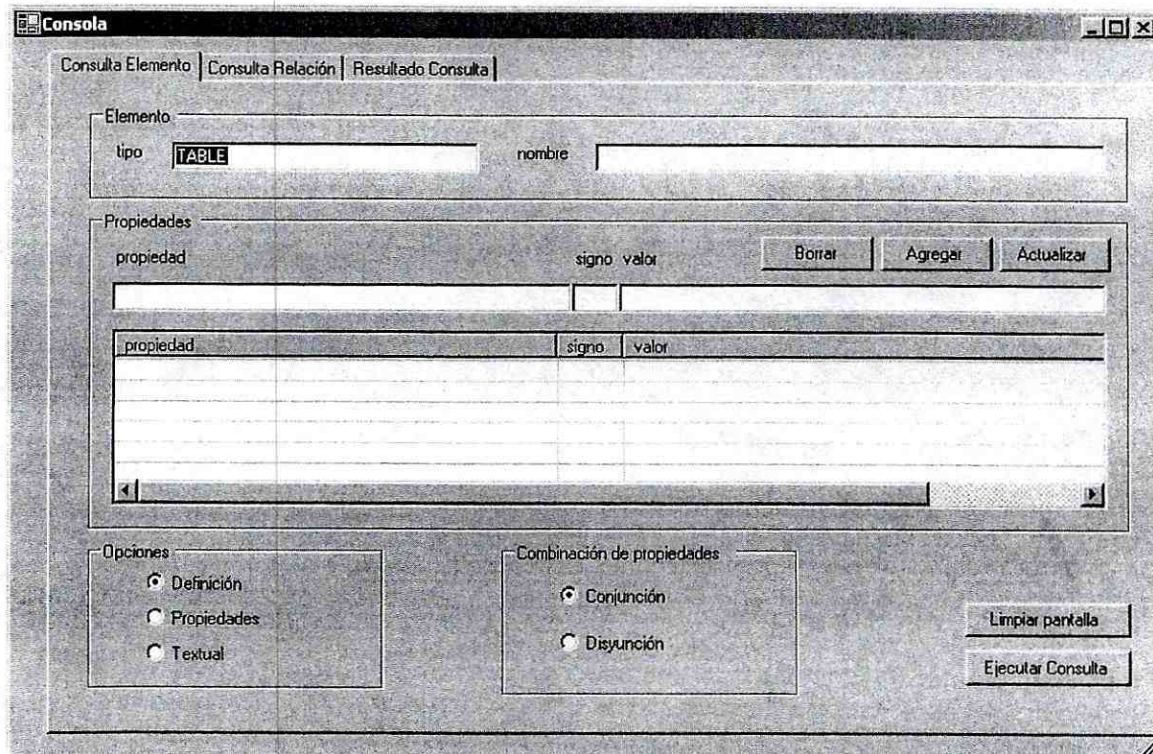


Figura 16. Primera pantalla de consulta

2) Segunda pantalla de consulta. La segunda pantalla de consulta se utiliza para obtener información de las relaciones entre los elementos de análisis. En esta pantalla se implementaron los diferentes tipos de consultas, así como los diferentes tipos de análisis de dependencias que serán expuestos posteriormente.

A continuación se exponen diferentes partes que al fusionarlas representan todo lo que se puede hacer con la segunda pantalla de consulta.

a) Filtros de búsqueda. Los filtros de búsqueda que se pueden utilizar en la segunda pantalla de consulta se describen a continuación:

(i) Tipo de elemento padre. En esta casilla se ingresa el tipo de elemento de análisis para el elemento padre. Si la casilla se deja en blanco indica cualquier tipo de elemento de análisis. En el caso de escribir un valor en la casilla se puede utilizar el símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

También, se puede llenar una lista de valores separados por coma, no deben dejarse espacios en blanco entre los valores y las comas. Estos valores deben corresponder a tipos de elementos de análisis.

Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes tipos de elementos de análisis.

Esta opción se observa en la parte superior izquierda del recuadro cuyo nombre es Elemento Padre de la pantalla que se encuentra en la Figura 18.

(ii) Tipo de elemento hijo. En esta casilla se ingresa el tipo de elemento de análisis para el elemento hijo. Si la casilla se deja en blanco indica cualquier tipo de elemento de análisis. En el caso de escribir un valor en la casilla se puede utilizar el símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

También, se puede llenar una lista de valores separados por coma, no deben dejarse espacios en blanco entre los valores y las comas. Estos valores deben corresponder a tipos de elementos de análisis.

Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes tipos de elementos de análisis.

Esta opción se puede ver en la parte superior izquierda del recuadro cuyo nombre es Elemento Hijo de la pantalla que se encuentra en la Figura 18.

(iii) Nombre del elemento padre. En esta casilla se ingresa el nombre del elemento de análisis para el elemento padre. Si la casilla se deja en blanco indica cualquier nombre de elemento de análisis. En el caso de escribir un valor en la casilla se puede utilizar el símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

También, se puede llenar una lista de valores separados por coma, no deben dejarse espacios en blanco entre los valores y las comas. Estos valores deben corresponder a nombres de elementos de análisis.

Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes nombres de elementos de análisis.

Esta opción se observa en la parte superior derecha del cuadro cuyo nombre es Elemento Padre de la pantalla que se encuentra en la Figura 18.

(iv) Nombre del elemento hijo. En esta casilla se ingresa el nombre del elemento de análisis para el elemento hijo. Si la casilla se deja en blanco indica cualquier nombre de elemento de análisis. En el caso de escribir un valor en la casilla se puede utilizar el símbolo de porcentaje para indicar cualquier ocurrencia de caracteres.

También, se puede llenar una lista de valores separados por coma, no debe dejarse espacios en blanco entre los valores y las comas. Estos valores deben corresponder a nombres de elementos de análisis.

Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes nombres de elementos de análisis.

Esta opción se puede ver en la parte superior derecha del cuadro cuyo nombre es Elemento Hijo de la pantalla que se encuentra en la Figura 18.

(v) Parámetro de hijo último. El parámetro de hijo último utiliza las condiciones ingresadas en las opciones de tipo y nombre del elemento hijo, indicando que la búsqueda no para en el momento de encontrar el primer elemento hijo que cumpla con la condición especificada en dicho elemento, sino que en la rama de búsqueda que se forma se encuentre el último elemento hijo que cumple con la condición.

Este parámetro se muestra en la parte inferior central, en el cuadro cuyo nombre es Parámetros que se encuentra en la pantalla, ver Figura 18. Si no se especifica este tipo de parámetro, el cual es el valor por omisión, la búsqueda para en el momento que se encuentre el primer elemento hijo que cumpla la condición especificada en ese elemento hijo.

(vi) Parámetro de nivel. El valor del parámetro de nivel es igual a cero por omisión. El valor cero indica que no hay restricción en el número de niveles en profundidad que puede realizar el algoritmo de búsqueda con base en los filtros. En el caso que el valor sea un número entero positivo, indica la profundidad máxima para hacer la búsqueda.

Esta opción se muestra en la parte inferior central en el cuadro cuyo nombre es Parámetros que se encuentra en la pantalla (ver Figura 18).

(vii) Listado de relaciones. Este listado está conformado por la

escogencia de los diferentes tipos de relaciones de los elementos de análisis seleccionados e indica las posibles relaciones donde se puede hacer la búsqueda. Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes tipos de relaciones de elementos de análisis.

Para hacer la lista hay un mantenimiento que tiene las siguientes acciones: borrar, agregar, actualizar que aplican a los elementos de la lista. Esta opción se visualiza en la parte izquierda en medio de la pantalla (ver Figura 18).

(viii) Listado de existencia de relaciones. Este listado está integrado por la escogencia de los diferentes tipos de relaciones de los elementos de análisis seleccionados e indica la existencia de la relación en la rama de búsqueda. Se puede presionar la tecla de escape para que muestre una pantalla que sirva de escogencia entre los diferentes tipos de relaciones de elementos de análisis.

Para hacer la lista hay un mantenimiento que tiene las siguientes acciones: borrar, agregar, actualizar que aplican a los elementos de la lista. Esta opción se visualiza en la parte derecha en medio de la pantalla (ver Figura 18).

(ix) Tipos de consultas. Esta opción indica el tipo de información a ser desplegada de la relación entre los elementos de análisis y tiene los siguiente valores: Definición, Textual y Gráfico. Cada una de estos valores tiene un equivalente como tipo de consulta expuesto en la sección Tipos de consultas de los elementos de análisis seleccionados de este capítulo. Por ejemplo, el valor “definición” representa la consulta relacional, el valor “textual” representa la consulta textual y el valor “gráfico” representa la consulta gráfica. El tipo de información desplegada se describe en el siguiente cuadro.

Valor	Información
Definición	El esquema del elemento padre, tipo del elemento padre, nombre del elemento padre, esquema del elemento hijo, tipo del elemento hijo, nombre del elemento hijo, tipo de relación
Textual	Texto original del archivo fuente de la relación entre los dos elementos de análisis y la misma información del valor definición
Gráfica	Al ejecutar estas consulta grafica, se genera un archivo de texto formateado con las ramas encontradas que cumplen los criterios de búsqueda. Luego, la herramienta de modelación Visio se encarga de interpretar y dibujar este archivo, obteniéndose un grafo de dependencias

Cuadro 11. Tipo de información desplegada por cada valor

Esta opción con sus valores es visualizado en la parte inferior izquierda del recuadro cuyo nombre es Opciones que se encuentra en la pantalla (ver Figura 18).

En el caso que se seleccione el valor gráfico, quedan habilitados los siguientes parámetros: forma, línea e íconos. El parámetro de forma tiene el valor que representa una estructura discreta de un árbol, el parámetro línea tiene el valor que especifica líneas rectas y el parámetro íconos representa si se utilizan o no los íconos que representan los elementos de análisis. Estos filtros gráficos son visualizados en la parte inferior central en el cuadro cuyo nombre es Gráfico que se encuentra en la pantalla, ver Figura 18.

Una vez generada la vista gráfica en la herramienta de modelación Visio 2002. Se puede utilizar esta misma herramienta para realizar una personalización del diagrama a gusto del usuario. Para ello se recomienda utilizar la opción *Lay Out Shapes*.

b) Formatos de archivos de salida. En esta sección se describen los diferentes archivos que se pueden generar por la herramienta desarrollada. Existen 3 tipos de formatos de archivos que se detallan a continuación:

(i) Archivo de consulta relacional. Este archivo se genera en el tipo de consulta relacional. El nombre de este archivo es "respuesta.txt". Éste se puede visualizar en Word, Notepad o Wordpad, se recomienda cambiar el tipo de letra a *New Courier*, ya que este tipo de letra le asigna el mismo tamaño a todos los símbolos. De este archivo se puede desplegar las ramas de búsquedas encontradas al realizar la consulta y aplicar los filtros de búsqueda.

Cada rama tiene un conjunto de relaciones. La información que se puede encontrar por relación en una fila del archivo es: el esquema del elemento padre, tipo del elemento padre, nombre del elemento padre, esquema del elemento hijo, tipo del elemento hijo, nombre del elemento hijo, y tipo de relación. La información en la rama tiene un orden donde el hijo de una relación es el padre de la siguiente relación. Este archivo se muestra en la Figura 13.

(ii) Archivo de consulta textual. Se genera en el tipo de consulta textual de la primera pantalla o de la segunda pantalla. El nombre de este archivo es "respuesta.txt". El cual se puede ver en Word, Notepad o Wordpad, se recomienda

cambiar el tipo de letra a *New Courier*, ya que éste le asigna el mismo tamaño a todos los símbolos. De este archivo se puede observar el texto original de donde proviene el elemento de análisis o la relación, según sea el tipo de pantalla de consulta que se utilice. Este archivo se muestra en la Figura 12.

(iii) Archivo de consulta gráfica. Se genera en el tipo de consulta gráfica. La herramienta de modelación de Visio lo utiliza para dibujar el grafo de dependencias que resultó de la consulta realizada al aplicar los filtros de búsqueda. El nombre de este archivo es “consulta.txt”. El formato de la fila de este archivo de texto es un comando seguido de argumentos separados por tabuladores. Los argumentos pueden ser escritos entre dobles comillas cuando llevan espacios en blanco. Entre los comandos están:

- *Template* es el encargado de indicar en dónde se encuentra el esténcil. El esténcil es una librería externa que contiene figuras, cada figura recibe el nombre de *Master*. La lista del comando y sus argumentos es: *Template* y ubicación de donde se encuentra el esténcil.
- *Master* es el nombre de un elemento específico que se encuentra en el esténcil. La lista del comando y sus argumentos es: *Master*, identificador único del *master*, nombre del *master* en el esténcil, y nombre del esténcil.
- *Shape* es el nombre del nodo que se va dibujar y tiene un *master* asociado. La lista del comando y sus argumentos es: *Shape*, identificador único de *shape*, nombre del *master*, texto, posición vertical, posición horizontal, ancho, altura, costo, duración, e información del recurso.
- *Link* es la conexión entre dos nodos y tiene un *master* asociado. La lista del comando y sus argumentos es: *Link*, identificador único del conector, nombre del *master*, identificador del *shape* del que sale, identificador del *shape* al que va.

