

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

106
ZEJ

FACULTAD DE INGENIERIA

"SISTEMA EJECUTIVO DE ADMINISTRACION
DE PROYECTOS CON HERRAMIENTAS
CASE DE NATURAL"

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :
FERNANDO FEDERICO VALLE BUENDIA
GABRIEL HERNANDEZ PEREZ
GERARDO LUNA SALAZAR
MIGUEL ANGEL TORRES BARRAGAN
SAUL SANCHEZ MARTINEZ



ASESOR: ING. MARIA EUGENIA RAMIREZ NARVAEZ

CIUDAD UNIVERSITARIA, MEXICO, D. F.,

1995

FALLA DE ORIGEN

**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.

A:
Ma. del Carmen Buendía y Fernando Valle

A mis padres les diría tantas cosas pero sólo les puedo decir **gracias por todo** lo que me han dado y por ser lo que soy. **Gracias...**

A:
Claudia, Gerardo y Antonieta

A mis hermanos **gracias** por su apoyo y por soportarme durante toda su vida.

A:
Dios
Gracias.

A:
Mis compañeros de estudio y Amigos

Con los que pasé buenos y malos momentos.

A:
Mis profesores
Por proporcionarme sus conocimientos para mi vida académica.

Federico

Agradezco a todos los que estuvieron conmigo durante toda mi vida académica; por los consejos que me fueron útiles, a la amistad sincera que me brindaron durante los momentos buenos y malos. Los sabios consejos de mis profesores, amigos y familiares. Gracias.

G.H.P.

AGRADECIMIENTOS

A mis padres

Por todo su cariño y entusiasmo para que cumpliera con esta meta, pero sobre todo por tener la dicha de ser su hijo.

A Mayra

Por todo su apoyo y paciencia durante tanto tiempo.

A Rodriguito

Por la alegría que me despierta.

A Chela

Por todo lo que representa en mi vida.

A Dios

Por darme todo sin siquiera intentar merecerlo.

GERARDO

A mi padre:

Por todo el apoyo brindado en mis estudios. A su buen ejemplo que estuvo presente a cada momento.

A mi madre:

Por su motivación a seguir adelante, sus tiernas palabras, su cariño y consejos.

A mis hermanas:

Por aceptarme tal y como soy.

A Maru:

Sobretudo por su paciencia y confianza brindados.

A Katy y el viejo tío Ernesto:

Por su maravilloso apoyo y comprensión para poder realizar esta tesis.

A Lupita:

Por su linda y sincera cooperación.

SINCERAMENTE MIGUEL ANGEL.

A mis padres:

Por solo haber recibido amor y cariño de ustedes, gracias.

A mis hermanos:

Por apoyarme y estar siempre conmigo.

A mis amigos:

Por todos los buenos momentos que hemos pasado juntos.

SAUL SANCHEZ MARTINEZ

Agradecemos a la Universidad Nacional Autónoma de México (UNAM) por permitirnos ser parte de la excelsa comunidad Universitaria.

De la misma forma agradecemos a la Facultad de Ingeniería por la formación como profesionistas de cada uno de nosotros, a sus académicos.

A nuestra asesora Maria Eugenia, por su tiempo, paciencia y confianza siempre brindados durante el desarrollo del trabajo presente y sus valiosos comentarios.

Fernando, Gabriel, Gerardo, Miguel, Saul.

INDICE

INTRODUCCION

CAPITULO 1 CONCEPTOS TEORICOS

1.1 Teoría de bases de datos	1
1.1.1 Tipos de enfoque de bases de datos	2
1.1.2 Normalización	4
1.1.3 Desnormalización	7
1.2 El ciclo de Vida	9
1.2.1 Ciclo de vida tradicional	9
1.2.2 Modelo del ciclo de vida tradicional	11
1.3 Herramientas CASE	13
1.3.1 Origen y generaciones	13
1.3.2 Historia de CASE	13
1.4 Cambios hacia el ciclo de vida CASE	15
1.5 Componentes de su sistema CASE	17
1.6 Las principales herramientas CASE	18
1.6.1 Juegos de herramientas	19
1.6.2 Bancos de trabajo	20
1.6.3 Asistentes de metodología	20

CAPITULO 2 TECNICAS DE DIAGRAMACION

2.1 Modelo Entidad - Relación	21
2.1.1 Objetivos del modelo Entidad - Relación	21
2.1.2 Conceptos básicos	22
2.1.3 Tipos de notaciones	25
2.1.4 Pasos a seguir para la construcción del modelo entidad Entidad - Relación	26
2.1.5 Formas de definir las llaves de acuerdo al tipo de relación	27
2.2 Modelado de funciones y procesos	27
2.2.1 Objetivos del modelado de funciones y procesos	27
2.2.2 Jerarquía de funciones	27
2.2.3 Modelado de flujo de datos	31

CAPITULO 3 LENGUAJES DE CUARTA GENERACION

3.1 Generación de lenguajes	35
3.2 Productos 4GL	39
3.2.1 PROGRESS	39

3.2.2 FOCUS	42
3.2.3 NATURAL	44
3.3 Natural Architect	46
3.4 Predict	50
3.4.1 Control de la aplicación en predict (PAC)	51
3.4.2 Manejador lógico y físico de estructura de datos	53
3.5 Natural Construct	54
3.6 Adabas	65

CAPITULO 4 METODOLOGIA

4.1 Análisis y diseño estructurado	72
4.1.1 Metodología de análisis	74
4.1.2 Definición del entorno de la aplicación	75
4.1.3 Documentación de análisis	77
4.1.4 Diseño de sistemas	78
4.1.5 Metodología de diseño	79
4.1.6 Estructura del sistema	80
4.1.7 Diseño de base de datos prototipo	81
4.1.8 Prototipos	82
4.1.9 Diseño de base de datos definitivo	83
4.1.10 Diseño de procesos	84
4.1.11 Documentación del diseño	85
4.1.12 Construcción de sistemas	86
4.2 Técnicas de RAD	86
4.2.1 Alta Calidad	88
4.2.2 Proceso de desarrollo RAD	88
4.2.3 Miembros primarios del grupo de usuarios	90
4.2.4 RAD - Desarrollo Rápido de Aplicaciones	91
4.2.5 JAD - Diseño de Aplicaciones en Grupo	92
4.2.6 Prototipos	93

CAPITULO 5. ANALISIS, DISEÑO Y CONTRUCCION DEL SISTEMA

5.1 Antecedentes	96
5.2 Situación Actual	97
5.3 Necesidades de información	101
5.4 Análisis	102
5.5 Documentación del análisis	105
5.6 Documentación del diseño	105
5.7 Construcción de programas , mapas y base de datos	107
5.7.1 Documentación de construcción	108

CONCLUSIONES
APENDICE MANUAL DE USUARIO
BIBLIOGRAFIA

INTRODUCCION

El objetivo de toda empresa es lograr que su cliente quede con un alto nivel de satisfacción de los servicios que le proporciona. Actualmente las necesidades de información son prioritarias, por lo que se busca mejorar el acceso rápido, confiable y oportuno a la información requerida. Para lograr esto nuestra organización en estudio (SAG) se propuso alcanzar un alto nivel del servicio para con sus clientes.

Existen empresas de consultoría que brindan a sus clientes soluciones completas de software y hardware. En base a los requerimientos del cliente se proporciona soporte de análisis y desarrollo de sistemas.

Software AG en su intención de brindar un servicio oportuno y actual al usuario, ofrece una respuesta constante al cliente. Los servicios que ofrece el personal de consultoría mantienen una planeación estratégica, diseño, análisis y diagnósticos en sus componentes, arquitectura y tecnología avanzada, así como una adecuada implantación de sus sistemas operativos, software de base y apoyo, también supervisa las instalaciones de hardware.

Por estos motivos, se ha tomado la decisión de diseñar y desarrollar un sistema que permita a los ejecutivos interpretar con mayor facilidad los datos obtenidos para poder llevar un control de todos sus clientes y áreas involucradas. De esta manera los ejecutivos podrán consultar con mayor rapidez la información de sus clientes y tomar decisiones en base a la información consultada.

El tipo de información que se busca automatizar es la relacionada con el control de clientes y sus respectivos proyectos.

El sistema busca:

- Mantener informada a la presidencia sobre la situación de las cuentas y los proyectos.
- Información concentrada de cuentas y proyectos para consulta en línea.
- Proporcionar los elementos informativos necesarios para las juntas ejecutivas con el cliente.

Para lograr el desarrollo e implantación de este software se utilizó un conjunto de herramientas integradas CASE optimizando tiempo y recursos humanos con la finalidad de obtener un sistema efectivo y con posibles adecuaciones a un corto plazo en base a nuevos requerimientos.

Los temas que se contemplan en esta tesis se dividen en cinco capítulos:

El capítulo I trata conceptos relacionados con bases de datos, ciclo de vida del software y CASE.

En el capítulo 2 se habla de las técnicas de diagramación más importantes usadas por cualquier organización o industria.

En el capítulo 3 se describen las herramientas CASE utilizadas para la elaboración del sistema. NATURAL ARCHITECT es una herramienta auxiliar para el análisis y diseño de aplicaciones. PREDICT, utilizado para definir la base de datos, el diccionario de datos, la relación entre estructuras, la documentación y los programas que controlan el acceso a la base de datos. NATURAL CONSTRUCT genera aplicaciones en NATURAL y utiliza PREDICT para obtener información de los campos y vistas utilizadas, así como dar mantenimiento a aplicaciones previamente creadas.

Se mencionan algunas características de lenguajes de cuarta generación como PROGRESS y FOCUS. Se describe el manejador de la base de datos utilizado por el sistema (ADABAS).

En el capítulo 4 se describe la metodología utilizada para la elaboración del sistema, la cual es conocida como Análisis y Diseño Estructurado aunado con las técnicas de RAD y JAD.

El capítulo 5 se enfoca al análisis, diseño y construcción del sistema, presentando además la documentación generada por CASE.

Por último se presenta el manual del usuario, el cual contiene una explicación del funcionamiento de los módulos que componen el sistema.

CAPITULO 1

CONCEPTOS TEORICOS

En este capitulo se tratarán conceptos relacionados con el manejo y desarrollo del sistema; entre ellos bases de datos, ciclo de vida del software y herramientas CASE.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. Realmente el software ha pasado de ser una resolución de problemas especializada y herramienta de análisis de la información, a una industria por sí misma.

1.1 TEORIA DE BASES DE DATOS

Una base de datos es:

Una colección de datos interrelacionados almacenados en un conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una o más aplicaciones; los datos son independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y modificar o extraer los datos almacenados.

El enfoque de bases de datos nos permite:

- Controlar la redundancia.
- Mantener la consistencia.
- Lograr la integración de los datos.
- Compartir los datos entre las diferentes aplicaciones.
- Cumplir con los estándares.
- Tener facilidad en el desarrollo de aplicaciones.
- Uniformar los controles de seguridad, privacidad e integridad.
- Independencia entre los datos y los programas.
- Reducir el mantenimiento a los programas.

Un sistema de manejo de bases de datos (DBMS) consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos. El conjunto de datos se conoce comúnmente como base de datos. Esta contiene información acerca de una organización determinada.

CAPITULO 1

CONCEPTOS TEORICOS

En este capitulo se tratarán conceptos relacionados con el manejo y desarrollo del sistema; entre ellos bases de datos, ciclo de vida del software y herramientas CASE.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. Realmente el software ha pasado de ser una resolución de problemas especializada y herramienta de análisis de la información, a una industria por sí misma.

1.1 TEORIA DE BASES DE DATOS

Una base de datos es:

Una colección de datos interrelacionados almacenados en un conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una o más aplicaciones; los datos son independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y modificar o extraer los datos almacenados.

El enfoque de bases de datos nos permite:

- Controlar la redundancia.
- Mantener la consistencia.
- Lograr la integración de los datos.
- Compartir los datos entre las diferentes aplicaciones.
- Cumplir con los estándares.
- Tener facilidad en el desarrollo de aplicaciones.
- Uniformar los controles de seguridad, privacidad e integridad.
- Independencia entre los datos y los programas.
- Reducir el mantenimiento a los programas.

Un sistema de manejo de bases de datos (DBMS) consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos. El conjunto de datos se conoce comunmente como base de datos. Esta contiene información acerca de una organización determinada.

OBJETIVOS DE UN DBMS

- No tener datos repetidos y no almacenar datos derivados.
- Garantizar la consistencia de los datos. Obtener la misma información por peticiones similares en un momento dado.
- Integridad de los datos. Las reglas dictadas por políticas o normas de la empresa y que los datos deben cumplir.
- Seguridad de los datos. La protección de los datos contra accesos, modificaciones o pérdidas ya sea en forma intencional o no intencional.
- Controlar la concurrencia. Múltiples usuarios pueden acceder a la misma información al mismo tiempo, sin que con ello tengan problemas con los datos.
- Proteger los datos contra fallas del sistema. La capacidad de restaurar la integridad y consistencia después de errores en el sistema.
- El diccionario de datos. Es la capacidad que da el manejador de la base de datos de poder tener la descripción de los datos que están almacenados en la base de datos.

Dentro del ambiente de bases de datos se han identificado ya ciertos tipos de roles de personas involucradas en el desarrollo, mantenimiento, diseño y uso de las bases de datos. Entre ellas los programadores, analistas, soporte técnico y usuarios finales.

El administrador de la base de datos (DBA) es la persona o grupo responsable del control del DBMS. Esto implica:

- Definir el esquema conceptual es decir, decidir que los datos serán controlados por el DBMS, y contruir así el modelo de datos.
- Definir el esquema interno. Crear las estructuras físicas de almacenamiento de forma que aseguren un acceso eficiente a los datos.
- Conocer el negocio en contacto con usuarios finales y detectar necesidades.
- Definir la estructura de seguridad. Propietarios de datos, dar autorización de accesos y establecer procedimientos de auditoría.
- Definir estrategias de respaldo y recuperación.
- Monitorear la eficiencia del DBMS.

1.1.1 TIPOS DE ENFOQUE DE BASES DE DATOS

Los modelos de bases de datos que existen son:

- Modelo jerárquico
- Modelo de red.
- Modelo relacional.

ENFOQUE JERARQUICO.

La estructura lógica en la cual se sustenta la base de datos jerárquica es el árbol. Un árbol se compone de un nodo raíz y varios nodos sucesores ordenados jerárquicamente.

Cada nodo representa una entidad (tipo de registro) y las relaciones entre entidades son las conexiones entre los nodos.

El nodo colocado en la parte superior es llamado padre y los nodos inferiores son los hijos.

En el sistema jerárquico, las conexiones entre archivos no dependen de la información contenida en ellos; se definen al inicio y son fijos.

La característica sobresaliente de este modelo es el manejo de la conexión uno a muchos, entre un padre y varios hijos, en otras palabras, cada hijo solo tiene un padre. El procesamiento es top-down, navegacional.

DESVENTAJAS EN EL ENFOQUE JERÁRQUICO

- No modela sencillamente las relaciones muchos a muchos.
- Anomalías de inserción.
- Anomalías de borrado.
- Anomalías de actualización.
- Se pueden dar consultas inconsistentes.

ENFOQUE DE RED

Los datos se representan como registros ligados formando un conjunto de datos interconectados.

La base de datos de red a diferencia de las jerárquicas, permite cualquier conexión entre entidades, es decir, se pueden representar relaciones de muchos a muchos. En una red, un hijo puede tener varios hijos y varios padres a la vez.

DESVENTAJAS EN EL ENFOQUE DE RED

- Resulta difícil definir nuevas relaciones.
- Es complicado darle mantenimiento ya que cualquier cambio en la estructura requiere una descarga en los datos.
- Representa desperdicio de recursos.
- Anomalías de inserción.
- Anomalías de borrado.

ENFOQUE RELACIONAL

La estructura lógica de una base de datos relacional esta basada en la representación de entidades mediante tablas, las cuales constan de columnas (campos) y renglones (registros). Las relaciones entre tablas se llevan a cabo a través de un conjunto de columnas que se tengan en común, logrando una conexión dinámica entre un número ilimitado de ellas a través del contenido de esas columnas.

La ventaja de los sistemas relacionales es el poder modificar la información sin preocupación de especificar las combinaciones entre registros.

CARACTERÍSTICAS

- Diccionario de datos integrado.
- Flexibilidad: mantenimiento de estructuras y de datos , en el tipo de consultas.
- La navegación es responsabilidad del DBMS, no del programador.
- Todas las consultas son posibles.

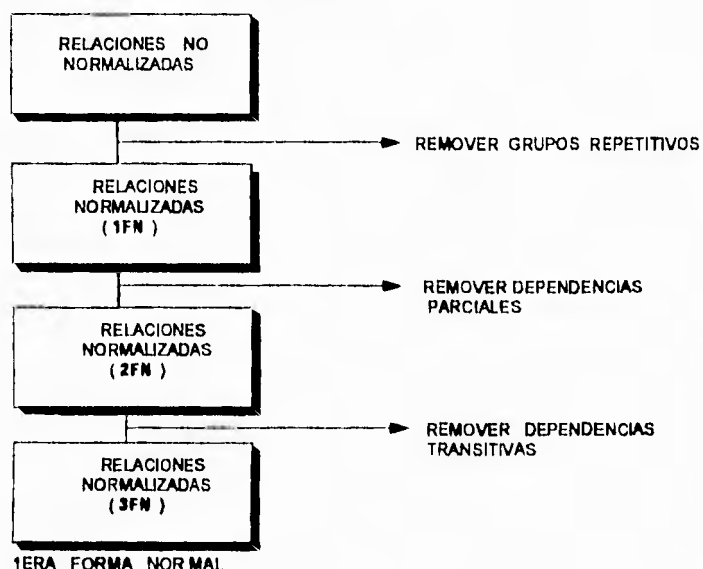
1.1.2 NORMALIZACION

Técnica para asegurar que los datos analizados estan relacionados de la mejor manera por medio de chequeos y balances para encontrar las llaves de acceso mas cortas y eliminar la redundancia de datos, asegurando la integridad de los mismos reduciéndolos a su estructura más simple.

Pasos:

- 1) Identificar relaciones críticas.
- 2) Transformar cada entidad en una tabla.
- 3) Definir llaves candidatas (primarias y alternas).
- 4) 1a forma normal.
- 5) 2a forma normal.
- 6) 3a forma normal.
- 7) Dibujar E-R (diagrama entidad-relación) normalizado.

PASOS DE LA NORMALIZACION



1ERA FORMA NORMAL

ELIMINAR GRUPOS REPETIDOS

NO. DEPTO	NOMB-DEPTO	DIV1	DIV2	DIV3	DIV4
-----------	------------	------	------	------	------

SE VUELVE :

NO. DEPTO	NOMB-DEPTO
-----------	------------

NO. DEPTO	No DIV
-----------	--------

2DA. FORMA NORMAL

DEPENDENCIA FUNCIONAL

CADA ATRIBUTO DEBE DEPENDER DE TODA LA LLA VE

NO.PARTE	NO.PROVEEDOR	NOMB-PROV	PRECIO
----------	--------------	-----------	--------

SE VUELVE :

NO.PROVEEDOR	NOMB-PROV
--------------	-----------

NO.PARTE	NO.PROVEEDOR	PRECIO
----------	--------------	--------

3ERA. FORMA NORMAL

DEPENDENCIA FUNCIONAL

UN DATO QUE NO ES LLAVE IDENTIFICA A OTRO DATO

NUM-EMP	NOMB-EMP	SALARIO	NUM-PROY	DIA-TERMINACION
---------	----------	---------	----------	-----------------

SE VUELVE :

NUM-EMP	NOMB-EMP	SALARIO	NUM-PROY
---------	----------	---------	----------

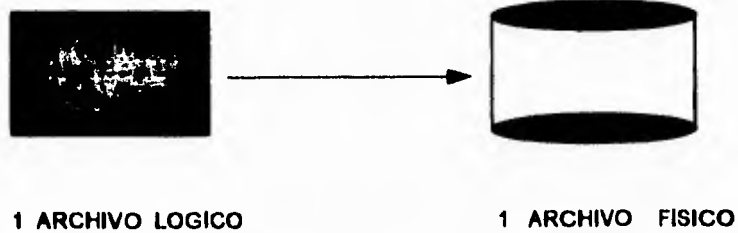
NUM-PROY	DIA-TERMINACION
----------	-----------------

1.1.3 DESNORMALIZACION

Paso contrario a la normalización, surge ante la necesidad de mejorar el desempeño y es parte del diseño de la base de datos.

Una vez obtenido el modelo normalizado se tiene la necesidad de desnormalizar por ejemplo en el modelo jerárquico padres con un solo hijo son posibles de absorberse. Ante la necesidad de esta acción se tienen beneficios pero con ciertas desventajas:

ARCHIVO DESNORMALIZADO



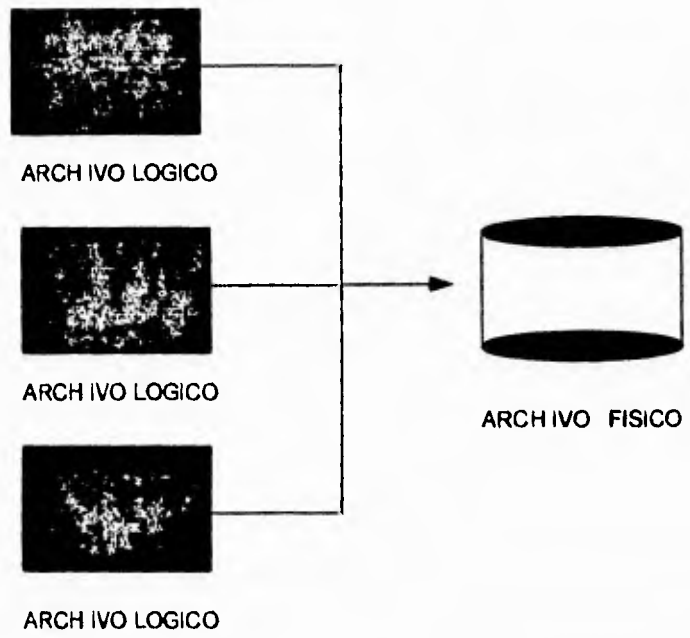
Ventajas:

- Relaciones simples
- Diseño muy flexible
- Pocas restricciones
- Registros muy cortos

Desventajas:

- Llamadas a ADABAS(DBMS que usa el sistema.)
- Sobrecarga en memoria
- Duplicidad de datos

ARCHIVO NORMALIZADO



Ventajas:

- Poca duplicidad de datos
- Poca sobrecarga
- Menos llamadas

Desventajas:

- Mayores grupos periódicos
- Programación muy compleja

1.2 EL CICLO DE VIDA

El ciclo de vida del software puede ser visto como una serie de transformaciones. Primero, se definen los requerimientos conocidos como especificación del sistema de estos se obtiene el diseño; después se transforman en programas y finalmente estos se convierten en instrucciones de máquina.

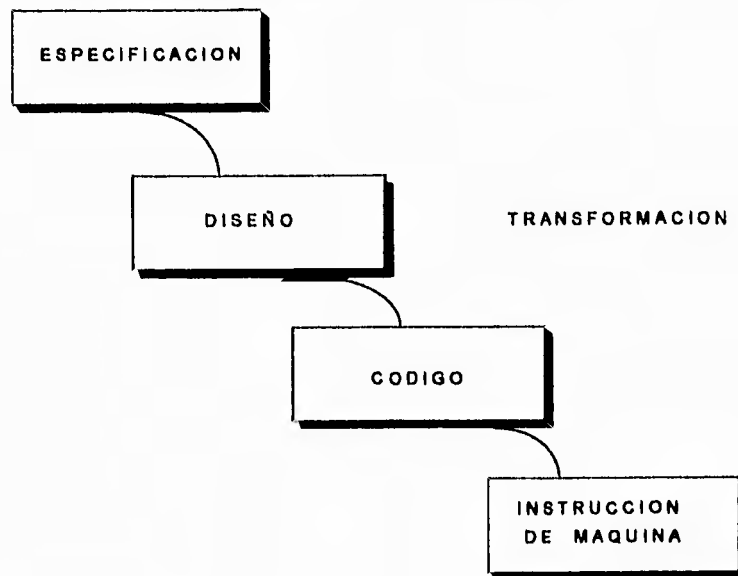


fig. 1.1

1.2.1 CICLO DE VIDA TRADICIONAL

Es un concepto fundamental en la ingeniería del software, describe la secuencia de pasos para el desarrollo, uso y mantenimiento de los sistemas. En esencia, es una técnica de manejo para mantenimiento y desarrollo de software estructurado.

El ciclo de vida consiste de cinco fases básicas:

- Análisis
- Diseño
- Codificación
- Prueba
- Operación y mantenimiento.

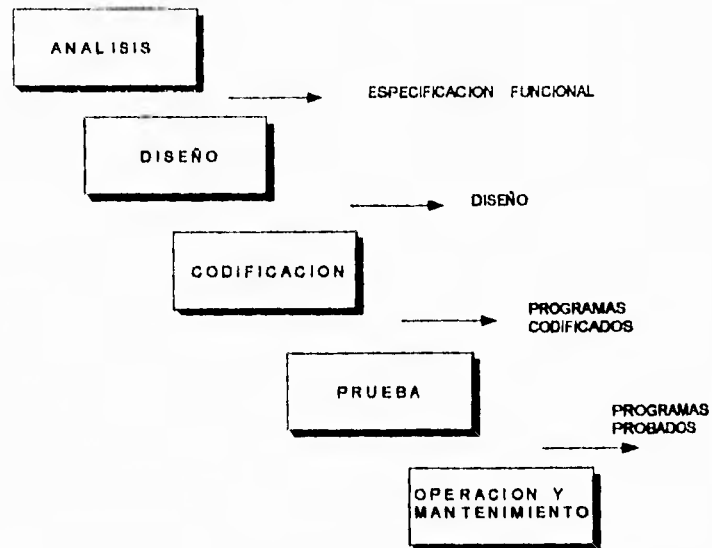


fig. 1.2

Análisis es el proceso para definir requerimientos y dar solución a un problema. En esta etapa es necesario que los usuarios sean cuestionados y después definir las propiedades que el sistema debe tener para que las necesidades sean cubiertas. La salida en esta fase es la especificación del sistema en la cual se establecen las propiedades requeridas de una manera precisa y formal.

Diseño. En esta fase se planea la construcción del sistema; esto es, determinar los datos y procedimientos necesarios y como estos deberán ser integrados para dar una solución óptima. Los algoritmos son desarrollados para describir como se manejarán tanto datos como procedimientos.

La especificación del sistema es usada como entrada en la etapa de diseño.

Codificación es el proceso de transformar el diseño en instrucciones para la computadora. El objetivo es producir programas eficientes y correctos. Los módulos codificados conjuntan una unidad de prueba que es desarrollada durante este paso.

Prueba. Se demuestra que un sistema satisface los requerimientos de un problema y funciona correctamente para todas las posibles entradas de datos. El objetivo es encontrar condiciones de programa no consideradas, fallas y descubrir cualquier implantación

incorrecta de los requerimientos. Los módulos se integran en grupos cada vez más grandes y se prueban hasta que el sistema sea verificado totalmente.

Operación y mantenimiento. Se ejecuta el sistema en un modo de producción, manteniéndolo en buen funcionamiento. Involucra:

- Corrección de errores y desajustes que no fueron descubiertos durante el desarrollo.
- Modificar el sistema y adaptar los cambios en el ambiente de operación, mejorando su eficiencia y calidad. Todos los recursos y actividades requeridos se llevan a cabo asegurando que el software se mantenga en condiciones aceptables.

