

201



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

MIGRACION DEL SISTEMA DE INFORMACION DE INGENIERIA DE SISTEMAS DE ORACLE DE MEXICO A UN AMBIENTE DE WEB

## T E S I S

QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION

P R E S E N T A N :

CATALINA BARBEYTO CHALTE

KARINA IDALIA RAMIREZ GONZALEZ

JORGE BARBA ATILANO

HECTOR VIVEROS GALVEZ

DIRECTOR DE TESIS: M. I. JUAN CARLOS ROA BEIZA



MEXICO, D. F.

1998

TESIS CON FALLA DE ORIGEN

263768



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

---

**Migración del Sistema de Información de Ingeniería de  
Sistemas de Oracle de México a un Ambiente de Web.**

---

---

## **OBJETIVO**

**Proporcionar un mecanismo de acceso remoto sencillo al Sistema de Información de Ingeniería de Sistemas basado en el ambiente de Web, buscando la incorporación de interfaces gráficas amigables al usuario.**

---

---

## **Agradecimientos**

---

---

*Gracias por la confianza que me han brindado y por el apoyo que siempre he tenido de todos y cada uno de los miembros de la familia Barbeyto Chalte.*

Con cariño Catalina, junio-98

---

---

*A mi esposo Alejandro ... por ser mi más grande razón para vivir.*

*A mis papas ... por quererme.*

*A mis hermanos Rosy, Miriam y Manolo ... por confiar en mi.*

*A mis amigos ... por escucharme.*

**Gracias a Todos.**  
*Karina*

---

A mis Compañeros de Tesis por su gran esfuerzo.

A mi Familia por el apoyo invaluable y sus valiosos consejos.

A mis Amigos por su ayuda y compañía a lo largo de mi carrera.

A mis Profesores por sus enseñanzas y por dedicarme parte de sus vidas.

A a la Facultad de Ingeniería y a la Universidad por darme la gran oportunidad de estudiar una carrera profesional.

A la Unidad de Cómputo por su participación en mi formación como profesional.

A Dios por permitirme llevar a cabo mis metas en esta vida.

Dedico este trabajo al ser que me dió la vida.

A mi Madre, gracias por todo.

Jorge Barba Atilano.

---



---

*A mis padres por darme la vida, su apoyo y la oportunidad de recibir la mejor de las herencias, el estudio.*

*A mis hermanos por su amor y apoyo incondicional.*

*A mi familia y amigos por el cariño y entusiasmo recibido.*

*A mis profesores por brindarme sus conocimientos y consejos.*

*A mi esposa por ser mi compañera de la vida.*

*Y a mis sobrinos e hijos por hacerme recordar cada día lo importante que es no perder la capacidad de asombro y la alegría de vivir. Gracias.*

*Héctor Viveros Gálvez*

---

---

## **Indice**

---

---

**Introducción**

i

**Capítulo 1 Teoría Básica**

1.1 Metodologías de diseño y desarrollo de sistemas	1-1
1.2 Conceptos generales del manejador de base de datos relacionales Oracle	1-33
1.3 Características, ventajas y desventajas de las principales arquitecturas de redes comerciales	1-55
1.4 Características, funcionamiento, ventajas y desventajas de la arquitectura cliente-servidor de tres capas	1-122
1.5 Diseño, características y uso de la herramienta CASE de Oracle	1-135
1.6 Metodologías de diseño de ingeniería en reversa	1-164
1.7 Fundamentos del ambiente de trabajo Internet, Intranet y WWW	1-177
1.8 Características, comparación y selección del lenguaje para la generación de páginas de web	1-203

**Capítulo 2 Planteamiento del Problema y Propuesta de Solución**

2.1 Problemática y organigrama de la situación actual	2-1
2.2 Análisis y Clasificación de la Información	2-8
2.3 Requerimientos del usuario	2-10
2.4 Situación que se pretende alcanzar	2-23
2.5 Plan de trabajo de la solución propuesta	2-27

**Capítulo 3 Diseño e Implementación del Sistema**

3.1 Especificaciones de diseño y diagrama de flujo general	3-1
3.2 Ingeniería en reversa de la base de datos	3-13
3.3 Ingeniería en reversa de la aplicación	3-22
3.4 Regeneración de la base de datos a la versión requerida	3-52
3.5 Diseño y desarrollo del front-end	3-65
3.6 Integración, Pruebas y Transición del Sistema	3-90
3.7 Etapa de Producción	3-103

**Capítulo 4 Conclusiones**

4-1

**Bibliografía****Apéndices**

Apéndice A Manuales de usuario y técnico del Sistema de Control de Preventa	A-1
Apéndice B Manuales de usuario y técnico del Sistema de Canales Alternos	B-1
Apéndice C Manual de Usuario	C-1
Apéndice D Manual Técnico	D-1

---

---

## **Introducción**



---

---

## INTRODUCCIÓN

Hoy día Ingeniería de Sistemas cuenta con un sistema de información que le permite llevar al cabo el control de actividades de los ingenieros de sistemas, el control de los productos existentes en inventario, así como el control de las diversas actividades que se realizan por parte del área de ingeniería de sistemas con los socios comerciales.

Para acceder y operar el sistema es necesario hacer uso de terminales tontas conectadas al equipo multiusuario en donde corre el sistema, o bien, utilizando una computadora personal se puede tener acceso al mismo utilizando un software de emulación de terminal, lo cual es una limitante en el acceso al sistema.

Para que los usuarios puedan operar el sistema, es indispensable que estén en las oficinas de la empresa o en su defecto traigan consigo una computadora portátil con módem para conectarse a la máquina en la que reside el sistema y tener que trabajar con el problema de diferentes tipos de teclados y funciones. Además, la interfaz modo carácter de despliegue del sistema lo hace poco amigable.

El sistema esta hecho con herramientas de desarrollo de Oracle que son Sql\*Forms 3.0, Sql\*Menu 5.0 y Sql\*ReportWriter 1.1, accediendo al manejador de base de datos Oracle7 versión 7.0.16, que no son las últimas versiones ni de la base de datos ni de las herramientas y que en un futuro a corto plazo dejarán de ser soportadas, lo que las llevará a un estado de obsolescencia. Adicionalmente, el sistema está en una ambiente centralizado, en el que en el mismo servidor reside el manejador de base de datos y la aplicación, lo que lleva a tener una carga del procesamiento en el servidor.

Es necesario que el área que le da mantenimiento al sistema tenga que seguir usando tecnología vieja para incorporar los nuevos cambios que los usuarios demandan, lo cual trae como consecuencia una baja productividad de esta área.

---

Hoy en día el dinamismo de la empresa va exigiendo nuevas formas de cómo llevar al cabo la operación diaria y la pone en el plano de poder hacer más eficiente y rápido el trabajo administrativo que se realiza con el sistema, es decir, de qué manera poder hacer que el usuario no recorra distancias dentro de la ciudad para operar el sistema.

En consecuencia a tales exigencias, es necesario poner el sistema en un ambiente que proporcione flexibilidad en el acceso, que sea amigable a través de una interfaz gráfica y dejarlo en un esquema de procesamiento descentralizado para un mejor rendimiento de la aplicación, para lo cual se puede, en primera instancia, pensar en ponerlo en el entorno del mundo Internet que proporcione todas las características deseadas actualmente para el sistema.

Analizando los recursos de cómputo disponibles dentro de la empresa para la realización del proyecto se cuenta con un servidor Sun SPARC con sistema operativo solaris 2.4, 64 MB de RAM y 2 GB de espacio en disco; un servidor Sequent con sistema operativo Dynix/PTX 4.1; equipos PC y laptops con procesador pentium, 32 MB de RAM, 1GB de espacio en disco y monitor super VGA a color. Adicionalmente, se tiene un sistema de red con topología Ethernet y acceso a Internet. Como recursos de software se tienen el manejador de base de datos Oracle 7 versión 7.3, Oracle Web Server 2.0, Designer/2000, Developer/2000 y navegadores de internet.

Teniendo como fundamento lo anterior se decidió incorporar la estructura de una Intranet y de un esquema de tres capas, en la que se tiene un servidor de base de datos, un servidor de aplicaciones y un cliente. El servidor Sequent se utilizará como servidor de datos, el servidor Sun Sparc se utilizará para el manejo de la aplicación y clientes PC's con capacidad para trabajar con cualquier navegador en un ambiente gráfico, ameno y sencillo para el usuario.

El sistema que se desea cambiar se basará en el manejador de base de datos Oracle como servidor de datos. En el caso de la aplicación ésta será cambiada a páginas de Web dinámicas, para lo cual se utilizará Developer/2000 sólo para la migración de la misma, Designer/2000 para hacer la ingeniería en reversa de los módulos y generar el sistema en HTML, así como para obtener la documentación del sistema.

---

# **Capítulo 1**

## **Teoría Básica**

## 1.1 METODOLOGÍAS DE DISEÑO Y DESARROLLO DE SISTEMAS

La metodología del desarrollo de sistemas es el camino que siguen los analistas de sistemas al realizar su trabajo. Se emplea el término genérico de *analista de sistemas* para describir a la persona que tiene la responsabilidad principal de conjuntar los componentes estructurales, dándoles forma y sustancia en conformidad con las fuerzas del diseño para construir sistemas de información exitosos. En una compañía pequeña, el analista quizás no sólo diseñará el sistema de información, sino que también hará la programación y operará la computadora. En una compañía grande, el analista de sistemas puede preparar las especificaciones del diseño que se dan a los técnicos, como los programadores, ergonomistas, diseñadores de formas y especialistas en comunicaciones. El analista de sistemas coordinará las tareas de todos los especialistas para la implementación final del sistema completo.

### Fases principales de la Metodología del Desarrollo de Sistemas

En la figura 1.1.1 se muestra un esquema general de ésta. Sus fases principales son el análisis de sistemas, el diseño general de sistemas, la evaluación de sistemas, el diseño detallado de sistemas y la implementación de sistemas. Dentro de cada fase se incluyen las principales actividades o tareas. Las primeras cuatro fases están dirigidas a proporcionar valores específicos para los componentes estructurales. La última fase se ocupa de que los componentes estructurales sean operacionales. Algunos autores, compañías y consultores dividen su metodología de desarrollo de sistemas en fases o etapas diferentes a las presentadas en este trabajo, sin embargo independientemente del número o nombres de las fases o etapas, la metodología del desarrollo de sistemas racionaliza y asigna una rutina al proceso de construcción de sistemas de información. Su sello distintivo lo forman sus fases discretas. La meta principal de la metodología del desarrollo de sistemas es reducir los inicios falsos, reciclamiento indebido, retrabajos y callejones sin salida. Además, aumenta la probabilidad de que el sistema que se construya e instale finalmente sea el que los usuarios desean y necesitan. Pero de ninguna manera la metodología del desarrollo de sistemas deberá ahogar la creatividad o la innovación. Proporciona un faro guía para el recorrido a lo largo de un sistema de información. El qué tan excitante, productivo y



creativo desee el analista de sistemas que sea este recorrido depende, en gran medida, del propio analista.

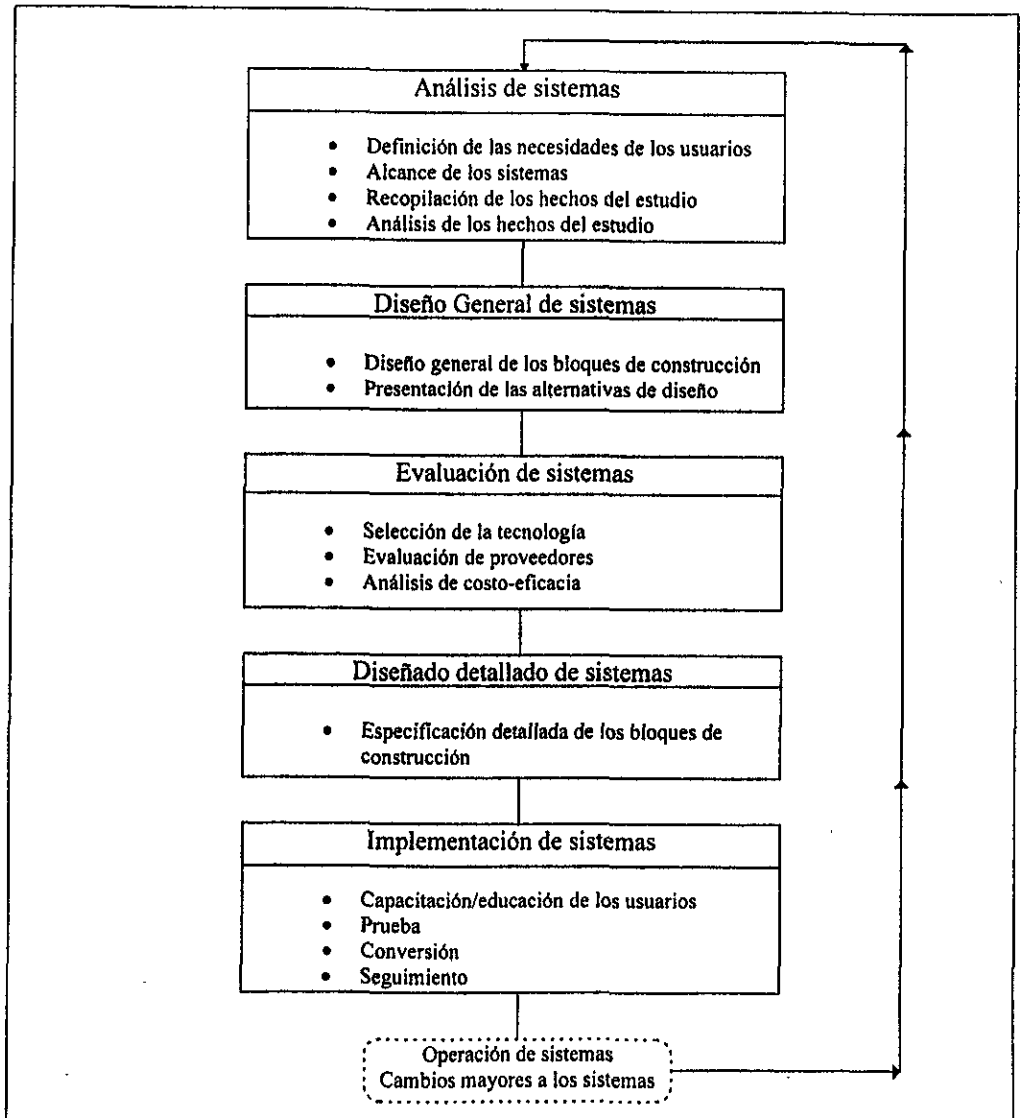


Figura 1.1.1 Metodología del desarrollo de sistemas.

## Definición de la Metodología del Desarrollo de Sistemas

Durante la fase del *análisis de sistemas*, los analistas de sistemas ayudan a los usuarios a identificar qué información se necesita. Se llevan a cabo varias entrevistas y se plantean preguntas como: “¿Qué información está usted recibiendo actualmente?”, “¿Qué clase de información necesita para realizar su trabajo?” Sin embargo, en este punto los analistas de sistemas deben ser más que escuchas pasivos o “mecnógrafos”; deben ser líderes y maestros. Con frecuencia los usuarios no tienen una idea completa de cuál es la información que realmente necesitan o cómo pueden obtenerla. En la mayoría de los casos, los analistas deben interactuar y determinar con los usuarios cuáles son sus verdaderas necesidades. Después de un buen número de entrevistas, observaciones y muestreos, los analistas de sistemas empiezan a conjuntar muchos hechos del estudio para un análisis posterior. Hacia el final de la fase de análisis, los analistas conceptualizan diseños generales y gradualmente van construyendo un modelo.

Antes de iniciar el diseño general de sistemas, la fase de análisis debe completarse y los usuarios deben estar de acuerdo con los resultados. El diseño es el proceso de traducir los requerimientos definidos durante el análisis a varias alternativas de diseño para la consideración de los usuarios. También implica descubrir todas las fuerzas de diseño para ver cómo impactarán e influirán en los diferentes diseños. Los usuarios pueden responder a las alternativas de diseño y empezar a trabajar hacia un consenso. Sin embargo, pueden ocurrir varias iteraciones entre el análisis y el diseño, en que cada iteración implique malabear y dar nueva forma a los componentes estructurales.

En la fase de *diseño*, los usuarios seleccionan dos o tres de los mejores modelos o le piden al analista que “regrese al restirador” para desarrollar un mejor modelo. Si se recopilaron y analizaron suficientes hechos del estudio en la fase de análisis, se reduce la probabilidad de tener que regresar a dicha fase. También se debe recalcar que durante el *diseño* los analistas de sistemas con frecuencia empezarán a hacer uso de técnicos como científicos de la administración, ergonomistas, expertos en robótica, programadores, especialistas de formas y expertos en

comunicaciones. En cualquier caso, los analistas concluyen la fase del diseño general con un entendimiento bastante claro de lo que se desea y de lo que tendrán que hacer para obtenerlo.

La *evaluación de sistemas* en realidad no es una fase discreta; en todo momento durante el trabajo en sistemas se realiza una gran cantidad de evaluaciones, pero para resaltarla, aquí se presenta como una fase separada. Esta fase implica la selección de la tecnología que soportará a los otros componentes estructurales, la evaluación de esta tecnología y los proveedores que la suministran, y un análisis completo de costo-eficacia de cada una de las opciones propuestas del diseño de sistemas para determinar aquella con la mejor proporción de eficacia con relación al costo. Obviamente, los analistas no desean tener que recorrer el camino largo y difícil hacia el *diseño detallado de sistemas* teniendo varias opciones de diseño aún en el aire.

Antes de iniciar este trabajo sustancial, los analistas desean asegurarse de que todas las partes hayan llegado a un consenso final en cuanto al sistema que desean que se implemente. Aquí, a cada componente de construcción se le da una definición precisa y detallada. A decir verdad, sería un desperdicio excesivo de tiempo y dinero, por no mencionar la frustración y consternación, diseñar en detalle un sistema sólo para encontrar finalmente que nadie lo quiere y que no corresponde a las fuerzas del diseño.

La *implementación de sistemas* es la capacitación y educación de los usuarios, la prueba y la conversión para hacer que el sistema sea operacional. Aquí es donde todo el trabajo de desarrollo y diseño llega a un clímax. Los analistas fijan las fechas límite o "puntos de congelación", capacitan y coordinan al personal usuario, instruyen a los técnicos, prueban el nuevo sistema y eliminan errores, instalan nuevos procedimientos y formas, y ven si hay descuidos u omisiones. En algún punto, el nuevo sistema diseñado es aceptado por el personal que lo operará y administrará bajo una base cotidiana. Poco después de su aceptación, el analista de sistemas deberá realizar un seguimiento para ver que el sistema está operando según lo esperado. En muchos casos, los analistas pueden hacer algunos ajustes menores o "afinaciones" que resulten en mejoras significativas. Después de que el sistema haya estado operando durante cierto tiempo,

las necesidades de los usuarios cambiarán y los analistas de sistemas se verán involucrados en la preparación de prototipos especiales para cubrir necesidades particulares.

En ocasiones será necesario efectuar algún trabajo de mantenimiento. Finalmente, en algún momento, tiempo después de que el sistema haya estado en operación, el analista de sistemas se verá de nuevo fuertemente involucrado en un mantenimiento mayor del sistema, o en el desarrollo y diseño de uno nuevo, y entonces se repetirá el ciclo de vida de la metodología del desarrollo de sistemas.

Puede ser necesario que los analistas desarrollen un subsistema con base en una solicitud especial. Para corresponder a esta solicitud los analistas pueden trabajar en la terminal del usuario para ayudar a diseñar formatos de pantalla alternos de acuerdo a las especificaciones del usuario. Esta interacción y consulta de uno-a-uno con el usuario se denomina elaboración de *prototipos* y, en gran medida, es un subconjunto de la metodología del desarrollo de sistemas pero se centra en una aplicación para un usuario (o unos cuantos usuarios) y su alcance no es amplio.

La sesión de elaboración de prototipos no dura más de una hora y produce un modelo (prototipo) de una aplicación. Con el usuario presente e interactuando con el analista de sistemas, se puede obtener en la pantalla un bosquejo general que sea bastante parecido a lo que el usuario desea. El usuario y el analista de sistemas diseñan, prueban, modifican y rediseñan conjuntamente los prototipos hasta que el usuario elige uno para su implementación posterior.

El papel de los analistas de sistemas y su empleo de la metodología del desarrollo de sistemas es, por supuesto un cuadro compuesto. En una organización grande el sistema de información es una entidad vasta y compleja, y muchos analistas están involucrados en el diseño e implementación de sólo un subsistema de todo el sistema de información de la organización. En un momento dado, un analista podría muy bien estar realizando actividades relacionadas con más de una fase de la metodología del desarrollo de sistemas, y estas actividades se deben coordinar adecuadamente. Para que el sistema sea más eficaz, el analista de sistemas participará en un

amplio espectro de actividades que van desde las formales a las informales, cuantitativas a cualitativas, estructuradas a no estructuradas, específicas a generales. Al realizar estas actividades, el analista puede hacer uso de diagramas de flujo, diagramas de flujo de datos, tablas de decisión, matrices, gráficas, reportes narrativos, entrevistas, modelos y prototipos. Estas técnicas se refuerzan entre sí y, cuando se ven en conjunto, proporcionan las herramientas básicas para el trabajo en sistemas.

Después de la fase de análisis de sistemas, en muchos casos se ve con bastante claridad lo que debe hacerse.

### **Opciones de Desarrollo**

Los analistas de sistemas cuentan por lo menos con tres opciones básicas, cada vez que se evalúa un conjunto de requerimientos de los usuarios. El analista puede recomendar que no se haga nada, que se modifique un sistema existente o que se diseñe un nuevo sistema.

1. *No hacer nada.* En toda decisión de sistemas referente a la forma de satisfacer los requerimientos de información de los usuarios o las solicitudes de mejora en los sistemas, el analista tiene la oportunidad de recomendar que no se tome ninguna acción en ese momento. Las razones para elegir esta opción incluyen (a) una pobre identificación de los requerimientos y necesidades, (b) la determinación de que no es factible desarrollar un sistema significativo o una solución a las necesidades de los usuarios, (c) otras solicitudes de sistemas con mayor prioridad o debido a que los recursos para el desarrollo ya se han asignado completamente, o (d) las necesidades de los usuarios, según se ha planteado, no son necesidades reales.
2. *Modificación de un sistema existente.* La mayoría de todas las investigaciones de sistemas que se realizan en las organizaciones incluyen alguna consideración de los sistemas y subsistemas existentes. Para satisfacer requerimientos nuevos o revisados de los usuarios, el analista con frecuencia recomienda la modificación de los sistemas existentes en lugar de

diseñar sistemas nuevos. Dependiendo del tamaño de la organización y del subsistema en particular que se está evaluando, las modificaciones de sistemas pueden tener un mayor impacto sobre la organización que el desarrollo de un subsistema totalmente nuevo.

Este impacto puede resultar del tamaño del esfuerzo dedicado a los sistemas o del cambio que se está dando en la organización. Cuando se aplica un sistema como soporte a la solución de un problema organizacional, el énfasis está en los resultados inmediatos. De esta forma, con frecuencia se implementan cambios a los sistemas existentes hasta que se pueda definir y desarrollar un nuevo sistema. Adicionalmente, el nivel de desarrollo de sistemas de información que existe actualmente en muchas organizaciones de tamaño mediano a grande ha llegado a un punto en el que las nuevas demandas de los usuarios con frecuencia requieren cambios relativamente pequeños a la captura de datos y elementos de almacenamiento, y el énfasis se da en el acceso a los datos disponibles en un nuevo formato o en una forma más oportuna.

3. *Diseño de un nuevo sistema.* La opción final a disposición del analista consiste que el analista recomiende un nuevo sistema para satisfacer los requerimientos de los usuarios. Esta opción es obviamente la solución más compleja y difícil de implementar. Esta puede verse como una combinación de dos elecciones de acción posteriores. Cuando un analista recomienda que se implemente un nuevo sistema, se debe tomar la decisión acerca de si el sistema se va a desarrollar desde su principio o si se va a comprar de otras fuentes un sistema aceptable. Tradicionalmente esto se conoce como la decisión de "hacer-o-comprar".

### **Definición del Diseño**

El diseño general de sistemas puede definirse como el dibujo, planeación, bosquejo o arreglo de muchos elementos separados en un todo viable y unificado. En tanto que la fase de análisis de sistemas responde a las preguntas de lo que está haciendo el sistema y de lo que debería estar haciendo para satisfacer los requerimientos de los usuarios, la fase del diseño de sistemas se ocupa de la forma en que se desarrolla el sistema para satisfacer estos requerimientos.

La fase del diseño de sistemas está orientada técnicamente al punto en que los analistas deben contestar la pregunta: "¿Cómo lo vamos a hacer?". Por otra parte, el diseño es un arte, orientado en forma creativa.

### **La Mente de Diseño**

En su trabajo, los analistas de sistemas poseen y utilizan tres mentes: *la mente de análisis, la mente de diseño y la mente de evaluación*. Pero estas mentes no están separadas como tales. Se combinan y se mezclan y, en donde es apropiado, una está más enfocada y activa que las otras dos. A medida que se analizan los hechos de estudio y se revisan los hallazgos, en la mente de diseño empiezan a cristalizarse conceptos claros de diseño. El analista de sistemas formula y convierte estas ideas de diseño en modelos de diseño empleando algunas de las técnicas de modelación, como los diagramas de flujo de datos, los diagramas de entidad-relación y los diagramas de estructura. Ya sea en papel o en pantalla, estos modelos se visualizan, se evalúan y se vuelven a dibujar hasta que son apropiados y factibles. Las opciones de diseño se malaborean mentalmente y se evalúan aún más hasta que algunas de ellas empiezan a sobresalir de las demás. Los detalles que no son críticos se dejan a un lado para su solución durante la fase del diseño detallado de sistemas. Algunas de las ideas de diseño más pertinentes que se generan con lluvias de ideas y entrevistas de grupos también pueden accederse y utilizarse en este momento, y mezclarse y combinarse con los modelos de diseño.

En algunos casos, la mente de diseño no hará su trabajo ni proporcionará al analista de sistemas ideas de diseño claramente definidas. Si esto sucede, el analista de sistemas deberá concentrarse en los hechos de estudio y en los requerimientos de los usuarios hasta un punto de agotamiento.

### **Proceso del Diseño y presentación de Opciones de Diseño**

El proceso del diseño general de sistemas se ilustra en la figura 1.1.2. El analista de sistemas conoce los requerimientos de los usuarios, el alcance de los sistemas y los recursos disponibles. Las fuerzas de diseño se consideran y se ponderan para determinar sobre los componentes

estructurales y los diseños de sistemas finales. Se crean diferentes alternativas de diseño para tomar en cuenta diferentes funciones de ponderación. Por ejemplo, en una opción se puede dar mucho peso a la integración, a la interfaz usuario/sistema, a las fuerzas competitivas y a los factores humanos. En otra opción se puede dar más peso a los requerimientos de sistemas y a los requerimientos de costo-eficacia.

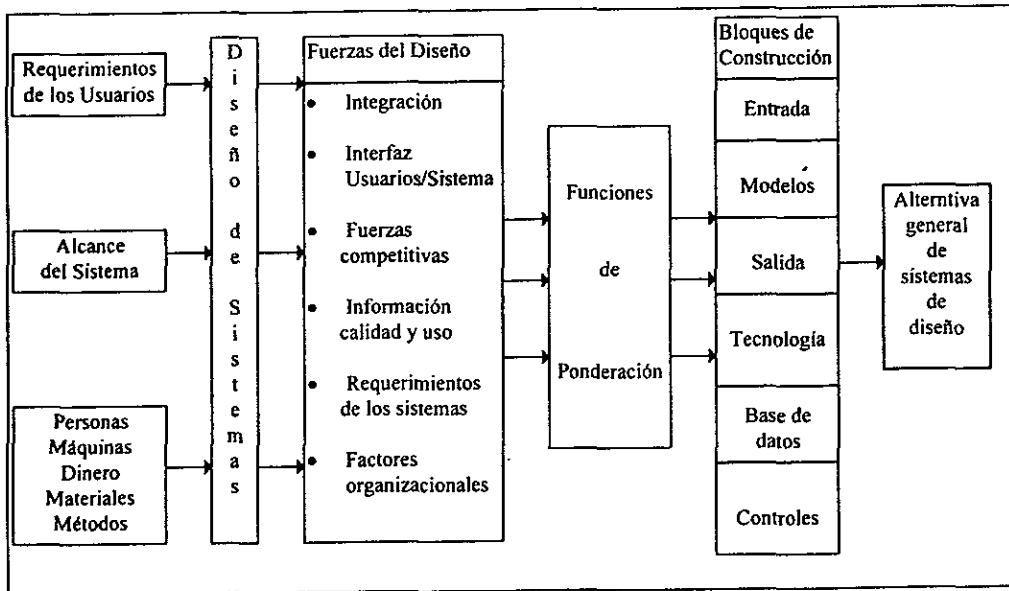


Figura 1.1.2 Proceso general del diseño de sistemas.

Con base en las fuerzas de diseño se pueden crear diversas opciones de diseño que se someten a la consideración de los usuarios. Por ejemplo, una opción puede estar basada en una topología de red en estrella, en tanto que otra puede basarse en una topología de anillo. Una opción puede extenderse en el ambiente, como en los sistemas de los clientes, y otra puede manejar los pedidos en una forma tradicional. Una puede ofrecer grandes capacidades de filtrado, monitoreo e interacción; otra puede ofrecer pocas. Una opción de diseño puede incluir un componente estructural de tecnología compuesta de muchos avances recientes, en tanto que otra solo incluya la tecnología establecida. O bien, las opciones de diseño pueden originarse de la decisión entre



rentar o comprar equipo, o de la decisión entre desarrollar paquetes de software o comprarlos a proveedores de software.

El punto es que el analista de sistemas es un profesional y presenta a los usuarios diversas opciones de diseño de entre las cuales pueden elegirse algunas para una evaluación más profunda. El analista de sistemas trata de cubrir todas las contingencias dando a los usuarios la oportunidad de ver diferentes caminos para satisfacer sus requerimientos. La presentación de opciones aumenta la probabilidad de que finalmente se implemente el diseño correcto, ya que en muchos casos los usuarios realmente no saben lo que desean.

### **Preparación del Reporte de la propuesta del Diseño General de Sistemas**

El reporte de la propuesta del diseño general de sistemas se prepara para comunicar a la gerencia y a los usuarios de la organización la forma, a nivel general, en que el sistema diseñado podrá satisfacer sus requerimientos de sistemas de información y procesamientos de datos. Las siguientes guías se ofrecen para ayudar al analista a preparar el reporte de la propuesta del diseño general de sistemas:

1. Plantear nuevamente la(s) razón(es) para iniciar el trabajo en sistemas, incluyendo los objetivos específicos. Relacionar todos los requerimientos originales de los usuarios y los objetivos con la propuesta actual de diseño de sistemas.
2. Preparar un modelo conciso pero completo del diseño de sistemas propuesto. Tratar siempre de incluir opciones de diseño de entre las cuales la gerencia pueda elegir, en vez de presentar sólo un enfoque, además con frecuencia se puede determinar que una opción particular tendrá un impacto significativamente diferente sobre la organización. Por ejemplo, la propuesta de diseño B puede cubrir un 90 por ciento de los requerimientos de la propuesta del diseño A, pero B puede costar solamente un 40 por ciento del costo de A. El analista nunca debe entrar en una situación en la que deba elegir entre un diseño particular o nada.

3. Mostrar todos los recursos requeridos para implementar y mantener cada alternativa.
4. Identificar toda suposición crítica o problema no resuelto que pueda afectar al diseño final de sistemas.

Ciertamente, el formato del reporte de la propuesta del diseño general de sistemas está sujeto a una amplia variación de una organización a otra. Sin embargo, el punto principal que debe tenerse presente al preparar una propuesta de diseño es que la(s) persona(s) que debe autorizar el desarrollo de una de la opciones o una combinación de las mismas debe tener suficientes elementos sobre los cuales basar su decisión. En efecto, el analista de sistemas está haciendo una invitación a las críticas y sugerencias de los usuarios al presentar el reporte de la propuesta para el diseño general de sistemas. Esta propuesta obviamente tiene una probabilidad elevada de que las opciones del diseño general de sistemas tendrán que modificarse varias veces antes de que los usuarios acepten finalmente las que consideren que cubrirán completamente sus requerimientos.

### **Bosquejos y Prototipos**

A menos que la clase de proyectos de sistemas que se esta desarrollando sea tradicional, como un sistema básico de contabilidad, los usuarios no siempre podrán definir sus requerimientos en forma adecuada y precisa. Simplemente no pueden especificar sus requerimientos de manera previa. Tienen que descubrirlos. Tres técnicas de análisis y diseño que ayudan a este proceso circunstancial son el bosquejo de componentes estructurales, el bosquejo en papel en blanco y la elaboración de prototipos.

### **Bosquejo de Componentes Estructurales**

La reunión de todos los componentes estructurales en una hoja de papel, como se muestra en la figura 1.1.3, o en una pantalla para cada opción de diseño, le dan al analista un bosquejo y una vista general del sistema total. Las opciones pueden bosquejarse para que los usuarios presenten sus reacciones y ofrezcan una retroalimentación valiosa a los analistas de sistemas. Estas

imitaciones del funcionamiento de los sistemas aumentan la comprensión tanto de los analistas de sistemas como de los usuarios finales, ayudan a detectar los elementos faltantes y generan ideas de diseño adicionales. Cada uno de los diseños finales posibles está contenido en una hoja de componentes estructurales. Estas hojas se convierten en una parte importante del reporte de la propuesta del diseño general de sistemas.

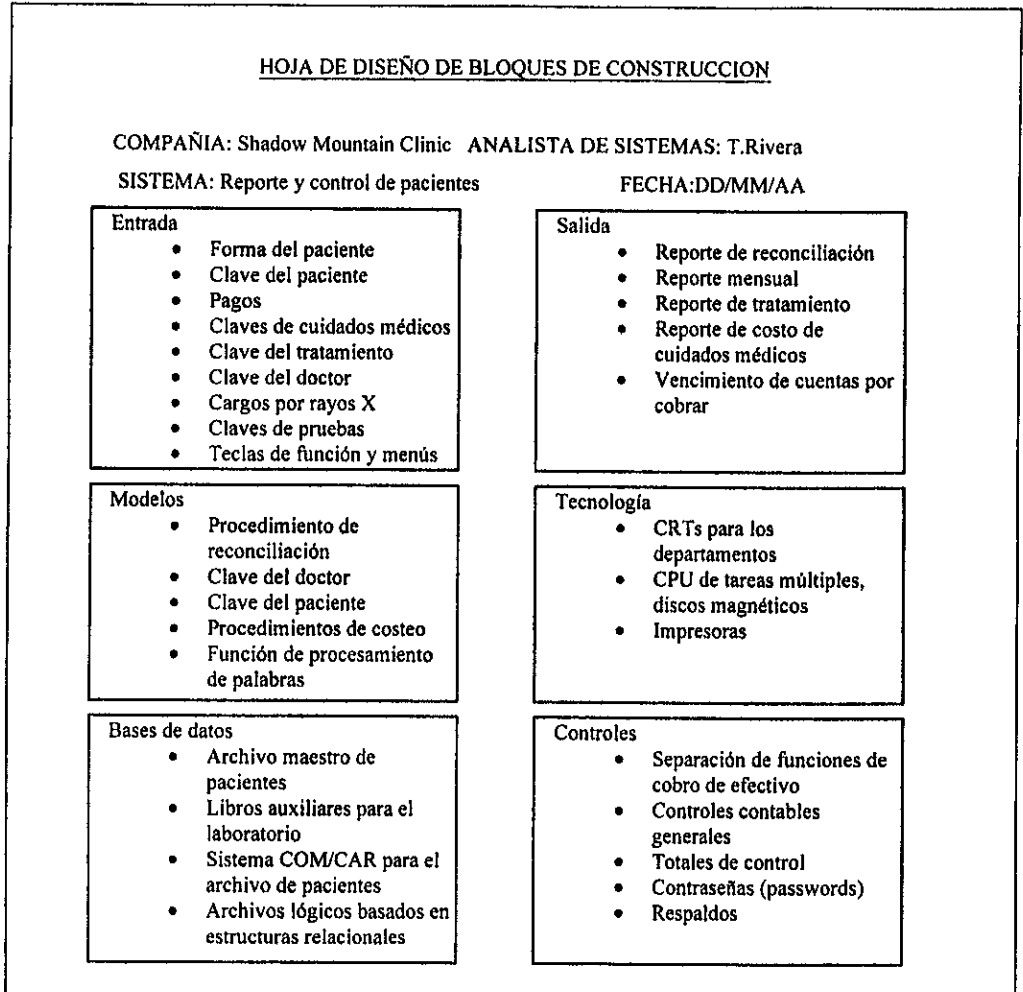


Figura 1.1.3 Hoja de diseño de los componentes de construcción.

## Bosquejo en Papel en Blanco

El bosquejo en papel blanco es una técnica de diseño que tiene sus raíces en los prototipos, la simulación y la lluvia de ideas. A los usuarios potenciales simplemente se les da una hoja en blanco o una pantalla y se les pide que bosquejen lo que desean como salida. Los usuarios, con una mínima ayuda y guía de los analistas, bosquejan elementos como tablas y diagramas y el contenido de la salida deseada. Los bosquejos iniciales pueden ser rudimentarios e incompletos. Sin embargo, con el desarrollo de cada nuevo bosquejo tiene lugar un aprendizaje iterativo. De hecho, un proceso de diseño bastante complejo puede iniciarse con bosquejos y progresar hasta representaciones con un mayor detalle y significado. Con el paso del tiempo, se obtiene un nivel de detalle a partir del cual se puede implementar el bosquejo final. Y, lo más importante de todo, es que los analistas de sistemas pueden asegurarse en cierto modo que los sistemas que implementen serán los que los usuarios desean y necesitan.

## Elaboración de Prototipos

Una de las peores cosas que le pueden suceder a un analista de sistemas es diseñar e implementar un sistema que los usuarios no desean ni necesitan. Todas las técnicas y conceptos presentadas hasta este momento ayudan a evitar que se presente este problema. Durante la planeación estratégica de sistemas de información, a los analistas de sistemas se les da una guía y misiones específicas de sistemas. En el análisis de sistemas se recopilan hechos de estudio de varias fuentes. Estos hechos de estudio se analizan y se estudian a conciencia. Se hacen presentaciones de reportes de terminación y propuestas de análisis para que los usuarios proporcionen sus críticas y evaluación. Se resuelven las diferencias y se logran acuerdos. Los analistas de sistemas, contando con una visión aún mas clara y mas detalles específicos, inician la fase del diseño general de sistemas y desarrollan varias alternativas de diseño con base en lo que se ha aprendido hasta ese momento. Los usuarios tienen nuevamente una oportunidad de aceptar, rechazar o combinar estas opciones de diseño.

¿No es suficiente todo este trabajo en sistemas para evitar que se implemente un sistema malo? No necesariamente. ¿Porqué? Debido a que los analistas de sistemas con experiencia saben que, en algunos casos, "lo que los usuarios piden no es lo que desean; lo que desean no es realmente lo que necesitan". A decir verdad, algunos usuarios simplemente no saben que necesitan hasta que lo ven en terminos físicos y trabajan con él. Ahí es donde entra en juego la elaboración de prototipos. Esta técnica es más avanzada que el bosquejo de los componentes estructurales y el bosquejo en papel en blanco por el hecho de que el usuario trabaja efectivamente con una imitación del sistema que se va a implementar. Si a los usuarios no les gusta, entonces se construye otro modelo para su prueba y evaluación. Aplicada de esta forma, la elaboración de prototipos es esencialmente la construcción de sistemas mediante aprendizaje y descubrimiento.

En general, la elaboración de prototipos no es aplicable para sistemas estándar de funciones específicas como contabilidad, pero es particularmente útil para el desarrollo de proyectos de sistemas únicos e innovadores que traen consigo un gran número de beneficios cualitativos o en el diseño de subsistemas pequeños y limitados para satisfacer las necesidades especiales de reportes y toma de decisiones de uno o unos cuantos usuarios. La elaboración de prototipos también puede utilizarse eficazmente para afinar y mejorar partes del sistema existente.

Típicamente, los analistas de sistemas desarrollan prototipos centrándose en el componente estructural de la salida, las entidades y atributos de datos, y en las interfaces usuario/sistema, considerando solamente en forma secundaria otros aspectos como los controles y la tecnología.

Los analistas de sistemas, trabajando conjuntamente con los usuarios, crean diferentes reportes o salidas por pantalla empleando lenguajes de consulta o generadores de reportes. Los analistas de sistemas discuten con los usuarios las necesidades de éstos y luego desarrollan prototipos en la terminal conjuntamente con los usuarios. Si las necesidades son complejas, los analistas pueden requerir una semana aproximadamente para producir una muestra. En cualquier caso, los prototipos deben demostrarse. Los usuarios ven como trabajan y los utilizan. A continuación, los usuarios con frecuencia solicitan cambios y los analistas de sistemas crean más prototipos

mostrando diferentes *formas de salida*, como tablas y gráficas. A través de cada iteración, los prototipos se mejoran hasta que satisfacen con precisión los requerimientos de los usuarios.

Se pueden crear varios prototipos en una terminal. La necesidad de una programación clásica de computadoras se minimiza gracias a la interacción en línea con las bases de datos relacionales y los lenguajes de consulta, lenguajes fáciles de usar para la creación de modelos, teclas de función, menús y una variedad de formas en pantalla.

Obviamente, el ciclo de vida de los prototipos es mucho más corto que el ciclo de vida de la metodología de desarrollo de sistemas (MDS). La metodología para la elaboración de prototipos generalmente no es tan formal como la de la MDS, pero aún así también sigue una metodología. En realidad, la elaboración de prototipos es un microcosmos de la MDS, excepto que la primera produce un modelo de trabajo físico del diseño propuesto, en tanto que los diseños generales de sistemas propuestos por la MDS normalmente son más conceptuales y más amplios en alcance.

En la figura 1.1.4 se ilustra como funciona la elaboración de prototipos con la MDS.

Si el prototipo aceptado es parte de un sistema más grande, entonces se integra en el diseño general de sistemas y se desarrolla de acuerdo a las fases restantes de la MDS. Sirve como base para un conjunto completo y válido de requerimientos de los usuarios, y la MDS mantiene el avance del proyecto de sistemas. Por otra parte, si el prototipo aceptado es un sistema único o independiente que pueda implementarse, entonces el analista de sistemas realiza las tareas necesarias para convertir directamente el nuevo prototipo en una operación total.

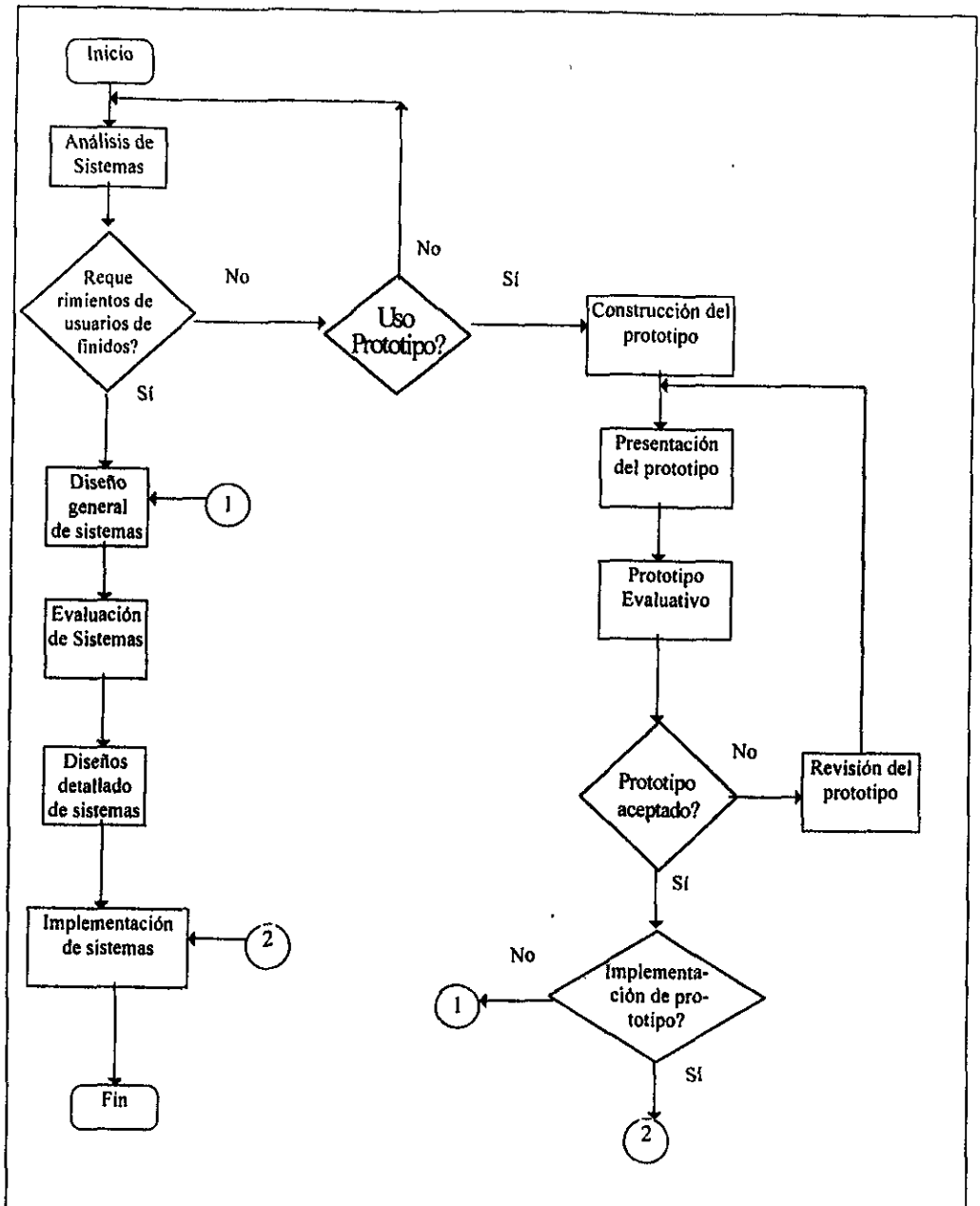


Figura 1.1.4 Diagrama de flujo de cómo se emplea un prototipo para ayudar a definir los requerimientos de los usuarios y diseñar sistemas

## Metodologías de Desarrollo

Dentro de las metodologías de desarrollo, se pueden distinguir tres corrientes :

- La francesa, que dió como fruto la metodología MERISE, potenciada por la administración francesa a partir del año 1977.
- La inglesa, también impulsada por la administración en Gran Bretaña y que dió lugar, a partir de 1981, al SSADM.
- La americana, basada en las teorías de EDWARD YOURDON y que tiene algunas variantes aportadas por otros autores como DEMARCO, GANE Y JAMES MARTIN.

El resto de las metodologías existentes, tanto públicas como privadas, deben considerarse como adaptaciones, más o menos mejoradas, de las citadas anteriormente, así mismo, las metodologías de Orientación a Objetos está fuera del alcance de esta tesis.

### Metodología Merise

Surge en Francia a partir del año 1977, tras una petición del Ministerio de Industria, como un intento de definir una metodología a emplear en la Administración Pública para el desarrollo y diseño de sistemas informáticos.

Esta petición se realiza en un momento en que la gran diversidad de lenguajes y formalismos empleados para la representación de sistemas informáticos hacen necesario un esfuerzo de unificación de criterios y métodos. En un momento en que hay que tratar de eliminar los problemas derivados de la rotación de personal.

Los principios generales en que se apoya MERISE son :

- Desglose del desarrollo en etapas.



- Definición de los documentos estándar de cada una.
- Uso del modelo ENTIDAD/RELACION y sus formalismos para la representación de datos.
- Uso de las REDES DE PETRI para la representación de procesos y tratamientos.
- Definición de grupos de trabajo, reparto de las responsabilidades y funciones a lo largo del desarrollo.
- Especificación del reparto de tareas y tratamientos entre los usuarios y el ordenador.
- Definición de los flujos de información entre las unidades del sistema.

El sistema se contempla desde diferentes niveles de abstracción y esto da lugar a una descripción del mismo a tres niveles : conceptual, lógico u organizativo y físico .

En la fase de concepción se trabaja básicamente sobre dos elementos : datos y tratamientos. La descripción de los datos reflejará la información existente en el entorno y las relaciones entre ellas. La representación de los tratamientos reflejará los procesos a realizar con los datos así como su secuencia en el tiempo. Con la descripción de estos dos elementos habremos conseguido reflejar tanto el contenido del sistema como su funcionamiento. Se podría establecer el siguiente cuadro (ver figura 1.1.5) en cuanto a los diferentes niveles, tanto de decisión como de descripción de datos y tratamientos :

NIVELES DE DECISIÓN	NIVEL DE DESCRIPCIÓN	
	TRATAMIENTOS	DATOS
Conceptual	Conceptual	Conceptual
Organizativo	Organizativo	Lógico
Técnico	Operativo	Físico

Figura 1.1.5 Niveles de decisión y de descripción.

Veremos en primer lugar lo que significan estos niveles y cual es su contenido. Empecemos por los tratamientos.

En cuanto a la DESCRIPCIÓN DE TRATAMIENTOS, el NIVEL CONCEPTUAL consiste en la descripción del QUÉ hay que hacer, es decir, en la descripción en términos de operaciones y resultados de la gestión que debe resolver el sistema independientemente de quién sea el que la realice, hombre o máquina, y de qué modo.

En el NIVEL ORGANIZATIVO, se desglosan las operaciones descritas a nivel conceptual en procedimientos funcionales, es decir, en tareas realizadas sucesivamente en un puesto de trabajo. A este nivel se concreta ya QUIÉN, CUÁNDO y DÓNDE se han de realizar estas tareas.

En el NIVEL OPERATIVO se responde a la pregunta de CÓMO hay que hacer las cosas. Se detallan, tanto para procedimientos automatizados como para los manuales, las normas para realizarlos correctamente.

Veamos a continuación como se estructura la descripción de los datos desde diferentes niveles de abstracción.

En la DESCRIPCIÓN DE DATOS contemplamos en primer lugar un NIVEL CONCEPTUAL, en el que se observa la información del sistema en términos de objetos o entidades, se describen sus propiedades, la información de cada uno de ellos y las relaciones entre los mismos. Este modelo conceptual de datos es, en principio, bastante estable a lo largo del ciclo de vida del sistema.

En el NIVEL LÓGICO se traduce el modelo conceptual en agrupaciones o estructuras lógicas de datos para su tratamiento por el sistema. Todavía este nuevo modelo debe ser independiente de la opción técnica a elegir en cuanto a soporte de software del mismo.

En el nivel más bajo de la descripción de datos es decir, en el NIVEL FÍSICO, se concreta ya cual va a ser la estructura final de los datos de acuerdo al sistema gestor elegido (base de datos, tipos de ficheros, etc) Una vez conocido esto se podrán hacer optimizaciones del modelo lógico para mejorar rendimientos.

En estos dos procesos, descripción de datos y de tratamientos, se hacen necesarias validaciones para comprobar la coherencia entre unos y otros. En este sentido se podrían estructurar los pasos a seguir en el siguiente diagrama (ver figura 1.1.6) que propone un estudio paralelo de datos y tratamientos.

Por otra parte, MERISE como todas las metodologías, propone una serie de etapas en el desarrollo de un proyecto.

- Estudio preliminar.
- Estudio detallado.
- Realización.
- Puesta en marcha o implantación.

Esto es lo que compone el ciclo de vida del sistema.

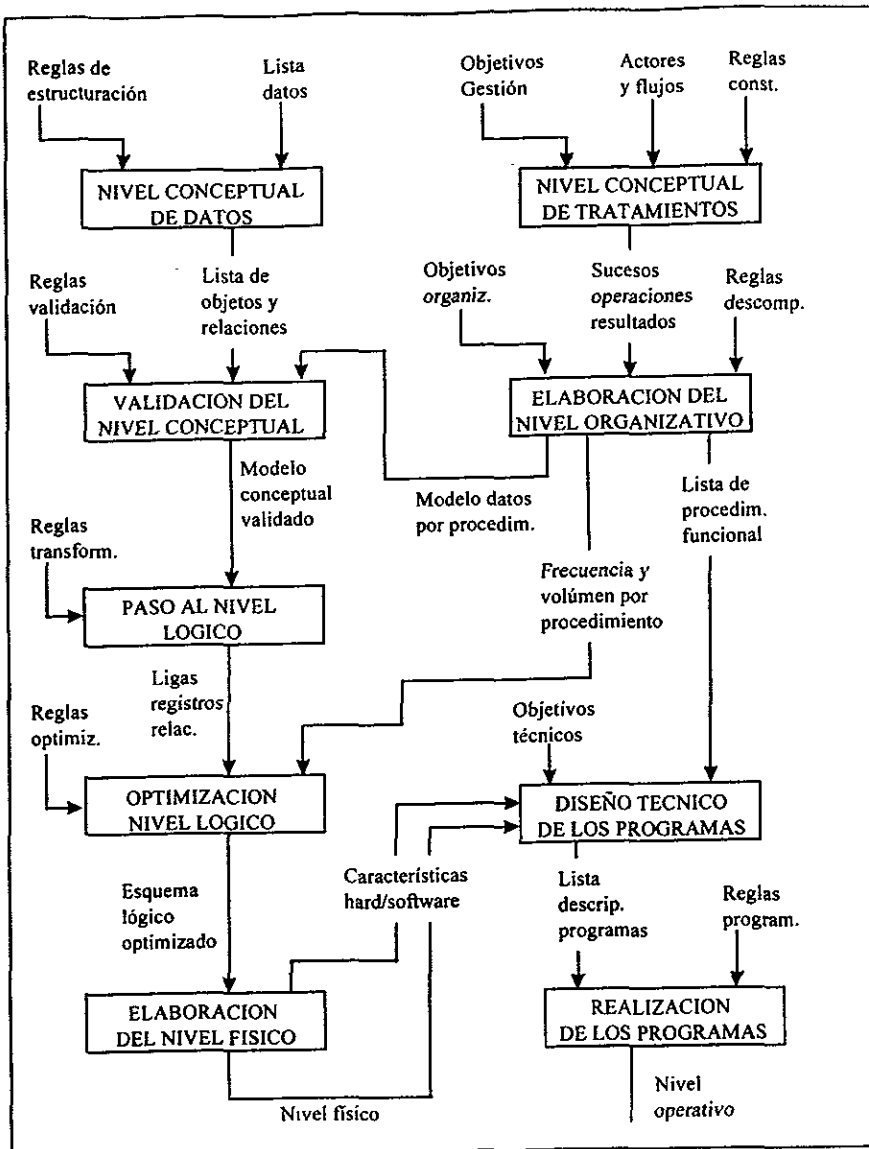


Figura 1.1.6 Niveles de descripción de datos y tratamientos.

Veremos ahora el conjunto de acciones a realizar en las etapas propuestas por MERISE y su estructuración en distintas fases. El índice general de etapas y fases sería el siguiente :

## ETAPA 1 : ESTUDIO PRELIMINAR

### FASE 1 : Recopilación de datos

- Recopilación inicial
- Estudio de la situación actual
- Síntesis y crítica de la situación actual

### FASE 2 : Concepción de la nueva solución

- Objetivos a alcanzar
- Descripción de la solución

### FASE 3 : Evaluación y plan de desarrollo

- Evaluación de la nueva solución
- Plan de desarrollo

## ETAPA 2 : ESTUDIO DETALLADO

### FASE 1 : Concepción general

### FASE 2 : Concepción detallada de las fases

- Realización de las especificaciones detalladas de los procesos

### FASE 3 : Plan de desarrollo

## ETAPA 3 : REALIZACIÓN

### FASE 1 : Estudio técnico

### FASE 2 : Producción

## ETAPA 4 : PUESTA EN MARCHA

FASE 1 : Preparación de los recursos físicos y humanos

FASE 2 : Recepción y lanzamiento del sistema

Otro punto contemplado por MERISE es la descripción de los diferentes grupos de trabajo, es decir, el personal implicado en el desarrollo del sistema a todos los niveles. Así, habrá :

- **COMITÉ DIRECTOR** que indicará los objetivos a cubrir por el sistema en las primeras fases de concepción.
- **COMITÉ DE USUARIOS** que aportará el punto de vista de los futuros utilizadores del mismo.
- **GRUPO DE DESARROLLO** o personal técnico encargado de satisfacer esos requerimientos a través del sistema informático.

### Metodología SSADM

Las siglas de esta metodología corresponden a las iniciales de Structured System Analysis and Design Method. Nace en el Reino Unido a petición del Gobierno Inglés con la intención de ser un sistema de desarrollo público de Gran Bretaña. En la elaboración de esta metodología intervienen dos empresas consultoras : Learmonth and Burchett Management Systems y Central Computer and Telecommunications Agency.

Es aceptada en 1981 en que se presenta la primera versión de la misma. Se hace uso obligatorio en toda la administración en 1983, aunque se siguen sacando sucesivas versiones hasta el modelo

actual que van incorporando mejoras o contemplando fases del ciclo de vida no cubiertas en un principio.

La metodología consiste en una estructuración de los pasos a seguir en el desarrollo de un proyecto informático en las fases iniciales del ciclo de vida del mismo y en la descripción de unas técnicas y formalismos sobre las que se basan los trabajos a realizar en cada fase.

Así, según un carácter puramente jerárquico, podríamos distinguir de la figura 1.1.7

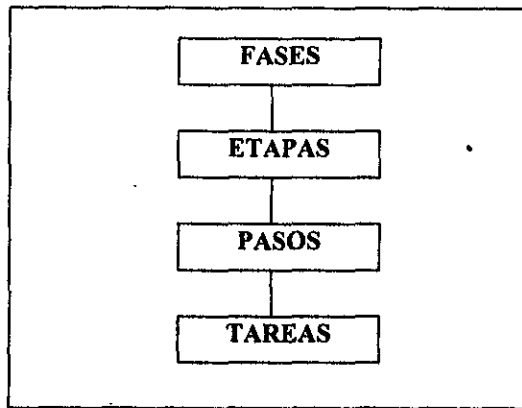


Figura 1.1.7 Estructuración de pasos en un proyecto informático.

El método contempla, en principio, las tres primeras fases del desarrollo (estudio de viabilidad, análisis y diseño) divididas a su vez en una serie de etapas, esto se muestra en la figura 1.1.8:

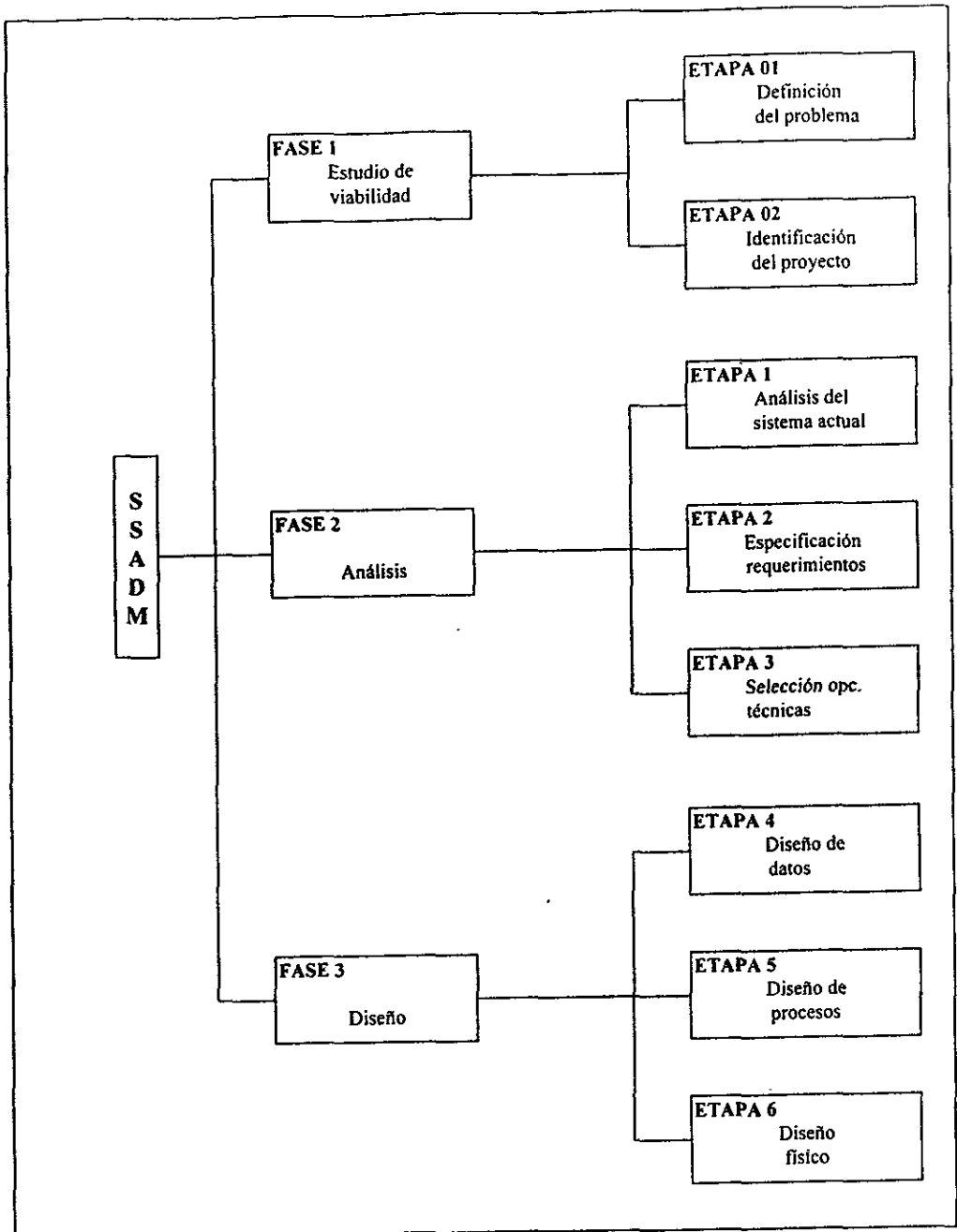


Figura 1.1.8 Fases y etapas de SSADM.



Cronológicamente, las seis etapas, una vez comprobada la viabilidad del proyecto, se desarrollarán en el siguiente orden :

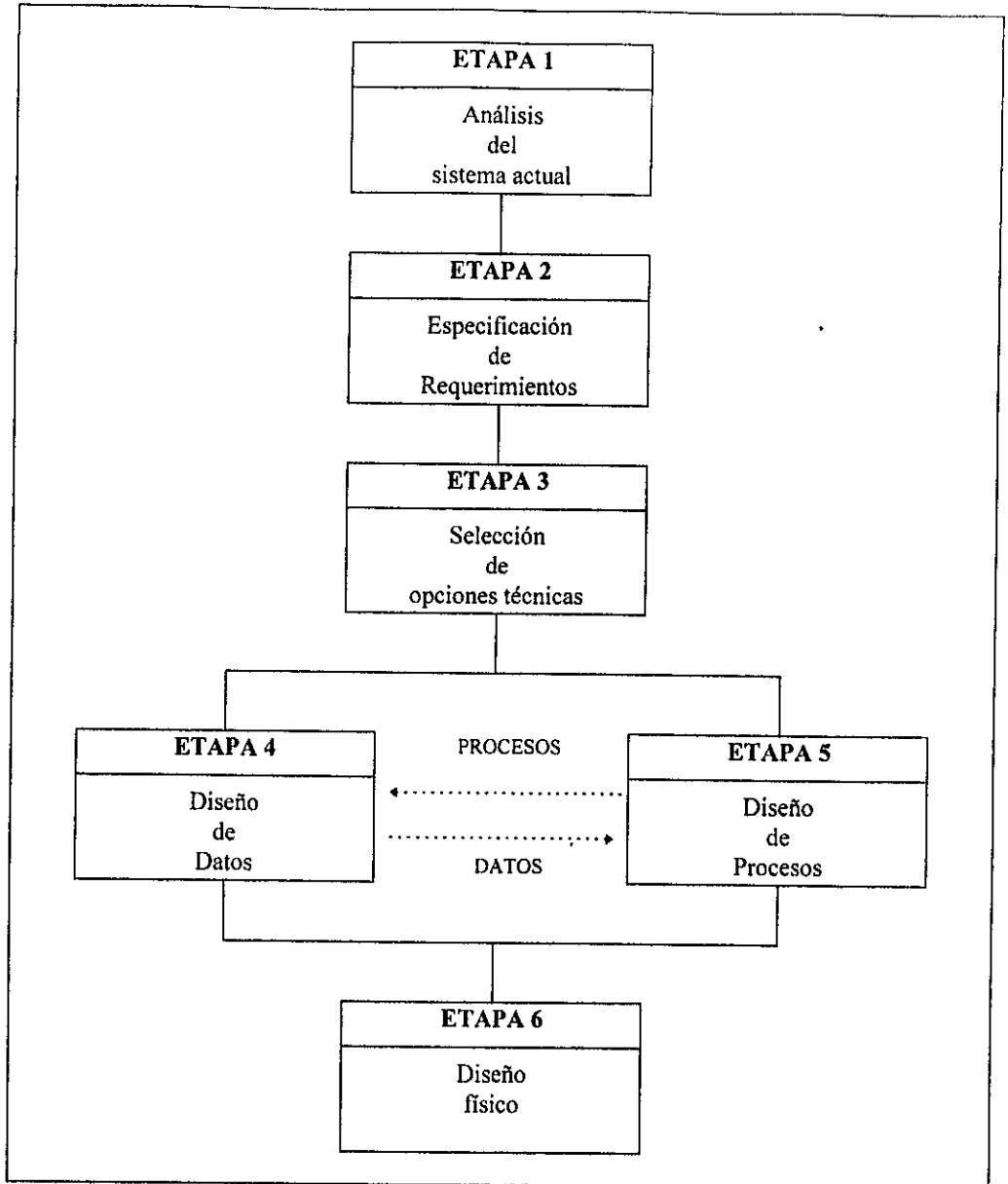


Figura 1.1.9 Orden de ejecución de las etapas.

Las técnicas utilizadas por SSADM en cada una de las fases y etapas son :

- Diagramas de Flujo de Datos (DFD), que son una formal representación de los flujos de información en el interior del sistema contemplado y entre el sistema y el exterior, es decir sus relaciones con otros.
- Estructura Lógica de Datos (LDS). Mediante la representación de las entidades del sistema y las relaciones entre ellas. Para ello se utilizarán los formalismos y teorías del modelo ya visto entidad/relación.
- Historia en la Vida de la Entidad (ELH). Representa la descripción de como las entidades descritas son afectadas por diferentes sucesos que ocurren en el sistema.
- Tercera Forma Normal (TFN), en la descripción de datos. Es un método matemático para la definición de datos que ayuda a evitar inconsistencias y ambigüedades en la estructura de los mismos.

### **Metodología Yourdon**

Es el representante de la corriente metodológica más importante de Estados Unidos, aunque hay numerosos autores que aportan variantes, matices y formalismos de representación al método de Yourdon.

A lo largo de sus obras Yourdon describe técnicas para la realización de análisis estructurado de sistemas basado principalmente en los siguientes conceptos :

- Diagramas de flujo de datos para la representación de procesos.
- Diagramas de transición de estados para la representación estructurada de las funciones a realizar en los procesos.

- Modelo Entidad/Relación para la representación conceptual de datos.
- Diccionario de datos como base o soporte de información del sistema.
- Diagramas o mapas de estructura para la representación modular de los procesos y las variables intercambiadas entre ellos.
- Especificaciones de programas basadas en lenguaje estructurado y tablas de decisión.

Yourdon define las siguientes etapas y niveles en el ciclo de vida de los sistemas informáticos.

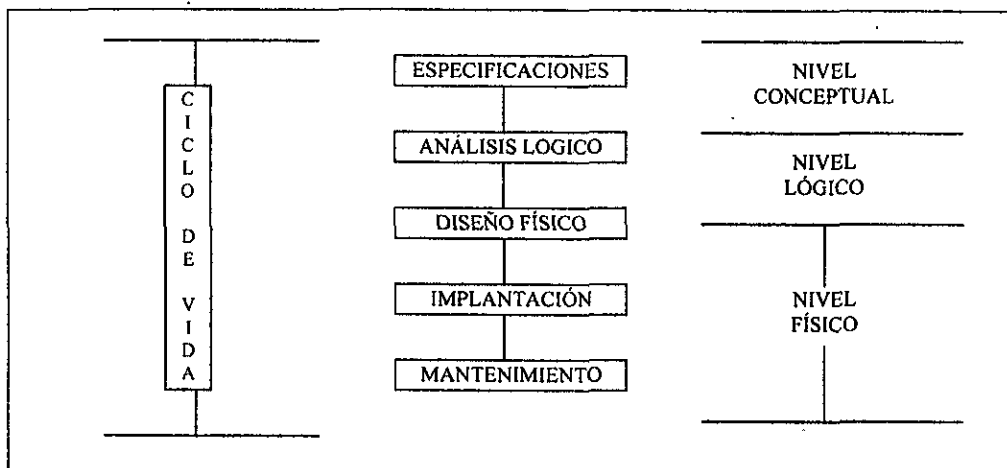


Figura 1.1.10 Ciclo de vida según Yourdon.

Por tanto aplica las técnicas descritas anteriormente a lo largo de estas etapas, distinguiendo las siguientes actividades a realizar :

- **Estudio de Viabilidad**

En este punto se debe identificar el proyecto a realizar, los usuarios responsables y se debe hacer un estudio de la situación actual, representando la misma a través de Diagramas de Flujo de Datos de primer nivel o diagramas de contexto en los que, de forma simple, se indiquen los procesos simples más relevantes. En este estudio de la situación actual se identificarán las

deficiencias del mismo como puntos a resolver por el nuevo sistema. Esta actividad debe ocupar entre un 5 y un 10% del total del proyecto.

- **Análisis del Sistema**

En esta fase se debe tener en cuenta los requerimientos de los usuarios relativos a cambios o funcionalidad del sistema y esta labor es facilitada por herramientas CASE de diagramación y por técnicas de prototipado del sistema.

- **Diseño**

En esta tercera fase, se pasa del nivel conceptual descrito anteriormente a un nivel de representación lógica de los datos mediante un diseño dependiente del modelo de base de datos elegida (CODASYL, relacional, etc) y una estructuración de los procesos utilizando diagramas de estructura de los mismos y generando las especificaciones de programas correspondientes.

- **Implementación o Producción**

Comprende la generación de código y el ensamblaje e integración de todos los módulos.

- **Pruebas del Sistema**

A realizar con la totalidad del sistema hasta llegar a la aceptación del mismo por parte del usuario. En esta fase se harán pruebas de integración y de funcionamiento conjunto de programas y cadenas.

- **Control de Calidad**

El objeto de esta actividad es garantizar los controles de calidad del software que puedan estar definidos para la empresa. Esta fase complementa la anterior de forma que el producto final sea de un buen nivel de calidad y cumpla los estándares fijados.

- **Documentación**

En este apartado se generará toda la documentación necesaria para la instalación del sistema: manuales de usuario, de operación, etc. La documentación interna, o sea, las especificaciones de programas, habrán sido creadas ya anteriormente y utilizadas por los programadores. Con ello el nuevo sistema queda completamente documentado, interna y externamente.

- **Conversión de los Datos del Sistema Anterior**

La ejecución de esta fase depende, evidentemente del estado anterior a la mecanización del entorno afectado por el proyecto. Si existía ya un sistema informatizado se deben realizar los programas de conversión de datos al nuevo sistema y si anteriormente los archivos eran manuales puede requerirse una grabación y carga previa a la puesta en marcha del sistema.

- **Instalación**

Comprende la puesta en marcha del sistema y en esta fase son de aplicación las consideraciones hechas acerca de aspectos tales como formación y entrenamiento del usuario, entrega de manuales, procesos paralelos, etc.

### **Estudio Comparativo entre las diferentes Técnicas**

Evidentemente, en las tres metodologías expuestas hay puntos comunes y técnicas similares en determinadas fases del desarrollo. Para establecer una comparación entre ellas hay que tomar

como base esos puntos comunes que toda metodología debe contemplar y después cómo son abordados estos puntos desde cada una de las metodologías. Tomaremos como referencia lo siguiente:

- Etapas del Ciclo de Vida
- Modelado de Datos
- Modelado de Procesos

### **Etapas del Ciclo de Vida**

En general todas ellas estructuran el desarrollo en diversas etapas. Si examinamos esta estructuración se puede observar una mayor precisión en cuanto a los pasos a seguir en la metodología SSADM. Resulta ser casi un recetario, siguiéndolo tal cual podemos desarrollar cualquier proyecto.

Hay que resaltar que ninguna de las tres llega a determinar qué técnicas de programación estructurada se debe seguir, aunque todas recomiendan su uso. En el caso de SSADM incluso no se llega a esta etapa y la metodología llega hasta el diseño físico. Se podría decir que MERISE es la que más trascendencia da al estudio organizativo de la empresa como base de todo el sistema informático. La selección de las personas que van a intervenir en el proyecto a través de los Comités de Dirección, Usuarios y Equipo de Desarrollo y en qué momentos o fases intervienen cada uno de ellos, hacen incidencia en el trasfondo organizativo que MERISE quiere aportar.

### **Modelado de Datos.**

Es común en todas las metodologías la definición de los niveles conceptual, lógico y físico en la definición de los datos. A nivel conceptual, se admite por parte de todas el modelo entidad/relación como el más adecuado para reflejar la información del sistema. También es una tendencia general del mercado la utilización de bases de datos relacionales como soporte del nivel lógico del modelo de datos. Desde este punto de vista, todas recomiendan la normalización

de estructuras siguiendo las reglas de normalización de Codd para después desnormalizar de forma controlada y por razones de rendimiento y tiempos de respuesta del sistema.

### **Modelado de Procesos**

Nuevamente MERISE introduce una descripción más precisa de los mismos a través de las redes de Petri. En ellas la información acerca de los procesos, del qué hay que hacer, cómo, cuándo y quiénes es muy completa. Esta técnica puede tener el inconveniente de la dificultad no sólo para el que la realiza sino también para la validación por parte del usuario por su mayor complejidad y nivel de detalle.

Las otras dos metodologías realizan esta modelización a través de los Diagramas de Flujo de Datos de diferente nivel de detalle. Los Diagramas de Flujo de Datos resultan claros para el usuario y relativamente fáciles de realizar. Evidentemente, si hay un apoyo en una herramienta CASE de diagramación, esta técnica resulta cómoda de realizar y de interpretar por el usuario.

De cualquier forma, las redes de Petri serán una opción más en las herramientas de soporte de la metodología SSADM en sus próximas versiones.

## 1.2 CONCEPTOS GENERALES DEL MANEJADOR DE BASE DE DATOS ORACLE

### Bases de Datos y Manejo de la información

Un servidor de base de datos es la llave para resolver los problemas de manejo de la información. En general un servidor debe ser capaz de manejar grandes cantidades de datos en un ambiente multiusuario, de tal manera que muchos usuarios puedan acceder los datos concurrentemente. Todo esto debe estar acompañado de un alto nivel de rendimiento, así como también prevenir accesos no autorizados y poder proveer soluciones eficientes para recuperaciones a fallas.

El Servidor Oracle provee soluciones eficientes y efectivas con las siguientes características:

- *Arquitectura cliente-servidor (Procesamiento distribuido)* .

La computadora donde se encuentra la Base de Datos maneja todas las responsabilidades del servidor de base de datos mientras las estaciones de trabajo corren la aplicación y se encargan de desplegar los datos.

- *Capacidad para manejar grandes volúmenes de información*  
Oracle maneja las bases de datos más grandes, hasta Terabytes .

- *Usuarios concurrentes.*

Oracle es capaz de manejar un gran número de usuarios concurrentes ejecutando diversas aplicaciones minimizando la contención de datos.

- *Alta disponibilidad*

En muchos centros de cómputo Oracle trabaja las 24 horas por día, los 365 días del año sin degradación del rendimiento.



- Control de disponibilidad

Oracle puede controlar la disponibilidad de datos, base de datos y datos por aplicación. Por ejemplo, el administrador puede deshabilitar el uso de una aplicación específica para cambiar los datos de los catálogos sin afectar a las demás aplicaciones que se encuentran en la misma base de datos.

- Abierto a estándares industriales.

Oracle se adhiere a los estándares aceptados por la industria para lenguajes de acceso de datos, sistemas operativos e interfaces de usuario y protocolos de comunicación. Es un sistema abierto que protege la inversión hecha por sus clientes.

- Manejo de seguridad

Para proteger contra accesos no autorizados, Oracle provee características para limitar, monitorear y auditar el acceso a datos. Estas características pueden ser tan fáciles o tan complejas de acuerdo a las necesidades del cliente.

- Integridad de la base de datos

Se fuerza la integridad de datos a través de las reglas del negocio que dictan los estándares para aceptar los datos. Como resultado de esto el costo de codificar y manejar el chequeo a nivel aplicación es eliminado.

- Sistema distribuido

Para redes de ambiente distribuido, Oracle permite combinar una sola base de datos lógica en varias bases de datos localizadas en diferentes máquinas y/o centros de cómputo, también se puede tener acceso a la información de bases de datos no Oracle como parte de la base de datos distribuida.

- **Portabilidad**

Las bases de datos pueden ser portadas a través de diferentes sistemas operativos sin ningún cambio. Lo mismo sucede con las aplicaciones que pueden correr en diferentes sistemas operativos con pequeños o con ningún cambio.

- **Conectividad**

Es posible tener software para diferentes tipos de computadoras y diferentes tipos de sistemas operativos para compartir información a través de una red.

- **Replicación**

Es posible replicar grupos de tablas en múltiples centros de cómputo. Oracle soporta replicación a nivel tabla o esquema, (Conjunto de objetos de un usuario). También se encuentra disponible la replicación primaria del centro de cómputo como modelos más avanzados y complejos de replicación.

### **Servidor Oracle**

El servidor Oracle es un RDBMS (Sistema Manejador de Bases de Datos Relacional) que consiste de una base de datos Oracle y de una o mas instancias Oracle. El concepto instancia se verá más adelante.

### **RDBMS**

Los sistemas manejadores de bases de datos han evolucionado desde las jerárquicas hasta el modelo relacional, que actualmente es el mas aceptado. El cual tiene 3 aspectos principales:

- **Estructuras**

Objetos bien definidos, tales como tablas, vistas, índices, etcétera. Las estructuras y los datos contenidos que pueden ser manipulados mediante operaciones.

- **Operaciones**  
Operaciones bien definidas que permiten a los usuarios manipular datos y estructuras de la base. Las operaciones de la base deben apegarse a un conjunto de reglas de integridad pre-definidas.
- **Reglas de integridad**  
Estas reglas son las leyes que gobiernan todas las operaciones en los datos y estructuras de la base de datos.

Entre las ventajas que tiene el RDBMS, se tiene:

- Independencia de almacenamiento físico y lógico
- Facilidad de acceder a todos los datos
- Flexibilidad en el diseño de la base de datos
- Mínima redundancia de datos

### **SQL (Lenguaje Estructurado de Consulta)**

SQL es el lenguaje de programación que define y manipula la base de datos, las bases de datos definidas por SQL son relacionales, esto significa que los datos están almacenados con un conjunto de relaciones entre ellos. Una base de datos puede tener una o más tablas. Cada tabla tiene columnas y renglones.

Todos los comandos de SQL se agrupan en dos partes:

**DDL** (Lenguaje de definición de datos), que permite crear y modificar bases de datos, tablas, índices, etc.

**DML** (Lenguaje de manipulación de datos), mediante el cual se puede actualizar, consultar, borrar e insertar datos.

Además de SQL, Oracle cuenta con un lenguaje procedural llamado PL/SQL que permite utilizar comandos SQL con estructuras repetitivas, de control y manejo de errores.

## **Estructura de la Base de Datos**

Una base de datos Oracle tiene dos estructuras: la física y la lógica.

### **a) Estructura Física**

La estructura física está determinada por los archivos de sistema operativo que forman a la base de datos. Se tienen tres tipos de archivos:

- **Archivos de Datos (Datafiles)**

Los archivos de datos contienen toda la información de una base de datos. Las estructuras lógicas como tablas e índices están almacenados físicamente en estos archivos, un archivo de datos puede estar asociado a solo una base de datos y además pueden crecer o decrecer dinámicamente.

Los datos de un archivo de datos son leídos a solicitud de los usuarios y son almacenados en memoria cache . Si se modifica o se insertan nuevos datos no se escriben inmediatamente a disco en los archivos de datos sino que se utilizan estructuras de memoria como son los redo log buffers y archivos de log en disco para agilizar esta operación.

- **Archivos de Log**

Al conjunto de archivos de log se les conoce como archivos redo log, la función primaria de éstos es guardar todos los cambios que son hechos a los datos.

Estos archivos son la pieza clave para proteger a la base de datos contra fallas. Como pueden ser multiplexados, generalmente se tiene más de una copia de cada uno de ellos en diferentes discos.

La información contenida en estos archivos es utilizada solamente para recuperar la base de datos en caso de fallas, tales como una suspensión de energía eléctrica con la base de datos trabajando.

- Archivos de Control

Cada base de datos debe tener al menos un archivo de control. Este archivo tiene información de como está la estructura física de la base de datos, por ejemplo tiene la siguiente información:

- nombre de la base
- nombres y ubicaciones de los archivos de datos y de los archivos de log
- fecha y hora de la creación de la base

Cada vez que una instancia Oracle es levantada, el archivo de control es utilizado para identificar la base de datos y los archivos de toda la base que deben ser abiertos para que los usuarios puedan acceder a la información.

## b) Estructura Lógica

La estructura lógica de la base de datos está determinada por

- Los objetos de los esquemas. Un esquema es una colección de objetos. Los objetos de los esquemas son estructuras lógicas que directamente hacen referencia a los datos de la base de datos. Los objetos de los esquemas incluyen tablas, vistas, índices, procedimientos almacenados, secuencias, sinónimos, clusters (almacenamiento especial para dos o más tablas) y ligas que permitan acceder a la información de dos o más bases de datos remotas, llamadas **database links**.
- **Tablespaces**. Es el nombre técnico para identificar en dónde se almacenan lógicamente los objetos de los esquemas. Una base de datos se encuentra dividida en unidades lógicas llamadas tablespaces, un tablespace es usado para contener estructuras lógicas juntas, por

ejemplo el tablespace para contabilidad contiene todas las tablas de la aplicación de contabilidad.

La estructura lógica incluye tablespaces, segmentos y extents (conjunto de bloques que forman un objeto, este puede ser una tabla o un índice).

La relación entre bases de datos, datafiles y tablespaces se encuentra ilustrada en la figura 1.2.1

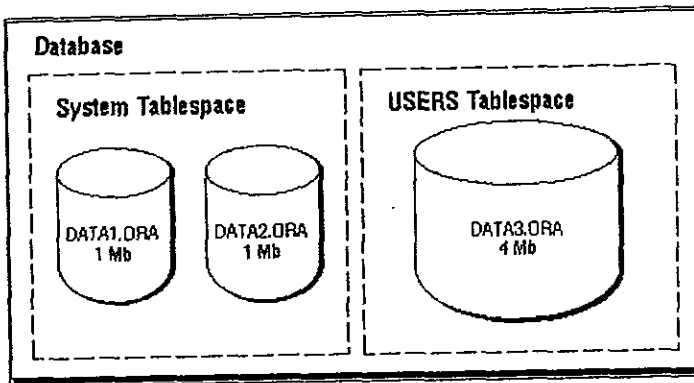


Figura 1.2.1 Base de datos, Datafiles y Tablespaces.

La figura 1.2.1 ilustra:

- Cada base de datos está dividida en uno o mas tablespaces
- Uno o mas archivos de datos son creados para cada tablespace para almacenar físicamente los datos de todas las estructuras lógicas de ese tablespace.
- La suma del espacio ocupado por los archivos de datos que forman un tablespace es el espacio del tablespace.
- La suma de todos los tablespaces es el total de espacio para datos de la base de datos.

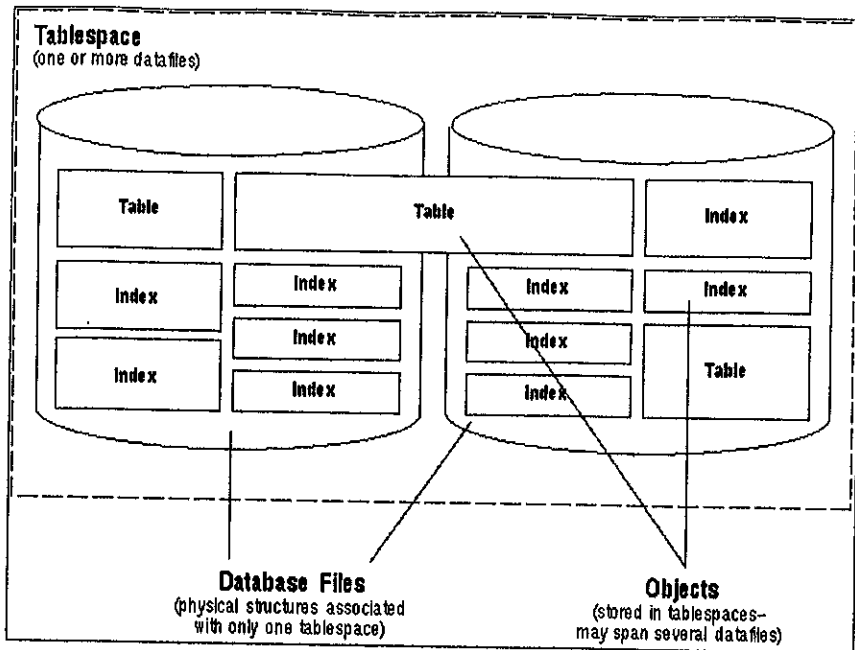


Figura 1.2.2 Datafiles, Tablespaces, Segmentos y Extents.

