

2 ej



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

APLICACION MAESTRA DE INFORMACION GENERAL DE SOPORTE TECNICO

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N :

JOSE FRANCISCO ANDRADE BRAVO

ANTONIO CASTILLO GARCIA

JOEL GABRIEL FLORES HERNANDEZ



DIRECTOR DE TESIS: ING. ADOLFO MILLAN NAJERA

MEXICO, D. F.

1999.

TESIS CON
FALLA DE ORIGEN



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.



Este documento se proporciona sólo para fines informativos y los contenidos del mismo están sujetos a cambio sin previo aviso. Oracle corporation no garantiza que este documento este libre de fallas, ni brinda ninguna garantía o condición, ya sea expresa oralmente o por ley, incluyendo garantías y condiciones comerciales o de estado físico para ningún propósito en particular. Oracle corporation se libera de cualquier responsabilidad con respecto a este documento y declara que no adquiere ninguna obligación contractual directa o indirecta con este documento.

DEDICATORIAS

A mi madre

Principalmente y de manera muy especial quiero dedicar esta tesis a mi madre, por todos esos años de sacrificio, por todo el amor, comprensión y apoyo incondicional, aún en las situaciones más difíciles. Todas las palabras que yo pudiera plasmar en estas hojas no serían suficientes para decirte ¡GRACIAS MADRE!

A Dios

Que fue mi guía espiritual en todo momento.

A mi padre

Aunque no te tuve físicamente cerca de mí como yo hubiera querido, quiero darte las gracias por tu apoyo y ejemplo, que me han servido para concretar esto, que ahora es una realidad.

A mi querida esposa

Por creer en mí; tu comprensión y amor me han dado las fuerzas necesarias para seguir adelante.

A mis hijas

A ese par de criaturas que han cambiado mi vida, han sido el aliciente que me ha impulsado a continuar y terminar este proyecto.

A mis suegros

Lic. Ezequiel Zúñiga Galeana y Profra. Anita Berdeja De Zúñiga
Por su inmenso apoyo; les estaré eternamente agradecido.

A mis amigos

M.V.Z. R. Miguel Rosas, José Luis Rubí, Ing. Ricardo Ruíz,
Alfie Tapia, Joel Flores y José Fco. Andrade.
Gracias por sus palabras de aliento, y por brindarme el honor
de compartir su amistad y convivencia.

ANTONIO

Quiero agradecer la culminación de mi carrera a la incansable labor de mis padres que siempre estuvieron presentes bajo cualquier circunstancia, apoyándome en todo lo necesario.

A mi hermano que me recuerda cada día la esperanza y los sueños de la adolescencia y la fuerza de luchar por ellos.

A esa bellísima persona que estuvo presente durante todo el trayecto de mi carrera y que me permitió dislumbrar las riquezas del esfuerzo y dedicación y que formo parte especial en mi vida(S.R.O.)

A la unidad de cómputo académico de la Facultad de Ingeniería por haber contribuido a mi desarrollo como profesional.

A mis amigos Alfie Tapia, Francisco Andrade y Antonio Castillo por compartir conmigo 5 de los años más especiales vividos hasta el momento.

A mis familiares y compañeros de la Facultad que compartieron una gran cantidad de experiencias durante todo el trayecto.

A dios que me ha dado fuerza esperanza y mucha fe para enfrentar cada obstáculo

JOEL G.

Dedico esta obra a las personas más importantes que existen
en mi vida:

A Dios porque me dio la vida, la libertad y la fuerza para
continuar por el camino del éxito.

A mis padres por todo el cariño y apoyo que me han brindado;
porque confiaron en mi, por los principios que me han
inculcado y las bases que me dieron para que me supere día a
día, por todos los desvelos y preocupaciones que les hice pasar
y porque gracias a ellos estoy aquí.

Quiero agradecer a mis hermanos y mi cuñada, por todos los
consejos y experiencias que me brindaron para salir adelante a
lo largo de estos años, los cuales me sirvieron para no darme
por vencido y seguir caminando a la meta final.

Con cariño a Yesica, Berenice, Angel y Luis para que les sirva
de ejemplo y motivación porque algún día quiero que ellos
experimenten esta felicidad que ahora comparto con ellos.

De una forma muy especial a mi esposa, el viento que impulsa
mis alas, por todos esos momentos de alegrías y tristezas que
hemos pasado juntos, porque gracias a ellos hemos crecido
más como pareja y por todos sus ideales, sentimientos y
esfuerzos que ha compartido conmigo y me han ayudado a
concluir esta etapa en mi vida.

A mis amigos Alfie, Antonio y Joel, por todos los fabulosos
momentos de estrés que pasamos juntos a lo largo de la
carrera, y por la gran amistad que nos mantiene unidos a pesar
de la distancia.

A todos aquellos que integran la unidad de cómputo
académico por la gran oportunidad que me brindaron al ser
uno de ellos.

JOSE FRANCISCO

A la Universidad Nacional Autónoma De México, y en especial a la Facultad de Ingeniería

Por haber sido nuestra casa durante 5 años y por su valiosa aportación en el desarrollo profesional de nosotros los Universitarios.

A Nuestro Asesor

Ing. Adolfo Millán Najera

Quien con su dedicación, paciencia y estímulo logró hacer posible el presente trabajo.

A Todos Nuestros Profesores

Que han compartido sus conocimientos y parte de su vida en nuestra formación profesional.

NUESTRO MAS SINCERO AGRADECIMIENTO

INTRODUCCION

INTRODUCCION

Actualmente los avances tecnológicos en el campo de la computación se desarrollan muy rápidamente, ofreciendo nuevos conceptos, terminologías, equipo, etc., que proporcionan herramientas que facilitan el desarrollo principalmente en el área científica e industrial.

Algunas herramientas nuevas que han tenido mucho éxito y se han desarrollado exitosamente son las llamadas bases de datos, las cuales permiten definir sistemas de cómputo que utilizan una gran cantidad de información, proporcionando métodos eficientes para la manipulación de datos. Estas herramientas han sido esenciales para la creación de otros sistemas que tengan una aplicación más particular.

El objetivo de este trabajo es presentar el diseño de un sistema integral de información basado en herramientas de cuarta generación como manejadores de bases de datos relacionales, utilizando plataformas o arquitecturas en un ambiente Windows permitiendo de esta manera un mejor control de las llamadas que ingresan al área de soporte técnico.

Este trabajo engloba la utilización de herramientas y metodologías para el desarrollo e implementación de sistemas. En base a la metodología CASE y la tradicional, se desarrollará un sistema prototipo con alta adaptabilidad y funcionalidad consiguiendo resultados satisfactorios, utilizando para ello metodologías de desarrollo de sistemas, básicamente el método CASE.

Un sistema de información lo entenderemos como un conjunto de elementos (software, hardware, personas, datos, etc) relacionados entre sí, y que mediante interfaces adecuadas nos sirven para procesar, administrar y generar información.

Las razones que pueden conducir a una empresa a la realización de un sistema de información son muy diversas. Existen diversos tipos de sistemas de información enfocados cada uno de ellos a cubrir necesidades específicas de la empresa. Algunos de estos tipos de sistemas de información son :

- *Sistemas de procesamiento de datos.* Procesan grandes volúmenes de información de las funciones administrativas de rutina.
- *Sistemas informáticos para la administración (MIS).* Proporcionan informes periódicos para la planeación, el control y la toma de decisiones.
- *Sistemas de apoyo para la toma de decisiones (DSS).* Ayudan a quien toma las decisiones, cuando le proporcionan la información que solicita.
- *Sistemas expertos.* Asimilan la experiencia de quienes toman las decisiones en la solución de problemas.

Estos se analizan y diseñan mediante la aplicación de los conceptos y las técnicas del diseño y del análisis de sistemas.

El análisis y diseño de sistemas, pretenden estudiar sistemáticamente la operación de ingreso de los datos, el flujo de los mismos y la salida de la información; todo ello dentro del contexto de una empresa en particular.

En general, el análisis y el diseño de sistemas sirve para analizar, diseñar y fomentar o proponer mejoras en la operación de la empresa, es decir, un sistema de información se crea como solución a una necesidad específica o general, de acuerdo a ciertos requerimientos y en base a una metodología de diseño, lo cual es realizado en la actualidad mediante el uso de sistemas de información computarizados; esto es, la construcción del sistema es realizado de principio a fin con el apoyo de la tecnología informática y mediante el uso de herramientas computacionales que nos permiten optimizar cada uno de los pasos del diseño y desarrollo del sistema de información.

El desarrollo de sistemas sigue un ciclo conocido como ciclo de vida del sistema de información (CVSI), y no es otra cosa que el enfoque sistemático y por etapas del análisis y diseño del sistema de información. Existen diversas opiniones en cuanto al número exacto de etapas que conforman el ciclo de desarrollo de los sistemas, sin embargo se reconoce la importancia de su enfoque sistemático, es decir, nunca se lleva a cabo un elemento independientemente aunque cada etapa se presente de manera discreta.

Existen diversos métodos que se pueden seguir para el desarrollo del sistema. Dentro de estos métodos a seguir encontramos también diversas técnicas que nos servirán para implantar las fases del ciclo de vida del desarrollo del sistema.

El método tradicional contempla 5 pasos principalmente:

1. Análisis
2. Diseño
3. Desarrollo
4. Instalación
5. Mantenimiento (correctivo y preventivo)

Otros métodos contemplan hasta 7 pasos:

1. Identificación del problema y objetivos
2. Determinación de los requerimientos de información
3. Análisis de las necesidades de información

4. Diseño del sistema recomendado
5. Desarrollo y documentación del software
6. Pruebas y mantenimiento del sistema
7. Implantación y evaluación del sistema

Este compendio de información forma básicamente de cinco puntos:

- Antecedentes o estudio preliminar
- Marco Teórico
- Análisis y diseño del sistema
- Desarrollo del sistema
- Conclusiones

Las etapas antes mencionadas involucran en sí más pasos como se muestra en el índice temático, y no implica que cada uno de ellos se realice de manera independiente, ya que habrá actividades que se estén realizando y que luego irán finalizando poco a poco al tiempo que otras estarán en proceso.

Considerando los métodos mencionados, hemos decidido unirlos y detallar un poco más cada etapa siguiendo el ciclo de vida CASE (ingeniería de Sistemas Asistida por Computadora), otra de las razones por la cual adoptamos este método es porque al detallar cada etapa se tendrá un mayor control sobre cada una de ellas y de esta manera se podrán detectar errores que pudieran estar afectando el buen funcionamiento del sistema en general; nos auxiliaremos del software de Oracle Developer 2000 para la etapa de desarrollo, así como de Oracle7 Server para el manejo de la base de datos

En la primera etapa, llamada etapa de antecedentes o estudio preliminar se determina la forma de operación del área de negocios con el fin de identificar las problemáticas actuales, así como las probables deficiencias que pudieran llegar a existir y comenzar a bosquejar las probables alternativas de solución que podrían llegar a aplicarse.

Dentro de la sección de "Marco Teórico" se plasman los aspectos teóricos sobre los cuales se fundamenta el trabajo y los ideas fundamentales de la teoría utilizada para el desarrollo del trabajo que se presenta. Por una parte se determinan las características de la Metodología CASE. Además de la teoría del modelo entidad-relación, así como la especificación del tipo de nomenclatura utilizada para poder llevar a cabo el desarrollo de un sistema

Uno de los aspectos que también se llegan a tocar en este punto es la teoría que encierra la operación y funcionalidad del software que controlará el buen desempeño del sistema, en este caso nos estamos haciendo referencia al manejador de base de datos Oracle. Donde se explica los procesos que

involucran al trabajar con este manejador, las operaciones básicas que realiza cada proceso, así como la interacción entre ellos. Por otra parte se indica también la forma en que opera el software de comunicación entre los clientes, o las terminales que ejecutarán el sistema AMIGS, y el manejador Oracle.

En la etapa de análisis se investigan los requerimientos de los usuarios y se identifican los recursos con que se cuentan para realizar el sistema. La identificación de objetivos es un componente importante en esta primera etapa. En primera instancia se deberá descubrir lo que la empresa intenta realizar, ya sean modificación al sistema anterior o implantación de un nuevo sistema, en nuestro caso estaríamos hablando del primero.

En cuanto a la identificación de los objetivos tenemos lo siguiente: enfocándonos específicamente al tema de la presente tesis, el análisis y diseño está centrado en cubrir una necesidad específica de la empresa; hacer del sistema de soporte técnico con el que se cuenta actualmente un sistema más amigable y fácil de utilizar, esto se pretende realizar cambiando el antiguo sistema con ambiente 100% texto en un sistema con ambiente gráfico, que facilite la captura, la búsqueda y envío de resultados al usuario. Esto necesariamente tendrá que realizarse de manera óptima, es decir, que cada uno de los procesos; captura, búsqueda y resultados, sean realizados en un tiempo no mayor al que se lleva realizarlos en la actualidad, porque de lo contrario estaríamos cayendo en el caso de que el sistema se verá más agradable a la vista pero será más lento, y entonces la implantación del nuevo sistema ya no respondería a las necesidades de la empresa.

Tener en forma rápida y segura la información de las acciones que se realizan en el área de soporte técnico, llevando un registro de las interacciones que se tienen con un cliente con respecto de un problema en particular, así como tener una forma de dar seguimiento a sus clientes, registrando y actualizando los contratos de los mismos.

En el área de soporte técnico de alguna empresa, cuando se proporciona esta ayuda vía telefónica, la función principal es auxiliar a los clientes a resolver un problema que se les presenta en un momento dado, con alguno de los productos de esa empresa de una manera eficaz y rápida.

Existen diferentes tipos de situaciones por las cuales llama un cliente a soporte técnico. Con lo cual se analizaron niveles de severidad que pueden existir al recibir una llamada por parte del cliente para que de esta forma se le de la atención con la prontitud que esta requiere. Dicha clasificación la podemos resumir de la siguiente forma

I. Severidad 1

La falla provoca pérdida completa de los servicios sin tener en cuenta el ambiente del cliente (*por ejemplo: producción, pruebas, desarrollo, etc.*) y el trabajo no puede continuar. El Problema / Falla presenta alguna o más de las siguientes características:

- **Corrupción de Datos.** Los datos física o lógicamente no están disponibles o son incorrectos. Ejemplo: corrupción del formato del bloque, entradas invalidas de índices, corrupción de meta-datos, resultados incorrectos.
- **Sistema suspendido.** El proceso se suspende indefinidamente o existe una degradación severa del performance provocando que la respuesta de los recursos sea muy lenta, como si el sistema estuviera colgado.
- **El sistema se cae repetidamente.** Los procesos de la base de datos, o los procesos de background fallan y continúan fallando después de varios intentos al tratar de reiniciarlos.
- **Las funciones críticas no están disponibles.** La aplicación no puede continuar debido a que son características vitales no están disponibles, los datos no pueden ser asegurados, respaldados, etc.

II Severidad 2

La falla provoca un error interno (software), o un comportamiento incorrecto causando una pérdida severa de los servicios. No hay ningún workaround disponible para el cliente; sin embargo, las operaciones pueden continuar en un modo restringido. El Problema / Falla presenta alguna o varias de las siguientes características.

- **Error interno de software el cual causa que el sistema falle, pero si es posible reiniciar o recuperar.**
- **Degradación severa del performance debido al error de software**
- **Algunas funciones importantes no están disponibles, sin embargo el sistema puede seguir operando en un modo restringido.**

III Severidad 3

La falla provoca una pérdida mínima del servicio. El impacto del problema / falla es menor o una inconveniencia, tal como una desviación manual para

restaurar la funcionalidad del producto. El Problema/Falla presenta alguna o más de las siguientes características:

- Un error de software para el cual existe un *workaround* aceptable
- Degradación mínima del performance ($\leq 10\%$) debido al error de software.
- Error de software o comportamiento incorrecto con un impacto menor en la operación del sistema.
- El error de software requiere modificar la configuración manualmente o algún *script* para solucionar el problema.

IV. Severidad 4

La falla NO causa pérdida del servicio. El Problema / Falla es un error menor, comportamiento incorrecto o un error de documentación que de ninguna manera impide la operación del sistema.

V. Severidad 5

Aquella en la cual el cliente solicita la actualización de algún producto, y en la cual realmente no se le presenta ningún error de ninguna clase.

VI Severidad 6

El cliente llama para hacer una consulta no técnica, ya sea referente a su contrato o para cualquier otro tipo de información.

En base a lo anterior, podemos darnos cuenta que es muy necesario llevar un control de todos los problemas que entran a soporte técnico, porque en varias ocasiones se presentan problemas similares que pueden ser solucionados rápidamente consultando y aplicando las acciones que se llevaron a cabo en un problema anterior, pero para esto se debe contar con un sistema que permita llevar dicho control. Además, podemos constatar que si se tiene un sistema eficiente y confiable, la atención y el servicio al cliente se incrementará en forma notable, resultando en una mejor satisfacción para el cliente y un mayor beneficio para la empresa.

En la sección de análisis y diseño se especifica la propuesta de solución. Dentro de las cuales se agregó un par de objetivos más los cuales hacen referencia a una interfaz vía WEB que permita a los clientes poder interactuar con los problemas reportados al área de soporte técnico sin necesidad de realizar una

llamada telefónica, además, de consultar los diferentes problemas de todos los diversos clientes con el fin de poder tener alternativas de solución consultando directamente la base de datos del sistema AMIGS antes de consultar al área de soporte técnico.

Además se plantea graficamente, a través de los diagramas de flujo de datos, las operaciones que involucran el área de trabajo. De que forma se llevo a cabo la generación de prototipos y la importancia de la interacción del usuario final con los prototipos generados. Se presenta el diagrama entidad-relación el cual se concluyó en base a todo el flujo de información identificado en la etapa de generación de flujo de datos. Finalmente se describen las tablas lógicas que involucran a todo el sistema, incluyendo el nombre de las tablas, los campos asociados y el tipo de dato que cada campo manejará, además del diccionario de datos.

En el capítulo "Desarrollo del sistema" de primera instancia se habla sobre las herramientas de desarrollo que se utilizarán para realizar la generación del sistema, dichas herramientas como ya se comentará en su momento son Designer 2000 y Developer 2000.

Se plantea la justificación del desarrollo del sistema. Básicamente se replantean los objetivos con el fin de dejar claro el punto hacia el cual se debe de llegar en base a mejoras o nuevos servicios que pudieran llegar a darse como el servicio a domicilio, conocido como servicio on-site y que en esta sección se comenta a detalle la implementación de este servicio.

Por otra parte se determinan los requerimientos con los cuales debe contar cada máquina cliente para poder ejecutar el sistema AMIGS. Requerimientos como el tipo de procesador, la memoria RAM mínima, el espacio libre en disco, tipo de monitor a utilizar, etc.

Una de las secciones que está relacionada con la generación y prueba de prototipos es la parte de programación, en esta parte se especifica el tipo de programación utilizada, especificando de manera clara la que es generada por la herramienta CASE, en esta caso Designer/2000, y el tipo de programación creada por el programador.

Se indican las pruebas a las cuales fue sometido el sistema, aplicando básicamente dos tipos de pruebas, pruebas de la caja negra, y pruebas del sistema. Además de la explicación de cada una de ellas y lo que se pretende cubrir con cada una de las pruebas realizadas.

Se plantea los pasos necesarios para poder realizar la generación de ayuda desde Designer/2000. Con el fin de facilitar el uso y entendimiento del sistema en el momento en que el usuario final lo requiera.

Se explica los procedimientos necesarios para que el usuario sea capaz de acceder la información del sistema desde ambiente WEB, tal y como uno de los objetivos lo requería. Se presentan las pantallas que conforman esta interfaz y el tipo de información que puede llegar a consultar el usuario final.

Se habla de los dos métodos de implantación que existen para llevar a cabo la implantación del sistema. Se especifican las formas o ventanas y reportes con los cuales interactua el usuario final. El conjunto y tipo de archivos que conforman la aplicación, como por ejemplo: librerías, iconos y archivos tipo ayuda.

Al final de la sección del "Desarrollo del sistema" se explica el tipo de mantenimiento que se puede llegar a realizar a la aplicación utilizando la misma herramienta de desarrollo como es Designer/2000, como es el manejo de versiones así como la posibilidad de poder agregar al repositorio de Designer/2000 la definición de algún módulo creado directamente con Developer 2000.

Finalmente se habla sobre las conclusiones de este trabajo, si fueron cubiertas o no las expectativas inicialmente fijadas.

Los resultados que se pretenden obtener al desarrollar dicho sistema es tener un control eficiente y total de todas las llamadas referentes a las asesorías por parte del equipo del área de soporte técnico. Además de que los analistas o ingenieros encargados de prestar sus servicios tengan una manera sencilla de consultar qué clientes tienen contrato de soporte, y en base a ello determinar si se le da soporte a dicho cliente en ese momento o no.

INDICE

		Página
CAPÍTULO 1	ANTECEDENTES	1
1.1	Un sistema de Base de Datos	6
1.1.1	Datos de operación	8
1.1.2	Ventajas de tener un control centralizado	8
1.1.3	Independencia de los datos	9
1.1.4	Almacenamiento de datos	10
1.1.5	Arquitectura de un sistema de Base de Datos	11
1.1.6	Manejo de tablas	11
CAPÍTULO 2	MARCO TEORICO	13
2.1	Análisis y Diseño(metodología CASE)	14
2.1.1	Componentes de un sistema CASE	19
2.2	Diseño de la Base de Datos	21
2.3	Oracle RDBMS	30
2.3.1	El servidor Oracle	32
2.3.1.1	Estructura de la Base de Datos	32
2.3.1.2	Una instancia Oracle	33
2.3.1.3	Estructura de la Base de Datos y manejo del espacio	34
2.3.2	Estructuras de memoria y procesos	34
2.3.2.1	Estructuras de memoria	36
2.3.2.2	Procesos	37
2.4	Conexión vía SQL*NET	38
2.4.1	Configuración	41
CAPÍTULO 3	ANALISIS Y DISEÑO DEL SISTEMA	42
3.1	Definición del problema	43
3.2	Propuesta de solución	45
3.3	Diagrama de Flujo de Datos(DFD)	48
3.4	Prototipos	59
3.4.1	Desarrollo de un prototipo	60
3.4.2	Aplicación Maestra de Información General de Soporte técnico(AMIGS)	64
3.5	Diagrama Entidad-Relación	67
3.6	Diccionario de Datos	71

CAPITULO 4	DESARROLLO DEL SISTEMA	79
4.1	Herramientas de Desarrollo	80
4.1.1	Generalidades de Designer 2000	80
4.1.2	Oracle Developer/2000, características clave	86
4.2	Justificación del sistema	90
4.3	Requerimientos del sistema	92
4.4	Programación del sistema	94
4.5	Pruebas	105
4.5.1	Prueba de la caja blanca	105
4.5.2	Prueba de la caja negra	106
4.5.2.1	Partición equivalente	107
4.5.2.2	Prueba de tareas	107
4.5.2.3	Prueba de Integración o comportamiento integrado	107
4.5.3	Prueba del sistema	108
4.5.3.1	Pruebas de recuperación	108
4.5.3.2	Pruebas de seguridad	108
4.5.3.3	Pruebas de resistencia	109
4.5.3.4	Pruebas de rendimiento	109
4.5.4	Pruebas realizadas al sistema AMIGS	109
4.6	Implantación	114
4.6.1	Instalación local en el cliente	114
4.6.2	Instalación y configuración del sistema AMIGS	118
4.6.2.1	Configuración SQL*NET	120
4.6.2.2	Configuración del REGEDIT	123
4.6.2.3	Instalación sobre una red tipo LAN	124
4.7	Generación de ayuda desde DESIGNER/2000	127
4.7.1	Característica de un sistema de ayuda Microsoft Windows	127
4.7.1.1	Configuración del MS Help Generator	127
4.7.1.2	Procedimiento para la generación de ayuda de Microsoft Windows	129
4.8	Implementación en WEB	132
4.9	Mantenimiento	141
CONCLUSIONES		150
GLOSARIO		155
BIBLIOGRAFIA		159

ANTECEDENTES

1

ANTECEDENTES

Como ya lo habíamos mencionado anteriormente, la ciencia de la computación ha tenido un auge enorme, y el avance tecnológico que se tiene es constante, por tanto es muy importante la modernización tanto de software como de hardware, porque de otro modo se tendrán graves dificultades para enfrentar los requerimientos que se presentan y poder tener un ambiente adecuado para enfrentar los problemas que enfrentan los clientes cuando nosotros trabajamos en el área de soporte técnico; además de tener en cuenta que cada vez es más sencillo el ambiente y manejo para los clientes, de forma tal que se le faciliten las cosas, con respecto al soporte técnico de algún producto en particular, pero esto implica, por otro lado que al equipo técnico se le complican más sus actividades, debido a que cada vez son más opciones las que tiene al alcance de su mano y debe saber acerca de las características y funcionalidad de sus productos para poder auxiliar a sus clientes cuando éstos le llaman para consultarle.

Pero qué es lo que sucede cuando se queda estancado el área de soporte técnico con respecto a las herramientas que ellos utilizan para auxiliar a los clientes; esto da como resultado una *insatisfacción por parte de los clientes*, lo cual se traduce en pérdidas para la empresa que presta el soporte, porque cada vez menos clientes confían en la rapidez que tendrán si llaman al área de soporte, los cuales debido a las herramientas que manejan se ven imposibilitados para responder de una manera rápida a las solicitudes de los clientes, provocando de esta forma que la cartera de clientes disminuya de forma considerable, dando como resultado ya sea la disminución de la nómina, o en casos extremos, podría llevar incluso al cierre de pequeñas, y por que no decirlo, medianas empresas también, las cuales tengan como principal sostén a sus clientes, y sin los cuales no tienen razón de ser.

Actualmente tenemos una estrecha relación con el área de Soporte Técnico de Oracle de México, ya que nos encontramos como becarios en dicha área de servicios. Con lo cual nos percatamos de la forma de operación y del sistema con el que cuenta esta subsidiaria y las posibles mejoras que se le podrían realizar. Se plantearon los objetivos y nos facilitaron los medios necesarios para poder realizar la actualización de dicho sistema sin garantizarnos que en el futuro se llegarán a implementar las sugerencias aquí comentadas.

El primer punto que debemos analizar, es qué tipo de sistema se está utilizando, además de cómo está organizado; tanto la infraestructura de la empresa como el tipo de red que están manejando.

Cabe hacer notar que debido a las pérdidas que se han tenido últimamente, el director está de acuerdo en hacer una inversión suficiente para solventar los problemas que se tienen antes de que lleguen a ser catastróficos e

irreversibles.

Actualmente cuentan con el siguiente hardware, cabe hacer mención que este equipo ha sido de reciente adquisición, lo cual nos beneficia en gran manera debido a que prácticamente cumple con las características que solicitamos en el punto de requerimientos.

- 1 máquina Hewlet Packard
Procesador Pentium a 133 Mhz
32 MB de RAM
1GB de disco duro
Sistema Operativo Windows NT 3.51
- 15 máquinas Acer
Procesador 486 a 100 Mhz
8 MB de RAM
500 MB de disco duro
Sistema Operativo Windows 95

Cabe destacar que este equipo sólo es el que pertenece al área de soporte, ya que la empresa cuenta con otras áreas en donde todavía no se *moderniza* el equipo. Pero como podemos observar prácticamente con estos recursos podemos trabajar con el sistema sin mucho problema.

El tipo de red que están utilizando es una bus, con cable UTP. A continuación se muestra una gráfica en la cual podemos apreciar la distribución que tienen las máquinas en el área de soporte, por el momento no hacemos mención a las otras áreas en donde se cuenta igualmente con una red bus con cable UTP, pero en este caso el equipo con el que se cuenta es obsoleto, debido a que estamos hablando de computadoras con procesador 80386 y en algunos casos inclusive con máquinas 80286; las cuales son una minoría.

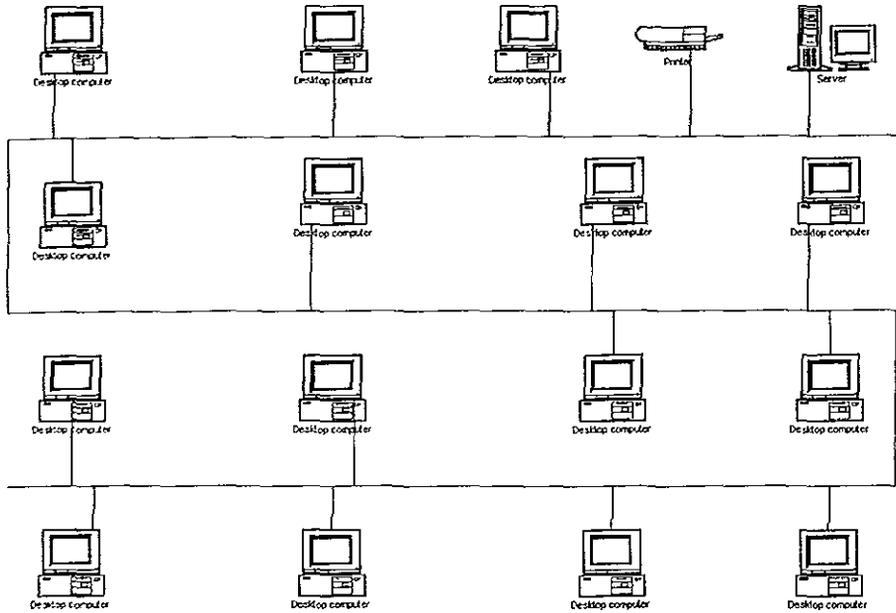


Figura 1.1 Red interna del área de soporte

Como podemos darnos cuenta, podemos explotar en gran medida los recursos con los que cuenta la empresa. Solo es cuestión de realizar un sistema que realmente cubra las necesidades que se están presentando hoy en día.

Si nos enfocamos al sistema anterior, el cual estaba realizado con Forms 3.0, y considerando los puntos más relevantes de dicha herramienta, podemos resaltar de ella los siguientes puntos:

- Nos permite definir transacciones que combinan datos de múltiples tablas en una simple forma.
- Crear de una manera rápida formas estandarizadas.
- Nos permite la definición personalizada de las formas creadas por default.
- Portable para una gran cantidad de plataformas.

Sin embargo, esta herramienta fue desarrollada para trabajar con las versiones 6.0 y 7.1 de la base de datos de Oracle, aunque también puede ser utilizada con versiones posteriores de la base de datos de Oracle, se debe tener un ambiente de trabajo cliente-servidor. Pero el seguir trabajando con esta herramienta presenta algunas desventajas, tales como la de no contemplar el

cambio de siglo, el no contar con la gama de triggers y demás posibilidades que, por otro lado, nos ofrecen las nuevas herramientas de Oracle; Designer 2000 y developer 2000. Las características de estas herramientas se detallan y tratan a fondo en el capítulo 4, en el apartado 4.1 Herramientas de desarrollo.

Forms 3.0 contempla únicamente 5 tipos de objetos. El objeto más externo, es decir, el contenedor de todos los demás es la forma. Esta forma puede ser ligada a otra forma, grupos de formas, menús, reportes y demás componentes. Los demás objetos manejados por Forms son:

Bloques. Describe una sección o sub-sección de una forma, y funciona como la base para realizar la interacción con la base de datos.

Fields (campos). Representan las columnas o áreas de inserción de datos, y en el se puede describir la manera en la que serán desplegados los datos.

Pages (páginas). Es el conjunto de capas en las cuales pueden ser desplegados los demás objetos, como constantes de textos y los campos o fields.

Triggers (disparadores). De manera similar que en Forms 4.5 es el conjunto de programas que en determinadas circunstancias se llegan a disparar.

Procedimientos a nivel forma. Conjunto de programas que pueden ser invocados desde algún trigger u otro procedimiento a nivel forma. En forma 4.5 este tipo de procedimientos pueden ser funciones, procedimientos o paquetes (conjunto de programas).

Actualmente la aplicación la tienen instalada sobre plataforma UNIX. Lo que se realiza para ejecutar la aplicación es lo siguiente: ejecutan un emulador de terminal desde las PC's con Windows 95, con este emulador cada analista se conecta al servidor UNIX y de esta forma ejecutan su aplicación en ambiente UNIX.

En la figura 1.2 se muestra una de las formas del actual sistema, la cual fue desarrollada con Forms 3.0. En la pantalla que se muestra de ejemplo se dan de alta los reportes o SAT's (Solicitud de Asistencia Técnica) de los clientes.

Contratos Educacional **SopORTE** Rec Humanos catálogos Utilerías salir

Q/E_ING_SOPORTE [REDACTED]
 ===== SATS =====

No SAT [REDACTED] Responsable [REDACTED] Fecha [REDACTED] NIC [REDACTED]

Nombre cliente [REDACTED]
 Nombre contacto [REDACTED]
 Dirección [REDACTED]

Teléfono [REDACTED] Plataforma [REDACTED]
 Fax [REDACTED] Media [REDACTED]
 Modem [REDACTED] Versión SO [REDACTED]

Prioridad [REDACTED] Status [REDACTED] Producto [REDACTED]
 Fecha actualización [REDACTED] RDBMS [REDACTED]
 Componente [REDACTED]

Error [REDACTED] Escalamiento [REDACTED]

Descripción del error [REDACTED] Fecha de escalamiento [REDACTED]

[REDACTED]

Figura 1.2 Forma de captura de una SAT

Existen algunos aspectos teóricos con relación a un sistema de base de datos que consideramos importantes, dichos aspectos se describen a continuación

1.1 UN SISTEMA DE BASES DE DATOS

Un sistema de bases de datos es un sistema de mantenimiento de registros basado en computadoras, es decir, un sistema cuyo propósito general es registrar y mantener información. Tal información puede estar relacionada con cualquier aspecto que sea significativo para la organización donde el sistema opera.

Una base de datos, se puede ver como un repositorio de datos almacenados y, en general, es tanto integrada como compartida.

Por integrada se entiende que la base de datos puede considerarse como una unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier redundancia entre los mismos.

Por compartida se entiende que partes individuales de la base de datos pueden compartirse entre varios usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la base de datos (y utilizaría con propósitos diferentes). Tal comportamiento es en verdad consecuencia del hecho de que la base de datos es integrada. En otras palabras, usuarios diferentes percibirán de modos muy distintos una base de datos específica. Aunque dos usuarios compartan el mismo conjunto de elementos en la base de datos, sus percepciones o vistas de ese conjunto pueden diferir mucho al nivel de detalle.

Un sistema de base de datos incluye cuatro componentes principales: datos, hardware, software y usuarios.

Datos. Se emplea la palabra datos para referirse a los valores registrados físicamente en la base de datos. Los datos almacenados en un sistema pueden dividirse en una o más bases de datos; sin embargo, es más conveniente suponer que sólo hay una base de datos, la cual contiene todos los datos almacenados en el sistema.

Hardware. El hardware se compone de los volúmenes de almacenamiento secundario como pueden ser discos; donde reside la base de datos, junto con dispositivos asociados como pueden ser las unidades de control, puesto que la base de datos es demasiado grande para caber en su totalidad en la memoria principal del sistema.

Software. Entre las bases de datos físicas en sí (es decir, el almacenamiento real de los datos) y los usuarios del sistema, existe un nivel de software, que a menudo recibe el nombre de Sistema de Administración de Base de Datos o DBMS. Este maneja todas las solicitudes de acceso a la base de datos realizadas por los usuarios. Una función general del DBMS, por tanto, es proteger a los usuarios de la base de datos contra los detalles al nivel de hardware. En otras palabras, el DBMS ofrece una vista de la base de datos que esta por encima del nivel de hardware y apoya las operaciones del usuario.

Usuarios Se consideran tres clases generales de usuarios. La primera la representa el programador de aplicaciones, encargado de escribir programas de aplicación que utilicen base de datos. Estos programas de aplicación operan con los datos de todas las maneras usuales: recuperan información, crean información nueva, suprimen o modifican información existente, etc. Todas estas operaciones se realizan formulando las operaciones adecuadas al DBMS.

Los programas en sí pueden ser aplicaciones convencionales de procesamiento por lotes o programas en línea diseñados para apoyar a un usuario final, que interactúa con el sistema desde una terminal remota.

La segunda clase de usuarios es entonces, el usuario final que accesa a la base de datos desde una terminal. Un usuario final puede emplear un lenguaje de consulta proporcionado como parte integral del sistema. Recurrir a un programa de aplicación escrito por un usuario programador que acepte órdenes desde la terminal y a su vez formule solicitudes al DBMS en nombre del usuario final. De cualquier manera, el usuario final puede realizar en general, todas las funciones de recuperación, creación, supresión y modificación, aunque tal vez podemos afirmar que la recuperación es la función más común de este usuario.

La tercera clase de usuarios la representa el administrador de la base de datos o DBA.

1.1.1 DATOS DE OPERACION

Los datos de una base de datos los podemos llamar "datos de operación", distinguiéndolos de los datos de entrada y de salida. Por ejemplo: Cualquier empresa necesita disponer de una gran cantidad de datos acerca de sus funciones:

<i>Empresa</i>	<i>Datos de operación</i>
Banco	Datos de cuenta
Hospital	Datos de pacientes
Universidad	Datos de estudiantes
Compañía manufacturera	Datos de planeación

Los datos de entrada se refieren a la información que entra al sistema desde el exterior; tal información puede ocasionar un cambio en los datos de operación, pero en sí misma no forma parte de base de datos. Del mismo modo, los datos de salida conciernen a mensajes e informes que proceden del sistema (ya sean impresos o desplegados en una terminal); de nuevo esos datos contienen información derivada de los datos de operación, pero no constituyen parte de la base de datos.

1.1.2 VENTAJAS DE TENER UN CONTROL CENTRALIZADO DE LOS DATOS

Se debe optar por almacenar los datos de operación en una base de datos integrada puesto que un sistema de base de datos proporciona un control centralizado de los datos de operación los cuales constituyen uno de los activos más validos.

Puede disminuirse la redundancia. En sistemas que no usan base de datos, cada aplicación tiene sus propios archivos privados, esto a menudo origina enorme

Puede disminuirse la redundancia. En sistemas que no usan base de datos, cada aplicación tiene sus propios archivos privados, esto a menudo origina enorme redundancia en los datos almacenados, así como desperdicio resultante del espacio de almacenamiento. En un sistema de base de datos, la redundancia debe controlarse; es decir, el sistema debe estar al tanto de la redundancia y asumir la responsabilidad de propagar las actualizaciones.

Puede evitarse la inconsistencia. Cuando una información se representa por dos entradas diferentes en la base de datos, y el sistema no está al tanto de esta duplicidad (en otras palabras, la redundancia no se ha controlado), habrá entonces algunas ocasiones en que las dos entradas no concuerden, esto es, cuando una y sólo una de ellas se haya actualizado entonces, en tales circunstancias se dice que la base de datos es inconsistente; lo que da como resultado que se obtenga información incorrecta o contradictoria.

Los datos pueden compartirse. Las necesidades de los datos de las nuevas aplicaciones pueden atenderse sin tener que crear nuevos archivos almacenados.

Puede conservarse la integridad. El problema de la integridad es garantizar que los datos de la base de datos sean exactos. La inconsistencia entre dos entradas que representan al mismo hecho es un ejemplo de falta de integridad, que por supuesto sólo ocurre si existe redundancia en los datos almacenados. Para contrarrestar este suceso, se pueden aplicar restricciones de seguridad:

1. Asegurarse que el único medio de acceder la base de datos sea a través de los canales establecidos
2. Definir controles de autorización para que se apliquen cada vez que se intente el acceso a datos sensibles.