1.2.2 MODELO DEL CICLO DE VIDA TRADICIONAL

El modelo del ciclo de vida del software describe las acciones y decisiones requeridas para definir, desarrollar, probar, descubrir, operar y mantener un sistema. El modelo tradicional se deriva del modelo hardware, el cual incluye las fases de requerimientos, diseño, fabricación prueba y operación/mantenimiento.

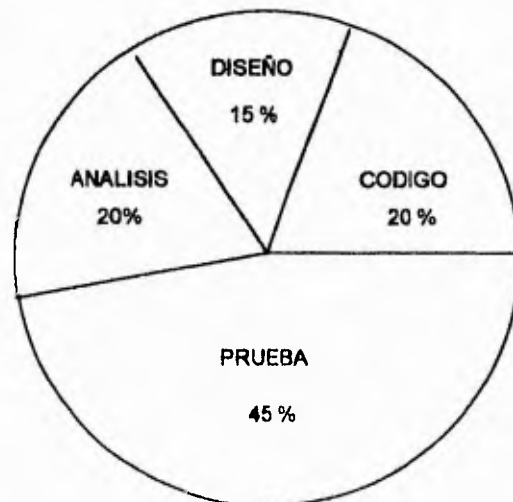


Fig 1.3

Como se muestra en el ciclo de vida tradicional se enfatizaba en las últimas etapas del desarrollo de software con un 65 por ciento del ciclo total.

Esta distribución generó sistemas especificados pobremente y de un alto costo, menos del 35 por ciento se dedicaba al análisis o diseño.

Definir requerimientos resultaba una tarea difícil, no eran estables. En la práctica el análisis era repetitivo.

Cuando fue descubierto que los errores más costosos ocurrían en etapas de desarrollo, se enfatizó más dando un 60 por ciento del ciclo al análisis y diseño, en base a la introducción de técnicas estructuradas de las cuales parte fundamental esta en tomar el tiempo estimado para definir requerimientos cuidadosamente e investigar el mejor diseño antes de llegar a los detalles de implantación.

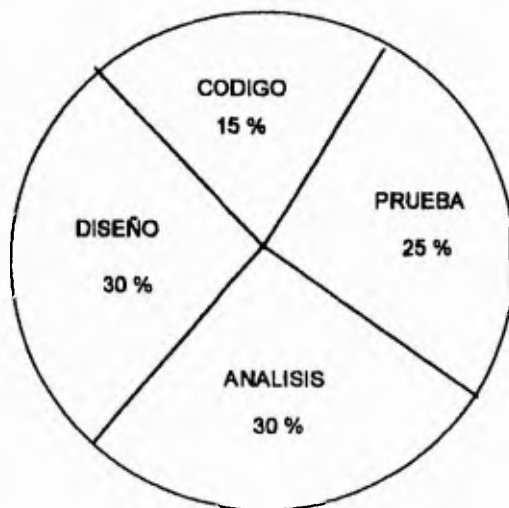


Fig. 1.4

1.3 HERRAMIENTAS CASE

1.3.1 ORIGEN Y GENERACIONES

La historia de la tecnología CASE comienza a principios de los años ochenta con la introducción de la documentación asistida por computadora y de las herramientas de diagramación, las cuales representan los primeros intentos para automatizar el análisis y diseño de tareas. Su propósito era producir automáticamente la documentación estructurada requerida por las distintas metodologías de desarrollo estructurado.

Este cambio en las herramientas de software es parte de una nueva tecnología denominada **Computer Aided Software Engineering** ó **CASE** (ingeniería de software asistida por computadora). Es una tecnología para automatizar el desarrollo y mantenimiento del **software**, siendo además una respuesta muy práctica a los últimos 25 años de crisis del **software**. Realmente CASE no es totalmente nueva ya que está construida sobre las técnicas estructuradas que se desarrollaron en la década de 1970.

1.3.2 HISTORIA DE CASE

1a. GENERACION (84 - 86)	Herramienta para diagramación simple sobre PC/XT.
2a. GENERACION (87 - 89)	Herramienta de diagramación apoyado en métodos, generación de documentación y generación de código sobre PC/AT y workstation nivel proyecto.
3a. GENERACION (90 - ?)	Herramientas múltiples integradas a través de un repositorio sobre PC/386 y workstation a nivel empresa.

EVOLUCION DE LA TECNOLOGIA CASE

Principios de los 80	Ayuda de computadora para documentación. Ayuda de computadora para diagramación. Herramientas de análisis y diseño.
Mediados de los 80	Chequeo automático de análisis y diseño. Almacenamiento de diagramas estructurados en bibliotecas automatizadas (repositorio).

Finales de los 80	Generación automática de código. Ligar el diseño automatizado a la programación automatizada.
Principios de los 90	Manejo de metodologías inteligentes. Habilitación de interfases con el usuario. Reutilización como metodología de desarrollo

En sus inicios las herramientas CASE se enfocaron principalmente a la automatización de la documentación para mejorar la productividad del software. A mediados de los años ochenta, las herramientas CASE se mejoraron para proporcionar dos funciones muy importantes:

- Comprobación automática de diagramas estructurados.
- Almacenamiento de diagramas estructurados en librerías de diseño automático llamadas diccionarios, depósitos o enciclopedias.

La base de CASE está en el diseño automático, porque el análisis y el diseño se consideran como las fases iniciales del ciclo de vida del software.

El paso siguiente en la evolución de la tecnología CASE fue la unión del diseño automático con la programación automática. Mientras que el diseño automático cubre las tareas de análisis y diseño, la programación automática significa generadores automáticos de código. La unión implica que del 80 al 90 por ciento del sistema del software puede generarse a partir del diseño de diagramas estructurados.

El diseño automático lo soportan las herramientas CASE basadas en computadoras personales desarrolladas durante la primera mitad de la década de los años ochenta. La programación automática la proporcionan los generadores de aplicaciones y de código de la cuarta generación, la mayoría de los cuales están basados en mainframes. Para unir el diseño y la programación automática se requiere un puente entre dos entornos hardware (computadoras personales - mainframes) y dos tecnologías de software (CASE y lenguajes de cuarta generación). Una parte muy importante para esta unión será añadir un repositorio al entorno CASE.

El repositorio CASE es un mecanismo para almacenar y organizar toda la información sobre un sistema de software. Esta información es necesaria para la gestión del proyecto, y para la generación automática de código. Incluye información del problema que se va a resolver, el dominio del problema, los procesos del software que están siendo utilizados, los modelos de procesos, prototipos, recursos del proyecto, y del contexto organizativo.

Además de la unión entre el diseño automático y la programación automática, el repositorio CASE convierte por primera vez, el concepto del software reutilizable en algo práctico. Los componentes del software reutilizables son la clave del vertiginoso aumento

de la productividad del software. En lugar de ir creando cada vez nuevo software o de mejorar el dudoso, se pueden utilizar los componentes del software reutilizable almacenado en el repositorio CASE. No solamente se podrán reutilizar los módulos de código fuente sino que también podrán ser usados los planes de proyectos, modelos de prototipos, modelos de datos y especificaciones de diseño.

La CASE ha cambiado radicalmente la forma de construir los sistemas software 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.
- 2) La automatización de muchas tareas de desarrollo y mantenimiento del software.
- 3) Una programación visual proporcionada por potentes estándares gráficos.

La meta de la tecnología CASE es automatizar todo el ciclo de vida del software mediante un conjunto de herramientas de software integradas. No obstante, para el estado tecnológico actual se trata de una meta a conseguir, no de una realidad.

La tendencia de CASE es que el desarrollo y el mantenimiento del software debería verse como una actividad formal y disciplinada capaz de una comprobación mejor de la exactitud y automatización del proceso. La única forma real para incrementar significativamente la calidad y la productividad del software es a través de la automatización.

1.4 CAMBIOS HACIA EL CICLO DE VIDA CASE

Los cambios y mejoras de la tecnología CASE en el desarrollo y mantenimiento de software son variados. En la siguiente tabla se muestran las diferencias entre el desarrollo de software tradicional y el de CASE a través de la automatización.

Desarrollo Tradicional	Automatización del software
Enfasis en codificación y prueba	Enfasis en análisis y diseño
Especificación basada en papel	Uso de prototipo interactivo
Codificación manual	Generación automática de código
Documentación manual	Generación automática
Pruebas de software	Chequeo automático de diseño
Mantenimiento del código	Mantener especificaciones de diseño

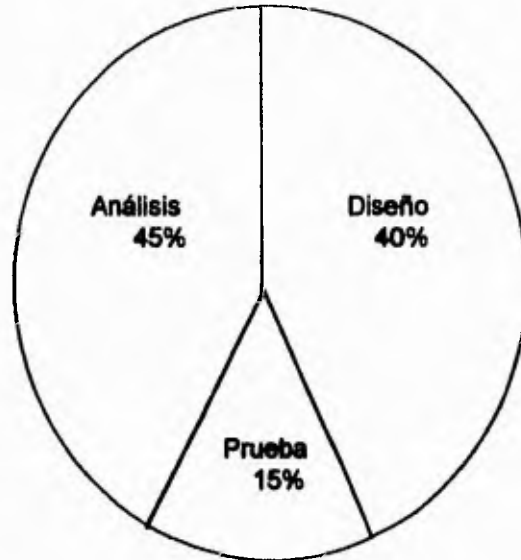


Fig. 1.5

Como se muestra en la figura, los desarrolladores se enfocan más a las etapas de análisis y diseño que a las de codificación o prueba. La codificación manual es sustituida por los generadores de código CASE que producen de un 80 a 100 por ciento del código del sistema en base a especificaciones de diseño, las cuales son estructuras de datos, entidades de datos y procedimientos.

Estas especificaciones son creadas interactivamente en la computadora por analistas de sistemas y son introducidas en librerías automáticas CASE para poder ser utilizadas después si es necesario.

La tecnología CASE cambia la metodología de análisis de sistemas tradicional. Se crea un prototipo que reemplaza a los métodos basados en papel de usuario. Las herramientas CASE tales como diseñador de pantallas y reportes o lenguajes de especificación ejecutables pueden ser usados para construir modelos prototipo de sistemas, permitiendo así determinar los requerimientos del usuario.

CASE reduce enormemente la prueba de software, el análisis y chequeo automático, y por ende elimina errores. A su vez el mantenimiento se simplifica con librerías de diseño y los generadores de código.

Finalmente, entra el concepto de reutilizable en el desarrollo de software; esto es, código fuente, modelos de planes de proyectos, modelos prototipo, modelos de datos y especificaciones de diseño, los cuales pueden ser reusados por los desarrolladores.

1.5 COMPONENTES DEL SISTEMA CASE

Podemos dividir un sistema CASE en tres componentes básicos:

- 1) Front-end (Upper Case - o - Case de alto nivel)
- 2) Repositorio
- 3) Back-end (Lower Case - o - Case de bajo nivel)

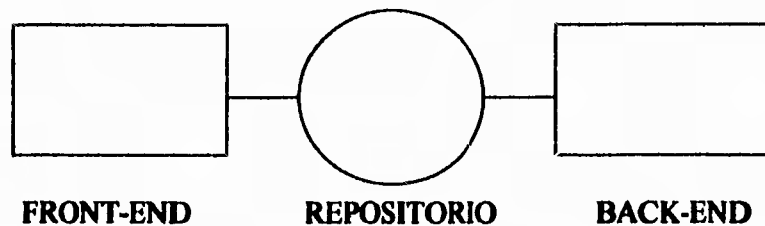


Fig. 1.6

El componente **front-end** corresponde a las fases primarias del ciclo de vida del software; es decir, el análisis y el diseño. El front-end puede corresponder a la parte del computador 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. Estas actividades se realizan mucho mejor en una computadora personal dedicada con tiempo de respuesta rápida, mapa de bits gráficos de alta resolución y ratón.

El componente **back-end** corresponde a las últimas fases del ciclo de vida del software; es decir, la implantación y el mantenimiento del programa. El back-end también corresponde a la porción del computador principal en la plataforma de hardware Case. Las herramientas del back-end CASE automatizan el código, las comprobaciones, la generación de las bases de datos, la normalización de los datos y el análisis del impacto en el sistema existente. Estas tareas requieren la potencia y capacidad de almacenamiento de una minicomputadora o computadora principal.

El repositorio 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 software se gestiona y comparte. Por medio del repositorio el trabajo del sistema en los diferentes equipos se puede combinar, analizar y consolidar en una representación del sistema consistente, en progreso y completa.

Lógicamente, el repositorio CASE está dividido en librerías de proyectos y en modelos del sistema. Físicamente, está dividido en niveles que no corresponden con las plataformas de hardware del sistema CASE. A nivel computadora personal existe un repositorio local para soportar el desarrollo individual. A nivel computadora principal existe un repositorio para mantener toda la información corporativa. A nivel de departamento o de proyecto hay un repositorio intermedio para guardar la información.

Debe existir un enlace entre todos los repositorios anteriores para que los datos puedan cargarse y descargarse entre los niveles.

Lo anterior lleva a considerar en la conectividad entre los sistemas hardware lo siguiente:

- Selección de los datos a descargar
- Carga y consolidación de los datos
- Necesidades de reconstrucción del formato de los datos
- Control y seguridad de los datos
- Velocidad
- Eficiencia
- Acceso multiusuario a los archivos
- Copias de seguridad (backup)

1.6 LAS PRINCIPALES HERRAMIENTAS CASE

Las diferentes herramientas CASE se enfocan hacia el soporte de diferentes fases del ciclo de vida del software o al desarrollo de diferentes tipos de sistemas software. Para distinguir hay tres categorías básicas de herramientas CASE:

- 1) Juegos de herramientas CASE.
- 2) Bancos de trabajo CASE.
- 3) Asistentes de metodologías CASE.

Los juegos de herramientas (toolkits) son el tipo más simple de herramientas CASE. Son un conjunto integrado de herramientas que automatizan un tipo de tarea del ciclo de vida del software, como el diseño del sistema, el mantenimiento del programa o un tipo de trabajo, como el análisis del sistema.

Los bancos de trabajo CASE se componen de un conjunto de herramientas integradas que automatizan las tareas a lo largo de todo el ciclo de vida del software. Son entornos de propósito general para el desarrollo del software soportando la totalidad del rango del trabajo para el desarrollo de software.

Un asistente de metodología CASE proporciona ayuda por computadora para una particular metodología de desarrollo del software, como el análisis estructurado de De Marco, el diseño estructurado de Jackson o la ingeniería de la información de Martin.

1.6.1 JUEGOS DE HERRAMIENTAS CASE

Los juegos de herramientas CASE enfocan el soporte de una fase en particular del ciclo de vida del software o de un tipo de sistema de software. Pueden clasificarse, además, según el entorno hardware y del sistema operativo en el que se procesan, la línea del producto al que pertenecen, si tienen arquitectura abierta o cerrada y según la metodología que soportan.

Por ejemplo, unos se procesan en PC con MS-DOS o PS/2. Otros en estaciones de trabajo o computadoras principales.

Dentro de las fases del ciclo de vida del software se encuentran los siguientes juegos de herramientas:

1) De análisis los cuales hacen más fácil seguir los principios de un buen análisis y diseño de sistemas al hacerse cargo automáticamente del pesado trabajo burocrático y de los innumerables detalles del sistema. El objetivo es automatizar la creación de las especificaciones de un sistema describiendo las necesidades del mismo.

Tiene cuatro componentes básicos:

- Las herramientas de diagramación estructurada. Un conjunto de herramientas informatizado para el dibujo, manipulación y almacenamiento de los diagramas estructurados.
- Las herramientas de prototipos. Se emplean para ayudar a determinar las necesidades del sistema y responder a las cuestiones sobre la capacidad de rendimiento del sistema antes de construirlo.
- El repositorio. Toda la información de las especificaciones del sistema se almacenan en este. Es realmente el corazón del CASE y base para la integración de las herramientas, el control de la consistencia y la integridad de las especificaciones del sistema, documentación y reutilización del software.
- El comprobador de especificaciones. Comprobación automática proporcionada por las herramientas CASE.

2) De diseño. Soportan el diseño lógico y físico de los archivos y de las bases de datos así como el modelado lógico de los datos, la conversión automática de los modelos de datos a tercera forma normal, la generación automática de esquemas de bases de datos para un sistema gestor en particular y la descripción de archivos a nivel de programa en código.

3) De programación. El propósito es obtener un programa documentado y probado que satisfaga todas las especificaciones.

Dentro de estas herramientas una muy importante es el generador de código. Este posibilita la producción automática del código desde el diseño de programación. Incrementa la productividad sobre la escritura manual, los prototipos se soportan mediante generadores de pantallas, además de la generación de código compilado y estructurado, la reducción de costos en mantenimiento.

4) De mantenimiento. Dominan el ciclo de desarrollo de software. El propósito es corregir, cambiar, mejorar, analizar, validar, convertir y modificar la especificación de los sistemas existentes.

1.6.2 BANCO DE TRABAJO CASE

Un banco de trabajo CASE (workbench) es un conjunto de herramientas integradas cuya función es automatizar el desarrollo y mantenimiento de los sistemas, además de la gestión de las actividades de un proyecto. Contiene las herramientas que integran las fases básicas del desarrollo de software, a saber: análisis, diseño e implantación. Las herramientas se integran de forma que la salida de una fase del ciclo de vida pase directa y automáticamente a la fase siguiente. El producto final de un banco de trabajo CASE es un sistema de software ejecutable acompañado de su documentación.

Se clasifican según el proceso en distintos entornos de hardware y sistemas operativos, soportando diferentes metodologías de desarrollo y de distintos tipos de sistemas de salida.

1.6.3 ASISTENTES DE METODOLOGIA CASE.

Un asistente de metodología CASE puede ser tanto un juego de herramientas como un banco de trabajo que estructura el proceso de desarrollo del software de acuerdo con las fases y reglas de una metodología estructurada en particular. La información sobre la metodología se incorpora por medio de los paneles de ayuda, del menú de opciones, de la introducción de funciones y de los controles de calidad.

La clasificación es de acuerdo al entorno hardware o sistema operativo, la metodología soportada, el nivel de soporte y el tipo de aplicación.

CAPITULO 2

TECNICAS DE DIAGRAMACION

El modelado ha sido aceptado siempre por ingenieros, científicos, artistas y contadores como una técnica invaluable para presentar ideas, ayudar a la comprensión y aún para predecir nuevas formas para realizar las cosas además de permitir conocer sus limitaciones.

Las buenas técnicas de modelado son basadas en rigurosos estándares y convenciones para poder eliminar la ambigüedad además de ayudar a la comunicación. Estas técnicas son aplicables a las necesidades de información de cualquier organización o industria, en el presente capítulo se tratarán las más importantes.

2.1 MODELO ENTIDAD-RELACION

El modelo Entidad-Relación es una técnica para especificar un esquema empresarial y permitir facilitar el diseño de bases de datos, además de proporcionar fundamentos sólidos para el desarrollo apropiado de sistemas de datos de alta calidad para cubrir las necesidades de las organizaciones.

En su forma más simple, el modelo Entidad-Relación involucra la identificación de los objetos importantes en una organización (Entidades), las propiedades de estos objetos (Atributos) y como se relacionan unas con otras (Relaciones).

2.1.1 OBJETIVOS DEL MODELO ENTIDAD-RELACION

Proporcionar un modelo adecuado de la información necesaria para la organización, el cual actuará como un esqueleto para el desarrollo de nuevos sistemas o para sus mejoras.

Proporcionar un modelo independiente de cualquier tipo de plataforma, formas de almacenamiento, métodos de acceso a la información, además de permitir la toma de decisiones objetivas sobre las técnicas de implantación y su coexistencia con sistemas en funcionamiento.

PORQUE ES IMPORTANTE EL MODELO ENTIDAD-RELACION

Durante los últimos años los sistemas de cómputo han llegado a ser cada vez más sofisticados, y a medida que este progreso ha ocurrido resultaba ya imposible minimizar la duplicidad y asegurar la integridad de los datos además de una real disponibilidad de la información. Esto debido en gran parte a la incapacidad de entender en su totalidad las necesidades de los usuarios que sobrepasaban la visión, habilidad, experiencia y herramientas que podían aplicarse a medida que los sistemas de información fueron también creciendo.

El moderno modelado Entidad-Relación en conjunto con sofisticadas herramientas CASE proporcionan un efectivo y oportuno medio para definir y controlar la información necesaria. Estas técnicas pueden utilizarse para proporcionar soluciones para sistemas de cómputo convencionales, relacionales, de red y jerárquicos, como se indica en la figura 2.1.

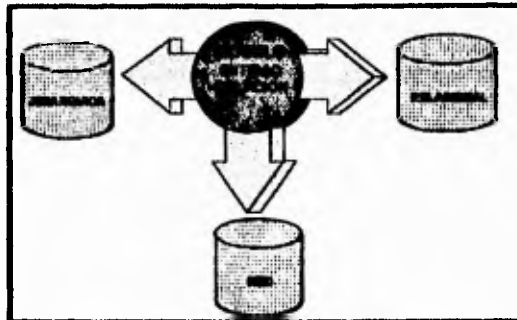


Figura 2.1

2.1.2 CONCEPTOS BASICOS

Los componentes del modelo Entidad-Relación y su simbología los mencionaremos a continuación.

ENTIDADES: Cualquier persona, lugar, cosa o evento del que se recopila información y es de interés para el sistema.

Notación:



TIPOS DE ENTIDADES

Físicas:

- Tangibles.
- Fáciles de comprender.
- (Seres vivos, equipo, edificios, mercancías, etc.)

Conceptuales:

- No tangibles.
- Menos comprensibles.

- Definidos en términos de otros tipos de entidades.
- (Organizaciones, acuerdos, abstracciones, etc.)

CARACTERISTICAS DE LAS ENTIDADES


- Se deben de poder identificar por separado sus ocurrencias.
- Cada una por si misma debe jugar un papel importante en el sistema bajo estudio.
- Puede describirse con dos o más atributos.

ATRIBUTOS: Información que describe las características de una entidad : color, valor, tamaño, etc. .

CARACTERISTICAS DE LOS ATRIBUTOS

- Deben describir a una entidad.
- No usar el nombre de la entidad.
- Identificador único.
- Nombres en singular.

RELACIONES: Representación de una asociación entre entidades.

Notación : (Flecha) 

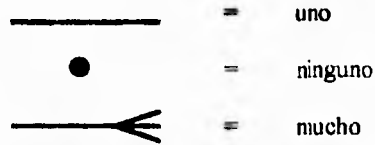
CARACTERISTICAS DE LAS RELACIONES

- Contiene información de la forma como se relacionan las entidades.
- Comparten atributos.
- Son multidireccionales
- Se identifican por una llave

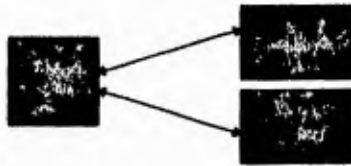
PROPIEDADES DE LAS RELACIONES

Cardinalidad : Número de ocurrencias de cada entidad.
(Uno, ninguno, muchos)



Notación :



Grado : Número de entidades que conectan.



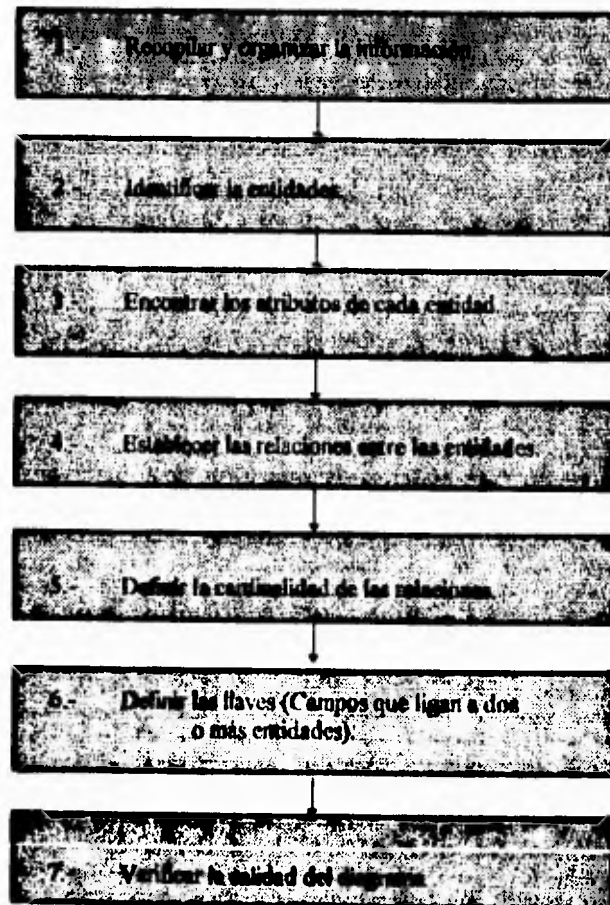
Opcionalidad : Nos indica la obligatoriedad de que una relación se asocie con otra mediante la relación.

- **Mandatorias** Notación : 
- **Opcionales** Notación : 

2.1.3 TIPOS DE NOTACIONES

	Chen/Bachman	Martin	I.R.M.
1 a 1			
1 a M			
1 a 0 o 1			
1 a 0 o M			
M a M			

2.1.4 PASOS A SEGUIR PARA LA CONSTRUCCION DEL MODELO ENTIDAD-RELACION



2.1.5 FORMAS DE DEFINIR LAS LLAVES DE ACUERDO AL TIPO DE RELACION

- Uno a uno : Se determina la dirección de la relación para que la llave del lado dominante pase como alterna al otro lado.
- Uno a muchos : Se añade la llave de lado uno a la del lado muchos.
- Muchos a muchos : Si la relación va en una dirección se toma la llave origen y se copia a la nueva, si va en ambas se rompe y se crea una entidad unión.

2.2 MODELADO DE FUNCIONES Y PROCESOS

El modelado de procesos es un conjunto de técnicas para describir que es lo que hace la organización. Estas actividades son muy útiles para ayudar a determinar las descripciones de las tareas y estructuras de la organización, y definir los requerimientos funcionales para sistemas nuevos o la revisión de los mismos que nos permitan conocer sus necesidades.

En su forma más simple, el modelado de procesos involucra la identificación de que hace la organización (funciones), que es lo que dispara estas actividades (eventos) y que objetos de importancia (entidades) o propiedades de estos objetos (atributos) están actuando sobre estas funciones.

2.2.1 OBJETIVOS DEL MODELADO DE FUNCIONES Y PROCESOS

Los objetivos principales de realizar un modelo de procesos o funciones de una organización son:

- Proporcionar un modelo adecuado de las necesidades funcionales de la organización, el cual actuará como un esqueleto para el desarrollo de sistemas nuevos o mejoras en los actuales.
- Establecer un modelo que es independiente de cualquier mecanismo o método de proceso, permitiendo realizar decisiones objetivas sobre las técnicas alternativas de implantación y coexistencia con sistemas existentes.

COMPLEJIDAD

El modelado de procesos requiere del uso de una variedad de técnicas como el diagrama de flujo de datos y el diagrama de jerarquía de funciones, algunas veces utilizadas en combinación, para cubrir las diferentes formas en las que una organización debe trabajar. Cada técnica es aplicable en diferentes circunstancias.

2.2.2 JERARQUIA DE FUNCIONES

La más sencilla y útil técnica para el modelado funcional es producir una jerarquía de estas, donde cada función es descrita mediante una simple sentencia sin ambigüedad

comenzando con un verbo. La jerarquía es similar a un árbol genealógico o a una estructura organizacional. Cada función padre en la jerarquía es descrita con mayor detalle por sus funciones hijas. Esta técnica permite entender que hace ahora la organización o que necesita hacer para el futuro.