En la Figura 17 se visualizan los comandos anteriores. Se especifican los elementos de análisis tipo tabla cuyos nombre son B y PAR, tipo columna cuyos nombres son BCC, BCF, BCN, BCC y PAD, tipo procedimiento con nombre BOL y tipo función con nombre CUENTA y ROUND. Además, se especificaron relaciones de CONTIENE y LLAMA.

```

consultav1.txt - Notepad
File Edit Format Help
Template Basic Flowchart.vst
PlacementStyle 1
Gridding 0
RoutingStyle 5
Master "TABLE" C:\aplicacion\template\elemento.vss
Master "COLUMN" C:\aplicacion\template\elemento.vss
Shape "S1" "TABLE" "B"
Shape "S2" "COLUMN" "BCC"
Link "L1" "dynamic connector" "CONTIENE" "S1" "S2"
Shape "S3" "COLUMN" "BCF"
Link "L2" "dynamic connector" "CONTIENE" "S1" "S3"
Shape "S4" "COLUMN" "BCN"
Link "L3" "dynamic connector" "CONTIENE" "S1" "S4"
Shape "S5" "COLUMN" "BCC"
Link "L4" "dynamic connector" "CONTIENE" "S1" "S5"
Shape "S6" "TABLE" "PAR"
Shape "S7" "COLUMN" "PAD"
Link "L5" "dynamic connector" "CONTIENE" "S6" "S7"
Master "PROCEDURE" C:\aplicacion\template\elemento.vss
Master "FUNCTION" C:\aplicacion\template\elemento.vss
Shape "S8" "PROCEDURE" "BOL"
Shape "S9" "FUNCTION" "CUENTA"
Link "L6" "dynamic connector" "LLAMA" "S8" "S9"
Shape "S10" "FUNCTION" "ROUND"

```

Figura 17. Archivo generado por consulta gráfica

c) **Algoritmo de búsqueda.** En la segunda pantalla de consulta se implementó un algoritmo de búsqueda recursivo en profundidad. El algoritmo determina cuáles son los elementos padres para iniciar la búsqueda, estos elementos son los que cumplen las condiciones de los filtros de elemento padre. Con el elemento padre se buscan los elementos hijos utilizando las relaciones, ya que cada relación tiene asociado un elemento padre, un elemento hijo y un tipo de relación. En este momento es donde se hace el filtro con la lista almacenada de la opción de relaciones, si es que hubiera elementos en esta lista.

Luego, con los elementos hijos encontrados se revisa las condiciones del filtro de elemento hijo. Si no se cumplen, se verifica que no se haya pasado del nivel máximo de búsqueda en profundidad y en caso de no haber una violación a la condición anterior se convierte cada elemento hijo de ese nivel en elemento padre y así sucesivamente formando la rama de búsqueda.

Si se cumplen las condiciones de elemento hijo, como resultado se tiene una rama de búsqueda y en ella se cheque la lista de existencia de relación, para determinar si en esa rama se encuentra alguna relación dentro de esa lista.

Las revisiones de las condiciones del filtro de elemento hijo se llevan a cabo cuando no está habilitada la opción de hijo último. En caso de que se encuentre habilitada

esta opción no se hacen las revisiones de las condiciones del filtro del elemento hijo, sino se deja que se vaya construyendo la rama hasta que encuentre un elemento que no tiene hijos o la rama se cortó al detectar nodos que ya se visitaron en esa misma rama. En este momento se hacen las revisiones de las condiciones de filtros de un elemento hijo desde el último nodo encontrado en la rama, ascendiendo en la rama hasta encontrar un elemento que cumpla las condiciones anteriores y así podar la rama.

Para evitar que la recursión no entre en un lazo infinito en cada rama, se registran los nodos que se han visitado por cada rama. Cada vez que se realice la búsqueda en un nodo se verifica si ya se visitó ese nodo. En caso de que el nodo se haya visto en esa rama, no se realiza la búsqueda y, con ello, se rompe el lazo infinito.

El orden del tiempo de ejecución del algoritmo es exponencial, en el peor de los casos. Para hacer un análisis del tiempo de ejecución hay que hacer un conteo sobre las ramas, en el peor de los casos se construirán todas las ramas. La cuota superior se realizará con los siguientes factores: "A" es el número máximo de tipo de relaciones entre cualesquiera dos tipo de elementos de análisis, "B" es el número máximo de tipo de elementos de análisis hijos para la combinación de un tipo de elementos de análisis padre y sus tipos de relación, "V" es el número máximo de instancias de elementos de análisis agrupados por su tipo y "W" es el número de instancias de elementos de análisis. El conteo de ramas de altura "i" es acotada por la expresión $WA^i(VB)^i$. Nótese que cada una rama de altura "i" tiene "i+1" vértices e "i" aristas. El primer elemento de la rama "i" puede ser seleccionado entre "W" elemento, el resto de los vértices son elementos hijos que pueden ser seleccionados a lo más en VB. Cada arista identifica una relación y el número máximo de relaciones entre dos elementos es a lo más "A". Del argumento anterior se deduce la expresión que sirve de cuota para el conteo de ramas de altura "i". La suma de ramas es acotada por la expresión $\sum_{i=1..W} WA^i(VB)^i = W((AVB)^{W+1} - 1) / (AVB - 1)$ con $AVB > 1$, el contador de la suma llega hasta W porque en el peor la rama más grande es conformada por la lista de los W elementos y el algoritmo poda la rama para evitar la recursión. Asumiendo que el tiempo de ejecución de las consultas a la base de datos sea constante. Se tendría que el tiempo de ejecución del algoritmo en el peor de los casos es $O(W(AVB)^W)$ con $AVB > 1$.

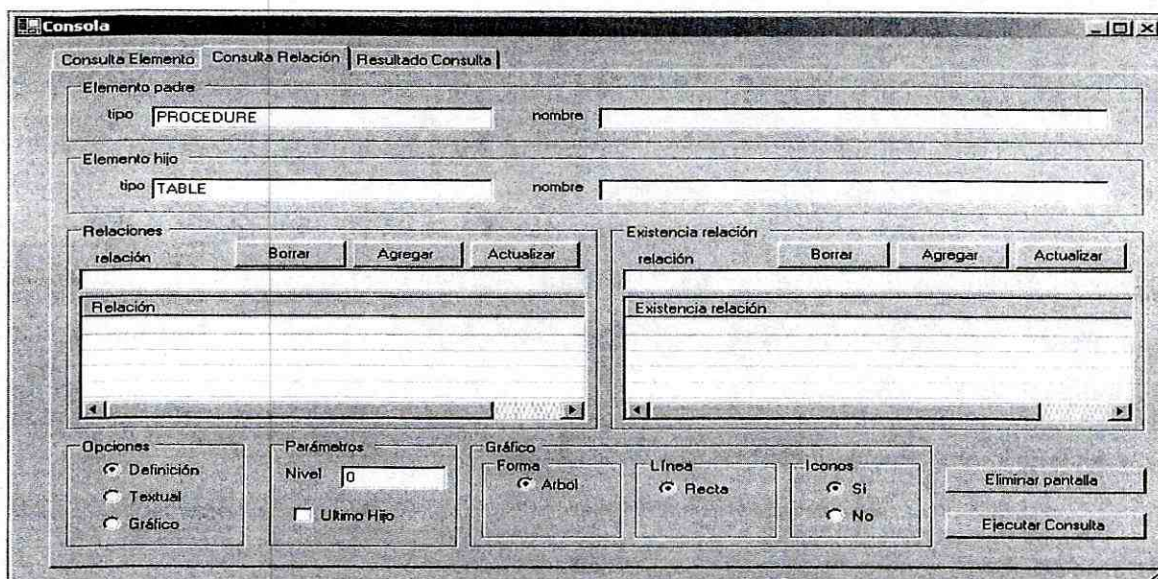


Figura 18. Segunda pantalla de consulta

3) **Pantalla del resultado.** El cliente de análisis tiene una pantalla para visualizar la respuesta obtenida por una consulta. Cuando la respuesta contiene mucha información se genera un archivo texto de respuesta (“respuesta.txt”). Este archivo se llena con base a los formatos especificados por las dos pantallas de consultas antes descritos. La pantalla de respuesta se puede ver en la Figura 19, indicando el camino de la ubicación del archivo texto de respuesta.

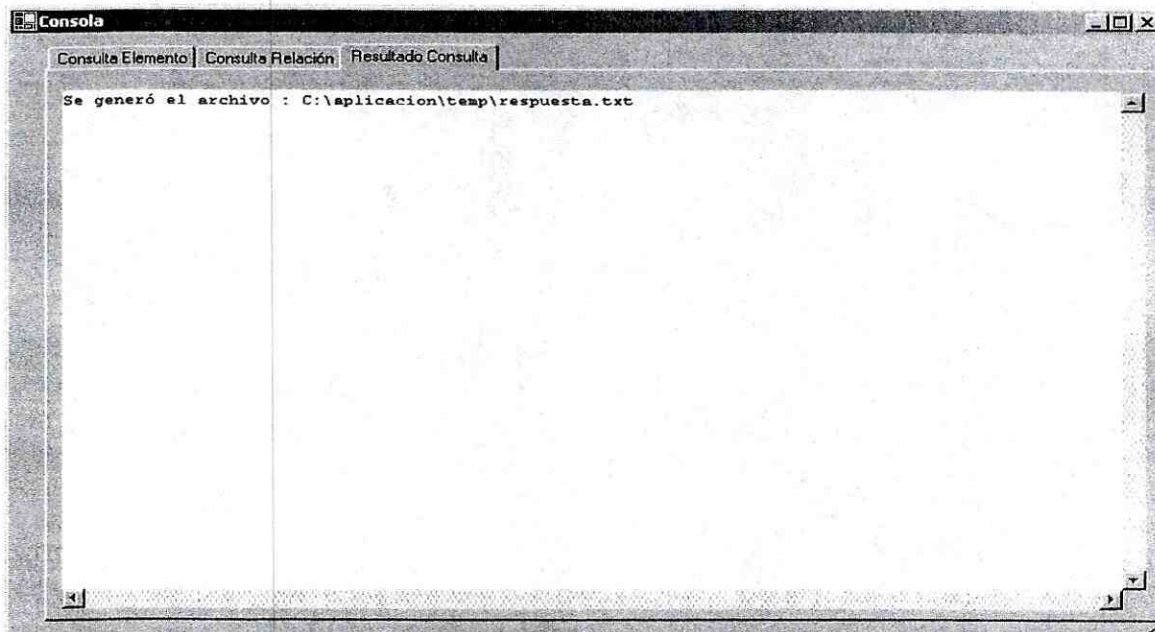


Figura 19. Pantalla de respuesta

a. Referencias simples. La referencia simple consiste en poner como pivote un elemento de análisis y dejar libre el otro elemento. Este elemento que sirve como pivote puede aparecer como elemento padre o hijo. En el caso que aparezca como elemento padre, la consulta relacional generará todos los elementos que son referenciados por él. De igual manera, en el caso que aparezca como elemento hijo, la consulta relacional generará todos los elementos que son referidos o que lo referencian a él.

Para ejemplificar su uso primero se coloca el elemento de análisis como padre. Esto se muestra en la pantalla de consulta (ver Figura 21), y también se muestra la respuesta de la consulta (ver Figura 22).

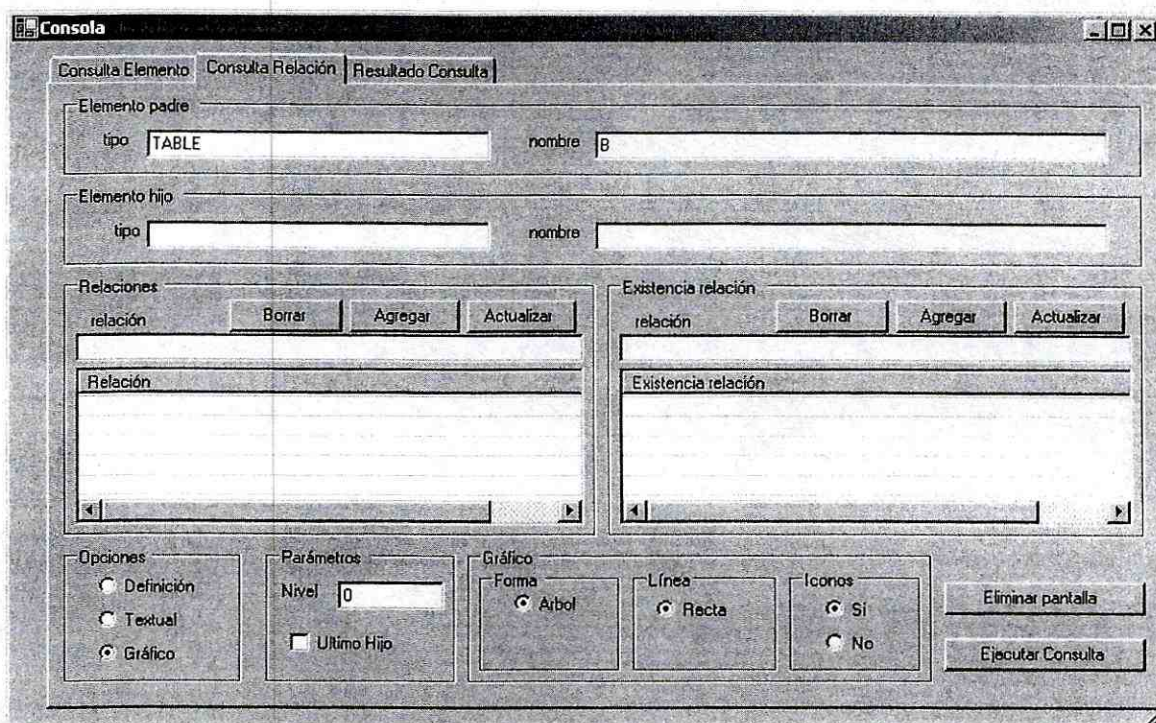


Figura 21. Pantalla de consulta del elemento padre

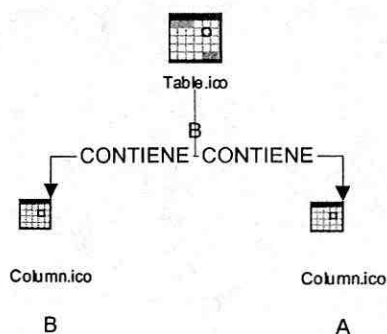


Figura 22. Resultado de la consulta del elemento padre

De igual manera, se coloca el elemento de análisis como hijo. Esto se muestra en la pantalla de consulta (ver Figura 23), y también se muestra la respuesta de la consulta (ver Figura 24).

