



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad Y Computación

“Implementación de la Migración de la Base de Datos del Sistema CANOPUS de Informix 9.4C a Oracle 10g”

TESIS DE GRADO

Previa la obtención del Título de:

**INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN
SISTEMAS DE INFORMACIÓN**

Presentado por:

Gerardo Villagomez de Oliveira e Souza
Harold Álvarez Álvarez
Danny Vivanco Toala

GUAYAQUIL – ECUADOR

AÑO

2006

AGRADECIMIENTO

A la Escuela Superior Politécnica del Litoral por habernos alimentado de conocimiento valioso para representar a la Institución de la mejor manera; a todas las personas que nos apoyaron para la realización de la tesis, especialmente al Ingeniero Fabricio Echeverría Director de Tesis por su invaluable ayuda y comprensión.

DEDICATORIA

A Dios por haberme dado fuerzas y salir avante en mi carrera.

A mis padres quienes siempre les dedicaré mis triunfos.

A mi esposa por el apoyo continuo e incondicional que durante la Universidad supo sobrellevar.

A mis suegros por el aliento de superación que en todo momento me daban.

Gerardo Junior Villagomez de Oliveira e Souza

A Dios todopoderoso por haberme permitido alcanzar esta meta importante en mi vida.

A mis padres por su apoyo incondicional brindado durante todo este tiempo.

A la Armada del Ecuador y Escuela Superior Politécnica del Litoral por haberme dado una sólida formación académica

Harold David Álvarez Álvarez

A Dios por haberme dado la fortaleza y el discernimiento para salir adelante en mi carrera.

A mis padres que siempre han estado conmigo velando por mis estudios y a quienes le dedico mis éxitos.

A mi tío Sargento Gabriel Toala por ser como mi segundo papá que con su ayuda moral y económica he podido progresar en la carrera.

A mi tía Lcda. Alexandra Toala por ser como mi segunda mamá que con sus sabios consejos me han orientado a ser una mejor persona.

A mi tío Ing. Carlos Villafuerte por su ayuda en mi orientación académica.

A mi novia por estar conmigo en los buenos y malos momentos de mi carrera.

Danny Marcelo Vivanco Toala

TRIBUNAL DE GRADUACIÓN



Ing. Holger Cevallos
SUBDECANO FIEC



Ing. Fabricio Echeverría
DIRECTOR DE TESIS



Ing. Galo Valverde Landivar
VOCAL PRINCIPAL



Ing. Marcelo Loor
VOCAL PRINCIPAL

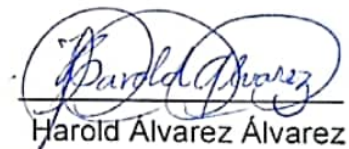
DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

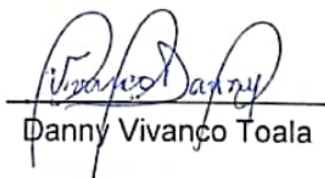
(Reglamento de Graduación de la ESPOL)



Gerardo Villagómez de O.e.S.



Harold Álvarez Álvarez



Danny Vivanco Toala

RESUMEN

La Armada del Ecuador busca la actualización, estandarización y calidad de las bases de datos utilizadas en los diferentes sistemas informáticos implementados en los distintos repartos navales, es así que el sistema naval CANOPUS, cuyo motor de base de datos es Informix versión 9.4C, ha sido considerado para migrarlo a Oracle versión 10g con la ayuda de las herramientas Power Builder y Power Designer de Sybase. Esta nueva base de datos es más liviana y segura además de soportar de mejor manera la escalabilidad de CANOPUS al permitir mantener un mayor grupo de usuarios.

Dentro del proceso de migración a la base de datos Oracle 10g, se va a trabajar específicamente con los tres módulos siguientes: **Modulo Principal**, al que se le va a modificar su codificación de tal manera que levante solamente el **Modulo Gerencial** y el **Modulo de Comité de Contrataciones**.

Para la migración procedimos a realizar primeramente un análisis de todas las tablas versus las diferentes operaciones y consultas que se realizan en los queries, de tal manera que había operaciones que no se utilizaban en el código, es decir, había código basura.

Posterior al análisis procedimos a realizar la migración de las tablas con sus respectivos índices luego de las vistas, triggers y al final de los procedimientos almacenados, con la utilización de las herramientas antes mencionadas.

Al final con todos los registros ya migrados procedimos a realizar las pruebas, tomando en cuenta que el código se encontraba en una tecnología antigua RDO (Random Data Object), se hizo cambios para mantenerlo a la tecnología ADO (ActiveX Data Objects) que es uno de los mecanismos que usan los programas de computadoras para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas.

ÍNDICE GENERAL

	Página
RESUMEN.....	VIII
ÍNDICE DE ABREVIATURAS.....	XIV
ÍNDICE DE FIGURAS.....	XVI
ÍNDICE DE TABLAS.....	XVII
ÍNDICE DE ANEXOS.....	XVIII
INTRODUCCIÓN.....	XIX

CAPÍTULO I.

I. DESCRIPCIÓN GENERAL

1.1 Reseña histórica del Sistema Canopus.....	1
1.2 Evolución del Sistema.....	4
1.3 Descripción del problema.....	16
1.4 Ambientes donde se produce el problema.....	18
1.5 Solución propuesta y su factibilidad.....	23
1.6 Objetivos de la solución.....	24

CAPÍTULO 2.

II. FUNDAMENTOS TEÓRICOS

2.1 Características relevantes de ORACLE 10g. aplicadas al proceso de migración.....26

2.2 Características relevantes de la base de datos INFORMIX 9.4C. aplicadas al Sistema Canopus.....36

2.3 Diferenciación y compatibilidad entre la base de datos INFORMIX versión 9.4C y ORACLE 10g.....44

CAPITULO 3.

III. REINGENIERÍA

3.1 Modelo lógico y físico del sistema Canopus en INFORMIX.....52

3.2 Propuesta del modelo lógico y físico del Sistema Canopus en ORACLE 10g.....57

3.2 Descripción de entidades susceptibles a mejoras.....59

CAPÍTULO 4.

IV. DESARROLLO DE LA MIGRACIÓN

4.1 Descripción de los pasos para realizar el movimiento de datos.....61

4.2 Rendimiento del Sistema Canopus en la nueva plataforma.....69

4.2.1 Comparación de tiempos y cambios realizados en el Front-end de la aplicación.....69

4.2.2 Comparación de tiempos utilizando consultas particulares.....	72
4.3 Codificación y prueba de aplicaciones.....	75
4.4 Funcionalidades.....	76

CAPÍTULO 5.

V. IMPACTO FINANCIERO

5.1 Beneficios cualitativos.....	77
5.2 Beneficios cuantitativos.....	79
5.3 Recursos y costos.....	80
5.3.1 Recursos usados para la implementación.....	80
5.3.2 Recursos mínimos necesarios.....	81
5.3.3 Recursos adicionales requeridos para la administración del sistema.....	82
5.3.4 Costo del software.....	82

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES.....	90
RECOMENDACIONES.....	92

BIBLIOGRAFÍA.....93

ANEXOS

[ANEXO A](#)

Diagnóstico de Migración de tablas

[ANEXO B](#)

Tablas Migradas

[ANEXO C](#)

Modelo físico del Sistema Canopus

[ANEXO D](#)

DDL en Informix y en Oracle

ANEXO E

Manual de Migración

ÍNDICE DE ABREVIATURAS

Abreviatura	Ingles	Español
DBA	Database administrator	Administrador de base de datos
DIGMAT		Dirección General del Material Armada del Ecuador
ODBC	Open Database Connectivity	Estándar de acceso a base de datos desarrollado por Microsoft.
ADO	ActiveX® Data Objects	Inteface Universal de acceso a data
RAID	Redundant Array of Independent Disks	Arreglo redundante de discos independientes
FMS	Fortran Monitor System	Sistema monitor de Fortran (Sistema Operativo)
IBSYS / SOS	SHARE Operating System	Sistema operativo de IBM
IBM	International Business Machines Corporation	
OS / SO	Operating System	Sistema Operativo
MS DOS	Microsoft Disk Operating System	Sistema operativo de Microsoft perteneciente a la familia DOS.
E/R	Entity-Relationship	Entidad relación
DED	Data Element Definition	Diagrama estructura de datos
DDL	Data Definition Language	Lenguaje de definición de datos
I/O	Input / output	Entrada / salida
CASE	Computer Aided Software Engineering	Ingeniería de Software Asistida por Ordenador
ANSI	American National Standards Institute	Instituto Nacional de Estándares Americanos
SPARC	Scalable Processor ARChitecture	Arquitectura escalable para procesador

SAN	Storage Area Network	Red de área de almacenamiento
DINFOR		Dirección de Informática de la Armada
DIGPER		Dirección General del Personal
DIGPERGYE		Base de datos de personal de la DIGPER Guayaquil
SGDB	Data Base Generator System	Sistema Generador de Base de Datos
DIRAFI		Dirección Administrativa y Financiera
DIGMAT		Dirección de General del Material
CEPROD		Centro de Procesamiento de Datos
DINDES		Dirección de Investigación y Desarrollo
DINCYP		Dirección de Ingeniería Civil y Portuaria
DINNAV		Dirección de Ingeniería Naval
DIECAR		Dirección de Electrónica, Comunicaciones y Armamento
DIRABA		Dirección de Abastecimiento de la Armada
CENABS		Centro de Abastecimiento
DDL	Data Define Lenguaje	Lenguaje de Definición de Datos

ÍNDICE DE FIGURAS

Figura1.1 Gráfico estadístico.....	13
Figura2.1 Instancia de Oracle.....	29
Figura2.2 Diagrama de Arquitectura Oracle.....	29
Figura2.3 Conexión remota.....	31
Figura2.4 Esquema general.....	32
Figura2.5 Estructura de la Tecnología ADO.....	44
Figura2.6 Procedimiento Migrado a Oracle.....	49
Figura2.7 Código del procedimiento.....	50
Figura2.8 Vistas migradas.....	51
Figura2.9 Triggers migradas.....	52
Figura2.10 Tablas migradas.....	52
Figura3.1 Modelo Físico COMCON.....	57
Figura4.1 Visualización de las tablas migradas.....	69

ÍNDICE DE TABLAS

Tabla 1.1 Requerimientos anteriores del Software.....	5
Tabla 1.2 Requerimientos actuales del Software.....	5
Tabla 1.3 Requerimientos del Hardware.....	7
Tabla 1.4 Módulos del Sistema CANOPUS.....	8
Tabla 1.5.1 Cuadro de reporte de los módulos e Sistema CANOPUS.....	9
Tabla 1.5.2 Cuadro de reporte de los módulos e Sistema CANOPUS.....	9
Tabla 1.6 Módulos de Sistema CANOPUS.....	12
Tabla 2.1 Tipos de datos ORACLE.....	34
Tabla 2.2 Recursos para migrar	39
Tabla 4.1 Reporte de estado del proceso de migración.....	64
Tabla 4.2 Rendimiento del Sistema CANOPUS.....	73
Tabla 4.3 Actividades de I/O del Sistema CANOPUS.....	74
Tabla 5.1 Puntos de red en los repartos.....	81
Tabla 5.2 Detalle de contrato con distribuidor IBM.....	85
Tabla 5.1 Lista de precios ORACLE.....	86

ÍNDICE DE ANEXOS

ANEXO A	Diagnóstico de Migración de tablas
ANEXO B	Tablas Migradas
ANEXO C	Modelo físico del Sistema Canopus
ANEXO D	DDL en Informix y en Oracle

INTRODUCCIÓN

Los avances tecnológicos han producido la necesidad de mantener actualizado a los Centros de Información tanto de Guayaquil como en Quito de la Armada del Ecuador, es por esto que entre los proyectos a implementar esta en hacer una Reingeniería del Sistema CANOPUS que fue desarrollada en 1993 con una duración de 10 meses, que agrupa varios módulos informáticos necesarios para el funcionamiento eficiente de la DIGMAT (Dirección General del Material de la Armada).

El objetivo el Sistema CANOPUS es automatizar los procedimientos de gestión de la Dirección General del Material y repartos subordinados, implementando sistemas de información diseñados para correr en una plataforma de arquitectura abierta, con características multiusuario cliente-servidor, que permita modernizar la administración y control de las direcciones técnicas y de abastecimientos para que contribuyan eficazmente en el proceso de toma de decisiones.

La solución del desarrollo de la migración del Sistema CANOPUS desde la Base de Datos de Informix a Oracle debe cumplir los siguientes objetivos:

1. Mover las estructuras de una Base de Datos a otra Base de Datos de diferentes marcas.
2. Definir el plan de pruebas que conllevan validación, verificación e inspección del código.
3. Utilizar la ingeniería reversa como herramienta de Ingeniería de Software para implementaciones de cambios de componentes e infraestructura.
4. Consolidar la información de manera segura estando siempre disponible además de contar con un tiempo de respuesta, escalabilidad y soporte requerido.
5. Obtener la más alta calidad de servicio en hardware modular.

CAPITULO 1

1. DESCRIPCIÓN GENERAL

1.1 Reseña histórica del Sistema Canopus

La Dirección General del Material (DIGMAT) dedicada a la compra, control y suministro de artículos de uso interno y externo; tales como: uniformes, suministros de oficina, material bélico, repuestos, entre otros, todo esto conforme a la demanda y al presupuesto asignado.

Del convenio Armada del Ecuador – Escuela Superior Politécnica del Litoral se inicia el proyecto CANOPUS en 1993 con una duración de 10 meses, que agrupa varios módulos informáticos necesarios para el funcionamiento eficiente de la DIGMAT

Definición del Sistema Canopus:

El Sistema Canopus es un Software que está compuesto por 8 módulos Operativos, 8 Administrativos y 5 Financieros (detallados en la Tabla No 3) e integrados entre si (cada módulo tiene su labor específica) que conforman el CANOPUS brindando así ayuda a todo el Sector del Material en el área administrativa y financiera, y apoyo a las unidades navales y operativas a cargo de la DIGMAT y los repartos navales subordinados.

Objetivo Canopus:

Automatizar los procedimientos de gestión de la Dirección General del Material y repartos subordinados, implementando sistemas de información diseñados para correr en una plataforma de arquitectura abierta, con características multiusuario cliente-servidor, que permita modernizar la administración y control de las direcciones técnicas y de abastecimientos para que contribuyan eficazmente en el proceso de toma de decisiones.

El sistema Canopus después de su implementación en la Armada ha permitido lo siguiente:

- Mejorar los requerimientos de información a los procesos.
- Permitir la captación de información tanto interna como externa.
- Incorporar canales eficientes de comunicación.
- Suministrar información oportuna y de fácil interpretación.
- Mejorar la capacidad para clasificar y agregar información.
- Agilizar la toma de decisiones.

Antecedentes:

- Se inicia en 1993 con DIRABA, 1'280,000 USD. IBM duración de 10 meses.
- 1995 CIA VTE paquete platinum 590,000 USD. con una duración de 9 meses.
- Varios estudios de campo que no llegaron a concretarse por onerosos.
- 14 Junio 1995 se genera un documento en que se establecen 7 fases (10 meses).

- Todos estos intentos se han diluido en estudios análisis sin existir un real comprometimiento de llevarlo a cabo.
- 1996 se creo un proyecto de sistematización T-02-1006 “Adquisición de equipos del sector del material” en el que se adquirió un servidor y 3 computadores los que servirían para desarrollar el proyecto planificado en 1995.
- 1997 se programó la evaluación y capacitación del personal que estará a cargo de la ejecución del proyecto en el desarrollo de módulos de acuerdo a la prioridad que fije la DIGMAT.

1.2 Evolución del Sistema

Requerimiento de Software:

ANTES

SOFTWARE	DESCRIPCIÓN DEL SOFTWARE
Microsoft Windows NT 4.0 en español	Este será utilizado en el Servidor y es necesario para el funcionamiento de la red.
Informix 7.x	Será el manejador de la Base de datos del Sistema Canopus.
Microsoft Windows 95 o	Este será utilizado en las estaciones

Windows 98 en español	de trabajo.
Microsoft Visual Basic 5.0 en español, Edición Empresarial con Licencia.	Será el lenguaje de programación utilizado para la implementación de los módulos.


Tabla 1.1 Requerimiento anterior del Software

ACTUAL

SOFTWARE	DESCRIPCIÓN DEL SOFTWARE
Microsoft Windows 2003 Server en español o LINUX Red Hat 9.0	Este será utilizado en el Servidor y es necesario para el funcionamiento de la red.
Informix 9.4C	Será el manejador de la Base de datos del Sistema Canopus.
Microsoft Windows XP en español	Este será utilizado en las estaciones de trabajo.
Microsoft Visual Basic 6.0 en español, Edición Empresarial con Licencia.	Será el lenguaje de programación utilizado para la implementación de los módulos.

Tabla 1.2 Requerimiento actual del Software

Requerimientos de Hardware:

EQUIPO	CARACTERÍSTICAS	DESCRIPCIÓN
Servidor 	Microprocesador	PENTIUM AKD26 MMX
	Velocidad	266 MHZ
	Arquitectura del BUS	PCI 64 bits
	Memoria RAM	128 MB
	Memoria Caché	512 KB
	Memoria de Vídeo	2 MB
	Disco Duro	6 GB
	Tipo de Monitor	SVGA 15'''
	Unidad de disquete	De 1.44 MB
	Teclado	101 Teclas
	Tarjeta de Vídeo	PCI
	Tarjeta de Red	3 COM
	Tarjeta de Sonido	32 Bits
	CD ROM	36X
	Microprocesador	PENTIUM MMX
	Velocidad	233 MHZ
	Arquitectura del BUS	PCI 32 bits

Estación de Trabajo	Memoria RAM	32 MB
	Memoria Caché	512 KB
	Memoria de Vídeo	2 MB
	Disco Duro	3.2 GB
	Tipo de Monitor	SVGA 15””
	Unidad de disquete	De 1.44 MB
	Teclado	101 Teclas
	Tarjeta de Red	3 COM
Impresora	Cualquier tipo de impresora de alta resolución, mínimo 360 x 360 DPI.	

Tabla 1.3 Requerimiento del Hardware

El Proyecto CANOPUS, consta de 21 sistemas modulares completamente integrados, los mismos que han sido desarrollados producto del Análisis de los procesos y la organización realizado en la Fase de Estudio.

Módulos Administrativos	Módulos Operativos	Módulos Financieros
Auditoria Interna	Activos Fijos	Contabilidad y Cuentas por cobrar
Comunicaciones Administrativas y Control de Archivo	Adquisiciones	Cuentas por Pagar
Control de Costos	Control de Reparaciones	Presupuesto Ejecución
Control Grafico de Repuestos	Control de Talleres	
Información Gerencial	Facturación y Pedidos	Presupuesto Planificación
Recursos Humanos	Importaciones	Activos Fijos
Seguridades	Inventario de Materiales y Repuestos	

Tabla 1.4 Módulos del Sistema CANOPUS

A continuación se detalla algunos de los módulos más utilizados por los señores Directores y Jefe de Departamentos así como sus respectivos reportes e informes:

Sistema	Opción
Presupuesto Planificación	CUADRO DE BIENES
	CUADRO DE SERVICIOS
Contabilidad	DETALLE DE CUENTAS POR KARDEX
	TOTAL DE CUENTA POR KARDEX
Cuentas Por Pagar	PAGOS PENDIENTES
	BENEFICIARIOS DE PAGOS
	REPORTE DE BENEFICIARIOS
	CUENTAS POR PAGAR POR REPARTO
Presupuesto Ejecucion	CHEQUES
	CEDULAS PRESUPUESTARIAS
	LISTADO DE COMPROMISOS
	LISTADO DE INVERSION E INGRESOS
	LISTADO DE DOCUMENTOS
	ANEXO BRAVO

Tabla 1.5.1 Cuadro de Reporte del los módulos del Sistema CANOPUS

Sistema	Opción
Facturación	LISTADO DE FACTURAS POR ITEM
	LISTADO DE NOTAS DE CREDITO
	LISTADO DE FACTURAS POR REPARTO
	LISTADO DE FACTURAS POR REPARTO DETALLADO
Pedidos	LISTADO DE PEDIDOS DE LAS BODEGAS
arios de Materiales y Repu	CONS. EXISTENCIAS POR ITEM
	CONS. EXISTENCIAS POR BODEGA
Adquisiciones	CONSULTA DE PROVEEDORES
	CONSULTA DE ORDENES DE COMPRA GENERAL
	CONSULTA DE ORDENES DE COMPRA DETALLADO
	REPORTE DE SEGUIMIENTO DE LA ORDEN DE COMPRA
Importaciones	PROVEEDORES
	NOTAS DE PEDIDO
	CÓDIGO DE IMPORTACIÓN

Tabla 1.5.2 Cuadro de Reporte del los módulos del Sistema CANOPUS

El proceso de migración comprende inicialmente los siguientes módulos:

- 1.- Sistema de Información General (SIG)
- 2.- Modulo de Comité de Contrataciones (COMCON)
- 3.- Módulo Sistema Distribuidor

1) Sistema de Información Gerencial:

Introducción:

El proyecto Canopus agrupa varios módulos informáticos necesarios para el funcionamiento eficiente de la Dirección General del Material DIGMAT, entre estos módulos se contempla el Sistema de Información Gerencial que se alimentara de varios módulos y proporcionara información estadística y globalizada disponible solo a los Directivos de la DIGMAT y personal autorizado.

Beneficios:

- Manipulación de datos y cálculos automáticos para evitar errores humanos.

- Obtener información oportuna y al día en el momento en que el usuario lo necesite.
- Ahorro de tiempo para la ejecución de todos los procesos automatizados.
- Existe un efectivo control de las actividades de la DIGMAT

Agilita la obtención de consultas y reportes por medio de un sistema que ofrece versatilidad de formatos para la presentación de los reportes estadísticos.

Logrará conocer con precisión como se están desarrollando las labores encomendadas a otros destacamentos de la DIGMAT, manejadas por el CANOPUS.

Comunicación constante con el resto de módulos del CANOPUS, lo que proporciona rapidez en la toma de decisiones.

El Sistema de Información Gerencial (SIG) se alimenta de la información almacenada en las tablas de los módulos siguientes:

Módulos	
Adquisiciones: Permite emitir reportes relacionados con las compras locales.	Inventario de materiales: Permite emitir reportes relacionados con los movimientos de los suministros, su consumo y/o demanda.
Importaciones: Permite emitir reportes relacionados con las compras en el exterior	Inventario de repuestos: Para emitir reportes relacionados con los movimientos de repuestos, su consumo y demanda.
Pagaduría: Permite emitir reportes relacionados con todas las cuentas por pagar que tenga la DIGMAT y su estado actual.	Planificación y Control de Obras: Permite emitir reportes de las obras planificadas y/o en ejecución
Activos fijos: Permite emitir reportes relacionados con los movimientos de los activos dentro de la DIGMAT	Reparaciones: Permite emitir reportes de las diferentes reparaciones realizadas a los buques, así como el estado de las mismas.
Presupuesto: Permite emitir reportes relacionados al presupuesto asignado o planificado a repartos de la DIGMAT	Recursos humanos: Permite emitir reportes relacionados con el personal de los diferentes repartos de la DIGMAT.

Tabla 1.6 Módulos del Sistema CANOPUS

SIG almacena la información seleccionada en una tabla temporal obteniéndose como salidas Reportes y Gráficos estadísticos.

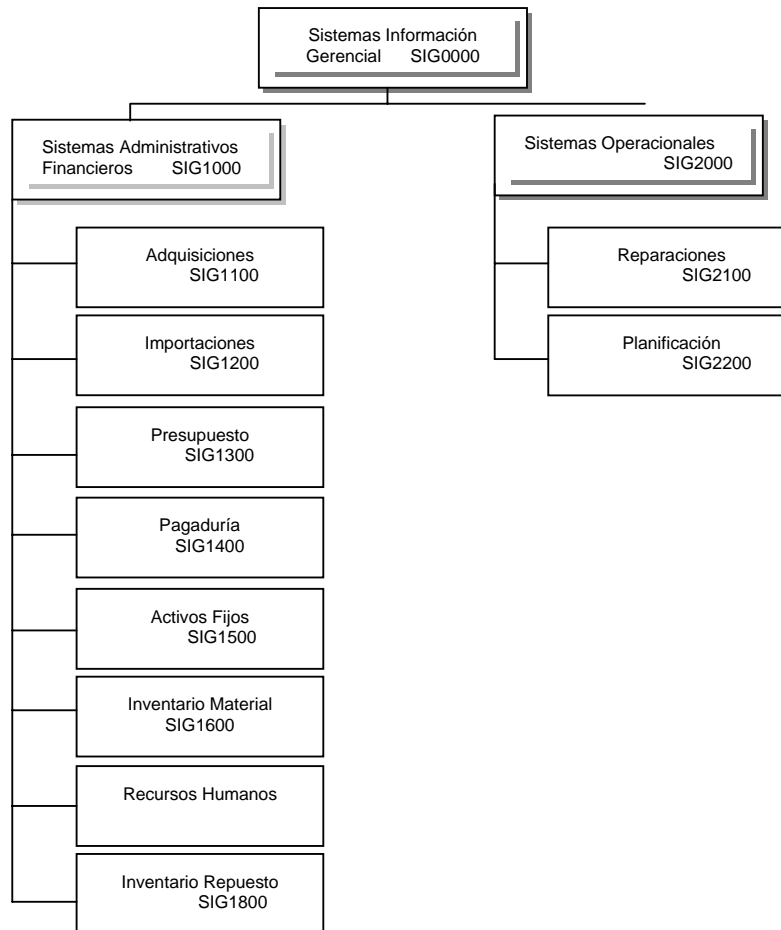


Figura # 1.1 Gráfico Estadístico

2) Modulo de Comité de Contrataciones COMCON

En el sistema CANOPUS existe el módulo de Comité de Contrataciones (COMCON), el cual fue creado para la

automatización del Departamento de Contratos a nivel administrativo a finales del año 2001.

Para el diseño de pantallas del módulo del COMCON se utilizó el estándar establecido para el sistema CANOPUS.

El módulo de COMCON tiene las siguientes opciones:

Mantenimiento:

Tipo de solicitud, tipo de materiales, tareas, proveedores, cargos, número de Sesiones.

Concurso:

Planificación anual, solicitud, sesiones.

Consulta/ Reportes:

Planificación, proveedores, concursos, concurso-permanencia en días, toma de decisiones.

Se evaluó la necesidad del departamento de Contratos para automatizar los procesos aplicados para el seguimiento de los proyectos que le son solicitados, y aplicar indicadores de gestión. Se ha determinado que en el módulo actual, COMCON, no permite realizar mencionados procesos.

Para ello se ha propuesto la siguiente solución:

Primera Etapa:

(Desarrollada por el CETEIG-DIGMAT)

1. Elaboración de una pantalla de ingreso de las tareas por proyecto. La misma que servirá para el registro de las tareas por cada proyecto, según su proceso, con el objetivo de realizar un seguimiento de avance de las etapas de cada proyecto, el cual no estaría enlazado con ningún módulo del sistema CANOPUS.
2. Pantalla de consulta en la cual el usuario podrá llevar el seguimiento de las tareas ingresadas.
3. Se realizará la migración de los datos que se encuentran ingresados en EXCEL.

Segunda Etapa:

(Desarrollada por el CETEIG-DIGMAT)

1. Definición y análisis de los indicadores que serán utilizados en la aplicación, el usuario deberá definir los indicadores que se utilizarán.

Tercera Etapa:

1. Desarrollo de una aplicación en Web que permita hacer consultas por parte del usuario.

1.3 Descripción del problema

Las bases de datos constituyen una parte fundamental de los sistemas de información en los que están integrados. El estado actual de la tecnología de base de datos es el resultado de la evolución que a lo largo de décadas ha tenido lugar en el procesamiento de los datos y en la gestión de la información. La tecnología de gestión de datos se ha desarrollado desde los métodos primitivos de los años cincuenta hasta los potentes sistemas de hoy en día, empujada por un lado por la demanda y las necesidades de gestión de la información y restringida por las limitaciones de la tecnología.

El Sistema “CANOPUS” cuyo propósito es la **Automatización del Sistema Logístico**, es un sistema que sirve para automatizar los procedimientos de gestión de la Dirección General del Material y repartos subordinados, implementando sistemas de información diseñados para correr en una plataforma de arquitectura abierta,

con características multiusuario cliente-servidor, que permita modernizar la administración y control de las direcciones técnicas y de abastecimientos para que contribuyan eficazmente en el proceso de toma de decisiones.

Este sistema que fue creado por completo en 1997 pero no con todos los módulos que se posee al momento, cuenta con una Base de datos INFORMIX versión 9.4 FC3 y el sistema operativo utilizado es Linux RedHat Advanced Server 3, para Itanium.

Una vez que IBM absorbió la Base de Datos INFORMIX con DB2 ha habido algunos problemas en las actualizaciones, es por esto una de las principales razones, que hace presumir que este sistema de base de datos está por desaparecer. Ya que las bases de datos se han convertido en un producto estratégico de primer orden, se ha llegado a la necesidad de mantener una tecnología de punta y realizar la migración de la Base de Datos de Informix a ORACLE, ya que este último es una base de datos que posee las siguientes características: obtenidas:

- Oracle posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Esto es porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.

- Oracle soporta bases de datos de todos los tamaños, desde bytes hasta severas cantidades de tamaño en gigabytes.
- Oracle provee salvar con seguridad de error lo visto en el monitor y la información de acceso y uso.

Oracle soporta un verdadero ambiente cliente servidor. Este establece un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.

1.4 Ambientes donde se produce el problema

La Dirección de Desarrollo Administrativo e Informático de la Armada con la misión de modernizar e implantar el desarrollo administrativo e informático de la Institución a través de la gestión por procesos, con personal especializado y apoyado por sistemas informáticos y de comunicación de datos, dentro del plan de tecnologías de la información considera la **optimización del sistema informático Canopus** con la implementación de la migración de su base de datos de Informix 9.4 FC3 a la plataforma Oracle 10g.

Esto además permitirá la **estandarización** con las demás Bases de Datos de otros repartos navales como de la Dirección de

Abastecimientos, el Instituto Oceanográfico, el Hospital Naval y entre otros.

Actualmente existe un solo proveedor de la plataforma Informix que es IBM, cuyas características son:

- Combina la robustez
- Alto performance
- Disponibilidad
- Escalabilidad

Situación Actual

Los principales problemas encontrados referentes a la aplicación de CANOPUS son:

- Bloqueo de los usuarios
- Bajo rendimiento para horas pico de trabajo
- Tiempo de respuesta.
- Programadores

Problemas de bloqueo de los usuarios.

Los bloqueos o locks permiten a la base de datos mantener la consistencia de datos entre las diferentes sesiones de usuario, y lo que hace es asegurar que ningún usuario pueda modificar o eliminar un dato mientras otro usuario este haciendo uso de él.

Los problemas de bloqueos de los usuarios no son exclusivos de base de datos INFORMIX, ocurren en todas las bases de datos que permitan trabajar en ambiente multiusuario. Cada SGDB (Sistema de Gestión de la Base de Datos) maneja de manera diferente la necesidad de asegurar la integridad de los datos que un usuario usa, y puede realizar los bloqueos a nivel de tabla, de página, y de registro.

La actual base de datos INFORMIX realiza en forma general el bloqueo de los datos a nivel de página, por lo que al utilizar al menos un campo de un registro, el SGBD bloqueara no solo el registro modificado sino también todos los registros que formen parte de la página, paralizando el trabajo de todos los usuarios que requieran aquellos datos.

EL SGBD actual nos brinda la posibilidad de aminorar los datos bloqueados usando la opción LOCK MODE ROW en la creación de

la tabla y con ello cuando hagamos uso de un registro éste será el único bloqueado, con lo que los usuarios que quieran acceder a otros registros de la misma página no resultaran impedidos de hacerlo.

Los problemas de bloqueo pueden deberse también a una mala estructura de programación de las transacciones que usan los objetos de la base de datos, lo que causa que un objeto pase bloqueado mas tiempo del estrictamente necesario.

También es causa de problemas de bloqueo el hecho de que un usuario que esta trabajando en las aplicaciones distraiga su atención a otros asuntos dejando tareas pendientes en su computador sin terminar la transacción ya sea grabando (COMMIT) o cancelando el proceso (ROLLBACK).

Bajo rendimiento de las Aplicaciones con carga de trabajo.

El bajo rendimiento de las aplicaciones con cargas de trabajo exigentes puede deberse a muchas razones entre las cuales tenemos:

Problemas de rendimiento del CPU, por ejemplo, como parte de los proyectos que está realizando la Armada para Estandarizar las Bases de Datos en Oracle, actualmente el Sistema CANOPUS consta de un Servidor Itanium para lo cual mejoró parte de su rendimiento pero aun tiene problemas en la velocidad de procesamiento.

- Mal diseño de estructuras de datos.- Relaciones incompletas entre tablas, falta de índices, etc.
- Mala técnica de programación, procesos que podrían desarrollarse en el servidor se realizan del lado del cliente.
- Problemas con la forma de conexión (ODBC) desde la aplicación al servidor, por ejemplo para la creación se necesita conseguir un cliente de Informix y a parte la interfaz no es tan amigable.
- Problemas de red (ver [ANEXO C](#) donde se muestra el modelo físico del Sistema CANOPUS), por ejemplo en el caso en que los usuarios realizan las transacciones, los tiempos de respuesta no son muy eficientes.

Nos veremos avocados a analizar los problemas inherentes al rendimiento de la Base de Datos como tal por lo que tocaremos los tres primeros puntos.

1.5 Solución propuesta y su factibilidad

Una solución para el problema en mención y basados en la ingeniería inversa y/o reingeniería, consiste en migrar la base de datos que radica en “Informix”, a “Oracle”, ya que Oracle presenta continuamente avances tecnológicos en su infraestructura de almacenamiento y de esta manera nos permite trabajar de forma novedosa y flexible.

A través de la migración, también podemos tener mejores tiempos de repuesta, muy reducido en cuanto al tiempo de respuesta propuesto por Informix; y permite hacer un buen Backup y Recuperación de datos, esto es debido a que existe un buen planeamiento y comprobación de los procedimientos de Backup y diferentes modos de recuperar un fallo en la base de datos.

Las buenas características presentadas por Oracle, han permitido optimizar las acciones por los DBA en cuanto se refiere a la toma de decisión de migrar de Informix a Oracle.

1.6. Objetivos de la Solución

El mercado de Manejadores de Bases de Datos (MBD) es muy amplio y ofrece las más variadas alternativas, por lo que se debe discernir:

- ¿Cuál de todas las plataformas de Base de Datos satisface las necesidades de brindar más y mejores servicios a sus usuarios?
- ¿Cuáles son los criterios en los cuales nos debemos basar para escoger de mejor forma la herramienta?

Analizando cada una de estas preguntas llegamos a la conclusión de que uno de los objetivos de la solución es contar con una base de datos reconocida como Oracle que soporte el desarrollo de aplicaciones en una variedad de herramientas y que una vez que la aplicación este en producción nos brinde la seguridad de que los trabajos realizados por la base de datos están 100% garantizados, nos ayudará mucho a la hora de resolver los múltiples problemas que pudieran surgir, lo que no ocurre si usamos bases de datos de código abierto que todavía en nuestro país son desconocidas cuyo soporte y experiencia son casi nulos, lo que resultará en la pérdida de valioso tiempo de investigación lo que no asegura la resolución del problema, y lo que es mas grave sin contar con el respaldo de una casa comercial.