CONSTRUCCION DE UNA JERARQUIA DE FUNCIONES

El modelado de la jerarquía de funciones se realiza adecuadamente produciendo la jerarquía desde una perspectiva de arriba-abajo y un chequeo con una perspectiva de abajo-arriba.

MODELADO DE ARRIBA-ABAJO

Este modelado empieza definiendo la esencia de la organización como un todo (objetivo o misión de la organización) apartir de esto enumerar las funciones que lo componen, este proceso se realiza repetidamente por niveles.

En este proceso el analista se debe de realizar las siguientes preguntas :

¿Qué funciones son necesarias para logra este objetivo?

¿Se necesita algo más que la función padre deba realizar?

Si es así agregar la función a la jerarquía.

¿Existe alguna función hijo la cual no es necesario que realice la función padre?

Si es así coclocar la función donde si lo sea.

Este proceso de chequeo de abajo-arriba consiste en tomar cada una de las funciones de la jerarquía y revisar que se encuentre en un lugar conveniente de la jerarquía desde una perspectiva abajo-arriba.

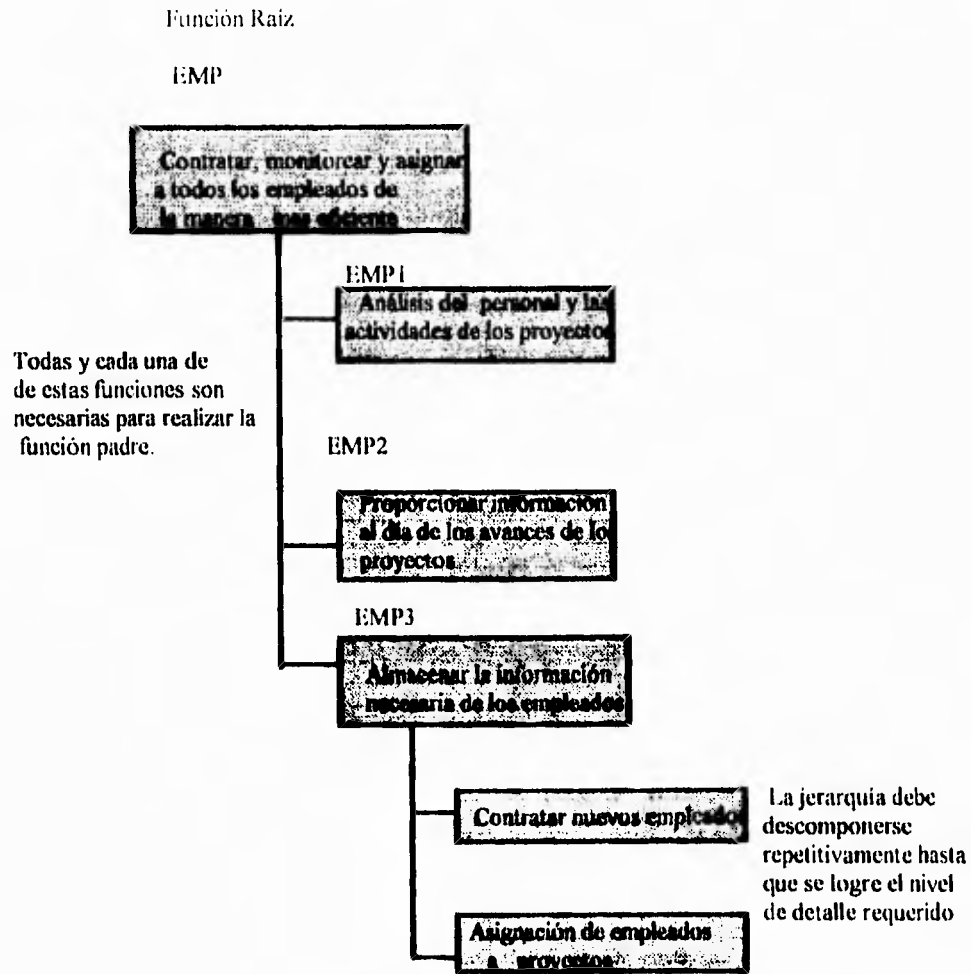
VENTAJAS

- Rápida y fácil de hacer y aprender
- El 80% se obtiene fácilmente.
- Buen medio de presentación para niveles de jerarquía altos

DESVENTAJAS

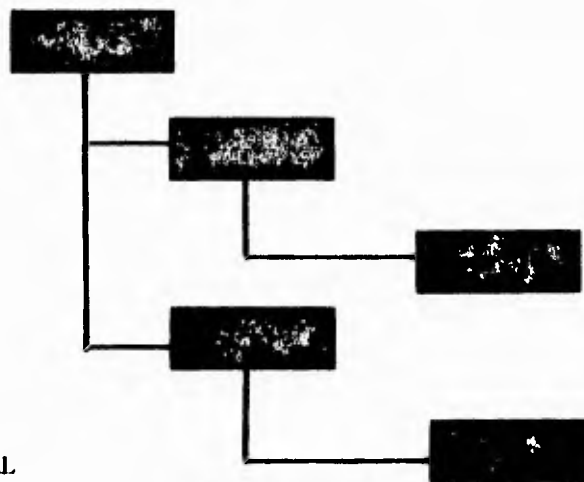
- Dificil de validar su calidad.
- No tiene buena presentación para jerarquias largas

ANATOMIA DE UNA JERARQUIA DE FUNCIONES

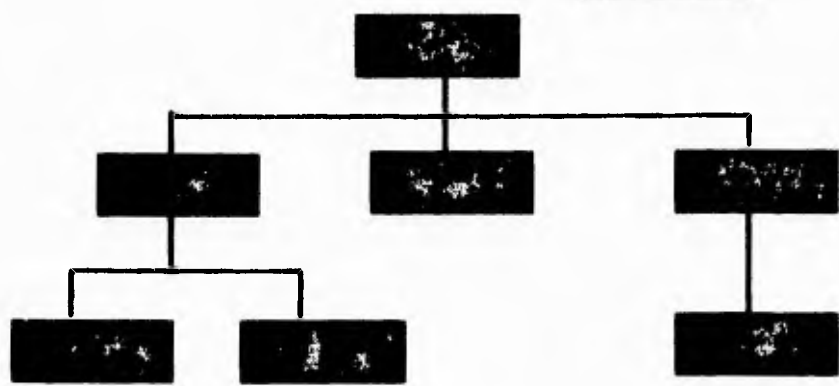


NOTACIONES

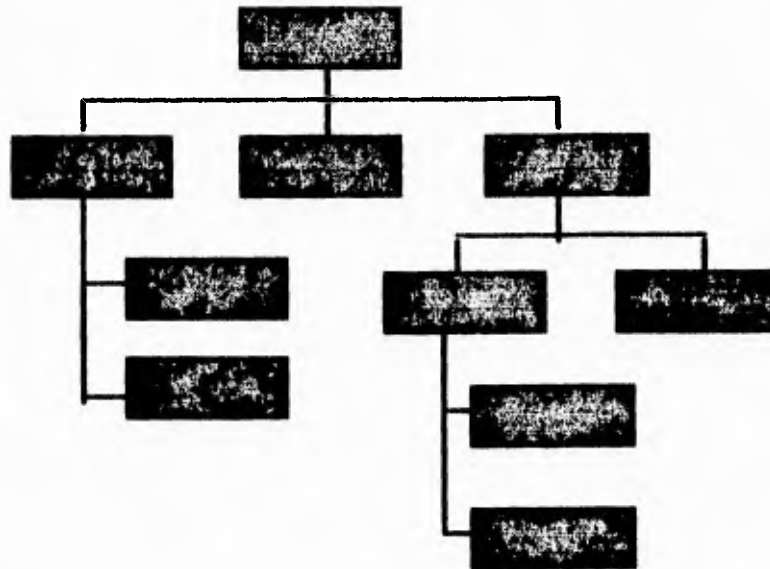
VERTICAL



HORIZONTAL



HIBRIDO



2.2.3 MODELO DE FLUJO DE DATOS

La interdependencia entre funciones puede ser mostrada por otra técnica llamada Modelado de flujo de datos. Frecuentemente las funciones no pueden ser realizadas hasta que la información esté disponible de alguna otra fuente.

El modelo de flujo de datos muestra la interdependencia entre funciones por la definición real o virtual de flujos de información entre funciones. Esta técnica es particularmente importante cuando se requiere modelar un proceso complicado para la organización.

El diagrama de flujo de datos es un diagrama gráfico que representa dentro de un sistema las actividades, los depósitos de información, los flujos de información y las entidades externas que generan o reciben los datos. Muestra también la transformación que se realiza sobre los datos de entrada para producir los datos de salida.

CARACTERISTICAS

Muestra gráficamente :

- El contexto del sistema.
- La interfase entre los componentes del sistema.
- El uso de la información entre las actividades.
- Las fuentes externas.

- Lo que hace el sistema.
- Modela el flujo de información de cualquier sistema, ya sea manual o automatizado.

USOS DEL DIAGRAMA DE FLUJO DE DATOS

El diagrama de flujo de datos muestra los procesos dentro de un sistema y la forma en que intercambian la información entre ellos así como el movimiento de datos tanto de entrada como de salida del sistema mismo.

El modelado de flujo de datos ha sido utilizado por décadas en el área de sistemas y es adecuado para ayudar a entender situaciones existentes o modelar nuevas donde el flujo y los procesos prescritos deben ser agregados.

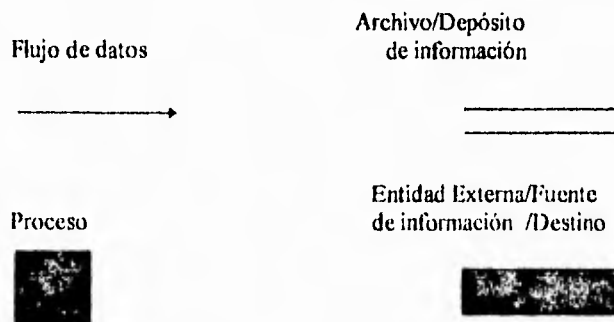
VENTAJAS DEL MODELO DE FLUJO DE DATOS

- Muestra la estructura del sistema como un todo.
- Adecuado para el desarrollo de sistemas de arriba-abajo sin perder detalle.
- Es útil cuando se requiere una presentación del sistema para su comprensión.

DESVENTAJAS DEL MODELO DE FLUJO DE DATOS

- Pobre para el manejo de sistemas complejos.
- Inadecuado para mantener una jerarquía balanceada cuando los requerimientos cambian rápidamente.

SIMBOLOGIA EMPLEADA PARA LOS COMPONENTES DE UN DIAGRAMA DE FLUJO DE DATOS



- Lo que hace el sistema.
- Modela el flujo de información de cualquier sistema, ya sea manual o automatizado.

USOS DEL DIAGRAMA DE FLUJO DE DATOS

El diagrama de flujo de datos muestra los procesos dentro de un sistema y la forma en que intercambian la información entre ellos así como el movimiento de datos tanto de entrada como de salida del sistema mismo.

El modelado de flujo de datos ha sido utilizado por décadas en el área de sistemas y es adecuado para ayudar a entender situaciones existentes o modelar nuevas donde el flujo y los procesos prescritos deben ser agregados.

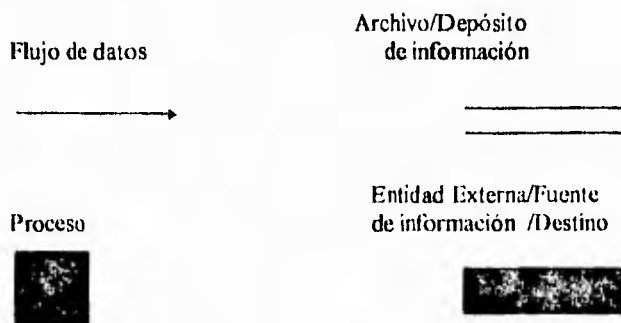
VENTAJAS DEL MODELO DE FLUJO DE DATOS

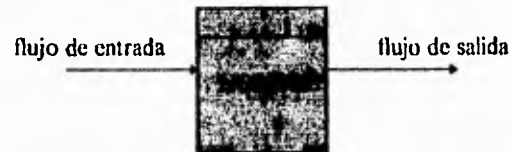
- Muestra la estructura del sistema como un todo.
- Adecuado para el desarrollo de sistemas de arriba-abajo sin perder detalle.
- Es útil cuando se requiere una presentación del sistema para su comprensión.

DESVENTAJAS DEL MODELO DE FLUJO DE DATOS

- Pobre para el manejo de sistemas complejos.
- Inadecuado para mantener una jerarquía balanceada cuando los requerimientos cambian rápidamente.

SIMBOLOGIA EMPLEADA PARA LOS COMPONENTES DE UN DIAGRAMA DE FLUJO DE DATOS



PROCESO

Es usado para representar una función o proceso.
 Transformación que se realiza con los datos de entrada para producir la salida. Se descompone en un conjunto de procesos en el siguiente nivel del diagrama de flujo de datos.

FLUJO DE DATOS

- Traza el flujo de la información a través de los procedimientos del sistema.
- Representa como se conectan o relacionan los procesos.
- Portan elementos de datos de un lugar a otro.
- Se descomponen en un conjunto de atributos o en un conjunto de flujos de datos de nivel inferior.

TIPOS DE FLUJOS

- **Simple**

R1-Flujo

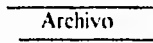
$R1\text{-Flujo} = \text{dato1} + \text{dato2} + \text{dato3}$
- **Compuestos**

C-Flujo

$C\text{-Flujo} = R1 + R2$
 $R1 = \text{dato1} + \text{dato2} + \text{dato3}$
 $R2 = \text{dato4} + \text{dato5} + \text{dato6}$

Los flujos de datos describen los datos reflejados en el diagrama Entidad-Relación.

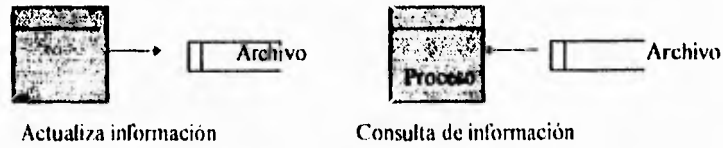
ARCHIVO O DEPOSITO DE INFORMACION



Representa :

El almacenamiento de información.
Un archivo lógico.

Se descompone en un conjunto de atributos.



AGENTE O IDENTIDAD EXTERNA

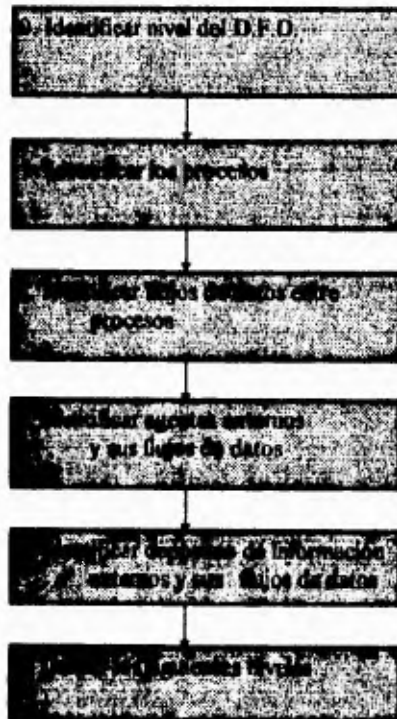
- Deben de indicarse en la parte externa del sistema a modelar.
- La interfase entre el sistema y la parte externa se hace por medio de flujos de datos. Ningún análisis y diseño pueden cambiar las políticas de las entidades externa.

NOTACIONES

	Yourdon/De Marco	Gane & Searson
Proceso		
Flujo de datos		
Archivo		
Entidad Externa		

PAGINA DUPLICADA

35

PASOS PARA ELABORAR LOS DIAGRAMAS DE FLUJO DE DATOS

CAPITULO 3

LENGUAJES DE CUARTA GENERACION (4GL)

3.1 GENERACION DE LENGUAJES

A través del tiempo se ha tenido un desarrollo muy importante en el campo del software y existen cientos de lenguajes de programación que han sido aplicados en uno u otro momento en algún esfuerzo serio de desarrollo de software. La evolución del software se define en las siguientes etapas:

PRIMERA GENERACION DE LENGUAJES

La primera generación de lenguajes nace por el surgimiento de la primera computadora que estaba basada en la tecnología de tubos de vacío y memoria de cilindro magnético, la principal aplicación de esta computadora era la de instrumento de cálculo. No contaban con sistema operativo y los lenguajes que se utilizaban para programar esta computadora eran lenguajes de máquina y ensambladores primitivos.

SEGUNDA GENERACION DE LENGUAJES

La segunda generación de lenguajes también cuenta con los avances en la tecnología de computadoras, que estaban hechas a base de transistores y ferritas, su principal aplicación era la de procesamiento de datos e instrumentos de cálculo. Estas computadoras contaban con un sistema operativo rudimentario que controlaban periféricos, iniciaban y terminaban tareas, los lenguajes que se utilizaban para programar eran lenguaje ensamblador, primeros compiladores de FORTRAN I, ALGOL 58, Flowmatic, IPL V, etc., que han servido como base de todos los lenguajes de programación modernos.

TERCERA GENERACION DE LENGUAJES

Los lenguajes de tercera generación también cuentan con los avances en la tecnología de computadoras, que estaban hechas a base de circuitos integrados y memoria de partículas magnéticas, su principal aplicación era de sistemas de información. El sistema operativo que utilizaban estas computadoras permitían el manejo de discos, multiprocesos, memoria dinámica, memoria virtual, etc.

Esta generación de lenguajes se caracteriza por sus potentes posibilidades procedimentales y de estructuración de datos. Los lenguajes de programación que surgieron en este tiempo fueron : Cobol, PL/1, DBMS RPG, Pascal, BASIC, etc. Computadoras compatibles con un conjunto de instrucciones universales, tecnología de circuitos integrados y RAM.

CUARTA GENERACION DE LENGUAJES

Los lenguajes de cuarta generación han incrementado la productividad en el desarrollo de sistemas de software. Manejan información central, desarrollo a usuarios finales, y un cambio en el manejo de prototipos. Es de lenguaje No-procedimental (no siguen una secuencia en las estructuras de programación), múltiples DBMS, DBMS relacionales.

Los lenguajes de cuarta generación o 4GL, son sistemas integrados en herramientas CASE de desarrollo que ofrecen una mayor productividad sobre lenguajes de tercera generación. El mayor beneficio que ofrece un 4GL depende del medio ambiente de desarrollo, del cual el lenguaje forma únicamente una parte. Las facilidades que ofrece este ambiente para el programador incluyen las formas de diseño, código, mejora al diccionario y pruebas a programas.

El factor que todos los 4GLs tienen en común es el uso del diccionario de datos, éste almacena muchos de los tipos definidos en un sistema, en un solo lugar. Por ejemplo, guarda campos, tabla de la base de datos, formas y procedimientos. El 4GL es el encargado de exportar la información guardada en el diccionario de datos. Así, una forma (pantallas o mapas) puede usar la validación por omisión de un campo almacenado en el diccionario, o el lenguaje desarrollará una revisión basado en reglas que están en el diccionario.

El diccionario de datos es un sistema de software para el almacenamiento y manejo de toda la información acerca del dato (metadato), también para el almacenamiento de formas, detalles del usuario, programas, etc.

CARACTERISTICAS DE LOS 4GL

Dentro de las características más importantes de un 4GL se encuentran las siguientes:

- Enfocado a un usuario final
- Basado en Mainframe
- Ligado a un DBMS y tiene un alto nivel en la navegación automática de la base de datos y vistas lógicas del usuario. Además soporta las interfaces con otros sistemas manejadores de base de datos y archivos.
- Diccionario de datos que es acoplado con el DBMS y puede controlar la integridad de funciones de datos.
- Soporta el mapeo y creación de pantallas.
- Propósito especial en sistemas de Información Gerencial
- Consultas
- Producción de Reportes
- Proceso de Transacción
- De tamaño pequeño o mediano
- Proceso de desarrollo de aplicación asistido por menú
- Sistemas Genéricos
- Lenguaje Procedimental y No-procedimental , ya que puede o no seguir una secuencia en las estructuras de programación y soporta un alto nivel de sentencias.
- Permite emigrar hacia nuevos ambientes
- Permite la liga de Mainframes a micros, es bidireccional.
- Su productividad es acrecentada por completo en un ciclo de vida del software, incluyendo la codificación, diseño, pruebas, documentación y mantenimiento del proyecto.
- Integridad, seguridad en estructuras de comandos
- Suplementario: paquetes estadísticos, respaldo y recuperación, seguridad, portabilidad, librerías y ayudas.

Los beneficios de usar herramientas de cuarta generación son los siguientes:

- Los sistemas son construidos con mucha más rapidez
- Las pruebas son más fáciles debido a que el código generado es altamente confiable.
- Los lenguajes de cuarta generación son amigables
- El desarrollo de aplicaciones es más rápido en prototipos y diseño de reportes.
- Muchos lenguajes de cuarta generación producen documentación
- Usuarios finales pueden definir y mantener aplicaciones completas de cuarta generación.

Existen algunas limitaciones en las herramientas de cuarta generación:

- Muchas herramientas de cuarta generación son para DBMS y sistemas operativos específicos limitando de esta forma su portabilidad.
- No proveen la capacidad de actualización de datos concurrencia y no tienen recuperación automática.
- La gran mayoría de las herramientas de cuarta generación son ineficientes para manejar el procesamiento de un alto volumen de transacciones.
- Muchos programas de cuarta generación son incompatibles con sistemas existentes escritos en lenguajes de segunda o tercera generación.
- Los lenguajes de cuarta generación no están estandarizados
- Los programas de cuarta generación usan una gran cantidad de recursos de máquina
- Las herramientas de cuarta generación producen código redundante

CASE y 4GL

El número 4GL está asociado con CASE. Los analistas de sistemas emplean las herramientas CASE en la descomposición funcional y crean estructuras y diagramas de flujos de datos y pueden ser usados en la generación de código de programas de 4GL.

El término "Upper CASE" es usado en la descripción de herramientas CASE, está basado en la función básica de diagramación y el diccionario de datos. El término "Lower CASE" es usado en la descripción de herramientas CASE, como se mencionó en el capítulo I, está basado en la interface del ambiente de desarrollo y generación de código.

Las herramientas CASE también proporcionan una revisión automática, ya sea de sintaxis y/o semántica en la programación, las cuales ayudan a la eliminación de algunos errores del analista. La productividad que ofrece al analista, usando los beneficios CASE es una garantía en la documentación y durante la etapa de diseño del proyecto, el analista almacena datos al diccionario y es fácil de usar durante la etapa de programación.

El gran problema con las herramientas CASE es que en la mayor parte no ofrecen un apropiado ambiente de desarrollo, y muchas veces las herramientas CASE no utilizan el diccionario durante el desarrollo. Puesto que en la mayoría de las herramientas CASE proporcionan las facilidades de Importación/Exportación a otros diccionarios.

3.2 PRODUCTOS 4GL

A continuación se describirán algunas de las características más importantes de algunos lenguajes de cuarta generación usados actualmente.

3.2.1 PROGRESS

Progress se ha establecido cómo un lenguaje de cuarta generación, en ambiente UNIX, y es utilizado en las PCLANs. Los lenguajes de cuarta generación tienen su origen en los ambientes multi-usuario.

Progress surgió en 1983 en el ambiente UNIX con versiones para DOS, y VAX/VMS en esta última versión se creó en 1987, recientemente se ha liberado para OS/2, OS/400, Novell NLM, CTOS y MS Windows.

ARQUITECTURA

Progress consiste de cinco componentes principales: lenguaje de cuarta generación, un editor inteligente de procedimientos, un pintador de pantalla y un formateador de reportes, un diccionario de datos y una base de datos relacional. Además existen seis productos opcionales, de los cuáles dos son los más importantes: Fast Track y Results.

Fast Track es un menú manejador de prototipos y un ambiente de escritura de reportes. Con Fast Track se puede crear un prototipo o generar un reporte, después se compila y se integra al sistema.

Progress tiene las mismas capacidades de otras bases de datos y además las puede acceder. Adicionalmente se pueden agregar sentencias de SQL en aplicaciones de C y tener acceso a la base de datos Progress con el Host de interfaz del lenguaje Progress, lo que no obliga el acceso al 4GL. Progress está dirigido para programadores principiantes para un desarrollo rápido de sistemas. Aunque posee herramientas CASE este producto no es orientado a objetos.

LENGUAJE

El 4GL Progress es un lenguaje de inglés común que soporta el estándar SQL ANSI, este tiene una librería integrada de funciones y puede manipular operaciones aritméticas, fechas, así como una cadena de caracteres y un control de estado, soporta la definición de arreglos. Contiene un editor interactivo que chequea la sintaxis detecta palabras y frases incorrectas, ofrece una posible solución a los errores. Esto también se complementa con una ayuda en línea disponible en cualquier momento, proporcionando información de todas las operaciones en el lenguaje. Contiene gran variedad de funciones como ventanas pop-up, escroleo, etc.

FORMAS

Las formas y menús son creados en el Fast Track el cual incluye un diseñador de pantallas, así como otros diseñadores de pantallas que pueden diseñar formas agregando campos, etiquetas y texto, se crean formas por omisión desde una tabla de base de datos, posee la capacidad de ventanas y consulta, y soporta el procesamiento de texto pero no soporta gráficos. Una de las ventajas de Progress es que lo que se ve en pantalla es lo que se va a obtener, es decir no se pueden incluir procesos especiales para uno o varios campos sin tener que recurrir al código de 4GL.

Una consulta por formas (QBF) del Fast Track permite unir datos entre diferentes tablas e índices. Progress únicamente soporta aplicaciones y desarrollo basado en caracteres.

ESCRITURA DE REPORTES

Esta es una de las primeras herramientas del programador, los campos se seleccionan uno por uno, posicionando los campos donde se requieran, tiene las ventajas de ordenar, relacionar campos lógicamente

DICCIONARIO

El diccionario de Progress es un manejador de menú que almacena toda la información de la estructura de la base de datos, así como los nombres de los archivos, campos, definición de llaves y llaves externas, etc. También posee las definiciones de formatos, etiquetas, validaciones y mensajes de ayuda.

Las definiciones de la base de datos son almacenadas dinámicamente proporcionando seguridad en los archivos campos e índices. Esto significa que los cambios pueden realizarse almacenándose y recargando la base de datos.

La validación puede ser definida a nivel archivo y a nivel campo con una revisión automática, con otros procedimientos de campos relacionados ofreciendo así una integridad de referencia total. Progress cuenta con un gran número de opciones extras para su diccionario incluyendo una interfaz con CASE, el cual permite integrar a otros productos CASE el diccionario de Progress. Cuando un esquema de base de datos externa es copiada dentro de Progress se crea un respaldo del esquema, el cual almacena el número de versión. En caso de que la base de datos se actualice Progress contará con un respaldo para comparar la base de datos anterior con la actual.

SOPORTE DISTRIBUIDO

Otra ventaja de la base de datos Progress es su arquitectura cliente/multi-servidor. No solamente soporta cliente/servidor, si no que también es muy eficiente en la arquitectura multi-servidor esto permite que la base de datos soporte servicio a grupos de procesos de clientes y pueden ser utilizados para explotar al máximo las capacidades de grandes máquinas. Progress provee su propio software para redes de trabajo que están enfocados al ambiente cliente/servidor y aplicaciones distribuidas.