1.1.3 INDEPENDENCIA DE LOS DATOS

La mayoría de las aplicaciones actuales son independientes de los datos, por lo que, la manera como los datos se organizan en almacenamiento secundario y a la manera como se accesan depende de los requerimientos de la aplicación y, además que el conocimiento de la organización de los datos y de la técnica de acceso forma parte de la lógica de la aplicación. Por ejemplo, puede decirse que un archivo particular se almacena en forma secuencial con índices. La aplicación, entonces, debe saber que el índice existe y conocer la secuencia del archivo (como la define aquél), de modo que la estructura interna de la aplicación se construirá con base en este conocimiento. Además, la forma precisa de los diversos procedimientos de acceso y de control de excepciones dentro de la aplicación dependerá, ante todo, de los detalles de la interfaz presentada por el

software secuencial indicado.

Se dice que una aplicación como ésta es dependiente de los datos porque es imposible cambiar la estructura de almacenamiento (la manera como se encuentra físicamente registrados los datos) o la estrategia de acceso (la manera como se accesan) sin afectar la aplicación, no sería posible reemplazar el archivo secuencial indicado anteriormente por un archivo direccionado por dispersión sin hacer modificaciones importantes de la aplicación.

La provisión de independencia de los datos es un objetivo esencial de los sistemas de bases de datos. Se puede definir la independencia de los datos como la inmunidad de las aplicaciones a los cambios de la estructura de almacenamiento y de la estrategia de acceso. Lo que implica que las aplicaciones no dependen de ninguna estructura de almacenamiento o estrategia de acceso en especial.

1.1.4 ALMACENAMIENTO DE DATOS

CAMPO ALMACENADO

Un campo almacenado es la unidad de datos con nombre más pequeña que se halla almacenado en la base de datos. Esta contendrá en general muchas ocurrencias de cada uno de los diversos tipos de campo almacenado; por ejemplo, una base de datos que contenga información sobre alumnos de una escuela, tal vez incluya un tipo de campo almacenado llamado "nombre o alumno" el cual tendría todos los nombres de los alumnos de dicha escuela y, habría una ocurrencia de ese campo para cada uno de los alumnos.

REGISTRO ALMACENADO

Un registro almacenado es un conjunto con nombres de campos almacenados asociados. Una ocurrencia de un registro almacenado se compone de un grupo de ocurrencias de campos almacenados relacionados (y representa una asociación entre ellos). Se dice que las bases de datos contienen múltiples ocurrencias del tipo de registro almacenado. En la mayoría de los sistemas, la ocurrencia de registro almacenado es la unidad de acceso a la base de datos, es decir, la unidad que el DBMS puede recuperar o almacenar en un acceso. Un archivo almacenado es el conjunto de todas las ocurrencias de un tipo de registro almacenado.

1.1.5 ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS.

El propósito de describir esta arquitectura es proporcionar un marco de referencia que sea útil para describir conceptos generales sobre bases de datos, pero no se pretende que cada sistema de bases de datos corresponda con fidelidad a este *marco de referencia* específico, no se quiere dar a entender que esta arquitectura particular ofrece el único marco de referencia posible. La arquitectura se divide en tres niveles generales: interno, conceptual y externo.

Nivel interno. Es el más cercano al almacenamiento físico, es decir el que concierne a la manera como los datos se almacenan en realidad.

Nivel externo. Es el nivel más cercano a los usuarios, es decir, el que atañe a la manera como cada usuario ve los datos.

Nivel conceptual. Este es un nivel de mediación entre los otros dos niveles.

1.1.6 MANEJO DE TABLAS

En una base de datos se tiene un conjunto de información organizada, de forma que ésta pueda ser recuperada rápidamente. Los campos de datos están delimitados en su contenido por el tipo de información que puede llegar a tener, esto es, si cuando se crea una base de datos se especifica que un determinado campo va a contener fechas, no se les podrá introducir cadenas alfanuméricas; los datos que admitirá el campo serán únicamente de tipo fecha.

Existen los siguientes tipos de campos definidos en una base de datos:

Alfanuméricos: Admiten letras, números y cualquier carácter ASCII. Estos campos son de longitud fija, admitiendo un texto predefinido por la longitud dada en su estructura.

Numéricos: Sólo admiten números de tipo entero y decimal. El signo también puede ser introducido

Lógicos: Aceptan solamente dos estados posibles: Verdadero y Falso.

Fecha: Los datos que aceptan serán de tipo de fecha, esto es, respetando el formato de fecha en días, meses, años, según se seleccione posteriormente en el programa.

Memo: Son campos especiales, debido a que permitan almacenar un texto largo.

ANTECEDENTES

Pueden contener cualquier carácter ASCII. Este campo se graba en un archivo distinto al de datos. Este archivo sólo contendrá el contenido de los campos *memo*. El archivo MEMO se creará simultáneamente con la base de datos al momento de seleccionar un campo memo y tendrá extensión *dbf*

La creación de una base de datos se realizará a través de la utilería DBU. Con esta utilería se pueden crear bases de datos, archivos de índices, así como editar sus contenidos, reemplazar valores, copiar y agregar registros.

MARCO TEORICO

2

2.1 ANALISIS Y DISEÑO (METODOLOGIA CASE)

La utilización de CASE (Ingeniería de sistemas Asistida por computadora) afecta a todas las fases del ciclo de vida del software, desde la definición de los requerimientos, pasando por el análisis, el diseño, la codificación, las pruebas y el mantenimiento. El empleo de CASE también cambia y perfila el ciclo de vida del software.

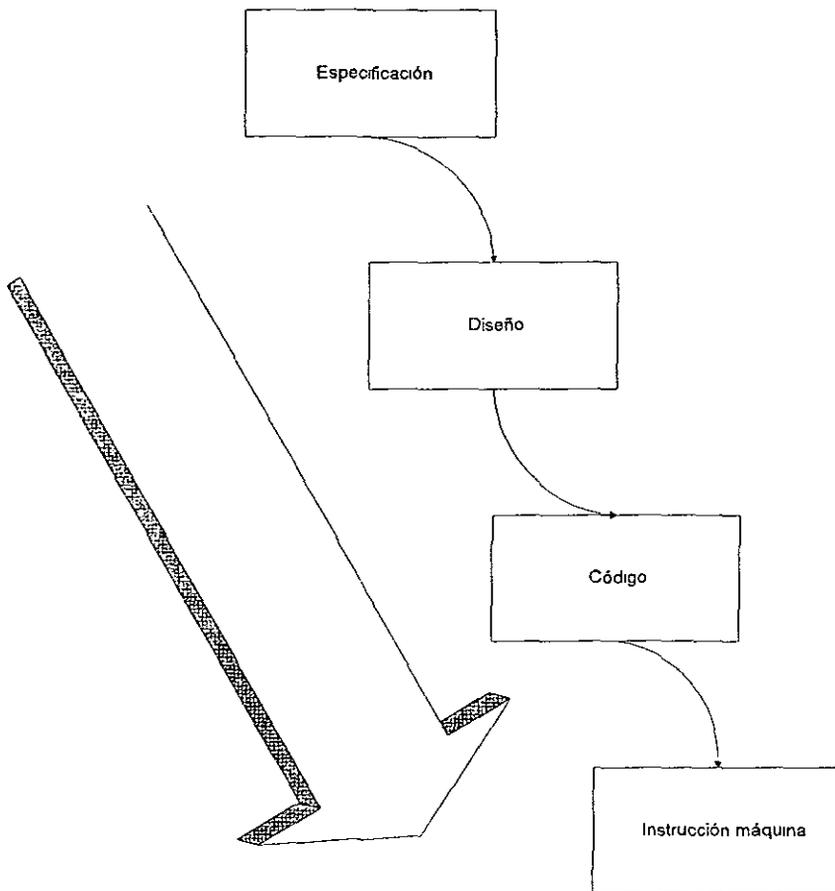


Figura 2.1.1 La parte de desarrollo dentro del ciclo de vida del software puede verse como una serie de transformaciones donde los objetivos del sistema se transforman en especificaciones de los requerimientos, después en especificaciones de diseño, seguidamente en programas en código y finalmente en instrucciones de máquina.

Un concepto unificador fundamental en la ingeniería de software es el ciclo de vida del software.

El ciclo de vida es un proceso de fases múltiples que comienza con la definición de un problema y continua hasta el envejecimiento, reemplazo o destrucción del sistema. El ciclo de vida típico se compone de cinco fases básicas (figura 2.1.2):

1. Análisis
2. Diseño
3. Codificación
4. Pruebas
5. Operación y mantenimiento

El *análisis* es el proceso de la definición de los requerimientos para una solución del problema. Durante este análisis se examinan las necesidades de los usuarios y se definen las propiedades que debe poseer el sistema para satisfacer esas necesidades. También se identifican las restricciones y necesidades de funcionamiento. Se deben definir de modo preciso las funciones a realizar, pero no su funcionamiento. El principal resultado de esta fase es la especificación del sistema. Idealmente la especificación del sistema establece las propiedades del sistema de forma precisa, comprobable y formal.

El *diseño* es el proceso de planificar cómo va a construirse el sistema; esto es, determinar los componentes de datos y procedimientos necesarios y cómo esos componentes se ensamblarán para formar la solución. Las especificaciones del sistema, los requerimientos del problema y las restricciones definidas en la fase de análisis se utilizarán como entrada en la fase de diseño.

La *codificación* es el proceso de transformar el diseño en instrucciones máquina. El objetivo de esta fase es producir programas correctos y eficientes. Las pruebas de los módulos codificados se suelen realizar durante esta fase.

Las *pruebas* es el proceso de demostrar que un software satisface los requerimientos del problema y funciona correctamente para todos los posibles datos de entrada. El objetivo de las pruebas es la eliminación de todas las condiciones y fallos inesperados del programa y descubrir cualquier implantación incorrecta de los requerimientos del problema. Los módulos de prueba se integran y prueban en grupos cada vez más grandes hasta que se ha probado todo el sistema.

La *operación y el mantenimiento* son los procesos de la ejecución del sistema en un modo de producción manteniendo un buen orden de operación. Esto implica.

- 1 Corregir errores y deficiencias que no se descubrieron durante el desarrollo del sistema.
- 2 Modificar el sistema para satisfacer los cambios de los requerimientos, adaptándolos a los cambios en el entorno y mejorando la eficiencia de operación y calidad general. Todos los recursos y actividades requeridos para asegurar que el software continúa satisfaciendo o excede las capacidades de operación requeridas se consideran parte del proceso de mantenimiento.

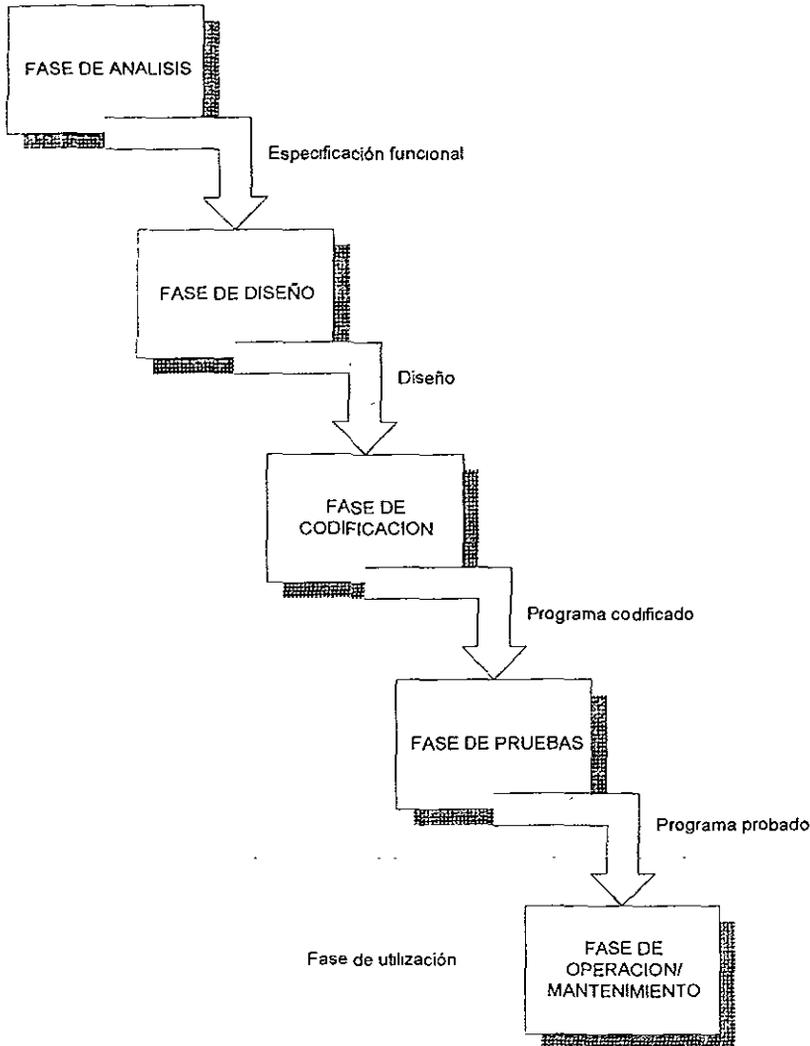


Figura 2.1.2. El ciclo de vida tradicional del software consta de cinco fases secuenciales: análisis, diseño, codificación, pruebas y operación/mantenimiento.

La figura 2.1.3 muestra un modelo sencillo del ciclo de vida del software CASE. Podemos observar que la fase del prototipo ha reemplazado a la tradicional fase de análisis del sistema. Todas las fases están soportadas por herramientas CASE automatizadas, las fases de comprobación del diseño y de generación de código son las fases más automatizadas del ciclo de vida.

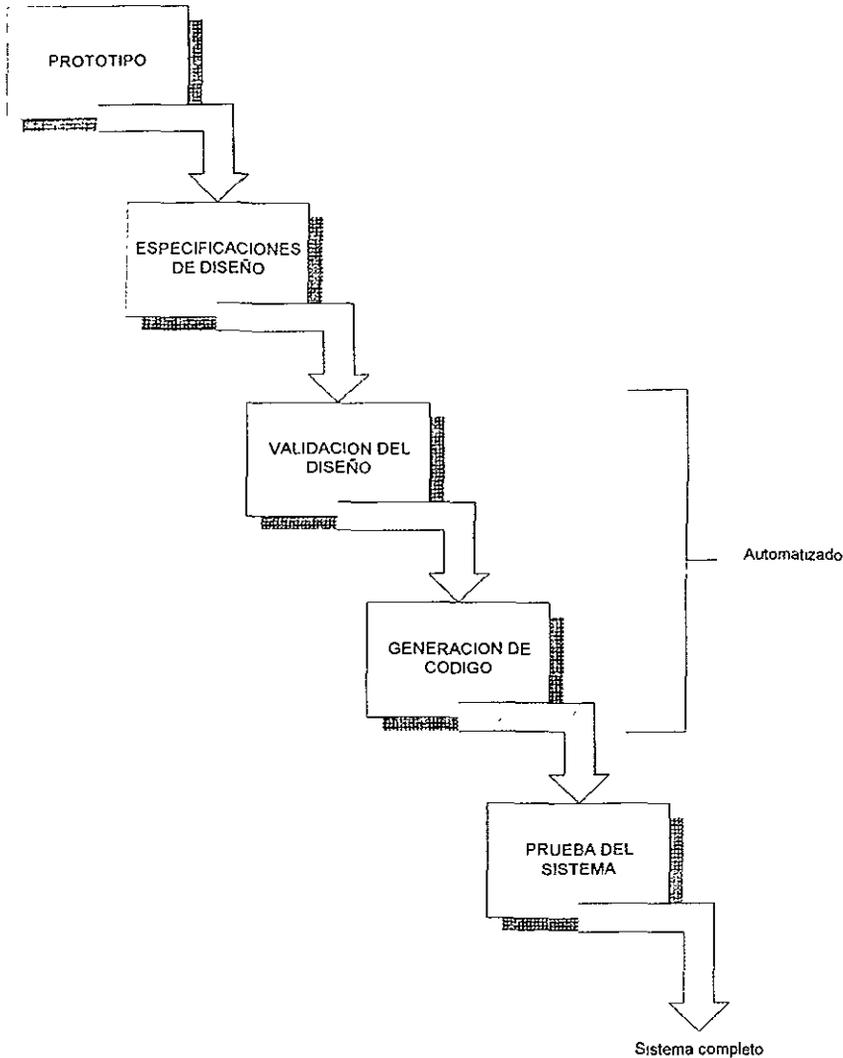


Figura 2.1.3. Una visión simple del ciclo de vida CASE del software, que muestra el empleo del prototipo y las fases automatizadas de la validación del diseño y de la generación del código.

Sin embargo, hay algunos problemas y limitaciones con este modelo de ciclo de vida. Al igual que con el modelo tradicional mostrado en la figura 2.1.3 desfigura el proceso de desarrollo del software en la práctica. Ambos describen el proceso como secuencia de pasos únicos en el cual las fases se siguen unas a otras. Pero en la práctica del desarrollo del software raramente es un simple proceso secuencial. Más frecuentemente es un proceso iterativo donde las fases se repiten y el proyecto entero se recicla una y otra vez durante la vida del sistema.

Un segundo problema con dos de estos modelos simples de ciclo de vida es que ponen demasiado énfasis en las primeras fases del ciclo de vida. Las formulaciones CASE para el desarrollo presuponen un análisis y diseño muy detallados para lograr una calidad más alta a un precio más bajo. El mantenimiento es la fase más extensa del ciclo de vida y la mayoría de las veces es mucho más costosa que todo el desarrollo. Para controlar los problemas de mantenimiento y los costos, se ha diseñado el sistema pensando en la facilidad de mantenimiento.

El modelo tradicional del ciclo de vida ignora las tareas del mantenimiento hasta llegar a esa fase, y ya es demasiado tarde para incorporar la posibilidad de mantenimiento. Sin embargo, la tecnología CASE consigue un sistema de mantenimiento fácil porque "puede mantenerse".

Definida de una forma simple, la ingeniería de sistemas asistida por computadora es la automatización del desarrollo del sistema, comúnmente referida como tecnología CASE (Computer Aided Systems Engineering).

De esta manera, CASE propone una nueva formulación del concepto de ciclo de vida de un sistema, basada en la automatización. La idea básica que subyace en la utilización de CASE para el desarrollo del presente proyecto es que nos proporciona un conjunto de herramientas bien integradas y que ahorran trabajo, enlazando y automatizando, sino todas, sí la mayoría de las fases del ciclo de vida del sistema.

La tecnología CASE es una combinación de herramientas de software y de metodologías. Debido a esto y a que es extensiva a todas las fases del ciclo de vida del sistema, CASE es una de las tecnologías de sistemas más completa hoy en día.

Algunas de las ventajas que consideramos para utilizar la tecnología CASE fueron las siguientes

- Permite las técnicas estructuradas.
- Aumenta la calidad del software mediante comprobación automática.
- Favorece la realización de prototipos.

- Simplifica el mantenimiento del programa.
- Acelera el proceso de desarrollo
- Posibilita la reutilización de los componentes del sistema.

Resumiendo, CASE ha cambiado radicalmente la forma de construir los sistemas al proporcionar tres avances principales:

1. Un entorno de desarrollo interactivo con un tiempo de respuesta rápido, recursos dedicados y una comprobación de errores desde el principio, ahorrándonos, además de tiempo, esfuerzo en la corrección de errores que pudieran detectarse demasiado tarde
- 2 La automatización de muchas tareas de desarrollo y mantenimiento del sistema Redundando de igual manera que en el punto anterior en un ahorro de tiempo y esfuerzo considerables
- 3 Una programación visual proporcionada por potentes interfaces gráficas. Esto es, sin duda, una de las características más consideradas por los usuarios finales, ya que el lenguaje de las imágenes es más rico que el de los textos, y se puede representar mucha más información en una imagen que en un texto además de que el lector puede comprender el significado más rápidamente.

El fin último de la tecnología CASE es automatizar todo el ciclo de vida del software mediante un conjunto de herramientas de software integradas. No obstante, el acceso a este tipo de software es un tanto difícil, sin mencionar el costo elevado del mismo y de presentar ciertas complicaciones en su manejo, al menos en la herramienta CASE de ORACLE que teníamos contemplada utilizar para el diseño de los diagramas: de flujo, estructurado y entidad-relación. Sin embargo, no fue ningún obstáculo para llevar a cabo dichos diagramas.

2.1.1 COMPONENTES DE UN SISTEMA CASE

Por otra parte, un sistema CASE lo podemos dividir en tres componentes básicos



Figura 2.1.4 El banco de trabajo (workbench) consta de tres componentes básicos. 1) el front-end para soportar el análisis y el diseño; 2) el depósito para almacenar toda la información del sistema; y 3) el back-end para soportar la implantación del programa.

El componente front-end corresponde a las fases primarias del ciclo de vida del sistema; es decir, el análisis y el diseño. El front-end puede corresponder a la parte de la computadora personal o estación de trabajo de la plataforma de hardware CASE. Las herramientas CASE front-end *proporcionan funciones para soportar las actividades de análisis y de diseño, como diagramación, prototipos y comprobación de especificaciones.*

El componente back-end corresponde a las últimas fases del ciclo de vida del sistema; es decir, la implantación y el mantenimiento del programa. Las herramientas del back-end CASE automatizan el código, las comprobaciones, la generación de las bases de datos, la normalización de los mismos.

Finalmente el depósito central o depósito de información es el enlace entre los componentes front-end y back-end de un sistema CASE. Es el vehículo de la comunicación por el cual toda la información del sistema reunida durante el ciclo de vida del sistema se gestiona y comparte

La combinación efectiva de estos tres componentes, aunado a una correcta toma de decisiones nos permitirá unificar procedimientos para la conclusión pronta y oportuna de nuestro sistema.

2.2 DISEÑO DE LA BASE DE DATOS

Para ser congruentes con la utilización de la tecnología CASE y debido a que en un sistema de información computarizado no sólo se requiere de los datos, sino de la estructura de esos datos (localización física), un sistema manejador de base de datos relacional (RDBMS, de las siglas en inglés Relational Data Base Management System) es lo más apropiado a considerar en nuestro caso. Un RDBMS puede reunir y estructurar los archivos relativos a muchos usuarios permitiendo que se realicen de manera fácil la obtención, manipulación, y almacenamiento de los datos. De las figuras 2.2.1 y 2.2.2 podemos apreciar la diferencia entre un sistema tradicional de archivos y un DBMS.

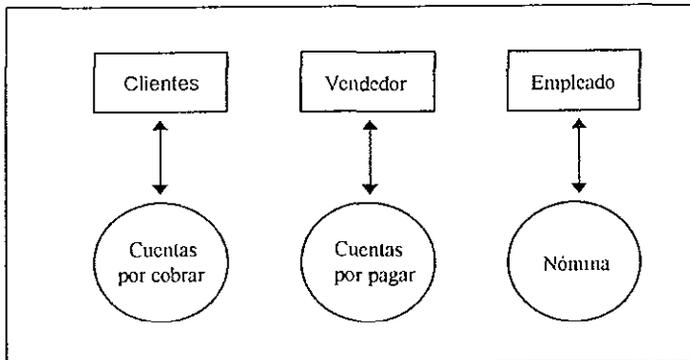


Figura 2.2.1. Sistema de archivos tradicional

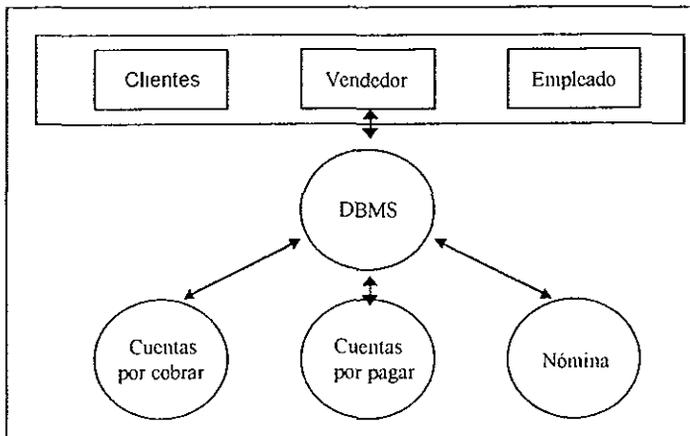


Figura 2.2.2. Sistema manejador de base de datos

Existen muchas características que hacen de una DBMS un sistema de procesamiento de datos muy eficiente, lo cual influyó de sobre manera en la toma de decisiones con respecto a la elección de este sistema manejador de base de datos para el manejo de la información en nuestro sistema. Algunas de estas características se enumeran a continuación.

- El sistema se ocupa de controlar los accesos a la base de datos.
- El sistema se encarga de gestionar la posible concurrencia de varios usuarios a la base de datos.
- El sistema se encarga de realizar las correspondientes operaciones para la seguridad e integridad de los datos. Un DBMS generalmente incluye un password para controlar el acceso a los datos que requieran una mayor seguridad. Por la limitante de acceso de sólo lectura o sólo escritura para los registros, los passwords pueden prevenir que los usuarios obtengan o alteren datos.
- Independencia de programas y archivos.
- Versatilidad de acceso. Los usuarios pueden obtener datos de muchas maneras. Se puede tener acceso secuencial para la muestra de datos en un orden preestablecido, y un acceso aleatorio para una obtención rápida de un registro específico.
- Fácil mantenimiento a programas. Los cambios y reparaciones a un sistema son relativamente fáciles de llevar a cabo.
- Fácil obtención de información especial. La generación de reportes de propósito especial pueden producirse con un mínimo esfuerzo.

Algunas de las grandes diferencias con respecto al sistema tradicional de archivos y que suelen considerarse como desventajas de los DBMS son:

- Hardware adicional. La utilización de un DBMS requiere de hacerse de memoria adicional y/o unidades de disco. La memoria es utilizada para almacenar el software del DBMS, y las unidades de disco extra son utilizadas para mantener los archivos especiales que el DBMS requiere. Sin embargo, con esto se estará consiguiendo una menor redundancia en el manejo de los datos.
- Equipo entrenado. Los programadores que no están familiarizados con los conceptos que se manejan en un DBMS y con su terminología necesitan de un entrenamiento especial para orientarlos con respecto al nuevo ambiente del DBMS.

- Software costoso
- Equipo especial. La organización puede llegar a necesitar de la contratación de un experto DBMS para supervisar y manejar el DBMS. Comúnmente conocido como administrador de la base de datos, esta persona establece las reglas de cómo se permitirá el uso de los datos, el monitoreo de la seguridad de los mismos para evitar la intromisión de personal no autorizado, y la orientación adecuada a los analistas y programadores acerca del mejor camino para utilizar el DBMS.

A pesar de las desventajas existentes en cuanto a la consideración de un DBMS concluimos que es el mejor camino a seguir, y tomando en cuenta que el cliente cuenta ya con el software adecuado para su óptima implantación, nos hemos decidido sin lugar a dudas en la adopción del DBMS como sistema de archivos a utilizar.

Las características principales que presentan las herramientas de Oracle para el manejo de la base de datos son:

- Es relacional
- Opera sobre PC's
- Permite conexión en red
- Se permite tener control sobre la base de datos
- Especialmente por su seguridad de acceso

A continuación se presentan las tablas (entidades) que conforman el sistema, las columnas (atributos) de que se componen dichas tablas, y el tipo de dato de que se trata.

Aquellos atributos que se encuentran subrayados por una línea continua indica que se trata de la llave principal, y aquellos que se encuentran subrayados por una línea punteada indica que se trata de llaves foráneas.

ACCIONES

<u>NO_ACCION</u>	NOT NULL	NUMBER (3)
<u>NO_SAT</u>	NOT NULL	NUMBER (5)
<u>FECHA_ACCION</u>	NOT NULL	DATE
<u>NO_STATUS</u>		NUMBER (2)
<u>NO_PRIORIDAD</u>		NUMBER (1)
<u>CVE_EMITOR</u>	NOT NULL	VARCHAR2 (8)
<u>STATUS_ACCION</u>	NOT NULL	NUMBER (1)
<u>CVE_RECEPTOR</u>	NOT NULL	VARCHAR2 (8)
<u>TIEMPO</u>		VARCHAR2 (8)

AREAS

<u>NO_AREA</u>	NOT NULL	NUMBER (2)
<u>NOMBRE</u>	NOT NULL	VARCHAR2 (32)

CLIENTE_TIPO

<u>NO_CLIENTE</u>	NOT NULL	NUMBER (6)
<u>CVE_TIPO</u>	NOT NULL	NUMBER (5)

CLIENTES

<u>NO_CLIENTE</u>	NOT NULL	NUMBER (2)
<u>NOMBRE</u>	NOT NULL	VARCHAR2 (20)
<u>NO_RAZON_SOCIAL</u>	NOT NULL	NUMBER (6)
<u>COMENTARIO</u>		VARCHAR2 (80)
<u>TELEFONO</u>		VARCHAR2 (30)
<u>FAX</u>		VARCHAR2 (20)
<u>ALIAS</u>		VARCHAR2 (25)
<u>CALLE_NUMERO</u>		VARCHAR2 (50)
<u>COLONIA</u>		VARCHAR2 (50)
<u>CP</u>		VARCHAR2 (5)
<u>CIUDAD</u>		VARCHAR2 (25)
<u>ESTADO</u>		VARCHAR2 (25)

CONTACTOS

<u>NO_CONTACTO</u>	NOT NULL	NUMBER (6)
NOMBRE	NOT NULL	VARCHAR2 (32)
TELEFONO	NOT NULL	VARCHAR2 (20)
<u>NO_CLIENTE</u>		NUMBER (6)
FAX		VARCHAR2 (20)
PUESTO		VARCHAR2 (32)
MODEM		VARCHAR2 (20)
CALLE_NUMERO		VARCHAR2 (50)
COLONIA		VARCHAR2 (50)
CP		VARCHAR (5)
CIUDAD		VARCHAR2 (25)
ESTADO		VARCHAR2 (25)

CONTACTO_TIPO

<u>CVE_TIPO</u>	NOT NULL	NUMBER (2)
<u>NO_CONTACTO</u>	NOT NULL	NUMBER (6)

CONTRATOS

<u>NO_CONTRATO</u>	NOT NULL	NUMBER (6)
<u>NO_CLIENTE</u>	NOT NULL	NUMBER (6)
<u>CVE_VENDEDOR</u>	NOT NULL	VARCHAR2 (8)
ESTADO		VARCHAR2 (16)
FECHA_DE_RECONOCIMIENTO		DATE
O		
TERMINOS_PAGO		VARCHAR2 (200)
TOTAL_DOLARES		NUMBER (12, 2)
TOTAL_A_PAGAR		NUMBER (12, 2)
FECHA_FIRMA		DATE
COMENTARIO		VARCHAR2 (80)
INICIO_VIGENCIA		DATE

CONTRATOS_CONTACTOS

<u>NO_CONTRATO</u>	NOT NULL	NUMBER (6)
<u>NO_CONTACTO</u>	NOT NULL	NUMBER (6)
SUPLEMENTO	NOT NULL	VARCHAR2 (3)

EMPLEADOS

<u>CVE_EMPLEADO</u>	NOT NULL	VARCHAR2 (8)
NOMBRE	NOT NULL	VARCHAR2 (32)
NO_AREA	NOT NULL	NUMBER (2)
FECHA_EGRESO		DATE
PUESTO		VARCHAR2 (32)
FECHA_INGRESO		DATE

ESTADO

NO_ESTADO	NOT NULL	NUMBER (3)
TEXTO		VARCHAR2 (32)

INFO_ACCIONES

NO_RENGLON	NOT NULL	NUMBER (3)
NO_ACCION	NOT NULL	NUMBER (3)
NO_SAT	NOT NULL	NUMBER (6)
TEXTO		VARCHAR2 (80)

LICENCIAS

<u>NO_LICENCIA</u>	NOT NULL	NUMBER (6)
NO_CONTRATO	NOT NULL	NUMBER (6)
MONTO		NUMBER (12, 2)
MONTO_DOLARES		NUMBER (12, 2)
NO_PLATAFORMA	NOT NULL	NUMBER (3)
FECHA_TERMINO		DATE

MEDIO_DIST

NO_MEDIO	NOT NULL	NUMBER (1)
DESCRIPCION	NOT NULL	VARCHAR2 (20)

NIVEL_SERVICIO

NO_NIV_SER	NOT NULL	NUMBER (1)
DESCRIPCION	NOT NULL	VARCHAR2 (20)

PLATAFORMAS

<u>NO_PLATAFORMA</u>	NOT NULL	NUMBER (3)
DESCRIPCION	NOT NULL	VARCHAR2 (50)

PLATAFORMAS_CLIENTE

<u>NO_SOPT_CLIE</u>	NOT NULL	NUMBER (6)
<u>NO_SERIR</u>	NOT NULL	VARCHAR2 (32)
<u>VERSION_SO</u>	NOT NULL	VARCHAR2 (20)
<u>NO_PLATAFORMA</u>	NOT NULL	NUMBER (3)
<u>MEDIO_DISTRIBUCION</u>		NUMBER (1)
<u>FECHA_NOTIF_NIC</u>		DATE

PRIORIDADES

<u>NO_PRIORIDAD</u>	NOT NULL	NUMBER (3)
<u>DESCRIPCION</u>	NOT NULL	VARCHAR (50)

PRODUCTOS

<u>NO_PRODUCTO</u>	NOT NULL	NUMBER (3)
<u>DESCRIPCION</u>	NOT NULL	VARCHAR2 (50)

PRODUCTOS_CONTRATADOS

<u>NO_PARTIDA</u>		NUMBER (6)
<u>NO_SOPT_CLIE</u>		NUMBER (6)
<u>NO_LICENCIA</u>	NOT NULL	NUMBER (6)
<u>NO_PRODUCTO</u>	NOT NULL	NUMBER (3)
<u>COSTO_DOLARES</u>		NUMBER (1, 2)
<u>VERSION</u>		VARCHAR2 (20)
<u>COSTO_PESOS</u>		NUMBER (10, 2)
<u>CANTIDAD</u>		NUMBER (4)
<u>TIPO_LICENCIA</u>		VARCHAR2 (1)

RAZON_SOCIAL

<u>NO_RAZON_SOCIAL</u>	NOT NULL	NUMBER (6)
<u>NOMBRE</u>	NOT NULL	VARCHAR2 (60)

SOPORTES

<u>NO_SOPORTE</u>	NOT NULL	NUMBER (6)
<u>FECHA_TERMINO</u>	NOT NULL	DATE
<u>NO_CONTRATO</u>	NOT NULL	NUMBER (6)
<u>MONTO</u>		NUMBER (12, 2)
<u>NO_SOPT_CLIE</u>		NUMBER (6)
<u>MONTO_DOLARES</u>		NUMBER (12, 2)
<u>NIVEL_SERV</u>		NUMBER (1)

STATUS_SATS

<u>NO_STATUS</u>	NOT NULL	NUMBER (2)
<u>DESCRIPCIO</u>	NOT NULL	VARCHAR2 (50)

SATS

<u>NO_SAT</u>	NOT NULL	NUMBER (6)
<u>ENCABEZADO</u>	NOT NULL	VARCHAR2 (80)
<u>FECHA_CREACION</u>	NOT NULL	DATE
<u>NOMBRE_CONTACTO</u>	NOT NULL	VARCHAR2 (32)
<u>NOMBRE_CLIENTE</u>	NOT NULL	VARCHAR2 (100)
<u>NO_PRIORIDAD</u>	NOT NULL	NUMBER (1)
<u>NO_STATUS</u>	NOT NULL	NUMBER (2)
<u>NO_PRODUCTO</u>	NOT NULL	NUMBER (3)
<u>FECHA_ACTUALIZACION</u>	NOT NULL	DATE
<u>TELEFONO</u>	NOT NULL	VARCHAR2 (30)
<u>CVE_ING_SOPORTE</u>		VARCHAR2 (8)
<u>NO_SOPORTE</u>	NOT NULL	NUMBER (6)
<u>NO_PLATAFORMA</u>	NOT NULL	NUMBER (3)
<u>CVE_RESPONSABLE</u>		VARCHAR2 (8)
<u>COMENTARIO</u>		VARCHAR2 (80)
<u>CVE_ERROR</u>		VARCHAR2 (3)
<u>CVE_PCR</u>		VARCHAR2 (7)
<u>MEDIO_DISTRIBUCION</u>	NOT NULL	NUMBER (1)
<u>FECHA_ESCALAMIENTO</u>		DATE
<u>VERSION_SO</u>		VARCHAR2 (20)
<u>VERSION_RDBMS</u>	NOT NULL	VARCHAR2 (20)
<u>VERSION_PRODUCTO</u>	NOT NULL	VARCHAR2 (20)
<u>NO_ERROR</u>		NUMBER (6)
<u>MODEM</u>		VARCHAR2 (30)
<u>FAX</u>		VARCHAR2 (30)
<u>CVE_TAR</u>		VARCHAR2 (11)
<u>DIRECCION1</u>		VARCHAR2 (60)
<u>NIVEL</u>		NUMBER (1)

TIPOS_D_CLIENTE

<u>CLAVE</u>	NOT NULL	NUMBER (5)
DESCRIPCION	NOT NULL	VARCHAR2 (80)

TIPOS_D_CONTACTO

<u>CLAVE</u>	NOT NULL	NUMBER (2)
DESCRIPCION	NOT NULL	VARCHAR2 (32)

2.3 ORACLE RDBMS

Existen varias razones por las cuales se eligió el manejador de bases de datos de Oracle. Pero antes de entrar en detalles del por qué la elección de este manejador, podemos listar brevemente las características básicas que debe cumplir un manejador de base de datos.

Un servidor de base de datos es la clave para resolver los problemas del manejo de la información. En general, un servidor debe manejar de forma confiable una gran cantidad de datos en un ambiente multi-usuario para que varios usuarios puedan acceder de forma concurrente a los mismos datos. Por supuesto que esto se debe cumplir con un buen grado de performance. Además un servidor de bases de datos también debe prevenir el acceso sin autorización a la información, así como proporcionar soluciones eficientes para fallas de recuperación. El Oracle Server proporciona soluciones efectivas con las siguientes características:

Ambiente cliente/servidor (Procesamiento distribuido). Para tomar una amplia ventaja de un sistema o de una red, Oracle permite que el procesamiento se divida entre el servidor de la base de datos y las aplicaciones del cliente. La computadora que corre el manejador de la base de datos maneja todas las responsabilidades del servidor mientras que las estaciones que ejecutan las aplicaciones de la base de datos se concentran en la interpretación y despliegue de la información.

Manejo de espacio y de bases de datos grandes. Oracle soporta la base de datos más grande, la cual potencialmente puede ser de terabytes. Para que se justifique y realmente sea eficiente el uso de dispositivos de hardware, Oracle permite el control completo del espacio.

Varios usuarios conectados concurrentemente a la base de datos. Oracle soporta gran cantidad de usuarios concurrentes ejecutando una gran variedad de aplicaciones de la base de datos operando con los mismo datos. Minimizando la contención y garantizando la concurrencia.