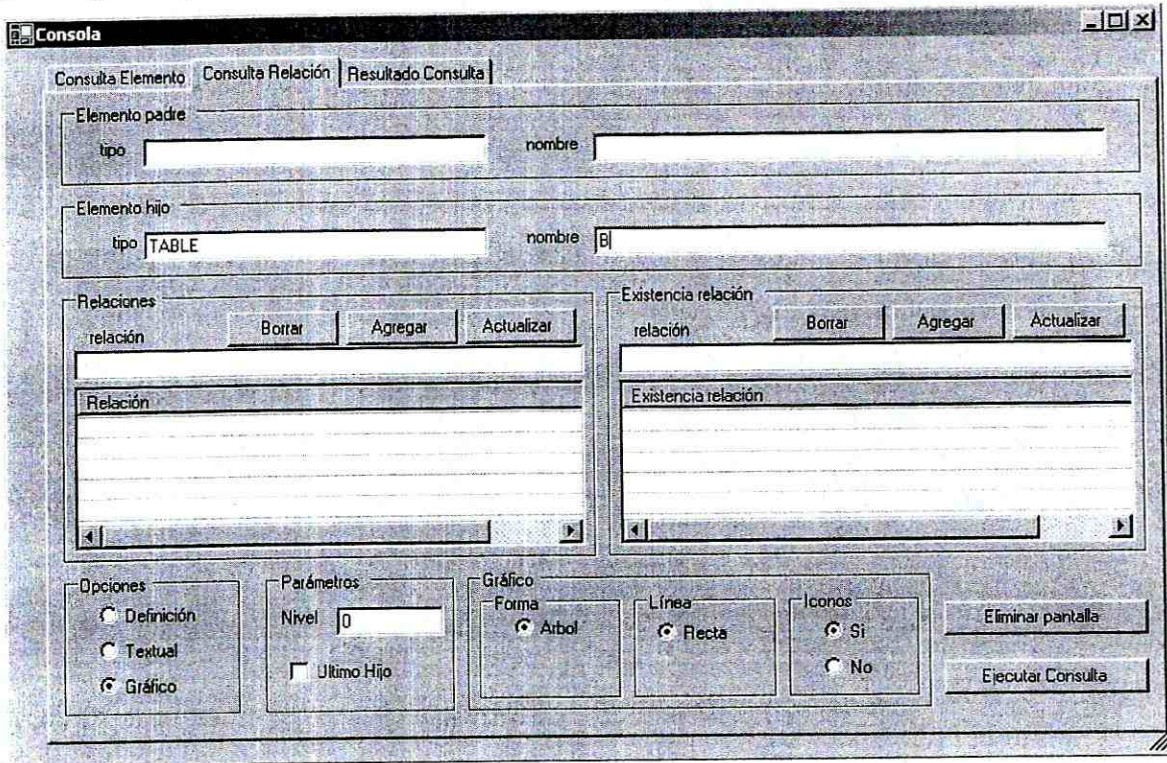


Figura 23. Pantalla de consulta del elemento hijo

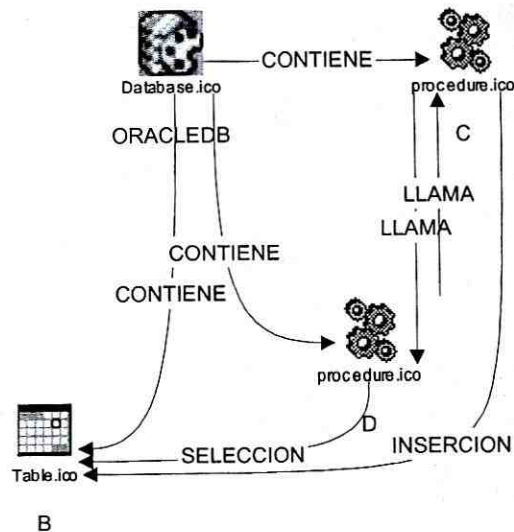


Figura 24. Resultado de consulta del elemento hijo

b. Referencias cruzadas. ésta se lleva a cabo cuando se fijan los elementos de análisis. En este caso la consulta relacional generará todos los elementos que son

referenciados por el elemento padre y que referencian al elemento hijo. Para mostrar su uso, se llenan las casillas del elemento padre e hijo en la pantalla de consulta (ver Figura 25). La respuesta de esta consulta se muestra en la Figura 26.

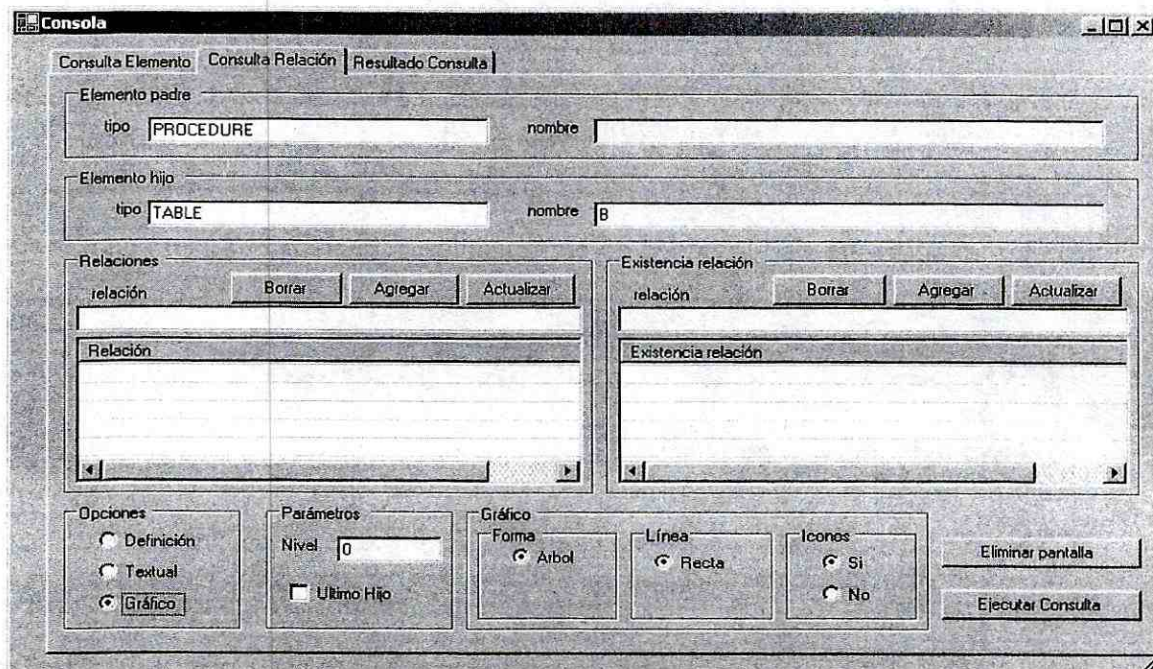


Figura 25. Pantalla de consulta de referencia cruzada

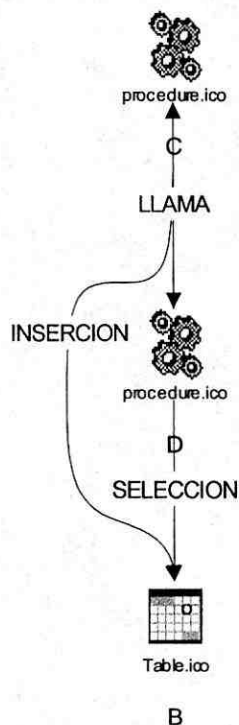


Figura 26. Resultado de la consulta de referencia cruzada

c. Quitar elementos. Al hacer dos referencias simples de un mismo elemento, en donde la primera referencia simple es un elemento padre, y la segunda referencia simple es un elemento hijo y no devuelve ninguna información, entonces dicho elemento no referencia y no es referenciado por ningún otro elemento. Por lo tanto, dicho elemento se puede quitar.

Las pantallas de consulta se muestran en la Figura 27 y Figura 29 . Los resultados de las consultas se muestran en la Figura 28 y Figura 30, respectivamente. A pesar de que se obtiene información en los resultados de las consultas, este elemento es un candidato para eliminación, ya que todo elemento tiene un elemento padre que es la base de datos y toda tabla tiene que tener columnas por su definición. Por lo cual no se provee información extra significativa.

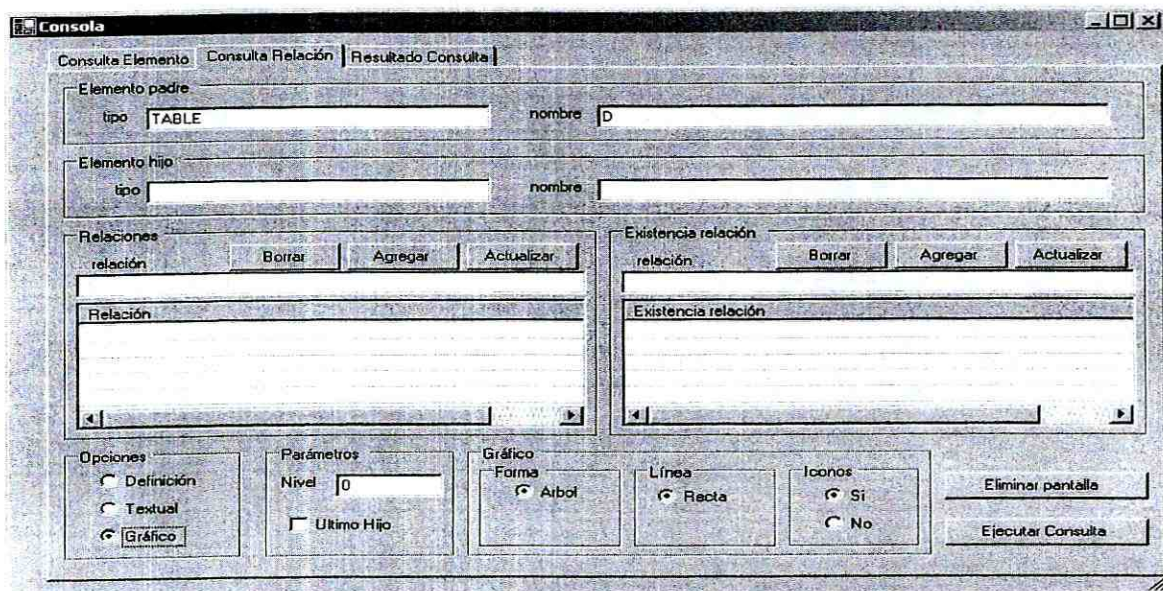


Figura 27. Pantalla de consulta del elemento padre

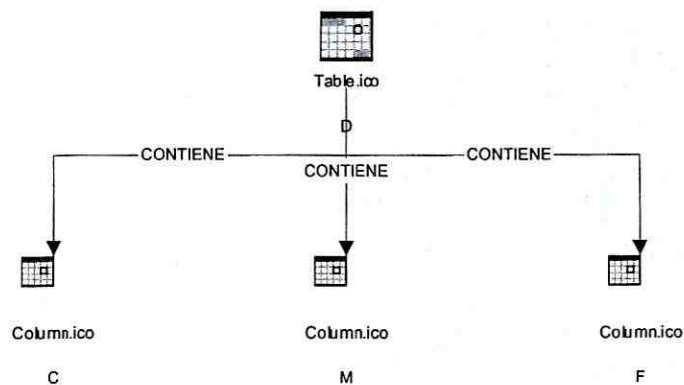


Figura 28. Resultado de la consulta del elemento padre

Figura 29. Pantalla de consulta del elemento hijo

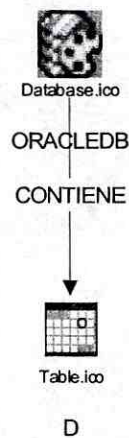


Figura 30. Resultado de la consulta del elemento hijo

d. Referencias recursivas. Es cuando al realizar una referencia cruzada con respecto al mismo elemento ésta devuelve información. Puede existir recursión directa e indirecta. La recursión directa se da cuando la referencia es inmediata. Y la recursión es indirecta cuando en la referencia hay intermediarios que son referenciados y referencian al mismo elemento.