Por otro lado Progress no soporta llamadas a procedimientos lejanos por lo tanto no soporta el procesamiento cooperativo (es compartir la carga de trabajo entre dos o más computadoras, tales como una macrocomputadora y una computadora personal), el cual es opuesto al procesamiento cliente/servidor (un ejemplo de cliente/servidor en un sistema de base de datos para trabajar en red, donde el cliente es la interfaz del usuario en el cual reside la estación de trabajo y las funciones de almacenamiento y recuperación residen en el servidor).

BASE DE DATOS

La base de datos que provee Progress, ofrece respaldo en línea, acceso indexado, acceso directo a línea de tablas que contienen el número de registros. Estos números de registros son utilizados independientemente por el usuario para establecer las relaciones entre archivos.

MIGRACION

Progress tiene la habilidad de correr aplicaciones anteriores en ambientes nuevos.

RAD (Rápida Aplicación en el Desarrollo)

Soporta el desarrollo de prototipos y el uso de procedimientos fuente, el uso de un código fuente en la librería de procesamiento llevará el control y la reutilización. Una de las limitantes de Progress es que no tiene un depurador.

HERRAMIENTAS ASOCIADAS

Las herramientas asociadas son Developer's Toolkit y Run-Time. Developer's Toolkit sirve para encriptar código objeto y el Run-Time Proporciona el desarrollo de aplicaciones sin código objeto.

3.2.2 FOCUS

La filosofía de Focus está en su fácil uso en el ambiente de desarrollo de aplicación. Esto tiene un resultado dentro de la evolución progresiva, en la interfaz con el usuario que proporciona soporte a todos los niveles. Focus produce reportes, rutinas de datos de entrada y gráficos. Focus cuenta con las siguientes herramientas: Table Talk produce reportes, Modify Talk genera transacciones en datos de entrada, Plot Talk genera gráficos de trabajo, y otras utilidades como File Talk que intenta proporcionar al usuario la definición de estructura de datos. Focus ofrece un ambiente sofisticado para la construcción de aplicaciones.

LENGUAJE

El lenguaje FOCUS, contiene las construcciones usuales para direcciones de requerimientos comunes. Sin embargo Focus tiene la facilidad de un lenguaje de cuarta generación que proporciona la navegación automática de la base de datos. El lenguaje es cubierto por el ambiente Talk, la entrada de datos en línea y la rapidez de programas pueden ser creados usando Modify Talk, esto proporciona la facilidad en el código generado. Comparado con otros productos, Talk y Modify Talk de Focus, proporciona un ambiente donde la entrada de datos tiene una labor de construcción de código. Este código puede ser editado con Modify Talk y sólo las modificaciones correctas sintácticamente son almacenadas. El Manejador del diálogo, proporciona el soporte en la creación de menús y diálogos, variables definidas y control en el ambiente de tiempos de ejecución.

FORMAS

La capacidad de GUI (Interfaz de Gráficos a Usuarios) en ambos niveles, cómo el de desarrollo y aplicación, es común solo en el Manejador OS/2. P.M./Focus proporciona un editor y archivo de aplicación, formas y facilidad de pintado. Soporta el siguiente tipo de formas de objeto: entrada de datos estándar, entrada de datos en multi-línea, radio buttons, lista de cajas, chequeo de cajas, y repetición de grupos en forma de matriz. Los radio buttons pueden ser de texto, gráficos o invisibles. El ambiente de focus está basado en el pintado de caracteres en pantalla, orientado a un generador de código que es usado por un menú de selección.

ESCRITURA DE REPORTES

Focus incorpora el Report Writer dirigido al desarrollo, y Table Talk es el principal producto de Focus y es un generador automático de reportes. Este menú solo es el manejador del ambiente, cada opción del punto crea un reporte. El ambiente es inteligente y solo las opciones relevantes son presentadas en cada paso del proceso.

Una vez definidos todos los campos en un archivo, estos pueden incluirse en un reporte.

DICCIONARIO

En la descripción del archivo maestro que posee archivos o tablas, donde incluye campos que contienen metadatos, formatos, reglas de validación, seguridad en el acceso, ayudas de información, campos derivados y opcionalmente relaciones entre tablas. En efecto esto crea vistas, además de la descripción del acceso al archivo, el cual contiene metadatos de cada base de datos.

Las razones por que es aprovechada esta información, es que usa una combinación de métodos en el almacenamiento del archivo, incluyendo varios tipos de base de datos en formatos convencionales.

SOPORTE DISTRIBUIDO

Una de las ventajas de Focus es que soporta un procesamiento cooperativo. Por medio de las llaves de FOCNET, soporta la comunicación entre DOS, OS/2, Mainframe de IBM, VAX de Digital, AS/400 y Unix.

BASE DE DATOS

La base de datos de Focus tiene un acceso jerárquico y relacional en algunas plataformas, esto puede ser soportada por un manejador de datos externo. Focus ofrece un alto nivel de portabilidad y un nivel alto de inter-operabilidad. Soporta base de datos como Rdb, Oracle, Ingress, Sybase, Adabas, Teradata, ALibase/SQL, CA-Datacom, CA-DB, DB2, DBase, CA-IDMS, IMS/DB, Informix, Model 204, NonStop SQL, OS/2 DB, OS/400 DB, Supra, TOTAL, Turbo IMAGE, Unify 2000.

RAD (Rápida Aplicación en el Desarrollo)

Focus soporta prototipos iterativos, en resultados finales, de los cuales pueden ser utilizados en un ambiente de producción. Sin embargo suceden casos en que el código necesita un ajuste para razones de desempeño.

HERRAMIENTAS ASOCIADAS

Focus incluye estadísticas, salida aún Plot Talk que produce gráficos. También cuenta con un lenguaje de reportes financieros.

3.2.3 NATURAL

NATURAL es un lenguaje de cuarta generación, y contiene en un sólo módulo, todos los componentes necesarios para la implementación de un ambiente práctico y productivo en las áreas de desarrollo de aplicaciones en línea, administración y seguridad de datos, definición de pantallas y prueba de programas. Incluye facilidades para desarrollo interactivo de aplicaciones y generación de reportes, incrementando considerablemente la productividad en el área de desarrollo y programación

Natural corre bajo VSE, VM, MVS en ambiente de Mainframe de IBM, en VMS de Digital, Unix, soporta el OSF/Motif y el Open Look de GUI, PC bajo MS-DOS, MS-Windows, OS/2, en Apple Macintosh y varias plataformas incluyendo Wang, ICL y Siemens-Nixdorf.

ARQUITECTURA

La arquitectura de desarrollo de aplicaciones en Natural se encuentra dividida en dos partes que pueden unirse mutuamente. La primera consiste del 4GL de Natural, Natural Construct un generador de código, Natural Expert es un sistema experto, Super Natural que es una herramienta de reportes para usuarios finales y Natural Security. El resto de la familia está compuesto por Predict que es un diccionario de datos activo y Predict CASE.

LENGUAJE

Natural es un lenguaje Procedimental altamente funcional, dirigido a aplicaciones críticas. La estructura de su sintaxis sigue los mismos conceptos para las rutinas de interfaz de los lenguajes de tercera generación, los cuales requieren desarrolladores que siguen los estándares. Soporta SQL que puede ser utilizado como en Natural y en programas de tercera generación.

Este conforma el estándar ANSI II y las extensiones para DB2, Oracle, Rdb, Sybase así como Adabas.

Como Natural Construct y Predict CASE generan programas en Natural 4GL, los programas son generados desde modelos, Construct cuenta con 90 modelos preconstruidos (estructuras de código para aplicaciones específicas), más un constructor de modelos para que los desarrolladores creen sus propios modelos y modifiquen los ya existentes. Opcionalmente, Construct puede ser utilizado para generar código 3GL. Esto no es recomendable cuando se desee implantar el acceso a un nivel de entrada y salida de un lenguaje de tercera generación, el cual puede hacer caso omiso a las reglas de seguridad. El código de Natural casi siempre es compilado, la excepción es que está limitado a un cierto rango de llamadas a DBMS.

FORMAS

A nivel de caracteres Natural provee un fuerte soporte para diseño de ventanas. No se requieren modificaciones para aplicaciones que soportan GUI (interfases gráficas de usuarios) y pantallas basadas en caracteres al mismo tiempo. EL soporte de GUI de Natural todavía no está totalmente desarrollado. Soporta los botones de push, de radio, las barras de escroleo etc. Las versiones GUI de Natural vienen con cuatro 'seleccione y presione', basado en editores que forman el corazón del sistema.

El editor de programas es utilizado para la creación y edición de programas Natural, rutinas de ayuda, etc. El editor del área de datos es para la creación y mantenimiento de las áreas de datos locales y de parámetros, el editor de DDM (vistas) permite la definición de módulos de datos o el uso de DDM's para crear y mantener las vistas de Natural. El editor de Mapas/Formas es el diseñador de pantallas actual, permite al diseñador generar formas y crear interfaz de usuarios finales dibujando objetos relevantes y campos de datos.

ESCRITURA DE REPORTE

Las facilidades de escritura de reportes son similares a la de las formas. El editor de mapas es utilizado para diseñar el marco del reporte y una función programable que puede ser utilizada para probar el reporte en modo del prototipo. Super Natural es un manejador de menús para usuarios finales, sin embargo es más que una herramienta de reportes, ya que permite la generación de funciones, así como el mantenimiento de programas.

DICCIONARIO DE DATOS

Natural utiliza el Predict y soporta el uso de CDD+ en DEC VAX y el DB2 catalogo en Mainframe de IBM.

Predict es un diccionario de datos, donde se puede activar y desactivar ciertas partes en un ambiente de tiempo de ejecución, lo cual le permite operar en modo pasivo, por ejemplo la ayuda en línea puede estar siempre activa pero las referencias cruzadas (referencias activas o cruzadas es información que se almacena durante la catalogación de un programa Natural)

pueden ser desactivadas para el desarrollo de sistemas, esto incrementará la eficiencia al tiempo de ejecución del sistema.

Predict está desarrollado en Natural esto significa que el usuario puede extender la meta-estructura de Predict definiendo sus propios tipos de objetos y relaciones. Esa es una ventaja particular que puede servir para aplicaciones especializadas.

SOPORTE DISTRIBUIDO

Para el soporte cliente/servidor tiene su propio RPC (Llamadas Remotas a Procedimientos). La tecnología de servidor provee acceso a funciones de datos y servicios en computadoras locales y remotas utilizada comúnmente en red, haciendo esto transparente.

BASE DE DATOS

La base de datos de Natural se llama Adabas y fue desarrollada originalmente como una base de datos navegacional. Soporta vistas relacionales de la base de datos a través de Natural y mientras que las vistas de relaciones de entidad se mantienen a través de Adabas. Así como las base de datos intrínsecamente jerárquicas, Adabas ofrece un excelente desempeño tanto en OLTP (Proceso de Transacciones En-Línea), así como para la consulta. Actualmente tiene la facilidad de manipular arreglos. Natural ofrece actualización total para DB2, Rdb, Oracle y Sybase incluyendo, Supra, Cincom's y Total, Compute Associates IDMS, Plus IMS, DL/I, etc. El acceso a las base de datos heterogéneos se realiza via Natural TP.

RAD (Rápida Aplicación en el Desarrollo)

Natural ofrece facilidades para realizar prototipos con varios componentes.

HERRAMIENTAS ASOCIADAS

Natural Expert provee un sistema experto con su propio lenguaje, de tal forma que un sistema experto creado con Natural Expert puede ligarse con otro programa Natural.

3.3 NATURAL ARCHITECT (HERRAMIENTA CASE)

Natural Architect ofrece un conjunto de herramientas integradas, disponible para aplicaciones en línea del análisis y diseño. El diseño de metodologías estructuradas pueden incrementar la calidad del diseño y la productividad puede ser mayor a un 300%, asegurando que sus desarrollos estén consistentes en el tiempo. El sistema automatiza el proceso de documentación y diagramación, se puede producir documentación del sistema de alta calidad,

diagrama de flujo de datos, diseño de la base de datos, definición de la estructura del programa y diseño de pantallas.

La integridad del diseño se asegura, ya que se requiere que el desarrollador defina los componentes significativos del sistema y establezca decisiones a cerca de la conducta de cada componente esta disciplina produce un diseño bien pensado, resultando una rápida codificación, menos depuración y un mantenimiento más fácil.

Natural Architect corre en estaciones de trabajo Macintosh, se utilizan menús pull-down, iconos y selección, control de tamaño de forma y posición de los componentes del diagrama y los campos. Controla la apariencia de las salidas especificando los caracteres y el tamaño de los mismos. Texto intensificado subrayado, itálicas, sombras, etc. En los diagramas diseñados las definiciones y otros componentes pueden ser fácilmente exportados a un ambiente integrado de gráficas de procesamiento de palabras y otras aplicaciones Macintosh.

Natural Architect se integra con el ambiente de desarrollo de Natural para maximizar el poder y la funcionalidad del Mainframe. Por ejemplo el sistema automáticamente genera la definición de los elementos de los datos de Predict y definiciones de pantallas para IBM y VAX.

Natural Architect proporciona herramientas para predecir y prevenir problemas durante el análisis y diseño haciendo que los errores sean fácil de resolver.

Sus metodologías estructuradas aunadas al análisis de diseño automatizado, el cual identifica inconsistencia y errores potenciales. Se pueden descubrir errores antes de codificar evitando cambios costosos que afecten a cientos de programas.

Natural Architect permite a los usuarios finales participar durante los procesos de análisis y diseño, a través de un proceso interactivo, los usuarios pueden participar en el desarrollo de la aplicación en el principio del proyecto definiendo así las necesidades del usuario evitando sorpresas al finalizar el proyecto.

Natural Architect tiene un conjunto de editores integrados que le permiten diagramar y diseñar. Todos los editores giran alrededor de un diccionario de datos basados en Predict.

EDITOR DE DIAGRAMA DE FLUJO DE DATOS

Este editor le permite ver el flujo de información a través del sistema propuesto. Architect soporta las metodologías de diagramación de flujo de datos más populares Yourdon, Gane and Sarson.

Las modificaciones en un nivel de un diagrama se reflejan automáticamente a otros niveles. Cuando se especifica un almacén de datos o la definición de un atributo del flujo de datos este se refleja automáticamente en el diccionario de datos.

EDITOR DE DIAGRAMA DE ENTIDAD RELACION

Este diagrama modela la relación entre las entidades de datos, incluyendo la definición del campo llave (descriptor o el superdescriptor) esta herramienta permite las definiciones físicas y lógicas, soporta el diseño de la base de datos relacional y permite el uso de las características adicionales de Adabas, tales como grupos periódicos y campos de valores múltiples, esta herramienta permite las tecnologías de Martí y una forma modificada de las tecnologías de Bachman/Chen.

EDITOR DE DIAGRAMAS DE ESTRUCTURA DE PROGRAMAS

Este editor permite expresar la lógica del programa opcionalmente con la metodología de Jackson.

Se pueden presentar componentes condicionales e incondicionales, ciclos, subrutinas. Permite la descomposición en múltiples niveles.

EDITOR DE FORMAS

Este editor permite diseñar pantallas de aplicación Natural formas y reportes más fácil y rápido cuando se utilizan métodos tradicionales se pueden traer los atributos desde el diccionario hacia la pantalla. El editor de formas automáticamente traduce los atributos en representaciones visuales, se puede ocultar, iluminar o utilizar video inverso en la selección de campos, se pueden reorganizar los campos en la pantalla posicionandolos en nuevos lugares, crea formas simples, diseña pantallas completas sin tener que escribirlas al diccionario de datos que almacena su diseño para futuros usos.

DICCIONARIO INTEGRADO

El Diccionario de datos de Architect integra todos los componentes del sistema y es el almacén central para todos los elementos de los datos nombres y definiciones especificadas en el diseño. Se puede definir los atributos del diccionario en cualquier estado del proceso del diseño, una vez definidos estos, están disponibles automáticamente para otros componentes del sistema y diagramas.

Los cambios se reflejan automáticamente en los segmentos del diseño relacionados con el sistema. Para mayor eficiencia varios diseños pueden hacer uso del mismo diccionario, las descripciones para cada componente del diseño pueden contener hasta 32000 caracteres, lo

cual es suficiente para descripciones muy largas. El diccionario de datos Architect soporta las especificaciones de definición de atributos Adabas. En resumen la utilidad del diccionario permite que varios diseñadores trabajen en un mismo diseño simultáneamente y combinen sus esfuerzos.

DOCUMENTACION AUTOMATIZADA

Architect provee copias de reportes para cada componente del sistema. El reporte incluye el texto y los atributos de los datos asociados con el diseño general y con cada componente del diseño, este reporte puede ser fácilmente editado de acuerdo a sus necesidades así mismo se puede abrir el reporte con un programa procesador de palabras y darle el formato deseado. Natural Architect soporta Page Maker, Style tags para un control avanzado en su formato de reportes.

COMPILADOR DEL DISEÑO

Este asegura la integridad del diseño analizando los diagramas del diseño para correcciones sintácticas y lógicas. Después de analizar el diseño el compilador reporta errores y advertencias de problemas potenciales. El sistema almacena las compilaciones en un archivo objeto, disponible para traducirse y transferirse al host de la computadora.

LA UTILIDAD DEL DICCIONARIO DE ARCHITECT

El diccionario es la parte central de cualquier diseño en Architect, la utilidad del diccionario permite manipular lo siguiente:

- Revisar y reparar el diccionario
- Combinar el diccionario con diferentes usuarios
- Transferir texto dentro y fuera del diccionario
- Referencias cruzadas a un diccionario

GATEWAY

Este módulo hace interface con el diseño compilado, con el host de Predict/Natural/Natural Construct. El Gateway consiste en rutinas que corren en Macintosh y en el host. Sus capacidades combinadas permiten alimentar la información del diseño directamente en Predict, Natural para generar las base de datos, archivos, campos, relaciones, programas y mapas. Las funciones del Gateway, así como los diseños realizados en Natural Architect para el host proveen seguridad en la calidad de las metodologías. Gateway provee:

- Entrada a través de un password
- Menús pull-down para comandos del host
- Múltiples ventanas o ambientes de trabajo

3.4 PREDICT

PREDICT DDMS (Sistema Manejador de Diccionario de Datos) es un sistema de software para el almacenamiento y manejo de toda la información acerca del dato (metadato) de una organización y consiste de:

Una meta-base de datos, un manejador y un sistema de control para el mantenimiento, evaluación, protección y seguridad de los metadatos.

(Un metadato es el nombre de un campo y una meta-base de datos va a contener metadatos o nombres de campos). A continuación se muestra un ejemplo de metadato.

Metadato: Datos acerca del dato.

DATOS(Usuario)	METADATOS
Susana González Analista	Nombre Apellido Pat. Puesto

PREDICT se utiliza para mantener y generar modelos de metadatos de sistemas de aplicación complejos. Los metadatos pueden ser creados manualmente o automáticamente para implantar código. Los modelos de metadatos pueden centralizar el mantenimiento y almacenamiento de información necesaria para el uso y producción de aplicaciones de software.

VENTAJAS

- La producción y uso de software gana eficiencia porque la información esta disponible cuando se necesita.
- Se puede generar código a partir de metadatos.
- La consistencia en la información se puede asegurar más fácilmente.

Predict ofrece facilidades para diferentes fases del ciclo de vida del software.

Diseño de aplicaciones	Los archivos conceptuales de Predict contienen la descripción de la estructura de datos de una aplicación.
Administración de datos	La estructura de las bases de datos, archivos, campos y campos por omisión pueden definirse centralmente en el diccionario de datos.
Documentación del programa	El código fuente (incluyendo todas las relaciones del sistema, módulos, usuarios, bases de datos, archivos y campos) pueden ser documentados centralmente en Predict.
Generación de código	Las estructuras de archivos de diferentes tipos de lenguajes de programación y sistemas de base de datos pueden generarse.
Control de implantación	Las referencias cruzadas activas comparan el código implantado, con su correspondiente documentación en Predict.
Creación de reportes	La descripción de los datos puede ser evaluada y desplegada. Se pueden generar reportes para todos los subconjuntos de entradas en Predict.
Re-ingeniería	Ofrece facilidades para volver a construir aplicaciones (referencias activas, incorporación de funciones, funciones de redocumentación).

3.4.1 CONTROL DE LA APLICACION EN PREDICT (PAC)

Ayuda automatizar los procedimientos para manipularlas y controlar la aplicación durante su ciclo de vida. Asegura la integridad de sus aplicaciones basados en Natural. Cuenta con las siguientes ventajas:

- Integridad de la aplicación

- Incrementa la eficiencia, Control y precisión
- Procedimiento de mantenimiento modernizado
- Ambiente de producción protegidos
- Revisión automática de reportes

INTEGRIDAD DE LA APLICACION

PAC ayuda a asegurar la integridad de una aplicación a través de su ciclo de vida. Toda la información acerca del estado de cada aplicación se localiza en un sólo lugar. Como resultado se puede centralizar el control, prevenir inconsistencias adicionalmente PAC permite definir y establecer niveles de seguridad y control de ambiente de tal manera que refuerza sus estándares.

Con PAC se puede identificar fácilmente todos los objetos y relaciones. PAC automáticamente mantiene las relaciones y dependencias entre versiones creadas en Natural y Predict.

INCREMENTA LA EFICIENCIA, CONTROL Y PRECISION

PAC ayuda a implantar las aplicaciones más rápido y eficientemente. Definiendo rutas de prueba, asegura que todas las pruebas realizadas se lleven a cabo en las versiones correctas de los objetos. PAC ahorra tiempo y elimina inconsistencias entre los estados del ciclo de vida. Frecuentemente la variedad de base de datos y archivos utilizados en diferentes ambientes presentan obstáculos cuando se mueven objetos y aplicaciones (entre pruebas y producción). Con PAC el código objeto esta situado en un sólo lugar, este es compilado si se va hacer un acceso o actualización de los datos. Opcionalmente conforme se vaya cambiando su aplicación del ciclo de vida PAC puede recompilar dinámicamente y redireccionar el acceso o actualización a la base de datos, requeridos para diferentes conjuntos de datos.

PROCEDIMIENTO DE MANTENIMIENTO MODERNIZADO

Para mantener la integridad de la aplicación durante el mantenimiento se puede utilizar PAC para asegurar el ambiente de control tales como restringir el acceso a un sólo programador una vez que un objeto ha sido actualizado, para mantener un registro completo de todas las versiones. PAC también puede ayudar a rastrear las actividades de mantenimiento y de los problemas con información detallada por prioridades y descripción.

AMBIENTE DE PRODUCCION PROTEGIDOS

Una de las metas del control de una aplicación es asegurar la integridad del ambiente de producción ya sea centralizado o distribuido. PAC provee el Predict Application Audit que es

un sistema de reportes para calendarizar la inspección independiente de respaldos y recuperación de aplicaciones. Se pueden utilizar reportes en procesos en lotes o en-Línea.

REVISION AUTOMATICA DE REPORTES

Como parte de control de nuevas versiones de aplicaciones y sus objetos, PAC controla todas las acciones con todos los objetos.

3.4.2 MANEJADOR LOGICO Y FISICO DE ESTRUCTURA DE DATOS

Predict soporta diseño técnico, implantación y mantenimiento de la estructura de datos como de la estructura funcionales de aplicaciones. Las funciones para crear, mantener, analizar y transformar la documentación de los datos, así como recuperar la información de todos los tipos individuales de objetos o grupos de objetos. Los tipos de objetos en la meta-estructura predefinida de Predict son utilizados para describir los datos y programas (así como sus interconexiones físicas y lógicas). Se definen relaciones entre base de datos, archivos, campos, redes, máquinas virtuales, sistemas y programas, reglas de verificación y relaciones de archivo. Los tipos de objetos definidos por el usuario y relaciones pueden ser agregados a la meta-estructura predefinida y adaptarse a Predict dependiendo de los requerimientos. La estandarización de la definición de los datos se pueden hacer utilizando archivos definidos de Predict. Cuando las definiciones de un campo son modificadas en el archivo definido, todas las vistas son actualizadas automáticamente.

IMPLANTACION Y DOCUMENTACION CONSISTENTE

Los datos y programas de aplicaciones implantados son documentados automáticamente con las referencias activas, almacenadas en Predict los usuarios pueden comparar esta implantación de documentación con los conceptos de diseño en Predict (DP).

DESARROLLO DE LA APLICACION A DIFERENTES PLATAFORMAS

Predict soporta gran variedad de sistemas de base de datos, incluyendo Adabas, ENTIRE; SQL-DB, DB2, DL/I, IMS/DB, SQL/DS, Rdb, RMS, VSAM, ISAM (BS2000), y SQL base de datos Oracle, Ingres, Sybase, e Informix. Predict soporta aplicaciones SAP, y ofrece funcionalidad compatible con MVS, MVS/XA, VSE, VM/CMS, BS2000, VAX/VMS, WANG/VS, y UNIX. Esta es una herramienta excelente para soportar desarrollos y producciones a la medida en ambientes computarizados heterogéneos, el diseño de datos y las ventajas en la manipulación de las aplicaciones aseguran la portabilidad de estas.

Los lenguajes que maneja Predict son Natural, Cobol, ADA, Fortran, PL/I y Assembler.

3.5 NATURAL CONSTRUCT

Es una herramienta de programación que automatiza gran parte del trabajo redundante en el desarrollo de aplicaciones. Es un sistema generador de aplicaciones de un código generador con las funciones de un modelador de sistemas y herramientas de simulación. Natural Construct utiliza las facilidades de Natural contiene un generador de programa (una librería de modelos programables que se utilizan para generar aplicaciones interactivas) un sistema de mantenimiento y un sistema de mantenimiento de ayudas.

Natural Construct produce aplicaciones de alta calidad rápidamente y se debe a dos razones:

1) Construct produce resultados con calidad por que su herramienta CASE combinada con su lenguaje de cuarta generación Natural, aprovecha las metodologías y herramientas automatizadas. Cuenta con las siguientes facilidades:

- Automatiza las tareas repetitivas
- Asegura que las tareas se realicen en forma correcta y consistentemente
- Se concentra en la resolución de problemas relacionados con una tarea específica.
- Automatiza el uso de procesos estándares

2) Ventajas

- Invierte menos tiempo en la realización de código redundante
- Reduce el tiempo de depuración
- Invierte más tiempo en los aspectos creativos del desarrollo
- Utiliza código estructurado estandarizado
- Minimiza el impacto de los cambios por mantenimiento
- Crea ayudas de texto independientes
- Utiliza un software integrado en el ambiente de desarrollo
- Incrementa su productividad

Natural Construct genera aplicaciones en Natural y utiliza Predict para obtener información de los campos y vistas utilizadas, combinando estas herramientas pueden generar aplicaciones, así como dar mantenimiento a aplicaciones previamente creadas. Combinando Natural Construct, Natural 4GL y Predict los programadores pueden desarrollar un amplio rango de sistemas de información integrados en una tecnología base. Los modelos Batch (son procesos de ejecución, en donde se manejan grandes cantidades de información controladas desde el sistema operativo) y Browse (hace una lectura a la información en cuestión), así como los

modelos de mantenimiento utilizan Predict para construir las vistas definidas por el generador de programas que corresponde a las vistas especificadas en Predict.