Otros de los objetivos de tener una Base de Datos segura y confiable como Oracle son:

1. Mover las estructuras de una Base de Datos a otra Base de Datos de diferentes marcas.
2. Definir el plan de pruebas que conlleven validación, verificación e inspección del código.
3. Utilizar la ingeniería reversa como herramienta de Ingeniería de Software para implementaciones de cambios de componentes e infraestructura.
4. Consolidar la información de manera segura estando siempre disponible además de contar con un tiempo de respuesta, escalabilidad y soporte requerido.
5. Obtener la más alta calidad de servicio en hardware modular.

CAPITULO 2

2 FUNDAMENTOS TEÓRICOS

2.1 Generalidades de la base de datos de ORACLE 10g aplicadas al proceso de migración.

Oracle es un sistema de administración de base de datos (o RDBMS Relational Data Base Management System por las siglas en inglés), fabricado por Oracle corporation, básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, Mysql, Sql Server, etc.

Oracle, es una base de datos Orientadas a Objetos (BDOO) con beneficios OODBMS (Object Oriented DataBase Management System), ya que permite almacenar y manipular información que puede ser digitalizada (representada) por Objetos, proporciona una estructura flexible con acceso ágil, rápido con gran capacidad de modificación. Además combina las mejores cualidades de los archivos planos, las bases jerárquicas y las relacionales (Informix); donde soportan el análisis, diseño y Programación Orientada a Objetos.

Para desarrollar en Oracle utilizamos PL/SQL un lenguaje de 5ª generación, bastante potente para tratar y gestionar la base de datos, también por norma general se suele utilizar SQL.

Oracle es sin duda una de las mejores bases de datos que tenemos en el mercado, es un sistema gestor de base de datos robusto, tiene muchas características que nos garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la

competencia de gestores de bases de datos comerciales y de la oferta de otros con licencia Software Libre como PostgreSQL, Mysql o FireBird.

Con todas las características antes mencionadas de Oracle nos garantizan la seguridad de todos los datos migrados teniéndolos en un repositorio de alta confidencialidad, los cuales tuvieron que pasar por un proceso de migración utilizando la herramienta "Power Builder 10.0" que es muy poderosa, con múltiples opciones de la herramienta presentadas al usuario y paneles de crear objetos y editar sintaxis SQL y PL/SQL, compatible (en drivers) con Oracle y otros motores y además esta herramienta me brinda la facilidad de migración completa de todo un esquema de Informix a Oracle.

Tipos de Procesos

Los procesos en un Sistema de Oracle pueden ser categorizados en dos grandes grupos:

- Procesos de Usuario, corren la aplicación o el código de la herramienta de Oracle.
- Procesos de Oracle, corren el código del servidor de la Base de Datos de Oracle, eso incluye los procesos del servidor y procesos secundarios.

La estructura del proceso varía para diferentes configuraciones de Oracle, esto depende del sistema de operación y de la opción que escojas de Oracle. El código para usuarios conectados puede ser configurado como un servidor dedicado o un servidor compartido.

En la figura 2.1 ilustra una configuración del servidor dedicado donde cada usuario conectado tiene un proceso de usuario por separado y muchos procesos secundarios corriendo en ORACLE.

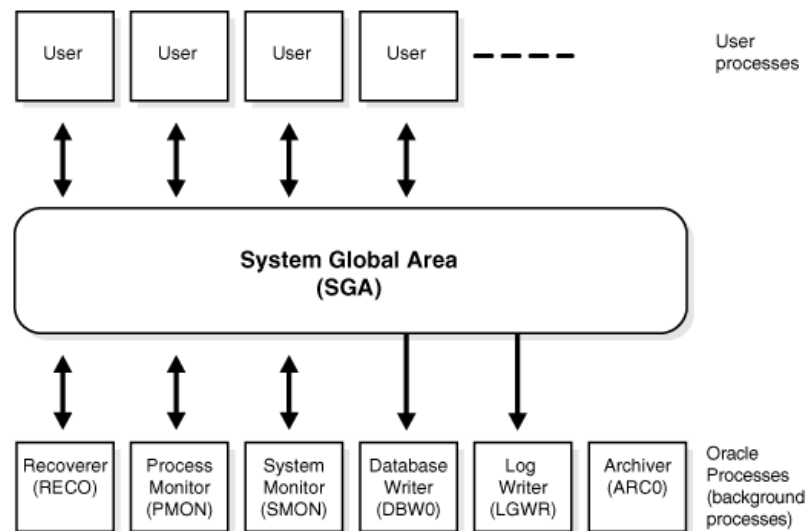


Figura # 2.1 Instancia de Oracle ⁽²⁾

En un servidor se pueden crear varias instancias, pero no es recomendable ya que cada instancia consume muchos recursos.

⁽²⁾ Por cada instancia de Oracle se tiene una sola base de datos.

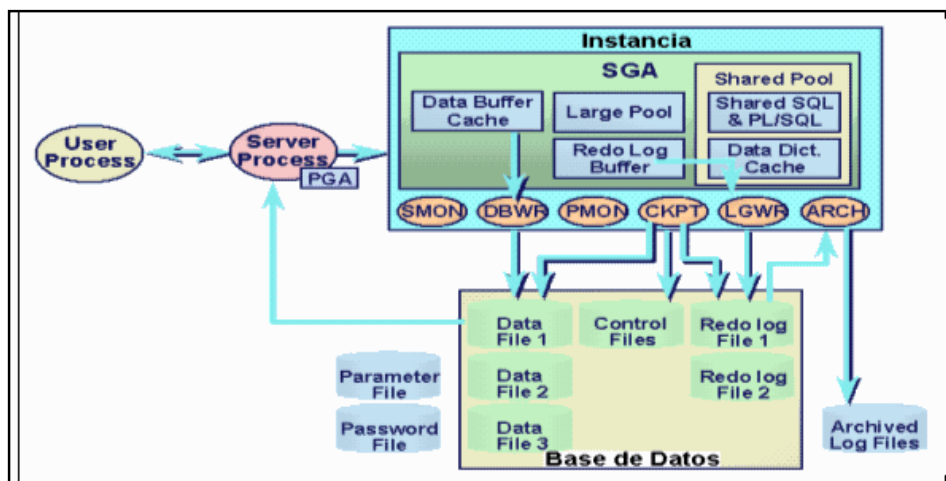


Figura # 2.2 Diagrama de arquitectura Oracle ⁽³⁾

Una instancia de Oracle está conformada por varios procesos de fondo y espacios de memoria compartida denominada System Global Area (SGA) que son necesarios para acceder a la información contenida en la base de datos.

La instancia está conformada por procesos del usuario, procesos que se ejecutan en el background de Oracle y los espacios de memoria que comparten estos procesos.

El SGA es utilizado para el intercambio de datos entre el servidor y los clientes.

⁽³⁾ Fuente: Manual de Oracle, autor Diego Burbano

Una instancia de oracle solo puede abrir una sola base de datos a la vez.

- **Proceso Usuario.** Programa, aplicación ó herramienta que usa el usuario para iniciar un proceso de usuario y establecer una conexión.
- **Proceso Servidor.** Una vez que el proceso del usuario establece conexión, un proceso servidor es iniciado, el cual manejará las peticiones del usuario. Un proceso servidor puede ser dedicado, es decir, solo atiende las peticiones de un solo proceso usuario, ó puede ser compartido, con lo cual puede atender múltiples procesos usuario.
- **Sesión.** Una sesión es una conexión específica de un usuario a un servidor Oracle. Se inicia cuando el usuario es validado por el servidor Oracle. Finaliza cuando el usuario termina la sesión en forma normal (log out) ó aborta la sesión.

Conexión Remota

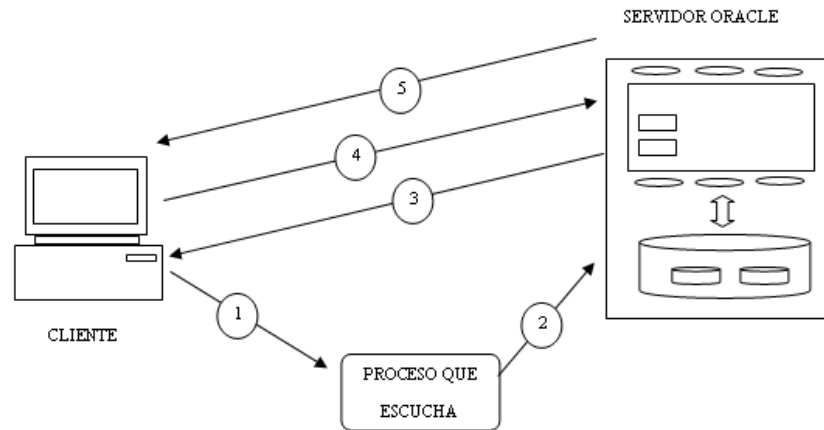


Figura # 2.3 Conexión Remota

Conexión a la instancia Oracle

De la relación de servicios creados durante la instalación de Oracle, por ahora nos interesan básicamente dos puntos:

- El servicio relacionado con la instancia y la base de datos, cuyo nombre tiene la siguiente estructura: OracleServiceXXX, donde XXX representa el nombre de la instancia.
- El servicio relacionado con la disponibilidad de servidor para el acceso remoto, el nombre del servicio es: OracleOraHome92TNSListener.

Para una correcta conexión, estos dos servicios deben estar ejecutándose.

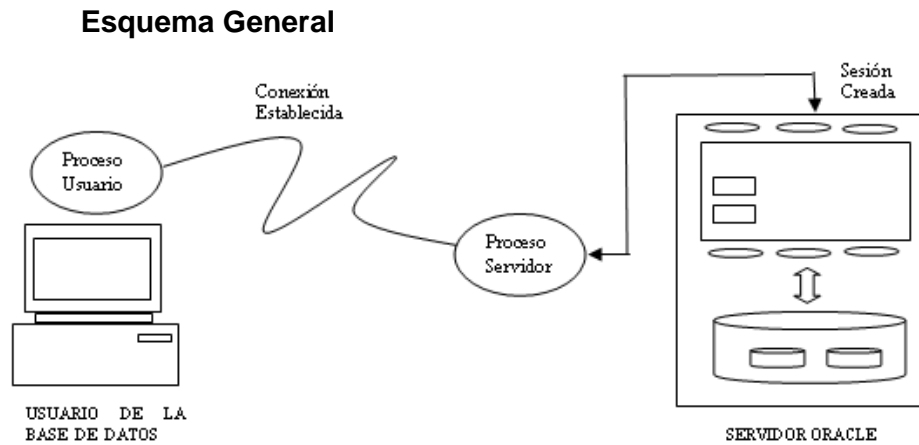


Figura # 2.4 Esquema general

Oracle tiene su herramienta de red que permite a las aplicaciones en general conectarse a servidores Oracle. Para que una aplicación pueda conectarse a un servidor Oracle, es necesario que el Proceso “Escucha” se encuentre ejecutándose en el servidor (OracleOraHome92TNSListener).

El proceso para la conexión se describe a continuación:

1. El cliente establece una conexión al proceso Escucha usando el protocolo configurado y envía un paquete CONNECT.
2. El proceso Escucha comprueba que el SID esté definido. Si es así, genera un nuevo proceso para ocuparse de la conexión. Una conexión, se establece en el proceso Escucha y el nuevo

proceso del servidor para pasarle la información del proceso de inicialización. Luego la conexión es cerrada.

3. El proceso del servidor envía un paquete al cliente.
4. Un nuevo paquete CONNECT es enviado al proceso servidor dedicado.
5. El proceso del servidor dedicado acepta la conexión entrante y remite un mensaje de ACEPTADO al nuevo cliente.

Tipos de datos

Los tipos de datos Oracle se agrupan en los siguientes conjuntos:

Tipos de Datos				
Alfanuméricos	Numéricos	Fecha	Binarios	Otros
CHAR	NUMBER	DATE	RAW	ROWID
VARCHAR(2)	FLOAT	TIMESTAMP	LONG RAW	UROWID
VARCHAR		TIMESTAMP WITH TIME ZONE	BLOB	
NCHAR		INTERVAL	CLOB	
NVARCHAR(2)			NLOB	
LONG			BFILE	

Tabla 2.1 Tipos de datos Oracle

Oracle 10g ofrece a los clientes diversas soluciones a sus problemas, ya que permite la integración de sistemas sumamente grandes sin tener inconvenientes con el manejo de los datos, ofrece también una buena integridad de datos.

La única base de datos diseñada para el cómputo en red, Oracle Database 10g ofrece desempeño, escalabilidad, disponibilidad, seguridad y facilidad de administración superior en una red de bajo costo de sistemas de almacenamiento y servidores estándar de la industria. Oracle Database 10g está diseñada para implementarse de manera efectiva en todos los sistemas, desde pequeños servidores tipo blade hasta los más grandes servidores SMP y clústeres de todos los tamaños. Cuenta con capacidades de administración automatizada para una operación fácil y económica. La capacidad única de Oracle Database 10g para el manejo de datos, desde información tradicional de negocios hasta documentos XML e información espacial y de ubicación, la convierte en la opción ideal para el procesamiento de transacciones en línea, soporte a decisiones y aplicaciones de administración de contenido.

2.2 Generalidades de la base de datos de INFORMIX 9.4C.

IBM Informix Dynamic Server 9.4 (IDS) continúa la larga tradición de IBM e Informix de desarrollar una base de datos de primer nivel. Combina la robustez, alto performance, disponibilidad y escalabilidad que necesitan las empresas en la actualidad.

Una base de datos de misión crítica requiere una combinación de procesamiento de transacciones online (OLPT), procesamiento de transacciones Batch y procesamiento análisis online (OLAP).

Informix IDS está construido sobre Informix Dynamic Scalable Architecture (DSA), la cual brinda una de las soluciones más efectivas disponibles – una próxima generación de bases de datos paralelos que permitirán mejor escalabilidad, administración y performance, aminorar las cargas del sistema, redistribución de las cargas de trabajo, y la capacidad de extender los servidores para manejar nuevos tipos de datos. Con la versión 9.4 IBM IDS altera el panorama con significativos incrementos en la capacidad de almacenamiento de información y de procesamiento que no se encontraran en ningún otro producto.

IBM IDS soporta una gran variedad de herramientas para un rápido desarrollo de aplicaciones bajo Linux, Windows y UNIX; y brinda soporte para 4GL, SQL, ODBC, JDBC, OLE/DB, SQLJ, etc.

IBM IDS soporta consistencia a nivel de transacciones e integridad de datos mientras simplifica la optimización y administración de los datos.

IBM IDS soporta Logging de transacciones, lo que los mismos que fueron incrementados en su tamaño hasta el máximo de 1GB.

Dado el amplio rango de tipos de datos usados por IBM IDS 9.4, los administradores de bases de datos y desarrolladores de aplicaciones pueden definir estructuras de datos y reglas que reflejen más detalladamente los ambientes de negocios.

IBM ha adicionado tipos de datos como boolean, int8, serial8, y lvarchar que es el tipo de dato carácter más grande que se ha creado 32Kb.

Dentro de IBM IDS 9.4 el tamaño máximo de cada Chunk es de 4Tb. Sumado a esto el tamaño máximo de cada instancia fue incrementado a 32767. Como resultado IBM IDS 9.4 tiene una capacidad total de 128 petabytes – es decir 128×10^{15} bytes. Esto asumiendo que el administrador de base de datos no usa disk-

Mirroring para fallos de tolerancia. Usando Mirroring dobla el número total de chunks y por ende la capacidad de almacenamiento.

IBM IDS posee herramientas de respaldo de recuperación automatizadas, incrementa la eficiencia en el manejo de los tape backup y tiene muchas mejores opciones de restauración.

IBM IDS posee herramientas de administración basadas en Web.

Podemos decir que IBM IDS 9.4 es una base de datos totalmente confiable, la misma que puede ser usado en los entornos más críticos de trabajo y con una gran cantidad de usuarios.

Es cierto que encontrar empresas que brinden soporte de Informix no es tan fácil como puede ser con otras bases de datos como ORACLE y SQL Server; no deja de ser cierto también que en el mercado laboral y en la comunidad informática de nuestro país hay mucha gente preparada y capacitada para absolver los problemas que pudiesen surgir por lo que mantenerse en Informix no debe dejar de ser una opción válida para el CETEIG.

Informix es compatible con las siguientes aplicaciones:

Java, including JDBC 3.0 and JRE 1.4

C, C++

Visual BASIC .NET

Visual C# .NET

Visual J# .NET

ASP.NET web

A continuación se muestra una tabla en la que se ve la cantidad de recursos necesarios tanto para Informix como para Oracle. Oracle necesita de más recursos para su instalación.

IBM Informix Dynamic Server 10		Oracle 10g
RAM	256 MB	750 MB
Download Size	80 MB	600 MB
Disk for Install	500 MB*	1000 MB
Overhead	22 Bytes/Page	53 Bytes/Page

Tabla 2.2 Recursos para migrar

Generalidades del proceso de migración.

Tecnología ADO

ADO (ActiveX Data Objects) es uno de los mecanismos que usan los programas de computadoras para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas.

Con ADO, un programa puede leer, insertar, editar, o borrar, la información contenida en diferentes áreas de almacenamiento dentro de la base de datos llamadas tablas. Además, se puede manipular la propia base de datos para crear nuevas áreas para el almacenamiento de información (tablas), como también alterar o eliminar las ya existentes, entre otras cosas.

Fue desarrollado por Microsoft y es usado en ambientes Windows por lenguajes de programación como Visual Basic, C++, Delphi entre otros, como también en la Web mediante el uso de Active Server Pages (ASP) y el lenguaje VBScript.

Evolución:

ADO substituyó tanto a DAO (Data Access Object), como a RDO (Remote Data Object), que eran los sistemas previos que se usaban para acceder a las bases de datos y bases de datos remotas, respectivamente. Tiene la mayor parte de la funcionalidad de ambos modelos y sin embargo es más sencillo de usar y de entender y por lo tanto más fácil y menos engorroso de programar.

La última versión de ADO, creada por Microsoft, se llama ADO.NET, y se usa en los entornos de programación de la plataforma .NET, de Microsoft, para manejar bases de datos tanto

en Windows como en la Web mediante ASP.NET, que es la nueva versión del ASP para la plataforma .NET.

Funcionamiento

ADO es un intermediario entre el programa y la base de datos. El programa no ve la base de datos directamente, sino que hace todo el trabajo a través de ADO. Usando ADO, el programa se comunica con la base de datos, consulta, edita, inserta, borra, registros, añade tablas, etc. ADO a su vez se comunica con la base de datos a través de un "proveedor de datos".

Principales componentes de ADO

- **Connection:** Permite establecer una conexión con la base de datos
- **Recordset:** Maneja un conjunto de records de la base de datos
- **Command:** Permite enviar órdenes SQL para ser ejecutados por la base de datos.

La Conexión

La conexión es como una autopista que permite el flujo de datos entre el programa y la base de datos. Por ella pueden viajar las órdenes que desde el programa se usan para hacer solicitudes de

información a la base de datos o para realizar una operación dentro de ella como borrar registros, añadir registros, modificar tablas, etc. También, por esta autopista, pueden ir y venir los datos, desde y hacia la base de datos, entre otras cosas.

Tanto el recordset como la orden usan la conexión para comunicarse con la base de datos.

La conexión se comunica con la base de datos a través de un intermediario llamado "proveedor de datos".

El proveedor de datos es un componente que se relaciona directamente con la base de datos. Hay un proveedor de datos por cada tipo de base de datos. Así, las bases de datos de tipo Access, SQL Server, Oracle, MySQL, tienen, cada una, un proveedor de datos específico.

La conexión ADO puede usar dos tipos de proveedores de datos, *OLE DB* y *ODBC*, siendo *OLE DB* el tipo de proveedor nativo⁽⁵⁾.

Cuando no existe un proveedor de *OLE DB* específico para una base de datos determinada, y en cambio existe un proveedor *ODBC*, la conexión ADO puede usarlo para comunicarse con la base de datos, sin embargo, no directamente, sino a través de un

⁽⁵⁾ Nativo se refiere a que la conexión ADO se inició con el proveedor *OLE DB*.

proveedor OLE DB especial que sirve de intermediario entre ADO y ODBC.

Recordset:

El recordset es, como su nombre lo indica, un conjunto de records (registros). En general, sus datos tienen su origen en una base de datos, aunque también pueden generarse independientemente de ésta.

Un recordset puede contener cero o más records (registros). Cada recordset tiene una colección de campos, que es común a todos los records. Podemos verlo como una matriz o tabla, en donde las filas son los records, y las columnas son los campos.

Command:

Command es el componente ADO que permite hacer solicitudes o dar órdenes a la base de datos mediante una sentencia SQL.

Se puede especificar la inserción de registros nuevos en una tabla, así como también, la eliminación la actualización y la obtención de registros con determinadas condiciones.

Además, se pueden crear, alterar y modificar las características de las tablas que conforman la base de datos.

En la figura 2.5 podemos observar como está estructurada la plataforma bajo la tecnología ADO.

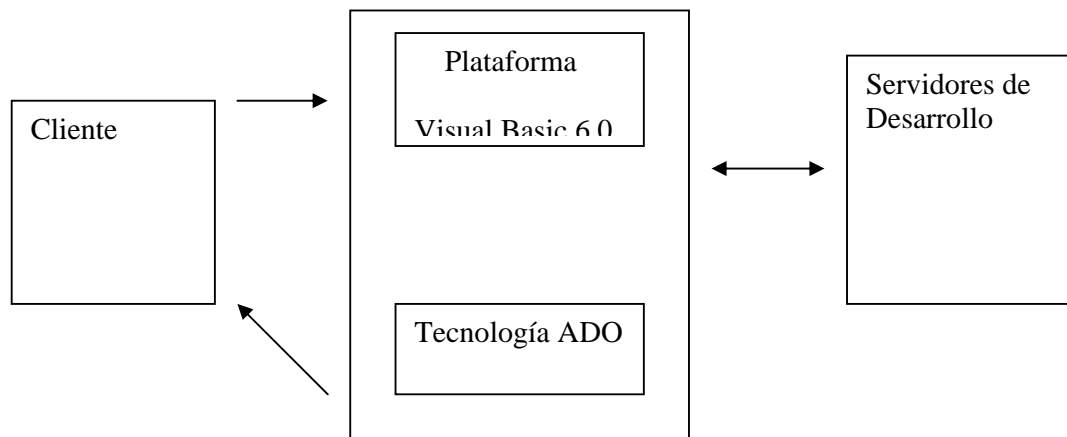


Figura # 2.5 Estructura de la tecnología ADO

2.3 Diferenciación y compatibilidad entre la base de datos INFORMIX versión 9.4C y ORACLE 10g.

Diferencias:

- Oracle es un motor de Base de Datos relacionales es decir, tiene una colección de relaciones, y se considera que la relación es representada por una tabla, y en la que cada fila representa una colección de valores que describen una entidad del mundo real se la conoce como registro y cada columna se denomina campo. En tanto Informix es un motor de Base de Datos OLTP es decir, aquellas bases en las que

los datos suelen estar organizados en tablas relacionadas con el fin de reducir la información redundante y aumentar la velocidad de las actualizaciones.

- Oracle siempre ha sido considerada una base de datos para uso más general que Informix. Informix por su lado, se especializó más en aplicaciones tipo GIS (datos geográficos), Datawarehouse y Datamining.
- En cuanto a precios, Informix tiene tendencia a ser más caro que Oracle en configuraciones parecidas.
- En la práctica, como buenos enemigos acérrimos, ambas tienen parecidas características y funcionalidades. Cada una tiene las típicas ampliaciones que permiten especializar la base de datos a un cierto tipo de aplicación (en Informix se llaman DataBlades, ver imagen 3). Para competir, en cada nueva versión que sacaban, las dos iban añadiendo de serie muchas de las extensiones que en la versión anterior eran opcionales (de pago por separado). De cara a una instalación y configuración básicas, las diferencias frente a Oracle que el lector va a notar son dos:
 - En Oracle, tienes que definir los usuarios dentro la base de datos (gestión interna de usuarios). En cambio, Informix utiliza

los mismos usuarios de Linux (los que creas con adduser), simplificando la administración.

- Para conectar externamente vía TCP/IP a Oracle, hacía falta colgar un daemon llamado listener de un puerto. Luego el listener traducía las llamadas al SQLNet para hablar con la base de datos. Luego había dos procesos: el daemon de la base de datos (oracle), y el daemon que escuchaba el puerto (listener). En Informix, el mismo daemon de la base de datos (ONINIT) atiende los puertos. De esta manera Informix ocupa menos memoria y recursos.
- La Base de Datos de 10g se demostró ser considerablemente más fácil de manejar que Informix 9.4C, haciendo posible el incremento de productividad y ahorro en costos para los administradores de bases de datos (DBAs).

Compatibilidades:

- Son lenguajes de 4ta. Generación.
- Poseen igual interacción en todas las plataformas (Windows, Linux, Macintosh, etc).
- Informix y Oracle tienen funcionalidades en cuanto a respaldos, administración, procesamiento Multithreading, Clustering, control de bloqueos, procesamiento paralelo de sentencias SQL, etc. que la mayoría de bases de código abierto no tienen.

- Oracle e Informix apuestan más bien por sistemas híbridos antes que por la orientación a objetos pura.

Durante la Migración

En el proceso de migración de Informix a Oracle se observaron diferencias en las dos plataformas.

Presentaremos a continuación un análisis de todas las estructuras migradas de Informix a Oracle:

- Durante la migración de las tablas se presentaron inconvenientes con el número de registros de alguna de ellas, es decir, cuando tratábamos de migrar tablas en volumen nos arrojaba un error de registros. **En informix** se escribe de la siguiente forma:

```
CREATE TABLE "INFORMIX"."ADQM_PORC_SEGU"
("PORC_CODI_ASEG"
  NUMBER(10) NOT NULL, "PORC_TIPO_TRAM"
  CHAR(3 byte) NOT NULL,
  "PORC_CODI_TRAM" CHAR(3 byte) NOT NULL,
  "PORC_VALO_PORC"
  NUMBER(5, 2))
```

- En Oracle se escribe de la siguiente forma:

```
create table ARMADA.ADQM_PORC_SEGU
(
  PORC_CODI_ASEG          FLOAT
  not null,
  PORC_TIPO_TRAM         CHAR(3)
  not null,
```

```

        PORC_CODI_TRAM          CHAR(3)
    not null,
        PORC_VALO_PORC
NUMBER(5,2)                    not
null
)

```

- En la migración de los procedimientos almacenados se dieron diferencias tanto en la sintaxis como en el manejo de errores.

En Informix se escribía de la siguiente forma:

```

create procedure "informix".contar_reg(tabla varchar(50))
    returning int, int;
    define cod_err int;
    define num_fila int;
    ON EXCEPTION SET cod_err
        RETURN cod_err, 0;
    END EXCEPTION
    SELECT COUNT(*) INTO num_fila FROM tabla;
    return 0, num_fila;
end procedure;

```

En cambio en Oracle el procedimiento almacenado se maneja de otra forma donde incluimos las excepciones para el manejo de errores:

```

create or replace procedure  contar_reg ( tabla in VARCHAR2 )
as
cod_err NUMBER(10);
num_fila NUMBER(10);
contador_no_permitido
EXCEPTION;
dmens_error varchar2(1024);
dmens_default varchar2(256);
datos_no_encontrado exception;
begin
    SELECT COUNT(*) INTO num_fila FROM tabla;
EXCEPTION
WHEN contador_no_permitido THEN
cod_err := -20001;
WHEN datos_no_encontrado THEN
cod_err := -20002;
dmens_DEFAULT := 'NO EXISTEN REGISTROS EN LA TABLA';
dmens_error := TRAE_ERROR (cod_err, dmens_DEFAULT);

```

```
RAISE_APPLICATION_ERROR (-20999, dMENS_ERROR);
end;
```

En el código Informix se puede observar palabras reservadas como “returning”, “return” y “define” que en Oracle no se utilizan, por lo que los mismos fueron reemplazados por su respectiva sintaxis.

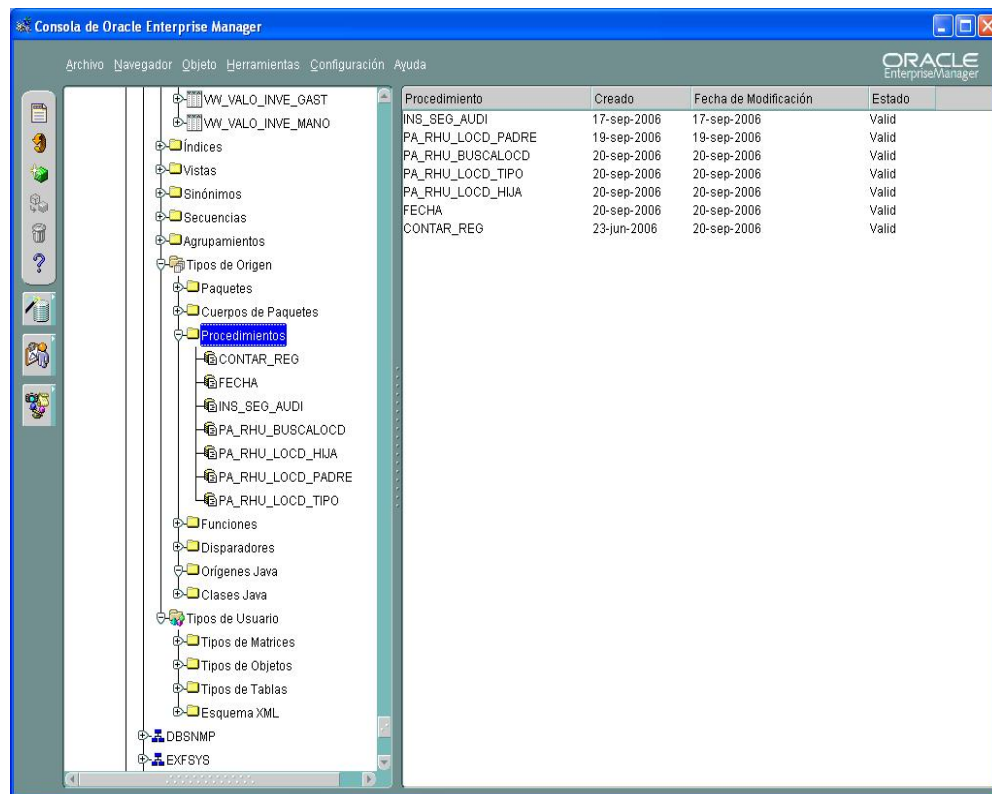
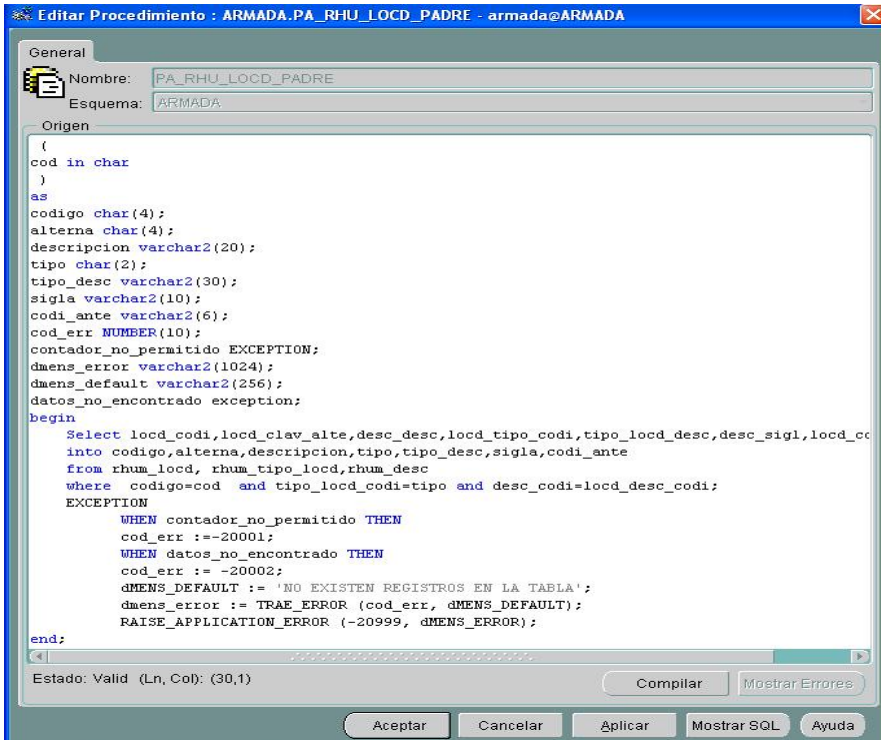


Figura # 2.6 Procedimiento migrado a Oracle

Lo que podemos concluir es que los datos no fueron migrados plenamente porque el manejo de errores en Informix como en la sintaxis de simbología es diferente. Así mismo utilizamos

como herramienta de migración el Enterprise Manager del cliente Oracle como se muestra en la figura # 2.4, el cual nos presentaba los errores de forma más detallada en caso de haberlo y así nos ayudó a tener éxito en la migración de los procedimientos almacenados .



```

General
Nombre: PA_RHU_LOCD_PADRE
Esquema: ARMADA
Origen
(
cod in char
)
)
as
codigo char(4);
alterna char(4);
descripcion varchar2(20);
tipo char(2);
tipo_desc varchar2(30);
sigla varchar2(10);
codi_ante varchar2(6);
cod_err NUMBER(10);
contador_no_permitido EXCEPTION;
dmens_error varchar2(1024);
dmens_default varchar2(256);
datos_no_encontrado exception;
begin
Select locd_codi,locd_clav_alte,desc_desc,locd_tipo_codi,tipo_locd_desc,desc_sigla,locd_co
into codigo,alterna,descripcion,tipo,tipo_desc,sigla,codi_ante
from rhum_locd, rhum_tipo_locd,rhum_desc
where codigo=cod and tipo_locd_codi=tipo and desc_codi=locd_desc_codi;
EXCEPTION
WHEN contador_no_permitido THEN
cod_err :=-20001;
WHEN datos_no_encontrado THEN
cod_err := -20002;
dmens_DEFAULT := 'NO EXISTEN REGISTROS EN LA TABLA';
dmens_error := TRAE_ERROR (cod_err, dmens_DEFAULT);
RAISE_APPLICATION_ERROR (-20999, dmens_error);
end;
Estado: Valid (Ln, Col): (30,1)
Compilar Mostrar Errores
Aceptar Cancelar Aplicar Mostrar SQL Ayuda

```

Figura # 2.7 Código del procedimiento

- En la migración de vistas no hubo mayores diferencias en la sintaxis, para la migración utilizamos la herramienta de PowerBuilder en donde el único cambio era el de eliminar la palabra “Informix” para crear el nombre de la vista. En la figura

2.6 podemos ver las vistas migradas. Podemos concluir que hubo compatibilidad en la migración.