La figura anterior muestra como una tabla puede estar alojada en dos archivos de datos que pertenecen a un mismo tablespace.

### Esquemas y Objetos de Esquema

Como habíamos mencionado anteriormente un esquema es un conjunto de objetos, dentro de los objetos que pueden encontrarse bajo un esquema tenemos tablas, índices, vistas, etc.

No existe una relación directa entre los tablespaces y los esquemas, por ejemplo los diferentes objetos de un esquema pueden encontrarse en diferentes tablespaces y a su vez un tablespace puede contener objetos de diferentes esquemas.

## Tablas

Una tabla es la unidad básica de almacenamiento de datos. Las tablas contienen todos los datos que son accesibles a los usuarios. Una tabla está compuesta por renglones y columnas. Cada columna tiene un nombre determinado, un tipo de dato y una longitud máxima y opcionalmente una precisión (solo para datos numéricos). Una vez que una tabla ha sido creada, renglones válidos se pueden empezar a insertar. Para forzar las reglas del negocio se cuenta con **constraints** (restricciones que se deben cumplir)

## Vistas

Una vista es una selección sobre una o más tablas con ciertos criterios. A las vistas también se les conoce como **query** (consulta) programada. Las vistas no almacenan datos por sí mismas, sino que consultan las tablas. De la misma manera que las tablas, con las vistas se pueden consultar, actualizar, insertar y borrar datos. Las vistas son usadas para garantizar un nivel adicional de seguridad, como por ejemplo para ocultar a ciertos usuarios campos confidenciales, como salarios. También se utilizan para manejar datos complejos de manera eficiente, por ejemplo una vista puede contener los datos de las doce tablas mensuales de ventas.

## Secuencias

Una secuencia es una tabla generadora de números secuenciales únicos. Las secuencias facilitan ciertas aplicaciones porque automáticamente generan números secuenciales.

## Unidades de Programación

Este término es usado para referirse a los procedimientos, funciones y paquetes almacenados en la base de datos. Un procedimiento o función es un programa en PL que es almacenado en la base de datos. Los procedimientos o funciones permiten combinar la facilidad y flexibilidad de



SQL con un lenguaje procedural. La única diferencia entre funciones y procedimientos es que las funciones regresan un solo valor, mientras que los procedimientos no regresan ningún valor.

Los paquetes permiten encapsular y almacenar procedimientos y funciones que tienen relación entre sí. Los paquetes facilitan la administración del código e incrementan el rendimiento de la base de datos.

### **Sinónimos**

Un sinónimo es una alias de una tabla, vista, secuencia o unidad de programación. Un sinónimo no es un objeto en sí mismo, pero hace referencia a un objeto. Los sinónimos son usados para:

- ocultar el nombre y el dueño reales del objeto
- permitir acceso público a un objeto
- permite transparencia para tablas, vistas y unidades de programación remotas.
- simplifican las instrucciones sql para los usuarios.

Los sinónimos pueden ser públicos, si todos los usuarios tienen acceso a él, o privados.

### **Índices, Clusters y Clusters de Hash**

Los índices, clusters y clusters de hash son estructuras opcionales asociadas con tablas, que permiten incrementar el rendimiento de la extracción de datos.

Los índices son creados para mejorar el rendimiento de la extracción de datos. Un índice ayuda a localizar la información más rápido que si no existiera éste. Cuando se está procesando un requerimiento, Oracle puede utilizar alguno o todos los índices para localizar los renglones requeridos mas eficientemente. Los índices son útiles cuando las aplicaciones consultan continuamente una tabla para un rango de renglones, por ejemplo todos los empleados con un salario mayor que 1,000 o un renglón en específico. Los índices se pueden crear en una o mas

columnas de una tabla. Una vez que son creados, el mantenimiento se realiza de manera automática para el usuario. Los cambios a la tabla, como agregar más renglones, actualizar y/o borrar son automáticamente incorporados en todos los índices relevantes con completa transparencia para los usuarios. Los índices son lógicamente y físicamente independientes de las tablas, esto es que son almacenados de manera independiente a la tabla que pertenecen. Pueden ser borrados sin afectar a la tabla ni a los demás índices de esa tabla.

Los clusters son un método alternativo para almacenar tablas. Los cluster son grupos de una o más tablas físicamente almacenadas juntas debido a que comparten columnas en común y son accedidas casi siempre juntas. Debido a que los renglones son almacenados conjuntamente el tiempo de acceso a disco es menor que cuando se guardan de manera separada.

La siguiente figura muestra un par de tablas almacenadas independientes y almacenadas en un cluster.

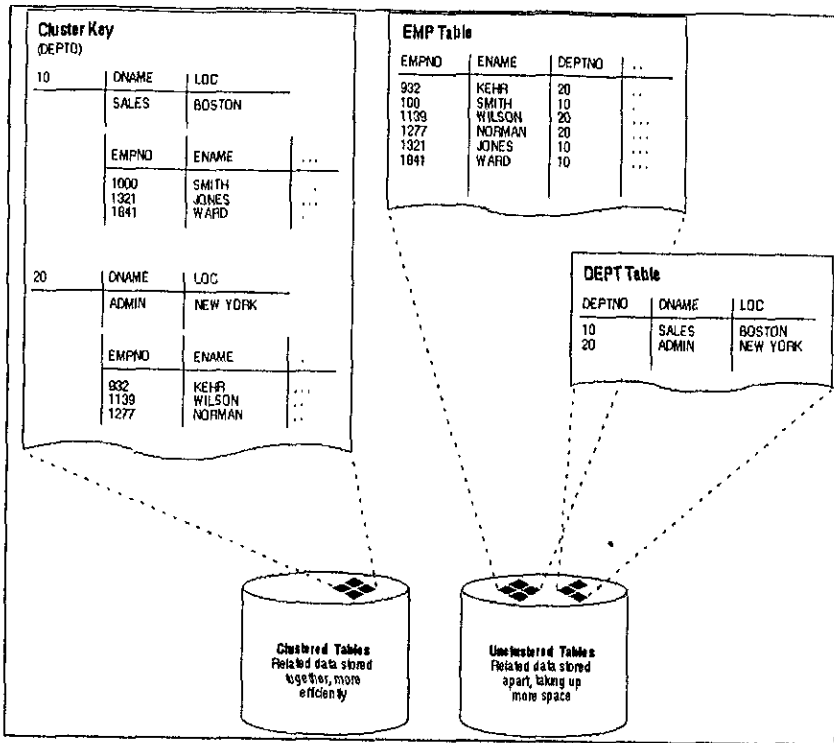


Figura 1.2.3 Tablas en cluster y tablas sin cluster.

Los cluster de hash son datos almacenados en forma similar a la normal con índices de hash. Un renglón es almacenado en un cluster de hash basado en el resultado de aplicar una función de hash a el valor de la llave del cluster. Todos los renglones con la misma llave de hash son almacenados juntos.

Este tipo de cluster se recomienda cuando la mayoría de las consultas tienen una condición de igual para la columna en común de las tablas, por ejemplo Traer todos los renglones del departamento 10.

## Ligas de Base de Datos

Una liga de base de datos o Database Link es un objeto que contiene la ruta de una base de datos a otra, éstas ligas son implícitamente usadas cuando se hace referencia a un objeto global en una base de datos distribuida.

## Bloques de datos, Extents y Segmentos

Oracle permite el más fino nivel de granularidad a través de las estructuras lógicas, incluyendo bloques de datos, extents y segmentos.

- Bloques de datos

El mas fino nivel de granularidad son los datos almacenados en un bloque de datos. El tamaño del bloque de datos se establece a nivel base de datos desde la creación de ésta. Un bloque corresponde al número específico de bytes almacenados físicamente en la base de datos en disco.

- Extents

Es un conjunto de bloques contiguos, el tamaño mínimo de un bloque es de 2KB y puede declararse hasta de 32KB

- Segmentos

Es un conjunto de extents agrupados con un nombre lógico. Entre los diferentes tipos de segmentos que se tienen, se encuentran:

- Segmentos de datos. Las tablas son segmentos de datos.
- Segmentos de índices. Los índices son segmentos de índices.
- Segmentos de "rollback" . Uno o más segmentos de rollback son creados por el administrador de la base datos para almacenar temporalmente información que no necesariamente se guardará en las tablas. Esta información es utilizada para tener lecturas consistentes, poder realizar

recuperaciones, es lo que permite a los usuarios hacer **rollback** (después de hacer una transacción se eliminan esos cambios mediante esta instrucción) a sus transacciones.

- Segmentos temporales. Estos segmentos son creados cuando un usuario necesita realizar una ordenación que no cabe en el área asignada para ordenación en memoria.

Los extents de los segmentos se almacenan dinámicamente conforme el segmento va creciendo.

### **Diccionario de Datos**

El diccionario de datos es un conjunto de tablas, vistas e índices de sólo lectura para todos los usuarios, excepto para el dueño del diccionario que es el usuario: sys.

El diccionario de datos almacena toda la información de la base de datos, como por ejemplo los nombres y contraseñas de todos los usuarios de la base, el nombre de todos los tablespaces, los nombres de todas las tablas, etc.

El diccionario de datos es creado al momento de la creación de la base de datos, se actualiza de manera automática por el RDBMS. es la memoria de la base de datos y es necesario para el funcionamiento de la base.

### **Arquitectura del RDBMS**

El RDBMS utiliza procesos y estructuras de memoria para acceder la base de datos. Todas las estructuras de memoria se manejan en la memoria principal de la computadora. Los procesos son tareas que trabajan en la memoria de la computadora. La figura 1.2.4 ilustra los procesos y estructuras de memoria.

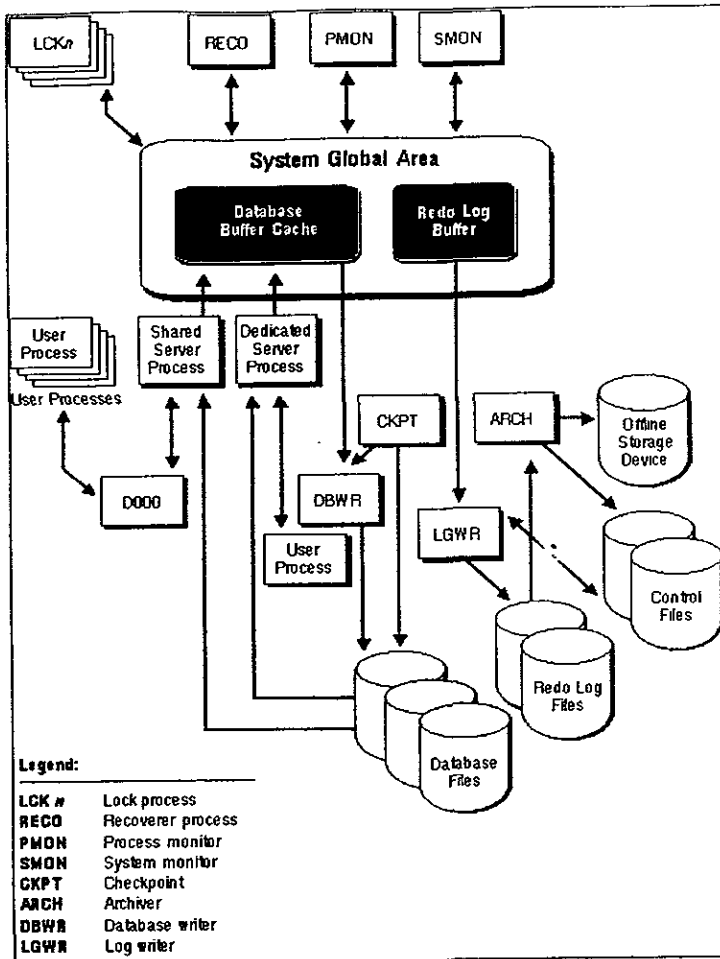


Figura 1.2.5 Procesos y segmentos de memoria de una instancia Oracle.

### Estructuras de Memoria: SGA (Área Global del Sistema)

La SGA es una región de memoria que contiene datos e información de control para una instancia. La SGA y los procesos de background constituyen la instancia.

Oracle reserva la SGA al momento en que la instancia se levanta y la libera la memoria ocupada cuando se da de baja la base de datos. Cada instancia tiene su propia SGA.

Los usuarios concurrentes comparten los datos de la SGA. Para un rendimiento óptimo es recomendable que la SGA sea tan grande como lo permita la memoria real de la máquina en la que se encuentra la base de datos, esto con el objetivo de minimizar el acceso a disco. La información almacenada en la SGA se divide en varios tipos de estructuras de memoria, como por ejemplo los **database buffers** (segmentos para datos almacenados en la SGA), los **redo log buffer** (estructuras de memoria para tener los cambios hechos a la información) y la **shared pool** (estructura de memoria que contiene las instrucciones SQL que ejecutan los usuarios).

- Database Buffer Cache.

El conjunto de database buffers en una instancia recibe el nombre de database buffer cache. Esta estructura almacena los bloques de aquellos datos que más recientemente han sido utilizados. Estos buffers pueden contener datos que no han sido escritos a disco de manera permanente.

- Redo Log Buffer

Este conjunto de buffers almacena **redo entries** (bitácora de cambios realizados a la base de datos). Estos buffers son escritos a los redo log files.

- Shared Pool

Esta estructura contiene las instrucciones SQL que los usuarios pueden compartir. Por ejemplo, cuando una misma instrucción de consulta es utilizada por 1000 usuarios de manera simultánea, solo para el primer usuario se coloca esta instrucción en memoria, con su parse (verificación de sintaxis) y su plan de ejecución; para los demás usuarios, como ya se encuentra en memoria, no es necesario volver a realizar el proceso de subir a memoria esta misma instrucción, de parsearla y de sacar el método de acceso de la información (plan de ejecución), con esto se gana en rendimiento. Una área en el shared pool es necesaria para ejecutar cualquier instrucción SQL.

- Cursores

Son nombres o apuntadores a memoria asociados con una instrucción específica. El manejo de cursores se puede realizar a través de PL/SQL o de PRO\*C. Dentro de la aplicación se controla el manejo del cursor.

- **PGA (Área global de programa)**

Es un buffer de memoria que contiene datos e información de control para el proceso servidor. La PGA es creada cuando un proceso servidor es levantado.

### **Procesos**

Es un mecanismo en un sistema operativo que realiza una serie de pasos. Algunos sistemas operativos utilizan los términos *job* y *task* para nombrar a los procesos de sistema operativo. Un proceso tiene su propia área de memoria en la cual se ejecuta o corre.

Un RDBMS Oracle tiene dos tipos de procesos: procesos de sistema o *background* y procesos de usuarios.

- **Proceso de usuario**

Estos procesos son ejecutados cuando el usuario corre una aplicación como *sql\*plus* o *Forms* o un programa en *PRO\*C*. Estos procesos se comunican con el RDBMS a través del **program interface** (programa de interface).

- **Procesos del servidor (shadow processes)**

Estos procesos son creados para manejar requerimientos de la base generados por procesos de usuario. Este proceso es el que se encarga de comunicarse directamente con el proceso de usuario y de interactuar con Oracle. Se pueden tener dos tipos de configuraciones para manejar estos procesos. En la configuración de servidores dedicados, un servidor de proceso solo maneja los requerimientos de un solo proceso de usuario. En la configuración **multi-threaded server** (servidores compartidos) se tienen muchos procesos de usuario manejados por unos cuantos procesos servidores para maximizar los recursos disponibles del sistema. Es posible tener configuraciones híbridas en las que algunos usuarios se conecten con usuarios dedicados y algunos otros con servidores compartidos.



La figura 1.2.5 muestra como interactúan los procesos con servidores compartidos.

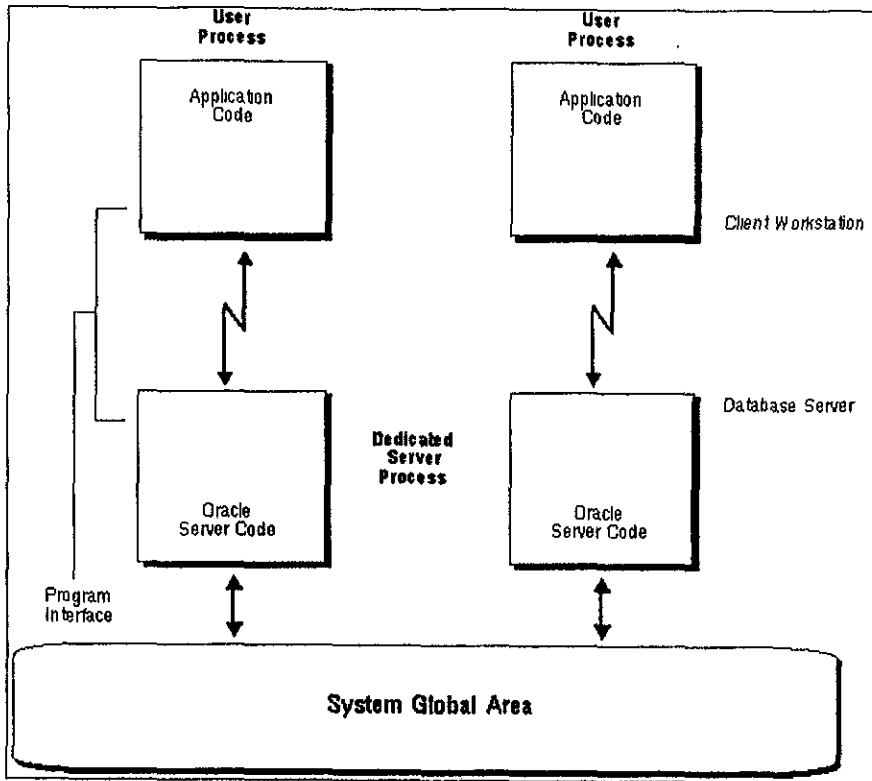


Figura 1.2.5 Servidores compartidos.

El funcionamiento de la configuración multi-threaded server se muestra en la figura 1.2.6, los procesos despachadores y servidores compartidos son procesos de background.

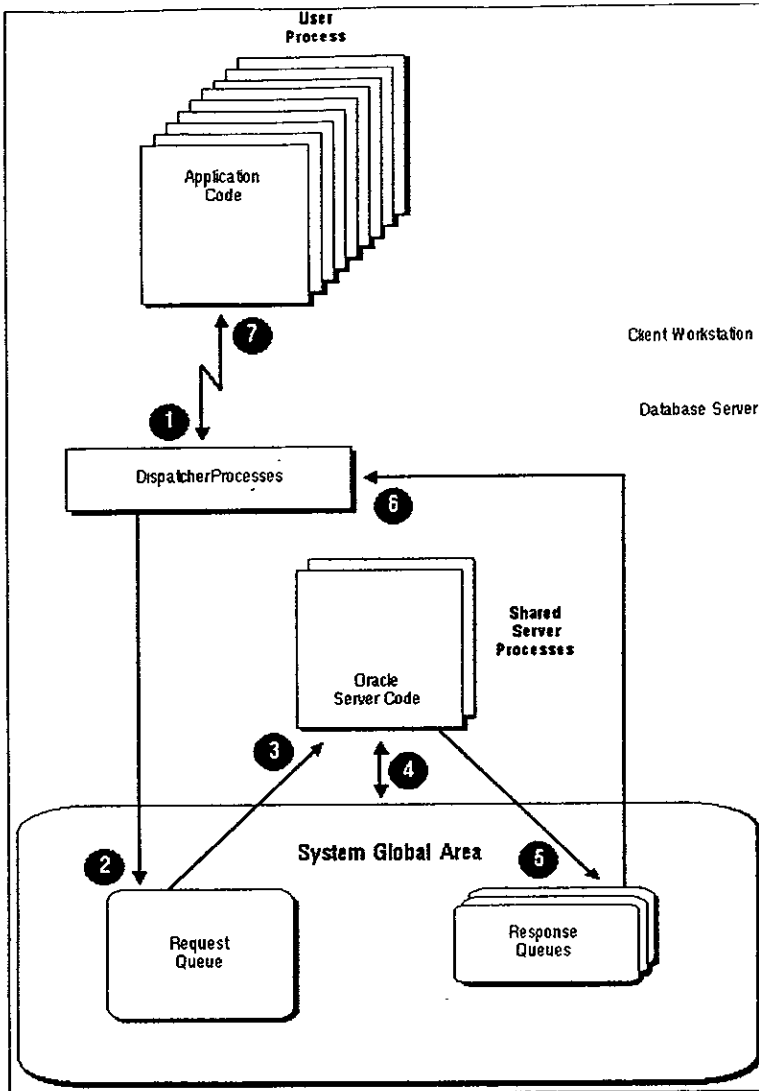


Figura 1.2.6 Configuración Multithreaded-Server.

1.- El proceso de usuario tiene un requerimiento hacia la base de datos que puede ser una consulta, selección, actualización, borrado de renglones o manipulación de algún objeto de la base y lo solicita al proceso despachador.

- 2.- El proceso despachador coloca este requerimiento en la cola de requerimientos para que un proceso servidor lo atienda.
- 3.- Un proceso servidor toma el requerimiento de la cola de requerimientos.
- 4.- El proceso servidor interactúa con la SGA para atender el requerimiento que tomó de la cola y espera tener una respuesta.
- 5.- El proceso servidor coloca la respuesta en la cola de respuestas.
- 6.- El mismo despachador que atendió al proceso usuario toma la respuesta de la cola de respuesta.
- 7.- El servidor le pasa la respuesta al proceso de usuario.

En sistemas cliente-servidor los procesos de usuario y servidor (servidores compartidos y despachadores, para configuración multi-thread) se encuentran en diferentes máquinas.

- Procesos de background

Estos procesos son propios de cada instancia y sirven para interactuar con una base de datos. Estos procesos realizan I/O y monitorean otros procesos Oracle, además realizan funciones para el funcionamiento de la instancia.

Cada instancia puede tener varios procesos, algunos de ellos son opcionales. Estos procesos son:

- **DBWR** (Database Writer, proceso que escribe y lee de los archivos de datos )

Este proceso escribe bloques modificados de los database buffer cache a los datafiles. Este proceso no necesariamente escribe bloques cuando una transacción da commit. Solo escribe cuando es necesario leer mas datos y ya no hay block buffers disponibles. Los bloques menos usados son los que se escriben primero a los archivos de datos.

- **LGWR** (Log Writer, proceso que escribe a los archivos de log )  
Los datos de redo son puestos en los redo log buffers de la SGA. Cuando hay una transacción el log buffer se escribe, el LGWR escribe estos buffers a los archivos de log.
  
- **CKPT** (Checkpoint )  
En tiempos especificados los buffers de la sga son escritos a los archivos de datos por el DBWR, este evento recibe el nombre de checkpoint. El proceso CKPT es responsable de señalar los checkpoints y actualizar los encabezados de los archivos de datos y los archivos de control para indicar el checkpoint. Este proceso es opcional y si no se encuentra, el LGWR realiza esta tarea.
  
- **SMON** (System monitor, monitor del sistema)  
Este proceso realiza la recuperación automática de la instancia al tiempo de levantarse en caso de ser necesario. En el caso de ambientes parallel server, este proceso realiza la recuperación de las instancias que hayan fallado. También limpia segmentos temporales que ya no son usados y compacta los espacios contiguos de los tablespaces.
  
- **PMON** (Process monitor, monitor de procesos)  
Realiza la recuperación del proceso cuando un proceso de usuario falla. El PMON es el responsable de limpiar el buffer caché y liberar los recursos que el proceso que falló estaba usando. También se encarga de checar los despachadores y servidores para levantarlos en caso de ser necesario.
  
- **ARCH** (Archive process, proceso de almacenamiento)  
Este proceso se encarga de copiar los redo log files a disco o a cinta cuando éstos se encuentran llenos. Este proceso se levanta cuando la base de datos se encuentra en **archive mode** (Estado de la base de datos, en el cual los log files deben ser respaldados antes de ser sobrescritos).

- **RECO** (Recoverer process, proceso para recuperación de transacciones distribuidas) El RECO es utilizado para recuperar transacciones distribuidas pendientes debido a alguna falla en la red.
- **Dnn** (Dispatcher, despachador)  
Son procesos opcionales, solo se utilizan cuando se tiene configuración Multi-thread. Al menos un despachador se debe tener por cada protocolo manejado. Cada proceso despachador es responsable de rutear los requerimientos de procesos usuarios a servidores compartidos y regresar la respuesta a los usuarios apropiados.
- **LKnn** (Lock, proceso para manejo de candados)  
Se pueden tener hasta 10 procesos manejadores de candados y son utilizados para manejo inter-instancia cuando es utilizado Parallel Server.

### **Program Interface**

Este programa es un mecanismo mediante el cual los procesos usuarios se comunican con un proceso servidor. Sus funciones son:

- Actuar como mecanismo de comunicación, formateando datos de los requerimientos, pasando datos y atrapando y regresando errores.
- Realizar conversiones de datos, particularmente entre diferentes tipos de computadoras.

## 1.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS PRINCIPALES ARQUITECTURAS DE REDES COMERCIALES.

### Características Físicas de una Red de Área Local.

Una Red de Area Local (LAN) es un sistema formado por dispositivos de procesamiento de la información interconectados por un medio común de comunicaciones. El control de los mismos puede estar centralizado, distribuido o ser una combinación de ambos. Una LAN, por definición, tiene impuesta una restricción de alcance, limitando el área de cobertura al entorno definido por un usuario o tipo de usuario.

La forma en que las computadoras sobre una red son conectadas se le conoce como *topología*. Las tres topologías más ampliamente utilizadas son **bus** (línea), estrella y anillo.

### Topología de Bus.

Una topología de bus o lineal se constituye de un cable para interconectar todas las estaciones de trabajo. En cada una de las estaciones de los extremos se coloca un terminador. En esta topología la información viaja en ambos sentidos, es decir hacia la derecha y a la izquierda de la estación de trabajo, por lo que es necesario controlar las posibles colisiones de información que se produzcan en los medios de comunicación.

El bus lleva un mensaje desde un extremo de la red hasta el otro, conforme el mensaje pasa por cada estación de trabajo, ésta verifica la dirección de destino del mensaje. Si la dirección del mensaje se relaciona con la de la estación de trabajo, ésta recibe el mensaje, si no, el bus lleva el mensaje a la próxima estación de trabajo y así sucesivamente.

Un tipo de bus llamado bus local se muestra en la figura 1.3.1. Un bus local utiliza un conector "T" para conectar el cable al adaptador de red de la estación de trabajo. Un terminador es conectado a cada conector "T" en cada extremo de la red.

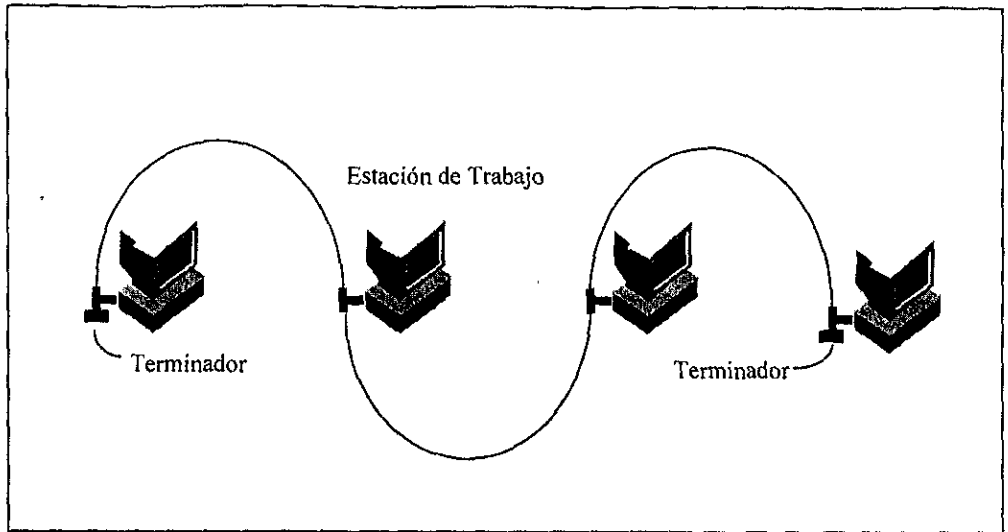


Figura 1.3.1 Red tipo bus conecta la red directamente a cada computadora por medio de conectores tipo "T".

Otro tipo de bus como se muestra en la figura 1.3.2 utiliza cables para conectar la estación de trabajo al Bus Principal.

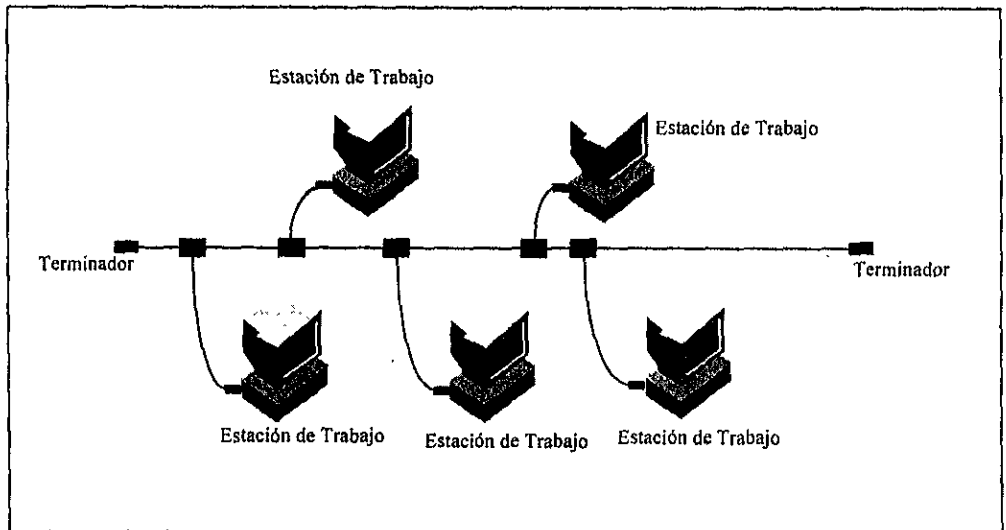


Figura 1.3.2 Uso de cables para conectar computadoras al cable central.

La topología de bus es una red pasiva, lo anterior significa que las estaciones de trabajo solo escuchan datos que son enviados a la red y no son responsables de llevar o mover datos de una estación a la siguiente.

Si una estación de trabajo falla, ésta no afecta a la red completa. Por otro lado si la conexión de una estación de trabajo se pierde ó el cable se daña o rompe, el segmento completo queda sin funcionar.

La figura 1.3.3 muestra algunas ventajas y desventajas de una red tipo bus.

VENTAJAS	DESVENTAJAS
La falla de una estación de trabajo no afecta a la red	El daño ó ruptura de un cable puede afectar a muchos usuarios
Conexión de cable fácil y flexible	Número de estaciones de trabajo y longitud de cable limitados
Conectores y cables relativamente económicos	Dificultad para encontrar los errores de cableado en la red

Figura 1.3.3 Ventajas y desventajas de una red tipo bus

### Topología de Estrella.

La topología de estrella se constituye de un cable por estación de trabajo, que se conecta a un dispositivo de conexión central, este dispositivo puede ser un concentrador, un repetidor ó un servidor, este dispositivo central puede ser pasivo, activo o inteligente. Véase figura 1.3.4

El concentrador pasivo simplemente conecta los elementos que forman la estrella, no regenera la señal. Un concentrador activo regenera las señales y los concentradores inteligentes realizan la operación del activo y realizan la selección de la mejor ruta, además puede realizar operaciones de administración de la red.



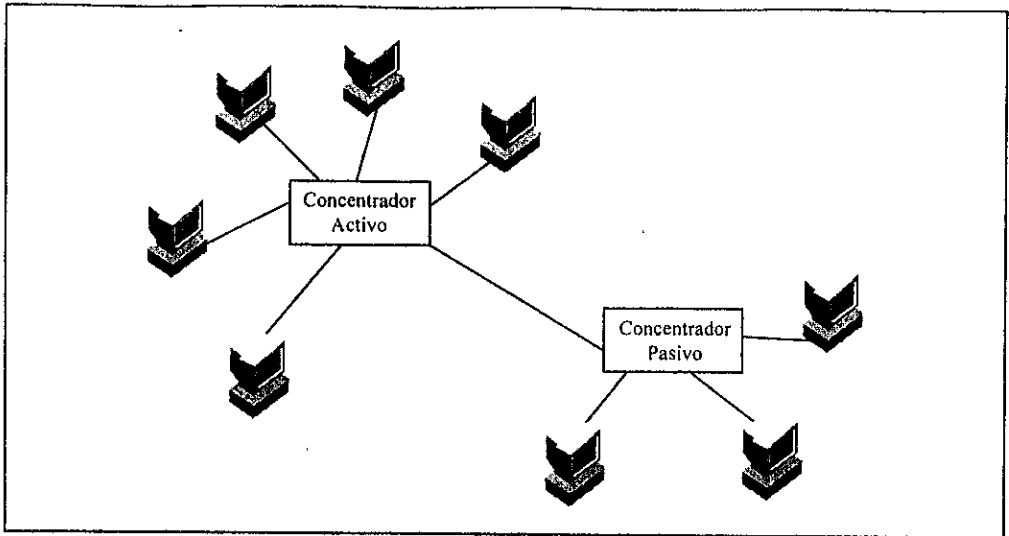


Figura 1.3.4 Cada computadora sobre una red estrella está conectada a un concentrador. Los concentradores pueden ser interconectados para extender la red.

Debido a la necesidad de un cable por cada estación de trabajo hacia el dispositivo de conexión central, se requiere de una cantidad de cable muy superior a los otros tipos de topologías (bus y anillo) pero aunque se tiene esta desventaja, se presenta una ventaja muy importante, la facilidad de aislar las fallas que se lleguen a presentar durante el funcionamiento de la red. El hecho de poder aislar las fallas de un nodo es una de las mejores cualidades de este tipo de topología ya que no genera problemas con las demás estaciones de trabajo, y de esta manera se ve totalmente aislado el problema de las estaciones que se encuentran trabajando en forma correcta.

Además de la gran ventaja mencionada anteriormente, se tiene la facilidad de localizar un nodo en específico, y el agregar o eliminar un nodo se realiza de manera muy sencilla.

El dispositivo central que se encarga de atender a cada una de las estaciones lo hace en forma continua y por orden; es decir, para atender a cada una de las estaciones de trabajo se atiende en forma progresiva para ver si tiene alguna necesidad de comunicación con otra estación o con el servidor, al realizar esta atención a cada estación de trabajo lo realiza en forma exclusiva

(servidor estación), es decir, que en ese momento no le pone atención a ninguna otra de las estaciones de trabajo, el recorrido de la atención a los diferentes nodos lo realiza en forma permanente.

La figura 1.3.5 muestra algunas ventajas y desventajas de una red tipo estrella.

VENTAJAS	DESVENTAJAS
Es fácil añadir nuevas estaciones de trabajo	La falla del concentrador inhabilita todas las estaciones conectadas a éste
Monitoreo y administración de la red centralizados	

Figura 1.3.5 Ventajas y desventajas de una red tipo estrella.

### Topología de Anillo.

En una red de este tipo las estaciones de trabajo son situadas en forma de anillo sobre el cual una señal es enviada desde una estación de trabajo a la siguiente. Aún cuando el nombre de la topología sea anillo, físicamente las estaciones de trabajo están conectadas en estrella y eléctricamente funcionan como anillo. Las estaciones de trabajo están centralmente conectadas a un concentrador y son cableadas en configuración de estrella. Véase figura 1.3.6.

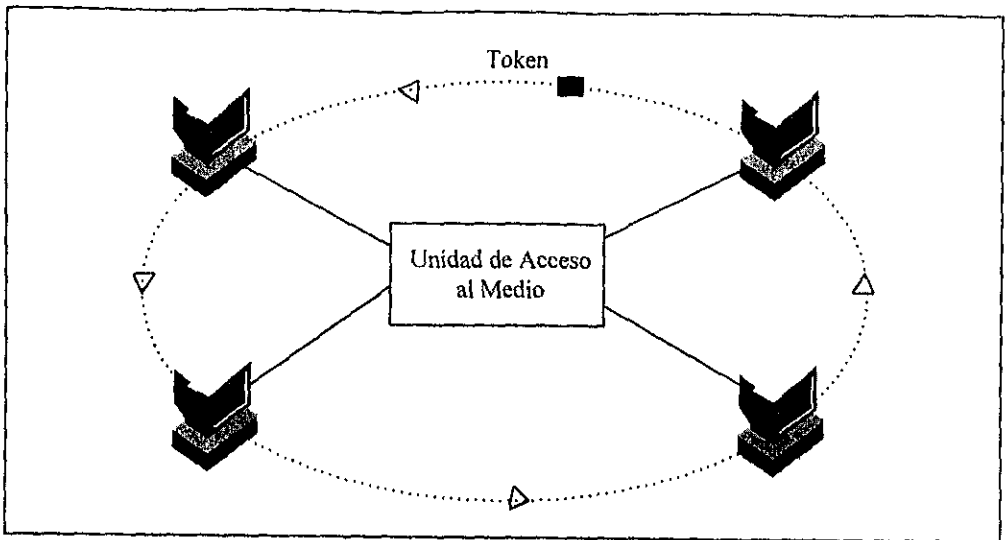


Figura 1.3.6 Las computadoras en un red de anillo se comunican por medio de un token.

Las estaciones de trabajo utilizan una señal llamada **token** para transmitir datos y cada una de ellas debe esperar por un token disponible para transferir mensajes.

El token contiene la dirección de quien envía y quien recibirá. Cuando el receptor ha copiado la información del mensaje recibido, éste regresa el token a la estación de trabajo que lo originó para verificar que el mensaje fue recibido. La estación de trabajo originaria del token envía el token a la próxima estación en el anillo.

La figura 1.3.7 muestra algunas ventajas y desventajas de una red tipo anillo.

VENTAJAS	DESVENTAJAS
La falla en un cable afecta a un número reducido de usuarios	Conectores y cable de alto costo
Poca degradación del rendimiento conforme la red crece	

Figura 1.3.7 Ventajas y desventajas de una red tipo anillo.

### Topología de Malla.

Una red de topología en malla es considerada como una conexión de punto a punto entre todos los nodos que se encuentran en la red. Una red con topología puramente en malla, no es considerada una buena solución práctica. Un problema es que cada dispositivo requiere una tarjeta de interface para cada dispositivo en la red. Otro problema es que el monto total de cable es demasiado alto. Véase figura 1.3.8.

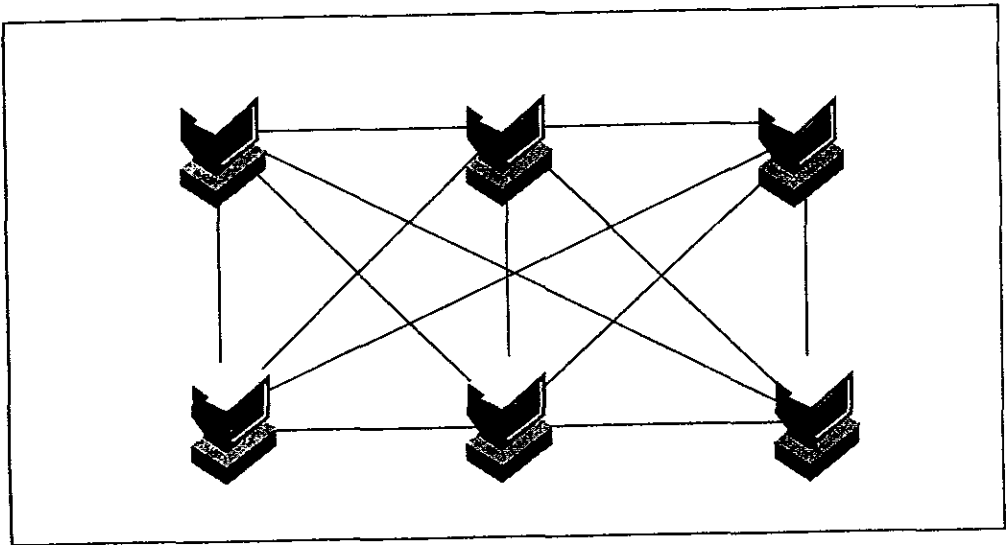


Figura 1.3.8 Topología de malla.

### Topología Híbrida.

La topología híbrida es aquella que se encuentra formada por múltiples topologías; como por ejemplo las redes WAN se encuentran formadas de esta manera. Véase figura 1.3.9.

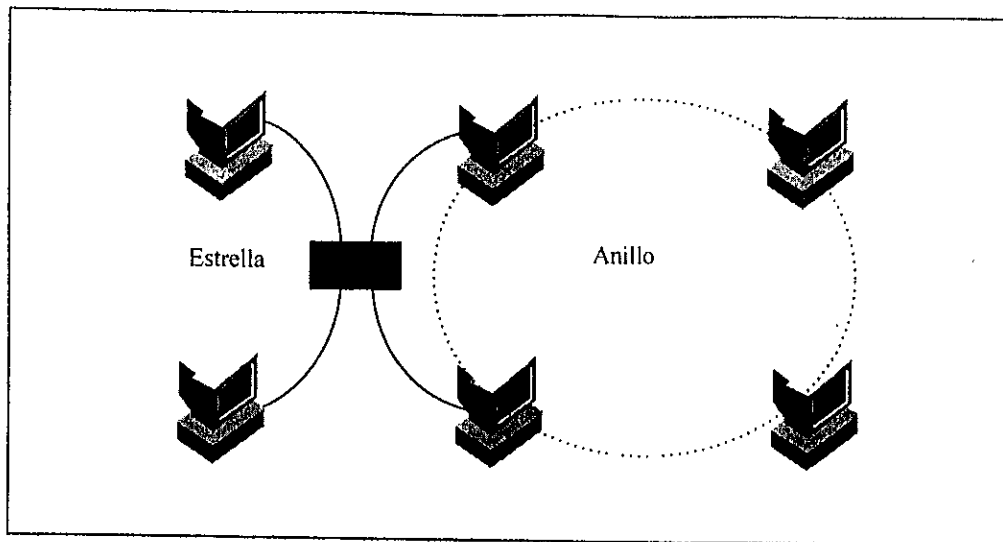


Figura 1.3.9 Topología híbrida.

Por tal motivo las redes con topología híbrida han tenido un gran crecimiento en los 90's. La razón es el incremento en la interconexión de redes, formando redes WAN.

### Cableado.

La industria de las redes de área local ha estandarizado tres medios físicos primarios que pueden ser usados en la capa física: cable coaxial, cable twisted pair (par trenzado), y cable de fibra óptica. Las velocidades de transmisión que pueden ser soportadas sobre cada uno de estos medios físicos son medidos en millones de bits por segundo o Mbps.

### Cable Coaxial.

Este cable está compuesto de un cable central conductor rodeado por una capa aislante, una capa de malla alámbrica (blindaje), y una capa exterior no conductiva. Este cable es resistente a interferencias. Ver figura 1.3.10

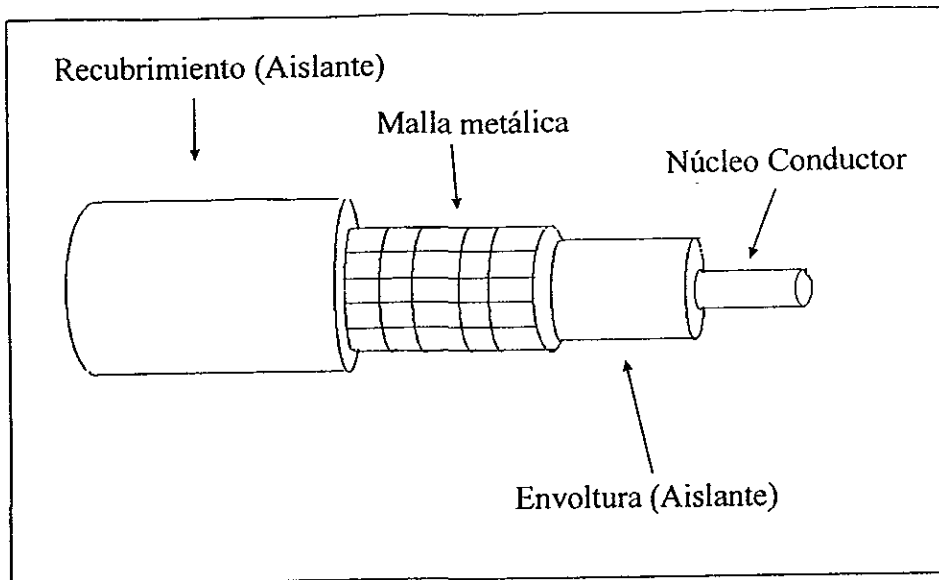


Figura 1.3.10. Cable coaxial.

El cable coaxial para redes tiene varias presentaciones, las más populares son Thinnet y Thicknet.

Thinnet se refiere al cable RG-58 y es un cable coaxial flexible de  $\frac{1}{4}$  de pulgada de grosor. Es usado para distancias de comunicación cortas relativamente y es muy flexible lo cual facilita el cableado entre las estaciones de trabajo. Se conecta directamente a la tarjeta de red por medio de un conector BNC tipo "T".

Thicknet o Ethernet estándar es un cable coaxial relativamente rígido de  $\frac{1}{2}$  pulgada de diámetro. Normalmente thicknet es usado como backbone para conectar muchas pequeñas redes Thinnet debido a la característica de soportar transferencia de datos a través de grandes distancias. Muchas de las veces un transceiver es conectado directamente al cable Thicknet usando un conector conocido como "piercing tap". La conexión del transceiver a la tarjeta de red es realizada por medio de un drop cable (segmento pequeño de cable). La figura 1.3.11 muestra las ventajas y desventajas del cableado coaxial

VENTAJAS	DESVENTAJAS
Bajo costo de mantenimiento	Distancia y topología limitadas
Fácil instalación	Baja seguridad
Mejor resistencia al ruido que el par trenzado sobre grandes distancias	Dificultad para hacer cambios a la topología del cableado
Transmisión de voz, video y datos	El costo es mayor al del par trenzado, sin embargo los componentes de soporte electrónico son mas baratos
Compatibilidad con equipo de diferentes vendedores	Algunos cables son pesados, costosos y poco manejables
Ancho de banda amplio	
Alta resistencia al movimiento, evitando roturas en el cable	

Figura 1.3.11 Ventajas y desventajas del cableado coaxial.

### Par Trenzado.

Consiste de dos tiras de alambre de cobre trenzados y aislados. Un número de pares son agrupados y cubiertos con una funda para formar el cable. El par trenzado sin blindaje es usado comunmente para sistemas de telefonía. Ver figura 1.3.12

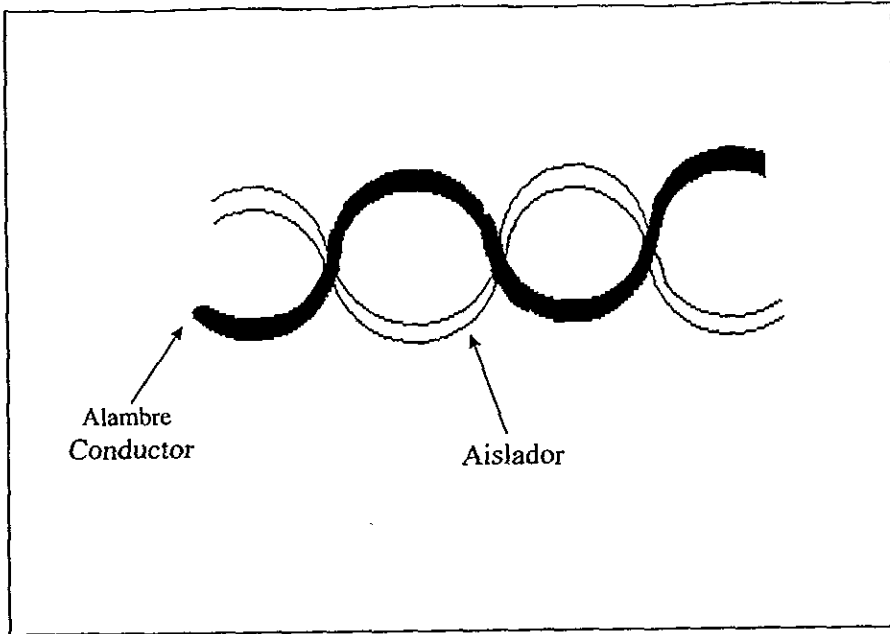


Figura 1.3.12 Alambre y cable de par trenzado.

La figura 1.3.13 lista algunas ventajas y desventajas del cable de par trenzado.

VENTAJAS	DESVENTAJAS
Tecnología conocida	Susceptible al ruido
Facilidad de añadir computadoras a la red	Ancho de banda limitado
Bajo costo	Distancia limitada
Se puede hacer uso de los cables de par trenzado instalados en los edificios para el sistema de teléfonos	Requiere dispositivos de alto costo
Compatibilidad con varias arquitecturas de redes	
Excelente relación en costo/rendimiento	

Figura 1.3.13 Ventajas y desventajas del cable de par trenzado.



## Fibra Óptica.

La fibra óptica lleva la señal de datos digitales en forma de pulsos de luz modulados. Una fibra óptica consiste de un cilindro de vidrio muy delgado conocido como "core" (núcleo), rodeado por una capa concéntrica de vidrio. Hay dos fibras por cable, una para transmitir y la otra para recibir. (Ver figura 1.3.14.)

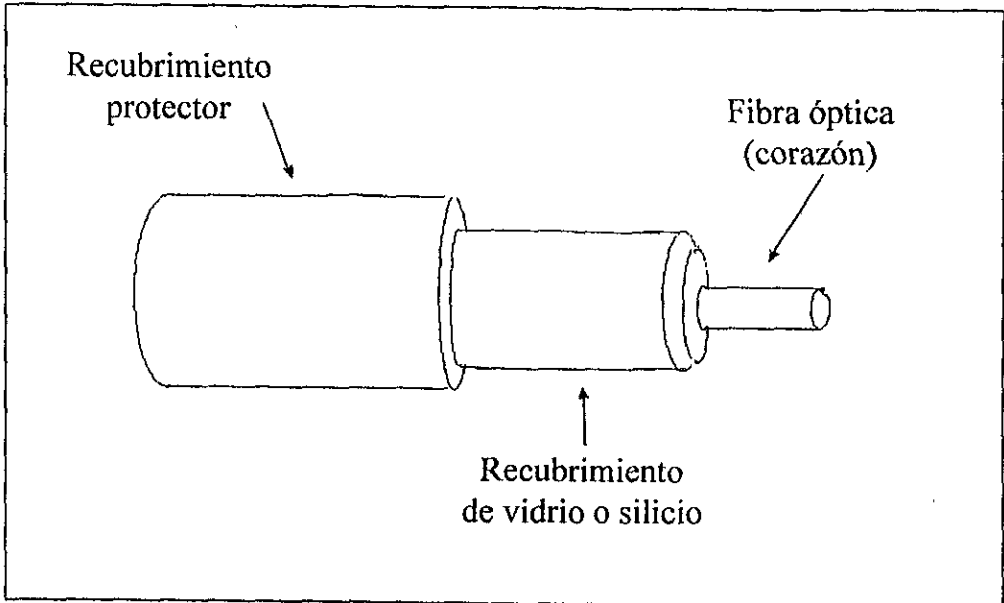


Figura 1.3.14 Cable de Fibra Óptica.

Las señales transmitidas por fibra óptica no son propensas a interferencia y son muy veloces (actualmente la transmisión es de aproximadamente 100 Mbps con velocidades comprobadas de hasta 200,000 Mbps).

La figura 1.3.15 muestra algunas ventajas y desventajas de la fibra óptica.

VENTAJAS	DESVENTAJAS
Aplicaciones de alta velocidad	Alto costo de instalación y mantenimiento
La más baja pérdida de transmisiones en grandes distancias	Cable inflexible. No puede ser doblado fácilmente
Inmune a interferencias electromagnéticas	Carencia de componentes estándar
Bueno para "Backbone" en las redes	Requiere de conocimientos especiales para su instalación y mantenimiento
Soporte para voz, datos y video por el mismo canal	Las interfaces de red y cables son relativamente costosos
Puede propagar una señal sin necesidad de utilizar un amplificador en grandes distancias	Las conexiones requieren una manufactura de alta precisión y manejo muy delicado
Tiene un gran ancho de banda	
Compatibilidad con Ethernet, Token Ring y FDDI	

Figura 1.3.15 Ventajas y desventajas de la fibra óptica.

### Transmisión en Banda Base y Banda Ancha.

Dos técnicas pueden ser usadas para transmitir las señales codificadas sobre el cable. Banda Base y Banda Ancha. La transmisión en Banda Base utiliza señalización digital y la transmisión en Banda Ancha utiliza señal analógica.

#### Transmisión en Banda Base.

Los sistemas de Banda Base usan señales digitales en una sola frecuencia. Las señales fluyen en forma de pulsos discretos de electricidad o luz. Con transmisión de Banda Base toda la capacidad del canal de comunicación es usado para transmitir una sola señal de datos.

Conforme la señal viaja a través del cable de red, ésta gradualmente se decrementa en potencia y puede distorsionarse. Si el cable es demasiado largo, resultará distorsionada la señal, la señal

recibida quizá no pueda ser reconocida. Un sistema de Banda Base seguro utiliza repetidores para recibir una señal y retransmitirla con mayor potencia y definición para incrementar la longitud práctica de un segmento de cable.

### **Transmisión en Banda Ancha.**

Los sistemas en Banda Ancha usan señales analógicas y un rango de frecuencias. Con transmisión analógica las señales empleadas son continuas y no discretas. Estas fluyen a través del medio físico en forma de ondas ópticas ó electromagnéticas.

Para Banda Base, el ancho de banda total del cable es la diferencia entre las frecuencias más alta y más baja que son llevadas por el cable. Con Banda ancha, el ancho de banda total está disponible, por lo que múltiples sistemas de transmisión analógica pueden ser soportados simultáneamente sobre el mismo cable (tales como cable de televisión y transmisiones de red). A cada sistema de transmisión se le asigna una parte del ancho de banda total. Todos los dispositivos asociados con un sistema de transmisión dado tales como computadoras en un cable de red de área local, deben ser sintonizadas de tal forma que usen solo las frecuencias que están especificadas para su rango.

Dispositivos analógicos, tales como teléfonos y equipo de video, pueden transmitir su señal analógica directamente sobre el cable. Sin embargo los sistemas digitales tales como computadoras, requieren modems para traducir la señal digital en una señal analógica antes de la transmisión.

Mientras que los sistemas de Banda Base usan repetidores, los sistemas de Banda Ancha usan amplificadores para regenerar la señal analógica a su potencia original.

Con transmisión en Banda Ancha, el flujo de la señal es en una sola dirección, por lo que para conseguir que la señal alcance todos los dispositivos se necesita que existan dos rutas para el flujo de ésta, hay dos formas de conseguirlo :

1. Configuración Mid-split, la cual divide el ancho de banda en dos canales, cada uno usando una frecuencia diferente o rango de frecuencias. Un canal es usado para transmitir señales y el otro para recibir.
2. Configuración de Doble-cable, cada dispositivo es conectado a dos cables. Un cable es usado para enviar y otro para recibir.

### **Tarjetas de Interface de Red.**

Las tarjetas de interface de red proporcionan la conexión física con la red. En la mayoría de los casos se adapta a la ranura de expansión de la computadora.

La tarjeta de interface obtiene información de la computadora, la convierte al formato adecuado y la envía a través del medio de comunicación (como por ejemplo un cable) a la otra tarjeta de interface de la red local. Esta tarjeta recibe la información, la traduce para que la computadora la pueda entender y la envía a la computadora.

Las funciones de la tarjeta de interface son : la comunicación de host (computadora) a tarjeta, buffering, formación de paquetes, conversión serial paralelo, codificación y decodificación, acceso al cable, transmisión y recepción. Estos pasos hacen que los datos de la memoria de una computadora pasen a la memoria de otra.

Los parámetros que debemos considerar son los siguientes: el ancho del bus (8, 16 ó 32 bits), tipo de bus (ISA, EISA ó Microcanal), el tipo de transferencia de memoria (la memoria compartida es más rápida que E/S y el acceso directo a la memoria), y si la tarjeta puede tener control sobre el bus.

## Opciones de Configuración.

Las tarjetas de red generalmente tienen tres parámetros de configuración, los cuales se deben seleccionar correctamente para tener un funcionamiento correcto con la computadora. Los parámetros referidos son: línea de interrupción (IRQ), dirección de entrada salida (I/O address) y la dirección de memoria base. Todos estos parámetros como se indicó, se deben configurar en un valor de forma que no entre en conflicto con algún otro dispositivo de la computadora.

### Línea de Interrupción. IRQ

Cuando un dispositivo envía una solicitud de atención al microprocesador se hace uso de una línea de interrupción al procesador. Esta línea es una señal eléctrica que se envía al procesador, cada dispositivo debe utilizar una línea diferente para no entrar en conflicto al solicitar la atención del procesador. La línea de interrupción que cada dispositivo utilizará se define al momento de configurarlo.

En la mayoría de los casos se puede utilizar la interrupción IRQ3 ó IRQ5, siendo de éstas la más recomendable la IRQ5. Si en un momento dado no se tiene disponible ninguna de estas dos líneas se puede usar otra que se encuentre sin uso. A continuación se muestra una tabla de las líneas de interrupción de una computadora AT. (figura 1.3.16)

IRQ	DISPONIBILIDAD
2 y 9	EGA/VGA
3	Normalmente disponible ó utilizada por COM1 ó COM4
4	COM1, COM3
5	Normalmente disponible, ó utilizada por LPT2
6	Controlador del disco flexible
7	LPT1
8	Reloj de tiempo real
10	Disponible
11	Disponible
12	Ratón (PS/2)
13	Coprocesador matemático
14	Controlador del disco duro
15	Disponible

Figura 1.3.16 Tabla de las líneas de interrupción de una computadora AT.

### Dirección de Entrada/Salida (I/O Address)

La dirección de entrada/salida especifica el canal a través del cual la información será transferida entre el hardware de la computadora y el microprocesador. Cada dispositivo que necesita comunicarse con el microprocesador requiere de una dirección de entrada salida, y debe ser *diferente una de otra*. La tabla (figura 1.3.17.) muestra las direcciones de entrada/salida en una computadora personal.

PUERTO	DISPOSITIVO	PUERTO	DISPOSITIVO
200-20F	Puerto de juego	300-30F	
210-21F		310-31F	
220-22F		320-32F	Controlador del disco duro
230-23F	Ratón de bus	330-33F	
240-24F		340-34F	
250-25F		350-35F	
260-26F		360-36F	
270-27F	LPT3	370-37F	LPT2
280-28F		380-38F	
290-29F		390-39F	
2A0-2AF		3A0-3AF	
2B0-2BF		3B0-3BF	LPT1
2C0-2CF		3C0-3CF	EGA/VGA
2D0-2DF		3D0-3DF	CGA/MCGA
2E0-2EF		3E0-3EF	
2F0-2FF	COM2	3F0-3FF	Controlador del disco flexible, COM1

Figura 1.3.17 Direcciones de entrada/salida en una computadora personal.

### Dirección de Memoria Base.

La dirección de memoria base define la dirección de la localización de memoria RAM dentro de la PC que será utilizada para intercambiar información entre la tarjeta de red y el microprocesador. La dirección más común para esta configuración es D8000. En algunos casos es necesario indicar el monto o total de memoria que será configurada para esta función.

**Arquitecturas.**

Se le llama arquitectura a un conjunto de estándares que se utilizan para desarrollar equipo y ser utilizado en la comunicación de redes. A continuación se analizarán las arquitecturas principales en el área de redes locales.

**Instituto de Ingenieros Eléctricos y Electrónicos (IEEE).**

En 1980, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) se dieron a la tarea de definir los estándares para las redes de área local (LAN).

Por las necesidades de las diferentes compañías involucradas en el desarrollo de las comunicaciones en redes de cómputo, la IEEE determinó que se requerían más de una norma para poder satisfacer a las partes involucradas, por tal motivo se determina desarrollar varias normas o estándares.

La IEEE crea un comité para el desarrollo de estas normas, éste comité es el proyecto 802 de la sociedad de cómputo de la IEEE, de tal forma que desarrollan los estándares de redes de área local, estos estándares abarcan las capas física y de liga de datos del modelo OSI.



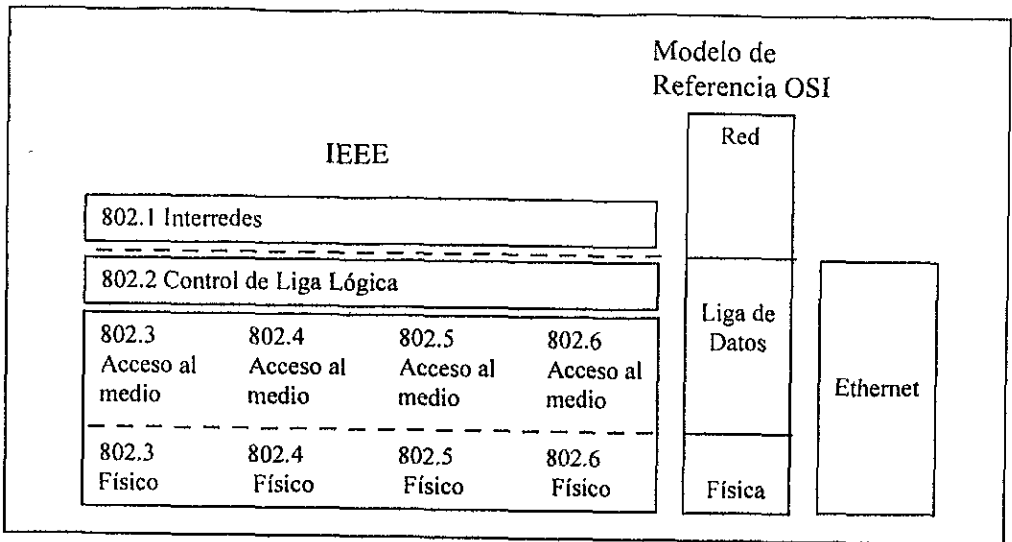


Figura 1.3.18. Modelo IEEE para las comunicaciones.

Así, en 1985 el comité del proyecto 802 de la sociedad de cómputo de la IEEE publica cuatro normas diferentes.

Al enfocarse a las dos capas bajas del modelo OSI, decidieron dividir la capa de liga de datos en dos subcapas: la subcapa de Control de Acceso al Medio (MAC) y la subcapa de Control de Liga Lógica (LLC). De estas dos subcapas la más baja es la MAC, la cual proporciona el acceso a la tarjeta de la red con el medio de comunicación. La subcapa MAC comunica directamente con la tarjeta de red y es responsable de librar de errores los datos entre dos o más computadoras de la red.

La subcapa LLC maneja la liga de datos en la comunicación y define un punto de interface lógico para que otras computadoras puedan referenciar y usar la información de la subcapa LLC hacia las capas superiores del modelo OSI.

La norma 802.2 de la IEEE proporcionará el control de Liga de datos (capa 2), la 802.3 define las redes bajo CSMA/CD, la 802.4 define el acceso por paso de token en bus y la 802.5 define el

acceso por paso de token en anillo. Estos estándares o normas son adoptados por el Instituto Nacional Americano de Estándares (ANSI) en 1985, son revisados y emitidos nuevamente por la ISO y referenciados por la ISO 8802.

El estándar o norma 802.1 de la IEEE proporciona una introducción a los estándares 802 y la relación existente entre los otros estándares de la 802.

La norma 802.6 de la IEEE fue creada para buscar una mejor solución al problema de la estandarización de la tecnología existente y utilizada para la expansión de una red a otra WAN, así que este comité evalúa la utilidad de muchas soluciones existentes.

La 802.7 y 802.8 proporciona un guión para otros grupos de trabajo y subcomités en el desarrollo de estándares para la tecnología de banda ancha y fibra óptica.

La 802.9 cubre la integración de voz y datos en LAN y la 802.10 se encarga de la seguridad en las LAN.

### **Ethernet y la IEEE 802.3.**

La arquitectura o tecnología Ethernet fue inventada por la compañía Xerox a mediados de los 70's. Fue diseñada como una simple y económica arquitectura para redes de área local. La versión 1.0 de Ethernet fue liberada en unión con Digital Equipment Corporation, Intel y Xerox en 1980. Ethernet versión 2.0 apareció dos años después, y tres años después la IEEE libera las primeras especificaciones en la 802.3

La norma IEEE 802.3 está basada en Ethernet, pero proporciona múltiples opciones en la capa física. En un conjunto que cumple con la IEEE los enlaces están basados generalmente en la norma IEEE 802.2. Actualmente el término Ethernet es utilizado para hacer referencia a las redes de área local que se basan en CSMA/CD (Carrier-Sense Multiple Access with Collision Detection, éste es un método de acceso múltiple con detección de colisiones).

IEEE 802.3 y Ethernet utilizan el método de acceso de contención, y por lo tanto el protocolo CSMA/CD, este protocolo de acceso al medio funciona como sigue: Al transmitir, una estación de trabajo debe primero verificar el medio de comunicación, si el medio se encuentra libre, la estación transmite el mensaje y verifica el medio de comunicación para ver si existe o no una colisión. Si el medio de comunicación está en uso, la estación espera un tiempo y en seguida verifica de nuevo el medio. Cuando la transmisión actual termina, la estación envía su mensaje y verifica el medio por si se presenta una colisión. Si una colisión se presenta, es detectada, la estación detiene la transmisión de datos, y envía un mensaje para alertar a todas las estaciones de que sucedió una colisión. Este mensaje de aviso consiste de 32 a 48 bits. Las estaciones esperan un tiempo aleatorio para verificar nuevamente el medio de transmisión, e intentar nuevamente la comunicación.

Como se mencionó anteriormente la norma IEEE 802.3 ofrece una variedad de opciones a nivel físico, incluyendo diferentes modos de enviar las señales, tipos de medios de comunicación, topologías, y velocidades de transmisión.

El nombre del estándar se encuentra formado por tres partes, y éstas son:

1. Velocidad en Mbps.
2. Banda base ó banda ancha.
3. Longitud del segmento de forma aproximada.

Las redes Ethernet pueden ser cableadas con tres diferentes tipos de cable:

1. La especificación 10base2, conocida como **thinnet** (cable coaxial delgado), se refiere a una red de área local ethernet en banda base que utiliza cable coaxial delgado. El cable coaxial delgado puede llevar una señal a aproximadamente 185 metros, debido a ésto la señal debe ser regenerada por un repetidor.

2. La especificación 10base5, conocida como **thicknet** (cable coaxial grueso), es una red de área local en banda base ethernet que utiliza cable coaxial grueso. Este tipo de cable puede llevar una señal hasta 500 metros antes de requerirse un repetidor.
3. La especificación 10baseT, conocida como **twisted pair** (par trenzado), es una red de área local en banda base ethernet que utiliza el cable de par trenzado, éste tipo de cable puede llevar un mensaje por 100 metros entre una estación de trabajo y el concentrador al que está conectada la estación de trabajo.

**Nota:** Los componentes de **hardware**, existen para facilitar la conexión de diferentes topologías ethernet, esto permite que una topología determinada sea seleccionada dependiendo de las necesidades del grupo de trabajo.

A continuación describiremos cada una de las tres topologías ethernet:

### **Thinnet (10base2)**

Es llamada así porque utiliza cable coaxial delgado (RG-58). Este cable puede llevar una señal aproximadamente 185 metros, por lo que la señal debe ser regenerada por un repetidor. Además de que existe un límite máximo, existe un límite mínimo para la longitud del cable. El cable entre estaciones debe ser de por lo menos 0.5 metros de largo.

La razón por la que esta red es llamada 10Base2 por la IEEE es debido a que transmite a 10 Mbps sobre un cable banda base y lleva una señal aproximadamente 2 X 100 metros. (la distancia actual no son 200 metros, sino 185 metros).

Las redes thinnet generalmente utilizan una topología de bus local, como se describió anteriormente. Los estándares de la IEEE para thinnet no permiten el uso de cables para conectar las estaciones a un cable central, en vez de esto se conectan los conectores tipo "T" al adaptador de red.

La red thinnet fue diseñada por ser una forma económica de implementar redes para grupos de trabajo pequeños, el cable utilizado es económico, fácil de configurar e instalar.

Una red thinnet puede soportar un máximo de 30 nodos (estaciones de trabajo y repetidores) por segmento. Un segmento es la longitud de cable entre dos terminadores. Utilizando repetidores se pueden tener hasta 5 segmentos para dar soporte a 150 computadoras.

### **Thicknet (10Base5).**

La IEEE lo llama así debido a que sus principales especificaciones son: 10Mbps, banda base y 500 metros (5 X 100). También conocido como ethernet estándar.

Generalmente usa una topología de bus y puede soportar hasta 100 nodos (estaciones de trabajo, repetidores y puentes) por segmento. Este tipo de red utiliza cables desde las estaciones para conectarlas al cable principal. Un máximo de 5 segmentos pueden ser conectados usando un repetidor (basado en la especificación 802.3 de la IEEE).

La mínima longitud de cable entre conexiones sobre el segmento principal es de 2.5 metros.

Thicknet fue diseñado para soportar redes que ocupan edificios completos.

### **Twisted Pair (10BaseT)**

10BaseT es una red de área local que utiliza cable de par trenzado para conectar las estaciones de trabajo. La mayoría de las redes de este tipo son configuradas con topología de estrella pero internamente utilizan un sistema de señalamiento de bus como las otras configuraciones ethernet.

Normalmente, el concentrador de una red 10BaseT sirve como repetidor. La longitud del cable para la conexión de cada estación de trabajo con el concentrador puede estar entre 2.5 y 100 metros.

En la figura 1.3.19 se muestra la relación entre ethernet y la IEEE 802.3

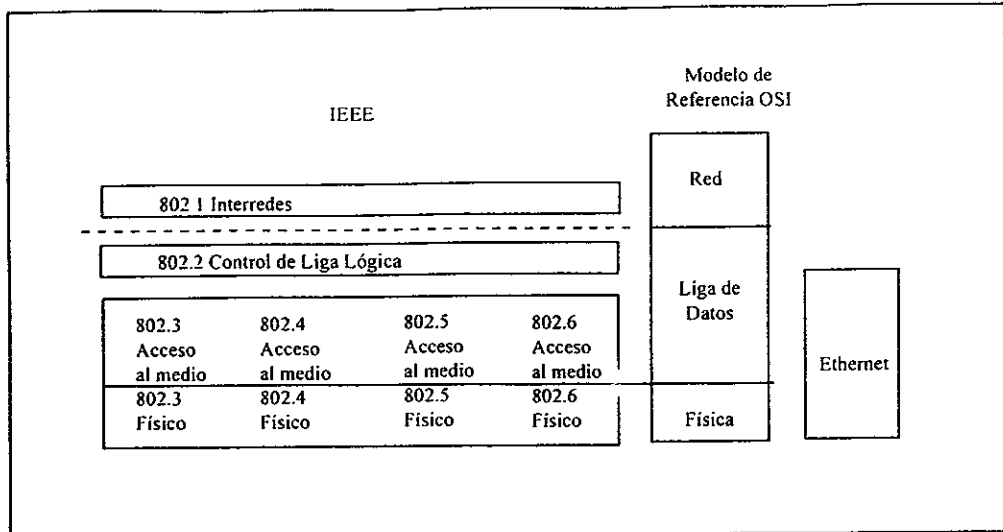


Figura 1.3.19. Relación entre ethernet y la IEEE 802.3

### Token Ring y la IEEE 802.5

Esta norma utiliza el control de acceso al medio de **token passing** (paso de señal) en anillo y fue inicialmente diseñado por IBM. La norma IEEE 802.5 es compatible con las redes token ring de IBM. Aunque IBM continua desarrollando, expandiendo, y promoviendo sus redes de área local, la diferencia que existe entre la IEEE 802.5 y token ring de IBM son muy pequeñas.

Este estándar está formado por dos partes: el estándar a nivel físico y las especificaciones de la subcapa **MAC** (Media Access Control) de la capa de enlace.

La norma IEEE 802.5 no especifica muchas limitaciones en la capa física, ésta describe un simple anillo de hasta 250 estaciones conectadas en serie por enlaces de punto a punto de STP. Las señales pueden tener una velocidad de 1 a 4 Mbps usando una codificación Manchester Diferencial. Las redes Token Ring de IBM ofrecen una velocidad de 4 y 16 Mbps. La IEEE 802.5 no especifica un cableado en estrella, usa un cableado central llamado Unidad de Acceso a

Múltiples estaciones (MSAU), éste incluye un **bypass** (derivación) que libera a las estaciones del anillo. Cada estación en el anillo funciona como un repetidor unidireccional, cada estación recibe una serie de bits de la estación predecesora, éstas reciben la información e inmediatamente retransmiten la señal a la siguiente estación. En la figura 1.3.20 se muestra la relación del modelo OSI y el estándar IEEE 802.5

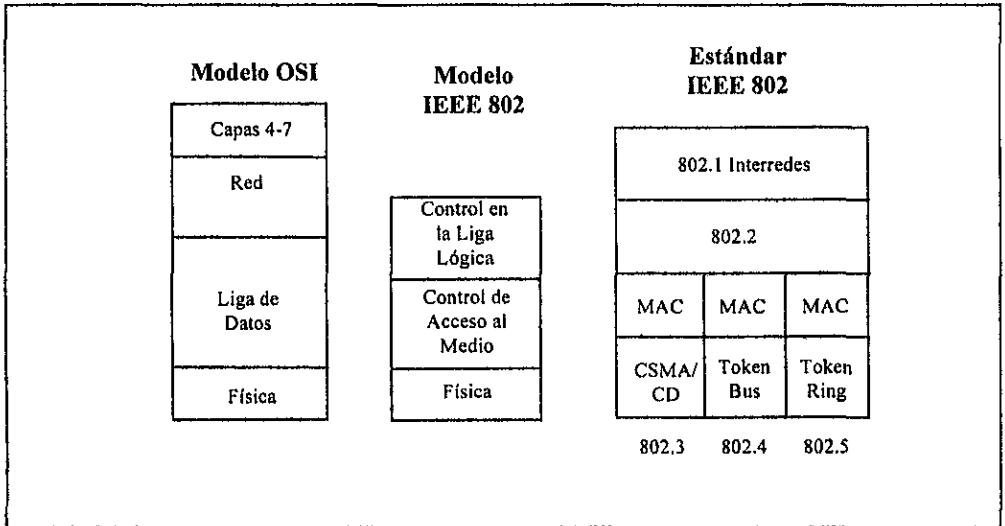


Figura 1.3.20. Relación del modelo OSI y el estándar IEEE 802.5

	IEEE 802.5	TOKEN RING DE IBM
Método de acceso	Token passing	Token passing
Topología	No específica	Estrella
Tipo de señal	Banda base	Banda base
Codificación	Manchester diferencial	Manchester diferencial
Velocidad de transmisión	1 y 4 Mbps	4 y 16 Mbps
Num. de estaciones	250	260 (STP) 72 (UTP)
Medio de comunicación	No específica	TP

Figura 1.3.21. Relación del estándar IEEE 802.5 y Token Ring de IBM.

La IEEE 802.5 especifica que el método de acceso al medio de comunicación es el token passing. El token (información especial) es el que proporciona el permiso para realizar la transmisión y solo se permite un token en el anillo. Cuando una estación recibe el token, se le da el permiso a la estación de realizar su transmisión. Esta estación puede transmitir el frame (con la información) a la siguiente estación, después de que el frame (la información enviada) retorna a la estación de origen, se elimina del anillo. Para checar si se presentaron errores en la transmisión, la estación transmisora compara la información enviada con la información retornada, si la comparación es igual se continúa con la siguiente transmisión, y si no, se transmite de nueva cuenta. Al terminar de transmitir la estación en cuestión, se genera otro token y se pasa a la siguiente estación, con lo que se le da el control de acceso al medio de comunicación.

Una de las estaciones del anillo actúa como un monitor activo, donde el monitor activo proporciona el control de tiempo en el anillo, es el que se encarga de remover del anillo frames (paquetes de datos) sin control y realiza otras operaciones de control sobre el anillo de comunicación. Todas las estaciones tienen la posibilidad de ser monitor activo en la red. Cuando no se tiene presente un monitor activo, se genera automáticamente uno.

El componente que se utiliza para conectar las estaciones es la Unidad de Acceso para Múltiples estaciones (MSAU).



**Arcnet.**

ARCnet (Attached Resource Computer Network) fue introducida por Datapoint Corporation en 1977. Varios años después, Datapoint permite el desarrollo de ARCnet a Standard Microsystems Corporation (SMC), el cual comienza a desarrollar circuitos para ARCnet. SMC presenta su primera tarjeta ARCnet en 1983, y posteriormente NCR Corporation crea la versión en **CMOS** (Complementary Metal Oxide Semiconductor) del circuito integrado de SMC. Datapoint continúa con los derechos de ARCnet, además de coordinar un proceso de certificación que ayuda a asegurar la compatibilidad entre los productos que se venden y ARCnet.

Se puede decir que todos los vendedores que se encuentran en el mercado con productos para ARCnet son miembros de la Asociación de Comercio de ARCnet (ATA ARCnet Trade Association), la cual es una asociación de usuarios y vendedores interesados en ARCnet, esta organización realiza conferencias para mantener estándares e intenta incrementar la exposición de ARCnet. La figura 1.3.22 muestra ARCnet y el modelo OSI.

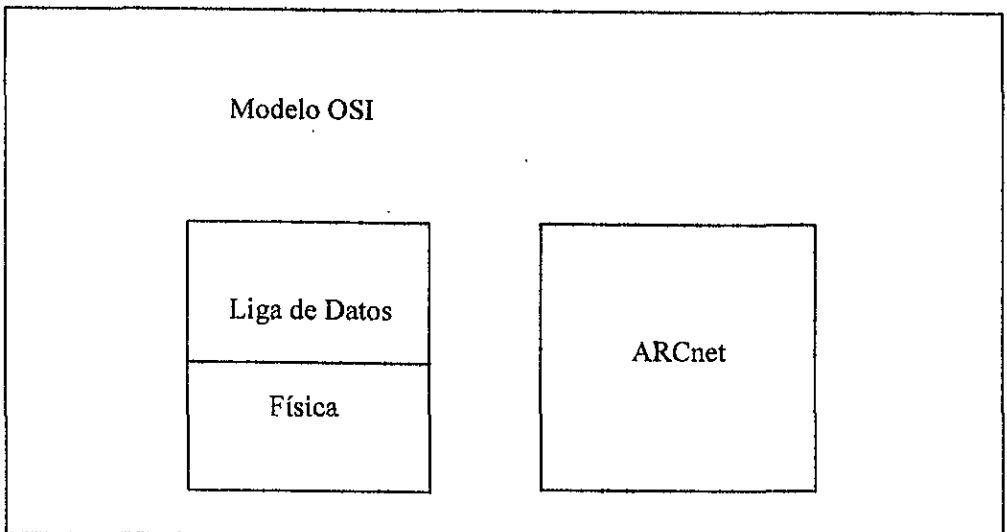


Figura 1.3.22. ARCnet y el modelo OSI

ARCnet soporta los medios de transmisión primarios (UTP, coaxial y fibra óptica) y dos topologías, la de bus y la de estrella. Se puede hacer uso de todos los tipos de medio y topologías para formar una red híbrida como se muestra en la figura 1.3.23.

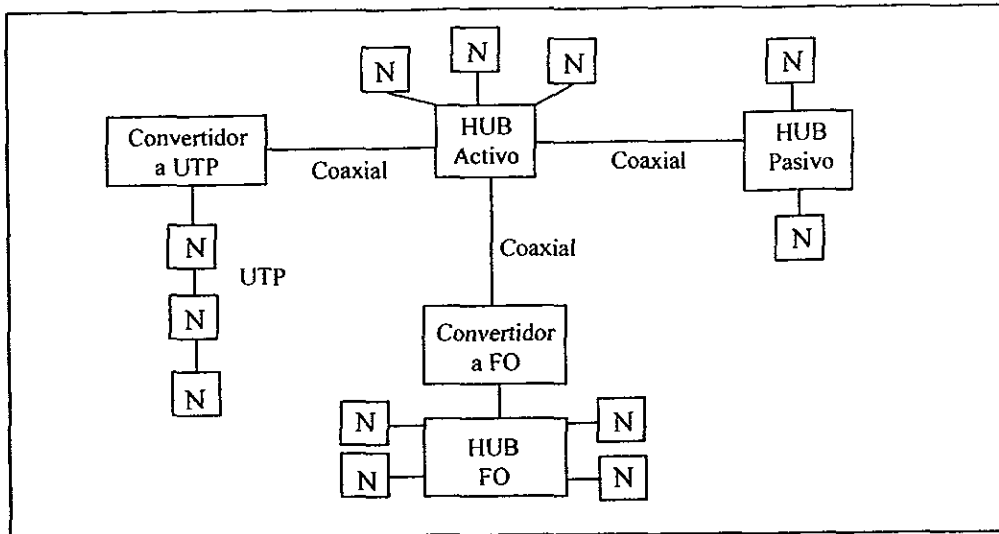


Figura 1.3.23. Posible configuración en una red ARCnet.

La mayoría de las redes ARCnet usan el cable coaxial RG62 con una topología de estrella, teniendo como elemento central a un concentrador, el cual puede ser pasivo ó activo. La longitud que se puede lograr teniendo un concentrador pasivo es de 100 pies, y si el concentrador es activo la longitud es de 2000 pies. Los concentradores pueden ser conectados a otros concentradores, los que pueden ser activos ó pasivos. La única limitante es el tiempo de propagación, el cual no debe pasar de 31 microsegundos.

Cada nodo en ARCnet tiene asignada una dirección numérica de 8 bits, esta dirección puede estar en el rango de 1 a 255. A diferencia de otras tecnologías, como la token Ring y la Ethernet, que tienen asignado de fábrica el número de la dirección, a la ARCnet se le puede cambiar la dirección de hardware mediante unos switches que se encuentran en la misma tarjeta.

Así como token ring usa el método de acceso al medio token passing, ARCnet también lo usa pero creando un anillo lógico, puesto que la topología es de estrella ó bus. Cada estación conoce su dirección (Source Identifier-SID) así como la dirección de la estación que sigue en cuanto a la dirección (Next Identifier-NID). El token es pasado de un nodo a otro en orden ascendente (tomando en cuenta la dirección del nodo). El nodo con la dirección mayor cierra el anillo completo, pasando el token al nodo con la dirección menor.

Cuando la estación entra o sale de la red, o bien ocurre una falla, la red ARCnet realiza una reconfiguración automática.

### **LocalTalk**

LocalTalk Link Access Protocol (el nombre completo de LocalTalk), es la implementación propietaria de Apple Computer's en las capas física y enlace de datos. Apple comenzó su desarrollo en 1983, y fue anunciado en 1984, año en el que se liberó la computadora Macintosh.

LocalTalk es la implementación tecnológica para grupos pequeños, utilizando una topología de bus en banda base. La interface física es EIA RS422, el cual es una interface eléctricamente balanceada como la RS449. El medio de comunicación utilizado para esta tecnología es el STP (par trenzado con blindaje) y una velocidad máxima de 230.4 Kbps. Cada segmento puede tener una longitud máxima de 300 metros y un número máximo de 32 nodos. En la figura 1.3.24 se muestra el modelo OSI y LocalTalk.

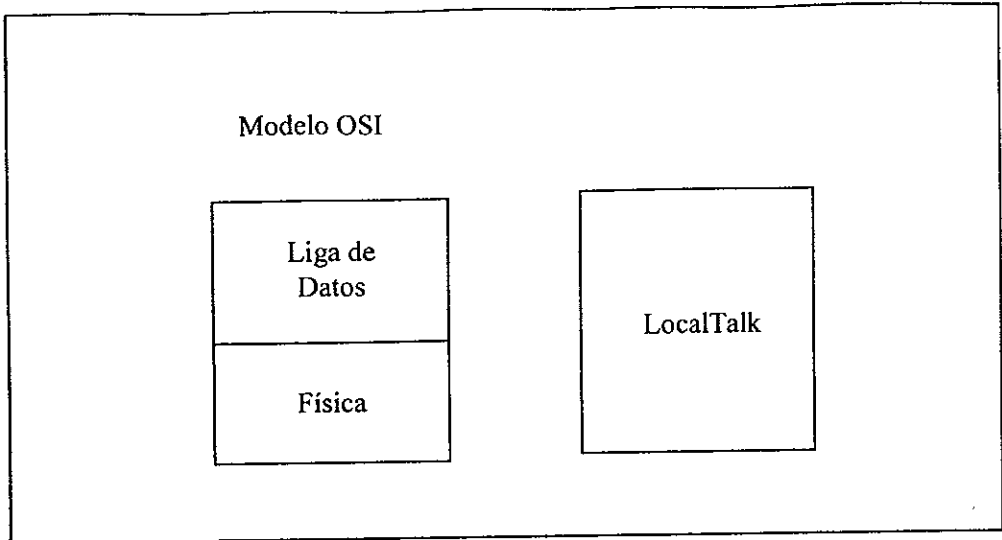


Figura 1.3.24. Modelo OSI y LocalTalk.

LocalTalk utiliza una asignación de direcciones dinámica, esto quiere decir que automáticamente se asigna una dirección cuando se inicializa. Con esta técnica se ahorra tiempo de forma muy clara, reduciendo la frecuencia de direcciones duplicadas, y es particularmente útil cuando se tiene un movimiento constante de los nodos dentro de la red.