La referencia recursiva se muestra en la pantalla de consulta (ver Figura 31), y el resultado de la pantalla de consulta se muestra en la Figura 32. Además, observe que en la Figura 32 hay una recursión directa e indirecta. La recursión directa es visualizada por un arco en forma de cola de cerdo.

The screenshot shows a window titled 'Consola' with three tabs: 'Consulta Elemento', 'Consulta Relación', and 'Resultado Consulta'. The 'Consulta Elemento' tab is active, showing fields for 'Elemento padre' (tipo: PROCEDURE, nombre: C) and 'Elemento hijo' (tipo: PROCEDURE, nombre: C). Below these are two panes for 'Relaciones' and 'Existencia relación', each with 'Borrar', 'Agregar', and 'Actualizar' buttons. At the bottom, there are sections for 'Opciones' (Definition, Textual, Gráfico), 'Parámetros' (Level: 0, Ultimo Hijo), 'Gráfico' (Forma: Arbol, Línea: Recta), and 'Iconos' (Si, No). Buttons for 'Eliminar pantalla' and 'Ejecutar Consulta' are also present.

Figura 31. Pantalla de consulta de referencia recursiva

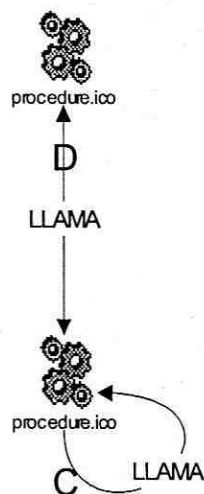


Figura 32. Resultado de consulta de referencia recursiva

E. Caso de estudio.

La herramienta desarrollada pretende ser un complemento en las prácticas de la empresa para analizar el impacto que tiene la realización de un requerimiento. Los requerimientos que se adecuan correctamente para el uso de la herramienta son los que, de alguna manera, se pueden expresar con base a los elementos de análisis y sus relaciones. Para ejemplificar se documenta el caso de estudio.

1. Ambiente. Para la captura de información contable de ciertas unidades de trabajo se utiliza una aplicación externa. Esta aplicación externa sirve de puente entre los sistemas informáticos de la unidad de trabajo y los sistemas informáticos de la administración (ver Figura 33).

Esta aplicación externa ejecuta procedimientos almacenados en la base de datos e ingresa información en tablas temporales. Estos procedimientos forman parte del sistema contable. La información que captura la aplicación externa genera partidas contables. Además, los procedimientos almacenados registran impuestos y generan resúmenes de información de las partidas contables.

La interfase expuesta anteriormente ha estado trabajando varios meses. Se tiene como objetivo que toda la información fluya a través de la misma y que ya no se estén realizando cargas manuales de información por parte del usuario para alimentar el sistema informático de la administración.

A esta interfase se van agregando de manera continua unidades de trabajo para el envío de la información. Apenas se cuenta con un pequeño número de unidades de trabajo que están haciendo uso de la interfase.

Esta interfase fue desarrollada por una empresa externa con productos de Microsoft y Oracle, y no se cuenta con una documentación adecuada para darle un soporte apropiado por parte del departamento de sistemas de información de la empresa. Solamente se tienen los fuentes y la aplicación en producción. El sistema de la administración fue desarrollado con productos de Oracle.

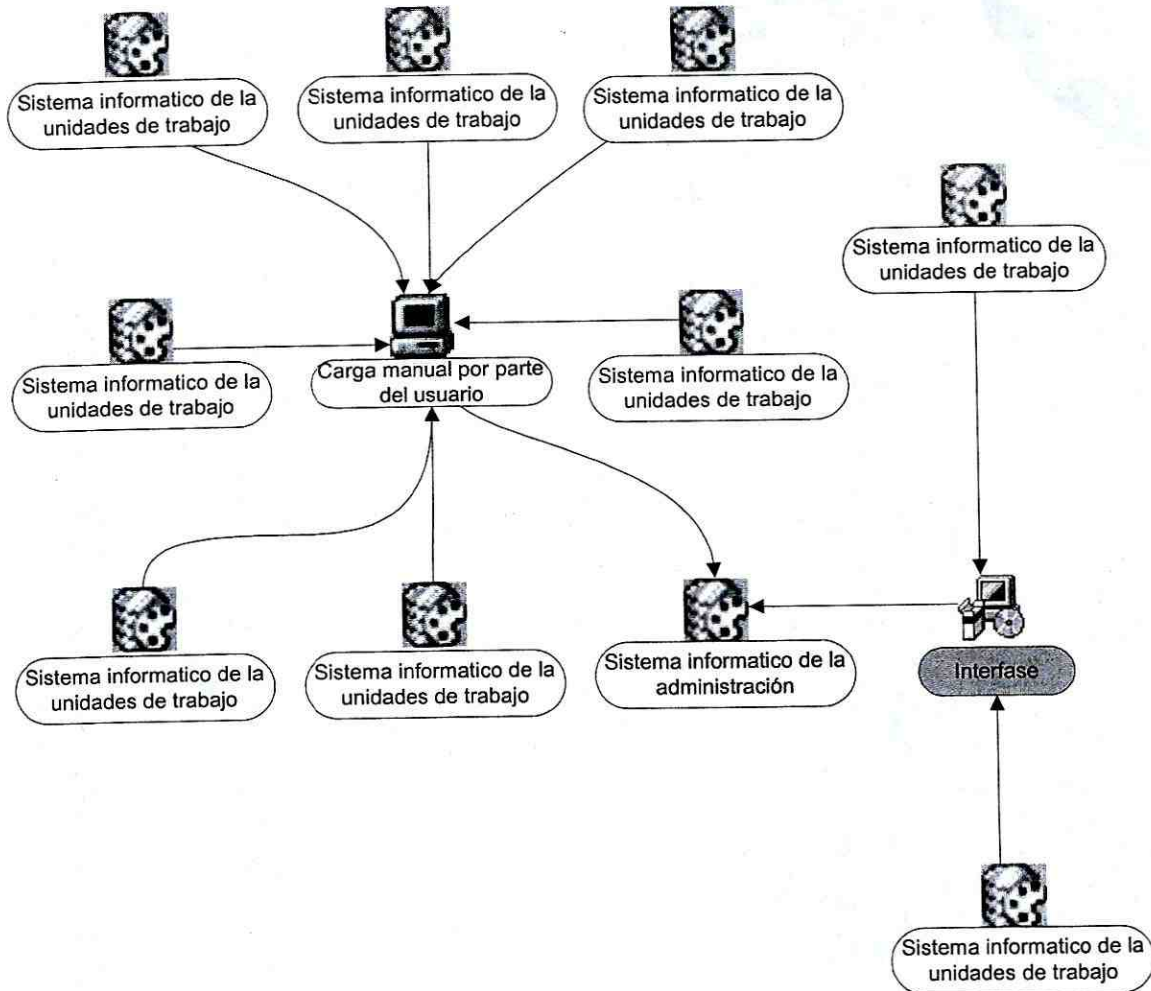


Figura 33. Interacción de la interfase con sistemas informáticos

2. Requerimientos. Se detectaron los siguientes inconvenientes que obstaculizan lograr el objetivo expuesto anteriormente.

Lista de requerimientos:

1. En el caso de que la unidad de trabajo haya mandado más de una vez la misma información, ésta debe ser procesada una vez.
2. Si la información que mandan las unidades de trabajo no puede generar partidas contables cuadradas, éstas no deben reflejarse en el sistema informático de la administración.
3. Las personas encargadas de verificar o analizar la información necesitan que se les notifique el estatus del procesamiento de la información que viene de la unidad de trabajo.

4. Se determinó que ciertos reportes de análisis de información no se han podido generar a tiempo porque la información no se encuentra disponible, ésta se encuentra en la etapa de procesamiento, y la unidad de trabajo había mandado la información en los plazos estipulados para poder generar los reportes a tiempo.
5. Hay información que viene de la unidad de trabajo que no se está utilizando en el procesamiento de la información para el sistema informático de la administración, el uso de esta información contribuirá en la automatización de ciertos procesos.
6. Por cuestiones de actualizaciones de la ley de impuestos, se necesita que la unidad de trabajo envíe cierto detalle de información para poderse alimentar correctamente los libros del IVA.

3. Aplicación de la herramienta. Los resultados de la lista de los requerimientos serán expuestos en el rubro Aplicación de la herramienta con base en los requerimientos de la sección de RESULTADOS Y DISCUSIÓN. Por cada requerimiento se detalla la forma en que se utilizó la herramienta haciendo énfasis sobre los elementos de análisis seleccionados, los tipos de consultas y tipos de análisis de dependencias.

V. RESULTADOS Y DISCUSIÓN

Los objetivos fueron cumplidos en el momento de implementada la herramienta y el uso de la misma a un caso real. A continuación se detallan los rubros que apoyan lo alcanzado en los objetivos y la arquitectura que sirvió de base en la herramienta desarrollada.

A. Elaboración de clientes de carga

Las herramientas para analizar código fuente son dependientes del lenguaje de programación, y hay una diversidad de fuentes de donde se puede obtener la información o código. Por ello se plantea la necesidad de tener un repositorio de carga. Este repositorio es un directorio que se puede dividir en subdirectorios para clasificar los fuentes, ya que cada fuente generado por una herramienta externa tiene un formato específico. Para cada formato hay que implementar un analizador léxico y sintáctico para capturar la información de lo que se desee analizar.

Al analizador léxico y sintáctico dependiente de este formato de archivos se le llamó cliente de carga. El cliente de carga se implementa con base en los elementos de análisis de interés, para que registre y centralice la información en otro tipo de repositorio de fácil consulta.

En la herramienta desarrollada se implementó el cliente de carga, el cual realiza dos pasadas en los archivos de texto. La primera pasada es para registrar la información de definición de todos los elementos de análisis, y la segunda pasada se almacena la información de las relaciones. Se realizó de esta manera porque hay relaciones donde aún no se han definido los elementos de análisis y no se dependa del orden de las instrucciones que conforman los archivos de texto.

También se ilustra la idea de resolver ambigüedades, ya que no siempre se tiene la información depurada y como se necesita. La resolución de estos casos consiste en mostrar el elemento con problema y compararlo con posibles elementos de análisis. Se especificaron tres tipos de acciones para resolver la ambigüedad: elemento de análisis nuevo, eliminar el elemento, actualizarlo por un elemento de análisis seleccionado. De igual manera se resuelven las ambigüedades en el caso de las relaciones. De esta manera

se evita que algún tipo de información no se tome en cuenta. Además, se elimina cierta complejidad de programación al manejar los inconvenientes de esta forma.

Se pueden incorporar nuevos elementos de análisis al cliente de carga que se implementó. Se recomienda hacer una revisión exhaustiva del código del analizador léxico y sintáctico de la herramienta desarrollada para poder integrar los *tokens* y reglas gramaticales, y tomar en cuenta el diagrama entidad relación del repositorio de consulta a la hora de registrar estos elementos de análisis y sus relaciones. En caso que no se pueda incorporar de una manera fácil, se recomienda elaborar un nuevo cliente de carga, este cliente de carga debe mantener la interfaz con el repositorio de consulta para que pueda ser utilizado posteriormente por el cliente de análisis.

B. Importancia de la metadata

La metadata es un componente muy importante para el uso de cualquier herramienta de análisis. Porque la información resultante del análisis depende del tipo de detalle de la información de la metadata. Al no registrar la información necesaria no se pueden obtener resultados o conclusiones confiables. No se recomienda tener muchos elementos de análisis, únicamente los necesarios, porque dificulta el manejo de la información en las consultas, repercute en la implementación y tiempo de ejecución del cliente de análisis de la herramienta en los tipos de consultas, y afecta en la implementación y tiempo de ejecución del cliente de carga de la herramienta al procesar la información de la aplicación, ya que se registran elementos de análisis y relaciones indeseables.

En la herramienta que se implementó, se identificó la información a nivel de columna. Pero, según sea el caso, puede ser necesario identificar los elementos a nivel de tipo de dato. Por lo que se recomienda la elaboración de un nuevo cliente de carga donde se identifiquen los elementos de análisis necesarios. Otra opción es la incorporación de nuevos elementos de análisis al cliente de carga que se implementó. Además, se recomienda agrupar los elementos de análisis con sus relaciones que hagan sentido, o sean compatibles, o valga la pena analizar para elaborar clientes de análisis específicos para estas agrupaciones.

Los elementos de análisis seleccionados en la herramienta que se implementó se escogieron porque son los elementos que se encuentran frecuentemente en cualquier aplicación de base de datos, y sus relaciones son las relaciones más frecuentes y naturales de encontrar. Estas relaciones se encontraron en las instrucciones de actualización de SQL (DML), y de las llamadas de procedimientos y de funciones. De la aplicación de la herramienta desarrollada al caso real se ratifica la justificación expuesta anteriormente, y muchas de las recomendaciones y resultados detallados anteriormente en este rubro son aprendidos después de haber aplicado la herramienta desarrollada al caso real.

C. Cliente de análisis genérico

El repositorio de consulta tiene que tener la característica de facilidad de consulta y centralización de la información. Por ello se escogió como repositorio una base de datos relacional por su lenguaje de consulta SQL. En este repositorio se registra la metadata. La metadata consiste en los elementos de análisis y sus relaciones. Nótese que en el repositorio de consulta se puede registrar cualquier tipo de información de elementos y relaciones.