Alta disponibilidad. En algunos lugares, Oracle trabaja las 24 horas del día sin permitir que se dé de baja el sistema. Por lo que las operaciones normales del sistema tales como respaldo de la base de datos y fallas parciales del sistema no interrumpen el uso de la base de datos

Disponibilidad controlada. Oracle puede controlar selectivamente la disponibilidad de los datos, al nivel de la base de datos y a un nivel más bajo. Por ejemplo, un administrador puede deshabilitar el uso de una aplicación en específico para que los datos de la aplicación puedan ser recargados sin afectar a otras aplicaciones

Apertura a los estándares de la industria. Oracle se adhiere a los estándares aceptados por los lenguajes de acceso a los datos, sistemas operativos, interfaces de usuarios, y protocolos de comunicación de redes.

Oracle también soporta el Simple Network Management Protocol (SNMP) estándar para el manejo de sistemas. Este protocolo permite a los administradores el manejo de sistemas heterogéneos con una sola interfaz de administración.

Manejo de la seguridad. Para proteger el acceso no autorizado a la base de datos, Oracle proporciona características de seguridad fail-safe para limitar y monitorear el acceso a los datos. Estas características le permiten manejar fácilmente incluso hasta el diseño más complejo para el acceso a los datos.

Sistemas distribuidos. Para ambientes distribuidos de redes, Oracle combina los datos físicamente localizados en diferentes computadoras en una base de datos lógica que puede ser accesada por todos los usuarios de la red. Los sistemas distribuidos tienen el mismo grado de transparencia y consistencia de los datos que los sistemas no distribuidos, pero los primeros reciben las ventajas del manejo de las bases de datos locales. Además, Oracle permite a los usuarios acceder a los datos de otras bases de datos de manera transparente.

Portabilidad. El software de Oracle es portable para trabajar bajo diferentes sistemas operativos. Las aplicaciones desarrolladas por Oracle pueden ser implementadas en cualquier sistema operativo, con algunas o ninguna modificación.

Compatibilidad. Es compatible con los estándares de la industria, incluyendo la mayoría de los estándares de los sistemas operativos.

Ambientes replicados. Permite replicar grupos de tablas y sus objetos a múltiples sitios.

2.3.1 EL SERVIDOR ORACLE

El servidor Oracle es un sistema de manejo de bases de datos relacional. El servidor de Oracle consiste de una base de datos Oracle y una instancia Oracle; para comprender mejor acerca de la relación entre una instancia y una base de datos lo describiremos a continuación de una forma breve, sin pretender entrar a fondo en la misma, pero tratando de abarcar los puntos más sobresalientes e importantes.

SQL (Structured Query Language). Es el lenguaje de programación que define y manipula la base de datos. Las bases de datos SQL son relacionales; esto significa simplemente que los datos son almacenados en un conjunto de relaciones simples. Una base de datos puede tener una o más tablas, y cada tabla tiene columnas y renglones. Se pueden definir y manipular los datos en una tabla con los comandos SQL. Los comandos DDL (Data Definition Language) son utilizados para crear y alterar las bases de datos y las tablas.

Es posible actualizar, eliminar o recuperar datos en una tabla con los comandos DML (Data Manipulation Language).

Además del SQL, el servidor de Oracle tiene un lenguaje de procedimientos llamado PL/SQL, el cual permite programar sentencias SQL, facilitando de esta manera el control del flujo de un programa SQL, para usar variables y escribir procedimientos de manejo de errores.

2.3.1.1 ESTRUCTURA DE LA BASE DE DATOS

Una base de datos Oracle tiene una estructura física y una estructura lógica. Debido a que dichas estructuras están separadas, el almacenamiento físico de los datos puede ser manipulado sin afectar el acceso a las estructuras de almacenamiento lógico.

Estructura física de la base de datos. La estructura física de una base de datos Oracle está determinada por los archivos del sistema operativo, los cuales constituyen la base de datos. Cada base de datos Oracle se compone de tres tipos de archivos: uno o más archivos de datos, dos o más archivos de redo log y uno o más archivos de control. El conjunto de estos archivos componen la parte física para el almacenamiento de la información.

Estructura lógica de la base de datos. La estructura lógica de la base de datos está determinada por:

Uno o más tablespaces. (Un tablespace es el área lógica de almacenamiento)
El esquema de objetos de la base de datos. Un esquema es una colección de objetos. Este esquema es el que hace referencia directa a los datos de la base de datos. Los esquemas incluyen estructuras tales como tablas, vistas, secuencias, procedimientos almacenados, sinónimos, índices, clusters y ligas.

2.3.1.2 UNA INSTANCIA ORACLE

Cada vez que una base de datos es iniciada, una área global del sistema (SGA) es asignada, los procesos de background de Oracle son iniciados. El área global del sistema es una parte de la memoria utilizada para que la información de la base de datos sea compartida por los usuarios de la base de datos. A la combinación de los procesos de background y los buffers de memoria se les llama instancia Oracle

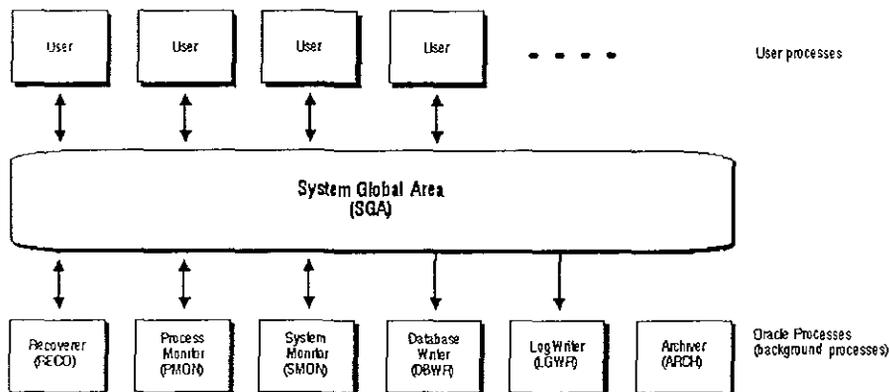


Figura 2.3.1 Una instancia Oracle

Una instancia Oracle tiene dos tipos de procesos: los procesos de usuario, y los procesos de servidor.

Un proceso de usuario ejecuta el código de una aplicación; un proceso servidor es el que se encarga de mantenimiento del servidor de Oracle.

2.3.1.3 ESTRUCTURA DE LA BASE DE DATOS Y MANEJO DEL ESPACIO

En nuestros días el modelo de bases de datos ampliamente aceptado es el modelo de base de datos relacional. Dicho modelo contempla tres puntos de suma importancia:

Estructuras. Las estructuras son objetos bien definidos (tales como tablas, vistas, índices, etc.) los cuales almacenan o accesan los datos de la base de datos. Las estructuras y los datos contenidos dentro de ellos pueden ser manipulados por operaciones.

Operaciones. Las operaciones son acciones claramente definidas las cuales permiten a los usuarios manipular los datos y estructuras de una base de datos. Las operaciones en una base de datos deben apegarse a un conjunto de reglas de integridad ya predefinidas.

Reglas de integridad. Las reglas de integridad son leyes que indican que operaciones son permitidas en los datos y estructuras de la base de datos. Las reglas de integridad protegen los datos de las estructuras de la base de datos.

Los sistemas de manejo de bases de datos relacionales ofrecen beneficios tales como.

- independencia del almacenamiento físico de los datos con respecto a la estructura lógica de los mismos.
- acceso variable y fácil a todos los datos
- reducción de la redundancia

Una base de datos Oracle es una colección de datos los cuales son tratados como una unidad. El propósito general de la base de datos es almacenar y recuperar la información relacionada. La base de datos tiene estructuras lógicas y físicas.

2.3.2 ESTRUCTURAS DE MEMORIA Y PROCESOS

Un servidor Oracle utiliza las estructuras de memoria y procesos para manejar y accesar la base de datos. Todas las estructuras de memoria residen en la memoria principal del sistema. Los procesos son tareas que trabajan en la memoria de las computadoras.

La siguiente figura muestra una variación típica de la memoria del servidor Oracle y los procesos.

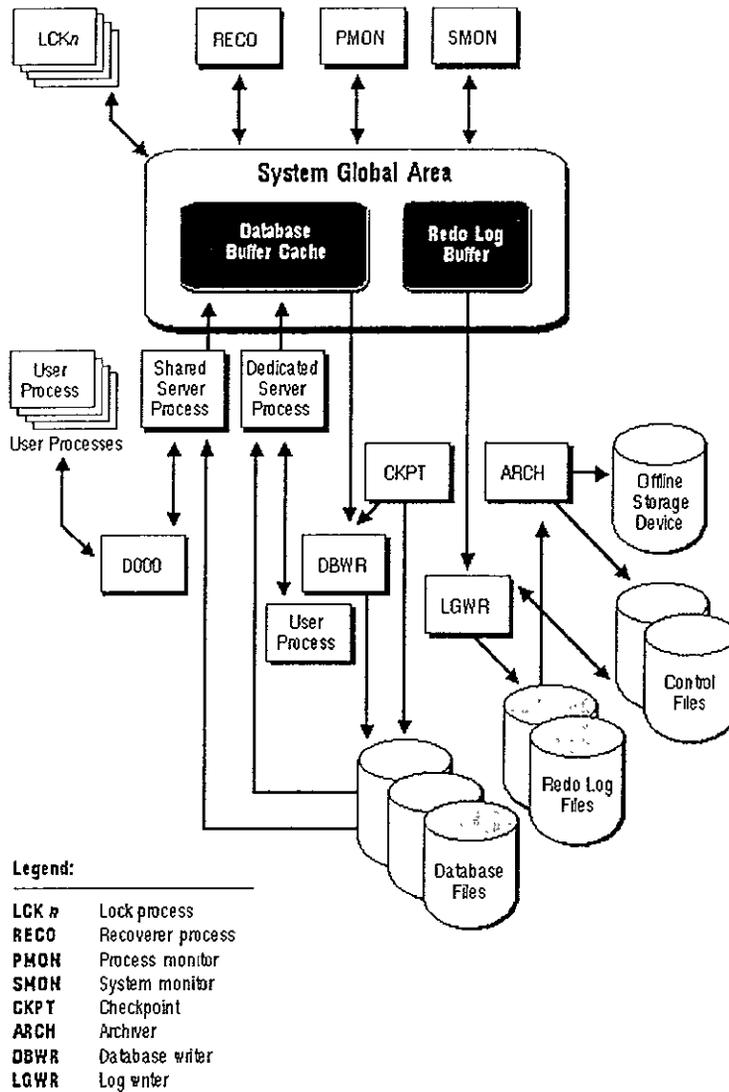


Figura 2.3.2 Estructura de memoria y procesos de Oracle

2.3.2.1 ESTRUCTURAS DE MEMORIA

Oracle crea y utiliza las estructuras de memoria para completar varias tareas. Por ejemplo, la memoria es utilizada para almacenar el código de un programa que se esté ejecutando y los datos que son compartidos por los usuarios. Existen varias estructuras de memoria que están estrechamente relacionadas con el servidor Oracle, tales como el área global del sistema (la cual incluye el database buffer cache, los redo log buffers y el shared pool) y las áreas globales de programa

Area Global de Sistema

El área global de sistema mejor conocida como SGA (System Global Area) es una región compartida de memoria que contiene datos e información de control para una instancia. El SGA y los procesos de background son los que constituyen a una instancia Oracle.

Oracle asigna el área global del sistema cuando se inicia una instancia y la remueve cuando la instancia se da de baja. Cada instancia tiene su propia área global del sistema.

Los usuarios conectados concurrentemente a un servidor Oracle comparten los datos en la SGA. Para que el performance sea el adecuado, la SGA debe ser tan grande como sea posible (siempre y cuando se conserve en memoria real) para almacenar tantos datos como sea posible y minimizando de esta forma la entrada/salida a disco. La información almacenada dentro de la SGA esta dividida en diferentes estructuras, incluyendo los database buffers cache, redo log buffer y el shared pool. Estas áreas tienen un tamaño fijo y son creadas cuando la instancia inicia.

Area Global de Programa

El área global de programa también conocida como PGA (Program Global Area) es un buffer de memoria que contiene datos y control de la información para un proceso servidor. Un PGA es creado por Oracle cuando un proceso del servidor es inicializado. La información en un PGA depende de la configuración de Oracle

2.3.2.2 PROCESOS

Un proceso es un mecanismo en un sistema operativo el cual puede ejecutar una serie de pasos. Algunos sistemas operativos manejan el término trabajo o tarea. Un proceso normalmente tiene su propia área de memoria privada en la cual se ejecuta.

Un servidor Oracle tiene dos tipos de procesos generales: procesos de usuario y procesos Oracle.

- *Procesos de usuario.* Un proceso de usuario se crea para ejecutar el código de una aplicación. El proceso de usuario también maneja la comunicación con el proceso servidor a través de la interfaz del programa.
- *Procesos Oracle.* Los procesos Oracle son llamados por otros procesos para ejecutar funciones que son requeridas por los procesos que los invocaron. Existen varios tipos de procesos de Oracle con funciones específicas.
- *Procesos de Servidor.* Oracle crea los procesos de servidor para manejar las peticiones de los procesos de usuario. Un proceso de servidor está a cargo de la comunicación con los procesos de usuario e interactuar con Oracle para llevar a cabo el requerimiento que fue solicitado por el proceso de usuario.

Oracle puede ser configurado para variar el número de procesos de usuario por procesos de servidor. En una configuración de servidor dedicado, un proceso de servidor maneja los requerimientos de un solo proceso de usuario. En una configuración multi-threaded server se permite que varios procesos de usuario compartan un pequeño número de procesos de servidor, minimizando el número de procesos de servidor y maximizando la utilización de los recursos disponibles del sistema.

En algunos sistemas, el proceso de usuario y el proceso de servidor están separados, mientras que en otros se combinan en un solo proceso. Si un sistema utiliza el multi-threaded server o si el proceso de usuario y el proceso de servidor se ejecutan en máquinas diferentes, el proceso de usuario y el proceso de servidor deben estar separados. En sistemas cliente / servidor dichos procesos se ejecutan en máquinas diferentes.

- *Procesos de Background.* Oracle crea un conjunto de procesos de background para cada instancia. Ellos consolidan funciones que de otra forma serían manejadas por múltiples programas ejecutándose por cada proceso de usuario. Los procesos de background ejecutan de forma asíncrona entrada / salida a disco y monitorean otros procesos para incrementar el paralelismo mejorando de esta forma el performance. Cada instancia Oracle puede utilizar varios procesos de background.

2.4 CONEXION VIA SQL*NET

Para que los clientes puedan tener acceso a la información contenida en el servidor Windows NT, es necesario hacer uso de un programa, así como de un protocolo de comunicación con los cuales se pueda llegar a establecer la comunicación deseada

SQL*Net V2 es la capa de comunicación por default de Oracle para todas las versiones de Oracle a partir de la versión 7 0.15. A continuación se describirá de manera básica la forma en que se debe de configurar SQL*Net V2.

SQL*Net es el programa que nos permite establecer la comunicación entre un proceso cliente y el proceso Server (proceso de back ground) en el manejador de la base de datos de Oracle, mediante un medio de red, usualmente llamado protocolo de comunicación.

El protocolo de comunicación vía SQL*Net V2 provee diferentes interfaces entre el código general de SQL*Net y la interfaz del Sistema Operativo, estas diferentes interfaces se presentan a continuación.

Sobre servidores UNIX todas las conexiones a la base de datos usan SQL*Net de una forma o de otra. La conexión local por default utiliza el adaptador para protocolo "Bequeath". Este usa los pipes de UNIX para establecer la comunicación entre en proceso Server de Oracle y el programa cliente. Para este caso en particular la interfaz no es hardware, en la figura anteriormente mostrada.

Los componentes que permiten establecer la conexión se muestran en la tabla siguiente.(Tabla 2.4.1). En la mayoría de los casos los tres componentes son necesarios para poder establecer la comunicación Una notable excepción es cuando la conexión se establece con el adaptador de protocolo "Bequeath".

Nombre	Cliente	Listener	Servidor
Propósito:	Utilizar la interfaz de comunicación entre el usuario y el RDBMS	Estar pendiente de las peticiones del cliente para establecer la conexión	Realizar las operaciones de la base de datos recibidas directamente por SQL
Ejemplo:	SQL/PLUS	tnslsnr	Oracle
Archivos de configuración:	Sqlnet.ora Tnsnames.ora	Listener.ora Sqlnet.ora	

Tabla 2.4.1

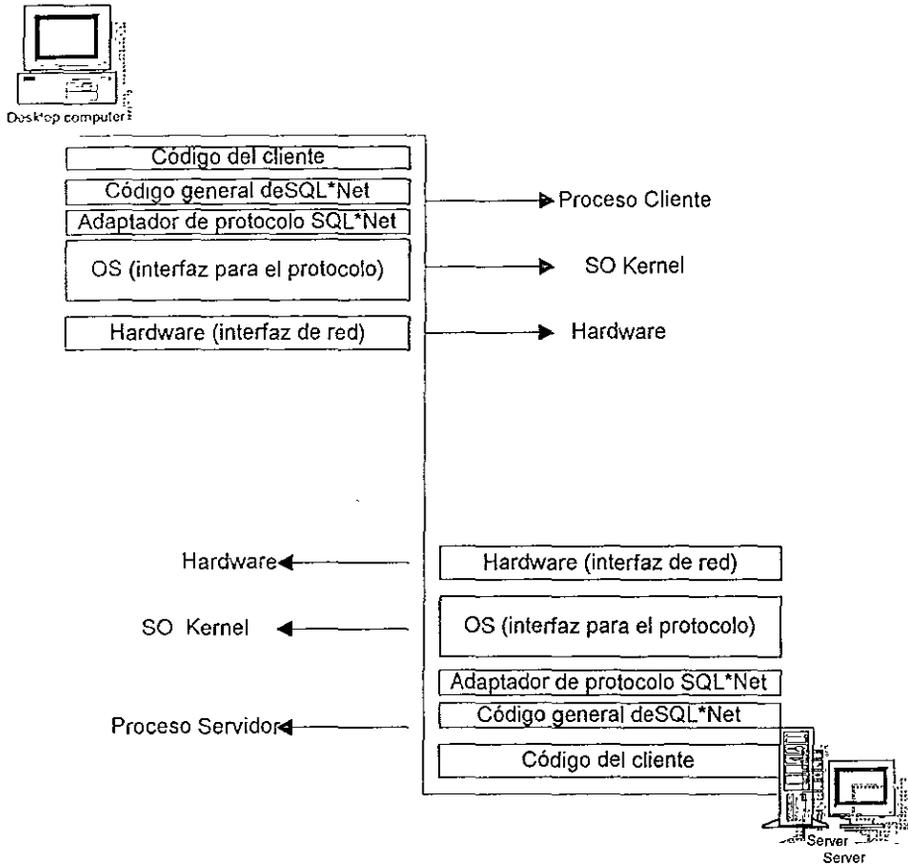


Figura 2.4 1 Procesos que se llevan a cabo tanto en cliente como en servidor

Al establecerse una conexión desde una herramienta cliente hacia el proceso servidor en el RDBMS en el servidor Windows NT se dispara la siguiente secuencia de pasos:

- a) El proceso cliente proporciona el identificador, password y la cadena de conexión. Ejemplo:

\$ sqlplus scott/tiger@MYALIAS

- b) El cliente procesa la información en base al archivo local de configuración en función del alias introducido (cadena de conexión). Realiza la búsqueda del archivo de configuración, para su lectura, en los siguientes lugares dentro del file system.

- \$HOME/tnsnames.ora
- TNS_ADMIN/tnsnames.ora

- /etc/tnsnames.ora
- \$ORACLE_HOME/network/admin/tnsnames.ora.

Sólo uno de estos archivos es cargado en primero que localice en base a la secuencia especificada anteriormente.

Si dentro del archivo tnsnames.ora localizado se encuentran definidos más de un alias, realiza una búsqueda del alias indicado al tratar de invocar la conexión. Para resolver correctamente el problema de conexión, el alias dentro del tnsnames debe de tener la siguiente estructura:

```
MYALIAS =  
  (DESCRIPTION =  
    (ADDRESS=(PROTOCOL=IPC) (KEY=mysid))  
    (CONNECT_DATA= (SID=mysid))  
  )
```

- c) Si el alias indicado, al tratar de realizar la conexión no es localizado dentro del archivo tnsnames.ora puede ser definido con el Easy Configuration.
- d) Si asumimos que el alias (cadena de conexión) fue localizado dentro del tnsnames, se realiza una llamada hacia la dirección IP definida dentro de la estructura de este alias. Esta llamada hacia la dirección se pretende que sea escuchada por el proceso listener. Es decir, el proceso cliente hace una petición hacia la dirección especificada para un servicio en particular.
- e) Si no responde es servicio solicitado así la dirección especificada el proceso cliente registra alguna otra expansión del alias y trata de realizar la conexión con la nueva expansión localizada.
- f) Asumiendo que el proceso listener escucho la petición en base a la dirección especificada y trata de localizar el servicio solicitado. Esto significa que el proceso listener identificó el servicio llamado "mysid". Si el servicio especificado no se llegará a localizar, por el lado del cliente despliega un error (ORA-12505).

Si el servicio es localizado entonces dependiendo de la máquina y del protocolo de comunicación es como se comporta el proceso listener

- i) En este momento el proceso listener establece la conexión entre el proceso Server y el proceso cliente.
- g) El cliente se encuentra ahora en comunicación con el proceso server.
- h) Normalmente no se llegan a generar problemas de conexión una vez que esta fue establecida. Pero en algunos casos en particular esta problemática se puede llegar a presentar.

2.4.1 CONFIGURACION

Como anteriormente se comentó, el proceso cliente utiliza el archivo llamado "tnsnames.ora" para localizar el alias o cadena de conexión especificado.

Una cosa importante a recordar cuando se configura la conexión por el lado del cliente es que en el alias se especifique.

- a) Que se determine de manera completa y clara la llamada al proceso listener.

Es decir que se delimite de manera detallada el bloque del alias a utilizar.

- b) Una vez que el servicio es solicitado el listener debe ser contactado.

Cuando el RDBMS fue instalado sobre plataformas UNIX, se puede hacer uso de la variable TNS_ADMIN para especificar de donde tomara el archivo tnsnames.ora. Si la variable de ambiente TNS_ADMIN no fue definida entonces realizará una búsqueda en el directorio /etc, si no se llegará a localizar aquí tampoco entonces busca el archivo bajo la ruta \$ORACLE_HOME/network/admin.

Tratando de establecer una conexión desde el cliente

- a) Crea un directorio en el cual tengas identificados los archivos de configuración de SQL*Net V2.
- b) Si te encuentras bajo ambiente UNIX crea la variable de ambiente TNS_ADMIN para especificar el directorio bajo el cual se encuentran los archivos de configuración. Asegúrate de exportar esta variable.
- c) Crea el archivo tnsnames.ora bajo el ejemplo indicado o utiliza el Easy Configuration.
- d) Revisa que el tnsnames.ora satisface los elementos necesarios para establecer la conexión.
- e) Asegúrate que alguna copia ejemplo del tnsnames.ora no se localiza bajo \$ORACLE_HOME.
- f) Prueba la conexión, normalmente se realiza esta prueba tratando de conectarse mediante SQL*PLUS.

Si la conexión llegara a fallar es necesario determinar en que parte del proceso de la conexión se presentó la problemática. Para realizar una búsqueda minuciosa del error es conveniente habilitar el archivo de "trace" (archivo de rastro), para identificar el error.

ANÁLISIS Y DISEÑO DEL SISTEMA

3

3.1 DEFINICION DEL PROBLEMA

A pesar de que el sistema actual cumple con las necesidades actuales del negocio es indispensable tener en cuenta el posible crecimiento del mismo en un futuro no muy lejano. Por tal motivo es de suma importancia poder prevenir el aumento en la demanda del servicio.

El sistema esta diseñado para que cualquier empresa o consorcio que se dedique a prestar servicios de soporte técnico lleve el control necesario del negocio, incluso a nivel Nacional

El sistema actual diseñado en Forms 3.0, con el tiempo y a manera vertiginosa, se esta convirtiendo en un sistema obsoleto que únicamente cubriría las necesidades de una empresa pequeña, es decir que opere a escala local en la ciudad de México, y cuyo equipo de cómputo se encontraría también rezagado en cuanto a avances tecnológicos.

Dado que la empresa en cuestión tiene en planes crecer dentro de un período muy corto. Y en función de esto esta el cubrir todas las expectativas de los clientes que se tienen aquí en la ciudad de México y los futuros clientes, que de acuerdo a la estrategia de crecimiento, se establecerá en algún punto geográfico de la República Mexicana en primera instancia.

Por lo anterior y debido a las necesidades del negocio es de suma importancia actualizar el equipo, teniendo como principal apoyo a los analistas que prestan el servicio y el equipo de cómputo que éstos últimos utilizan para desempeñar su trabajo.

Cuando nos referimos a actualizar el equipo, no únicamente nos referimos a las instalaciones de trabajo, y los requerimientos de los equipos necesarios. Es de suma importancia tener en cuenta los recursos humanos, ya que, debido a ellos y al servicio que dan, se presenta esta oportunidad de poder crecer (por lo tanto es importe capacitar a los analistas e incorporar a nuevos elementos con las características adecuadas). Lo anteriormente mencionado implica cubrir una gama más completa de productos a los que se les deberá dar servicio y por lo tanto equipos con las características necesarias para que los analistas se desarrollen de manera adecuada y para que el sistema en cuestión funcione correctamente y cubra la parte de control y administración de la información

Por tal motivo los requerimientos solicitados, ya que éstos contemplan en gran medida, lo mínimo requerido para el sistema y los posibles productos que cada analista deberá instalar en su correspondiente equipo para auxiliarse en cuanto algún cliente reporte un problema.

Otra de las características que implicó el desarrollo del sistema es el nivel de servicio que anteriormente se prestaba y que ahora es necesario cambiar, dadas las características de demanda. Es decir, el servicio anteriormente proporcionado sólo contemplaba servicio telefónico (conocido como hot-line) que cubría un horario de 9:00 de la mañana a las 18:00 horas. Pero dado las necesidades del cliente, es necesario renovar la forma de trabajo, y por lo tanto aumentar

Varios de los clientes con los cuales la empresa cuenta, se quejan del servicio. No por el nivel de los analistas, sino porque algunos problemas de nivel crítico se llegan a presentar después de las 18:00 horas, incluso en la madrugada. Y no saben como localizar a los analistas para solucionar su problema crítico en ese momento.

Otros han solicitado con gran insistencia que un analista fuera con ellos a resolver el problema o a realizar actividades previsoras, es decir antes de que se presenten los problemas (conocidas como actividades proactivas).

La figura 3.1.1 muestra la forma de trabajo actual.



Figura 3.1.1 Forma en la que se trabaja actualmente

3.2 PROPUESTA DE SOLUCION

Sobre la base de la problemática discutida en el punto 3.1, se considero la *importancia de renovar la manera en que se operará a partir del momento en que el sistema se ponga en producción.*

Basándonos en los incrementos en la demanda del servicio se han planteado básicamente cuatro niveles de servicio, los cuales podrán en cumplir las demandas que los clientes podrán solicitar en algún momento determinado. Estos niveles de servicio son los siguientes.

- a) Servicio Bronce.
- b) Servicio Plata.
- c) Servicio Oro
- d) Servicio Proactivo.

Los servicios anteriormente comentados cubren una variedad de servicios que se comentan de manera breve a continuación.

Servicio Bronce. Este tipo de servicio esta enfocado básicamente al tipo de servicio que anteriormente se estaba brindando. Es decir el servicio será únicamente vía telefónica brindándose de las 9:00 a.m. a las 18:00 horas. Se decidió seguir conservando este mismo esquema ya que la gran mayoría de los clientes hasta este momento están satisfechos con el servicio otorgado.

Servicio Plata. Este tipo de servicio surgió sobre la base de la necesidad de algunos clientes por tener la seguridad de poder comunicarse vía telefónica a Soporte Técnico, después de las 18:00 horas. El servicio de esta categoría se contempla en prestar el servicio vía telefónica después de las 18:00 horas hasta las 21.00 horas.

Servicio Oro. Este tipo de servicio fue creado basándose en los clientes que tiene aplicaciones de magnitud crítica y que por la importancia de sus operaciones y transacciones requieren un servicio las 24 horas del día. En este caso se tendría que establecer un horario de trabajo nocturno. Es decir que pueda brindar el servicio de las 21:00 horas a las 9:00 horas. Otra posible idea podría ser asignar a un analista con las características adecuadas y que un cliente determinado tenga los medios necesarios para localizar al analista en cuestión y que este pueda asesorarlo en cualquier momento.

Servicio Proactivo. Este tipo de servicio se creo con la idea de que un analista pueda asistir al lugar del cliente para resolverle sus problemáticas. Este tipo de servicio será proporcionado basándose en las horas de trabajo del analista, es decir, el cliente deberá haber adquirido un contrato con este tipo de servicio, donde se establezca la cantidad de tiempo contratado, y este tiempo

especificado será el que el analista estará otorgando los servicios al cliente en cuestión. A continuación se muestra una imagen donde se plasma la nueva forma de trabajar.

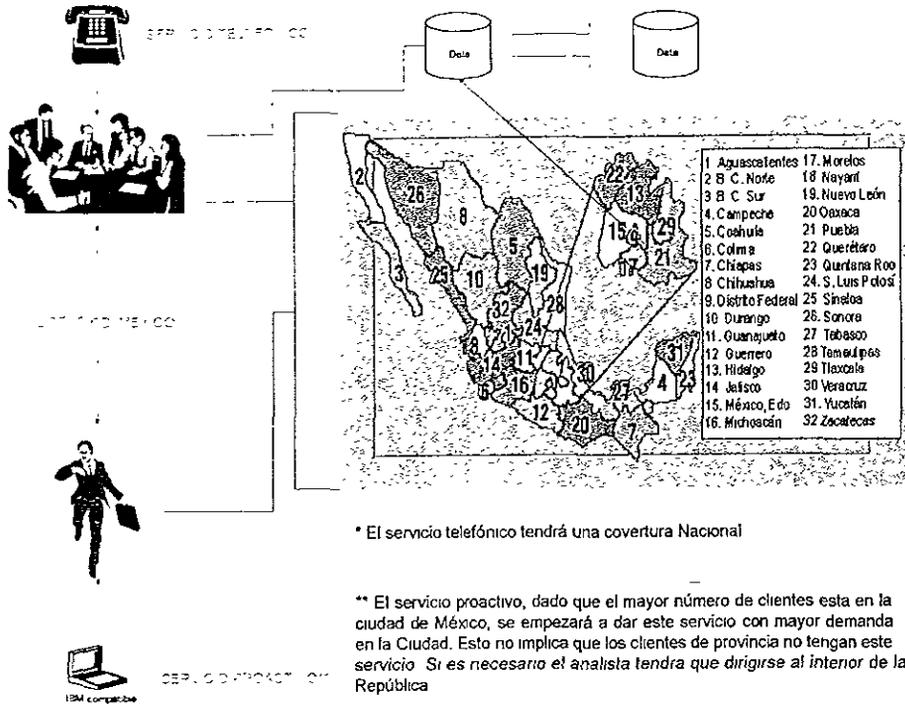


Figura 3.2.1 Nueva forma de trabajo, más eficiente y controlada

En función de lo anteriormente comentado se definió el siguiente objetivo:

Proporcionar al Departamento de Soporte Técnico un sistema de información capaz de registrar las actividades y funciones de los analistas, así como de manejar la estructura de trabajo que se comentó en líneas anteriores.

Además de poder realizar un registro ordenado de la información que los analistas deberán insertar, cubrirá otros puntos importantes como:

- Registro y consulta de las posibles soluciones que los analistas brinden a los clientes

- Poder asignar los reportes de los clientes a los analistas adecuados.
- Apoyo en la toma de decisiones de atención en las colas de reportes de los analistas
- El Proporcionar estadísticas de rendimiento a los coordinadores de los analistas para verificar su desempeño.
- El Proporcionar estadísticas en cuanto a tiempo de resolución de los reportes resueltos. Para que los propias analistas verifiquen su rendimiento.
- Un mejor servicio al cliente (indirectamente).
- La opción a los clientes de consultar vía internet la SAT(Solicitud de Asistencia Técnica), y de esta manera saber el progreso en la solución de su problema.

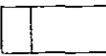
3.3 DIAGRAMA DE FLUJO DE DATOS (DFD)

Un diagrama de flujo de datos nos representara de forma gráfica el origen y destino de la información, y su paso a través de los procesos del sistema de información; es un modelo lógico del flujo de la información, es decir, no se especifican aspectos físicos de la implantación del sistema.

Existen principalmente dos tipos de notación para describir un DFD; la notación "Yourdon/DeMarco" y la notación "Gane/Sarson". Las diferencias principales entre ambas notaciones consisten en lo siguiente: en la notación Yourdon/DeMarco el proceso se representa mediante el uso de círculos, al contrario de la notación Gane/Sarson, en donde el proceso se representa mediante el uso de un rectángulo con las esquinas redondeadas. Por otra parte, la notación Yourdon/DeMarco utiliza líneas paralelas para almacenar los datos y Gane/Sarson utiliza un rectángulo cerrado por un extremo y abierto por el otro.

Este último tipo de notación (Gane/Sarson) es el que utilizaremos para la representación del DFD.

La notación es la siguiente:

	<p><i>Entidad externa</i>. Nos representa el origen o destino de la información y se identifica por medio de un nombre apropiado, pudiendo ser el de una persona, un lugar, organización, departamento, etc.</p>
	<p><i>Proceso</i>. Nos indica la transformación de datos de entrada en información de salida. Debido a esto, la información de entrada al proceso tendrá siempre un nombre diferente a la información de salida del mismo.</p>
	<p><i>Almacenamiento</i>. Indica el lugar en donde se realizará el almacenamiento de los datos. El lugar indicado aquí no hará referencia alguna al lugar físico de almacenamiento (cinta, diskette, etc.), simplemente indica un depósito de datos, el cual permite la adición y acceso de los datos.</p>
	<p><i>Flujo de datos</i>. Aquí se especifica la información en movimiento, es decir, la forma en la que los datos fluyen por el sistema de información. La punta de la flecha nos indicará el destino de los datos, y cada flecha se definirá con un nombre apropiado correspondiente al flujo de datos.</p>

La tabla mostrada a continuación nos muestra las diferencias principales entre los dos tipos de diagramas de flujo de datos que se consideraron para el modelado del diagrama de flujo de datos.