Predict debe estar disponible para poder realizar las funciones de Construct, Predict es una herramienta para la administración de datos. La estructura de los metadatos en Predict describe las relaciones físicas y lógicas de los datos y programas que se utilizan. Las relaciones entre las base de datos, campos, campos elementales, sistemas de aplicación, módulos y reportes se definen en Predict, así como las reglas de validación y las relaciones entre campos.

COMPONENTES EN LA GENERACION DE PROGRAMAS.

Especificaciones	Las especificaciones son los parámetros que Construct utiliza para generar código Natural para el tipo de programas a desarrollar, las especificaciones se almacenan con Construct y pueden ser modificadas generación tras generación de programas
Paneles	Los paneles son las pantallas donde se introducen las especificaciones
Modelo	Un modelo es un conjunto de plantillas de programas (code frame) y subprogramas relacionados. Cada modelo tiene al menos un panel
Código User Exit	Este código es escrito por el programador en Natural. Este contiene código particular bajo el cual el programa será desarrollado
Programa Generado	Un programa generado es un programa Natural completo, almacenado en una librería Natural del programador
Code Frame	El code frame es una combinación de sentencias Natural y operadores de Construct. Las sentencias definen la estructura básica del programa. Los operadores de consulta definidos por el usuario son los parámetros para generar el código tales como los archivos de la base de datos y mapas a hacer utilizados

Existen tres tipos de Modelos

Programas	Estos sirven para generar aplicaciones completas de programas estructurados. Estos modelos están disponibles en el subsistema de generación
Sentencias	Crea bloques de código (es decir cualquier conjunto de sentencias cohesivas en Natural) pueden ser generados directamente en el editor de programas Natural o del programa Natural, User Exit o el editor de code frame. Los modelos de sentencias están disponibles en el subsistema de generación
Creación de Modelos	Crea los componentes de un modelo Construct permitiendo a los usuarios aumentar la selección de modelos Construct. Los modelos pueden ser creados por el administrador del subsistema

MODELOS DE PROGRAMAS

Los modelos de programas son utilizados para generar programas, subprogramas, rutinas de ayuda, área de datos y mapas. Las ventajas de los modelos de Construct son las siguientes:

- Soporte de mensajes en varios lenguajes, pantallas y nombre de funciones.
- Pf-Keys definidos y con su nombre de cada PF (funciones del teclado).
- Soporte de interfaz con IBM, la interfaz se llama Acceso de Comunicaciones de Usuarios (CUA).
- Acceso directo a Construct por comando directo el cual permite navegación rápida utilizando comando directo en vez de menús.
- Verificación de entrada
- Capacidades extendidas en diferentes ambientes.

En la siguiente tabla se presentará algunos modelos con su función correspondiente:

Modelo	Función
STARTUP	Inicializa el ambiente de la aplicación
QUIT	Termina la aplicación
MENÚ	Controla el flujo de los módulos de trabajo de una aplicación
BROWSE-HELPR	Genera una rutina de ayuda que hace un barrido simple a la base de datos
BROWSE-SUBP	Genera un subprograma que hace un barrido simple a la base de datos
BROWSE-SELECT	Genera un programa complejo de la base de datos
BROWSE-SELECT-HELPR	Genera una rutina de ayuda de consulta de la base de datos
BROWSE-SELECT-SUBP	Genera un subprograma de browse complejo de la base de datos.
MAINT	Crea un programa de actualización y selección de la base de datos.
MAP	Genera mapas Natural.
HELP-TEX-SELECT	Accesa la ayuda pasiva a Construct del sistema
BATCH	Genera programas de actualización y reportes en batch.
JCL-DOS-NATBACH	Genera JCL en el ambiente DOS
JCL-OS-NATBACH	Genera JCL en el ambiente OS.
DRIVER	Prueba y depura subprogramas y rutina de ayuda
SHELL	Realiza patrones para programas escritos en Natural

MODELO DE OBJETOS

Es un grupo de entidades de datos relacionales al que se le puede dar mantenimiento con una transacción simple. Los modelos de objetos son un subconjunto de los modelos de programas, los modelos de objetos también se utilizan para generar procesos de mantenimiento a objetos.

El concepto de desarrollo orientado a objeto es una aplicación que consiste de componentes objetos, subprogramas Natural y componentes de diálogo (programas o subprogramas Natural). El componente de diálogo se comunica con el usuario final e invoca métodos (acciones a los datos) implantados por el componente objeto. En la siguiente tabla se indican los modelos de objetos y sus funciones:

Modelo	Función
OBJECT-PDA	Genera PDA utiliza para transferir datos
OBJECT-PDA-R	Genera PDA utiliza para generar información que es accesada a través de múltiples aplicaciones.
OBJECT-SUBP	Genera un subprograma que ensambla al objeto, realiza la actualización y reafirma la integridad de los datos.
OBJECT-MAINT-DIALOG	Genera un componente de diálogo de una aplicación orientada a objeto.
OBJECT-MAINT-DIALOG-SUBP	Genera un componente de diálogo de la aplicación orientada a objetos utilizando accesos a programas.

AMBIENTE DE NATURAL CONSTRUCT

Los componentes principales de Construct son subsistemas y los modelos utilizados en los procesos de generación. Construct es una herramienta para gran variedad de usuarios, sirve a las necesidades de diferentes usuarios y se divide en tres subsistemas.

SUBSISTEMA DE GENERACION:

Los desarrolladores lo utilizan para agregar sus especificaciones, después de que Construct genera el programa del código objeto Natural.

Este subsistema utiliza modelos reutilizables, un modelo consiste de code frames (estructuras de código), área de parámetros de datos (PDA) y subprogramas relacionados que contienen código de las especificaciones del usuario.

SUBSISTEMA DE ADMINISTRACION:

Se utiliza para crear y mantener los modelos en la generación de los objetos Natural.

Con este subsistema los modelos definidos pueden ser adaptados para agregarles o modificarles funciones de aplicación. Estos modelos pueden modificarse para reforzar los estándares de una organización tales como la seguridad e interfaces de usuario. También se pueden crear nuevos modelos.

SUBSISTEMA DE TEXTO DE AYUDA:

Este se utiliza para crear y mantener el texto de ayuda en línea para cualquier campo, programa o aplicación generada en Construct.

Sólo los textos de ayuda deben ser especificados, las otras necesidades de funcionalidad ya están incluidas en cualquier programa generado en Construct.

SUBSISTEMA DE GENERACION

El menú principal de generación da acceso a los modelos de Construct, sus paneles de especificación y el editor de User Exit, el subsistema de generación permite generar casi cualquier objeto Natural, así como las descripciones de programas en Predict y JCL (es un controlador del lenguaje en el manejo de tarjetas para procesos batch) esto incluye:

• Programas	• Mapas
• Subprogramas	• PDA's
• Rutinas de ayuda	

Se pueden crear 30 modelos de programas en este subsistema, en estos programas se pueden dar altas, bajas, consultas a registros de archivos de la base de datos, etc. A continuación se muestran algunos ejemplos de programas:

• Programas Browse	• Helproutine Browse
• Subprogramas Browse	• Programas Textos de Ayuda
• Programas de Mantenimiento	• Programas de Menú
• Programas Batch	• Mapas
• PDA's	• Programas de finalización
• Programas de iniciación	

A continuación se muestra el menú del Subsistema de Generación.

```

CSMAIN      NATURAL CONSTRUCT 3.2.1      CSGMNM1
Mar 27,93   -GENERATION MAIN MENU      05:46 PM
Code function      Spec Panel Model Options
*****
R  Read specification and object      R    O
M  Modify specification panel        O  O  R
U  User exit editor                  R
G  Generate source from specification      R  O
T  Test generated source              R
E  Edit generated source              R
S  Save specification and source       R    O
W  Stow specification and source       R    O
L  List generated programs             O
C  Clear specification and edit buffer      O
?/  Help terminate

*****
Code.... --Specification/Program _____ Panel ..: __ Library ..... SSGSE
Model ...: _____ Options: __ Object type: Program
Desc ....:
Command:
Enter -PF1--PF2--PF3--PF4--PF5--PF6--PF7--PF8--PF9--PF10--F11--PF12

```

Fig. 3.1 Pantalla del Subsistema de Generación.

Los campos en la parte inferior del panel se utilizan para introducir los parámetros, la leyenda a la derecha de los códigos de función indicando cuales campos son requeridos y cuales opcionales.

Campo	Descripción
Code	Indica el código de la función a utilizar
Specification/ Program	Indica el nombre del programa que se quiere utilizar para la función antes especificada.
Panel	Son números de paneles específicos
Library	Indica la librería en que se esta trabajando
Model	Nombre del modelo a generar.

Campo	Descripción
Options	Aparece una lista de las opciones disponibles para el modelo.
Object Type	Indica el tipo de objeto Natural que se esta generando
Desc	Descripción del modelo especificado.
Command	Comandos Natural que se puedan utilizar. Permite probar programas que se van a ejecutar.

Este subsistema consta de dos editores, el editor de User Exit y el editor de Código generado.

El editor de Código generado es idéntico al editor de Natural, dependiendo del objeto generado ya sea un mapa, un programa o un editor de datos.

User Exit: Este editor se utiliza para insertar procesos especiales en cualquier programa generado, el código introducido en cuando el usuario sale de sesión siempre se preserva tras regeneración. Los Modelos de Construct tienen variedad de User Exit (es donde se permite agregar código del usuario) y aparece una lista de User Exit dependiendo del modelo que se esta generando. A continuación se muestra un ejemplo de la lista de User Exit del modelo de Main Menú.

Exist	Required	Mark User Exits to be generated	Type	Conditional
		X CHANGE-HISTORY	Specif	
		- LOCAL-DATA	General	
		- BEFORE-FETCH		

Enter--PF1--PF2--PF3--PF4--PF5--PF6--PF7--PF8--PF9--PF10--PF11--PF12--
 help rtnm quit bkwrđ frwrđ

SUBSISTEMA DE ADMINISTRACION

Este subsistema se utiliza para mantener definiciones de modelos ya existentes, crear nuevos modelos y definir las especificaciones de las PF's (funciones del teclado).

```

CSDMAIN      ***** NATURAL CONSTRUCT *****      CSDMN1
Mar 08,93    -ADMINISTRATION MENU-                  09:50AM

Code Function
*****
M   Maintain Models
F   Maintain Code Frames
S   Maintain Subprograms
C   Maintain Control Record
O   Compare Models and Code Frames
?   Help
.   Terminate
*****

Code: _
Direct Command: _____
Enter -PF1--PF2--PF3--PF4--PF5--PF6--PF7--PF8--PF9--PF10--F11--PF12

```

Fig. 3.2 Pantalla del Subsistema de Administración.

Código	Función
M	Crea nuevos modelos o modifica los ya existentes
F	Crea nuevos code frame o modifica los ya existentes
C	Mantiene el control de registros, esta función permite darle mantenimiento a las especificaciones de las PF's y establecerles atributos dinámicos tales como color e intensificación de caracteres
S	Da mantenimiento a las PF's y especificaciones a ventanas para modelos de subprogramas
O	Compara los modelos de Construct y code frame que residan en diferentes archivos del sistema de Construct

SUBSISTEMA DE TEXTO DE AYUDA

Permite crear y mantener el texto de ayuda de programas generados en Construct.

CSHMAIN Mar 08,93	**** NATURAL CONSTRUCT **** -HELP TEXT MAIN MENU-	CSHMN1 09:52AM																																																																		
<table border="1"> <thead> <tr> <th>Code</th> <th>Function</th> <th>Minor</th> <th>Major</th> <th>Type</th> <th>Profile</th> </tr> </thead> <tbody> <tr> <td>E</td> <td>Edit the current help text</td> <td>O</td> <td>O</td> <td>O</td> <td>O</td> </tr> <tr> <td>T</td> <td>Test the current help text</td> <td>O</td> <td>O</td> <td>O</td> <td>p</td> </tr> <tr> <td>S</td> <td>Save the current text to file</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> </tr> <tr> <td>L</td> <td>List help text for selection</td> <td>O</td> <td>O</td> <td>O</td> <td></td> </tr> <tr> <td>P</td> <td>Purge help text from file</td> <td>R</td> <td>R</td> <td>R</td> <td></td> </tr> <tr> <td>C</td> <td>Clear the edit buffer</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>H</td> <td>Print hardcopy of saved help text</td> <td>R</td> <td>R</td> <td>R</td> <td></td> </tr> <tr> <td>M</td> <td>Maintain help display profiles</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>?</td> <td>Help</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>.</td> <td>Terminate help maintenance</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>			Code	Function	Minor	Major	Type	Profile	E	Edit the current help text	O	O	O	O	T	Test the current help text	O	O	O	p	S	Save the current text to file	R	R	R	R	L	List help text for selection	O	O	O		P	Purge help text from file	R	R	R		C	Clear the edit buffer					H	Print hardcopy of saved help text	R	R	R		M	Maintain help display profiles					?	Help					.	Terminate help maintenance				
Code	Function	Minor	Major	Type	Profile																																																															
E	Edit the current help text	O	O	O	O																																																															
T	Test the current help text	O	O	O	p																																																															
S	Save the current text to file	R	R	R	R																																																															
L	List help text for selection	O	O	O																																																																
P	Purge help text from file	R	R	R																																																																
C	Clear the edit buffer																																																																			
H	Print hardcopy of saved help text	R	R	R																																																																
M	Maintain help display profiles																																																																			
?	Help																																																																			
.	Terminate help maintenance																																																																			
Code: _____		R=Required																																																																		
Minor: _____		O =Optional																																																																		
Major: _____																																																																				
*Type: P (F,D,P,O)																																																																				
Help Profile SYSTEM _____																																																																				
Enter -PF1-PF2-PF3-PF4-PF5-PF6-PF7-PF8-PF9-PF10-F11-PF12																																																																				

Fig. 3.3 Pantalla del Subsistema del Texto de Ayuda

Parámetro	Función
MINOR	El componente menor especifica un programa o un mapa.
Parámetro	Función
MAJOR	El componente mayor especifica un sistema.
TYPE	Especifica el tipo de texto a darle mantenimiento.

CONSTRUCT Y EL CICLO DE VIDA DE UN SISTEMA

El concepto de ciclo de vida organiza el proceso de desarrollo y mantenimiento de aplicaciones, el cual provee una visión general del sistema para el desarrollo de proyectos de software, provee las siguientes funciones para manipular el proceso.

- Estructura el proceso en fases bien definidas
- Identifica los principales problemas en el desarrollo del sistema
- Especifica un plan de trabajo para cada fecha.

Existen cuatro estados de procesos típicos

- 1) Requerimientos (Análisis)
- 2) Diseño del sistema
- 3) Implantación
- 4) Operación y mantenimiento

REQUERIMIENTOS EN LA FASE DE ANALISIS

En esta fase se examina el proceso a hacer automatizado, se pone especial énfasis en que va a hacer el sistema y como lo va hacer finalmente, se obtiene la especificación de requerimiento que es una descripción detallada de las funciones del sistema. Construct no tiene un impacto significativo en el análisis.

FASE DE DISEÑO DEL SISTEMA

Durante esta fase se diseña la arquitectura del sistema, se crea un diagrama estructural que muestra jerárquicamente los programas y módulos de la aplicación para obtener finalmente un prototipo. Los prototipos se pueden crear con Construct; estos generalmente ponen énfasis en la interfaz con la aplicación y el usuario final.

Genera modelos de mapas, menús y consultas, si es necesario modificar el prototipo se regenera y se hacen los cambios necesarios.

FASE DE IMPLANTACION:

En esta fase se codifican y prueban los programas de módulos individuales.

Construct hace interfaz con Predict para asegurar que la información, con la fase de diseño sea fácilmente accesible para que el código generado encaje correctamente con el contexto del sistema desarrollado, accedendo la información de datos directamente desde Predict los resultados del análisis y diseño sirven para el proceso de generación de código y realizar la documentación del programa, si se crea un prototipo durante la fase de diseño este puede ser utilizado como una base para la fase de implantación, ya que Construct genera código reutilizable, por lo tanto es posible actualizar el prototipo. El editor de texto se utiliza también para que se diseñen las pantallas de ayuda para el usuario final.

FASE DE OPERACION Y MANTENIMIENTO

Una vez que se a instalado el sistema, y que se encuentra en operación, empieza la fase del mantenimiento. Gran parte de esta fase consta de evaluación que muchas veces requiere modificaciones al sistema. Las modificaciones se pueden hacer manualmente utilizando el editor de Natural o realizarlas automáticamente utilizando Construct.

3.6 ADABAS

Adabas (Sistema de Base de Datos Adaptable) surgió en 1971, desarrollado inicialmente para Mainframe de IBM, ahora soporta otras plataformas como Unix etc, es un DBMS amigable que puede manejar grandes volúmenes de datos, es compatible con lenguajes de segunda, tercera y cuarta generación (ejemplos de lenguajes Ada, Ensamblador, C, Natural, ect).

Es un manejador de Base de Datos cuasi-relacional, basado en "listas invertidas" (índices creados por cada descriptor o campo llave, el descriptor es utilizado como un criterio de búsqueda para localizar datos específicos), que establece la independencia entre los datos físicos y las relaciones lógicas en su manejo, permitiendo que éstas últimas sean establecidas y que evolucionen, de acuerdo a las necesidades del usuario final. Asimismo, soporta aplicaciones en línea con un alto volumen de transacciones donde refleja tiempos mínimos de respuesta. Su flexibilidad y eficiencia de listas invertidas también le permite soportar una gran cantidad de procesos Batch conjuntamente a los procesos en línea. Cuenta, además con la ventaja de realizar modificaciones o poner en operación nuevas aplicaciones sin necesidad de interrumpir su servicio.

CARACTERISTICAS

- Relaciones lógicas separadas de las estructuras físicas.
- Argumento de búsquedas basado en valores en oposición a apuntadores físicos. Se define como de tipo relacional.
- Localización a través de las listas invertidas y del asociador o en forma aleatoria por medio de un algoritmo HASH.
- Debido a los aspectos señalados, ADABAS es muy flexible a los cambios, eficiente y fácil de aprender a usar.
- Herramientas a todos los niveles.
- Utilidades en línea para diseño y carga de archivos, recuperación y reinicio; agregado de llaves y campos, ampliación física de la base de datos, auditoría.
- Llamadas (CALLS), estándares y directos.
- Bitácora de Comandos y de protección.
- Control de comandos contra la base de datos a nivel usuario, aplicaciones y terminal.
- Control de transacciones incompletas automáticamente al reinicio de la operación.
- Seguridad a nivel archivo, campo y contenido de campo.
- Encriptado de información.
- Compresión automática de información reduciendo el almacenamiento en disco hasta un 50% y haciendo más eficiente el I/O.
- Reutilización de espacios por supresión de campos, registros y archivos, haciendo innecesarias las reorganizaciones y recargas.

ESTRUCTURA Y ARQUITECTURA

El manejador de la base de datos consiste de Adabas y Predict que es el diccionario de datos el cual almacena los datos y metadatos, también incluye las diferentes interfaces para el acceso a SQL, acceso de texto, etc.

La arquitectura de Adabas varía entre plataformas, en el ambiente DEC VAX bajo VMS trabaja como multiservidor pero en otros ambientes sólo sirve para un servidor.

FUNCIONALIDAD

Adabas soporta múltiples modelos de datos. La vista relacional de datos es accesible desde Natural, su 4GL y a través de SQL, mientras que Adabas soporta la vista relacional de entidades total. La funcionalidad de Adabas puede accederse utilizando Predict o bien implementando Adabas. Con Predict se provee una vista dinámica de los datos.

Aunque Adabas no es puramente relacional soporta el modelo relacional a través de interfaces con SQL, y pueden agregarse campos y tablas dinámicamente a la base de datos, el nivel

general de funcionalidad continua siendo alto. También se pueden tener interfaces para el almacenamiento y recuperación de textos y datos geográficos. Adabas ENTIRE es la interface más reciente para Adabas.

ADABAS VSAM/BRIDGE

Es un importante módulo que permite emigrar nuestra información de archivos VSAM a archivos ADABAS sin sacrificar lo invertido en las aplicaciones existentes bajo VSAM y sin preocuparse de las mismas. Tienen la capacidad de migrar directamente sin necesidad de reprogramar ningún código.

ADABAS/VTAM

Provee el mecanismo ideal para acceder Base de Datos remotas a través de un puente de comunicación nodo a nodo. Esta capacidad permite a los usuarios acceder la Base de Datos ADABAS desde su equipo mientras ésta reside en otro equipo y teniendo interface con la Red completa.

ADABAS/VM

Es la herramienta que nos permite acceder a la Base de Datos ADABAS que resida en cualquier máquina bajo el ambiente de operación CMS/VM aún entre sistemas operativos diferentes (Unix, MVS/XA, VMS, OS/2, etc).

FUNCIONALIDAD DISTRIBUIDA

EFST (ENTIRE Tecnología de Función Servidor), es una tecnología para un ambiente de procesos distribuidos, este provee acceso a funciones, datos, o servicios a computadoras locales o remotas, utilizando comunicación de red donde sea necesario, en forma transparente para el usuario. Este proporciona múltiples servicios así como requerimientos de impresión, que deberían tener diferentes implantaciones en diferentes plataformas de hardware, estos servicios son accesibles a través de comandos estándar de SQL. El objetivo es establecer un protocolo lógico para los requerimientos de comunicación entre cualquier cliente o cualquier servidor. Esto significa que todos los programas (ya sea de clientes o servidores) tendrán la misma vista lógica de datos, servicios de hardware u otros procesos que necesiten invocar, a través de todas las variedades de hardware, sistemas operativos y software de comunicación soportados.

DESEMPEÑO

Adabas contiene estrategias avanzadas de indexado y gran eficiencia en el buffer. Como la mayoría de los DBMS los cuales utilizan arboles B, Adabas utiliza listas invertidas, así como su propio mecanismo para tener un rápido almacenamiento y recuperación de los datos.

UNIDAD DE PROCESAMIENTO CENTRAL

Adabas utiliza los recursos de la Unidad de Procesamiento Central eficientemente con una arquitectura individual para soportar cada plataforma utiliza tipo de datos nativos e implanta un algoritmo eficiente de compresión de datos, cuando es posible Adabas utiliza llamadas de bajo nivel (llamadas a sistema operativo) y en la mayoría de las ocasiones evita servicios de sistemas operativos de alto nivel. Esto contribuye significativamente a la eficiencia del aprovechamiento de la Unidad de Procesamiento Central, es posible utilizar llamadas directas a la base de datos de Adabas

MEMORIA

Los requerimientos de memoria para Adabas dependen del hardware y software del ambiente bajo el cual operan.

I/O DEL DISCO

Las estructuras de datos de las listas invertidas generan menos llamadas al I/O del disco. Esto puede ser claramente observado durante las actualizaciones. La lista invertida es una estructura flexible de datos que se caracteriza por tener OLPT (Procesos de Transacciones En-Línea) y transacciones de consulta. Es extremadamente rápida para consultas que sólo accesan el índice de datos y es ideal para optimizar el trabajo.

Adabas soporta un nivel en línea más eficiente que los arboles B. El Adabas de servicios en-línea forman una herramienta de administración interactiva de la base de datos para optimizar los recursos optimizados.

CARACTERISTICAS TECNICAS DE DATOS

MODELO DE DATOS SOPORTADOS

- Relacional
- Texto
- Entidad/Relación
- Secuencial, indexado, directo
- Red y jerárquico

TIPOS DE DATOS

- Arreglos dimensionales
- Arreglos de longitud variable
- Carácter fijo
- Carácter variable
- Entero
- Decimal fijo
- Punto flotante
- Fecha y tiempo
- Virtual (calculado)
- Binario

LIMITE DE DATOS

- 16 K máximo de columnas o tamaño del campo
- 926 campos por registro
- 4.16 millones de registros por archivo

DEFINICION DE DATOS

- Diccionario de datos
- Archivos catalogados
- Selección del usuario en la compresión del campo
- Compresión automática en campos múltiples

INDEXACION

- Indexado automático
- Composición en el indexado del campo
- Llave opcional de indexado
- Creación de nueva llave en la reorganización de la base datos
- Llaves fonéticas
- Indexado de columnas o campos particionados

CARACTERISTICAS DEL LENGUAJE DE LA BASE DE DATOS

- Compilado
- Interprete
- Interactivo con SQL.

ESTANDARES

- Compresión de datos
- Compresión en el indexado
- Reorganización
- Modificación de archivos y datos
- Datos cargados
- Backup y restauración
- Incremento en la restauración y respaldo
- Diagnóstico completo

SEGURIDAD Y CAPACIDAD DE INSPECCION

- Encriptación de datos
- Protección en la aplicación
- Protección de datos
- Contraseña de protección
- Comando logging y herramientas de inspección

RESTRICCIÓN EN LOS NIVELES DE ACCESO

- Usuario, base de datos, diccionario, tablas, registros, campos, grupos y el valor del nivel.

INDEXACION

- Indexado automático
- Composición en el indexado del campo
- Llave opcional de indexado
- Creación de nueva llave en la reorganización de la base datos
- Llaves fonéticas
- Indexado de columnas o campos particionados

CARACTERISTICAS DEL LENGUAJE DE LA BASE DE DATOS

- Compilado
- Interprete
- Interactivo con SQL.

ESTANDARES

- Compresión de datos
- Compresión en el indexado
- Reorganización
- Modificación de archivos y datos
- Datos cargados
- Backup y restauración
- Incremento en la restauración y respaldo
- Diagnóstico completo

SEGURIDAD Y CAPACIDAD DE INSPECCION

- Encriptación de datos
- Protección en la aplicación
- Protección de datos
- Contraseña de protección
- Comando logging y herramientas de inspección

RESTRICCION EN LOS NIVELES DE ACCESO

- Usuario, base de datos, diccionario, tablas, registros, campos, grupos y el valor del nivel.

LENGUAJES SOPORTADOS

- Ada, Ensamblador, Macro, C, COBOL, FORTRAN, NATURAL, PL/1, PASCAL y SQL.

AMBIENTES SOPORTADOS

- IBM y PCM: MVS/XA, MVS/ESA, VSE, VSE/ESA, VM/ESA, OS/2
- Digital: Open VMS
- Fujitsu: OSIV/F4, MSP, MSP/AE
- Siemens/Nixdorf: BS2000
- Wang: VS
- UNIX: HP, SUN, AT&T (NCR), IBM, SCO, ICL

CAPITULO 4

METODOLOGIA

Uno de los objetivos en una empresa es contar con una información actualizada, ya que ésta evoluciona constantemente. Esta es la razón por lo que un sistema resulta en poco tiempo obsoleto, entonces la empresa tiene la necesidad de crear y modificar sus aplicaciones mucho más rápido requiriendo de una metodología para dicha actualización.

Siguiendo el ciclo de vida del software tradicional se llega a tomar de dos a tres años construir un sistema, para entonces la empresa no podrá cumplir con sus objetivos estimados (no podrá ser lo suficientemente competente). Siendo el factor crítico la velocidad con que se libere alguna aplicación.

En éste capítulo se describirá la metodología para la elaboración del sistema, dicha metodología es la que se utiliza en la empresa, conocida como :

- Análisis y Diseño Estructurado
- Técnicas de RAD

4.1 ANALISIS Y DISEÑO ESTRUCTURADO

La metodología de Análisis y Diseño Estructurado se basa en las técnicas propuestas por Yourdon, Gane & Sarson, Jackson.

El análisis y diseño estructurado como todos los demás métodos, es una actividad de diseño y construcción de modelos (mapas, diagramas), con una estandarización de notaciones para representarlos, permitiendo reflejar el flujo y contenido de información.