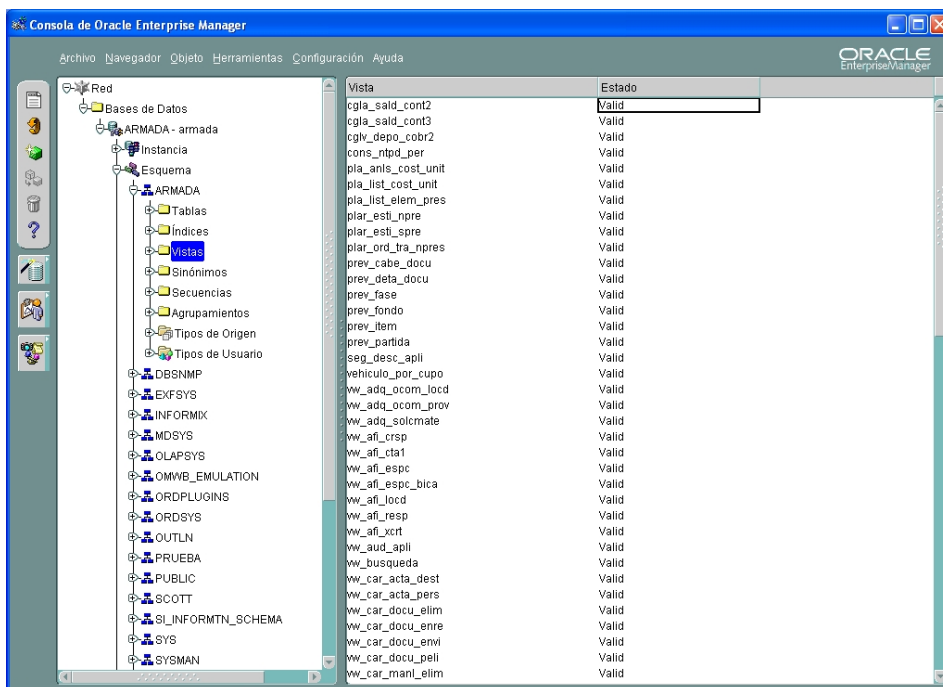


Figura # 2.8 Vistas migradas

- Durante la migración de los Triggers si hubo diferencias, se dieron errores como *“Faieled to parse trigger text (...) Parse error al line (...). Encountered...”*; así mismo utilizamos PowerBuilder como herramienta de migración, había que eliminar la palabra *“Informix”* en el nombre del trigger, por ejemplo *“create trigger “informix”.trga_adqt_fpagd delete on “informix”.adqt_fpag”*. En la figura 2.7 muestra los triggers muestra los triggers migrados.

Disparador	Tipo	Evento	Propietario de la Tabla	Tabla
TGLADOT_DETETA_OCOMO	AFTER EACH ROW	DELETE	ARMADA	ADOT_DETETA_OCOM
TGLADOT_DETETA_OCOMI	AFTER EACH ROW	INSERT	ARMADA	ADOT_DETETA_OCOM
TGLADOT_DETETA_OCOMU	AFTER EACH ROW	UPDATE	ARMADA	ADOT_DETETA_OCOM
TGLADOT_DOCU_RUBERD	AFTER EACH ROW	DELETE	ARMADA	ADOT_DOCU_RUBR
TGLADOT_DOCU_RUBRI	AFTER EACH ROW	INSERT	ARMADA	ADOT_DOCU_RUBR
TGLADOT_DOCU_RUBRU	AFTER EACH ROW	UPDATE	ARMADA	ADOT_DOCU_RUBR
TG_UPD_CABE_OCOM	AFTER STATEMENT	UPDATE	ARMADA	ADOT_CABE_OCOM
TG_UPD_CABE_OCOMD	BEFORE EACH ROW	DELETE	ARMADA	ADOT_CABE_OCOM
TG_UPD_CABE_OCOMI	AFTER EACH ROW	INSERT	ARMADA	ADOT_CABE_OCOM
TRGA_ADOT_CABE_D	AFTER STATEMENT	DELETE	ARMADA	ADOT_CABE_COTI
TRGA_ADOT_CABE_I	AFTER STATEMENT	INSERT	ARMADA	ADOT_CABE_COTI
TRGA_ADOT_CABE_SOLC	AFTER EACH ROW	DELETE	ARMADA	ADOT_CABE_SOLC
TRGA_ADOT_CABE_SOLCU	AFTER EACH ROW	UPDATE	ARMADA	ADOT_CABE_SOLC
TRGA_ADOT_CABE_SOLI	AFTER EACH ROW	INSERT	ARMADA	ADOT_CABE_SOLC
TRGA_ADOT_CABE_U	AFTER STATEMENT	UPDATE	ARMADA	ADOT_CABE_COTI
TRGA_ADOT_CONT_GARD	AFTER STATEMENT	DELETE	ARMADA	ADOT_CONT_GARA
TRGA_ADOT_CONT_GARI	AFTER STATEMENT	INSERT	ARMADA	ADOT_CONT_GARA
TRGA_ADOT_CONT_GARU	AFTER STATEMENT	UPDATE	ARMADA	ADOT_CONT_GARA
TRGA_ADOT_CUAD_OFERD	AFTER STATEMENT	DELETE	ARMADA	ADOT_CUAD_OFER
TRGA_ADOT_CUAD_OFERU	AFTER STATEMENT	UPDATE	ARMADA	ADOT_CUAD_OFER
TRGA_ADOT_DETETA_AFI	AFTER STATEMENT	DELETE	ARMADA	ADOT_DETETA_AFI
TRGA_ADOT_DETETA_AFIU	AFTER STATEMENT	INSERT	ARMADA	ADOT_DETETA_AFI
TRGA_ADOT_DETETA_AFIU	AFTER STATEMENT	UPDATE	ARMADA	ADOT_DETETA_AFI
TRGA_ADOT_DETETA_COTID	AFTER STATEMENT	DELETE	ARMADA	ADOT_DETETA_COTI
TRGA_ADOT_DETETA_COTI	AFTER STATEMENT	INSERT	ARMADA	ADOT_DETETA_COTI
TRGA_ADOT_DETETA_COTIU	AFTER STATEMENT	UPDATE	ARMADA	ADOT_DETETA_COTI
TRGA_ADOT_DETETA_OFERD	AFTER STATEMENT	DELETE	ARMADA	ADOT_DETETA_OFER
TRGA_ADOT_DETETA_OFERU	AFTER STATEMENT	INSERT	ARMADA	ADOT_DETETA_OFER
TRGA_ADOT_DETETA_OFERU	AFTER STATEMENT	UPDATE	ARMADA	ADOT_DETETA_OFER
TRGA_ADOT_DETETA_SOLCD	AFTER EACH ROW	DELETE	ARMADA	ADOT_DETETA_SOLC
TRGA_ADOT_DETETA_SOLCI	AFTER EACH ROW	INSERT	ARMADA	ADOT_DETETA_SOLC
TRGA_ADOT_DETETA_SOLCU	AFTER EACH ROW	UPDATE	ARMADA	ADOT_DETETA_SOLC
TRGA_ADOT_FIRM_APROD	AFTER STATEMENT	DELETE	ARMADA	ADOT_FIRM_APRO
TRGA_ADOT_FIRM_APROI	AFTER STATEMENT	INSERT	ARMADA	ADOT_FIRM_APRO
TRGA_ADOT_FIRM_APROU	AFTER STATEMENT	UPDATE	ARMADA	ADOT_FIRM_APRO
TRGA_ADOT_FPAO	AFTER STATEMENT	DELETE	ARMADA	ADOT_FPAO

Figura # 2.9 Triggers migradas

- Ya para la migración de las tablas no hubo diferencias para la migración entre Informix y Oracle, su sintaxis es la misma, y como herramienta utilizamos PowerBuilder. En la figura # se puede observar algunas tablas migradas.

Tabla	Tablespace	Filas	Último Análisis
ADOM_CONC_EVAL	USERS	14	30-mar-2008 08:32:46 AM
ADOM_PORC_SEGU	USERS	0	30-mar-2008 08:32:46 AM
ADOM_PROV	USERS	105	30-mar-2008 08:32:46 AM
ADOM_PROV_DOCU	USERS	35780	30-mar-2008 08:33:33 AM
ADOM_PROV_REFE	USERS	37	30-mar-2008 08:32:47 AM
ADOM_PROV_SERV	USERS	3508	30-mar-2008 08:33:14 AM
ADOM_PROV_TDIS	USERS	3121	30-mar-2008 08:33:22 AM
ADOM_REPR_LEGA	USERS	935	30-mar-2008 08:33:16 AM
ADOM_TIPO_COMI	USERS	4	30-mar-2008 08:32:47 AM
ADOP_DMON	USERS	10	30-mar-2008 08:32:48 AM
ADOP_MONI	USERS	4	30-mar-2008 08:32:48 AM
ADOP_RANG_CALJ	USERS	5721	30-mar-2008 08:33:19 AM
ADOT_CABE_COTI	USERS	6	30-mar-2008 08:32:48 AM
ADOT_CABE_OCOM	USERS	1	30-mar-2008 08:32:49 AM
ADOT_CABE_SOLC	USERS	6	30-mar-2008 08:32:50 AM
ADOT_CONT	USERS	93	30-mar-2008 08:32:16 AM
ADOT_CONT_GARA	USERS	19	30-mar-2008 08:32:50 AM
ADOT_CRIT_BELE	USERS	57	30-mar-2008 08:32:51 AM
ADOT_CUAD_OFER	USERS	43	30-mar-2008 08:32:51 AM
ADOT_DETETA_AFI	USERS	2	30-mar-2008 08:32:52 AM
ADOT_DETETA_COTI	USERS	130	30-mar-2008 08:32:52 AM
ADOT_DETETA_OCOM	USERS	345	30-mar-2008 08:33:17 AM
ADOT_DETETA_OFER	USERS	88	30-mar-2008 08:32:53 AM
ADOT_DOCU_RUBR	USERS	7	30-mar-2008 08:32:54 AM
ADOT_ENCL_CONT	USERS	58659	30-mar-2008 08:34:53 AM
ADOT_FIRM_APRO	USERS	938	30-mar-2008 08:32:54 AM
ADOT_FIRM_SOLC	USERS	7866	30-mar-2008 08:33:27 AM
ADOT_FPAO	USERS	4	30-mar-2008 08:32:55 AM
ADOT_OCOC_DETETA_AFI	USERS	0	30-mar-2008 08:32:56 AM
ADOT_OCOC_FACT	USERS	624	30-mar-2008 08:33:18 AM
AFIM_BICA_HIST	USERS	3340	30-mar-2008 08:33:20 AM
AFIM_ACOF_LOCD	USERS	6	30-mar-2008 08:32:56 AM
AFIM_BIEN_CARJ	USERS	21426	30-mar-2008 08:33:50 AM
AFIM_CABE_MOVI	USERS	0	30-mar-2008 08:32:56 AM
AFIM_CARJ	USERS	4527	30-mar-2008 08:33:21 AM

Figura # 2.10 Tablas migradas

En el [ANEXO D](#) hemos realizado el análisis comparativo de las dos plataformas, donde podemos darnos cuenta la diferencia que existe en la sintaxis, tipo de datos, etc.

CAPITULO 3

VI. REINGENIERÍA

3.1 Modelo lógico y físico del sistema Canopus en INFORMIX

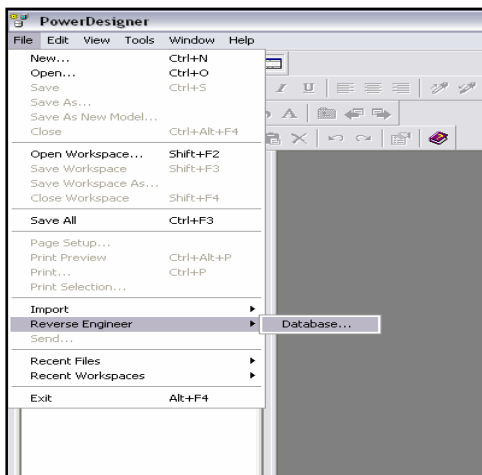
Para poder mostrar el modelo físico lo hicimos con la ayuda de Power Designer como herramienta de desarrollo porque es una herramienta para crear bases de datos y aplicaciones cliente/servidor basadas o no en Web, lo cual permite a los diseñadores de aplicaciones complejas de cliente/servidor tener una descripción general de los procesos particulares para comprender mejor a la organización, así mismo exporta información del modelo físico y extiende atributos al diccionario de 4GL, lo cual no afecta en lo que respecta a los procesos reales. Esta herramienta también proporciona un diseño conceptual de modelo de datos, generación automática de modelo de datos, diseño de normalización física, sistema de manejo de bases de datos múltiples (DBMS) y soporte de herramientas de desarrollo, y elementos de reportes con presentación y calidad.

El diseño se realiza en dos niveles:

- *Nivel conceptual*: entidades, relaciones, dominios, tipos de datos conceptuales, identificadores, y reglas de negocios.
- *Nivel físico*: tablas, columnas, dominios, llaves primarias, llaves foráneas, llaves alternadas, índices, constrains³ de integridad referencial declarativa, vistas, parámetros de almacenamiento físico, reglas de negocios, triggers y procedimientos almacenados.

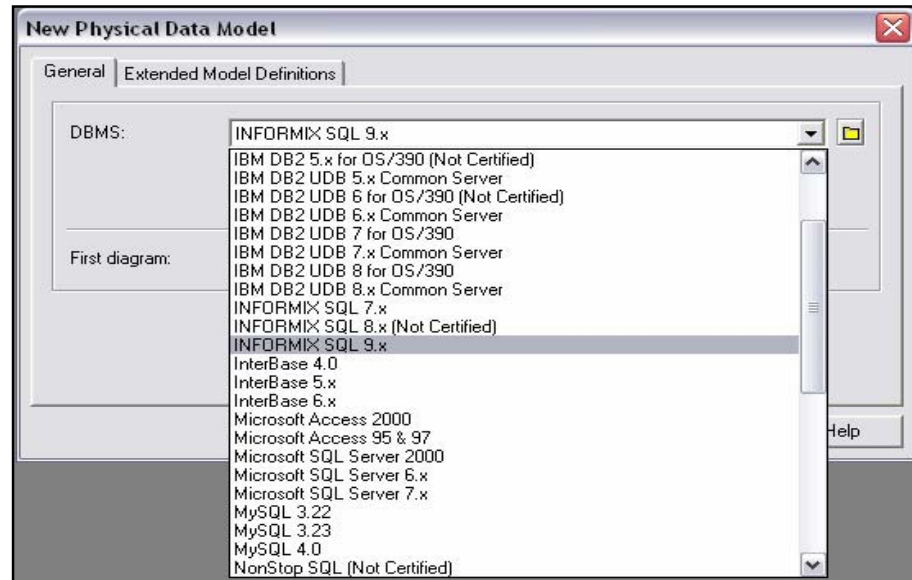
Para poder realizar la Ingeniería reversa y generar el script de cualquier dato que necesitemos, realizamos una reingeniería de la Base de Datos del Sistema, para lo cual hay que seguir los siguientes pasos:

1. En menú-archivo, seleccionar “Ingeniería Reversa” – Database-
enter, para poder ingresar a la configuración del modelo físico.



³ Constrains: Sirve para definir una **restricción** que se podrá eliminar cuando queramos sin tener que borrar la columna

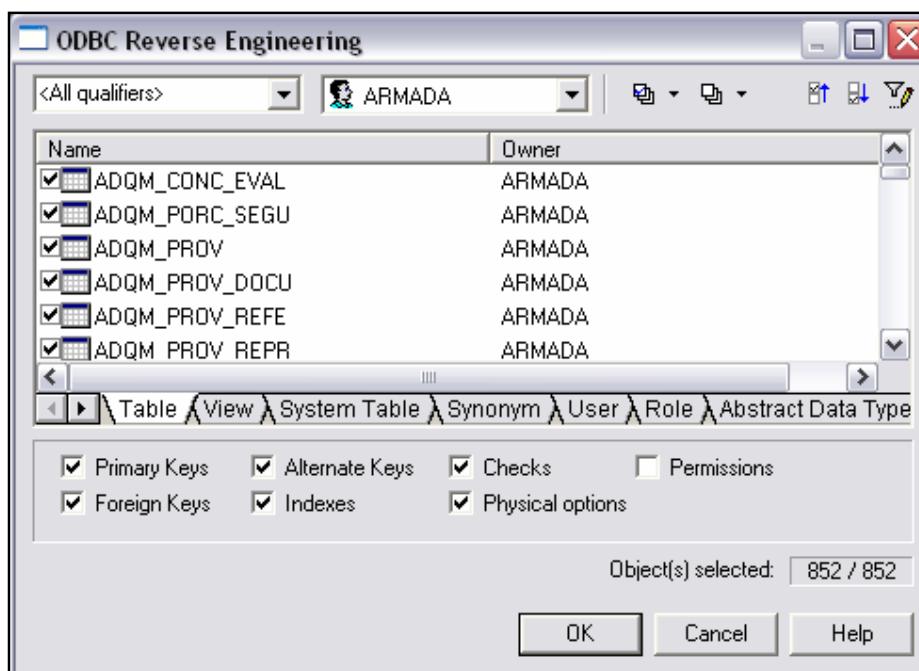
- Al abrirse la ventana seleccionar la Base de Datos en la cual se va a sacar el Script.



- Se selecciona el ODBC para el cual voy a realizar la conexión con la Base de Datos y realizo la respectiva prueba de conexión.



4. Aparecen todas las tablas, vistas, triggers, etc para lo cual se puede seleccionar lo que se quiere realizar la ingeniería reversa.



5. Una vez realizado todos estos pasos la herramienta procederá a realizar la reingeniería lo cual el proceso durará unos minutos. Al final aparecerá el modelo físico del sistema Canopus. Como ejemplo en la figura #3.1 muestra el modelo físico del módulo COMCON (Comité de contrataciones).

Así mismo, en el [ANEXO C](#) se puede ver como está estructurado el sistema CANOPUS, nos podemos dar cuenta que no están bien aplicadas las reglas de normalización (5 formas normales) para la

Mediante el uso correcto de las reglas de normalización estaríamos encaminados a eliminar redundancia e inconsistencias de dependencias en el diseño de las tablas.

3.2 Propuesta del modelo lógico y físico del Sistema Canopus en ORACLE 10g.

Al revisar el modelo lógico de ciertos módulos del Sistema CANOPUS nos hemos dado cuenta que existen ciertas estructuras que no están siendo utilizadas o están mal diseñadas, por lo que se hace una propuesta mejorada del modelo físico únicamente de los siguientes módulos:

- Módulo de Comité de Contrataciones (COMCON)
- Módulo de Sistema de Información Gerencial (SIG)
- Módulo principal distribuidor CANOPUS.

Para realizar un buen diseño de las tablas y para que el sistema sea eficiente y explotar al máximo las ventajas y beneficios que nos brinda Oracle, debemos cumplir las reglas de normalización, verificando como están relacionadas las tablas y corrigiendo todas las redundancias de dependencias.

El utilizar Oracle como Base de Datos podemos tener muchas ventajas cuando usamos las sentencias SQL, lo cual nos ayuda

cuando realizamos algunos procesos al mismo tiempo. Un ejemplo puede ser cuando 2 usuarios están conectados al sistema y proceden a modificar algún dato, Oracle nos brinda la seguridad de poder realizar estas acciones sin la necesidad de desconectar al usuario.

Como se muestra en la figura # 3.2 se puede utilizar bloques de PL/SQL y subprogramas que pueden agrupar sentencias antes de enviarlas a la Base de Datos para ejecutarlas.

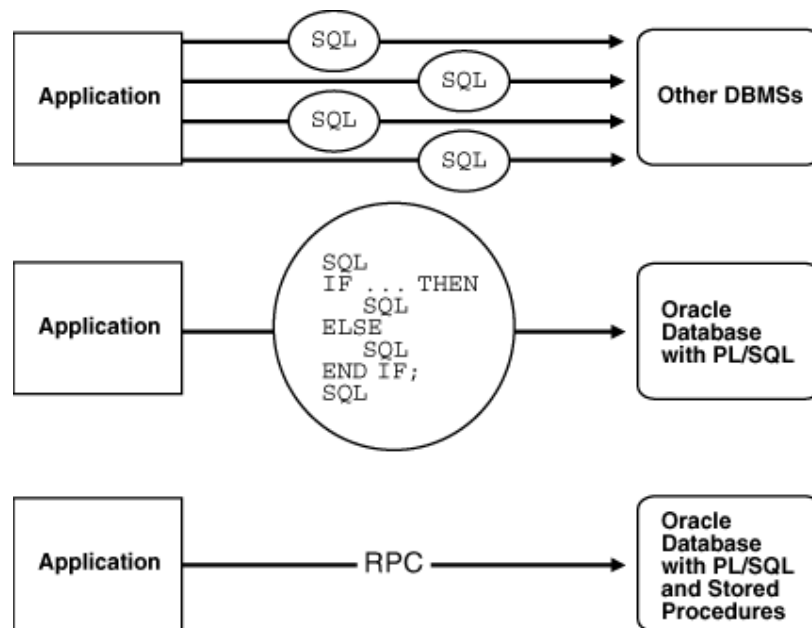


Figura # 3.2 Funcionamiento eficiente Oracle

Los procedimientos almacenados son compilados una vez y almacenados en forma ejecutable, es por esto que las llamadas a procedimientos son eficientes.

Utilizando Oracle como plataforma tendremos:

Alta productividad, ya que al escribir el código es bien compacto al querer manipular los datos.

Full Portabilidad, ya que las aplicaciones de PL/SQL pueden correr sobre cualquier plataforma y sistema operativo donde corra la Base de Datos Oracle.

Soporta programación orientada a objetos, Tipos de Objetos son una herramienta de modelo Orientado a Objeto, en que puedes usarlo para reducir el costo y el tiempo requerido para construir aplicaciones complejas.

3.3 Descripción de entidades susceptibles a mejoras

Durante todo el proceso de migración se realizó una serie de análisis como lo muestra el [ANEXO A](#), lo cual nos dio una mayor claridad de que tablas daban problemas para ser migradas ya sea porque poseía demasiados registros o porque era una tabla mal

configurada, es por esto que al pasarlo a la Base de Datos de ORACLE se fueron corrigiendo todas estas falencias como por ejemplo la tabla "rpl_transaccion" se encontraba registrado en 2 Bases de Datos lo cual perdía eficiencia para realizar las consultas.

CAPITULO 4

4. DESARROLLO DE LA MIGRACIÓN

4.1 Descripción de los pasos para realizar el movimiento de datos.

En un principio utilizamos la herramienta “Migration Workbench” que es la herramienta de Oracle que permite la conversión instantánea de información almacenada, incluyendo triggers y stored procedures, en cualquier versión de Informix a la base de datos de Oracle, lo cual maneja un ambiente integrado y visual y el proceso de migración más simple y rápido, lo cual para ayudar a asegurar la portabilidad, todos los componentes del Oracle Migration Workbench están escritos en Java.

La herramienta genera un reporte sobre el estatus del proceso de migración.

Características de Migration Workbench:

Reporte de Estado						
Soporte	MS SQL Server & Sybase	MS Access	Informix	MySQL	DB2/400	DB2/UDB
Tablas	Si	Si	Si	Si	Si	Si
Vistas	Si	Si (Queries)	Si	N/A	Si	No
Indices	Si	Si	Si	Si	Si	Si
Grupos/Roles	Si	N/A	Si	N/A	No	Si
Usuarios	Si	No	Si	Si	Si	Si
Constraints	Si	Si (Reglas validación)	Si	Si	Si	Si
Privilegios	Si	N	Si	Si	Si	Si
Stored Procedures	Si	N/A	Si	Si	Si	Si

Tabla 4.1 Reporte de estado del proceso de migración

Esta herramienta es muy eficiente pero no funcionó su utilización por motivo que al realizar la migración hubieron tablas que no podían ser migradas y el proceso se bloqueaba, lo cual causó fatiga y pérdida de tiempo, es por esto que utilizamos PowerBuilder por recomendación y por ser una forma más segura de migrar los datos a pesar de ser menos eficiente.

Para la migración de los datos desde en origen (INFORMIX) al destino (ORACLE), como se dijo, utilizamos PowerBuilder como

herramienta de trabajo porque es una herramienta dinámica y flexible que permite realizar traspasos de información de una base de datos a otra por medio de mecanismos llamados pipelines (tubería de datos entre una base de datos y otra), así mismo está orientada a aplicaciones de gestión contra bases de datos. Es una herramienta multiplataforma, cliente/servidor, orientado a objetos y capaz de acceder homogéneamente a cualquier base de datos que soporte SQL.

Durante el proceso se realizó la migración de las tablas, vistas, triggers, stored procedure e índices desde un servidor con plataforma Linux y base de datos Informix a un cliente con plataforma Windows y base de datos Oracle por motivo que se poseía los materiales a disposición pero si es muy recomendable haber tenido LINUX como plataforma por seguridad.

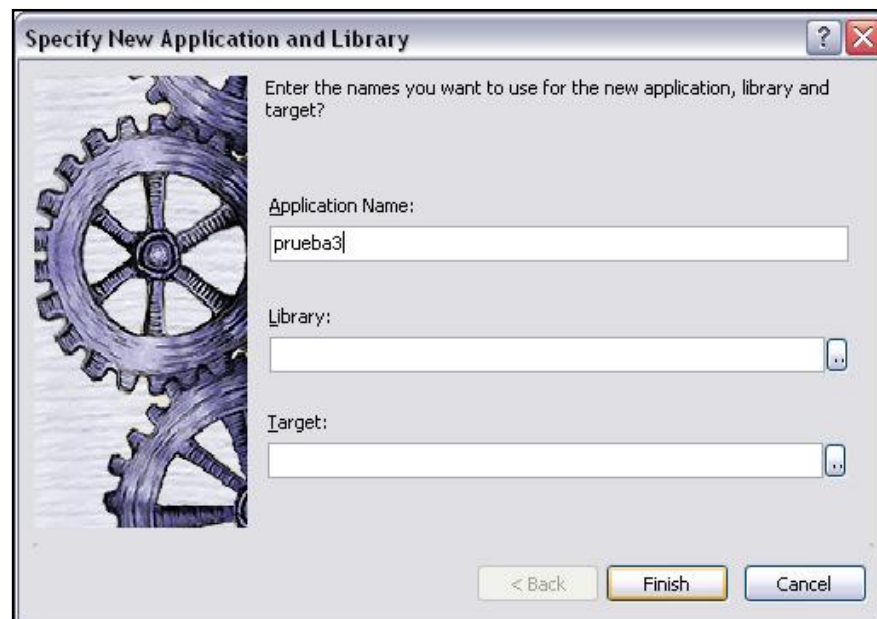
En el caso en que ciertas tablas, vistas, triggers, stored procedure e índices no se hayan logrado migrar correctamente debido a fallas de sintaxis, se utilizó la herramienta de PowerDesigner para poder crear los scripts y posteriormente ejecutarlos ya corregidos en PowerBuilder.

Los pasos para realizar la migración se detallan a continuación:

1.- Se procede a crear un Workspace.



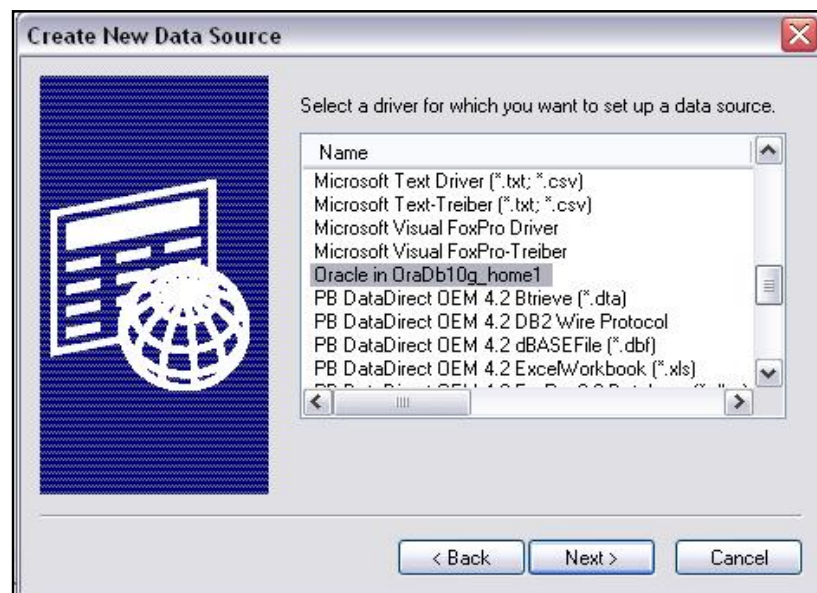
2.- Se crea una nueva aplicación.



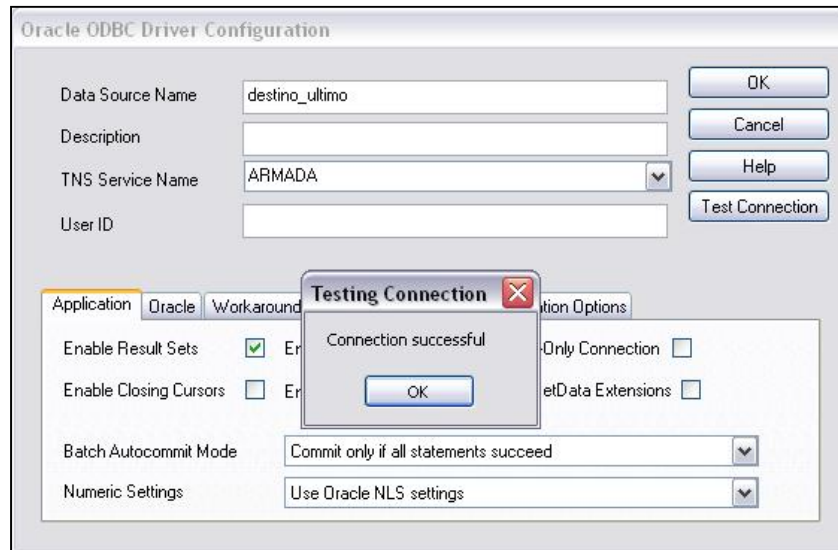
3.- Se crea el ODBC Open Database Connectivity que es un estándar de acceso a la base de datos desarrollado por Microsoft



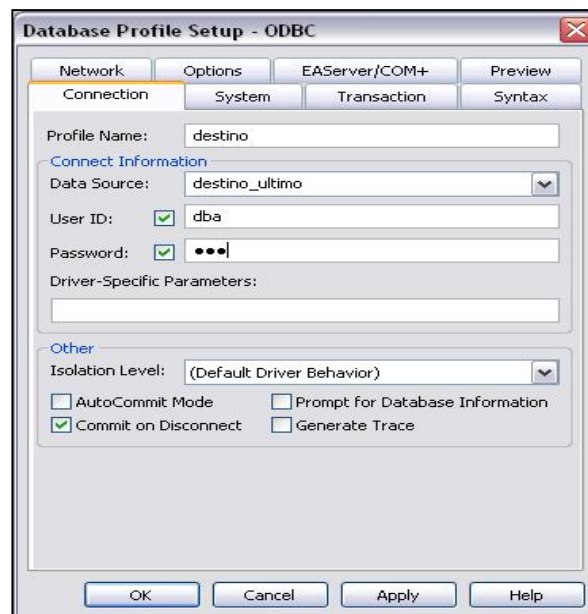
4.- Se selecciona el driver correspondiente.



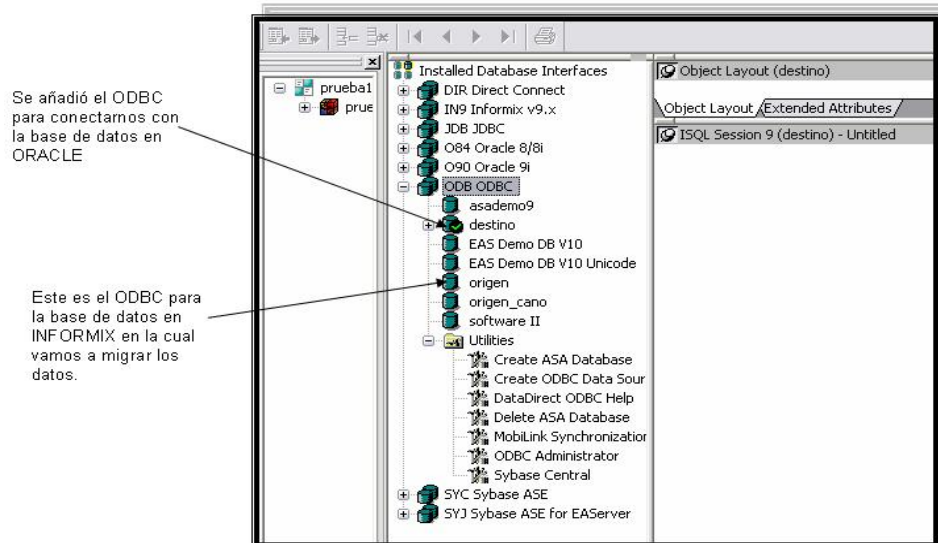
5.- Se configura el driver ODBC y además se hace una prueba de conexión.



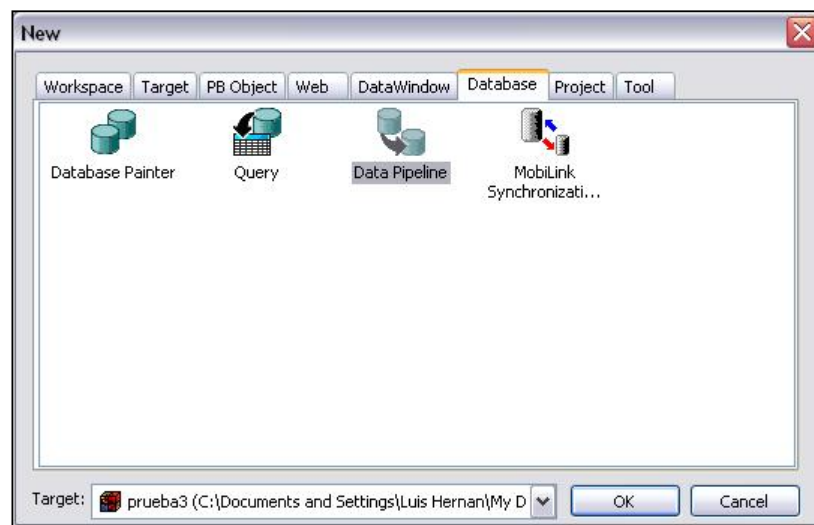
6.- Se configura el perfil del ODBC especificando un ID del usuario y la contraseña.



7.- Se visualiza el ODBC creado.