DESCRIPCIÓN	YOURDON - DE MARCO	GANE - SEARSON
<ul style="list-style-type: none"> Flujo de datos 		
<ul style="list-style-type: none"> Proceso de transformar los datos 		
<ul style="list-style-type: none"> Almacenamiento de datos 		
<ul style="list-style-type: none"> Origen externo o destino de los datos 		
<ul style="list-style-type: none"> Almacenamiento de los datos para mostrar múltiples veces en el diagrama 		
<ul style="list-style-type: none"> Origen externo o destino de los datos para mostrar múltiples veces en el diagrama 		

Tabla 3.3.1 Comparación entre las notaciones: Yourdon - De Marco y Gane - Searson

El siguiente DFD nos muestra de manera general la transformación y el flujo de los datos a través del sistema, el cual será definido posteriormente de una manera más detallada

DIAGRAMA DE FLUJO DE DATOS GENERAL

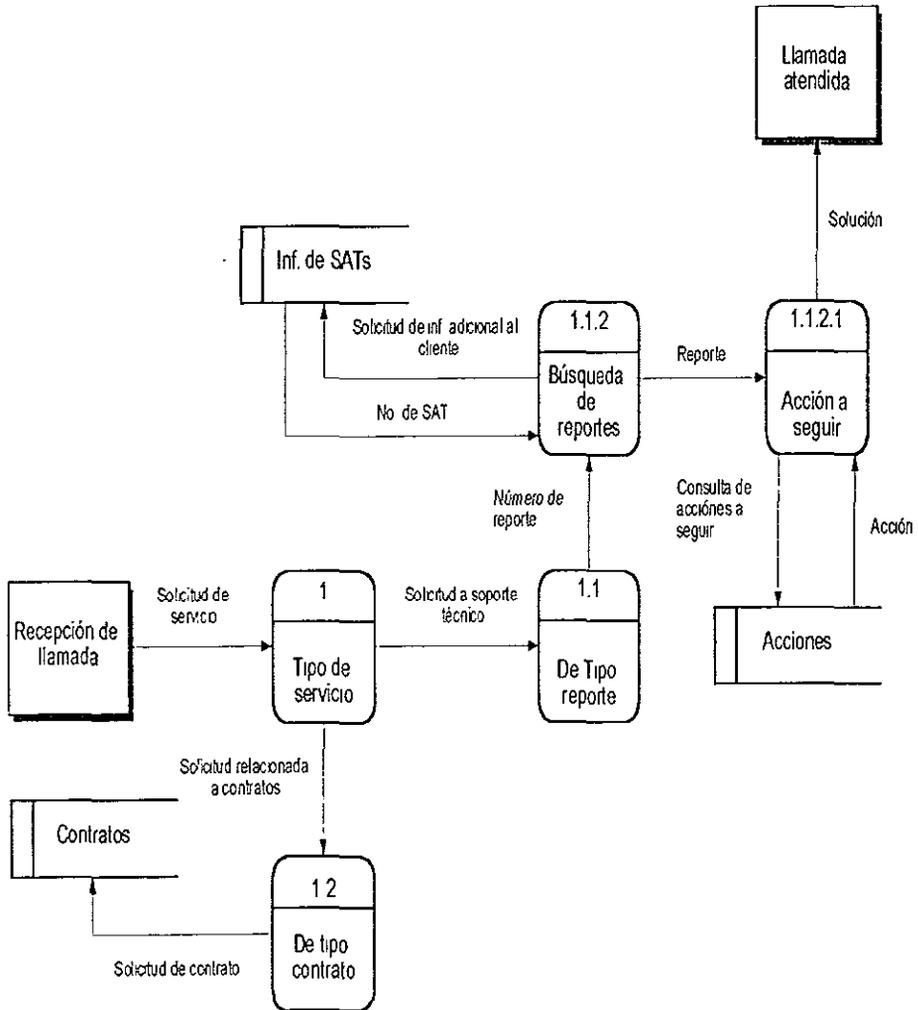


DIAGRAMA DE FLUJO DE DATOS GENERAL

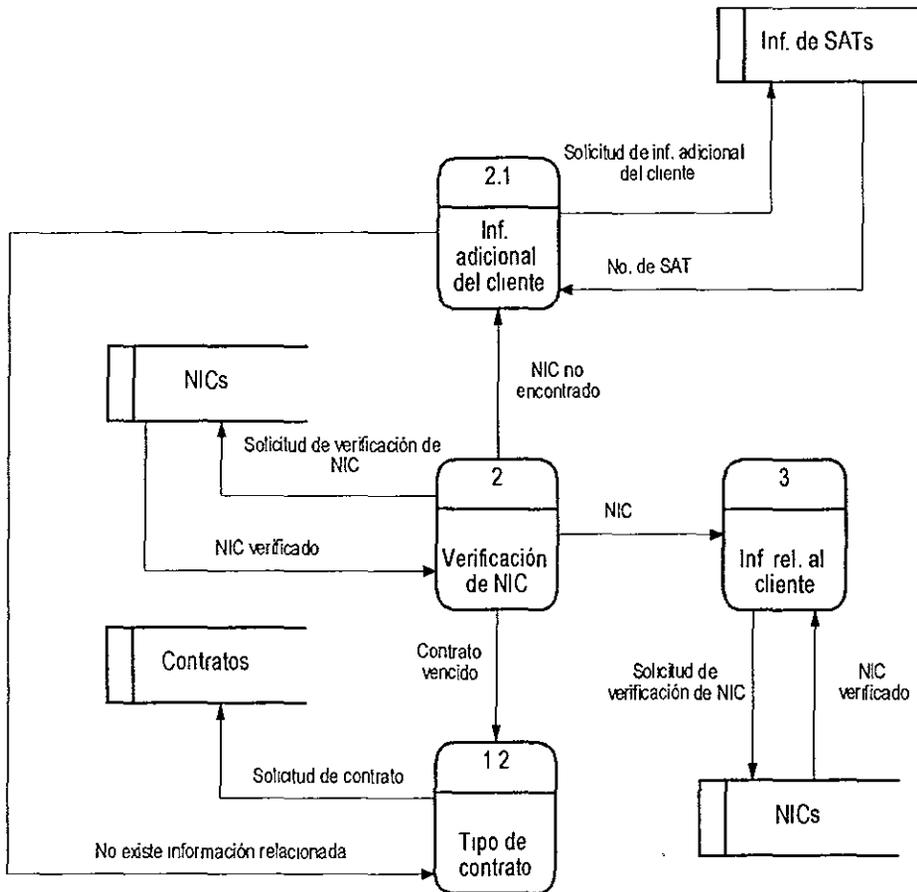


DIAGRAMA DE FLUJO DE DATOS GENERAL

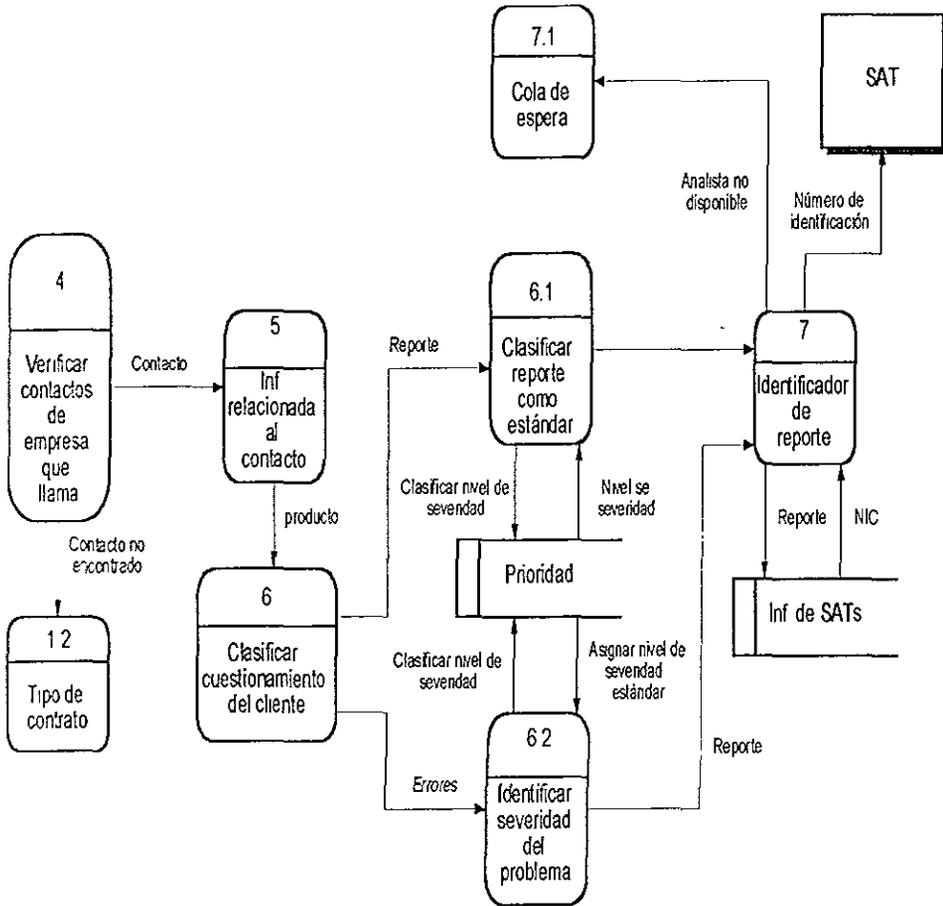


DIAGRAMA DE FLUJO DE DATOS DETALLADO(TIPO DE CONTRATO: CON SOPORTE TECNICO)

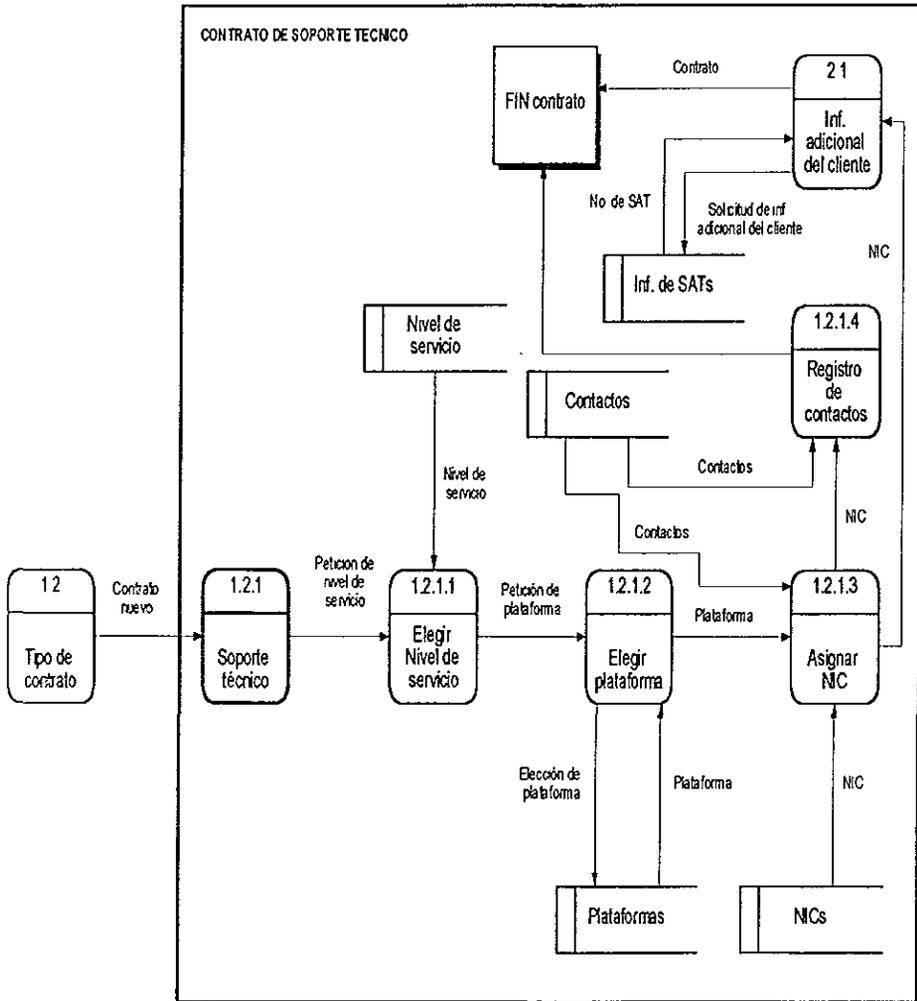


DIAGRAMA DE FLUJO DE DATOS DETALLADO(TIPO DE CONTRATO:
LICENCIA DE PRODUCTO CON SOPORTE TECNICO)

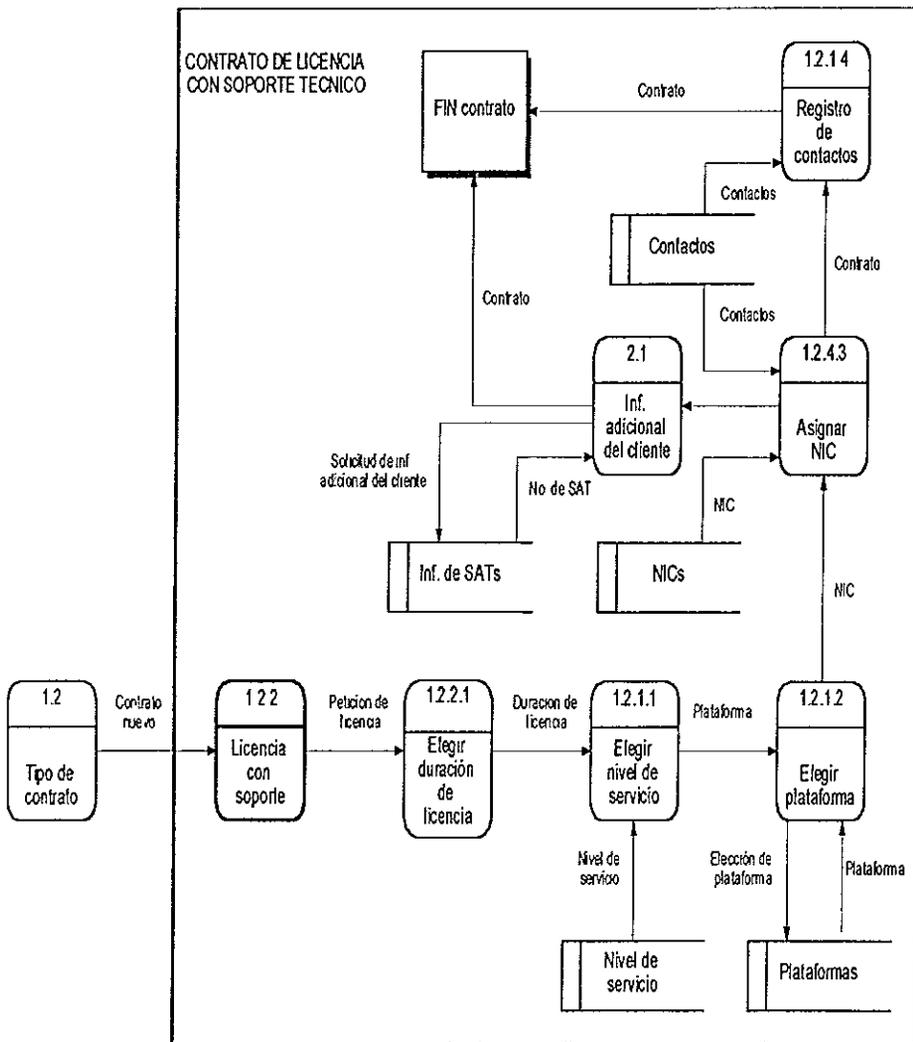


DIAGRAMA DE FLUJO DE DATOS DETALLADO(TIPO DE CONTRATO:
LICENCIA DE PRODUCTO)

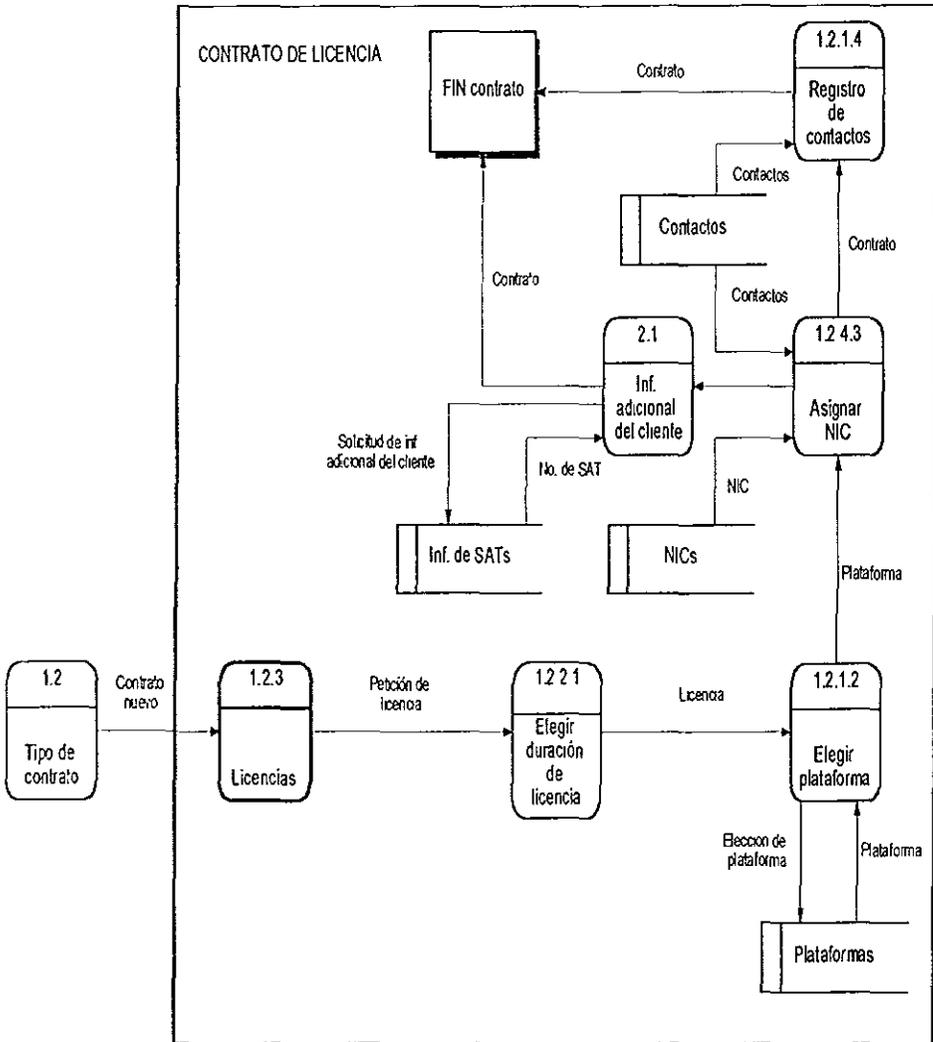


DIAGRAMA DE FLUJO DE DATOS DETALLADO (RENOVACION DE CONTRATO)

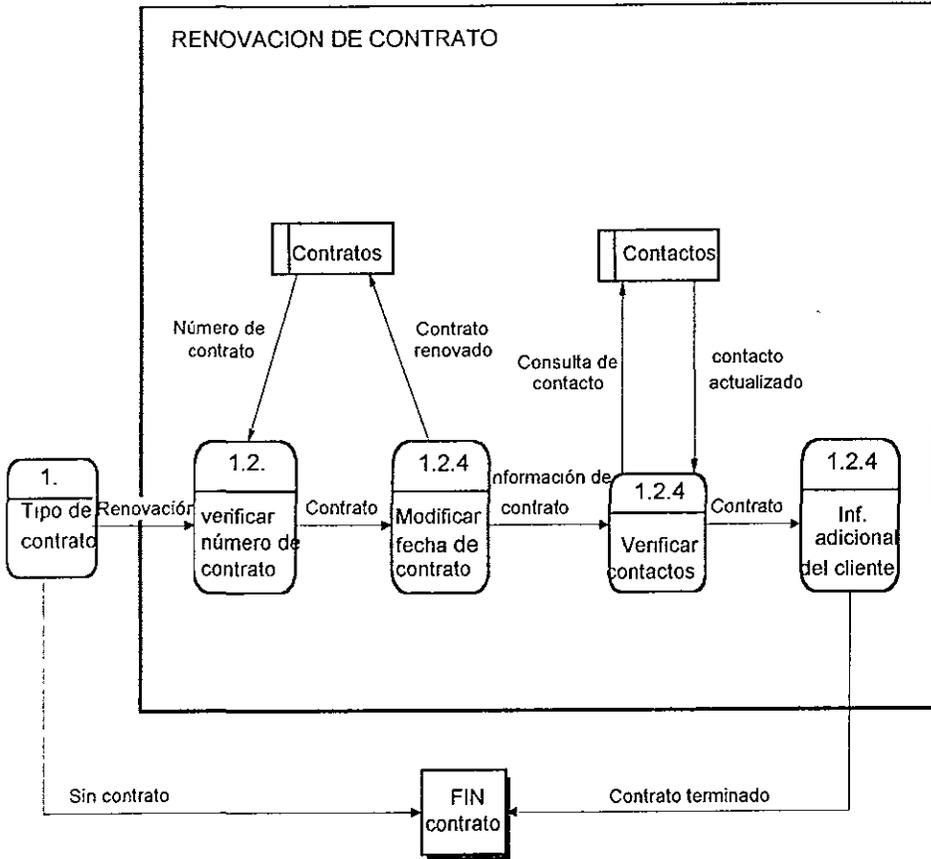


DIAGRAMA DE FLUJO DE DATOS DETALLADO(CLIENTE CONTACTADO)

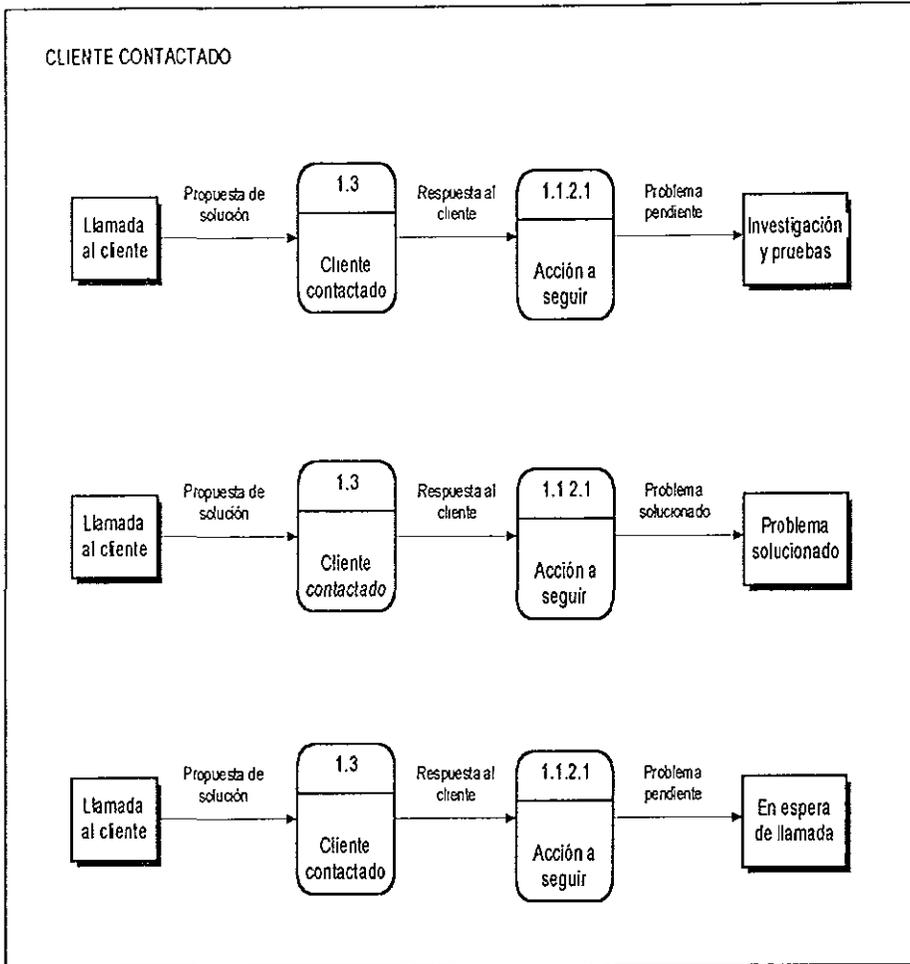
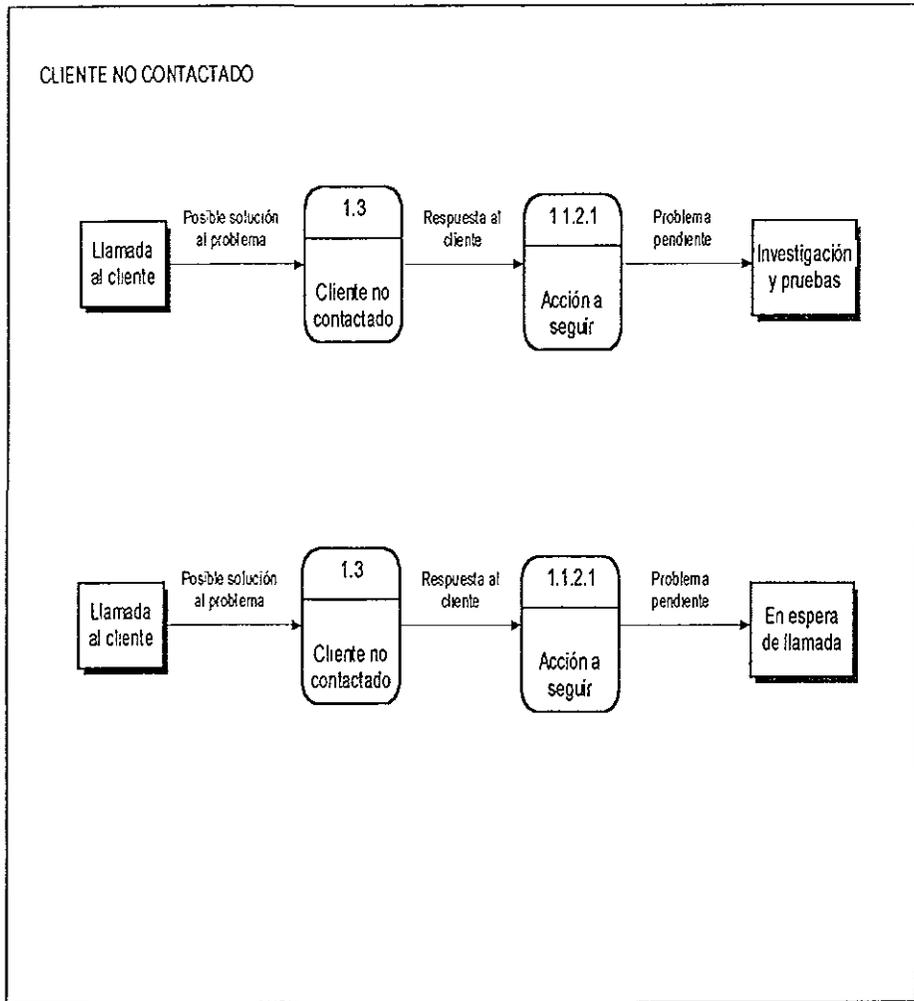


DIAGRAMA DE FLUJO DE DATOS DETALLADO(CLIENTE NO CONTACTADO)



3.4 PROTOTIPOS

Dentro del desarrollo de sistemas, una de las herramientas más importantes y útiles es la generación de prototipos, ya que en función de éstos, los usuarios finales pueden darse una idea sobre el estado final del sistema. Es importante tener en cuenta que la elaboración de prototipos se tiene que llevar a cabo desde las primeras etapas de desarrollo de la aplicación. Es uno de los métodos más completos para recopilar la opinión de los usuarios finales.

El objetivo principal de la elaboración de prototipos es observar las reacciones del usuario final, sugerencias por parte de los usuarios, fijar nuevas fechas de reunión para revisar las nuevas modificaciones sobre los prototipos inicialmente revisados.

En función de lo anteriormente comentado, se podrá observar que tan eficientemente se están cubriendo las necesidades que los usuarios finales inicialmente solicitaron. Durante la reunión con los usuarios finales es necesario recopilar las reacciones mediante formas de retroalimentación, con el fin de recoger los comentarios de todos los usuarios, a partir de esto definir si el sistema es del agrado de la mayoría de los usuarios y si por algún motivo se tendrán dificultades para implementar el sistema.

Cuando se le presenta un prototipo a un usuario final se deben de tomar en cuenta básicamente tres aspectos:

- a) Sugerencias del Usuario
- b) Innovaciones
- c) Planes de revisión

a) *Sugerencias del Usuario*

Todos los comentarios realizados por el usuario final son de gran importancia, dado que, sobre la base de ellos, se realizarán las refinaciones y cambios sobre el prototipo presentado. Para que los usuarios puedan dar una opinión más exacta de lo que requieren, es importante que los usuarios estén trabajando con el sistema durante un período de tiempo determinado. El cual varía en función de la complejidad del prototipo, así como del interés por parte de los usuarios.

b) *Innovaciones*

Con relación a las innovaciones, es importante señalar no se tenían contempladas antes de la interacción con el usuario. Son nuevas características que no forman parte de una modificación importante sobre el prototipo presentado.

c) Planes de revisión

Dado que los prototipos son una versión preliminar del producto finalmente terminado, los planes de revisión son útiles para establecer fechas y procedimientos donde se ven involucrados varios grupos de trabajo. En función de estos planes se determina sobre qué módulos es necesario generar un nuevo prototipo. En función de estos planes se determinan las prioridades a seguir y que camino es el menos costoso.

3.4.1 DESARROLLO DE UN PROTOTIPO

El desarrollo de un prototipo parte de los requerimientos inicialmente solicitados por el usuario, aunque la creación del primer prototipo cumpla con la mayoría de las necesidades finales, pero no todas ya que los usuarios finales no saben realmente lo que requieren hasta que ven algo tangible.

Por la propia naturaleza de los prototipos, éstos evolucionan constantemente y en su momento pueden requerir de una gran cantidad de revisiones.

Cuando se lleva a cabo el desarrollo de un prototipo se deben de considerar cuatro aspectos importantes. trabajar en módulos manejables, construir el prototipo rápidamente, modificar el prototipo en iteraciones sucesivas y enfatizar la interfaz del usuario. Estos cuatro aspectos cubren por si solos lineamientos específicos que deben de seguir una secuencia de aplicación para cubrir en su totalidad la etapa del desarrollo del prototipo.

Trabajando en módulos manejables

Este punto hace referencia a desarrollar un prototipo por cada módulo operacional del sistema. No tendría ningún sentido desarrollar un prototipo que engloba todas los requerimientos funcionales del sistema. Es decir, desarrollar prototipos sobre módulos que sean independientes, pero evidentemente dichos módulos deben de interactuar entre ellos.

Construcción rápida de prototipos

Esta actividad es de suma importancia ya que influye sobremanera en el tiempo que puede transcurrir desde la definición de requerimientos hasta la entrega del sistema completo

El poder construir un prototipo rápidamente le permite al usuario final interactuar en un período de tiempo pequeño con un prototipo que cubra de

manera parcial sus necesidades y en función de la respuesta del usuario poder hacerle adecuaciones finales al sistema para tratar de cubrir en su totalidad todas las necesidades del usuario

Es importante puntualizar que esta etapa del desarrollo el analista se encuentra en una de las etapas más importantes del desarrollo de sistemas. Ya que aún se puede recopilar información en función del comportamiento de usuarios al trabajar con el prototipo. Para poder recopilar la información sobre el funcionamiento del prototipo es necesario hacerlo mediante entrevistas, observaciones y encuestas. Pueden ser desarrolladas encuestas como la que se muestra a continuación para recopilar la información y punto de vista de cada usuario final.

FORMA DE EVALUACION DE PROTOTIPO					
Nombre del Observador:				Fecha:	
Nombre del Sistema o Proyecto:			Compañía o Ubicación:		
Nombre o número del programa:			Versión:		
	USUARIO 1	USUARIO 2	USUARIO 3	USUARIO 4	USUARIO 5
Nombre del usuario:					
Período observado:					
Reacciones del Usuario:					
Sugerencias del usuario:					
Innovaciones:					
Planes de revisión:					

Figura 3.4.1 Forma de evaluación de un Prototipo

Modificación del prototipo

La creación del prototipo inicial debe de contemplar la posibilidad real de ser modificado, esto implica que los módulos creados no deben de tener mucha interdependencia.

En base a lo anterior es importante tener en cuenta que el prototipo puede llegar a sufrir varias modificaciones, de tal forma que se vuelva iterativa la forma de trabajo, es decir, cada modificación o conjunto de modificaciones en base a los comentarios de los usuarios es necesario que dichas modificaciones al ser realizadas físicamente sobre el sistema, es imperativo que los usuarios finales realicen la revisión correspondiente.

Las modificaciones posteriormente realizadas a partir de las sugerencias y puntos de vista de los usuarios es necesario que también sean realizadas de la manera más rápida posible, aproximadamente entre dos y tres días.

Por otro lado es importante indicarle al usuario final que el forma parte de las personas que se deben de involucrar de manera importante para que el sistema a desarrollar tenga éxito ya que depende de él y de sus comentarios que, *realmente en el futuro el sistema puede cumplir todas sus expectativas y puede trabajar de manera adecuada.*

Enfatizando la Interfaz del usuario

En este punto lo que se debe de tomar en cuenta es que el sistema tenga la característica de poder interactuar con el usuario de manera adecuada. Es decir, que el usuario pueda trabajar fácilmente con el sistema y le haga más sencilla su forma de trabajar. Esto implica que con un mínimo de entrenamiento el usuario pueda trabajar con el sistema con el fin de que se adapte de manera rápida con el programa y no provocarle una sensación de complejidad con el fin de evitar que deje el sistema.

El papel del usuario dentro del desarrollo de prototipos

Como se comentó en el punto de modificación del prototipo, es muy importante que el usuario final pueda aportar sus comentarios e insatisfacciones con respecto al sistema y su funcionalidad. Por lo cual es necesario comentarle al principio del desarrollo que tendrá que interactuar de una gran cantidad de ocasiones con los analistas.

En este sentido se pueden observar tres puntos relevantes en los cuales los usuarios finales pueden participar.

1. Experimentar directamente con el prototipo
2. Responder adecuadamente al prototipo
3. Comentar abiertamente las correcciones necesarias sobre el prototipo

Experimentando directamente con el prototipo

En este punto el usuario tendrá la libertad de poder experimentar de manera abierta sobre el prototipo con el fin de poder colocar al usuario en una situación real de trabajo y de funcionalidad.

Por tal motivo es de suma importancia motivar al usuario con el fin de que pueda trabajar con el prototipo abiertamente. Es cierto que al entregar el sistema terminado, se acompañe de la documentación correspondiente para el uso adecuado del sistema, pero en este punto específicamente no se limitará en ningún sentido al usuario.

Otra de las actividades, que van implícitas, es que el analista puede observar de cerca el comportamiento del usuario ante el sistema, de tal forma que se puede llegar a dar cuenta de las posibles iteraciones que llegue a tener el usuario, o pretenda llegar a tener y el analista no las haya contemplado.

Respondiendo adecuadamente al prototipo

Este punto hacer referencia a que los usuarios puedan sentir la libertad para poder aportar información sin tener temor de ser reprimidos posteriormente. Para poder garantizar la aportación de los usuarios es conveniente realizar reuniones en las cuales no este presente ninguna influencia laboral con el fin de que se sientan con la confianza suficiente para poder responder.

Comentando abiertamente las correcciones necesarias sobre el prototipo

Finalmente se requiere que los usuarios aporten nuevas ideas y comentarios sobre el prototipo que sé esta evaluando, de tal forma que puedan ser agregadas o eliminadas algunas de las características hasta el momento observadas.

Uno de los aspectos relevantes en las evaluaciones de los prototipos es que, a pesar de que los usuarios aporten sugerencias es responsabilidad única del analista el poder evaluarlas y considerar en su momento si pueden ser tomadas en cuenta para la funcionalidad final del producto. Por otra parte los usuarios finales no deben aceptar algo que no los convenza dentro de la

funcionalidad del sistema siempre y cuando dicha funcionalidad no puede ser considerada, y si es así, presentar al usuario los motivos por lo cuales no pueden ser considerados.

3.4.2 APLICACIÓN MAESTRA DE INFORMACIÓN GENERAL DE SOPORTE TÉCNICO

El prototipo previo a ser evaluado por los usuarios finales fue generado con Designer/2000, en el capítulo 4 ahondaremos más acerca de esta herramienta de desarrollo. En este punto hablaremos específicamente sobre la generación de uno de los prototipos del sistema.

Los prototipos que pueden llegar a ser generados con Designer/2000 se construyeron con una herramienta llamada Diagramador de Módulos, en donde podemos especificar de manera clara y detallada, las características de las ventanas (formas a desarrollar) y los reportes.

La pantalla principal de Designer 2000 para invocar el diagramador de Módulos se presenta a continuación:

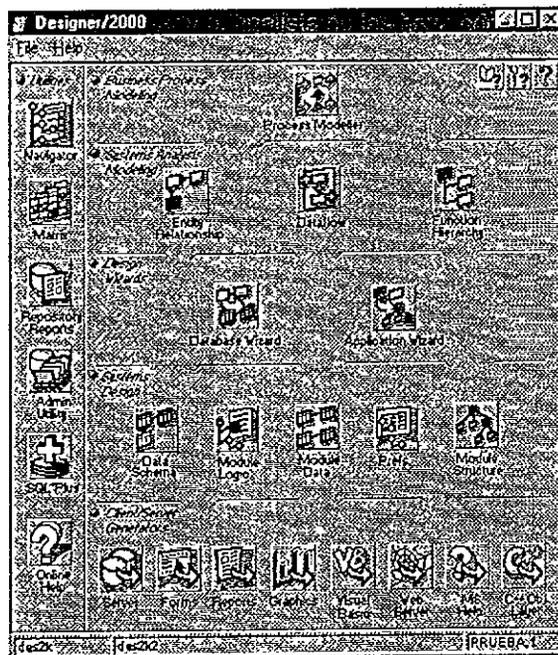


Figura 3.4.2 Pantalla principal de Designer 2000, que nos muestra al Diagramador de Módulos, herramienta que nos permite generar los prototipos de una manera sencilla y rápida.

Con relación a la pantalla para dar de alta la SAT (Solicitud de Asistencia Técnica) se presentó el siguiente módulo para generar el primer prototipo:

Figura 3.4.3 Pantalla para dar de alta una Solicitud de Asistencia Técnica

Dicho módulo cumplía con la mayor parte de los requerimientos inicialmente solicitados. Sin embargo, algunos usuarios coincidieron en eliminar campos como fecha de escalamiento, fecha de actualización y nivel.

Otra de las solicitudes, después de ver el prototipo, fue que en lugar de la barra de herramientas que se había colocado dentro de la pantalla, para salvar, realizar consultas y navegar entre los registros, se insertará una barra de herramientas con botones iconizados, en la parte superior de la ventana, de tal forma que dieran a entender por sí solos los botones, la operación que se pudiera llegar a realizar al presionar dicho botón.

Finalmente otra de las sugerencias que se comentaron fue que se agruparan los campos de captura para una rápida ubicación de los campos sobre la pantalla. Es decir un grupo de campos llamado "Información del Cliente", "Información del contacto", "Información técnica", etc. Y en cada agrupación se

concentraran aquellos campos que pudieran formar parte de la agrupación. Esto con el fin de que el usuario pudiera identificar o ubicar cada uno de los campos.

Basándose en estos últimos requerimientos solicitados se realizaron las modificaciones correspondientes y se generó la siguiente pantalla;

The screenshot shows a window titled "Nueva Sat" with the following fields and data:

- Analisis:** TESIS
- Fecha Actual:** 07-JUN-98
- Número de Sat:** (empty)
- TECIS:** TESIS
- NIG:** 2412
- Fecha de Vencimiento:** 31-JUL-97
- Fecha de Creación:** 07-JUN-98
- Información del Cliente:**
 - Clientes:** GRUPO SANTANDER MEXICO, S.A.
 - Dirección:** PASEO DE LA REFORMA 213 PISO 10
- Información del Contacto:**
 - Contacto:** TERESA SALGADO MUNOZ
 - Modem:** (empty)
 - Teléfono:** 629-5869
 - Fax:** 629-5873
- Información Técnica:**
 - Plataforma:** 2 HP 3000 Series 70000X
 - Producto:** 26 STARLAN
 - S.O.:** HP-UX 90
 - Paquetes Producto:** 356
 - DBMS:** 712
 - Medio:** 8 CU-ROM
- Problema del Encabezado:**
 - Cve. Error:** No Error
 - Descripción:** (empty)
 - oro:** 5678
- Detalle del Problema:**
 - Responsable:** MAHERNA
 - Sevis:** 25 Nueva SAT
 - Prioridad:** 3 Standard Fallo o D
- Comentario:** (empty text area)

At the bottom of the window, there is a footer: "La clave del Inventario de Soporte Técnico que se haya Error del reporte a donde sea: Contak-10" and "ContakLab".

Figura 3.4.4 Pantalla definitiva para dar de alta una SAT

Esta pantalla cumple con los requisitos establecidos, y fue el punto de partida en el cual nos basamos para seguir construyendo las demás pantallas

3.5 DIAGRAMA ENTIDAD-RELACION

Dentro del contexto del modelado encontramos dos tipos principales: el modelado de los datos; a través del Diagrama Entidad-Relación (DER), y el modelado del proceso, a través del Diagrama de Flujo de Datos (DFD). Ambas técnicas permiten al analista estudiar la estructura de la información del funcionamiento de una empresa, por ejemplo. Estos diagramas pueden ser capturados a detalle en papel en un "lenguaje" gráfico estándar que permita a analistas, usuarios, y programadores su comprensión.

Hablando específicamente del diagrama que atañe a este apartado (DER), y tomando en cuenta que existen diversas formas de representación; nos hemos basado en dos principalmente: la notación Chen y la notación Bachman.

Dentro de un diagrama Entidad-Relación existen 5 conceptos principales a considerar. las entidades, las instancias de las entidades, los atributos, las relaciones entre entidades y la cardinalidad de la relación.

Una *entidad* es cualquier objeto o evento, acerca del cual, se recolectan datos. Es real; una persona, un lugar o un objeto. Por ejemplo el nombre de un vendedor, de un estado o ciudad o de un producto en particular. También puede definir eventos o unidades de tiempo, como la venta de un producto, la compostura de un equipo, un día en particular de la semana, un mes, etc.

Otro importante concepto es el de la instancia de una entidad. En el DER observamos que existe una entidad denominada Cliente, la cual tiene varios atributos; como son: nombre, teléfono, colonia, ciudad, estado, etc. Estos atributos, así como la entidad Cliente tienen su instancia. Teléfonos de México, por ejemplo, es una instancia u ocurrencia de la entidad Cliente; esto es análogo a tener un registro dentro de un archivo. En la Tabla 3.5.1 se muestra un ejemplo de este concepto.

Tipo de Entidad	Cliente	Instancia de Cliente :	Teléfonos de México
Tipo de atributo :		Instancia del tipo de atributo :	
	Num_cliente		10
	Nombre		Teléfonos de México
	Teléfono		515-26-93
	Fax		515-40-15
	Alias		Telmex
	Calle_número		Jaguar No 10
	Colonia		Granjas
	CP		21739
	Estado		México D.F.

Tabla 3.5.1

Tomada por sí misma, una entidad (Cliente, por ejemplo) no contiene ningún dato. Los datos residen en los atributos de las entidades. Por lo tanto, Consideraremos a los

atributos como características o propiedades que describen a una entidad y describen lo que nosotros queremos almacenar en esta entidad. Un atributo es análogo a tener un campo dentro de un registro. A continuación tenemos un ejemplo, en donde se muestran los atributos que describen a la entidad Cliente.