La forma en que se asignan las direcciones es como sigue: cuando un nodo es encendido, se selecciona una dirección de forma aleatoria. Una vez que se ha seleccionado una dirección, el nodo envía un mensaje a todos los demás nodos (llamado broadcast) que contestan para verificar que dicha dirección no se encuentre duplicada. Los nodos que lleguen a tener la dirección duplicada lo anuncian en la respuesta del broadcast y el nodo intenta con una nueva dirección.

Las direcciones de LocalTalk se encuentran divididas en dos grupos: direcciones para usuarios y servidores. Las direcciones aplicadas a los usuarios se encuentran entre un rango de 1 a 127, y de 128 a 254 para servidores. La dirección 255 es reservada para broadcast, esto se respeta como en ARCnet.

Al utilizar dos grupos de direcciones se permite que LocalTalk separe los clientes de los servidores. Esto es importante cuando clientes y servidores deben ser tratados de forma diferente.

El acceso de LocalTalk y los diálogos de transmisión es como se menciona en los siguientes párrafos. Los diálogos de transmisión consisten en múltiples frames que comienzan una conversación lógica. Los diálogos de transmisión dirigida ocurren entre la transmisión de un nodo y la recepción de otro nodo. Los diálogos de transmisión para broadcast ocurren entre una transmisión de un nodo a todos los demás nodos de la red. El tiempo mínimo entre diálogos de transmisión (InterDialogue Gap IDG) es de 400 microsegundos y los frames distintos deben ser separados por un tiempo mínimo (InterFrame Gap-IFG) que es de 200 microsegundos.

### **FDDI (Fiber Distribution Data Interface)**

A mediados de los 80's las redes comienzan a ganar una mayor popularidad, y con esto se incrementa el número de nodos que se tienen conectados en una red. Con esto se dan cuenta que el ancho de banda de 10 Mbps para Ethernet y de 4 Mbps para Token Ring no es suficiente, así que se desarrollan otros estándares de red.

Se desarrolla para superar estos problemas el estándar FDDI, el cual es un enlace de alta velocidad, con el método de acceso token passing en anillo usando un cable de fibra óptica. La primera liberación fue en 1986.

Este estándar fue producido por el comité ANSI X3T9.5. Las implementaciones son producidas por muchas compañías, tales como IBM, Rockwell/CMC, Interphase Corp., Fibronics Communications, Codenoll, Cisco Systems, National Semiconductor e Intel. En la figura 1.3.25 se muestra el estándar FDDI y el modelo OSI.

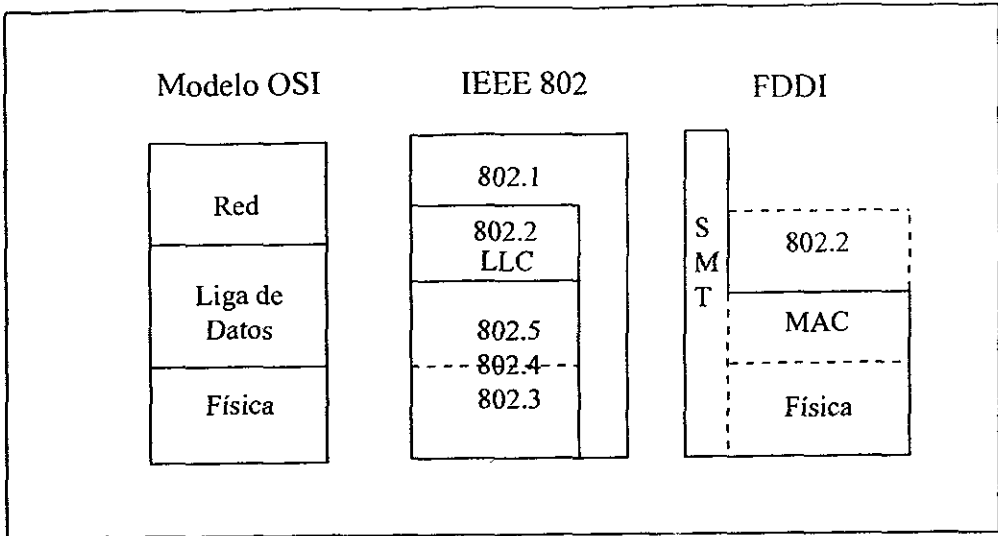


Figura 1.3.25. Modelo OSI y LocalTalk.

FDDI incluye la capa física, la subcapa MAC y especificaciones de administración de las estaciones (SMT). Ver figura 1.3.26. Este estándar fue creado para solventar las necesidades de alto ancho de banda y de un medio de comunicación seguro para redes de área local. Este estándar se usa principalmente como columna vertebral o esqueleto de redes, cuartos de conexión de computadoras en red y redes con alta velocidad de transmisión.

CARACTERÍSTICA	FDDI	IEEE 802.5
Medio	Fibra Óptica	Varios
Topología	Estrella, Anillo	Anillo
Velocidad de Transmisión	100 Mbps	1-4 Mbps
Método de acceso	Token Passing	Token Passing
Señal	Banda Base	Banda Base
Codificación	NRZI-4B/5B	Manchester Diferencial
Direcciones	16 y 48 bits	16 y 48 bits

Figura 1.3.26. Características de FDDI y IEEE 802.5.

En los esqueletos o columna vertebral de redes se proporciona una alta velocidad que permite conectar otras redes. Los cuartos de conexión de red sirven para conectar mainframes, minicomputadoras y otros periféricos en un simple salón de red. Las redes de área local que usan este estándar sirven para conectar minicomputadoras de muy alta velocidad.

FDDI es muy similar a IEEE 802.5, la razón es que ambos utilizan el método de acceso token passing, por lo tanto es determinístico. Ambos pueden ser configurados con topología de estrella, y debido al medio es totalmente inmune a interferencias electromagnéticas.

Las redes FDDI consisten de dos anillos, cada uno de ellos para una dirección de transmisión (en uno se transmite en un sentido y en el otro en sentido contrario). Uno de estos anillos es el primario y el otro es el anillo secundario. Bajo condiciones normales la información fluye sólo en un anillo (el primario), dejando al anillo secundario como respaldo. Cuando se presenta una falla en el anillo primario, se configura automáticamente la red para que no se quede sin operar (activando el anillo secundario).

Se menciona que en un anillo FDDI no se puede exceder de 500 estaciones en una longitud total de cable de 100 Km.

Las estaciones que se tienen en una red FDDI pueden estar conectadas hacia un concentrador de conexión punto a punto, teniendo estaciones de clase A y clase B. Las estaciones de clase A son aquellas que se encuentran conectadas a ambos anillos y las de clase B son aquellas que se tienen conectadas en un solo anillo. Como las estaciones de clase A se tienen conectadas a ambos anillos, éstas pueden permanecer en la red al presentarse una reconfiguración provocada por una falla en la red, en cambio las que se encuentran conectadas a un solo anillo no. Véase figura 1.3.27 Configuración típica de FDDI.

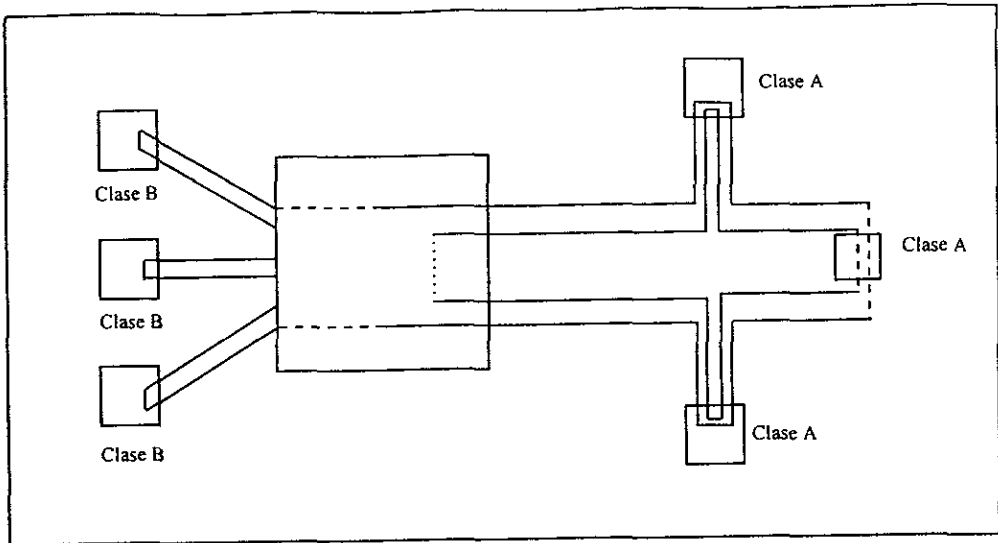


Figura 1.3.27. Configuración típica de FDDI.

### El Modelo de Referencia OSI.

Los estándares contemporáneos no están diseñados para ser cincelados en piedra con la finalidad de no ser modificados, por el contrario, están diseñados para ser aumentados y modificados de modo que nuevas tecnologías puedan coexistir con dispositivos anteriores.

En 1977 la ISO (International Organization for Standardization) formó un comité para estudiar la compatibilidad de equipo para redes, trabajo que condujo eventualmente a la publicación del modelo Open System Interconnection (OSI). En este contexto "Sistema Abierto" se refiere a un modelo de red abierto a equipo de fabricantes de la competencia. Como han señalado Frank Deefler y William Stalling, "el modelo de referencia OSI es útil para cualquier persona involucrada en la compra o manejo de una red local, porque ofrece un marco teórico..." mediante el cual se pueden entender problemas y oportunidades de la conexión en redes.

El modelo de referencia fue creado para hacer posible "la definición de procedimientos estandarizados que permiten la interconexión y el subsiguiente intercambio efectivo de



información entre usuarios". Usuarios en este sentido se refiere a sistemas que constan de una o más computadoras, software asociado, periféricos, terminales, operadores humanos, procesos físicos, mecanismos de transferencia de información y elementos relacionados. Estos elementos, juntos, deben poder realizar procesamiento y/o transferencia de información. Los estándares desarrollados a partir del modelo de referencia permitirán a diversas redes del mismo tipo o diferentes, comunicarse fácilmente entre sí, como si constituyeran una misma red.

Hay que tener presente que el apego al modelo de referencia no implica ninguna implantación o tecnología en particular.

El modelo de referencia OSI tiene siete capas. Los principios aplicados para el establecimiento de siete capas fueron los siguientes:

1. Una capa se creará en situaciones en donde se necesita un nivel diferente de abstracción.
2. Cada capa deberá efectuar una función bien definida.
3. La función que deberá realizar cada capa deberá seleccionarse con la intención de definir protocolos (reglas para llevar la comunicación) normalizados internacionalmente.
4. Los límites y las capas deberán seleccionarse tomando en cuenta la minimización del flujo de información a través de las interfaces.
5. El número de capas deberá ser lo suficientemente grande para que funciones diferentes no tengan que ponerse juntas en la misma capa y, por otra parte, también deberá ser lo suficientemente pequeño para que su arquitectura no llegue a ser difícil de manejar.

El alcance o campo de acción del modelo de referencia es relativamente amplio y puede resumirse en los cinco puntos siguientes:

1. Especificar una estructura lógica de aplicación universal que abarque vastas aplicaciones de comunicaciones, en especial las del CCITT (Consultative Committee on International Telegraphy and Telephony).
2. Actuar como referencia durante el desarrollo de nuevos servicios de comunicaciones.
3. Permitir a diversos usuarios establecer comunicación entre sí, alentando la implantación compatible de características de comunicación.
4. Hacer posible la evolución sostenida de las aplicaciones de comunicación, en particular las del CCITT, otorgando la flexibilidad suficiente para que puedan tener cabida los adelantos tecnológicos y las necesidades en evolución de los usuarios.
5. Hacer posible la satisfacción de nuevos requisitos de los usuarios en forma compatible con servicios existentes y que sean consistentes con el modelo OSI.

El objetivo fundamental del modelo OSI es definir un conjunto de recomendaciones que permitan cooperar a sistemas abiertos. La cooperación contempla una amplia variedad de actividades:

1. Comunicación entre procesos: intercambio de información y sincronización de la actividad entre procesos de aplicación del modelo OSI.
2. Interés en todos los aspectos de la creación y conservación de descripciones y transformaciones de datos para reformatear datos que se intercambian entre sistemas abiertos.
3. Interés en medios de almacenamiento, sistemas de archivos y bases de datos para manejar y ofrecer acceso a datos almacenados en los medios.

4. Administración de procesos y recursos mediante el cual se declaran, inician y controlan procesos de aplicación del modelo OSI; además, es el medio a través del cual adquieren recursos de OSI.
5. Integridad y seguridad de los datos durante la operación de sistemas abiertos.
6. Soporte de acceso integral a los programas ejecutados por procesos de aplicación del modelo OSI.

A continuación se presenta una gráfica que muestra el modelo OSI, para posteriormente describir cada una de las capas que lo forman.

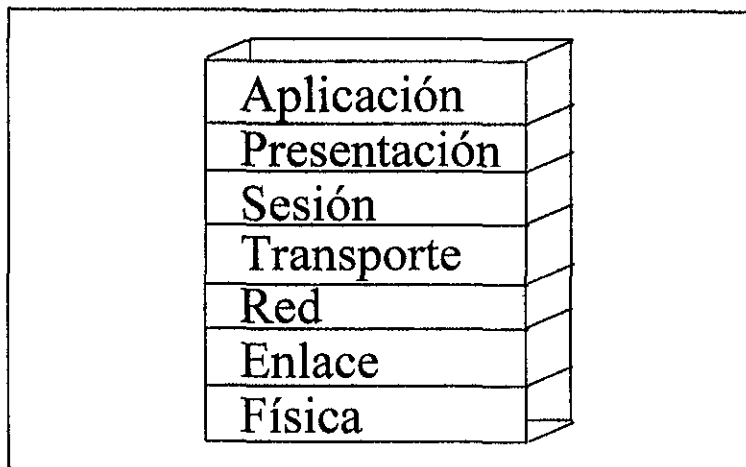


Figura 1.3.28. Capas del Modelo OSI.

### Capa Física.

La capa física se ocupa de la transmisión de bits a lo largo de un canal de comunicación. Su diseño debe asegurar que cuando un extremo envía un bit con valor 1, éste se reciba exactamente como un bit con ese valor en el otro extremo, y no como un bit de valor 0. Preguntas comunes

aquí son cuantos *volts* deberán utilizarse para representar un bit de valor 1 ó 0; cuántos microsegundos deberá durar un bit; la posibilidad de realizar transmisiones bidireccionales en forma simultánea; la forma de establecer la conexión inicial y cómo interrumpirla cuando ambos extremos terminan su comunicación; o bien cuantas puntas terminales tiene el conector de la red y cual es el uso de cada una de ellas. Los problemas de diseño a considerar aquí son los aspectos mecánicos, eléctrico, de procedimiento de interface y el medio de transmisión física.

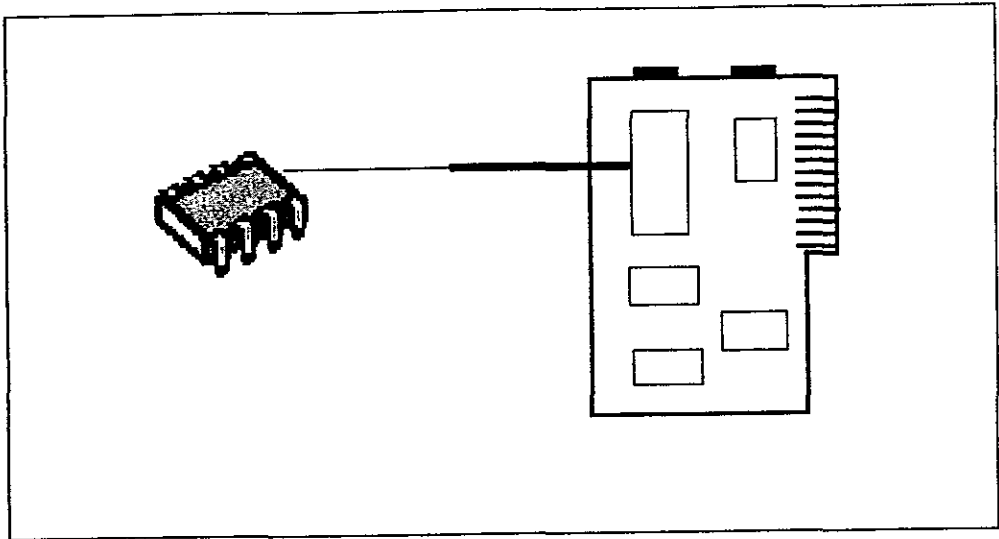


Figura 1.3.29. Capa Física.

### Capa de Enlace.

La función principal de la capa de enlace consiste en que, a partir de un medio de transmisión común y corriente, transformarlo en una línea sin errores de transmisión para la capa superior. Esta tarea se realiza al hacer que el emisor fragmente la información de entrada en tramas de datos, y las transmita en forma secuencial, posteriormente procese las tramas de asentimiento devueltas por el receptor, al realizar este seccionamiento de los datos se forma lo que se le llama frame, el cual tiene la información a enviar, así como información de control al inicio y final del paquete. Esta información puede incluir la dirección fuente y destino, longitud de frame, y otra

información requerida por capas superiores. Recae sobre la capa de red el enlace, la creación o reconocimiento de los límites de la trama. La trama puede destruirse debido al ruido en la línea, en cuyo caso el software de la capa de enlace, perteneciente a la máquina emisora, deberá retransmitir la trama. Sin embargo, múltiples transmisiones de la misma trama introducen la posibilidad de duplicar la misma. Corresponde a esta capa resolver los problemas causados por daño, pérdida o duplicidad de tramas. Otro de los aspectos que se toman en cuenta en la capa de enlace, es cómo evitar que un transmisor muy rápido saturé con datos a un receptor lento, se deberá emplear un mecanismo de regulación de tráfico que permita que el transmisor conozca el espacio de memoria que en ese momento tiene el receptor.

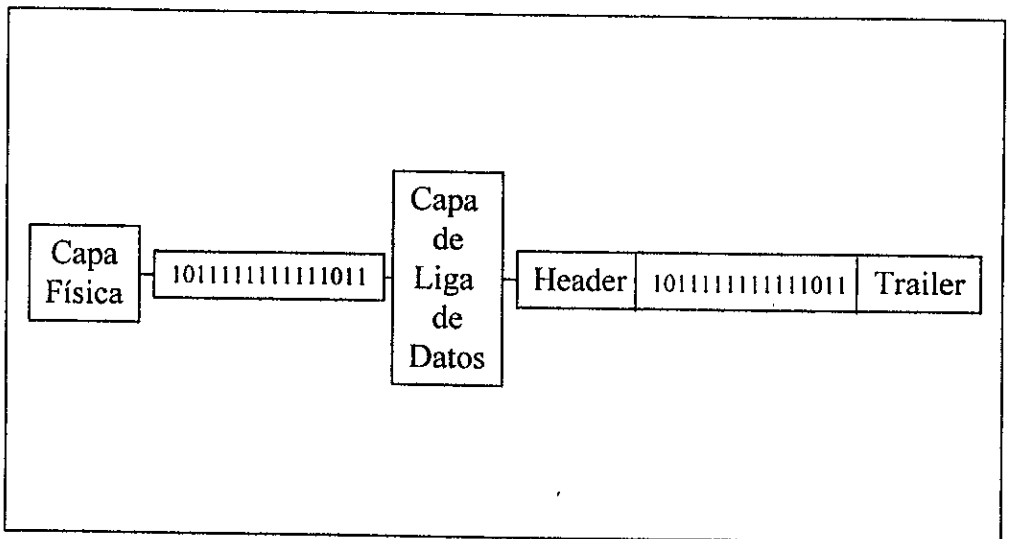


Figura 1.3.30. Capa de Enlace. (Liga de Datos).

### Capa de Red.

Un aspecto que se maneja aquí es la determinación sobre cómo encaminar los paquetes del origen al destino. Las rutas podrían basarse en tablas estáticas que se encuentran "cableadas" en la red y que difícilmente podrían cambiarse. Si en un momento dado hay demasiados paquetes presentes en la subred, ellos mismos se obstruirían mutuamente y darían lugar a un cuello de

botella. El control de la congestión dependerá también de esta capa. Por ejemplo, en la figura 1.3.31 se muestra una red formada por cuatro subredes, de tal forma que para que un cliente de A envíe información a otro cliente de D se debe dirigir la información por una ruta correcta.

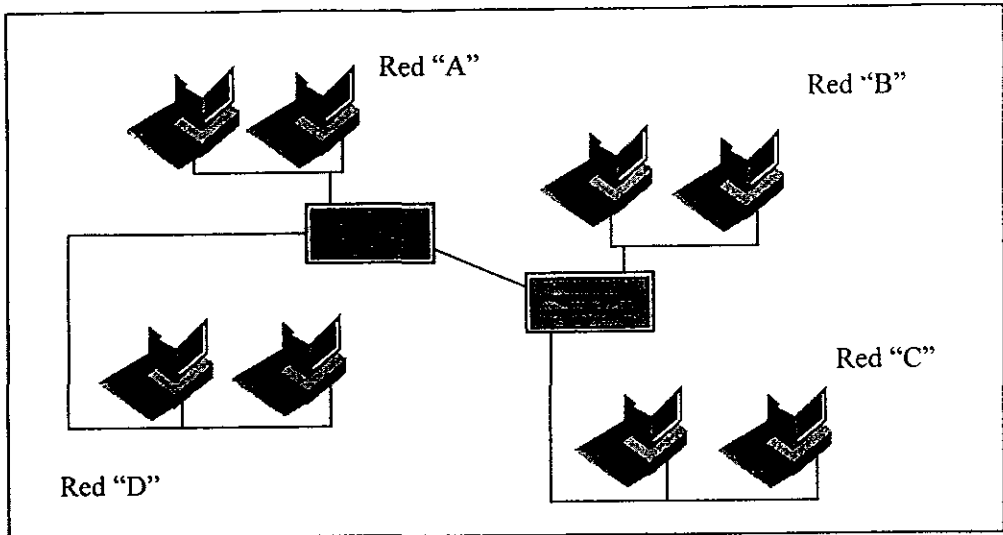


Figura 1.3.31. Capa de Red.

### Capa de Transporte.

La función principal de esta capa consiste en aceptar los datos de la capa de sesión, dividirlos, siempre que sea necesario, en unidades más pequeñas, pasarlos a la capa de red y asegurar que todas estas unidades lleguen correctamente al otro extremo. Todo este trabajo se debe hacer de manera eficiente, de tal forma que aisle la capa de sesión de los cambios inevitables a los que está sujeta la tecnología del hardware.

La capa de transporte crea una conexión de red distinta para cada conexión de transporte solicitada por la capa de sesión. Si la conexión de transporte necesita un gran caudal, ésta podría crear múltiples conexiones de red, dividiendo los datos entre las conexiones de la red con objeto de mejorar dicho caudal. Si la creación o mantenimiento de la conexión de una red resulta

costoso, la capa de transporte podría multiplexar varias conexiones de transporte sobre la misma conexión de red para reducir dicho costo. En todos los casos, la capa de transporte se necesita para hacer el trabajo de multiplexión transparente a la capa de sesión.

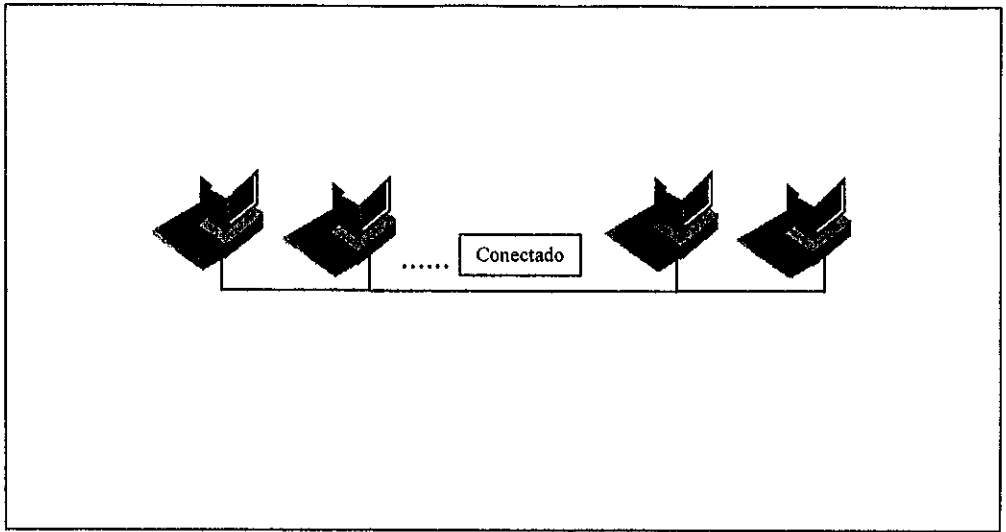


Figura 1.3.32. Capa de transporte.

### Capa de Sesión.

Esta capa permite que los usuarios de diferentes máquinas puedan establecer sesiones entre ellos. A través de una sesión se puede llevar a cabo un transporte de servicio ordinario, tal como lo hace la capa de transporte, pero mejorando los servicios que ésta proporciona y que se utilizan en algunas aplicaciones. Una sesión podría permitir al usuario acceder a un sistema de tiempo compartido a distancia, o transferir un archivo entre dos máquinas.

Uno de los servicios de la capa de sesión consiste en gestionar el control de diálogo. Las sesiones permiten que el tráfico vaya en ambas direcciones al mismo tiempo, en una sola dirección en un instante dado. Si el tráfico sólo puede ir en una dirección en un momento dado, la capa de sesión ayudará en el seguimiento de quien tiene el turno.

Otro de los servicios de esta capa es la sincronización. Por ejemplo, en la transferencia de archivos la capa de sesión proporciona una forma de insertar puntos de verificación en el flujo de datos.

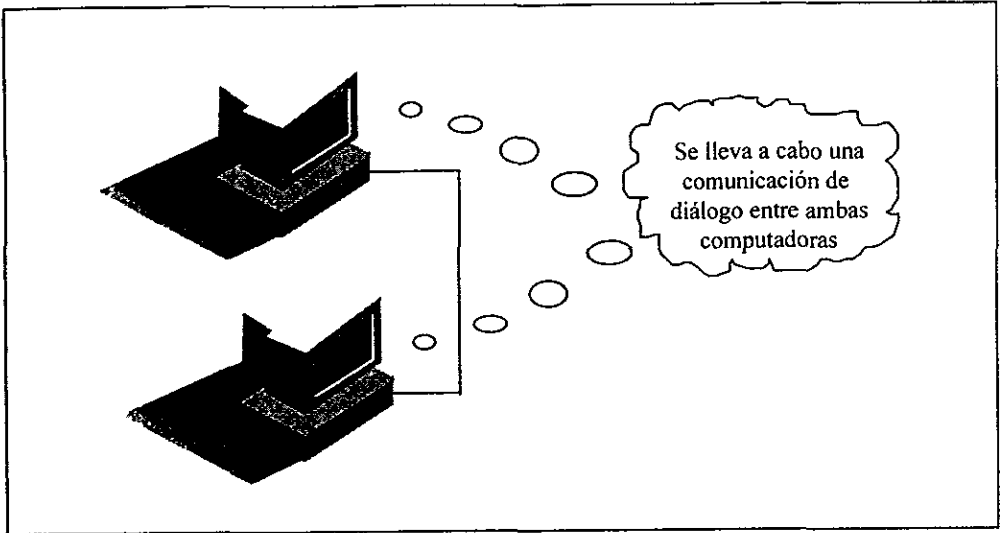


Figura 1.3.33. Capa de Sesión.

### Capa de Presentación.

Esta capa realiza funciones que se necesitan bastante a menudo como para buscar una solución general para cada problema, más que dejar que cada uno de los usuarios resuelva los problemas. En particular y a diferencia de las capas inferiores, que únicamente están interesadas en el movimiento fiable de bits de un lugar a otro, la capa de presentación se ocupa de los aspectos de sintaxis y semántica de la información que se transmite.

La capa de presentación está relacionada también con otros aspectos de representación de información. Por ejemplo, la compresión de datos se puede utilizar aquí para reducir el número de bits que tiene que transmitir, y el concepto de *criptografía* se necesita usar para efectos de privacidad y de autenticación.



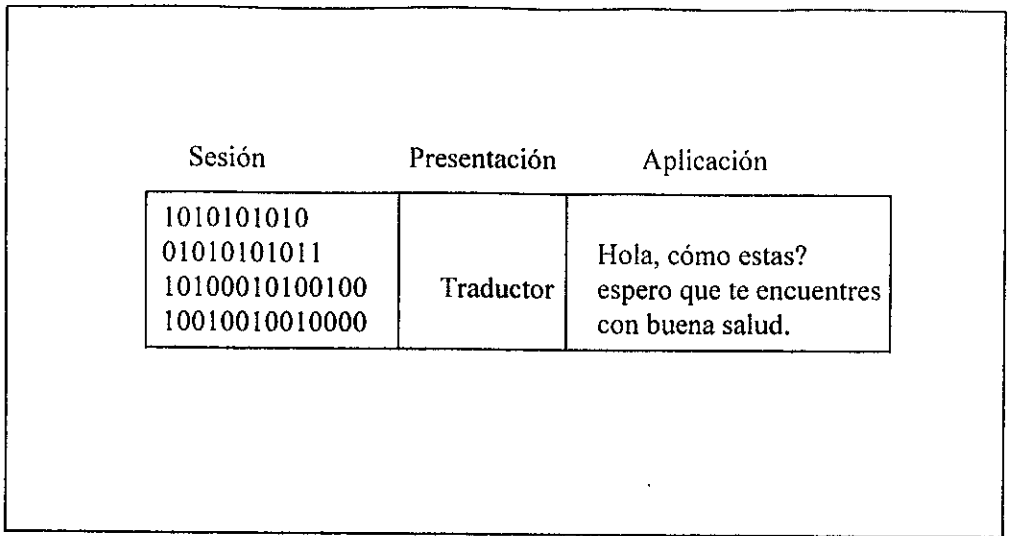


Figura 1.3.34. Capa de Presentación.

### Capa de Aplicación.

Una de las funciones de la capa de aplicación es la transferencia de archivos. Distintos sistemas de archivo tienen diferentes convenciones para denominar un archivo, así como diferentes formas de representar las líneas de texto, etc. La transferencia de archivos entre dos sistemas diferentes requiere de una resolución de éstas y otras incompatibilidades. Este trabajo, así como el correo electrónico, la entrada de trabajo a distancia, el servicio de directorio y otros servicios de propósito general y específico, también corresponden a la capa de aplicación.

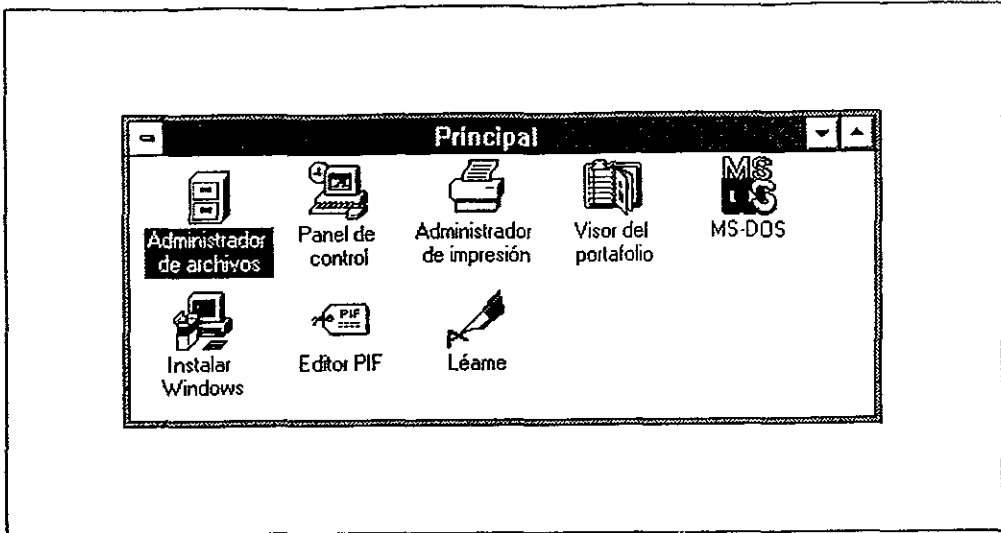


Figura 1.3.35. Capa de Aplicación.

La capa de aplicación contiene una gran variedad de protocolos que se necesitan frecuentemente. De manera general, especifica la interface de comunicación entre el usuario y la computadora. Como hemos visto el modelo OSI de la ISO (International Organization for Standardization), está orientado a permitir la comunicación entre dos o más computadoras que pueden tener o ser sistemas diferentes, esta comunicación se efectúa desde la capa superior hacia abajo en la computadora fuente, y en la computadora destino de la capa más baja hacia arriba.

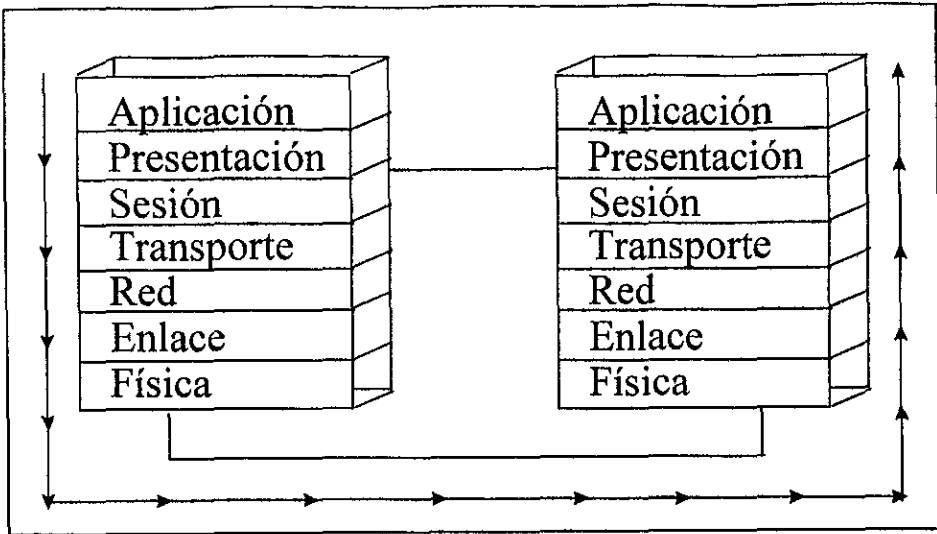


Figura 1.3.36. Comunicación entre dos equipos de cómputo.

Cada una de las capas usa los servicios de la capa superior para proporcionar servicio a la capa inferior. Por ejemplo; la capa de aplicación necesita comunicarse con su similar en la computadora destino, para lo cual necesita pasar su mensaje a la capa de presentación, ésta pasa la información a la capa de sesión y le coloca el encabezado correspondiente, luego esta capa pasa la información a la capa de transporte y le coloca el encabezado, este proceso se realiza para cada una de las capas hasta llegar a la física, en donde se transmite la información por el medio de comunicación mediante una señal eléctrica, esta señal es tomada en la computadora destino y se pasa de la capa física a la de liga de datos quitándole el encabezado y la información de control, la capa de liga de datos pasa la información a la capa de red, quitándole el encabezado; este proceso sigue hasta la capa de aplicación la cual es el destino.

Como vemos cada capa introduce un encabezado e información adicional para el control que realiza la capa correspondiente.

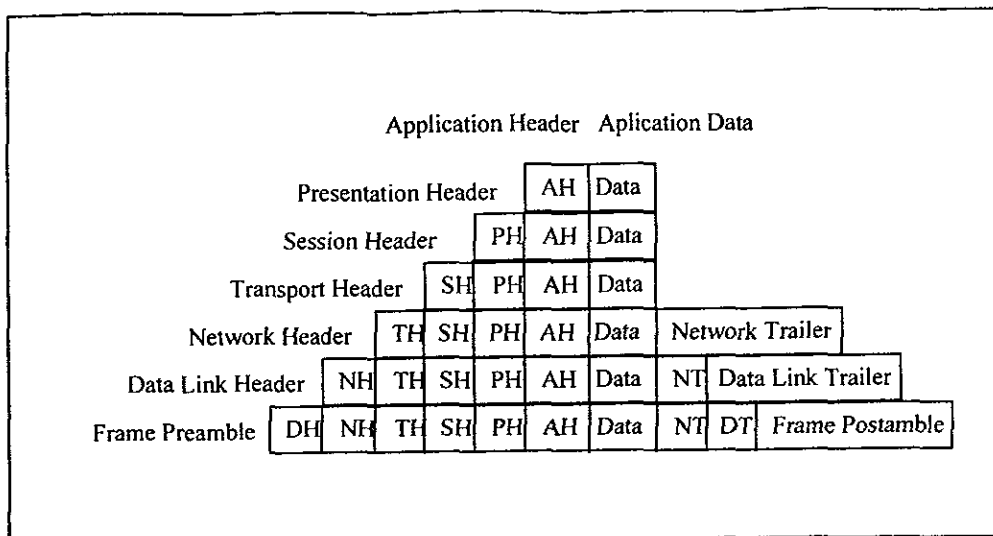


Figura 1.3.37. Encabezados introducidos en la información.

La forma de identificar a la información en cada capa es diferente, y estas son: a nivel físico se le llama bits, en la capa de liga de datos al conjunto de bits se le llama frame, en la capa de red se le da el nombre de datagrama, en la capa de transporte, sesión y presentación se le llama datagrama, y por último en la de aplicación se le da el nombre de mensaje.

### Comunicación entre Redes.

A continuación se verán los dispositivos que se encargan de realizar la comunicación entre las redes que ya existen.

Los dispositivos que se encargan de comunicar redes se dividen en categorías, dependiendo de la capa en la cual trabajan. El repetidor, trabaja en la capa física; el puente, trabaja en la capa de enlace; el ruteador, trabaja en la capa de red; y el gateway, trabaja en las siete capas.

### Repetidor.

La transmisión de las señales por el medio de comunicación se realiza a ciertas distancias limitadas, esto se debe a que la señal se atenúa con la distancia recorrida, se distorsiona por ruido, o se pierde la integridad de la información enviada.

Para solventar los problemas que se mencionan en el párrafo anterior se debe amplificar la señal, pero si se introduce un simple amplificador se podría amplificar también la señal del ruido, por tal motivo no es buena solución un simple amplificador.

El repetidor se encarga de reproducir la señal y retransmitirla. Este dispositivo no realiza un análisis de la señal ni de la estructura de la misma, sólo recibe la información, la reproduce eliminando ruido y la retransmite. La señal nueva es una réplica idéntica de la señal originalmente transmitida, y tiene las características necesarias para poder ser transmitida sobre otro segmento. De forma práctica se debe tener como máximo cuatro repetidores entre dos puntos.

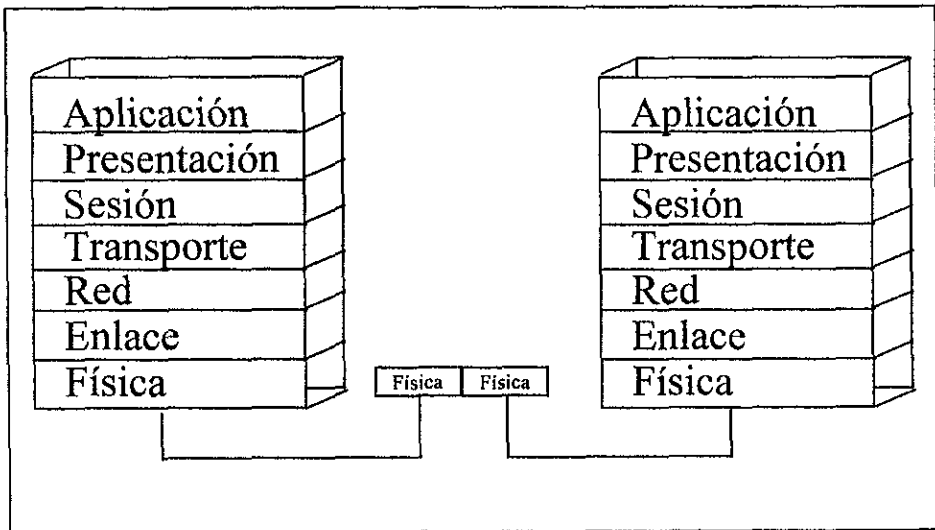


Figura 1.3.38. Capas abarcadas por un repetidor.

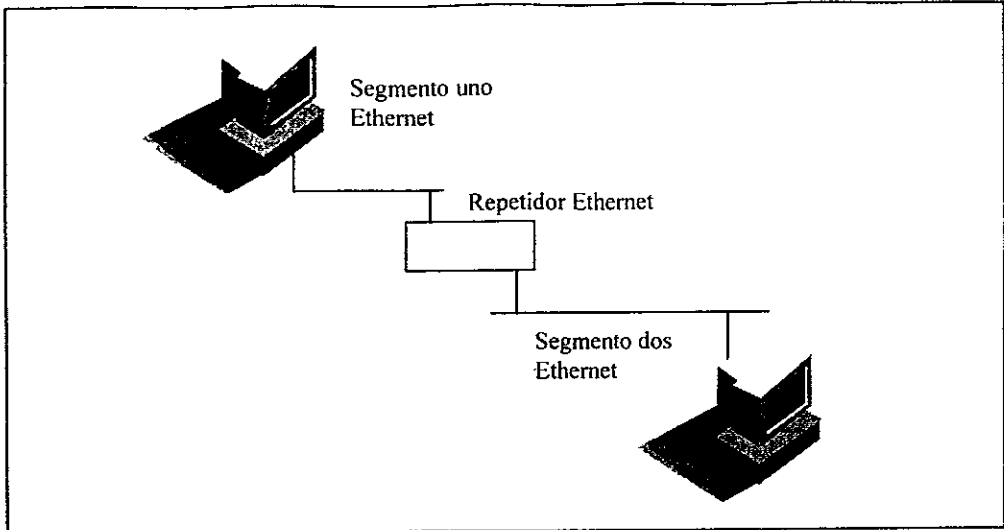


Figura 1.3.39. Funciones de un Repetidor.

### **Puente.**

Los puentes son usados para la conexión de dos segmentos de una LAN a nivel de la capa de enlace. En esta capa se tiene acceso a la dirección física del dispositivo, de tal forma que puede determinar la dirección fuente y destino de la información. Una vez que el repetidor ha determinado las direcciones, éste puede permitir o denegar el acceso al nuevo segmento.

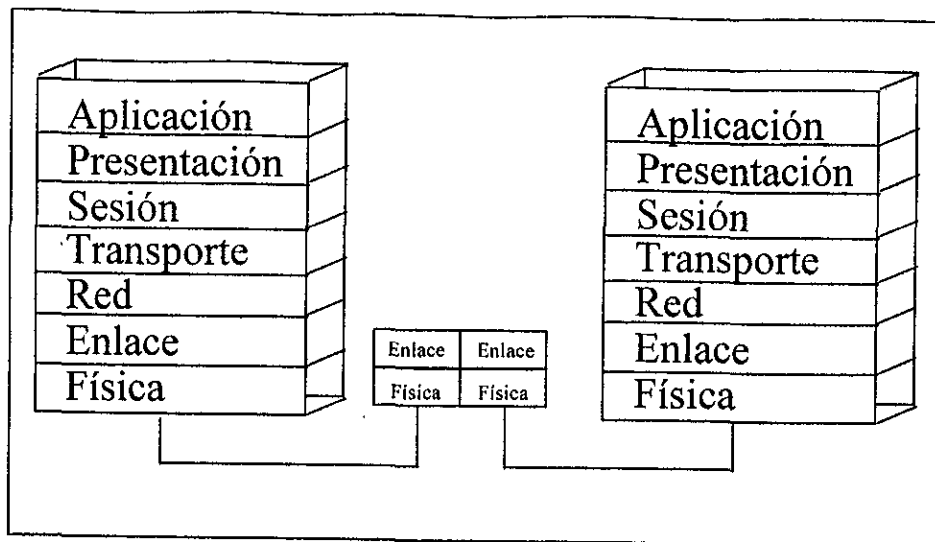


Figura 1.3.40. Capas abarcadas por un puente.

Debido a la habilidad de un puente para filtrar la dirección de las estaciones se usa para dividir en dos redes un segmento dado. Una vez hecha la división, el puente evita el tráfico de otras redes o segmentos. Cuando se tiene una red con trabajo pesado se puede dividir en dos segmentos para dividir el tráfico, al hacer esto se reduce el tráfico en cada segmento.

Se tienen dos tipos de puentes: *transparente* y *puente de ruteo*. El puente fuente de ruteo son encontrados principalmente en redes IBM, y los puentes transparentes son encontrados en cualquier otra red.

Los puentes transparentes no requieren programación inicial. Una vez instalados en una red aprenden la localización de los dispositivos de la red al asociar la dirección fuente de los paquetes con la línea que siguen para ser recibidos. Haciendo uso de esta característica, el puente crea una tabla con una relación de los pares de segmentos con los dispositivos. El puente delibera la necesidad de transmitir el paquete al siguiente segmento, esto lo realiza al analizar la dirección destino del paquete; si esta dirección se encuentra en el mismo segmento de donde se originó,

descarga el paquete, y si la dirección no se encuentra en el mismo segmento se realiza la transmisión al siguiente.

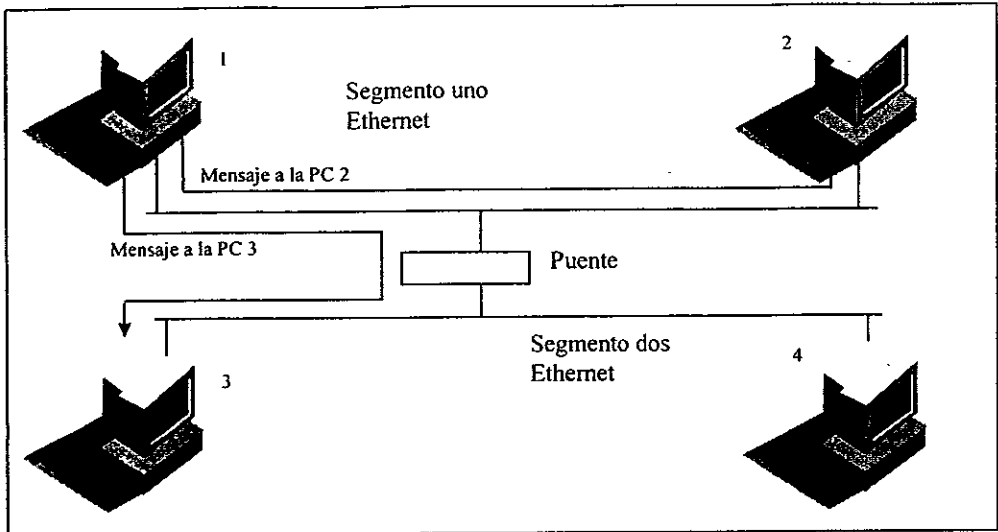


Figura 1.3.41. Funciones realizadas por un puente.

En el puente de ruteo fuente se incluye toda la dirección de la estación destino en el paquete. Como se tiene la dirección completa del dispositivo destino; el puente transmite el paquete al segmento correspondiente.

### Ruteador.

Los ruteadores trabajan sobre las tres capas inferiores (red, enlace y física), en la capa de red se incluye la dirección lógica de red, la cual es comúnmente asignada por el administrador de la red. La dirección física de los dispositivos normalmente no puede ser asignada por el administrador, ésta es de fábrica.

La dirección lógica normalmente es asociada por el administrador con un grupo específico de dispositivos con alguna característica en común.



Los ruteadores envían información a través de la red usando principalmente la dirección lógica. Las subdivisiones lógicas de red son normalmente llamadas subredes.

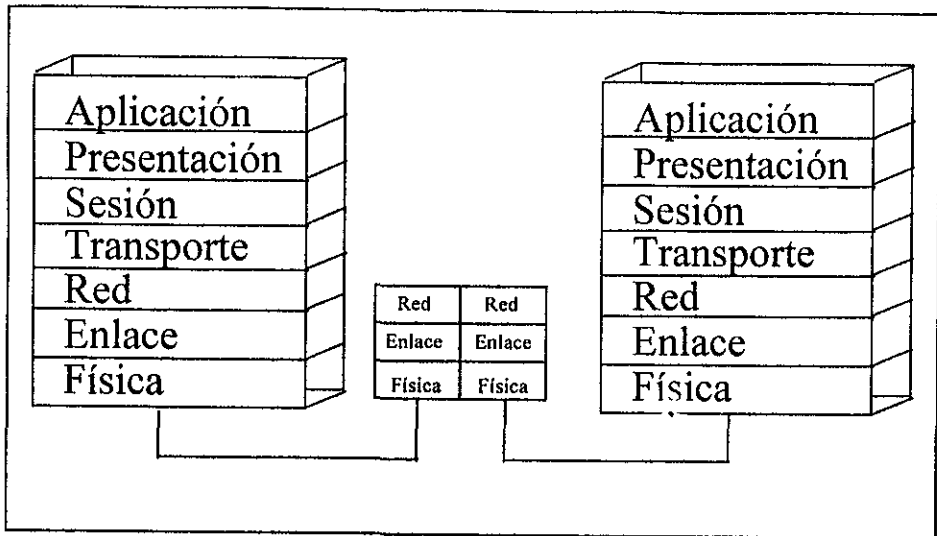


Figura 1.3.42. Capas abarcadas por un ruteador.

Los ruteadores difieren de los puentes en el uso de la dirección lógica sobre la dirección física, además de usar un algoritmo específico de ruteo para obtener la mejor ruta. Para el cálculo de la mejor ruta pueden tomar diferentes factores, como por ejemplo, se puede calcular en base al número de saltos (ruteadores por los que pasa la información) realizados para llegar al destino, o bien, puede basarse en el tiempo de viaje. Los ruteadores más modernos toman en cuenta varios factores para realizar el cálculo de la mejor ruta.

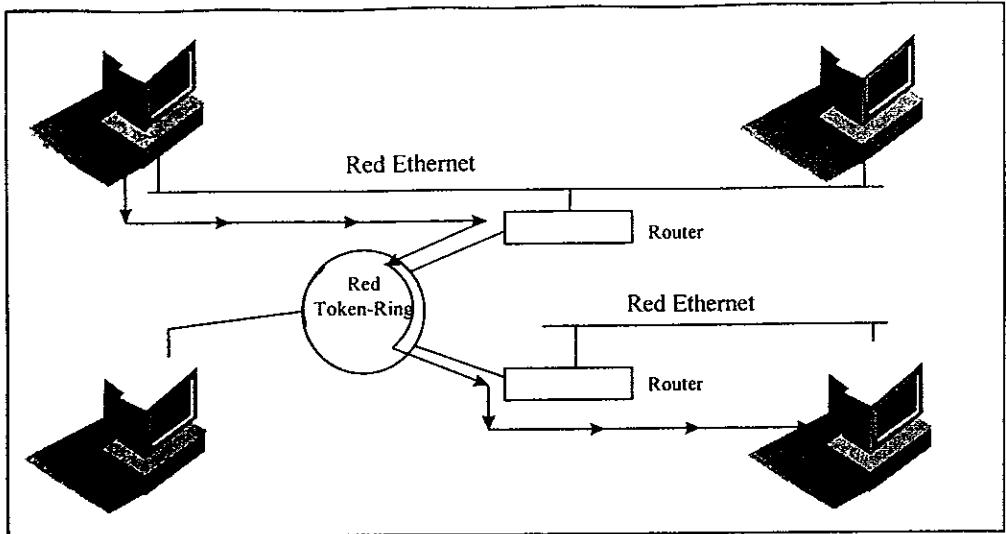


Figura 1.3.43. Funciones de un ruteador.

Mucho de los ruteadores actuales son en realidad b-ruteadores (brouters), los brouters son ruteadores que tienen funciones de puentes (bridge). Muchos de los algoritmos son específicos para protocolos particulares de redes. Los brouters primero checan un paquete para ver si soportan el algoritmo de ruteo, si no, simplemente pasan el paquete, este paquete es pasado usando la información de la capa de enlace.

### Gateway.

De los dispositivos analizados en las páginas anteriores no existe uno que maneje todas las capas del modelo OSI, es decir, que si se tienen dos redes con un manejo de protocolos diferentes en las capas superiores a la de red, no se podrían interconectar para llevar a cabo una comunicación.

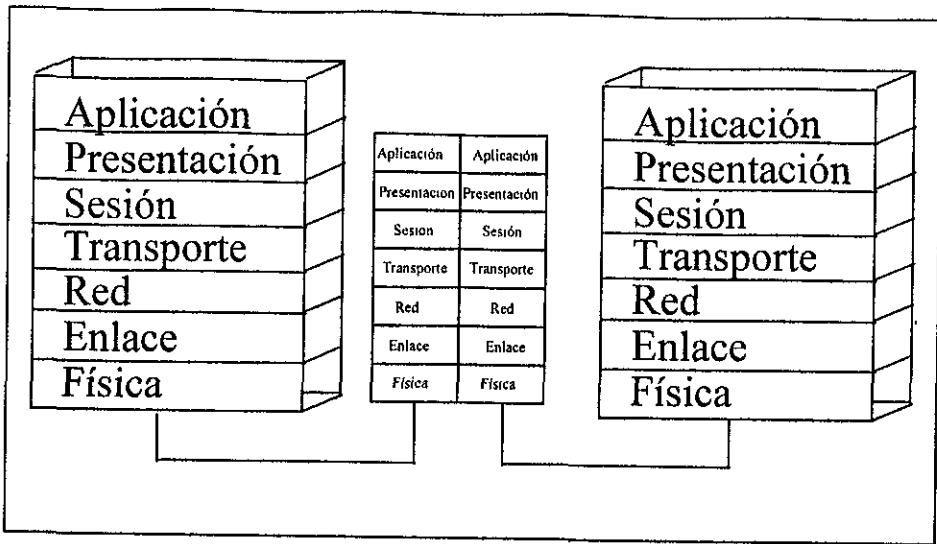


Figura 1.3.44. Capas abarcadas por un gateway.

Como se explicó en las líneas anteriores, se requiere de un dispositivo que pueda realizar una conversión de un protocolo a otro para las capas superiores. Por ejemplo si se deseara comunicar una red con un sistema operativo NetWare de Novell con una red SNA de IBM, en este caso se tiene que el hardware es diferente, pero también lo es el software, por tal motivo los protocolos de comunicación son diferentes. Para realizar esta comunicación se requiere de un gateway para realizar la conversión de los protocolos que no se comprenden de un sistema a otro.

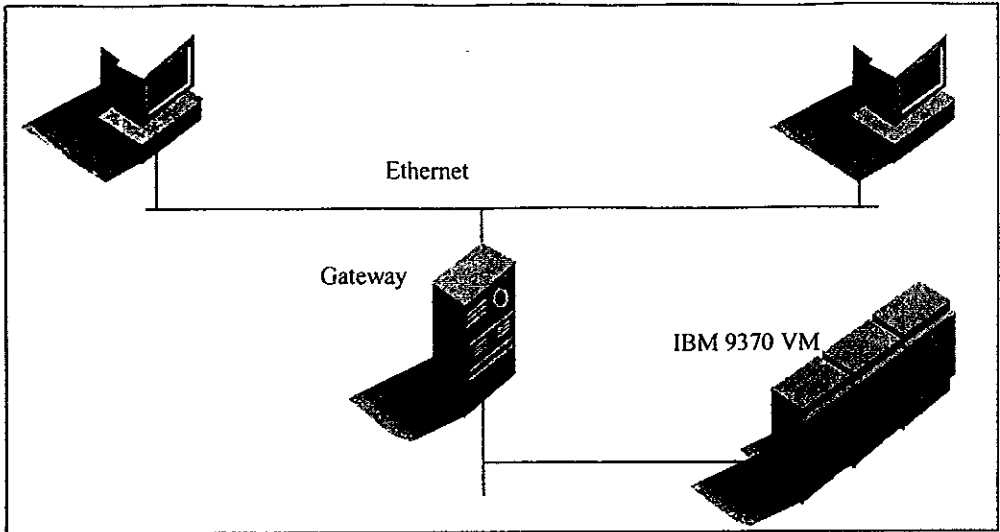


Figura 1.3.45. Funciones de un gateway.

Un gateway puede ser standalone, productos a nivel de caja o una combinación de tarjetas y software. Los dispositivos a nivel de caja (dispositivo especializado) son más costosos, pero proporcionan un desempeño más alto (performance). La combinación de tarjetas y software pueden ser dedicados y no dedicados.

### Conjunto de Protocolos de TCP.

TCP/IP tiene sus inicios como un experimento conducido por el Departamento de Defensa, esto fue a mediados de la década de los 70's. La finalidad de este proyecto fue el de proporcionar una conectividad entre computadoras con sistema operativo totalmente diferente.

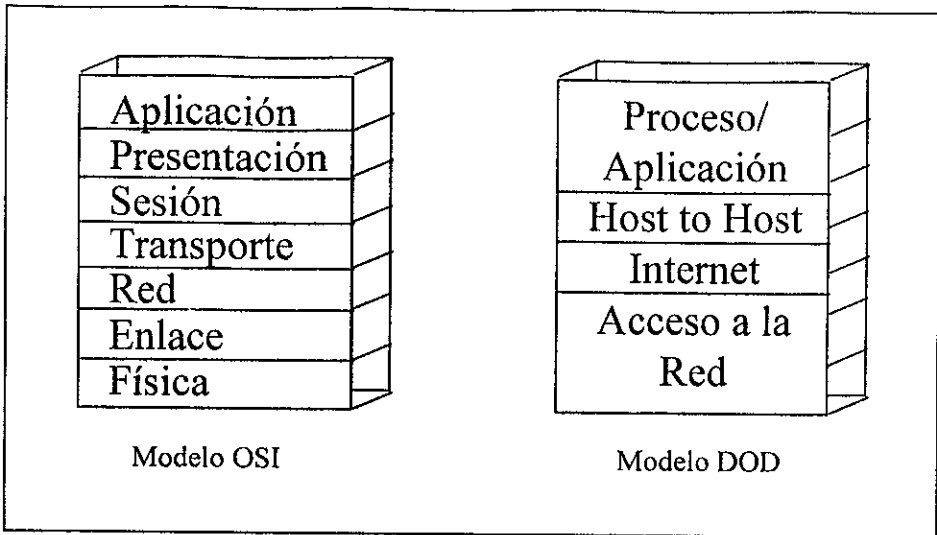


Figura 1.3.46. Modelo DOD.

Para realizar este proyecto el Departamento de Defensa de los Estados Unidos definió un modelo de red (modelo DOD) de cuatro capas, en cada una de las capas de este modelo se incluyeron un conjunto de protocolos para realizar ciertas funciones, al conjunto de todos estos protocolos en las diferentes capas se le conoce como el conjunto de protocolos TCP/IP.

Actualmente TCP/IP es un estándar para el sistema operativo Unix. Además TCP/IP se ha implementado para la mayoría de los equipos de cómputo, tales como PCs, Sun, VAX, Apple, minicomputadoras y mainframes.

Además el conjunto de protocolos de TCP/IP es ampliamente utilizado en todo el planeta, esto es en base a la conexión que se ha realizado de la mayoría de las redes grandes. A esta interconexión que se realizó, se le llamó Internet. Las redes que se conectaron son las siguientes:

#### ARPANET

Defense Advanced Research Project Agency (DARPA, formalmente llamada la red ARPA).

**MILNET**

Department of Defense Network.

**NFSnet**

Redes conjuntas de la Fundación Nacional de Ciencia.

**CSNET**

Red de ciencia de cómputo.

**Cypress Net**

Principalmente universidades.

La red que fue concebida e implementada por la DARPA, se basó en la transmisión de información en redes de *paquetes conmutados*. Una red de paquetes conmutados transmite la información por la red en pequeños segmentos, a estos segmentos se les denomina paquetes. Por ejemplo, si una computadora transmite un archivo grande a través de la red, este archivo es dividido en fragmentos, los cuales como se indicó en líneas anteriores reciben el nombre de paquetes, el número de paquetes puede ser muy grande. Los paquetes generados son enviados de la computadora fuente a la computadora destino en donde se ensambla el archivo original en base a todos los paquetes recibidos. El conjunto de protocolos de TCP/IP definen el formato de los paquetes que se forman, incluyendo el origen, destino, longitud y tipo de paquete, entre otros.

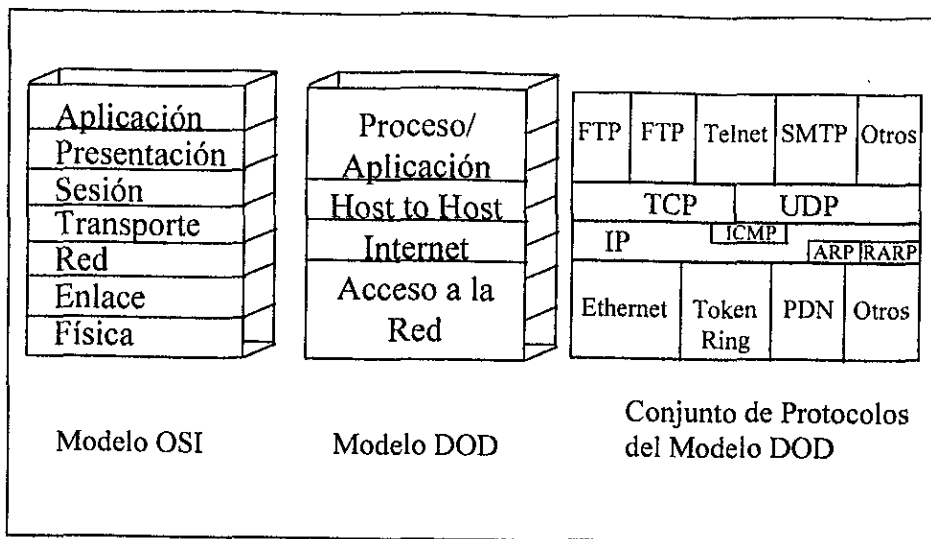


Figura 1.3.47. Protocolos de TCP/IP y el modelo DOD.

### Modelo del Departamento de Defensa (DOD)

Como ya se había mencionado TCP/IP consiste de cuatro capas de protocolos, estas capas se ven en el modelo del Departamento de Defensa, y se encuentra formado de la siguiente forma:

#### Capa de Proceso/Aplicación.

Este conjunto de protocolos forman la interface para el usuario, así que esta capa proporciona aplicaciones específicas para funcionar entre los dos hosts. Estas aplicaciones proporcionan transferencia de archivos (FTP), emulación de terminal (TELNET), correo electrónico (SMTP), administración de redes y otras aplicaciones más. A estos protocolos se les puede mapear con las tres capas superiores del modelo OSI (Aplicación, Presentación y Sesión).

### **Capa Host to Host.**

Esta capa establece y mantiene conexión simultánea entre dos hosts, y consiste de dos protocolos principales: Protocolo de Control de Transmisión (TCP, Transmission Control Protocol) y el Protocolo de Datagramas de Usuario (UDP, User Datagram Protocol). Esta capa se mapea hacia la capa de Transporte del modelo OSI.

### **Capa Internet.**

La función de esta capa es el direccionamiento de los paquetes entre diferentes hosts o redes. Los protocolos que se tienen en esta capa son: Protocolo Internet (IP- Internet Protocol), el Protocolo de Control de Mensajes Internet (ICMP- Internet Control Message Protocol), Protocolo de Resolución de Direcciones (ARP - Address Resolution Protocol) y el Protocolo de Resolución de Dirección Inversa (RARP - Reverse Address Resolution Protocol). Esta capa se mapea con la capa de red del modelo OSI.

### **Capa de Acceso de Red.**

Esta capa describe el tipo de acceso al medio físico de la red para comunicarse entre dos hosts. Los estándares que se tienen en esta capa son : Ethernet, Token-Ring y Token-Bus.

Las especificaciones de todo protocolo en el conjunto de protocolos de TCP/IP se encuentran definidos en uno o más RFC (Request for Comments), y son administrados por SRI International (Stanford Research Institute).

### **Procesos/Aplicación.**

Comenzaremos a analizar la capa de Procesos/Aplicaciones, y como se mencionó anteriormente se tiene la emulación de terminal, transferencia de archivos y correo electrónico entre otros. Estos programas normalmente requieren dos programas, uno de ellos en el cliente y otro en el



servidor. El programa que se tiene en el cliente se ejecuta por el usuario al momento de querer utilizar tales servicios. El software que se tiene en el servidor se ejecuta en el momento en el cual se da de alta el servidor, es decir, al momento de inicializar el servidor se ejecuta el programa, a este tipo de programas que se ejecutan se les da el nombre de demonios. Las aplicaciones que se tienen comúnmente son:

FTP (File Transfer Protocol) especificado en el RFC 959.

TFTP (Trivial File Transfer Protocol) especificado en el RFC 783.

TELNET (Virtual Terminal Protocol) especificado en el RFC 854.

SMTP (Simple Mail Transfer Protocol) especificado en el RFC 821.

## **FTP**

El FTP permite a un usuario transferir un archivo de una computadora a otra remota o viceversa. Este protocolo permite acceder la estructura de archivos remota como si fuese la local, permitiendo realizar cambios de directorio, y otras acciones comunes.

## **TFTP**

Este protocolo es similar a FTP, sólo que éste permite transferir un archivo entre dos sistemas sin especificar una cuenta o password. Las operaciones que permite realizar FTP con los archivos y directorios no son permitidas en este protocolo. El hecho de que en este protocolo no se usen cuentas y passwords, es que la mayoría de las implementaciones de este protocolo restringen el tipo de archivos que se pueden acceder.

## **TELNET**

Esta aplicación realiza una emulación de terminal para conectarse con un sistema remoto. Al realizar esta conexión y emulación de terminal remota se realiza cualquier operación sobre la cuenta a la cual se entró, esto como si se estuviese en la computadora local.

## SMTP

Esta aplicación permite comunicarse con otros usuarios en redes diferentes, es decir, que con esta aplicación se pueden enviar y recibir mensajes de correo, como si se estuviese enviando cartas de correo pero con una velocidad casi instantánea.

## HOST to HOST

La capa host to host es la equivalente a la capa de transporte en el modelo OSI, por lo tanto, como ya se trató en un capítulo anterior, esta capa se encarga de mantener la integridad de datos, y mantiene la comunicación que existe entre los dos sistemas. Al realizar una transmisión de información desde un host a otro, esta capa se asegura de que la información en el destino sea la misma que se envió en la fuente, que el orden de los datos sea el correcto, que no exista duplicación de información y que la información se encuentre libre de errores. En esta capa se encuentran principalmente dos protocolos, que se usan en la comunicación de TCP/IP. Estos protocolos son: TCP y UDP.

## TCP

El protocolo TCP proporciona una comunicación segura entre dos sistemas de cómputo, de los objetivos de este protocolo está la secuenciación de los paquetes enviados del fuente al destino. Al recibir la información, el destino envía un mensaje de reconocimiento del paquete, esto con la finalidad de que el fuente se entere de la recepción de la información enviada. Otra característica de este protocolo es el establecimiento de una comunicación como circuito virtual, asegurándose de la comunicación correcta de la fuente con el destino.

La utilización de este protocolo por los servicios (protocolos) de otras capas depende de la importancia que representa la seguridad en la comunicación. Cuando se desea utilizar este protocolo en la comunicación, es por que se desea una total seguridad en la recepción de la información, esto se hace mediante el envío de un mensaje para reconocer la recepción de la

información por parte del destino. Esta operación se realiza por cada paquete que se recibe en el destino, por tal razón el uso de este protocolo representa un mayor uso del medio de comunicación, y por consiguiente mayor congestión del mismo. Los protocolos de capas superiores que hacen uso de TCP son:

TELNET, FTP, SMTP y UDP.

El Protocolo UDP proporciona en contrapartida al TCP una comunicación sin aseguramiento de recepción de la información por parte del destino. Al realizarse una comunicación sin el mensaje de reconocimiento de recepción de la información, la carga en el medio de comunicación es menor, y como resultado se tiene que el medio no se encuentra sobrecargado en cuanto a su uso. Los protocolos superiores que hacen uso de UDP son:

TFTP (Trivial File Transfer Protocol).

NFS (Network File System).

Mensajes de Broadcast.

SNMP (Simple Network Management Protocol).

INTERNET

La capa que le sigue a host to host en orden descendente es la capa de internet, y esta capa tiene como objetivo principal la guía o dirección de la información entre los hosts que se desean comunicar. La comunicación de diferentes redes de área local forman una red de área vasta, y la liga que se realiza entre estas redes es en base a este protocolo. Anteriormente se indicó que la unión o liga entre dos o más redes de área local se realiza mediante un dispositivo llamado ruteador, que funciona bajo las capas 1, 2 y 3 del modelo OSI, o bien mediante otros dispositivos que trabajen sobre más capas.

## **IP**

El Protocolo Internet (IP) se encuentra en esta capa, y es el responsable del direccionamiento, fragmentación y ensamblaje de los paquetes para atravesar la red.

## **ICMP**

Se tiene al Protocolo de Control de Mensajes de Internet (ICMP) el cual permite a los ruteadores y a los hosts enviar mensajes de error y control a otros equipos de cómputo (ruteadores y hosts).

## **ARP**

El Protocolo de Resolución de Direcciones (ARP) se encarga de mapear la dirección lógica (software) con la dirección física (hardware) en la red.

## **RARP**

Por último el Protocolo de Resolución Inversa de Direcciones (RARP) determina una dirección lógica (software) a partir de una dirección física (hardware). Este protocolo se utiliza principalmente en workstations sin disco duro en el momento de inicializar el equipo.

## **Acceso al Medio.**

Por último, la capa más baja es la correspondiente a los protocolos que permiten enviar los mensajes por el medio de comunicación. La capa de acceso al medio corresponde a las capas de liga de datos y física del modelo OSI, y como se trató anteriormente estas capas se encargan de asegurar una liga libre de errores por el medio de comunicación y proporciona el medio físico para la transmisión de datos.

Los estándares utilizados en TCP/IP son : Ethernet o IEEE 802.3 (con los estándares; 10BASE5, 10BASE2, 10BASET y 1BASE5), el estándar IEEE 802.4 (Token-Bus) y la IEEE 802.5 o Token-Ring.

### **Formato del Frame Ethernet en TCP/IP**

Cuando se envía información de un host a otro en TCP/IP se manda desde las capas superiores a las inferiores en el fuente y de las inferiores a las superiores en el destino. De forma que en la capa de Procesos/Aplicación los datos se pasan a la capa de host to host, por ejemplo por el protocolo FTP, en la capa host to host es tomado por el protocolo correspondiente, ya sea por TCP o UDP, en este caso por TCP y le agrega a los datos un encabezado, donde este encabezado identifica a la capa superior de donde provienen estos datos. Otra información que contiene este encabezado es un número de secuencia y reconocimiento de la información, este conjunto de datos forma al mensaje que es enviado a la capa inferior (capa Internet), una vez que el mensaje es recibido por la capa de Internet, se forma un paquete al agregar un encabezado IP, donde este encabezado contiene la dirección lógica (software) del fuente y del destino, e identifica el protocolo de la capa host to host (la capa superior). Este paquete se pasa a la capa de acceso a la red, donde se agrega un encabezado MAC (Media Access Control), que contiene la dirección física (hardware) del host fuente y destino, al resultado de esto se le da el nombre de frame. Este frame contiene también un dato para checar una redundancia cíclica (su función es asegurar la integridad de los datos).

Al pasar el frame por el medio de comunicación hasta el host destino, se pasan por las mismas capas pero en orden inverso para llegar al usuario. En el destino se recibe el frame en la capa de acceso a la red, aquí se quita el encabezado MAC, y se pasa a la capa Internet, donde se le quita el encabezado IP y se pasa al protocolo correspondiente (TCP o UDP) de la capa host to host, una vez que se encuentra en esta capa se elimina el encabezado y se pasa a la capa superior, al protocolo que corresponde. Al pasar por cada capa se elimina el encabezado correspondiente, pero se realizan verificaciones utilizando la información recibida y el contenido de los encabezados correspondientes para asegurarse de que la información se encuentre correcta.

## **Direccionamiento.**

Como se explicó en unos párrafos anteriores la comunicación que se realiza entre dos hosts, se efectúa entre las capas correspondientes, es decir que la capa de Procesos/Aplicación en el host fuente se comunica con la capa Procesos/Aplicación en el host destino, la capa host to host en el fuente se comunica con la capa host to host en el destino, esto es para cada capa del modelo DOD (Department of Defense Model). La identificación que se realiza en cada capa se efectúa de acuerdo a los datos de la misma. En la capa Procesos/Aplicación se toma el nombre del host, en la capa Internet se toma la dirección IP, en la capa acceso al medio se toma la dirección física (hardware).

El nombre del host, como se indicó en el párrafo anterior se aplica en la capa de Procesos/Aplicación, la cual es la interface con el usuario. Al ser la interface con el usuario es necesario utilizar una notación que sea fácil de recordar, por tal motivo se proporciona un nombre a la computadora.

Para hacer uso de esta notación, es necesario tener en consideración que en la red Internet se tienen miles de computadoras, por tal motivo existe la posibilidad de que varias computadoras tengan el mismo nombre, pero se encuentran localizados en lugares geográficamente distintos; así que para evitar confusiones en la ubicación de la computadora se proporcionan dominios; y estos dominios pueden tener uno o varios subdominios; y estos a su vez otros más; dependiendo de las necesidades. Estos dominios tienen la función de organizar las redes que se tienen en todo el planeta. Por ejemplo se tienen unos dominios que la Internet proporcionó, y estos son:

### **MIL**

Utilizado para identificar al Departamento de Defensa de los Estados Unidos.

### **EDU**

Utilizado para identificar Universidades e Instituciones de educación.

**COM**

Se utiliza para corporaciones.

**NET**

Utilizado por organismos encargados de la administración de la Internet.

**MX**

Utilizado para identificar redes localizadas en México.

La dirección utilizada en la capa Internet es la IP, y esta dirección es obligatoria para todas las computadoras en la red, es una dirección lógica (software). Esta dirección identifica tanto la red en la cual se tiene conectado el host, como al mismo host, y es utilizada por los ruteadores para dirigir la información a través de las redes hasta llegar a su destino.

La dirección Internet (IP) de un host consiste de 4 bytes y contienen tanto la dirección de la red como la dirección del nodo. La dirección que se proporciona a todo host que se encuentre en la Internet es proporcionada por SRI International, o por un organismo al cual se le asignó una dirección de red para administrar. Esta dirección se encuentra escrita en una notación de puntos, esto es, cada dirección tiene como se indicó en líneas anteriores, cuatro bytes, estos bytes se tienen separados por un punto, por ejemplo: 144.22.1.193, la cual corresponde a mxpyr2.mx.oracle.com.

El Protocolo Internet soporta 5 clases de direcciones de Internet, estas son: clase A, B, C, D, E. Las clase D y E son especiales, de forma que sólo se utilizan en forma práctica las A, B y C.

**Clase A**

En la clase A el primer byte es la dirección de la red y los últimos tres bytes forman la dirección del host. El primer bit del primer byte debe ser cero, y en este primer byte sólo puede llegar hasta el número 128, por lo tanto se tiene como máximo 128 redes de clase A y cada una de ellas

puede tener hasta un total de 16, 777, 216 hosts. En la Internet todas las redes de clase A ya han sido asignadas.

### **Clase B**

En las redes de clase B, el primer byte se debe encontrar en el rango de 129 a 191. Los primeros dos bytes identifican a la red y los últimos dos bytes forman el host. El número de redes que se pueden tener en la clase B son 16,384 y el número de hosts es de 65,536.

### **Clase C**

En las redes de clase C, el primer byte se debe encontrar en el rango de 192 a 255. Los primeros tres bytes identifican a la red y el último byte identifica al host. El número de redes que se pueden tener en la clase C son 2,097,152 y el número de hosts es de 254.

### **Direcciones IP Reservadas.**

Existen unas direcciones IP que son reservadas para realizar algunas operaciones especiales, estas son:

#### **Dirección de Red.**

Es la parte de la dirección IP en que la porción del nodo o host están definidas a cero, como por ejemplo: 144.22.0.0, por el mismo uso que tiene esta dirección nunca es asignada a un host.

#### **Dirección de Broadcast.**

Esta es una dirección en la cual la porción del host esta definida a unos. Un paquete con dirección de destino de broadcast es enviada a todo nodo de la red, por tal motivo esta dirección no es asignada a un host o nodo.



## 1.4 CARACTERÍSTICAS, FUNCIONAMIENTO, VENTAJAS Y DESVENTAJAS DE LAS ARQUITECTURA CLIENTE/SERVIDOR DE TRES CAPAS

### Cliente/Servidor como un caso especial del Cómputo Distribuido

Un sistema de cómputo distribuido (DCS) es una colección de computadoras autónomas interconectadas a través de una comunicación de red para archivar funciones de negocio. Técnicamente hablando, las computadoras no comparten memoria principal por lo que la información no puede ser transferida mediante variables globales y la información es intercambiada sólo a través de mensajes vía la red, por ende las computadoras en este ambiente deben estar conectadas a través de la red que finalmente es la responsable del intercambio de información entre computadoras.

El cómputo distribuido puede ser realizado bajo los tres modelos siguientes:

- **Modelo de Transferencia de Archivos.-** Es uno de los más antiguos modelos del cómputo distribuido a un nivel mínimo. Básicamente, los programas de diferentes computadoras se comunican unos con otros mediante la transferencia de archivos, de hecho el correo electrónico es un caso especial de la transferencia de archivos. A pesar de que es un modelo viejo y extremadamente limitado se sigue usando en la actualidad para soportar computadoras distribuidas bajamente acopladas.
- **Modelo Cliente/Servidor.-** Este modelo permite que los procesos de la aplicación en diferentes centros de cómputo intercambien mensajes de manera interactiva, y es de esta manera que es un ambiente más transaccional que el modelo de transferencia de archivos. Inicialmente, el modelo Cliente/Servidor utilizaba llamadas a procedimientos remotos, el cual extiende el alcance de una simple llamada a un procedimiento local. Ahora, el modelo Cliente/Servidor

está siendo utilizado en el paradigma de objetos distribuidos, el cual extiende el alcance de un simple objeto local a una aplicación compuesta por objetos distribuidos.

- Modelo punto a punto.- permite que los procesos de diferentes centros de cómputo se invoquen unos a otros, la diferencia básica de un modelo Cliente/Servidor y uno punto a punto es que la interacción de los procesos puede ser de cliente, servidor o ambos, mientras que en el modelo Cliente/Servidor un proceso tiene el papel del proveedor de servicio, en tanto que el otro tiene el papel del consumidor de servicio.

### **Fundamentos del Modelo Cliente/Servidor**

El modelo Cliente/Servidor es un concepto que describe comunicaciones entre procesos de cómputo que son clasificadas como consumidores de servicios (Clientes) y proveedores de servicios (Servidores). De manera conceptual, clientes y servidores pueden correr sobre una misma máquina o en máquinas separadas, aunque hay un mayor interés en sistemas *Cliente/Servidor distribuido* que es cuando se encuentran en diferentes máquinas.

Las características del modelo Cliente/Servidor son las siguientes:

- Cliente y Servidor son modelos funcionales con interfaces bien definidas. Las funciones que desarrollan tanto cliente como servidor pueden ser implementadas por módulos de software, componentes de hardware o una combinación de ambos.
- La relación Cliente/Servidor es establecida entre dos módulos funcionales cuando un módulo inicia una petición (cliente) y la otra responde al servicio requerido (servidor).  
Un servidor puede transformarse en un cliente cuando éste requiere servicios de otro servidor.
- El intercambio de información entre clientes y servidores es estrictamente a través de mensajes, ninguna información es intercambiada mediante variables globales.

- El intercambio de mensajes entre clientes y servidores generalmente es interactivo, es decir, típicamente se manejan en tiempo real.
- Clientes y servidores residen en máquinas separadas conectadas a través de una red.

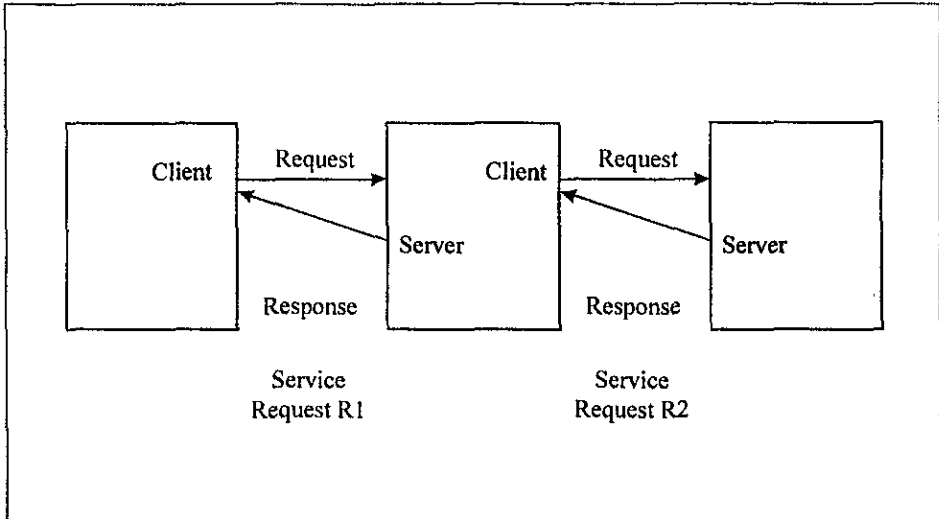


Figura 1.4.1 Modelo Conceptual Cliente/Servidor.

### Componentes del Modelo Cliente/Servidor

- **Middleware.-** Provee una amplia gama de funcionalidad, tales como, el establecimiento de sesiones entre los procesos clientes y procesos servidores, seguridad, compresión/descompresión y manejo de fallas. Se verán los dos lados del middleware cliente y servidor.
  - **Middleware del Cliente.-** Provee las interfaces entre procesos clientes y procesos servidores remotos, el middleware del cliente es esencialmente un conjunto de módulos de software, los cuales pueden ser invocados por procesos clientes a través de una interfaz programática API, por ejemplo, muchas bases de datos proveen su

API/SQL que es utilizado por los procesos clientes para poder enviar sentencias SQL al servidor donde residen el manejador de la base de datos y los datos en si mismos. Ejemplos de middleware del cliente son: **Sql\*Net** de Oracle, **ODBC** ( Conector abierto de base de datos ) de Microsoft, DCE RPC, etc.

Es deseable que el middleware del cliente provea un API estandarizado para que los procesos clientes pueden ser portables a través de múltiples plataformas, así mismo, debe establecer la conexión con el servidor mediante el uso de los servicios de red y el middleware del servidor y mandar la petición al middleware del servidor usando un protocolo comprensible para el servidor, por ejemplo, se puede hacer una petición en un formato Oracle si es usado Oracle del otro lado, sin embargo, en la actualidad los proveedores del middleware del cliente usan diferentes protocolos para no tener problemas de interoperabilidad. Además, el middleware del cliente debe recibir las respuestas del middleware del servidor, nuevamente, en un formato entendible para ambos, adicionalmente, debe manejar la sincronización de actividades, tener capacidades de manejo de fallas y manejar el control de acceso.

- **Middleware del Servidor.-** Monitorea las peticiones de los clientes e invoca procesos servidores apropiados, éste recibe las peticiones del cliente enviadas mediante los servicios de red y decodifica-*checa sintaxis del requerimiento*. Con el propósito de decodificar-*chechar sintaxis* el middleware del servidor debe entender el formato en el cual se le está enviando la sentencia SQL que fue enviada por el middleware del cliente. El protocolo de intercambio para llevar a cabo estas operaciones es la clave de la interoperabilidad del ambiente *Cliente/Servidor*, en muchos casos el protocolo es propietario y sólo podrán interoperar clientes y servidores del mismo proveedor. El middleware del servidor debe tener *mecanismos de seguridad* mediante la autenticación del cliente, también tiene un proceso de encolamiento de las peticiones del cliente, la mayoría del manejo de peticiones requieren el uso del servidor multihilos (multithreader) y cuando la concurrencia es alta se apoyan en monitores de transacciones, para así, tener un mejor manejo de las peticiones. Recibe respuesta de los procesos servidores y manda la respuesta a los clientes, así mismo, es responsable

del manejo de candados e intenta recuperaciones en caso de fallas de clientes, servidores, redes y cualquier otro componente.

- **Procesos Cliente.-** Realiza ciertas funciones de la aplicación del lado del cliente, éstos procesos pueden ser desde una simple interfaz de usuario hasta procesos más complejos, de manera muy significativa hoy en día se tiene que cada vez más los procesos clientes están basados en Internet, es decir, ellos utilizan navegadores para interactuar con los usuarios finales. Los procesos clientes tienen las siguientes características: interactúan con los usuarios a través de interfaces que son típicamente gráficas conocidas como GUI's u orientadas a objetos OOUI's que permiten invocar servicios mediante el uso de íconos; desarrollan funciones aplicativos si es necesario como manipulación de objetos, generación de reportes, etc.; interactúa con el middleware del cliente formando consultas y/o comandos de acuerdo al formato del API (Interface de aplicación programática) usado que es comprensible por el middleware del cliente; y recibe las respuestas del middleware del servidor y las despliega, si es necesario, sobre la interface de usuario.
- **Procesos Servidor.-** Realiza ciertas funciones de la aplicación del lado del servidor, los cuales tienen las siguientes características: Los procesos servidores pueden ser muy sencillos, como sólo dar la fecha del sistema, hasta desarrollar procesos sofisticados como el procesar una orden de compra, revisar inventarios, dar servicios de bases de datos, dar servicios de impresión, dar servicios de correo electrónico, etc.; idealmente, dan seguridad a la información interna y a la estructura de la base de datos, por ejemplo, el cliente no necesariamente debe conocer el tipo de dato que está siendo accedido, la ruta de acceso a los datos y, dependiendo del tipo de usuario, podrá ver o no ciertos datos; proveen de servicios de encolamiento de las peticiones de los clientes que están trabajando de manera concurrente; proveen también de servicios en caso de falla como recuperación; y en caso de aplicaciones transaccionales cuentan con capacidades de manejo de procesos transaccionales.