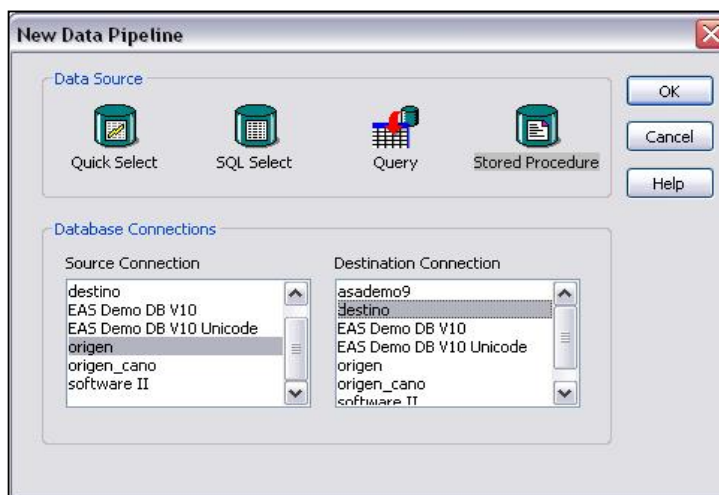


8.- Dentro de Database se selecciona Data Pipeline³ que es para el proceso de migración



³ Pipeline representa una “tubería” de datos entre una base de datos y otra, pudiendo ser estas de cualquier tipo dentro de los soportados por Power Builder. Es muy útil para hacer migraciones de bases de datos, ya que, simplemente indicando el origen y el destino, el proceso nos hará la copia de todos los objetos y datos (o aquellos que seleccionemos), ajustando aquellas sentencias que no sean compatibles en la base de datos destino.

9.- Se seleccionan las bases de datos de origen y de destino.



En la figura # 4.1 se puede visualizar alguna de las tablas que se migraron usando la herramienta Enterprise Management de Oracle 10g.

ORACLE Enterprise Manager 10g Database Control

Database: ARMADA > Tablas

Tablas

Buscar

Seleccione un tipo de objeto y, opcionalmente, introduzca un nombre de esquema y un nombre de objeto para filtrar los datos que aparecerán en el juego de resultados.

Tipo de Objeto: Tabla Esquema: ARMADA Nombre del Objeto:

Resultados

Seleccionar	Esquema	Nombre de la Tabla	Tablespace	Particionado	Filas	Ultimo Analisis
<input type="radio"/>	ARMADA	ADQM_CONC_EVAL	USERS	NO	14	30/03/2006 08:32:46 AM COT
<input type="radio"/>	ARMADA	ADQM_PORC_SEGU	USERS	NO	0	30/03/2006 08:32:46 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV	USERS	NO	105	30/03/2006 08:32:46 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_DOCU	USERS	NO	35780	30/03/2006 08:33:33 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_REFE	USERS	NO	37	30/03/2006 08:32:47 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_REPR	USERS	NO	5628	30/03/2006 08:33:12 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_SERV	USERS	NO	3606	30/03/2006 08:33:14 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_TDIS	USERS	NO	3121	30/03/2006 08:33:22 AM COT
<input type="radio"/>	ARMADA	ADQM_REPR_LEGA	USERS	NO	835	30/03/2006 08:33:16 AM COT
<input type="radio"/>	ARMADA	ADQM_TIPO_COMI	USERS	NO	4	30/03/2006 08:32:47 AM COT
<input type="radio"/>	ARMADA	ADQP_DMON	USERS	NO	10	30/03/2006 08:32:48 AM COT

Figura # 4.1 Visualización de las tablas migradas

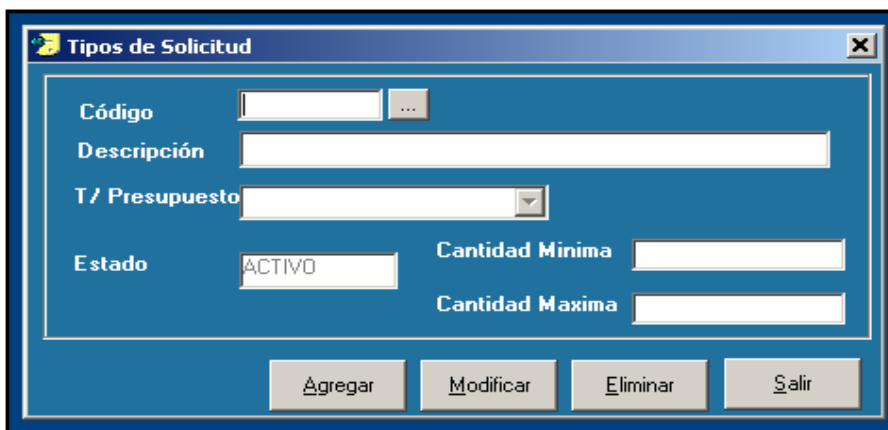
4.2 Rendimiento del Sistema Canopus en la nueva plataforma

4.2.1 Comparación de tiempos y cambios realizados en el Front-end de la aplicación

Modulo Comité de Contrataciones COMCON

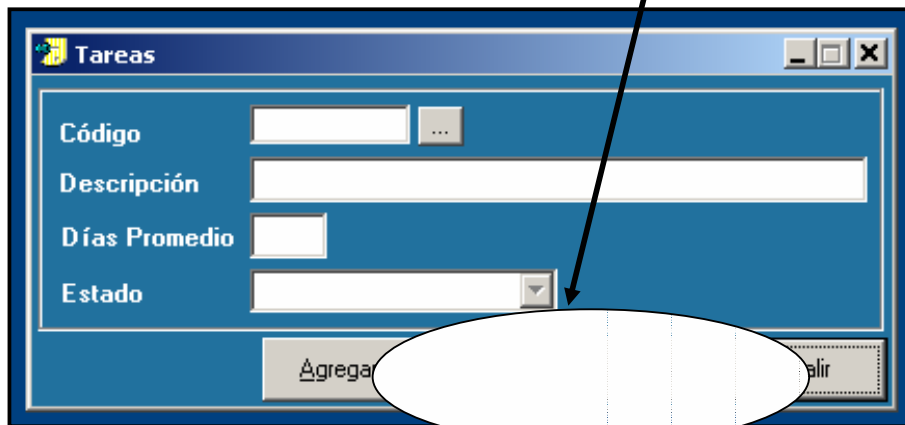
Opción de Mantenimiento:

Tipos de Solicitud:

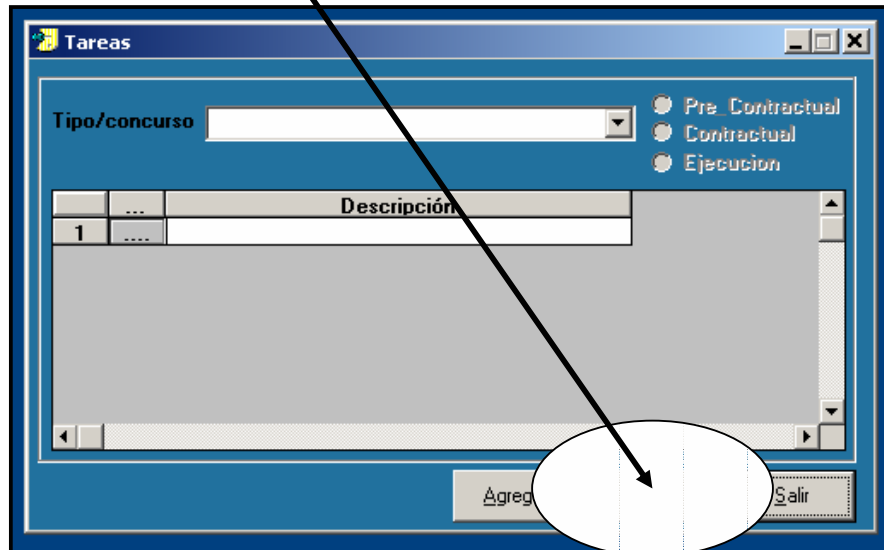


Se procedió a deshabilitar los botones "Modificar" y "Eliminar"

Tipo de Tareas:

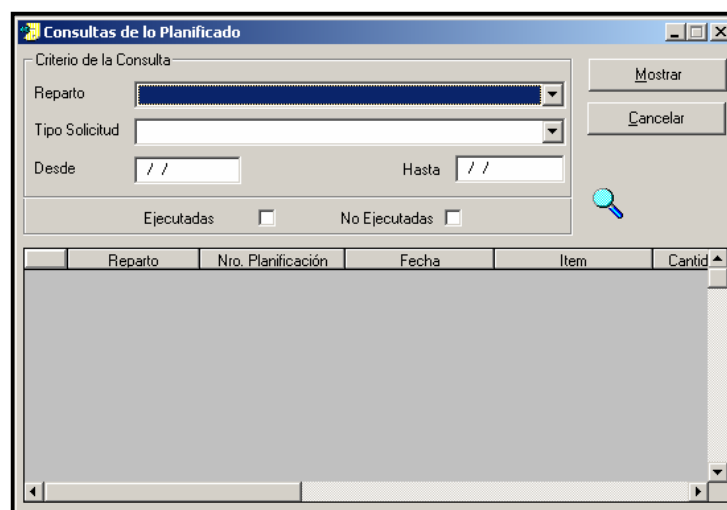


Tareas Por tipo de contrato:



OPCIONES DE CONSULTAS/REPORTES

Planificación



Proveedores

Consulta de Proveedores

Tipo de Material: [dropdown]

Proveedores

	Codigo	Nombre
1		

Todos

Salir

Concursos

Concursos

Desde: // Hasta: //

Agrupamiento por:

- Reparto
- Proveedor
- Estado

Todos

[dropdown]

Filtrado por:

- Año
- Tipo de Material
- Tipo de Solicitud

Año: [input]

Tipo de Material: [dropdown]

Tipo de Solicitud: [dropdown]

Concurso-Permanencia en días

Concursos

Filtrado por:

Año

Estado

Reparto

Año:

Estado:

Reparto:

Toma de Decisiones

Opciones

Datos a Presentar

Estados

ACTA DE ENTREGA Y RECEPCION

ADJUDICADO

AGREGADURIA

ASESORIA JURIDICA

COGMAR

COMACO

COMCON

COMISION TECNICA

Repartos

Sin Año

AVINAV

BASJAR

BASUIL

CETNAV

DIECAR

DIGMAT

DIGMER

Seleccionar Todos

Propiedades Consulta

Por Estados

Por Reparto

Promedio por Año

Tipo Gráfico

Totales

Detallado

Aceptar Cancelar

4.2.2 Comparación de tiempos utilizando consultas particulares

Previo a realizar las mediciones es necesario considerar los siguientes factores:

Problemas de rendimiento de CPU.

Se instaló en el servidor de CANOPUS utilidades para medir el uso de los procesadores, las mismas que se dejaron ejecutando durante toda la jornada laboral que después de tabularlos y resumirlos nos dan los siguientes resultados promedios:

	% USUARIO	% SISTEMA	% LIBRE	INT/S
CANOPUS (2 CPUS)	21.09	2.65	76.26	213.66

Tabla 4.2 Rendimiento del Sistema CANOPUS

Hay que resaltar que estos valores son el promedio desde las 8:00 hasta las 15:40 de la tarde de una jornada de trabajo.

Los valores de % USUARIO (porcentaje de tiempo que el procesador dedica a procesos de usuarios) llegan en el servidor CANOPUS hasta el 99% en muchísimos casos, lo que representa que en muchos momentos de día falta capacidad de procesamiento en el Servidor IBM Netfinity que alberga la base de datos CANOPUS.

Problemas de rendimiento de Discos Duros.

Se instaló en el servidor de CANOPUS utilidades para medir el uso de los discos duros, los mismos que se dejaron ejecutando durante toda la jornada laboral y después de tabularlos y resumirlos nos muestran los siguientes resultados:

ACTIVIDAD DE I/O EN CANOPUS							
					entre discos		
Disco	Tran/s	lecturas	escrituras	total	lectura	escritura	leer+escrit
hdisk0	8,906	16,360	126,136	142,496	3.49%	28.02%	15.51%
hdisk1	112,047	182,682	265,902	448,584	38.97%	59.08%	48.82%
hdisk2	39,897	154,518	5,292	159,810	32.96%	1.18%	17.39%
hdisk3	41,993	115,212	52,760	167,972	24.58%	11.72%	18.28%
		468,772	450,090	918,862			
		51.02%	48.98%				

Tabla 4.3 Actividades de I/O del Sistema CANOPUS

Como se puede apreciar las cargas de trabajo de lectura/escritura en los discos duros del servidor CANOPUS no se encuentran correctamente balanceadas, puesto que un solo disco duro realiza el 48% de las transacciones lo que en horas pico indefectiblemente se convertirá en un cuello de botella para las aplicaciones.

Se puede apreciar también que la base de datos de CANOPUS tiene el 51.02% de lecturas y 48.98% de escrituras.

Adicional a esto las bases de datos CANOPUS se encuentra instalada en configuraciones de disco con Arreglos de Categoría CINCO, las mismas que no son recomendables para bases de datos transaccionales OLTP como lo son las mencionadas.

4.3 Codificación y prueba de aplicaciones

Aquí se procedió a colocar los cambios realizados en la codificación para el front-end, y los datos estadísticos realizados en cada uno de los módulos.

En el [ANEXO B](#) se muestra un análisis que se realizó en algunos módulos del Sistema CANOPUS donde indica que tablas utiliza un

módulo, esto nos ayudó para poder comprobar su rendimiento al realizar las pruebas unitarias.

4.4 Funcionalidades

Oracle tienen funcionalidades en cuanto a respaldos, administración, procesamiento multithreading, clustering, control de bloqueos, procesamiento paralelo de sentencias SQL. Desarrollo en n-capas que hacen de ésta base de datos la mejor opción en cuanto a plataforma de desarrollo de nuevas aplicaciones de alta transaccionalidad sin dejar de lado su excelente desempeño sirviendo a aplicaciones para el Web.

CAPITULO 5

5. IMPACTO FINANCIERO

5.1 Beneficios cualitativos

Se puede destacar los siguientes beneficios a nivel Institucional:

Debido a que la Armada del Ecuador ha creado en los últimos tiempos bases navales, repartos y estaciones en gran parte del país y las islas Galápagos y es fundamental tener un alto nivel de operatividad de las distintas unidades, hace que la demanda de acceso remoto al Sistema Naval Informático Canopus se incremente sustancialmente buscando lograr la automatización de la logística naval, es por eso que la Dirección General del Material necesita en un futuro próximo un socio de tecnología con una plataforma Internet segura y confiable que permita este crecimiento. La plataforma Internet integrada de Oracle permitirá cubrir estas necesidades.

Comparativa de soporte Oracle e Informix:

El servicio de soporte Oracle incluye:

- Asistencia técnica las 24 horas al día 7 días a la semana (24 x 7).
- Acceso a OracleMetaLink – sistema de soporte al cliente basado en el Internet.
- Habilidad para registrar requerimientos a través de OracleMetaLink.
- Actualizaciones de software, versiones de mantenimiento.

Asistencia técnica a los problemas y preguntas de los clientes con relación a los productos oracle.

Se presta a través de su CSI (código de identificación de soporte) desde el centro de soporte en Orlando, al cual se accede por llamada telefónica ó por la conexión con Oracle MetaLink. Este proceso requiere el registro por parte del cliente de las solicitudes de asistencia técnica, la cuales se transfieren directamente y se resuelven por analistas técnicos que tienen acceso a un amplio rango de herramientas de software de diagnóstico. Cada solicitud de asistencia técnica recibe una prioridad establecida por el

analista y el cliente, en función del impacto que cause en el negocio del cliente. Todas las solicitudes son registradas, procesadas, solucionadas y sólo se cierran cuando el analista y cliente así lo acuerden.

Actualizaciones de productos y versiones de mantenimiento.

Oracle tiene el compromiso de la continua investigación y desarrollo de mejoras a sus productos que aseguren a sus clientes cumplir con los continuos retos que enfrentan en su negocio, por medio de las actualizaciones de productos.

Informix ofrece el sitio **PASSPORT ADVANTAGE CUSTOMER** que permite al cliente Informix una serie de servicios como el de bajar software, consultar licencias además de un soporte técnico

5.2 Beneficios cuantitativos

Los beneficios cuantitativos más relevantes son:

1. Existe una mejora de los tiempos empleados para las operaciones de mantenimiento de las bases de datos.

2. Al utilizar Oracle como Base de Datos tenemos mejoras en los procedimientos de respaldo de las bases de datos, logrando de esta manera optimizar este proceso.

3. Al disminuir el tiempo en una de sus actividades, el DBA podrá dedicar mayor cantidad de tiempo a otras labores igualmente importantes; como el mantenimiento de la base de datos, la seguridad de la red, la implementación de otros case de optimización, etc.

5.3 Recursos y costos

5.3.1 Recursos usados para la implementación

- Software de Instalación del Sistema Canopus para cada uno de los usuarios.
- Que las PC's de los usuarios cumplan con las características para la Instalación del Sistema.
- El computador debe tener tarjeta de red para la conexión al sistema.
- Capacitación para los usuarios.

5.3.2 Recursos mínimos necesarios

Como podemos visualizar en la Tabla # 5.1 los recursos mínimos se dan en base a la cantidad de usuarios que posee la red en uso de Sistema CANOPUS.

Repartos	Ubicación	Puntos de Red
Planta Baja	DIRAFI	3
Primer Piso	DIGMAT	19
Segundo Piso	DIRAFI	24
Segundo Piso	CEPROD	18
Segundo Piso	DINDES	22
Tercer Piso	DINCYP	18
Tercer Piso	DINNAV	12
Tercer Piso	MAESTRANZA	5
Tercer Piso	DIECAR	13
Cuarto Piso	DIECAR TALLERES	4
Cuarto Piso	DIRABA	21
Cuarto Piso	CENABS	9

Tabla #5.1 Puntos de red en los repartos

5.3.3 Recursos adicionales requeridos para la administración del sistema

Entre los recursos adicionales podemos mencionar:

- Actualizaciones en línea de las ayudas a la utilización del sistema el cual no se encuentra en funcionamiento por el momento.
- Tener un servidor propio de aplicaciones por motivo de que se encuentra compartido con aplicaciones que no son del sistema.

5.3.4 Costo del software

La adquisición de software y licencia del programa IBM INFORMIX por parte de la Dirección General del Material de la Armada del Ecuador permite el uso autorizado de los siguientes productos:

- IBM INFORMIX y WEBSHERE WORKGROUP BUNDLE PROCESSOR licencia y mantenimiento por doce meses.

Se pondrá a disposición de la DIGMAT la versión, release o la actualización mas actualizada disponible en el mercado. IBM proporcionara asistencia relacionada a la instalación, configuración y al uso del producto.

Además se considera autorizaciones para incrementar el uso de Informix V9.4, actualizaciones de IBM, actualizaciones competitivas y renovaciones de mantenimiento.

Las actualizaciones de IBM contemplan la adquisición de licencias para determinados programas que reemplacen los programas IBM por un coste reducido.

Las actualizaciones competitivas considera la adquisición de licencias de determinados programas que reemplacen programas de otros proveedores por un coste reducido.

Paquete INFORMIX y WEBSHERE WORKGROUP-IDS V9.40 incluye los siguientes medios magnéticos y documentación

Medios magnéticos CY730NA

1 CD IBM Informix Dynamic Server Documentation V9.4

1 CD IBM Informix Dynamic Server Traslated Documentation V9.4

1 CD IBM Informix Dynamic Server Datablade V8.11 Disk 1 of 2

1 CD IBM Informix Dynamic Server Datablade V8.11 Disk 2 of 2

1 CD Informix International Language Supplement V3.40.MC2

1 CD IBM Informix Spatial Datablade V8.20.TC2, 8.20.UC2,
8.20.FC2

Medios Magneticos CY72YNA

1 CD IBM Informix Dynamic Server V9.40.FC7

1 CD IBM Informix Dynamic Server V9.40.UC7

1 CD Informix Dynamic Server V9.40.FC3 for LINUX ITANIUM

1 CD Informix Dynamic Server V9.40.UC3 for IRIX

1 CD Informix Dynamic Server V9.40.FC7 for IRIX

1 CD Informix Dynamic Server V9.40.TC7 for WINDOWS

1 CD Informix Dynamic Server V9.40.FC7 for FUJITSU/SIEMENS

1 CD Advanced Global Systems. LTD.SERVER STUDIO JE V5.1
for JAVA

Documentación:

- IBM Informix Datablade Module Installation and Registration
Guide V4.0A
- IBM Informix Dynamic Server Installation Guide for UNIX and
LINUX V9.4

- IBM Informix Dynamic Server Installation Guide for MICROSOFT WINDOWS V9.4
- IBM Informix Dynamic Server Client Products Installation Guide V2.81
- IBM Informix Dynamic Server Getting Started Guide V9.4

La compañía SINERGY TEAM CIA. LTDA., distribuidor de IBM, suscribió un contrato con la Dirección General del Material con el detalle mostrada en la Tabla # 5.2.

Código	Descripción	Cantidad	V. Unitario	Valor Total
D53EALL	IBM INFORMIX and WEBSHERE WORKGROUP Bundle Licencia y mantenimiento de SW 12 meses.	2	3044.05	6088.10
BBOJ2NA	INFORMIX/WEBSHERE WORKGROUP-IDS V9.40 WIN2K, DIGI UNIX, LINUX, IRIX MP Instalación y configuración del producto IBM Informix adquirido.	1	150.00	150.00
		1	300.00	300.00

Tabla 5.2 Detalle de contrato con distribuidor de IBM

El costo total es de SEIS MIL QUINIENTOS TREINTA Y OCHO 10/100 dólares (USD. \$ 6538,10) aplicado a la partida presupuestaria "Adquisición de Software y Licencia".

Oracle mantiene una lista de precios obtenida de la página de Oracle www.oracle.com en la sección “Global Price List Oracle” la cual se encuentra mostrada en la tabla # 5.3:

Producto	Usuario Nombrado	Update, Licencia y Soporte	Licencia por procesador	Update, Licencia y Soporte
Base de datos				
Standard Edition One	149.00	32.78	4995	1098.90
Standard Edition	300.00	66.00	15000	3300.00
Enterprise Edition	800.00	176.00	40000	8800.00
Opciones de Enterprise Edition				
Real application Cluster	400	88.00	20000	4400.00
Partitioning	200	44.00	10000	2200.00
OLAP	400	88.00	20000	4400.00
Data Mining	400	88.00	20000	4400.00
Spatial	200	44.00	10000	2200.00
Advanced Security	200	44.00	10000	2200.00
Label Security	200	44.00	10000	2200.00
Aplicaciones para Mantenimiento				
Diagnostic Pack	60	13.20	3000	660.00
Tuning Pack	60	13.20	3000	660.00
Change Management Pack	60	13.20	3000	660.00
Configuration Management Pack	60	13.20	3000	660.00
Data WareHouse				
Express Server	800	176.00	40000	8800.00
Express Analyzer	800	176.00	n/a	n/a
Express Objects	5000	1100.00	n/a	n/a
Intenet Application Server				
Standard Edition One	149	32.78	4995	1098.90
Standard Edition	200	44.00	10000	2200.00
Enterprise Edition	600	132.00	30000	6600.00
Business Intelligence	400	88.00	20000	4400.00

Tabla # 5.3 Lista de Precios Oracle

Dentro de los productos más vendidos de Oracle, tenemos los relacionados con la base de datos y son los siguientes:

- Oracle Database Standar Edition One
- Oracle Database Standar Edition
- Oracle Database Enterprice Edition

Dentro de las necesidades de la Armada estaría considerado el Oracle Database Enterprice Edition.

Oracle Database Enterprice Edition.- Orientado a empresas grandes o que requieren características especiales, puede ser configurado sin restricción de procesadores, y su costo es de \$40000 por procesador o de \$800 por usuario (mínimo 25 usuarios por procesador instalado).

Los clientes de Oracle Database Enterprice Edition encuentran muy útil licenciar de manera adicional opciones como:

- Real Application Cluster
- Partitioning
- Diagnostic Pack
- Tuning Pack
- Change Management Pack

- Configuration Management Pack

Real Application Cluster. - Su finalidad es poder tener múltiples servidores soportando a la misma base de datos, tiene un costo de \$20000 por procesador, o de \$400 por usuario nombrado.

Obviamente el número de procesadores o usuarios a licenciar debe ser igual al número de procesadores o usuarios licenciados para Oracle Enterprise Edition.

Partitioning.- Una opción que permite administrar las tablas grandes de la base de datos en diferentes tablespaces, para ayudar al rendimiento de la base de datos. Su costo es de \$10000 por procesador o \$200 por usuario nombrado.

Oracle Packs.- Diagnostic, Tuning, Change Management, Configuration Management son características adicionales que permiten facilitar la administración del motor de base de datos Oracle, así como potenciar su rendimiento. El costo de cada uno de estos 4 paquetes es de \$3000 por procesador o de \$60 por cada usuario nombrado.

Otro de los productos requeridos es el Internet Application Server:

Internet Application Server Enterprise Edition. - Para empresas grandes, no tiene límite de escalabilidad a nivel de CPUs, y tiene un costo de \$30000 por CPU o de \$600 por usuario nombrado.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Después de haber utilizado “Workbench” en el proceso de migración determinamos que esta herramienta no fue eficiente debido a que el Sistema CANOPUS cuenta con muchos registros, lo que produjo que se bloquee y que no se migre al 100%, es por esto que utilizamos PowerBuilder que no es eficiente pero si es eficaz.
2. La delimitación de los módulos a migrar, la definición de la estrategia a seguir, la identificación de las herramientas a utilizar, la determinación de la viabilidad técnica y la factibilidad económica, permitió realizar una migración completa y exitosa.
3. La realización de pruebas de rendimiento nos permitió comprobar que las aplicaciones que hacen uso de la base de datos Oracle 10g funcionan correctamente, logrando la optimización de los tiempos y recursos.

4. Es importante para el éxito del proyecto contar con el apoyo de la alta dirección para poder enfrentar la resistencia al cambio especialmente por parte de los desarrolladores y administradores de la base de datos.
5. Factores como costos, eficiencia, integridad, seguridad, velocidad de respuesta y cantidad de usuarios, determinó que el motor de Base de Datos sea Oracle 10g por ser la mejor en adaptarse a los nuevos requerimientos de la organización.
6. El reconocimiento de la tecnología con la que cuenta el Sistema CANOPUS nos permitió determinar que ciertas estaciones de trabajo del usuario inciden en su desempeño al no cumplir con los requerimientos mínimos indicados.
7. Los nuevos requerimientos de la Institución para el Sistema CANOPUS hacen necesario una revisión completa de su arquitectura, lo que permitirá mantener una estructura sólida dado los cambios de tecnología.

RECOMENDACIONES

1. Para migraciones futuras se podría considerar desarrollar una propia aplicación que migre cualquier Base de Datos a Oracle, en especial que se pueda migrar cualquier base de datos que se encuentre en Informix, ya que Oracle utiliza una sintaxis PL/SQL distinta a los demás motores.
2. Realizar la creación de todos los scripts de los componentes de la Base de Datos (tablas, índices, vistas, disparadores, etc.) a ser migrada, en este caso crear todos los scripts de los componentes de la Base de Datos que se encuentra en Informix; para luego llevar a cabo una migración ordenada y segura, esto es cuando directamente con la herramienta PowerBuilder no se puede migrar.
3. Para versiones futuras se podría considerar la sobrescritura completa del Sistema CANOPUS en cuanto al código fuente por lo que existen funciones que no se utilizan, además que el código posea una documentación interna para un mejor entendimiento de la misma.

4. Se debería utilizar tecnología ADO en todos los módulos del sistema para mejorar el rendimiento en la conexión con la Base de Datos.

5. Mejorar la interfaz del usuario en el Sistema Canopus, de una manera que refleje que el usuario se sienta cómodo en su uso, con opciones claras y eficientes.

BIBLIOGRAFIA

1. El Rinconcito de Delphi, ORACLE PL/SQL tratamiento personalizado de Excepciones.
2. Donald K. Burleson, (Paperback - Jul 27, 2001), Oracle High-Performance SQL Tuning, Standford, California
3. Kevin L., Bob B. (2006), Oracle Database 10g: The Complete Reference (Osborne ORACLE Press Series), Standford: California.
4. Ryan S. and Ronald P. (2005), URL: [www.amazon.com/Database-Design-Sams-Teach-Yourself/ dp/ 0672317583/ sr % 3d8-42/ qid% 3d1159806015 / ref % 3dsr_1_42 / 102-0481889 - 0901714?ie %3dUTF8%26amp;s%3dbooks%27%27](http://www.amazon.com/Database-Design-Sams-Teach-Yourself/dp/0672317583/sr%3D8-42/qid%3D1159806015/ref%3Dsr_1_42/102-0481889-0901714?ie=UTF8&s%3Dbooks%27%27) Database Design (Sams Tech Yourself S.).
5. Sanchez, J. (2003), Manual de SQL para Oracle, Standford, Califórnia.
6. URL: [http:// es.wikipedia.org/ wiki/ Comparaci%C3%B3n _de_sistemas_administradores_de_bases_de_datos_relacionales](http://es.wikipedia.org/wiki/Comparaci%C3%B3n_de_sistemas_administradores_de_bases_de_datos_relacionales)
7. URL: [Http://www.redcientifica.com/oracle](http://www.redcientifica.com/oracle)

8. Sanchez, J. (2003), Manual de SQL para Oracle, Standford, Califórnia,
9. URL: Ont.oracle.com
10. URL: www.monografias.com, Manual de Informix.
11. URL: www.monografias.com, Manual de Oracle.
12. URL: www.oracle.com, Global Price List Oracle 6 Enero 2006
13. URL: www.mysql.com/ products/ tools/ migration - toolkit/ tutorials/ OracleMigrationTutorial.html, MySQL migration tool.
14. URL: www.databasejournal.com/feature/oracle/index/php, Steve Callan, "Oracle Migration Workbench", Octubre 24 del 2004.
15. www.zonaoracle.com

ANEXOS

ANEXO A

Diagnóstico de Migración de tablas

ANEXO B

Tablas Migradas

ANEXO C

Modelo físico del Sistema CANOPUS

ANEXO D

DDL en Informix y en Oracle

ANEXO E

Manual de Migración

ANEXOS

ANEXO A

MÓDULO DE ADQUISICIÓN

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
Control de Obras (ADQ)	ADQMD_001	x	x			x
Control de Obras (ADQ)		adqt_fpag				x
Control de Obras (ADQ)		adqt_deta_ocom	x			
Control de Obras (ADQ)		adqt_cabe_solc				x
Control de Obras (ADQ)		adqt_cuad_ofer				x
Control de Obras (ADQ)		adqt_deta_ofer				x
Control de Obras (ADQ)		adqp_rang_cali				x
Control de Obras (ADQ)		adqm_prov				x
Control de Obras (ADQ)		adqm_tipo_comi				x
Importaciones (IMP)		impt_deta_part_pre				x
Importaciones (IMP)		impm_dicc				x
Importaciones (IMP)		impt_enlc_docu			x	
Importaciones (IMP)		impt_tipo_camb				x
Planificación Presupuestaria (PRE)		prem_part_proy				x
Contabilidad General (CGL)		Cglm_Grup				x
Contabilidad General (CGL)		Cglm_subs_Grup				x
Contabilidad General (CGL)		Cglm_mayo_gene				x
Contabilidad General (CGL)		Cglm_plan_ctas				x
Contabilidad General (CGL)		cglm_masc				x
Contabilidad General (CGL)		Cglm_subs_Grup				x
Recursos Humanos (RHU)		rhum_carg				x
Recursos Humanos (RHU)		rhut_carg_pers				x
Recursos Humanos (RHU)		rhum_pers				x
Recursos Humanos (RHU)		rhum_desc_carg				x
Recursos Humanos (RHU)		rhum_espe				x
Recursos Humanos (RHU)		rhum_grad				x
Cuentas por Cobrar (SEG)		segt_mont				x
Seguridades (SEG)		segp_prmt				x
Cuentas por Cobrar (SEG)		segt_sesi	x		x	
Cuentas por Cobrar (SEG)		segm_apli				x
Importaciones (IMP)	ADQMD_002	impm_mone				x
Importaciones (IMP)		impm_dicc				x

Importaciones (IMP)		impm_cabe_ntdp				X
Importaciones (IMP)		impt_delta_liqu				X
Importaciones (IMP)		impm_pais				X
Importaciones (IMP)		impt_cabe_liqu				X
Planificación Presupuestaria (PRE)		prem_rela_acti				X
Planificación Presupuestaria (PRE)		prem_prgr_rela				X
Planificación Presupuestaria (PRE)		prem_rela_repu				X
Planificación Presupuestaria (PRE)		prem_rela_acti				X
Planificación Presupuestaria (PRE)		prem_prgr_rela				X
Contabilidad General (CGL)		cgl_t_deta_cont				X
Contabilidad General (CGL)		cgl_t_cabe_cont				X
Contabilidad General (CGL)		cgl_t_deta_inte				X
Contabilidad General (CGL)		cgl_t_cabe_inte				X
Contabilidad General (CGL)		cglm_grup				X
Contabilidad General (CGL)		cglm_subs_grup				X
Contabilidad General (CGL)		Cglt_docu_firm				X
Control de Inventario de Materiales (INM)		inmm_item				X
Recursos Humanos (RHU)		rhum_pers				X
Aquisiciones (CXP)		cxpt_deta_cons	X	X		X
Seguridades (SEG)		segp_prmt		X		X
	ADQMD_003					
Seguridad SEG		segp_prmt				X
	ADQMD_004					
Control de Obras (ADQ)		ADQT_CABE_SOLC				X
Control de Obras (ADQ)		ADQT_CABE_OCOM				X
Control de Obras (ADQ)		adqm_repr_lega				X
Control de Obras (ADQ)		adqm_prov_repr				X
Control de Obras (ADQ)		ADQT_CONT				X
Control de Obras (ADQ)		ADQT_CUAD_OFER				X
Control de Obras (ADQ)		adqm_prov				X
Control de Obras (ADQ)		adqm_tipo_comi				X
Importaciones (IMP)		IMPM_DICC				X
Contabilidad General (CGL)		cgl_t_cred				X
Contabilidad General (CGL)		CGLM_KARD				X
Comunicaciones Administrativas (PLA)		plah_obra				X

Control de Inventario de Materiales (INM)		inmm_item				x
		INVM_REPU				x
		INVM_DENO				x
		AFIM_DENO				x
		invn_tipo_item				x
		mccg				x
		mid				x
		linmarc				x
		mcxc				x
		dic				x
		plant				x
		manual				x
		minv				x
		dinv				x
		ref				x
		linea				x
		GLINEA				x
		LINmarc				x
	ADQMD_005					
Recursos Humanos		rhum_pers				x
		DICC_CODI				x
Seguridad		Segm_User				x
Control de Obras		ADQT_CABE_SOLC				x
Control de Obras		adqt_cabe_solc				x
	ADQMD_006					
Adquisiciones		cxpt_deta_cons				x
		IMPM_DICC				x

MÓDULO DE PLANIFICACIÓN

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
Comunicaciones Administrativas PLA	PLAMD_0004					
Seguridad		Segm_User				X
	PLAMD_0001					
Comunicaciones Administrativas PLA		Plah_Obra				X
		Plam_Rubr				X
		plap_secu_obra	X	X		X
Planificacion Presupuestaria PRE		Pret_salm_mond				X
Contabilidad General CGL		Cglt_Cred				X
		cglt_docu_apli				X
Control de Inventarios de Materiales INM		INMM_REPA		X		X
Recursos Humanos RHU		rhum_locd				X
		Rhum_Desc				X
		rhum_carg				X
		rhut_carg_pers				X
		rhum_pers				X
		rhum_desc_carg				X
		Rhum_Grad				X
		Rhut_Deta_Ufin				X
		Rhut_Cabe_Ufin				X
Importaciones IMP		Impt_Deta_Part_Pre				X
Seguridad SEG		Segm_User				X
		segp_prmt				X
		Segm_Comp				X
		SEGT_SESI	X		X	
	PLAMD_0002					
Planificacion Presupuestaria PRE		PRET_TRAN_COGA	X	X		X
		Prem_Deta_Solc				X
		pret_salm_mond				X
Comunicaciones administrativas PLA		Plah_Orde_Trab				X
		Plat_PLan_Cab				X
		PLah_Obra				X