Tipo de Entidad	Cliente
Atributos :	
	Num_cliente
	Nombre
	Teléfono
	Fax
	Alias
	Calle_número
	Colonia
	CP
	Estado

Observando el diagrama de la figura 3.5.1 notamos que las entidades están conectadas por medio de *relaciones*. Los CLIENTES "tienen" CONTRATOS, en donde "tienen" es la relación. Los CONTRATOS "llevan" LICENCIAS, donde "llevan" es la relación.

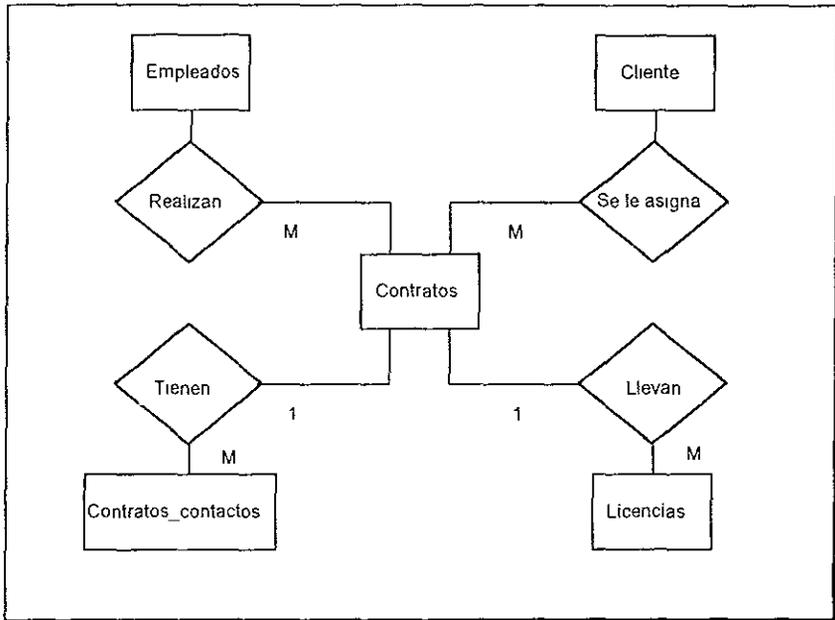


Figura 3.5.1 Relaciones entre entidades

Por último tenemos el concepto de cardinalidad, que no es otra cosa que el número de instancias que son permitidas entre dos relaciones. La información de la cardinalidad permite que del modelo de datos obtengamos información profunda y detallada acerca del funcionamiento de la empresa. En la figura 3.5.2 observamos las ligas de asociación entre entidades las cuales se denominan con un verbo, y su representación es la siguiente:

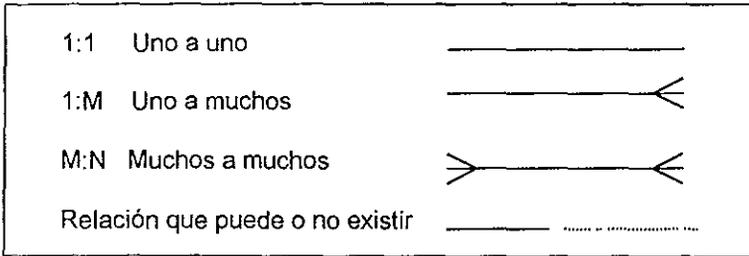
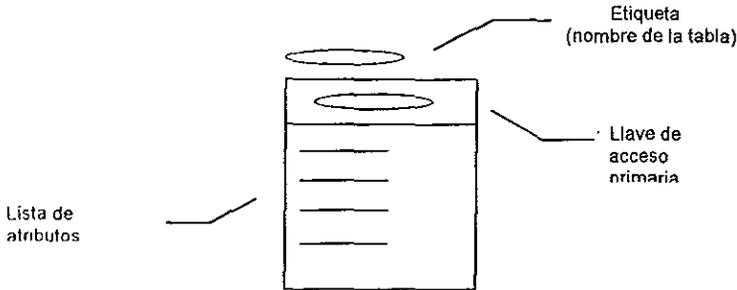


Figura 3.5.2. Representación de la cardinalidad en un Diagrama Entidad-Relación

La representación gráfica de los datos mediante un modelo entidad-relación será visto de la siguiente manera :



Las llaves de acceso secundarias serán identificadas por las letras FK encerradas entre paréntesis, éstas indicarán que se trata de una llave foránea (del inglés Foreign Key)

3.6 DICCIONARIO DE DATOS

El diccionario de datos definirá cada flujo de datos, o los datos almacenados que se encuentran en el diagrama de flujo, es decir, aquí encontraremos la información producida por el DFD. Esta información estará cuantificada por tipo, identificador, etiqueta (nombre del proceso, de la entidad, del almacenamiento del dato, o si se trata de un flujo de datos), una descripción narrativa de la acción que se está ejecutando. Todos estos datos nos permitirán tener un control total sobre la información que produce nuestro sistema, modificarlo si es necesario o detectar errores en el modelado del mismo.

ETIQUETA	IDENTIFICADOR	DESCRIPCIÓN	CONTENIDO
Recepción de llamada (EX)		Esta entidad hace referencia al hecho de cuando se reciben las llamadas.	Nombre de la persona que está llamando
Solicitud de servicio(FD)		La persona que realiza la llamada al área de soporte técnico indica la petición de servicio.	Nombre de la compañía de la cual está llamando.
Tipo de servicio(P)	1.1	En este punto el analista determina el tipo de ayuda que requiere la persona que está llamando.	Basicamente consiste en cuestionar si el servicio será atendido por el técnico directamente o será necesario contactar a personal del área de preventa.
Solicitud relacionada a contratos (FD)		Consiste en dejar en espera al cliente durante unos segundos en lo que se contacta a una persona del área e preventa.	Se le indica al personal de preventa el nombre del contacto y empresa de la cual está llamando.
De tipo de contrato (P)	1 2	Una vez que la persona de preventa esta en comunicación vía teléfono con el cliente se le pregunta sobre el tipo de contrato con el cual desea renovarlo o aclarar algún cuestionamiento.	En este punto el cliente proporciona si el cuestionamiento es en relación a un contrato sobre licencias del producto o sobre el servicio de soporte técnico.

Contratos (AD)		Almacenamiento de datos en el cual se encuentra la información relacionada con contratos.	El contenido puede ser el nombre del cliente, el nombre de la compañía, en número de contrato, número de NIC.
Solicitud de soporte técnico (FD)		En este punto el analista identifica que el problema del cliente es de tipo técnico.	El cliente proporciona información relacionada con su plataforma, producto y error que se le presenta.
De tipo soporte (P)	1.1	En este punto se determina si la llamada consiste en un nuevo problema o si el cliente está llamando por una SAT ya existente.	Comentario por parte del cliente, indicando que es un nuevo caso o en caso contrario el número de SAT por el cual está llamando y el nombre del analista que lo estaba atendiendo.
Número de reporte (FD)		Este caso es específico para la opción en la cual la persona que llama es por una SAT existente.	Número de la SAT.
Búsqueda de reportes (P)	1.1.2	En este punto el analista realiza la consulta sobre el SAT que el cliente le indicó esta siendo atendido.	Puede ser que el cliente no tenga el número de SAT. Con lo cual se realizaría la búsqueda en base al nombre de la persona que llama, el producto con el cual tiene el problema, etc.
Solicitud de información adicional al cliente (FD)		En base a la información proporcionada por la persona se realiza la búsqueda correspondiente.	Se manejan todos los datos posibles que el cliente proporcionó para buscar la correspondiente SAT.
No. de SAT (FD)		En función de la búsqueda realizada se obtiene el número de SAT.	En número de SAT por el cual el cliente está llamando.
Información de SATs (AD)		Unidad de almacenamiento que contiene las solicitudes de asistencia técnica.	Esta compuesto por toda la información que relaciona a las SATs como el tipo de problema, plataforma en la que se presenta, etc.

Reporte (FD)		Finalmente se tiene el reporte o número de SAT.	Lo importante en este punto es verificar el analista que estaba atendiendo la SAT.
Acción a seguir (P)	1.1.2.1	En base a la ubicación del analista que estaba atendiendo el reporte se determina la acción a seguir.	La acción puede variar, desde transferir la llamada al analista en cuestión o pararse a buscarlo.
Consulta de acciones a seguir (FD)		Puede ser que el analista que estaba atendiendo el caso dejó indicado la solución en la última acción con lo cual, el analista que en ese momento esta en contacto telefónico con el cliente puede atender.	El analista consulta la unidad de almacenamiento de Acciones con el fin de ayudar al cliente de primera instancia.
Accion (FD)		Información que se estrajo del almacenamiento de datos de acciones.	Puede ser la solución al problema o alguna alternativa, o unicamente indicarle al cliente que están en investigación en ese momento.
Acciones (AD)		Corresponde al almacenamiento de datos de acciones. Es decir toda actividad que involucre la solución de una SAT.	Esta almacenado todo tipo de información relacionada con la solución y seguimiento de las SATs.
Solución (FD)		Consiste en la solución plasmada por el analista que la estaba atendiendo o alguna alternativa de solución o simplemente el estatus de investigación.	El comentario del analista que está atendiendo esa llamada.
Llamada atendida (EX)		La conclusión de este primer ciclo de proceso de operación de la unidad de soporte técnico.	Puede ser desde una SAT cerrada hasta, una modificación del estatus del problema, incluso la severidad del mismo.

Verificación de NIC (P)	2	En el caso de tratarse de un nuevo reporte o SAT, se solicita al cliente el número de NIC para ser validado.	El número de NIC proporcionado por el cliente
Solicitud de verificación de NIC (FD)		Una vez teniendo el número de NIC, se realiza la consulta con la correspondiente clave.	Número de NIC.
NIC verificado (FD)		Es la información relacionanda con la verificación del NIC, es decir, si el servicio de soporte está vigente o no.	Notificación en relación a si el contrato del servicio está vigente.
NICs (AD)		Corresponde a la unidad de almacenamiento de datos donde se encuentran los NICs.	La información concerniente al NIC y el contrato de soporte relacionando.
Contrato Verificado (FD)		En caso de que el contrato no este vigente. Se transfiere la llamada a un empleado del área de preventa.	Información del analista en relación a que el contrato de soporte técnico del cliente en cuestión no está vigente.
Tipo de contrato (P)	1.2	Una vez que la persona de preventa esta en comunicación vía teléfono con el cliente se le pregunta sobre el tipo de contrato con el cual desea renovarlo o aclarar algún cuestionamiento.	En este punto el cliente proporciona si el cuestionamiento es en relación a un contrato sobre licencias del producto o sobre el servicio de soporte técnico.
Solicitud de contrato (FD)		Se realiza la búsqueda del contrato del servicio de soporte correspondiente.	Número de contrato con el cual se está presentando el incidente.
NIC (FD)		Corresponde al identificador único del equipo del cliente.	Consiste básicamente de un número no mayor de 6 dígitos.
Información relacionada al cliente (P)	3	Se verifican algunos datos relacionados con el cliente, para cerciorarnos que en realidad el cliente es quien está llamando	El tipo de información puede ser como su dirección, teléfono, plataforma, etc.

Solicitud de verificación de NIC (FD)		Se realiza la petición en base al NIC para consultar información adicional como por ejemplo la dirección.	Esta compuesta por el número de NIC en este momento.
NIC verificado (FD)		Se realizó la verificación correspondiente del NIC, de tal forma que se pueden consultar algún tipo de información adicional del cliente.	Esta compuesto por un conjunto de información adicional como la dirección, teléfono, número de fax, etc
NIC no encontrado (FD)		En este caso el número de NIC no fue localizado con tal motivo se solicita información adicional con el fin de localizarlo.	No existe dato.
Información Adicional del cliente (P)	2.1	Se le comenta al cliente que no se localiza el número de NIC. Que proporcione información adicional para consultar más a detalle.	Se le solicita por ejemplo, el nombre de la persona que llama, plataforma que maneja, etc.
Solicitud de información adicional al cliente (FD)		El base a la información recaudada, se realiza una búsqueda en SATs probablemente atendidas con anterioridad.	La consulta está compuesta de información que el cliente proporcionó adicionalmente.
No de SAT (FD)		Finalmente se localiza algún número de SAT anteriormente localizado para extraer de allí el número de NIC y proceder a crear un nuevo SAT o transferir la llamada con el personal de preventa.	Número de SAT anteriormente atendido.
Información de SATs		Unidad de almacenamiento en la cual están localizados todos los reportes o SATs anteriores.	Todo el tipo de información concerniente a las SATs como la plataforma en la cual se presentó el problema, producto, versiones, errores, etc

NIC (FD)		Consiste del número previamente verificado.	NIC.
Verificar contactos de la empresa que llama (P)	4	En base al NIC se despliegan los contactos reportados para ese número de NIC	Nombres de las personas que pueden hacer uso del servicio de soporte técnico.
Contacto (FD)		Consiste del nombre de la persona que está llamando y solicitando el servicio.	Nombre de la persona.
Contacto no encontrado (FD)		La persona que está pidiendo el servicio no se encuentra registrada por tal motivo es necesario transferir la llamada al personal de preventa.	Transferencia de la llamada del cliente a preventa indicando su nombre y el número de NIC con el cual desea pedir el servicio
Información relacionada al contacto (P)	5	Información relacionada a la persona que esta llamando.	El tipo de información puede ser ,su dirección laboral, teléfono, fax, etc.
Producto (FD)		Consiste del producto en el cual se esta presentando el problema.	Nombre del producto en el cual se está presentando el problema.
Clasificar cuestionamiento del cliente (P)	6	El cliente explica todo lo relacionado a su problema. Como reproducirlo. Realizando un análisis previo del problema.	El cliente explica a detalle su problema técnico indicando errores y problemas que se presentan.
Reporte (FD)		Consiste en parte de la información que generará la SAT.	Información como el nombre del cliente, contacto, dirección, teléfono, etc
Errores (FD)		Consiste de los errores reportados por el cliente.	Dichos errores pueden variar dependiendo del producto que se está utilizando.
Clasificar reporte como estándar (P)	6.1	En este proceso normalmente, no es regla, pero los errores de desarrollo generalmente son de severidad 3.	Este tipo de errores no detiene ningún proceso significativo de desarrollo o producción con lo cual se le asigna el nivel de severidad 3.

Identificar severidad del problema (P)	6.2	En este punto el problema puede ser de severidad 2 ó 1 con lo cual requiere un mayor detenimiento y estudio del problema.	En este caso el problema reportado causa que algún proceso de desarrollo o producción se detenga, con lo cual la severidad puede ser 2 ó 1.
Clasificar nivel de severidad (FD)		Clasifica el nivel de severidad, es decir 1, 2 ó 3.	Consiste del dígito correspondiente.
Nivel de severidad (FD)		Corresponde a la descripción del nivel de severidad 3.	Consiste del dígito 3.
Asignar nivel de severidad estándar (FD)		Corresponde en asignar el correspondiente valor estandar de severidad. En esta caso para la severidad 1 y 2.	Corresponde de la descripción correspondiente de la severidad 1 y 2.
Reporte (FD)		Consiste de la información más detallada y proporcionada por el cliente.	En este nivel tendríamos además la severidad y tipo de errores reportados.
Identificador de reporte (P)	7	Este proceso se encarga de asignar el número de SAT correspondiente.	Número compuesto no por más de 6 dígitos.
Reporte (FD)		Toda la información recopilada se almacenará en el almacenamiento de datos de SATs.	En este nivel tendríamos además la severidad y tipo de errores reportados
No. SAT(FD)		Consiste del número identificador del caso reportado.	Se asigna número correspondiente al reporte o SAT.
Número de identificación (FD)		El flujo de información también hace referencia al número de reporte pero proporcionado al cliente.	Corresponde nuevamente al número obtenido pero ahora proporcionándose al cliente.
Analista no disponible (FD)		En algunos casos el reporte no puede ser atendido porque no se encuentra en ese momento algún analista desocupado.	Se proporciona el nombre del analista que atenderá el problema.

Cola de espera (P)	7.1	Se asigna a una cola de espera determinada en función del analista asignado.	Conjunto de reportes que cada analista tiene a su cargo.
SAT (EX)		Da por entendido el problema reportado y con el número de identificador asignado.	Conjunto de 6 dígitos.

Nota Para identificar el tipo de elemento que se trata en la representación del DFD, se han asignado valores en relación a la siguiente lista con el fin de distinguir al elemento y su ubicación en el DFD.

- AD Almacenamiento de Datos
- EX Entidad eXterna
- FD Flujo de Datos
- P Proceso

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

DESARROLLO DEL SISTEMA

4

4.1 HERRAMIENTAS DE DESARROLLO

Para el desarrollo de las aplicaciones en nuestro sistema hemos elegido dos de las herramientas más comerciales y de mayor difusión en el ámbito empresarial con las que cuenta la firma Oracle: Oracle Designer/2000 y Oracle Developer/2000. Designer/2000 soporta el diseño de complejos sistemas, análisis, y diagramadores de diseño. Developer/2000 por su parte brinda a las organizaciones la capacidad de construir rápida y productivamente sistemas sofisticados que escalan de grupo de trabajo a empresa.

Se han elegido estas herramientas por contar con algunas características especiales:

- Repositorio común
- Soporte de metodología y diseño flexible
- Entorno de desarrollo de servidor y cliente unificado
- Arquitectura portable abierta

Sistemas Visuales y Modelaje de Diseño

Mediante la utilización de los diagramadores de modelaje de función e información podemos construir modelos que capturen las necesidades de los usuarios mediante el registro de estructuras lógicas y físicas requeridas en los sistemas de depósito de datos y transaccionales.

Repositorio Abierto

Podemos Integrar los repositorios y herramientas que no son de Oracle con el entorno Designer/2000. Utilizando la Interfaz de Programador de Aplicaciones (API) abierta de Designer/2000 para integrar herramientas y aplicaciones de terceros

4.1.1 GENERALIDADES DE DESIGNER 2000

Modelaje de Procesos

Podemos Visualizar y mejorar radicalmente los procesos fundamentales de negocios con las capacidades de modelaje de proceso de Designer/2000. Obteniendo una considerable ventaja competitiva, una reducción de costos y la mejora en la calidad mediante la comprensión de dependencias entre organizaciones y duraciones de ciclo de proceso. Tenemos la opción de aplicar iconos para identificar pasos de proceso e interacciones, podemos también ingresar parámetros de costo y tiempo para cada paso de proceso.

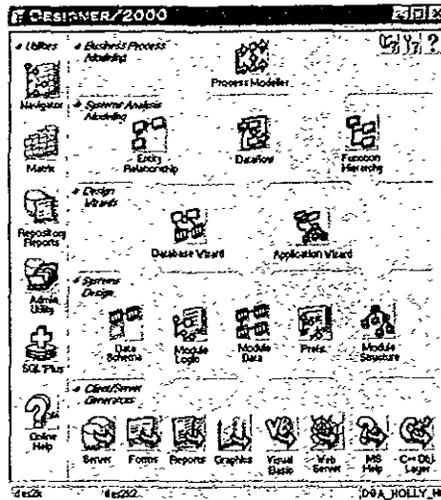


Figura 4.1.1 Pantalla principal de DESIGNER 2000

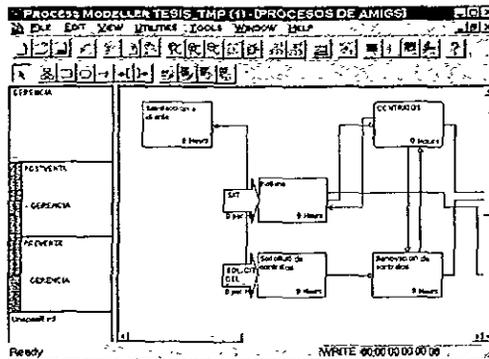


Figura 4.1.2 Pantalla principal del modelador de procesos

Con la utilización de la animación de procesos y multimedios podemos revelar problemas en la administración y brindar oportunidades a los usuarios a medida que observan toda la ejecución de pasos de proceso de principio a fin en un entorno de escala en el tiempo

La claridad y facilidad de uso de las herramientas de modelaje de sistemas combinadas con las sofisticadas instalaciones de navegación, aseguran un sólido cimiento para el diseño e implementación de soluciones de depósito de datos y aplicaciones.

Diseño de Sistemas

Utilizando Designer/2000 podemos implementar los sistemas distribuidos de cliente/servidor y las aplicaciones de depósito. Tomando input directamente de los desarrolladores y utilizando Wizards de Diseño podemos derivar rápidamente diseños de sistemas de las definiciones de requerimientos. La estructura de datos, y los diseños de programas de Developer/2000 y Visual Basic, se materializan rápidamente y con precisión.

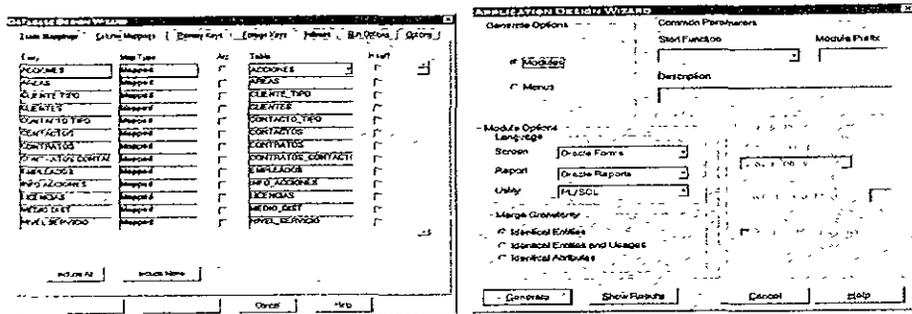


Figura 4.1.5 Wizards de diseño

Con la combinación de diagramadores y navegadores podemos crear y manejar detalles de tablas, claves foráneas y reglas de validación de datos; y definir triggers basados en servidores junto con las condiciones, funciones y procedimientos de ejecución (triggering). Designer/2000 facilita el particionamiento de código de cliente/servidor. También podemos controlar el diagramador de estructura de módulo para definir la estructura de las aplicaciones, sus módulos de programa de componente y las interacciones entre los mismos. Aplicando el navegador de lógica de módulos podemos de una manera gráfica e inteligente construir y modificar las definiciones de procedimientos dirigida por sintaxis. El diagramador de datos de módulo define fácilmente las interacciones entre programas individuales y los datos a los que acceden con el uso de ventanas, páginas y canvases. Podemos modificar características de acceso de datos de módulos mediante la definición de operaciones autorizables de selección, inserción, actualización y supresión y el agregado de especializaciones específicas de módulo de diseños generales.

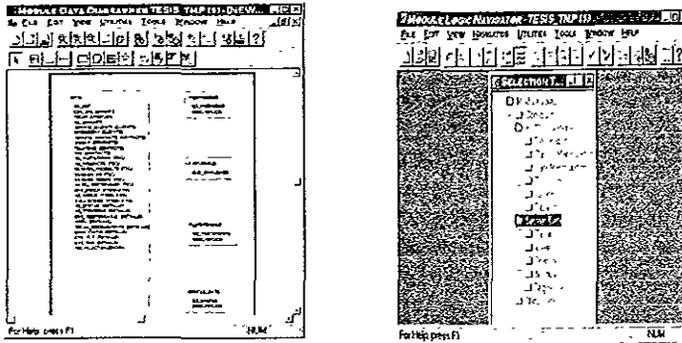


Figura 4.1.6 Diagramador de datos y navegador de lógica de módulos

Utilizando el navegador de preferencias y plantillas se pueden definir las características generales visuales y semánticas de los sistemas generados que elimina tiempo dedicado a pintar pantallas y define una presentación standard de todas las aplicaciones.

Designer/2000 provee las herramientas necesarias para diseñar e implementar rápidamente sistemas de cliente/servidor y aplicaciones de depósito y para responder a las cambiantes necesidades de negocios.

Generación de Aplicación de Cliente

Designer/2000 nos permite construir automáticamente aplicaciones Developer/2000. Las aplicaciones generadas incluyen configuraciones de pantallas múltiples, informes sofisticados, navegación a través de menús, botones y listas desplegadas, lógica de aplicación del lado del cliente, y funcionalidad total de acceso a la base de datos; todo con origen en los diseños de tabla, módulo de programa y preferencia almacenados en el repositorio de Designer/2000. Las plantillas dirigen la apariencia de una gran variedad de controles de GUI en pantallas generadas, tales como imágenes, diálogos, listas desplegadas, casilleros de control, botones de iconos y otros. Designer/2000 genera aplicaciones que no sólo corren en múltiples plataformas con presentación nativa completa, sino que también explotan las funciones nativas de cada plataforma tales como controles OLE2.

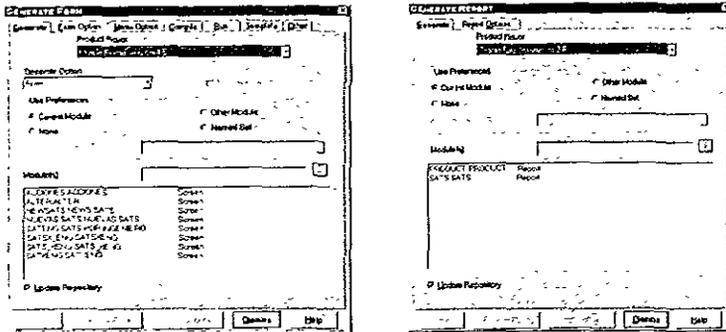


Figura 4.1.7 Generación de formas y reportes

Generación de Servidor

Designer/2000 genera no sólo aplicaciones del lado del cliente sino que también completa la entrega de aplicaciones empresariales y soluciones de depósito con la creación de todos los componentes del lado del servidor. ANSI SQL DDL puede crearse desde el diseño de servidor mantenido por repositorio, incluyendo las estructuras de tabla y columna, la validación de lado de servidor de restricciones de clave foránea, condiciones de control y claves primarias.

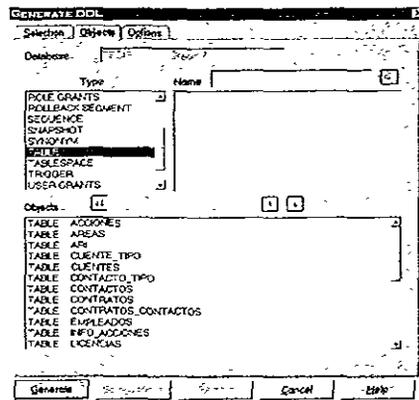


Figura 4.1.8 Generación de archivos DLL del lado del servidor

4.1.2 ORACLE DEVELOPER/2000 CARACTERISTICAS CLAVES

Developer/2000 además de ser un producto nativo con Oracle presenta una *característica muy importante*; su orientación al 100 por ciento hacia el manejo de las bases de datos. Además de ser una herramienta de eficiente operatividad, lo cual reduce el tiempo de programación de manera considerable.

Developer /2000 puede interactuar no sólo con el manejador de Oracle, también acepta otros manejadores comerciales entre los cuales tenemos Informix, Paradox, Sybase, DB2, entre otros. Esta conexión se puede establecer a partir de un producto denominado Oracle Client Adapter, mejor conocido como Gateway, el cual permite establecer como backend cualquier manejador que no sea Oracle, teniendo como frontend a Developer/2000.

Otra de las grandes ventajas de Developer/2000 es su gran portabilidad, ya que las aplicaciones realizadas tienen la misma funcionalidad en una gran cantidad de plataformas tanto en ambiente desktop (Windows 95, Windows NT y Windows 3.1) como en UNIX (HP-UX, Digital DEC, SCO UNIX, SUN, etc.). Además de que tiene implícito el enfoque orientado a objetos, lo cual facilita en gran medida el tiempo de programación, lo que provoca una respuesta más rápida en las aplicaciones.

Una de las últimas ventajas de Developer/2000 es su facilidad de poder interactuar con el ambiente WEB, es decir, las aplicaciones desarrolladas en Developer/2000 pueden ser implementadas en el WEB mediante la utilización de una herramienta también de Oracle conocida como *Web Server*.

Una de las características más sobresalientes de Developer/2000 es su enorme compatibilidad y su seguridad de acceso a la información. Si por alguna razón no se deseara trabajar con Developer/2000 para el desarrollo del frontend, el manejador de datos de Oracle tiene la facilidad de poder trabajar con herramientas de terceros, tales como Visual Basic, Acces, Visual Fox Pro, C++, entre otros. Para poder trabajar con este tipo de plataforma desktop es necesario hacer uso del ODBC de Oracle, ya sea para 16 ó 32 bits, dependiendo de la plataforma de desktop a utilizar. Además se cuenta con un acceso restringido a informes, consultas y bibliotecas.

Además Developer/2000 cuenta con algunas otras características destacables:

Capacidades de Procedimiento

Lenguaje de procedimiento integrado, PL/SQL
Administración de biblioteca para reutilización de PL/SQL
Extensiones a través de programas 3GL:

C
C++
COBOL
FORTRAN

Control Programático

Crear y modificar objetos de aplicación, incluyendo:

Cuadros
Consultas

Modificar atributos de cuadros, incluyendo:

Tipo de cuadro
Triggers de formato
Formatos de etiqueta
Fuentes de etiqueta
Título
Ubicación de leyenda

Crear o duplicar cualquier objeto gráfico, incluyendo:

Imágenes
Rectángulos
Arcos
Líneas
Polígonos
Grupos

Cambiar atributos gráficos de objetos tales como:

Color
Diseño
Tamaño
Posición
Rotación

Prestaciones de Administrador de Traducción

Traducir Toda Aplicación Developer/2000
Soporta Forms, Informes, y Gráficos
Carga transparentemente los archivos de recursos
Separa texto a traducir de otra información
Fusiona traducciones fácilmente

Editor Fácil de Utilizar

Editor de ventana dividida

Independiente de formato de archivo de recurso
Browser para navegación por strings
Información de contexto

Poderosa Administración de Proyecto

Organiza tareas en proyectos y módulos
Controla una gran cantidad de aplicaciones
Define roles de usuarios para seguridad
Busca control de versión
Procesamiento de lote

Glosario

Soporta creación de glosarios definidos por usuario
Provee sugerencias durante traducción

Soporte de Archivo ASCII

Extraer texto traducibles de archivos de texto no formateados
Delimitadores de strings definidos por usuario

Prestaciones de Conectividad de Base de Datos, soporte SQL

Sentencia centrales de manejo de ODBC:

Seleccionar (con soporte para Order By, Group By, y Having)

Insertar

Actualizar

Suprimir

Sentencias de definición central de Datos ODBC:

Crear Tabla

Crear índice

Crear Vistas

Alterar Tabla

Eliminar Tabla

Eliminar índice

Eliminar Vistas

Otorgar

Revocar

Procesamiento de transacción:

Ejecutar

Restablecer

Acceso a Datos

Identificadores de caso superior/inferior/mixto
Acceso de sólo lectura y lectura-escritura

Soporte PL/SQL

Modos de Soporte Developer/2000

Clave Primaria - Actualizable y No Actualizable

Bloqueo - Demorado

Tipos de datos soportados por PL/SQL

BINARY INTEGER, NUMBER, CHAR y VARCHAR2, VARCHAR, LONG,
RAW, LONG RAW, BOOLEAN, DATE

Sentencias soportadas por PL/SQL

DECLARE, BEGIN, END, COMMIT, WORK, ROLLBACK, OPEN,
CURSOR, FETCH, CLOSE, GOTO, SELECT INTO, UPDATE, INSERT,
DELETE, NULL

Funciones

SUM, AVG, MAX, MIN, COUNT, ABS, FLOOR, SIGN, SQRT, ROUND,
ASCII, LOWER, UPPER, LTRIM, RTRIM, SOUNDIX

4.2 JUSTIFICACION DEL SISTEMA

A pesar de que este punto es tocado en el capítulo de antecedentes es importante tener en cuenta cuál es la justificación o conjunto de justificaciones que dieron origen a la elaboración de la tesis.

La justificación esta fundamentada básicamente en 4 puntos de gran importancia:

- a) Modificación del ambiente de trabajo del usuario. Anteriormente era ambiente en modo carácter, en el cual, el acceso al sistema era sólo a través del teclado, lo cual limitaba en gran manera al usuario. Por tal motivo y por los avances tecnológicos a nivel computación y, además, por tener un sistema amigable completamente para el usuario se decidió cambiar a un ambiente tipo GUI (Graphics User Interface).
- b) Utilizar un método con el cual se garantice la organización de la información de manera consistente. Permitiendo a futuro el crecimiento de cualquier empresa a escala nacional. Sin tener limitaciones geográficamente hablando.
- c) Utilizar un método de análisis que determine la problemática actual de negocio, que permita desarrollar un sistema en función con las necesidades actuales del mismo a partir de las problemáticas actuales, así como los factores de impacto y los nuevos objetivos a nivel negocio que se desean alcanzar (motivo por el cual se decidió utilizar *Designer/2000*, que es la herramienta CASE de Oracle para el desarrollo del sistema).
- d) Una justificación más sería la de utilizar un sistema que contemplara el cambio de siglo, para que el manejo de fechas no tuviera mayor complicación. Esta es una de las principales justificaciones por las cuales se ha seleccionado en manejador de base de datos de Oracle.
- e) Sobre la base de los objetivos especificados y como uno de los objetivos principales, tenemos el de desarrollar una interfaz vía WEB que permita a los clientes consultar las SATs de manera remota y sin tener la necesidad de consultar vía teléfono al analista técnico. Además de tener la facilidad de que los cliente con soporte tipo oro pueden consultar las SAT's durante las 24 horas del día.

Estas son las principales circunstancias por las cuales el desarrollo de la tesis. Es importante enfatizar que El RDBMS de Oracle no sólo esta casado con

sus productos. Oracle, además de ser una de las empresas líderes como proveedora de soluciones, es completamente portable, es decir, puede funcionar en una gran variedad de plataformas sin que el performance disminuya.

Otra de las importantes causas por las que se decidió utilizar el manejador de Oracle es que permite trabajar de varias formas que le den gran seguridad y consistencia a la información. Tal es el caso de Parallel Server y Replication

Por otra parte Oracle permite trabajar con varios servidores de información, evitando con esto trabajar de manera centralizada en un futuro. Estas características permiten evitar posibles pérdidas de información por, por errores humanos, fenómenos naturales o por problemas de hardware o software.

Por el momento se está proponiendo que el RDBMS este montado sobre un servidor Windows NT. Pero esta sugerencia no establece que la información permanezca de manera definitiva. El sistema esta diseñado tomando en cuenta que la capacidad de la empresa se encuentra limitada.

Si la empresa en cuestión tiene la visión de crecer a futuro, el sistema se lo permitirá. Además de poder migrar su información a un servidor UNIX si sus necesidades así lo requieren. Simplemente consiste en cambiar la información del Servidor Windows NT que se encuentre en ese momento hacia el servidor UNIX, sobre el cual el cliente desee migrar su información.

4.3 REQUERIMIENTOS DEL SISTEMA

El AMIGS (Aplicación Maestra de Información General de Soporte) es una aplicación que puede ser instalada de forma independiente en cualquier consorcio, empresa, organización o similar que tenga la capacidad de otorgar el servicio de soporte técnico vía telefónica.

La estructura de AMIGS esta compuesta básicamente de dos módulos, el primero de ellos es el relacionado con el área de Soporte Técnico, en lo relacionado a el levantamiento, seguimiento (modificación) y finalización de las *consultas técnicas de los clientes*. La segunda de ellas es la que se encarga básicamente el área de Ventas de Soporte Técnico, cuya función principal es la de registrar que empresas tienen contrato de Soporte Técnico, o renovación de contrato, los empleados de dicha empresa que estarán dados de alta en la AMIGS como contactos y quienes, de manera directa, podrán solicitar el servicio.

Para el caso particular de la AMIGS, para poder ejecutar la aplicación en cada cliente es necesario utilizar una serie de programas, nativos al desarrollo de la aplicación, como "Runtime Forms 4.5" (4792 Kb aproximadamente), "Runtime Reports 2.5" (8034 Kb aproximadamente), que en su totalidad ocupan aproximadamente 12 MB. Por tratarse de una aplicación gráfica es necesario un procesador 486 ó superior a 100 Mhz y monitor VGA, así como 32 Mb en memoria RAM y espacio libre en disco duro de 100 Mb.

Teniendo en cuenta que el corazón del sistema se centraliza en el servidor, es decir, toda la información se encontrará almacenada en el servidor NT, es necesario que tanto el servidor como el cliente cuenten con un programa con el cual se pueda establecer la *comunicación desde los clientes hacia el servidor*. En nuestro caso se utilizará el programa llamado SQL*NET v 2.3.X, el cual es utilizado para verificar que el protocolo de comunicación que se esta utilizado, dentro del ambiente cliente/servidor. Utilizaremos el protocolo TCP/IP para la comunicación, por tal motivo es necesario instalar el SQL*NET Client (5846 Kb aproximadamente). Al realizar esta instalación automáticamente se instala también Oracle TCP/IP Adapter 2.3.X. Para poder garantizar la comunicación es necesario que el proveedor del protocolo TCP/IP este soportado por las herramientas de ORACLE, ya que puede presentarse el caso en que se utilice un proveedor no soportado y la capa Transporte del standard ISO probablemente no opere de la manera adecuada. Para el caso de los clientes que este utilizando Windows 3 X el proveedor a utilizar es el Padway (Wongollong) (ya que con pruebas anteriormente realizadas, se ha comprobado la eficiencia de este proveedor) y para el caso de Windows 95 es necesario utilizar el protocolo TCP/IP proporcionado por Microsoft, ya que cualquier otro proveedor no esta soportado.

En cuanto a los requerimientos por parte del servidor, se propone que sea una máquina con procesador Pentium a 200 Mhz, con memoria RAM de 64 Mb y monitor VGA. El RDBMS ocupa aproximadamente 80 Mb en disco duro, el sistema operativo de red (Windows NT ocupa 100 Mb), más SQL*NET para el protocolo de comunicación (5846 Kb aproximadamente). Por lo tanto estamos hablando de aproximadamente 200 Mb, sin embargo, por ser el servidor seguramente se le instalarán nuevas aplicaciones las cuales pueden ser útiles para el monitoreo o rastreo de la base de datos, así como del sistema operativo.

Sobre la base de las necesidades del área de soporte técnico que ofrezcan este tipo de servicios, los requerimientos para software y hardware se presentarán a continuación de manera resumida. Dichos requerimientos se especificarán tanto para el servidor (donde se encontrará el corazón del sistema RDBMS) así como de los clientes (desde donde los analistas podrán acceder la información en el servidor).

CLIENTE-SERVIDOR
Cliente :
Plataforma: Windows 95
S.O. : Windows 95
PC 386 ó superior
Monitor VGA
Disco duro de 500 MB
16 MB en RAM
Forms Runtime V 4.5.X
SQLNET V 2.X preferentemente 2.3
SQLPLUS V 3.X
Servidor :
Plataforma: NT 3.51 ó superior
RDBMS: Oracle 7.3.X o superior
Espacio disponible en disco duro de 370 MB p/RDBMS
SQLNET V 2.0.1.5.0.3
Oracle ODBC Driver 1.11.0.2
Hardware: Procesador Pentium a 100 Mhz
Disco Duro de 2 GB
Tarjeta de red para UTP
64 MB de memoria RAM

Tabla 4.3.1 Características de hardware y software, tanto para el cliente como para el servidor.