- **Servicios de Red.-** Ofrece el más bajo nivel de servicios, es decir, el transporte de la información en ambientes de cómputo distribuido y Cliente/Servidor. En este contexto, una red es una colección de equipamiento, módulos de software, y medios físicos; que interconectan dos o más computadoras, una alta variedad de configuraciones existen en éste rubro. Este punto se tocó en el inciso anterior de éste capítulo.
  
- **Servicios Locales.-** Ofrece acceso y manipulación de datos y procesos localizados en máquinas en un ambiente Cliente/Servidor, ejemplos de software local son:
  - **Manejadores de bases de datos.-** También conocidos como Sistemas Manejadores de Bases de datos (**DBMSs**) que dan acceso a bases de datos para usuarios en línea o batch. En un ambiente típico de bases de datos los usuarios pueden ver, acceder y manipular datos en una base de datos. Esta diseñado para: manejar vistas lógicas de los datos para que diferentes usuarios puedan acceder y manipular los datos sin conocer las representaciones físicas de los datos que están almacenados; manejar el acceso concurrente a los datos, forzando al manejo del aislamiento de las transacciones; manejar la seguridad de la información mediante diferentes privilegios otorgados a los usuarios. Los sistemas antiguos de información utilizaban bases de datos jerárquicas como IMS, pero en la actualidad típicamente son usados los manejadores de bases de datos relacionales como son: Oracle, DB2, Sybase, Informix, Progress, Ingres, etc. Las bases de datos Orientadas a Objetos están emergiendo, sin embargo, tendrán que pasar por un período de maduración, el cual implica para muchas bases de datos comerciales pasar primero por un esquema objeto-relacional antes de ir a un modelo puramente de objetos.
  
  - **Manejadores de transacciones (TMs).-** También son conocidos por monitores de transacciones (**TP-Monitors**) monitorea la ejecución de transacciones (secuencia de sentencias que deben ser tratadas como una unidad). TM se especializa en el manejo de transacciones desde su inicio hasta su terminación ya sea planeada o no planeada, algunos TMs se integran con las bases de datos de manera estrecha para facilitar las

consultas a las bases de datos desde diferentes transacciones para el acceso o la actualización de datos, sin embargo, otros TMs se especializan sólo en el manejo de transacciones como son Tuxedo, Encina y CICS.

- **Manejadores de archivos.**- Son responsables de dar acceso y manipulación a textos, diagramas, gráficas, imágenes y archivos indexados. Es una problemática del ambiente Cliente/Servidor dar acceso a los archivos que están geográficamente dispersos y han habido desarrollos en este sentido desde los años 60s, en la actualidad un ejemplo de ello es el NFS (Servidor de Archivos de Red) de SUN.
- **Manejadores de impresión.**- Son los responsables de la impresión de documentos. En la actualidad casi todas las redes LAN proveen el acceso a servidores de impresiones sobre la red LAN.
- **Sistemas Operativos.**- Administran los recursos de la computadora, entre ellos están la memoria principal, el CPU, los discos, etc. Desde una perspectiva Cliente/Servidor el principal papel que juega un sistema operativo es encolar o agendar procesos que necesitan recursos tales como memoria principal, archivos, bases de datos e impresoras. En algunos sistemas operativos como es el caso de Unix , es relativamente fácil agendar nuevos requerimientos usando procesos que residen en memoria, que pueden estar activos permanentemente, o bien pueden habilitarse o deshabilitarse manualmente mediante comandos del sistema operativo. Sin embargo, en sistemas operativos como MVS el agendado de procesos es tarea del manejador de transacciones como IMS o CICS que están orientados a un ambiente de procesamiento tipo mainframe. Es conveniente hacer mención de dos servicios que son particularmente importantes desde la perspectiva Cliente/Servidor, uno de ellos es lo que se conoce como threads que provee unidades de concurrencia dentro de un programa y son usados para crear concurrencia y programas servidores manejados por eventos; el otro es la comunicación entre procesos (IPC) que es necesaria para el intercambio de información entre procesos independientes. Por ejemplo, Unix provee una amplia gama de servicios de IPC como son: pipes, streams, sockets y TLI.

- **Hardware.-** En un ambiente Cliente/Servidor los procesos clientes y servidores pueden residir en PC (Computadoras Personales), estaciones de trabajo, minis y mainframes. El hardware también ha tenido un gran desarrollo empezando a ser barato, rápido, pequeño y cada vez más confiable. El uso de equipos de cómputo con multiprocesamiento ha tenido un auge importante en este ambiente, el multiprocesamiento utiliza los CPUs disponibles en la máquina para maximizar el paralelismo de las operaciones, en este tipo de arquitecturas se puede mencionar SMP (procesadores múltiples simétricos), MPP (Procesadores Masivamente Paralelos) y Clusters (máquinas bajamente acopladas).

### **Arquitecturas para Aplicaciones**

La determinación de un apropiado nivel de distribución de los componentes de la aplicación es de fundamental importancia cuando se usa el modelo Cliente/Servidor, esta decisión impacta en la selección de la infraestructura adecuada, en el rendimiento del sistema y en las consideraciones de la implementación.

El paso principal en el establecimiento de la arquitectura de la aplicación es identificar los niveles aplicativos y los mensajes entre esos niveles. Los tres niveles lógicos de una aplicación -interface de usuario, procesamiento y datos- son también conocidos como capas lógicas, el siguiente paso es estructurar y hacer coincidir esos niveles en una configuración física Cliente/Servidor. Por ejemplo, el nivel de la aplicación puede ser colocado en una sola máquina, a lo que se le llama simple capa física; o en una máquina cliente y en una máquina servidor, a lo que se le llama dos capas físicas; o también en una máquina cliente, en una máquina servidor y en otra adicional a ellas, a lo que se le llama tres capas físicas.

En la mayoría de los ambientes corporativos se puede encontrar el uso de todas estas configuraciones, pues cada una de ellas resuelve necesidades específicas, por lo que en el mundo real podemos encontrar el uso de ellas con diferentes tipos de arquitecturas de hardware, con varios protocolos de comunicación y con diferentes proveedores de software.



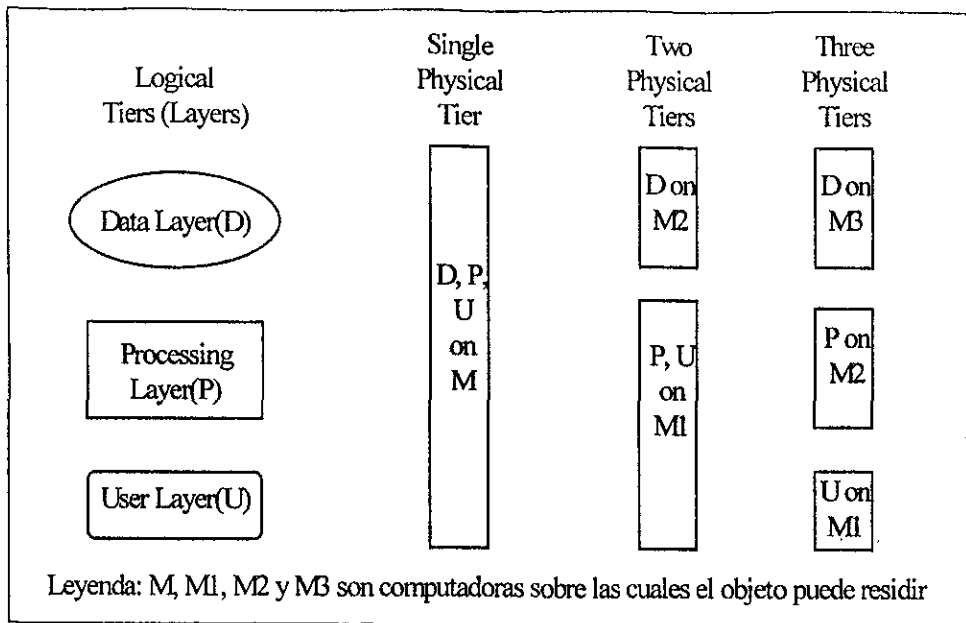


Figura 1.4.2 Capas de Lógicas a Físicas.

### Arquitectura de una Capa Física

La arquitectura basada en terminales (terminal-host) representa el modelo antiguo basado en mainframes, sin embargo, este modelo es bastante aceptado para aplicaciones de misión crítica, transaccionales (OLTP) y seguirá siendo una buena opción para aplicaciones que requieran un fuerte control central y un manejo de altos volúmenes de transacciones por segundo para cientos de usuarios.

“Las modernas NCs (computadoras de red) potencialmente hacen el modelo de una capa más popular puesto que esas computadoras son un poco más que terminales programables que necesitan de un respaldo del servidor”. (Umar, Amjad, 1997, pag. 66)<sup>1</sup>

<sup>1</sup> Umar, Amjad. 1997. Client/Server Internet Environments: Object-Oriented. Ed. Prentice Hall PTR, NJ USA

## Arquitectura de Dos Capas Físicas

En la arquitectura de dos capas en donde se cuenta con una máquina cliente y una máquina servidor, se pueden dar una serie de combinaciones que dan como resultado varias arquitecturas Cliente/Servidor de dos capas: la primera de ellas es la configuración basada en lo que se podría llamar presentación remota, los datos están residiendo en el servidor, además la lógica de la aplicación también reside en el servidor, sólo la interfaz de usuario esta localizada en el cliente; una segunda configuración podría estar basada al igual que la primera en los datos y la programación están posicionados en el servidor, pero a diferencia de ella, se sitúa la interface de usuario en el cliente y también en el servidor; otra arquitectura más, es la que esta basada en programación distribuida, en la cual los programas que comprenden la aplicación son divididos una parte hacia el cliente y otra parte en el servidor, la totalidad de los programas se comunican a través de **RPCs** (Llamadas a procedimientos remotos), esta es muy usada para aplicaciones más robustas, como pueden ser de tipo corporativo, hay proveedores que soportan esta arquitectura como es el caso del manejador de base de datos y herramientas de Oracle; una más es la que la lógica de la aplicación reside totalmente del lado del cliente y los datos en el servidor, esta arquitectura es usada para aplicaciones de tipo departamental, es ampliamente soportada por proveedores tales como PowerBuilder y GuptaSQLWindows; y la última es la arquitectura que está basada en que además de que la lógica esté del lado del cliente también los datos están tanto del lado del cliente como del lado del servidor, el tener datos distribuidos lleva a tener cuidado en el manejo y administración de los datos.

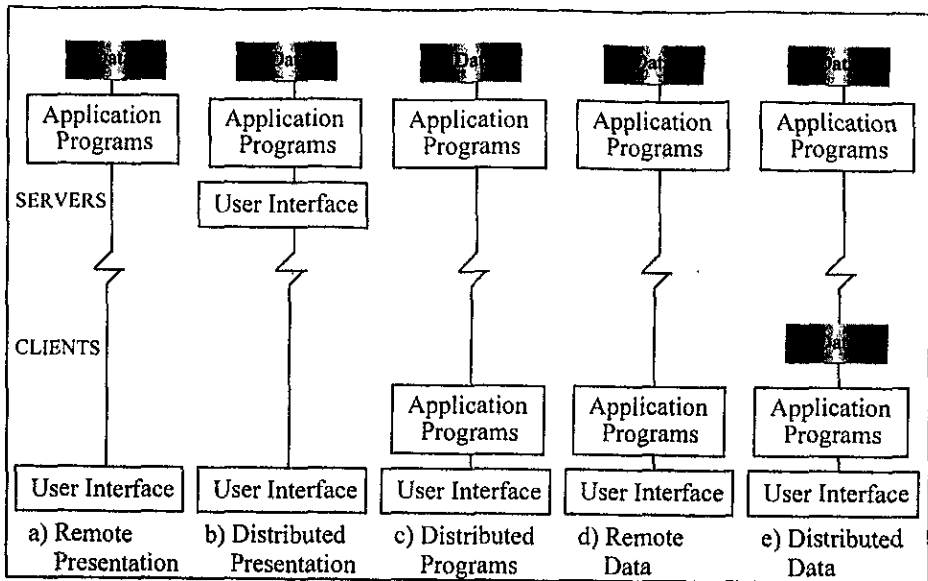


Figura 1.4.3 Arquitecturas Cliente/Servidor de dos capas.

### Arquitectura de Tres Capas Físicas

En un modelo Cliente/Servidor de tres capas, la aplicación Cliente/Servidor está repartida en tres tipos de máquinas: una máquina front-end, que puede ser una PC o NC, una máquina intermedia que puede ser una mini y una máquina back-end que puede ser incluso un mainframe, es importante que la interacción entre cada capa use un modelo Cliente/Servidor. Cuando sucede que la interacción entre un servidor y otro es a través de transferencia de archivos entonces estamos hablando de una arquitectura híbrida en vez de una arquitectura de tres capas Cliente/Servidor. En la figura que abajo se muestra, se pueden observar las diferentes combinaciones de cómo acomodar las tres capas lógicas ( front-end, middle y back-end ) en tres capas físicas.

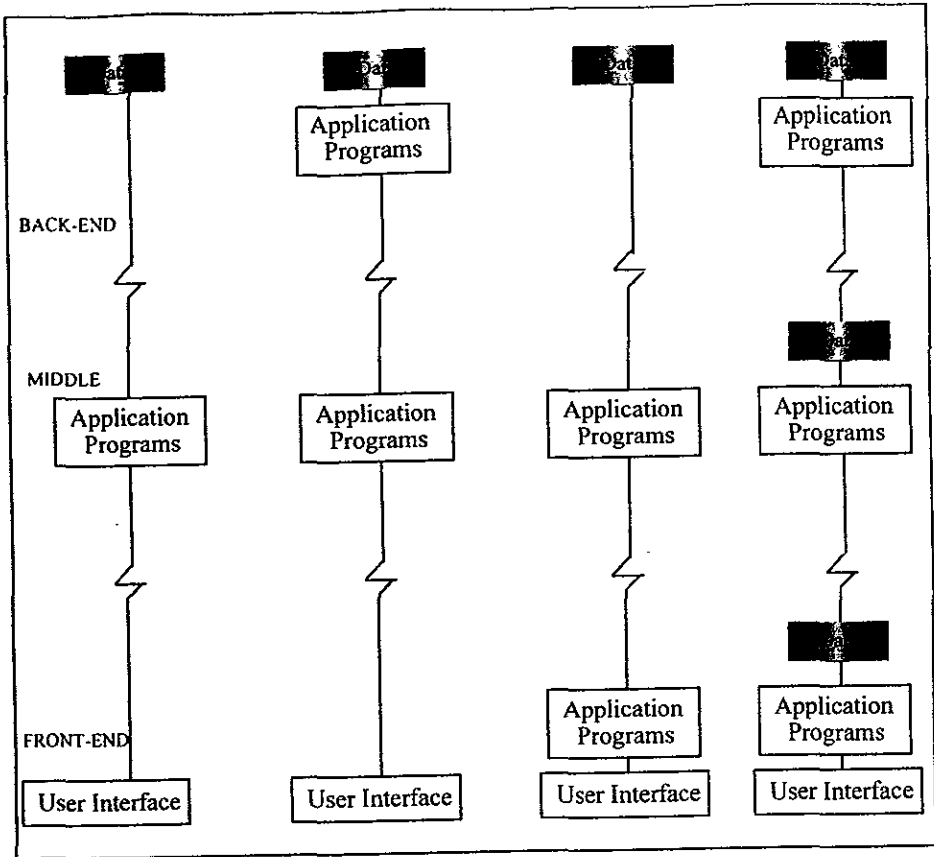


Figura 1.4.4 Arquitecturas Cliente/Servidor de tres capas.

Una aplicación puede ser configurada como una, dos, tres o arquitectura mixta de capas físicas, lo que se quiere mostrar son las diferencias entre las arquitecturas en términos de tipo de aplicación, flexibilidad, independencia de usuarios, rendimiento, disponibilidad, costo inicial, costo de actualización, manejabilidad y seguridad. Se presenta a continuación una tabla hecha por Umar, Amjad en su libro "Client/Server Internet Enviroments" en donde se pueden ver sus diferencias.

	<b>UNA CAPA O SISTEMA CENTRALIZADO</b>	<b>DOS CAPAS FÍSICAS</b>	<b>TRES CAPAS FÍSICAS</b>
<b>TIPO DE APLICACIÓN</b>	Aplicaciones de Tipo Corporativas, Misión Crítica, OLTP	Soporte de decisiones departamentales	Aplicaciones Corporativas
<b>FLEXIBILIDAD Y CRECIMIENTO</b>	No Bueno	Bueno	Muy Bueno
<b>INDEPENDENCIA DE USUARIO</b>	No Bueno	Bueno	Muy Bueno
<b>RENDIMIENTO</b>	Congestión en el Servidor	Congestión en la red	Muchas opciones
<b>DISPONIBILIDAD</b>	No Bueno	Bueno	Muy Bueno
<b>COSTO INICIAL</b>	Bajo	Medio	Alto
<b>COSTO DE ACTUALIZACIÓN</b>	Alto	Medio	Bajo
<b>CONTROL Y MANEJO</b>	Muy Bueno	Bueno	No Bueno
<b>SEGURIDAD</b>	Muy Bueno	Bueno	No Bueno

Figura 1.4.5 Comparación entre diferentes capas.

## 1.5 DISEÑO, CARACTERÍSTICAS Y USO DE LA HERRAMIENTA CASE DE ORACLE.

El desarrollo de sistemas efectivos de información siempre ha representado un gran reto. Los desarrolladores de software hoy día hacen uso de mejores prácticas y metodologías de desarrollo de sistemas que les permiten utilizar diferentes ambientes de desarrollo y utilizar las nuevas tendencias tecnológicas. El objetivo principal de los desarrolladores es contribuir al éxito de sus usuarios. Estos usuarios pueden ser clientes de paquetes de software o usuarios de sistemas desarrollados en casa.

Los sistemas de información exitosos por lo tanto, tendrán características reconocibles como:

*Enfocados al negocio* - Los usuarios y administradores deben dirigir el desarrollo cuando el alcance es una reingeniería de procesos de la empresa, un plan estratégico de sistemas de información o un sistema departamental.

*Desarrollados rápidamente* - Un sistema no tiene valor mientras no esté en uso. A la luz de los constantes cambios tecnológicos y el ambiente del negocio, si un sistema no está disponible rápidamente, éste probablemente sea descartado.

*Flexibles* - Dado que los sistemas implementan necesidades reales de los usuarios, al ser implementados estarán cambiando constantemente tanto como las necesidades de los usuarios cambien.

*Confiables* - La calidad en los sistemas es requerida y es justo otra faceta que los usuarios necesitan. Significa que el sistema se comporte consistentemente y que se pueda predecir que tanto en una parte del sistema como en otra se tendrá el mismo comportamiento.

La tecnología CASE (Computer Aided Software Engineering) permite a los desarrolladores de software generar aplicaciones como las anteriormente descritas. Esta tecnología esta formada de dos partes, la metodología y las herramientas.

Mientras las herramientas automatizan el software, la metodología define los pasos del proceso, la entrada y salida para cada paso y el control de calidad.

Es útil para programadores, analistas e ingenieros de sistemas, así como para directivos y ejecutivos de negocios de todos niveles y para negocios de todos tamaños y estructuras. Proporciona herramientas para ayudar a los directores corporativos a planear y documentar su trabajo; a los analistas les da apoyo en el análisis estratégico, diseño de sistemas y en la documentación de su trabajo, además de que ayuda a hacer menos pesada la documentación haciéndola en paralelo a la construcción.

La filosofía CASE involucra el uso de computadora como una herramienta de desarrollo para construir modelos que describen el negocio, el ambiente del negocio y la planeación corporativa, así como, documentar el desarrollo de los sistemas de cómputo desde la planeación hasta la implementación.

Para el desarrollo de sistemas, la tecnología CASE de Oracle cuenta con la herramienta Designer/2000. Esta herramienta da soporte a los desarrolladores en el uso de una amplia variedad de enfoques de desarrollo de sistemas de información y permite encontrar el mejor enfoque de desarrollo para cada proyecto dado que un enfoque no necesariamente es el correcto para todos los proyectos. Una implantación rápida será siempre un objetivo a alcanzar, pero bajo diferentes circunstancias se llegan a tener diferentes puntos de inicio. Aun así, independientemente de cual sea el punto de inicio, existen elementos comunes para todos los proyectos:

*Repositorio compartido* - Para llevar acabo desarrollos rápidos y consistentes, los desarrolladores necesitan utilizar el trabajo de otros así como compartir definiciones comunes. Sin un repositorio

compartido, la reusabilidad y el compartir es imposible, la consistencia y la verificación de la calidad es mucho mas difícil y demanda mas tiempo.

*Trabajo en equipo* - Múltiples desarrolladores y múltiples equipos necesitan compartir el repositorio para:

- Permitir compartir cuando sea necesario.
- Proveer un ambiente seguro de trabajo.
- Permitir buscar en el repositorio el componente adecuado para volverlo a usar.
- Mantener la documentación adecuada y al día de los componentes para que el siguiente desarrollador que lo necesite.

Designer/2000 está agrupado en conjuntos de herramientas que reflejan las necesidades primarias de los diferentes tipos de usuarios, y está basado en los enfoques de desarrollo siguientes:

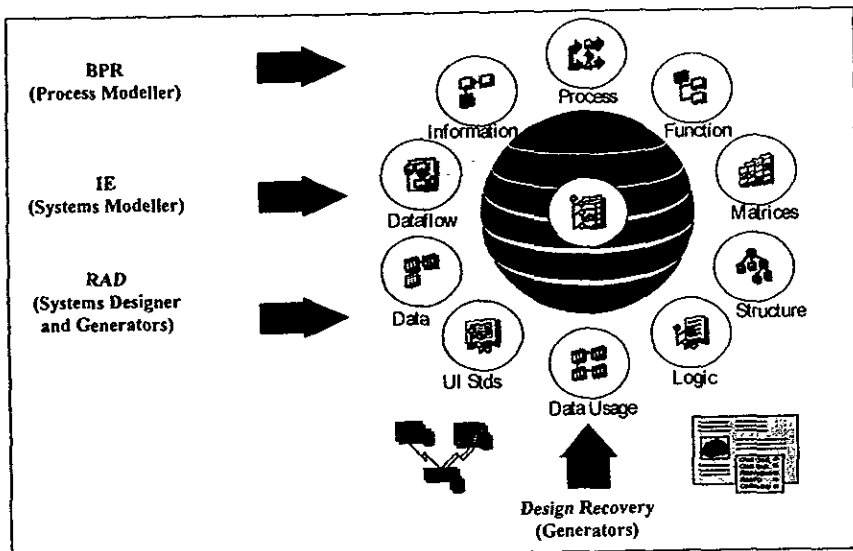


Figura 1.5.1. Enfoques de desarrollo que soporta Designer/2000.



- *Process Modeller* - Es el punto usual de inicio para el desarrollo con un enfoque de procesos. En este enfoque se comienza por mapear los procesos principales utilizando *Process Modeller* y entonces, mediante el refinamiento, se crea un modelo del negocio completo con *Systems Modeller* que en su momento definirá los proyectos de implementación que serán explotados por *Systems Designer* y *Generators*.
- *Systems Modeller* - Es el punto usual de inicio para el desarrollo con un enfoque de información. En este enfoque los proyectos basarán su alcance en un modelo de información del negocio y definirán su funcionalidad asociada mediante el uso de *Systems Modeller* antes de proceder a su implementación utilizando *Systems Designer* y *Generators*.
- *Systems Designer* - Es el punto usual de inicio para el desarrollo con un enfoque de usuario final. En este enfoque los proyectos pueden ser basados en rudimentarios modelos del negocio y así proceder rápidamente a su implementación. Se puede hacer caso omiso de un modelo completo del negocio utilizando solamente las herramientas de *Systems Designer* y *Generators*.
- *Systems Designer y Generators* - Incluyen las utilerías necesarias de recuperación del diseño para el desarrollo con un enfoque de sistemas heredados. En este enfoque los proyectos comenzarán por recuperar tanto como sea posible el diseño de los sistemas heredados usando las utilerías de ingeniería en reversa de los *Generators*. La raíz de esto es usualmente el diseño de la base de datos pero, la recuperación del diseño debería además, direccionar la funcionalidad del sistema. No existe una solución completa al 100% al problema de recuperación del diseño pero mucha información de gran valor puede ser recuperada con un relativo bajo esfuerzo.

Dependiendo del enfoque de desarrollo adoptado, el punto de inicio para un proyecto puede ser un conjunto diferente de herramientas en *Designer/2000*.

Los componentes de Designer/2000 han sido agrupados de esta forma para simplificar la instalación y para que cada desarrollador tenga a la mano las herramientas que necesite.

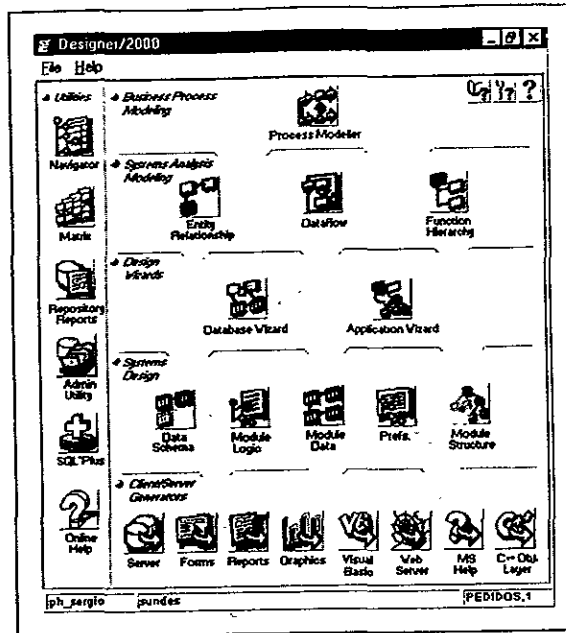


Figura 1.5.2. Pantalla principal de acceso a las herramientas de Designer/2000.

## Process Modeller



### Process Modeller Diagrammer

Process Modeller captura los pasos esenciales de un proceso del negocio, cómo éstos están ligados y quién es responsable de llevarlos a cabo. La herramienta utiliza anotaciones multimedia ( video clips, imágenes, sonidos, etc.) para realmente promover la participación del usuario en la creación y validación de los modelos.

Estos modelos pueden ser de importancia fundamental para el negocio. Representan la forma en que las organizaciones llevan a cabo su negocio, por lo que deben ser exactos y perfectamente entendidos.

Los objetos modelados son los mismos objetos que se manipulan en las herramientas de modelado funcional de Systems Modeller, esto es, funciones del negocio, flujos de datos, almacenamientos y unidades organizacionales, todos ellos reutilizables en el proceso de desarrollo.

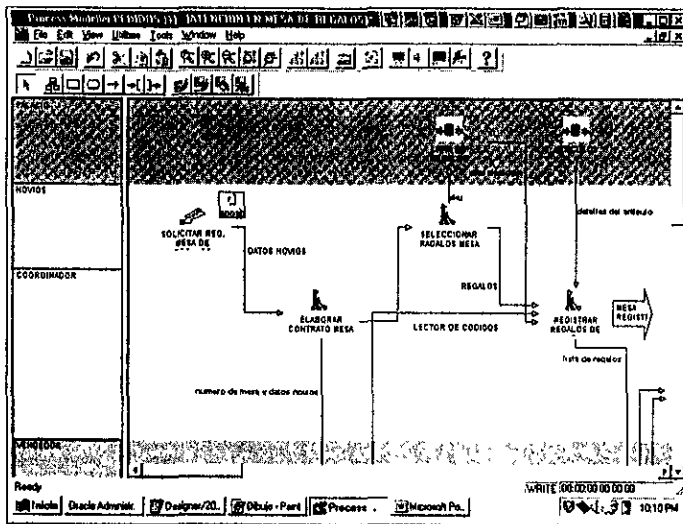


Figura 1.5.3. Process Modeller Diagrammer para modelar procesos.

## Systems Modeller

Systems Modeller provee las herramientas que son necesarias para especificar los requerimientos de los sistemas de información los cuales pueden ser escalables a un plan estratégico completo de sistemas de información. Incluye los siguientes diagramadores:



## Entity Relationship Diagrammer

Los sistemas basados en información no pueden ser exitosos sin una clara definición de qué información deben manejar y procesar. El modelado entidad relación es una forma para definir esto sin tomar en cuenta detalles de implementación.

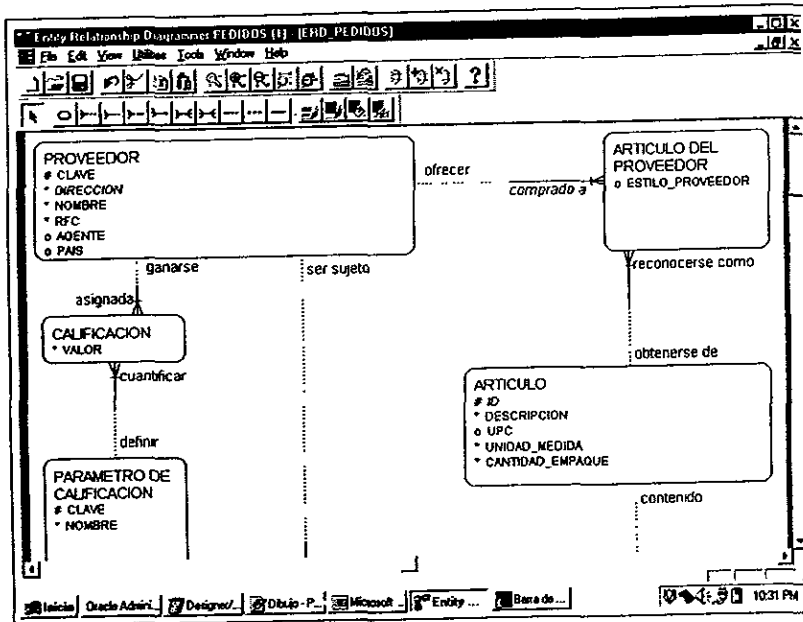


Figura 1.5.4. Entity/Relationship Diagrammer para modelar requerimientos de información.



## Function Hierarchy Diagrammer

Los requerimientos de información son solamente una parte de las necesidades. Una especificación del sistema debe además identificar qué es lo que se va a hacer con la información. Este diagrama muestra la descomposición de actividades dentro de una organización de una manera jerárquica. Actividades de gran escala pueden ser descompuestas recursivamente dentro de múltiples actividades de escala menor. En el nivel más fino de

mismo y por lo tanto, los componentes de Systems Designer están enfocados a producir estos entregables: definiciones de datos, reglas del negocio y definiciones de procedimientos, definiciones de las transacciones, definiciones de uso de datos, consultas, reportes y definiciones de navegación y flujos de trabajo. Incluye las siguientes herramientas:



#### Data Diagrammer

Muchos proyectos fracasan sin una vista sencilla y completa del diseño de la base de datos; los desarrolladores especifican de manera individual su propia versión de tablas o hacen deducciones inválidas acerca de la definición correcta de los datos; pruebas locales de la base de datos se convierten en pruebas inconsistentes y el trabajo del administrador de la base de datos se convierte en una tarea casi imposible de llevar a cabo. El Data Diagrammer proporciona un diseñador de la base de datos a través de un ambiente completamente gráfico para el diseño de un esquema de base de datos Oracle 7 o bien un esquema de base de datos ANSI (Instituto Nacional Americano de Estándares) estándar.

Adicionalmente, este diagramador, es una herramienta de uso fácil para los desarrolladores que les permite crear, compartir, particionar y reutilizar otros diseños de base de datos.

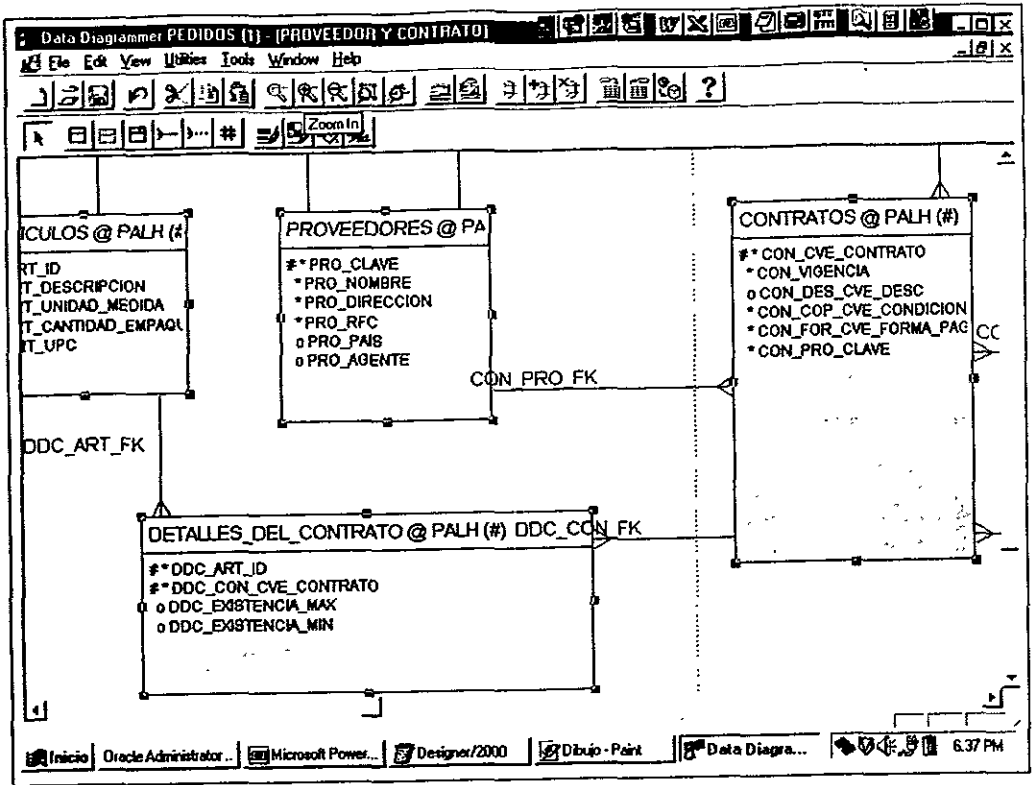


Figura 1.5.7. Data Diagrammer para visualizar gráficamente los objetos de Base de Datos.



### Module Structure Diagrammer

Un sistema de información es más que una base de datos y un conjunto de programas que la accesan y modifican. Dichos programas deben estar arreglados y ligados de manera tal que den soporte a las tareas de los usuarios. Los menús y los botones son implementaciones comunes de ligas y estructuras en aplicaciones GUI (interface de usuario gráfica). El Module Structure Diagrammer permite al diseñador de la aplicación definir la funcionalidad que el sistema presentará a los usuarios, cómo esta funcionalidad será presentada y cómo los flujos de trabajo serán implementados por las ligas.

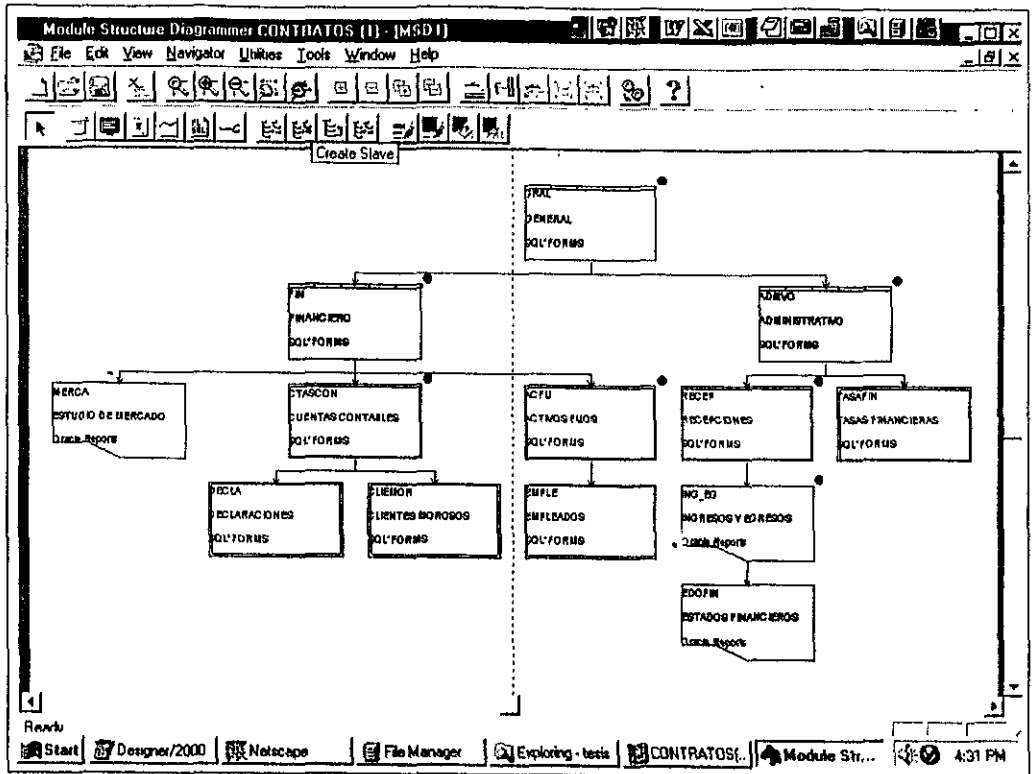


Figura 1.5.8. Module Structure Diagrammer para visualizar la estructura general de la aplicación.



Module Data Diagrammer

Cada una de las opciones en un menú de usuario o una barra de herramientas representa una pantalla de transacciones o un requerimiento de reporte. Así como el Module Structure Diagrammer especifica cómo estos programas embonan juntos, el Module Data Diagrammer especifica la estructura interna de un módulo individual o programa. Este diagramador especifica qué datos serán accedados, así como las tablas y columnas de la base de datos que se utilizan; especifica la forma de despliegue de los datos (text list, check box, etc.); adicionalmente, vía este

diagramador se especifica qué datos aparecen en qué ventana, cuáles estarán permanentemente visibles y cuáles serán de tipo pop-up.

El Module Data Diagrammer permite al desarrollador diseñar módulos de una manera gráfica, rápida y sencilla, dejándolos listos para que los Generators los transformen en aplicaciones. Dado el alto nivel de abstracción que el diagramador utiliza, el Module Data Diagrammer es una excelente herramienta para desarrollar prototipos. Mientras que con un lenguaje de programación el reestructurar una pantalla de dos ventanas “maestro detalle” a una sola ventana con múltiples bloques sincronizados puede ser una tarea muy compleja, con el Module Data Diagrammer esta puede ser una tarea muy sencilla.

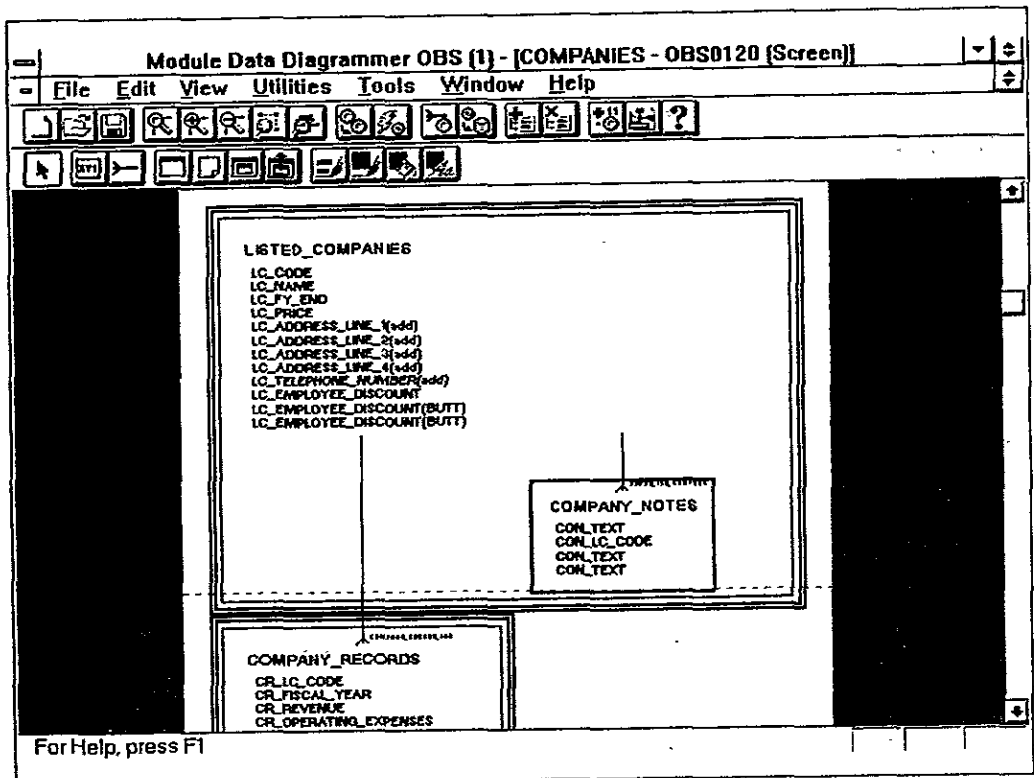


Figura 1.5.9. Module Data Diagrammer para definir gráficamente la funcionalidad de los módulos.





### Module Logic Navigator

En una aplicación del mundo real, no todos los programas están basados en pantallas, pueden llegar a existir reglas complejas del negocio que se pueden llegar a implementar como procedimientos. El Module Logic Navigator permite la creación y edición de paquetes, procedimientos, **triggers** (código que se ejecuta cuando un evento sucede) y funciones **PL/SQL** (Lenguaje Procedural / Lenguaje Estructurado de Consulta) para su posterior implementación vía el Server Generator.

El Module Logic Navigator permite a los desarrolladores hacer **cut and paste** (cortar y pegar) de los módulos existentes, codificar **PL/SQL** directamente o hacer **drag and drop** (seleccionar y depositar) de objetos **PL/SQL** desde un catálogo para crear procedimientos sintácticamente correctos. Este navegador, puede además hacer corrección de sintaxis en los módulos y permite verificar el uso de datos del mismo mediante una comparación contra las definiciones del esquema de base de datos almacenado en el repositorio de Designer/2000.



### Preferences Navigator

Los Generators hacen un uso extensivo de reglas para controlar la creación de código. Las reglas permiten definir por ejemplo, la cantidad de comentarios que se deben incluir en el código fuente o cómo se debe dar formato a las llamadas **SQL** embebidas en el código fuente. Estas preferencias, también controlan el **look and feel** (apariencia visual) de la aplicación; por ejemplo, cómo se deben dibujar los radio buttons, cómo manipular los mensajes de alerta, qué colores y **fonts** (fuente tipográfica) deben de utilizarse para las etiquetas de los campos, etc.

Las preferencias son, en efecto, una guía de estilo en línea, y como toda buena guía de estilo, las preferencias manejan una jerarquía de precedencia. Reglas diferentes pueden ser especificadas para diferentes proyectos, para módulos específicos o para aquellos módulos que hacen uso en

particular de una tabla o columna de la base de datos. El Preferences Navigator permite un acceso de fácil uso a una jerarquía de preferencias potencialmente muy grande y le da al desarrollador el poder de crear módulos con un look and feel sofisticado pero estándar.

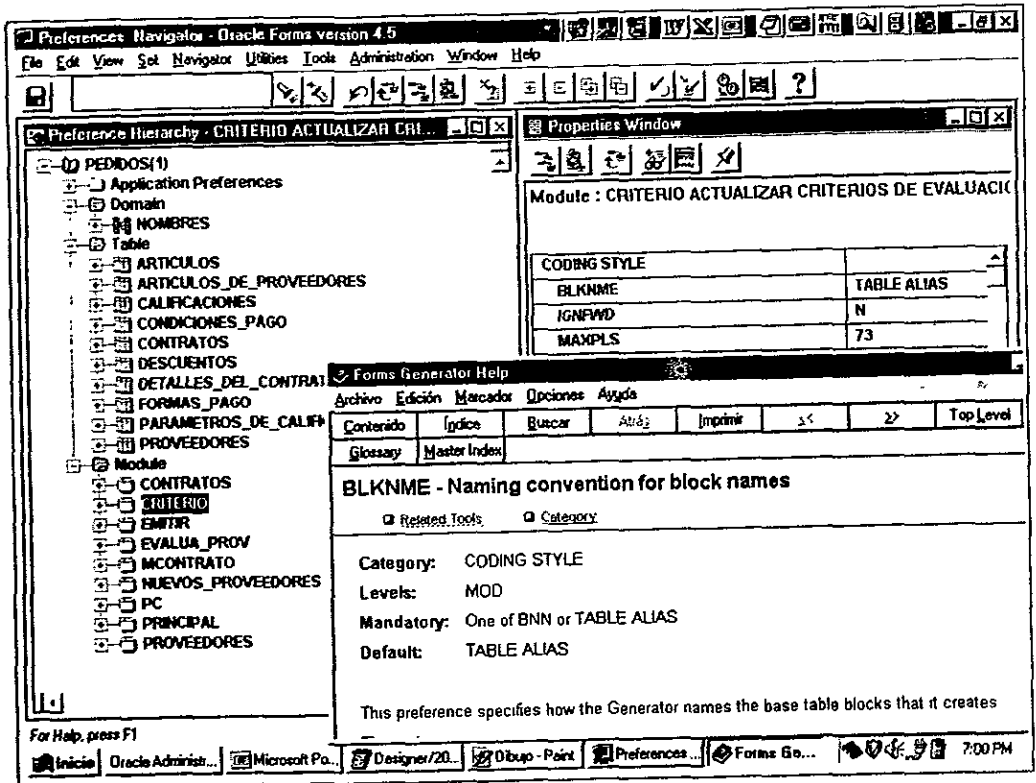


Figura 1.5.10. Preferences Navigator que muestra a manera de árbol las preferencias de generación.

## Design Wizards

Muchos desarrolladores usan Systems Designer como una herramienta *stand-alone* (aislada) para la creación de aplicaciones y bases de datos. Sin embargo, muchos otros utilizan Systems Modeller y Process Modeller para modelar el alcance de los requerimientos de un sistema previamente a su desarrollo. Otros, simplemente definirán el esqueleto de un modelo

probablemente creado en un laboratorio para dimensionar un proyecto de RAD (Desarrollo Rápido de Aplicaciones).

Cualquiera que sea el punto de inicio, los desarrolladores desean poder reutilizar tanto como sea posible la información de requerimientos especificada en el análisis y los Design Wizards les permiten hacer esto.



#### Database Design Wizard

Esta utilidad crea un diseño viable de base de datos relacional a partir del modelo entidad relación, deduciendo tablas, columnas, índices y llaves.



#### Application Design Wizard

Esta utilidad deduce que módulos programables serán necesarios para soportar las funciones en los modelos jerárquicos de funciones, de flujos de datos y de procesos, así como también, deduce el acceso a datos requerido por los módulos. Adicionalmente, deduce cómo los menús deberían ser estructurados para permitir un acceso apropiado a los módulos.

A partir de los modelos de requerimientos, los Design Wizards pueden crear un diseño del sistema apropiado reduciendo el consumo de tiempo y esfuerzo por parte de los diseñadores de la base de datos y la aplicación.

### Generators

Los Generators transforman los modelos almacenados en el Repositorio en código para el cliente y el servidor que en conjunto forman un sistema trabajando automáticamente. El proceso de generación Oracle está basado en un modelo radicalmente diferente que el de la mayoría de los

generadores de código. El modelo común para la generación de código es una especificación procedural dentro de un lenguaje de relativamente alto nivel transformado a código completamente detallado en un lenguaje procedural que se puede implementar, a diferencia de los Generators de Oracle trabajan a partir de especificaciones declarativas y reglas utilizando especificaciones procedurales solamente para reglas del negocio muy complejas.

De esta manera, el **layout** (distribución de campos y objetos), la mayoría de las validaciones, las reglas de referencia e integridad, la navegación y el 'look and feel' son creados a partir de reglas reutilizables especificadas en templates (machotes) , preferencias o defaults.

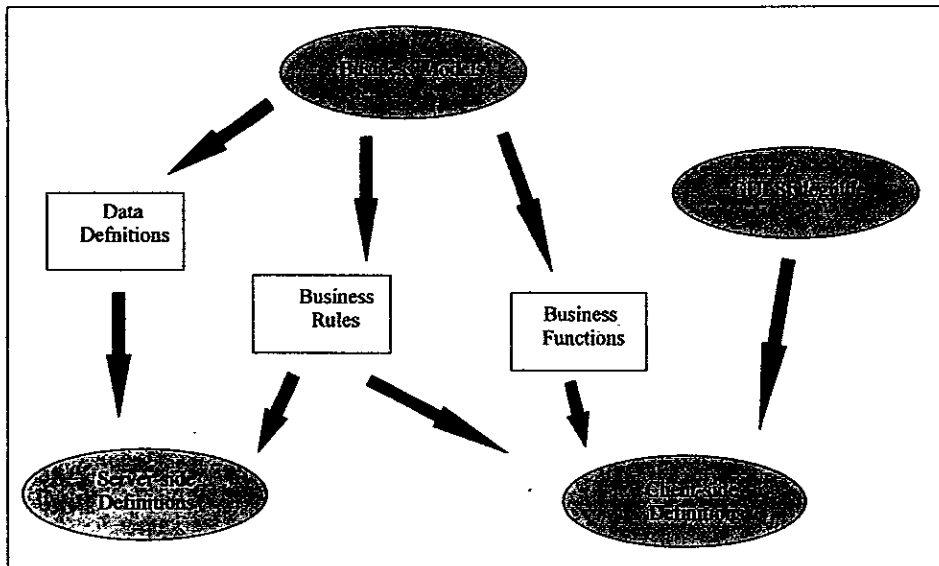


Figura 1.5.11. Esquema general de funcionamiento de los Generators.

Los generadores incluidos en Designer/2000 son:



## Forms Generator

El Forms Generator explota completamente las características de Oracle Forms 4.5. El generador utiliza el modelo de repositorio comprensivo para construir una definición interna completa de la forma que será creada. Dado que los desarrolladores cuentan con utilerías gráficas para especificar todos los aspectos de apariencia y entorno de las formas generadas, el uso extensivo de las reglas y defaults del generador implica que aunque el desarrollador no haya especificado todo, el entorno estándar del modulo será heredado.

Por ejemplo, si el desarrollador incluye el estado de la cuenta de un cliente en la forma, el generador primero buscara en el Detailed Column Usage (la especificación de uso detallado del campo) para identificar el tipo de despliegue, si no existe, entonces el de default es heredado, en este caso, podría ser un radio group. Después de esto, el generador consultará la guía de estilos del módulo (Preferences y template) para identificar cómo los radio groups son dibujados, podría ser horizontalmente en itálicas con **prompts** (etiqueta del campo) en color gris. El resultado es que se logra economizar en la especificación y se reutilizan los componentes existentes, las definiciones y las reglas, obteniendo así una alta productividad y un alto grado de estandarización.



## Reports Generator

El poderoso modelo de generación de Designer/2000 aplica de la misma manera para los reportes que para las formas. Los desarrolladores cuentan con utilerías análogas a las disponibles en Forms Generator para especificar reportes complejos rápidamente así como para hacerles cambios también de una manera muy rápida. Hacer prototipos con una serie de reportes que operan sobre el mismo conjunto de datos pero usando estructuras de consulta diferentes es simple. La creación de un reporte a partir de una pantalla equivalente es cuestión de unos cuantos pasos y viceversa. Esta habilidad de cada uno de los generadores de código en el cliente de

reinterpretar una especificación que en un principio fue desarrollada para otro generador, permite que se desarrollen prototipos de subsistemas los cuales pueden lograrse en una sola sesión, sin olvidar, que cada generador agrega su propio valor.

El Reports Generator permite agregar totales, cortes, un numero ilimitado de niveles, incluir **drill-downs** (navegación a un nivel de mayor detalle) e inclusive hacer reportes matriciales directamente de las definiciones contenidas en el repositorio. También es posible llevar a cabo desarrollos rápidos , implementar prototipos flexibles y obtener documentación completa a partir del repositorio.



### Server Generator

La generación del código del lado del cliente, es solo la mitad de la historia en lo que se refiere a la generación de sistemas. Gracias a la capacidad procedural de Oracle 7, mucha de la funcionalidad de un sistema puede ser almacenada en el servidor.

El Server Generator transforma las definiciones de procedimientos, paquetes, triggers y funciones que fueron creadas utilizando Module Logic Navigator , en código implementado PL/SQL.

La esencia de la generación del lado del servidor es la administración de los objetos de la base de datos: tablas, vistas, roles, shapshots (fotografías de la información), nodos y todos los otros objetos de Oracle 7. Estos son generados directa y automáticamente a partir de las definiciones almacenadas en el repositorio, las cuales pudieron ser dadas de alta vía el Data Diagrammer o vía el Database Design Wizard.

Adicionalmente, el Server Generator puede restringirse a un conjunto menos rico de objetos reconocidos por el estándar ANSI.



### Visual Basic Generator

Esta basado en las mismas definiciones del repositorio de Designer/2000 utilizadas por el Forms Generator y el Reports Generator. Este generador habilita la rápida construcción de aplicaciones Visual Basic totalmente funcionales y orientadas a datos, con layout automatizado y acceso a Oracle 7 vía **ODBC** (Conexión Abierta a la Base de Datos).

Mediante el Module Data Diagrammer, el Data Diagrammer y el Preferences Navigator se define cómo las aplicaciones de Visual Basic utilizaran los datos y como éstos serán desplegados. Con estas especificaciones, el Visual Basic Generator puede, entonces, crear los layouts de pantalla y el código de acceso a la base de datos, controlando características como la navegación y la sincronización de datos entre ventanas.

El generador utiliza templates y preferencias para fomentar un look and feel estándar en las aplicaciones generadas.

Cada aplicación Visual Basic generada comprende una o más ventanas, cada una de las cuales contiene una o más zonas en donde los datos pueden ser consultados o manipulados. Estas aplicaciones pueden ser cargadas dentro del ambiente de diseño de Visual Basic con el que se pueden construir los runtime ejecutables o bien se pueden afinar ciertas características del módulo adicionalmente.



### C++ Object Layer Generator

Este generador ayuda a los desarrolladores a crear aplicaciones con C++ que puedan manejar transparentemente datos almacenados en la base de datos. Esto se hace mediante la creación de clases C++ y las implementaciones basadas en un modelo de negocios definido en el repositorio de Designer/2000.

Las clases de C++ generadas se convierten en bloques de la aplicación, representando el tipo de objetos almacenados en la base de datos, además de proveer una interface semejante con la base de datos. Esto permite a los desarrolladores tratar dichos objetos de la misma forma que lo haría C++ y concentrarse en los detalles de la aplicación sin preocuparse por los mecanismos requeridos para hablar con la base de datos.



### Graphics Generator

Esta herramienta genera aplicaciones utilizando Oracle Graphics de Developer/2000 usando las definiciones de los módulos registrados en el repositorio de Designer/2000. Así mismo, permite que una amplia variedad de gráficas puedan ser generadas en pantalla y que éstas tengan la facilidad de hacer cortes o drill-down sobre los datos.

El generador incluye diversos templates para una fácil y rápida generación. Estos templates pueden ser modificados para controlar con más precisión las pantallas generadas.



### Microsoft Help Generator

Este generador permite crear las ayudas tipo MS-Windows utilizando la información definida en los módulos que se encuentran dentro del repositorio. La ayuda tipo Windows es una forma de proveer asistencia en línea para las aplicaciones basadas en esta plataforma desarrolladas y diseñadas utilizando Forms Generator o Visual Basic Generator.

La fuente principal para el generador de la ayuda tipo MS-Windows es el texto de ayuda, el cual está definido a detalle en base a los módulos, las tablas y las columnas utilizadas. Los archivos del generador de ayuda pueden ser usados en stand-alone o pueden ser integrados a las aplicaciones generadas.





## Web Server Generator

Este generador crea aplicaciones totalmente funcionales para la publicación de datos obtenidos de la base de datos Oracle a través de lo que se conoce como World Wide Web (WWW). Estas aplicaciones están basadas en las especificaciones de diseño de los módulos y de la base de datos registradas en el repositorio de Designer/2000.

La entrada principal para el proceso de generación es la especificación de diseño de un módulo registrado en el repositorio usando Module Data Diagrammer. Se registran las tablas y columnas usadas por el módulo, la relación entre ellas y la información detallada de cómo el módulo utiliza esta información.

Los módulos pueden ser relacionados entre ellos usando Module Structure Diagrammer. Dentro de las aplicaciones generadas, esta herramienta especifica la navegación permitida entre los diferentes módulos definidos.

Durante la generación, el Web Server Generator crea un grupo de paquetes de PL/SQL los cuales son ejecutados para instalar la aplicación de web en el Oracle Web Server. Las preferencias del usuario determinan el look and feel de la aplicación generada y puede ser personalizada para cualquier requerimiento en particular.

### **Repository Administrator**

Las herramientas de Process Modeller, Systems Modeller y Systems Designer proveen soporte completo al desarrollo de sistemas de información. Todas ellas son más útiles en un ambiente de trabajo en equipo en donde los desarrolladores están compartiendo las definiciones y reutilizando los componentes creados por sus colegas. En esta situación, cientos si no es que miles de objetos se llegan a definir en el repositorio. Sin una adecuada administración y control es imposible entender qué objetos existen, para qué se usan y quién necesita tener acceso a ellos. Repository

Administrator pone en manos de los desarrolladores y de los administradores del desarrollo las herramientas necesarias para realmente explotar y tomar ventaja del trabajo en equipo. Incluye las siguientes herramientas:



### Repository Object Navigator

El repositorio de Designer/2000 maneja muchos diferentes tipos de objetos y miles de instancias de estos pueden ser creados y usados en un solo proyecto de desarrollo. El Repository Object Navigator presenta esta gran cantidad de información de manera estructurada con una fácil navegación. Este navegador utiliza el paradigma de la jerarquía en el bien conocido formato de *Windows Explorer*. Por supuesto, en un repositorio no solo la jerarquía es importante (que columnas pertenecen a que tabla, por ejemplo), sino que también son importantes las relaciones que existen entre los diferentes tipos de objetos (qué módulos acceden qué columnas, por ejemplo). El Repository Object Navigator presenta las jerarquías y las relaciones de la misma forma sencilla, y hace una diferenciación de manera icónica.

Los desarrolladores pueden encontrar, ver y editar cualquier objeto del repositorio a través de esta herramienta así como aplicar cambios a conjuntos de objetos seleccionados.

El Repository Object Navigator permite a sus usuarios empezar la navegación a partir de cualquier objeto y a partir de ahí ir trazando sus asociaciones:

- de que tipo es (por ejemplo: es una función del negocio)
- que usa (por ejemplo: entidades)

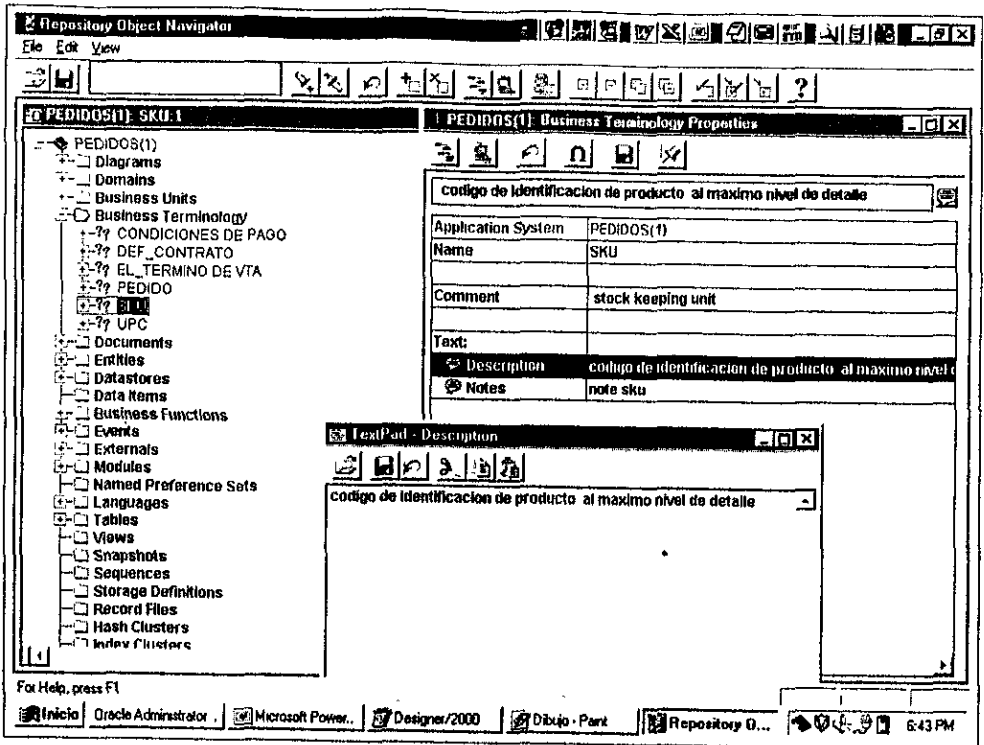


Figura 1.5.12. Repository Object Navigator para acceder los objetos definidos en el repositorio.



### Matrix Diagrammer

Este diagramador le da a los desarrolladores una forma diferente de ver el repositorio. Muestra todas las asociaciones de un tipo específico; por ejemplo, una matriz de todas las tablas y todos los módulos que las acceden. Se pueden establecer filtros y ordenamientos de los objetos incluidos.

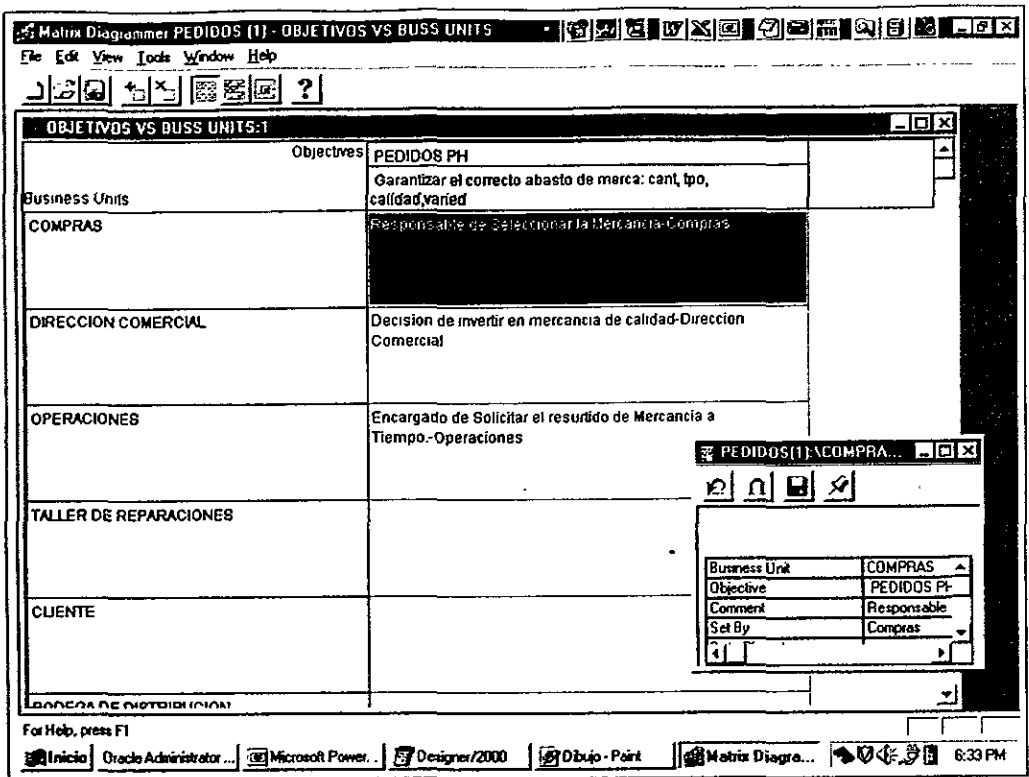


Figura 1.5.13. Matrix Diagrammer para establecer referencias cruzadas entre los objetos.



## Repository Reports

Esta herramienta permite el acceso a aproximadamente 150 reportes predefinidos que se pueden ejecutar; cada uno proporciona una vista diferente del contenido del repositorio.

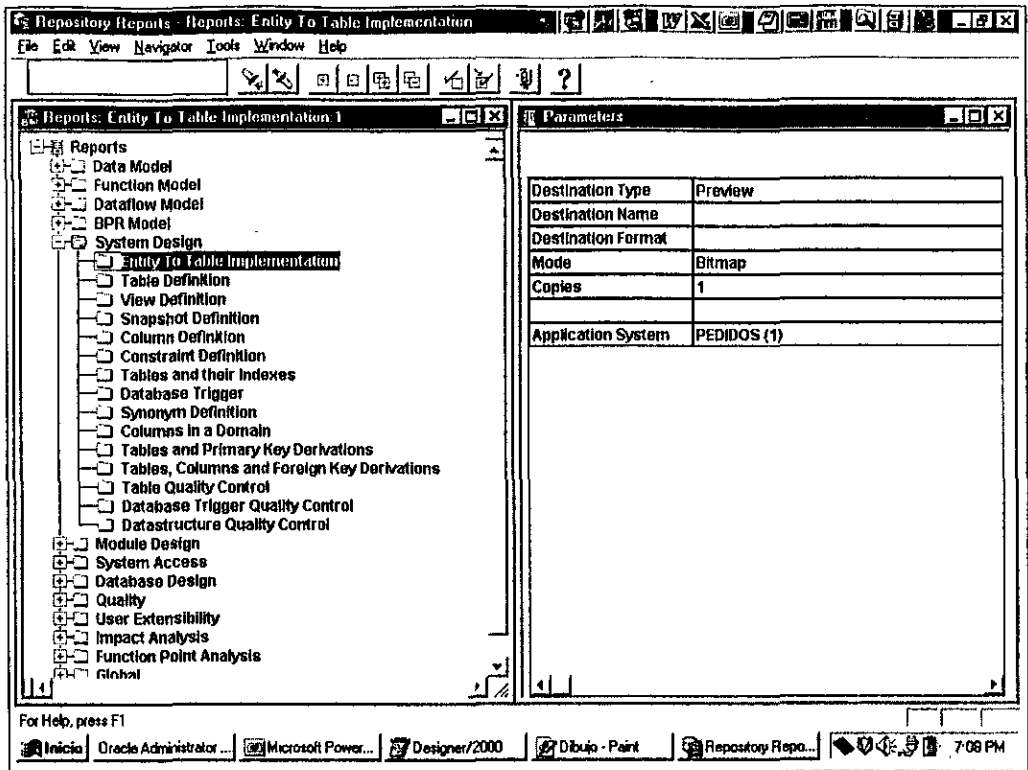


Figura 1.5.14. Repository Reports para generar reportes en general y a detalle de los objetos definidos en el repositorio.



### Repository Administrator Utility

Esta herramienta es una utilidad diseñada para los administradores del repositorio de Designer/2000, habilitándolos para su instalación y administración del ambiente del repositorio. Adicionalmente, vía esta utilidad, los administradores pueden extender la estructura de la definición del repositorio utilizando las características de **User Extensibility** (extensiones de usuario).

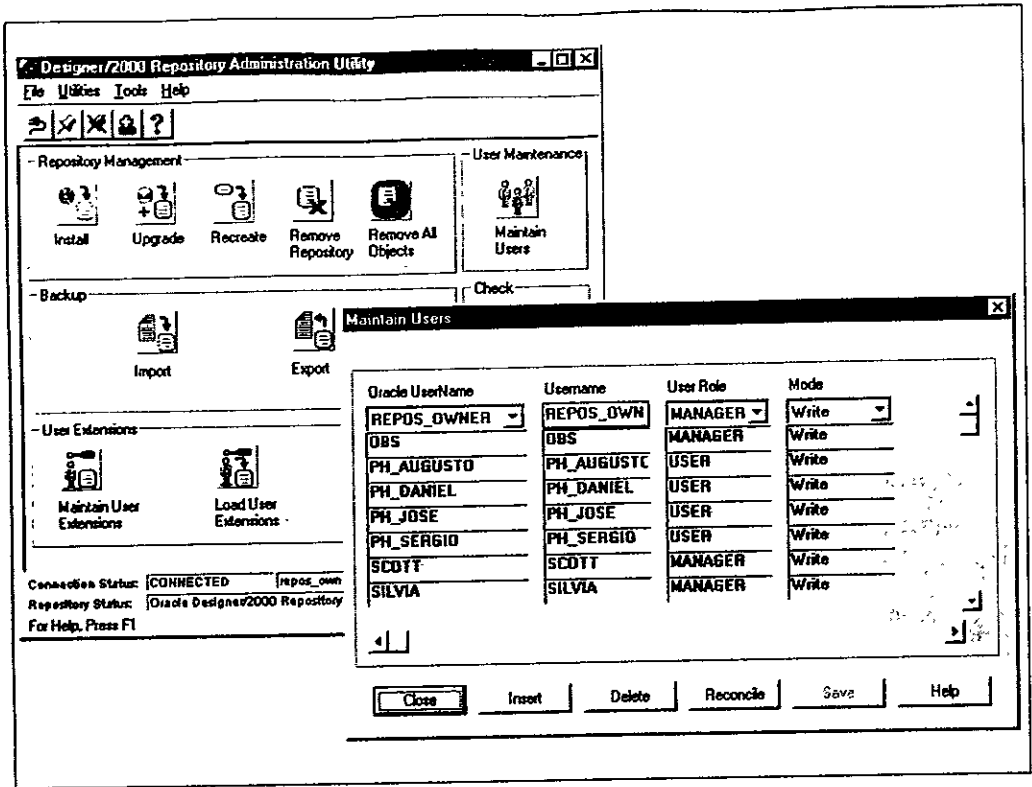


Figura 1.5.15. Repository Utility para administrar el acceso de usuarios de Designer/2000.

## Online Documentation System

Designer/2000 es un amplio conjunto de productos y cuenta con diversos componentes que son usados por diferentes tipos de usuarios en diferentes combinaciones. Designer/2000 cuenta con un sistema de documentación en línea extensivo en formato Microsoft Help para ayudar a los usuarios de todos los niveles de experiencia.

Una forma de acceder la documentación en línea es mediante la ayuda en línea sensible al contexto, la cual una vez que se accede dispone de ligas de hipertexto y botones de navegación para acceder otras secciones de la documentación.

Otra forma de acceder la documentación es a través del menú de ayuda de cada herramienta mediante el cual los usuarios pueden hacer búsquedas de tópicos en particular o palabras clave.

En términos generales, la documentación en línea esta dividida en tres grandes categorías:

*Introduction/Concepts*: Información que describe cómo funciona cada herramienta y bajo qué situaciones el usuario debería usar cada una de ellas.

*Tasks*: Instrucciones de cómo usar las herramientas.

*Reference*: Descripción detallada de cada campo en la pantalla (por ejemplo: opciones del menú, botones, paletas de botones, cajas de diálogo, etc.).

## **Resumen**

Designer/2000 es un nuevo tipo de ambiente de desarrollo. Trae a partir de su herencia las mejores características de las tecnologías existentes de desarrollo basadas en el repositorio dentro de su conjunto de herramientas Systems Modeller y va mas allá.

Con la introducción del soporte a la reingeniería de procesos del negocio incluida en Process Modeller, Designer/2000 ha respondido a los más fuertes desarrollos recientemente en la ciencia de la administración y el análisis de los sistemas del negocio.

Con la inclusión de herramientas de modelado gráfico de los sistemas incluidas en Systems Designer, Designer/2000 responde a la necesidad de brindar tecnología RAD para llevar a cabo desarrollos basados en un repositorio con las ventajas de poder trabajar en equipo y reutilizar componentes previamente especificados.

Mediante los Generators, Designer/2000, independientemente del punto de inicio elegido por el desarrollador, brinda:

*Productividad:* gracias a que se puede genera código a partir de modelos abstractos.

*Administración del proceso de desarrollo:* ya que no se requiere de procedimientos pesados y complejos para garantizar que las especificaciones almacenadas en el repositorio están completas y son de calidad.

*Mantenimiento fácil de los sistemas implementados:* dado que se obtiene documentación automática a partir del repositorio.

*Reutilización de los objetos definidos:* ya que el repositorio está centralizado y es compartido para el equipo de trabajo.

*Consistencia:* puesto que la interface gráfica de los sistemas generados está controlada mediante templates y preferencias de generación.



## 1.6 METODOLOGÍAS DE DISEÑO DE INGENIERÍA EN REVERSA.

### Introducción

El objetivo de la ingeniería en reversa es extraer información del diseño a partir del código programado. La ingeniería en reversa se realiza para ampliar el entendimiento de los programas durante el mantenimiento y para facilitar la reingeniería de software. Así pues, la ingeniería en reversa es el proceso de entender, analizar y abstraer los sistemas a una nueva forma en un nivel de abstracción mas alto.

Aparentemente, el entendimiento del software existente es una tarea trivial, pero la complejidad y dificultad implicada en el proceso no lo es tanto y ante la ausencia de cualquier documentación externa o guía, la tarea se convierte en algo prácticamente imposible. Esta es la razón por la que la ingeniería en reversa se ha desarrollado como una gran área de investigación en la ingeniería de software. La investigación en la ingeniería en reversa incluye, entre otras cosas, el desarrollo de metodologías de sondeo y la construcción de herramientas como sistemas de software utilizados como ayudas en la ingeniería en reversa.

Dependiendo de las necesidades de un proyecto en particular, se hace ingeniería en reversa de diferentes fuentes y se producen muchos tipos de vistas del diseño al aplicar diferentes heurísticas de ingeniería en reversa.

La tecnología de hacer ingeniería en reversa aplica también a compañías que están considerando convertir programas a otros lenguajes o base de datos o que están considerando la adopción de una herramienta CASE y que quieren tener los programas existentes bajo el control de dicha herramienta.

Los beneficios ofrecidos por la reingeniería de software versus los ofrecidos por el re-desarrollo de software a partir de los requerimientos originales son considerados como una mejor solución para el manejo del código existente heredado ya que mucha de la funcionalidad del software

existente se ha obtenido al paso del tiempo y debe ser preservada por muchas razones incluyendo la de proveer continuidad a los usuarios actuales del software.

### **Alcance y Tareas de la Ingeniería en Reversa.**

Dado que el propósito principal de la ingeniería en reversa es el de recuperar información a partir de un código existente o de cualquier otra documentación intermedia, cualquier actividad que requiera entender los programas a cualquier nivel puede fallar sin tener claro el alcance de la ingeniería en reversa. Lo que se obtiene como resultado de cualquier actividad de ingeniería en reversa varía de acuerdo a lo que se hace con la información extraída.

Las áreas en donde la ingeniería en reversa es aplicable (sin ser limitantes) son:

- Comprensión de programas
- Redocumentación o generación de documentación
- Enfoques de recuperación del diseño y detalles del diseño a cualquier nivel de abstracción
- Identificación de componentes reutilizables
- Identificación de componentes que necesitan ser reestructurados
- Recuperación de las reglas del negocio
- Entendimiento de la descripción de los sistemas a un nivel alto

Los procesos que tienen como finalidad rediseñar, reestructurar y ampliar funcionalidad a los sistemas no caen dentro del alcance de la ingeniería en reversa.

La ingeniería en reversa engloba un amplio arreglo de tareas relacionadas con el entendimiento y modificación de los sistemas de software. Estos arreglos de tareas pueden ser agrupados dentro de distintas clases. Algunas de estas clases son tratadas a continuación.

*Mapeo entre dominios de aplicación y programa* : Los programas de computadora son representaciones de situaciones problema de algún dominio de aplicación. Los programas,

usualmente no contienen ningún texto de ayuda que hable al respecto del problema. La tarea de la ingeniería en reversa es reconstruir el mapeo entre el dominio de la aplicación y el dominio del programa.

*Mapeo entre niveles concretos y abstractos:* Los procesos de desarrollo de software van de una abstracción de alto nivel a un mayor detalle en el diseño y llegan hasta una implementación concreta. Una ingeniería en reversa tiene que moverse hacia atrás y crear una representación abstracta de la implementación a partir de la mayor parte de los detalles concretos.

*Redescubrimiento de estructuras de alto nivel:* Un programa es la personificación de una estructura coherente de alto nivel con un propósito bien definido. Sin embargo, ese propósito y la estructura pueden perderse con el paso del tiempo a través de las actividades de mantenimiento como corrección de errores, cambios de plataforma, modificaciones y mejoras. Una de las tareas de la ingeniería en reversa es la de detectar el propósito y la estructura de alto nivel de un programa cuando éste pudo haber cambiado y donde, de hecho, podría ya no existir un propósito sencillo.

*Encontrar ligas perdidas entre la sintaxis de los programas y su semántica:* Los programas de computadora son formales, es decir, tienen una sintaxis y una semántica bien definida. Dentro del mundo formal, el significado de un programa sintácticamente correcto determina la salida para una entrada específica. Los sistemas que requieren ingeniería en reversa generalmente han perdido su semántica original. Más aún, ciertos lenguajes, como los lenguajes orientados a objetos, no tienen bases formales fuertes. El proceso de ingeniería en reversa podría determinar la semántica de un programa dado a partir de su sintaxis.

## **Análisis de Dominio**

La tecnología de ingeniería en reversa actual está típicamente basada en métodos de análisis de programas tales como el **parsing** (análisis sintáctico) y el análisis de flujo de datos. El enfoque de análisis de dominio ha sido considerado como una metodología alternativa. Un dominio es un

área problema y un análisis de dominio es un intento por identificar objetos importantes, operadores y relaciones dentro de un área problema o un dominio. El proceso de ingeniería en reversa proporciona un tremendo beneficio dado el conocimiento obtenido a través del análisis de dominio.

### **Método Jarzabek**

El autor Jarzabek<sup>1</sup> propone un enfoque manejado por modelos de dominio para la reingeniería de sistemas de negocio procedurales dentro de sistemas que tengan vistas orientadas a objetos. Jarzabek adopta un enfoque de manejo de datos en vez de un enfoque de manejo de procesos lo cual permite desarrollar las tareas mas fácilmente dado que el paradigma de orientación a objetos por si mismo es manejado por datos. Mas aun, los datos son mas permanentes que los procesos.

Este enfoque, además, toma en cuenta dos propiedades básicas del paradigma de orientación a objetos: las llamadas pre-condiciones y post-condiciones. El objetivo de este método es crear un repositorio de componentes de software reutilizables identificados y obtenidos a través del análisis de dominio, el cual, es desarrollado utilizando patrones de reconocimiento, particionamiento de programas y otras técnicas de análisis de programas estáticos.

Jarzabek resume su método como un proceso que consiste de diez pasos en el cual, los primeros ocho son esencialmente fases de ingeniería en reversa.

### **Método de Análisis de Consultas**

Jean-Marc Petit, Farouk Toumani y Jacques Kouloumdjian<sup>2</sup> han desarrollado un método de ingeniería en reversa para bases de datos relacionales operacionales. En este método, los

---

<sup>1</sup> Jarzabek, S. Domain Model-Driven Software Reengineering and Maintenance. Ed: Journal of System Software. USA. 1993

<sup>2</sup> Kouloumdjian, Jacques; Petit, Jean-Marc y Toumani, Farouk. Relational Database Reverse Engineering: A Method Based on Query Analysis. Ed: International Journal of Cooperative Information System. USA.

esquemas conceptuales son derivados utilizando información extraída de los diccionarios de datos, las extensiones de la base de datos y los programas de aplicación. La fortaleza principal del método descansa en las consideraciones hechas sobre el conocimiento a-priori disponible de la base de datos, las restricciones sobre los atributos y la competencia de los usuarios. Los autores argumentan que la mayoría del conocimiento necesario para construir el esquema conceptual, si no está descrito en el DDL (lenguaje de descripción de datos), está embebido en los programas de aplicación de varias formas.

Este método está basado en cuatro pasos principales:

Primer paso: Se desarrolla un análisis de los programas de aplicación y un conjunto de equi-joins es obtenido.

Segundo paso: Un esquema conceptual es derivado a partir del conjunto anteriormente obtenido, de las extensiones de la base de datos y del esquema relacional.

Tercer paso: El esquema conceptual anterior es validado a través de un diálogo interactivo con el usuario experto, quien es ayudado en esta tarea con indicaciones dadas por el método.

Cuarto paso: Se obtiene una reorganización del esquema bajo el control del usuario para un mejor empate de los requerimientos del usuario.

Este método permite que otro tipo de consultas puedan ayudar en la tarea del descubrimiento de la semántica y adicionalmente identifica fases donde la interacción del usuario es requerida.

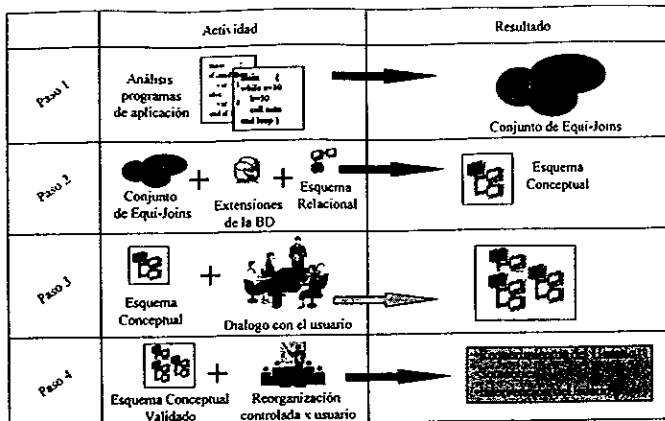


Figura 1.6.1. Pasos del Método de Análisis de Consultas.

### Método de dos Fases

Este método implica un enfoque de dos fases para la ingeniería en reversa que integra un proceso de abstracción formal de especificaciones a partir del código de un programa con una técnica para identificar objetos candidatos dentro del código del programa. Ver figura 1.6.2.

Una de las dificultades en la automatización de la abstracción de especificaciones formales a partir del código de un programa es que frecuentemente la especificación puede estar también limitada a la implementación. Este acoplamiento requiere de la interacción del usuario para poder obtener correctamente una especificación de alto nivel precisa que esté libre del sesgo de implementación.

Para tomar ventaja completa de las propiedades lógicas de las construcciones de programación, se puede representar a un alto nivel de abstracción una determinación precisa del propósito del programa, así como se comparó el código del programa. Las especificaciones formales construidas, junto con la información provista por un experto de dominio (alguien que tiene conocimiento acerca del dominio específico, detalles de implementación y requerimientos de

funcionalidad) facilitan la determinación de un programa correcto utilizando técnicas razonablemente automatizadas.

Este método incluye un conjunto de procedimientos para la abstracción de especificaciones formales a partir del código de un programa mediante la traducción básica de las construcciones de programación a representaciones equivalentes formales (lógica del predicado). Específicamente, los procedimientos manejan asignaciones, alternativas e iteraciones. Actualmente se ha direccionado la abstracción de especificaciones formales involucrando procedimientos. En todos los casos, las  $wp$  (pre-condiciones débiles) como las definidas por Dijkstra y Gries son usadas en el proceso de abstracción. Para todas las instrucciones de programación existe una transformación de predicado  $wp$  que es usada para definir la semántica de la instrucción con respecto a una post-condición  $R$ . Las  $wp$  es el conjunto de todos los estados dentro de los cuales una instrucción dada puede empezar la ejecución y cerca de la terminación, la post-condición  $R$  es verdadera. En el proceso de abstracción se inicia con la programación de la instrucción y se busca la post-condición  $R$  utilizando las definiciones  $wp$  para guiar la derivación.

Este método cuenta con una herramienta prototipo preliminar llamada Autospec que se ha comenzado a desarrollar para aplicar estos procedimientos al código del programa.

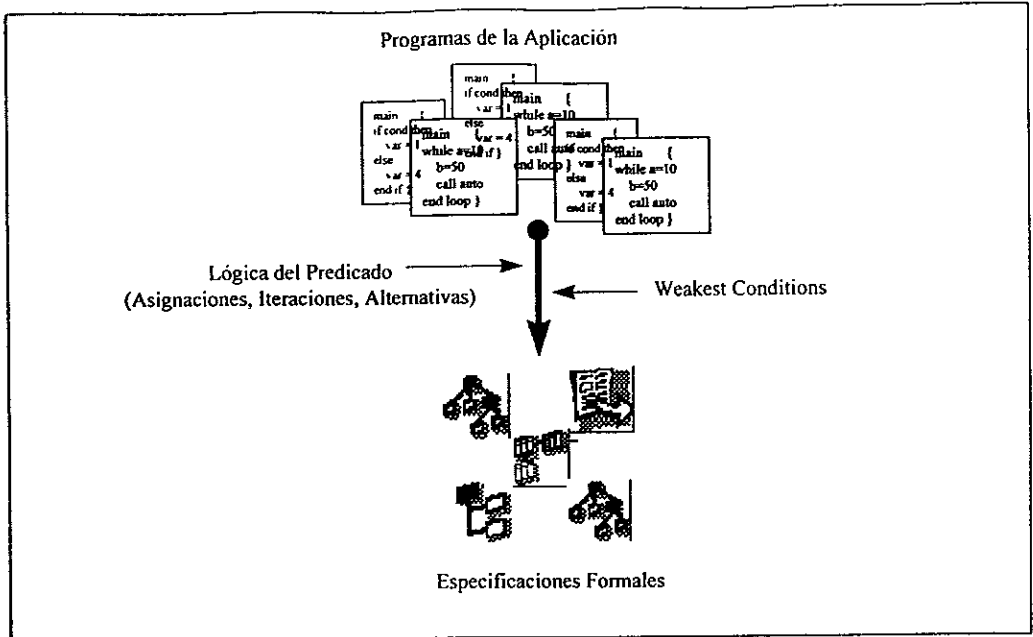


Figura 1.6.2. Procesos del Método de Dos Fases.