El cliente de análisis es un cliente genérico, está basado en dos tipos de pantallas para elaborar consultas parametrizables que obtienen información del repositorio de consulta. El primer tipo de pantalla de consulta únicamente se realiza sobre los elementos de análisis, para este tipo de pantalla de consulta se pueden realizar filtros por nombre, tipo, y propiedad. Como resultado se obtienen los elementos de análisis que cumplan los filtros. Además, se tiene la opción de poder obtener la información de las propiedades y el texto original de los elementos de análisis seleccionados como resultado de la consulta. Esta consulta se implementó pensando en que hay cierta información que se desea buscar de los elementos individuales.

El segundo tipo de pantalla de consulta únicamente se realiza sobre las relaciones de los elementos de análisis. Para este tipo de consulta se pueden realizar filtros por nombre del padre, tipo del padre, nombre del hijo, tipo del hijo, y tipo de relaciones. Como resultado se obtienen las ramas de relaciones con sus elementos padre e hijo. Además, se tiene la opción de poder obtener información del texto original y gráfica. Esta consulta se implementó pensando en encontrar todas las ramas que cumplan los filtros. En

la opción grafica de esta consulta si hay demasiados objetos es necesario personalizarla con la herramienta de modelación Visio para obtener una mejor visualización.

El cliente de análisis representa un poderoso componente por la forma y flexibilidad de sus dos tipos de pantallas de consulta parametrizables para la identificación de elementos de análisis y sus dependencias. Además, puede ser utilizado para cualquier otro tipo de información donde se necesite identificar dependencias. Lo único que hay que hacer es cargar la información en el repositorio de consulta, respetando el significado de cada elemento que conforma este repositorio. Estos resultados también se ratifican después de haber aplicado la herramienta desarrollada al caso real.

D. Aplicación de la herramienta con base en los requerimientos

Al aplicar la herramienta a un caso real, se obtuvieron los resultados que se expondrán a continuación. En este rubro se detallan las construcciones de las consultas utilizadas y el objetivo de encontrar ciertos elementos de análisis.

Por asuntos de confidencialidad no se especificarán los nombres de los elementos de análisis ni se colocarán los tipos de consulta de manera visual para evitar que no se den a conocer los procesos de la forma de trabajar de la empresa .

Para facilitar la comprensión de los resultados, estos serán expuestos en cuadros y cada cuadro está compuesta de dos columnas: actividad y resultado. En la mayoría de los cuadros en esta sección se numeran las actividades en orden ascendente de arriba hacia abajo para hacer referencia a las actividades.

De la instalación de la herramienta se obtuvieron los siguientes resultados que se detallan en el próximo cuadro .

Actividad	Resultado
Elaboración del repositorio de carga	Tardó aproximadamente 2 minutos
Elaboración del repositorio de consulta	Demoró aproximadamente 5 minutos
Disco virtual	Se utilizo el mismo disco C de la máquina
Instalación del cliente de carga	Tardó aproximadamente 7 minutos
Instalación del cliente de análisis	Demoró aproximadamente 12 minutos
Herramienta externa de <i>scripts</i> de base de datos	Versión gratis de TOAD de 30 días
Generación de <i>scripts</i> de la aplicación real	Tardó aproximadamente 13 minutos
Numero de máquinas de clientes de carga	Una máquina
Numero de máquinas de clientes de análisis	La misma máquina de carga

Cuadro 12. Resultados de la instalación de la herramienta desarrollada

Los resultados de la carga de la aplicación del caso real se detalla en el siguiente cuadro. El cliente de carga es el encargado de realizar estas actividades.

Actividad	Resultado
Número de archivos que conforman el repositorio de carga	Un archivo de texto de 2.5 MB de tamaño
Identificación de los elementos de análisis por parte del cliente de carga	Tardó 14 minutos con 17 segundos
Identificación de las relaciones por parte del cliente de carga	Tardó 10 minutos con 54 segundos
Ambigüedades resueltas por parte del usuario utilizando el cliente de carga	32 ambigüedades
Conteo de los elementos de análisis registrados en el repositorio de consulta	9057 columnas, 854 tablas, 104 vistas, 132 <i>triggers</i> , 6 paquetes, 321 procedimientos, y 81 funciones.

Cuadro 13. Resultados de la carga de la aplicación en la herramienta desarrollada

El cliente de análisis es el encargado de elaborar las consultas y obtener información para resolver los requerimientos o apoyar con nueva información para la resolución de los requerimientos.

1. Requerimiento uno. En el caso de que la unidad de trabajo haya mandado más de una vez la misma información, ésta debe ser procesada una vez.

Para resolver este requerimiento, se pretenden determinar los elementos de análisis tipo tabla en donde se registra la información de manera no temporal y verificar si tienen la propiedad de llave primaria o chequeos de unicidad. Estos elementos de análisis objetivo se llamarán tablas históricas porque almacenan permanentemente la información.

Los resultados de este requerimientos serán expuestos en la siguiente tabla.

Actividad	Resultado
1. Identificación de elementos de análisis en el código fuente de la interfase	Se encontraron un elemento de análisis de tipo procedimiento y cinco elementos de análisis de tipo tabla
2. Elaboración de las consultas para identificar posibles tablas candidatas a ser históricas. Se espera que la información fluya de una relación de selección a otra relación de inserción, el elemento de análisis de tipo tabla que tenga la última inserción conformará la lista de tablas históricas	Las consultas relacional y grafica de referencia cruzada donde los elementos padres están conformados por los elementos encontrados en la actividad uno de este cuadro, y los elementos hijos están con formados por elementos de análisis de tipo tabla. Incluir en las consultas la opción de ultimo hijo y agregar las relaciones de selección y e inserción en la opción de existencia relación.

Actividad	Resultado
3. Ejecución de las consultas elaboradas en la actividad dos de este cuadro para determinar una lista de elementos de análisis candidatos a ser tablas históricas	Las consultas demoraron tres minutos aproximadamente en conjunto y se encontraron nueve elemento de análisis de tipo función, 33 elementos de análisis de tipo tabla, 11 elementos de análisis de procedimiento y tres elementos de análisis de tipo <i>trigger</i>
4. Elaboración de una lista de elementos de análisis del resultado de las consultas de la actividad tres de este cuadro	Se abrió el archivo "consulta.txt" con Excel y con base a macros se elaboró la lista. Este archivo se generó al hacer consulta gráfica de la actividad tres
5. Diseño de la consulta para revisión de la lista de elementos elaborada en la actividad cuatro de este cuadro	Se diseñó una consulta textual de definición donde los parámetros de tipo y nombre son llenados con base en la lista elaborada en la actividad cuatro de este cuadro
6. Ejecución de la consulta diseñada en la actividad cinco de este cuadro	La consulta tardó un minuto aproximadamente
7. Revisión del resultado de la consulta ejecutada en la actividad seis de este cuadro	Se hallaron cinco elementos de análisis tipo tabla que conforman las tablas históricas
8. Diseño de la consulta para determinar la existencia de llaves primarias en las tablas históricas	Se diseñó una consulta textual de definición donde los parámetros de tipo y nombre son llenados con base en las tablas históricas, encontradas en la actividad siete de este cuadro
9. Ejecución de la consulta diseñada en la actividad ocho de este cuadro	La consulta tardó un minuto aproximadamente y generó el archivo "respuesta.txt"
10. Revisión del resultado de la consulta obtenido en la actividad nueve de este cuadro	Las tablas históricas no tienen llaves primarias
11. Investigar cuales son las llaves primarias que se adecuen correctamente a los requerimientos de información de estas tablas. Se recomienda revisar los datos de las tablas históricas e ir con los usuarios que utilizan esta información para determinar cuales deben ser los campos para conformar las llaves primarias en las tablas históricas	Se procedió a colocar llaves primarias que mejor se adecuan a las tablas históricas con base en los datos ya almacenados y la información que utiliza el usuario
12. Diseño de las consultas para determinar los elementos de análisis en donde hay que colocar los chequeos y analizar el impacto de las llaves primaria	Se diseñaron las consultas relacional y gráfica de referencias simple donde los elementos hijos sean las tablas históricas.
13. Ejecución de las consultas diseñadas en la actividad doce de este cuadro	Las consultas demoraron trece minutos aproximadamente en forma conjunta y se encontraron cuatro elementos de análisis de tipo función, 18 elementos de análisis de tipo procedimiento, y cinco elementos de análisis de tipo tabla
14. Elaboración de una lista de los elementos de análisis que forman el resultado obtenido en la actividad trece de este cuadro	Se abrió el archivo "consulta.txt" con Excel y con base a macros se elaboró la lista. Este archivo se generó al hacer la consulta gráfica de la actividad trece de este cuadro
15. Diseño de la consulta para hacer la revisión de la lista elaborada en la actividad catorce de este cuadro	Se diseño una consulta textual de definición donde los parámetros de tipo y nombre son conformados por la lista elaborada en la actividad catorce de este cuadro

Actividad	Resultado
16. Ejecución y revisión de la consulta de los elementos encontrados en las consultas de la actividad quince de este cuadro	La consulta se tardó aproximadamente un minuto y generó el archivo "respuesta.txt". De la revisión se identificó poner los chequeos en cinco elementos de análisis de tipo procedimiento y cuatro elementos de análisis de tipo función

Cuadro 14. Actividades desarrolladas del requerimiento uno

2. Requerimiento dos. Si la información que mandan las unidades de trabajo no puede generar partidas contables cuadradas, éstas no deben reflejarse en el sistema informático de la administración.

Para llevar a cabo este requerimiento, se pretende determinar en dónde hay que colocar los chequeos de partidas descuadradas para evitar su ingreso en el sistema informático de la administración.

Los resultados de este requerimientos se muestran en el próximo cuadro.

Actividad	Resultado
1. Identificación de elementos de análisis en los fuentes de los reportes donde se muestra la información de las partidas contables	Se hallaron dos elementos de análisis de tipo tabla
2. Búsqueda de elementos de análisis en el código fuente de la interfase	Se encontró un elemento de análisis de tipo procedimiento
3. Elaboración de las consultas para determinar los elementos de análisis donde hay que colocar los chequeos de cuadro de las partidas contables	Las consultas relacional y gráfica de referencia cruzada donde el elemento padre es el elemento encontrado en la actividad dos de este cuadro, y los elementos hijos son conformados por los elementos encontrados en la actividad uno de este cuadro. Incluir en las consultas la opción de hijo último
4. Ejecución de las consultas de la actividad tres de este cuadro	La consulta se tardó un minuto aproximadamente y se hallaron seis elementos de análisis de tipo procedimiento, un elemento de análisis tipo de función y dos elementos de análisis de tipo tabla
5. Diseño de la consulta para revisión de los elementos análisis encontrados en la actividad cuatro de este cuadro	La consulta textual de definición donde los parámetros del tipo y nombre están conformados por los elementos de análisis de tipo procedimiento y función, encontrados en la actividad cuatro de este cuadro
6. Ejecución y revisión de la consulta de la actividad cinco de este cuadro	La consulta se tardó un minuto aproximadamente y generó el archivo "respuesta.txt". De la revisión se determinó poner el chequeo de cuadro en un elemento de análisis de tipo procedimiento

Cuadro 15. Actividades desarrolladas del requerimiento dos

3. Requerimiento tres. Las personas encargadas de verificar o analizar la información necesitan que se les notifique el estatus del procesamiento de la información que viene de la unidad de trabajo.

Para desarrollar este requerimiento, se necesita determinar los lugares en donde se debe registrar una bitácora de mensajes. Esta bitácora de mensajes tendrá la siguiente información: el mensaje mismo, fecha, unidad de trabajo, tipo de mensaje e identificador del proceso.

Entre los tipos de mensajes están: mensajes de notificación para el usuario, mensajes de error, mensajes de rastreo, etcétera. El identificador del proceso servirá para agrupar los mensajes que pertenecen a una misma transacción en el procesamiento de la información. Los mensajes de rastreo deben ser parametrizables para ser activados y los mensajes de error deben colocarse en las excepciones de los bloques de PL/SQL.

Con el registro de estos mensajes se notificará al usuario del estatus del procesamiento de la información y servirán para detectar inconvenientes.