4.4 PROGRAMACION DEL SISTEMA

Utilizando el método CASE como herramienta de desarrollo se puede invertir un período de tiempo significativo en el proceso de programación, ya que parte de ésta es desarrollada por la herramienta. En el caso de Designer 2000, son utilizados dos módulos para la generación de la herramienta. El primero de estos módulos se llama Diagramador de Módulos y Datos (Module Data Diagrammer), con el cual se pueden desarrollar todas las interfaces necesarias para interactuar con el usuario final.

Dentro de los tipos de interfaces desarrolladas por el Diagramador de Módulos y Datos se encuentran básicamente dos tipos de módulos; formas y reportes. Las formas son las pantallas que se presentarán al usuario con el fin de poder realizar operaciones de inserción, actualización y eliminación de información. Los reportes son los módulos que permiten únicamente el despliegue de información en un determinado formato de presentación.

El segundo de los módulos utilizados para la generación de la codificación es el módulo denominado Estructurador de Módulos (Module Structure Diagrammer), básicamente consiste en desarrollar y especificar la forma en que fluirá la información durante la ejecución de la aplicación.

Dentro de la pantalla principal de Designer 2000 se pueden observar los dos iconos correspondientes a los módulos de Designer para la generación de los módulos del sistema.

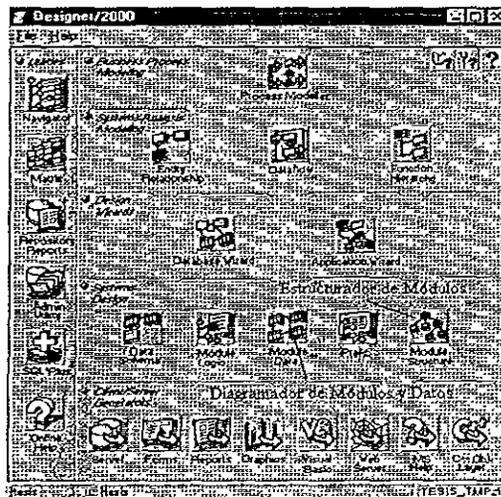


Figura 4.4.1 Pantalla principal de DESIGNER 2000, mostrando las herramientas principales de programación(Estructurador de módulos y Diagramador de módulos y datos).

Para ejemplificar la generación de la aplicación se hará referencia a la forma de captura de Solicitudes de Asistencia Técnica. En este caso el nombre del módulo a generar lo llamaremos new_sats.

La forma de generar un módulo tipo forma se lleva a cabo de la siguiente manera. Invocar el Diagramador de Módulos y Datos cuya pantalla principal de este generador se presenta a continuación:

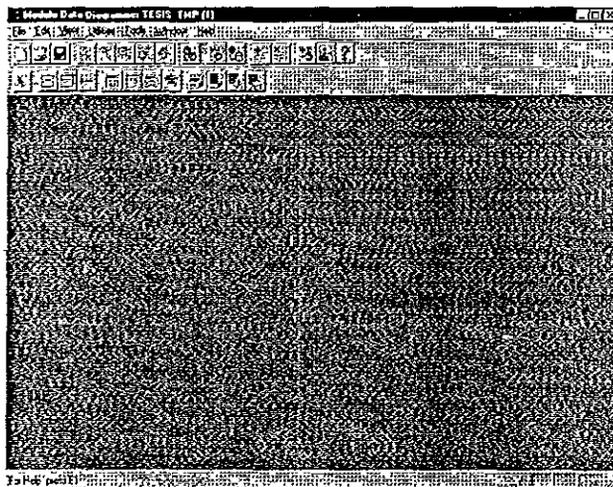


Figura 4.4.2 Pantalla principal del Diagramador de módulos y datos

Para crear un nuevo diagrama se puede indicar directamente sobre el menú de File o presionando el icono de NEW, correspondiente a la hoja en blanco de la barra de herramientas.

Posteriormente nos desplegaría una ventana donde debemos indicar las características básicas del módulo a generar. La información básica a indicar consiste de, el nombre corto del módulo; el nombre del módulo; propósito; lenguaje, es decir si será generado para Oracle Forms, Oracle Reports o Oracle Graphics; el tipo de módulo a generar que puede ser Screen (pantalla), Report (reporte) o Chart (Grafica); como dato opcional se puede especificar si se desea crear el nuevo módulo como copia de otro ya existente. La pantalla que se presenta es la siguiente:

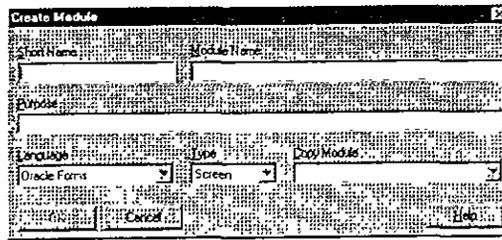


Figura 4.4.3 Pantalla que nos pide las características básicas del módulo a generar

La información a insertar sería como la que se presenta a continuación:

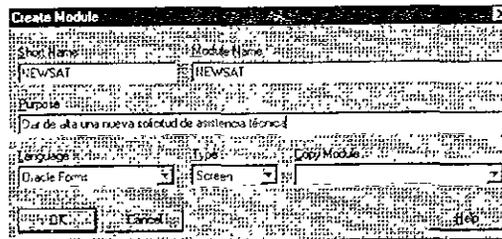


Figura 4.4.4 Ejemplo que nos muestra la forma de introducir los datos necesarios para generar un módulo

Se insertaría la información correspondiente y se presionaría el botón OK. De esta forma se crearía un nuevo módulo y se presentaría en el monitor como se muestra en la siguiente imagen:

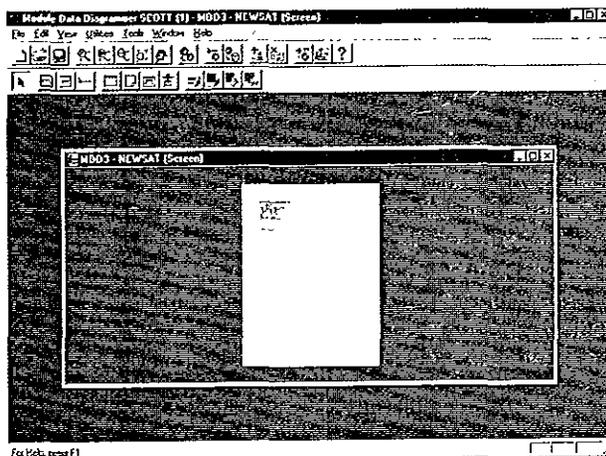


Figura 4.4.5 Módulo generado

Si maximizáramos la ventana del módulo creado y tratáramos de visualizar el recuadro que se encuentra en la hoja, se observaría la información concerniente al módulo. Si realizáramos este par de acciones observaríamos la siguiente ventana:

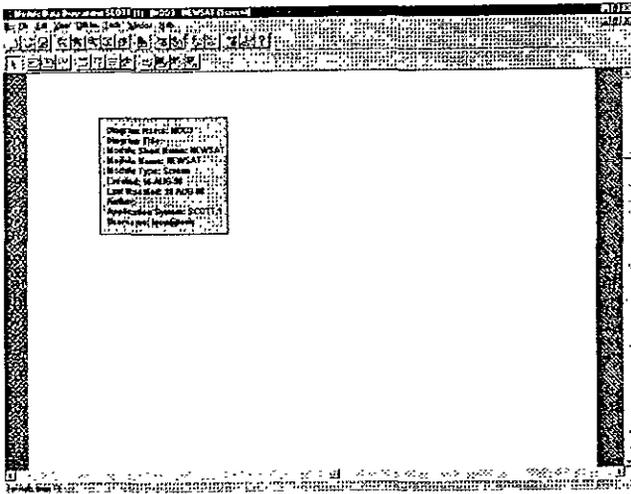


Figura 4.4.6 Vista maximizada del módulo generado

Posteriormente de haber creado el módulo consiste se deben especificar los usos de tablas sobre los cuales se realizará la generación de la forma. En el caso del módulo new_sats, se hará uso de las tablas Sats, Prioridades, Plataformas, Empleados, Status_sats, Medio_dist y Productos. Estos usos de tablas gráficamente estarán representados por recuadros con el nombre de la tabla en la parte superior del recuadro.

Ya que dichos usos de tablas fueron especificados en el módulo, a cada uno de los usos de tablas sería necesario establecer los usos de columnas para cada uso de tabla. Correspondería un uso de columna a cada una de las columnas de las tablas que serán involucradas en este módulo. Estos usos de columnas estarían representados gráficamente por el nombre de la columna dentro del recuadro del uso de tabla correspondiente.

Como existen relaciones entre las tablas, estas relaciones serán necesarias especificarlas también en la generación del módulo de tal forma que cuando se trabaje directamente sobre la interfaz, la integridad de la información con respecto a las tablas se cumpla y no exista ningún problema en la información.

Aparte de la funcionalidad que se le desea proporcionar a la pantalla se pueden especificar algunos detalles como los siguientes.

- El tipo de operaciones que se van a poder realizar sobre la interfaz. Operaciones como inserciones, actualizaciones, consultas y eliminaciones.
- La etiqueta que queramos sea desplegada para cada campo, así como la operación sobre cada campo.
- De que manera deseamos sean distribuidos los campos, si van a pertenecer a un grupo específico de campos de tal manera que visualmente se encuentren enmarcados y con un título del grupo en común
- Se puede determinar también si al posicionarlos en un campo específico aparezca una leyenda al final en la barra de status de la ventana que indique el objetivo básico del campo
- Indicar el tamaño de la ventana que conformará el módulo.
- El tamaño de los campos.
- El tipo de dato que se va a manipular.
- Indicar alguna condición específica al consultar la información de la base de datos.
- Si deseamos ejecutar algún proceso específico al posicionarnos en algún campo determinado

Una vez que todas las características del módulo fueron especificadas, el siguiente paso consiste en generar el módulo. Esta operación nos generará los correspondientes ejecutables y archivos fuente de la forma que se generará. En el caso del ejemplo que se está presentando se generaran un par de archivos, el *newsat.fmb* y el *newsat.fmx*.

El primero de ellos corresponde al archivo fuente, que es el que en su momento se podría llegar a modificar si se quisiera agregar algún tipo de funcionalidad específica. Estas modificaciones se realizarían en Forms Designer.

El segundo de ellos corresponde al archivo ejecutable que es el archivo que formaría parte de la aplicación final.

Para poder realizar la generación del módulo se pueden realizar de las dos formas Presionando la opción de "Generate as..." del menú Utilerías, o presionando el icono de generación, representado por un engrane.



Al presionar dicha opción de generación se presentará la pantalla que se muestra a continuación:

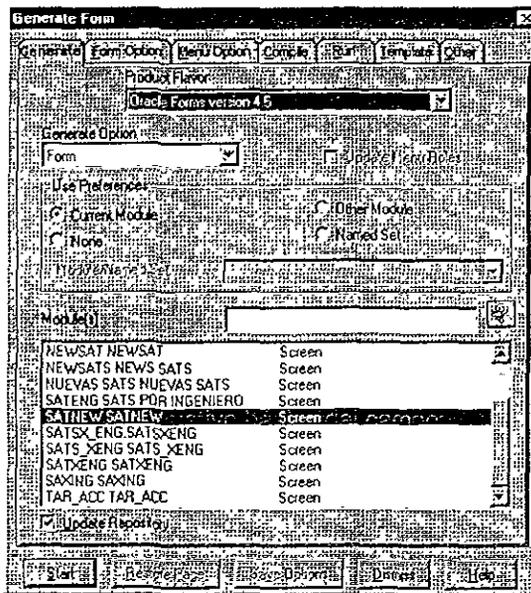


Figura 4.4.9 Generación de módulo

En esta ventana se especifica algunas características de generación como las que se comentan a continuación:

- Módulo que se generará.
- Ruta en la cual se desea se coloquen los archivos generados.
- La cadena de conexión hacia la base de datos.
- Comandos necesarios para poder realizar la generación.
- Especificación de algún módulo tipo menú que se anexará a la forma.
- Se puede indicar si dicho módulo generado se desea correr automáticamente para poder observar la presentación final al usuario.
- El template que será utilizado para generar la forma.
- Se puede especificar si se desea genera algún tipo de archivo como reporte para registrar todos los pasos de la generación.

Una vez que se generaron los archivos es necesario revisar la presentación de los módulos ya que en algunos casos es necesario reubicar los campos para mejor ubicación para el usuario final.

Es importante comentar que en algunas ocasiones es necesario que el desarrollador le proporcione algún tipo de funcionalidad adicional al módulo generado por la herramienta ya que dicha funcionalidad no se puede conseguir con Designer 2000.

Al realizar la generación la herramienta automáticamente inserta la codificación necesaria en función de lo que se haya indicado en el módulo y los tipos de constraints correspondientes a cada tabla y columna y que en este momento, al realizar la generación de los módulos es necesario que dichas tablas se encuentren físicamente en la base de datos

Un ejemplo de la codificación que genera la herramienta se presenta a continuación:

```
1 /* CGGN$KEY_CLRFRM */
2 /* Perform the key's standard functionality */
3 BEGIN
4 clear_form;
5 IF :SYSTEM.FORM_STATUS = 'CHANGED' THEN
6 RAISE FORM_TRIGGER_FAILURE,
7 END IF;
8 END;
9
10 /* CGCF$PERFORM_STARTUP */
11 /*Execute the WHEN-NEW-FORM-INSTANCE code that was created by Forms 12*/
12 /* Generator */
13 BEGIN
14 BEGIN
15 CG$WHEN_NEW_FORM_INSTANCE;
16 EXCEPTION
17 WHEN OTHERS THEN
18 CGTE$OTHER_EXCEPTIONS;
19 END;
20 END;
```

Esta codificación la creo en caso de que se presionará la opción de limpiar la forma, es decir, realizar la acción para poder dar de alta una nueva Solicitud de Asistencia Técnica. Como se puede observar primero se limpia la forma y con esta acción se coloca el cursor en el primer campo de la forma (4)¹. Si el status de la forma se encontrará en status de "CAMBIO" entonces se dispararía un error (5,6)¹ Posteriormente se cierra ese bloque de codificación.

En el segundo bloque de codificación se llama a ejecutar la unidad de programa llamada "CG\$WHEN_NEW_FORM_INSTANCE" (16)¹. En caso de que se presente alguna problemática se dispararía una excepción llamada "WHEN

¹ Nota: corresponde a la numeración especificada en la codificación.

OTHERS THEN" (18)¹, en dicho caso se llamaría a ejecutar la unidad de programa "CGTE\$OTHER_EXCEPTIONS" (19)¹.

Otro ejemplo referente a la codificación es la unidad de programa llamada "CG\$WHEN_NEW_FORM_INSTANCE" que es invocada por el primer ejemplo presentado. Dicha codificación se indica a continuación.

```

1      /* CG$WHEN_NEW_FORM_INSTANCE */
2      PROCEDURE CG$WHEN_NEW_FORM_INSTANCE IS
3      BEGIN
4      /* CGGN$GET_DATE_AND_USER */
5      /* Get the current user and date; store in items for general use */
6      BEGIN
7          DECLARE
8              CURSOR C IS
9                  SELECT SYSDATE, USER
10                 FROM SYS DUAL;
11      BEGIN
12          OPEN C;
13          FETCH C
14          INTO :CG$CTRL.CG$$SYSDATE, :CG$CTRL.CG$$USER;
15          IF C%NOTFOUND THEN
16              message('Internal Error. No row in table SYS.DUAL');
17              RAISE FORM_TRIGGER_FAILURE;
18          END IF;
19          CLOSE C;
20      EXCEPTION
21          WHEN OTHERS THEN
22              CGTE$OTHER_EXCEPTIONS;
23      END;
24      END;
25
26      /* CGLY$COPY_USER_DATE */
27      /* Values from USER and DATE items on page 0 in the control */
28      /* block will be "fired" to all other USER and DATE items */
29      /* on other pages. */
30      BEGIN
31          :CG$CTRL.CG$$US := :CG$$USER;
32          :CG$CTRL.CG$$DT := :CG$$SYSDATE;
33      END;
34
35      /* CGLY$INIT_CANVASES */
36      /* Call procedure to ensure correct canvases are visible */
37      BEGIN
38          CGLY$CANVAS_MANAGEMENT;
39      END;
40      END;

```

Primeramente se declara el nombre del procedimiento (2)². Se declara la creación de un cursor llamado "c", es decir, es un almacenamiento temporal de memoria donde se guardará temporalmente la información extraída en la base de

² Idem

datos por la sentencia indicada en el cursor. En este caso "Select sysdate, user from sys.dual" (8,9,10)². Posteriormente se abre el cursor declarado anteriormente (12)². Esto se realiza utilizando la sentencia "OPEN".y el nombre del cursor. Después se extrae la información de dicho cursor con la sentencia "FETCH" (13)². La información extraída se asignará a un par de campos correspondientes al bloque de control Dichos campos son "CG\$CTRL.CGU\$SYSDATE" y "CG\$CTRL.CGU\$USER" y que se encuentran en la página 0. En la primera de ellas se guardará la fecha del sistema y en la segunda de ellas el nombre del usuario que esta accediendo la base de datos (14)².

En caso de que el cursor no haya extraído la información se desplegará un mensaje de error (15,16,17)². Posteriormente se cierra el cursor utilizando la cláusula "CLOSE" y el nombre del cursor (19)². En caso de que se disparará algún tipo de error que no fuera el "NOTFOUND" se dispararía la excepción "WHEN OTHERS". Siendo este el caso se ejecutaría la unidad de programa "CGTE\$OTHER_EXCEPTIONS" (20,21,22)².

Posteriormente se realiza una copia, tanto del usuario como de la fecha del sistema que en ese momento se encuentran en los campos "CG\$CTRL.CGU\$SYSDATE" y "CG\$CTRL.CGU\$USER" del bloque de control (CTRL) a otros campos en otras páginas que conforman la forma (31,32)².

Finalmente se ejecuta el procedimiento "CGLY\$CANVAS_MANAGEMENT" para asegurar que el canvas o lienzo (contenedor donde están colocados los objetos) adecuado esta siendo visualizado. (38)².

El caso especial, cuando no se tiene la funcionalidad que uno desea, como ya se comentó, es necesario insertar codificación personalizada. Una muestra de esto es el siguiente ejemplo, que consiste de el trigger (disparador) PRE-INSERT, colocado a nivel bloque sobre el bloque "SATS". Esta codificación básicamente consiste de asignar el número de Solicitud de Asistencia Técnica, después de haber insertado toda la información concerniente a la SAT.

```
1 declare
2 last_tar sats.no_sat%TYPE;
3 uno number(1) := 1;
4 begin
5     select max(no_sat)
6         into last_tar
7         from sats;
8
9
10    if last_tar is null THEN
11        :sats.no_sat := 1;
12    else
13        :sats.no_sat = last_tar + 1;
14    end if;
```

```
15 : sats.ari .= 1;  
16   end;
```

En este caso se declaran un par de variables, la primera de ellas es del mismo tipo de dato que el campo `no_sat` de la tabla `SATS`, dicha variable se llama `"last_sat# (2)"`. Después se declara una variable numérica llamada "uno" a la cual se le asigna el valor de 1 (3)². Posteriormente se realiza una consulta a la base de datos obteniendo el valor máximo del número de SAT registrado hasta ese momento en la tabla `SATS` en la columna `"no_sat"` y se asigna el valor a la variable `"last_tar" (5,6,7)`². Después se pregunta si la variable `last_tar` es nula, entonces quiere decir que no existe ninguna SAT, hasta ese momento, por lo que se le asignara el valor de 1 a la primera asignando dicho valor al campo `no_sat` que se encuentra ubicado en el bloque `SATS (10.11)`². En caso contrario de que no sea nulo se incrementa a uno el valor extraído de la base de datos y se le asigna al mismo campo (13)². Finalmente se asigna el valor de 1 al campo `ari` del bloque `SATS (15)`².

Es importante notar la diferencia que existe entre la nomenclatura utilizada por `Designer 2000` y por la codificación personalizada. La más importante radica en que todas las variables, bloques y demás elementos de programación, `Designer 2000` les antepone el prefijo "CG", en cambio la codificación realizada por el programador no necesariamente requiere inicializar con dicho prefijo.

Por último es importante comentar que antes de insertar la codificación personalizada es importante realizar las modificaciones necesarias que se requieran hacer desde los módulos de generación de `Designer`. Es decir, cubrir la etapa de generación de prototipos con el fin de que se genere el último prototipo más acorde con las necesidades del usuario final.

Finalmente esta etapa de programación esta íntimamente relacionada con la etapa de generación de prototipos.

4.5 PRUEBAS

La etapa de pruebas es una de las etapas más importantes dentro del desarrollo de sistemas ya que la prueba del software implica el nivel de calidad en que puede evaluarse el mismo.

Según Deutsch:

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son enormes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos... pueden estar especificados de forma errónea o imperfecta, así como los errores que aparecen en los posteriores pasos de diseño y desarrollo... Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo de software ha de ir acompañado de una actividad que garantice la calidad.

El autor Glen Myers establece una serie de reglas que permiten ser tomadas como objetivos de prueba:

1. La prueba de un proceso de ejecución de un programa con la intención de descubrir el error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba tiene éxito si descubre un error no detectado.

Algunas de las pruebas que se llegan a realizar al desarrollar un sistema computacional son básicamente dos tipos de pruebas la prueba de la Caja Blanca y la prueba de la Caja Negra.

4.5.1 PRUEBA DE LA CAJA BLANCA

Este tipo de pruebas permite a los desarrolladores crear casos de pruebas con los cuales el sistema sea revisado minuciosamente. Al aplicar esta prueba se puede garantizar que:

1. Se ejecuten por lo menos una vez todos los caminos de manera independiente en cada módulo
2. Que todas las decisiones lógicas, falsas y verdaderas, se prueben.
3. Todas las estructuras cíclicas se ejecuten en sus límites.
4. Que todas las estructuras internas de validez se ejecuten

Dentro de las pruebas de la caja blanca se encuentra la prueba denominada "*Prueba del camino básico*". Esta prueba fue inicialmente propuesta por Tom McCabe. Esta prueba permite al desarrollador medir de alguna forma la complejidad lógica del diseño de los procedimientos y utilizar esta medida como punto básico de caminos de ejecución. Si se llega aplicar esta prueba se puede garantizar que por lo menos el código del sistema por lo menos se ejecuto al menos una vez. A este tipo de pruebas se le puede denominar también *prueba de la estructura de control*.

Dentro del tipo de pruebas de la caja blanca como son la prueba del camino básico se llevan a cabo pruebas detalladas de las condiciones establecidas dentro del flujo del programa, a partir de una condición simple o lógica o una expresión relacional. Esta prueba de condiciones está intimamente relacionada con la prueba denominada flujo de datos, ya que de las pruebas de condiciones realizadas dependerá la prueba del flujo de datos de acuerdo a la ubicación de las definiciones y usos de las variables el programa. Además dentro de este tipo de pruebas se encuentran las de estructuras cíclicas las cuales son punto fundamental dentro del desarrollo de sistemas.

4.5.2 PRUEBAS DE LA CAJA NEGRA

Las pruebas de la caja negra son realizadas básicamente para probar la funcionalidad del sistema. Es decir, establecer condiciones de entrada que ejerciten de manera completa todas las posibles alternativas de operación del usuario final. Este tipo de pruebas no es una alternativa al tipo de pruebas de la caja blanca se considera como una operación complementaria.

Las pruebas de la caja negra pretenden identificar problemas como:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores de estructuras de datos o en accesos a bases de datos externas.
4. Errores de rendimiento.
5. Errores de inicialización y terminación.

Dentro de las pruebas de la caja negra se pueden mencionar las siguientes:

- *Partición Equivalente*
- *Prueba de Tareas*
- *Prueba de Integración o comportamiento integrado*
 - *Integración ascendente y descendente*

4.5.2.1 PARTICION EQUIVALENTE

Es un tipo de prueba de la caja negra en la cual se determina datos de entrada que pueden derivar casos de prueba y generar por alguna causa algún error, que de manera normal no se llega a presentar en condiciones normales de operación del sistema. Esto reduce el número total de casos de prueba que hay que desarrollar.

4.5.2.2 PRUEBA DE TAREAS

Esta prueba básicamente consiste en probar cada uno de los módulos generados independientemente, de tal forma que podamos identificar algún problema en cada uno de los módulos sin que intervengan más factores. De tal forma es de muy importante que cada módulo funcione de manera correcta y de manera independiente, sin tomar en cuenta con que otros módulos esta interrelacionado.

4.5.2.3 PRUEBA DE INTEGRACION O COMPORTAMIENTO INTEGRADO

Después de haber realizado la prueba de tareas es indispensable realizar una prueba de integración en la cual se creen entradas de datos con las cuales se simule lo mas cercano a la realidad del negocio, una operación *normal*. En algunas ocasiones es común que los módulos independientes funcionan de manera satisfactoriamente, pero al realizar la integración con otros módulos se presentan errores. Es recomendable para este tipo de pruebas realizar una integración incremental de tal forma que si se llegan a presentar errores puedan ser fáciles de acotar e identificar.

Dentro de la prueba de integración existen básicamente dos técnicas la ascendente y la descendente.

La *integración descendente* consiste en un planteamiento incremental en la que se integran los módulos moviéndose hacia abajo por la jerarquía de control, empezando con el módulo de control principal.

En cambio la *integración ascendente* comienza la construcción y la prueba con los módulos independientes. Este proceso implica integrar los *módulos de abajo hacia arriba* con lo cual el procesamiento requerido de los módulos subordinados esta en disponibilidad.

4.5.3 PRUEBA DEL SISTEMA

Finalmente se realizan pruebas donde interactúan factores como nuevo hardware e información, este tipo de pruebas no pueden realizarse o contemplarse dentro del proceso de ingeniería de software, sin embargo las pruebas previamente realizadas incrementan en gran medida las posibilidades de éxito del sistema en el ambiente real de trabajo.

Todas las pruebas de este tipo se llevan a cabo con el fin de verificar que se han integrado adecuadamente todos los elementos del sistema y que se están realizando las funciones apropiadas.

Dentro de las pruebas del sistema podemos encontrar las siguientes:

- Pruebas de recuperación
- Pruebas de seguridad
- Pruebas de resistencia
- Pruebas de rendimiento

4.5.3.1 PRUEBAS DE RECUPERACION

Esta prueba está enfocada a probar el sistema bajo condiciones de falla. Esto implica que los fallos de procesamiento no debe hacer que cese el funcionamiento de todo el sistema.

Las pruebas realizadas consisten en someter al sistema a diferentes situaciones e falla y verificar que las operaciones de recuperación son realizadas satisfactoriamente en un período recomendable. Si la operación de recuperación requiere de la intervención humana es necesario evaluar los tiempos medios de reparación.

4.5.3.2 PRUEBAS DE SEGURIDAD

Todo sistema que manipule información confidencial debe de ser sometido a este tipo de pruebas con el fin de averiguar el nivel de seguridad con el que cuenta. Al realizar este tipo de pruebas en recomendable intentar cualquier operación para poder acceder la información.

4.5.3.3 PRUEBAS DE RESISTENCIA

Este tipo de pruebas esta diseñado para medir el nivel de resistencia a la carga de trabajo antes de que presente algún tipo de problemas. Es decir, realizar pruebas con el sistema con el fin de demandar recursos en cantidad, frecuencia o volúmenes anormales. Como por ejemplo:

- a) Generar pruebas especiales que generen una cantidad alta de interrupciones.
- b) Aumentar el rango de frecuencias de datos de entrada.
- c) Trabajar con casos de prueba que requieran una gran cantidad de memoria o de otro tipo de recursos.
- d) Generar casos de prueba que puedan dar problemas con memoria virtual.
- e) Finalmente, generar casos de prueba que realicen búsquedas de datos residentes en disco.

4.5.3.4 PRUEBAS DE RENDIMIENTO

Las pruebas de rendimiento normalmente son aplicadas a sistemas de tiempo real. Esta prueba esta diseñada para medir el rendimiento del software en tiempo de ejecución dentro de un contexto de sistema integrado.

4.5.4 PRUEBAS REALIZADAS AL SISTEMA AMIGS

Partiendo de que básicamente existen dos tipos de pruebas, la prueba de la Caja Blanca y la prueba de la Caja Negra, en el desarrollo del sistema nos enfocamos de manera directa sobre las pruebas de la Caja Negra y pruebas en Tiempo Real ya que la prueba de la Caja blanca se enfoca de manera muy específica a probar de manera independiente cada secuencia de instrucción de tal forma que se ejecuten por lo menos una vez cada una de las líneas de codificación desarrolladas. De tal forma que si el sistema AMIGS fue desarrollado en su totalidad con una herramienta tipo CASE como es el caso de Designer 2000, un porcentaje del 90 por ciento de la programación fue desarrollado por la herramienta CASE y si nos tomarámos tiempo en probar las líneas de codificación generadas por Designer 2000 no explotariamos al máximo la ventaja de disminuir el tiempo de desarrollo del sistema a partir de una herramienta tipo CASE como Designer 2000.

Como se comenta en el capítulo de prototipos, éstos fueron generados en su totalidad por Designer 2000. Unicamente se desarrollaron algunas modificaciones a las formas y reportes generados por cuestiones de presentación y en algunos casos de funcionalidad e integración entre módulos.

De manera independiente se probaron cada uno de los módulos prototipo generados y se presentaron de manera aislada e integrados para que verificarán la funcionalidad de manera independiente así como la operatividad al integrarse.

Por ejemplo el Método de la *Partición Equivalente* fue aplicado por ejemplo en el módulo forma llamado *Newsat* al momento de registrar un nuevo problema en el área de soporte técnico ya que al tratar de registrar el problema el primer dato que se da es el número de NIC, el cual identifica de manera única a cada cliente entonces en ese momento se aplica una condición de entrada en base a un valor específico. Este mismo método fue aplicado, por ejemplo, al tratar de crear un nuevo contrato de soporte técnico, al dar de alta un nuevo empleado e indicar el área en la cual va a trabajar. Esta prueba de *Partición Equivalente* se puede considerar como una *prueba de validación* de esta forma se garantiza que se cumplen en su totalidad todos los requisitos de rendimiento y funcionalidad.

Este método fue utilizado normalmente para seleccionar los valores de algunos campos en base a lista de valores extraídos de la base de datos.

Algunas otras pruebas que se realizaron fueron en Tiempo Real interactuando el sistema desarrollado con Designer 2000 y la aplicación desarrollada en WEB para consulta de los clientes sobre sus SATs en el área de soporte técnico. En este tipo de pruebas se aplicaron básicamente cuatro pasos.

El primer paso consistió en aplicar la *prueba de tareas*, es decir probar cada una de las partes del sistema con el fin de determinar el buen funcionamiento de cada módulo, especificando de manera clara cada entrada y la salida obtenida. Un caso muy claro es que a partir de un número insertado en el módulo de *Newsat*, que es el correspondiente al NIC del cliente, se obtendrá un número que corresponde a la SAT, esta salida debe de ser correcta y válida ya que es la llave primaria de la tabla sats y este valor debe ser único para cada caso reportado a soporte técnico.

Posteriormente se aplicó la *prueba de comportamiento* integrado en la cual se revisó en cada módulo los casos en que se llevará a cabo todo el proceso de captura de información, cancelación de algún proceso, eliminación de información y esto a su vez cotejándolo de manera directa con la base de datos para asegurar el buen manejo de la misma. Además de las líneas de codificación que Designer 2000 generó en cada módulo la validación se realiza en el manejador de Oracle garantizando la integridad de la información.

Como tercera prueba se aplicó la prueba *intertareas* es decir trabajar de manera paralela entre dos clientes con el sistema instalado y que ambos

usuarios finales trabajaran de manera independiente en los casos reportados sin interferirse mutuamente entre sus actividades. Otra de las pruebas que se realizaron fue interactuando con la aplicacion de WEB desarrollada y consultando de manera inmediata las modificaciones realizadas en el sistema AMIGS

Otra de las pruebas aplicadas fue la prueba de integraci3n ya que en la mayoria de las ocasiones los m3dulos *trabajan de manera correcta* individualmente pero al integrar los m3dulos se pueden llegar a presentar varios problemas de integraci3n. Aplicamos en la etapa de integraci3n la integraci3n descendente, es decir, a partir del m3dulo principal y el men3 general de la aplicaci3n se prob3 la integraci3n y la invocaci3n del resto de los m3dulos. Se aplic3 una prueba especifica sobre la integraci3n de muy peculiar de un par de m3dulos que *trabajando de manera independiente* no se presentaba ning3n problema pero al integrarlos el resultado no era satisfactorio. Estos m3dulos fueron la forma *saxing* y *safacc*. La primera de estas formas muestra las SAT abiertas, cerradas o inactivas por el analista que uno indique. A partir de esta forma y presionar un bot3n asociado a cada SAT se invoca de manera autom3tica el m3dulo *safacc* el cual muestra el detalle de la SAT seleccionada en *saxing* y las acciones realizadas en dicha SAT. Situaci3n que no ocurría ya que solamente se desplegaba el detalle de la SAT pero no las acciones realizadas sobre dicha SAT.

Se revis3 a detalle la secuencia de triggers que se iban ejecutando al correr la forma y se not3 que al insertar una lnea m3s en la codificaci3n creada por Designer 2000 el problema de integraci3n se solucionaba. La codificaci3n que se modific3 se encuentra en el trigger when-new'block-instance a nivel forma. La codificaci3n fue la siguiente:

```
/* CGLY$MANAGE_CANVASES */
/* Call procedure to ensure correct canvases are visible */
BEGIN
  CGLY$CANVAS_MANAGEMENT,
END;

/* CGBSSWHEN_NEW_BLOCK_INSTANCE */
/* ensure the current canvas is correctly coordinated */
BEGIN
  CGBSS$NEW_BLOCK(:SYSTEM CURSOR_BLOCK, 'GLOBAL CGB$QUERY_MODE),
  execute_query; /* Lnea que solucion3 el problema de integraci3n */
END;
```

Al revisar la codificaci3n y secuencia de ejecuci3n de 3sta se determin3 que si en ese punto de ejecuci3n se realizaba una consulta al bloque de acciones del m3dulo *safacc* autom3ticamente tambi3n las acciones realizadas a esa SAT serían desplegadas.

Finalmente se llevaron a cabo *pruebas del sistema*, esto implicó realizar *pruebas de recuperación*, pruebas de seguridad, pruebas de resistencia y pruebas de rendimiento.

Dentro de las *pruebas de recuperación* se pueden llegar a presentar algunos casos en los que el sistema se puede llegar a corromper. Algunos de estos casos de pruebas nos han llevado a determinar que algunos aspectos pueden influir en el buen funcionamiento del sistema. El primero de ellos es que en las máquinas cliente donde se ejecutará el sistema se eviten cualquier tipo de savescreeens ya que en algunos de los casos puede llegar a provocar un debordamiento de la pila en la PC y provocar que el sistema termine de manera anormal. Y otro caso es porque el suministro de energía haya fallado en algún momento y alguno de los archivos ejecutables de la aplicación llegará a sufrir alguna corrupción

Se llevaron a cabo *pruebas de seguridad* en las cuales se verificó la seguridad proporcionada por el manejador de Oracle. No hay forma alguna de determinar el password de los usuarios al menos que obtuvieran los archivos fuente de la aplicación, realizarán las correcciones correspondientes para capturar los passwords de los analistas, llegar a generar esas posibles modificaciones y generar el correspondiente archivo y sustituirlo por el original. Por otro lado la aplicación maneja seguridad a nivel de menú, es decir que no todos los usuarios pueden llegar a tener acceso a todas las formas en tiempo de ejecución. Si por algún motivo alguien desea ejecutar una forma a la cual no tiene privilegios para ejecutarla de manera normal pero la ejecuta de manera independiente, AMIGS maneja roles a nivel base de datos los cuales otorgan los privilegios necesarios a cada usuario para realizar las diferentes acciones (inserción, modificación, eliminación) sobre las tablas correspondientes. Esto fue probado desde una sesión de SQL*PLUS y en tiempo de ejecución del sistema AMIGS.

En cuanto a *pruebas de resistencia* se identificaron casos en los que se puede llegar a presentar problemas por insuficiencia de espacio en disco duro para realizar de manera correcta la operación swap y la ejecución de los reportes por falta de este recurso. Por tal motivo se recomienda que se indiquen unos 50 megabytes de memoria virtual y se tenga el mismo espacio libre en disco duro para el proceso de swapeo. Este espacio no es consumido en su totalidad por el proceso asociado al sistema AMIGS sino se toma el valor como un aproximado, considerando algunas otras aplicaciones ejecutándose en ese momento

Finalmente las *pruebas de rendimiento* se llevaron a cabo en el servidor en el cual por el momento no se presentó ningún problema en el rendimiento ya que como esta configurada la SGA en este momento puede soportar hasta 200 conexiones simultáneas sin problemas y que dichas conexiones realicen las

operacions normals de inserción, actualización y eliminación. Se pueden llegar a presentar problemas de conexión e insuficiencia de recursos cuando se lleguen a superar este nivel de conexiones.

4.6 IMPLANTACION

La implantación se puede realizar aplicando dos procedimientos. El primero de ellos consiste en realizar la instalación de todos los archivos de las correspondientes formas y reportes, así como del Runtime de Forms y de Reports y SQL*NET para poder ejecutar la aplicación desde las máquinas cliente. La segunda alternativa es tener una instalación de red tipo LAN (Local Area Network) en la cual tengamos un servidor de archivos donde estén colocados tanto los archivos correspondientes a las formas como de los reportes, además de los runtime de las aplicaciones.

A continuación se explicará de manera detallada la forma de realizar la instalación de cada una de éstas.

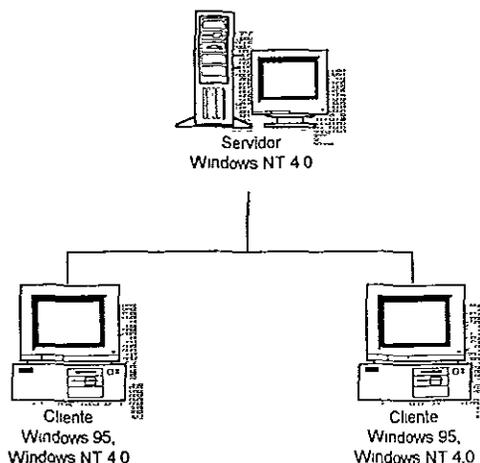


Figura 4.6.1 Esquema cliente/servidor con instalación local de las formas y reportes en cada cliente

4.6.1 INSTALACION LOCAL EN EL CLIENTE

Este tipo de instalación consiste en realizar la copia de todos los archivos correspondientes a las formas y los reportes, y así como la instalación de los runtime de Forms y Reports.

Para realizar la instalación de los runtime de Forms y Reports se deben de llevar a cabo los siguientes pasos:

- a) Insertar el CD de Developer 2000 r 1.3.2 en una unidad de CD y automáticamente se ejecutará el programa de instalación de productos Oracle para Windows 95 ó Windows NT.
- b) Primero se presentará una ventana donde se deberá especificar el idioma en el que se instalará el producto. La ventana que se presenta es la siguiente:

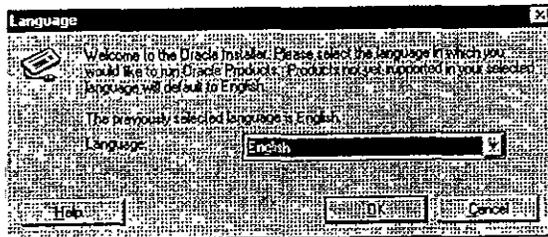


Figura 4.6.2 Ventana que nos pide el idioma en el que se instalara la herramienta Developer 2000

Por default siempre la instalación se realiza en idioma ingles.