Seguridad		segp_prmt		X		X
Importaciones IMP		Impt_Deta_Part_Pr				X
		impm_stad				X
		impm_ciud				X
		Impt_Deta_Part_Pre				X
Control de Inventario de Materiales INM		INMT_CABE_PEDI				X
Control de Obras ADQ		ADQT_CABE_OCOM				X
		ADQT_CABE_SOLC				X
Recursos Humanos RHU		rhum_locd				X
		Rhum_Desc				X
		rhut_carg_pers				X
		rhum_pers				X
		Rhum_Grad				X
Adquisiciones CXP		Cxpm_ctas_pagr				X
??????????		INVT_DMOV				X
??????????		T_AudiIndi	X	X		
??????????		Vw_Pres				X
??????????		Invm_Unid				X
	PLMD_0003					
Adquisiciones CXP		cxpt_deta_cons	X	X		X
Importaciones IMP		Impt_Deta_Part_Pre				X
Planificacion Presupuestaria PRE		pret_tran_coga				X
Contabilidad General CGL		cgl_t_deta_cont				X
		cgl_t_cabe_cont				X
		CGLM_GRUP				X
		CGLM_SUBS_GRUP				X

MÓDULO DE CONTABILIDAD

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
	Modulo aplica					
Contabilidad General (CGL)		CGLT_CABE_INTE	X			
		CGLT_DETA_INTE	X			
		cglm_masc				X
		CGLM_CABE_ASIE				X
Importaciones (IMP)		IMPT_ENLC_CONT	X			
Seguridad (SEG)		segp_prmt		X		X
ADQUISICIONES (CXP)		cxpm_ctas_pagr		X		
Importaciones (IMP)		impm_mone				X
		IMPT_DETA_PART				X
		IMPM_CABE_NTPD				X
		IMPM_CABE_LIQU				X
		IMPT_DETA_LIQU				X
		impt_enlc_docu				X
		impm_codi_impo				X
		impm_dicc				X
		IMPT_GAST				X
		IMPT_TIPO_CAMB				X
Recursos Humanos (RHU)		rhum_locd				X
		rhum_desc				X
		RHUM_ESPE				X
		RHUM_GRAD				X
		RHUM_CARG				X
		RHUT_CARG_PERS				X
No se sabe a que modulo pertenece		repm_soli_trab				X
		invn_repu				X
		invn_deno				X
		rept_requ				X
Control de Inventario de Materiales (INM)		inmm_item				X

		inmt_cabe_pedi				X
Planificacion Presupuestara (PRE)		Prem_Relacion_Plpa				X
		pret_tran_coga				X
Adquisiciones (CXP)		cxpm_ctas_pagr				X
Control de Obras (ADQ)		ADQM_PROV				X

MÓDULO DE EJECUCIÓN PRESUPUESTARIA

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
	PLAMD_001					
Control de Obras		Plah_Obra				x
Contabilidad General		Cglt_Cred				x
Control de Obras		Plam_Rubr				x
		repo_mate				x
Recursos Humanos		rhum_locd				x
Recursos Humanos		Rhum_Desc				x
Recursos Humanos		rhum_carg				x
Recursos Humanos		rhut_carg_pers				x
Recursos Humanos		rhum_pers				x
Recursos Humanos		rhum_desc_carg				x
Recursos Humanos		Rhum_Grad				x
Seguridad		Segm_User				x
Contabilidad General		cglt_docu_apli				x
Control de Obras		Plap_Secu_Obra	x	x		x
Recursos Humanos		Rhut_Deta_Ufin				x
Recursos Humanos		Rhut_Cabe_Ufin				x
Recursos Humanos		rhum_locd				x
Seguridad		segp_prmt				x
Control de Obras		plah_obra				x
Control de Inventarios de Materiales		INMM_REPA		x		x
Seguridad		Segm_Comp				x
Importaciones		Impt_Deta_Part_Pre				x
Planificacion Presupuestaria		Pret_salm_mond				x
Seguridad		SEGT_SESI	x		x	
	PLAMD_002					
Importaciones		Impt_Deta_Part_Pre				x
Planificacion Presupuestaria		Prem_Deta_Solc				x
Control de Obras		Plah_Orde_Trab				x
Control de Obras		Plat_PLan_Cab				x
Control de Inventarios de Materiales		INMT_CABE_PEDI				x
Adquisiciones		ADQT_CABE_OCOM				x
Adquisiciones		ADQT_CABE_SOLC				x

Recursos Humanos		rhum_locd				x
Recursos Humanos		Rhum_Desc				x
Recursos Humanos		rhut_carg_pers				x
Recursos Humanos		rhum_pers				x
Recursos Humanos		Rhum_Grad				x
Control de Obras		PLah_Obra				x
Cuentas por pagar		Cxpm_ctas_pagr				x
Planificacion Presupuestaria		pret_salm_mond				x
Seguridad		Segp_prmt		x		x
Importaciones		Impt_Deta_Part_Pre				x
Planificacion Presupuestaria		Pret_tran_coga				x
Importaciones		impm_stad				
Importaciones		impm_ciud				
Control de Inventarios de Repuestos		INVT_DMOV				
		PRET_TRAN_COGA	x	x		
	PLAMD_003					
Cuentas por pagar		cxpt_deta_cons	x	x		x
Importaciones IMP		Impt_Deta_Part_Pre				x
Planificacion Presupuestaria PRE		pret_tran_coga				x
Contabilidad General		cgltdeta_cont				x
Contabilidad General		cgltcabe_cont				x
Contabilidad General		CGLM_GRUP				x
Contabilidad General		CGLM_SUBS_GRUP				x

MÓDULO DE CONTROL DE OBRAS

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
<i>PLAMD_001</i>						
Control de Obras		Plah_Obra				x
Contabilidad General		Cglt_Cred				x
Control de Obras		Plam_Rubr				x
		repo_mate				x
Recursos Humanos		rhum_locd				x
Recursos Humanos		Rhum_Desc				x
Recursos Humanos		rhum_carg				x
Recursos Humanos		rhut_carg_pers				x
Recursos Humanos		rhum_pers				x
Recursos Humanos		rhum_desc_carg				x
Recursos Humanos		Rhum_Grad				x
Seguridad		Segm_User				x
Contabilidad General		cglt_docu_apli				x
Control de Obras		Plap_Secu_Obra	x	x		x
Recursos Humanos		Rhut_Deta_Ufin				x
Recursos Humanos		Rhut_Cabe_Ufin				x
Recursos Humanos		rhum_locd				x
Seguridad		segp_prmt				x
Control de Obras		plah_obra				x
Control de Inventarios de Materiales		INMM_REPA		x		x
Seguridad		Segm_Comp				x
Importaciones		Impt_Deta_Part_Pre				x
Planificacion Presupuestaria		Pret_salm_mond				x
Seguridad		SEGT_SESI	x		x	
<i>PLAMD_002</i>						
Importaciones		Impt_Deta_Part_Pre				x
Planificacion Presupuestaria		Prem_Deta_Solc				x
Control de Obras		Plah_Orde_Trab				x
Control de Obras		Plat_PLan_Cab				x
Control de Inventarios de Materiales		INMT_CABE_PEDI				x
Adquisiciones		ADQT_CABE_OCOM				x
Adquisiciones		ADQT_CABE_SOLC				x

Recursos Humanos		rhum_locd			X
Recursos Humanos		Rhum_Desc			X
Recursos Humanos		rhut_carg_pers			X
Recursos Humanos		rhum_pers			X
Recursos Humanos		Rhum_Grad			X
Control de Obras		PLah_Obra			X
Cuentas por pagar		Cxpm_ctas_pagr			X
Planificacion Presupuestaria		pret_salm_mond			X
Seguridad		Segp_prmt		X	X
Importaciones		Impt_Deta_Part_Pre			X
Planificacion Presupuestaria		Pret_tran_coga			X
Importaciones		impm_stad			
Importaciones		impm_ciud			
Control de Inventarios de Repuestos		INVT_DMOV			
		PRET_TRAN_COGA	X	X	
	PLAMD_003				
Cuentas por pagar		cxpt_deta_cons	X	X	X
Importaciones IMP		Impt_Deta_Part_Pre			X
Planificacion Presupuestaria PRE		pret_tran_coga			X
Contabilidad General		cgltdeta_cont			X
Contabilidad General		cgltcabe_cont			X
Contabilidad General		CGLM_GRUP			X
Contabilidad General		CGLM_SUBS_GRUP			X

MÓDULO DE FACTURACIÓN

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
	FACMOD_001					
xxx	xxx	xxx				
	FACMOD_002					
Contabilidad General		Cglt_Docu_firm				x
Facturacion		Facm_Tipo_Kard				x
Control de Inventario de Material		Inmm_Espe_Vehi				x
Control de Inventario de Material		Inmm_Tipo_Vehi				x
Inventario		Invt_Cmov				x
Recursos Humanos		rhum_carg				x
Recursos Humanos		rhut_carg_pers				x
Importaciones		impt_tipo_camb				x
Planificacion Presupuestaria		Pret_tran_Coga		x		x
Planificacion Presupuestaria		Pret_tran_Coga				x
Importaciones		IMPT_TIPO_CAMB				x
Inventario		Invm_Bode				x
Recursos Humanos		Rhut_Cabe_Ufin				x
Recursos Humanos		Rhut_Deta_Ufin				x
Control de Inventario de Material		inmt_bode_item				x
Inventario		invm_tipo_item				x
Recursos Humanos		Rhum_Locd				x
Recursos Humanos		Rhum_Desc where				x
Contabilidad General		CGLM_CABE_ASIE				x
Contabilidad General		CGLM_DETA_ASIE				x
Contabilidad General		CGLM_MASC				x
Seguridad		segp_prmt		x		x
Control de Inventario de Material		inmm_docu		x		x
Recursos Humanos		Rhum_Tipo_Locd				x
Planificacion Presupuestaria		Prem_Fond_Part				x
Contabilidad General		CGLT_CABE_INTE	x			
Contabilidad General		CGLT_DETA_INTE	x			
Seguridad		segt_Sesi	x		x	
	FACMOD_005					
Inventario		INVT_DMOV				x

Inventario		INVT_CMOV				X
	Mod_Localidades					X
Recursos Humanos		rhum_locd				X
Recursos Humanos		rhum_desc				X
	Modulo_Mike					X
Seguridad		segp_prmt				X

MÓDULO DE COMUNICACIONES

Módulos	Submódulos	Tablas	Insert	Update	Delete	Select
Consultas Gerenciales CAR	<i>Modulo1</i>	CART_HIST_MENS				x
		cart_bita	x	x		x
		carm_prio				x
		carm_para	x	x		x
		carm_estd_mesg				x
		carm_bita_user		x		x
		cart_anex				x
		carm_cuer	x		x	
		carm_mens		x		
Recursos Humanos RHU		rhum_locd				x
		rhum_desc				x
		rhum_tipo_auto				x
Importaciones IMP		impm_ciud				x
Seguridades SEG		segp_prmt				x
	<i>ModImpr.Bib</i>					
Consultas Gerenciales CAR	<i>ModBib</i>	carm_Deno_Arch				x
		cart_calf_arch				x
Recursos Humanos RHU		Rhut_Carg_Pers				x
Recursos Humanos RHU		Rhum_Locd				x
Recursos Humanos RHU		Rhum_Desc				x
Recursos Humanos RHU		Rhut_Carg_Pers				x
Importaciones IMP		Impm_Ciud				x
Seguridad SEG		segm_User				x
Seguridad SEG		segt_Sesi	x		x	
Consultas Gerenciales CAR	<i>ModValD</i>	Carm_Arch_Almc				x
		Carm_Deno-Arch				x
		Carm_Carp_Fold				x
Recursos Humanos RHU		Rhum_Pers				x
		Rhum_Locd				x
		Rhum_Desc				x
Seguridad SEG		Segp_Prmt		x		x
	<i>Modulo 2</i>					
Consultas Gerenciales CAR		Carm_Resp	x			

		Carm_Mens	x			
		Cart_Hist_Mens	x			
Seguridad SEG		Segp_Prmt	x			

ANEXO B

BASE DE DATOS DIGMAT-INFORMIX			
TABLAS_DIGMAT	EXITO	ERROR	Observación
adqm_conc_eval	x		Sin Novedad
adqm_porc_segu	x		Sin Novedad
adqm_prov	x		Máximo error alcanzado
adqm_prov_docu	x		Sin Novedad
adqm_prov_refe	x		Sin Novedad
adqm_prov_repr	x		Sin Novedad
adqm_prov_serv	x		Sin Novedad
adqm_prov_tdis	x		Sin Novedad
adqm_repr_lega	x		Máximo error alcanzado
adqm_tipo_comi	x		Sin Novedad
adqp_dmon	x		Máximo error alcanzado
adqp_mont	x		Sin Novedad
adqp_rang_cali	x		Sin Novedad
adqt_cabe_coti	x		Sin Novedad
adqt_cabe_ocom	x		Máximo error alcanzado
adqt_cabe_solc	x		Máximo error alcanzado
adqt_cont	x		Máximo error alcanzado
adqt_cont_gara	x		Máximo error alcanzado
adqt_crit_sele	x		Máximo error alcanzado
adqt_cuad_ofer	x		Máximo error alcanzado
adqt_deta_afi	x		Sin Novedad
adqt_deta_coti	x		Máximo error alcanzado
adqt_deta_ocom	x		Máximo error alcanzado
adqt_deta_ofer	x		Máximo error alcanzado
adqt_deta_solc	x		Máximo error alcanzado
adqt_docu_rubr	x		Sin Novedad
adqt_enlc_cont	x		Sin Novedad
adqt_firm_apro	x		Sin Novedad
adqt_firm_solc	x		Sin Novedad
adqt_fpag	x		Sin Novedad
adqt_ocom_deta_afi	x		Sin Novedad
adqt_ocom_fact	x		Sin Novedad
afih_bica_hist	x		Sin Novedad
afim_acop_locd	x		Sin Novedad
afim_bien_cara	x		Máximo error alcanzado
afim_cabe_movi	x		Sin Novedad
afim_cara	x		Sin Novedad
afim_cara_actf	x		Sin Novedad
afim_cara_old	x		Sin Novedad
afim_cent_acop	x		Sin Novedad
afim_comp	x		Sin Novedad
afim_deno	x		Máximo error alcanzado
afim_encar	x		Sin Novedad
afim_encar_locd	x		Sin Novedad
afim_sist	x		Sin Novedad
afim_sist_locd	x		Sin Novedad
afim_tran	x		Sin Novedad
afit_deta_bien	x		Máximo error alcanzado
afit_deta_movi	x		Sin Novedad
afit_poli	x		Sin Novedad
afit_poli_actf	x		Sin Novedad
afit_poli_bono	x		Sin Novedad
afit_temp_cabe	x		Errores

afit_temp_deta	x		Sin Novedad
afit_temp_deta2	x		Sin novedad
afit_tran_cont	x		Sin novedad
alat_docu_nsub	x		Sin novedad
alat_tran_docu	x		Sin novedad
audm_cart_pres	x		Sin novedad
audm_clas_insp	x		Sin novedad
audm_dato_audi	x		Sin novedad
audm_info_resu	x		Sin novedad
audm_moti_insp	x		Sin novedad
audm_rubr	x		Sin novedad
audm_tipo_info	x		Sin novedad
audm_tipo_resp	x		Sin novedad
audm_zona	x		Sin novedad
audt_asig_resp	x		Sin novedad
audt_cbdc_erra	x		Sin novedad
audt_dtcd_erra	x		Sin novedad
audt_equi_insp	x		Sin novedad
audt_insp	x		Sin novedad
audt_loca_insp	x		Sin novedad
audt_log	x		Invalid datatype
audt_rubr_insp	x		Sin novedad
audt_tran	x		Invalid datatype
aux_cgla_suma_sald	x		Sin Novedad
aux_cglh_sald_auxi	x		Sin Novedad
aux_cglh_sald_auxi	x		Sin Novedad
aux_cglh_sald_cont	x		Sin Novedad
aux_cgl_t_cabe_cont	x		Máximo error alcanzado
aux_cgl_t_deta_cont	x		Máximo error alcanzado
bak_sald_sucr	x		Sin Novedad
bak_salm_mond	x		Sin Novedad
borrar	x		Sin Novedad
carh_anex	x		Sin Novedad
carh_cabc_tbod	x		Sin Novedad
carh_deta_tbod	x		Sin Novedad
carh_recl_docu	x		Sin Novedad
carm_acta_cabc_des	x		sin Novedad
carm_arch_almc	x		Sin Novedad
carm_bita_user	x		Sin Novedad
carm_cuer	x		Sin Novedad
carm_cuer2	x		Sin Novedad
carm_deno_arch	x		Sin Novedad
carm_estd_mesg	x		Sin Novedad
carm_mens	x		Máximo error alcanzado
carm_para	x		Sin Novedad
carm_pers_resp	x		Sin Novedad
carm_prio	x		Sin Novedad
carm_tipo_dest	x		Sin Novedad
carp_tipo_auto	x		Sin Novedad
cart_acta_deta	x		Sin Novedad
cart_anex	x		Máximo error alcanzado
cart_bita	x		Máximo error alcanzado
cart_calf_arch	x		Sin Novedad
cart_carp_fold	x		Sin Novedad
cart_divs_carp	x		Sin Novedad
cart_docu_ment	x		Sin Novedad

cart_hist_mens	x		Máximo error alcanzado
cart_manl_libr	x		Sin Novedad
cart_otro	x		Máximo error alcanzado
cart_pres_tamo	x		Sin Novedad
cart_subd_carp	x		Sin Novedad
cgl_a_asie_aper	x		Sin Novedad
cgl_a_cabe_cont	x		Máximo error alcanzado
cgl_a_cont_pres	x		Sin Novedad
cgl_a_deta_aper	x		Sin Novedad
cgl_a_deta_cont	x		Máximo error alcanzado
cgl_a_inte_cont	x		3 errores
cgl_a_pres	x		Sin Novedad
cgl_a_sald_cont	x		Sin Novedad
cgl_a_sald_cont2	x		Sin Novedad
cgl_a_sald_cont3	x		Sin Novedad
cgl_a_sald_pres	x		Sin Novedad
cgl_a_suma_sald	x		Sin Novedad
cgl_a_tran_cont	x		Máximo error alcanzado
cgl_e_deta_comp	x		Máximo error alcanzado
cgl_h_cabe_cont	x		Máximo error alcanzado
cgl_h_cabe_inte	x		Máximo error alcanzado
cgl_h_deta_cont	x		Máximo error alcanzado
cgl_h_deta_inte	x		Máximo error alcanzado
cgl_h_sald_auxi	x		Máximo error alcanzado
cgl_h_sald_auxi_copy		x	Un error de sintaxis
cgl_h_sald_cont	x		Sin Novedad
cgl_m_cabe_asie	x		Sin Novedad
cgl_m_conc_deud	x		Sin Novedad
cgl_m_conc_inca	x		Sin Novedad
cgl_m_cxc_pre	x		Sin Novedad
cgl_m_deta_asie	x		Sin Novedad
cgl_m_deud	x		Máximo error alcanzado
cgl_m_docu	x		Sin Novedad
cgl_m_grup	x		Sin Novedad
cgl_m_kard	x		Máximo error alcanzado
cgl_m_masc	x		Sin Novedad
cgl_m_masc_grup	x		Sin Novedad
cgl_m_masc_mayo	x		Sin Novedad
cgl_m_masc_subs	x		Sin Novedad
cgl_m_mayo_gene	x		Sin Novedad
cgl_m_plan_ctas	x		Sin Novedad
cgl_m_rela_ctka	x		Sin Novedad
cgl_m_subs_grup	x		Sin Novedad
cgl_m_tabl_dicc	x		Sin Novedad
cgl_m_tipo_deud	x		Sin Novedad
cgl_m_tipo_inba	x		Sin Novedad
cgl_m_tipo_kard	x		Sin Novedad
cgl_t_apli_asie	x		Sin Novedad
cgl_t_auto	x		Sin Novedad
cgl_t_cabe_cont	x		Máximo error alcanzado
cgl_t_cabe_cont_2003		x	Un error de sintaxis
cgl_t_cabe_cont_31012005		x	Un error de sintaxis
cgl_t_cabe_cont2004	x		Sin Novedad
cgl_t_cabe_inte	x		Máximo error alcanzado
cgl_t_caib	x		Sin Novedad
cgl_t_caib_caic	x		Sin Novedad

cgl_t_caib_cart	x		Sin Novedad
cgl_t_caib_depo	x		Máximo error alcanzado
cgl_t_caic	x		Sin Novedad
cgl_t_caic_cart	x		Sin Novedad
cgl_t_caic_ctac	x		Sin Novedad
cgl_t_cart_ctac	x		Sin Novedad
cgl_t_cheq	x		Sin Novedad
cgl_t_comb_kard	x		Sin Novedad
cgl_t_cred	x		Sin Novedad
cgl_t_deta_cart	x		Máximo error alcanzado
cgl_t_deta_con2	x		Sin Novedad
cgl_t_deta_con3	x		Sin Novedad
cgl_t_deta_cont	x		Máximo error alcanzado
cgl_t_deta_cont_2003	x		Sin Novedad
cgl_t_deta_cont_31012005	x		Sin Novedad
cgl_t_deta_cont2004	x		Máximo error alcanzado
cgl_t_deta_inte	x		Máximo error alcanzado
cgl_t_docu_apli	x		Sin Novedad
cgl_t_docu_firm	x		Sin Novedad
cgl_t_pres	x		Sin Novedad
cgl_t_rela_cuen	x		Sin Novedad
cgl_t_rela_giop	x		Sin Novedad
cgl_t_veri	x		Sin Novedad
cgl_v_depo_cobr2	x		Sin Novedad
cgrm_emba	x		Sin Novedad
cgrm_libr	x		Sin Novedad
cgrm_sist	x		Sin Novedad
cgrt_comp_numb	x		Sin Novedad
cgrt_comp_piez	x		Sin Novedad
cgrt_escu_sist	x		Sin Novedad
cgrt_repu_manu	x		Sin Novedad
cgrt_sist_equi	x		Sin Novedad
cgrt_sist_manu	x		Sin Novedad
comh_cont_tare	x		Sin Novedad
comm_carg_asis	x		Sin Novedad
comm_prov	x		Sin Novedad
comm_sesi	x		Sin Novedad
comm_tare	x		Sin Novedad
comm_tare_tico	x		Sin Novedad
comm_tipo_prov	x		Sin Novedad
comm_tipo_soli	x		Sin Novedad
comt_asit	x		Sin Novedad
comt_cabe_plan	x		Sin Novedad
comt_cabe_soli	x		Máximo error alcanzado
comt_cert_fina	x		Máximo error alcanzado
comt_cont_tare	x		Sin Novedad
comt_deta_plan	x		Sin Novedad
comt_deta_soli	x		Máximo error alcanzado
comt_prov_tipo	x		Sin Novedad
comt_sesi	x		Máximo error alcanzado
comt_sesi_tare	x		Sin Novedad
comt_soli_prov	x		Máximo error alcanzado
cons_ntpd_per	x		Sin Novedad
cxph_conc_banc	x		Máximo error alcanzado
cxph_conc_deta	x		Sin Novedad
cxph_tran_banc	x		Máximo error alcanzado

cxpm_ajus_banc	x		Sin Novedad
cxpm_audi_cheq	x		Sin Novedad
cxpm_bank	x		Sin Novedad
cxpm_bnfc	x		Máximo error alcanzado
cxpm_cheq	x		Máximo error alcanzado
cxpm_cheq_retc	x		Sin Novedad
cxpm_cmeg	x		Máximo error alcanzado
cxpm_csri	x		Sin Novedad
cxpm_ctas_banc	x		Máximo error alcanzado
cxpm_ctas_pagr	x		Máximo error alcanzado
cxpm_deta_ajus	x		Sin Novedad
cxpm_irtf	x		Sin Novedad
cxpm_ntdb	x		Máximo error alcanzado
cxpm_temp_tran	x		Sin Novedad
cxpt_abon	x		Sin Novedad
cxpt_anul_docu	x		Sin Novedad
cxpt_cheq_entg	x		Sin Novedad
cxpt_deta_cons	x		Máximo error alcanzado
cxpt_icdc_ctpg	x		Sin Novedad
cxpt_tran_deta	x		Máximo error alcanzado
cxpt_unpm	x		Sin Novedad
deci_tran_coga	x		Máximo error alcanzado
doc_dol	x		Sin Novedad
doc_mone	x		Sin Novedad
facm_tipo_kard	x		Sin Novedad
fact_cabc_trns	x		Máximo error alcanzado
fact_cobr	x		Sin Novedad
fact_ctas_pagr	x		Sin Novedad
fact_deta_cupo	x		Sin Novedad
fact_deta_trns	x		Sin Novedad
fact_giro_banc	x		Máximo error alcanzado
genmopci	x		Máximo error alcanzado
impm_agre	x		Sin Novedad
impm_agre_bank	x		Sin Novedad
impm_cabe_acta_ins	x		Sin Novedad
impm_cabe_liqu	x		Máximo error alcanzado
impm_cabe_lste	x		Máximo error alcanzado
impm_cabe_ntpd	x		Máximo error alcanzado
impm_cabe_ntpd_aux	x		Máximo error alcanzado
impm_cabe_ntpd_pag	x		Sin Novedad
impm_ciud	x		Sin Novedad
impm_codi_impo	x		Máximo error alcanzado
impm_comu	x		Máximo error alcanzado
impm_dicc	x		Sin Novedad
impm_egre_pres	x		Sin Novedad
impm_emba	x		Sin Novedad
impm_mone	x		Sin Novedad
impm_ntpd_gere	x		Sin Novedad
impm_ofic_ntpd	x		Sin Novedad
impm_orde_admi	x		Máximo error alcanzado
impm_orde_admi_sci	x		Sin Novedad
impm_pais	x		Sin Novedad
impm_segu_impo	x		Sin Novedad
impm_stad	x		Sin Novedad
impt_deta_acta_ins	x		Sin Novedad
impt_deta_item_liq	x		Sin Novedad

impt_deta_liqu	x		Sin Novedad
impt_deta_lste	x		Sin Novedad
impt_deta_ntpd	x		Sin Novedad
impt_deta_ntpd_aux	x		Máximo error alcanzado
impt_deta_part_pre	x		Máximo error alcanzado
impt_deta_peti	x		Sin Novedad
impt_deta_segu	x		Sin Novedad
impt_docu_liqu	x		Máximo error alcanzado
impt_enlc_cont	x		Sin Novedad
impt_enlc_docu	x		Sin Novedad
impt_enlc_segu_imp	x		Sin Novedad
impt_gara	x		Sin Novedad
impt_gast	x		Sin Novedad
impt_lste_repa	x		Sin Novedad
impt_ntpd_cntl	x		Sin Novedad
impt_ntpd_deta_afi	x		Sin Novedad
impt_ntpd_situ	x		Sin Novedad
impt_tipo_camb	x		Sin Novedad
impt_tran_pres	x		Sin Novedad
impw_repo_liqu	x		Máximo error alcanzado
inme_exis_fech	x		Sin Novedad
inme_sald_comb	x		Sin Novedad
inmm_docu	x		Sin Novedad
inmm_docu_ante	x		Sin Novedad
inmm_espe_vehi	x		Sin Novedad
inmm_grup	x		Sin Novedad
inmm_grup_clas	x		Sin Novedad
inmm_item	x		Máximo error alcanzado
inmm_part_pano	x		Sin Novedad
inmm_prec_pano	x		Sin Novedad
inmm_prec_pano_feb	x		Sin Novedad
inmm_repa	x		Sin Novedad
inmm_subc	x		Sin Novedad
inmm_tipo_vehi	x		Sin Novedad
inmm_vehi	x		Sin Novedad
inmt_asig_cbvh	x		Máximo error alcanzado
inmt_asig_dtvh	x		Máximo error alcanzado
inmt_bode_item	x		Sin Novedad
inmt_cabe_cprv	x		Sin Novedad
inmt_cabe_pedi	x		Máximo error alcanzado
inmt_ctrl_comb	x		Sin Novedad
inmt_deta_cprv	x		Sin Novedad
inmt_deta_pedi	x		Máximo error alcanzado
inmt_item	x		Sin Novedad
inmt_kdrx_cons	x		Sin Novedad
inmt_mant_vehi	x		Sin Novedad
inmt_sald_diar	x		Sin Novedad
inmt_temp_comb	x		Sin Novedad
inmt_vehi_cupo	x		Sin Novedad
invh_cabe_bode	x		Sin Novedad
invh_ccon	x		Máximo error alcanzado
invh_dcon	x		Máximo error alcanzado
invh_deta_item	x		Sin Novedad
invm_barr_item	x		Sin Novedad
invm_bode	x		Sin Novedad
invm_cabe_plan	x		Sin Novedad

invm_cabe_stoc	x		Sin Novedad
invm_cate_bode	x		Sin Novedad
invm_dema	x		Sin Novedad
invm_deno	x		Sin Novedad
invm_equi_nsn	x		Sin Novedad
invm_exis	x		Sin Novedad
invm_exis_19ene	x		Sin Novedad
invm_exis_feb	x		Sin Novedad
invm_fina	x		Sin Novedad
invm_marc	x		Sin Novedad
invm_mate	x		Sin Novedad
invm_mate_bode	x		Sin Novedad
invm_perc	x		Sin Novedad
invm_prec_item	x		Sin Novedad
invm_proc	x		Sin Novedad
invm_repu	x		Máximo error alcanzado
invm_tipo	x		Sin Novedad
invm_tipo_bode	x		Sin Novedad
invm_tipo_item	x		Sin Novedad
invm_tipo_sust	x		Sin Novedad
invm_tipo_unid	x		Sin Novedad
invm_tran	x		Sin Novedad
invm_ubic	x		Sin Novedad
invm_unid	x		Sin Novedad
invt_bode	x		Sin Novedad
invt_cabe_actc	x		Máximo error alcanzado
invt_cabe_cata	x		Máximo error alcanzado
invt_cabe_dema	x		Sin Novedad
invt_clas_subc	x		Sin Novedad
invt_cmov	x		Máximo error alcanzado
invt_deta_actc	x		Sin Novedad
invt_deta_dema	x		Sin Novedad
invt_deta_plan	x		Sin Novedad
invt_deta_stoc	x		Máximo error alcanzado
invt_deta_tran_pre	x		Sin Novedad
invt_dmov	x		Máximo error alcanzado
invt_dmov_deta	x		Sin Novedad
invt_item_nive	x		Sin Novedad
invt_plan	x		Sin Novedad
invt_prec	x		Sin Novedad
invt_repu	x		Sin Novedad
invt_repu_part	x		Sin Novedad
invt_sust	x		Sin Novedad
invt_temp_sald	x		Sin Novedad
invt_temp_sald_rep	x		Sin Novedad
invt_temp_ubic	x		Sin Novedad
men_dol	x		Sin Novedad
men_mone	x		Sin Novedad
ml_connection_script		x	Un error de sintaxis
pla_anls_cost_unit	x		Máximo error alcanzado
pla_list_cost_unit	x		Máximo error alcanzado
pla_list_elem_pres	x		Sin Novedad
plah_avan_deta	x		Sin Novedad
plah_avan_obra	x		Sin Novedad
plah_cont	x		Sin Novedad
plah_obra	x		Máximo error alcanzado

plah_ofer_comi	x		Sin Novedad
plah_orde_trab	x		Máximo error alcansado
plah_orde_trab1	x		Sin Novedad
plah_pres_anls	x		Máximo error alcansado
plah_pres_esti	x		Máximo error alcansado
plam_equi_cons	x		Sin Novedad
plam_rubr	x		Sin Novedad
plam_subr	x		Máximo error alcansado
plam_subr_item	x		Sin Novedad
plam_tipo_trab	x		Sin Novedad
plap_fpag	x		Sin Novedad
plap_remu	x		Sin Novedad
plap_secu_obra	x		Sin Novedad
plar_esti_npre	x		Máximo error alcansado
plar_esti_spre	x		Máximo error alcansado
plar_ord_tra_npres	x		Máximo error alcansado
plat_anls_cost	x		Máximo error alcansado
plat_anls_rubr	x		Máximo error alcansado
plat_cabe_camb	x		Sin Novedad
plat_cabe_ofer	x		Sin Novedad
plat_coti_solc	x		Sin Novedad
plat_desc_cont	x		Máximo error alcansado
plat_deta_camb	x		Sin Novedad
plat_deta_cont	x		Máximo error alcansado
plat_deta_coti	x		Sin Novedad
plat_deta_ofer	x		Sin Novedad
plat_deta_plan	x		Máximo error alcansado
plat_deta_plan1	x		Sin Novedad
plat_deta_solc	x		Sin Novedad
plat_dist_rubr	x		Sin Novedad
plat_espe_tecn	x		Sin Novedad
plat_form_pago	x		Sin Novedad
plat_gara	x		Sin Novedad
plat_orde_deta	x		Máximo error alcansado
plat_orde_deta1	x		Sin Novedad
plat_pago_solc	x		Sin Novedad
plat_plan_cab	x		Máximo error alcansado
plat_pres_deta	x		Máximo error alcansado
plat_rela_prov	x		Sin Novedad
plat_sala	x		Sin Novedad
plat_soli_cont	x		Sin Novedad
preh_movi_pres	x		Sin Novedad
preh_sald_sucr	x		Sin Novedad
preh_salm_menm	x		Máximo error alcansado
preh_salm_mond	x		Sin Novedad
preh_sals_mens	x		Máximo error alcansado
preh_tran_coga	x		Máximo error alcansado
prem_acci_cabe	x		Sin Novedad
prem_acci_deta	x		Sin Novedad
prem_cabe_bise	x		Sin Novedad
prem_cabe_docm	x		Máximo error alcansado
prem_calt_bise	x		Sin Novedad
prem_dalt_bise	x		Sin Novedad
prem_deta_bise	x		Sin Novedad
prem_deta_docm	x		Sin Novedad
prem_deta_solc	x		Máximo error alcansado