Actividad	Resultado
1. Implementación de los elementos que conforman la bitácora de mensajes	Se crearon un elemento de análisis de tipo paquete que contiene tres elementos de análisis de tipo procedimiento, y tres elementos de análisis de tipo tabla. Además, los procedimientos creados deben de tener el <i>pragma</i> de transacción autónoma
2. Búsqueda de elementos de análisis en el código fuente de la interfase	Se hallaron un elemento de análisis de tipo procedimiento y cinco elementos de análisis de tipo tabla
3. Diseño de las consultas para determinar los elementos de análisis donde hay que colocar el registro de mensajes	Se diseñaron consultas relacional y grafica de referencia simple donde los elementos padres son los elementos de análisis encontrados en la actividad dos de este cuadro
4. Ejecución de las consultas diseñadas en la actividad tres de este cuadro	La consulta se tardó aproximadamente 5 minutos y se encontraron 16 elementos de análisis de tipo función, trece elementos de análisis de tipo procedimiento, quince elementos de análisis de tipo tabla y 24 elementos de análisis de tipo <i>trigger</i>
5. Elaboración de la lista de elementos de análisis que se encontraron en la actividad cuatro de este cuadro	Se abrió el archivo "consulta.txt" con Excel y con base a macros se elaboró la lista. Este archivo se generó al hacer la consulta gráfica de la actividad cuatro de este cuadro
6. Elaboración de una consulta para revisión de la lista de elementos elaborada en la actividad cinco de este cuadro	La consulta textual de definición donde los parámetros tipo y nombre son llenados con base a la lista elaborada en la actividad cinco de este cuadro
7. Ejecución de la consulta elaborada en la actividad seis de este cuadro	La consulta demoró aproximadamente un minuto y generó el archivo "respuesta.txt"
8. Revisión de la consulta efectuada en la actividad siete de este cuadro	Se identificó poner el registro de mensajes a cinco elementos de análisis de tipo función y once elementos de análisis de tipo procedimiento

Actividad	Resultado
9. Notificación al encargado que le da mantenimiento a los fuentes de código de la interfase para incorporación de la bitácora de mensajes	El usuario sea notificado del estatus del procesamiento de la información

Cuadro 16. Actividades desarrolladas del requerimiento tres

4. Requerimiento cuatro. Se determinó que ciertos reportes de análisis de información no se han podido generar a tiempo porque la información no se encuentra disponible, ésta se encuentra en la etapa de procesamiento, y la unidad de trabajo había mandado la información en los plazos estipulados para poder generar los reportes a tiempo.

Para la contribución en el avance de este requerimiento, se desea incorporar la bitácora de mensajes, expuesta en el Requerimiento tres, en los elementos de análisis que intervienen en el flujo de la información que proviene de los elementos de análisis que conforman los reportes.

Hay que definir un identificador de proceso para estas transacciones de procesamiento de información y con base en los registros de estos mensajes se va poder determinar cuáles son los elementos de análisis que están demorando la transacción o verificar la existencia de algún inconveniente que impida el registro de la información en los elementos de análisis.

Actividad	Resultado
1. Identificación de elementos de análisis en los fuentes de los reportes de donde se quiere obtener la información	Se hallaron cuatro elementos de análisis de tipo tabla
2. Búsqueda de elementos de análisis en el código fuente de la interfase	Se encontró un elemento de análisis de tipo procedimiento
3. Elaboración de las consultas para determinar los elementos de análisis que intervienen en el flujo de la información	Las consultas relacional y gráfica de referencia cruzada donde el elemento padre es el elemento encontrado en la actividad dos de este cuadro, y los elementos hijos son conformados por los elementos encontrados en la actividad uno de este cuadro. Incluir en las consultas la opción de hijo último
4. Ejecución de las consultas de la actividad tres de este cuadro	La consulta se tardó aproximadamente 5 minutos y se encontraron cuatro elementos de análisis de tipo procedimiento y cuatro elementos de análisis de tipo tabla
5. Diseño de la consulta para revisión de los elementos análisis encontrados en la actividad cuatro de este cuadro	La consulta textual de definición donde los parámetros del tipo y nombre están conformados por los elementos de análisis de tipo procedimiento, encontrados en la actividad cuatro de este cuadro
6. Ejecución de la consulta diseñada de la actividad cinco de este cuadro	La consulta se tardó un minuto aproximadamente y generó el archivo "respuesta.txt".

Actividad	Resultado
7. Revisión de los resultados obtenidos de la consulta de la actividad seis de este cuadro	De la revisión se determinó colocar los registros de mensajes a los cuatro elementos de análisis de tipo procedimiento

Cuadro 17. Actividades desarrolladas del requerimiento cuatro

5. Requerimiento cinco. Hay información que viene de la unidad de trabajo que no se está utilizando en el procesamiento de la información para el sistema informático de la administración, el uso de esta información contribuirá en la automatización de ciertos procesos.

Para llevar a cabo este requerimiento, hay que determinar la información que proviene de la unidad de trabajo y de esta información cual es procesada en el sistema informático de la administración. Además, se busca un patrón del procesamiento de la información que hay que tomar en cuenta en el momento de realizar los cambios. En caso de no existir este patrón habría que crear uno para ordenar el flujo de información y para facilitar el mantenimiento de los cambios.

Actividad	Resultado
1. Identificación de la información que es capturada por la interfase	Lista de campos
2. Búsqueda de elementos de análisis en el código fuente de la interfase, esta es la información procesada por la interfase	Se encontraron un elemento de análisis de tipo procedimiento y cinco elementos de análisis de tipo tabla
3. Investigación de la forma en que se tiene que procesar la información encontrada en la actividad uno de este cuadro. Además investigar los controles y reportes necesarios que hay que tomar en cuenta	Se determinó que la información que no está siendo procesada no complementa a la información que si se procesa
4. Diseño de consultas para determinar los elementos de análisis que conforman el patrón de procesamiento de información	Se diseñó las consultas gráfica y relacional de referencia simple donde los elementos padres son los elementos de análisis encontrados en la actividad dos de este cuadro
5. Ejecución de las consultas diseñadas en la actividad cuatro de este cuadro	La consulta se tardó aproximadamente cuatro minutos y se hallaron 16 elementos de análisis de tipo función, trece elementos de análisis de tipo procedimiento, quince elementos de análisis de tipo tabla y 24 elementos de análisis de tipo <i>trigger</i>
6. Revisión del resultado obtenido de las consultas de la actividad cinco de este cuadro	Se determinó de la consulta relacional que la rama más grande tiene seis niveles
7. Diseño de consultas para determinar los elementos de análisis que conforman el patrón de procesamiento de información de forma gradual por niveles	Se diseñó las consultas gráfica y relacional de referencia simple donde los elementos padres son los elementos de análisis encontrados en la actividad dos de este cuadro. A estas consultas hay que llenar la opción por nivel

Actividad	Resultado
8. Ejecución de las consultas diseñada en la actividad siete de este cuadro utilizando el nivel uno	Las consultas duraron un minuto y se encontraron siete elementos de análisis de tipo función, tres elementos de análisis de tipo procedimiento, cuatro elementos de análisis de tipo tabla y doce elementos de análisis de tipo <i>trigger</i>
9. Ejecución de las consultas diseñada en la actividad siete de este cuadro utilizando el nivel dos	Las consultas tardaron dos minutos y se encontraron diez elementos de análisis de tipo función, once elementos de análisis de tipo procedimiento, seis elementos de análisis de tipo tabla y catorce elementos de análisis de tipo <i>trigger</i>
10. Ejecución de las consultas diseñada en la actividad siete de este cuadro utilizando el nivel tres	Las consultas duraron tres minutos y se encontraron 16 elementos de análisis de tipo función, trece elementos de análisis de tipo procedimiento, once elementos de análisis de tipo tabla y 19 elementos de análisis de tipo <i>trigger</i>
11. Ejecución de las consultas diseñada en la actividad siete de este cuadro utilizando el nivel cuatro	Las consultas tardaron cuatro minutos y se encontraron 16 elementos de análisis de tipo función, trece elementos de análisis de tipo procedimiento, trece elementos de análisis de tipo tabla y 22 elementos de análisis de tipo <i>trigger</i>
12. Ejecución de las consultas diseñada en la actividad siete de este cuadro utilizando el nivel cinco	Las consultas demoraron seis minutos y se encontraron 16 elementos de análisis de tipo función, trece elementos de análisis de tipo procedimiento, quince elementos de análisis de tipo tabla y 24 elementos de análisis de tipo <i>trigger</i>
13. Ejecución de las consultas diseñada en la actividad siete de este cuadro utilizando el nivel seis	Las consultas tardaron 4 minutos y se encontraron 16 elementos de análisis de tipo función, trece elementos de análisis de tipo procedimiento, quince elementos de análisis de tipo tabla y 24 elementos de análisis de tipo <i>trigger</i>
14. Elaboración de una lista de los elementos de análisis indicando el número de nivel donde apareció por primera vez el elemento de análisis	La lista se elaboró utilizando macros en Excel, consolidando la información de los archivos "consulta.txt" de las actividades ocho, nueve, diez, once, doce y trece de esta tabla. El nivel uno tiene 26 elementos de análisis, el nivel dos tiene quince elementos de análisis, el nivel tres tiene 18 elementos de análisis, nivel cuatro tiene cinco elementos de análisis, nivel cinco tiene cuatro elementos de análisis y el nivel 6 no tiene elementos de análisis.
15. Diseño de las consultas para revisión por niveles de los elementos análisis encontrados en la lista elaborada en la actividad catorce de este cuadro	La consulta textual de definición se hace por nivel donde los parámetros del tipo y nombre están conformados por los elementos de análisis que cumplen el nivel indicado en lista elaborada en la actividad catorce de este cuadro
16. Ejecución de las consultas diseñada de la actividad quince de este cuadro	Las consultas tardaron cada una aproximadamente un minuto y generaron cada una el archivo "respuesta.txt"

Actividad	Resultado
17. Revisión de los resultados obtenidos de la consulta de la actividad 16 de este cuadro. El orden de la revisión lo indica el nivel y es ascendente.	El patrón de procesamiento de la información que se detectó fue: validación de datos temporales (conformado por un elemento de análisis de tipo procedimiento y cuatro elementos de análisis de tipo función) , traslado de datos históricos (constituido por seis elementos de análisis de tipo procedimiento), cuadro de la transacción (compuesto por un elemento de análisis de tipo función) , contabilización de la transacción (conformado por tres elementos de análisis de tipo procedimiento) y registro de documentos que soportan la contabilización (integrado por dos elementos de análisis de tipo procedimiento)
18. Creación de elementos de análisis para la incorporación de la información nueva en el sistema informático de la administración , tomando en cuenta los resultados de las actividades 17 y tres de este cuadro	Se crearon cuatro elementos de análisis de tipo tabla, dos elementos de análisis de tipo función y dos elementos de análisis de tipo procedimiento
19. Elaboración de la consulta de los elementos de análisis obtenidos en la actividad 17 de este cuadro	La consulta textual de la definición donde los parámetros del tipo y nombre están formados por los elementos de análisis que se encuentran en el patrón de procesamiento de información expuesto en la actividad 17 de este cuadro
20. Ejecución de la consulta diseñada de la actividad 19 de este cuadro	La consulta se tardó un minuto aproximadamente y generó el archivo "respuesta.txt"
21. Revisión de los elementos de análisis obtenidos en la actividad veinte de este cuadro para determinar el impacto de los cambios que hay que realizar en la incorporación de los elementos de análisis creados en la actividad 18 de este cuadro, tomando en cuenta la bitácora de mensajes desarrollada en el Requerimiento tres y el patrón de procesamiento de información encontrado en el Requerimiento cinco	Se concluyó que hay que hacer modificaciones a un elemento de análisis de tipo procedimiento que integra el traslado de datos históricos y un elemento de análisis de tipo procedimiento que conforma la validación de temporales

Cuadro 18. Actividades desarrolladas del requerimiento cinco

6. Requerimiento seis. Por cuestiones de actualizaciones de la ley de impuestos, se necesita que la unidad de trabajo envíe cierto detalle de información para poderse alimentar correctamente los libros del IVA.

Para llevar a cabo este requerimiento, hay que identificar que elementos de análisis deben ser modificados para que los libros de IVA queden con el detalle de información que se necesita.

Actividad	Resultado
1. Identificación de elementos de análisis en los fuentes de los reportes de donde se muestra la información de los libros del IVA	Se hallaron dos elementos de análisis de tipo tabla
2. Búsqueda de elementos de análisis en el código fuente de la interfase	Se encontró un elemento de análisis de tipo procedimiento

Actividad	Resultado
3. Modificaciones de los elementos de análisis encontrados en la actividad uno de este cuadro , con base en el detalle de la información que se necesita registrar en los libros del IVA	Hay que incorporar un nuevo elemento de análisis de tipo columna para almacenar la información que hace falta en un elemento de análisis de tipo tabla encontrado en la actividad uno de este cuadro
4. Diseño de consultas para determinar los elementos de análisis donde hay que realizar las modificaciones	Se diseñó las consultas gráficas y relacional de referencia cruzada donde el elemento padre es el elemento de análisis encontrado en la actividad dos de este cuadro y los elementos hijos son conformados por los elementos de análisis encontrados en la actividad uno de este cuadro. En las consultas utilizar la opción de hijo último.
5. Ejecución de las consultas de la actividad cuatro de este cuadro	La consulta se tardó aproximadamente cuatro minutos y se hallaron tres elementos de análisis de tipo procedimiento y un elemento de análisis de tipo tabla
6. Diseño de la consulta para revisión de los elementos análisis encontrados en la actividad cinco de este cuadro	La consulta textual de definición donde los parámetros del tipo y nombre están conformados por los elementos de análisis de tipo procedimiento, encontrados en la actividad cinco de este cuadro
7. Ejecución de la consulta diseñada de la actividad seis de este cuadro	La consulta se tardó un minuto aproximadamente y generó el archivo "respuesta.txt"
8. Revisión de los resultados obtenidos de la consulta de la actividad siete de este cuadro	Se identificó que no hay que realizar ningún cambio
9. Elaboración de consultas para obtener los elementos de análisis que intervienen en el flujo de información de los elementos de análisis que conforman los libros del IVA, estos elementos serán los candidatos para hacer las modificaciones	Las consultas relacional y grafica de referencia simple donde los elementos hijos son los elementos de análisis encontrados en la actividad uno de este cuadro
10. Ejecución de las consultas elaborada en la actividad nueve de este cuadro	La consulta tardó aproximadamente trece minutos y se encontraron un elemento de análisis de tipo paquete, quince elementos de análisis de tipo procedimiento y dos elementos de análisis de tipo tabla
11. Hacer una lista de los elementos de análisis encontrados en las consultas ejecutadas en la actividad diez de este cuadro	Se abrió el archivo "consulta.txt" con Excel y con base a macros se elaboró la lista. Este archivo se generó al hacer la consulta gráfica de la actividad diez de este cuadro
12. Diseño de la consulta para revisión de los elementos de análisis seleccionados de la lista elaborada en la actividad once de este cuadro	La consulta textual de definición donde los parámetros del tipo y nombre están formados por los elementos de análisis que se encuentran en la lista elaborada en la actividad once de este cuadro
13. Ejecución de la consulta diseñada de la actividad doce de este cuadro	La consulta se tardó un minuto aproximadamente y generó el archivo "respuesta.txt"
14. Revisión de los resultados obtenidos de la consulta de la actividad trece de este cuadro, tomando en cuenta la lista de tablas históricas encontradas en el Requerimiento uno, el patrón de procesamiento de información encontrado en el Requerimiento cinco, y la bitácora de mensaje desarrollada en el Requerimiento tres	Se concluyó que hay que hacer modificaciones a un elemento de análisis de tipo procedimiento y un elemento de análisis de tipo tabla

Cuadro 19. Actividades desarrolladas del requerimiento seis

E. Reducción de tiempos por el uso de la herramienta

A simple vista parece que el uso de la herramienta no contribuye en la reducción de tiempos a la hora de implementar los cambios, por el hecho de que hay que realizar una carga inicial y hay que invertir tiempo para hacer las consultas.