**Métricas de la Ingeniería en Reversa**

El autor Muller<sup>1</sup> describe un conjunto de métricas que guían a las herramientas de ingeniería en reversa en una composición de subsistemas. Se definen criterios de calidad del software y medidas para evaluar las estructuras de los subsistemas basados en qué tan fuerte es la interconexión entre los subsistemas y los clientes/proveedores comunes del mismo.

Estas medidas son usadas para calcular medidas normalizadas relacionadas a la calidad de la composición de un subsistema. La calidad en la encapsulación es una medida del ocultamiento de los datos y la cohesividad de un subsistema. La calidad en el particionamiento es el inverso de la medida de acoplamiento. La calidad normalizada de particionamiento se incrementa a medida que se tiene en los subsistemas interfaces pequeñas. La calidad de composición general

<sup>1</sup> Müller, H.A. Verifying Software Quality Criteria Using an Interactive Graph. Ed: Proc. 8<sup>th</sup> Annual Pacific Northwest Software Quality Conf. Portland, USA. 1990



de un sistema se incrementa en la medida que aumente la calidad en la encapsulación y la calidad en el particionamiento.

### **Herramientas de Ingeniería en Reversa incluidas en la herramienta CASE de Oracle Designer/2000**

Un proyecto de sistemas heredados o legados es aquel en donde el requerimiento es el de reemplazar el sistema heredado o agregar dicho sistema a uno existente. La primer tarea es la de recuperar en la medida de lo posible la mayor parte del diseño del sistema para poder entender cómo las mejoras deberían interoperar con él o cómo el sistema que lo reemplaza debería de funcionar.

Con la herramienta Designer/2000, un diseño documentado en el repositorio, inmediatamente pone al sistema bajo control. El análisis de impacto es automático. Fácilmente se pueden ver los efectos de los triggers sobre el servidor y se pueden identificar imperfecciones en el diseño mucho mas fácilmente cuando éste es visto desde un nivel de abstracción más alto con las herramientas de Designer/2000. Por ejemplo, los reportes que hacen esencialmente las mismas cosas se identifican rápidamente cuando se observa un reporte matricial de módulos contra tablas que los módulos acceden.

Existen diferentes enfoques de recuperación del diseño de sistemas legados mediante Designer/2000 y estos son:

*Oracle CASE* : Si el sistema fue construido utilizando el predecesor de Designer/2000 - Oracle CASE, el diseño del sistema estará accesible directamente. Al instalar Designer/2000 se tendrá actualizado el repositorio. Las mejoras o el reemplazo pueden empezar inmediatamente al afinar o rediseñar el sistema basándose en las nuevas herramientas disponibles en Designer/2000.

*Servidor Oracle y herramientas Developer/2000*: Designer/2000 incluye utilerías de ingeniería en reversa.

El Server Generator puede recapturar todo el diseño de un servidor Oracle 7, incluyendo la distribución de datos y los procedimientos. Aunque no se tengan planeadas subsecuentes mejoras, el servidor recapturado puede ser visto utilizando el Data Diagrammer, el Module Logic Navigator y el Repository Object Navigator. En si misma, la herramienta Designer/2000 es una valiosa herramienta de administración para los administradores de bases de datos.

Los generadores del cliente, pueden capturar las definiciones de los módulos del programa, así como qué datos usan y cómo los usan. Así, por ejemplo, un sistema que conste de decenas o cientos de módulos de programa hechos con Oracle Forms 4.5 y tablas de la base de datos, puede ser rápidamente capturado y los resultados pueden verse a través de:

**El Module Data Diagrammer:** Herramienta que mostrará qué tablas y columnas eran usadas para las operaciones de Query, Delete, Insert o Update.

**El Matrix Diagrammer:** Herramienta que mostrará todos los módulos y su uso de tablas en una sola vista, permitiendo a los desarrolladores identificar inconsistencias, duplicaciones y omisiones.

**El Module Structure Diagrammer:** Herramienta que permitirá al desarrollador modelar la nueva estructura de la aplicación. Una limitante de la herramienta es que no se puede hacer ingeniería en reversa de la estructura de la aplicación actual.

**Otro repositorio:** Si el diseño existe en otro repositorio, es posible utilizar Oracle Exchange para crear un mapeo del repositorio previo al repositorio de Designer/2000. Oracle Exchange cuenta con herramientas para definir el mapeo de un objeto de un repositorio a otro, por ejemplo, un proceso en un repositorio puede ser una función en el otro repositorio. También, Oracle Exchange puede ser usado para migrar un diseño a Designer/2000 y así tener un desarrollo mas fuerte; se puede usar como parte de una estrategia de coexistencia, en donde pueden emerger implementaciones alternativas cada una con su conjunto de herramientas de desarrollo preferido, a partir de un modelo de requerimientos de alto nivel. Se puede mantener un solo modelo

corporativo y migrar vía Oracle Exchange ciertos subconjuntos que se desee implementar en un conjunto diferente de herramientas de implementación.

*Otras fuentes:* A través del uso de la tecnología de gateways en el servidor Oracle y a través de pasos intermedios de scripts SQL, el Server Generator puede recapturar las definiciones básicas de la base de datos de cualquier manejador relacional de base de datos estándar ANSI. Los generadores del cliente pueden recapturar el uso de datos de cualquier fuente ASCII que contenga instrucciones SQL embebidas, de tal manera, que aunque la lógica de esos programas se pierda, no se perderá el control y la documentación estará siempre disponible.

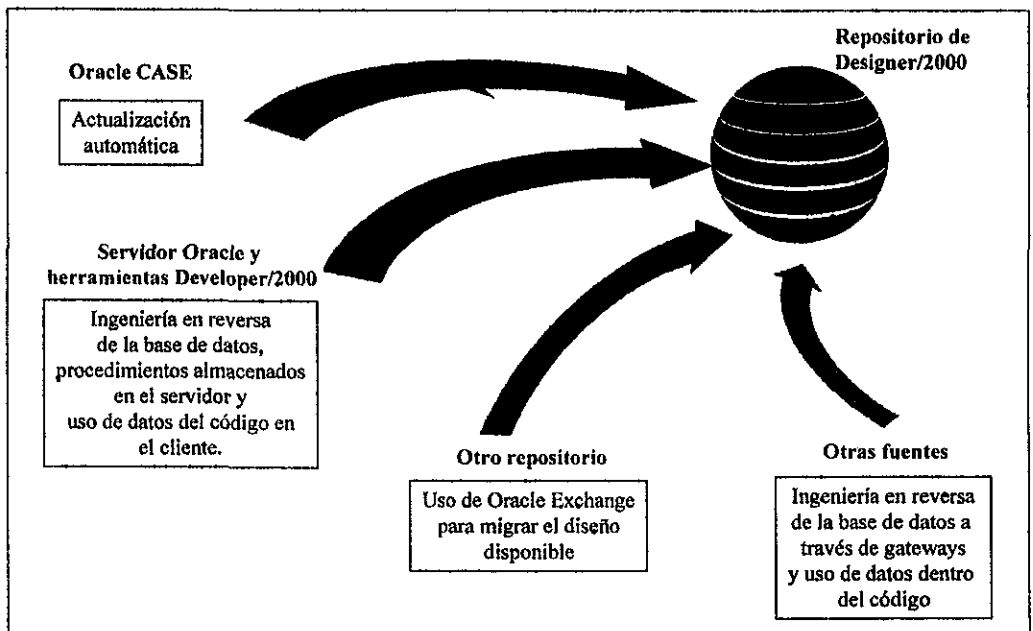


Figura 1.6.3. Enfoques de recuperación del diseño soportados en Designer/2000.

Independientemente de cual de los enfoques de recuperación del diseño anteriores se use, el punto final es que se contará con un modelo del diseño básico accesible a través del conjunto de herramientas incluidas en Systems Designer de Designer/2000. A partir de este modelo, los desarrolladores pueden elegir:

- Refinar el modelo y completarlo mediante una inspección manual o
- Involucrar a los usuarios finales y desplegarles la solución requerida a partir de un diseño básico mediante un desarrollo iterativo o un desarrollo RAD.

La primera opción generará un nuevo sistema con una funcionalidad muy similar al sistema original. La segunda opción generara una implementación del sistema con una funcionalidad mucho mas cercana a las necesidades actuales de los usuarios finales.

Es claro que ambas soluciones tienen un costo, pero también es claro, que al recuperar el diseño se protege la inversión de desarrollo que se tiene en los sistemas heredados.

### **Conclusión**

La ingeniería en reversa juega un papel muy significativo en lo que a mantenimiento de software se refiere. Buenas herramientas de ingeniería en reversa pueden hacer que la tarea de dar mantenimiento al software sea menos costosa y tenga menos problemas. Distintos estudios muestran que mas de dos terceras partes del dinero que se gasta en sistemas de software son destinadas a las actividades de mantenimiento. En este orden de ideas, el papel y la importancia que la ingeniería ha conseguido es muy alto. Si una herramienta de ingeniería en reversa puede reducir el costo de mantenimiento aunque sea en un margen muy pequeño, por decir un cinco por ciento, esto significa que se tendrían reducciones de gastos de billones de dólares a nivel mundial.

Dado que la ingeniería en reversa aplica a cualquier fase del desarrollo de un sistema o a cualquier nivel de abstracción, ésta puede ser empleada efectivamente en puntos estratégicos durante el proceso de desarrollo de software. Es mas, la ingeniería en reversa podría retroalimentar de manera importante a los desarrolladores y permitirles retroceder y rectificar errores y así, diseñar e implementar software mucho mas eficiente. Esto, tal vez, podría incrementar el costo del desarrollo, pero eventualmente, se disminuirían los gastos de mantenimiento.

Muchos enfoques de ingeniería en reversa son semiautomáticos por naturaleza. Las herramientas extraen la información, pero la gente tiene que ir guiando a las herramientas e ir tomando decisiones de cómo quieren ver la información. Por lo anterior, hoy día, se hacen muchos esfuerzos para desarrollar herramientas mas automatizadas, sin embargo, cabe mencionar, que se puede correr el riesgo de que éstas lleguen a fallar al trabajar con sistemas de software que no son estrictamente formales.

El software tiene gran influencia del proceso cognoscitivo humano y hasta ahora, solamente los humanos pueden entender los modelos mentales y las intuiciones que guían el desarrollo de un software en particular. Así pues, aunque no se requiera una herramienta particular para un sistema en particular, se debe prevenir que la intervención humana debe estar presente para guiar el proceso de ingeniería en reversa.

La importancia de la ingeniería en reversa es cada vez mas significativa y se está incrementando enormemente con el hecho de que en el mundo del software los cambios de paradigma se dan mas frecuentemente que antes. El tiempo que tomó la transición del paradigma de programación monolítica plana al paradigma de programación estructurada fue mas largo que el tiempo que tomó el cambiar de este último a un paradigma de orientación a objetos. Las herramientas de ingeniería en reversa son un mal necesario para hacer que la transición de un paradigma a otro sea mas rápida, fácil y menos costosa. Por ejemplo, a pesar de los probados beneficios que se tienen con el diseño orientado a objetos, las compañías y los usuarios del software no se están moviendo muy entusiastamente hacia él, esto se debe a que cuentan con un gran número de sistemas legados de software a los que se les tendría que hacer reingeniería hacia el nuevo modelo de objetos, lo cual representa una nueva inversión que no todas las compañías están en posibilidades de llevar a cabo inmediatamente.

## 1.7 FUNDAMENTOS DEL AMBIENTE DE TRABAJO, INTERNET, INTRANET Y WORLD WIDE WEB

### Antecedentes

El origen de la red de redes conocida como Internet es posible localizarlo en las postrimerías de la década de los 60's, particularmente hacia el año de 1969 en las oficinas del Gobierno Norteamericano, tiempos en los cuales emerge un proyecto denominado ARPANET (Agencia de la Red de Proyectos de Investigaciones Avanzadas). El principal objetivo de este proyecto consistía en brindar apoyos a las acciones y programas de las áreas militares del DOD (Departamento de la Defensa) precisamente desempeñando funciones que posibilitaran el hecho de que la información se pudiese compartir, al mismo tiempo en que fuese posible tener acceso a computadoras con ubicaciones remotas, no importando el lugar en dónde éstas se encontrarán. El modelo de red que en ese entonces se creó fue altamente distribuido para habilitar la facilidad de enrutamiento de comunicaciones en caso de ataques bélicos.

En aquellas épocas, que ahora parecen tan lejanas, el usuario de Internet tenía sólo una interface basada en texto, razón por la cual durante muchos años los científicos y programadores únicamente podían transferir archivos, así como mandar y recibir mensajes electrónicos.

En 1989 se registró un cambio sumamente importante para el desarrollo del ambiente Internet, ello tuvo que ver con la aparición del World Wide Web (WWW), conocido más comúnmente como Web. En este nuevo ambiente ya se permite el acceso, la navegación y lo que es más importante: la capacidad para poder compartir la información dispersa alrededor del mundo; esto es posible lograrlo a través de las GUI (Interfaces Gráficas de Clientes) denominadas Web Browser, mismas que también son conocidas como navegadores de Internet. Una de las características distintivas de los navegadores es que permiten que, de una manera transparente, se puedan ver diversas piezas de información localizadas en distintas computadoras en diferentes ciudades o países.

El propósito inicialmente asignado a WWW fue la creación de un sistema de hipertexto que permitiera brindar la posibilidad de compartir información geográficamente dispersa para uso fundamentalmente de grupos de investigadores. Las acciones destinadas a la consecución de este fin fue desarrollado por European Particle Physics Lab.

Hacia finales de 1990 fue desarrollado un navegador de internet llamado www, el mismo que logró implementar por vez primera los principios del hipertexto y que demostró capacidad para leer diferentes tipos de documentos. En 1992 fueron desarrollados nuevos navegadores y en 1993 hace su aparición un navegador sumamente conocido bajo la denominación de MOSAIC, el cual desempeña las funciones de un catalizador para la explotación del intercambio de información que hoy en día está ocurriendo, este navegador incluye una interface gráfica para el usuario que fue desarrollado por NCSA (National Center for Supercomputing Applications) de la Universidad de Illinois.

A partir de 1994 el Web ha tenido un notable auge, lo que se ha visto reflejado en el considerable incremento del número de navegadores, de ingenierías de búsquedas, servidores de Web y su uso.

Actualmente el mercado ha aceptado un producto comercial basado en el modelo de MOSAIC llamado Netscape, el cual es proveído por una firma comercial cofundada por Andreessen, esta nueva compañía está ofreciendo el soporte y la calidad necesaria que se espera de un software comercial. Y como dato adicional en agosto de 1995 Andreessen se estaba convirtiendo en un multimillonario.

Este ambiente de trabajo ha tenido un auge tremendo, hasta llegar a nuestros días en que las empresas o negocios están empezando a poner información en el Web de anuncios, publicidad y comercio electrónico.

## Internet e Intranet

Técnicamente hablando, Internet es una red basada en el protocolo TCP/IP. El término Internet es usado para referirse a una gran colección de redes TCP/IP estrechamente interconectadas a través de dispositivos tales como ruteadores y gateways.

Un término interesante para ser comentado es el de Cyber-espacio. Este término originalmente fue introducido por la ciencia ficción y ha llegado para quedarse en nuestro vocablo informático, sirve para representar a miles de computadoras y recursos de cómputo interconectados a lo largo del mundo vía Internet.

Es necesario dejar bien establecidas las diferencias de dos conceptos importantes :

- **Internet (Internet Pública):** Es precisamente lo que se describió con anterioridad, en realidad *no pertenece a ninguna entidad, organismo o institución. Consiste en miles de redes TCP/IP independientes pero enlazadas entre sí. Actualmente se han incorporado a esta red de redes, redes comerciales y privadas para enriquecer más el alcance de la misma.*

Las computadoras en este ambiente son conocidas mediante una dirección IP (Internet Protocol) el cual es único para cada máquina.

- **Intranet (Internet Privada):** De manera similar son redes TCP/IP pero el uso de ellas está vinculado únicamente a una corporación con el simple objetivo de ayudar al negocio o giro de la empresa, especialmente explotando la tecnología de Web, lo cual hace posible que cualquier aplicación que está en Internet pueda ser usada en Intranet y muchas compañías están usando esta tecnología para sus aplicaciones internas. En este concepto presumiblemente se tiene mayor control, son más seguras, más pequeñas y privadas.

Después de haber hecho esta diferencia de dos grandes conceptos se comentarán definiciones que son usadas en Internet pero que también son aplicables para Intranets.



Domain Naming Services (DNS) este concepto es usado para localizar diferentes recursos en donde se define una jerarquía de nombres que son fáciles de recordar mucho más que una dirección IP, además, define tipos de organización. Ejemplos de esto se tienen los siguientes:

- oracle.com = Compañía comercial llamada Oracle
- unam.edu = Institución educativa, Universidad Nacional Autónoma de México
- sun.com.mx = Compañía comercial llamada Sun que está en México
- omg.org = Organismo independiente

Todos estos nombres a su vez se traducen en una dirección IP . Ejemplo de una dirección IP podría ser 135.24.6.78

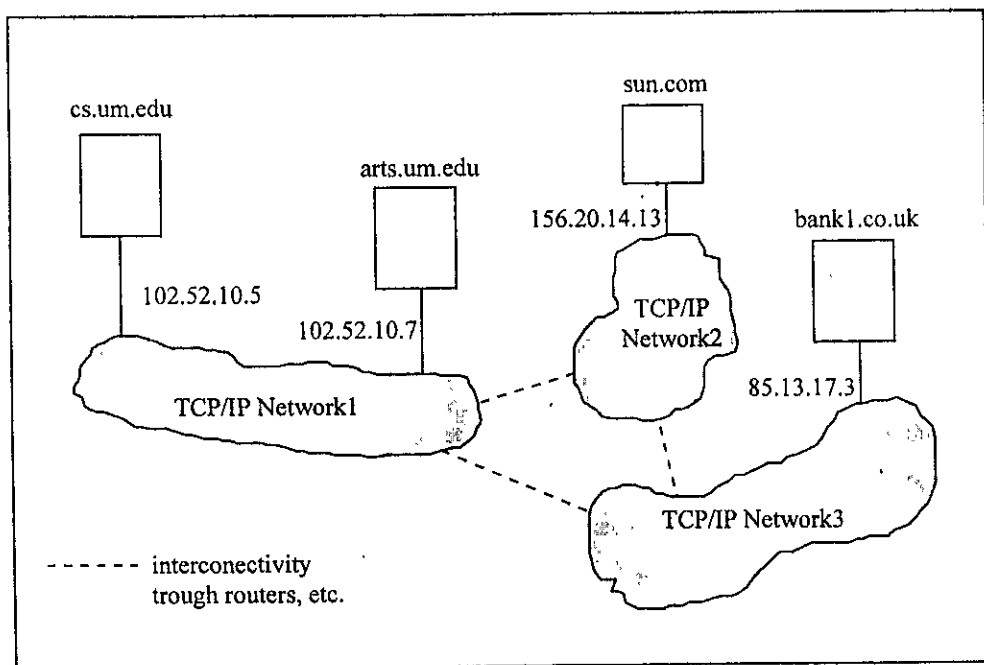


Figura 1.7.1 Vista Parcial de Internet.

Los servicios que están asociados a TCP/IP y por ende a Internet que en la actualidad son usados por muchos usuarios de Internet son:

- **Correo Electrónico (E-mail)** : Está basado en el Simple Mail Transfer Protocol (**SMTP**) que a su vez está sustentado en el protocolo TCP que es el mecanismo de intercambio de mensajes electrónicos. Este servicio es el que los usuarios usan con mayor frecuencia. Para poder tener acceso a este servicio es indispensable que dispongan de una dirección de correo, tal como, `mdiaz@mx.oracle.com`, de tal manera que así puedan recibir correo electrónico orientado a dicha dirección.
- **Telnet** : El servicio Telnet es el que nos proporciona el emulador de terminal necesario para poderse conectar remotamente a una computadora, es necesario en este servicio conocer el nombre de la máquina o su dirección IP. además del usuario y el password correctos para entrar a la computadora.
- **Transferencia de archivos (FTP)** : Este servicio permite la transferencia de información basado en el File Transfer Protocol (**FTP**)
- **Gopher** : Esta interface fue desarrollada por la Universidad de Minnesota, se puede decir que es el antecesor del WWW dado que provee una interface gráfica a herramientas tales como, FTP, Telnet, etc.
- **Servidor de Información de Área Amplia (WAIS)**: Permite la búsqueda de información en una gran cantidad de bases de datos almacenadas en Internet con un patrón de búsqueda.

### **World Wide Web**

El propósito del WWW es soportar el crecimiento del número de usuarios y abarcar una amplia gama de aplicaciones que van desde las de entretenimiento hasta los sistemas de información

corporativa. En estricto sentido técnico el WWW es una colección de middleware que opera sobre las redes TCP/IP y cuya base se encuentra en los siguientes conceptos y tecnologías:

- **Web Sites** : Es un catálogo de información que contiene cada suministrador de información que el usuario de Web requiere. Es importante mencionar que una de las fortalezas del WWW es mediante los Web Sites, que en número se han incrementado, conteniendo una amplia variedad de información, conocidas como recursos, pueden ser accedidos transparentemente por los usuarios de Web.

El Web site está compuesto por tres elementos:

- Servidor de Web
- Archivos de contenido (páginas de Web)
- Gateways

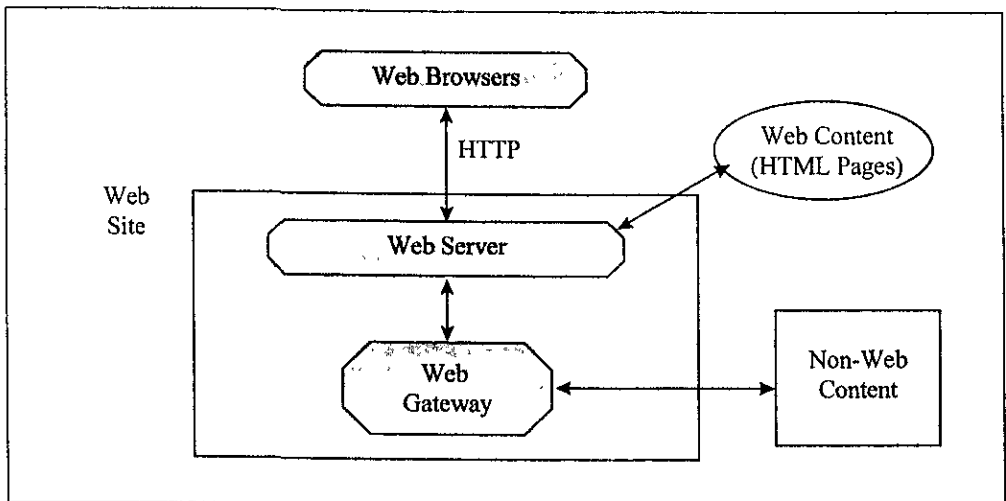


Figura 1.7.2 Un Web Site

- Servidor de Web : El trabajo que desarrolla esta pieza de software es fundamentalmente recibir peticiones del cliente de Web y accede la información, esencialmente enlista las peticiones, las agenda por decirlo de una manera más coloquial.

En sus orígenes el servidor de Web fue desarrollado por NCSA un paquete conocido como **httpd** (HTTP Daemon) primero para Unix y después fue portada para otras plataformas.

Actualmente en el mercado existen diferentes servidores de Web, tales como, Netscape, Microsoft y Oracle entre otros.

- Navegadores : Los navegadores son la interface al usuario final hacia los servidores de Web, esos navegadores son conocidos como clientes Web que típicamente reside en PCs, Macs, y estaciones de trabajo en Unix. Desde un punto de vista de usuario final, los navegadores tienen una interface gráfica y brindan una vista de Internet que es fácil de usar, también proveen menús pull-down/pop-up y botones para acceder a servidores remotos y poder navegar al interior de documentos, imprimir resultados, guardar documentos en la máquina local donde corra el navegador y desarrollar búsquedas. El primer navegador GUI (Interface Gráfica de Usuario) fue **MOSAIC**, en la actualidad existen muchos navegadores que proveen diferentes casas de software como son : NCSA X-Mosaic, NCSA Mosaic para Windows, Netscape Navigator, Spyglass, Air Mosaic, Win-Tapestry, Oracle Power Browser y Web-Explorer.

Muchos navegadores que son populares tales como, Netscape Navigator y NCSA Mosaic corren sobre muchas plataformas como son : PC Windows, Macintosh, Unix. Esta es una de las razones por la cual es muy popular WWW en el mundo corporativo, pues en el pasado los sistemas pudieron haber sido diseñados usando una interface determinada para usuarios ahora parece más natural para las empresas el uso de los navegadores como interface de usuario.

En el uso cotidiano de los navegadores se encuentra el manejo de hiperligas, que son ligas a otra información, las cuales aparecen subrayadas y como texto o imágenes realizadas. En donde, si un usuario hace un click a dichos objetos entonces el navegador usa HTTP, que es

un estándar para los navegadores, para traer la documentación requerida del apropiado Web site.

Los navegadores están diseñados para desplegar información preparada en un lenguaje conocido como HTML y comunicarse a los servidores a través de HTTP.

Los navegadores pueden acceder recursos localizados en servidores de Web, servidores de FTP, servidores de Telnet, servidores de Gopher, etc.

Hay dos funciones básicas en los clientes de Web que son : Navegación, que facilita viajar a través del Cyber-espacio desde un recurso a otro; y Consulta detallada o Browsing, que facilita la lectura a detalle de la información que está localizada en el navegador. Como estas dos son funciones que están estrechamente ligadas en los navegadores, es por esta razón que son llamados Navegadores o Browsers.

Comercialmente los navegadores poseen una serie de características que son :

- **Estandarización:** La estandarización está relacionada con el grado de concordancia entre las características operacionales del cliente con las expectativas de conexión del servidor. Básicamente esto se reduce hacia los estándares HTTP y HTML. Sin embargo, el hecho de que HTTP sea un estándar, lleva al gran peligro de comunicaciones no seguras para lo cual ahora existe el protocolo de seguridad de HTTP o Secure Socket Layer (SSL) . También el estándar HTML es importante porque existen varias versiones de HTML (versión 1, 2 & 3) y existen extensiones de HTML que vienen en el navegador Netscape.
- **Rendimiento:** Este es un punto extremadamente importante para los navegadores. La limitación del ancho de banda de muchas comunicaciones en Internet y el tiempo de carga de los recursos multimedia hace que el rendimiento del cliente sea un factor fundamental. Muchos productos de Web usan un caché para ayudar a mejorar el

rendimiento. El caché retiene los documentos ya consultados o páginas localmente para reducir el proceso lento de reconconsultar información en Internet. El caché puede existir durante la sesión, y a esto es llamado soft caching o puede ser transferido al disco, y a esto es llamado hard caching. Así mismo, los clientes pueden usar multihilos (multithreader) para tener múltiples sesiones de Web concurrentes para mejorar el rendimiento.

- **Reconfigurabilidad:** Es la capacidad de configurar el cómo se muestra o se ve el cliente Web para satisfacer necesidades del usuario. Por ejemplo, se pueden definir valores para que se desplieguen ciertas páginas de Web al momento de cargar el navegador en vez de cargar las páginas del proveedor del Navegador, otro tipo de reconfiguración podría ser cambiar el tipo de letra o colores.
- **Integración:** La integración es importante en el sentido de poder acoplarse a la máquina donde esta residiendo el navegador.
- **Ayuda de navegación:** El apoyar el proceso de navegación implicaría reducir el tiempo de navegación. El problema básico que ocurre en este rubro en Internet es que las hiperligas no escalan bien, entonces la ayuda para la navegación crea bitácoras en el cliente Web para obtener de allí de los recursos más recientemente accedidos.

La funcionalidad de los navegadores hasta cierto punto es limitada pues pasa los requerimientos al servidor de Web y despliega resultados, sin embargo, esto está cambiando muy vertiginosamente por la aparición de Java que es un lenguaje de programación que creó SunMicrosystems y con sus programas llamados Java Applets los navegadores pueden tener una funcionalidad mucho más amplia.

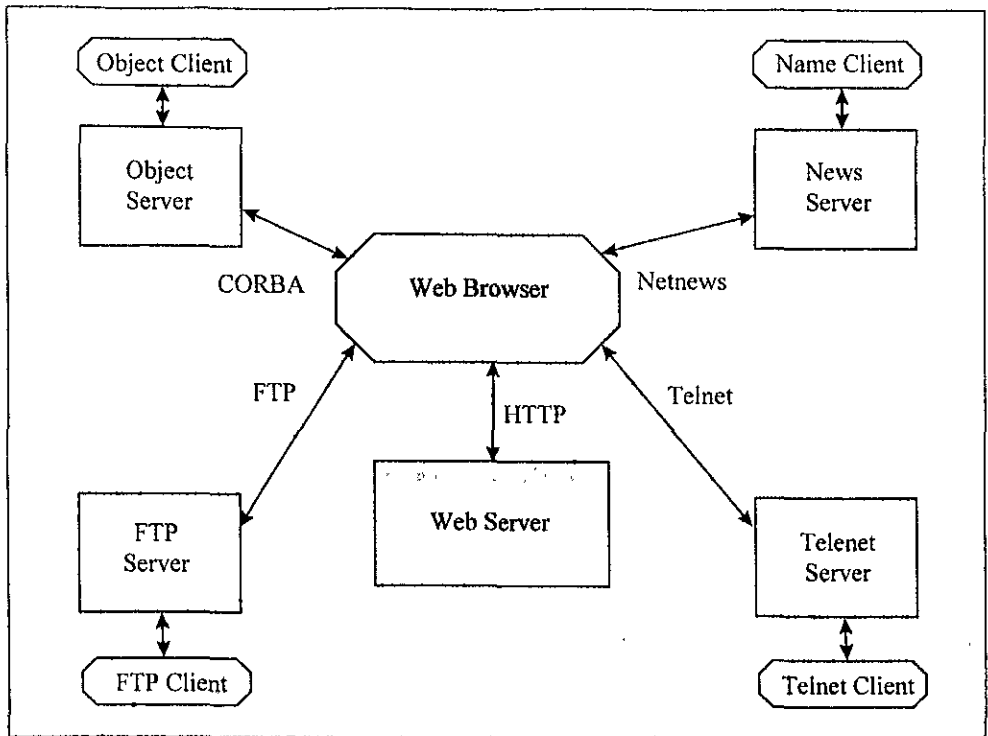


Figura 1.7.3 Interfaces de Navegadores de Web.

- **Localizador de recursos uniforme (URL)**; Es la base para localizar un recurso en WWW, que realmente es una cadena de caracteres, que funge como un identificador único para cada recurso.

Se puede localizar a un recurso escribiendo el URL en el navegador o haciendo click a una hiperliga que implícitamente invoque a un URL.

El formato general del URL es *.- protocol ://host :port/path*, donde

*protocol* representa el protocolo que regresa o manda información. Ejemplos de protocolos válidos son HTTP, FTP, Telnet, Gopher y NNTP ( Network News Transfer Protocol)

*host* es la computadora donde reside en recurso

*port* es un número de puerto

*path* es una identificación, típicamente es un nombre de archivo, que reside en la computadora

Ejemplo de un URL

“<http://cs.um.edu/faculty.html>”

donde

[http](http://cs.um.edu/faculty.html) .- le indica al servidor que un requerimiento HTTP está siendo iniciado

[cs.um.edu](http://cs.um.edu/faculty.html) .- nombre de la máquina donde corre el servidor de Web

[faculty.html](http://cs.um.edu/faculty.html) .- nombre del archivo que reside en la máquina [cs.um.edu](http://cs.um.edu)

- Protocolo de transferencia de hipertexto (**HTTP**) : Es un protocolo que está implementado sobre TCP/IP, el cual está diseñado a nivel aplicación para dar a los usuarios la ligereza y rapidez necesaria para el trabajo en estos ambientes de hipermedia.

Es necesario definir un concepto que posteriormente se va a utilizar :

Identificadores Universales de Recursos (**URIs**) : La familia completa se llama URI, la cual comprende varios identificadores.



- URL es en nombre de la localización física de objetos en WWW.
- URN (Nombres Universales de Recursos) que identifica el nombre del recurso sin dar la localización.
- URC (Cita Universal del Recurso) que describe las propiedades de los objetos.

Como HTTP debe llevar a cabo sus tareas razonablemente rápido, el modelo en el cual está basado es muy simple y consiste de los siguientes pasos :

- Conexión .- El cliente establece la conexión con el servidor, una conexión por defecto usa el puerto 80 cuando se utiliza TCP/IP. sin embargo, otros puertos se pueden especificar en el URL.
- Petición .- El cliente manda una petición al servidor, después de que el programa cliente establece conexión con un programa servidor, éste manda un requerimiento al servidor. El mensaje de petición contiene el URI, el método de petición, versión del protocolo, modificadores de petición, información del cliente y posiblemente contenido. El método es la parte más importante de este mensaje porque muestra el tipo de operación que necesita desarrollarse.
- Respuesta .- El servidor responde al cliente con una respuesta, el mensaje consiste en una línea de estado, información del servidor, información meta y posible contenido. El servidor manda la información requerida al navegador en un estándar de Internet llamado MIME ( extensiones de mail en Internet con múltiples propósitos ). MIME soporta el intercambio de Texto/HTML, imágenes, video, audio y código de las aplicaciones. El servidor precede al dato regresado con un encabezado tipo MIME que contiene bits de información, incluyendo el contenido. Los navegadores usan estos encabezados para interpretar el tipo de información que esta siendo enviada y para desplegarla al usuario.

- **Cerrar** - El cliente o el servidor cierra la conexión. En muchos casos sucede que el cliente establece una conexión antes de cada petición y el servidor cierra la conexión después de mandar la respuesta, HTTP no lo requiere pues cliente o servidor pueden iniciar y/o cerrar la conexión. Los clientes y servidores sobre HTTP deben ser capaces de manejar el evento de que se cierre prematuramente una conexión de cualquier lado de la conexión pues pueden existir fallas, tiempos de espera muy largos (timeouts) o acciones del usuario.

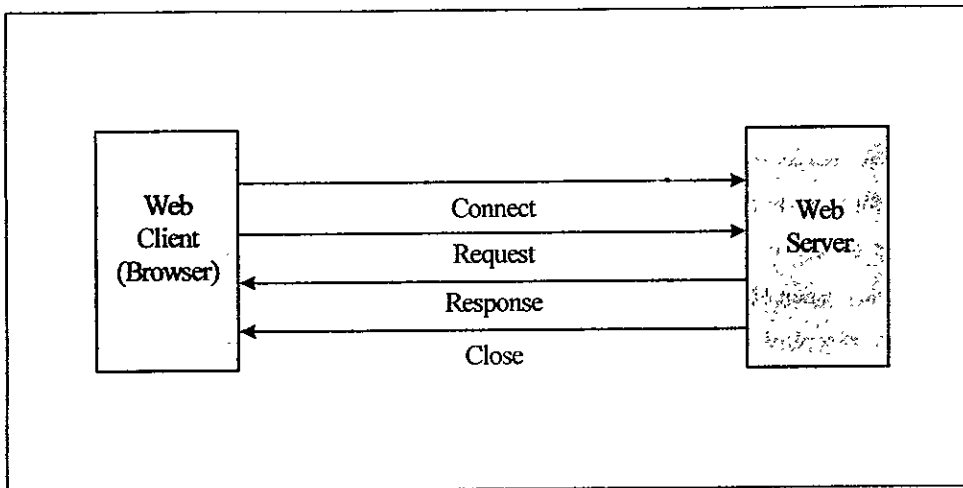


Figura 1.7.4 Una Típica sesión HTTP.

- **Lenguaje de hipertexto (HTML)**: Es el lenguaje de documentación multiplataformas, cualquier computadora que provee de un Navegador podrá leer y desplegar un documento en Internet. También HTML ayuda a la creación de ligas de hipertexto que como ya se ha mencionado da la ruta de un documento a otro. El principal propósito de éste lenguaje es mostrar documentos de manera muy similar a como se verían en una revista, libro o catálogo. Se tendrá una descripción más detallada de este lenguaje en el siguiente inciso de este capítulo.

- **Servicios de búsqueda:** Son usados para la búsqueda de información en el vasto mundo de Internet. El modelo general de búsqueda es mostrar el índice de la información disponible en Internet, de los Web Sites y el mecanismo de búsqueda es a través de palabras claves u orientadas a temas mediante los índices, que da como resultado los URLs de todos los documentos que cubrieron con el patrón de búsqueda. Ejemplos de este servicio son: Yahoo, Lycos y Alta Vista, algunos navegadores al invocar su producto muestran como página inicial (home page) algunas herramientas disponibles para este servicio.
- **Gateways a fuentes no-Web :** Los Web gateways son puentes entre los navegadores, las aplicaciones corporativas y las bases de datos. Actualmente, es urgente la necesidad de integración entre las fuentes corporativas tradicionales de información y el nuevo mundo del WWW, porque el verdadero éxito de WWW es ofrecer una interface al usuario final con contenido de información valiosa, necesaria de cualquier tipo que es provista mediante las bases de datos corporativas tradicionales. Aquí un reto importante es que los sistemas actuales tienen la característica que la respuesta por ejemplo a una consulta, depende de qué tan lejos o cerca se encuentre el usuario, además de que se manejan protocolos en donde los mensajes tienen sentido dependiendo del anterior, que es una diferencia muy marcada con HTTP, que trata de que cada interacción sea independiente a la anterior.

Los puentes pueden manejar casi cualquier cosa que no esté adaptado a los navegadores o HTML. Existen diferentes aproximaciones para desarrollar un puente de Web (Web Gateways).

- **Interface común de puentes (CGI) :** Esta clase de puentes CGI son muy populares, es un programa que reside en el Servidor de Web, este programa puede ser un script, escrito en Unix Shell o Perl; o un ejecutable de C o C++, el cual reside para su ejecución en el Servidor de Web, en el directorio cgi.bin o en cualquier otro que se haya designado para este fin. Por lo que las hiperligas a esta clase de programas pueden ser incluidos en documentos HTML de la misma manera que hiperligas a otros recursos. Cuando un usuario hace un click a este tipo de hiperligas, el URL del

programa puente es pasado hacia el servidor de Web, el servidor de Web localiza el programa puente en el directorio predeterminado y lo ejecuta, y finalmente la salida que produce el programa puente lo regresa al Navegador. Entonces como ya se puede ver, la diferencia fundamental de acceder un archivo HTML y un programa CGI, es que con HTML trae la información y la despliega a diferencia del CGI que es ejecutado en el servidor de Web para desarrollar funciones específicas incluyendo creación de páginas HTML.

Hay dos tipos de CGIs

- Puente CGI de un paso .- Un programa de aplicación es ejecutado como un CGI ejecutable por lo que el proceso de la aplicación se ve altamente restringido para cada petición. En este caso el programa ejecutable CGI contiene la lógica invocada por el cliente Web.
- Puente CGI de dos pasos .- Un programa de aplicación, en este caso corre como un proceso daemon. Un ejecutable CGI sólo despacha el requerimiento más que desarrollar cualquier función aplicativa. El puente CGI no tiene reglas de negocios y justamente es usado como despachador.

El primero de estos dos tipos es usado típicamente para funciones relativamente simples y rápidas, el segundo tipo es usado para funciones complejas, y en muchos casos, para aplicaciones de legado.

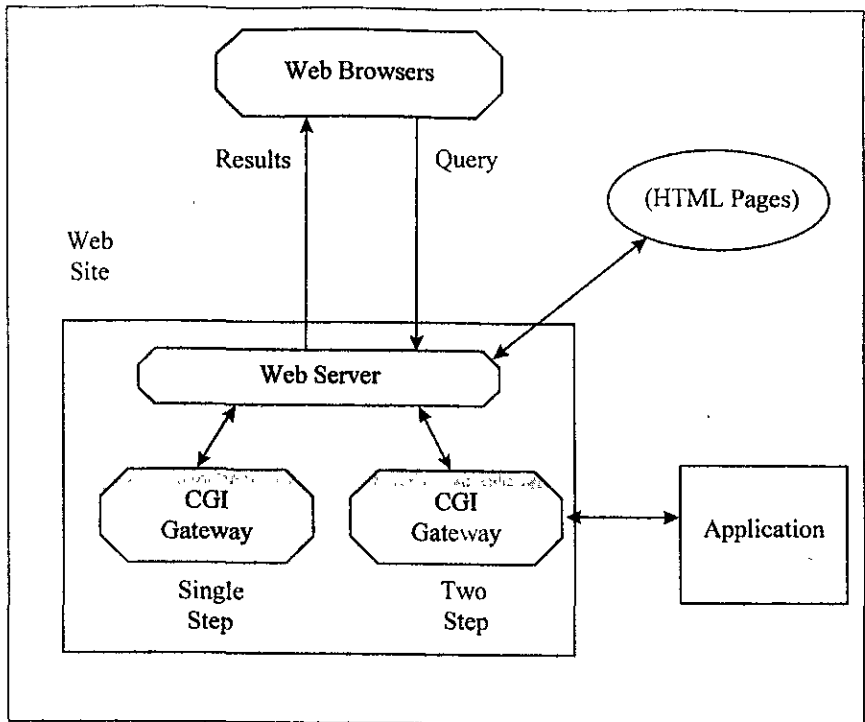


Figura 1.7.5 Gateways CGI.

- Inclusión del lado del servidor ( SSI ) : Existe otra opción para crear puentes que es SSI que consiste en una secuencia especial de caracteres (tags) en una página HTML. Esta secuencia es procesada por el servidor cómo también manda el documento HTML hacia el Navegador. Esto significa que los documentos HTML deben ser analizados por el servidor. Por consiguiente, los documentos que incluyen SSI no son estándar de HTML. Así mismo, el parseo de HTML lleva como consecuencia a problemas de rendimiento por el tiempo que ocupa el servidor en ésta tarea. Por lo tanto, no es recomendable usar SSI en un número abundante de páginas HTML. Además no todos los servidores de Web soportan SSI aunque otros se han preocupado por esto y han introducido extensiones a SSI, por ejemplo una extensión a SSI es SSI+, el cual es usado para desarrollar puentes hacia bases de datos, correo electrónico y desarrollar otras funciones.

Como resultado de la invocación de un SSI puede ser una llamada al sistema o programa, lo que puede resultar ser peligroso desde el punto de vista de la seguridad, por esta razón la mayoría de los sites comerciales han deshabilitado el soporte a SSI.

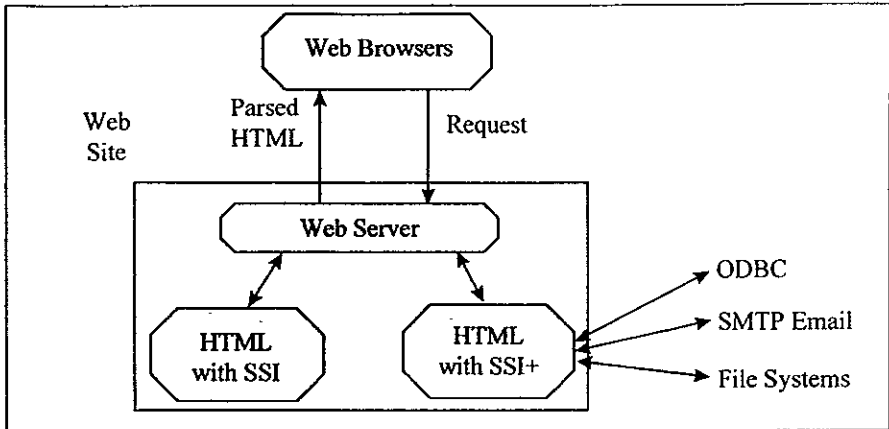


Figura 1.7.6 Inclusiones del lado del Servidor.

- Puentes como un servidor standalone : Es un servidor dedicado para tener funciones de puente que realmente podrá desarrollar mucho mejor dichas funciones que puentes CGIs o SSIs., básicamente cada llamada recibida por el servidor es asumida para desarrollar funciones de puente. Por ejemplo, un servidor dedicado a ser puente de base de datos podría recibir llamadas SQL de un Navegador y mandar la sentencia SQL a la base de datos requerida, recibir los resultados de la sentencia, construir páginas HTML de los resultados y mandar los resultados hacia el Navegador.
- Sistemas de Código Móvil (Puentes de Java) : Hasta aquí los puentes que se han visto corren del lado del Servidor, la aparición de applets de Java están cambiando esto. La idea central es distribuir el código de la aplicación y mandarlo al cliente Web donde éste se ejecuta, ya que una característica de los applets de Java es que pueden estar dentro de las páginas HTML y mandar a los navegadores, donde ellos se ejecutan. Usando este argumento el acceso a aplicaciones remotas y bases de datos pueden ser

invocados directamente desde el navegador, los applets de Java pueden preguntar por el usuario y hacer la petición, por ejemplo, de una consulta y luego mandar esta consulta a una aplicación remota o base de datos. Esto es especialmente interesante para el acceso hacia bases de datos, donde la funcionalidad del puente para base de datos corre del lado del cliente de Web. Un estándar está siendo perfeccionado para el acceso hacia base de datos que es **JDBC** (Conectividad a través de Java hacia base de datos) que permite a los programas en Java hacer llamadas hacia bases de datos relacionales.

Sin embargo, no hay que perder de vista que aplicaciones de legado no son fácilmente traducibles hacia applets de Java, pero tal vez se puede pensar en alguna parte, cómo podría ser, la interface al usuario tenerlo bajo este esquema.

Ahora se verá en forma rápida algunas soluciones o estrategias de proveedores para estos ambientes.

- Netscape : Ha desarrollado una amplia gama de productos de Web, por ejemplo, el Netscape Commerce Server para comercio electrónico; Fast Track para soportar Java y Java script; SuiteSpot para Intranets empresariales; y LiveWire Pro para soportar desarrollo en Java.
- Microsoft : La estrategia de esta compañía comprende el de Internet Information Server, que es un servidor HTTP en un ambiente Windows NT; Internet Explorer, que es un Navegador; Internet Assistant for MS Word Windows 96, que es un editor HTML para MS Word; Media Server, que es un servidor de video y audio con soporte para HTTP; Merchant Server , que es un servidor para comercio electrónico; y muchos otros como ActiveX para ligar el Web con objetos distribuidos.
- Sun : Ha anunciado algunos productos alrededor de Java, que incluye soporte a Java distribuido que podría ir a través de firewalls.

- Oracle : La estrategia de Oracle para Intranet/Internet es la arquitectura NCA (Arquitectura de Cómputo en Red); Oracle Web Application Server; que es un Servidor de Web con funciones avanzadas y del cual se hablará con detalle posteriormente; Designer/2000, que es una herramienta Case que genera código HTML y Developer/2000; Developer/2000, que es una herramienta para el desarrollo de sistemas que en tiempo real genera applets de Java; Oracle Electronic Commerce, para comercio electrónico; e Interoffice, que es una solución para correo electrónico y workflow.

### **Oracle Web Application Server**

Para soportar la evolución de Internet desde una biblioteca hasta una plataforma de negocios, Oracle, junto con proveedores de hardware y software han conjuntado la Arquitectura de Cómputo en Red (NCA), la infraestructura en diversas plataformas para el desarrollo y la ejecución basados en objetos. La NCA combina:

- Los amplios y confiables servicios del mundo Cliente/Servidor, tales como, seguridad, integridad y soporte a transacciones distribuidas.
- La robustez, escalabilidad y reusabilidad del mundo de los objetos.
- La facilidad de uso emanado del ambiente Web.

Operando dentro de la NCA, el Oracle Web Application Server 3.0 provee un servidor de aplicaciones de misión crítica que pone los fundamentos para una arquitectura abierta, escalable e interoperable, que no solo soporta diversas aplicaciones y diversos estándares, tales como, Java, CORBA y X/Open, pero que activamente se abraza de ellos.

- Fundamento para la Computación en Red. El Oracle Web Application Server 3.0 es un componente fundamental para la NCA. Un servidor de aplicaciones escalable para Web, el Web Request Broker, soporta cartuchos de aplicaciones basados en HTTP/HTML. Esto provee los servicios de un Object Request Broker que soporta CORBA hacia cartuchos de Web, ofreciendo una integración entre aplicaciones basadas en Web y Objetos. La



independencia de HTTP y los servicios de transacciones escalables aseguran un servidor de aplicaciones robusto para el Web.

- La arquitectura de aplicaciones robustas. El Web Application Server 3.0 provee una arquitectura de objetos abierta y basada en estándares ideal para alojar y lanzar diversas clases de aplicaciones Internet e Intranet. Su escalabilidad y distribución es esencial en un mundo que crece de manera exorbitante y tomando en cuenta que está envuelto en los estándares establecidos y su robustez, su diseño tolerante a fallas es apropiada para aplicaciones de misión crítica.

El Oracle Web Application Server incluye un servidor HTTP robusto, pero el corazón del producto es el Web Request Broker (**WRB**), un mecanismo de despacho de alto rendimiento, balanceo de carga y extensiones de servidores de terceros. Construye una verdadera arquitectura multihilos y multiprocesos, el WRB ofrece un ambiente aplicativo superior a los APIs HTTP. Corriendo todas las extensiones como procesos individuales, el WRB da procesamiento independiente.

- Cartuchos. La arquitectura de cartuchos provee una oportunidad única para el particionamiento de la aplicación, balanceo de carga y la separación de procesos. Los cartuchos pueden ser cargados o no dependiendo de la carga del servidor y las necesidades de la aplicación.
- Distribuido. La tecnología de objetos ha involucrado desde una simple colección de objetos reusables para traer el poderío de cómputo cliente/servidor a la red. Con objetos distribuidos, un nuevo rango de servicios pueden ser traídos a la infraestructura de cómputo, incluyendo servicios de seguridad, servicios de transacciones, servicios persistentes y otros.

Como parte de la NCA, la arquitectura OWAS 3.0 está basada sobre el modelo de objetos distribuidos de CORBA (Common Object Request Broker Architecture). Esto permite a los componentes de las aplicaciones de Oracle Web Application Server escalar a través Intranet

amplias e interoperar un vasto mundo de aplicaciones. Los cartuchos no son objetos reutilizables que pudieran ser llamados por un despachador del Oracle Web Application Server, ellos también participan completamente en los servicios ofrecidos por el API Object Request Broker y Web Request Broker.

- Tolerante a Fallas. A través de su arquitectura de procesamiento independiente, el Web Request Broker garantiza extensiones de terceros que no afectarán a otras partes del sistema. Los desarrolladores pueden construir, correr y manejar aplicaciones independientemente. Una aplicación de misión crítica demanda alta disponibilidad. La separación de procesos, una arquitectura de objetos y un manejo de cartuchos independientes permiten a los administradores construir, manejar y servir sus sistemas sobre un componente básico.

Una típica infraestructura tecnológica abierta está compuesta de diversos elementos -un amplio rango de sistemas operativos, lenguajes, redes, aplicaciones, servidores de web y bases de datos. Una verdadera arquitectura robusta no solo debe proveer soporte a todos estos estándares, debe también abrazarse de ellos lanzando los fundamentos necesarios para tener un ambiente aplicativo abierto, interoperable y heterogéneo.

- La Arquitectura Unificada. Una plataforma que verdaderamente soporta interoperabilidad es una que soporta y puentea diversos estándares. En este sentido, la plataforma por si misma empieza a ser la base para un ambiente aplicativo abierto y con extensiones. Una arquitectura abierta como CORBA, por ejemplo, ofrece lo necesario para construir aplicaciones que son reutilizables, interoperables, e idealmente independiente de lenguajes, plataformas, redes y proveedores. CORBA por si mismo es el corazón de la NCA, es un estándar. El protocolo CORBA-Internet Inter-Orb (IIOP) provee no solo la médula espinal para comunicar objetos CORBA a través de una red TCP/IP, esto también da los fundamentos para interoperar con otras arquitecturas de objetos tales como COM (Modelo de objetos común). Donde CORBA da el estado de como objetos y arquitecturas de objetos se comunican una a otra, la NCA va abajo para proveer el marco de trabajo para que esas diversas aplicaciones sean implementadas y comunicadas una con otra dentro de la misma arquitectura.

- Trayendo lo corporativo/empresarial al Web. El modelo X/Open de procesamiento de transacciones distribuidas (**X/Open DTP**) incluye tres componentes: un programa aplicativo, un manejador de recursos y un manejador de transacciones, que realmente el Oracle Web Application Server soporta este modelo dentro de su arquitectura. Sin embargo, dentro de la arquitectura del Servidor de Web de Oracle no sólo soporta el modelo X/Open DTP, sino también usa estándares abiertos como son: SQL, X/Open's XA y X/Open TX. Estos estándares significan que el Servidor de Web de Oracle trabajará con cualquier manejador de recursos XA. ICX provee el marco de trabajo para la interoperabilidad. El servicio ICX permite que distintos cartuchos escritos en distintos lenguajes participen en un mismo ambiente aplicativo. Esto es un puente que actualmente permite a las aplicaciones escritas en Perl, Java y Oracle PL/SQL comunicarse y compartir información. Esto también empezará a ser un puente para unificar distintos modelos de objetos y otros ambientes operativos tales como COM, Java, sistemas de legado y aplicaciones.

Unificando el **API** ( Interface de aplicación programática) del Servidor de Web de Oracle como escuchador independiente trabaja con el escuchador de Spyglass y también soporta servidores de Web de Netscape, Microsoft y Apache. Esto significa que la misma aplicación escrita usando WRB API, trabajará de la misma manera a través de diferentes servidores. Esto no es posible con aplicaciones escritas con NSAPI o ISAPI (APIs propietarios de Netscape o Microsoft, respectivamente).

Por lo tanto, el WRB API es portable no solo a través de un rango de sistemas operativos, sino que también en un rango de servidores de Web. Esto permite a los desarrolladores proteger sus inversiones (aplicaciones existentes) de un cambio de infraestructura tecnológica.

### **Servicios de Seguridad en Red**

ISO publicó un documento en 1989 que lleva como nombre Sistemas de Procesamiento de Información-Interconexión de Sistemas Abiertos-Modelo de Referencia Básica, en donde identifica un conjunto específico de servicios de seguridad que son los siguientes:

- Continuidad en el servicio de operaciones .- Este punto es incluido porque es de vital importancia para el éxito de las operaciones por la red. Está diseñado para asegurar que una red esté disponible para proveer el nivel de servicios requeridos para así llegar al objetivo o propósito que fue definido para la red. Muy comúnmente, la negación del servicio de red es el más serio de los problemas que pueden ocurrir en este ambiente, para detectar a tiempo cualquier caída, las conexiones puerto a puerto intercambian un conjunto de mensajes para verificar que una ruta abierta y clara sigue existiendo entre dos entidades que se comunican, incrementando la frecuencia del intercambio de mensajes reducirá la cantidad de tiempo requerido para identificar un daño en la red. Los mecanismos que se usan para garantizar la continuidad de las operaciones son:
  - Componentes altamente confiables y tolerantes a fallas
  - Componentes redundantes y rutas de comunicación
  - Un Routing alternativo
  - Planes y procedimientos del manejo de contingencia
  - Almacenamiento redundante (fuera del site) y facilidades de procesamiento
  
- Servicios de Integridad .- Provee protecciones necesarias para proteger información de las manipulaciones o modificaciones no autorizadas, predomina fundamentalmente en dos áreas: La protección de un dato simple o la protección de un data stream, conexión o sesión compuesta de más de un elemento de datos. Para el primero de ellos a un contenido o un valor de chequeo es asociado con una unidad de datos que es normalmente originado por el proceso que lo envía, el contenido o el valor de chequeo es proveído a través de un número de mecanismos incluyendo el uso de CRC (Ciclo Común Redundante) o a través del uso de encriptación punto a punto. La protección para un data stream es proveído a través del uso de etiquetas de fechas (time stamping) o secuencias de números. Uno de los mecanismos de seguridad de redes comúnmente usado para soportar servicios de integridad es la encriptación, la cual protege de modificaciones o actualizaciones no autorizadas, el proceso de encriptación provee de una protección que se podría decir que esta sellada o

herméticamente cerrada que se romperá y se reconocerá si la información es alterada o manipulada sobre la ruta hacia el proceso de descryptación.

- **Servicios de Autenticación** .- Hay dos tipos de autenticación, el primer tipo está referido a la *autenticación del dato original* esto permite que puedan ser llamadas las fuentes de origen de los datos para su verificación con los datos recibidos. El servicio de la autenticación del dato original no ofrece protecciones contra la duplicación o modificación de unidades de datos, y además es normalmente usado durante la fase del transporte de los datos. El segundo tipo esta referido hacia la autenticación puerto a puerto, ofrece la posibilidad de verificar que un puerto en asociación con otro sea el único llamado, por lo que asegura que un puerto o entidad no esté intentando llevar a cabo una respuesta de alguna asociación ya establecida. La autenticación puerto a puerto se desarrolla durante la fase de conexión o rara vez durante la fase de transportación de los datos.

La identificación del usuario y la autenticación son medidas importantes de seguridad, antes de permitir el acceso a la información, acceso a recursos, desarrollo de software o el uso de alguna aplicación , cada usuario debe ser identificado y autenticado por la red o por los recursos de red que están siendo accedidos.

- **Servicios de Control de Acceso**.- Es usado para proteger recursos, archivos, datos, y aplicaciones de red de accesos no autorizados. Un usuario o proceso debe ser identificado y autenticado antes de que el mecanismo de control de acceso pueda dirimir o mediar el acceso hacia recursos, servicios e información. Este control es proveído a través de los mecanismos:
  - **Lista de control de acceso**.- Es una lista de todos los autorizados a acceder información específica, aplicaciones o recursos de red.

- **Etiquetas de información/Datos.-** Es una forma de controlar el acceso mediante el uso de componentes confiables que soportan el uso de protocolos de seguridad. Las etiquetas están asociadas a la información que está transmitida dentro del paquete que a su vez pueden soportar uno o varios protocolos como son: **IPSO** (Opción de Protocolo de Seguridad en Internet), **RIPSO** (Opción revisada de Protocolo de seguridad en Internet ), **CIPSO** (Opción Comercial de Protocolo de Seguridad en Internet), **DNSIX** (Seguridad en red para el intercambio de Información), etc.
- **Capacidades/Funciones-basadas en el control de acceso.-** Se basa en la definición de accesos y autorizaciones a capacidades o funciones de la red, una capacidad es normalmente implementada a través del uso de una señal que sirve como identificador de un recurso.
- **Controles Lógicos.-** En la mayoría de los casos los atributos inherentes a los componentes de la red dan el significado necesario para separar usuarios, recursos, servicios y redes completas. *Puentes y ruteadores son dispositivos de red que pueden ser muy efectivos.*
- **Servicios de Confidencialidad .-** ISO/IEC 7489-2 define cuatro formas de servicios de confidencialidad: *Conexión confidencial, provee datos confidenciales de n-conexiones; Connectionless confidencial, provee confidencialidad de un single connectionless data unit; Confidencialidad selectiva, provee confidencialidad de ciertos campos dentro del dato durante la conexión o en un single connectionless data unit; Flujo de tráfico confidencial, provee protección de información que de otra manera pudiera ser expuesta o derivada de la observación del tráfico de la red. El propósito de los servicios de confidencialidad es proteger la información de una divulgación desautorizada de la misma a otros procesos, individuos o entidades.*

Los servicios de confidencialidad son proveídos a través del uso de la encriptación. Por ejemplo, la confidencialidad de conexión puede ser proveída a través de mandar datos encriptados desde un nodo/proceso con una llave privada hasta la recepción del mismo hasta otro nodo/proceso, así mismo, el servicio de confidentiality connectionless se lleva a cabo mediante el uso de encriptación y una llave privada desde donde se origina el mensaje hasta el recipiente donde se comprometió al mensaje llegar. También ambos servicios pueden ser proveídos a través del uso de mecanismos de control de rutas con el cual se pueden establecer rutas de transmisión de datos más seguras.

El servicio de confidencialidad selectiva puede ser proveído a través del uso de encriptación selectiva, es decir, sólo la encriptación de un campo específico del mensaje y el uso de una llave privada, de igual manera el servicio de flujo de tráfico confidencial puede ser comprometido a través del uso de la encriptación entre mandar-recibir nodo/proceso.

- Servicios de No-reconocimiento.- Son usados para la protección contra el originador de un mensaje o acción, negando que él o ella originó el mensaje o acción. También para proteger contra el recipiente de mensajes, negando que él o ella recibió el mensaje. Este servicio se ofrece de dos maneras: El No-reconocimiento con la prueba de origen, esta prueba es hecha hacia el recipiente del mensaje; El No-reconocimiento con prueba de entrega, que es hecha al que manda un mensaje.

Adicionalmente, existen mecanismos de notarización que pueden ser usados para envolver el registro de datos con un tercero confiable (notario). Este registro puede dar un alto nivel de aseguramiento y confianza en la precisión del dato en el origen, destino, tipo e integridad.

## 1.8 CARACTERÍSTICAS, COMPARACIÓN Y SELECCIÓN DEL LENGUAJE PARA LA GENERACIÓN DE PÁGINAS DE WEB

### HTML (Lenguaje de Marcas de Hipertexto)

Es un lenguaje fácil de usar para crear documentos con tags (códigos) que son leibles por Navegadores, que permite mezclar texto, ligas a otros documentos, ligas a texto, ligas a gráficos. El principal objetivo de HTML es permitir al usuario navegar a través de documentos tal como se haría con un libro, revista o catálogo, pero trayendo imágenes, sonido y vídeo solo si lo escogemos. Los documentos HTML pueden contener texto, imágenes, audio y vídeo. Este es un lenguaje de documentación, que cualquier computadora equipada con un navegador de Web puede interpretar .

Los documentos no son WYSIWYG (what you see is what you get = lo que se ve es lo que se obtiene). Los documentos HTML se crean con editores de texto ASCII, utilizando tags, para indicar encabezados, tipos de letra, listas ordenadas gráficas, sonido y otras piezas de información ubicados en el documento.

Por ejemplo, en la figura 1.8.1 se muestra el archivo html y en la figura 1.8.2 la salida de dicho archivo, vista desde un Navegador.



```
<html>
<head>
<TITLE>A Simple HTML Example</TITLE>
</head>
<body>
<H1>HTML is Easy To Learn</H1>
<P>Welcome to the world of HTML. This is the first paragraph. While short it is still a paragraph!</P>
<P>And this is the second paragraph.</P>
</body>
</html>
```

Figura 1.8.1 Archivo en HTML.

**HTML is Easy To Learn**

Welcome to the world of HTML. This is the first paragraph. While short it is still a paragraph!

And this is the second paragraph.

Figura 1.8.2 Salida de un archivo HTML.

El hipertexto y la hypermedia son el corazón de HTML.

El hipertexto describe el sistema que permite que un documento tenga ligas o referencias cruzadas, de tal manera que un lector puede explorar otros documentos escogiendo una palabra en texto iluminado o subrayado. Estas ligas o referencias que son visibles en pantalla son conocidas como ligas de hipertexto.

Hypermedia es el mismo concepto que hipertexto solo incluyendo ligas de todo tipo, como texto, audio, gráficas y vídeo. La hypermedia permite que gráficas, audio, vídeo y elementos de vídeo puedan ser ligados a otros documentos o a otros elementos multimedia. En otras palabras

la hypermedia permite ligar elementos multimedia por medio de ligas hypermedia (se puede escoger con el ratón una opción dentro de una gráfica).

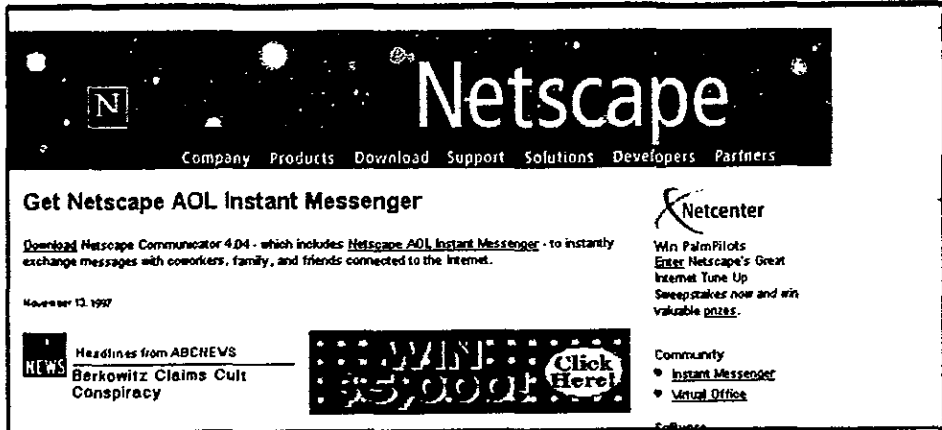


Figura 1.8.3 Ejemplo de página de Web.

En la figura 1.8.3 se aprecia una página HTML con ligas a anuncios comerciales y de características del Netscape, tales como Company, Product, Support, etc.

Lo que hace a HTML y a WWW únicos es que tienen multimedia, como lo define el modelo DEXTER. Este modelo para multimedia es el resultado de un taller de diseñadores multimedia que se llevó a cabo en el Dexter Inn de Sunapee, New Hampshire, en octubre de 1988. En subsecuentes reuniones, el grupo Dexter desarrolló un modelo de datos y procesos que pudiera ser una referencia estándar en la cual los nuevos sistemas de hipertexto pudieran ser analizados, evaluados y comparados.

Las hotlinks son la característica principal del hipertexto, hypermedia y HTML. Estas hotlinks son hiperligas que contienen la trayectoria de un documento a otro. HTML incluye tags de tal manera que se puedan tener hiperligas en cualquier parte del documento. Generalmente las hiperligas son resaltadas de alguna manera (por ejemplo con color y subrayados) cuando son desplegadas con los Navegadores. El uso de HTML permite navegar a través de documentos del

Web, como si nos encontráramos en un biblioteca, pero de una manera muy poderosa, ya que con solo hacer click en una hyperliga se puede acceder el documento, gráfica, vídeo, o sonido que se encuentra en alguna parte de la Web.

Por ejemplo suponga que está navegando en un artículo HTML, que describe varios productos de comunicación, y cada producto tiene una liga hacia el **home page** (página principal de web perteneciente a una empresa, usuario o institución educativa, es el equivalente al índice y cubierta de un libro) del distribuidor de cada producto. De tal manera que podríamos empezar a navegar en un home page de Chicago, después leer otro de Nueva York y después saltar a Inglaterra, o bien darle la vuelta al mundo a través de las ligas de hipertexto.

En la fig. 1.8.4 se aprecia una ruta que se puede seguir fácilmente al navegar a través de la Red de redes (World Wide Web).

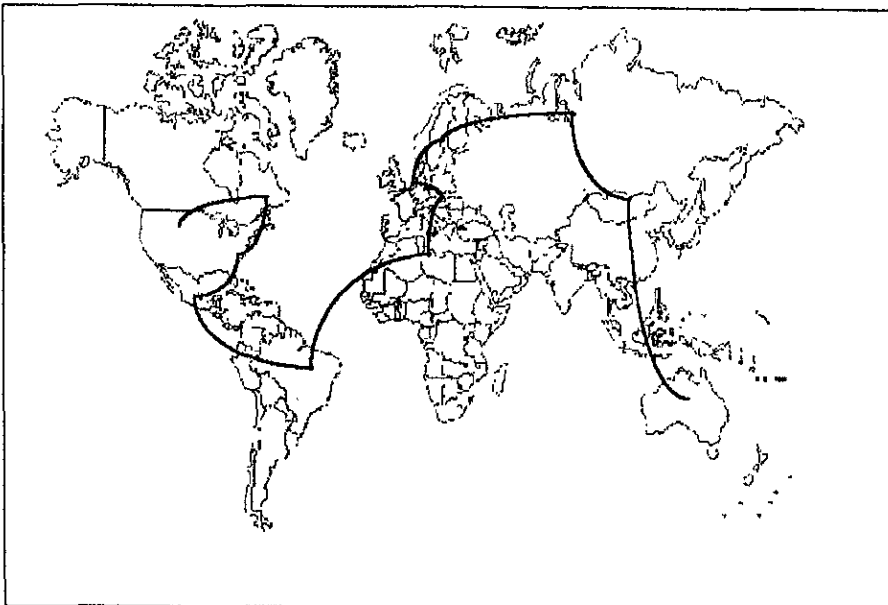


Figura 1.8.4 Navegación a través del Web.

Las hyperligas utilizan los URL para localizar la información, de tal manera que cada hyperliga inicia una sesión http.

Las capacidades de HTML incluyen características básicas como **fill-in-forms** (formatos para ser llenados), para enviar búsquedas sobre un valor capturado en la forma. Las características principales de HTML incluyen definición de títulos, listas, párrafos, ubicación de imágenes, etc. La capacidad fill-in-forms permite la instrucción **form** (forma). La forma es una construcción de HTML que ha sido utilizada para desarrollar gateways de Web a Sistemas de Información Corporativa y Bases de Datos Relacionales. Típicamente una forma contiene campos donde los usuarios introducen datos de una manera estructurada y después pueden enviar su forma a través del Web. Los navegadores usan la instrucción form para construir un URL y datos a ser enviados al servidor de Web. El servidor de Web pasa esta información a un script que realiza las operaciones necesarias y regresa los resultados al cliente. Una forma es implementada como se muestra en el siguiente ejemplo de las figuras 1.8.5 (parte del archivo fuente) y 1.8.6 (página de Web).

```

For a new account choose <B>Register</B> above.
<BR> To login fill in your userid and password &mp; select
<strong>Login</strong>.
<BR> If you are having trouble logging in see <A HREF="/cgi-
bin/iv/do.pl/CAT?WwwCid=021301:31:24.25375&file=loginprobs&
Tab=ADMIN">login problems</A>

<Form method=POST action="/cgi-bin/iv/do.pl/LOGON">
<INPUT TYPE=hidden NAME=WwwPrevAction VALUE="">
<INPUT TYPE=hidden NAME=WwwPrevSource VALUE="">
<INPUT TYPE=hidden NAME=WwwPrevID VALUE="">
<TABLE ><TR>
<TD>UserID (name<B>.</B>cc):</TD>
<TD><INPUT TYPE=TEXT SIZE=14 NAME="TMP.UserID" VALUE=""> </TD>
<TD COLSPAN=4>
Display as:<SELECT NAME="TMP.Colour">
<OPTION VALUE=""> per Preferences
<OPTION VALUE="col"> Colour
<OPTION VALUE="bw"> Black & White
<OPTION VALUE="text"> Text Only
</SELECT></TD></TR>
<TR><TD>Password:</TD>
<TD><INPUT TYPE=password NAME=TMP.WwwPasswd SIZE=10></TD>
<TD></TD>
<TD></TD>
<TD><INPUT TYPE=submit VALUE="Login"></TD>
</TR></TABLE>
</FORM>

```

Figura 1.8.5 Programa HTML.

Web

Ora\*Notes Search Lookup Ora\*Web Utilities Admin

Login Register Feedback Help Reset

For a new account choose **Register** above.  
 To login fill in your userid and password & select **Login**.  
 If you are having trouble logging in see [login problems](#)

UserID (name.cc)  Display as

Password:

Figura 1.8.6 Salida de un programa HTML.

La palabra METHOD especifica el método a ser usado para enviar la información al servidor. Method utiliza dos argumentos, Get y Post. Muchos navegadores utilizan post porque es más eficiente. La palabra ACTION especifica el URL al cual los datos serán enviados. Esta URL generalmente es una liga a un script ejecutable o archivo binario que es utilizado como gateway . Las instrucciones INPUT son utilizadas para recibir diferentes tipos de entradas como texto (TYPE=TEXT). Típicamente una serie de instrucciones tipo TEXT son usadas para recibir datos de clientes y asignar nombres a los campos de entrada (estos campos son usados para procesar scripts). Las figura 1.8.6 muestra parte del código HTML que permite al usuario capturar un usuario y contraseña.

Muchos navegadores en el momento actual utilizan HTML versión 2.0, aunque la versión 3.0 se está popularizando rápidamente. Existen algunas diferencias entre los estándares aprobados, estándares propuestos y extensiones que están siendo utilizadas. Los navegadores usualmente saltan los tags que no pueden reconocer, esto crea problemas a los desarrolladores que están probando sus páginas de Web contra una amplia variedad de navegadores para asegurar que el material desarrollado pueda ser mostrado de manera aceptable en diferentes navegadores. Para

evitar ésto, el IETF (Internet Engineering Task Force) introdujo un HTML estándar en noviembre de 1995. Sin embargo muchas páginas de HTML se presentan utilizando superconjuntos de instrucciones de este estándar.

Los documentos HTML pueden ser desarrollados utilizando un editor de texto ASCII, pero ya han salido al mercado editores HTML para ayudar a los usuarios a construir documentos HTML. Estos editores automáticamente colocan los tags en el documento, el usuario simplemente apunta y hace clicks en el icono de la barra de herramientas. Ejemplos de estos editores son Microsoft Internet Assistant for Word, Quarterdeck's Webauthor, HTML Assistant, HTML Editor, etc. Adicionalmente existen filtros que convierten documentos creados con procesador de texto a HTML.

## Java

Java es un lenguaje de programación orientado a objetos que ocupa un lugar muy importante en el WEB. Java es capaz de soportar diversos arreglos de aplicaciones del WEB y también está preparado para aplicaciones distribuidas sobre Internet.

Este lenguaje de programación fue introducido por Sun Microsystem para desarrollar software avanzado para consumidores electrónicos. “ James Gosling y un pequeño grupo de programadores comenzaron a extender el compilador de C++ en 1991, la meta fue desarrollar un lenguaje de programación orientado a objetos *neutral de la plataforma* que permitiera a dispositivos de consumo electrónico como cajas set-top y láser disc player hablarse uno al otro. El mercado para el producto falló para materializar sus objetivos. Y entonces Sun anunció el lenguaje de programación Java y el Hot-Java en mayo de 1995, los productos fueron aclamados por diferentes razones : los desarrolladores buscaban herramientas para escribir una vez y correr en cualquier plataforma, de tal manera que pudieran crear aplicaciones de misión crítica independientes de la plataforma y desplegarlas sobre el Web, y además poder atraer un mundo cada vez mas grande de usuarios que quisieran convertir páginas estáticas del Web a

documentos interactivos, dinámicos e independientes de la plataforma, en tiempo real y con seguridad para poder navegar en estos documentos.

Y Java era más que un café por la mañana o una isla lejana, su líder James Gosling se convirtió en un ídolo para los fans de Java. Gosling comenzó a trabajar con Sun desde 1984, fue el ingeniero líder para el sistema de ventanas News así como también el autor de **EMACS** (el editor de texto para Unix) y participó en otros muchos proyectos de compiladores y sistemas.

La meta, decía Gosling, fue habilitar Java para que pudiera ser manejado de un lugar a otro, sin barreras y separado de los datos. Gosling frecuentemente describe Java como un C++ con seguridad. Java y las herramientas Web son inherentes en su organización al ambiente cliente-servidor. Si se construye una aplicación para el cliente en Java, al entrar a la aplicación el cliente solo hará switch a una página, la instalación es trivial, solo coloque la aplicación en el servidor de Web, y no hay puertos, solo existirá una versión de la aplicación."<sup>1</sup>

Pequeños programas de Java, son conocidos como **Java Applets** y estos applets pueden ser incluidos en páginas Web (estas páginas son conocidas como páginas con el poder de Java, **Java-powered pages**). Estas páginas son descargadas del lado del cliente y hace a los navegadores una herramienta poderosa. Los navegadores de Web, en estos momentos son relativamente tontos (su funcionalidad reside en los servidores de Web, no en los navegadores en si). Java está cambiando esto porque los applets pueden correrse en navegadores con Java habilitado. Cuando un usuario accesa estas páginas (con applets), estos applets se descargan al navegador. Los applets corren en el lado del cliente, estos convierten al navegador en un componente inteligente. Básicamente cuando un usuario hace click en una Java-powered page, los applets son descargados al centro de cómputo del navegador, donde corre como fue programado. Esto tiene varias implicaciones:

---

<sup>1</sup> Texto tomado de "Java the Inside Story", 1996 Sun Microsystems

- Los applets son un ejemplo de código móvil el cual es desarrollado en un centro de cómputo y es migrado a otro centro de cómputo en demanda, esto involucra muchos puntos de seguridad pero también crea muchas oportunidades de investigación.
- Los applets convierten a las aplicaciones de Web en aplicaciones cliente-servidor, porque el código de Java puede correr unidades lógicas en el centro de cómputo del cliente (el navegador de Web es la primera capa de la arquitectura de 3 capas).
- La pantalla de despliegue puede ser cambiada dinámicamente en base al tipo de usuario. Los programas de Java pueden determinar el tipo de usuario y de acuerdo a esto cambiar la pantalla de despliegue.
- Diferentes avisos pueden ser iluminados dependiendo de las características del usuario (por ejemplo edad, tipo de trabajo, nivel de educación, historial crediticio, nivel salarial, etc.)
- El acceso a la base de datos puede ser invocado directamente del navegador, en lugar de invocar a un gateway que se encuentre en el servidor de Web. Un estándar llamado JDBC ha sido desarrollado para permitir a los programadores de Java realizar llamadas a bases de datos relacionales.
- Se pueden producir gráficas y cuadros dinámicamente en el navegador en lugar de hacer un fetch predefinido de gráficas y cuadros hacia el servidor de Web
- Se pueden correr animaciones, invocar transacciones de negocios y correr hojas de cálculo en el centro de cómputo donde se encuentra el navegador



## Características de Java

- Simple. Fue desarrollado para ser similar a C++ para hacerlo más comprensible a los programadores de C++. Java omite muchas características de C++ como los operadores overloading. La recolección automática de basura fue añadida para simplificar las programación en Java. Fue diseñado para correr standalone en pequeñas máquinas, el tamaño del intérprete y el soporte de clases es de aproximadamente 40K, las librerías estándar y el soporte a thread implica 175K adicionales.
- Orientado a objetos.
- Distribuido. La principal ventaja de Java es que las aplicaciones desarrolladas pueden acceder objetos sobre la internet vía URL's de manera similar a acceder filesystems locales.
- Robusto. Java hace mucho énfasis en el chequeo para evitar posibles problemas, realizando chequeo dinámico y eliminando situaciones de error. Java requiere declaraciones y no soporta declaraciones implícitas estilo C. La gran diferencia entre Java y C/C++ es que Java no soporta aritmética de apuntadores. En suma Java no permite que números enteros sean convertidos a apuntadores.
- Arquitectura neutral . Java fue diseñado para soportar aplicaciones en red. El compilador de Java genera archivos objeto de arquitectura neutral que pueden ser ejecutados en muchas plataformas, con la presencia del sistema run-time de Java. El compilador genera instrucciones de código de byte que son independientes de la arquitectura de la máquina. Los códigos byte son diseñados para ser fáciles de interpretar en cualquier máquina y fácilmente trasladados a código máquina en el vuelo.
- Portable. Java especifica tamaños de datos primitivos con la respectiva ventaja aritmética de ello, por ejemplo int siempre significa un entero de 32 bits.

- **Alto rendimiento.** El código byte generado por Java es interpretado al vuelo en código máquina en el centro de cómputo del cliente.
- **Dinámico.** Java fue diseñado para adaptarse al medio ambiente. Java entiende interfaces, un concepto que es usado ampliamente en sistemas distribuidos a través de IFL (Interface Definition Language) . Una interface es simplemente una especificación de un conjunto de métodos al cual un objeto responde. Las interfaces hacen posible el uso de objetos en un ambiente distribuido.

### Java Applets

Los applets de Java son pequeños programas, la diferencia fundamental con una aplicación desarrollada con Java es que la aplicación de Java puede correr standalone y puede utilizar texto de entrada-salida. Los applets solo pueden correr como parte de un navegador habilitado para Java. Los applets contienen métodos (subrutinas) para inicializarse a sí mismas, dibujarse a sí misma, responder a clicks, etc. Estos métodos son invocados por el navegador habilitado para Java. La manera por medio de la cual el navegador sabe que debe descargar un applet es a través del tag `APPLET` que le indica la localización del applet en el servidor. Cuando un navegador encuentra este tag, lo descarga y lo corre.

Los applets son lo suficientemente pequeños para ser incluidos en páginas Web, pero lo suficientemente grandes para hacer una tarea útil. Los applets son transferidos al centro de cómputo donde se encuentra el navegador junto con todo lo que está incluido en la página del Web (texto, imágenes, vídeo clips, etc.). Una vez que son transferidos, se ejecutan en el centro de cómputo del cliente y no sufren los retrasos del tráfico de la red entre el cliente y el servidor del Web. Como la aplicación corre en el cliente la ejecución se vera más natural y con una ejecución más eficiente.

Debido a la gran popularidad de los applets, una gran variedad de applets de plug-and-play se encuentran disponibles. Una vez construidos los applets pueden correr en muchas máquinas

diferentes. El código de Java es primero compilado en código byte (instrucciones de máquina que son independientes de la plataforma). El código byte del applet es cargado en el navegador, donde corre eficientemente en diferentes máquinas utilizando un intérprete de run-time. Copiando la apariencia de los applets de Java, otros lenguajes de programación como C++ y Cobol han comenzado a producir código byte que puede ser invocado por los navegadores de Web (al navegador no le interesa como fue creado el código byte).

Los applets tienen acceso a un amplio rango de librerías que les permiten realizar operaciones como gráficas, descargar imágenes, reproducir archivos de audio e interfaces de creación (como botones, barras de scroll, ventanas, etc.). Estas librerías son incluidas como parte de los applets API de Java. Estos API's son soportados por todos los navegadores compatibles con Java. Se espera que con el tiempo todas estas librerías crezcan y que los applets cada vez sean mas poderosos y se diversifiquen.

## JDBC

Java Soft, Oracle y otros han hecho las especificaciones para JDBC como la interfaz dinámica estándar (nivel llamadas) por medio de la cual los programas de Java pueden realizar accesos SQL .

JDBC es una clase independiente del proveedor de bases de datos, incluye manejo de conexiones múltiples a varias bases de datos, manejo de transacciones, queries, manipulación de instrucciones SQL, llamadas a procedimientos almacenados, acceso al diccionario de datos y manipulación de cursores. JavaSoft provee una implementación de JDBC construida sobre un driver ODBC. Este tipo de implementación es conocido como driver de tipo I, y soporta virtualmente cualquier base de datos que tenga un driver ODBC.

Oracle cuenta con 2 implementaciones de JDBC optimizadas para clientes tradicionales y para clientes pequeños (thin clients). Ambos drivers cumplen al 100% con la implementación de la versión estándar 1.22 de JDBC y comparten varias extensiones para el RDBMS Oracle,

incluyendo diseño de mejor performance para optimizar la comunicación cliente-servidor, soporta también los tipos de datos específicos de Oracle como number, date y ref cursor, como también tipos para objetos (Oracle 8).

**Driver OCI o tipo II :** Este driver implementa llamadas de bajo nivel JDBC como métodos nativos, que fueron escritos en C, utilizando la interfaz de programación en C para Oracle (OCI) . Este driver tiene un excelente performance y beneficios de los servicios de SQL\*Net ( como tnsnames, amplia portabilidad).

**Thin driver o tipo IV :** está completamente implementado en Java, no necesita ninguna instalación especial y es descargable. Esto lo hace ideal para los Java applets que son interpretados por navegadores habilitados para Java que trabajan en clientes como PC's o network computers.

## **JSQL**

Mientras JDBC está diseñado para programas que dinámicamente accesan tablas SQL cuyos esquemas son desconocidos al tiempo de compilación, mucho código escrito por desarrolladores está actualmente diseñado para esquemas estáticos que son conocidos al tiempo de compilación. Oracle ha trabajado junto con IBM y Tandem en el diseño de JSQL, como un estándar mediante el cual las instrucciones SQL pueden estar incluidas en el texto de programación Java . Comparando JDBC a su equivalente JSQL los programas en JSQL son mucho más concisos, seguros y eficientes ( gracias al potencial de la compilación estática del SQL incluido) .

## **Javascript**

Java script maneja eventos que no requieren respuesta del servidor de Web para validaciones y manejo de eventos. Igual que Java, Java script permite extender las capacidades de Web más allá de las capacidades de HTML y del servidor de aplicaciones de Web Oracle. Java script es menos complejo que Java y también menos poderoso . Es posible combinar Java y Java script

El siguiente cuadro muestra las diferencias entre Java script y Java :

	JAVASCRIPT	JAVA
<b>ESTRUCTURA DEL LENGUAJE</b>	Lenguaje de programación limitado, mas como macros	Lenguaje de programación orientado a objetos
<b>TIEMPO DE DESCARGA</b>	Menor debido a que es menos código	Código completamente compilado deberá ser descargado para correr. Las funciones requieren más código que Java Script
<b>COMPATIBILIDAD DE NAVEGADORES</b>	Netscape's Navigator o Microsoft's Internet Explorer	no hay ningún navegador específico
<b>COMPLEJIDAD</b>	Es una alternativa simple para pequeños proyectos. Este código es puesto con HTML e interpretado por navegadores habilitados para Java. Fácil de aprender	Escribir applets requiere escribir código fuente, compilarlo y poner el applet disponible en el servidor de Web. Difícil de aprender

	JAVASCRIPT	JAVA
<b>COMPATIBILIDAD CON HTML</b>	Javascript es combinado directamente con HTML	Los applets son descargables como archivos separados, y son llamados desde HTML
<b>ESTRUCUTURA DEL LENGUAJE</b>	Más fácil que Java, pero con funcionalidad limitada	Más complejo que JavaScript, pero más flexible y potente
<b>PLATAFORMAS</b>	Cualquier plataforma con Netscape Navigator o con Internet Explorer	Cualquiera
<b>APLICACIONES STANDALONE</b>	No	Si
<b>PERFORMANCE</b>	Generalmente bueno, pero malo para cálculos numéricos complejos	Una vez descargado, los Applets tienen un muy buen performance. El código se corre a través de un intérprete, así que el performance no es tan bueno como el de un ejecutable
<b>SEGURIDAD</b>	No tiene esta característica . Datos encriptados pueden ser pasados usando funciones	API's de seguridad para autenticación, encriptación y control de seguridad has sido desarrolladas por Sun

Figura 1.8.7 Comparativo de Java vs. JavaScript.