prem_fapr_rela	x		Sin Novedad
prem_fase_pres	x		Sin Novedad
prem_fond_part	x		Sin Novedad
prem_item_pres	x		Sin Novedad
prem_locd_armd	x		Sin Novedad
prem_nump_proy	x		Sin Novedad
prem_obse_odre	x		Sin Novedad
prem_pagd_pres	x		Sin Novedad
prem_part_proy	x		Máximo error alcanzado
prem_plan_basi	x		Sin Novedad
prem_porc_segf	x		Sin Novedad
prem_prgr_rela	x		Sin Novedad
prem_proy_pres	x		invalid datatype
prem_prpg_rela	x		Sin Novedad
prem_rela_acti	x		Sin Novedad
prem_rela_docm	x		Sin Novedad
prem_rela_lcpg	x		Sin Novedad
prem_rela_locd	x		Sin Novedad
prem_rela_plpa	x		Sin Novedad
prem_rela_repu	x		Sin Novedad
prem_segf_item	x		Sin Novedad
prem_sgfi_bise	x		Sin Novedad
prem_situ_proy	x		Sin Novedad
prem_solc_alte	x		Sin Novedad
prem_tech_fina	x		Sin Novedad
prem_tipo_proy	x		Sin Novedad
prep_audi_cedu	x		Sin Novedad
prep_cepr_dola	x		Sin Novedad
prep_cepr_mone	x		Sin Novedad
prep_comp	x		Sin Novedad
prep_rela_part	x		Sin Novedad
prep_salm_menm	x		Máximo error alcanzado
prep_sals_mens	x		Máximo error alcanzado
prep_tabl_pres	x		Sin Novedad
pret_bise_cabe	x		Sin Novedad
pret_bise_pres	x		Sin Novedad
pret_dbis_segi	x		Sin Novedad
pret_deno_auxi	x		Sin Novedad
pret_movi_pres	x		Sin Novedad
pret_prfi_pres	x		Sin Novedad
pret_rela_epcr	x		Sin Novedad
pret_rela_lcpr	x		Sin Novedad
pret_sald_sucr	x		Sin Novedad
pret_salm_mond	x		Sin Novedad
pret_tran_coga	x		Sin Novedad
prev_cabe_docu	x		Sin Novedad
prev_cedula	x		Sin Novedad
prev_deta_docu	x		Revisar si migro todos los datos
prev_fase	x		Sin Novedad
prev_fondo	x		Sin Novedad
prev_item	x		Sin Novedad
prev_partida	x		Sin Novedad
prev_proyecto	x		Sin Novedad
prueba_permisos	x		Sin Novedad
rhum_atra_temp	x		Sin Novedad
rhum_cabc_rgua	x		Sin Novedad

rhum_camb_part	x		Sin Novedad
rhum_carg	x		Sin Novedad
rhum_cate_civi	x		Sin Novedad
rhum_catr	x		Sin Novedad
rhum_catr_locd	x		Sin Novedad
rhum_cert	x		Sin Novedad
rhum_cpto_cali	x		Sin Novedad
rhum_crtt	x		Sin Novedad
rhum_desc	x		Sin Novedad
rhum_desc_carg	x		Sin Novedad
rhum_dicc	x		Sin Novedad
rhum_espe	x		Sin Novedad
rhum_func	x		Sin Novedad
rhum_grad	x		Sin Novedad
rhum_locd	x		Sin Novedad
rhum_pers	x		Error: Campo decimal solo soporta 18
rhum_pert	x		Sin Novedad
rhum_pfis	x		Sin Novedad
rhum_prom	x		Sin Novedad
rhum_stdo	x		Sin Novedad
rhum_subs	x		Sin Novedad
rhum_temp	x		Sin Novedad
rhum_tipo_auto	x		Sin Novedad
rhum_tipo_locd	x		Sin Novedad
rhum_tipo_movi	x		Sin Novedad
rhum_tipo_part	x		Sin Novedad
rhum_tipo_perm	x		Sin Novedad
rhum_titu	x		Sin Novedad
rhum_viat	x		Sin Novedad
rhut_argu	x		Sin Novedad
rhut_asce	x		Sin Novedad
rhut_auto	x		Sin Novedad
rhut_avan	x		Sin Novedad
rhut_cabe_tarj	x		Sin Novedad
rhut_cabe_ufin	x		Sin Novedad
rhut_cagu	x		Sin Novedad
rhut_cali	x		Sin Novedad
rhut_calo	x		Sin Novedad
rhut_calt	x		Sin Novedad
rhut_camb	x		Sin Novedad
rhut_carg_pers	x		Sin Novedad
rhut_cate	x		Sin Novedad
rhut_cond	x		Sin Novedad
rhut_curs	x		Sin Novedad
rhut_curs_capa	x		Sin Novedad
rhut_ddia	x		Sin Novedad
rhut_deta_cali	x		Sin Novedad
rhut_deta_rgua	x		Sin Novedad
rhut_deta_subs	x		Sin Novedad
rhut_deta_tarj	x		Sin Novedad
rhut_deta_ufin	x		Sin Novedad
rhut_deta_viat	x		Sin Novedad
rhut_deto	x		Sin Novedad
rhut_dett	x		Sin Novedad
rhut_diar	x		Sin Novedad
rhut_dpfi	x		Sin Novedad

rhut_dres	x		Sin Novedad
rhut_educ	x		Sin Novedad
rhut_expe_doce	x		Sin Novedad
rhut_fald	x		Sin Novedad
rhut_form_acad	x		Sin Novedad
rhut_falt	x		Sin Novedad
rhut_hobb	x		Sin Novedad
rhut_hora_perm	x		Sin Novedad
rhut_mejo_carg	x		Sin Novedad
rhut_movi	x		Sin Novedad
rhut_part	x		Sin Novedad
rhut_perm	x		maximo error alcanzado
rhut_plan_curs	x		Sin Novedad
rhut_prof	x		Sin Novedad
rhut_repo	x		Sin Novedad
rhut_resp	x		Sin Novedad
rhut_tgua_asig	x		Sin Novedad
rhut_titu	x		Sin Novedad
rhut_titu_pers	x		Sin Novedad
rhut_tran	x		Sin Novedad
rhut_vaca	x		Sin Novedad
rhut_valo	x		Sin Novedad
rhut_viat_pres	x		Sin Novedad
rpl_transaccion	x		maximo error alcanzado
rtkm_dire	x		Sin Novedad
rtkm_parm	x		Sin Novedad
rtkm_rmov	x		Sin Novedad
rtkp_emsg	x		Sin Novedad
rtkp_mex	x		Sin Novedad
sctt_soli_tran	x		Sin Novedad
seg_desc_apli	x		Sin Novedad
sep26_sald_sucr	x		Sin Novedad
sep26_salm_menm	x		Sin Novedad
sep26_salm_mond	x		Sin Novedad
sep26_sals_mens	x		Sin Novedad
sucre_inmm_item	x		maximo error alcanzado
sucre_invm_exis	x		Sin Novedad
sucre_invm_repu	x		maximo error alcanzado
sysdomains	x		Sin Novedad
tab_col	x		Sin Novedad
tab_log	x		Sin Novedad
tem_audt_log	x		error sintaxis
temp_cglt_cabe	x		maximo error alcanzado
temp_plan_tot	x		Sin Novedad
temp_plan_valo	x		Sin Novedad
temporal	x		maximo error alcanzado
tmp_salm_menm	x		Sin Novedad
tmp_sals_mens	x		Sin Novedad
tram_locd	x		Sin Novedad
tram_ruta	x		Sin Novedad
trat_cabe_cove	x		Sin Novedad
trat_deta_cove	x		Sin Novedad
vehiculo_por_cupo	x		Sin Novedad
vw_adq_ocom_locd	x		Sin Novedad
vw_adq_ocom_prov	x		Sin Novedad
vw_adq_solcmate	x		maximo error alcanzado

vw_afi_crsp	x		Sin Novedad
vw_afi_cta1	x		Sin Novedad
vw_afi_espc	x		Sin Novedad
vw_afi_espc_bica	x		maximo error alcanzado
vw_afi_locd	x		Sin Novedad
vw_afi_resp	x		Sin Novedad
vw_afi_xcrt	x		Sin Novedad
vw_aud_apli	x		Sin Novedad
vw_búsqueda	x		Sin Novedad
vw_car_acta_dest	x		Sin Novedad
vw_car_acta_pers	x		Sin Novedad
vw_car_docu_elim	x		Sin Novedad
vw_car_docu_enre	x		Sin Novedad
vw_car_docu_envi	x		Sin Novedad
vw_car_docu_peli	x		Sin Novedad
vw_car_manl_elim	x		Sin Novedad
vw_car_manl_peli	x		Sin Novedad
vw_car_manl_reci	x		Sin Novedad
vw_car_pres_tamo	x		Sin Novedad
vw_car_tras_docu	x		Sin Novedad
vw_car_ubic_docu	x		Sin Novedad
vw_car_ubic_manl	x		Sin Novedad
vw_cgl_comp	x		maximo error alcanzado
vw_cgl_comp_hist	x		maximo error alcanzado
vw_cgl_mayo	x		Sin Novedad
vw_cgl_prue	x		Sin Novedad
vw_cgl_prue2	x		Sin Novedad
vw_cheq_reci	x		Sin Novedad
vw_cupo_desp	x		Sin Novedad
vw_cupo_rept	x		Sin Novedad
vw_cxp_bolt_chco	x		Sin Novedad
vw_cxp_concil	x		Sin Novedad
vw_cxp_ctas_cons		x	identificador invalido
vw_cxp_ctas_pagr	x		maximo error alcanzado
vw_dba_sp	x		Sin Novedad
vw_dba_systables	x		Sin Novedad
vw_dba_tablas	x		Sin Novedad
vw_dba_triggers	x		Sin Novedad
vw_dba_vistas	x		Sin Novedad
vw_dpre_subr	x		maximo error alcanzado
vw_guar	x		Sin Novedad
vw_imp_cons_cont	x		Sin Novedad
vw_imp_emba	x		Sin Novedad
vw_imp_item_ntpd	x		maximo error alcanzado
vw_imp_pers_deta	x		Sin Novedad
vw_inm_cons_comb	x		Sin Novedad
vw_inm_tras_bode	x		Sin Novedad
vw_trans_movi	x		caracter numeric converter error
vw_inr_depexis	x		Sin Novedad
vw_inr_tras_bode	x		verificar si se migraron todos los datos
vw_list_obra	x		numero maximo error alcanzado
vw_ord_tra_equi	x		numero maximo error alcanzado
vw_part	x		verificar si se migraron todos los datos
vw_pers_carg_locd	x		Sin Novedad
vw_pers_func	x		Sin Novedad
vw_personal	x		Sin Novedad

vw_pla_comi_eval	x		Sin Novedad
vw_pla_mate_bode	x		numero maximo error alcanzado
vw_pla_pedi_mate	x		numero maximo error alcanzado
vw_plan_gast	x		numero maximo error alcanzado
vw_plan_orde	x		numero maximo error alcanzado
vw_pres	x		Sin Novedad
vw_rhu_anti_grad	x		Sin Novedad
vw_rhu_pers_anti	x		Sin Novedad
vw_rhu_pers_espe	x		Sin Novedad
vw_rhu_subs	x		Sin Novedad
vw_segu_audi	x		Sin Novedad
vw_trab_const	x		Sin Novedad
vw_trans_bode	x		Sin Novedad
vw_trans_movi_repu	x		caracter numeric converter error
vw_valo_inve_gast	x		verificar si se migraron datos bien
vw_valo_inve_mano	x		numero maximo error alcanzado

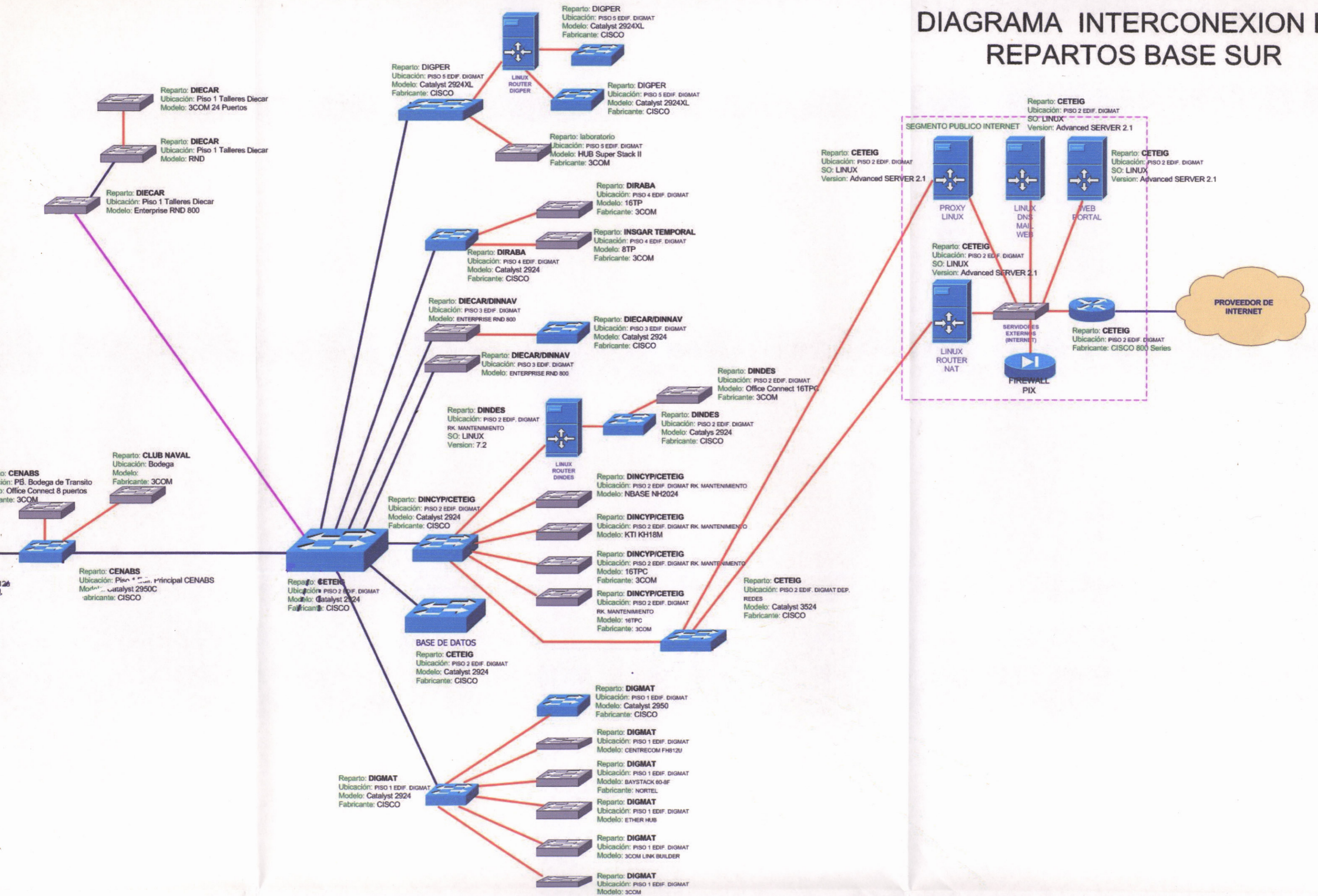
SYSTEM TABLE			
sysfragments	x		Sin Novedad
sysfragauth	x		Sin Novedad
syserrors	x		Sin Novedad
sysdistrib	x		Sin Novedad
sysdepend	x		Sin Novedad
sysdefaults	x		Sin Novedad
sysconstraints	x		Sin Novedad
syscolumns	x		Sin Novedad
syscoldepend	x		Sin Novedad
syscolauth	x		Sin Novedad
syscolattrs	x		Sin Novedad
syschecks	x		Sin Novedad
syscasts	x		Sin Novedad
sysblobs	x		Sin Novedad
sysattrtypes	x		Sin Novedad
sysams	x		Sin Novedad
sysaggregates	x		Sin Novedad
sysindexes	x		Sin Novedad
sysindices	x		Sin Novedad
sysinherits	x		Sin Novedad
syslangauth	x		Sin Novedad
syslogmap	x		Sin Novedad
sysobjstate	x		Sin Novedad
sysopclasses	x		Sin Novedad
sysopclstr	x		Sin Novedad
sysprocauth	x		Sin Novedad
sysprocbody	x		Sin Novedad
sysprocedures	x		Sin Novedad
sysprocplan	x		Sin Novedad
sysreferences	x		Sin Novedad
sysroleauth	x		Sin Novedad
sysroutinelangs	x		Sin Novedad
sys synonyms	x		Sin Novedad
sys syntable	x		Sin Novedad
sys tabamdata	x		Sin Novedad
sys tabauth	x		Sin Novedad
sys tables	x		Sin Novedad

systraceclasses	x		Sin Novedad
systracemsgs	x		Sin Novedad
systrigbody	x		Sin Novedad
systriggers	x		Sin Novedad
sysusers	x		Sin Novedad
sysviews	x		Sin Novedad
sysviolations	x		Sin Novedad
sysxtddesc	x		Sin Novedad
sysxtdtypeauth	x		Sin Novedad
sysxtdtypes	x		Sin Novedad

BASE DE DATOS SEGURIDAD-INFORMIX			
TABLAS_SEGURIDAD	EXITO	ERROR	Observación
cam_cuer			sin novedad
rpl_transaccion			MISMO NOMBRE TABLA DIGMAT
sege_clav_user	x		sin novedad
segh_audi	x		numero maximo de errores
segh_hist_perm	x		sin novedad
segh_hist_prmt	x		sin novedad
segh_hist_repa	x		sin novedad
segh_hist_user	x		sin novedad
segm_apli	x		sin novedad
segm_comp	x		sin novedad
segm_mens	x		sin novedad
segm_opcn	x		sin novedad
segm_user	x		errores en null
segm_user_web	x		sin novedad
segp_prmt	x		sin novedad
segp_prmt_audi	x		sin novedad
segt_apli_maqu	x		sin novedad
segt_apli_rept	x		sin novedad
segt_audi	x		numero maximo de errores
segt_audi_cxp	x		numero maximo de errores
segt_cone	x		sin novedad
segt_copy_vers	x		sin novedad
segt_list_arch	x		sin novedad
segt_list_exte	x		sin novedad
segt_menu	x		numero maximo de errores
segt_meop	x		sin novedad
segt_mont	x		sin novedad
segt_perf	x		sin novedad
segt_perm	x		sin novedad
segt_perm_web	x		sin novedad
segt_sesi	x		sin novedad
segt_tipo_mens	x		sin novedad
segt_user_mont	x		sin novedad
segt_user_repa	x		sin novedad
sysdomains			MISMO NOMBRE TABLA DIGMAT
sysindexes	x		sin novedad

ANEXO C

DIAGRAMA INTERCONEXION DE REPARTOS BASE SUR



ANEXO D

ANALISIS COMPARATIVO DDL

INFORMIX

ORACLE

Create TABLE

Create TABLE [nombre tabla]

(nombre-columna Tipo-Dato)[Clausulas Default] [NOT NULL]
[Clausulas CONSTRAINT]

Ejemplos:

```
CREATE TABLE "INFORMIX"."ADQM_CONC_EVAL" ("CONC_CODI" CHAR(3
byte) NOT NULL, "CONC_DESC" VARCHAR2(50 byte),
"CONC_FACT_IDEA_MIN" NUMBER(4, 1) NOT NULL,
"CONC_FACT_IDEA_MAX" NUMBER(4, 1) NOT NULL, "CONC_STAT_ADIC"
CHAR(1 byte) NOT NULL, "CONC_STAT" CHAR(1 byte) NOT NULL)
```

```
CREATE TABLE "INFORMIX"."ADQM_PORC_SEGU" ("PORC_CODI_ASEG"
NUMBER(10) NOT NULL, "PORC_TIPO_TRAM" CHAR(3 byte) NOT NULL,
"PORC_CODI_TRAM" CHAR(3 byte) NOT NULL, "PORC_VALO_PORC"
NUMBER(5, 2))
```

```
CREATE TABLE "INFORMIX"."ADQM_PROV" ("PROV_CODI" NUMBER(10) NOT
NULL, "PROV_NRUC_CEDU" VARCHAR2(20 byte) NOT NULL,
"PROV_STAT_RUC" CHAR(1 byte) NOT NULL, "PROV_CODI_PAIS"
CHAR(3 byte) NOT NULL, "PROV_CODI_ESTA" CHAR(3 byte),
"PROV_CODI_CIU" CHAR(3 byte), "PROV_NOMB" VARCHAR2(50 byte)
NOT NULL, "PROV_DIRE" VARCHAR2(250 byte), "PROV_PROC" CHAR(1
byte) NOT NULL, "PROV_TELE" VARCHAR2(20 byte), "PROV_FAX"
VARCHAR2(20 byte), "PROV_EMAI" VARCHAR2(50 byte), "PROV_CASI"
VARCHAR2(10 byte), "PROV_COME" VARCHAR2(50 byte),
"PROV_TIPO_PERS" CHAR(1 byte) NOT NULL, "PROV_TIPO_EMPR"
CHAR(1 byte) NOT NULL, "PROV_CATE_PROV" CHAR(1 byte) NOT NULL,
"PROV_CAPI_SOC" NUMBER(17, 4), "PROV_ACTI_CORR" NUMBER(17,
4), "PROV_PASI_CORR" NUMBER(17, 4), "PROV_ACTI_TOTA"
NUMBER(17, 4), "PROV_PASI_TOTA" NUMBER(17, 4),
```

create table [esquema.] nombreDeTabla (nombreDeLaColumna1 tipoDeDato(cantCaracteres))

Ejemplos:

```
create table ARMADA.ADQM_CONC_EVAL (
CONC_CODI CHAR(3) not null,
CONC_DESC VARCHAR2(50) not null,
CONC_FACT_IDEA_MIN NUMBER(4,1) not null,
CONC_FACT_IDEA_MAX NUMBER(4,1) not null,
CONC_STAT_ADIC CHAR not null,
CONC_STAT CHAR not null
)
```

```
create table ARMADA.ADQM_PORC_SEGU (
PORC_CODI_ASEG FLOAT not null,
PORC_TIPO_TRAM CHAR(3) not null,
PORC_CODI_TRAM CHAR(3) not null,
PORC_VALO_PORC NUMBER(5,2) not null
)
```

```
create table ARMADA.ADQM_PROV (
PROV_CODI FLOAT not null,
PROV_NRUC_CEDU VARCHAR2(20) not null,
PROV_STAT_RUC CHAR not null,
PROV_CODI_PAIS CHAR(3) not null,
PROV_CODI_ESTA CHAR(3) not null,
PROV_CODI_CIU CHAR(3) not null,
PROV_NOMB VARCHAR2(50) not null,
PROV_DIRE VARCHAR2(250) not null,
PROV_PROC CHAR not null,
PROV_TELE VARCHAR2(20) not null,
PROV_FAX VARCHAR2(20) not null,
PROV_EMAI VARCHAR2(50) not null,
```

"PROV_TOTA_INGR" NUMBER(17, 4), "PROV_TOTA_GAST" NUMBER(17, 4), "PROV_VALO_LIQU" NUMBER(5, 2), "PROV_VALO_SOLV" NUMBER(5, 2), "PROV_VALO_RENT" NUMBER(5, 2), "PROV_FECH_CONS" DATE NOT NULL, "PROV_ESTA_CONS" CHAR(1 byte) NOT NULL, "PROV_OBSE" VARCHAR2(100 byte), "PROV_STAT_FABR" CHAR(1 byte) NOT NULL, "PROV_TIPO_ASEG" CHAR(1 byte) NOT NULL, "PROV_STAT" CHAR(1 byte) NOT NULL, "PROV_STAT_CALF" CHAR(1 byte), "PROV_STAT_CTES" CHAR(1 byte), "PROV_NOMB_COMP" VARCHAR2(50 byte), "PROV_LIST_NEGR" CHAR(1 byte))

CREATE TABLE "INFORMIX"."ADQM_PROV_DOCU" ("PROV_DOCU_CODI_PRO" NUMBER(10) NOT NULL, "PROV_DOCU_TIPO_DOC" CHAR(3 byte) NOT NULL, "PROV_DOCU_CODI_DOC" CHAR(3 byte) NOT NULL, "PROV_DOCU_OBSE" VARCHAR2(250 byte), "PROV_DOCU_CUMP" CHAR(1 byte) NOT NULL)

CREATE TABLE "INFORMIX"."ADQM_PROV_REFE" ("PROV_REFE_CODI_PRO" NUMBER(10) NOT NULL, "PROV_REFE_TIPO_DIC" CHAR(3 byte) NOT NULL, "PROV_REFE_CODI" CHAR(3 byte) NOT NULL, "PROV_REFE_NUME_SEC" NUMBER(10) NOT NULL, "PROV_REFE_NOMB" VARCHAR2(50 byte))

PROV_CASI VARCHAR2(10) not null,
 PROV_COME VARCHAR2(50) not null,
 PROV_TIPO_PERS CHAR not null,
 PROV_TIPO_EMPR CHAR not null,
 PROV_CATE_PROV CHAR not null,
 PROV_CAPI_SOCIAL NUMBER(17,4) not null,
 PROV_ACTI_CORR NUMBER(17,4) not null,
 PROV_PASI_CORR NUMBER(17,4) not null,
 PROV_ACTI_TOTA NUMBER(17,4) not null,
 PROV_PASI_TOTA NUMBER(17,4) not null,
 PROV_TOTA_INGR NUMBER(17,4) not null,
 PROV_TOTA_GAST NUMBER(17,4) not null,
 PROV_VALO_LIQU NUMBER(5,2) not null,
 PROV_VALO_SOLV NUMBER(5,2) not null,
 PROV_VALO_RENT NUMBER(5,2) not null,
 PROV_FECH_CONS DATE not null,
 PROV_ESTA_CONS CHAR not null,
 PROV_OBSE VARCHAR2(100) not null,
 PROV_STAT_FABR CHAR not null,
 PROV_TIPO_ASEG CHAR not null,
 PROV_STAT CHAR not null,
 PROV_STAT_CALF CHAR not null,
 PROV_STAT_CTES CHAR not null,
 PROV_NOMB_COMP VARCHAR2(50) not null,
 PROV_LIST_NEGR CHAR not null
)

create table ARMADA.ADQM_PROV_DOCU (
 PROV_DOCU_CODI_PRO FLOAT not null,
 PROV_DOCU_TIPO_DOC CHAR(3) not null,
 PROV_DOCU_CODI_DOC CHAR(3) not null,
 PROV_DOCU_OBSE VARCHAR2(250) not null,
 PROV_DOCU_CUMP CHAR not null
)

create table ARMADA.ADQM_PROV_REFE (

```

PROV_REFE_CODI_PRO  FLOAT          not null,
PROV_REFE_TIPO_DIC  CHAR(3)        not null,
PROV_REFE_CODI      CHAR(3)        not null,
PROV_REFE_NUME_SEC  FLOAT          not null,
PROV_REFE_NOMB      VARCHAR2(50)   not null
)

```

Create VISTAS

INFORMIX

create view nombre-Vista [(Lista-Columnas)]

```

as
select-Sentencia [with check option]
Ejemplos:
create or replace view "prev_proyecto"(
"proyecto", "proy_denominacion", "proy_abreviacion", "rme",
"proy_rep_control_tecnico", "proy_rep_control_administrativ", "proy_tipo_prioridad",
"proy_numero_prioridad", "proy_fecha_inicio", "proy_fecha_fin", "loca_codigo",
PROY_FECH_FINA, PROY_CODI_LOCD)
as
select x0.proy_plan_basi, x0.proy_codi_prog, x0.proy_codi_tipo,
x0.proy_num_ejecc, x0.proy_deno_proy, x0.proy_abre_deno, x3.desc_sigl,
x5.desc_sigl, x7.desc_sigl, x0.proy_prio_proy [1,1], x0.proy_prio_proy [2,3],
x0.proy_fech_inic, x0.proy_fech_fina, x0.proy_codi_locd
from prem_proy_pres x0, segp_prmt x1, rhum_locd x2, rhum_desc x3,
rhum_locd x4, rhum_desc x5, rhum_locd x6, rhum_desc x7
where (((((((x0.proy_anio_ejec = x1.prmt_valr ) AND (x0.proy_esta_proy = 'A' ) )
AND (x1.prmt_codi LIKE 'prmt_anio_ejec' ) ) AND (x2.locd_codi =
x0.proy_codi_dire ) ) AND (x3.desc_codi = x2.locd_desc_codi ) ) AND
(x4.locd_codi = x0.proy_codi_rct ) ) AND (x5.desc_codi =
x4.locd_desc_codi ) ) AND (x6.locd_codi = x0.proy_codi_rca ) ) AND
(x7.desc_codi = x6.locd_desc_codi ) );

create or replace view "vw_cxp_ctas_cons"("plan_codi", "tipo_kard",
"codi_kard", "codi_skar", "total", "desc", "tipo_docu", "nume_docu", "ctpg_secc",
"fecha", "estado", "PLAN_NOME_CORT", "MAYO_DESC", "KARD_DESC",

```

ORACLE

create or replace view [esquema.] nombreDeVista

```

as
select [nombre de atributos]
from [nombre de las tablas]
where [las condiciones]
with check option;
Ejemplos:
create or replace view ARMADA."cgla_sald_cont2"
as
select x0.sald_ano, x0.sald_mes, x0.sald_plan_codi,
x0.sald_plan_fond, x0.sald_mont_debe, x0.sald_mont_habe
from cgla_sald_cont x0
with check option;

create or replace view ARMADA."cgla_sald_cont3"
as
select x0.sald_ano, x0.sald_mes, x0.sald_plan_codi,
x0.sald_plan_fond, x0.sald_mont_debe, x0.sald_mont_habe
from cgla_sald_cont x0
with check option;

create or replace view ARMADA."cglv_depo_cobr2"
as
select x2.caib_anio, x2.caib_codi, x1.caib_depo_secu,
x0.caib_caic_anio_icc, x0.caib_caic_codi_icc
from cglv_caib_caic x0, cglv_caib_depo x1, cglv_caib x2

```

"kard_desc2", "deha")

as

```
select x0.cons_plan_codi ,x0.cons_tipo_kard ,x0.cons_codi_kard ,
x0.cons_codi_skar ,x0.cons_tota ,x0.cons_desc ,x0.cons_tipo_docu ,
x0.cons_ume_docu ,x0.cons_ctpg_secc ,x0.cons_fech ,x0.cons_stat ,
x1.plan_nome_cort ,x4.mayo_desc ,x6.kard_desc ,x7.kard_desc ,'D'
from cxpt_deta_cons x0 , cglm_plan_ctas x1 , cglm_grup x2 , cglm_subs_grup x3 ,
cglm_mayo_gene x4 , cgl_t_comb_kard x5 , cglm_kard x6 , cglm_kard x7
where ((((((((((x0.cons_plan_codi = x1.plan_codi ) AND (x0.cons_tipo_kard =
x5.comb_kard_codi ) ) AND (x0.cons_codi_kard = x6.kard_codi ) ) AND
(x0.cons_codi_skar = x7.kard_codi ) ) AND (x1.plan_grup_codi = x2.grup_codi ) )
AND (x1.plan_grup_codi = x3.subs_grup_codi ) ) AND (x1.plan_subs_codi =
x3.subs_codi ) ) AND (x1.plan_grup_codi = x4.mayo_grup_codi ) )
AND (x1.plan_subs_codi = x4.mayo_subs_codi ) ) AND
(x1.plan_mayo_codi = x4.mayo_codi ) ) AND (x5.comb_kard_prim =
x6.kard_tipo ) ) AND (x5.comb_kard_segu =x7.kard_tipo ) )
union all select x8.cont_plan_codi ,x8.cont_tipo_kard ,x8.cont_codi_kard ,x
8.cont_codi_skar ,x8.cont_tota ,x8.cabe_glos ,x8.cabe_docu_codi , x8.cabe_codi ,
0',x8.cabe_fech_tran ,x8.cabe_stat ,x8.plan_nome_cort ,x8.mayo_desc ,
x8.kard_desc1 ,x8.kard_desc2 ,x8.cont_deha
from vw_cgl_comp x8
where ((x8.cabe_stat != 'E' ) AND (x8.cont_plan_codi = '2120723000' ) )
union all select x10.deta_inte_plan ,x10.deta_inte_tkar ,x10.deta_inte_kard ,
x10.deta_inte_skar ,x10.deta_inte_tota ,x9.inte_glos ,x9.inte_tipo_docu ,
x9.inte_codi_orig ,'0' ,x9.inte_fech_tran ,x9.inte_stat ,x11.plan_nome_cort ,
x14.mayo_desc ,x16.kard_desc ,x17.kard_desc ,x10.deta_inte_deha
from cgl_t_cabe_inte x9 , cgl_t_deta_inte x10 , cglm_plan_ctas x11 , cglm_grup x12 ,
cglm_subs_grup x13 , cglm_mayo_gene x14 , cgl_t_comb_kard x15 ,
cglm_kard x16 , cglm_kard x17
where ((((((((((((((x9.inte_ume_tran = x10.deta_inte_ntra ) AND
(x10.deta_inte_plan = x11.plan_codi ) ) AND (x10.deta_inte_tkar =
x15.comb_kard_codi ) ) AND (x10.deta_inte_kard = x16.kard_codi ) ) AND
(x10.deta_inte_skar = x17.kard_codi ) ) AND (x10.deta_inte_stat = 'A' ) ) AND
(x11.plan_grup_codi = x12.grup_codi ) ) AND (x11.plan_grup_codi =
x13.subs_grup_codi ) ) AND (x11.plan_subs_codi = x13.subs_codi ) ) AND
(x11.plan_grup_codi = x14.mayo_grup_codi ) ) AND (x11.plan_subs_codi =
```

where (x2.caib_tipo_ibco = '01')

with check option;

create or replace view ARMADA."cons_ntpd_per"(IMPO_SECU,
CABE_NTPD_ANIO, CABE_NTPD_SIGL, CABE_NTPD_NUME,
CABE_NTPD_FECH_ELA, CABE_NTPD_CODI_MON, CABE_NTPD_MONT_DIV,
CABE_NTPD_MONT_SUC, CABE_NTPD_REFE, IMPO_CODI_RPTO,
IMPO_CODI_DDTT, IMPO_DOCU_ORIG, IMPO_FECH_ELAB, REPTO, DIRECC,
CABE_NTPD_TIPO_SOL, DICC_DESC)

as

```
select x2.impo_secu ,x4.cabe_ntpd_anio ,x4.cabe_ntpd_sigl ,x4.cabe_ntpd_ume ,
x4.cabe_ntpd_fech_ela ,x4.cabe_ntpd_codi_mon ,x4.cabe_ntpd_mont_div ,
x4.cabe_ntpd_mont_suc ,x4.cabe_ntpd_refe ,x2.impo_codi_rpto ,x2.impo_codi_ddtt,
x2.impo_docu_orig ,x2.impo_fech_elab ,x0.desc_desc ,x6.desc_desc ,
x4.cabe_ntpd_tipo_sol ,x7.dicc_desc
from rhum_desc x0 , rhum_locd x1 , impm_codi_impo x2 , impt_enlc_docu x3 ,
impm_cabe_ntpd x4 , rhum_locd x5 , rhum_desc x6 , impm_dicc x7
where ((((((((((x0.desc_codi = x1.locd_desc_codi ) AND (x1.locd_codi =
x1.locd_desc_codi ) AND (x1.locd_codi = x2.impo_codi_rpto ) ) AND (x2.impo_secu =
x3.enlc_docu_impo ) ) AND (x3.enlc_docu_anio = x4.cabe_ntpd_anio ) ) AND
(x3.enlc_docu_sigl = x4.cabe_ntpd_sigl ) ) AND (x3.enlc_docu_ume =
x4.cabe_ntpd_ume ) ) AND (x2.impo_codi_ddtt = x5.locd_codi ) ) AND
(x4.cabe_ntpd_tipo_dic = x7.dicc_tipo ) ) AND (x4.cabe_ntpd_tipo_mat =
x7.dicc_codi ) );
with check option;
```

create or replace view ARMADA."pla_anls_cost_unit"(SIGLA, DESC_EQUI,
CODI_TRAB, DESC_TRAB, CODI_MATE, DESC_MATE, UNID_MATE, CANTIDAD,
POTENCIA, VALOR2, VALOR3, CODI_CIU, DESC_CIU, CODI_PROV,
DESC_PROV, CODI_PAIS, DESC_PAIS, CODI_SUBR, DESC_SUBR, SUBR_MEDI,
FORMA_PAGO, PORC_IMPR, PORC_UTIL, PORC_GENE, PORC_FISC, PORC_REND,
OBSE_ANLS, ANLS_OBRA, SUBR_ESPE, ANLS_PORC, ANLS_OBSE_DETA,
POTENCIA)

as

```
select x0.anls_cost_sigl ,x0.anls_desc_equi ,x0.anls_codi_trab ,x1.trab_desc ,
x0.anls_codi_item_mat ,x3.deno_desc || ' ' || x2.item_desc ,x2.item_unid_codi ,
```

```
x14.mayo_subs_codi ) ) AND (x11.plan_mayo_codi = x14.mayo_codi ) ) AND
(x15.comb_kard_prim = x16.kard_tipo ) ) AND (x15.comb_kard_segua =
x17.kard_tipo ) ) AND (x9.inte_esta IN ('E' , 'T' ) ) ) AND
(x9.inte_fech_tran > '01/01/2005' ) ) AND (x10.deta_inte_plan = '2120723000' ) );
```

```
x0.anls_cant_unid ,x0.anls_valo ,x0.anls_valo_dist ,x0.anls_auxi ,x5.ciud_codi ,
from plat_anls_cost x0 ,plam_tipo_trab x1 ,inmm_item x2 , invm_deno x3 ,
where (((((((((((((((x4.anls_codi_ciud = x5.ciud_codi ) AND (x4.anls_codi_pais =
x5.ciud_codi_pais ) ) AND (x4.anls_codi_prov = x5.ciud_codi_esta ) ) AND
(x5.ciud_codi_pais = x7.pais_codi ) ) AND (x6.stad_codi_pais = x7.pais_codi ) ) AND
(x6.stad_codi_pais = x7.pais_codi ) ) AND (x5.ciud_codi_esta = x6.stad_codi ) );
with check option;
```