Para la elaboración de los requerimientos: Requerimiento uno, Requerimiento dos, Requerimiento tres y Requerimiento cuatro, se pueden realizar búsquedas manuales para encontrar la lista de elementos que habría que revisar posteriormente para realizar los cambios. La realización de este proceso de búsqueda manual puede llevar a elementos olvidados, visualizaciones incorrectas, y elementos no visitados, por lo que los resultados encontrados pueden no ser confiables. En el peor de los casos, el proceso de búsqueda se puede ir realizando a ciegas visitando elementos que no valdría la pena revisar. Por lo tanto, hay que invertir tiempo en realizar este proceso de búsqueda manual. En cambio utilizando la herramienta se puede dejar que ella procese la consulta y paralelamente realizar otra actividad. Además de tener la seguridad de obtener datos confiables.

Si se realizarán modificaciones con el proceso de búsqueda manual, se podría correr el riesgo que en la etapa de realizar las pruebas haya que regresar a realizar cambios, en el peor de los casos se daría en la fase de producción, ya que no se sabe si realmente las modificaciones tienen impactos en otros lugares de la aplicación que no se encontraron en la búsqueda manual. Este riesgo debe disminuirse considerablemente con el uso de la herramienta desarrollada.

En la elaboración de los requerimientos: Requerimiento cinco y Requerimiento seis, habría que hacer una búsqueda exhaustiva para encontrar la lista de elementos donde hay que realizar los cambios. Por lo cual, se invertiría mucho tiempo y esfuerzo a la hora de hacer la búsqueda.

El hecho de utilizar el análisis de dependencias para determinar el impacto que tiene un cambio y tener diferentes tipos de consultas para visualizar estas dependencias, contribuye a poder determinar los recursos y tiempos de una manera más adecuada para obtener resultados más confiables en tiempos más razonables, con lo cual la reducción de tiempo se obtendrá en las etapas siguientes.

F. Productividad en el uso de la herramienta

En esta sección se detallan consejos y recomendaciones para incrementar la productividad en el uso de la herramienta desarrollada

1. Procedimiento propuesto. De los resultados obtenidos del rubro Aplicación de la herramienta con base en los requerimientos, se puede visualizar una serie de pasos genéricos que se aplicaron para cada requerimiento. Estos pasos en conjunto forman el procedimiento propuesto.

Las etapas o pasos que conforman este procedimiento son:

- Carga inicial del sistema informático.
- Identificación y documentación de los elementos de análisis del enunciado del requerimiento.
- Identificación de elementos de análisis objetivo por medio de consultas para cada requerimiento.
- Revisión de consultas textuales para determinar el impacto de los cambios de cada requerimiento.
- Obtención de la información que se necesita para hacer los cambios de los requerimientos en forma individual.
- Revisión que toma en cuenta los requerimientos en conjunto.
- Implementación de los cambios.

A continuación se explica cada una de las etapas del procedimiento propuesto.

a. Carga inicial del sistema informático. Siempre que se desee usar la herramienta desarrollada hay que realizar una carga inicial de la información. Esto se debe porque los sistemas informáticos son un reflejo de la forma de trabajar de la empresa, la cual es muy cambiante. Para esta actividad se recomienda utilizar una máquina para realizar la carga de las aplicaciones y todas las demás máquinas colocarlas como clientes de análisis.

b. Identificación y documentación de los elementos de análisis del enunciado del requerimiento. Cada enunciado de un requerimiento nace por algún inconveniente o mejora del sistema informático. La forma de expresión o escritura de un requerimiento

conlleve elementos que tiene un significado dentro del sistema informático. Estos elementos deben tener una contraparte en los elementos de análisis seleccionados porque cada elemento que se encuentra en el sistema informático tiene una razón de ser.

Por ejemplo, la contraparte de los libros de IVA son los elementos de análisis de tipo tabla donde se almacena esta información y la contraparte de un proceso de cuadro de una transacción es el elemento de análisis de tipo función que contiene la regla del negocio para poder realizar el cuadro de la transacción.

Esta etapa de identificación de los elementos de análisis que deben obtenerse del significado de los elementos expresados por el enunciado del requerimiento es importante, ya que estos elementos identificados servirán de filtro para realizar consultas que proveerán nueva información para resolver el requerimiento o complementar el mismo.

En esta etapa se necesita experiencia en el sistema informático, la cual se puede lograr poco a poco por parte del usuario conforme utiliza la herramienta desarrollada. Se recomienda elaborar un catálogo de elementos de análisis y sus relaciones donde se registre el significado del elemento en el sistema y que contenga dos índices. El primer índice debe ser sobre los elementos de análisis y el segundo índice sobre el significado del elemento de análisis en el sistema informático. El catálogo anterior servirá como documentación y facilitará el proceso de identificación de los elementos de análisis.

Se recomienda hacer revisiones de código fuente de reportes, formas, interfases y conocer el proceso que implementan para la identificación de estos elementos de análisis.

c. Identificación de elementos de análisis objetivo por medio de consultas para cada requerimiento. Cada requerimiento se desea resolver o apoyar con nueva información. Para lograr lo anterior se necesitan hacer cambios o modificaciones en el sistema informático. Antes de realizar los cambios es conveniente determinar el impacto de los mismos sobre un conjunto de elementos de análisis. Estos elementos de análisis se llamarán elementos de análisis objetivo.

Al principio se tiene un conjunto de elementos de análisis objetivos candidatos que se obtienen de la elaboración de consultas donde intervienen los elementos de análisis identificados en el paso "b". De los resultados de las consultas se puede ir

depurando haciendo otras consultas con otros filtros para encontrar los elementos de análisis objetivo.

Los tipos de consultas que se elaboran son relacionales y gráficas con cualquier tipo de referencia, donde se pueden hacer combinaciones de las opciones parametrizables de la consultas, según sea el caso. Por ejemplo estas opciones son: escribir listas de elementos de análisis para conformar los elementos padres y elementos hijos, opción por niveles, opción de hijo último, opción de lista de existencia de relaciones, y opción de lista de relaciones.

De los archivos de respuesta que generan las consultas se pueden elaborar las lista de estos elementos de análisis objetivo apoyándose en otras herramientas. Por ejemplo, Excel.

d. Revisión de consultas textuales para determinar el impacto de los cambios de cada requerimiento. Una vez elaborada la lista de los elementos de análisis que hay que revisar para hacer los cambios. Se procede hacer una consulta textual de estos elementos.

De la revisión de la consulta textual se deben obtener conclusiones para determinar como afecta o cual será su impacto a la hora de realizar el cambio

Además se debe obtener una lista de los elementos de análisis que hay que modificar y la modificación respectiva.

En caso de identificar que no hay que realizar ningún cambio entre la lista de elementos de análisis objetivo, habría que determinar una nueva lista de elementos de análisis objetivos y regresar al paso “c”.

e. Obtención de la información que se necesita para hacer los cambios de los requerimientos en forma individual. Una vez determinado cuáles son los elementos de análisis que hay que crear o alterar, y cuales son las modificaciones o cambios que hay que realizar.

Se encuentra en una etapa donde se pueden estimar los tiempos y recursos que se necesitan para la elaboración de los requerimientos de forma individual

f. Revisión que toma en cuenta los requerimientos en conjunto. Este proceso de

revisión debe tomar la información obtenida por cada requerimiento y consolidarla de tal manera que a la hora de hacer la implementación de los requerimientos no halla conflictos entre los mismos.

También, ayudará para hacer una estimación más adecuada de los tiempos y recursos que se necesitan para su implementación de los requerimientos en forma conjunta.

Se recomienda agrupar los requerimientos para repartirlos por afinidad para disminuir el tiempo de revisión en esta etapa.

g. Implementación de los cambios. Esta etapa es en sí la realización de los cambios tomando en cuenta los recursos definidos por la etapa anterior.

2. Consejos para la realización de consultas. Lista de consejos para realizar consultas que fueron aprendidos después del uso de la herramienta desarrollada a un caso real:

- En una consulta relacional donde se tenga que utilizar la opción de búsqueda del hijo último tratar de filtrar lo más posible en las opciones de búsqueda de tipo y nombre del elemento padre.
- Para realizar los procesos de revisión hay que elaborar consultas textuales con base en una lista en la primera pantalla de consulta del cliente de análisis. Los valores de esta lista pueden ser escritos contiguamente separados por coma en los parámetros de tipo y nombre del elemento.
- Tratar en lo posible de no hacer consultas relacionales o gráficas de referencia simple donde únicamente se llenen las opciones del elemento hijo, ya que estas consultas hacen un barrido sobre todos los elementos de análisis. Para este caso se recomienda llenar la casilla de tipo de elemento padre con el valor *DATABASE*.
- En las consultas relacionales de referencia cruzada tratar de escribir la lista de elementos separados por comas tanto en el elemento padre como en el elemento hijo cuando sea posible.
- Tratar de usar la opción de lista de existencia relación cuando sea necesario, ya que estos chequeos se realizan sobre toda la rama de búsqueda después de haber

encontrado la rama.

- Cuando se hace una consulta relacional o gráfica, de la cual se espera elaborar una lista muy extensa, que posteriormente será usada en un proceso de revisión, en esta consulta utilizar la opción por nivel. Con esta opción se pueden generar las consultas de manera gradual, reduciendo la cantidad de elementos que conforman la lista por cada nivel.
- Filtrar sobre un conjunto de tipo de relaciones al realizar una consulta cuando sea posible. Esto evitará que se visiten relaciones indeseables y podará el árbol de búsqueda.

3. Complementar con otras herramientas. Para la obtención de nueva información que ayude en la elaboración de un requerimiento, se pueden utilizar las salidas de la herramienta que servirán de entrada en otra herramienta para obtener nueva información.

Estas salidas de la herramienta son archivos generados en el momento de ejecutar las consultas. Al realizar la consulta gráfica se genera el archivo “consulta.txt” y al realiza la consulta relacional o textual se genere el archivo “respuesta.txt”. Los formatos de estos archivos fueron expuestos en el rubro Implementación del cliente de análisis perteneciente a la sección DESARROLLO DE LA PROPUESTA.

Por ejemplo, para la elaboración de listas y estadísticas se puede abrir el archivo “consulta.txt” con Excel y con base a macros obtener la información que se necesita.

4. Apoyo por parte de la administración. Se necesita el apoyo de la administración por parte del departamento de sistemas informáticos para que los usuarios utilicen la herramienta desarrollada antes de realizar los cambios de una aplicación. Siempre indicándoles que no es un reemplazo de la metodología que ya tienen sino como un complemento.

Una vez cargada la información en el repositorio de consulta, los usuarios de la herramienta desarrollada pueden utilizar el cliente de análisis, con base en los resultados obtenidos por la herramienta desarrollada se pueden justificar los cambios a realizar, y conocidos todos los elementos afectados, se puede estimar tiempos y recursos.

La mejor manera para ser productivo en el uso de la herramienta desarrollada es utilizando la herramienta misma cada vez que tenga sentido hacerlo. Con ello los usuarios que utilizan la herramienta van adquiriendo creatividad, disciplina, voluntad, y confianza en el uso continuo de la herramienta. Además de ir encontrando los beneficios de porque utilizar la herramienta.