El análisis y diseño estructurado se basa en tener un control de lo que se está realizando, con ayuda de la definición de modelos de diagramación nos permite visualizar la información. La utilización de las herramientas CASE logra que resulte más dinámico y con un mejor mantenimiento de la consistencia y la corrección de la información.

En la siguiente tabla se muestra los requerimientos mínimos para las tres etapas críticas del ciclo de vida del software.

Si se cumple con lo que expuesto en la tabla se tendrá una aplicación completa y se evitará algún retraso o un sistema que no cubra con todos los requerimientos definidos por el usuario en un inicio.

CAPITULO 4

METODOLOGIA

Uno de los objetivos en una empresa es contar con una información actualizada, ya que ésta evoluciona constantemente. Esta es la razón por lo que un sistema resulta en poco tiempo obsoleto, entonces la empresa tiene la necesidad de crear y modificar sus aplicaciones mucho más rápido requiriendo de una metodología para dicha actualización.

Siguiendo el ciclo de vida del software tradicional se llega a tomar de dos a tres años construir un sistema, para entonces la empresa no podrá cumplir con sus objetivos estimados (no podrá ser lo suficientemente competente). Siendo el factor crítico la velocidad con que se libere alguna aplicación.

En éste capítulo se describirá la metodología para la elaboración del sistema, dicha metodología es la que se utiliza en la empresa, conocida como :

- Análisis y Diseño Estructurado
- Técnicas de RAD

4.1 ANALISIS Y DISEÑO ESTRUCTURADO

La metodología de Análisis y Diseño Estructurado se basa en las técnicas propuestas por Yourdon, Gane & Sarson, Jackson.

El análisis y diseño estructurado como todos los demás métodos, es una actividad de diseño y construcción de modelos (mapas, diagramas), con una estandarización de notaciones para representarlos; permitiendo reflejar el flujo y contenido de información.

El análisis y diseño estructurado se basa en tener un control de lo que se esta realizando, con ayuda de la definición de modelos de diagramación nos permite visualizar la información. La utilización de las herramientas CASE logra que resulte más dinámico y con un mejor mantenimiento de la consistencia y la corrección de la información.

En la siguiente tabla se muestra los requerimientos mínimos para las tres etapas críticas del ciclo de vida del software.

Si se cumple con lo que expuesto en la tabla se tendrá una aplicación completa y se evitará algún retraso o un sistema que no cubra con todos los requerimientos definidos por el usuario en un inicio.

REQUERIMIENTOS POR ETAPA

ANALISIS	DISEÑO	CONSTRUCCION
DATOS - DATOS IMPORTANTES DE LA EMPRESA. - MODELO DE DATOS	- DISEÑO DE REGISTROS.	- VISTA DE DATOS POR PROGRAMA APLICATIVO.
ACTIVIDADES - PANORAMA ESTRATEGICO DE SUS METAS. - PROCESOS NECESARIOS Y COMO SE REALICIONAN.	- DISEÑO DE PROCESOS	- CODIFICACION DETALLADA DE PROCESOS.
TECNOLOGIA - ESTRATEGIA TECNOLÓGICA DEL NEGOCIO. - IDENTIFICACION DE LA TECNOLOGIA.	- DEFINICION DE HARWARE Y SOFTWARE BASE.	- APLICACION DEL HARDWARE Y SOFTWARE.

Fig. 4.1

Para cubrir las etapas es necesario llevar un control todo su ciclo de vida. El desarrollo del análisis deberá apegarse a un plan de trabajo y éste deberá especificar las diferentes actividades a realizar, los responsables de dichas actividades y las fechas planeadas de terminación.

Existen diversos tipos de calendarización, uno de los más usados y que se utilizó para el proyecto es el diagrama de Gantt.

4.1.1 METODOLOGIA DE ANALISIS

CALENDARIO DE REALIZACION

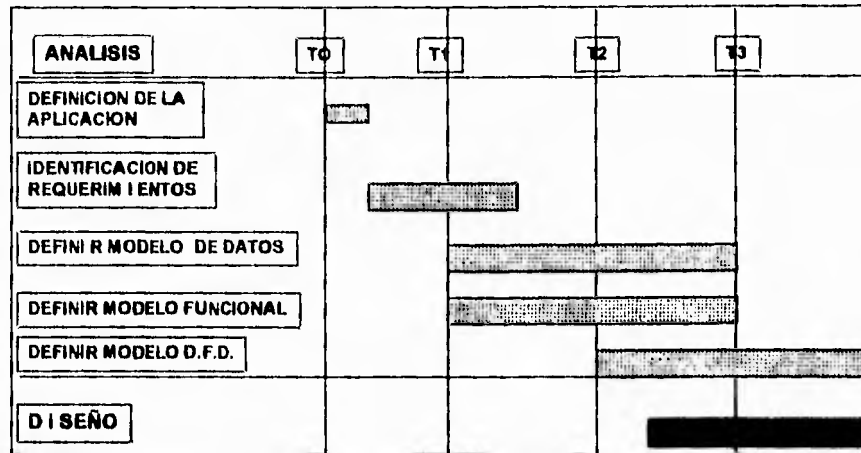


Fig. 4.2

Este calendario que propone el análisis estructurado para su correcta realización. Como puede observarse en el calendario existen actividades que se pueden realizar en paralelo y otras necesariamente necesitan un antecesor como es el caso de la identificación de requerimientos el cuál le antecede la definición del entorno de la aplicación, no se puede empezar a atacar el problema sin antes conocer su historia, el ambiente en que se desenvuelve y su interacción con otras aplicaciones si es el caso.

En la siguiente figura se ejemplifica lo que es definir el entorno de la aplicación y la información que se debe obtener.

4.1.2 DEFINICION DEL ENTORNO DE LA APLICACION

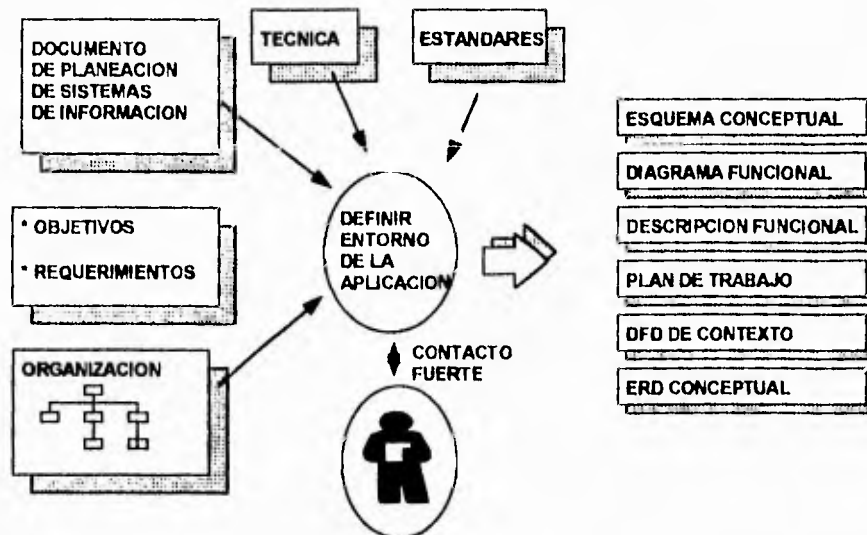


Fig. 4.3

ANALISIS DEL AREA DE NEGOCIO

- Identifica los requerimientos de información en el área de negocio.
- Crea un modelo de información que refleje la interrelación entre los datos y las actividades de la empresa.
- Identifica los cambios deseables por el usuario en el modelo existente de información.
- Evalúa la construcción del sistema de información.

ENTORNO DE LA APLICACION

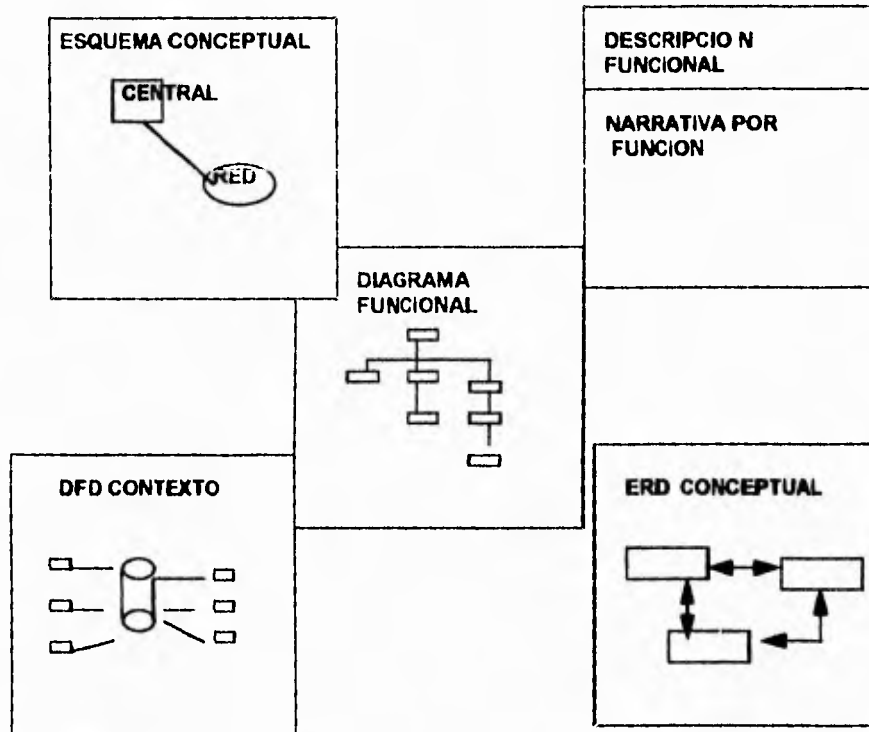


Fig. 4.4

Como se puede ver el tipo de información que se obtiene al definir el entorno de la aplicación es muy representativa y nos ayudará en gran medida para dar seguimiento y poder pasar a la etapa de diseño. Esto es realmente sencillo si se está trabajando con alguna herramienta de diagramación.

El problema se empieza a definirse en esta etapa.

4.1.3 DOCUMENTACION DE ANALISIS



Fig. 4.5

Como se puede observar en la figura se hace un recuento de la información que se obtiene con la metodología de análisis y diseño estructurado.

Una vez obtenida la información y como se observo en el calendario de análisis cuando se tiene un avance de las ultimas actividades (definición del modelo de datos, modelo funcional y DFD). Con esto se empieza la etapa de Diseño por lo que no se define cuando termina una etapa y empieza otra.

A continuación se esquematizará los pasos necesarios para un diseño estructurado.

Las técnicas de la metodología del diseño estructurado se basa en los siguientes pasos:

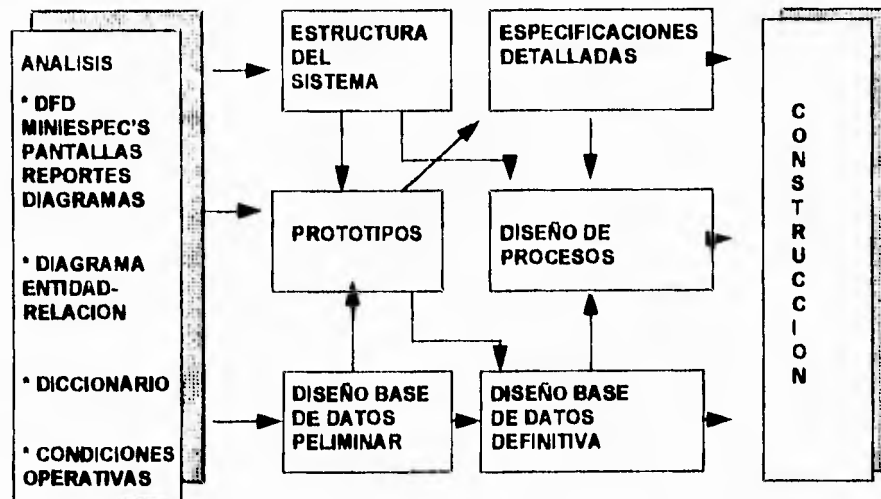


Fig. 4.6

4.1.4 DISEÑO DE SISTEMAS

- Define como los procesos y datos seleccionados en el análisis, pueden ser automatizados.
- Transforma los requerimientos del análisis en especificaciones detalladas de la aplicación.
- El diseño deberá satisfacer los requerimientos del usuario apoyado en prototipos.
- Se determina el ambiente operativo de las aplicaciones.

Como se observa en la figura, con los datos resultantes del análisis son la entrada para el diseño, información que al conjuntarse con una herramienta CASE resulta más tangible para el usuario, al decir más tangible es porque en la actualidad éstos producen prototipos desde la etapa de diseño y el usuario empieza a ver los resultados más rápidos y visualizarlos, todo esto con ayuda de las técnicas de RAD la cual se describe más adelante.

4.1.5 METODOLOGIA DE DISEÑO
CALENDARIO DE REALIZACION

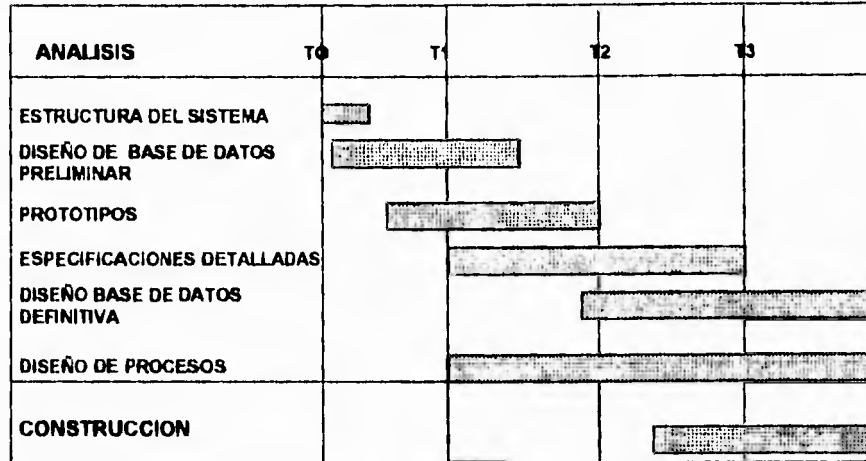


Fig.

4.7

Este calendario al igual que en la etapa de análisis, nos sugiere el seguimiento para el diseño estructurado, las actividades contempladas y recordando que debe diseñarse un diagrama de Gantt.

4.1.6 ESTRUCTURA DEL SISTEMA

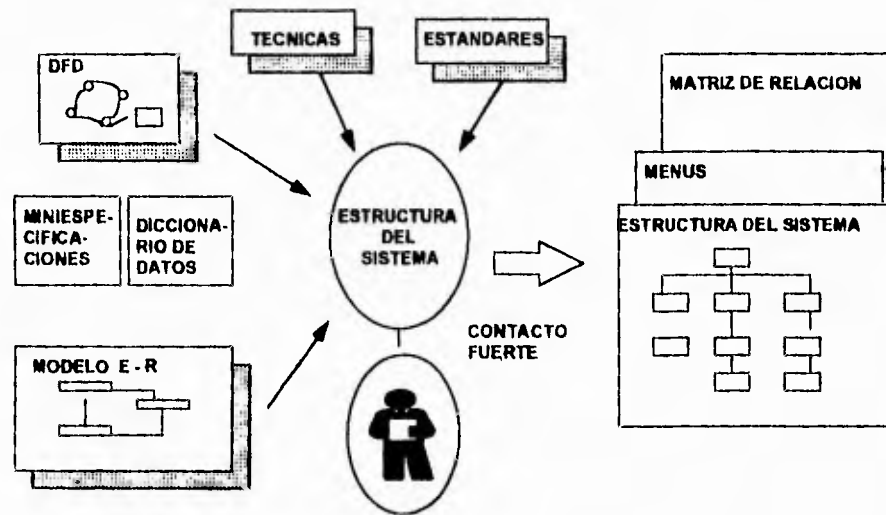


Fig. 4.8

En la definición de la estructura del sistema se debe de procurar que el usuario realmente participe con nosotros, lograr una relación fuerte. Es en esta etapa donde el usuario es quien diseña su sistema y al ser diseñado por él, se evitará la etapa de mantenimiento.

4.1.7 DISEÑO BASE DE DATOS PROTOTIPO

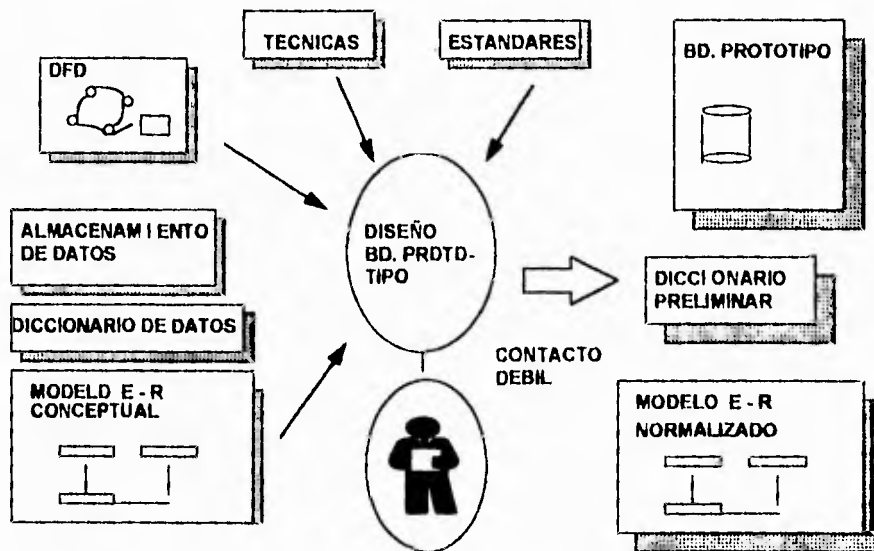


Fig. 4.9

En cuanto al diseño de la base de datos la participación del usuario es débil ya que solo se le consultará para la definición de algunas entidades.

4.1.8 PROTOTIPOS

PROPOSITOS:

- Tener mayor iteración con el usuario.
- Examinar alternativas más realistas.
- Desarrollo completo del sistema.

HERRAMIENTAS

- Lenguajes de 4a. Generación.
- Sistema administrador de Base de Datos.
- Generador de código y aplicaciones.

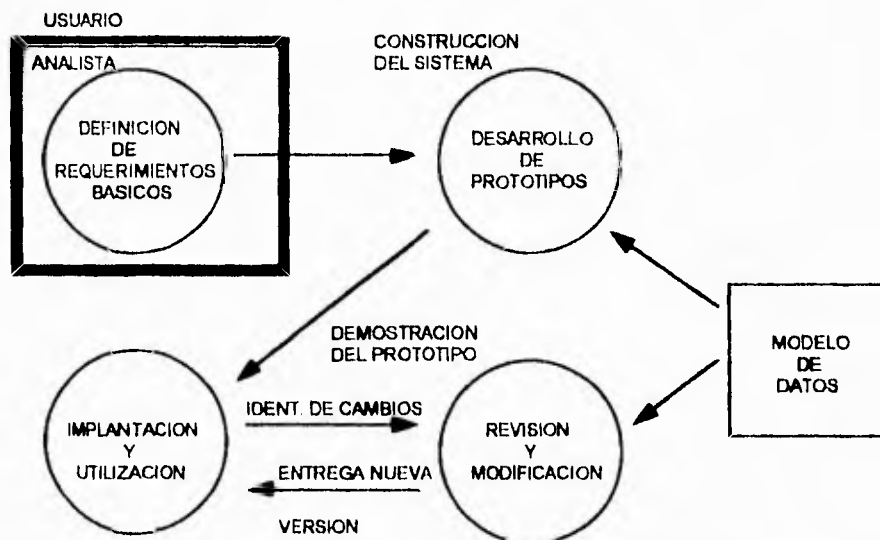


Fig. 4.10

Con los prototipos el usuario puede ver realmente como nace su sistema y no como tradicionalmente se hacía, donde el usuario participa en el inicio y final del ciclo de vida de su sistema

4.1.9 DISEÑO BASE DE DATOS - DEFINITIVA

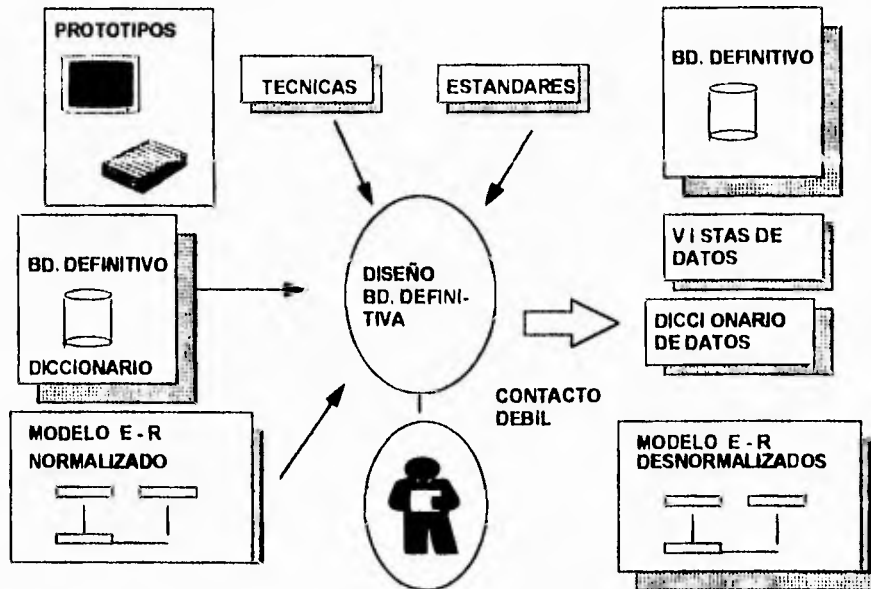


Fig. 4.11

Aquí ya se tiene definida la base de datos definitiva, con la ayuda de los prototipos se logra definir rápidamente. La construcción de vistas donde el usuario logra visualizar su sistema.

4.1.10 DISEÑO DE PROCESOS

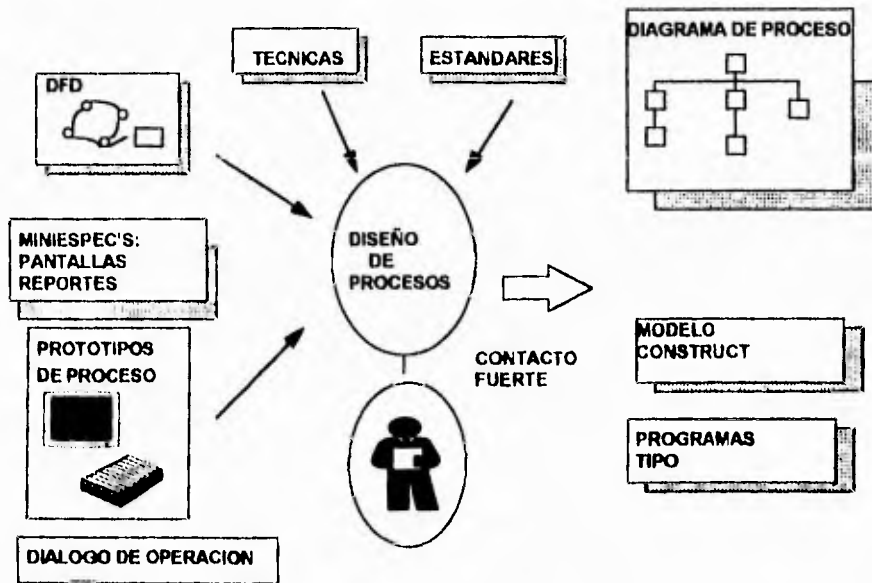


Fig. 4.12

Para el diseño de procesos se necesita la opinión del usuario constantemente y pasar a su construcción físico del sistema. Como puede observarse en la figura 4.12, los datos resultantes de esta fase son para la generación de código de programación.

4.1.11 DOCUMENTACION DEL DISEÑO

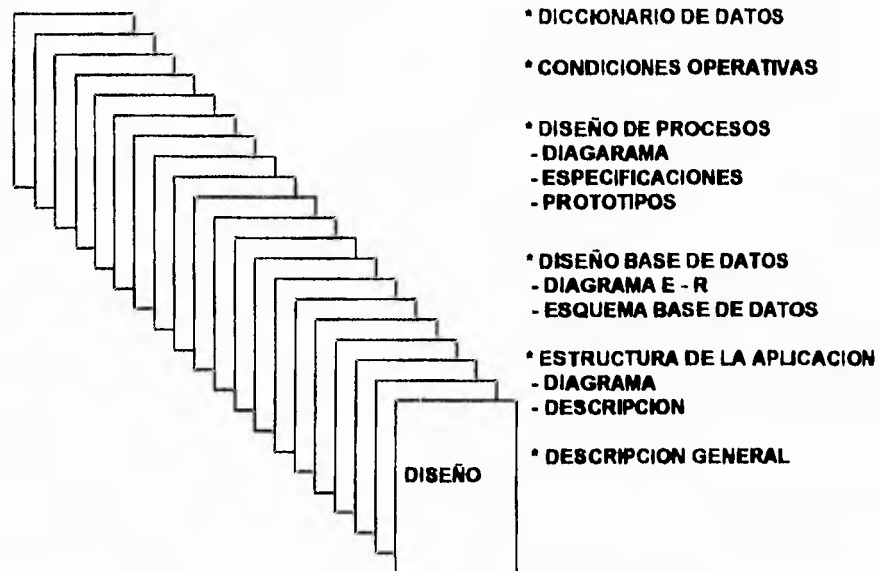


Fig. 4.13

Esta es la documentación que se debe obtener en el diseño estructurado alguna información será ocupada para la construcción del sistema (programas y base de datos).

4.1.12 CONSTRUCCION DE SISTEMAS

- Crear el sistema de información de acuerdo a las especificaciones del diseño.
- Emplear el prototipo como parte de las especificaciones del sistema.
- Desarrolla procedimientos para su operación y mantenimiento.
- Se realiza la capacitación de los usuarios.

En la construcción lo único que hay que hacer es bajar la información definida en el diseño restando entregarla a los programadores y al administrador de la base de datos, por lo que el usuario desaparece en esta etapa.

Una vez que el sistema es terminado (ligándose todos los prototipos y programas generados) pasa a la revisión por parte del usuario, si se siguieron los pasos vistos anteriormente no debe existir correcciones o en su defecto serán mínimas.

4.2 TECNICAS DE RAD

RAD (Rapid Application Development ó Desarrollo Rápido de Aplicaciones) es un conjunto de técnicas enfocadas para dar resultados más rápidos y de mayor calidad en comparación con los métodos tradicionales. RAD esta diseñada para aprovechar al máximo las herramientas CASE.

RAD es un proceso en el ciclo de vida del software, desarrollado por James Martin . RAD permite a las organizaciones responder rápidamente y estratégicamente a sus requerimientos.

Permite conocer las necesidades de la organización para :

- Entregar productos al mercado rápidamente
- Incrementar la calidad mientras se reduce el costo
- Trabajar astutamente con pocos recursos

RAD es una propuesta para mantener los requerimientos combinando cinco técnicas para el desarrollo de aplicaciones, cada una contribuye al mejoramiento de la productividad.

Estas técnicas permiten que las aplicaciones se desarrollen rápidamente, con alta calidad y bajo costo.