Create TRIGGERS

INFORMIX

create TRIGGER

```
{BEFORE|AFTER} {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]
[OR {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]. . .}]
into table
[REFERENCING OLD AS oldname, NEW as newname]
[FOR EACH ROW [WHEN (condition)]]
```

Ejemplos:

create trigger trg_cglu_prov

```
before delete on adqm_prov
referencing old as ante
for each row
```

```
insert into rpl_transaccion (id_base_origen,nombre_tabla_origen,
id_base_destino,nombre_tabla_destino,accion,condicion,estado)
```

values

```
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'D' , 'prov_codi=' || ante.prov_codi , 'A' );
```

create trigger trg_cgli_prov

```
before insert or update on adqm_prov
referencing new as post
for each row
```

```
insert into cglm_kard (kard_tipo,kard_codi,kard_desc,kard_stat)
```

values

```
('02' , post.prov_codi, post.prov_nomb , 'A' )
```

```
insert into rpl_transaccion (id_base_origen,nombre_tabla_origen,id_base_destino
```

ORACLE

create [OR REPLACE] TRIGGER

```
{BEFORE|AFTER} {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]
[OR {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]. . .}]
ON table
[REFERENCING OLD AS oldname, NEW as newname]
[FOR EACH ROW [WHEN (condition)]]
```

Ejemplos:

create trigger trg_cglu_prov

```
before delete on adqm_prov referencing old as ante
for each row
```

BEGIN

```
insert into rpl_transaccion (id_base_origen, nombre_tabla_origen,
insert into rpl_transaccion (id_base_origen, nombre_tabla_origen,
id_base_destino,nombre_tabla_destino,accion,condicion,estado)
```

values

```
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'D' , 'prov_codi=' || ante.prov_codi , 'A' );
```

```
END;
```

create trigger trg_cgli_prov

```
after insert on adqm_prov referencing new as post
for each row
```

BEGIN

```
insert into cglm_kard (kard_tipo,kard_codi,kard_desc,kard_stat) values
('02' , post.prov_codi, post.prov_nomb , 'A' ), insert into rpl_transaccion
(id_base_origen,nombre_tabla_origen,id_base_destino,
```

```

,nombre_tabla_destino,accion,condicion,estado)
values
(1,'adqm_prov',3,'adqm_prov','I','prov_codi=' || post.prov_codi,'A');

create trigger trg_cglu_prov
before delete on adqm_prov
referencing old as ante
for each row

insert into rpl_transaccion (id_base_origen,nombre_tabla_origen,id_base_destino,
nombre_tabla_destino,accion,condicion,estado)
values
(1,'adqm_prov',3,'adqm_prov','D','prov_codi=' || ante.prov_codi,'A');

create trigger trga_adqt_cabe_i
after
insert on adqt_cabe_coti

insert into audt_log (log_fech,log_user,log_host,log_sql) select CURRENT year to
fraction(3),x1.username,x1.hostname,x0.scs_sqlstatement
from syssqlcurses x0,syssessions x1
where (((x0.scs_sessionid = DBINFO ('sessionid')) AND
(x0.scs_isolationlevel = 'COMMITTED READ')) AND
(x1.sid = DBINFO ('sessionid')));

create trigger trga_adqt_cabe_u
after
update on adqt_cabe_coti

insert into audt_log (log_fech,log_user,log_host,log_sql) select CURRENT year to
fraction(3),x1.username,x1.hostname,x0.scs_sqlstatement
from syssqlcurses x0,syssessions x1 where (((x0.scs_sessionid =
DBINFO ('sessionid')) AND (x0.scs_isolationlevel = 'COMMITTED READ'))
AND (x1.sid = DBINFO ('sessionid')

```

```

nombre_tabla_destino,accion,condicion,estado) values
value
(1,'adqm_prov',3,'adqm_prov','I','prov_codi=' || post.prov_codi,'A');
END;

create trigger trg_cglu_prov
after update on adqm_prov
BEGIN
insert into rpl_transaccion (id_base_origen, nombre_tabla_origen,
id_base_destino,nombre_tabla_destino,accion,condicion,estado)
values
(1,'adqm_prov',3,'adqm_prov','U','prov_codi=' || post.prov_codi,'A');
END;

create trigger trga_adqt_cabe_i
after insert on adqt_cabe_coti
BEGIN
insert into audt_log (log_fech,log_user,log_host,log_sql)
select CURRENT year to fraction(3),x1.username,x1.hostname,
x0.scs_sqlstatement from syssqlcurses x0,syssessions x1
where (((x0.scs_sessionid = DBINFO ('sessionid')) AND
(x0.scs_isolationlevel = 'COMMITTED READ')) AND
(x1.sid = DBINFO ('sessionid')));
END;

create trigger trga_adqt_cabe_u
after update on adqt_cabe_coti
BEGIN
insert into audt_log (log_fech,log_user,log_host,log_sql)
select CURRENT year to fraction(3),x1.username,x1.hostname,x0.scs_sqlstatement
from syssqlcurses x0,syssessions x1 where (((x0.scs_sessionid = DBINFO
('sessionid')) AND (x0.scs_isolationlevel = 'COMMITTED READ')) AND
(x1.sid = DBINFO ('sessionid')));
END;

```


Create PROCEDURES

INFORMIX

```
CREATE [OR REPLACE] PROCEDURE name [(param [IN|OUT|IN OUT]) datatype) . . .]
[IS|AS] pl/sql_subprogram
```

Ejemplos:

```
create procedure "informix".ins_seg_audi
(apli like seguridad:segt_audi.audi_apli ,menu like seguridad:
segt_audi.audi_menu, prog like seguridad:segt_audi.audi_prog,
opc like seguridad:segt_audi.audi_opcn, tdocu like seguridad:
segt_audi.audi_tipo_docu, ndocu like seguridad:segt_audi.audi_ume_docu,
usuario like seguridad:segt_audi.audi_user,dire like seguridad:segt_audi.audi_dire,
fechaini like seguridad:segt_audi.audi_fech_ini, horaini like seguridad:
segt_audi.audi_hora_ini, tipo like seguridad:segt_audi.audi_tipo_proc,
valor like seguridad:segt_audi.audi_mont_emis)
returning integer;
define numero int;
define sql_err integer;
on exception set sql_err;
return sql_err;
end exception;
select prmt_valr into numero
from SEGURIDAD:SEGP_PRMT_AUDI
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
update SEGURIDAD:SEGP_PRMT_AUDI set prmt_valr = TRIM(PA_IMP_VAL
(prmt_valr + 1) || ' ') where PRMT_APLI_CODI = 'SEG' and PRMT_CODI =
'prmt_audi_secu';
insert into SEGURIDAD:SEGT_AUDI values (numero,apli,menu,prog,opc,fechaini,
today,horaini,current hour to second,tdocu,ndocu,usuario,dire,tipo,valor);
return 0;
end procedure;
```

```
CREATE PROCEDURE "informix".pa_rhu_buscalocd
(Loed like RHUM_LOCD.LOCD_CODI)
RETURNING integer, integer, varchar(100),char(2),char(4), varchar(10);
```

ORACLE

```
CREATE [OR REPLACE] PROCEDURE name [(param [IN|OUT|IN OUT]) datatype) . . .]
[IS|AS] pl/sql_subprogram
```

Ejemplos:

```
create or replace procedure ARMADA.INS_SEG_AUDI(apli in char,
menu in varchar2,prog in varchar2, opc in char,fechaini in date, today in date,
horaini in date, horafin in date, tdocu in char, ndocu in varchar2,usuario in varchar2,
dire in varchar2, tipo in char,valor number)
as
numero number(10);
sql_err number(10);
dmens_error varchar2(1024);
dmens_default varchar2(256);
contador_no_permitido EXCEPTION;
datos_no_encontrado exception;

begin
select prmt_valr into numero
from SEGP_PRMT_AUDI
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
update SEGP_PRMT_AUDI set prmt_valr = TRIM((prmt_valr + 1) || ' ')
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
update SEGP_PRMT_AUDI set prmt_valr = TRIM((prmt_valr + 1) || ' ')
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
insert into SEGT_AUDI values (numero,apli,menu,prog,opc,fechaini,today,horaini,
horafin,tdocu,ndocu,usuario,dire,tipo,valor);
EXCEPTION
WHEN contador_no_permitido THEN
sql_err :=-20001;
WHEN datos_no_encontrado THEN
sql_err := -20002;
dmens_default := 'NO EXISTEN REGISTROS EN LA TABLA';
dmens_error := TRAE_ERROR (sql_err, dmens_default);
RAISE_APPLICATION_ERROR (-20999, dmens_error);
end;
```

```

DEFINE V_Locd_Codi like RHUM_LOCD.LOCD_CODI;
DEFINE V_Locd_Desc like RHUM_DESC.DESC_DESC;
DEFINE V_Locd_Clave like RHUM_LOCD.LOCD_CLAV_ALTE;
DEFINE V_SqlErr integer;
DEFINE V_Errlsam integer;
DEFINE V_DescErr varchar(100);
DEFINE V_TipoErr char(2);
DEFINE V_PadErr char(4);
DEFINE V_sigla varchar(10);

ON EXCEPTION SET V_SqlErr, V_Errlsam, V_DescErr
RETURN V_SqlErr, V_Errlsam, V_DescErr,V_TipoErr,V_PadErr, V_sigla;
END EXCEPTION;

SELECT locd_codi, desc_desc, locd_clav_alte,locd_tipo_codi,locd_tipo_rela, desc_sigl
INTO V_Locd_Codi, V_Locd_Desc, V_Locd_Clave,V_TipoErr,V_PadErr, V_sigla
FROM rhum_locd, rhum_desc
WHERE locd_codi = Locd AND locd_desc_codi = desc_codi
AND locd_stat = 'A';

if V_Locd_Codi is null then
    return 0, 1, "Codigo de localidad no existe","00","0000", "";
end if;

if V_Locd_Clave is not null then
    if V_Locd_Clave = V_Locd_Codi then
        return 0, 0, V_Locd_Desc,V_TipoErr,V_PadErr, V_sigla;
    else
        return 0, 1,"Codigo de localidad no valido","00","0000", "";
    end if;
else
    return 0, 0, V_Locd_Desc,V_TipoErr,V_PadErr, V_sigla;
end if;
END PROCEDURE;

```

```

create or replace procedure ARMADA.PA_RHU_BUSCALOCD(Locd in char)

```

```

as

```

```

    V_Locd_Codi char(4);
    V_Locd_Desc varchar2(70);
    V_Locd_Clave char(4);
    V_TipoErr char(2);
    V_PadErr char (4);
    V_sigla varchar2 (10);
    cod_err NUMBER(10);
    contador_no_permitido EXCEPTION;
    dmens_error varchar2(1024);
    dmens_default varchar2(256);
    datos_no_encontrado exception;

```

```

begin

```

```

SELECT locd_codi, desc_desc, locd_clav_alte,locd_tipo_codi,locd_tipo_rela, desc_sigl
INTO V_Locd_Codi, V_Locd_Desc, V_Locd_Clave,V_TipoErr,V_PadErr, V_sigla
FROM rhum_locd, rhum_desc
WHERE locd_codi = Locd AND locd_desc_codi = desc_codi AND locd_stat = 'A';
EXCEPTION
WHEN contador_no_permitido THEN cod_err :=-20001;
WHEN datos_no_encontrado THEN cod_err := -20002;
dmens_DEFAULT := 'No existen Registros en la Tabla';
dmens_error := TRAE_ERROR (cod_err, dmens_DEFAULT);
RAISE_APPLICATION_ERROR (-20999, dmens_ERROR);
if V_Locd_Codi=null then
    cod_err := -20003;
    dmens_DEFAULT := 'Codigo de localidad no existe';
    dmens_error := TRAE_ERROR (cod_err, dmens_DEFAULT);
    RAISE_APPLICATION_ERROR (-20999, dmens_ERROR);
end if;
if V_Locd_Clave <> V_Locd_Codi then
    cod_err := -20004;
    dmens_DEFAULT := 'Codigo de localidad no valido';
    dmens_error := TRAE_ERROR (cod_err, dmens_DEFAULT);
    RAISE_APPLICATION_ERROR (-20999, dmens_ERROR);

```

```
create procedure "informix".contar_reg
  (tabla varchar(50))
  returning int, int;
  define cod_err int;
  define num_fila int;
  ON EXCEPTION SET cod_err
  RETURN cod_err, 0;
  END EXCEPTION
  SELECT COUNT(*) INTO num_fila
  FROM tabla;
  return 0, num_fila;

end procedure
```

```
end if;
end;
```

```
create or replace procedure ARMADA.CONTAR_REG(tabla in VARCHAR2)
as
```

```
        cod_err NUMBER(10);
        num_fila NUMBER(10);
        contador_no_permitido EXCEPTION;
        dmens_error varchar2(1024);
        dmens_default varchar2(256);
        datos_no_encontrado exception;
```

```
begin
```

```
        SELECT COUNT(*) INTO num_fila FROM tab_col;
```

```
EXCEPTION
```

```
WHEN contador_no_permitido THEN
```

```
cod_err := -20001;
```

```
WHEN datos_no_encontrado THEN
```

```
cod_err := -20002;
```

```
dMENS_DEFAULT := 'NO EXISTEN REGISTROS EN LA TABLA';
```

```
dmens_error := TRAE_ERROR (cod_err, dMENS_DEFAULT);
```

```
RAISE_APPLICATION_ERROR (-20999, dmens_error);
```

```
end;
```

;

ANALISIS COMPARATIVO DDL

INFORMIX

ORACLE

ALTER TABLE

Sintaxis:

alter table nombreTabla **ADD**(nombreColumna TipoDatos [Propiedades]
[, columnaSiguiente tipoDatos [propiedades]]

alter table nombreTabla **Drop**(columna)

alter table nombreTabla **Modify** (columna tipo [propiedades] [columnaSiguiente
tipo [propiedades]])

Ejemplos:

alter table INFORMIX.ADQM_PORC_SEGU
drop primary key cascade;

alter table INFORMIX.ADQM_PROV
drop primary key cascade;

alter table INFORMIX.ADQM_PROV_DOCU
drop primary key cascade;

alter table INFORMIX.ADQM_PROV_REFE
drop primary key cascade;

alter table INFORMIX.ADQM_PROV_REPR
drop primary key cascade;

Sintaxis:

alter table nombreTabla **ADD**(nombreColumna TipoDatos [Propiedades]
[, columnaSiguiente tipoDatos [propiedades]]

alter table nombreTabla **Drop**(columna)

alter table nombreTabla **Modify** (columna tipo [propiedades] [columnaSiguiente
tipo [propiedades]])

Ejemplos:

alter table ARMADA.ADQM_PORC_SEGU
drop primary key cascade;

alter table ARMADA.ADQM_PROV
drop primary key cascade;

alter table ARMADA.ADQM_PROV_DOCU
drop primary key cascade;

alter table ARMADA.ADQM_PROV_REFE
drop primary key cascade;

alter table ARMADA.ADQM_PROV_REPR
drop primary key cascade;

ALTER VISTAS

INFORMIX

ORACLE

ALTER view nombre-Vista [(Lista-Columnas)]

as

select-Sentencia [with check option]

create or replace view [esquema.] nombreDeVista

as

select [nombre de atributos]

Ejemplos:

```

ALTER view INFORMIX."prev_proyecto"({
"proyecto", "proy_denominacion", "proy_abreviacion", "rne",
"proy_rep_control_tecnico", "proy_rep_control_administrativ", "proy_tipo_prioridad",
"proy_numero_prioridad", "proy_fecha_inicio", "proy_fecha_fin", "loca_codigo",
PROY_FECH_FINA, PROY_CODI_LOCD)
as
select x0.proy_plan_basi, x0.proy_codi_prog, x0.proy_codi_tipo,
x0.proy_ume_secc, x0.proy_deno_proy, x0.proy_abre_deno, x3.desc_sigl,
x5.desc_sigl, x7.desc_sigl, x0.proy_prio_proy [1,1], x0.proy_prio_proy [2,3],
x0.proy_fech_inic, x0.proy_fech_fina, x0.proy_codi_locd
from prem_proy_pres x0, segp_prmt x1, rhum_locd x2, rhum_desc x3,
rhum_locd x4, rhum_desc x5, rhum_locd x6, rhum_desc x7
where (((((((x0.proy_anio_ejec = x1.prmt_valr ) AND (x0.proy_esta_proy = 'A' ) )
AND (x1.prmt_codi LIKE 'prmt_anio_ejec' ) ) AND (x2.locd_codi =
x0.proy_codi_dire ) ) AND (x3.desc_codi = x2.locd_desc_codi ) ) AND
(x4.locd_codi = x0.proy_codi_rct ) ) AND (x5.desc_codi =
x4.locd_desc_codi ) ) AND (x6.locd_codi = x0.proy_codi_rca ) ) AND
(x7.desc_codi = x6.locd_desc_codi ) ) ;

```

```

ALTER view INFORMIX."vw_cxp_ctas_cons"("plan_codi", "tipo_kard",
"codi_kard", "codi_skar", "total", "desc", "tipo_docu", "nume_docu", "ctpg_secc",
"fecha", "estado", "PLAN_NOME_CORT", "MAYO_DESC", "KARD_DESC",
"kard_desc2", "deha")
as
select x0.cons_plan_codi, x0.cons_tipo_kard, x0.cons_codi_kard,
x0.cons_codi_skar, x0.cons_tota, x0.cons_desc, x0.cons_tipo_docu,
x0.cons_ume_docu, x0.cons_ctpg_secc, x0.cons_fech, x0.cons_stat,
x1.plan_nome_cort, x4.mayo_desc, x6.kard_desc, x7.kard_desc, 'D'
from cxpt_deta_cons x0, cglm_plan_ctas x1, cglm_grup x2, cglm_subs_grup x3,
cglm_mayo_gene x4, cgl_t_comb_kard x5, cglm_kard x6, cglm_kard x7
where (((((((((((x0.cons_plan_codi = x1.plan_codi ) AND (x0.cons_tipo_kard =
x5.comb_kard_codi ) ) AND (x0.cons_codi_kard = x6.kard_codi ) ) AND
(x0.cons_codi_skar = x7.kard_codi ) ) AND (x1.plan_grup_codi = x2.grup_codi ) )
AND (x1.plan_grup_codi = x3.subs_grup_codi ) ) AND (x1.plan_subs_codi =
x3.subs_codi ) ) AND (x1.plan_grup_codi = x4.mayo_grup_codi ) )

```

```

from [nombre de las tablas]
where [las condiciones]

```

with check option;

Ejemplos:

```

ALTER view ARMADA."cgla_sald_cont2"
as
select x0.sald_ano, x0.sald_mes, x0.sald_plan_codi,
x0.sald_plan_fond, x0.sald_mont_debe, x0.sald_mont_habe
from cgla_sald_cont x0
with check option;

```

```

ALTER view ARMADA."cgla_sald_cont3"
as
select x0.sald_ano, x0.sald_mes, x0.sald_plan_codi,
x0.sald_plan_fond, x0.sald_mont_debe, x0.sald_mont_habe
from cgla_sald_cont x0
with check option;

```

```

ALTER or replace view ARMADA."cglv_depo_cobr2"
as
select x2.caib_anio, x2.caib_codi, x1.caib_depo_secu,
x0.caib_caic_anio_icc, x0.caib_caic_codi_icc
from cgl_t_caib_caic x0, cgl_t_caib_depo x1, cgl_t_caib x2
where (x2.caib_tipo_ibco = '01' )
with check option;

```

```

ALTER view ARMADA."cons_ntpd_per"(IMPO_SECU,
CABE_NTPD_ANIO, CABE_NTPD_SIGL, CABE_NTPD_NUME,
CABE_NTPD_FECH_ELA, CABE_NTPD_CODI_MON, CABE_NTPD_MONT_DIV,
CABE_NTPD_MONT_SUC, CABE_NTPD_REFE, IMPO_CODI_RPTO,
IMPO_CODI_DDTT, IMPO_DOCU_ORIG, IMPO_FECH_ELAB, REPTO, DIRECC,
CABE_NTPD_TIPO_SOL, DICC_DESC)
as
select x2.impo_secu, x4.cabe_ntpd_anio, x4.cabe_ntpd_sigl, x4.cabe_ntpd_nume,
x4.cabe_ntpd_fech_ela, x4.cabe_ntpd_codi_mon, x4.cabe_ntpd_mont_div,
x4.cabe_ntpd_mont_suc, x4.cabe_ntpd_refe, x2.impo_codi_rpto, x2.impo_codi_ddtt,

```

```

AND (x1.plan_subs_codi = x4.mayo_subs_codi ) ) AND
(x1.plan_mayo_codi = x4.mayo_codi ) ) AND (x5.comb_kard_prim =
x6.kard_tipo ) ) AND (x5.comb_kard_segua =x7.kard_tipo ) )
union all select x8.cont_plan_codi ,x8.cont_tipo_kard ,x8.cont_codi_kard ,x
8.cont_codi_skar ,x8.cont_tota ,x8.cabe_glos ,x8.cabe_docu_codi , x8.cabe_codi ,
0',x8.cabe_fech_tran ,x8.cabe_stat ,x8.plan_nome_cort ,x8.mayo_desc ,
x8.kard_desc1 ,x8.kard_desc2 ,x8.cont_deha
from vw_cgl_comp x8
where ((x8.cabe_stat != 'E' ) AND (x8.cont_plan_codi = '2120723000' ) )
union all select x10.deta_inte_plan ,x10.deta_inte_tkar ,x10.deta_inte_kard ,
x10.deta_inte_skar ,x10.deta_inte_tota ,x9.inte_glos ,x9.inte_tipo_docu ,
x9.inte_codi_orig ,'0' ,x9.inte_fech_tran ,x9.inte_stat ,x11.plan_nome_cort ,
x14.mayo_desc ,x16.kard_desc ,x17.kard_desc ,x10.deta_inte_deha
from cglm_cabe_inte x9 , cglm_deta_inte x10 , cglm_plan_ctas x11 , cglm_grup x12 ,
cglm_subs_grup x13 , cglm_mayo_gene x14 , cglm_comb_kard x15 ,
cglm_kard x16 , cglm_kard x17
where (((((((((((((((((x9.inte_nume_tran = x10.deta_inte_ntra ) AND
(x10.deta_inte_plan = x11.plan_codi ) ) AND (x10.deta_inte_tkar =
x15.comb_kard_codi ) ) AND (x10.deta_inte_kard = x16.kard_codi ) ) AND
(x10.deta_inte_skar = x17.kard_codi ) ) AND (x10.deta_inte_stat = 'A' ) ) AND
(x11.plan_grup_codi = x12.grup_codi ) ) AND (x11.plan_grup_codi =
x13.subs_grup_codi ) ) AND (x11.plan_subs_codi = x13.subs_codi ) ) AND
(x11.plan_grup_codi = x14.mayo_grup_codi ) ) AND (x11.plan_subs_codi =
x14.mayo_subs_codi ) ) AND (x11.plan_mayo_codi = x14.mayo_codi ) ) AND
(x15.comb_kard_prim = x16.kard_tipo ) ) AND (x15.comb_kard_segua =
x17.kard_tipo ) ) AND (x9.inte_esta IN ('E' , 'T' ) ) AND
(x9.inte_fech_tran > '01/01/2005' ) ) AND (x10.deta_inte_plan = '2120723000' ) ) ;

```

```

x2.impo_docu_orig ,x2.impo_fech_elab ,x0.desc_desc ,x6.desc_desc ,
x4.cabe_ntpd_tipo_sol ,x7.dicc_desc
from rhum_desc x0 , rhum_locd x1 , impm_codi_impo x2 , impt_enlc_docu x3 ,
impm_cabe_ntpd x4 , rhum_locd x5 , rhum_desc x6 , impm_dicc x7
where (((((((((((((((((x0.desc_codi = x1.locd_desc_codi ) AND (x1.locd_codi =
x1.locd_desc_codi ) AND (x1.locd_codi = x2.impo_codi_rpto ) ) AND (x2.impo_secu =
x3.enlc_docu_impo ) ) AND (x3.enlc_docu_anio = x4.cabe_ntpd_anio ) ) AND
(x3.enlc_docu_sigl = x4.cabe_ntpd_sigl ) ) AND (x3.enlc_docu_nume =
x4.cabe_ntpd_nume ) ) AND (x2.impo_codi_ddtt = x5.locd_codi ) ) AND
(x4.cabe_ntpd_tipo_dic = x7.dicc_tipo ) ) AND (x4.cabe_ntpd_tipo_mat =
x7.dicc_codi ) ) );
with check option;

```

```

create or replace view ARMADA."pla_anls_cost_unit"(SIGLA, DESC_EQUI,
CODI_TRAB, DESC_TRAB, CODI_MATE, DESC_MATE, UNID_MATE, CANTIDAD,
POTENCIA, VALOR2, VALOR3, CODI_CIU, DESC_CIU, CODI_PROV,
DESC_PROV, CODI_PAIS, DESC_PAIS, CODI_SUBR, DESC_SUBR, SUBR_MEDI,
FORMA_PAGO, PORC_IMPR, PORC_UTIL, PORC_GENE, PORC_FISC, PORC_REND,
OBSE_ANLS, ANLS_OBRA, SUBR_ESPE, ANLS_PORC, ANLS_OBSE_DETA,
POTENCIA)
as
select x0.anls_cost_sigl ,x0.anls_desc_equi ,x0.anls_codi_trab ,x1.trab_desc ,
x0.anls_codi_item_mat ,x3.deno_desc || ' ' || x2.item_desc ,x2.item_unid_codi ,
x0.anls_cant_unid ,x0.anls_valo ,x0.anls_valo_dist ,x0.anls_auxi ,x5.ciud_codi ,
from plat_anls_cost x0 ,plam_tipo_trab x1 ,inmm_item x2 , invm_deno x3 ,
where (((((((((((((((((x4.anls_codi_ciud = x5.ciud_codi ) AND (x4.anls_codi_pais =
x5.ciud_codi_pais ) ) AND (x4.anls_codi_prov = x5.ciud_codi_esta ) ) AND
(x5.ciud_codi_pais = x7.pais_codi ) ) AND (x6.stad_codi_pais = x7.pais_codi ) ) AND
(x6.stad_codi_pais = x7.pais_codi ) ) AND (x5.ciud_codi_esta = x6.stad_codi ) ) );
with check option;

```

ALTER TRIGGERS

INFORMIX

ALTER TRIGGER

```
{BEFORE|AFTER} {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]}
```

ORACLE

ALTER [OR REPLACE] TRIGGER

```
{BEFORE|AFTER} {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]}
```

```
[OR {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]. . .}]  
INTO table  
[REFERENCING OLD AS oldname, NEW as newname]  
[FOR EACH ROW [WHEN (condition)]]
```

Ejemplos:

ALTER trigger trg_cglu_provd

```
before delete on adqm_prov  
referencing old as ante  
for each row
```

```
insert into rpl_transaccion (id_base_origen,nombre_tabla_origen,  
id_base_destino,nombre_tabla_destino,accion,condicion,estado)  
values  
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'D' , 'prov_codi=' || ante.prov_codi , 'A' );
```

ALTER trigger trg_cgli_prov

```
before insert or update on adqm_prov  
referencing new as post  
for each row
```

```
insert into cglm_kard (kard_tipo,kard_codi,kard_desc,kard_stat)  
values  
( '02' , post.prov_codi, post.prov_nomb , 'A' )  
insert into rpl_transaccion (id_base_origen,nombre_tabla_origen,id_base_destino  
,nombre_tabla_destino,accion,condicion,estado)  
values  
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'I' , 'prov_codi=' post.prov_codi , 'A' );
```

ALTER trigger trg_cglu_provd

```
before delete on adqm_prov  
referencing old as ante  
for each row
```

```
insert into rpl_transaccion (id_base_origen,nombre_tabla_origen,id_base_destino,  
nombre_tabla_destino,accion,condicion,estado)  
values  
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'D' , 'prov_codi=' || ante.prov_codi , 'A' );
```

```
[OR {DELETE|INSERT|UPDATE [OF col1, col2, . . . , colN]. . .}]  
ON table  
[REFERENCING OLD AS oldname, NEW as newname]  
[FOR EACH ROW [WHEN (condition)]]
```

Ejemplos:

ALTER trigger trg_cglu_provd

```
before delete on adqm_prov referencing old as ante  
for each row
```

BEGIN

```
insert into rpl_transaccion (id_base_origen, nombre_tabla_origen,  
insert into rpl_transaccion (id_base_origen, nombre_tabla_origen,  
id_base_destino,nombre_tabla_destino,accion,condicion,estado)  
values  
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'D' , 'prov_codi=' || ante.prov_codi , 'A' );  
END;
```

ALTER trigger trg_cgli_prov

```
after insert on adqm_prov referencing new as post  
for each row
```

BEGIN

```
insert into cglm_kard (kard_tipo,kard_codi,kard_desc,kard_stat) values  
( '02' , post.prov_codi, post.prov_nomb , 'A' ), insert into rpl_transaccion  
(id_base_origen,nombre_tabla_origen,id_base_destino,  
nombre_tabla_destino,accion,condicion,estado) values  
value  
(1 , 'adqm_prov',3 , 'adqm_prov' , 'I' , 'prov_codi=' post.prov_codi , 'A' );  
END;
```

ALTER trigger trg_cglu_provd

```
after update on adqm_prov
```

BEGIN

```
insert into rpl_transaccion (id_base_origen, nombre_tabla_origen,  
id_base_destino,nombre_tabla_destino,accion,condicion,estado)  
values  
(1 , 'adqm_prov' ,3 , 'adqm_prov' , 'U' , 'prov_codi=' || post.prov_codi , 'A' );  
END;
```

```

ALTER trigger trga_adqt_cabe_i
after
insert on adqt_cabe_coti

insert into audt_log (log_fech,log_user,log_host,log_sql) select CURRENT year to
fraction(3) ,x1.username ,x1.hostname ,x0.scs_sqlstatement
from syssqlcurses x0 ,sysessions x1
where (((x0.scs_sessionid = DBINFO ('sessionid') ) AND
(x0.scs_isolationlevel = 'COMMITTED READ' ) ) AND
(x1.sid = DBINFO ('sessionid') ) );

```

```

ALTER trigger trga_adqt_cabe_u
after
update on adqt_cabe_coti

insert into audt_log (log_fech,log_user,log_host,log_sql) select CURRENT year to
fraction(3) ,x1.username ,x1.hostname ,x0.scs_sqlstatement
from syssqlcurses x0 ,sysessions x1 where (((x0.scs_sessionid =
DBINFO ('sessionid') ) AND (x0.scs_isolationlevel = 'COMMITTED READ' ) )
AND (x1.sid = DBINFO ('sessionid'

```

```

ALTER trigger trga_adqt_cabe_i
after insert on adqt_cabe_coti
BEGIN
insert into audt_log (log_fech,log_user,log_host,log_sql)
select CURRENT year to fraction(3) ,x1.username ,x1.hostname ,
x0.scs_sqlstatement from syssqlcurses x0 ,sysessions x1
where (((x0.scs_sessionid = DBINFO ('sessionid') ) AND
(x0.scs_isolationlevel = 'COMMITTED READ' ) ) AND
(x1.sid = DBINFO ('sessionid') ) );
END;

```

```

ALTER trigger trga_adqt_cabe_u
after update on adqt_cabe_coti
BEGIN
insert into audt_log (log_fech,log_user,log_host,log_sql)
select CURRENT year to fraction(3) ,x1.username ,x1.hostname ,x0.scs_sqlstatement
from syssqlcurses x0 ,sysessions x1 where (((x0.scs_sessionid = DBINFO
('sessionid') ) AND (x0.scs_isolationlevel = 'COMMITTED READ' ) ) AND
(x1.sid = DBINFO ('sessionid') ) );
END;

```

ALTER PROCEDURE

INFORMIX

```

ALTER [OR REPLACE] PROCEDURE name [(param [IN|OUT|IN OUT]] datatype) . . .]
[IS|AS] pl/sql_subprogram

```

Ejemplos:

```

ALTER procedure "informix".ins_seg_audi
(apli like seguridad:segt_audi.audi_apli ,menu like seguridad:
segt_audi.audi_menu, prog like seguridad:segt_audi.audi_prog,
opc like seguridad:segt_audi.audi_opcn, tdocu like seguridad:
segt_audi.audi_tipo_docu, ndocu like seguridad:segt_audi.audi_ume_docu,
usuario like seguridad:segt_audi.audi_user,dire like seguridad:segt_audi.audi_dire,

```

ORACLE

```

ALTER [OR REPLACE] PROCEDURE name [(param [IN|OUT|IN OUT]] datatype) . . .]
[IS|AS] pl/sql_subprogram