VI. RECOMENDACIONES

- En la instalación del repositorio de consulta, una vez instalada la instancia se puede crear los usuarios y los objetos de dichos usuarios. Se recomienda crear los objetos de los usuarios en un *tablespace* aparte para uso exclusivo de la herramienta.
- Para la creación de más usuarios de consulta de la aplicación, además del usuario CONSULTA_ANALISIS, se pueden adaptar los *scripts* “usuario_consulta_analisis.sql” y “sinonimos_usuario_consulta.sql”. En el primer *script* se reemplaza el nombre del usuario CONSULTA_ANALISIS por el nuevo usuario y en el segundo *script* se corre conectándose con el nuevo usuario.
- Como la arquitectura es flexible en la incorporación de los elementos de análisis. Se pueden agregar tipos de objetos de Forms, ya que Forms puede generar reportes de documentación o utilizar las librerías Forms API para obtener la información. Entre los elementos de análisis que se pueden incorporar son: Bloques, items, triggers, procedimientos y funciones. Todos estos elementos son definidos en la forma. Se recomienda agregar un padre que se llame Repositorio, del cual todos estos elementos serían descendientes. Para incorporar estos elementos con los elementos de análisis seleccionados anteriormente.
- También se pueden agregar otros tipos de relaciones que pueden surgir de las instrucciones *GRANT* y *REVOKE* de SQL. Además, hay que agregar otros tipos de elementos de análisis como *ROLES* y *USERS*, ya sea que se desee incorporarlo en la misma aplicación o no. Con base a estas relaciones se puede tener vistas o consultas de la dependencia de información de privilegios.
- Una vez generada la vista gráfica en la herramienta de modelación Visio 2002, se puede utilizar la herramienta misma para realizar una personalización del diagrama a gusto del usuario, para ello usar la opción *Lay Out Shapes*.
- En el cliente de carga descrito de la herramienta desarrollada se le puede incorporar las instrucciones de sinónimos públicos y privados para resolver las referencias o llamadas. Con estos elementos también aumentaría la aplicabilidad de la herramienta a un tipo más amplio de aplicaciones.
- Para la implementación del cliente de análisis se puede utilizar la caracteriza de

relaciones recursivas de SQL3 cuando en los productos comerciales hallan madurado este tipo de característica para no tener que realizar búsquedas recursivas.

- Para la generación de los *scripts* se puede utilizar la Herramienta de Quest TOAD, en esta herramienta se le puede indicar de manera parametrizable que al final de la instrucción sea un símbolo de punto y coma. En el caso de utilizar la herramienta Navigator de Quest habría que colocar el punto y coma manualmente para quitar la diagonal (que en algunas ocasiones es delimitador de fin de instrucción).
- Utilizar los archivos de salida que genera la herramienta desarrollada para obtener más información al complementar esta herramienta con otras herramientas como Excel. Cuando se utilice una herramienta de edición colocar el tipo de letra *New Courier*.
- En la elaboración de un nuevo cliente de carga utilizar resolución de ambigüedades por parte del usuario, para eliminar su complejidad y asegurar que la información se encuentre correctamente.
- Antes de hacer una consulta es importante conocer el orden de las relaciones y revisar los consejos para realizar consultas, expuestos en la sección de RESULTADOS Y DISCUSIÓN.
- Utilizar los pasos del siguiente procedimiento para atacar o resolver los requerimiento: carga inicial del sistema informático, identificación y documentación de los elementos de análisis del enunciado del requerimiento, identificación de elementos de análisis objetivo por medio de consultas para cada requerimiento, revisión de consultas textuales para determinar el impacto de los cambios de cada requerimiento, obtención de la información que se necesita para hacer los cambios de los requerimientos en forma individual, revisión que toma en cuenta los requerimientos en conjunto, e implementación de los cambios.
- Para la elaboración de un nuevo cliente de carga se puede utilizar como repositorio de carga un conjunto de vistas que trae la base de datos de Oracle. Entre estas vistas están: DBA_OBJECTS (se describen todos los objetos de la base de datos), DBA_TABLES (se describen únicamente las tablas), DBA_VIEWS (se describen únicamente las vistas), DBA_TRIGGERS (se describen únicamente los triggers), DBA_TRIGGER_COLS (se describen las columnas que disparan el *trigger* al

momento de hacer una actualización), DBA_SOURCE (se describen los paquetes, procedimientos y funciones), DBA_DEPENDENCIES (se describen las relaciones entre los objetos de la base de datos), DBA_CONSTRAINTS (se describen las llaves primarias, las llaves foráneas, y se chequeos por columna), DBA_CONS_COLUMNS (se describe a detalle que campos conforman la llave primaria, y la llave foránea), DBA_USERS (se describen los usuarios), DBA_ROLES (se describen los roles), DBA_ROLE_PRIVS (se describe la relación de roles que tiene un usuario o rol), DBA_SYS_PRIVS (se describe la relación que privilegios del sistema tiene un rol o usuario), DBA_TAB_PRIVS (se describe la relación que privilegios sobre un objeto tiene un rol o usuario) y así sucesivamente hay una infinidad de vistas donde se puede obtener información.

- Elaboración de nuevas opciones de búsqueda en el cliente de análisis como listas de elementos de análisis que deben conformar una rama (la rama de búsqueda que se forme debe cumplir con la condición que todos sus elementos de análisis pertenezcan a la lista de elementos de análisis seleccionados), lista de existencia de elementos de análisis que deben aparecer en una rama (la rama de búsqueda que se forme debe cumplir con la condición que existan elementos de análisis pertenezcan a la lista de elementos de análisis seleccionados), desarrollo de subconsultas (poder hacer filtros sobre los resultados de las consultas desarrolladas anteriormente), operaciones entre consultas (poder juntar resultados de las consultas para conformar un resultado de la nueva consulta), o hacer negaciones de las opciones ya existentes.

VII. RECONOCIMIENTOS

Agradezco a todos aquellos que ayudaron en la preparación del documento y la herramienta. Especialmente a mis padres Rubén Mejía y Luz de Mejía, y amigos Arturo Rivera, Aníbal García y Juan Antonio Cabrera por la colaboración brindada, y a mi novia Ingrid Ramos por la comprensión y apoyo. También a Microsoft de Guatemala por proporcionarme la versión beta de .Net y por haber hecho de Visual Basic un lenguaje de programación totalmente orientado a objetos, y a la corporación Oracle por facilitarme la versión beta de la base de datos Oracle 8i, y por haber hecho un lenguaje de programación y una base de datos relacional tan maravillosos como lo son PL/SQL y Oracle 8i respectivamente.

VIII. REFERENCIAS BIBLIOGRÁFICAS

- Aho, Alfred V.; Sethi, Ravi y Ullman, Jeffrey D. 1990. *Compiladores Principios, técnicas y herramientas*. 1ª ed. Delaware, Addison Wesley. 902 págs.
- Awai, Michael; Bortniker, Matthew; Carnell, John; Dillon, Sean; Erwin, Drew; Goodman, Jaeda; Hólm, Bjarki; Horton, Ann; Hubeny, Frank; Kyte, Thomas; Mitchell II, Glen; Mukhar, Kevin; Nicol, Gary; O'Connor, Daniel; Ruth-Hammond, Guy y Zucca, Mario. 2000. *Oracle 8i Application Programming with Java, PL/SQL and XML*. Canadá, Wrox Press. 1275 págs.
- Bohner ,Shawn A. y Arnold, Robert S. *Software Change Impact Analysis*. IEEE Computer Society Press. 376 págs.
- Chen, Yih-Farm. *A CIA Tutorial*. IEEE Transactions on Software Engineering.
- Date, C. J. 1995. *An introduction to Database System*. 6ª ed. New York, Addison Wesley. 562 págs.
- Dorsey, Paul y Hudicka, Joseph R. 1999. *Oracle 8 Diseño de bases de datos con UML*. Edición Oracle Press. España, Osborne. 394 págs.
- Kendal y Kendal. 1991. *Análisis y diseño de sistemas*. 1ª ed. México, Prentice Hall. 881 págs.
- Laws Wenger, James. *Program Dependency Visualization*. Intelligent Visual Software Engineering. 4 págs. Obtenido en la Red Mundial el 30/10/2001: dirección <http://www.dcs.exeter.ac.uk/~jlwenger/research/TearOneHandin/node21.html>
- Oracle Corporation. 1999. *Oracle8i Concepts Release 2 (8.1.6)*. 902 págs.
- Oracle Corporation. 1999. *Oracle 8i SQL Reference Realese 2 (8.1.6)*. 986 págs.
- Oracle Corporation. 1999. *PL/SQL User's Guide and Reference Realese 2 (8.1.6)*. 626 págs.
- Pressman, Roger S. 1993. *Ingeniería del Software Un enfoque práctico*. 3ª ed. España, Mc Graw Hill. 824 págs.
- Ray, Mark. *Dependency Diagrams*. ACM Crossroads Student Magazine. 6 págs. Obtenido en la Red Mundial el 30/10/2001: dirección <http://www.acm.org/crossroads/xrds2-3/dependency.html>.
- Ullman, Jeffrey D. y Widom, Jennifer. 1997. *A first course in Database System*. 1ª ed. New Jersey, Prentice Hall. 470 págs.
- Visio Corporation. 1999. *Developing Visio Solutions Visio 2000*. Washington. United State of America. 562 págs.



IX. Apéndices

A. Instalación de la herramienta

Hay que determinar cuáles serán las computadoras en donde se instalará la herramienta desarrollada. Se recomienda escoger una computadora para el cliente de carga y varias computadoras para clientes de análisis. Además, de identificar las computadoras de desarrollo o mantenimiento de la herramienta.

Antes de instalar cualquier *software* se debe tener el licenciamiento correspondiente. En las computadoras seleccionadas se debe instalar el siguiente software para su preparación:

- Actualización de componentes de .Net, este *software* se encuentra en los CD de los productos de .Net y la tecnología de .Net únicamente se puede utilizar en las versiones de sistemas operativos Windows 98 o más recientes. También este *software* se pueden bajar del sitio de Microsoft (www.microsoft.com).
- *Service pack* de los sistemas operativos correspondientes, este *software* se pueden bajar del sitio de Microsoft (www.microsoft.com).
- Productos de Oracle Net 8 y OLEDB de Oracle, este software se encuentra en el CD de instalación de la base de datos.

Adjunto al documento se encuentra un CD con los fuentes de la herramienta desarrollada. Se debe utilizar por lo menos una computadora de desarrollo con las licencias correspondientes del producto Visual Basic .Net, para abrir los proyectos desarrollados en la herramienta Visual Basic y construir los archivos ejecutables, que posteriormente hay que copiar en las computadoras donde se usará la herramienta desarrollada.

En el CD hay una carpeta “herramienta” y dentro de esta carpeta se encuentran otras dos carpetas, las cuales son: “cargarepositorio” y “consultarepositorio”. La carpeta de “cargarepositorio” tiene los fuentes de código del componente del cliente de carga y la

carpeta de “consultarepositorio” tiene los fuentes de código del componente del cliente de análisis. Los proyectos son archivos con extensión “vbproj” y al construir los archivos ejecutables con base a estos proyectos, se crea una nueva carpeta “bin” en donde se encuentran los archivos ejecutables con extensión “exe” y librerías con extensión “dll”. Esta carpeta “bin” es la que hay que copiar en la máquina donde se desea ejecutar la herramienta desarrollada.

También hay una carpeta “aplicación” con todas las carpetas que se necesitan para construir el repositorio de carga. Cada vez que se haga una copia de la información del CD al disco de la computadora, no se le olvide quitarle a las carpetas o archivos el cheque de sólo lectura, y revisar los pasos que se encuentran en el rubro Repositorio de carga de la sección Implementación del cliente de análisis.

En el caso de que el repositorio de carga tenga que ser compartido por varias computadoras hay que seleccionar un disco virtual para asociar el directorio de la aplicación compartido del repositorio de carga en la computadora donde se quiere hacer la carga. El disco virtual seleccionado en el momento de hacer la carga tiene que ser el mismo que hay que asociar en las computadoras que serán utilizadas como clientes de análisis.

Los archivos para la construcción del repositorio de consulta se encuentran en la carpeta “*scripts*”. Para la construcción del repositorio hacer los pasos que se encuentran en el rubro Repositorio de consulta de la sección Implementación del cliente de análisis. Se debe tener el licenciamiento correspondiente de los productos de Oracle para la construcción y uso del repositorio.

Además, en la computadoras que fueron definidas como clientes de análisis hay que instalar la herramienta de modelación Visio 2002. Esta herramienta de modelación viene en la versión *Enterprise Architect* de los productos de .Net. Si no hay que utilizar los discos de instalación de la herramienta de modelación de Visio. Se debe tener el licenciamiento correspondiente para utilizar la herramienta de modelación Visio.

B. Listado de abreviaturas

Término	Significado
ADT	Tipo de dato abstracto
ANSI	Instituto de normas nacionales americanas
CD	Disco compacto
DBMS	Sistema de administración de la base de datos
DCL	Lenguaje de control de datos de SQL
DDL	Lenguaje de definición de datos de SQL
DML	Lenguaje de manipulación de datos de SQL
HTML	Extensión del archivo que identifica que este documento es una página que puede ser visualizar en un <i>browser</i>
IVA	Impuesto al valor agregado
NULL	Valor que identifica la carencia de dato
OLEDB	Implementación de Microsoft del concepto de acceso a datos universal
ORACLE	Corporación de desarrollo de software orientado a aplicaciones de bases de datos e Internet
SQL	Lenguaje de consulta estructurado utilizado por la mayoría de base de datos relacionales comerciales
URL	Localizador uniforme de recursos
XML	Lenguaje de marcas extensibles