- c) Posteriormente se desplegará la ventana donde se proporcionará el nombre de la compañía y el punto de instalación, el cual es conocido como el directorio casa de Oracle. En el caso de Windows 95 el directorio casa es OraWin95 y para Windows NT el directorio casa se identifica por Orant. La segunda ventana que se presenta es la siguiente:

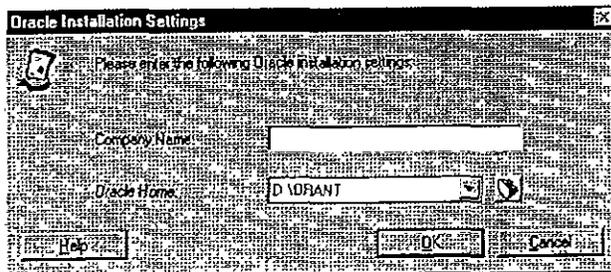


Figura 4.6.3 Ventana que nos pide el nombre de la compañía, y el directorio casa de Oracle.

- d) La segunda ventana de instalación que se presentará consiste en indicar el tipo de instalación que se realizará. Teniendo básicamente tres opciones. Modo interactivo, Opciones de Default e Instalación personalizada o desinstalación. La ventana que se presenta es la siguiente:

Figura 4.6.5 Ventana que nos permitirá realizar la instalación de manera personalizada de los productos que deseemos, y nos mostrará la lista de productos Oracle para Windows a 32 bits que ya se encuentran instalados.

En esta ventana lo que se realiza es la selección de los productos que se desean instalar en la ventana presentada del lado izquierdo. Al presionar el boton "Install" que se encuentra localizado en parte central de la ventana principal, los productos seleccionados quedarán instalados y se indicarán en la ventana de la derecha

Los productos necesarios para poder correr el sistema AMIGS son los siguientes: Forms Runtime 4.5.7.1.6, Reports Runtime 2.5.5.2.7, SQL*NET Client 2.3 2.1.6A.

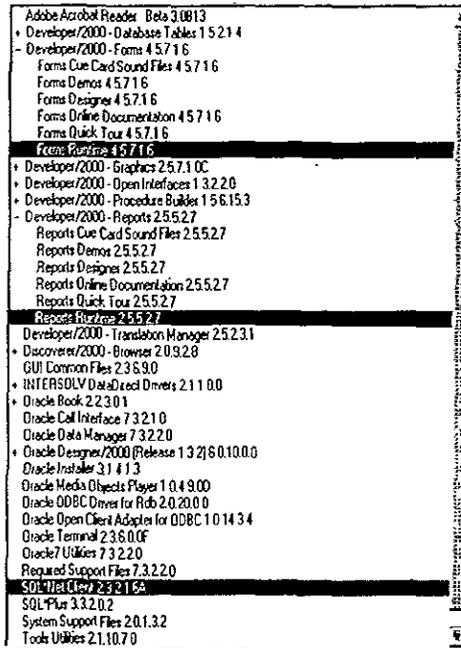


Figura 4.6.6 Selección de productos a instalar(Forms Runtime, Reports Runtime, SQL*NET Client .

Al realizar la instalación de los productos indicados realiza también la instalación de otros componentes que requieren los productos y que son instalados aún sin indicarlo explícitamente. Estos elementos son; Oracle Installer 3.1.4.1.3, Required Support Files 7.3.2.2.0, los System Support Files 2.0.1 3.2, los GUI Common Files 2.3.6.9.0 y los Tools Utilities.2.1.10 7.0.

f) Después de haber realizado la instalación de los productos se tiene que revisar si éstos fueron instalados correctamente. La forma de realizar la prueba es ejecutando el Runtime de Forms y el Runtime de Reports. Para realizar esta prueba se ejecuta directamente, dentro del ambiente de Windows 95 ó Windows NT, accediendo a la barra de herramientas de Windows, presionando Inicio o Start, presionando la opción de programas podremos observar el conjunto de programas identificado por Developer 2000 for Windows 95 ó Developer 2000 for Windows NT. Dentro de este conjunto de programas localizaremos los runtime de forms y de reports los cuales pueden ser ejecutados.

Otra alternativa para verificar la correcta instalación consiste en comprobar que los ejecutables del runtime de forms y reports existe. Estos son colocados después de haber realizado la instalación bajo el subdirectorio *bin* del directorio casa de los productos Oracle (para Windows 95 es típicamente *c:\Orawin95* y para Windows NT es *c:\Orant*). El nombre del ejecutable asociado al runtime de Forms 4.5 es *F45run32.exe* y para el runtime de Reports 2.5 es *R25run32.exe*. Estos programas pueden ser ejecutados desde el explorador de Windows.

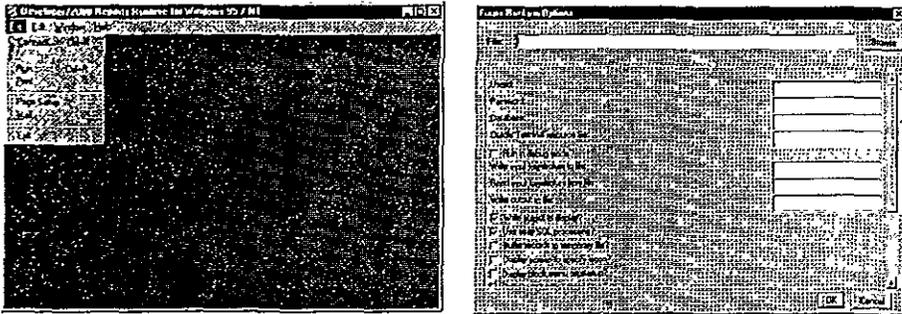


Figura 4.6.7 Ventanas asociadas al Runtime de Reports 2.5.

A continuación se debe proceder a instalar la aplicación y configurar el ambiente para poder ejecutar la aplicación.

4.6.2 INSTALACION Y CONFIGURACION DEL SISTEMA AMIGS

Primero se indicarán los archivos necesarios que se deberán de copiar para poder ejecutar el sistema. Todo sistema desarrollado con Designer 2000 y Developer 2000 está compuesto de tres tipos de programas con las siguientes extensiones: *frx* (archivos ejecutables generados con Forms Designer); *rep*

(archivos ejecutables generados con Reports Designer); *plx* (archivos ejecutables generados con Forms Designer y que corresponden a las librerías que utiliza la aplicación); *mmx* (menús generados con Forms Designer); los archivos con extensión *ico* (imagenes iconizadas); y los archivos con extensión *hlp* (archivos ejecutables de la ayuda).

A continuación se listan los archivos necesarios que conforman todo el sistema.

a) Formas:

main.fmx	contsop.fmx
client.fmx	satnew.fmx
emplea.fmx	saxing.fmx
licenc.fmx	satacc.fmx
platclie.fmx	

b) Reportes:

cartclie.rep	satsana.rep
cteplat rep	contra.rep
plataf.rep	

c) Librerías:

ofg4bsl.plx	ofg4tel.plx
ofg4hlp.plx	

d) Menús.

start mmx

e) Iconos:

afsave.ico	afdelrw.ico
afprint.ico	afclrw.ico
afclrall.ico	afedit.ico
affind ico	aflist.ico
afinsrw.ico	afhlpw.ico

f) Archivos hlp:

main.hlp	contsop.hlp
client.hlp	satnew.hlp
emplea.hlp	saxing.hlp
licenc.hlp	satacc.hlp
platclie.hlp	

Antes de realizar la copia de los archivos correspondientes en la máquina se debe de crear la siguiente estructura de directorio. Primero debemos de

crear un directorio llamado AMIGS bajo la unidad del disco duro de la máquina y bajo este directorio crear tres directorios más: formas, reportes, icons.

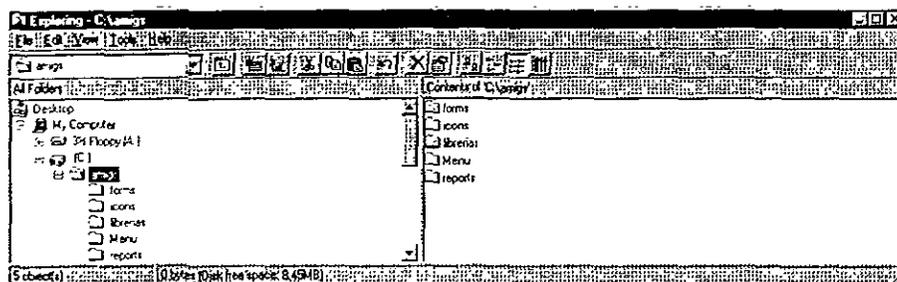


Figura 4.6.8 Creación de los directorios: formas, reportes e icons.

Se tendrá que realizar la copia de los archivos de las formas en el directorio forms, las librerías en el directorio librerías y el menú bajo el directorio menú. Bajo el directorio reports se deben de copiar los correspondientes reportes. Los archivos tipo *ico* bajo el subdirectorio icons. Finalmente los archivos *hlp* deberán ser copiados bajo el subdirectorio *bin* del directorio casa de los productos Oracle.

4.6.2.1 CONFIGURACION SQL*NET

Posteriormente se tendría que realizar la configuración de SQL*NET para establecer la comunicación con la base de datos en el servidor Windows NT. Esta configuración puede realizarse de dos formas. La primera de ellas es realizando la configuración de manera manual, es decir, editando y modificando el archivo *tnsnames.ora* y que se encuentra ubicado bajo el subdirectorio *c:\Orawin95\network\admin*. Se tendría que agregar un alias de conexión, en este caso lo llamaremos *amigs* y su definición quedaría de la siguiente forma:

```
amigs.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = tcp.world)
        (PROTOCOL = TCP)
        (Host = ip_windowsNT_server)
        (Port = 1521)
      )
    )
    (ADDRESS =
      (COMMUNITY = tcp.world)
```

```

        (PROTOCOL = TCP)
        (Host = ip_windowsNT_server)1
        (Port = 1526)
    )
)
(CONNECT_DATA = (SID = AMIGS)
)
)

```

El segundo método consiste en realizar la configuración de SQL*NET con un asistente gráfico llamado *Easy Configuration* que se encuentra localizado dentro del grupo de programas Oracle for Windows 95 u Oracle for Windows NT. Este asistente nos presentaría la siguiente secuencia de ventanas y al insertar los valores el modificaría o crearía el archivo tnsnames.ora.

Al ejecutar el *Easy Configuration* se presentaría la primera ventana donde indicaríamos si deseamos agregar, modificar, borrar o rebisar la configuración que se tiene en ese momento. Seleccionamos la opción de agregar un nuevo alias y continuamos la instalación. A continuación se presenta la primera ventana que el *Easy Configuration* presenta.

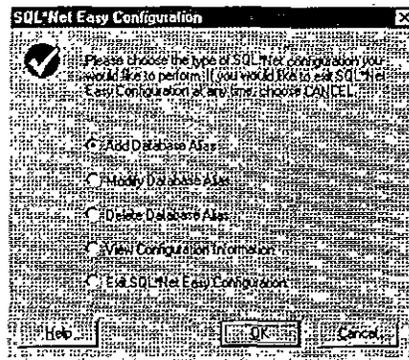
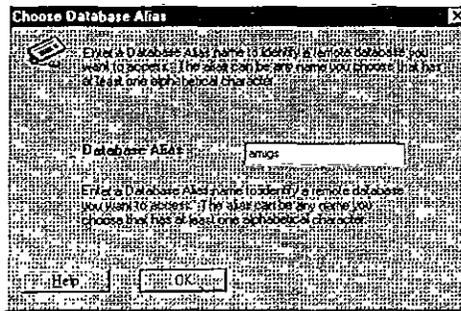


Figura 4 6 9 Forma sencilla de configurar SQL*NET

En la segunda ventana se solicitará el nombre del alias con el cual se desea realizar la conexión hacia la base de datos. En este caso como ya se comentó se indicaría *amigs*.

¹ *ip_windowsNT_server* corresponde a la dirección IP del servidor Windows NT donde se encuentra la base de datos



4.6.10 Solicitud del nombre del alias con el cual se desea realizar la conexión con la base de datos. El alias es amigs

Posteriormente te solicitará el protocolo de comunicación que utilizará SQL*NET para establecer la comunicación. En este caso se indicaría el protocolo TCP/IP.

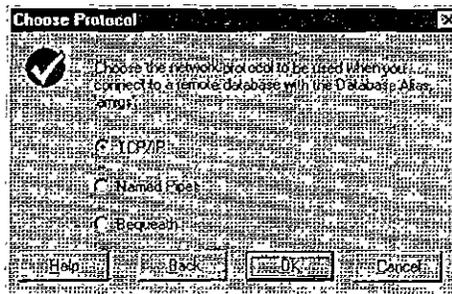


Figura 4.6.11 Especificación del protocolo de comunicación. En este caso es TCP/IP.

En la siguiente ventana se indicará la dirección IP que corresponde al servidor Windows NT donde se encuentra la base de datos y la instancia o proceso asociado a la base de datos en el servidor Windows NT. En el caso del Sistema Amigs el nombre la instancia y del alias de conexión son los mismos.

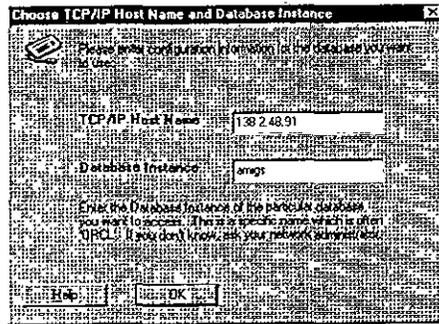


Figura 4.6.12 Dirección IP que corresponde al servidor Windows NT donde se encuentra la base de datos

Finalmente te presenta una ventana con la información que se insertó y nos da la opción de aceptar para modificar el archivo *tnsnames.ora* a partir de los parámetros insertados o eliminar la modificación.

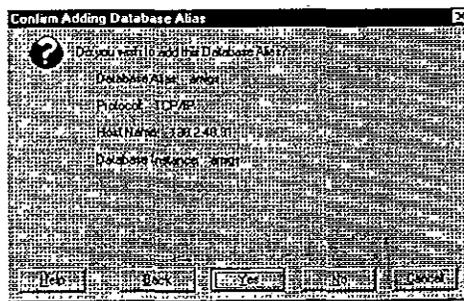


Figura 4.6.13 Ventana de confirmación

4.6.2.2 CONFIGURACION DEL REGEDIT.

Dentro del Regedit (punto de configuración de windows95 y windows NT) se deberán de especificar las rutas donde están ubicados los reportes, las formas, el menú, las librerías y los iconos. Se deben de especificar tres variables de configuración. Las cuales están ubicadas dentro del regedit bajo la ruta *My computer\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE*. Estas variables deben de contener las rutas donde están colocados los archivos.

```
FORMS45_PATH = C:\AMIGS\FORMAS;C:\AMIGS\LIBRERIAS;C:\AMIGS\MENU
```

```
REPORTS25_PATH = C:\AMIGS\REPORTS
```

```
TK23_ICON = C:\AMIGS\ICOSNS
```

Para acceder al Regedit, unicamente tenemos que ejecutar la opción de RUN o Ejecutar dentro de la barra de herramientas de Windows y en la caja de texto teclea regedit. De esta forma se presentará la siguiente pantalla

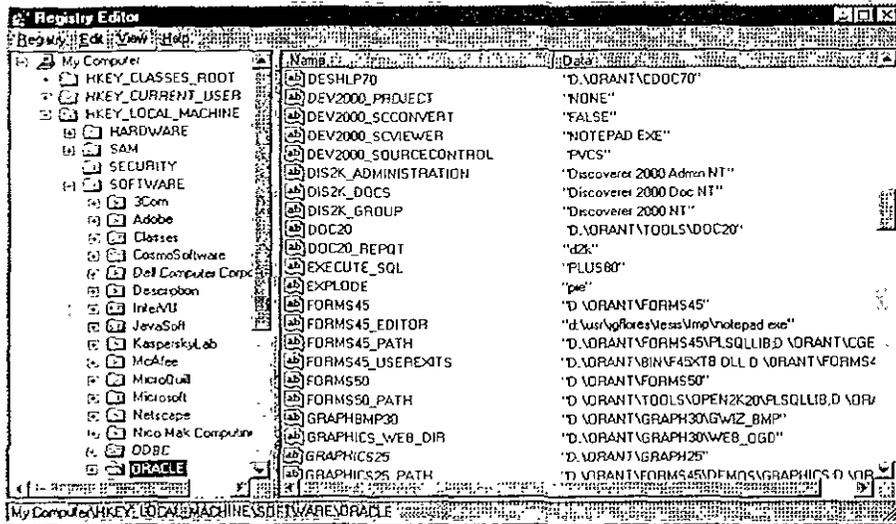


Figura 4.6.14 Acceso al REGEDIT de Windows

Finalmente se tendría que crear un acceso directo para poder ejecutar el sistema. Dentro de la línea de comandos para ejecutar la aplicación se tendría que colocar la siguiente línea:

```
C:\ORAWIN95\BIN\F45RUN32.EXE MODULE=MAIN.FMX USERID=AFUELLO/ @AMIGS
```

Con esta línea de comando se invoca el runtime de Repots que ejecutará la forma principal llamada main.fmx. La ventana de conexión presentará los datos del usuario y la cadena de conexión a la base de datos, de tal forma que el usuario unicamente tiene que insertar su password.

4.6.2.3 INSTALACION SOBRE UNA RED TIPO LAN

Este tipo de instalación consiste en la realización de cuatro pasos:

- Instalar Developer 2000 sobre un servidor de archivos.
- Importar el regedit a todos los clientes
- Crear un grupo de programas en todos los cliente.
- Instalar los System Support Files

Dentro de este proceso de instalación primero se realiza todo el proceso de instalación completo sobre una máquina cliente y posteriormente se importa en todos los demás clientes el regedit.

Este paso consiste en realizar el mapeo hacia el disco duro del servidor de archivos desde una máquina cliente, de tal forma que ese mapeo de unidad se vea como una unidad adicional de almacenamiento en la máquina cliente.

Ya realizado el mapeo se realiza la instalación del producto como se comentó en la parte *Instalación local en el cliente* pero haciendo referencia a la unidad mapeada al servidor de archivos

Después de haber realizado la instalación de manera normal y haber verificado el buen funcionamiento del producto, sobre esta máquina cliente se creará una estructura del regedit y se copiarán en esta máquina los archivos necesarios para poder acceder los ejecutables en el servidor de archivos. Estos archivos son: Required Support Files 7 3.2.2.0, los System Support Files 2 0.1 3.2, los GUI Common Files 2 3.6 9.0 y los Tools Utilities.2.1.10.7.0.

Posteriormente se tendrán que realizar todos los pasos necesarios para poder instalar y configurar el sistema AMIGS² sobre esa máquina cliente. Ya que el sistema halla funcionado de manera correcta sobre esta máquina cliente es necesario hecer el export del Regedit de esta máquina cliente y la operación de import del Regedit a las máquinas cliente que se conectarán a la base de datos AMIGS.

Es importante realizar un export previo del Regedit de cada máquina cliente la cual utilizará el sistema AMIGS.

Para realizar el import y export del Regedit se utiliza la opción Import Registry File... y Export Registry File... del menu Registry de la ventana Registry Editor. El proceso de importación y exportación puede llevarse a cabo de todo el Regedit o solo de una parte. En el caso del sistema AMIGS únicamente están involucrados los productos Oracle por lo cual sólo se importará y exportá la rama correspondiente a Oracle, basta con posicionar el cursor en *My computer\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE*.

² Pasos que se comentan en *Instalación y configuración del sistema AMIGS, Configuración SQL*NET, Configuración del Regedit y Acceso Directo*

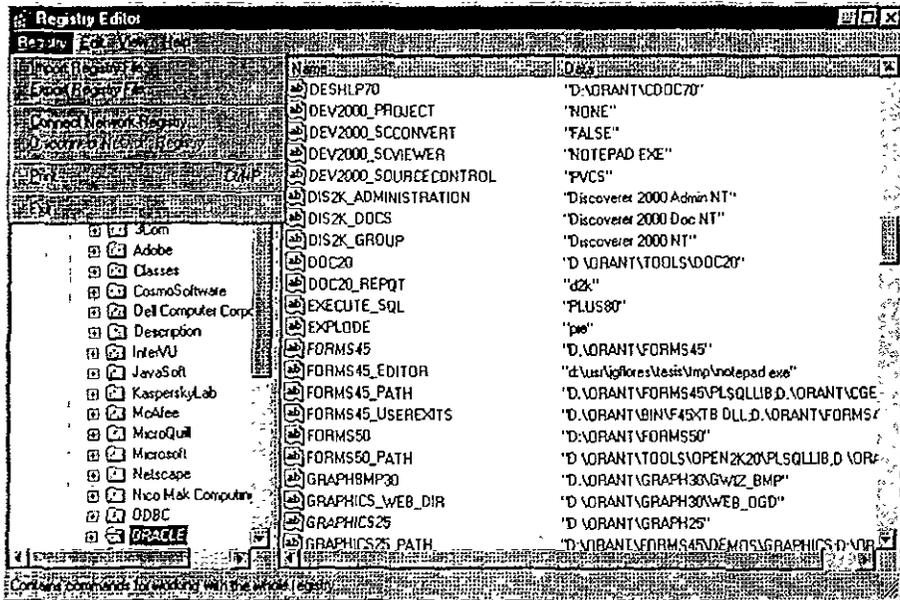


Figura 4.6.15 Importación y exportación del Regedit

4.7 GENERACION DE AYUDA DESDE DESIGNER/2000

Uno de los aspectos más importantes en el manejo de un sistema es la ayuda que le puede brindar al usuario final, en la mayoría de los casos, se entrega del sistema y el manual del usuario. Sin embargo, siendo realistas, la mayoría de veces los manuales no los tiene la persona que utiliza el sistema, o están bajo llave, o simplemente no existen las copias necesarias para todos los usuarios del sistema.

Una ventaja que ofrece Designer / 2000, es la facilidad que se tiene para generar ayuda en línea, y de esta forma cualquier usuario pueda acceder a la ayuda simplemente colocando los cursos en determinado campo. En esta parte hablaremos sobre como se genera la ayuda en línea para el sistema.

4.7.1 CARACTERISTICAS DE UN SISTEMA DE AYUDA MICROSOFT WINDOWS

Un sistema de ayuda Microsoft Windows consiste en uno o más archivos HELP (.hlp) cuya información se despliega a través de tópicos y ventanas "popup". Cada archivo contiene:

- Página de contenido: Organización de los temas en forma jerárquica, listando la categoría de los temas disponibles y permitiendo a través de ellos profundizar en los detalles de la aplicación.
- Glosario: Definiciones de términos del negocio e ilustraciones presentadas en ventanas " popup ".
- Indice: Utilidad para búsqueda rápida de información a través de palabras claves.
- Ayudas sensibles al contexto: Características que permiten el despliegue de la información correspondiente al tema del ítem de la aplicación que se encuentra en foco al invocar la ayuda.

4.7.1.1 CONFIGURACION DEL MS HELP GENERATOR

MS Help Generator requiere del compilador y del ejecutable de ayuda de Microsoft Windows para compilar y ejecutar, respectivamente, los archivos generados. Por lo tanto, algunas tareas preliminares deben ser realizadas para establecer el ambiente apropiado para la generación de ayuda Windows desde Designer/2000. Estas tareas se describen a continuación:

1. Definir el directorio de trabajo del MS Help Generator: El directorio de trabajo especificado para Designer / 2000 es el mismo que utiliza MS Help Generator para almacenar los archivos de ayuda. Dado este hecho, antes de utilizar el generador se debe verificar el directorio de trabajo para modificarlo, si es necesario establecer el ambiente apropiado para la generación de ayudas Windows desde Designer/2000. Estas tareas se describen a continuación:
2. *Instalar el compilador y ejecutable de ayudas de Windows.* MS Help Generator no genera los ejecutables de los archivos de ayuda, sino los archivos fuentes (rtf y hpJ). Para generar el ejecutable, se requiere el compilador de ayudas de Windows, HC31.EXE HC,EXE o HCP.EXE.

Luego de haber compilado los archivos fuentes es necesario, para correrlos poseer el ejecutable de ayudas Windows, WINHELP.EXE.

3. *Configurar los parámetros de ayuda Windows.* Después de verificar la existencia del compilador y ejecutable de ayuda, se debe especificar el nombre y la ubicación de estos archivos para que el generador de ayudas de Designer/2000 pueda encontrarlos. Los pasos por seguir para la configuración de estos parámetros son los siguientes:

Para Windows95/NT

- a) Invocar el Registry Editor (regedit.exe en Windows 95 y regedit32.exe en Windows NT).
- b) Expandir HKEY_LOCAL_MACHINE .. SOFTWARE ... ORACLE ... CGENH10.
- c) Editar el valor de las variables: HelpCompiler y HelpExecutable para modificarlo de la siguiente forma:

Value name: HelpCompiler
String: /<ruta>/hcp.exe
Value name- HelpExecutable
String- <ruta>WinHe1p.exe

Si las variables no existen, éstas deben ser creadas con los valores anteriores.

- d) Cerrar el Registry Editor.

4.7.1.2 PROCEDIMIENTO PARA LA GENERACION DE AYUDAS DE MICROSOFT WINDOWS

1. Definición de la estructura del sistema de ayuda. Una aplicación en Designer/2000 consiste de módulos con información detallada de los datos que utilizan. El enlace entre estos módulos, *red de módulos*, conforma la aplicación Windows. Con base en esta red, se define la estructura del sistema de ayuda, escogiendo una de las siguientes opciones:

Opciones

Un único archivo de ayuda para representar toda la red de módulos.

Un archivo de ayuda para cada subsistema de módulos.

Un archivo de ayuda para cada módulo individual.

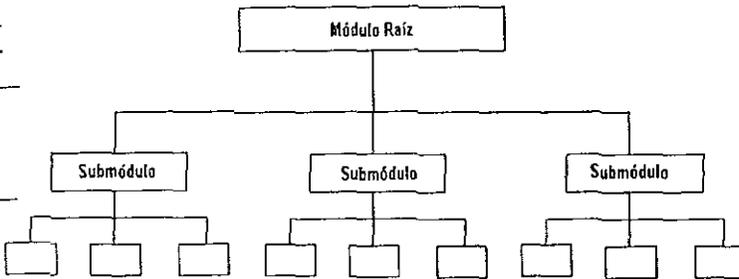


Figura 4.7.1 Estructura de un sistema de ayuda

Si la decisión es crear un archivo de ayuda para toda la red de módulos, se debe identificar como módulo raíz, el módulo que origina la red. Si la decisión es generar archivos de ayuda para los subsistemas de módulos, los módulos raíz son los que originan las subredes. Y, si en cambio, se desea generar un archivo de ayuda para módulos individuales, cada uno de ellos será el módulo raíz.

Ya identificados los módulos raíz, a cada uno de ellos se les debe asignar el nombre de los archivos fuentes de ayuda que contendrán su información y la de los módulos que existen bajo ellos.

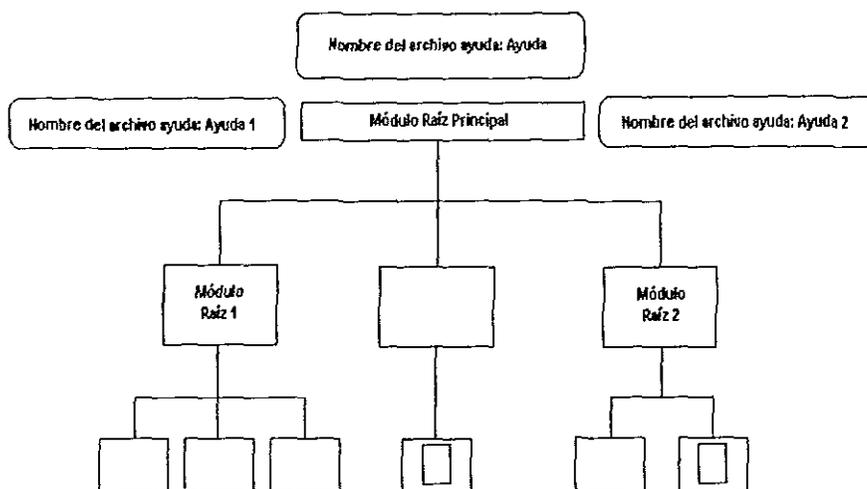


Figura 4.7.2 Asignación de nombre a los archivos fuentes de ayuda

El archivo Ayuda 1 va a contener el texto de ayuda de los módulos del subsistema del Módulo raíz 1.

El archivo Ayuda 2 contendrá el texto de ayuda de los módulos del subsistema del Módulo raíz 2.

El archivo Ayuda va a contener el texto de ayuda definido para el módulo raíz y los dos módulos que no se encuentran en los subsistemas 1 y 2.

Definición de la propiedad Help File Name

Desde el Module Structure Diagrammer

- Abrir el diagrama que contiene la red de módulos.
 - Seleccionar el módulo raíz y editar sus propiedades.
 - Seleccionar la opción del *Help Client Details* de la carpeta de propiedades, y especificar el nombre del archivo de ayuda en la propiedad Help file name.
2. Especificación de las propiedades de ayuda. Al tener definida la estructura del sistema de ayuda, se deben especificar el texto de ayuda por desplegar en los archivos, las palabras claves, el glosario, las gráficas que ilustrarán la ayuda y las preferencias del diseño del sistema.

Las palabras claves son automáticamente generadas por el MS Help Generator de los títulos definidos para los módulos, tablas, grupos de items y columnas; existe la disponibilidad de adicionar nuevas palabras claves a los objetos mencionados anteriormente.

La presentación de los archivos de ayuda se puede modificar configurando preferencias que establecen la estructura de temas, las acciones por llevar a cabo cuando el texto de ayuda no se encuentra definido y la generación automática de palabras claves.

- 3 Generar los archivos fuentes de ayuda.
4. Compilar los archivos fuentes
- 5, Ejecutar los archivos de ayuda generados.
- 6 Integrar el sistema de ayuda en la aplicación.

4.8 IMPLEMENTACION EN WEB

Uno de los principales obstáculos a los cuales se enfrentan los clientes de un centro de soporte, es cuando se ven obligados a laborar en horas fuera de oficina, o bien los fines de semana; esto no tendría importancia si al realizar sus labores programadas, todo tuviera un resultado exitoso; sin embargo, es bien sabido que la mayoría de las ocasiones se tiene algún problema cuando se está trabajando a esas horas. El punto aquí es que no siempre los centros de soporte cuentan con personal las 24 horas los 7 días de la semana. O bien en ocasiones los clientes tienen dudas o problemas demasiado sencillos en los cuales no sería necesario que otra persona por teléfono lo esté guiando para resolver esa situación.

Es por ello que una opción que tienen los clientes del centro de soporte, es utilizar las herramientas que tienen disponible, y una de ellas es el uso de internet. Conociendo que la mayoría de los clientes tienen acceso al internet, y ayudados por los productos que están en el mercado, decidimos establecer un medio de comunicación con los clientes el cual esté disponible las veces que ellos lo necesiten, y a la hora que ellos prefieran

Ayudados de las herramientas Oracle, en este caso de Web Application Server, pudimos crear una interfaz que accesa directamente a la base de datos del grupo de soporte, permitiendo al cliente el tener soporte las 24 horas del día los 7 días de la semana

Seguridad en el Web Application Server

Esta herramienta, presenta varias medidas de seguridad, las cuales no profundizaremos, pero daremos una breve descripción de tal forma que se llegue a comprender del porqué el uso de esta herramienta.

Existen tres tipos de seguridad que maneja el Web Application Server:

Autenticación Básica

Si se utiliza este tipo de autenticación, el browser solicita un usuario y un password cuando se intenta acceder a una página o aplicación protegida. En caso de que se esté utilizando una conexión sin un SSL (Security Socket Layer), el usuario y el password son enviados a través de la red sin antes haberlos encriptado, y cualquier persona podría interceptar y obtener el usuario y el password; por esta razón, Oracle recomienda que se utilice la seguridad SSL en conjunto con la autenticación básica.

Autenticación Digest

Este protocolo es parte de http:1.1." >HTTP 1.1. Es el mismo que el anterior, excepto que este sí encripta el password. Actualmente, solo algunos browsers soportan este tipo de autenticación, (Spyglass Mosaic si lo soporta, pero Netscape Navigator 3.0 y Microsoft Internet Explorer 3.0 no). Si esta autenticación es utilizada para una página, y un usuario utiliza un browser que no la soporta, e intenta acceder a la página, el listener automáticamente utiliza la autenticación básica. Por esta razón cuando se utiliza esta autenticación, se debe considerar que es la autenticación básica.

Restricción por dirección IP

El administrador puede permitir o negar el acceso a una dirección IP específica. Las direcciones IP son organizadas jerárquicamente. Este mecanismo permite que subárboles completos puedan ser incluidos o excluidos.

Restricción del Nombre del Dominio

En lugar de permitir o restringir el acceso a través de la dirección IP, el administrador lo puede hacer utilizando el nombre de un dominio. Se pueden utilizar porciones del dominio para restringir el acceso; por ejemplo, *.xyz.com niega el acceso a todos los clientes que provengan del dominio xyz.com. El nombre del dominio es una mejor forma que la dirección IP para permitir o negar el acceso a las páginas o aplicaciones.

Autenticación a través de la base de datos

La autenticación a través de la base de datos está disponible solamente a través del Servidor de Autenticación, y no del listener. En este tipo de autenticación, los usuarios y passwords son almacenados en la base de datos Oracle y no en un archivo de sistema operativo tal como en los esquemas de autenticación básica y de digest. Esto simplifica el trabajo del administrador, debido a que solamente un conjunto de usuarios y passwords es necesario que sean manejados tanto por el Web y por la base de datos.

También incrementa la seguridad, debido a que el Oracle7 Server proporciona un acceso gradual más controlado que el sistema operativo. El administrador configura un grupo, el cual consiste de una cadena de conexión, si se trata de una base de datos remota, o bien solamente el identificador de una base de datos local. Además ese grupo puede contener un role de la base de datos, el cual permite solo a los usuarios que tienen privilegios en ese role.

Otro elemento de seguridad, que se puede considerar, sin que realmente sea propiamente dicho, es la Descripción del Acceso a la base de Datos (DAD) . Sin embargo es un elemento muy importante al implementar un esquema de seguridad.

Los DAD son utilizados para controlar el acceso a la base de datos desde un cartucho de PL/SQL. El DAD es una sección de la configuración que se almacena en el archivo wrb.app, el cual es el archivo que almacena la configuración del Web Application Server. El DAD puede contener o no el nombre y password del usuario, los cuales son utilizados para conectarse al servidor Oracle7. Cabe resaltar que este método previene que los usuarios y passwords sean transmitidos a través de HTTP.

Si el DAD no contiene el usuario y el password, o si falla la conexión al tratar de utilizar los almacenados; el cartucho de PL/SQL ejecuta una autenticación básica. Esto significa que solicitará al Web Application Server que solicite el usuario y el password para poder realizar la conexión. Se debe tomar en cuenta que el usuario y el password son enviados a través del browser al listener, por lo tanto, si la seguridad es importante, es necesario utilizar la seguridad SSL al invocar a los cartuchos de PL/SQL.

En base a las definiciones anteriores, decidimos utilizar el DAD para acceder a la base de datos, y considerando que no todos los clientes tienen browser que soporte algún otro esquema de seguridad, decidimos ayudarnos de las características de seguridad que nos ofrece el Oracle7 server para poder mantener los datos confidenciales de una forma segura.

Para comprender de una manera más sencilla el método de seguridad que utilizamos para la creación del Repositorio de Soporte Técnico, utilizaremos las pantallas del repositorio.

Para conservar la integridad de la base de datos, se crearon vistas de las tablas

Una vista es una presentación de los datos de una o más tablas. Las vistas de hecho, no contienen datos; más bien, sus datos provienen de las tablas de las cuales están basadas. Las vistas proporcionan un nivel adicional de seguridad, restringiendo el acceso a un conjunto predeterminado de registros y columnas de una tabla. Por ejemplo, las vistas que utilizamos para poder acceder los datos a través del web, no contienen datos sensitivos. De esta forma solo le está permitido al usuario almacenado en el archivo wrb.app y que es el que se utiliza para realizar la conexión a la base de datos el seleccionar ciertos campos de las columnas que conforman las vistas que se crearon; así ningún usuario a través del web puede acceder a las tablas verdaderas de la base de datos, y mucho menos puede realizar algún cambio a dichas tablas.

En primer lugar, cuando se accesa al Repositorio de Soporte, se presenta la pantalla que se muestra en la figura 4.8.1, la cual nos muestra el saludo inicial, y una caja de texto en la cual es necesario acceder el número de identificación que se proporciona cuando se hace un contrato de soporte; es el mismo número que se utiliza cuando se llama por teléfono directamente a los ingenieros del centro de soporte.

Dicho número es enviado a través del http, y se verifica que su contrato correspondiente se encuentre vigente, o bien que su fecha de expiración del contrato sea menor o igual a treinta días naturales. Si cumple esta condición, el cliente puede acceder al repositorio, de lo contrario no permite el acceso. También verifica que el contrato exista, de lo contrario también niega el acceso.

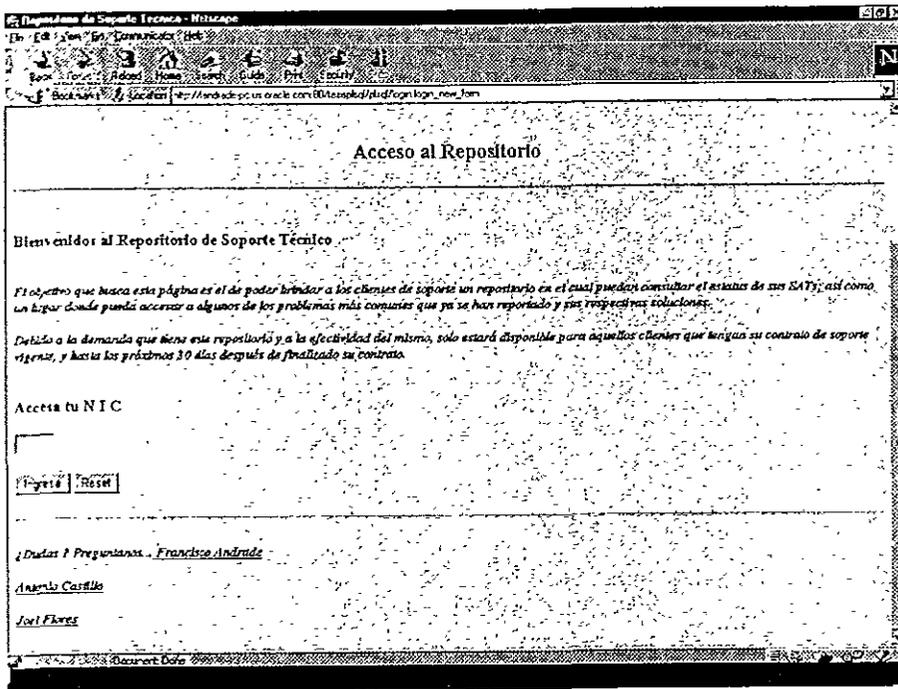


Figura 4.8.1 Pantalla de acceso al repositorio

Después de que el cliente fue validado y se le ha permitido el acceso, la pantalla de la figura 4.8.2 es desplegada. En dicha pantalla tiene las diferentes opciones que puede realizar a través del web.