```

Ejemplos:

```

ALTER procedure ARMADA.INS_SEG_AUDI(apli in char,
menu in varchar2,prog in varchar2, opc in char,fechaini in date, today in date,
horaini in date, horafin in date, tdocu in char, ndocu in varchar2,usuario in varchar2,
dire in varchar2, tipo in char,valor number)
as
numero number(10);

```



```

fechaini like seguridad:segt_audi.audi_fech_ini, horaini like seguridad:
segt_audi.audi_hora_ini, tipo like seguridad:segt_audi.audi_tipo_proc,
valor like seguridad:segt_audi.audi_mont_emis)
returning integer;
define numero int
define sql_err integer;
on exception set sql_err
return sql_err;
end exception;
select prmt_valr into numero
from SEGURIDAD:SEGP_PRMT_AUDI
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
update SEGURIDAD:SEGP_PRMT_AUDI set prmt_valr = TRIM(PA_IMP_VAL
(prmt_valr + 1) || ' ') where PRMT_APLI_CODI = 'SEG' and PRMT_CODI =
'prmt_audi_secu';
insert into SEGURIDAD:SEGT_AUDI values (numero,apli,menu,prog,opc,fechaini,
today,horaini,current hour to second,tdocu,ndocu,usuario,dire,tipo,valor);
return 0;
end procedure;

```

ALTER PROCEDURE "informix".pa_rhu_buscalocd

(Locd like RHUM_LOCD.LOCD_CODI)

RETURNING integer, integer, varchar(100),char(2),char(4), varchar(10);

```

DEFINE V_Locd_Codi like RHUM_LOCD.LOCD_CODI;
DEFINE V_Locd_Desc like RHUM_DESC.DESC_DESC;
DEFINE V_Locd_Clave like RHUM_LOCD.LOCD_CLAV_ALTE;
DEFINE V_SqlErr integer;
DEFINE V_Errlsam integer;
DEFINE V_DescErr varchar(100);
DEFINE V_TipoErr char(2);
DEFINE V_PadErr char(4);
DEFINE V_sigla varchar(10);

```

ON EXCEPTION SET V_SqlErr, V_Errlsam, V_DescErr

RETURN V_SqlErr, V_Errlsam, V_DescErr,V_TipoErr,V_PadErr, V_sigla;

```

sql_err number(10);
dmens_error varchar2(1024);
dmens_default varchar2(256);
contador_no_permitido EXCEPTION;
datos_no_encontrado exception;

```

begin

```

select prmt_valr into numero
from SEGP_PRMT_AUDI
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
update SEGP_PRMT_AUDI set prmt_valr = TRIM((prmt_valr + 1) || ' ')
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
update SEGP_PRMT_AUDI set prmt_valr = TRIM((prmt_valr + 1) || ' ')
where PRMT_APLI_CODI = 'SEG' and PRMT_CODI = 'prmt_audi_secu';
insert into SEGT_AUDI values (numero,apli,menu,prog,opc,fechaini,today,horaini,
horafin,tdocu,ndocu,usuario,dire,tipo,valor);

```

EXCEPTION

```

WHEN contador_no_permitido THEN
sql_err := -20001;
WHEN datos_no_encontrado THEN
sql_err := -20002;
dmens_default := 'NO EXISTEN REGISTROS EN LA TABLA';
dmens_error := TRAE_ERROR (sql_err, dmens_default);
RAISE_APPLICATION_ERROR (-20999, dmens_error);
end;

```

ALTER ARMADA.PA_RHU_BUSCALOCD(Locd in char)

as

```

V_Locd_Codi char(4);
V_Locd_Desc varchar2(70);
V_Locd_Clave char(4);
V_TipoErr char(2);
V_PadErr char (4);
V_sigla varchar2 (10);
cod_err NUMBER(10);
contador_no_permitido EXCEPTION;
dmens_error varchar2(1024);

```

```
END EXCEPTION;
```

```
SELECT locd_codi, desc_desc, locd_clav_alte,locd_tipo_codi,locd_tipo_rela, desc_sigla  
INTO V_Locd_Codi, V_Locd_Desc, V_Locd_Clave,V_TipoErr,V_PadErr, V_sigla  
FROM rhum_locd, rhum_desc  
WHERE locd_codi = Locd AND locd_desc_codi = desc_codi  
AND locd_stat = 'A';
```

```
if V_Locd_Codi is null then
```

```
    return 0, 1, "Codigo de localidad no existe","00","0000", "";
```

```
end if;
```

```
if V_Locd_Clave is not null then
```

```
    if V_Locd_Clave = V_Locd_Codi then
```

```
        return 0, 0, V_Locd_Desc,V_TipoErr,V_PadErr, V_sigla;
```

```
    else
```

```
        return 0, 1,"Codigo de localidad no valido","00","0000", "";
```

```
    end if;
```

```
else
```

```
    return 0, 0, V_Locd_Desc,V_TipoErr,V_PadErr, V_sigla;
```

```
end if;
```

```
END PROCEDURE;
```

```
ALTER procedure "informix".contar_reg
```

```
    (tabla varchar(50))
```

```
    returning int, int;
```

```
    define cod_err int;
```

```
    define num_fila int;
```

```
    ON EXCEPTION SET cod_err
```

```
    RETURN cod_err, 0;
```

```
    END EXCEPTION
```

```
    SELECT COUNT(*) INTO num_fila
```

```
    FROM tabla;
```

```
    return 0, num_fila;
```

```
end procedure;
```

```
dmens_default varchar2(256);
```

```
datos_no_encontrado exception;
```

```
begin
```

```
SELECT locd_codi, desc_desc, locd_clav_alte,locd_tipo_codi,locd_tipo_rela, desc_sigla  
INTO V_Locd_Codi, V_Locd_Desc, V_Locd_Clave,V_TipoErr,V_PadErr, V_sigla  
FROM rhum_locd, rhum_desc
```

```
WHERE locd_codi = Locd AND locd_desc_codi = desc_codi AND locd_stat = 'A';
```

```
EXCEPTION
```

```
WHEN contador_no_permitido THEN cod_err :=-20001;
```

```
WHEN datos_no_encontrado THEN cod_err := -20002;
```

```
dMENS_DEFAULT := 'No existen Registros en la Tabla';
```

```
dmens_error := TRAE_ERROR (cod_err, dMENS_DEFAULT);
```

```
RAISE_APPLICATION_ERROR (-20999, dMENS_ERROR);
```

```
if V_Locd_Codi=null then
```

```
    cod_err := -20003;
```

```
    dMENS_DEFAULT := 'Codigo de localidad no existe';
```

```
    dmens_error := TRAE_ERROR (cod_err, dMENS_DEFAULT);
```

```
    RAISE_APPLICATION_ERROR (-20999, dMENS_ERROR);
```

```
end if;
```

```
if V_Locd_Clave <> V_Locd_Codi then
```

```
    cod_err := -20004;
```

```
    dMENS_DEFAULT := 'Codigo de localidad no valido';
```

```
    dmens_error := TRAE_ERROR (cod_err, dMENS_DEFAULT);
```

```
    RAISE_APPLICATION_ERROR (-20999, dMENS_ERROR);
```

```
end if;
```

```
end;
```

```
ALTER procedure ARMADA.CONTAR_REG(tabla in VARCHAR2)
```

```
as
```

```
    cod_err NUMBER(10);
```

```
    num_fila NUMBER(10);
```

```
    contador_no_permitido EXCEPTION;
```

```
    dmens_error varchar2(1024);
```

```
    dmens_default varchar2(256);
```

```
    datos_no_encontrado exception;
```

```
begin
```

```
SELECT COUNT(*) INTO num_fila FROM tab_col;
```

EXCEPTION

```
WHEN contador_no_permitido THEN
```

```
cod_err := -20001;
```

```
WHEN datos_no_encontrado THEN
```

```
cod_err := -20002;
```

```
dMENS_DEFAULT := 'NO EXISTEN REGISTROS EN LA TABLA';
```

```
dmens_error := TRAE_ERROR (cod_err, dMENS_DEFAULT);
```

```
RAISE_APPLICATION_ERROR (-20999, dmens_error);
```

```
end;
```

ANALISIS COMPARATIVO DDL

INFORMIX

ORACLE

DROP TABLE

Sintaxis:

drop table nombreTabla;

Ejemplos tomados del proceso de migración:

drop table INFORMIX.ADQM_CONC_EVAL cascade constraints;

drop table INFORMIX.ADQM_PORC_SEGU cascade constraints;

drop table INFORMIX.ADQM_PROV cascade constraints;

drop table INFORMIX.ADQM_PROV_DOCU cascade constraints;

drop table INFOMIX.ADQM_PROV_REFE cascade constraints;

Sintaxis:

drop table nombreTabla;

Ejemplos tomados del proceso de migración:

drop table ARMADA.ADQM_CONC_EVAL cascade constraints;

drop table ARMADA.ADQM_PORC_SEGU cascade constraints;

drop table ARMADA.ADQM_PROV cascade constraints;

drop table ARMADA.ADQM_PROV_DOCU cascade constraints;

drop table ARMADA.ADQM_PROV_REFE cascade constraints;

DROP VISTA

Sintaxis:

drop view BaseDatos.nombreVista;

Ejemplos tomados del proceso de migración:

drop view INFORMIX."pla_anls_cost_unit";

drop view INFORMIX."cons_ntpd_per";

drop view INFORMIX."cglv_depo_cobr2";

drop view INFORMIX."cgla_sald_cont3";

drop view INFORMIX."cgla_sald_cont2";

Sintaxis:

drop view BaseDatos.nombreVista;

Ejemplos tomados del proceso de migración:

drop view ARMADA."pla_anls_cost_unit";

drop view ARMADA."cons_ntpd_per";

drop view ARMADA."cglv_depo_cobr2";

drop view ARMADA."cgla_sald_cont3";

drop view ARMADA."cgla_sald_cont2";

DROP TRIGGERS

Sintaxis:

drop trigger BaseDatos.nombreTriggers;

Ejemplos tomados del proceso de migración:

drop trigger INFORMIX.TRG_CGLU_PROVD;

drop trigger INFORMIX.TRG_CGLI_PROV;

drop trigger INFORMIX.TRG_CGLU_PROV;

drop trigger INFORMIX.TRGA_ADQT_CABE_I;

drop trigger INFORMIX.TRGA_ADQT_CABE_U;

Sintaxis:

drop trigger BaseDatos.nombreTriggers;

Ejemplos tomados del proceso de migración:

drop trigger ARMADA.TRG_CGLU_PROVD;

drop trigger ARMADA.TRG_CGLI_PROV;

drop trigger ARMADA.TRG_CGLU_PROV;

drop trigger ARMADA.TRGA_ADQT_CABE_I;

drop trigger ARMADA.TRGA_ADQT_CABE_U;

DROP PROCEDIMIENTOS

Sintaxis:

drop procedure BaseDatos.nombreProcedimiento;

Ejemplos tomados del proceso de migración:

drop procedure INFORMIX.PA_RHU_LOCD_PADRE;

drop procedure INFORMIX.PA_RHU_LOCD_HIJA;

drop procedure INFORMIX.PA_RHU_BUSCALOCD;

drop procedure INFORMIX.INS_SEG_AUDI;

drop procedure INFORMIX.CONTAR_REG;

Sintaxis:

drop procedure BaseDatos.nombreProcedimiento;

Ejemplos tomados del proceso de migración:

drop procedure ARMADA.PA_RHU_LOCD_PADRE;

drop procedure ARMADA.PA_RHU_LOCD_HIJA;

drop procedure ARMADA.PA_RHU_BUSCALOCD;

drop procedure ARMADA.INS_SEG_AUDI;

drop procedure ARMADA.CONTAR_REG;

ANEXO E



MANUAL DE MIGRACIÓN

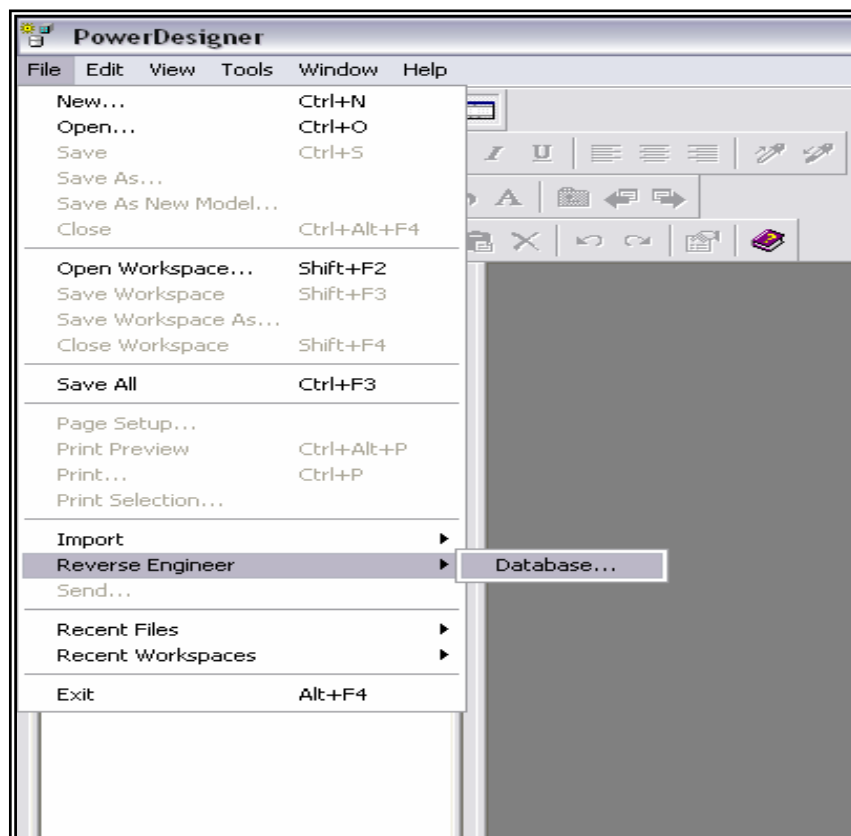




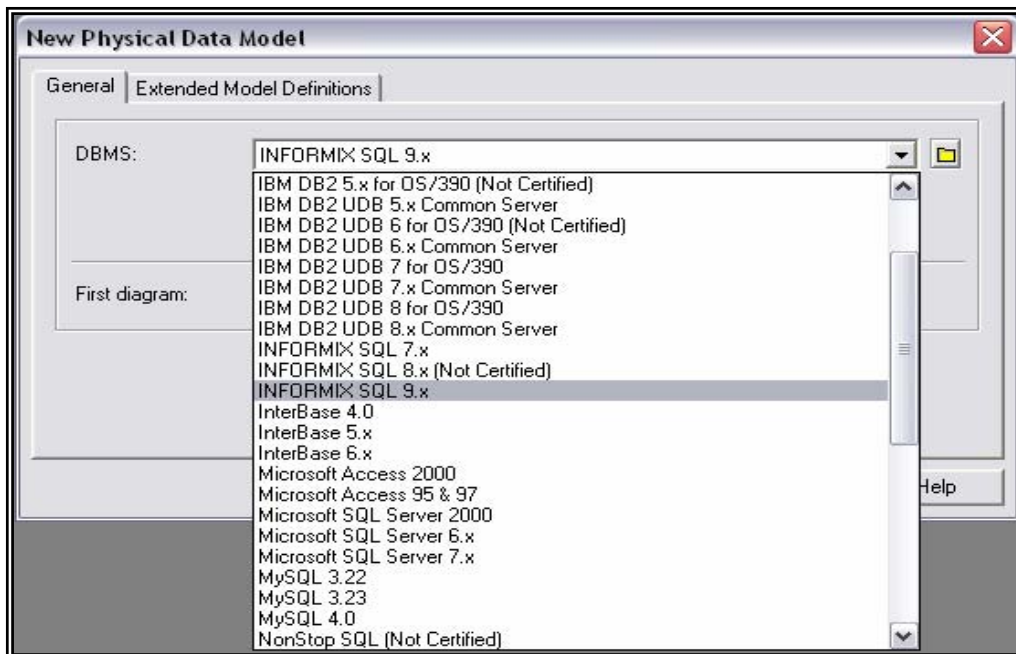
PROCESO DE REINGENIERÍA

Para poder realizar la Ingeniería reversa y generar el script de cualquier dato que necesitemos, realizamos una reingeniería de la Base de Datos del Sistema, para lo cual hay que seguir los siguientes pasos:

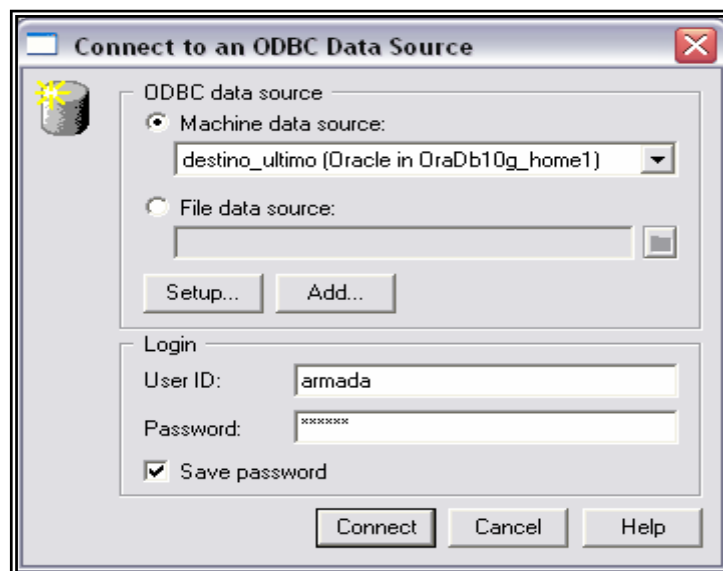
1. En menú-archivo, seleccionar “Ingeniería Reversa” – Database-
enter, para poder ingresar a la configuración del modelo físico.



2. Al abrirse la ventana seleccionar la Base de Datos en la cual se va a sacar el Script.



3. Se selecciona el ODBC para el cual voy a realizar la conexión con la Base de Datos y realizo la respectiva prueba de conexión, con el USER y PASSWORD ya asignado en la creación del ODBC.



4. Aparecen todas las tablas, vistas, triggers, etc para lo cual se puede seleccionar lo que se quiere realizar la ingeniería reversa.



PROCESO DE MIGRACIÓN

En el caso en que ciertas tablas, vistas, triggers, stored procedure e índices no se hayan logrado migrar correctamente debido a fallas de sintaxis, se utilizó la herramienta de PowerDesigner para poder crear los scripts y posteriormente ejecutarlos ya corregidos en PowerBuilder.

Los pasos para realizar la migración se detallan a continuación:

1.- Se procede a crear un Workspace.



2.- Se crea una nueva aplicación.





3.- Se crea el ODBC Open Database Connectivity que es un estándar de acceso a la base de datos desarrollado por Microsoft

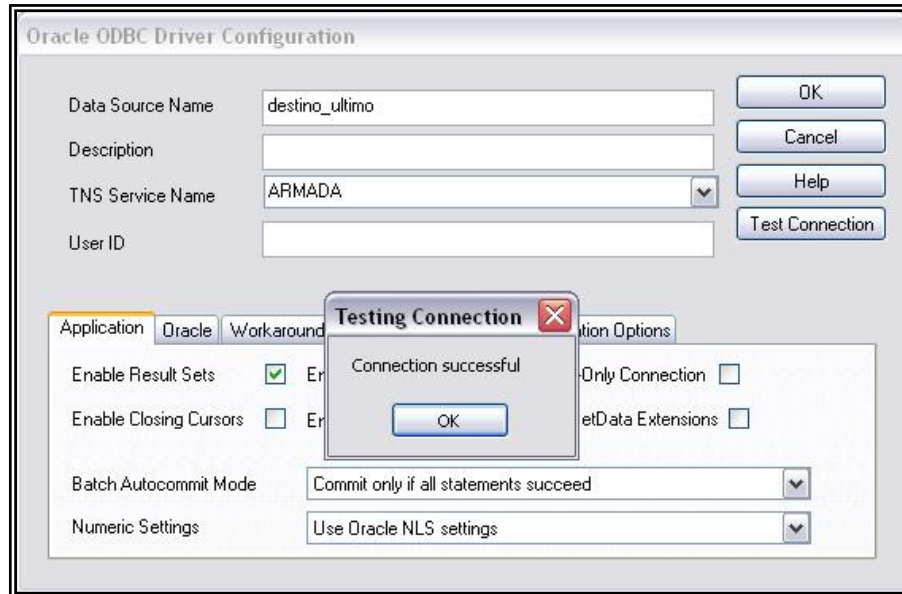


4.- Se selecciona el driver correspondiente.

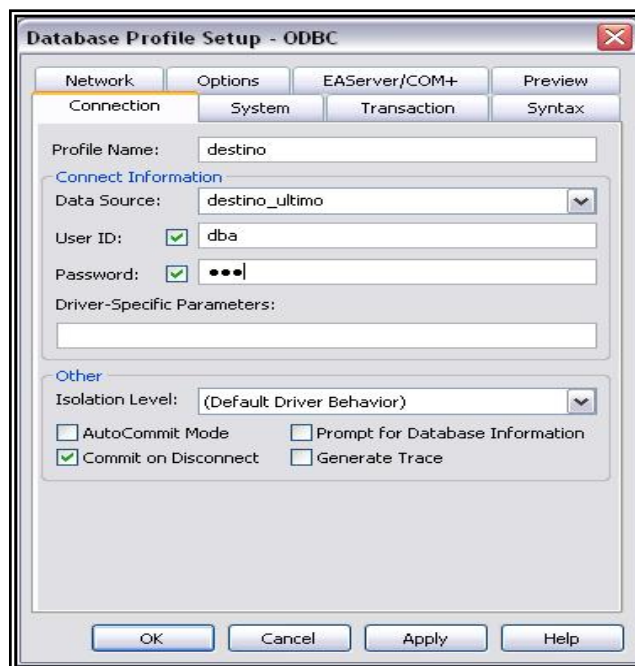




5.- Se configura el driver ODBC y además se hace una prueba de conexión.

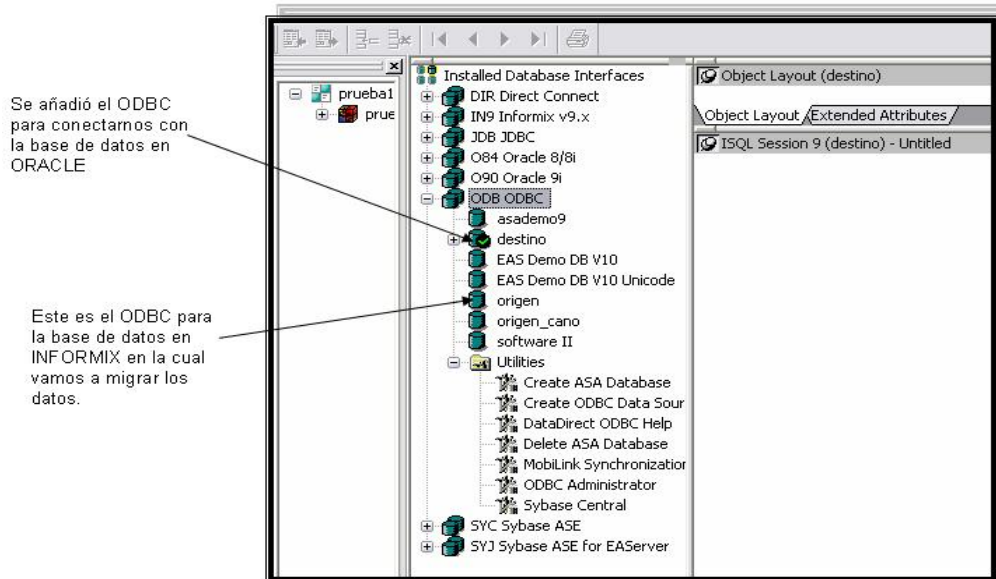


6.- Se configura el perfil del ODBC especificando un ID del usuario y la contraseña.

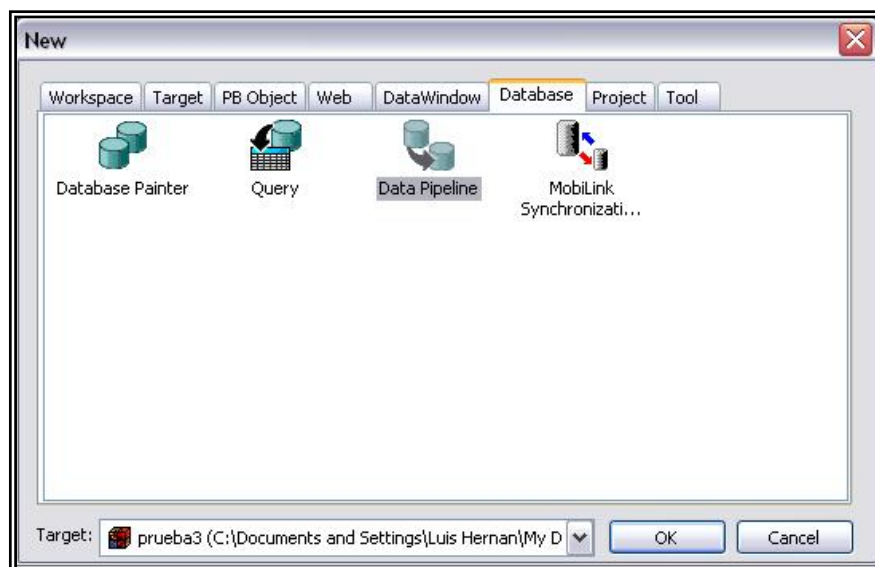




7.- Se visualiza el ODBC creado.



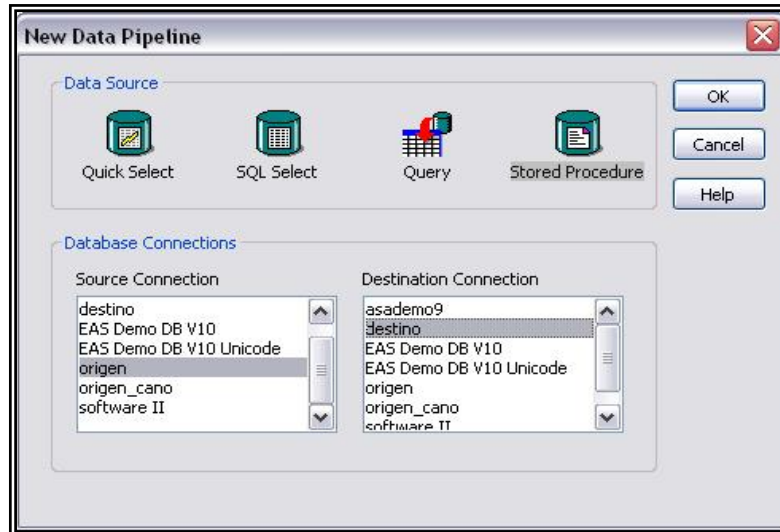
8.- Dentro de Database se selecciona Data Pipeline¹ que es para el proceso de migración



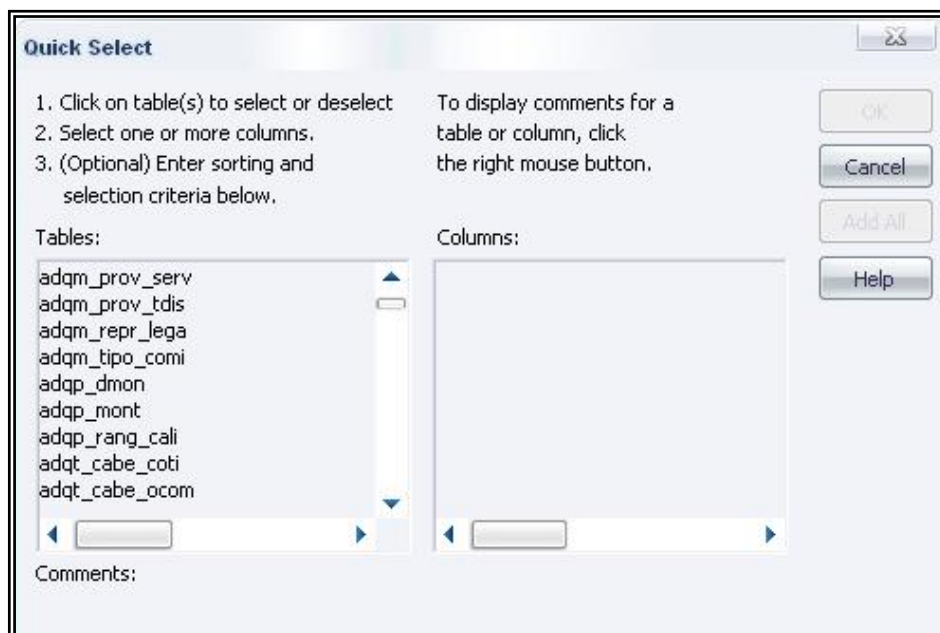
¹ Pipeline representa una “tubería” de datos entre una base de datos y otra, pudiendo ser estas de cualquier tipo dentro de los soportados por Power Builder. Es muy útil para hacer migraciones de bases de datos, ya que, simplemente indicando el origen y el destino, el proceso nos hará la copia de todos los objetos y datos (o aquellos que seleccionemos), ajustando aquellas sentencias que no sean compatibles en la base de datos destino.



9.- Se seleccionan las bases de datos de origen y de destino.

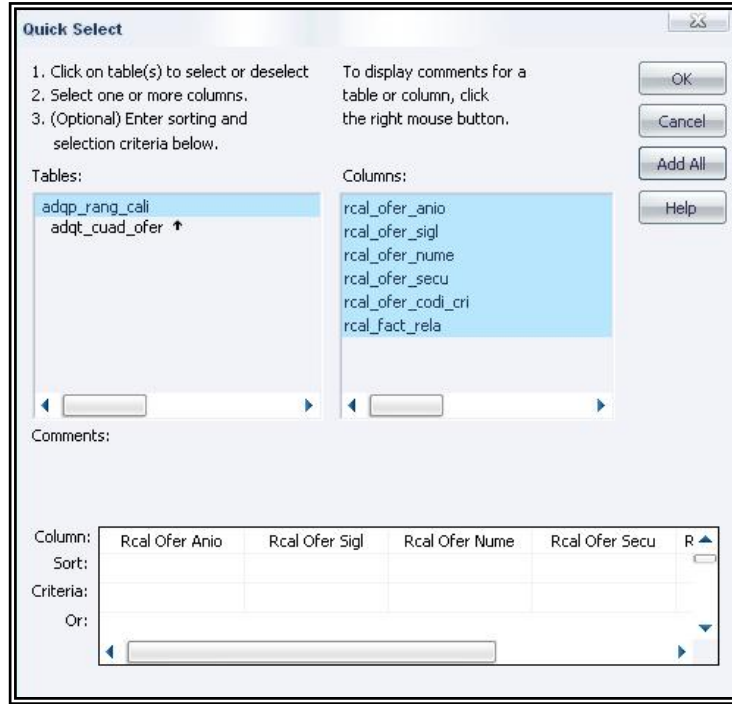


10.- Se muestran todas las tablas de la base de datos en Informix y se tiene que seleccionar una de ellas para luego ser migrada.

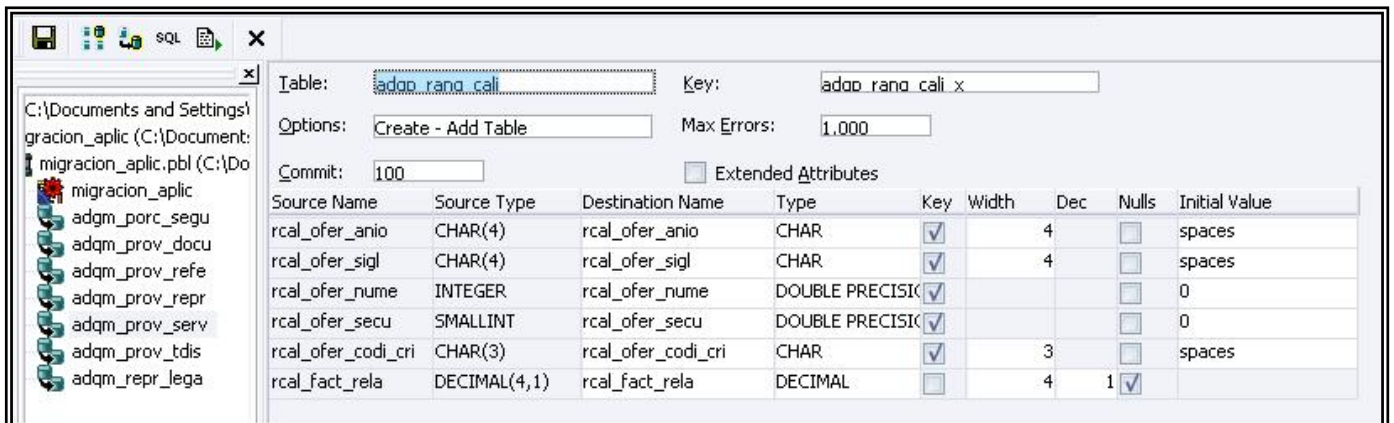




11.- Se selecciona una de las tablas y luego aparecerán todos los campos de las mismas.



12.- Si al momento de migrar una tabla, presenta una interfaz de advertencia que no se ha podido concluir la migración, se debe al número de campos y registros que posee, ya que el margen de error que muestra es muy pequeño para migrar esa cantidad de registros; se debe encerrar el margen de error con 1.000 para que no haya problemas al momento de migrar.





13.- En la siguiente figura se puede visualizar alguna de las tablas que se migraron usando la herramienta Enterprise Management de Oracle 10g.

ORACLE Enterprise Manager 10g Database Control

Database: ARMADA > Tablas

Tablas

Buscar

Seleccione un tipo de objeto y, opcionalmente, introduzca un nombre de esquema y un nombre de objeto para filtrar los datos que aparecerán en el juego de resultados.

Tipo de Objeto: Tabla Esquema: ARMADA Nombre del Objeto: Ir

Resultados

Seleccionar	Esquema	Nombre de la Tabla	Tablespace	Particionado	Filas	Ultimo Analisis
<input checked="" type="radio"/>	ARMADA	ADQM_CONC_EVAL	USERS	NO	14	30/03/2006 08:32:46 AM COT
<input type="radio"/>	ARMADA	ADQM_PORC_SEGU	USERS	NO	0	30/03/2006 08:32:46 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV	USERS	NO	105	30/03/2006 08:32:46 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_DOCU	USERS	NO	35780	30/03/2006 08:32:33 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_REFE	USERS	NO	37	30/03/2006 08:32:47 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_REPR	USERS	NO	5628	30/03/2006 08:33:12 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_SERV	USERS	NO	3506	30/03/2006 08:33:14 AM COT
<input type="radio"/>	ARMADA	ADQM_PROV_TDIS	USERS	NO	3121	30/03/2006 08:33:22 AM COT
<input type="radio"/>	ARMADA	ADQM_REPR_LEGA	USERS	NO	835	30/03/2006 08:33:16 AM COT
<input type="radio"/>	ARMADA	ADQM_TIPO_COMI	USERS	NO	4	30/03/2006 08:32:47 AM COT
<input type="radio"/>	ARMADA	ADQP_DMON	USERS	NO	10	30/03/2006 08:32:48 AM COT