Las técnicas usadas por RAD son:

- 1) Equipos Pequeños: RAD se enfoca en un grupo pequeño con capacitación intensiva (llamados equipos SWAT) en el uso de herramientas I-CASE, las cuales manejadas adecuadamente, pueden incrementar la productividad del software de un rango de 8 a 13 funciones punto por persona-mes con una meta de 20 funciones punto por persona-mes durante la fase de construcción de una aplicación. Una función punto es una medida de complejidad del software, correspondiendo aproximadamente a 105 líneas de código fuente de COBOL. Esto es una métrica de producción.
- 2) Rápidos Prototipos: RAD esta basado en la evolución del ciclo de vida del software y los prototipos de una aplicación estan constantemente desarrollandose para la producción de un sistema final. Los prototipos no son un desperdicio, ya que nos estan constantemente alimentando a detalle para el diseño de la información.
- 3) Integración de Herramientas CASE: RAD trabaja mejor con la integración de herramientas CASE disponibles en el mercado. Las herramientas I-CASE incorpora un generador de código, mantiene código para múltiples ambientes, incluyendo Pc's, Mainframes, Workstations.
- 4) JAD Interactivo: La comunicación de las especificaciones de los usuarios a los analistas es mucho más fácil con las técnicas de JAD (Joint Applications Development). El grupo de trabajo RAD captura los requerimientos y diseña la información en el I-CASE. El personal utiliza el I-CASE para producir un modelo de la aplicación, mientras interactua con el usuario final en la sesión de trabajo.
- 5) Límite Rígido en el Tiempo de Desarrollo: RAD impone un tiempo rígido para especificar los requerimientos, construcción de prototipos y la construcción de aplicaciones. Las aplicaciones estan construidas por una secuencia de prioridades por los grupos SWAT en un periodo de desarrollo limitado, típicamente suele ser de 90 días.

4.2.1 ALTA CALIDAD

Se suele decir que los desarrollos rápidos significan "rápido y sucio", lo cual resulta ser falso utilizando RAD. RAD cumple la calidad ya que satisface de la mejor manera las necesidades de los usuarios y de la empresa, creando sistemas:

- Sistemas que pueden evolucionar constantemente
- El desarrollo de sistemas que se ajusten a la estructura de la ingeniería de información.
- Construcción de sistemas complejos

RAD define la calidad como "En el momento en que el sistema es liberado, cubre efectivamente los requerimientos del negocio ó del usuario. Esto lo resuelve al acortar el tiempo entre la definición del usuario y la liberación del sistema. Para que los resultados sean satisfactorios se requiere tener una estrecha relación con el usuario en todas las etapas.

Los factores que perfeccionan la calidad mientras se incrementa la velocidad de desarrollo incluye:

- Limitar el involucramiento del usuario final en el diseño del sistema y en todas las fases de desarrollo.
- Prototipos para ayudar a los usuarios a visualizar la funcionalidad del sistema y participar en este.
- Uso de CASE para detectar errores lógicos en el diseño de especificaciones y generar código automáticamente.
- Uso de un deposito de información de diseño para facilitar el reuso de modelos de datos correctos y procesos.

Por lo tanto RAD cumple con la alta calidad, desarrollo rápido y por lo tanto un bajo costo de mantenimiento.

4.2.2 PROCESO DE DESARROLLO RAD

Como lo muestra la figura A., la metodología RAD tiene 4 fases:

- 1) **PLANEACION DE REQUERIMIENTOS:** La primera fase de RAD incluye una esforzada investigación tres días de JRP (Joint Requirement Planing) . El objetivo es definir las funciones del negocio que el sistema soportará, para especificar el alcance del sistema y para proporcionar una justificación de costo para el sistema con un alto nivel en los modelos de entidades y procesos.
- 2) **DISEÑO :** La fase de diseño incorpora dos talleres JAD cuya función es completar el diseño detallado de los datos, modelos de procesos, correr prototipos de componentes críticos del sistema.

- 3) **CONSTRUCCION:** Durante la tercera etapa, las transacciones individuales del negocio definidas en la fase anterior, estan construidas por un pequeño grupo de desarrolladores (llamados SWAT - Skiller With Advanced Tools: Expertos con herramientas avanzadas) y con un periodo limitado de desarrollo. El gupo SWAT trabaja estrechamente con los usuarios para redefinir los requerimientos funcionales y probar las transacciones definidas del negocio.
- 4) **LIBERACION:** La cuarta fase incluye entrenamiento con el usuario final, conversión de bases de datos y revisión del sistema para un estado de producción.

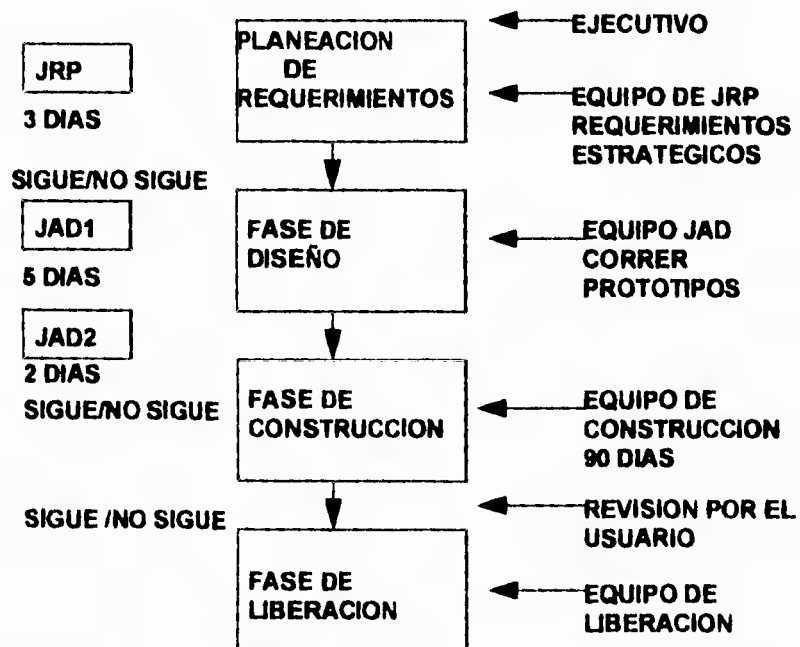


Fig. 4.13

4.2.3 MIEMBROS PRIMARIOS DEL GRUPO DE USUARIOS

- **Ejecutivo Responsable:** Un administrador de alto nivel quien hace el presupuesto para el desarrollo, el ejecutivo responsable típicamente viene del lado del usuario final pero también puede venir de una organización de desarrollo de productos. La primera responsabilidad del ejecutivo es eliminar los obstáculos burocráticos para un desarrollo rápido.
- **Coordinador del Usuario:** Usuario designado por el ejecutivo responsable para manejar el proyecto desde la perspectiva del usuario.
- **Grupo de Planeación de Requerimientos:** Estratégicamente orientado al grupo de usuario final quien participa en la planeación de requerimientos durante la sesión de JAD.
- **Grupo de Diseño:** Operacionalmente orientado al grupo de usuarios finales, participando en el diseño durante la sesión de JAD.
- **Grupo Asistente de Construcción:** Grupo del usuario final que trabaja con el grupo SWAT, durante la fase de construcción para probar cada transacción, producir documentación y preparar la liberación.
- **Revisión por el Usuario:** Grupo de usuarios que revisan el sistema completo al final de la fase de construcción. El grupo decide si el sistema satisface las necesidades de los usuarios y preparar la liberación para la producción.
- **Adicionalmente el usuario que puede participar en el proceso de desarrollo RAD incluye a los especialistas:**
- **Administrador del Proyecto Estratega de Información:** El analista estratega de información es asignado por un alto nivel para administrar el proyecto. El administrador del proyecto puede doblar al líder del grupo SWAT asignado en la fase de construcción.
- **Líder de la Sesión RAD:** También llamado Facilitador. Esta persona está específicamente entrenada para liderar la sesión JAD en la planeación de requerimientos y durante la fase de diseño con el usuario.
- **Documentador:** Analista de Estrategias, experto en el uso de I-CASE, quien participa en las sesiones JAD y almacena las especificaciones del diseño en el I-CASE.

4.2.3 MIEMBROS PRIMARIOS DEL GRUPO DE USUARIOS

- **Ejecutivo Responsable:** Un administrador de alto nivel quien hace el presupuesto para el desarrollo, el ejecutivo responsable típicamente viene del lado del usuario final pero también puede venir de una organización de desarrollo de productos. La primera responsabilidad del ejecutivo es eliminar los obstáculos burocráticos para un desarrollo rápido.
- **Coordinador del Usuario:** Usuario designado por el ejecutivo responsable para manejar el proyecto desde la perspectiva del usuario.
- **Grupo de Planeación de Requerimientos:** Estratégicamente orientado al grupo de usuario final quien participa en la planeación de requerimientos durante la sesión de JAD.
- **Grupo de Diseño:** Operacionalmente orientado al grupo de usuarios finales, participando en el diseño durante la sesión de JAD.
- **Grupo Asistente de Construcción:** Grupo del usuario final que trabaja con el grupo SWAT, durante la fase de construcción para probar cada transacción, producir documentación y preparar la liberación.
- **Revisión por el Usuario:** Grupo de usuarios que revisan el sistema completo al final de la fase de construcción. El grupo decide si el sistema satisface las necesidades de los usuarios y preparar la liberación para la producción.
- Adicionalmente el usuario que puede participar en el proceso de desarrollo RAD incluye a los especialistas:
- **Administrador del Proyecto Estratega de Información:** El analista estratega de información es asignado por un alto nivel para administrar el proyecto. El administrador del proyecto puede doblar al líder del grupo SWAT asignado en la fase de construcción.
- **Líder de la Sesión RAD:** También llamado Facilitador. Esta persona está específicamente entrenada para liderar la sesión JAD en la planeación de requerimientos y durante la fase de diseño con el usuario.
- **Documentador:** Analista de Estrategias, experto en el uso de I-CASE, quien participa en las sesiones JAD y almacena las especificaciones del diseño en el I-CASE.

- **Equipo de Construcción:** También llamado grupo SWAT, un grupo pequeño de tres ó cuatro analistas IS, quienes tienen un alto entrenamiento en la metodología RAD y el uso de Herramientas I-CASE. Un proyecto pequeño requiere un equipo SWAT, pero si el proyecto es grande se requerirán múltiples equipos SWAT.
- **Experto en Modelos de Datos:** Cada proyecto tiene un experto en modelado de datos asignado para ser el responsable de la creación y coordinación de modelos de datos en el proyecto.

4.2.4 RAD - RAPIDO DESARROLLO DE APLICACIONES

En la figura se muestra como interactua RAD durante el ciclo de vida del software basandonos en la herramientas CASE con las que fue realizado el sistema de Administración Ejecutiva de Proyectos.

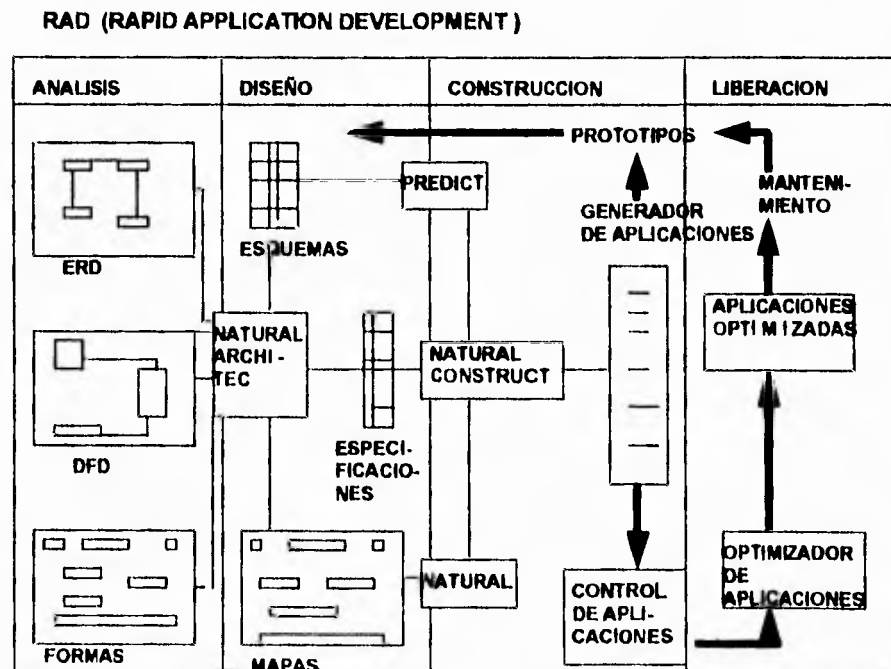


Fig. 4.14

4.2.5 JAD (JOINT APPLICATION DESIGN, DISEÑO DE APLICACIONES EN GRUPO)

Es una técnica mediante la cual se realizan sesiones de trabajo que permiten definir fácil y rápidamente las especificaciones de los procesos que conforman un sistema, con la participación de los usuarios finales.

ANTECEDENTES

La necesidad de instrumentar una técnica que permita el acercamiento de los usuarios finales a sistemas, mediante una comunicación adecuada obedece a problemas tales como:

- Usuarios que no participan en el desarrollo de sus aplicaciones.
- Usuarios insatisfechos.
- Aplicaciones no implementadas.
- Aplicaciones con mantenimiento constante.
- Mala imagen del área de sistemas ante los usuarios.

Adicionalmente la necesidad de desarrollos más rápidos y de mejor calidad.

BENEFICIOS DEL JAD

- JAD involucra a los usuarios finales en el proceso de diseño y ayuda a evitar usuarios insatisfechos.
- Reemplaza especificaciones voluminosas en papel, con diseños tangibles, mediante: pantallas, reportes, prototipos, diagramas de diseño.
- Minimiza tiempos de especificaciones de diseño.
- Incrementa la calidad del sistema.
- Reduce costos evitando la preprogramación o modificación de diseños inadecuados.
- Reduce gastos de mantenimiento.
- Se evita conflictos con usuarios finales.

4.2.6 PROTOTIPOS

Los prototipos aplicados en nuestra metodología son de gran importancia por lo que se explicarán algunas características de estos.

PROPIEDADES

- Funcionales.
- Modelo en miniatra del sistema propuesto.
- Fácil de modificar.
- Escrito por lo general en un 4GL.
- Modelo usado para determinar que tan correcto fue el diseño.
- Promueve la comunicación entre el usuario y el desarrollador.

DESARROLLANDO EL PROTOTIPO INICIAL

OBJETIVOS

- Presentar al usuario un modelo rápidamente.
- Que el modelo sea fácil de modificar.
- Incluir datos que sean familiares al usuario.
- Usar una estructura de datos flexible.

MANERA DE HACERLO

- Crear la base de datos.
- Crear datos de prueba o copiar datos.
- Crear menús.
- Crear las funciones de más alto nivel y que sean las más importantes.

DEMOSTRACION DEL PROTOTIPO

- No más de una hora de demostración.
- El objetivo es encontrar defectos, no arreglarlos.
- Se debe de dar a los usuarios documentación anterior y posterior a la presentación.
- Indicarle al usuario el tiempo con que cuenta para estudiar la información proporcionada.
- Fijar la fecha para la siguiente presentación, que no sea muy lejana con el objeto de darle continuidad al proceso.

DERIVACION DE DISEÑO

- Hasta ahora no se ha codificado.
- Codificación final con 4GL.
- Si es 3GL:
 - Código estructurado.
 - Código bien estructurado.
- Checar acoplamiento entre módulos.
- Acabar de definir especificaciones de los procesos.
- Acabar de definir atributos de las entidades.
- Definir documentación.

CAUSAS DE FALLA DE PROTOTIPOS

- Prototipo significa fin a la "parálisis del análisis"
- Prototipos rápidos y sucios.
- No llevar a cabo revisiones formales.
- Usuarios no convencidos.
- Mala definición de responsabilidades del usuario.
- Entregas prematuras.
- Resistencia del equipo.
- Equipo de trabajo no capacitado

QUE ES UN PROTOTIPO DE SOFTWARE

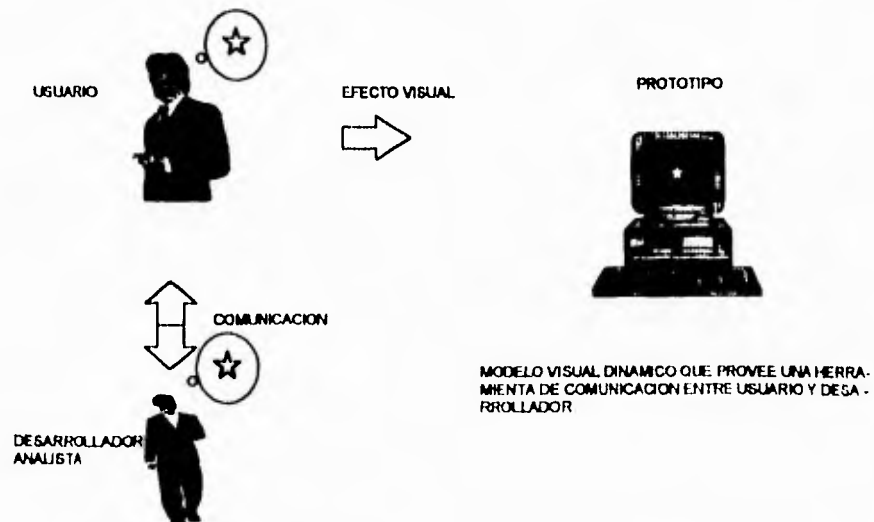


Fig. 4.15

Los seres humanos casi nunca realizan una tarea compleja correctamente la primera vez. Sin embargo, la gente es extraordinariamente buena, realizando muchos refinamientos y adecuaciones pequeñas. Los métodos de desarrollo deben explotar este tipo de iteración, en lugar de combatirlo.

CAPITULO 5

ANALISIS, DISEÑO Y CONSTRUCCION DEL SISTEMA

El objetivo de realizar el análisis de requerimientos consiste en identificar las necesidades de información de los usuarios y producir un modelo conceptual de estas para obtener una visión a muy alto nivel del sistema para conocer los servicios principales que debe ofrecer, así también como sus restricciones.

5.1 ANTECEDENTES

Software AG de México, S. A. de C.V. Es una empresa con un giro comercial que abarca:

- Venta de soluciones informáticas integrales, conformadas por software de base.
- Sistemas desarrollados a la medida y paquetes (NATURAL, PREDICT, ARCHITEC).
- Servicios de consultoría especializados.

Con una misión " Ser el mejor aliado estratégico de nuestros clientes y proveedores, que deseen alcanzar mayor eficiencia y calidad, a través de soluciones integrales de informática oportunas y confiables, que garanticen el logro de sus objetivos y permitan la satisfacción de nuestros socios, la realización profesional de nuestros colaboradores y el desarrollo de nuestro país."

HISTORIA

- Software AG nace en Darmstadt, Alemania en 1969.
- En 19971 implementó comercialmente por primera vez el manejador de Base de Datos "ADABAS" en México.
- Se presentó NATURAL como un entorno de desarrollo de 4ta Generación
- Se funda Software AG de México, con la participación de Software AG de Norteamérica y un grupo nacional asociado a la tecnología desde hace más de 10 años.

ESTRATEGIA ORGANIZACIONAL

Ofrecer a los clientes soluciones integrales, para la automatización de su negocio, buscando convertirse en su Socio Tecnológico (Principal proveedor de tecnología informática)

PRODUCTOS Y SERVICIOS

Software AG ofrece a sus clientes un conjunto de productos y servicios integrales, ya que actualmente las necesidades de información se han dirigido hacia el objetivo de contar con sistemas enlazados entre áreas específicas.

Se describirán a continuación los productos y servicios que SAG ofrece a sus clientes.

Servicios:

- Planeación
- Consultoría
- Desarrollo
- Capacitación
- Implantación

Aplicaciones:

- ARGENTA
- ISIS
- FACTOR-AG
- ARREND-AG

Tecnología de Punta:

- Super Natural
- Adabas
- Predict
- Natural for windows
- Natural Architect
- Natural Construct

5.2 SITUACION ACTUAL

Recientemente la empresa ha detectado un problema en cuanto al tiempo de respuesta de información de una cuenta (El flujo de información). Cuando un cliente tiene una queja sobre un servicio/producto, los directivos necesitan conocer el estado del mismo. Lo cual actualmente resulta ser un proceso tardado debido a que no se tiene automatizado el acceso a la información en las diferentes áreas.

Cada Gerencia controla su información, por lo que no existe un banco de información integral.

Esta situación representa pérdida de tiempo en caso de consulta, lo cual es un factor crítico para la prestación del servicio con la calidad que se exige la empresa, además de existir redundancia en la información con la latente posibilidad de tener inconsistencia en la misma. Lo cual repercute en los costos y calidad de los servicios que brinda la empresa.

A continuación se muestra la estructura organizacional de la empresa.

ORGANIGRAMA DE EMPRESA

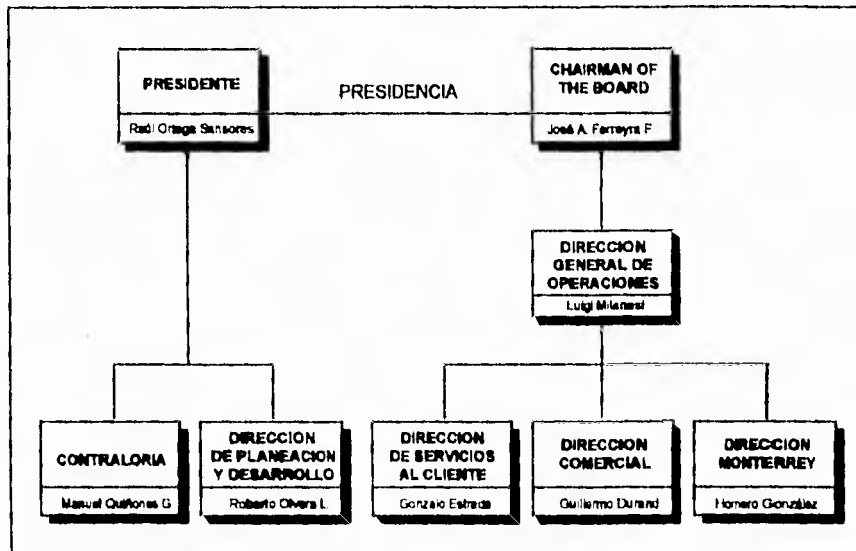


FIGURA 1-A

Fig. 5.1

FLUJO DE INFORMACION DETALLADO

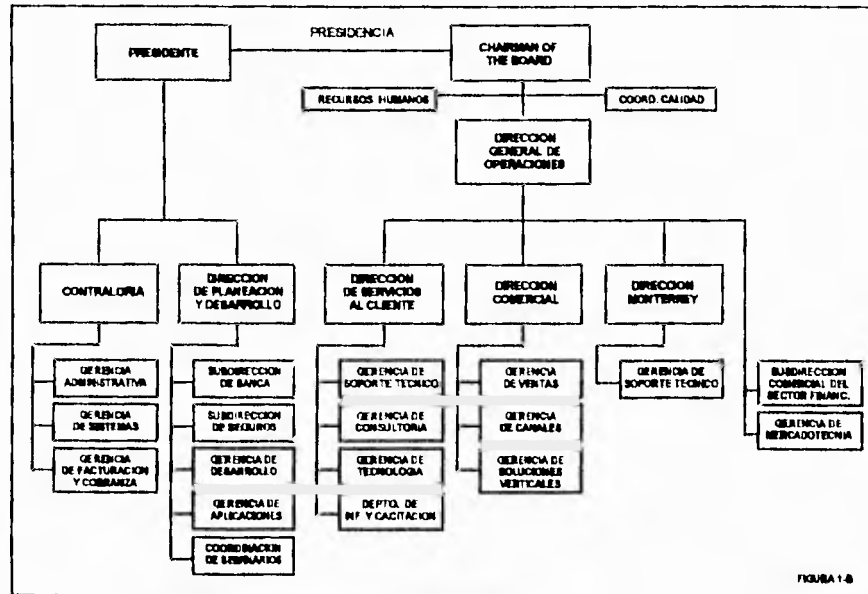


Fig. 5.2

FLUJO DE INFORMACION ACTUAL

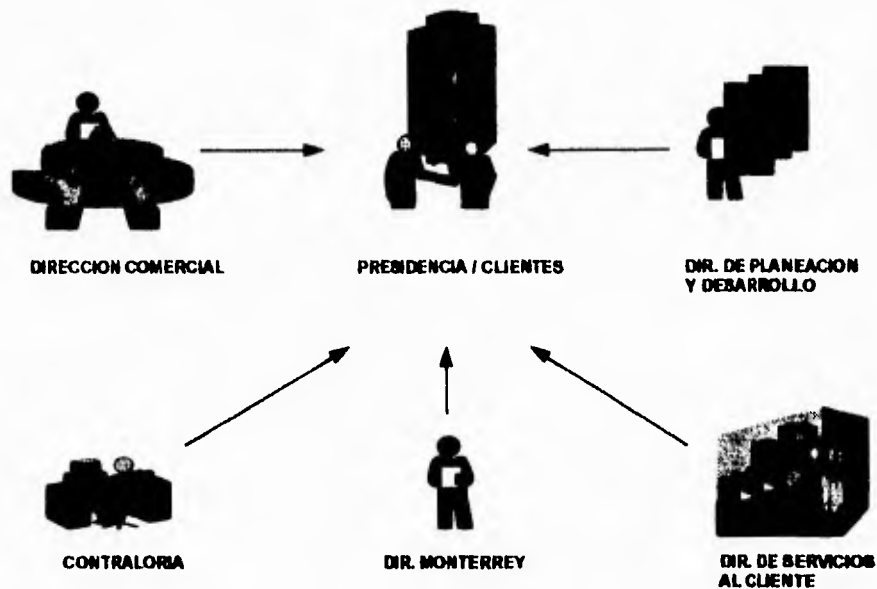


Fig. 5.3

AREAS INVOLUCRADAS CON LA ATENCION A CLIENTES

- **DIRECCION COMERCIAL** : Area encargada de las ventas directas con el cliente .
- **DIRECCION DE PLANEACION** : Su función es la administración de los proyectos desarrollados.
- **CONTRALORIA** : Area administrativa de la empresa, control financiero, costos y presupuestos.
- **MONTERREY** : Sucursal de Software A.G. en la ciudad de Monterrey, encargada de contactar clientes.
- **DIRECCION DE SERVICIOS AL CLIENTE** : A esta área le corresponde proporcionar soporte técnico, seminarios y cursos.
- **PRESIDENCIA** : Representante legal de la empresa, presente en la aprobación de un contrato con un cliente.

5.3 NECESIDADES DE INFORMACION

La empresa como prestadora de servicio, se preocupa por mejorar la calidad del mismo con sus clientes y proveedores para ofrecer servicios de verdadera calidad y profesionalismo. Pero se han detectado algunas necesidades de información que deben de solucionarse para cumplir sus objetivos con calidad y evitando demoras.

El tipo de información más importante sobre la cual Presidencia desea estar al tanto es:

- Información referente a las visitas que el área comercial ha realizado con los clientes.
 - * Qué cliente se visitó ?
 - * Quién estuvo a cargo de la visita ?
 - * Cuando se llevó a cabo ?
 - * Qué puntos se trataron ?