## HTML vs JAVA

Java es todo un lenguaje capaz de crear aplicaciones standalone a diferencia de HTML que es solo un lenguaje de marcas de hipertexto. La superioridad de Java sobre HTML quedó de manifiesto con toda la información comentada anteriormente.

HTML es un lenguaje que incluye Java Applets, con lo cual se hace muy poderoso, pero sin la inclusión de estos applets es solo un lenguaje de marcas de Hipertexto. La información está cambiando constantemente y se hace necesario que esta información que está cambiando segundo a segundo se despliegue dinámicamente en páginas de Web, esto Oracle lo soluciona generando html's dinámicos.

Designer que es la herramienta que se utilizará para la aplicación genera HTML con la creación de páginas al vuelo y Developer es capaz de generar Java, además de que Oracle soporta JDBC y cuenta con JSQL. Con esto se verifica que con Oracle se cubren varias opciones para desarrollar una aplicación al Web.

Como el caso de nuestra aplicación es un ingeniería en reversa de una aplicación ya existente, se ha escogido Designer y por lo tanto se generarán páginas HTML al vuelo.

En la figura 1.8.8 se ilustra como trabaja la opción de Web

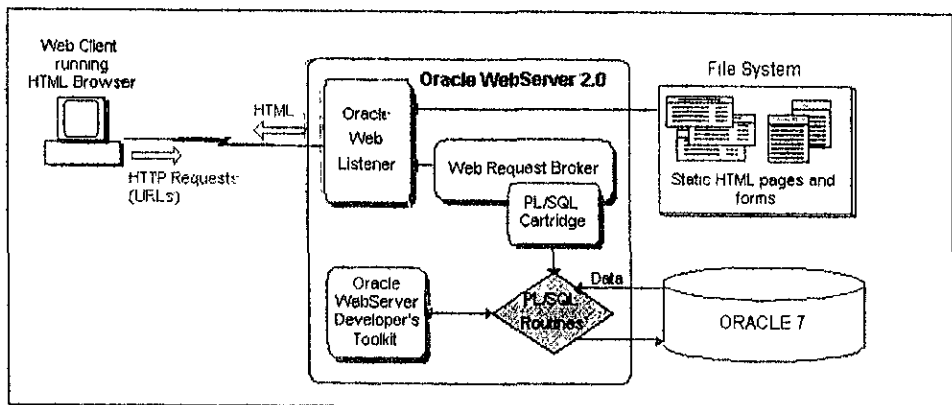


Figura 1.8.8 Web Server.

El Web Server de Oracle es un servidor HTTP integrado con la base de datos Oracle. Las reglas del negocio son almacenadas en la base y son formateadas en documentos HTML con el Web Server y transmitidas a los clientes del Web por petición de estos. Los usuarios solicitan

información HTML introduciendo un URL adecuado. Este URL especifica la máquina fuente donde se encuentra instalado el centro de cómputo para el Servidor de Web, el descriptor de la conexión a la base de datos (información necesaria para conectarse a la base de datos) y la rutina de PL/SQL que se correrá en el servidor de bases de datos.

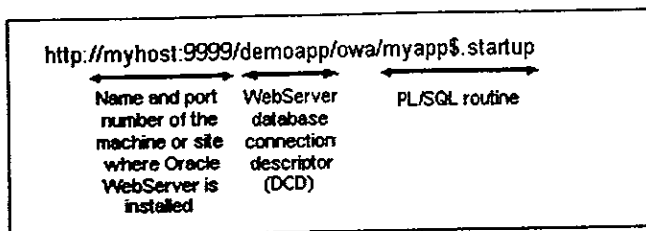


Figura 1.8.9 Ejemplo de conexión Web.

La rutina especificada PL/SQL con el URL hace fetch de los datos de la Base de Datos y entonces hace uso de las rutinas PL/SQL provistas con el Toolkit de Desarrollo de Web para formatear datos con la sintaxis HTML. Los datos formateados son ruteados al usuario que originó el requerimiento, vía el Web Request Broker.



---

**Capítulo 2**  
**Planteamiento del Problema y Propuesta de Solución**

.

---

## 2.1 PROBLEMÁTICA Y ORGANIGRAMA DE LA SITUACIÓN ACTUAL

El Objetivo de este capítulo es definir claramente la situación actual en la que se encuentra funcionando el Sistema de Información de Preventa y Canales Alternos de la compañía Oracle de México y la problemática a la que se enfrenta, así mismo, es establecer una posible solución y los alcances de la misma.

### **Disponibilidad**

Dentro de la organización el Sistema es utilizado principalmente por las siguientes personas:

- Ingenieros de Sistemas de Preventa.
- Gerentes de las áreas de Preventa.
- Personal de apoyo para captura de información.

Los Ingenieros de Sistemas de Preventa lo utilizan principalmente para introducir las actividades que están realizando. La gerencia lo usa para obtener reportes y analizar la información para conocer el desempeño del personal así como el estado de las transacciones llevadas a cabo con los clientes.

Es importante hacer notar que actualmente el Sistema se encuentra en modo **multiusuario**, es decir centralizado en una máquina en donde para tener acceso es necesario utilizar una terminal modo carácter ó bien una computadora con emulación de terminal. Para esto es necesario que la computadora que se utiliza esté dentro de la intranet de la empresa para que pueda tener acceso al servidor vía **telnet** (sesión remota) ó contar con un módem para realizar la conexión.. Ver figura 2.1.1.

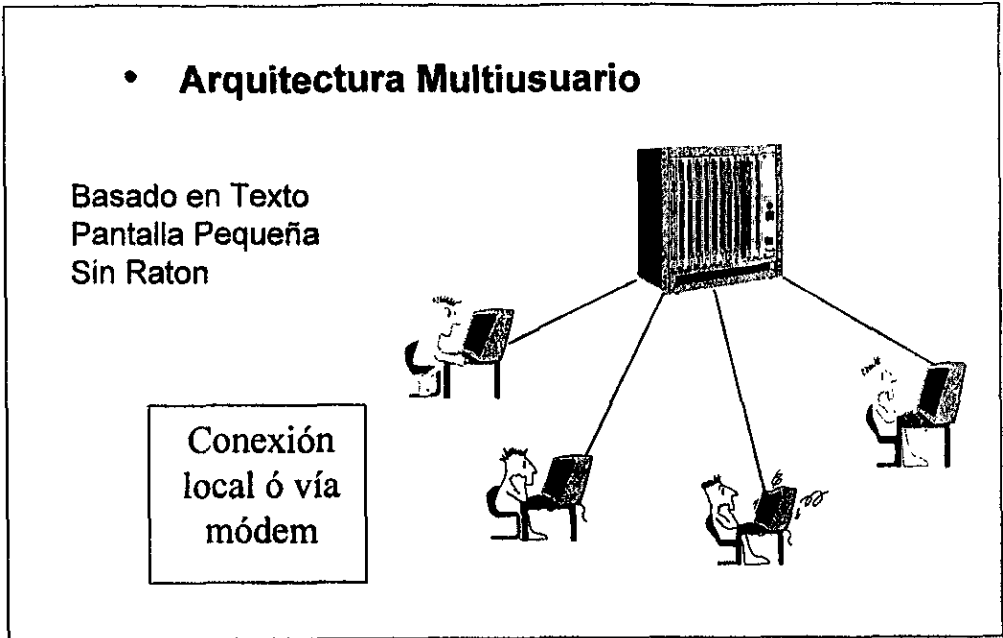


Figura 2.1.1 Intranet y Sistema Multiusuario. Aplicación accesible sólo en oficinas de la propia empresa ó por medio de un módem.

Los Ingenieros de Sistemas de Preventa tienen diversas actividades entre las cuales está la de realizar instalaciones, presentaciones, demostraciones y **benchmarks** (pruebas de rendimiento) en las instalaciones del cliente, por lo que muchas veces puede pasar un período relativamente largo (una semana) en el cual no se presentan en las instalaciones de la compañía ó lo tengan que hacer para capturar las actividades que han realizado a lo largo del día. Otra causa por la que no se acude a las oficinas es porque también se requiere realizar visitas a clientes fuera de la cd. de México, por lo que el registro de las actividades realizadas en provincia se retrasa hasta que el personal regresa a la oficina.

El hecho de registrar las actividades después de tiempo atrasa a la gerencia para la toma de decisiones debido a que el Sistema les proporciona datos que son utilizados para medir el desempeño del personal que aquí labora.

Por lo anteriormente señalado es deseable que se pueda tener acceso al Sistema desde otro lugar que no sean las oficinas de la empresa, y sin la necesidad de contar con hardware extra como lo es el módem.

### Presentación, Obsolescencia, Flexibilidad.

Hoy en día se cuenta con una gran gama de herramientas que permiten el diseño y desarrollo de sistemas de información las cuales ofrecen un producto de fácil uso y que necesitan muy pocos recursos para su mantenimiento. Al decir fácil uso nos referimos a que la aplicación adquiere todas las comodidades y ventajas que ofrece una aplicación que es ejecutada en modo gráfico. El ambiente gráfico (conocido como GUI -Graphical User Interface-), ha sido un gran éxito debido a que facilita la utilización de los sistemas de cómputo y es más rápido de comprender por los usuarios finales. Ver figura 2.1.2.

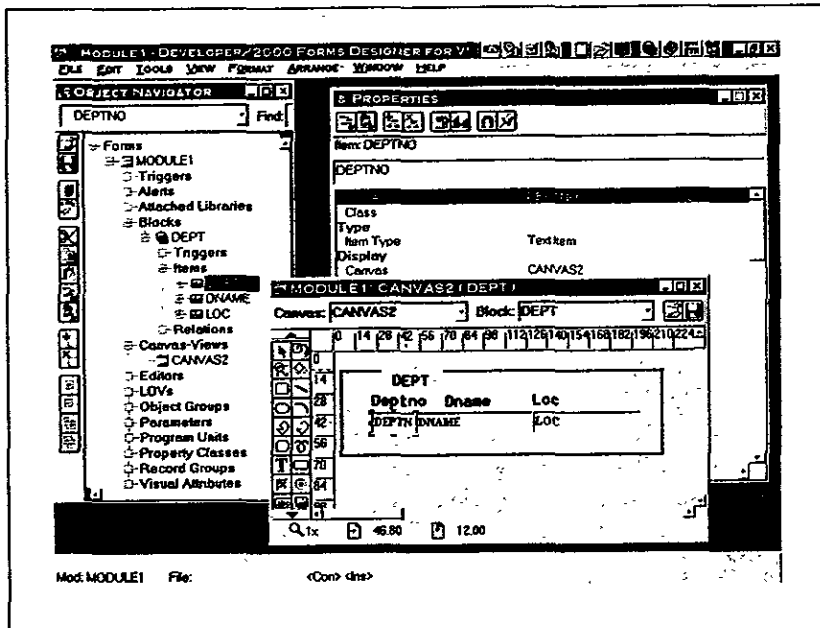


Figura 2.1.2. Pantalla modo gráfico.

Actualmente el Sistema de Preventa y Canales alternos se ejecuta en modo carácter. Ver figura 2.1.3.

Page 1 of 1  
Date 12/03/2006

**ORACLE de MEXICO**  
**Sistema de Control Ingenieria de Sistemas**

Informacion de las actividades realizadas por cada preventa.

Clave Preventa	05	Nombre	[REDACTED]	
Clave Actividad	11	No Cliente	000158	o No Distribuidor [REDACTED]
Fecha	06-01-06	Hora Actividad	18:00	Status 1
Lugar	[REDACTED]			
Comentario	[REDACTED]			

Plataforma/Herramienta	Horas Utilizadas
[REDACTED]	5

CLAVE DE IDENTIFICACIÓN DEL ESTADADO DE PREVENTA

Count: 2      ^ q      <List><Replace>

Figura 2.1.3. Pantalla modo carácter.

El sistema fue generado con productos que hoy en día son obsoletos y solo producían aplicaciones en modo carácter y que requieren mayor mantenimiento por lo que se complica el hacerle cambios o extensiones.

Además de lo anterior las versiones de las herramientas utilizadas para la generación de las formas y reportes están a punto de ser obsoletas. La desventaja de esto es que no se contará en un futuro cercano con soporte técnico para estos productos por lo que cualquier falla o problema detectado en el producto, como por ejemplo bugs (defectos del producto) ya no serán corregidos. Ver figura 2.1.4.

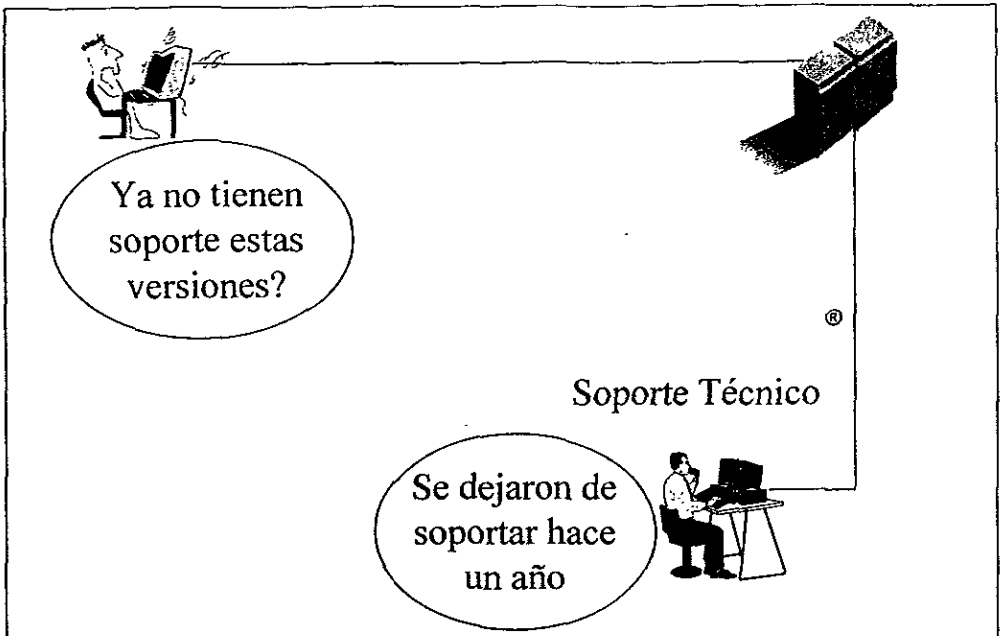


Figura 2.1.4. Obsolescencia de productos.

Para hacer uso del sistema se debe navegar entre las diferentes opciones que proporciona el menú principal, la desventaja de esto es que no se tiene una visión general de lo que es el sistema por lo que se tiene que acudir al manual para conocer en primera instancia el funcionamiento del mismo y sus capacidades. Este problema es resuelto al migrar el sistema a un ambiente gráfico que facilita al usuario la operación del mismo y su rápida comprensión.

#### **Información Insuficiente.**

El sistema actual contempla una serie de pantallas de captura desarrolladas en Forms 3.0 y un conjunto de reportes desarrollados en *Sql\*ReportWriter 1.1* que son llamados desde las formas. Este conjunto de reportes son utilizados para la toma de decisiones. En el tiempo que este sistema fue desarrollado, cumplía con las expectativas de la gerencia pero hoy en día como toda organización, Oracle ha cambiado algunas formas de operar y ha extendido sus actividades y procesos administrativos por lo que la información que se puede obtener del sistema ya no

cumple con las expectativas de la gerencia. Debido a lo anterior el sistema también se está convirtiendo en obsoleto respecto a las funciones que debe desempeñar.

## ORGANIGRAMAS



Figura 2.1.5. Organigrama Dirección Comercial/Preventa.

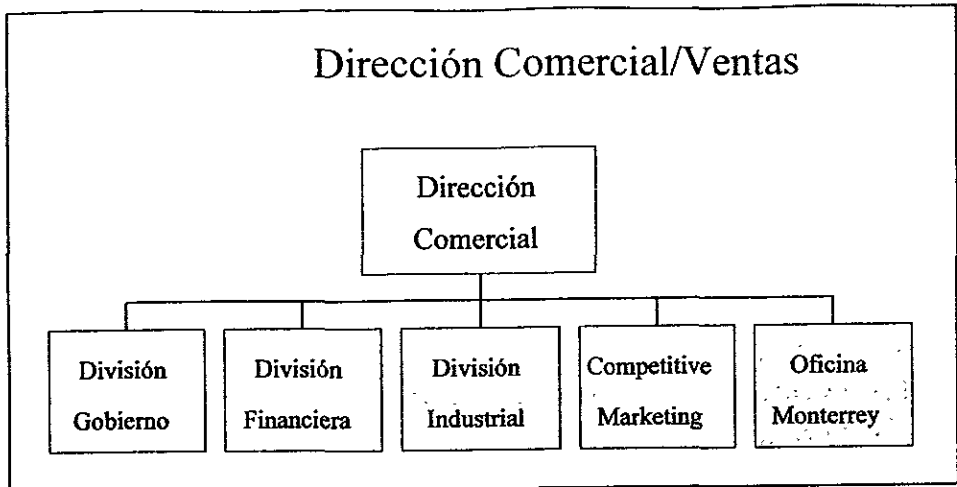


Figura 2.1.6. Organigrama Dirección Comercial/Ventas.

El personal de Oracle de México que utiliza el sistema pertenece a las áreas indicadas en las figuras 2.1.5 y 2.1.6.

Las desventajas del Sistema actual explicadas a lo largo de este capítulo se presentan para la gran mayoría del personal de las áreas de Ventas y Preventa.



## 2.2 ANÁLISIS Y CLASIFICACIÓN DE LA INFORMACIÓN

El proceso de análisis y clasificación de sistemas existentes es realizado cuando el objetivo del proyecto es reemplazar o mejorar una aplicación que ya esta en producción, que es el propósito de este trabajo: Mejorar el funcionamiento del Sistema de Información de Preventa y Canales Alternos, para lo cual se realizaron las siguientes tareas:

- Revisión de antecedentes. Se revisaron los procedimientos que seguían antes de tener el sistema actual, llegando a la conclusión que no había un sistema formal en el cual llevaran un control de actividades por preventa y no se sabía a ciencia cierta qué actividades se habían hecho con clientes. Cada Ingeniero de Sistemas llevaba, en el mejor de los casos, su propio control de actividades pero nunca había una integración de la información para hacer un análisis sencillo de la misma.
- Revisión del proceso de negocio actual y necesidades de información mediante la revisión de material de referencia existente y documentación del usuario, teniendo entrevistas con usuarios y sesiones de trabajo. Se tuvieron sesiones de trabajo con los usuarios a manera de determinar cuales son las necesidades de información y éstas se ven reflejadas en el punto 2.3 al que se le llamó requerimientos de usuario. En el apéndice de esta tesis se encuentra el manual de usuario de este sistema en el que se puede ver información de la funcionalidad del sistema así como también la presentación de los reportes generados en dicho sistema.
- Análisis de los sistemas a ser reemplazados con la revisión de documentación técnica, viendo los sistemas y entrevistando al personal técnico que le da mantenimiento. Se tuvieron pláticas con Jocelyn García que es la encargada de darle mantenimiento al sistema y dio la información sobre procedimientos operacionales usados por el sistema existente, así mismo, dio la documentación técnica, arquitectura técnica y volúmenes de información. En el apéndice se puede ver el manual técnico del sistema actual que se refiere a los tipos de datos utilizados por cada uno de los campos de las diferentes formas de captura y despliegue de información.

- Identificación y documentación de los problemas o áreas de oportunidad relacionados al proceso del negocio actual, necesidades de información y sistemas. Se identificaron algunas áreas de oportunidad que están reflejadas en el punto 2.3 de esta tesis.
- Asimilación de la información y capturarla en el repositorio de Designer/2000. Esto se va a realizar con la ingeniería en reversa que provee el mismo producto, la cual se desarrollará en el capítulo 3 de esta tesis.
- Confirmación de lo ya entendido mediante la retroalimentación y sesiones de revisión
- Redefinición del modelo Entidad-Relación.

## 2.3 REQUERIMIENTOS DE USUARIO

En este apartado se muestran los requerimientos del usuario con el fin de poder llegar a un sistema que efectivamente llene las expectativas actuales de los usuarios. En primera instancia se muestran las distintas personas que deben de tener acceso a diferentes funciones del sistema. En la siguiente tabla se representan los diferentes tipos de personas que podrán acceder el sistema :

ROL DESEMPEÑADO	DESCRIPCIÓN
ASISTENTE	Este rol es para aquellas personas que van a capturar información general del área de sistemas y se encargan de informar acerca de generalidades como los productos del área
BECARIO	Becarios que deberán capturar actividades que realizan
CANALES	Es para las personas que se encargan de este sector
COORDINADOR	Este rol es específico para los coordinadores de sectores del área de Ingeniería de sistemas Preventa
GERENTE	Para los gerentes del área de Ingeniería de sistemas Preventa
PREVENTA	Este rol es para que las personas que trabajan en el área de Ingeniería de sistemas Preventa puedan entrar al sistema

Figura 2.3.1 Roles

## Requerimientos Generales

Todas las pantallas deberán contener el número de forma o reporte que se está ejecutando, la fecha en español, el logo de la empresa y el nombre del usuario con que se está conectado a la base de datos. Cada usuario que pertenezca a un rol determinado solo verá las opciones a las que tiene acceso. Así mismo, es una propuesta del usuario que las pantallas se puedan ver a través de un navegador de Web.

Deberán existir listas de valores para captura y consulta, y se escogerá con ayuda del ratón la opción seleccionada, no se podrá teclear información en los campos que tengan listas de valores.

El rol de gerente deberá tener permisos de consulta sobre todas las tablas y desde el menú principal acceso a una herramienta para explotación de datos , que le permita obtener de manera fácil y rápida cualquier consulta requerida. Esta herramienta podrá generar archivos planos con la consulta o mandarlos imprimir. Toda la aplicación se deberá poder manejar con ayuda de un ratón. Los reportes que lo permitan deberán tener la posibilidad de mostrar gráficas. Se deberá tener acceso a una ayuda en línea desde cualquier módulo del sistema, esta ayuda mostrará información general del módulo desde el que se le invoca. El formato gráfico para las formas se muestra en la figura 2.3.2

ORACLE<sup>®</sup>  
Enabling Information age through network computer

Titulo de la forma

Usuario

Nombre módulo  
dd-mm-yy

Five black rectangular buttons are located at the bottom of the form.

Figura 2.3.2 Formato para formas

**HOME PAGE**

La página principal de la aplicación se muestra en la figura 2.3.3

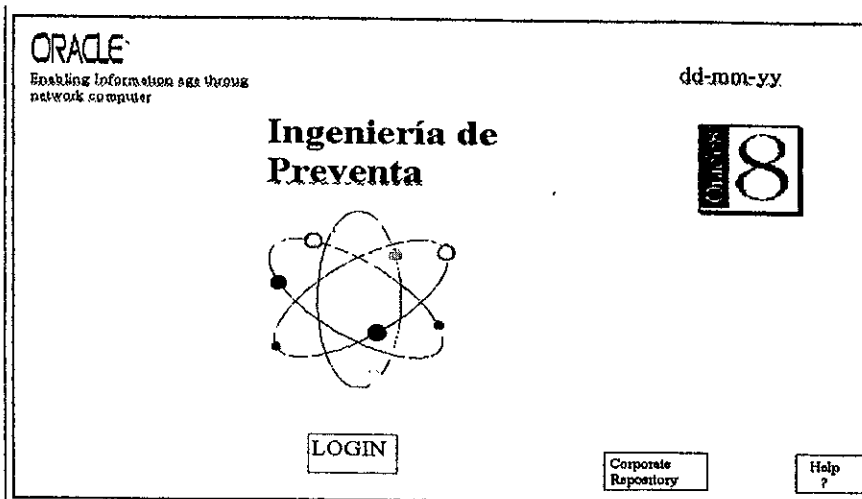


Figura 2.3.3 Home Page

Una vez que se hayan conectado con su usuario/password se desplegará la siguiente página que contiene el menú principal , y desde ahí podrán cambiar el usuario con el que están conectados a la base. El menú principal se muestra en la fig. 2.3.4

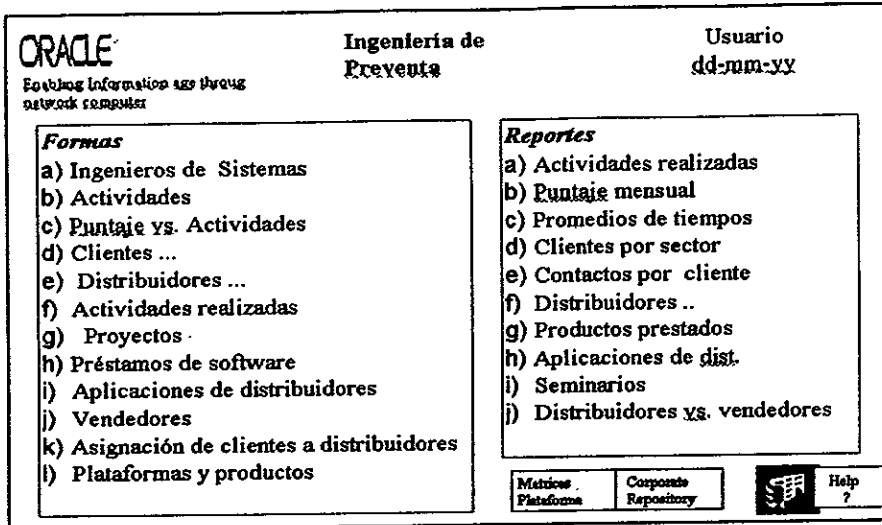


Figura 2.3.4 Menú Principal

### Navegación

Después de conectarse en el home page, se desplegará la pantalla que contiene el menú principal. Desde ahí se escogerá una forma o un reporte y desde el reporte una vez ejecutado se podrá regresar al menú principal. Desde el menú principal se tendrán dos ligas , una al home page del Corporate Repository y otra a alguna página del intranet de Oracle donde se encuentre información de versiones disponibles de los productos Oracle. Desde el mismo menú principal se podrá cambiar al usuario con que se está conectado.

Todos los módulos que se encuentran dentro de la aplicación actual se requiere que estén en la nueva versión de la aplicación, solo se desean algunos pequeños cambios , como por ejemplo juntar dos formas en una.

## Especificaciones Generales de Reportes

Deberán contener el número de página y la fecha de elaboración. Se podrán obtener salidas a pantalla, archivo o impresora. Deberá incluirse el logo de Oracle y el usuario con el que se conectó a la base de datos, como también el número de reporte que se está corriendo.

La pantalla de los reportes, deberá incluir scroll vertical y horizontal, así como también ayuda en línea. Los reportes que necesiten datos para poder ser ejecutados, deberán utilizar una pantalla de forma para la captura de datos de entrada. Por ejemplo la navegación de un reporte con parámetros se muestra en la figura 2.3.5

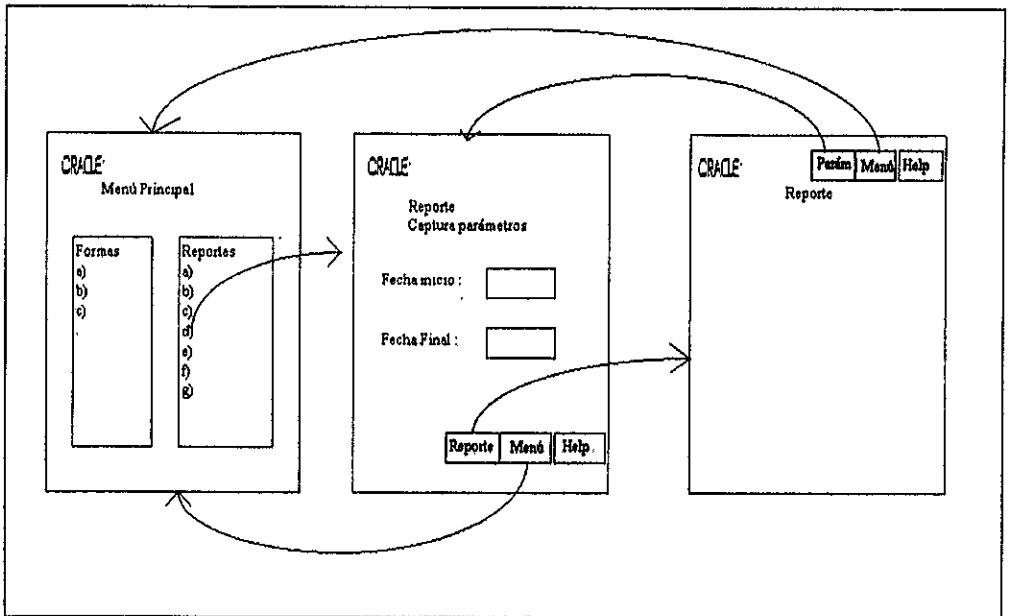


Figura 2.3.5 Navegación

El formato de los reportes se muestra en la figura 2.3.6

ORACLE  
Enabling Information Age through  
database computers

Título del reporte

Usuario  
dd-mm-yy

Menú  
Principal

Help  
?

Area de datos

Figura 2.3.6 Formato de reportes

En el caso de los reportes se requiere dar formatos mas generales y con ello se reduce el número de reportes solicitados, además como se contará con una nueva herramienta de explotación de la información, con la cual se pretende obtener todos los reportes adicionales.

## HELP

El help contendrá una breve descripción de todos los campos de las pantallas en el caso de pantallas de formas y una breve descripción del reporte en el caso de los reportes. Esta ayuda en línea ayudará a operar el sistema a los nuevos empleados. Además se requiere que se despliegue un mensaje explicativo de los campos de captura en las formas.

Por ejemplo si se va a capturar el número de radio localizador, que se despliegue el siguiente mensaje :

Número de radio localizador del Ing. de Sistemas \_\_\_\_\_



A continuación se hace una breve descripción de los módulos deseados :

### **Módulo de Datos Personales de los Ingenieros de Sistemas Preventa**

En esta opción se capturará toda información relevante del personal tal como nombre completo, dirección, dirección de correo electrónico, sector al que pertenece, localizador electrónico o celular (en caso de que cuente con cualquiera de ellos), puesto, perfil técnico (rdbms, herramientas, web, etcetera) , fotografía digitalizada del ingeniero de preventa, extensión telefónica, ubicación dentro de la oficina( piso, número de cubo).

### **Módulo de Actividades**

Para captura, modificación, borrado del catálogo de todas las actividades que puedan realizar los ingenieros de Sistemas.

### **Módulo de Puntaje**

Captura, modificación y borrado de los puntos asignados a cada actividad de acuerdo a la plataforma. Este módulo deberá estar enlazado como maestro detalle con el módulo de actividades.

### **Módulo de Captura de Clientes**

Captura y modificación de los datos de los clientes, incluyendo razón social, dirección, sector al que pertenece, teléfono, fax , home page (en caso de que cuente con ella) y en los casos que sea posible el logo del cliente.

### **Módulo de Contactos de Clientes**

Este módulo deberá estar enlazado al módulo anterior de tal manera que en la misma pantalla se pueda ver la información del cliente. Sirve para capturar, modificar y borrar contactos de las compañías, deberá incluir datos como nombre, teléfono, radio localizador, dirección electrónica, puesto que ocupa, área de influencia, fax.

### **Módulo de Distribuidores**

Captura y modificación de distribuidores, Los datos deberán ser razón social, nombre, dirección, home page (en caso de que cuente con ella), logo (opcional), rfc, tipo de canal distribuidor, tipo de mercado al que está enfocado, vendedor de Oracle que tiene asignado el distribuidor.

### **Módulo de Contactos de Distribuidores**

Este módulo deberá estar enlazado al módulo anterior de tal manera que en la misma pantalla se puedan ver la información del distribuidor. Sirve para captura, modificación y borrado de contactos de las compañías, deberá incluir datos como nombre, teléfono, radio localizador, dirección electrónica, puesto que ocupa, fax.

### **Módulo de Actividades de los Ingenieros de Sistemas**

En esta pantalla se debe poder capturar y modificar las actividades realizadas por los Ingenieros de Sistemas, debe incluir datos como clave del Ingeniero de Sistemas, cliente o distribuidor para el que se realizó dicha actividad, fecha de realización, estatus de la actividad, horas dedicadas, plataforma/herramienta que utilizó , el lugar y un campo para un comentario de 80 caracteres.

**Módulo de Proyectos**

Captura, modificación de proyectos. Algunos de los datos que se deben incluir en este módulo son cliente o distribuidor para el cual se realiza el proyecto o interno, alcance, objetivo, lugar de realización, estimación de tiempo necesario para cumplirlo y gentes asignadas a dicho proyecto.

**Módulo de Control de Préstamos de Software a Distribuidores**

Captura y modificación de la información para préstamo de software a distribuidores, deberá contener datos como número y nombre del distribuidor, clave de la plataforma a la cual pertenece el software, producto prestado, fecha de entrega, versión del producto, contacto que solicitó el préstamo, fecha de devolución, comentario.

**Módulo del Reporte de Actividades Realizadas para Clientes por Rango de Fechas**

En este reporte se podrá verificar todas las actividades y recursos que se han asignado a un cliente en un rango de fechas y también a todos las actividades y recursos asignados a todos los clientes para un rango de fechas.

**Módulo del Reporte de Actividades Realizadas para Distribuidores por Rango de Fechas**

Este reporte debe mostrar todas las actividades realizadas por los ingenieros de Sistemas para los distribuidores para un rango de fechas.

**Módulo del Reporte del Puntaje Mensual obtenido por un Ingeniero Preventa por Actividad**

El reporte mostrará información desglosada del puntaje que ha alcanzado un ingeniero. de Sistemas para un mes determinado.

**Módulo del Reporte Mensual de Promedios de Tiempo por Actividad**

El reporte debe mostrar los tiempos promedios por actividad realizada

**Módulo del Reporte Mensual de Horas Totales por Actividad**

Este reporte debe mostrar el total de horas por actividad por ingeniero.

**Módulo del reporte de Actividades Realizadas por los Ingenieros de Preventa en un Rango de Fechas**

Aquí se mostrarán todas las actividades que uno o todos los ingenieros de preventa han realizado para un período de tiempo determinado.

**Módulo de Reporte de Clientes por Sector**

Enlista todos los clientes de un sector específico

**Módulo de Reportes de Contactos por Cliente**

En este reporte se tendrán todos los contactos que están dados de alta para un cliente

**Módulo de Información sobre Plataformas**

Aquí se mostrarán las plataformas soportadas y deberá tener una liga hacia alguna página del intranet de Oracle donde se encuentren las matrices de compatibilidad de todas las plataformas soportadas. Se deberá poder imprimir o enviar por correo electrónico un documento específico de una matriz de compatibilidad.

**Módulo de Información de Productos**

Contendrá información de todos los productos soportados a la fecha y opcionalmente cada producto tendrá una liga hacia una pagina del intranet de Oracle donde se encuentre un documento de mercadotecnia, se deberá poder imprimir esta información o enviarla por correo electrónico. Se podrán dar de alta y modificar productos ya existentes.

**Módulo de Distribuidores**

Se podrán dar de alta y actualizar los datos de los distribuidores autorizados por Oracle, los datos que debe contener son razón social, rfc, dirección, teléfonos, fecha de ingreso como distribuidor Oracle.

**Módulo de Contactos de Distribuidores**

Se podrán dar de alta, baja o modificar los contactos para los distribuidores.

**Módulo de Productos Prestados a Distribuidores o Vendidos por Distribuidores**

Se darán de alta los préstamos de productos solicitados por distribuidores, así como también los productos vendidos a distribuidores.

**Módulo de Aplicaciones Desarrolladas por Distribuidores**

Se darán de alta y se podrán modificar todas las aplicaciones desarrolladas por distribuidores, se deberá incluir el tipo de aplicación, requerimientos necesarios para instalarse (software y hardware).

**Módulo de Hardware que pueden Vender los Distribuidores**

Se capturarán, modificarán o borrarán los diferentes equipos de cómputo que puede vender un distribuidor.

**Módulo de Asignación de Clientes a Distribuidores**

Se enlazarán los clientes que son atendidos a un distribuidor especificado

**Módulo de Productos Adquiridos por Clientes a través de Distribuidores**

Se darán de alta los productos que adquieren los clientes a través de distribuidores

**Módulo de Vendedores**

Se darán de alta, baja o modificación de los vendedores asignados a el área de canales alternos, se incluirán los mismos datos que los del módulo de datos personales de Ingenieros de Sistemas.

**Módulo de Reporte de Clientes por Sector**

En este reporte se obtendrán los clientes dados de alta por sector

**Módulo del Reporte de Distribuidores**

Se enlazarán todos los distribuidores autorizados de productos Oracle y el vendedor de Oracle que atiende a ese distribuidor

**Módulo de Reporte de Contactos por Distribuidor**

Aquí se mostrarán todos los contactos que están dados de alta para un distribuidor

hablando de cómo los integrantes del departamento de preventa pueden capturar sus actividades sin que les lleve mucho tiempo.

En consecuencia a tales exigencias, es necesario poner el sistema en un ambiente que proporcione flexibilidad en el acceso, que sea amigable a través de una interfaz gráfica y dejarlo en un esquema de procesamiento descentralizado para un mejor rendimiento de la aplicación, para lo cual se puede, en primera instancia, pensar en poner el entorno del mundo Internet que proporcione todas las características deseadas actualmente por el usuario para el sistema.

Analizando los equipos de cómputo disponibles dentro de la empresa para la realización del proyecto se cuenta con un servidor Sun SPARC con sistema operativo Solaris 2.5; un servidor Sequent; equipos PCs, Laptops con procesador Pentium, 32 MB en RAM, 1 GB de espacio en disco duro y monitor superVGA color. Adicionalmente, se tiene un sistema de red con topología Ethernet y acceso a Internet.

Como recursos de software se tienen el manejador de base de datos Oracle7 en su versión 7.3, Oracle Web Server version 2.1, Designer/2000, Developer/2000 y navegadores Internet.

Teniendo como fundamento lo anterior se decidió incorporar la estructura de una Intranet y un esquema de tres capas, en la que se tiene un servidor de base de datos, un servidor de aplicaciones y un cliente. El servidor Sequent se utilizará como servidor de base de datos, el servidor Sun SPARC se utilizará para el manejo de la aplicación, clientes PCs con capacidad para trabajar con cualquier navegador en un ambiente gráfico, ameno y sencillo para el usuario.

El sistema que se desea cambiar se basará en el manejador de base de datos Oracle como servidor de datos. En el caso de la aplicación ésta será cambiada a páginas de Web dinámicas de acuerdo con lo requerido por el usuario, para lo cual se utilizará Developer/2000 sólo para la migración de la misma, Designer/2000 para hacer la ingeniería en reversa de los módulos y datos; y posteriormente generar el sistema en HTML, así como para poder obtener la documentación de la misma. Ver figura 2.4.1

Adicionalmente y de común acuerdo con el usuario se llegó a la conclusión de que en una etapa posterior se implementarían los siguientes cambios a la aplicación:

- Despliegue del número de forma o de reporte en las pantallas
- Despliegue del nombre del usuario en toda la aplicación
- Los reportes que lo permitan deberán de mostrar gráficas
- Acceso a la ayuda en línea desde cualquier módulo del sistema
- Fecha en español
- Desde una pantalla de logon podrá cambiar el usuario
- Liga al Repositorio Corporativo de Oracle
- Liga hacia información de versiones de Oracle
- Fecha de elaboración de reportes
- Los reportes que tengan salida a un archivo
- Impresión del nombre del usuario que ejecutó el reporte
- Número de reporte
- Ayuda en línea para reportes
- Añadir a la base de datos para el ingeniero de sistemas: dirección de correo electrónico, sector al que pertenece, localizador electrónico o celular, puesto, perfil técnico, foto, extensión telefónica, ubicación dentro de la oficina; para el cliente: radio localizador, dirección electrónica, puesto, área de influencia; y para el distribuidor: el logotipo de la empresa
- Desde el módulo sobre plataformas. Imprimir o enviar por correo electrónico la matriz de compatibilidad de productos Oracle
- Liga hacia Mercadotecnia



También por consenso se determinó que los siguientes puntos no se verán reflejados en la aplicación:

- En los campos de lista de valores no se podrá teclear información. Este punto no se tomará en cuenta en sistema pues en ocasiones es más sencillo teclear directamente la información y no abrir una lista de valores
- La ayuda, que en el apartado anterior aparece como HELP. Pues se considera que es suficiente con el manual del usuario

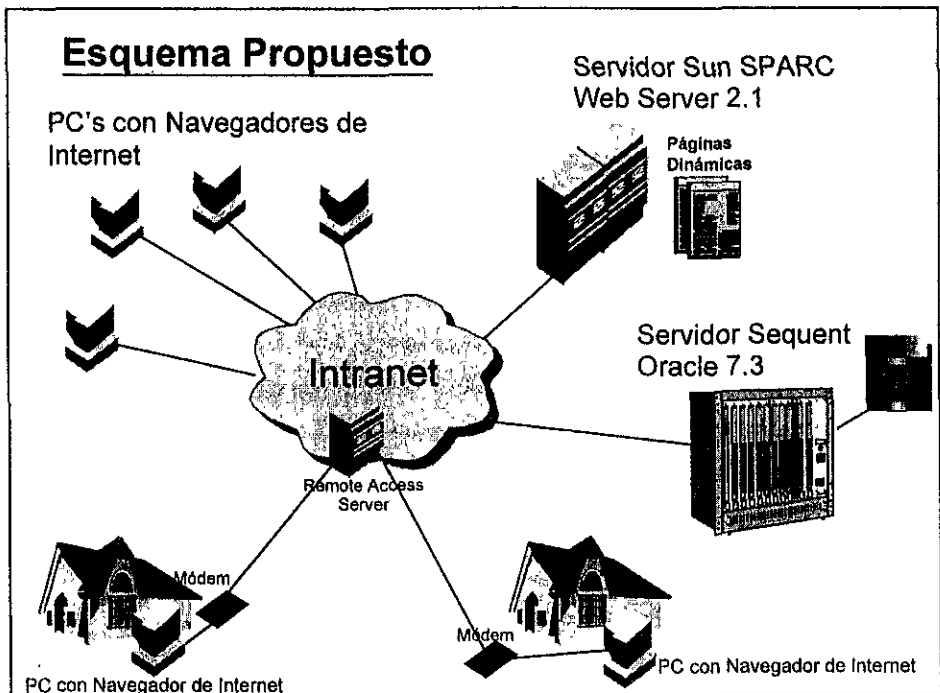


Figura 2.4.1 Esquema que se pretende alcanzar.

## 2.5 PLAN DE TRABAJO DE LA SOLUCIÓN PROPUESTA

En esta parte se explicará el plan elaborado para la consecución del objetivo de este trabajo. Colocando en primera instancia la lista detallada del software que se deberá instalar de acuerdo a los equipos de cómputo disponibles para este fin.

Se listarán las actividades que se deberán desempeñar y los tiempos que llevará cada una de ellas con el fin de establecer el tiempo total de desarrollo del plan propuesto.

### Productos de Software que se deberá Tener Instalados

- Sistema Operativo
- Oracle7 Server Versión 7.3.3
- Oracle WebServer Versión 2.1
- Designer/2000 Versión 1.3.2
- Developer/2000 Versión 1.3.2, que incluye:
  - Oracle Forms
  - Oracle Reports
- Sql\*Net V2.3

### Actividades

- Instalación del Sistema Operativo de la máquina Sequent.
  - Configuración de parámetros del protocolo de comunicación TCP/IP.
- Instalación del Manejador de Base de Datos (Oracle7 Server) en el Servidor Sequent
  - Configuración del “listener”.

- Instalación de la herramienta CASE (Designer/2000) en el Servidor Sequent
  - Creación del “repositorio” de Designer.
- Instalación del Oracle Web Server en el Servidor SUN
  - Configuración de parámetros.
- Conectividad con el equipo Pyramid que actualmente mantiene la aplicación. (MXPYR2)
- Migración de Formas de versión 3.0 a 4.5
- Migración de Reportes de versión 1.1 a 2.5
- Ingeniería en reversa de la Base de Datos.
- Ingeniería en reversa de la Aplicación.
- Regeneración de la Aplicación.
- Diseño y desarrollo del Front End.
- Integración, pruebas y depuración del sistema.

A continuación se presentan las actividades con las fechas establecidas.

PLAN DE ACTIVIDADES			
Num	Actividad	Inicio	Fin
1	Instalación del Sistema Operativo	20-AUG-97	26-AUG-97
2	Instalación del Manejador de Bases de Datos	23-AUG-97	26-AUG-97
3	Instalación de la Herramienta CASE (Designer)	26-AUG-97	03-SEP-97
4	Migración de Formas de version 3.0 a 4.5	27-AUG-97	03-SEP-97
5	Migración de Reportes de version 1.1 a 2.5	01-SEP-97	04-SEP-97
6	Instalación del Oracle Web Server	03-SEP-97	06-SEP-97
7	Ingeniería en reversa de la Base de Datos	05-SEP-97	09-SEP-97
8	Ingeniería en reversa de la Aplicacion	09-SEP-97	12-SEP-97
9	Regeneración de la Aplicacion	12-SEP-97	16-SEP-97
10	Diseño y desarrollo del Front End	16-SEP-97	20-SEP-97
11	Integración, pruebas y depuración del Sistema	20-SEP-97	25-SEP-97

Figura 2.5.1. Plan de Actividades.

A continuación se presenta el diagrama de Gantt con la serie de actividades antes mencionadas:

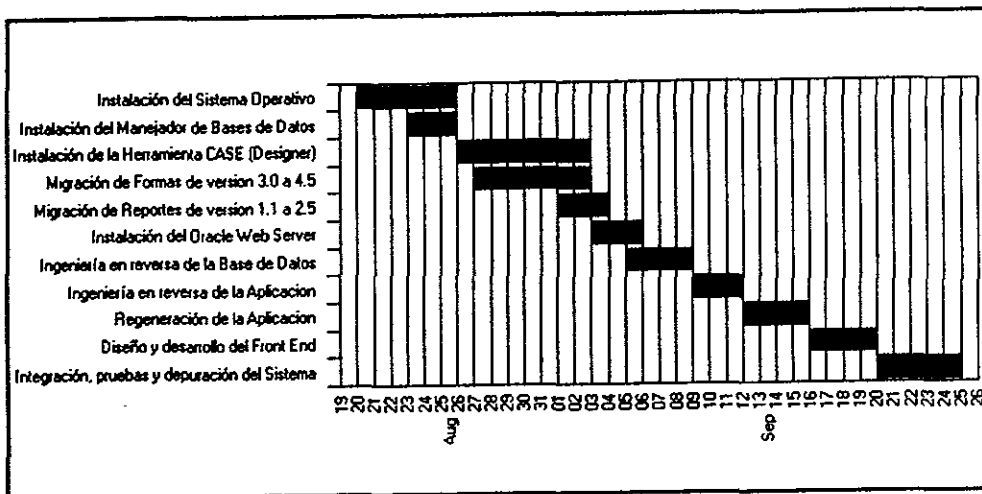


Figura 2.5.2. Diagrama de Gantt,

---

## **Capítulo 3**

### **Diseño e Implementación del Sistema**

### 3.1 ESPECIFICACIONES DE DISEÑO Y DIAGRAMA DE FLUJO GENERAL

A continuación se mencionarán las especificaciones de diseño que se pretende implementar como propuesta de desarrollo. Estas especificaciones fueron elaboradas en base a los requerimientos de usuario y al alcance de la implementación que se pretende obtener.

#### Arquitectura

Para brindar una mayor flexibilidad en el acceso, desempeño en tiempo de respuesta y facilidad en la administración se pretende implementar una arquitectura de 3 capas debido a las grandes ventajas que ofrece tener un servidor dedicado al almacenamiento y procesamiento de la información, otro servidor para la ejecución de la aplicación y finalmente equipos PCs o Laptops desde las cuales se logre acceder al sistema de información independientemente del lugar donde se encuentre un usuario tomando las ventajas que ofrece una conexión a Internet.

A continuación se mencionarán las características que tendrán los equipos de cómputo y el software requerido para implementar una arquitectura de tres capas.

#### Servidor de Base de Datos

- Servidor Sequent Modelo SE Serie 5000
- Sistema Operativo Dynix/PTX 4.1
- 256 Mb en RAM
- 20 Gb de espacio en disco duro
- Manejador de Bases de Datos Relacional ORACLE versión 7.3
- SQL\*Net versión 2.3
- Protocolo TCP/IP

**Servidor de Aplicaciones**

- Equipo Sun SPARCserver-1000
- Sistema Operativo Solaris 2.5.1
- 512 Mb de RAM
- 48 Gb de espacio en disco duro
- Webserver versión 2.1
- SQL\*Net versión 2.3
- Protocolo TCP/IP

**Equipos PC y Laptops**

- Sistema Operativo Windows 95 ó Windows NT 4.0
- Procesador 486 o superior
- 8 MB de RAM
- 300 Mb de espacio en disco duro
- Monitor Super VGA a color.
- Navegador de Internet
- Tarjeta de Red
- Módem
- Protocolo TCP/IP

**Sistema de Red**

- Topología Ethernet
- Acceso a Internet

Esta arquitectura es representada por la figura 3.1.1

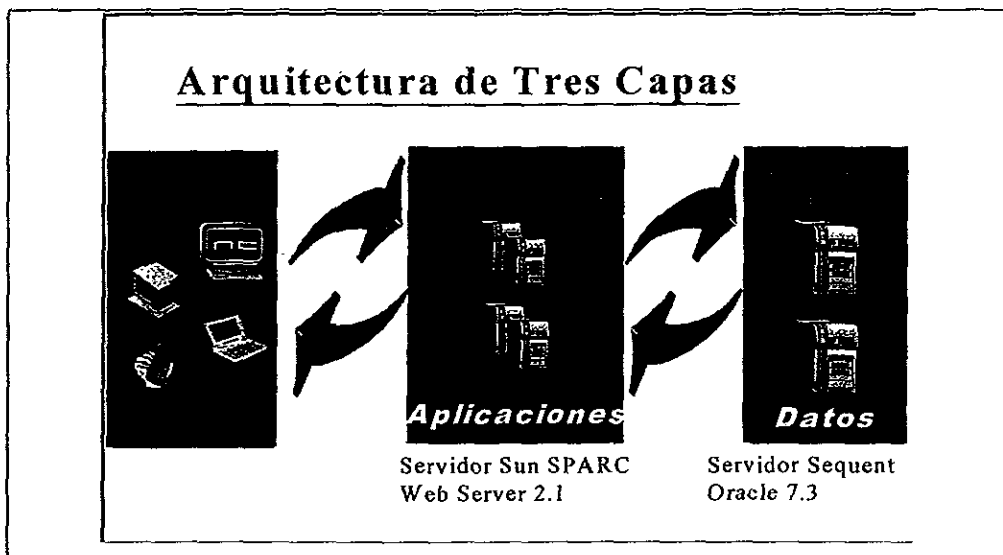


Figura 3.1.1 Arquitectura de Tres Capas

### **Requerimientos de Usuario**

A partir de la especificación de los requerimientos de usuario se ha concluido realizar las siguientes extensiones a la aplicación.

### **Especificaciones de Diseño Generales**

Todas las pantallas incluirán el título de la página de Web que se está ejecutando, se incluirá así mismo el logo de la empresa en la parte superior izquierda, el nombre del usuario de la base de datos que está realizando las consultas o captura de datos será colocado en la parte superior derecha, la fecha en el formato dd-mm-yyyy será colocado en la parte superior derecha inmediatamente debajo del nombre del usuario y constará de un campo dinámico que estará siendo refrescado por la fecha del sistema. Las pantallas de captura incluirán botones de "Aceptar" o de "Deshacer" para darle la facilidad al usuario de aceptar los cambios hechos a los



datos o en caso de ser necesario de no hacerlos permanentes, la navegación dentro de las pantallas será proporcionada por barras de scroll y para la navegación a otras pantallas será realizado por ligas, lo anterior podrá ser accesado con la ayuda del ratón o del teclado. El sistema de ayuda consistirá en mostrar leyendas breves pero concisas en la parte inferior de cada página explicando al usuario el propósito y función de cada campo cada vez que son accedidos.

El formato general para las páginas de Web de captura o despliegue de información será el siguiente:

ORACLE®  
Enabling Information Age Through  
Network Computing

Titulo de la página

Usuario  
Nombre del Módulo  
dd-mm-yyyy

Zona de despliegue de los campos de  
captura o despliegue de información

Figura 3.1.2 Formato General para las páginas de Web del Sistema.

Debido a la función primordial que desempeñan los Ingenieros de Preventa se incorporarán las siguientes extensiones dentro de la aplicación. Se definirá de manera completa el nombre de cada Ingeniero de Preventa, se especificará de manera precisa su domicilio actualizado, su correspondiente dirección de correo electrónico o e-mail de la empresa, puesto que desempeña, perfil técnico, sector al que pertenece, extensión telefónica, localizador electrónico o celular, ubicación dentro de la oficina (número de piso y número de cubo).

El módulo de contactos de clientes tendrá un enlace directo con el módulo de captura de clientes lográndose desplegar información de ambos módulos en la misma pantalla. Información de contactos de clientes incluirá datos como el nombre del contacto, puesto que desempeña, teléfono(s), localizador electrónico o celular, dirección electrónica, área de influencia y número de fax.

Por otro lado respecto al módulo de contactos de distribuidores se ha considerado definir las siguientes especificaciones. Tendrá un enlace directo respecto al módulo de distribuidores de tal forma que se desplegará información de ambos módulos en la misma pantalla, se incluirán datos como el nombre de los contactos, puesto que desempeñan, teléfono(s), localizador electrónico o celular, dirección electrónica y número de fax.

Los restantes módulos de la aplicación serán definidos como el resultado de la migración del sistema de Información al ambiente de Web esto por considerar que cumplen con las características de diseño y de información requeridas.

### ***Identificación de Roles Existentes***

Los roles que se pueden identificar de acuerdo a las actividades que realizan los usuarios de la base de datos y el tipo de acceso a la información que requieren son los siguientes :

#### **a) Gerente**

Este rol tendrá acceso total al sistema de la aplicación logrando realizar solo consultas a la información. No tendrá la facilidad de realizar cambios a la estructura de las tablas y objetos existentes dentro de la base de datos. Contará con una herramienta de explotación de consultas no planeadas para recuperación de información en una forma sencilla y rápida, para la generación de reportes sencillos a archivos planos o a la impresora, tendrá la facilidad de generar gráficas; ésta herramienta le ayudará en la toma de decisiones.

**b) Administrador**

Este rol está orientado al usuario de la base de datos que requiere tener control total de las estructuras de los objetos de la base de datos así como también para realizar modificaciones a la aplicación cada vez que éstas sean necesarias. Podrá tener acceso total al sistema de la aplicación. Su labor principal será la de proporcionar servicios de mantenimiento tanto a la base de datos como a la aplicación. Estará también entre sus funciones el llevar el control de los productos existentes en el inventario así como catálogos en general.

**c) Preventa**

Los usuarios quienes cumplen con la labor de Ingenieros de Preventa podrán tener privilegios de consulta de información y de captura para cada una de las pantallas que requieran de acuerdo a la labor específica que deban cumplir, entre ellas, la documentación de las actividades realizadas a los clientes.

Cabe señalar que estos roles tienen una correspondencia directa con los privilegios que se otorgan dentro de la base de datos.

**Interface de la Aplicación**

La aplicación contará con una interface gráfica como resultado de trabajar en un ambiente de Web esto representará una ventaja importante debido a que interfaces de este tipo son lo suficientemente amigables para lograr que cualquier usuario pueda aprender a utilizar una aplicación en un período de tiempo breve.

La interface de la aplicación con el usuario constará de dos pantallas, la primera de ellas realizará la validación al sistema solicitando para ello un "username" y un "password", la segunda

contendrá el ambiente de navegación de la aplicación y el despliegue de las páginas de Web de captura de datos y de información.

A continuación se presenta la pantalla que servirá como interface de validación al sistema.

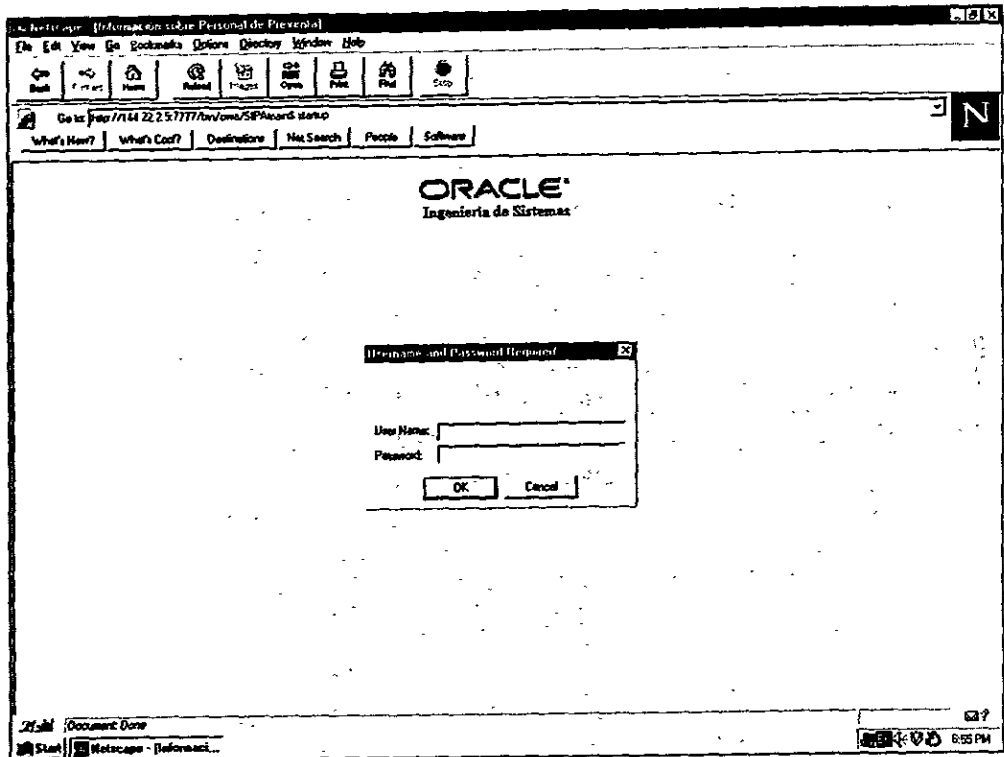


Figura 3.1.3 Interface propuesta de Validación al Sistema.

La pantalla de navegación del sistema ha sido diseñada de tal forma que permite al usuario elegir una opción, ya sea de captura o de consulta, y contar en todo momento con una perspectiva general del sistema. Para lograr esto se ha implementado el uso de frames, esto evita que el usuario se pierda en el contexto de la aplicación y la actividad de navegación dentro de las diferentes opciones se vuelva algo caótico y difícil de manejar. La interface mostrará en la parte izquierda de la pantalla el conjunto de menús de acceso a las páginas de Web de captura y de

consulta, y el conjunto de ligas a otros servidores de WEB de Oracle Corporation, todo estará contenido en un frame, en la parte superior de la pantalla se colocará un frame que contendrá el logo de la compañía y el nombre de la aplicación, por último se incluirá un tercer frame que se encontrará en la parte central de la pantalla y que será utilizado para el despliegue de las páginas de Web de captura y despliegue de información.

A continuación se muestra el prototipo propuesto para la pantalla de navegación y de despliegue de las páginas de Web de captura de datos y de información.

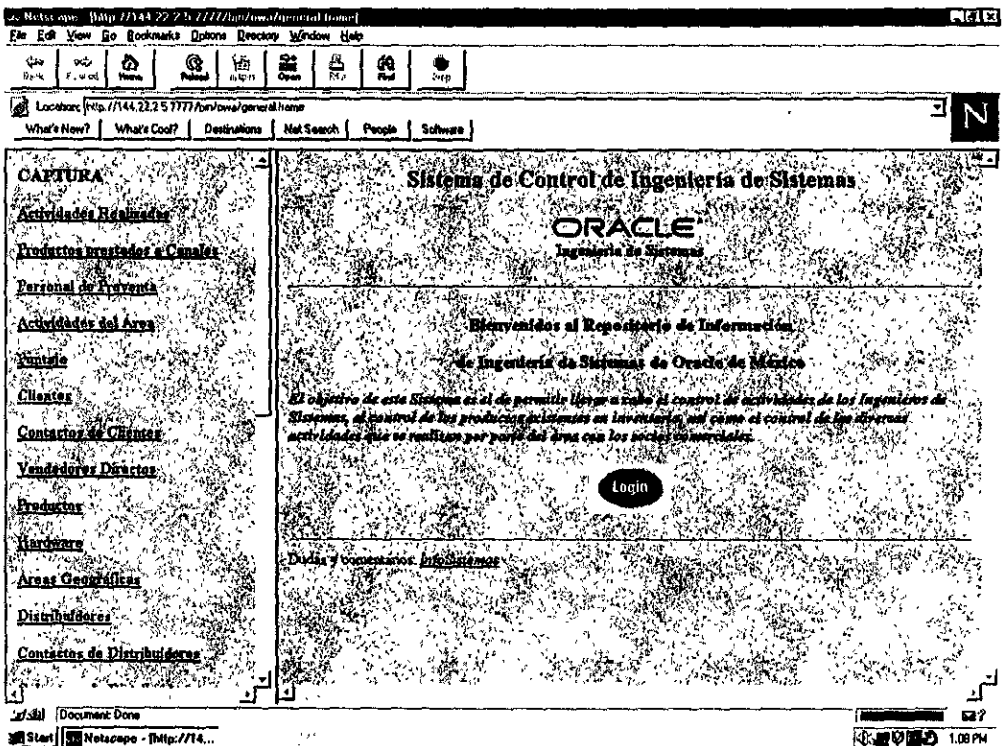


Figura 3.1.4 Prototipo de la Pantalla de Navegación y de despliegue de las páginas de Web.

Como podrá observarse la navegación y la percepción del contexto general resulta muy sencillo con el uso de frames ya que aún después de seleccionar una liga y de observar el resultado en la pantalla en todo momento se mantiene visible el resto de las opciones disponibles.

### **Diagrama de Flujo General**

Dada la simplicidad de la interface de la aplicación, el diagrama de flujo general del sistema es así mismo simple constando de solo tres niveles de profundidad, lo que nos habla de lo sencillo y funcional que resultará para cualquier usuario utilizar la aplicación.

Se puede observar de la siguiente figura que los niveles de los cuales consta el diagrama de flujo general son en un primer plano la pantalla de "Validación de Entrada al Sistema" este nivel nos permitirá ofrecer un método de seguridad al sistema autorizando entrar al usuario solo a la aplicación después de introducir tanto un USERNAME como un PASSWORD válidos. En un segundo nivel se puede apreciar los menús principales de los cuales constará la aplicación, estos son el menú de "Preventa", el menú de "Canales Alternos", el menú de "Reportes" y por último el menú de "Varios". Finalmente el tercer nivel del diagrama consta de las opciones disponibles para cada uno de los menús anteriores, estos en si mismo representan las páginas de Web de captura y de despliegue de información desde las cuales el usuario podrá ingresar información relevante al sistema y realizar consultas de la información disponible.

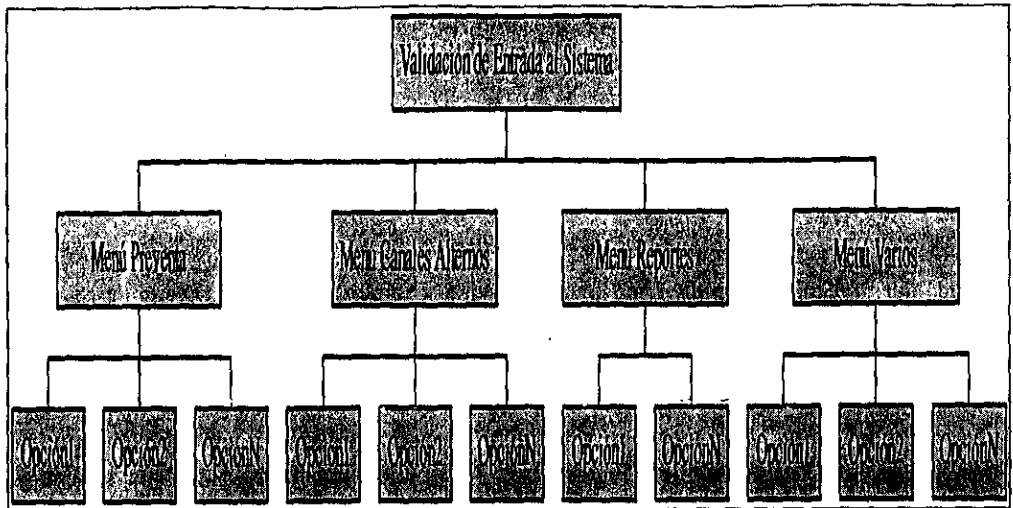


Figura 3.1.5 Diagrama de Flujo General.

A continuación se mencionan las opciones disponibles para cada uno de los menús.

### Menú Preventa

- Actividades realizadas por cada Preventa
- Información del Personal de Preventa
- Actividades del Area
- Información sobre Puntaje
- Información de Clientes
- Información sobre Contactos de los Clientes
- Vendedores de otros Sectores
- Productos

### Menú Canales Alternos

- Productos prestados o vendidos a Distribuidores

- Información de Clientes
- Información sobre Contactos de los Clientes
- Información de Hardware
- Areas Geográficas
- Información de los Distribuidores
- Contactos de Distribuidores
- Areas supervisadas por Distribuidores
- Aplicaciones hechas por Distribuidores
- Hardware que pueden vender los Distribuidores
- Clientes atendidos por Distribuidores
- Proyectos hechos a Clientes por Distribuidores
- Productos adquiridos por Clientes a través de Distribuidores
- Vendedores de Canales Alternos

### **Menú Reportes**

- Reporte General/Detallado de Contactos de Distribuidores
- Seminarios tomados por los Contactos
- Reporte de los Distribuidores
- Distribuidores por Tipo y Area
- Proyectos hechos por Distribuidores a Clientes
- Aplicaciones realizadas por Distribuidores
- Distribuidores atendidos por Vendedores
- Puntaje obtenido por Actividad
- Promedios de Tiempo por Actividad
- Horas totales por Actividad
- Actividades Realizadas por los Ingenieros de Sistemas de Preventa
- Actividades Realizadas a Clientes
- Actividades Realizadas a Distribuidores



- Clientes que pertenecen a un Sector específico
- Detallado de Contactos de Clientes

### **Menú Varios**

- Home page del Corporate Repository de Oracle <http://crweb.us.oracle.com>
- Home page de Oracle Corporation <http://www.oracle.com>
- Home page de Oracle Inglaterra <http://www-sup.uk.oracle.com>
- Home page de Oracle México <http://mxseq2.mx.oracle.com>

## 3.2 INGENIERÍA EN REVERSA DE LA BASE DE DATOS

La Ingeniería en Reversa permite que aplicaciones ya existentes se actualicen a herramientas actuales y además para aquellas aplicaciones que *no* se encuentran documentadas, se tenga la documentación necesaria.

Al proceso realizado por la Ingeniería en reversa se le conoce también como diseño 'bottom-up' y consiste en :

- Derivar las definiciones y módulos del mundo real
- Derivar el modelo del negocio de los datos después de la Ingeniería en Reversa
- Analizar las nuevas reglas del negocio y compararlas con las existentes en el sistema
- Rediseñar e implementar el nuevo sistema, si se requiere, para satisfacer las nuevas reglas del negocio
- Rediseñar e implementar el nuevo sistema a ambientes actuales, como el Web.

En nuestro caso el objetivo de pasar la aplicación con funcionalidad que tiene actualmente al WEB. Para poder hacer esto es necesario llevar a cabo una serie de pasos previos antes de comenzar con la Ingeniería en Reversa.

La base de datos de producción se encuentra actualmente en un equipo Pyramid con Unix, con el objetivo de poder realizar la ingeniería en reversa y pruebas del sistema se decidió migrar todos los objetos del usuario CANALES que es el dueño de la aplicación a un equipo NT con la versión 7.3

Las herramientas utilizadas para poder realizar este cambio de plataforma fueron las utilierías de Export e Import.

Para exportar los datos :

```
exp canales/canales owner=canales file=/tmp/canales.dmp log=/tmp/canales.log
```

El archivo dmp se transfirió en formato binario al equipo NT y se procedió a crear al usuario :

```
sql> create user canales identified by sipa ;
sql> grant connect, resource to canales ;
sql> alter user canales default tablespace datos quota unlimited on datos ;
sql> revoke unlimited tablespace from canales ;
```

Con las instrucciones anteriores se tiene todo listo para importar sus objetos :

```
$ imp73 canales/sipa file=c:\tmp\canales.dmp full=y log=c:\tmp\canales.log commit=y
buffer=1000000
```

El total de objetos que se exportaron se muestra en la fig. 3.2.1

TIPO DE OBJETO	TOTAL
Indices	245
Procedimientos almacenados	3
Secuencias	13
Tablas	75
Vistas	13
Constraints de llave primaria	67
Constraints referenciales	137
Constraints de llave única	37

Fig. 3.2.1 Tabla de objetos en la aplicación de producción.

Todas estas definiciones son las que se van a subir a la herramienta Designer.

El procedimiento de la Ingeniería en reversa se muestra en la figura 3.2.2

En nuestro caso se van a tomar las definiciones de los objetos para subirlos al repositorio de Designer.

Como productos de salida de esta etapa se obtendrán :

- Script DDL de generación de todos los objetos que utilizará la aplicación
- Diagrama Entidad-Relación de todas las entidades que utiliza la aplicación
- Diagrama de datos que incluye todas las tablas de la aplicación

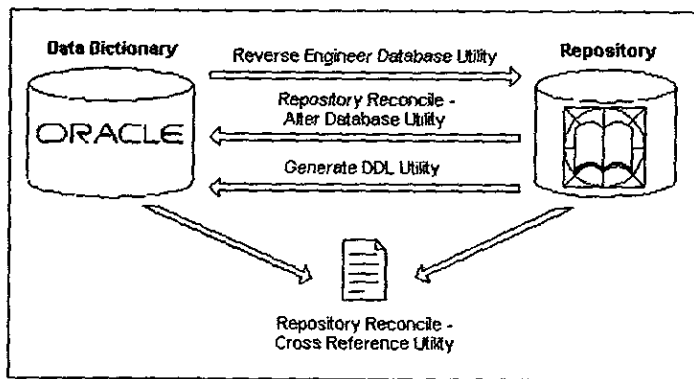


Fig 3.2.2 Utilerías de apoyo al hacer Ingeniería en Reversa a la Base de Datos.

El proceso de Ingeniería en Reversa incluye dos etapas.

## **INGENIERÍA EN REVERSA - ETAPA I**

Esta etapa puede incluir

- Ingeniería en Reversa de objetos de la base
- Creación de usos de módulos
- Ingeniería en Reversa de Oracle Forms
- Implementación de cambios y mejoras

Con esta etapa se completa la mayor parte del proceso de Ingeniería en Reversa. El sistema que hasta la fecha no ha sido controlado o documentado puede ahora documentarse. En el presente inciso no se cubrirá a detalle lo referente a módulos y Oracle Forms porque eso se presenta en el inciso 3.3.

### **• Ingeniería en Reversa de Objetos de la Base**

Para este paso se utiliza la herramienta Reverse Engineer Database de Designer para cargar las definiciones de objetos existentes en una base de datos Oracle al Repositorio de Designer. Las definiciones de objetos de la Ingeniería en Reversa pueden ser usados para identificar y aislar :

- Llaves primarias, únicas y foráneas
- Sub-tipos y super-tipos de entidades para definir una o mas tablas
- Datos desnormalizados
- Datos duplicados

Es conveniente que en este paso los usuarios finales sean consultados para ayudar al análisis. En este paso se establecen las convenciones a ser utilizadas para etapas posteriores del desarrollo.

Con el Repository Object Navigator de Designer, se abre la ventana de Reverse Engineer Form, ver la figura 3.2.3

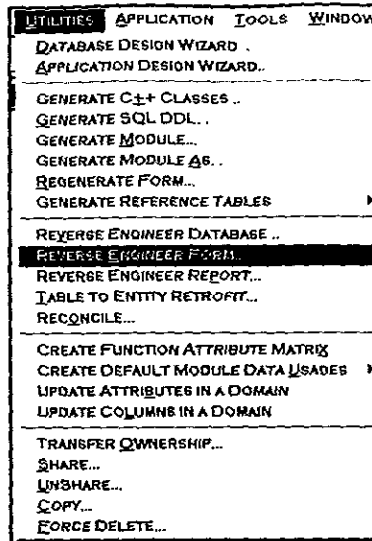


Figura 3.2.3 Ventana Utilities del Object Navigator.

Y la forma que aparece es la mostrada en la Figura 3.2.4, en la cual se proporcionan los datos del dueño de la aplicación que se va a subir al repositorio de Designer, se deben escoger todos los objetos.

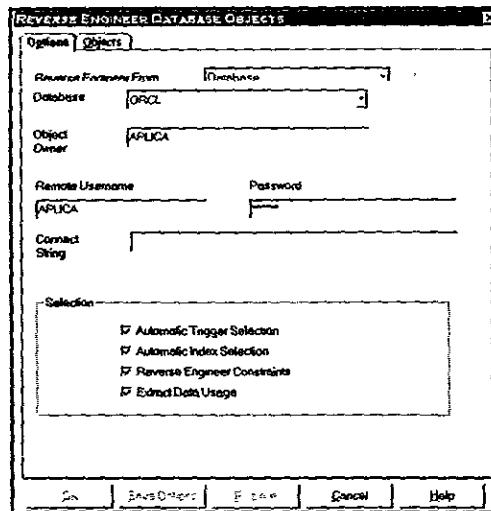


Figura 3.2.4 Wizard para Ingeniería en Reversa.

Una vez que se termina con este paso, desde el Object Navigator se pueden ver todos los objetos de la aplicación, tal y como aparece en las figuras 3.2.5 y 3.2.6.

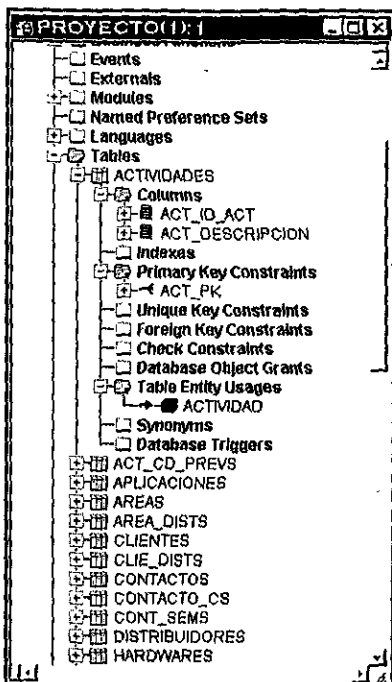


Figura 3.2.5 Pantalla con la lista de Tablas después de la Ingeniería en Reversa de la aplicación.

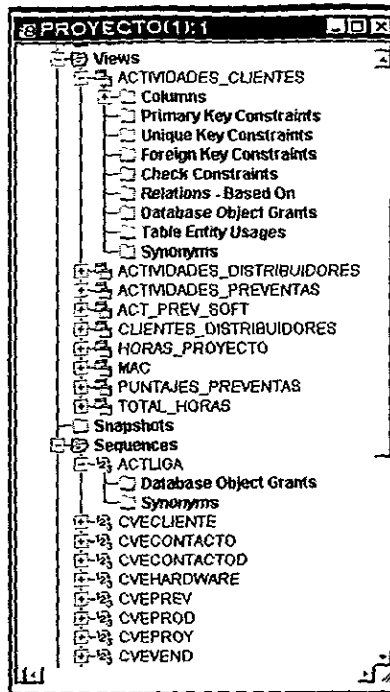


Figura 3.2.6 Pantalla con la lista de Vistas y Secuencias después de la Ingeniería en Reversa de la aplicación.

- **Implementación de Cambios y Mejoras**

El repositorio puede ser usado ahora para definir cambios y mejoras a la base de datos existente. Si la definición de nuevas tablas u otros objetos son creadas en el repositorio ellos pueden ser implementados generando el DDL con la utilidad de generación de DDL de Designer. Si el Repositorio y el diccionario de la base no coinciden se debe utilizar la opción de Referencias Cruzadas para tomar las acciones necesarias.

En el Sistema de Ingeniería de Preventa se decidió agregar 3 nuevos campos a las tablas de Ingenieros Preventas y Vendedores para almacenar su biper, piso en el que se encuentran y su extensión telefónica, después se generó el DDL de todos los objetos y se recrearon con este



DDL, con lo que el Repositorio y el Diccionario quedaron con las mismas definiciones. A continuación de esto se volvieron a cargar los datos de la base de producción para tener datos reales para la etapa de pruebas. Todo el proceso de generación de DDL se explica en el inciso 3.4. En la fig. 3.2.2 se muestra las utilerías que se usan para esta etapa de la Ingeniería en Reversa.

- **Creación de Usos de Módulos**

En este paso se definen las asociaciones entre módulos y tablas, vistas, snapshots, columnas para comenzar la documentación de referencias cruzadas para la etapa de implementación. Al subirse los módulos estas asociaciones quedan establecidas, pero es necesario revisarlas, este tema se trata a detalle en el inciso 3.3

- **Ingeniería en Reversa de Formas**

Este punto es útil para aplicaciones que están generadas en Oracle Forms y el módulo que se sube al repositorio. En el caso de la aplicación de Ingeniería de Sistemas está generada en Oracle Forms, por lo que se puede subir la definición de todos los módulos de Oracle Forms. ( Ver inciso 3.3)

## **INGENIERÍA EN REVERSA - ETAPA 2**

Este paso es opcional, pero provee una mejor documentación del sistema, puede incluir :

- Diagrama jerárquico de funciones
- Diagrama Entidad-Relación, subiendo las tablas a Entidades con la utilería Entity to Table Retrofit
- Revisar el modelo Top-Down del modelo de negocios con los datos de la Ingeniería en Reversa.
- Generar referencias cruzadas entre los nuevos niveles de datos y los ya existentes utilizando el Object Navigator.

### Utilería "Table to Entity Retrofit"

Se usa para generar las definiciones de entidades a partir de las tablas de la aplicación. La utilería genera entidades, atributos, relaciones e identificadores únicos de la siguiente manera:

- Las definiciones de las tablas son creadas a partir de las definiciones de las tablas.
- Los atributos son creados de las columnas no involucradas en constraints de llave foránea.
- Las relaciones son creadas a partir de las llaves foráneas de las tablas especificadas y de las tablas a las cuales se hace referencia.
- A las llaves primarias se le aplica Ingeniería en Inversa a identificadores de llave primaria.
- Los constraints de llave única se mapean a identificadores secundarios únicos.

La pantalla de la utilería se muestra en la figura 3.2.7

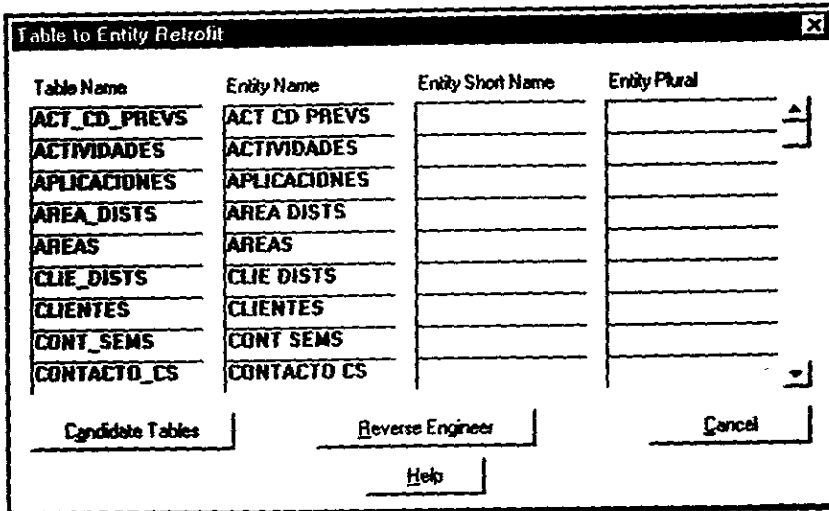
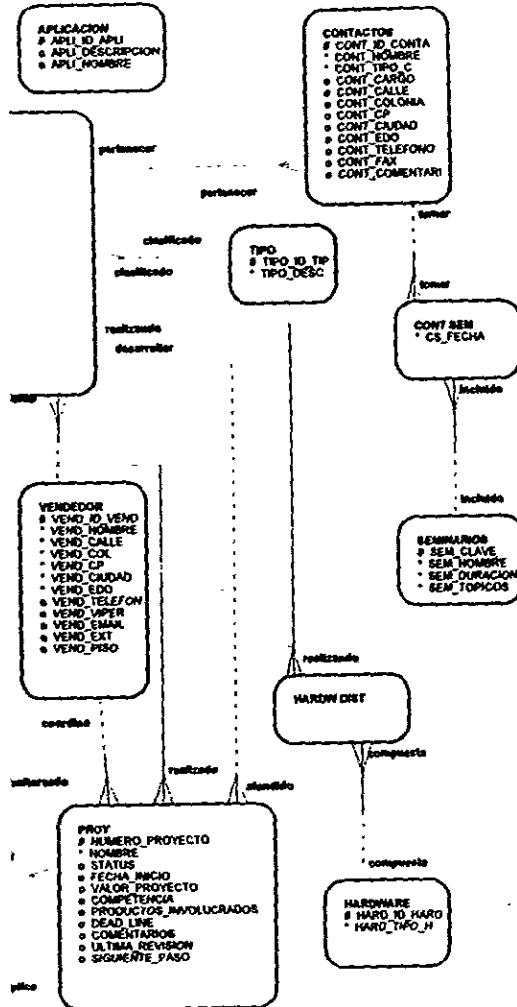


Figura 3.2.7 Pantalla para generar Entidades a partir de Tablas.

El diagrama Entidad-Relación y el Diagrama de Datos se muestran a continuación.





### 3.3 INGENIERÍA EN REVERSA DE LA APLICACIÓN

La Ingeniería en Reversa consiste en tomar una forma o reporte existente y crear o actualizar un módulo del repositorio de Designer.

En este apartado se detalla el proceso llevado a cabo para poder realizar la ingeniería en reversa, la cual consiste en cargar especificaciones de diseño de las formas y reportes al repositorio de “Designer” como módulos.

La ventaja principal de realizar la ingeniería en reversa es que todos los datos mantenidos en el Repositorio pueden ser utilizados para generar información acerca del sistema.

#### **Actualización de los Módulos a la Versión Requerida e Ingeniería en Reversa.**

Para poder llevar a cabo la Ingeniería en Reversa es necesario migrar tanto los Reportes como las Formas que conforman el Sistema.

#### ***Iniciaremos con la Migración e Ingeniería en Reversa de los reportes :***

Para poder realizar la ingeniería en reversa, se verificó que los reportes funcionaran bien en la versión de Sql\*ReportWriter 1.1.

Los reportes se encontraban almacenados en la Base de Datos; en teoría, la utilidad para realizar la Ingeniería en Reversa de los Reportes puede tomar como fuente reportes de la versión 1.1 de Sql\*ReportWriter.

La figura 3.3.1 muestra la ventana especificando lo mencionado en el párrafo anterior.

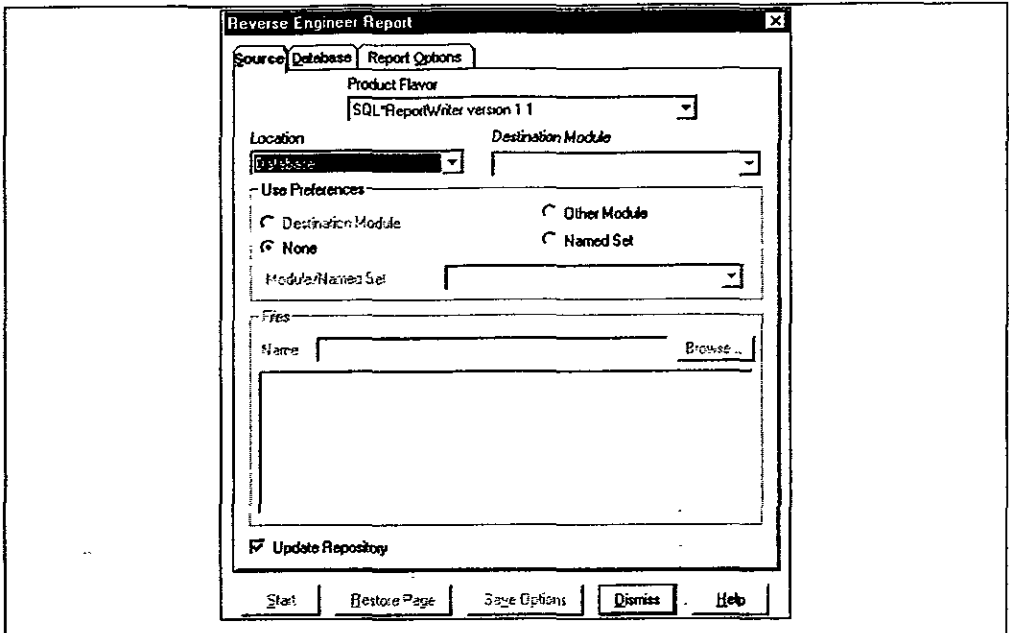


Figura 3.3.1 Ingeniería en reversa tomando como fuente los reportes almacenados en la Base de Datos.