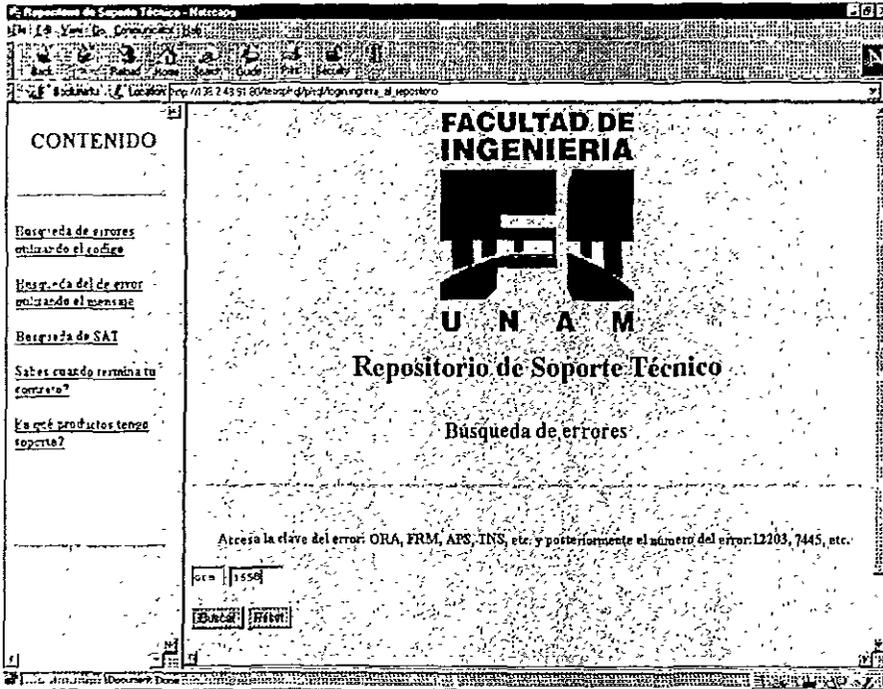


Figura 4.8.3 Búsqueda de errores

Al ejecutar la búsqueda, el sistema presenta el número de SAT y el encabezado de la misma que corresponde con el número de error que se especificó. De esta forma el cliente tiene la opción de revisar varios casos, para apegarse al que le esté ocurriendo en ese momento.

La figura 4.8.4 muestra los resultados de la búsqueda anterior, aquí el cliente puede acceder a los datos correspondientes de cada SAT.



Figura 4.8.4 Resultados de la búsqueda del código de error

En la figura 4.8.5 se muestra el desplegado de una de las SAT, en la cual se puede apreciar el seguimiento que se le da al caso hasta su solución.

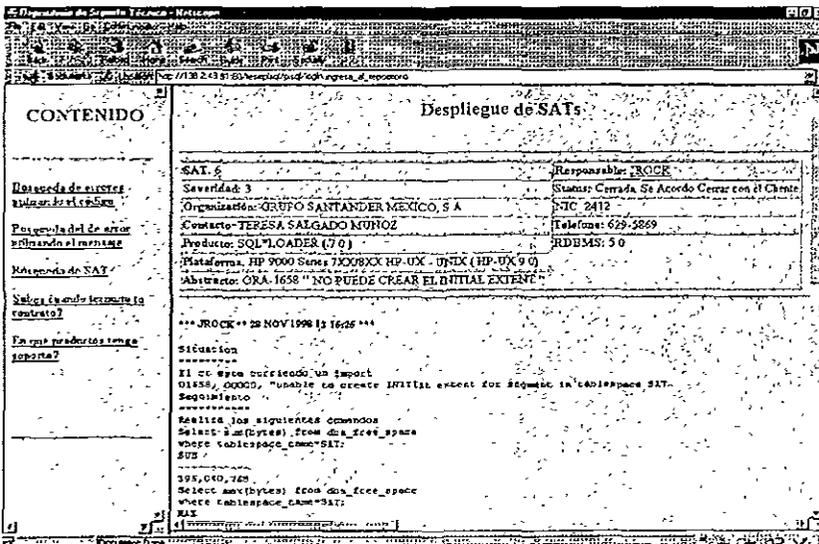


Figura 4.8.5 Despliegue de la información

Sin embargo, no siempre son problemas lo que los clientes reportan al centro de soporte, hay muchos casos en los cuales el cliente solicita determinada información que es vital para la toma de decisiones en una empresa. O bien el cliente no tiene las habilidades necesarias para realizar determinada tarea, la forma más sencilla de solventar estos contratiempos es llamar al centro de soporte.

La opción de Búsqueda del error utilizando el código el mensaje, es una buena opción para buscar información que realmente no es un error.

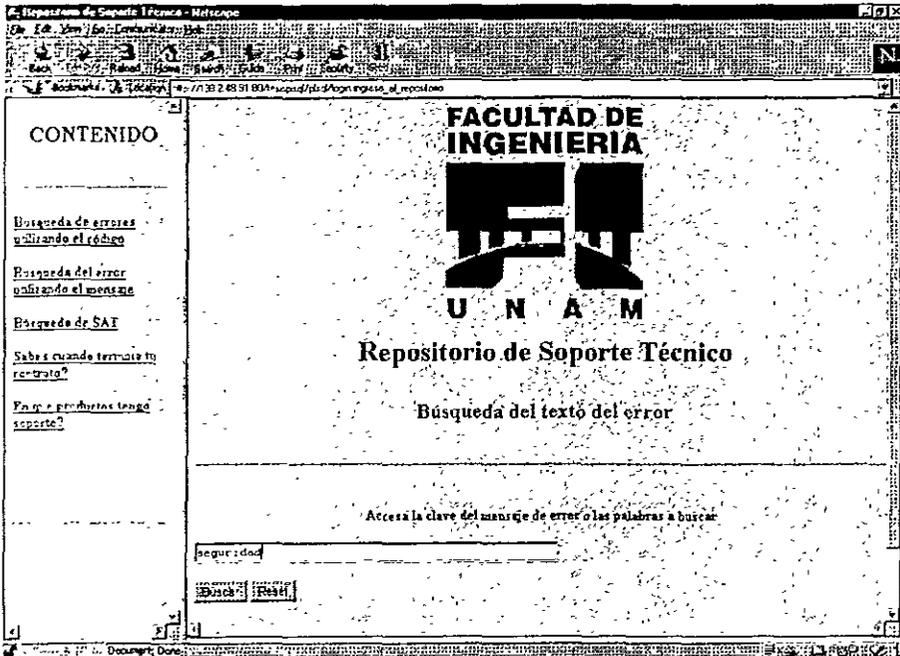


Figura 4.8.6 Búsqueda utilizando el texto del error

De esta forma el cliente puede obtener la información pertinente cuando no se trata de un error, el resultado que obtiene son las SAT que tienen en el encabezado la palabra 'seguridad'.

Las otras dos opciones son muy parecidas, porque las dos utilizan el número de NIC para buscar la información, en base a ellos el sistema hace la búsqueda en las vistas y despliega la información, sin interferir en la integridad de la base de datos.

La cuarta liga permite conocer a ciencia cierta cuando finaliza el contrato de soporte que el cliente tiene, figura 4.8.7. La quinta liga despliega la variedad de productos y las plataformas que fueron contratadas cuando se realizó el contrato de soporte, figura 4.8.8.

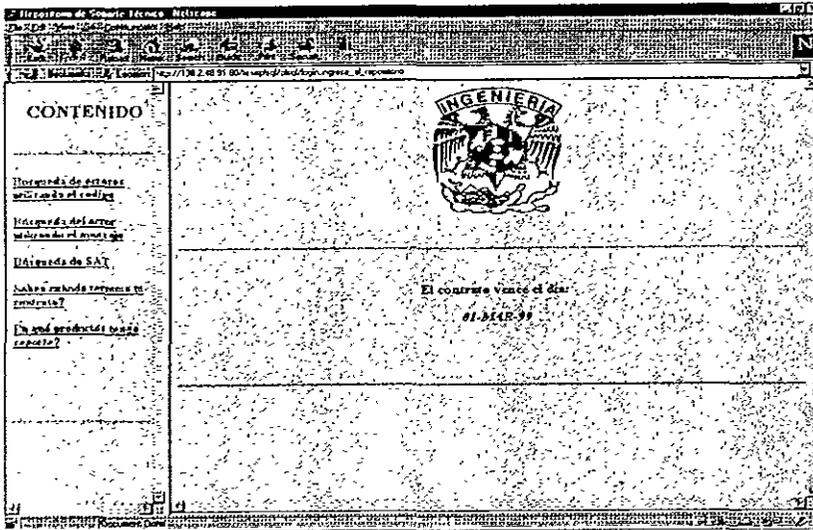


Figura 4.8.7 Fin del contrato de soporte



Figura 4.8.8 Productos contratados

plataforma original en la cual fueron generados, en este caso Windows 95, hacia otro tipo de plataforma, como la unix.

Posteriormente ya con la instalación de Developer 2000 sobre plataforma unix y los correspondientes archivos binarios del sistema se debe de proceder a crear los ejecutables sobre la plataforma unix, recordando que los ejecutables son archivos con extensión fmx y rep, para formas y reportes respectivamente. Este proceso es sumamente rápido y no puede llevar más de dos días

El proceso de generación se puede llevar a cado mediante dos métodos. Uno puede ser generando el reporte en ambiente gráfico y el otro es realizándolo directamente en línea de comandos de unix

Para realizar la generación de los módulos tipo forma en ambiente gráfico es necesario invocar el correspondiente Forms Designer sobre la plataforma unix con el comando:

```
$ f45desm &
```

Abrir la correspondiente forma, acceder el menú de File, la opción de Administration y generate. Automáticamente nos generará el correspondiente archivo ejecutable para la plataforma unix.

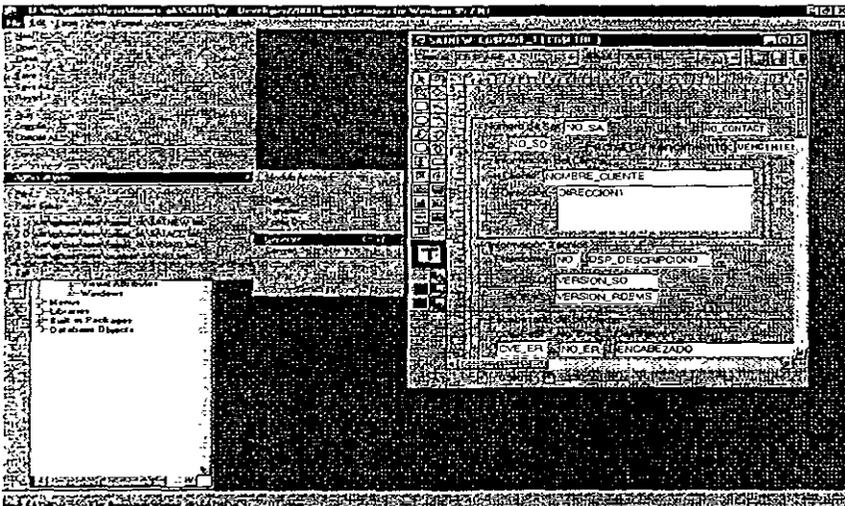


Figura 4.9.3 Generación de archivo ejecutable para la plataforma UNIX, en el caso de Forms Designer

En el caso de Reports Designer, se realizaría el mismo procedimiento. Se tendría que invocar Reports Designer en la plataforma unix con el comando:

```
$ r25desm &
```

Abrir el correspondiente reporte, acceder el menú de File, la opción de Administration y generate. Automáticamente nos generará el correspondiente archivo ejecutable para la plataforma unix.

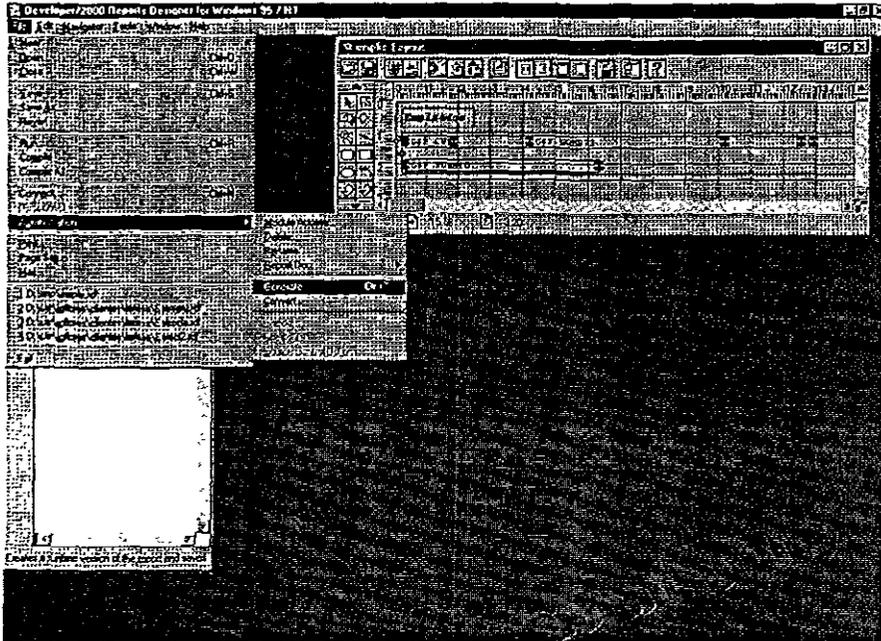


Figura 4.9.4 Generación de archivo ejecutable para la plataforma UNIX, en el caso de Reports Designer

Si deseáramos generar los ejecutable desde la línea de comandos por no contar con las herramientas necesarias para poder generarlo en ambiente gráfico, la generación de las formas se tendría que realizar utilizando el programa f45genm. La línea de ejecución se indica a continuación:

```
$ f45genm useric=usr/psd module=newsat.fmb output_file=ruta/archivo batch=yes
```

Donde en el parámetro *userid* se indica el usuario y password con el cual se conectará a la base de datos para generar la forma. El parámetro *module* para

indicar el archivo binario de forms a partir del cual se generará el ejecutable o el archivo fmx. El parámetro `output_file` en el cual se especifica la ruta y el nombre del archivo si se desea modificar el nombre del archivo ejecutable.

Finalmente el parámetro `batch` para que en la generación nos permita seguir trabajando con la terminal.

En el caso de Reports, se tendría que ejecutar el programa `r25conv` desde el prompt de unix:

```
$ r25conv userid=usr/psd dtype=repfile source=emp.rdf dest=ruta logfile=archivo.log batch=yes
```

Donde en el parámetro `userid` se indica el usuario y password con el cual se conectará a la base de datos para generar el reporte. El parámetro `dtype` para indicar el tipo de archivo que se va a generar, en este caso el valor es `repfile`, ya que deseamos generar el correspondiente ejecutable del reporte. El parámetro `source` para indicarle al programa cual es el archivo binario fuente a partir del cual se realizará la generación del ejecutable. El parámetro `dest` para indicar la ruta en la cual se desea colocar el archivo que se generará. Alternativamente se puede especificar el parámetro `logfile` donde se especifica algún archivo de log donde se puede especificar la salida de algún probable error de generación. Y finalmente el parámetro `batch` para que en la generación nos permita seguir trabajando con la terminal.

En lo relacionado a mantenimiento *perfectivo* es indispensable considerar que las definiciones de cada módulo están contenidos en objetos de la base de datos formando un repositorio, por lo tanto, es factible realizar diferentes versiones de la aplicación, en este caso AMIGS. Si en determinado momento se llegará a determinar una modificación que llevará a perfeccionar la aplicación se puede llegar a generar una nueva versión de la aplicación y comenzar a trabajar sobre ella.

Designer 2000 también tiene la característica de poder realizar ingeniería en reversa tanto de los reportes como de las formas

Manejo de versiones

Para el manejo de versiones se debe de abrir el Repository Object Navigator, acceder el menú de Application y seleccionar la opción de New Versión. Al presionar la opción se presenta la siguiente ventana:

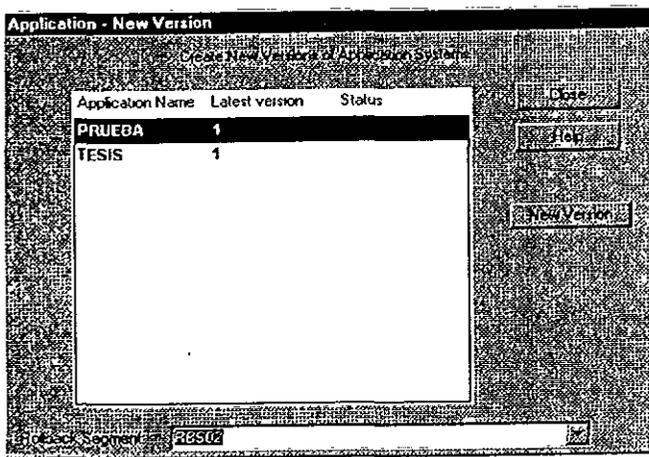


Figura 4.9.5 Manejo de versiones

En donde podemos observar el nombre de la aplicación y la versión actual con la cual se está trabajando. Si deseamos crear una nueva versión de la aplicación es necesario únicamente seleccionar la correspondiente aplicación y presionar el botón de New Version. De esta forma al acceder la ventana nuevamente la versión que estaría activa sería la indicada, tal y como se muestra en la siguiente imagen.

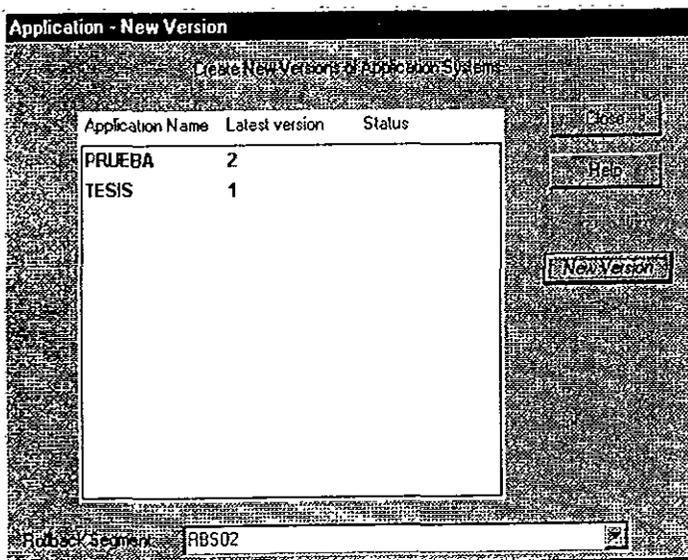


Figura 4.9.5 Cambio en la versión de la aplicación PRUEBA

En este caso la aplicación 1 quedaría congelada. Esto implicaría que sobre la aplicación 1, no se podrían realizar ningún tipo de modificaciones, únicamente en la nueva versión. Si quisieramos averiguar que aplicaciones se encuentran atoradas o paralizadas, debemos accesar la opción de Application y a su vez la de Freeze/Unfreeze, y de esta forma podríamos observar que aplicación se encuentra paralizada. La ventana que se presentaría sería la siguiente:

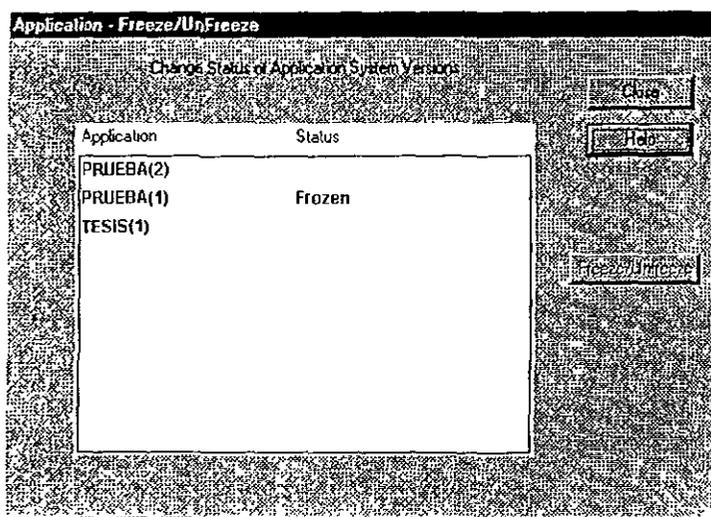


Figura 4.9.6 Pantalla que nos indica que se realizaron cambios a la aplicación PRUEBA.

Todas las modificaciones se empezarán a realizar en la nueva versión. En este caso la versión 2.

En relación a la ingeniería en reversa de las formas y reportes se podría utilizar si en dado caso de que se hallan realizado algunas formas y Reportes directamente en Developer 2000 y dichas definiciones de los módulos se necesitarán agregar al repositorio existente.

Para relizar este proceso de ingeniería en reversa de una forma se requiere ejecutar la opción de Utilities y la opción de Reverse Engineer Form y se indicaría con el botón de browse los archivos fmb correspondientes a las formas que se desearían agregar al repositorio. Se desplegaría la siguiente ventana:

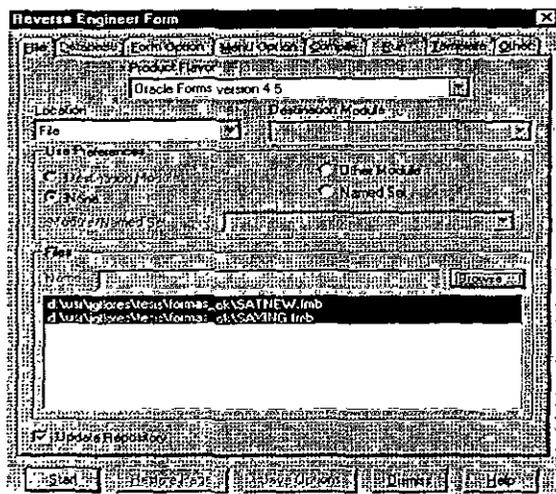


Figura 4.9.7 Utilización de la herramienta Reverse Engineer Form para el caso de una forma

El proceso para la generación del módulo para un reporte es similar. Tendríamos que acceder la opción de Reverse Engineer Report y en el menú de Utilities. Tendríamos, en este caso indicar el correspondiente archivo *rdf* del reporte que deseamos definir como módulo en el repositorio. La ventana que se presentaría sería la siguiente:

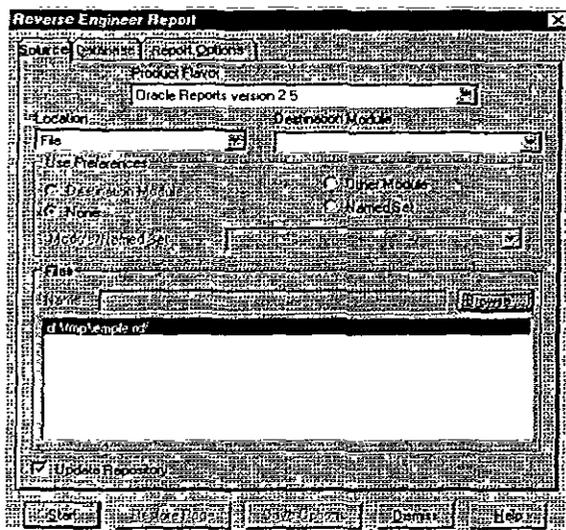


Figura 4.9.8 Utilización de la herramienta Reverse Engineer Form para el caso de un reporte

A partir de estas nuevas definiciones se pueden realizar algunas modificaciones a estos módulos y posteriormente generar la forma definitiva con las correcciones perfectivas ya realizadas.

Dentro del Diagrama Estructural de Módulos, estas nuevas definiciones de módulos pueden ser involucradas para formar parte integral de todo el nuevo sistema

CONCLUSIONES

CONCLUSIONES

En la actualidad se requiere de profesionales dentro del área de la computación para solucionar problemas, proporcionar ventajas específicas a las organizaciones, ayudar a mejorar los desempeños de producción automatizando los centros de trabajo utilizando algún tipo de metodología de información. Este tipo de actividades se ha convertido con el tiempo en un recurso muy valioso para muchas empresas.

En la actualidad, con el avance a nivel hardware, la tecnología ha cambiado de manera substancial los procesos de almacenamiento, la manera de acceder, manejar y compartir la información dentro de las organizaciones. Como resultado del empleo de este tipo de tecnología una gran cantidad de organizaciones han dado un valor agregado a los productos y servicios que venden, además de modificar sus procesos administrativos con el fin de ser más rentables, flexibles, innovativos, ágiles y orientados a cumplir con un aspecto de gran importancia, como es la satisfacción del cliente.

Como parte importante del desarrollo actual dentro de la tecnología de información se encuentran las metodologías y herramientas auxiliares para desarrollar un sistema confiable y que cumpla con la gran mayoría de requerimientos que sean necesarios.

Estudiando las metodologías se puede concluir que es el conjunto de actividades de tal manera que el proceso del desarrollo del sistema sea repetible y medible. Al aplicar una buena metodología se puede garantizar que los resultados serán satisfactorios.

De primera instancia puede parecer que dichas actividades pueden alargar el tiempo de desarrollo del sistema. Estas actividades o procesos no se deben de seguir como una receta. El seguir a detalle todos los pasos de una metodología no garantiza que todo saldrá bien durante el desarrollo del sistema. Se ven involucrados varios factores, como el tiempo disponible, los recursos del hardware y software, etc.

Indudablemente una de las actividades de gran importancia es el involucrar al cliente con el desarrollo del sistema. La manera de hacer participar a los clientes es variada, puede ser utilizando entrevistas, examinando documentos, dejando aportar al cliente sus ideas, analizando flujos de información, etc.

En el desarrollo del sistema fue de gran importancia el desarrollo de prototipos ya que al realizar esta actividad se cubrieron dos aspectos de suma importancia, por una parte involucrar de manera directa al cliente con el desarrollo del producto, hacerle ver de alguna manera el avance obtenido en

cada prototipo desarrollado, de tal manera que tenga conocimiento del trabajo que se está desarrollando, y finalmente que el cliente o usuario final comience a familiarizarse con el sistema.

Dentro de los objetivos que originalmente fueron establecidos se cumplieron básicamente todos.

El sistema finalmente desarrollado cumple con las expectativas de un sistema desarrollado en ambiente GUI, en el cual el usuario es capaz de poder interactuar con el sistema a través del apuntador del mouse, así como objetos de interfaz como botones, menús desplegables, listas de valores y visualizar alertas. Por otro lado la interacción con el usuario es más amigable utilizando la barra de herramientas que cada ventana del sistema presenta.

El objetivo principal de toda metodología es especificar una serie de actividades para poder obtener un resultado satisfactorio, en este sentido es de suma importancia para el desarrollo de sistemas. Al utilizar algún tipo de metodología se pueden tener algunos beneficios como son:

- Disminuir la probabilidad de fallas en el sistema.
- Consistencia, es decir, permite facilitar las actividades de mantenimiento y las posibles mejoras en un futuro.
- Convenios entre los usuarios finales y el equipo de trabajo sobre las actividades a seguir, así como responsabilidades.
- Establecer estándares de productividad a través de herramientas compatibles y conjunto de habilidades

La metodología CASE es bastante completa ya que cumple de manera detallada todas las actividades que se deben de seguir para que un sistema cumpla con todos los requerimientos establecidos. Las actividades a seguir según la metodología CASE son:

- Análisis
- Diseño
- Pruebas
- Implantación

Es indudable que un área de servicio como lo es el área de soporte técnico es muy indispensable dentro de cualquier empresa de desarrollo de software, con lo cual es inevitable la expansión de este servicio y esto implica un rápido crecimiento del área y por lo tanto la implementación de nuevas técnicas de servicio y nuevos procesos de operación de tal forma que el sistema presentado se vea sujeto a mejoras y mantenimientos de manera constante, con tal motivo la metodología CASE y la herramienta Designer 2000 nos permiten tener registrado hasta el momento todos los aspectos estudiados en cuanto a la operación actual y

a partir de éstos poder plantear los tipos de mantenimiento preventivo o correctivo necesarios y los subsecuentes procesos de operación sobre la base de lo desarrollado y documentado hasta el momento. De tal forma se cumple con el objetivo de utilizar una metodología y herramienta capaz de cubrir las necesidades de operación del área de soporte técnico.

Otro de los objetivos planteados fue el desarrollo de un sistema capaz de controlar y mantener las operaciones en buen funcionamiento a partir del 1o. de enero del año 2000. Con lo cual dentro del desarrollo de la aplicación así como a nivel configuración del manejador de Oracle se utiliza el formato especialmente diseñado para manejar el cambio de siglo, dicho formato denominado RR y RRRR, haciendo alusión a los dígitos del año. De tal forma que las operaciones del área de soporte técnico no se verán afectadas por este aspecto.

Finalmente el cliente es capaz, con la interfaz desarrollada para ambiente WEB revisar el estatus de todas las solicitudes de asistencia técnica que hasta ese momento se hallan atendido, sin la necesidad de llamar al área de soporte técnico y preguntar sobre el avance de la SAT. Esta interfaz es de gran ayuda, ya que en algunas ocasiones el analista técnico queda con el cliente de investigar con relación al problema reportado y el analista acuerda en regresar la llamada al cliente con la alternativa de solución. De esta forma cuando el analista llama al cliente con la alternativa en una gran cantidad de casos el cliente no se llega a localizar por algún motivo, de esta forma el cliente cuando se entere de la llamada del analista puede consultar vía WEB y aplicar por si solo la solución documentada por el analista que esta atendiendo la SAT.

Esta interfaz vía WEB también es una fuente de información para todos los clientes con nivel de soporte oro podrán consultar vía WEB la base de datos con el fin de tener acceso a todas las TARs de tal manera que podrían enriquecer sus conocimientos y además percatarse de nuevas alternativas de solución a nuevas situaciones que podrían presentarse al cliente. Teniendo acceso vía este medio las 24 horas del día.

Conforme la tecnología avanza salen al mercado nuevas herramientas de desarrollo y de la misma manera siguen evolucionando las metodologías. En la actualidad es muy común que se generen prototipos de rápida construcción y optimización, de tal manera que se trabaje de manera iterativa con el resto de las fases del sistema.

Hoy en día las herramientas auxiliares para el desarrollo de sistemas ayudan de manera significativa a la creación del sistema de forma eficiente y eficaz utilizando técnicas de ingeniería de información. A este tipo de herramientas se les conoce como herramientas CASE.

Es importante considerar este tipo de herramientas para el desarrollo de sistemas. Son de gran utilidad si se apegan a las necesidades que inicialmente se establecieron y además se debe de llegar a tener un conocimiento de las mismas.

En resumen, los rápidos progresos en tecnología, los cambios dinámicos del mercado, y las demandantes, y aún así cambiantes necesidades de usuario, exigen una innovación de avanzada en los marcos de desarrollo de sistemas. Con modelaje preciso de oportunidades de sistemas y negocios, la refinación de estos modelos a diseños flexibles y el poder provisto por la generación dirigida por modelo.

GLOSARIO

ADP (*Adapter/Dispatcher*). Adaptador /despachador

API (*Aplication Program Interface*). Interfaz para los programas de aplicación

ARCH (*Archiver*). Archivador. Copia los archivos redo log en línea para almacenarlos cuando están llenos.

Back-end. Últimas fases del ciclo de vida del sistema; implantación y mantenimiento

Background. Acción realizada en segundo plano

Buffer. Memoria intermedia.

Cache. Memoria inmediata.

Canvase. Superficie donde se colocan los objetos.

Cartridges. Cartuchos. Ejecutan funciones tales como la autenticación, resolución de nombres, etc. Contienen la aplicación lógica.

CASE (*Computer Aided Systems Engineering*). Ingeniería de sistemas asistida por computadora.

CGI (*Common Gateway Interface*)

Ciclo de vida. Secuencia de estados o fases, desde el análisis de los requerimientos hasta el

mantenimiento, involucrados en el desarrollo del software.

CKPT (*Checkpoint*). Este evento sucede cuando todos los buffers modificados de la base de datos en la SGA son escritos a los datafiles por el DBWR.

Commit. Comando que permite guardar las últimas operaciones de DML a la base de datos.

Control Files. Archivos de control.

DAD. Descripción del Acceso a la base de Datos.

Database buffer cache. Aquí se almacenan los bloques de datos más recientemente usados de la base de datos.

DBMS (*Data Base Management System*). Sistema manejador de base de datos.

DBWR (*Database writer*). Escritura a la base de datos.

DDL (*Data Definitioin Language*). Lenguaje de definición de dato.

Dedicated server Process. Procesos del servidor dedicado.

Desktop computer. Computadora de escritorio.

DFD. Diagrama de Flujo de Datos.

Dispatcher. Despachador (Ver Dnnn)

DML (*Data Manipulation Language*). Lenguaje de manipulación de datos.

Dnnn (*Dispatcher*). Cada proceso dispatcher es responsable de dirigir el requerimiento del proceso de usuario al proceso de servidor compartido disponible.

Drag and Drop. Arrastrar y soltar.

Front-end Fases primarias del ciclo de vida del sistema; análisis y diseño

Gateway. Programa que nos permite establecer comunicación entre dos entidades

GUI (*Graphic User Interface*). Interfaz gráfica de usuario.

Hardware. Parte tangible de un equipo de cómputo (teclado, monitor, etc.)

Instancia. Area de memoria compuesta por la SGA que interactua a través de los procesos de background.

Kernel. Corazón del sistema operativo, manejador de la base de datos.

LCKn (*Lock*). Es utilizado para bloquear la instancia cuando se está manejando parallel server.

LGWR (*Log writer*). El log writer Escribe los cambios hechos a la base de datos en disco.

Listener. Proceso que permite establecer la comunicación entre los clientes y el servidor a través de SQL Net.

Loopback. Prueba de conexión realizada en el propio servidor a través de SQL Net

Modelo relacional. Independencia de la representación lógica de los datos de su almacenamiento físico.

Multi-usuario. Ambiente en el que varios usuarios acceden de manera simultanea a los mismos datos.

Offline Storage Device. Dispositivo de almacenamiento fuera de línea o inactivo.

OLE (*Object Linking and Embeding*). Ligado e inserción de objetos.

ORB (*Object Request Broker*).

Padway. Emulador de terminal UNIX.

Parallel server. Servidor paralelo.

Performance. Prestación, desempeño.

PGA (*Program Global Area*). Area global de programa. Buffer de memoria que contiene datos y controla la información para un proceso servidor

PL/SQL (*Procedural Language /Structured Query Language*) Lenguaje procedural para la escritura de aplicaciones lógicas y la manipulación de datos desde fuera de la base de datos.

PMON (*Process Monitor*). Monitoreo de proceso. Aquí se ejecutan procesos de recuperación cuando un proceso de usuario falla.

Printer. Impresora

Protocolo Conjunto de normas usadas para comunicarse entre los diferentes niveles funcionales de un modelo de interconexión de red.

Prototipo. Versión a escala de algo, construida antes de construir el ítem completo, para asegurar la viabilidad o utilidad de la versión total.
Protocolo.

RECO (*Recoverer*). Es utilizado para resolver transacciones distribuidas que están pendientes debido a una falla del sistema o de la red en una base de datos distribuida

RDBMS (*Relational Data Base Management System*). Sistema manejador de base de datos relacionales

Redo log. Aquí se escriben las modificaciones hechas a la base de datos.

Redo log buffer. Aquí se almacenan los cambios hechos a la base de datos.

Rollback. Comando que permite deshacer las últimas operaciones de DML

SAT. Solicitud de Asistencia Técnica.

Server (*Servidor*) equipo de cómputo con características especiales.

Servidor dedicado Equipo de cómputo conectado en red y que está dedicado a una tarea específica.

Severidad. Nivel de dificultad que podría presentar el problema planteado por el cliente al área de soporte técnico

SGA (*System Global Area*). Área global del sistema.

Shared pool. Porción del SGA que contiene áreas compartidas, por ejemplo el área SQL.

Shared Server Process Procesos compartidos del servidor.

SMON (*System Monitor*). Monitoreo del sistema Ejecuta la recuperación de la instancia, una vez que ésta ha iniciado.

SNMP (*Simple Network Management Protocol*). Protocolo manejador de redes simples.

Software. Parte intangible de un equipo de cómputo (software, programas, etc.)

SQL (*Structured Query Language*). Lenguaje estructurado de consultas.

SQL Net Programa que interactúa con el Listener a través de un protocolo de comunicación para establecer la comunicación entre cliente y servidor

TCP/IP (*Transmission Control Protocol/Internet Protocol*). Protocolo de control de transmisión/Protocolo Internet

Trigger. Disparador, activador, disparo de activación

User. Usuario.

User processes. Procesos de usuario.

UTP (*Unshielded Twisted-Pair*). Par trenzado sin blindaje. Estandar de cableado para uso telefónico y comunicaciones a 8 hilos.

Wizard Hechizo, forma simplificada (por medio de un esquema predefinido) de realizar una tarea.

Workaround Solución alternativa a un problema

Workbench. Banco de trabajo.

WRB (*Web Request Broker*). Capa intermedia el Web Application Server que actúa como autenticador, distribuidor y coordinador de las peticiones a las aplicaciones.

WWW (*World Wide Web*). El ancho mundo de la red

BIBLIOGRAFIA

BARKER Richard & LONGMAN Cliff. CASE*METHOD Function and Process Modelling Addison-Wesley Publishing Company. U.S.A, 1992. 386 páginas.

KENNETH E. Kendall y JULIE E. Kendall. Análisis y Diseño de Sistemas Traducido de la primera edición en inglés de: Systems analysis and design Prentice-Hall hispanoamericana. México, 1991. 881 páginas.

KENNETH E. Kendall y JULIE E. Kendall. Análisis y Diseño de Sistemas. Tercera edición. Prentice-Hall hispanoamericana. México, 1997 913 páginas.

KENNETH E. Kendall & JULIE E. Kendall. Systems Analysis and Design. Third edition, Prentice-Hall, Inc. U S.A, 1995. 894 páginas.

KOCHHAR Neena & KRAMER Debby. Introduction to Oracle: SQL and PL/SQL Using Procedure Builder. Volume One, Volume two, Volume three, Volume four. Oracle Corporation. U S.A. 1996.

LOPEZ-Fuensalida Antonio. Metodologías de desarrollo. Macrobit Editores. México, 1991. 226 páginas.

MCCLURE Carma. CASE, la automatización del software. Addison-Wesley Iberoamericana. U.S.A , 1993. 381 páginas.

MULLER Robert J. Oracle Developer/2000 Handbook. Osborne McGraw. U.S.A.

LEVERENZ Lefty. Oracle7 Release 7.3 Concepts manual. Oracle corporation. U. S. A., 1996.

LONNROTH Magnus. Oracle Web Application Server Handbook. Osborne McGraw. U.S.A., 1998.

LORENTZ Diana. Oracle7 Release 7.3 SQL Reference Manual. Oracle Corporation. U.S.A., 1996.