Al indicar el nombre del reporte y el dueño del mismo en el folder de Database de la figura 3.3.1 e intentar realizar el proceso se obtuvo un mensaje de error mostrado en la figura 3.3.2

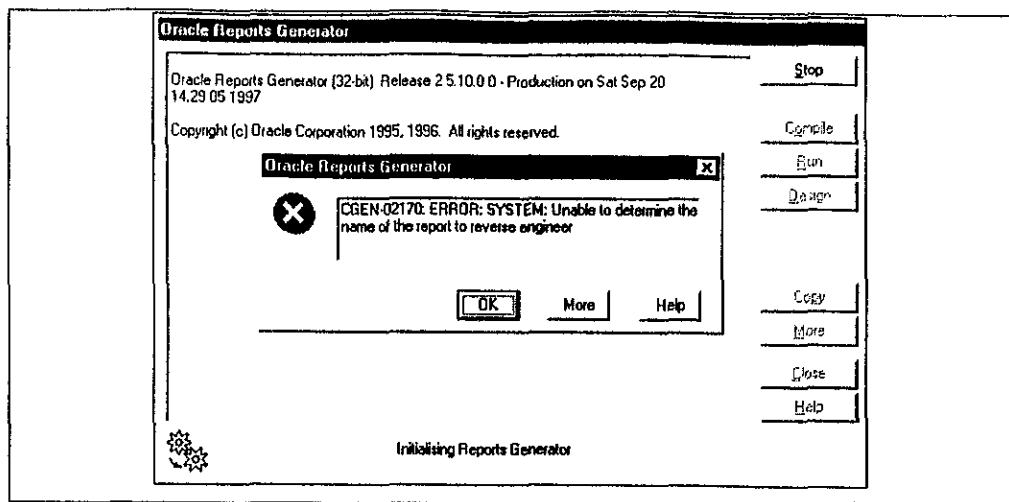


Figura 3.3.2 Error al intentar realizar la Ingeniería en Reversa.

Internamente los reportes deben mantener un nombre que el Generador (módulo que realiza el proceso de Ingeniería en Reversa) no encuentra, por lo que el proceso no se puede realizar.

Debido a lo anterior se deben migrar los reportes de la versión 1.1 de Sql\*ReportWriter a la versión 2.5 de Oracle\*Reports para posteriormente realizar la Ingeniería en Reversa.

Para migrar reportes de versión 1.1 a 2.5, se utiliza el ejecutable R25MOV32. Para utilizar esta herramienta es necesario saber los formatos de reportes en ambas versiones :

FORMATOS V1.1	FORMATOS V2.5
database	database (como en versión 1.1 pero las tablas son diferentes)
.rep	.rep (como en versión 1.1 código binario para ejecución, no contiene información de diseño)
.rex	.rex (texto, como en versión 1.1)
	.rdf (es binario, contiene información de diseño)

Figura 3.3.3 Formatos de los reportes en versiones 1.1 y 2.5.

La figura 3.3.4 ilustra las posibles rutas de migración.

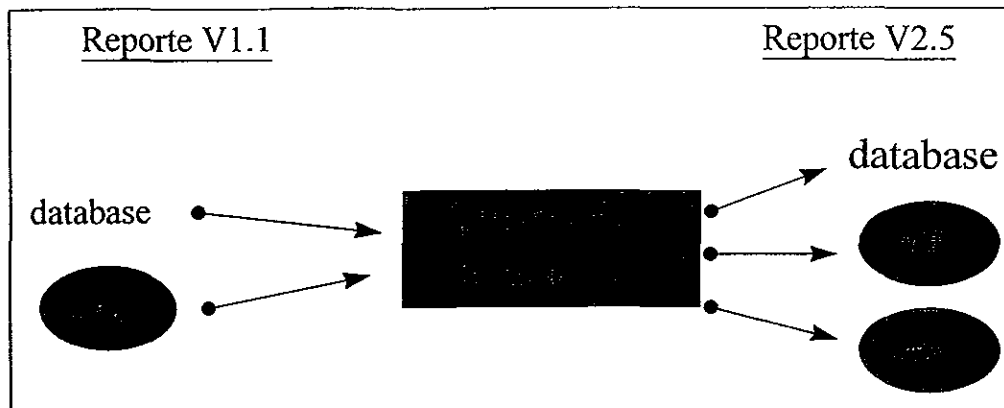


Figura-3.3.4 Rutas posibles de migración.

Se pueden migrar reportes V1.1 de la Base de Datos o .rex (la migración de reportes .rep no está soportada). Para decidir cuál es la ruta mas apropiada hay que verificar los siguientes puntos :

- La migración de archivos .rex es el método más rápido (debido a que la utilidad no accesa la Base de Datos por cada reporte), pero los caracteres comodines no pueden ser usados para especificar los reportes a migrar.
- La migración de los reportes en la Base de Datos es más lenta pero se pueden ocupar los caracteres comodines (% y \_) para especificar los reportes a migrar.

Se pueden migrar reportes mediante una caja de diálogo o desde la línea de comandos.



En la figura 3.3.5 se muestra la caja de diálogo para la migración.

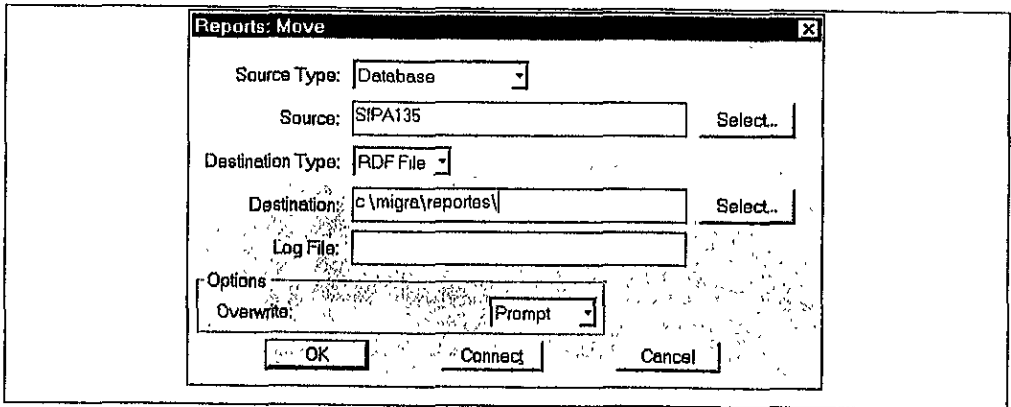


Figura 3.3.5 Caja de diálogo para la migración de reportes.

En la figura 3.3.6 se muestran los argumentos requeridos para las diferentes rutas de migración. Los argumentos serán explicados posteriormente.

FUENTE V1.1	DESTINO V2.5	ARGUMENTOS REQUERIDOS
database	database	USERID, SOURCE
database	files	USERID, SOURCE, DTYPE, DEST
files	database	USERID, STYPE, SOURCE
files	files	USERID, STYPE, SOURCE, DTYPE, DEST

Figura 3.3.6 Argumentos requeridos para las diferentes rutas de migración.

El uso de la utilería en la línea de comandos es de la siguiente forma:

```
r25mov32 [ [ keyword = ] value ] ...
```

Donde *keyword=value* pueden ser los siguientes :

```
[ [ USERID = ] userid ]
```

```
[ { STYPE = } { DATABASE | REXFILE } ]  
[ [ SOURCE = source_name ]  
[ { DTYPE = } { DATABASE | RDFFILE | REXFILE } ]  
[ { DEST = } destination_name ]  
[ { CMDFILE = } cmdfile ]  
[ [ LOGFILE = ] logfile ]  
[ { OVERWRITE = } { YES | NO | PROMPT | NEWNAME } ]
```

A continuación se explica cada valor :

### ***Connect/USERID***

Es el login para la Base de Datos (ejemplo : scott/tiger@dbalias).

### **Source Type/STYPE**

Es el formato de los reportes en versión 1.1 que serán migrados.

Opciones :

**DATABASE** significa que el reporte fuente está almacenado en Oracle.

**REXFILE** significa que el archivo fuente está almacenado en un archivo de texto (un archivo con la extensión .rex)

Si este argumento se omitió se tomará **DATABASE**.

**Source/SOURCE**

Es el nombre (y opcionalmente la ruta) de los reportes V1.1 que serán migrados.

Opciones :

Si STYPE es REXFILE, especificar cualquier nombre de archivo (por ejemplo sipa131.rex), los caracteres comodines no son permitidos.

Si STYPE es DATABASE, especificar cualquier nombre de reporte ; los caracteres comodines son válidos (% y \_).

Reglas :

- El valor especificado para este argumento es dependiente del sistema operativo donde se está trabajando.
- Se deben tener permisos de lectura sobre el archivo que se especifique.
- Si se migran reportes de Sql\*ReportWriter V1.1 almacenados en la Base de Datos, el valor de SOURCE puede contener caracteres comodines (% y \_). Por ejemplo, r% migrará reportes cuyo nombre inicie con "r".

**Destination Type/DTYPE**

Es el formato al que se migrarán los reportes.

Opciones :

**DATABASE** significa que los reportes migrados deben ser almacenados en Oracle.

**RDFFILE** significa que los reportes migrados deben ser almacenados en uno o más archivos de definición de reporte (archivos con la extensión .rdf).

**REXFILE** significa que los reportes migrados deben ser almacenados en uno o más archivos de texto (archivos con la extensión .rex).

El valor por omisión es DATABASE.

### **Destination/DEST**

Es el lugar en donde se depositarán los reportes V2.5 migrados.

Opciones :

Cualquier directorio válido para depositar los reportes migrados. Los reportes migrados se almacenarán en este directorio con nombres por omisión como se detalla más adelante.

Por omisión :

Si DEST no es especificado, R25MOV32 genera nombres usando la siguiente lógica :

- Si DTYPE es DATABASE, entonces DEST es igual a SOURCE.
- Si DTYPE es RDIFFILE, entonces DEST es SOURCE con la extensión .rdf.
- Si DTYPE es REXFILE, entonces DEST es SOURCE con la extensión .rex.

Reglas :

- Cabe aclarar que si el nombre del archivo es mayor a 8 caracteres, los nombres son truncados.
- Si SOURCE y DEST tienen el mismo nombre de archivo y extensión y OVERWRITE es igual a YES, el archivo SOURCE será sobrescrito con el archivo migrado.
- El valor que se especifica para este argumento es dependiente del sistema operativo en el que se está trabajando.

- Si no se especifica directorio para DEST y DTYPE es RDIFFILE ó REXFILE, el archivo migrado será colocado en el directorio actual.
- Si SOURCE es un archivo donde se encuentran varias definiciones de archivos (por ejemplo un archivo .rex con 5 definiciones de reportes), y DTYPE es RDIFFILE ó REXFILE, un archivo por cada reporte será generado.

## **CMDFILE**

Es el archivo (y ruta opcional) que contiene argumentos para el ejecutable R25MOV32. Esta opción permite migrar reportes sin la necesidad de especificar demasiados argumentos cada vez que se invoca. Esta opción está disponible solo en la línea de comandos.

Reglas :

- El valor que se especifica para este argumento es dependiente del sistema operativo en donde se está trabajando.
- Un archivo de comandos puede mandar llamar a otro archivo de comandos.
- La sintaxis de un archivo de comandos debe ser compatible con la línea de comandos.
- Los valores especificados en la línea de comandos reemplazarán aquellos especificados en el archivo de comandos.

## **Overwrite/OVERWRITE**

Significa sobrescribir un reporte existente con uno migrado, o preguntar por un nuevo nombre.

Opciones :

**YES** significa siempre sobrescribir un archivo existente u objeto de la Base de Datos.

**NO** significa nunca sobrescribir un archivo existente u objeto de la Base de Datos.

**PROMPT** significa preguntar si el nombre en DEST ya existe (de tal forma que se puede especificar un nuevo nombre o sobrescribir).

**NEWNAME** significa siempre preguntar el nombre para cada reporte migrado.

El valor por omisión es **PROMPT**.

Hasta aquí se explica la migración de reportes en general, adelante describiremos los pasos realizados en particular para nuestro trabajo.

Se tomó la opción directa de convertir reportes en la Base de Datos a archivos de reporte (.RDF) de Oracle\*Reports 2.5. Aunque es más lenta, se consumiría más tiempo en migrar los reportes uno por uno y se tendría que realizar un paso extra que es obtener el archivo .rex por medio de una utilería llamada "dumprep".

A continuación se presentan los reportes antes y después de la migración.

VT320/VT340 - MXPYR2.VT

File Edit Commands View Help

04-OCT-97 Pagina 6 de 82

Reporte de Actividades hechas a Clientes  
entre el 04-JAN-97 y el 04-FEB-97

CLIENTE: BANCO MEXICANO  
Número : 717

Fecha	Actividad	Realizada por	Horas Utilizadas
27-JAN-97	Junta Tecnica	Victor Cabra	1.5
04-FEB-97	Activ.de oficina	Edgar Perez	2
TOTAL DE HORAS:			3.5

CLIENTE: BANCO PARICIONAL DE MEXICO  
Número : 130

Fecha	Actividad	Realizada por	Horas Utilizadas

For Help, press HELP key, or ALT-F1 (001,001)

Figura 3.3.7 Reporte modo carácter antes de la migración.

Developer/2000 Reports Designer for Windows 95 / NT - [SIPA132F: Previewer]

File Edit Window Help

Prev Next First Last Page: 14 Print Mail Close New

CLIENTE: NACIONAL FINANCIERA S. N. C.  
 Numero : 238

Fecha	Actividad	Realizada por	Horas Utilizadas
05-JUL-95	Instalacion	Jocelyn Garcia	1
05-JUL-95	Asesoria a socios	Jocelyn Garcia	1
17-JUL-95	Asesoria a socios	Jocelyn Garcia	1
17-JUL-95	Asesoria a socios	Jocelyn Garcia	2
18-JUL-95	Asesoria a socios	Jocelyn Garcia	1
20-JUL-95	Asesoria a socios	Jocelyn Garcia	2
21-JUL-95	Asesoria a socios	Jocelyn Garcia	2
26-JUL-95	Instalacion	Jocelyn Garcia	4
27-JUL-95	Instalacion	Jocelyn Garcia	2
27-JUL-95	Asesoria a socios	Jocelyn Garcia	1
27-JUL-95	Asesoria a socios	Jocelyn Garcia	1
27-JUL-95	Asesoria a socios	Jocelyn Garcia	4
31-JUL-95	Instalacion	Jocelyn Garcia	4
TOTAL DE HORAS:			26

Figura 3.3.8 Reporte modo gráfico (después de la migración).

Una vez realizada la migración y obtenidos los archivos .RDF (Oracle\*Reports 2.5), el siguiente paso es realizar la Ingeniería en Reversa de los Reportes.

**Ingeniería en Reversa de los Reportes :**

Para poder realizar la Ingeniería en Reversa se debe tener:

- Acceso a los reportes.
- Permiso de escritura sobre una Aplicación del Repositorio.
- Definiciones en el Repositorio de todos los objetos referenciados por el reporte.

Nota : La utilería para realizar la Ingeniería en reversa no intenta inferir el diseño de la base de datos a partir de la definición del reporte. Si definiciones de objetos usados por el reporte no



existentes en el Repositorio, se debe utilizar la utilidad para llevar a cabo la Ingeniería en Reversa de dichos objetos.

Para hacer Ingeniería en Reversa de un Reporte se debe:

1. Seleccionar Utilities->Reverse Engineer->Report del “Module Data Diagrammer” o del “Repository Object Navigator”.

La caja de diálogo para la Ingeniería en reversa de un reporte aparece:

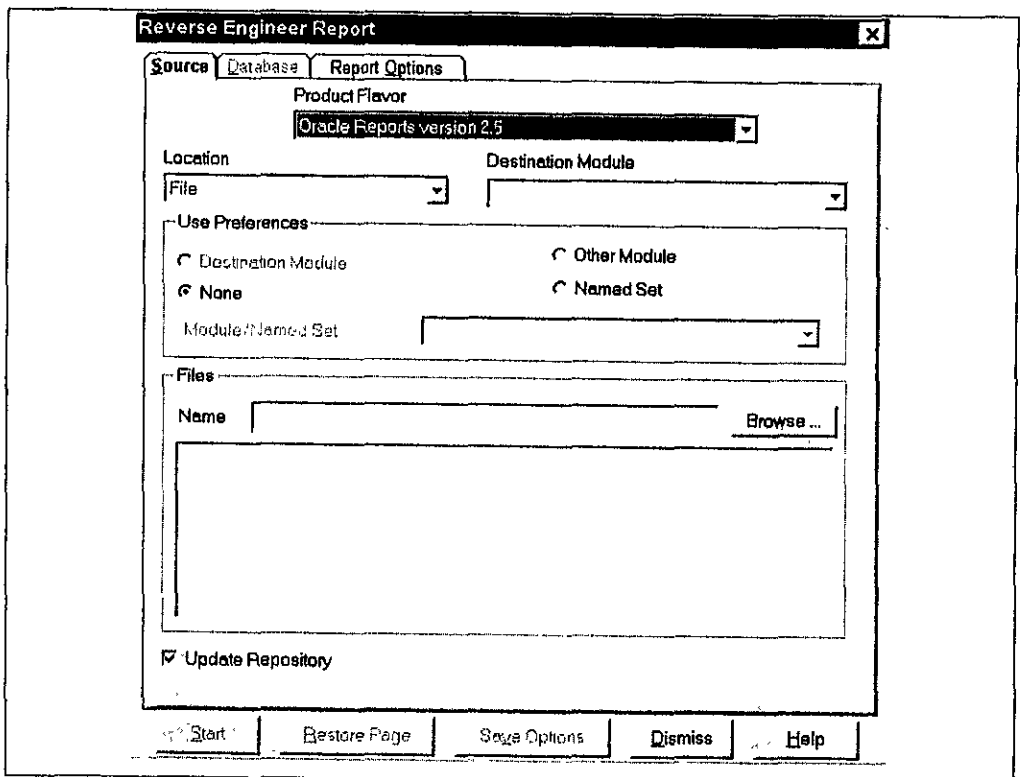


Figura 3.3.9 Caja de diálogo para la Ingeniería en Reversa de un reporte.

2. Se selecciona el tipo de reporte al que se le hará la Ingeniería en reversa en el lista llamada “Product Flavor”.

3. En el campo llamado “Location” se especifica si el reporte al que se le hará Ingeniería en Reversa está localizado en archivo o en una Base de Datos.
4. Si se desea actualizar un módulo existente, se selecciona el nombre corto del módulo en el campo llamado “Destination Module”
5. Elegir el tipo de preferencias que se desean usar para hacer Ingeniería en Reversa del reporte.
6. Especificar el nombre del reporte al que se desea hacer Ingeniería en Reversa.
7. Si se desea actualizar el Repositorio con la información del proceso de Ingeniería en Reversa, seleccionar el checkbox indicado como “Update Repository”
8. Si se está realizando la Ingeniería en Reversa de un reporte almacenado en la Base de Datos, especificar el nombre del dueño del reporte en el campo “Owner” del folder “Database”.
9. Seleccionar el botón de “Start”. El Generador despliega la caja de diálogo del Generador de Reportes y realiza la Ingeniería en Reversa del reporte. Si se seleccionó el checkbox del “Update Repository”, el Generador registra la información en el Repositorio.

Actividades del proceso de Ingeniería en Reversa de reportes.

El Generador crea una nueva definición de Módulo para cada reporte que el Generador hace IR. Si la definición del módulo ya existe el Generador consolidará la nueva definición del módulo con la anterior.

El Generador establece la jerarquía de los “queries” y la secuencia de las tablas referenciadas usando la estructura de grupos en el reporte.

El Generador identifica los “items” seleccionados en cada query y luego crea referencias a tablas y columnas para los “items” seleccionados. El Generador usa la condición de unión (join) o subqueries para crear las referencias de tablas y columnas que no están desplegados en el módulo.

El Generador extrae los siguientes objetos del “Data Model”:

- Referencias de columnas “summary”
- Parámetros
- “Break groups”

### **Información Registrada durante el Proceso de Ingeniería en Reversa.**

El Generador crea los siguientes elementos y registra las propiedades de estos elementos en el repositorio:

- Definición del módulo reporte
- Referencias a tablas
- Referencias a columnas
- Parámetros

### **Definición de módulos reporte.**

El generador registra las siguientes propiedades en el Repositorio siempre que se crea un nuevo módulo:

PROPIEDAD	VALOR ASIGNADO O FUENTE
Short Name	El nombre del reporte original
Name	El nombre del reporte original
Purpose	"Reverse Engineer by Oracle Reports Generator"
Language	Oracle Reports or Sql*ReportWriter
Type	REPORT
Format	Master-Detail
Notes	Mantiene las del módulo de Oracle Reports

Figura 3.3.10 Propiedades de un nuevo Módulo.

### **Nuevas Referencias a Columnas.**

El *Generador* registra las siguientes propiedades en el Repositorio cuando una nueva referencia a columna se crea (Detailed Column Usage):

PROPIEDAD	VALOR ASIGNADO O FUENTE
Seq	Valor específico del módulo
Display checkbox	Seleccionado si la columna es usada en un ítem de select o una columna summary
Select	Y, si la columna es usada en un ítem de select N, si existe una referencia secundaria como un summary
Datatype	El tipo de dato de la columna
Width	Definido como el valor de default de Oracle*Reports, si fué seleccionado el checkbox de "Add default column display information", de otra forma queda definido como el ancho máximo o el default
Height	Mismo que el anterior
Prompt	Mismo que el anterior
Summary Function	Es definido solo si la columna es un "summary"
Summary Type	Definido solo si la columna es una Función y el tipo puede ser determinado
Wrap Style	Valor específico de la columna
Break	Definido como el indicador especificado a la hora de realizar la IR, si la columna está dentro de un grupo de corte "break group"
Comment	"Created by Oracle Reports Generator"

Figura 3.3.11 Propiedades de referencias a columnas.

Nuevo Módulo de Parámetro.

El generador registra las siguientes propiedades en el Repositorio siempre que el proceso de IR crea un nuevo módulo de parámetro.

PROPIEDAD	VALOR ASIGNADO O FUENTE
Name	El nombre del parámetro en Oracle*Reports
Seq	Valor específico del módulo
Datatype	El tipo de dato de la columna
Width	Definido como el valor de ancho por omisión en Oracle*Reports. Definido a la longitud máxima si no existe un ancho por omisión.
Operator	El carácter "="
Prompt	Definido con el prompt por omisión
Description	"Reverse Engineered by Oracle Reports Generator"

Figura 3.3.12 Propiedades del Módulo de Parámetro.

**Nota:**

A los summary a nivel registro no se les hace Ingeniería en Reversa.

Al intentar realizar la IR de nuestros reportes obtuvimos el mensaje mostrado a continuación:

CGEN-02504: ERROR: SYSTEM: Unable to parse query query\_name.

Este error indica que no se pudo realizar el análisis (parse) de la consulta que pertenece al reporte.

Investigando al respecto se encontró que los objetos referenciados en la consulta del reporte tienen que pertenecer al usuario que es dueño de la aplicación en el Repositorio.

En nuestra base de datos el dueño de los objetos del sistema es CANALES/SIPA y el dueño de la aplicación es APLICA/APLICA.

Una forma de corregir el problema es que el usuario CANALES otorgue permisos de SELECT sobre sus objetos al usuario APLICA y luego éste realice la creación de sinónimos hacia dichos

objetos. Los comandos necesarios para conseguir lo anterior se muestran en las figuras 3.3.13 y 3.3.14.

```
SELECT 'GRANT SELECT ON '||TABLE_NAME||' TO APLICA;'  
FROM USER_TABLES  
/
```

El script anterior genera una lista de comandos con el siguiente formato:

```
GRANT SELECT ON OBJECT_NAME TO APLICA;
```

Figura 3.3.13 Comandos ejecutados como usuario CANALES.

```
SELECT 'CREATE SYNONYM '||TABLE_NAME||' FOR CANALES.'||TABLE_NAME||';'  
FROM USER_TAB_PRIVS  
WHERE OWNER='CANALES'  
/
```

El script anterior genera una lista de comandos con el siguiente formato:

```
CREATE SYNONYM OBJECT_NAME FOR CANALES.OBJECT_NAME;
```

Figura 3.3.14 Comandos ejecutados como usuario APLICA.

Una vez realizado lo anterior se realizó la IR de los reportes.

Internamente los nombres que se le asignaron a los módulos tiene el formato RS999AA.

Donde:

9 significa número.

A significa letra.

En el Object Navigator de Designer se muestran los reportes a los que ya se realizó Ingeniería en Reversa en el nodo de "Modules".

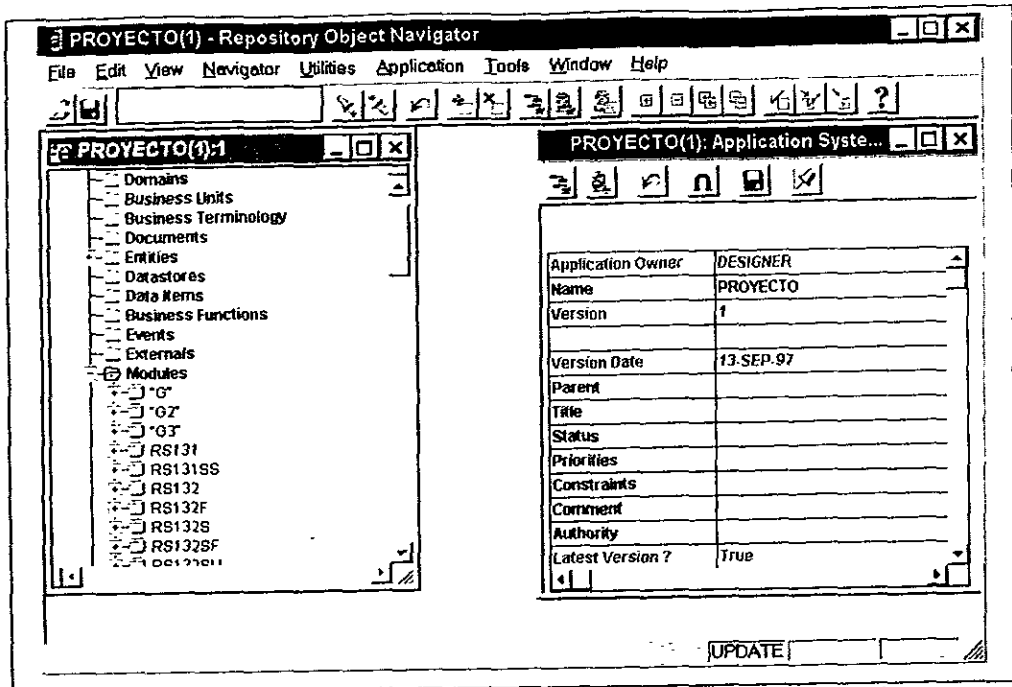


Figura 3.3.15 En el nodo de "Modules" del Object Navigator se muestran módulos tipo reporte, formas y procedimientos.

### *A continuación hablaremos de la Migración e Ingeniería en Reversa de las formas :*

Para poder llevar a cabo la IR de las Formas fué necesario migrar las formas de versión 3.0 a versión 4.5 de Oracle\*Forms debido a que Designer no puede realizar la IR de formas de versión 3.0.

### **Actualización de Formas a Versión 4.5**

Oracle Forms Versión 4.5 incluye un generador para actualizar:

- Formas creadas con Sql\*Forms Versión 2.3 y 3.0.
- Menús creados con Sql\*Menu Versión 5.0.



En versiones anteriores el Generador y el Convertidor eran herramientas separadas. En la versión 4.5 el componente Generate realiza ambas operaciones tanto conversión como generación utilizando los siguientes parámetros:

**upgrade.** Especifica que se desea realizar la operación de conversión.

**version.** Especifica la versión de la forma actual.

Para actualizar de versión 2.3 o 3.0 a versión 4.5, utilizar la siguiente instrucción:

```
f45gen32 <module_name> <username>/<password> upgrade=yes
```

Por omisión `version=30` para actualizar formas de versión 3.0. Si se desea actualizar de versión 2.3 `version=23` y para versión 2.0 `version=20`.

Con la siguiente instrucción se migraron las formas del sistema:

```
f45gen32 SIPA999AA.inp canales/sipa upgrade=yes
```

Donde:

9 número

A letra.

Nota:

La entrada al generador son los archivos con extensión **.inp** que son los que contienen información de diseño de las formas.

A continuación se muestran las formas antes y después de la migración:

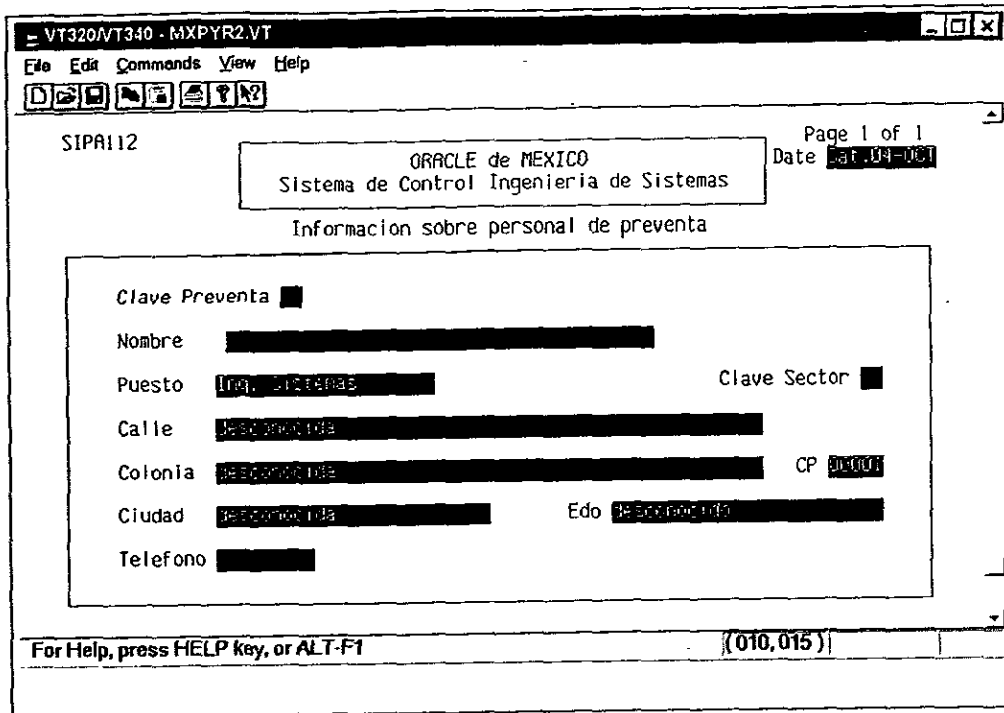


Figura 3.3.16 Forma en modo carácter (antes de la migración).

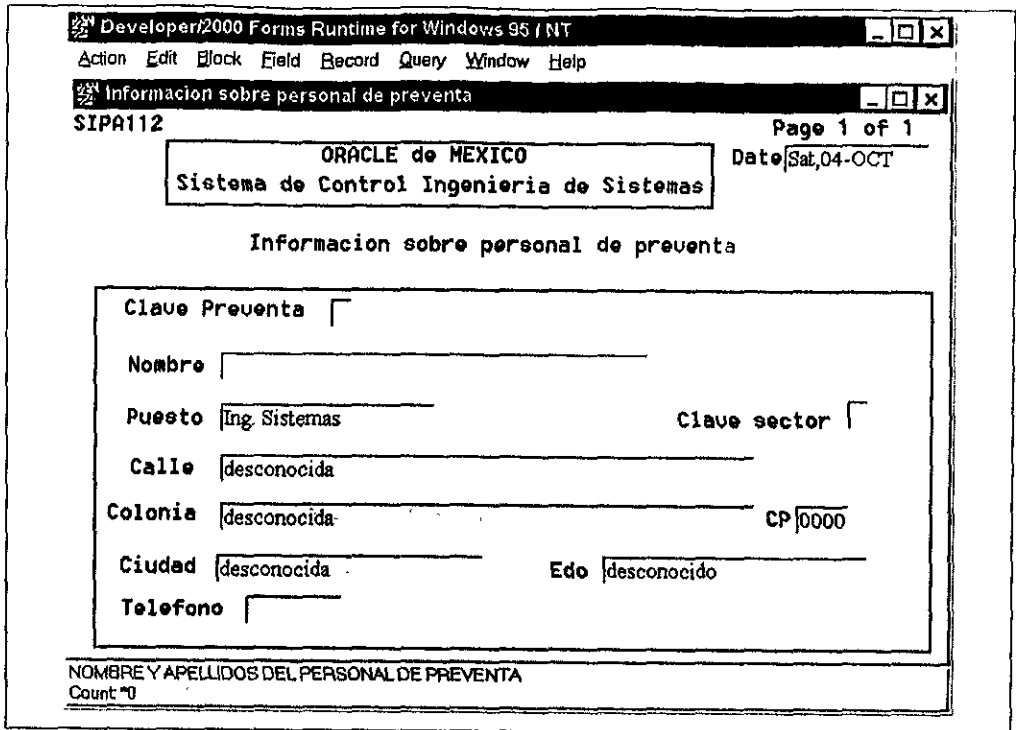


Figura 3.3.17 Forma en modo gráfico después de la migración.

Una vez realizada la migración se procede a la IR de las formas.

Antes de hacer IR de una forma se debe tener:

- Acceso a la forma a la que se le hará IR (la forma debe haber sido construida usando Oracle Forms 4.5; para hacer IR de una forma creada en una versión anterior de forms se debe convertir la forma a versión 4.5)
- Privilegios de escritura sobre una aplicación del repositorio de Designer.
- Definiciones en el Repositorio de todos los objetos de base de datos referenciados por la forma.

Para hacer IR de una forma se debe:

1. Iniciar el "Module Data Diagrammer".
2. Seleccionar Utilities -> Reverse Engineer -> Form.

La caja de diálogo para la IR de una forma aparece:

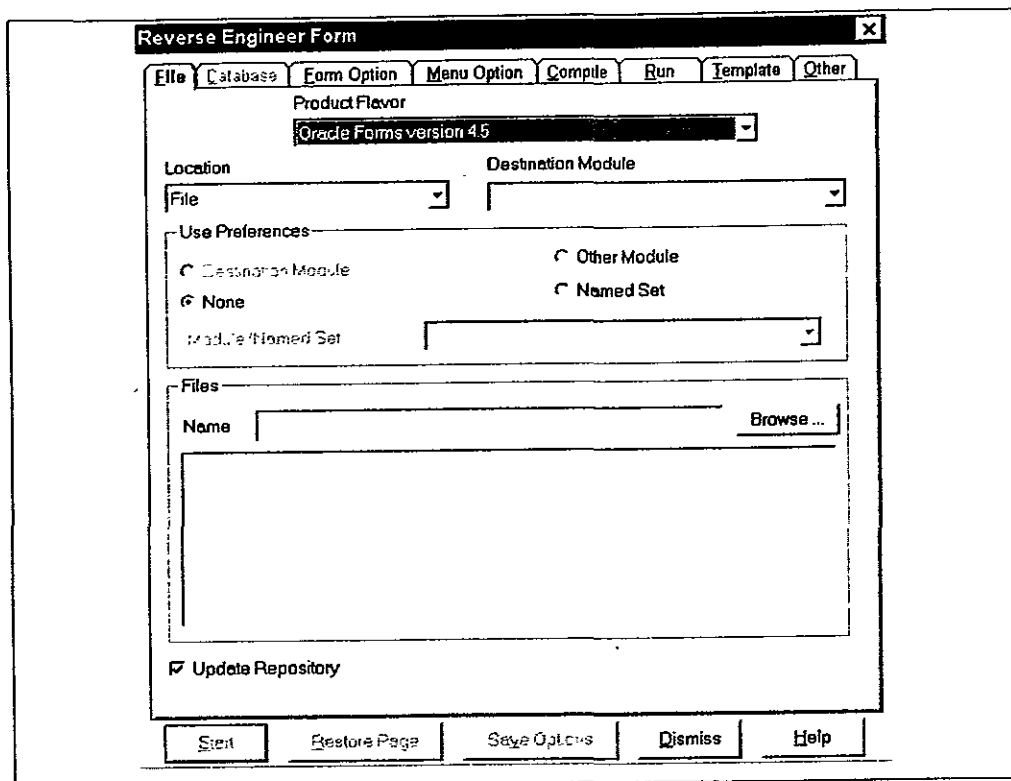


Figura 3.3.18 Caja de diálogo para la Ingeniería en Reversa de una forma.

3. Especificar el tipo de almacenamiento "Location" en que se encuentra la forma (File o Database).

4. Introducir el nombre del módulo del Repositorio en el que se guardará la información generada por la IR.
5. Especificar el nombre de la forma de la que se desea realizar la IR.
6. Presionar sobre el botón se “Start” para iniciar la IR de la forma.
7. Cuando el proceso de IR es terminado, cerrar el cuadro de mensaje.

A continuación se describirá lo que se registra en el Repositorio cada vez que se realiza la IR sobre una forma.

PROPIEDAD	VALOR ASIGNADO O FUENTE
Short Name	Definido como el nombre del módulo al realizar la IR.
Name	El Generador inicialmente asume que es el mismo que el Short Name. Sin embargo si entra en conflicto con otro nombre de módulo se le pregunta al usuario introducir uno nuevo.
Purpose	Definido como el título de la primera Ventana con título de la forma. Si no existe el Generador coloca el Short Name.
Language	Oracle Forms
Type	SCREEN
Format	MASTER DETAIL
Top Title	Definido como el título de la primera Ventana con título de la forma. Si no existe el Generador lo deja en blanco.
Comments	Si el Generador está creando un nuevo módulo, éste copia los comentarios a nivel forma.
Notes	Si la preferencia ADNTRT está en "Y", la siguiente cadena de texto se coloca:  Reverse Engineered on <date> by user <user> using Oracle Forms Generator Version <version>

Figura 3.3.19 Propiedades del módulo.

PROPIEDAD	VALOR ASIGNADO O FUENTE
Seq No	El número de secuencia de la referencia a tabla que es mapeado al bloque previo e incrementado en 1. Si no hay bloque previo el número es 1.
Delete flag	Definido de acuerdo al atributo DELETE del bloque.
Insert flag	Definido de acuerdo al atributo INSERT del bloque.
Query flag	Definido de acuerdo al atributo QUERY del bloque.
Update flag	Definido de acuerdo al atributo UPDATE del bloque.
Display title	Toma el valor del título para la definición de la tabla, el cual es registrado en el repositorio.
WHERE/Validation Condition	Copiado de la cláusula WHERE registrada para el bloque.
Comments	Definido como "Reverse Engineered by <user> on <date>."
Type	BASE

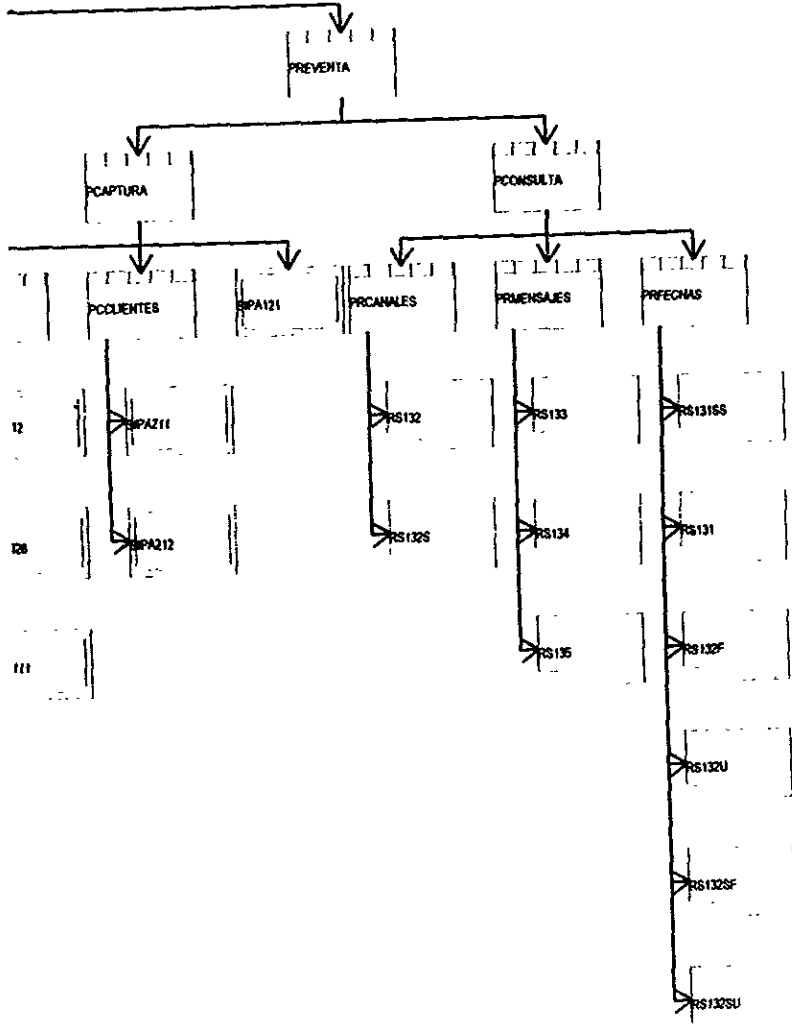
Figura 3.3.20 Propiedades de los detalles de referencia a tabla de un módulo.

Ingeniería en Reversa de los Procedimientos almacenados.

Nuestra aplicación hace uso de tres procedimientos almacenados en la Base de Datos. Al concluir la Ingeniería en Reversa de los mismos, quedaron como módulos con nombres G, G2 y G3.

### Diagrama Estructurado de Módulos.

En la figura 3.3.21 se muestra el Diagrama Estructurado de Módulos.





### Diagrama de Datos de Módulos.

El Module Data Diagrammer proporciona una forma fácil y rápida de crear referencias para Pantallas y Reportes. Estas referencias son usadas por los generadores para afectar la estructura, funcionalidad y presentación de las aplicaciones generadas.

En la figura 3.3.22 se muestra el Diagrama de Datos para el Módulo SIPA121 el cual está formado por una relación Maestro-Detalle.

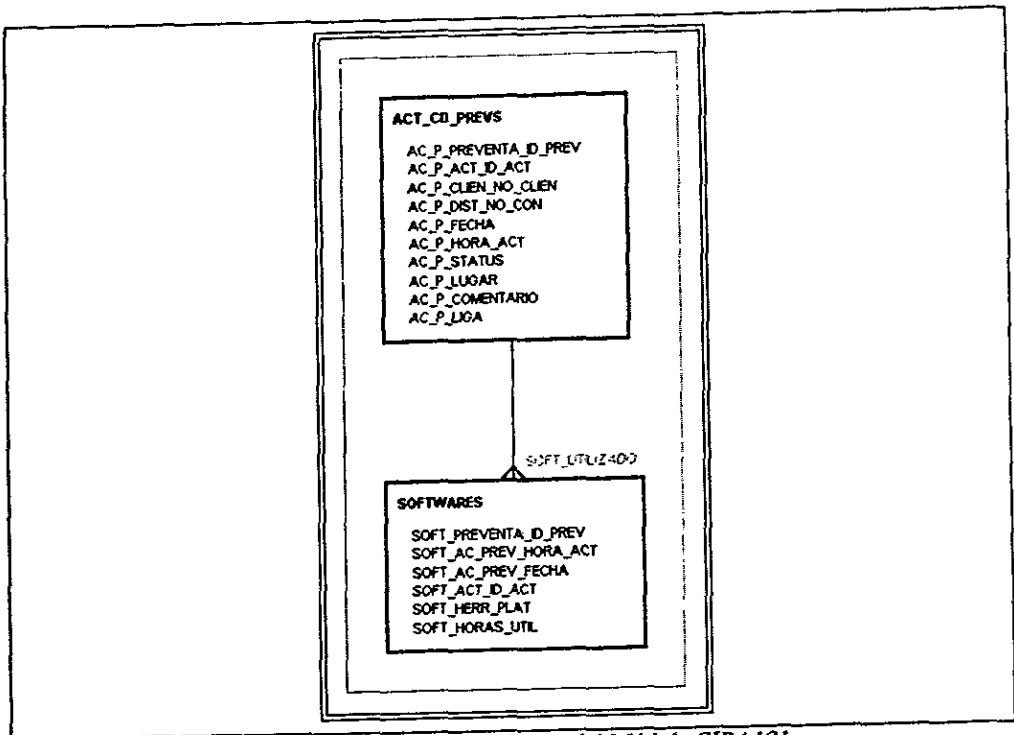


Figura 3.3.22 Diagrama de Datos del Módulo SIPA121.

### Diagrama de Referencias Cruzadas

El diagramador de matrices de "Designer" permite obtener entre otras cosas el Diagrama de Referencias Cruzadas con la información de cuáles módulos hacen referencia a qué tablas y columnas. A continuación se presenta un ejemplo donde podemos observar que el módulo

SIPA121 hace referencia a la tabla ACT\_CD\_PREVS y en la intersección se encuentra el nombre de las columnas de dicha tabla a las cuales hace referencia el módulo.

The screenshot shows a window titled "Matrix Diagrammer PROYECTO (1) - MXD11". The main area displays a matrix with the following data:

Columns	Modules	SPA121	SPA312	SPA326
ACT_CD_PREVS	ACT_CD_PREVS	AC P PREVENTA ID PREV		
ACT_CD_PREVS	ACT_CD_PREVS	AC P FECHA		
ACT_CD_PREVS	ACT_CD_PREVS	AC P COMENTARIO		
ACT_CD_PREVS	ACT_CD_PREVS	AC P LIGA		
ACT_CD_PREVS	ACT_CD_PREVS	AC P HORA ACT		
ACT_CD_PREVS	ACT_CD_PREVS	AC P CUEN NO CUEN		
ACT_CD_PREVS	ACT_CD_PREVS	AC P DIST NO CON		
ACT_CD_PREVS	ACT_CD_PREVS	AC P ACT ID ACT		
ACT_CD_PREVS	ACT_CD_PREVS	AC P STATUS		
ACT_CD_PREVS	ACT_CD_PREVS	AC P LUGAR		
ACTIVIDADES	ACTIVIDADES	ACT_ID_ACT		
ACTIVIDADES	ACTIVIDADES	ACT_DESCRIPCION		
APLICACIONES	APLICACIONES			
APLICACIONES	APLICACIONES			
APLICACIONES	APLICACIONES			
APLICACIONES	APLICACIONES			
AREA DIST	AREA DIST			
AREA DIST	AREA DIST			
AREAS	AREAS			AREA TERRITORIO
AREAS	AREAS			AREA ID AREA

At the bottom of the window, it says "For Help, press F1" and "NUM".

Figura 3.3.23 Parte de la matriz que muestra las tablas y columnas a las que hace referencia cada módulo que compone el sistema.

### **3.4. REGENERACIÓN DE LA BASE DE DATOS A LA VERSIÓN REQUERIDA.**

#### **Introducción**

Mediante la ingeniería en reversa de la base de datos, se ha recuperado en la medida de lo posible, el diseño de datos actual del sistema de preventa. Tomando esta información como base, ahora es necesario regenerar la base de datos, de manera tal que incluya los nuevos requerimientos del usuario mencionados en las especificaciones del diseño.

Al regenerar la base de datos, se obtienen las definiciones de datos, reglas del negocio y definiciones de procedimientos, definiciones de las transacciones, definiciones de uso de datos, así como la definición de la base de datos que será utilizada para el nuevo sistema de preventa y para ello se hace uso de diversos componentes de Systems Designer, del Repository Object Navigator, así como del Server Generator de Designer/2000.

Mediante el Data Diagrammer, se obtiene el diseño completo de la base de datos, a partir de los modelos del negocio, de las bases de datos a las que se les hizo ingeniería en reversa o en base a definiciones nuevas de tablas, vistas y snapshots.

#### **Afinación del Diseño Inicial de la Base de Datos**

El proceso de enmendar y enriquecer el diseño de la base de datos es una de las tareas más importantes en el ciclo de desarrollo de aplicaciones. Los elementos de diseño definidos durante el diseño de la base de datos son usados por los Generators para producir el diseño de la aplicación, por lo que se tienen que definir todos los objetos de la base de datos a detalle lo más preciso posible.

El esquema de base de datos requerido por el sistema de preventa es el siguiente:

Base de Datos:

ORACLE\_SID: ORCL

Versión : 7.3

Nodo: Pyramid ó NT

Tablespaces y archivos de datos:

System : Para guardar las tablas del diccionario de datos de la base de datos, las definiciones de usuarios, roles, vistas y secuencias utilizadas en la aplicación.

Archivo de datos: /oracle7/dbs/preventa/systemorcl.dbf, tamaño: 50 Mb

cláusula de almacenamiento: default

RBS: Para almacenar los segmentos de Rollback.

Archivo de datos: /oracle7/dbs/preventa/rbsorcl.dbf, tamaño: 10 Mb

cláusula de almacenamiento: (initial 300k next 300k pctincrease 0)

Temp : Para almacenar segmentos temporales

Archivo de datos: /oracle7/dbs/preventa/temporcl.dbf, tamaño: 15 Mb

cláusula de almacenamiento: default

Datos: Para almacenar las tablas de la aplicación.

Archivo de datos: /oracle7/dbs/preventa/datosorcl.dbf, tamaño: 350 Mb

cláusula de almacenamiento: (initial 3M next 500K pctincrease 0)

Indices: Para almacenar los índices de las tablas de la aplicación.

Archivo de datos: /oracle7/dbs/preventa/indicesorcl.dbf, tamaño: 150 Mb

cláusula de almacenamiento: (initial 1M next 300K pctincrease 0)

Usuarios: Para almacenar las tablas temporales que generen los usuarios al momento de utilizar alguna herramienta de consulta a la base de datos para hacer consultas no planeadas.

Archivo de datos: /oracle7/dbs/preventa/usuariosorcl.dbf, tamaño: 150 Mb  
cláusula de almacenamiento: (initial 200k next 300K pctincrease 0)

En la figura 3.4.1, se puede observar que una base de datos lógicamente esta conformada por uno o varios tablespaces, los cuales a su vez están formados por uno o varios archivos de datos. En los tablespaces, se almacenan los objetos de base de datos tales como vistas, tablas, índices, segmentos (temporales y de rollback), así como las definiciones de usuarios y roles.

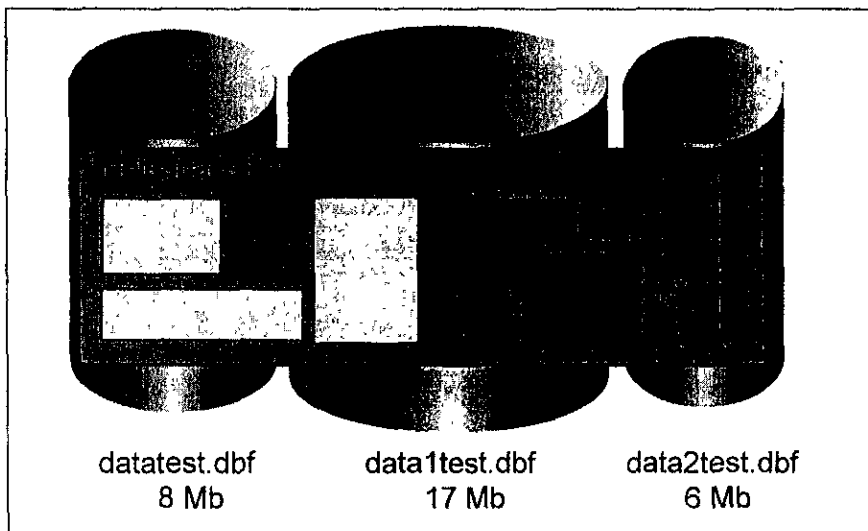


Figura 3.4.1. Diagrama de la estructura de archivos de una base de datos.

Roles y usuarios:

Administrador: Permiso de SELECT, INSERT, UPDATE y DELETE sobre todas las tablas de la aplicación.

Usuarios agrupados en este rol: aplica

Privilegios adicionales a los usuarios de este rol: dba, connect, resource

Gerente: Permiso de SELECT sobre todas las tablas de la aplicación.

Usuarios agrupados en este rol: mcruz, mdiaz, dibanez

Privilegios adicionales a los usuarios de este rol: connect, resource

Preventa: Permiso de SELECT, INSERT y UPDATE sobre todas las tablas de la aplicación.

Usuarios agrupados en este rol: kramirez, aomana, cvega, orojas, lalvarad, yperez, vcabra, jgarcia, jflores, ngonzale, gkaiten.

Privilegios adicionales a los usuarios de este rol: connect, resource

Tablas, vistas, secuencias e índices: Las obtenidas vía la ingeniería en reversa de la base de datos.

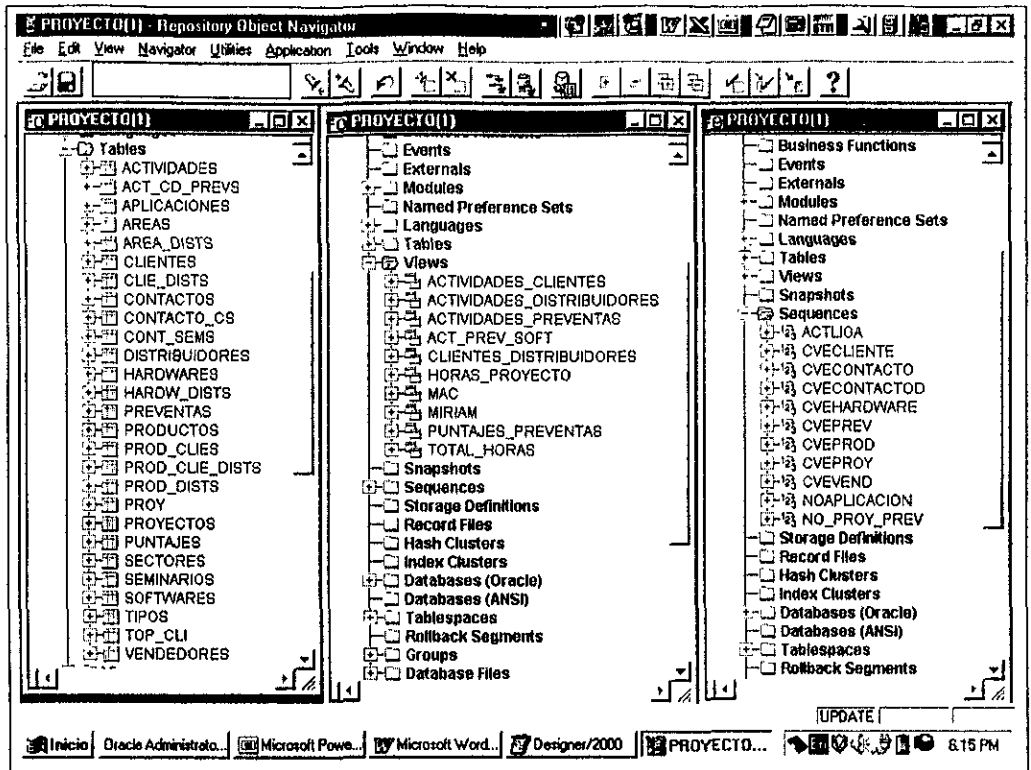


Figura 3.4.2. Tablas, vistas y secuencias obtenidas mediante la ingeniería en reversa de la base de datos.

El diseño de la base de datos anterior, puede representarse gráficamente utilizando Diagramas de Datos, los cuales modelan la estructura lógica del esquema de base de datos en términos de tablas, vistas, snapshots, columnas y constraints de integridad.

En el Data Diagrammer, por cada tabla, es necesario hacer uso de la caja de diálogo para especificar detalles de implementación de la tabla, usando los diferentes folders que aparecen en la misma. Así pues, es posible especificar por cada tabla detalles como nombre de la tabla, alias, prefijo de columnas, título de despliegue, base de datos y tipo de base de datos, número de registros iniciales y finales, cláusulas de almacenamiento y tablespace en donde la tabla será guardada; nombres de columnas, secuencias, dominio, tipo de dato, longitud promedio, longitud

máxima, posiciones decimales, porcentaje de ocupación de la columna, opcionalidad, y valor de default; secuencia en la que las columnas serán desplegadas y prompt de cada columna; reglas de validación, accesos e índices que se requieran por cada tabla.

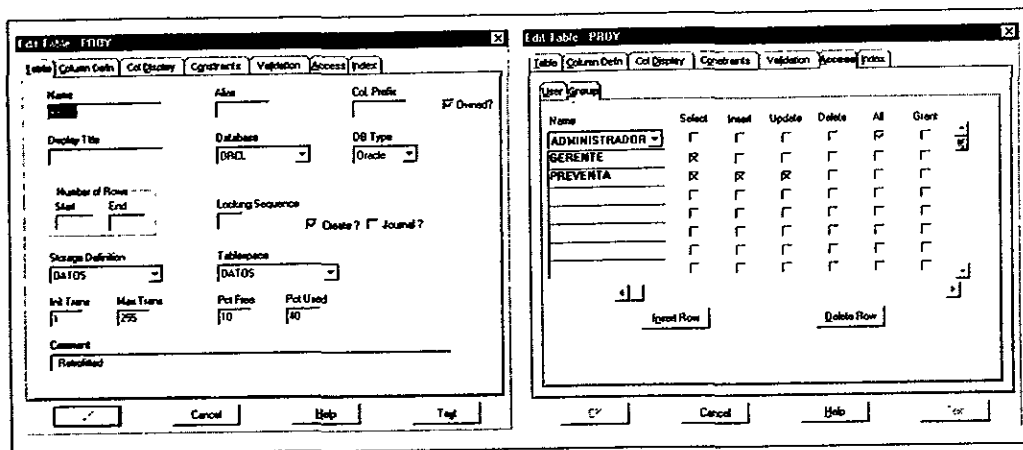


Figura 3.4.3. Caja de Diálogo para especificar las propiedades de las tablas de la aplicación.

Haciendo uso del Repository Object Navigator se definen cláusulas de almacenamiento, características de la base de datos, tablespaces, archivos de datos de cada tablespace, segmentos de rollback, grupos o roles, usuarios y nodos.

Al definir cada uno de estos objetos, se hace uso de la paleta de propiedades a través de la cual se especifican las características del objeto. En la paleta de propiedades existe un código de colores que indica la obligatoriedad de la propiedad, así pues, aquellas propiedades verdes, negras o rojas serán no modificables, opcionales u obligatorias respectivamente.



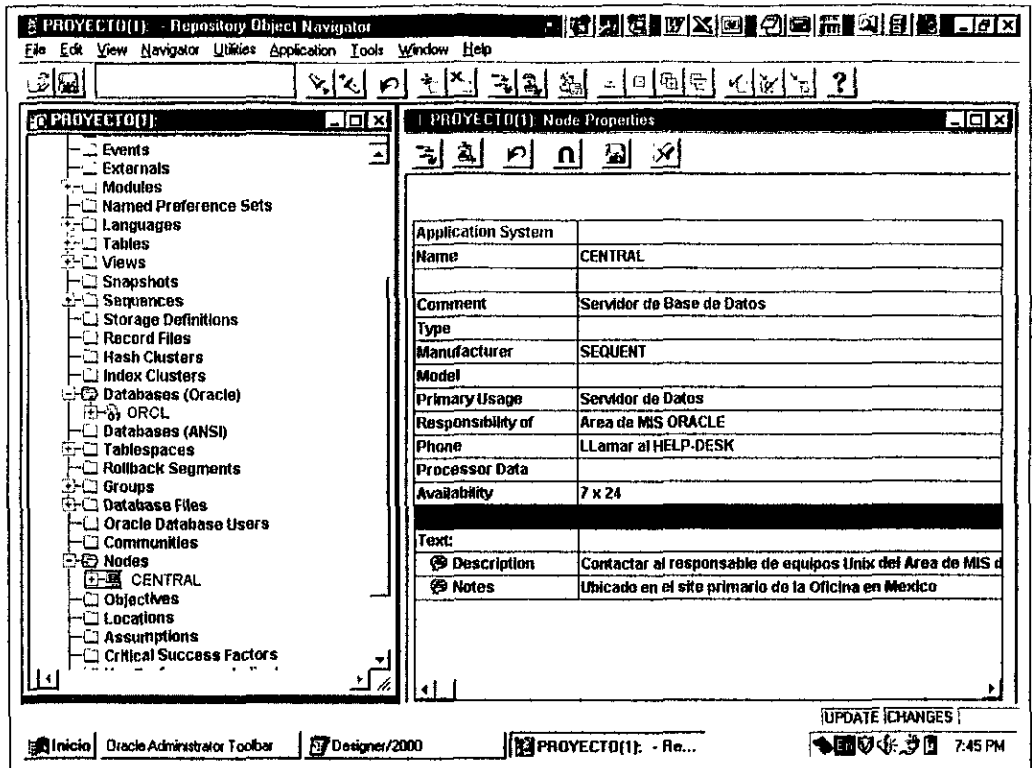


Figura 3.4.4. Repository Object Navigator para definir propiedades de los objetos del repositorio.

### Generación del DDL para la Creación de Objetos de la Base de Datos.

Una vez que se ha definido en el repositorio de Designer/2000 el diseño completo de la base de datos, se hace uso de la utilería del lenguaje de definición de datos (DDL Utility) contenida en el Server Generator de Designer, la cual, como su nombre lo dice, genera las instrucciones DDL (Data Definition Language) para crear los objetos de la base de datos a partir de las definiciones registradas en el repositorio. Estas instrucciones son escritas hacia archivos de comandos y son ejecutadas en la base de datos destino. El DDL generado, incluye además, instrucciones que controlan el acceso a datos de los usuarios tales como la creación de roles.

Para generar las instrucciones DDL que serán usadas para definir los objetos de la base de datos se llevan a cabo los siguientes pasos:

1. Invocar la Utilería SQL DDL desde el menú de Utilities del Repository Object Navigator o bien desde la pantalla principal de Designer/2000 invocar el Server Generator.
2. En la caja de diálogo Generate DDL desplegada acceder cada uno de los folders:

Folder Selection:

- a. En la lista de valores Database, seleccionar el nombre de la base de datos en donde los objetos de la base de datos serán creados.
- b. A través de los check boxes, indicar si se desea selección automática de triggers e índices.
- c. Indicar la Base de datos por default en donde los objetos que no tengan asignada una base de datos serán colocados.

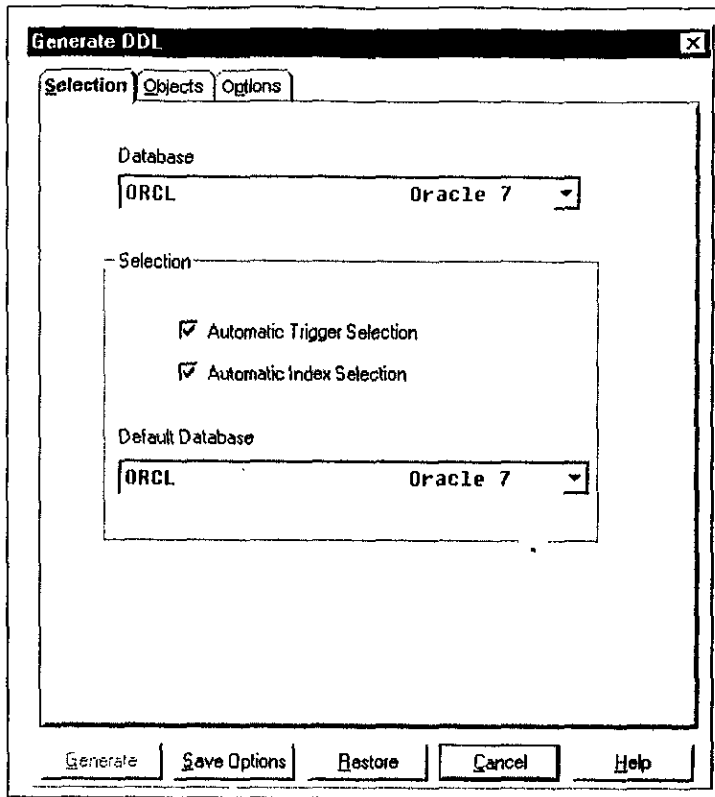


Figura 3.4.5. Pantalla de selección de base de datos del SQL DDL Utility.

Folder Objects:

- a. Seleccionar el tipo de objeto que se quiere generar en la ventana Type. Por ejemplo:  
Table
- b. En base a la selección del ejemplo anterior, en la ventana Name aparecerán los nombres de las tablas que se pueden generar. Seleccionar ayudándose del ratón aquellas tablas de interés.

- c. Utilizando los botones con flechas, mover las tablas (u objetos seleccionados) a la ventana de Select (solamente aquellos objetos que se encuentren en esta ventana serán generados).

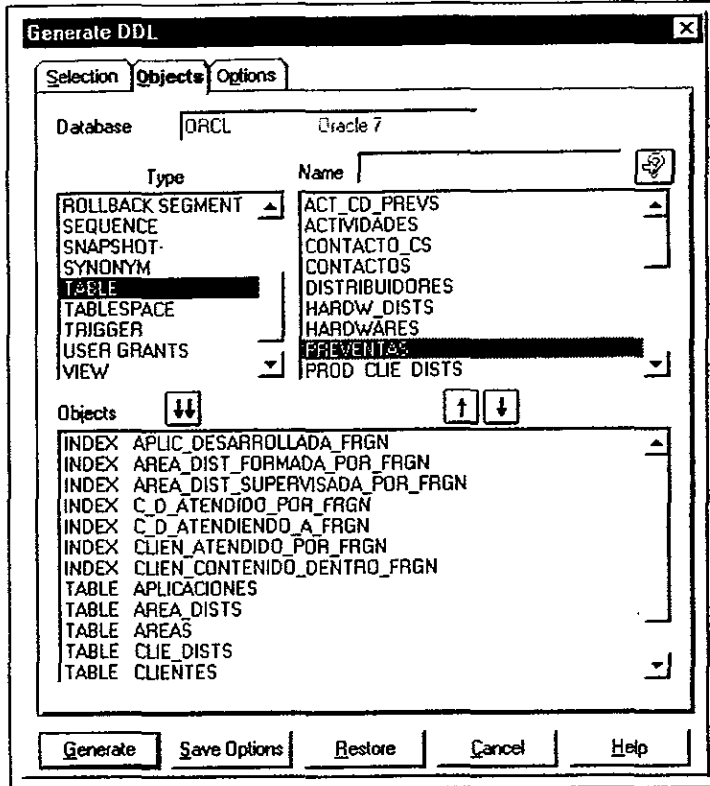


Figura 3.4.6. Pantalla de selección de objetos de la base de datos del SQL DDL Utility.

#### Folder Options:

- a. Si se quiere crear la especificación de paquetes y/o el cuerpo de los paquetes, seleccionar el radio botón apropiado de la sección Generate Package.

- b. En el campo de prefix, definir el prefijo que tendrá el nombre de los archivos de comandos que sean generados.
- c. En la sección de Options, especificar la preferencia de generación requerida en lo que a constraints de tablas, creación de la base de datos, distribución de objetos, comentarios de columnas, constraints de valores válidos.
- d. Elegir la versión de Base de Datos Oracle para la que serán creados los objetos.

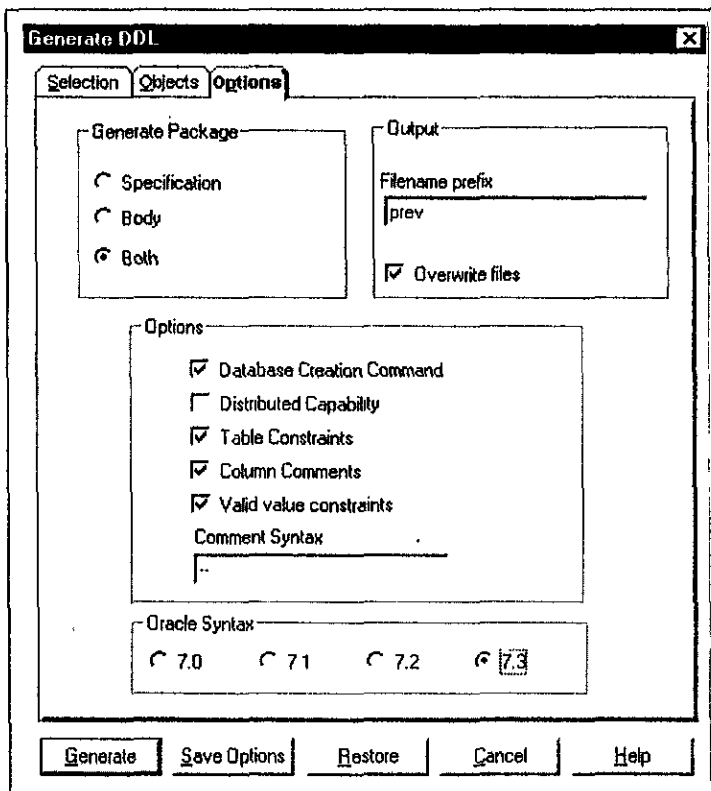


Figura 3.4.7. Pantalla de opciones de generación del SQL DDL Utility.

3. Oprimir el botón Generate para que los archivos de comandos de creación de objetos de la base de datos sean generados.

Esta utilidad genera un archivo maestro de comandos y uno o varios archivos de comandos de creación de objetos de la base de datos (el numero depende de los diferentes tipos de objetos seleccionados en el folder Objects). Dichos archivos, son mostrados en una caja de diálogo que permite examinar el contenido de los mismos o bien su ejecución.

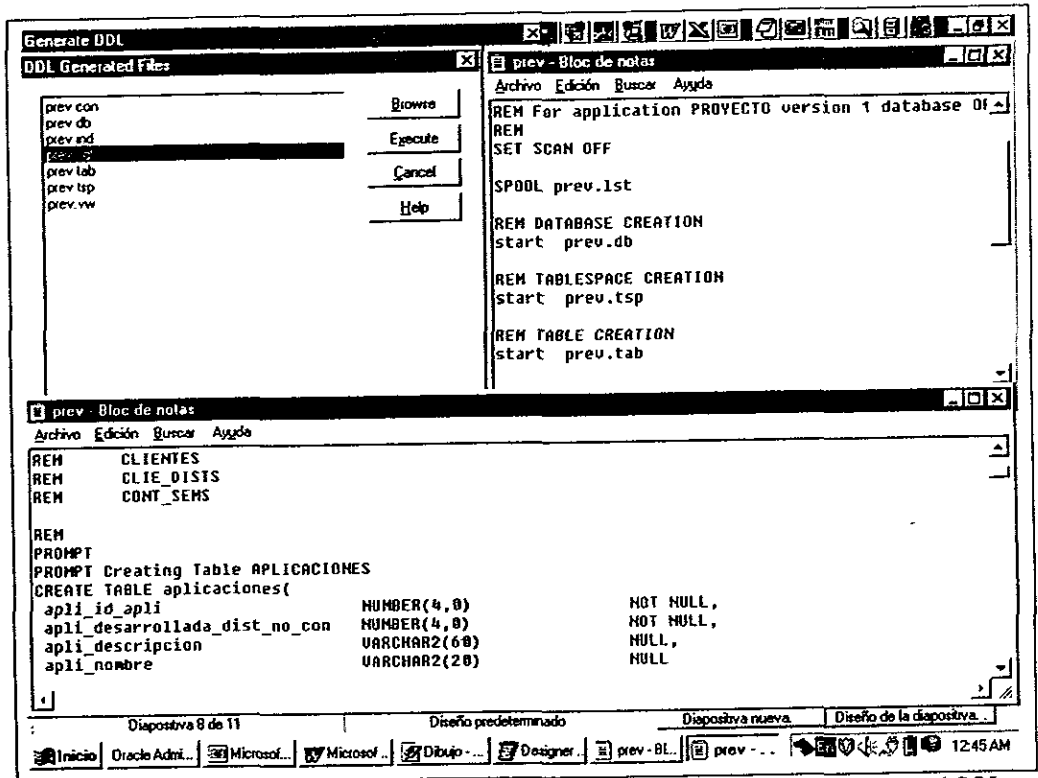


Figura 3.4.8. Vista del contenido de algunos de los archivos de comandos generados por el SQL

DDL Utility.

Una vez que se han corrido los scripts DDL se tiene la implementación de la base de datos definida en el repositorio. Esta base de datos contiene:

constraints	grants	users
indexes	packages	views
procedures	roles	tablespaces
rollback segments	sequences	

Ahora, esta base de datos, es práctica, funcional y su diseño está completamente documentado.

### 3.5 DISEÑO Y DESARROLLO DEL FRONT-END

#### Introducción al Generador para Web de Designer/2000

Para llevar a cabo la generación de la aplicación hacia el ambiente Web, es necesario que los pasos requeridos para hacer la ingeniería en reversa estén concluidos, así mismo, la generación de la base de datos a la versión actualizada de Oracle7, cuestiones que se darán por hecho en este inciso y posterior a esto es necesario haber instalado el Oracle Web Server y configurarlo. Cabe mencionar que el Oracle Web Server y Designer/2000 están íntimamente ligados, dado que sin el uno u otro no se podría hacer la regeneración de la aplicación que concierne en esta tesis, hacia el ambiente Web de manera transparente. El generador de Web de Designer/2000 ahorra el producir procedimientos almacenados en la base de datos hechos a mano puesto que usa un conjunto de herramientas para el desarrollo del Servidor de Web de Oracle, que realmente es un conjunto de paquetes que contienen procedimientos y funciones que se pueden invocar para generar HTML.

El Oracle WebServer es un servidor HTTP altamente integrado con la base de datos Oracle7, los modelos de datos del negocio almacenados en Oracle7 y formateados a documentos Web (HTML) dentro del WebServer y luego transmitidos hacia clientes Web, permite crear aplicaciones totalmente funcionales para un ambiente de Web, y así los usuarios pueden hacer consultas, actualizaciones, inserciones o borrado de información hacia una base de datos Oracle en WWW, como se muestra en la figura 3.5.1. Las aplicaciones generadas están basadas sobre el diseño de módulos y de base de datos que están almacenados en el repositorio de Designer/2000.



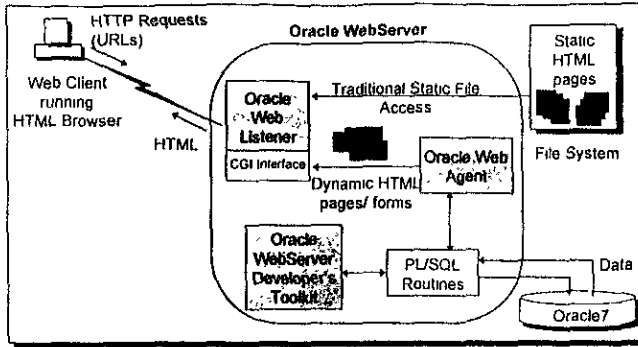
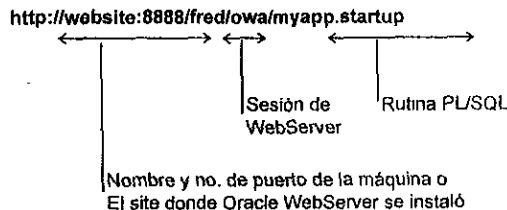


Figura 3.5.1 Aplicación corriendo en ambiente Web

La principal materia prima del proceso de generación son las especificaciones registradas a través del componente llamado Diagramador del Modelo de Datos de Designer/2000, que registra las tablas y las columnas además de ligas entre ellas, y una información detallada de cómo el módulo usa los datos. Adicionalmente, también necesita la liga entre módulos y preferencias, cabe mencionar que las ligas de módulos indican la navegación de la aplicación que fue creada por el Diagramador de Estructuras de Módulos, por otro lado, las preferencias indican y determinan el 'look & feel', es decir, determinan la presentación final de la aplicación hacia los usuarios.

Los usuarios en el Web requieren información HTML e identificarla con un apropiado URL, este URL especifica la localización del site del Oracle WebServer, la sesión de Web, es decir, la información para conectarse a la base de datos Oracle y la rutina PL/SQL a correr sobre el usuario de base de datos.

Ejemplo:



La rutina de PL/SQL especificada con el URL trae los datos de la base de datos Oracle y luego hace uso de otras rutinas de PL/SQL proveídas por el Oracle WebServer Developer's Toolkit para formatear los datos con la sintaxis requerida de HTML. Posteriormente los datos ya formateados son ruteados hacia el usuario de Web que originó el requerimiento vía el Web Agent.

El generador de Web crea aplicaciones totalmente funcionales para un ambiente de Web, que permite a los usuarios hacer consultas, actualizaciones, meter o borrar información hacia una base de datos Oracle en WWW. Las aplicaciones generadas están basadas sobre el diseño de módulos y de base de datos que está almacenado en el repositorio de Designer/200, como ya se dijo anteriormente, ver figura 3.5.2.

Durante el proceso de generación, el generador para Web crea un conjunto de paquetes PL/SQL, que son ejecutados para instalar la aplicación hecha en Web en el Servidor de Web de Oracle.

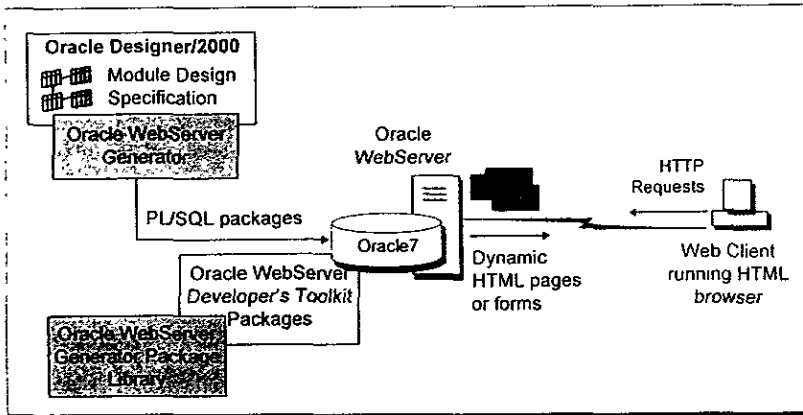


Figura 3.5.2 Proceso de Generación y aplicación de Web corriendo.

Antes de pasar propiamente a la generación de la aplicación, se explicará de manera muy somera como se configura el Web Server.

Se configura el Oracle Web Server usando el agente de Web por defecto que es `OWA_DEFAULT_SERVICE` éste usa el puerto #80 y el escuchador de Web es un servidor HTTP que maneja los requerimientos de cualquier Navegador. A continuación se presenta la pantalla de administrador del Web Server de Oracle:

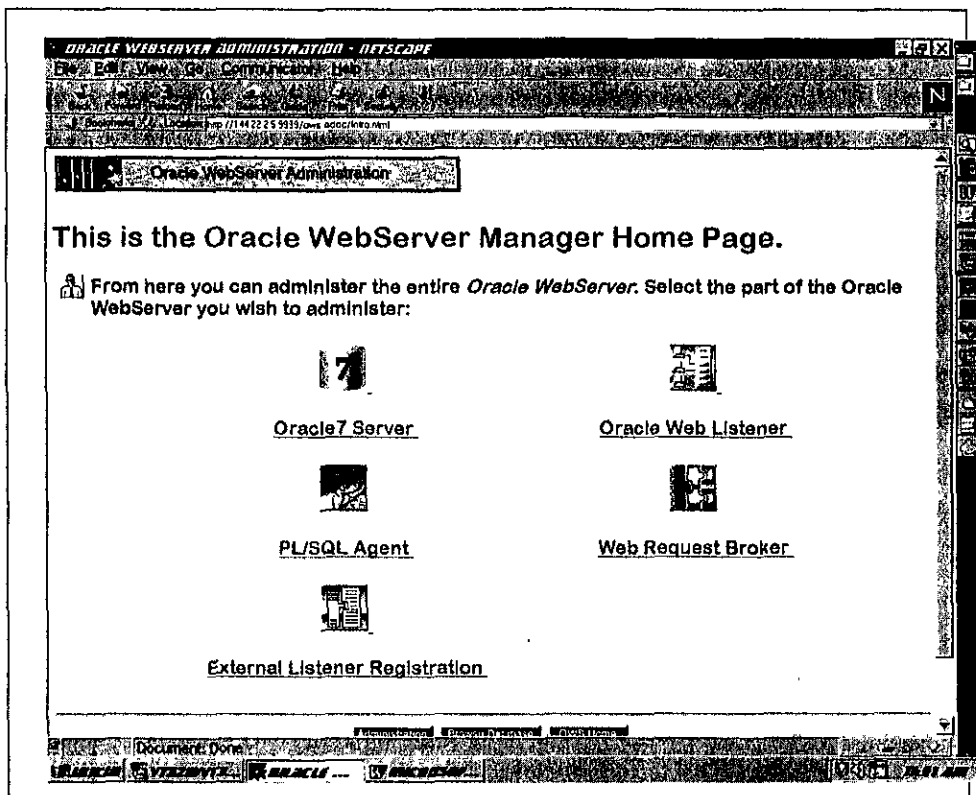
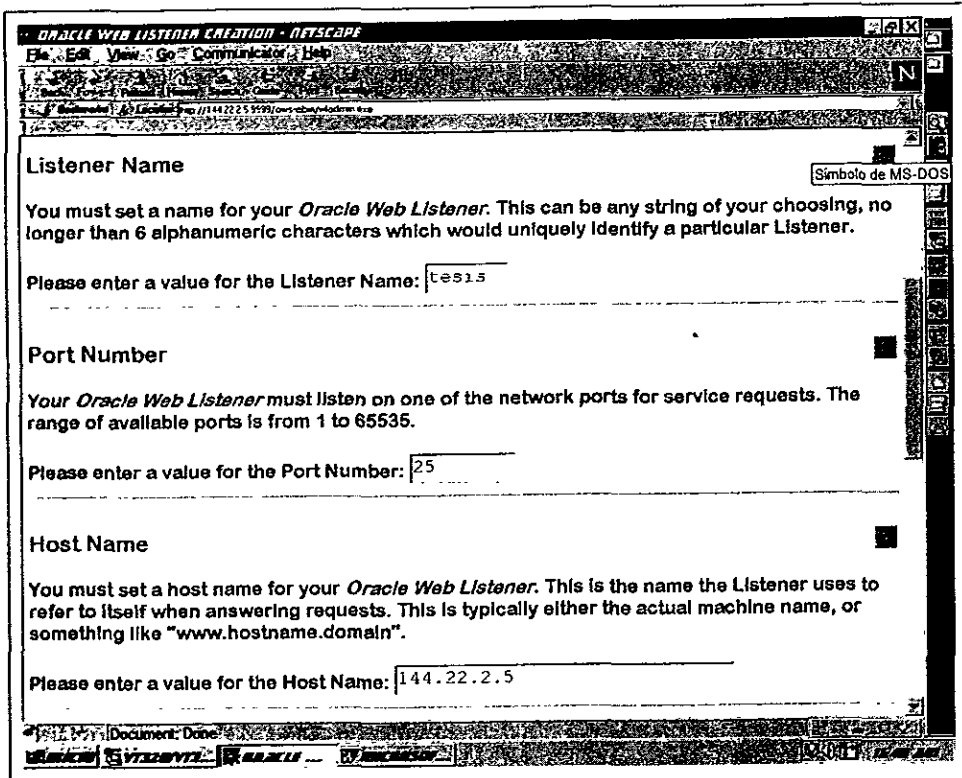


Figura 3.5.3 Administrador del Oracle WebServer.

Para el caso muy particular de esta aplicación se decidió tomar un escuchador de Web que no es el default para que a simple vista se identificara con el proyecto en cuestión, se creó uno nuevo con el nombre de TESIS y se direccionó al puerto #25, para lo cual se utilizó la siguiente pantalla:



**ORACLE WEB LISTENER CREATION - NETSCAPE**

File Edit View Go Communicator Help

Listener Name Símbolo de MS-DOS

You must set a name for your *Oracle Web Listener*. This can be any string of your choosing, no longer than 6 alphanumeric characters which would uniquely identify a particular Listener.

Please enter a value for the Listener Name:

Port Number

Your *Oracle Web Listener* must listen on one of the network ports for service requests. The range of available ports is from 1 to 65535.

Please enter a value for the Port Number:

Host Name

You must set a host name for your *Oracle Web Listener*. This is the name the Listener uses to refer to itself when answering requests. This is typically either the actual machine name, or something like "www.hostname.domain".

Please enter a value for the Host Name:

Document: Done

Figura 3.5.4 Página para creación del Listener.

Y adicionalmente, para poder parar, levantar o configurar el escuchador o escuchadores que se hayan creado en el Servidor de Web de Oracle se usa la siguiente pantalla:

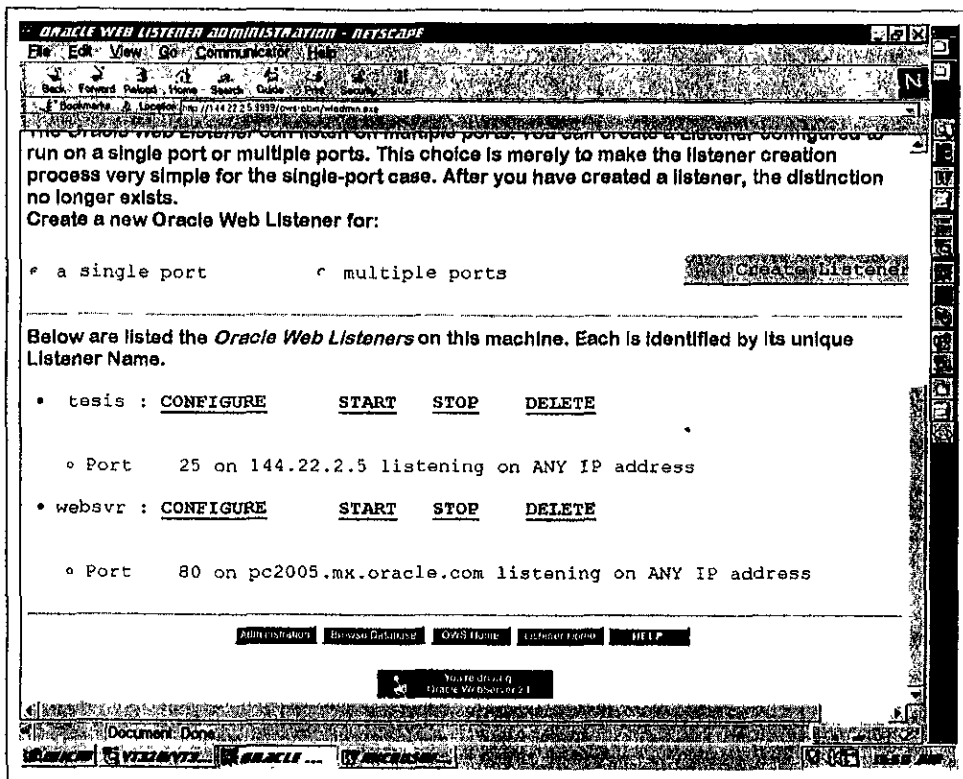


Figura 3.5.5 Página para la administración de los Listeners creados.

Cuando el escuchador de Web recibe un URL que especifica el requerimiento para ejecutar un procedimiento almacenado en la base de datos, éste manda el requerimiento al WRB (Web Request Broker) que ya fue explicado en el capítulo 1, el cual se comunica con el Agente PL/SQL que finalmente produce una conexión al Servidor de Base de Datos Oracle a través de SQL\*Net y ejecuta el procedimiento de PL/SQL, el procedimiento almacenado en la base de datos produce HTML que es pasado de regreso al escuchador de Web mediante el WRB y finalmente el escuchador lo envía al navegador, es decir, al cliente que lo solicitó.

### Configurando el Generador para Web de Designer/2000

La única manera por la cual se puede acceder al repositorio de Designer/2000 es precisamente usando la herramienta Designer/2000, donde reside el análisis y diseño de la aplicación y que es necesario para la generación. Se puede usar Designer/2000 1.2ª o mayor, sin embargo, hay que asegurarse que coincidan las versiones tanto del Cliente como del repositorio de Designer.

Estrictamente hablando, para la generación de paquetes, no es necesario acceder al Servidor de Web. El generador de Web produce tres archivos: un archivo maestro con extensión .sql, un archivo con extensión .pks el cual contiene los comandos para crear las especificaciones de los paquetes y un .pkb que contiene los comandos para crear el cuerpo o contenido de los paquetes.

Para la regeneración de este sistema a Web fue necesario instalar y ejecutar los paquetes ya generados para poder ver como iba quedando la generación de la aplicación:

- 1.- Se usó la herramienta de administración del Servidor de Web de Oracle para crear un **DCD** (descriptor de conexión de la base de datos) . Este servicio debe ser ligado a un usuario válido de base de datos, sin embargo, no es necesario que este usuario sea el mismo que el usuario del repositorio. Para minimizar confusiones, se nombró el DCD como un usuario de la base de datos. También se asoció el DCD con uno o más escuchadores de Web. Además también se instalaron las herramientas de desarrollo para el Servidor de Web que como ya se dijo antes es un número de paquetes que están referenciados por la librería del generador para Web de Designer/2000 (WSGL.sql).
- 2.- Para cada escuchador de Web se definió un directorio virtual con el mismo nombre que el DCD
- 3.- Se instaló la librería WSGL.sql dentro del usuario DCD de la base de datos

4.- Se configuró el generador para Web de la siguiente forma: Se especificó el usuario de la base de datos , su contraseña y el string de conexión del DCD, en el campo de Web Agent URL se especificó lo siguiente:

`http://<máquina>:<escuchador>/<el nombre del DCD>/owa`

que para el caso de la aplicación es:

`http://144.22.2.5:25/tesis/owa`

Con esta información el generador para Web construye automáticamente un URL para poder ver el paquete ya generado con el navegador que se desee.

Y finalmente para ver lo que se generó desde el navegador se teclea:

`http://144.22.2.5:25/tesis/owa/test$.startup`

### Cómo trabaja el Generador para Web de Designer/2000

El generador toma tres entradas principales para el proceso de generación, esas entradas principales son Preferencias de Usuario, Diseño de la base de datos y el Diseño de los módulos.

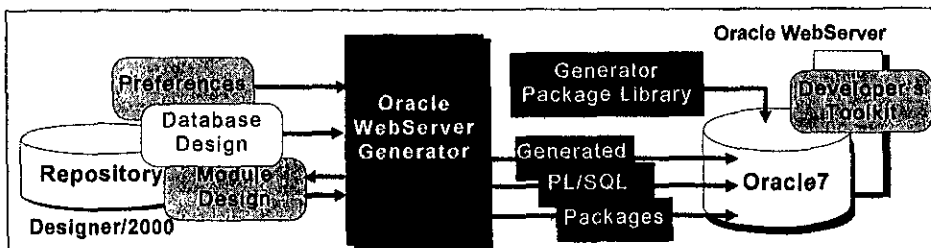


Figura 3.5.6 Principales entradas para la generación hacia Web.

- Preferencias .- Se pueden introducir a través del navegador de preferencias que permite definir las características del formato de presentación (layout) y conducta de las aplicaciones generadas. Se pueden poner las preferencias y afinarlas hasta llegar a la aplicación deseada o también dejar las preferencias que trae por default el producto Designer/2000. Las preferencias definen el estilo general de la aplicación, se pueden hacer diseños del formato de presentación por pantalla o por página, lo que significa que es necesario definir las una vez para activar el look & feel en toda la aplicación.

Oracle Designer/2000 incorpora un número de preferencias de usuario para customizar muchos aspectos de una aplicación dada. Para el generador de Web de Designer/2000, las preferencias de usuarios son proveídas para permitir que el usuario customice los siguientes aspectos de una aplicación:

- Colores
  - DBA
  - Frame Layout
  - Generales
  - Encabezados y Pies de páginas
  - Lista de valores
  - Formato para pantalla de inserción de datos
  - Formato para pantallas de consultas
  - Formato para presentar la lista de registros
  - Contenido de la primera pantalla (Startup Page)
  - View form layout
- 
- Diseño de la base de datos .- Mediante el uso de Designer/2000 se pueden definir todos los componentes del diseño de la base de datos y guardarlos en el repositorio para después generar lo que se le denomina **DDL** (Lenguaje de Definición de Datos) que crea los objetos de la base de datos físicamente en el servidor, así mismo, es posible dejar por default las características de despliegue de las columnas de una tabla, vista o snapshot. Esas



características de despliegue son usadas en el Diagramador de Módulos de Datos, que finalmente se utilizan en el generador. También es posible definir restricciones (constrains) o reglas del negocio con disparadores (triggers) y procedimientos almacenados en la base de datos y asociarle un despliegue de errores más amigable. Además contribuye con tres aspectos principales para la generación de una aplicación en Web: el primero de ellos es contra cada definición de columnas, que es posible jugar con la manera de despliegue de la columna; el segundo es que se puede especificar varias reglas que no se hubiese podido poner dentro del diseño de la base de datos, por ejemplo se pueden establecer relaciones entre tablas que se podrían implementar en el server via el DDL o del lado del cliente usando el apropiado client-side para las aplicaciones Web estas reglas serán implementadas usando JavaScript el cual es ejecutado por algún Browser; tercero el diseño de la base de datos puede tener definiciones procedurales de reglas de negocios las cuales son implementadas como disparadores (triggers) de base de datos o procedimientos almacenados en el servidor.

- Diseño de Módulos .- Un módulo para designer/2000 es un programa que forma parte de una aplicación, una parte importante de información registrada acerca de un módulo es la definición de los datos que el módulo accede. Se debe de escoger qué tablas serán la base del módulo, así mismo, se puede definir qué tipo de acceso se va a tener de las columnas de las tablas, vistas o snapshots como podría ser de inserción, consulta, actualización, o borrado. Una aplicación de Oracle Web Server generada se comprende de una o más páginas dinámicas de HTML, cada página generada contiene una o más secciones que despliega texto estático HTML y datos obtenidos de una consulta basada en SQL. Los componentes de los módulos forman las bases para crear los queries hechos en SQL y están definidos visualmente en el Diagramador de Módulos de Datos.

Una definición sencilla de módulos puede contener uno o más componentes ligados mediante llaves foráneas, así mismo, los componentes de los módulos están basados sobre una tabla específica o tabla base pero también pueden hacer uso de otras tablas para ofrecer información más a detalle (lookup tables).

La definición del módulo ilustrado en la figura 3.5.7 tiene dos componentes, uno basado sobre la tabla de CONTACTOS y otro sobre la tabla de CONT\_SEMS.

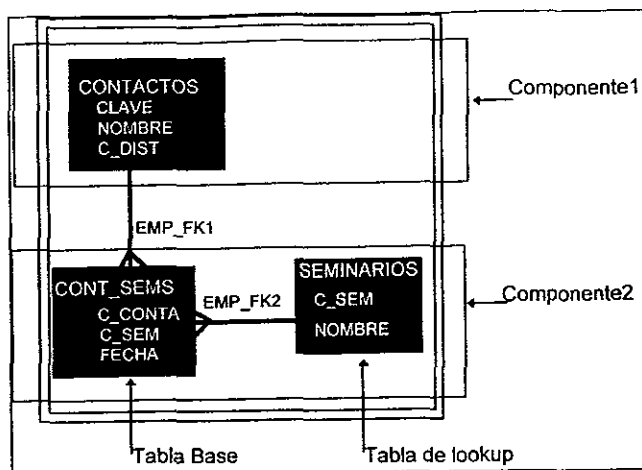


Figura 3.5.7 Definición de un Módulo.

Para cada componente se deben definir cuales van a ser las columnas que se van a incorporar dentro de la aplicación, su conducta y las características de despliegue, es decir, si es un campo que permite inserción, actualización, cual va a ser su etiqueta, algún formato especial HTML, etc.

### Proceso de Generación

- **Proceso de generación para el Servidor de Web.** Primeramente se escoge el módulo que se va a generar y también qué módulos se van o no a generar que son llamados directa o indirectamente por el módulo escogido por el generador para Web, el cual, examina las especificaciones de diseño de cada módulo, detecta las ligas entre módulos y determina los valores de preferencias.

Durante la generación, el generador crea dos tipos de archivos de PL/SQL para cada módulo:

- El archivo de especificación del paquete .PKS
- El archivo del cuerpo o contenido del paquete .PKB

Un paquete PL/SQL es creado para cada componente definido dentro de un módulo, más uno para el módulo en sí mismo, rutinas dentro de esos paquetes proveen la funcionalidad requerida en páginas de Web.

También como resultado de la generación se obtiene un archivo .SQL el cual simplemente llama a los dos ya mencionados anteriormente. Posteriormente se podrá ejecutar este archivo para la instalación de la aplicación ya generada dentro de la base de datos Oracle7, para que después se pueda ejecutar la aplicación desde cualquier navegador.

En suma, para generar paquetes el generador para Web hace uso de su propia librería de paquetes PL/SQL (WSGL) para proveer funcionalidad genérica dentro de la aplicación.

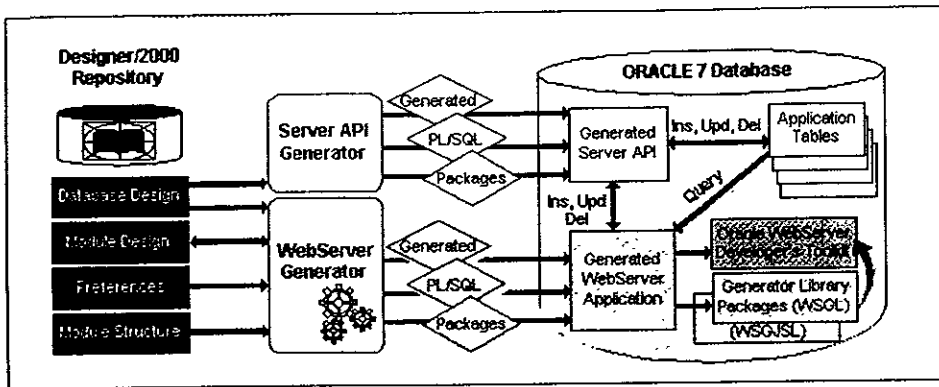
- Proceso de generación para API PL/SQL. La aplicación ya generada para Web da mantenimiento a los datos mediante un API servidor, esto significa que el API servidor debe existir en la base de datos antes de que la aplicación ya generada sea instalada.

El API servidor es generado de las definiciones de las tablas que están residiendo en el repositorio de Designer/2000, antes de generar un servidor de API se debe asegurar que el diseño de las tablas esté terminado, si se hicieran cambios al diseño de las tablas entonces el servidor API debe regenerarse. Cuando se genera el API se deben escoger las tablas a las que se va a dar mantenimiento mediante los módulos que se escogieron generar y entonces se generan un par de archivos que son:

- archivo de especificación de paquete                      tabla\_identificador.PKS
- archivo de cuerpo o contenido del paquete              tabla\_identificador.PKB

Un paquete PL/SQL es creado para cada tabla y esos paquetes contienen procedimientos que hacen inserciones, actualizaciones, borrados y candados sobre la tabla.

Al igual que el anterior se crea un archivo .SQL el cual llama a los .PKS y .PKB y al correr este archivo se instala el servidor API en la base de datos.



Figura

Figura 3.5.8 Proceso de Generación.

### Tipos de Páginas que Genera Oracle WebServer

A partir de la especificaciones hechas en el diseño del módulo, Oracle WebServer genera un conjunto de paquetes los cuales al momento de corrida de la aplicación construye dinámicamente una serie de páginas ligadas de Web, las páginas que se crean pueden ser clasificadas en siete tipos:

- **Página de comienzo (Startup page).** Es creada por cada módulo dentro de la aplicación generada, si el módulo es de alto nivel entonces esta página provee un conjunto de ligas de hipertexto para el resto de los módulos dentro de la aplicación; para los otros módulos, esta página sirve como entrada al módulo y puede, si es apropiado, desplegar la forma de consulta.

Las ligas entre módulos pueden definirse en el Diagramador de Estructuras de Módulos, las llamadas a otros módulos pueden ser organizadas dentro de una estructura jerárquica mediante la inclusión de módulos tipos MENU, las llamadas son desplegadas como ligas de hipertexto. Si es requerido, el generador puede incluir una página la cual despliega el detalle sobre la generación del módulo en cuestión (About page) y es llamada mediante una liga de hipertexto. Para el caso de esta tesis no se utilizaron este tipo de páginas, sin embargo, para ejemplificarlas se presenta la siguiente pantalla:

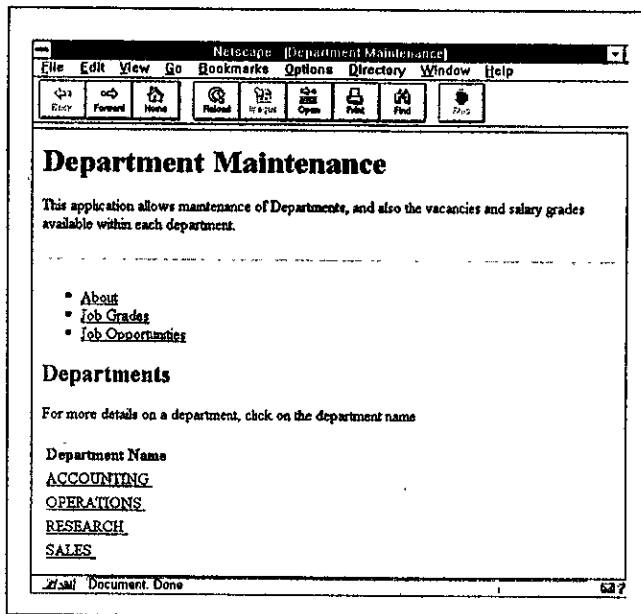


Figura 3.5.9 Ejemplo de página Startup.

- Forma de Consulta. Permite al usuario meter ciertos criterios de búsqueda a ser incorporado cuando se ejecuta una consulta, esta página despliega las columnas que fueron marcadas con el atributo de 'Select' en el diseño del módulo.

Una forma de Consulta solo es generada si al menos una columna es marcada con el atributo anteriormente mencionado, de lo contrario solo presenta una página con una consulta a la base de datos sin restricciones.



Figura 3.5.10 Ejemplo de forma de consulta.

- Lista de registros. Despliega un conjunto de registros que son buscados en la tabla base o en tablas lookup ya explicadas con anterioridad en este mismo apartado, normalmente despliega un subconjunto de columnas que facilitan al usuario la identificación del registro con facilidad, además una forma de detalles es generada para presentar el detalle de cualquier registro seleccionado en la lista.

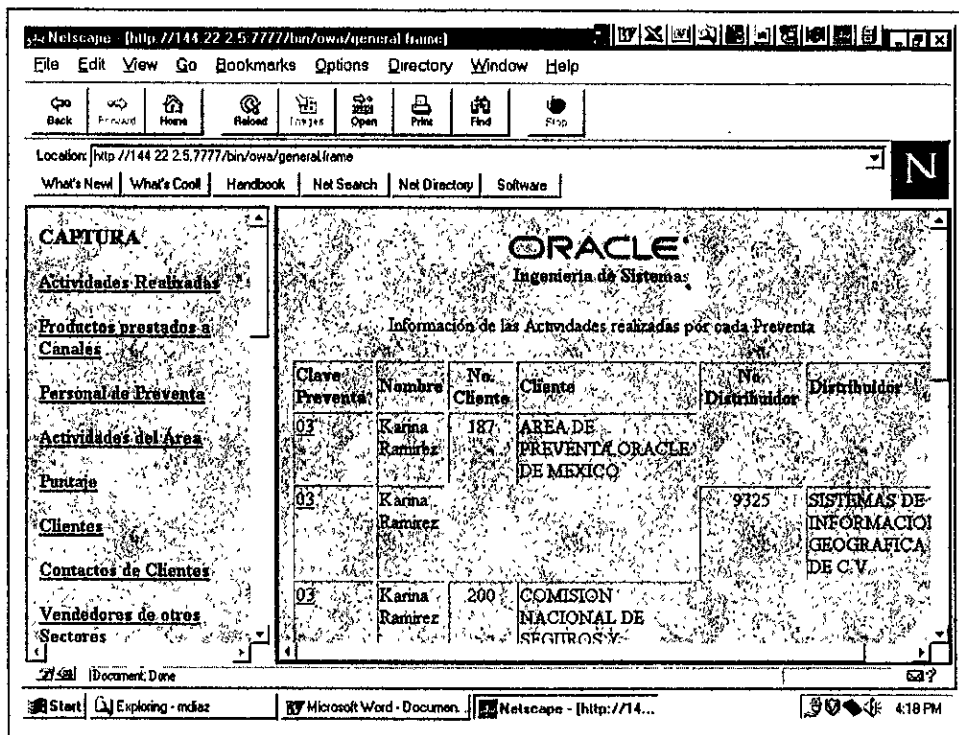


Figura 3.5.11 Ejemplo de una página de lista de registros.

- Forma de detalles. Esta forma despliega los detalles de cualquier registro, contiene todas las columnas definidas en el diseño del módulo. Cuando un usuario escoge una entrada de hipertexto en el tipo de forma que se describió en el punto anterior, entonces se despliega la forma de detalles en una nueva página de Web.

Si el usuario modifica un campo y presiona el botón de actualización, la aplicación trata de poner un candado en el registro en cuestión en la copia de información que se obtuvo al traer dichos datos desde la consulta. Si la operación tiene éxito entonces la actualización es escrita a la base de datos y la página de Web es resfrescada. Si otro usuario ha modificado cualquier valor desplegado en la aplicación, entonces la operación que trata de poner el candado del otro usuario falla y es notificado dicho usuario que otro usuario ha modificado la información. En el caso de que en el diseño del módulo tenga componentes que actúen como detalles, éstos pueden ser desplegados en la misma página. La colocación de los componentes maestro-detalle en la misma o en páginas separadas es definido en el Diagramador de Módulos de Datos.

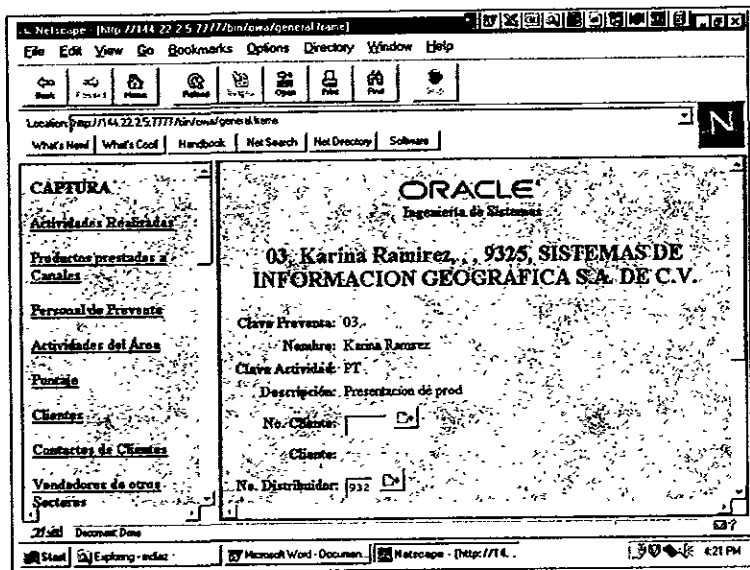


Figura 3.5.12 Ejemplo de página de detalles.

- Formato Lista/Forma. El formato Lista/Forma permite al generador de WebServer explotar Frames (marcos) para desplegar una lista de registros y una forma de detalles en la misma página. Cuando se selecciona un registro de la lista de registros su detalle es desplegado dentro de la forma de detalle. Este tipo de pantallas tampoco fueron utilizadas en el sistema, sin embargo, se muestra a continuación un ejemplo.



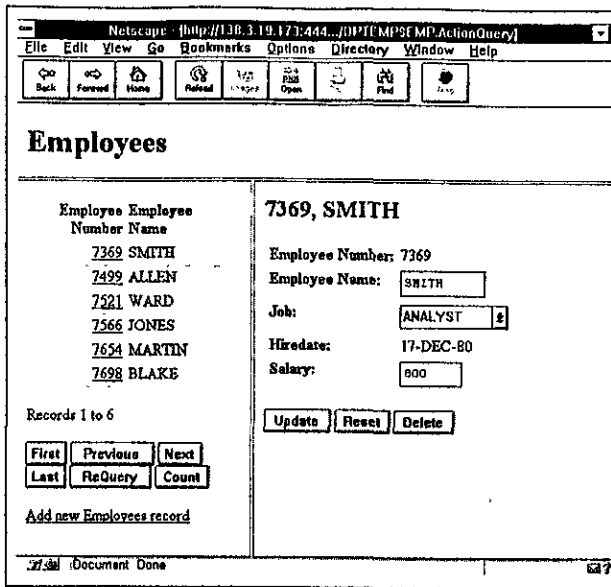


Figura 3.5.13 Ejemplo de página Lista/Forma.

- Forma para Inserción. Permite al usuario introducir nuevos registros a la base de datos para un módulo en particular, esta página despliega un conjunto de columnas que fueron marcadas como 'Insert' en el diseño del módulo.

Cuando se selecciona 'Agregar registro de ..', el cual esta localizado en la parte de abajo de la lista de Registro, la forma de insercion es desplegada.

La apariencia de este HTML está determinada en el conjunto de preferencias de la Forma de Inserción.

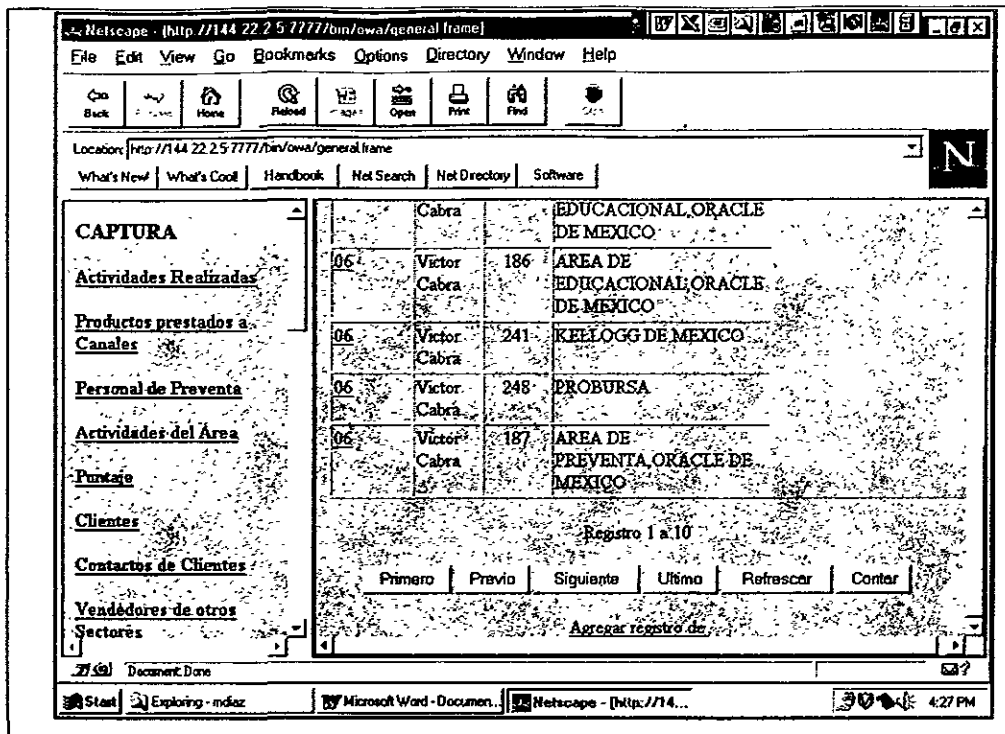


Figura 3.5.14 Ejemplo de página para inserción (“Agregar registro de ..”).

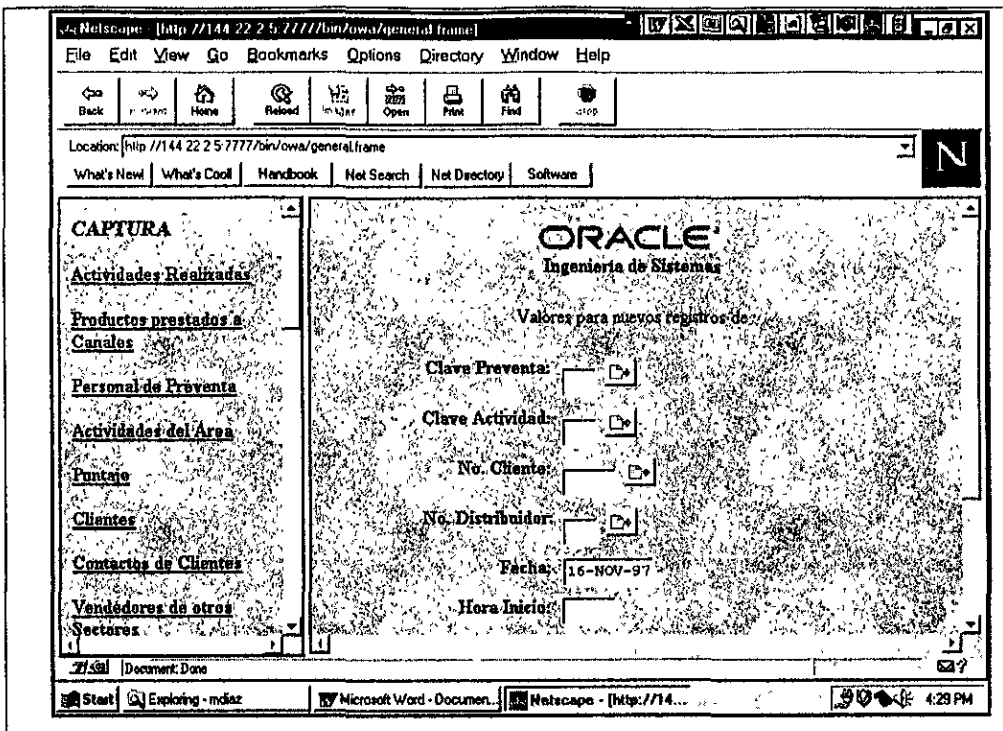


Figura 3.5.15 Ejemplo de página para inserción.

- Forma para borrado. La Forma para confirmación de borrado permite a los usuarios, como su nombre los dice, confirmar si quieren o no borrar un registro. Esta forma se despliega cuando se presiona el botón de 'Borrar' sobre una forma de detalles y en navegador no soporta JavaScript. Si el navegador soporta JavaScript un mensaje Pop-up es desplegado.

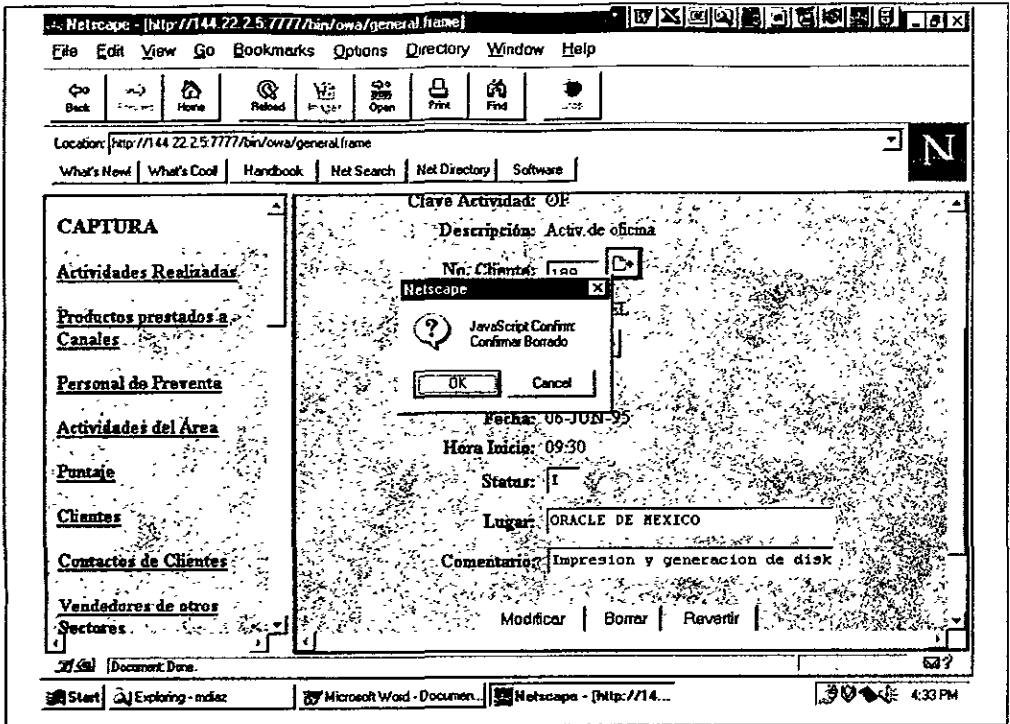


Figura 3.5.16 Ejemplo de página para borrar información.

Adicionalmente se generaron reportes en el sistema, la manera de hacer reportes con el generador de Designer/2000 para Web es generar una lista de registros y a continuación se presentan algunos de ellos.

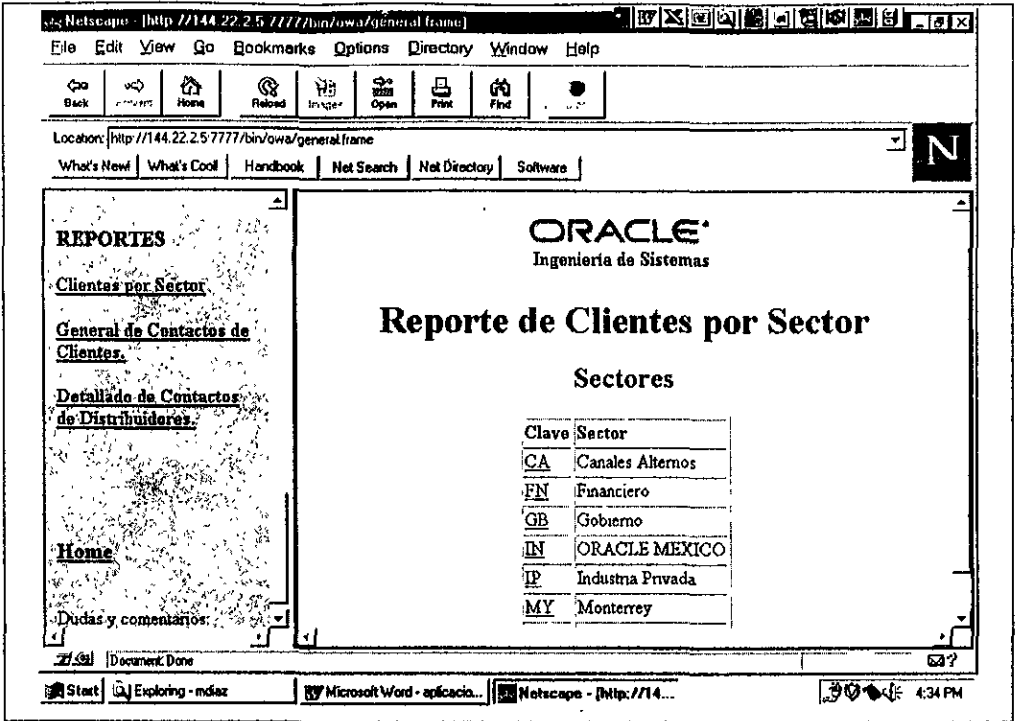


Figura 3.5.17 Reporte de Sectores.

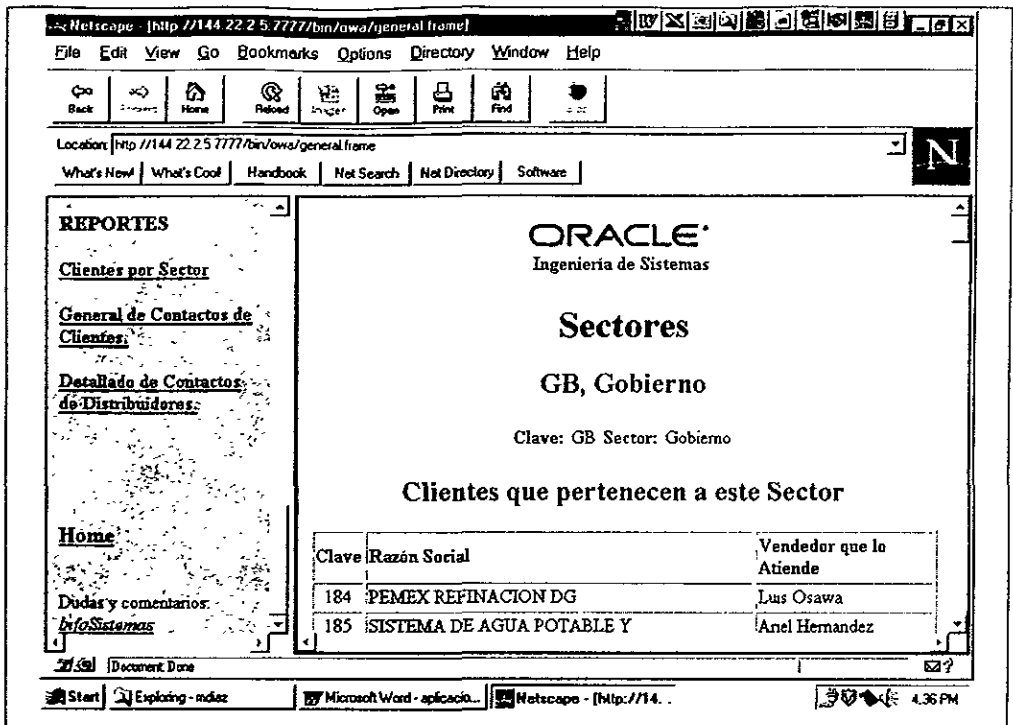


Figura 3.5.18 Reporte de Clientes en el Sector Gobierno.

Así mismo, cuando se trata de hacer un reporte en dónde se le pasen algunos parámetros, primero se hace una forma de Consulta y después una Lista de Registros, se muestra un ejemplo a continuación.

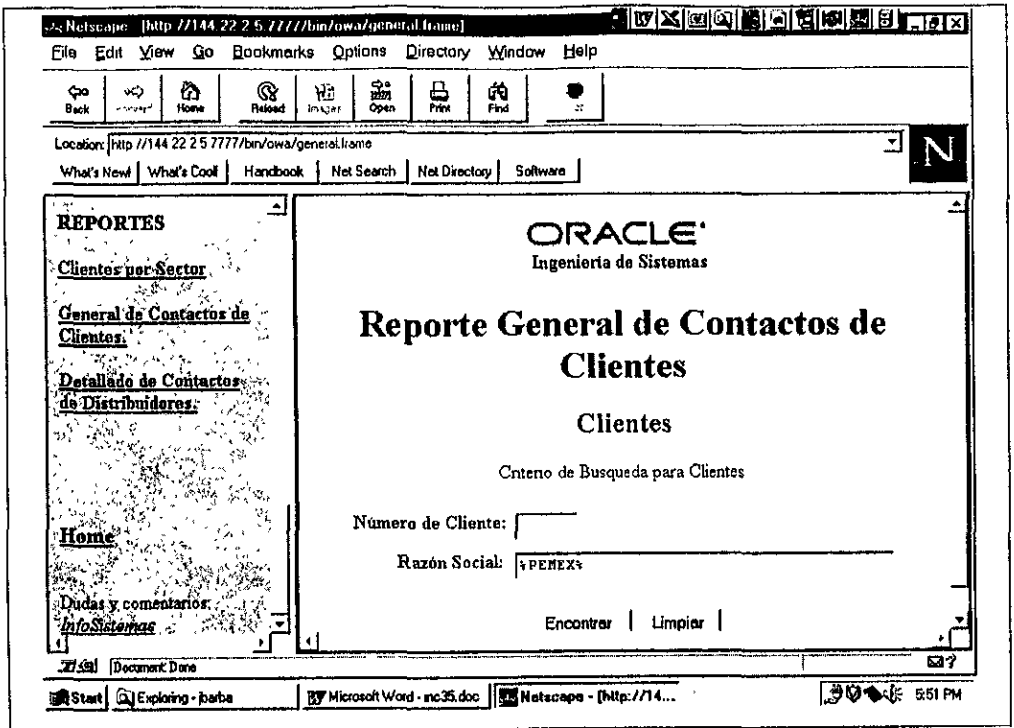


Figura 3.5.19 Ejemplo de un reporte basado en una forma de consulta.

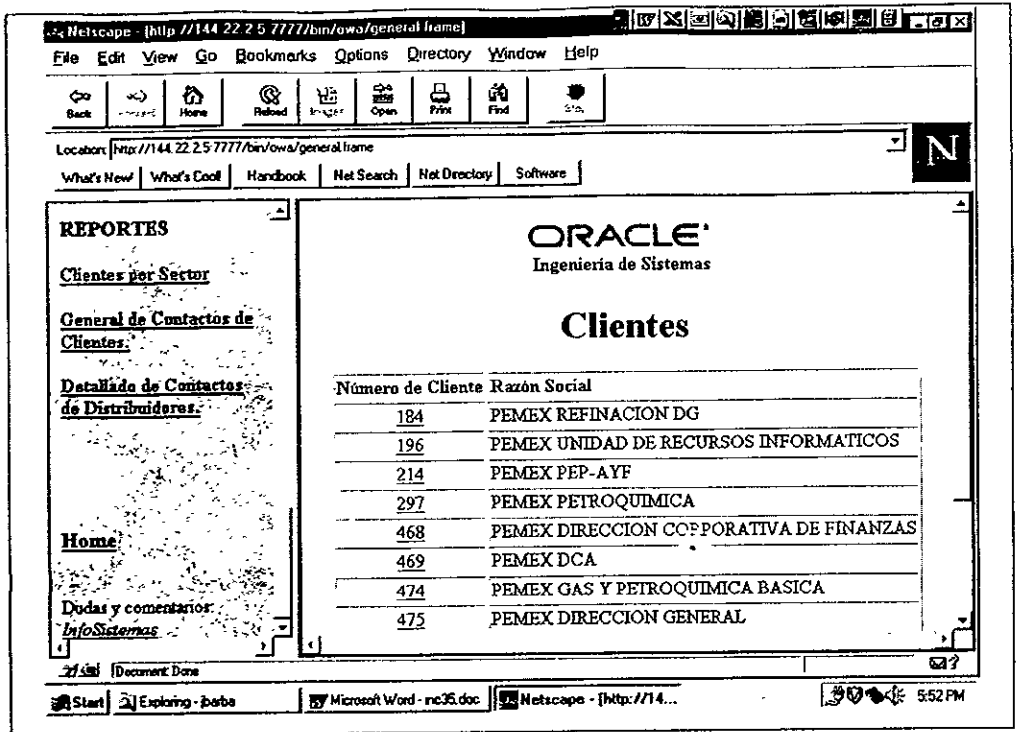


Figura 3.5.20 Ejemplo de un reporte con paso de parámetros vía forma de consulta.



### 3.6 INTEGRACIÓN, PRUEBAS Y TRANSICIÓN DEL SISTEMA

Después de haberse generado de manera completa el conjunto de módulos de la aplicación se procedió a integrarlos para conformar el sistema en su totalidad, durante el proceso de integración se realizó también de manera paralela un conjunto de pruebas y validaciones para cada módulo de la aplicación con el objetivo de detectar el mayor número de errores posibles antes de liberar el sistema a producción. Cabe hacer notar que se consideró de manera importante la participación del usuario final en el desarrollo de las pruebas y en la alimentación del sistema con datos válidos e inválidos para comprobar el buen funcionamiento de la aplicación y la corrección de las partes imperfectas, con el objetivo de lograr un sistema que cumpla con las expectativas del usuario.

En el proceso de integración se procedió a identificar las partes componentes de cada menú de la aplicación tomando en consideración el diagrama de la figura 3.1.5

Para el desarrollo de las pruebas y validación del sistema se consideraron dos estrategias de prueba, la *prueba ascendente* y la *prueba descendente*, a continuación se dará una breve descripción de cada una de estas estrategias.

#### Estrategias de Prueba

##### a) *Prueba Ascendente*

El enfoque ascendente empieza por probar módulos individuales pequeños en forma separada; esto a menudo se conoce como prueba de unidades, prueba de módulos, o prueba de programas. Luego los módulos individuales se combinan para formar unidades cada vez más grandes que se probarán en masa; esto se conoce como prueba de subsistemas. Finalmente, todos los componentes del sistema se combinan para probarse; esto se conoce como prueba del sistema, y suele estar seguido de las pruebas de aceptación, donde se permite al usuario usar sus propios casos de prueba para verificar que el sistema esté trabajando de manera correcta.

**b) Prueba Descendente**

El enfoque de prueba descendente empieza con un esqueleto del sistema; es decir, la estrategia de prueba supone que se han desarrollado los módulos ejecutivos de alto nivel del sistema, pero que los de bajo nivel existen sólo como módulos vacíos. Dado que muchas de las funciones detalladas del sistema no se han implantado, las pruebas iniciales están muy limitadas; el propósito es simplemente comenzar a ejercitar las interfaces entre los subsistemas principales. Las pruebas siguientes abarcan y tratan aspectos cada vez más detallados del sistema.

Después de haberse analizado ambas estrategias se decidió optar por la *prueba ascendente* debido a que esta estrategia presupone la existencia de módulos inferiores completamente desarrollados (a diferencia de la *prueba descendente* donde los módulos inferiores se presuponen como vacíos), y a que la aplicación que se ha generado parte también de módulos completamente desarrollados como el resultado de aplicar la metodología de la Ingeniería en Reversa de una aplicación ya existente.

Una vez elegida la estrategia de prueba se procedió a definir los diferentes *tipos de pruebas* que se desarrollarían junto con el usuario, se definieron pruebas de tipo funcional, pruebas de recuperación y pruebas de desempeño. A continuación se expone una breve descripción de cada uno de estos tipos de pruebas.

**Tipos de Pruebas****a) Prueba Funcional**

Esta es la forma más común de prueba; su propósito es asegurar que el sistema realiza sus funciones normales de manera correcta. Así, los casos de prueba se desarrollan y se alimentan al sistema; las salidas se examinan para ver si son correctas.

**b) Prueba de Recuperación**

El propósito de este tipo de prueba es asegurar que el sistema pueda recuperarse adecuadamente de diversos tipos de fallas. Esto es de particular importancia en los sistemas en línea grandes, al igual que en varios tipos de sistemas de tiempo real que controlan dispositivos físicos y/o procesos de fabricación. Las pruebas de recuperación pueden requerir que el equipo que realiza el proyecto simule (o provoque) fallas de hardware, fallas de corriente, fallas en el sistema operativo, etc.

**c) Prueba de Desempeño**

El propósito de este tipo de prueba es asegurar que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación del usuario, además de asegurar que tenga el tiempo de respuesta requerido. Esto puede requerir que el equipo que realiza el proyecto simule una gran cantidad de conexiones al sistema, de manera de que se pueda engañar al sistema para que “crea” que está operando con una gran carga.

A continuación se mencionará el procedimiento de pruebas y validaciones que se implementaron para comprobar el buen funcionamiento del módulo “Información sobre el Cliente”.

Es importante resaltar que el conjunto de acciones realizadas para este módulo se llevaron a cabo de la misma manera para los módulos restantes.

**Módulo “Información sobre el Cliente”***Pruebas de Tipo funcional*

Para cada una de las pantallas que consiste el módulo de “Información sobre el Cliente” se realizaron pruebas de validación de datos tales como incluir valores negativos, valores no existentes en la base de datos, dejar los campos con valores nulos, asignación de datos a través de listas de valores, por otro lado se validó también la navegación entre las pantallas, así como la obtención de la información mediante criterios de búsqueda.

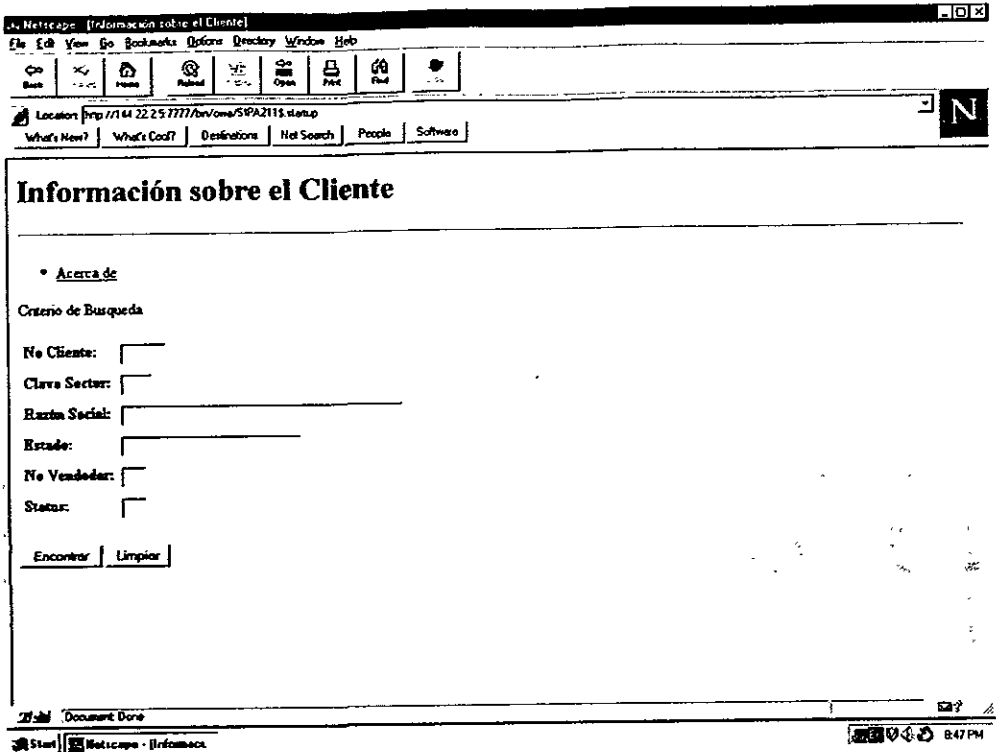


Figura 3.6.1 Información sobre el Cliente

De la figura 3.6.1 se muestra la pantalla de inicio del módulo “Información sobre el Cliente”, esta pantalla nos permite definir criterios de búsqueda solicitando la información mediante el botón “Encontrar” y para el caso que se desee restablecer los campos de la pantalla para especificar otro criterio de búsqueda basta con oprimir el botón “Limpiar”. Para esta pantalla se realizaron pruebas de validación para cada uno de los campos de los datos permitidos con el propósito de que la aplicación detectara cuando un dato era válido o cuando este no lo fuera, por ejemplo que el número del vendedor fuera un número entero positivo, que la clave del sector en caso de no encontrarse en la base de datos se pudiera definir al momento de agregar un registro, etc. A continuación se muestra el resultado de dejar los campos con valores nulos (la aplicación interpreta esto como recuperar toda la información disponible de todos los clientes almacenados en la base de datos) y oprimir la tecla “Encontrar”.

Información sobre Clientes

No Cliente	Razón Social
000138	BANCO NACIONAL DE MEXICO
000156	BANCOMER S.A.
000157	ABN-AMRO BANK, N.V.
000176	RADIO LASER, S.A. DE C.V.
000177	CHICLE ADAMS / WARNER LAMBERT
000178	AEI SISTEMAS
000179	BUFETE INDUSTRIAL
000180	FISHER CONTROLES DE MEXICO, S.A. DE C.V.
000181	ORGANIZACION EDITORIAL MEXICANA
000182	SERVICIOS AEROS DEL CENTRO (SACSA)

Registro 1 a 10

[Agregar registro de](#)

Figura 3.6.2 Información sobre el Cliente

Como puede observarse la información desplegada solo contiene los primeros diez registros del total de clientes, por lo que, para navegar entre los registros restantes nos podemos valer de los botones “Siguiente” para obtener el siguiente conjunto de diez registros, “Ultimo” para posicionarnos en los últimos diez registros, “Refrescar” para redibujar la pantalla, “Contar” para conocer el total de registros que encontró la aplicación de acuerdo al criterio de búsqueda definido en la pantalla anterior, etc. La información obtenida fue cotejada con los datos almacenados en la base de datos. Por otro lado si se desea obtener información mas detallada acerca de un cliente en particular basta solo con seleccionar el número de cliente de interés y se obtendrá información como la siguiente.

The screenshot shows a web browser window with the following content:

**138, BANCO NACIONAL DE MEXICO**

No Cliente: 000138  
 Clave Sector: FN  
 Razón Social: BANCO NACIONAL DE MEXICO  
 Calle: [Barranca del Muerto #24-3] ala norte  
 Colonia: [Barranca del Muerto]  
 CP: 0102  
 Ciudad: México  
 Estado: D.F.  
 Teléfono: 725-11-42  
 Fax:  
 No Vendedor: 06 LOZ  
 Nombre de Vendedor: Carlos Flores  
 Status: NO  
 Representante: Ing. Antonio Villegas  
 Comentarios:

Buttons: Modificar, Borrar, Revertir

System tray: 8:55 PM

Figura 3.6.3 Información sobre el Cliente

Esta pantalla nos despliega información de un cliente en particular proporcionándonos datos como el nombre, su clave, la razón social, etcétera. Se cuenta también con la facilidad de modificar datos del cliente tales como el nombre de la calle, la colonia, código postal, entre otros.

Se validaron el buen funcionamiento de los botones “Modificar” para hacer permanentes los cambios a la información, “Borrar” para eliminar el registro, “Revertir” para deshacer los cambios que se hubieran realizado a los campos y que no se desearan definir como permanentes; la validación a los cambios hechos se hicieron consultando directamente la información de la base de datos.

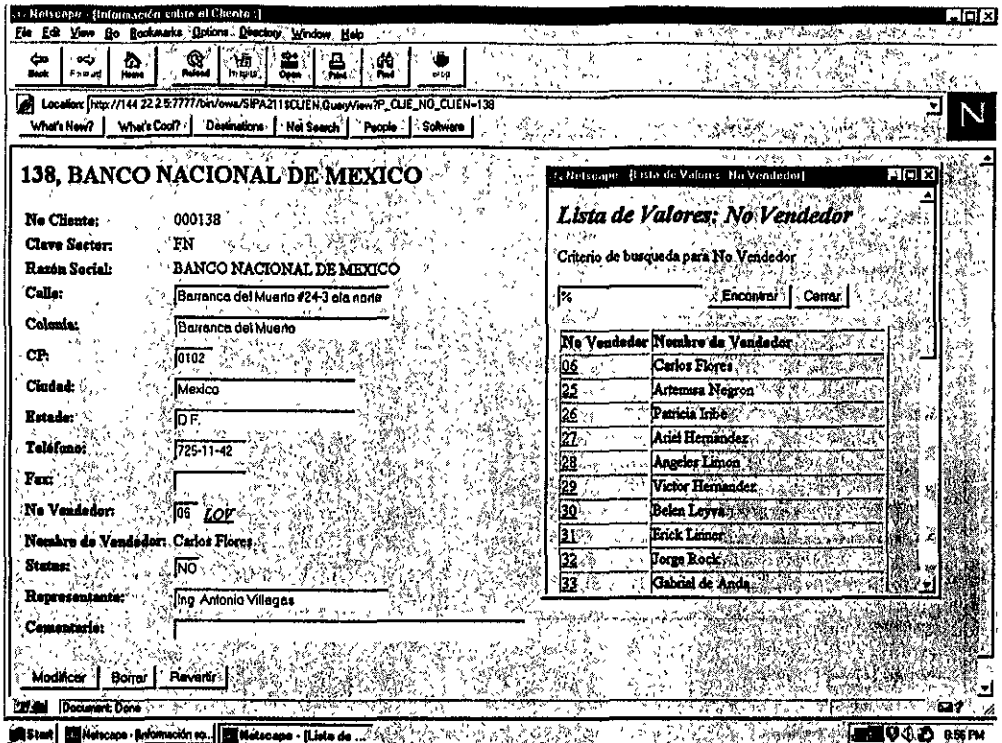


Figura 3.6.4 Información sobre el Cliente

Se validó también el uso de la lista de valores para el campo “Número de Vendedor”, esta facilidad es de mucha utilidad cuando se desea cambiar el vendedor asociado a un cliente y no se cuenta en ese momento con los números disponibles para cada vendedor, lo anterior se puede observar de la figura 3.6.4.

NetScape: Información sobre el Cliente

File Edit View Go Bookmarks Options Directory Window Help

Location: <http://144.22.2.57777/Am/owa/SPA2116DJEN/FormNew>

What's New? What's Cool? Customizations Net Search People Software

Valores para nuevos registros de:

Clave Sector:

Razón Social:

Calle:

Colonia:

CP:

Gender:

Estado:

Teléfono:

Fax:

No. Vendedor:

Representante:

Comentarios:

Document Done 9:57 PM

Figura 3.6.5 Información sobre el Cliente

Para lograr ingresar un nuevo cliente a la base de datos basta con seleccionar la línea “Agregar registro” de la pantalla representada por la figura 3.6.2 obteniéndose como resultado la pantalla que se aprecia en la figura 3.6.5, se pueden ingresar datos tales como la clave del sector al cual pertenece el cliente, su razón social, la calle, colonia, etc. También puede hacerse uso de las listas de valores como es el caso para los campos “Clave Sector” y “No. Vendedor”. Las listas de valores fueron validadas consultando directamente la información disponible en la base de datos. Se validaron asimismo el funcionamiento de los botones “Insertar” para agregar un registro de un cliente, y “Limpiar” para despejar los datos ingresados a la pantalla. Se comprobó también que no existiera duplicidad en la información, o que se estuviera violando algún constraint de tipo referencial o de llave primaria.



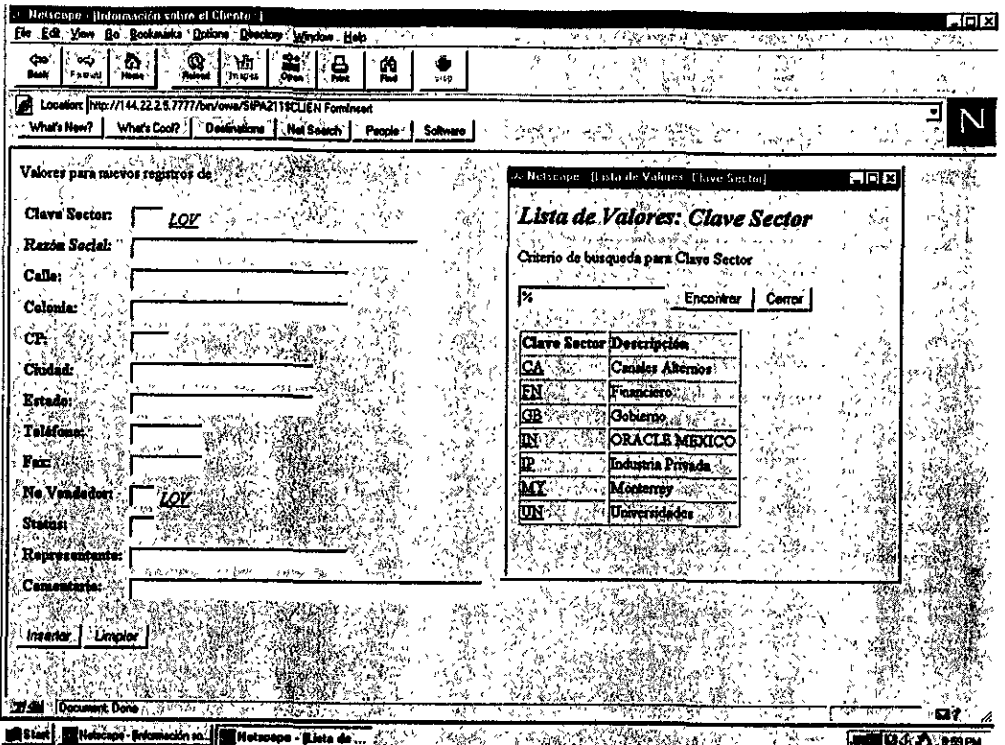


Figura 3.6.6 Información sobre el Cliente

Las figuras 3.6.6 y 3.6.7 muestran la funcionalidad de las listas de valores de que se disponen al momento de ingresar datos para dar de alta un nuevo cliente; se puede apreciar de las pantallas que las listas de valores nos permiten definir también criterios de búsqueda sobre el campo que se encuentra resaltado en color azul y subrayado definiendo el simbolo % como comodín para substituir cualquier número de caracteres. Se validó que tanto la clave del sector como el número del vendedor correspondieran a los datos almacenados.

The screenshot shows a web browser window with two main panels. The left panel is a form titled 'Valores para nuevos registros de:' with fields for 'Clave Sector:' (LOV), 'Razón Social:', 'Calle:', 'Colonia:', 'CP:', 'Ciudad:', 'Estado:', 'Teléfono:', 'Fax:', 'No Vendedor:' (LOV), 'Status:', 'Representante:', and 'Comentarios:'. Below the form are 'Insertar' and 'Limpiar' buttons. The right panel is a window titled 'Lista de Valores: No Vendedor' with a search criterion 'Criterio de búsqueda para No Vendedor' and a search bar. Below the search bar are 'Encontrar' and 'Cerrar' buttons. A table lists salespeople with columns 'No Vendedor' and 'Nombre de Vendedor'.

No Vendedor	Nombre de Vendedor
26	Cedros Flores
25	Artemisa Negraon
26	Patricia Eibe
27	Ariel Hernandez
28	Angelica Linao
29	Victor Hernandez
30	Belen Leyva
31	Erick Linao
32	Jorge Roca

Figura 3.6.7 Información sobre el Cliente

### Pruebas de Recuperación

La aplicación desarrollada, por su naturaleza, no requiere de un sistema sofisticado de recuperación en caso de fallas, por ejemplo debidas a una interrupción en la energía eléctrica, problemas causados por hardware o bien debido a problemas de software. El manejador de base de datos de Oracle provee de un mecanismo de resolución de transacciones pendientes como consecuencia de una falla en el sistema, colocando la información de la base de datos en un estado consistente.

Para propósitos de pruebas de recuperación se simuló una falla en la energía eléctrica después de haberse realizado cambios en los datos de clientes, de distribuidores, del personal de preventa,

etc., una vez restaurada la conexión se procedió a reinicializar la base de datos, a revisar el archivo de alert.log para observar si al momento de reinicialización se hubiera registrado algún problema, a continuación se generaron diversas consultas a la base de datos empleando los módulos de la aplicación obteniéndose información consistente.

### ***Pruebas de Desempeño***

Con el propósito de medir el performance de la aplicación se generaron 20 conexiones a la base de datos en forma simultanea, generando cada una de ellas transacciones de inserción, actualización, borrado y selección de datos a través de la red en horarios pico, obteniéndose al principio un tiempo de respuesta no aceptable, por lo que se procedió a modificar parámetros de la base de datos con el propósito de mejorar su rendimiento, una vez hecho lo anterior el tiempo de respuesta mejoró en gran medida.

### **Transición**

La etapa de transición es una parte crítica del sistema ya que se llevan a cabo todas las tareas necesarias para la implementación de la nueva aplicación; provee de un periodo inicial de soporte y capacitación a los usuarios, y mantenimiento de ambos sistemas en paralelo hasta que se tiene la seguridad de que los empleados han entendido y aceptado la nueva manera de realizar su trabajo. La transición debe ser realizada teniendo cuidado de no perturbar demasiado el desarrollo de las actividades cotidianas de trabajo de los usuarios.

Esta etapa es también caracterizada por encontrar en los usuarios resistencia al cambio ya que implica tener que aprender una nueva manera de hacer las cosas, sin embargo esto no debiera de ser visto como una reacción negativa al nuevo sistema sino como actitud transitoria. En el caso particular de los Ingenieros de Preventa de Oracle de México la resistencia al cambio fue nula ya que vieron en la nueva aplicación la oportunidad de realizar su trabajo de manera más eficiente, además de contar con la ventaja adicional de no tener que acudir necesariamente a la oficina para llevar a cabo el registro y consulta de sus actividades .

### *Capacitación a los usuarios*

Durante la etapa de transición es muy importante brindar capacitación al personal que interactuará con la aplicación ya que es necesario que conozcan el uso de las funciones disponibles, evitando así un pobre entendimiento de la aplicación. Una buena o mala capacitación recibida dependerá en gran medida del éxito o fracaso del sistema en su totalidad.

El personal del departamento de Preventa de Oracle de México es el responsable de mantener la información del sistema, su capacitación debe entonces asegurar que puedan manejar las opciones disponibles de la aplicación, conociendo a gran detalle cada una de las partes que serán de su interés para cumplir con sus actividades cotidianas.

La capacitación a el personal de Preventa fué sumamente fácil de impartir debido a que estaba muy familiarizado con el sistema que estaba en producción anteriormente y a que conocían el concepto de la aplicación en su generalidad. Se incluyeron métodos a seguir en la captura de datos o en la preparación de datos necesarios para apoyo a las actividades de toma de decisiones, modificación de datos, métodos de consultas y borrado de registros de datos.

Todas las actividades de capacitación a los usuarios se llevaron a cabo dentro de las instalaciones de ORACLE de México. Dentro de este curso se puso especial atención en las preguntas y errores que presentaban los usuarios con el fin de que consideraran este nuevo sistema como una ventaja que les iba a facilitar el trabajo y no como un problema más al que habían que enfrentarse.

### *Pruebas en Paralelo*

El método más seguro para sustituir un sistema anterior por uno nuevo es utilizar ambos en paralelo por un determinado tiempo. Con este enfoque, los usuarios siguen operando el sistema anterior de la forma acostumbrada y a su vez comienzan también a utilizar el nuevo sistema. Este método es el enfoque de conversión más seguro ya que garantiza que en caso de surgir problemas

con la nueva aplicación tales como, errores en el procesamiento de transacciones de datos, la organización pueda regresar al sistema anterior sin pérdida de tiempo, ingreso o servicio. El periodo de tiempo que cubren estas pruebas en paralelo están determinadas por el desempeño sin errores de la nueva aplicación y el pleno entendimiento y aceptación de la misma por parte de los usuarios.

Las desventajas de los sistemas paralelos pueden ser significativas. En primer lugar los costos del sistema se duplican, ya que existen dos sistemas trabajando al mismo tiempo. En algunos casos es necesario contratar personal temporal para operar ambos sistemas en paralelo. En segundo lugar, el hecho de que los usuarios sepan que es posible regresar a las formas antiguas puede ser una desventaja si existe una resistencia potencial al cambio o si los usuarios prefieren el sistema anterior. En general el método de sustituir sistemas en forma paralela ofrece el plan de implantación más seguro, pero los costos y riesgos de un juicio injusto no pueden ser pasados por alto.

La fase de implementación del sistema se realizó en forma paralela con el sistema anterior, trabajando con datos de producción. El objetivo de esta fase consistió en la simulación de todo un ciclo de las operaciones realizadas por los Ingenieros de Preventa con la intención de ejecutar todos los programas involucrados en este proceso para detectar errores en la aplicación, así también se cotejó información entre las dos aplicaciones. El periodo de esta etapa fue de quince días, durante los cuales se estuvo en contacto permanente con los usuarios con el propósito de asistirlos en las dudas que llegaran a tener en la operación del sistema así como en la resolución de los posibles errores, una vez concluida la fase de transición se retiró en forma definitiva el sistema anterior. Cabe señalar que la aceptación hacia la nueva aplicación por parte de los Ingenieros de Preventa fue totalmente natural debido a que en todo momento vieron al nuevo sistema como una oportunidad de realizar su trabajo de manera más eficiente.

## 3.7 ETAPA DE PRODUCCIÓN

### Monitoreo del Desempeño del Sistema

Después de haber concluido las etapas de anteriores como son: la reingeniería de base de datos, la reingeniería de la aplicación, la generación a Web, intergración del sistema, pruebas y transición, se decidió iniciar la etapa de producción. En un principio, se pensó en hacer una nueva transferencia de datos al sistema para evitar inconsistencias y que la información estuviese al día, sin embargo, como ya se realizaron las pruebas en paralelo del sistema anterior con el sistema nuevo en Web y fueron satisfactorias, se tomó la determinación de seguir éste paralelo hasta actualizar la información y desde el instante en que comenzó la operación del Sistema de Preventa en Web y dejó de funcionar el anterior, se comenzó a monitorear el desempeño del sistema en Web.

Con lo que respecta al monitoreo, Oracle cuenta con herramientas para monitoreo como son el Server Manager, el Enterprise Manager, que permite obtener la información de: los usuarios que están trabajando en un momento determinado; cuales programas se están ejecutando; el estado de varios objetos dentro de la base de datos como áreas temporales para que la base de datos ejecute los ordenamientos de datos (sorts), los segmentos de rollback que utiliza la base de datos para guardar una copia de la información cuando ésta esta siendo actualizada y así poder garantizar que se regresen los datos al estado original cuando en usuario decida no hacer efectivos los cambios que hizo en la base de datos, etc.

Para detectar cuales eran las tablas más utilizadas, se estuvo consultando dentro de la herramienta la opción que despliega la información correspondiente a las tablas, desplegando el nombre de las tablas , el identificador y el nombre del usuario, el número del proceso que corresponde al usuario y se pudo observar como era de esperarse, que las tablas de uso frecuente y más transaccionales fueron las de actividades de preventa y de proyectos, las tablas menos utilizadas fueron los catálogos.

De igual manera, se realizó el monitoreo de los procesos que se estaban utilizando en un determinado momento, de esta forma, se puede determinar que usuario está ejecutando algún proceso en especial y si existe algún problema o bloqueo, se puede detener la ejecución de éste proceso. Esta misma actividad puede llevarse a cabo con un bloqueo de una tabla, si se da algún tipo de problema con una tabla, es necesario determinar que usuario está accediendo a la tabla, la manera de saberlo es con el monitor de uso de tablas que se mencionó anteriormente, con el nombre del usuario y el nombre de la tabla se detecta el lugar de la problemática.

Existe otra opción muy interesante dentro de la herramienta de monitoreo de la base de datos que permite obtener estadísticas de usuarios, rollbacks, cursores, lecturas físicas, etc. Se puede monitorear el comportamiento de un usuario, observando para algún proceso determinado el número de cursores abiertos a través de un acumulativo o el número de commits que ha efectuado en un momento dado, el número de rollbacks que ha efectuado dentro del proceso, etc., por lo tanto con esta herramienta se puede detectar si un proceso que está en ejecución abre demasiados cursores.

Dentro de las estadísticas, es posible detectar el tiempo muerto para un proceso determinado, el número de lecturas que hacen a los bloques de la base de datos que se alojan en el disco duro, el número de escrituras que en un momento dado se hacen al disco, etc. dentro de la información que nos despliega para cada rubro, nos da un promedio, el valor máximo y mínimo en ese instante, el valor que actualmente se encuentra y un valor total, esto permitió detectar el comportamiento de cada proceso y en aquellos que se estaban haciendo muchas lecturas físicas, se optimizó el programa.

Se monitoreó el comportamiento de los segmentos de rollback, en esta pantalla se despliega el nombre del segmento, su tamaño el número de extents, etc. Con la ayuda de la información desplegada en esta pantalla pudimos detectar que era necesario aumentar el tamaño de un segmento de rollback creando uno adicional, de esta manera se mejoró el aspecto de la contención.

Existen otras opciones dentro de la herramienta que también ayudaron a observar el comportamiento de la base de datos, tales como el monitoreo de las entradas y salidas (I/O), el monitoreo de usuarios, el monitoreo de candados, etc. con base a la información desplegada, se tomaron algunas acciones que ayudaron a que el comportamiento del sistema mejorara.

### **Afinación de la Base de Datos**

Dentro de este punto, existen dos conceptos importantes que son la afinación de la aplicación y la afinación de la base de datos. Para la afinación de la aplicación, básicamente se optimizaron las consultas (queries) que se estaban realizando y en algunos procesos se modificó la lógica del programa, con respecto a la optimización de la base de datos, se modificaron algunos parámetros de Oracle que se encuentran en un archivo en el disco duro (init.ora), que Oracle utiliza para levantar la base de datos de acuerdo a lo establecido en dicho archivo, para que la aplicación operase de una manera óptima.

Adicionalmente, existe una herramienta que también es proporcionada por Oracle llamada `explain plan` (plan de ejecución) que permite identificar la forma en que una determinada consulta está siendo ejecutada por Oracle.

Durante las etapas de transición y producción, se pudo observar que existían varias consultas dentro de las formas y reportes que estaban ejecutándose en forma lenta y se optó por utilizar el `explain plan` para identificar el problema y a su vez solucionarlo. Básicamente, alguna de estas consultas utilizaban la tabla de actividades de preventas, se introdujo una consulta en el `explain plan` y se observó que se estaba haciendo un full scan (búsqueda completa) en la tabla de actividades de preventas, esto indicó que no se estaba utilizando el índice de esa tabla, también se consultó en el Repositorio de Designer/2000 los índices que debería tener esa tabla y se detectó que algunos de estos no estaban en la base de datos, por lo que se crearon las estructuras de éstos y se construyeron en la base de datos.

En otras consultas, lo que se hizo fue modificar el orden de la declaración de las tablas en la cláusula `from` (de) de la consulta para hacerla eficiente, así mismo, se modificaron aquellas



consultas que estaban utilizando la instrucción `in (en)` en las subconsultas, esto debido a que eran lentas y se sustituyó `in` por `exists (existe)` y para que las consultas fueran más entendibles, se adicionaron alias a las columnas para identificar la tabla o vista que contenía la columna.

La gran labor dentro de la optimización de una aplicación consiste en la revisión y en su caso, la modificación de las consultas declaradas en toda la aplicación, la razón de hacer esto es que es muy importante la forma en que se encuentra declarada una consulta pues la manera en la que obtiene la información depende de esto, en algunos casos será necesario el uso de un índice y en otras será totalmente necesario inhibir el índice, todo depende del comportamiento de la consulta, otro punto que merece recalcar y que no es tan frecuente, es la revisión de la lógica de la programación, que en un momento determinado puede provocar que el tiempo del proceso sea más alto. En el caso de esta sistema, se determinaron algunos estándares de programación, esto ayudó de una manera importante.

### **Mantenimiento**

Con el sistema en producción, el usuario pudo experimentar la forma en que operaba el sistema y aún cuando en las etapas anteriores a esta se verificó que la funcionalidad del sistema fuese la adecuada, se encontró con algunos módulos del sistema que debían ser modificados, en esencia, el concepto de éste era el correcto, pero se adecuaron algunas pantallas y reportes a las necesidades del usuario

La forma en la que se le dió el seguimiento a estas modificaciones fue a través de una bitácora en la cual el usuario anotaba el nombre del módulo, fecha, nombre del usuario y comentario, indicando explícitamente el funcionamiento actual del módulo.

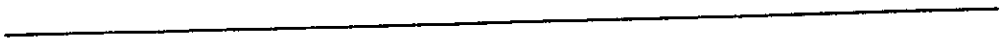
Con esta bitácora, se analizaba el problema y se asignaba la modificación a un programador, asignando una fecha estimada de entrega y cada vez que una modificación era terminada, se verificaba con el usuario. Ya que la validación del usuario terminaba correctamente, se actualizaba esta bitácora colocando el estado actual de la modificación.

Además, como parte de un requerimiento final del usuario en esta etapa fue poner el sistema corriendo en NCs (Computadora de Red) por lo que se abocó a la tarea de instalar NCs de diferentes marcas dentro de la empresa, lo cual implicó instalar varios servidores de booteo para las diferentes marcas de NCs, varios softwares denominados NC Server y la configuración de las NCs en la red. Se hicieron pruebas de funcionamiento y después se corrió el sistema sin ningún problema ya que las NCs traen dentro del NC Server el Navegador Netscape, para el caso de las Funai con el cual se pudo ver el sistema sin problemas, para el caso de las NCs de marca Acorn este tipo de máquina trae consigo un navegador propietario que de la misma manera no se tuvo ningún problema aunque se sabe que este tipo de NCs no soporta Applets de Java, sin embargo, el sistema esta en HTML y JavaScript que esto esta totalmente soportado en este tipo de máquinas. La otra parte del requerimiento es que el usuario quería tener algunas consultas no planeadas en el sistema para lo cual se utilizó una herramienta de usuario final llamada Discoverer, las cuales fueron puestas en el sistema a través de una liga.

---

## **Capítulo 4**

### **Conclusiones**



## CONCLUSIONES

Con este mantenimiento realizado al sistema se dio un giro en la interfaz que era tradicional de tipo carácter a una interfaz gráfica basada en acceso a través del Web, el cual cumplió con la característica de ser amigable a los usuarios, debido a que está en un ambiente totalmente gráfico y como se sabe este tipo de ambientes son muy sencillos de operar, y solo es necesario saber navegar en el Web para utilizar esta aplicación, aunque cabe mencionar, que los usuarios de dicho sistema no son típicos usuarios finales puesto que todos ellos tienen la carrera terminada en Computación, Informática o carreras afines. En algunos cuantos años más será muy común que la gran mayoría de los sistemas de Oracle de México se encuentren en Web, por lo que con esta migración, el Sistema de Ingeniería de Sistemas se puso a la vanguardia y listo para colocarse en el intranet de la empresa. Con esta labor que fue migrar el sistema existente, se actualizó la aplicación a la tecnología de vanguardia, además de poder explotar todos los beneficios que tiene un sistema en Web, como por ejemplo, los usuarios móviles no tienen que trasladarse a la oficina para poder capturar sus actividades del día, solo necesitan una línea telefónica para conectar su computadora portátil.

La funcionalidad del sistema quedó hecha a la medida de las necesidades actuales de los usuarios y gerentes del área, por lo que su implementación y puesta en operación no significó un gran esfuerzo, ni dolores de cabeza.

La administración y mantenimiento del sistema se facilita debido a que la aplicación se concentra en un equipo que es el servidor de aplicaciones, que es la segunda capa en el modelo de tres capas que ya se explicó en esta tesis y en el caso de haber modificaciones al sistema solo es necesario cambiar la aplicación una vez y esto será transparente a los usuarios, a diferencia de las aplicaciones cliente-servidor de dos capas que cuando se hace alguna modificación es necesario modificar la aplicación en todos los clientes, puesto que se necesita un cliente robusto en el cual reside la aplicación. El ahorro en el tiempo por actualizaciones de la aplicación depende del número de usuarios de la aplicación y entre más usuarios tenga el sistema el ahorro en tiempo es mayor.

Un punto muy importante a considerar es que con la migración de este sistema, Oracle tiene un sistema con la tecnología que esta promoviendo -que es la arquitectura cliente/servidor de tres capas a la que Oracle llama **Internet Computing** (Cómputo en Internet) en donde toma lo mejor de tres mundos: Ambiente cliente/servidor con su experiencia en aplicaciones transaccionales; Internet con su capacidad de obtener información desde cualquier parte del mundo, acceso casi inmediato y su fácil manejo vía los navegadores de Web y Objetos Distribuidos, en el cual se pueden hacer objetos o programas que pueden ser llamados a través de la red para dar como resultado una aplicación integral sin la necesidad de recodificar cierta funcionalidad, sino da la oportunidad de la reutilización de código- entre sus clientes y puede utilizarlo como referencia en algunas cuentas.

El hecho de haber utilizado en esas fechas las últimas versiones de los productos para realizar la migración del sistema facilita el mantenimiento de la aplicación debido a las bondades de las nuevas versiones que ahora permiten realizar las mismas tareas con más facilidad que sus versiones anteriores , además que el mantenimiento al sistema es también más rápido.

El soporte a la aplicación es muy importante y gracias a la migración el sistema se encuentra en versiones soportadas y no se encuentra más en la obsolescencia. Debido a que está realizado en las últimas versiones de las herramientas fue posible realizar la migración de la base de datos al último release del manejador de base de datos, lo cual antes no era posible por problemas de compatibilidad entre versiones, en pocas palabras después de estar estancado por mucho tiempo en cuanto a versiones de software se refiere, ahora es factible seguir realizando actualizaciones tanto del manejador de base de datos como de las herramientas a un costo muy bajo, pues el proceso de la actualización a nuevas versiones solo requerirá la regeneración de la aplicación, sin ninguna modificación adicional.

En cuanto a los integrantes del equipo todos ellos trabajan para Oracle México, lo que facilitó la tarea de llevar a cabo la investigación necesaria para realizar esta tarea, ya que todos tienen acceso a las fuentes primarias donde se obtuvo la información necesaria para la elaboración de esta tesis.

El trabajo en equipo enriquece la visión de cualquier proyecto, porque permite ampliarlo desde diversos ángulos tanto técnicos como administrativos y humanos, pues cada elemento aportó al proyecto desde su muy particular quehacer en Oracle lo que se consideró importante, por lo que se formó una visión integral alrededor de esta tesis, los enfoques que se tuvieron fueron :

- Enfoque de Soporte Técnico en cuanto a utilizar las últimas versiones de los productos, esta área sabe cuales son los problemas de desarrollo que tiene alguna versión o versiones de los productos de Oracle y también aprovechar al máximo las nuevas características ofrecidas por estas nuevas versiones.
- Enfoque Gerencial del área de Ingeniería de Sistemas, fue muy importante la interfaz amigable del sistema y la administración de recursos humanos y materiales necesarios para hechar a andar el sistema, así como también tener la visión del sistema en un futuro cercano cumpliendo con las políticas de la compañía..
- Enfoque de Ingeniería de Sistemas en cuanto a cuidar el más mínimo detalle técnico para que el sistema cumpliera con los objetivos planteados utilizando los productos Oracle de la manera más óptima.

La ingeniería en reversa se realizó con Designer/2000, el detalle de este proceso se revisó a lo largo de esta tesis y se generaron las formas y reportes de la aplicación hacia Web. Solo algunos reportes se hicieron en Discoverer pues el usuario necesitaba verlos en una herramienta de usuario final y son llamados a partir de una liga desde la página de web de la aplicación. En realidad, si se recomienda hacer la ingeniería en reversa, cuando ya se tiene un sistema hecho con las herramientas de desarrollo de Oracle (Developer/2000 & Designer/2000) pues facilita el desarrollo puesto que permite cambiar algunas definiciones aprovechando el trabajo ya realizado, es decir, la ventaja es que no se parte desde cero, así mismo, queda documentado y listo para cualquier mantenimiento que el equipo de trabajo u otras personas quieran o deban hacerle.

Otra alternativa era utilizar solo la herramienta de desarrollo Developer/2000 que no es una herramienta CASE -es la herramienta 4GL de Oracle que permite hacer formas, reportes y gráficas- en lugar de Designer/2000, sin embargo, no se habría alcanzado la documentación implícita de Designer/2000, que es el modelo de Entidad-Relación, los módulos de la aplicación entre otras cosas, y cabe recordar que la documentación es un parte fundamental de todos los sistemas, ya que debido a esto se pueden abatir costos de mantenimiento. Como experiencia propia del equipo de trabajo, el hecho de utilizar Designer/2000 facilitó el llevar a cabo el mantenimiento necesario del sistema y cumplió con todas las expectativas que se tenían de la herramienta.

La seguridad del sistema es controlada por el manejador de la base de datos, ya que las actualizaciones e inserciones se realizan a través de paquetes almacenados en la base de datos y el password del usuario con que se realizan las conexiones a la base de datos es desconocido por el usuario de la aplicación. Además de que también es desconocido para el usuario el lugar en donde residen los datos. Aunado a todo esto la aplicación se colocará en un futuro cercano en el intranet de la compañía, situación que está fuera del alcance de esta tesis.

Con todos los puntos presentados anteriormente se hacen evidentes las ventajas que tiene el haber migrado el sistema a un entorno de Web y se hace patente que se cumplió con el objetivo de la tesis "Migrar un sistema obsoleto a un ambiente de vanguardia, conservando y ampliando la funcionalidad que la aplicación tenía en modo caracter"

---

## **Bibliografía**

---



---

## BIBLIOGRAFÍA

Burch, John G. y Grudnitski, Gary. Diseño de Sistemas de Información. Ed: Grupo Noriega Editores. Primera Edición. 1992

Cheng, Betty H. c. y Gannod, Gerald C. Constructing Formal Specifications from Program Code. Ed: in Proc. of the IEEE Third International Conference on Tools in Artificial Intelligence. USA. 1991

Cruz Marín, Noe y Hernández Cortés, Miguel Angel. Sistema de Información de la Unidad de Servicios de Cómputo Académico. Tesis de Licenciatura de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México. Mex. 1996

DeBaud, J-M.; Moopen, B. y Rugaber, S. Domain Analysis and Reverse Engineering. Ed: Proc. Intl. Conf. Software Maintenance. 1994

Fairley, Richard. Ingeniería de Software. Ed: McGraw. Primera Edición. 1988

Gannod, Gerarld C. y Cheng, Betty H. C. A Two-Phase Approach to Reverse Engineering using Formal Methods. Ed: in Lecture Notes in Computer Science; Springer -Verlag, Proc. of Formal Methods in Programming and their Applications Conference. USA. 1993

Guengerich, Steve L y Smith, Patrick N. Client/Server Computing. Ed: SAMS Publishing. USA. 1994

Jarzabek, S. Domain Model-Driven Software Reengineering and Maintenance. Ed: Journal of Systems Software. USA. 1993.

---

Jarzabek, S y Tan, P.K. Design of a Generic Reverse Engineering Assistant Tool Proc. 2<sup>nd</sup> Working Conference on Reverse Engineering, WCRE. Ed: IEEE Computer Society Press. Los Alamitos, USA. 1995

Kendall, Julie E. y Kendall Kenneth E. Análisis y Diseño de Sistemas. Ed: Prentice Hall. Primera Edición. 1991

Kouloumdjian, Jacques; Petit, Jean-Marc y Toumani, Farouk . Relational Database Reverse Engineering: A Method Based on Query Analysis. Ed: International Journal of Cooperative Information Systems. USA.

Madron, Thomas W. Redes de Área Local. Ed: Limusa Grupo Noriega Editores. 1992

Microsoft Corporation. Microsoft Windows for Workgroups Resource Kit. USA. 1992

Müller, H. A. Verifying Software Quality Criteria Using an Interactive Graph Ed: Proc. 8th Annual Pacific Northwest Software Quality Conf. Portland, USA. 1990.

Novell Education. Networking Technologies.USA. 1993

Oracle Corporation. Oracle Designer/2000 On Line Documentation. USA. 1997

Oracle Corporation. Oracle Designer/2000Release 1:A Guide to Developer/2000 Generation. USA. 1997

Oracle Corporation. Oracle Designer/2000 Release 1.3 Product Overview. USA. 1997

Oracle Corporation. Executive Overview of Oracle Web Application Server. USA. 1997

Pressman. Roger S. Ingeniería de software. Ed: McGraw-Hill. Segunda Edición. 1988

Shaffer, Steven L. y Simon. Alan R. Network Security. Ed: AP Professional. USA. 1994

Sommerville, Ian. Ingeniería de Software. Ed: Iberoamericana. Segunda Edición. 1988

Tanenbaum. Andrew S. Redes de Ordenadores. Ed: Prentice Hall Hispanoamérica, S.A. USA. 1991

Tapscott, Don. The Digital Economy. Ed: McGraw-Hill. USA. 1995

Umar, Amjad. Client/Server Internet Environment: Object Oriented. Ed: Prentice Hall PTR. NJ USA. 1997

Yourdon, Edward. Análisis Estructurado Moderno. Ed: Prentice Hall. Primera Edición. 1993

[http://shimano.me.utexas.edu/papers/reverse/reverse\\_1996/dtml523/rev2.html](http://shimano.me.utexas.edu/papers/reverse/reverse_1996/dtml523/rev2.html)  
[irem@bertoni.me.utexas.edu](mailto:irem@bertoni.me.utexas.edu)