- Información sobre la situación de los proyectos tanto de desarrollo como de implantación ó adecuación de paquetes con los clientes.
 - * Qué proyecto ?
 - * Actividades
 - * Avances (planeados y reales)
 - * Desviaciones
 - * Causas de dichas desviaciones
 - * Participantes del proyecto
 - * Responsables

- Información sobre los problemas de tipo técnico que se han encontrado en las instalaciones de los clientes con los productos.
 - * Indicando en número de reporte de soporte técnico
 - * Fecha de recepción del reporte
 - * Fecha en que se atendió
 - * Quién atendió el reporte ?
 - * Descripción del problema
 - * Acciones para solucionarlo
 - * Tiempo de respuesta que se tomó para resolver el problema

- Información referente a la situación de facturación y cobranza con los clientes
 - * Indicando los montos facturados
 - * Pendientes de pago
 - * Montos a facturar
 - * Situación de cada factura
 - * Explicación sobre los retrasos en pagos por parte del cliente.

- La empresa debido a su organización, no cuenta con una información actualizada y organizada.
- Cuando los presidentes y directivos de la compañía requieren conocer del estado de su cartera de clientes ó el estado de cuenta de los mismos, no se tiene la información de inmediato.
- Cuando un cliente empieza a quejarse del avance de un proyecto ó de un producto, la presidencia debe organizar una junta para saber el estado del mismo con los directivos de las áreas, debido a que se debe de coordinar a los responsables del proyecto y reunir los antecedentes del mismo para tener los elementos suficientes en la reunión con el cliente, este proceso puede tomar varios días.

5.4 ANALISIS

OBJETIVO DEL ANALISIS

Automatizar el Flujo de información de todas las áreas de la empresa, con un acceso inmediato, actualizado y de caracter relevante.

SOLUCION

Para tal efecto se decidió establecer un método que permita tener dicha información al día y que se encuentre disponible en todo momento, para lo cual se decidió la realización de un sistema de información ejecutiva que permita fundamentalmente:

- Que la presidencia tenga un acceso inmediato a la información sobre la situación que cada área mantiene con sus clientes, que permita una adecuada toma de desiciones a alto nivel que se refleje en una mejora de la calidad de los servicios prestados.
- Tener la información centralizada de todas las áreas asociadas en la relación empresa-cliente.
- Tener un adecuado seguimiento de los proyectos realizados y poder aplicar medidas correctivas para el caso en que se desvien de los objetivos trazados.
- Rápido acceso a la información financiera existente con cada uno de los clientes.
- Tener una historia de los problemas que ocurran con cada uno de los clientes con los servicios y productos ofrecidos.

El sistema por el tipo de información que contiene recibe el nombre de **Sistema de Administración Ejecutiva de Proyectos (AEP)**. Está diseñado para que pueda ser la base de un sistema informático más completo.

AREAS INVOLUCRADAS

Para que el sistema funcione, es necesaria la colaboración constante y comprometida de las diferentes áreas (siguiendo el organigrama), para dar seguimiento y mantenimiento a la información de la cual es responsable cada una de ellas, siendo de vital importancia que se definan las responsabilidades, tanto de cual es la información que debe mantener cada área, así como de quién ó quienes actualizarán la información en el sistema.

De igual forma debe discriminarse toda aquella información que no sea de carácter relevante, para efecto de matener el volumen de información a consultar al mínimo indispensable.

FLUJO DE INFORMACION CON EL AEP

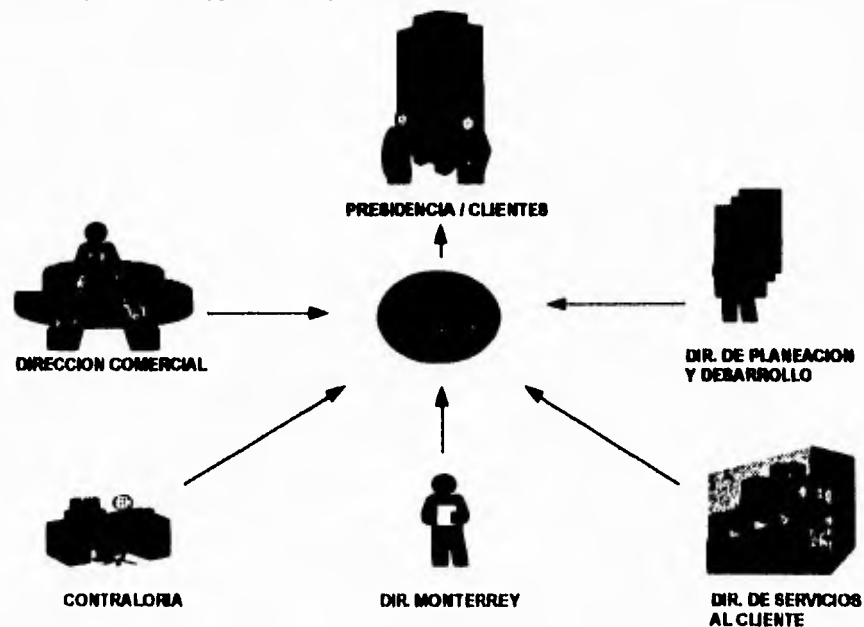


Fig. 5.4

Con el sistema implantado se mejorará los servicios con el cliente ya que la presidencia puede tomar decisiones de inmediato si se tiene problemas con el cliente-empresa.

La relación que existe entre los módulos, permite tener la información actualizada y disponible, cualquier modificación, alta ó baja de la información se registrará automáticamente, actualizando los datos en los registros correspondientes.

Con la información se obtuvieron las entidades principales y sus atributos, utilizando las herramientas de la empresa, en este caso CASE NATURAL ARCHITECT, el cual contiene un editor de Diagrama de Entidad Relación (ERD), editor de Diagrama de Flujo de Datos (DFD), editor de Mapas, editor de Procesos.

El editor de ERD es una herramienta para un diseño gráfico con el cual se crean los atributos y diccionario de datos.

Las entidades representativas del sistema son :

- PROYECTOS
- CLIENTES
- FACTURACION
- PROBLEMAS

De los cuales al analizarlos se determinarán los siguientes:

- | | | |
|------------------------------|---------------------------------|------------------------------|
| - Problemas-cliente | - Clasificación de Problemas | - Tipos de cliente |
| - Contactos | - Empleados | - Visitas del area comercial |
| - Reporte de Soporte Técnico | - Situación de los reportes | - Situación de los proyectos |
| - Actividades por Proyecto | - Equipo de Cómputo del Cliente | - Areas |
| - Situación de las Facturas | | |

Continuamente se fueron encontrando más entidades dando como resultado el ERD diseñado con Architect (ERD1).

Después de que se tiene el ERD se procede a explotar las entidades, es decir, describir los atributos que caracterizan a la entidad. (El tipo de diagramación que se utiliza se describió en el capítulo 2)

Natural Architect, cuando explota las entidades para obtener sus atributos, crea el diccionario de datos, descripción de los campos, longitud, tipo de supresión. Datos que serán utilizados para el diseño físico de la base de datos.

Con el sistema implantado se mejorará los servicios con el cliente ya que la presidencia puede tomar decisiones de inmediato si se tiene problemas con el cliente-empresa.

La relación que existe entre los módulos, permite tener la información actualizada y disponible, cualquier modificación, alta ó baja de la información se registrará automáticamente, actualizando los datos en los registros correspondientes.

Con la información se obtuvieron las entidades principales y sus atributos, utilizando las herramientas de la empresa, en este caso CASE NATURAL ARCHITECT, el cual contiene un editor de Diagrama de Entidad Relación (ERD), editor de Diagrama de Flujo de Datos (DFD), editor de Mapas, editor de Procesos.

El editor de ERD es una herramienta para un diseño gráfico con el cual se crean los atributos y diccionario de datos.

Las entidades representativas del sistema son :

- PROYECTOS
- CLIENTES
- FACTURACION
- PROBLEMAS

De los cuales al analizarlos se determinarán los siguientes:

- | | | |
|------------------------------|---------------------------------|------------------------------|
| - Problemas-cliente | - Clasificación de Problemas | - Tipos de cliente |
| - Contactos | - Empleados | - Visitas del area comercial |
| - Reporte de Soporte Técnico | - Situación de los reportes | - Situación de los proyectos |
| - Actividades por Proyecto | - Equipo de Cómputo del Cliente | - Areas |
| - Situación de las Facturas | | |

Continuamente se fueron encontrando más entidades dando como resultado el ERD diseñado con Architect (ERD1).

Después de que se tiene el ERD se procede a explotar las entidades, es decir, describir los atributos que caracterizan a la entidad. (El tipo de diagramación que se utiliza se describió en el capítulo 2)

Natural Architect, cuando explota las entidades para obtener sus atributos, crea el diccionario de datos, descripción de los campos, longitud, tipo de supresión. Datos que serán utilizados para el diseño físico de la base de datos

Natural Architect tiene un editor de Diagrama de Flujo, Mapas (pantallas), y Diagramas de Proceso o Programas.

Toda la información que se obtiene con natural architect se presenta en las páginas siguientes así como algunas pantallas de Natural Architect.

5.5 DOCUMENTACION DEL ANALISIS

La documentación que se incluye en el análisis de un sistema se especifica en la figura siguiente, la cual establece la forma en que se realizó el modelo que nos define las requerimientos de información de nuestro sistema.

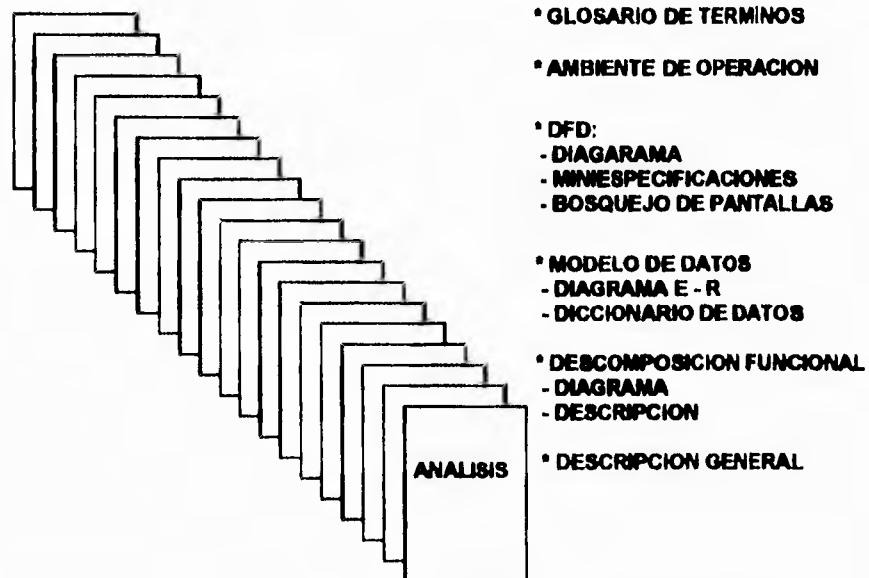


Fig. 5.5

A continuación se presenta la información del análisis de los requerimientos del sistema que nos proporciona la herramienta CASE, la documentación del sistema se genera a la vez que se realiza el análisis.

5.6 DOCUMENTACION DEL DISEÑO

La documentación que se incluye en el diseño de un sistema se especifica en la figura siguiente, la cual establece la forma en que se mapeó el modelo conceptual a un diseño de base de datos funcional.

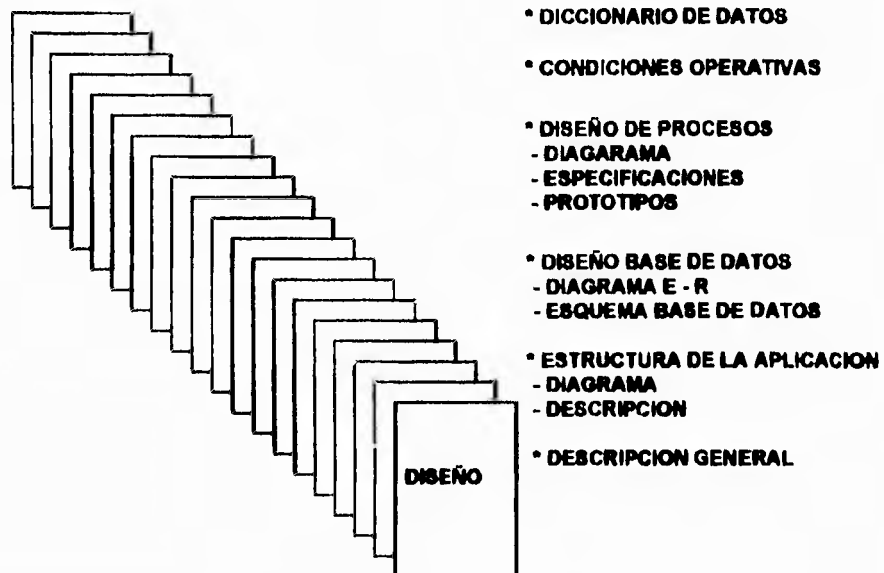


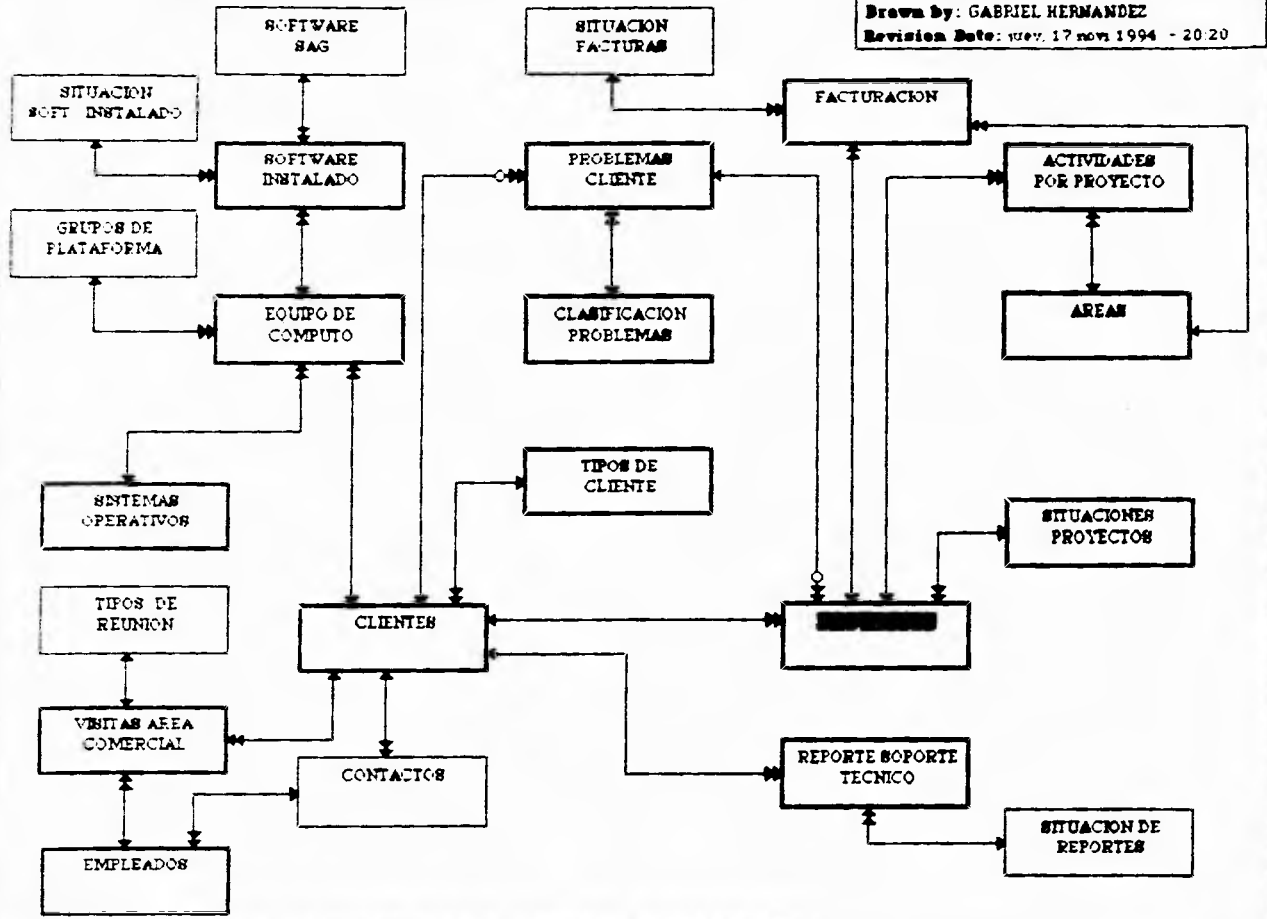
Fig. 5.6

A continuación se presenta la información del diseño del sistema que nos proporciona la herramienta CASE, la documentación del sistema es generada al mismo tiempo que es realizado el diseño, lo cual nos permite realizar las modificaciones en el diseño que sean requeridas sin que por ello la documentación deba ser nuevamente alterada.

DIAGRAMA ENTIDAD RELACION

SISTEMA DE ADMINISTRACION EJECUTIVA DE PROYECTOS

Modelo: EAP-ERD-GR
 Drawn by: GABRIEL HERNANDEZ
 Revision Date: jueves, 17 nov 1994 - 20:20



DOCUMENTACION DEL MODELO ENTIDAD-RELACION

ERD Report: EAP-ERD-GR vier, 2 dici 1994 18:32

Architect Data Dictionary: SIS-EJEC-ADMN-PROYECTO

Host Type: ADABAS

Entities: (22)

1. ACTIVIDADES POR PROYECTO

DESCRIP
DESV
DURACION-PLAN
DURACION-REAL
FEC-INI-PLAN
FEC-INI-REAL
FEC-TER-PLAN
FEC-TER-REAL
PORC-AVAN-PLAN
PORC-AVAN-REAL
REFERENCIA
CVE-ACT
CVE-CLIENTE
CVE-PROY

Related Objects:

Entity: PROYECTOS
Relationship: ACTIVIDADES POR PROYECTO#PROYECTOS
Cardinality: (one)

Entity: AREAS
Relationship: ACTIVIDADES POR PROYECTO#AREAS
Cardinality: (one)

Get Info:
ACTIVIDADES POR PROYECTO

REPORTE DE LAS ACTIVIDADES REALIZADAS EN LA CUENTA

2. AREAS

CVE-AREA
NOMBRE
CONTABLE

Related Objects:

Entity: FACTURACION
Relationship: AREAS#FACTURACION
Cardinality: (many)

Entity: ACTIVIDADES POR PROYECTO
Relationship: ACTIVIDADES POR PROYECTO#AREAS
Cardinality: (many)

Get Info:

AREAS

CATALOGO DE LAS AREAS QUE SE INVOLUCRAN EN UNA CUENTA.

**-DIR.SERV. CLIENTE
-GCIA.SOP.TEC.
-GCIA.SOP.TEC.COMERCIAL
-JEFATURA DE INFORMACION**

3. CLASIFICACION PROBLEMAS

**FEC-SITFIN
MONEDA
EJEC-CTA**

Related Objects:

Entity: PROBLEMAS CLIENTE
Relationship: PROBLEMAS PROYECTO#CLASIFICACION

PROBLEMAS

Cardinality: (many)

Get Info:

CLASIFICACION DE PROBLEMAS

CATALOGO DE PROBLEMAS

**- FALTA DE COMUNICACION
- PROBLEMA DEL SISTEMA OPERATIVO
- VERSION DE LOS PRODUCTOS**

4. CLIENTES

**COMENT-TECNO
COMUNICACIONES
DIREC-NOMBRE
DIREC-PUESTO
DIREC-TELEFONO
DIRECCION
GRUPO
IMP-ESTRAT**

IMP-SECTOR
RAZON-SOCIAL
RFC
TELEFONOS
TIPO-CLIENTE

Related Objects:

Entity: PROYECTOS
Relationship: CLIENTES#PROYECTOS
Cardinality: (many)

Entity: EQUIPO DE COMPUTO
Relationship: EQUIPO DE COMPUTO#CLIENTES
Cardinality: (many)

Entity: REPORTE SOPORTE TECNICO
Relationship: CLIENTES#REPORTE SOPORTE TECNICO
Cardinality: (many)

Entity: CONTACTOS
Relationship: CONTACTOS#CLIENTES
Cardinality: (many)

Entity: PROBLEMAS CLIENTE
Relationship: CLIENTES#PROBLEMAS
Cardinality: (zero or many)

Entity: TIPOS DE CLIENTE
Relationship: TIPOS DE CLIENTE#CLIENTES
Cardinality: (one)

Entity: VISITAS AREA COMERCIAL
Relationship: VISITAS AREA COMERCIAL#CLIENTES
Cardinality: (many)

Get Info:
CLIENTES

CATALOGO DE LOS CLIENTES DE SAG

5. CONTACTOS

Related Objects:

Entity: EMPLEADOS
Relationship: EMPLEADOS#CONTACTOS
Cardinality: (many)

Entity: CLIENTES
Relationship: CONTACTOS#CLIENTES
Cardinality: (one)

Get Info:
CONTACTOS

EL CONSULTOR QUE CONTACTO AL CLIENTE, PUEDE SER UN
CONSULTOR O CLIENTE

6. EMPLEADOS
CVE-EMP
CVE-AREA
PUESTO
NOMBRE-EMP

Related Objects:

Entity: CONTACTOS
Relationship: EMPLEADOS#CONTACTOS
Cardinality: (one)

Entity: VISITAS AREA COMERCIAL
Relationship: VISITAS AREA COMERCIAL#EMPLEADOS
Cardinality: (many)

Get Info:
EMPLEADOS

EMPLEADOS QUE INVOLUCRARAN EN EL PROYECTO/SERVICIO

7. EQUIPO DE COMPUTO
ACCESORIOS
GRP-PLAT
NUM-EQUIPO
UBICACION
SOP-CVE-SISOP
DESCRIP
CVE-CLIENTE

Related Objects:

Entity: CLIENTES
Relationship: EQUIPO DE COMPUTO#CLIENTES
Cardinality: (one)

Entity: SISTEMAS OPERATIVOS
Relationship: SISTEMAS OPERATIVOS#EQUIPO DE COMPUTO
Cardinality: (one)

Entity: SOFTWARE INSTALADO
Relationship: EQUIPO DE COMPUTO#SOFTWARE INSTALADO
Cardinality: (many)

Entity: GRUPOS DE PLATAFORMA
Relationship: GRUPOS DE PLATAFORMA#EQUIPO DE COMPUTO
Cardinality: (one)

Get Info:
EQUIPO DE COMPUTO

EQUIPO DE COMPUTO DEL CLIENTE

8. FACTURACION

CVE-AREA
ACREDITADO
COBRADO
CONCEPTO
FEC-EMISION
IMPORTE
NUMERO
STATUS
CVE-PROY
CVE-CLIENTE
FOLIO

Related Objects:

Entity: PROYECTOS
Relationship: FACTURACION POR PROYECTO#PROYECTOS
Cardinality: (one)

Entity: AREAS
Relationship: AREAS#FACTURACION
Cardinality: (one)

Entity: SITUACION FACTURAS
Relationship: SITUACION FACTURAS#FACTURACION
Cardinality: (one)

Get Info:
FACTURACION

REPORTE DE LA FACTURACION DE UNA CUENTA

9. GRUPOS DE PLATAFORMA

Related Objects:

Entity: EQUIPO DE COMPUTO
Relationship: GRUPOS DE PLATAFORMA#EQUIPO DE COMPUTO
Cardinality: (many)

Get Info:
GRUPOS DE PLATAFORMA

SE REFIERE A LA PLATAFORMA QUE TIENE EL CLIENTE

- UNIX
- MAIN FRAME
- PC

10. PROBLEMAS CLIENTE

TIPO-PROBLEMA
CVE-PROY
STATUS-SIN
GRAVEDAD
CVE-CLIENTE
NUM-PRB
DESC-PROBLEMA
CAUSA
SOLUCION

Related Objects:

Entity: CLASIFICACION PROBLEMAS
Relationship: PROBLEMAS PROYECTO#CLASIFICACION PROBLEMAS
Cardinality: (one)

Entity: CLIENTES
Relationship: CLIENTES#PROBLEMAS
Cardinality: (one)

Entity: PROYECTOS
Relationship: PROBLEMAS #PROYECTOS
Cardinality: (zero or many)

Get Info:

PROBLEMAS CLIENTE

REPORTE DE LOS PROBLEMAS QUE SE PUEDEN TENER EN UNA CUENTA

11. PROYECTOS

CVE-AREA
COSTOS-CPTO
COSTOS-IMP
CVE-PROY
FACT-CRIT
FEC-INI
FEC-TER
FEC-ULT-ACT
IMP-FACTURADO
IMP-PAGADO
IMP-TOTAL
IMPORTANCIA

NUM-CONTRATO
PORC-AVANCE
RETRASO-PRY
SUP-CRIT
REC-HUM-PRY
NOMBRE-PROY
DESV
STATUS-SIN
CVE-CLIENTE
MONEDA
EJEC-CTA
FEC-SITFIN

Related Objects:

Entity: CLIENTES
Relationship: CLIENTES#PROYECTOS
Cardinality: (one)

Entity: FACTURACION
Relationship: FACTURACION POR PROYECTO#PROYECTOS
Cardinality: (many)

Entity: ACTIVIDADES POR PROYECTO
Relationship: ACTIVIDADES POR PROYECTO#PROYECTOS
Cardinality: (many)

Entity: PROBLEMAS CLIENTE
Relationship: PROBLEMAS #PROYECTOS
Cardinality: (one)

Entity: SITUACIONES PROYECTOS
Relationship: SITUACIONES PROYECTOS#PROYECTOS
Cardinality: (one)

Get Info:
PROYECTOS

REPORTE DE UN PROYECTO

- AVANCE
- PROBLEMAS
- EQUIPO DE TRABAJO

12. REPORTE SOPORTE TECNICO

FEC-REPORTE
FEC-ATENCION
FEC-SOLUCION

NUM-REPORTE
EMP-ATENDIO
EMP-RECIBIO
DESC-PROBLEMA
STATUS-RST
CVE-CLIENTE
ACCION

Related Objects:

Entity: CLIENTES
Relationship: CLIENTES#REPORTE SOPORTE TECNICO
Cardinality: (one)

Entity: SITUACION DE REPORTES
Relationship: REPORTE SOPORTE TECNICO#SITUACION DE

REPORTES

Cardinality: (one)

Get Info:

SE CONTEMPLA EL MATENIMIENTO DE LOS REPORTES DE SOPORTE TECNICO QUE FUERON REALIZADOS A DETERMINADO CLIENTE (CUENTA).

13. SISTEMAS OPERATIVOS

SOP-CVE-SISOP
SOP-DESCRIP

Related Objects:

Entity: EQUIPO DE COMPUTO
Relationship: SISTEMAS OPERATIVOS#EQUIPO DE COMPUTO
Cardinality: (many)

Get Info

SISTEMAS OPERATIVOS

CATALOGO DE SISTEMAS OPERATIVOS

-MS-DOS

-UNIX

-VMS

ETC

14. SITUACION FACTURAS

Related Objects

Entity: FACTURACION
Relationship: SITUACION FACTURAS#FACTURACION
Cardinality: (many)

Get Info

SITUACION FACTURAS
EL ESTADO DE LAS FACTURAS
- CANCELADA
- PEND. COBRO
- PAGADA

15. SITUACION SOFT. INSTALADO

Related Objects:

Entity: SOFTWARE INSTALADO
Relationship: SITUACION SOFT. INSTALADO#SOFTWARE INSTALADO
Cardinality: (many)

Get Info:

SITUACION SOFT. INSTALADO

SE REFIERE AL ESTADO EN QUE SE ENCUENTRA EL SOFTWARE:

- BUENO
- MALO
- BETA

16. SITUACION DE REPORTE

Related Objects:

Entity: REPORTE SOPORTE TECNICO
Relationship: REPORTE SOPORTE TECNICO#SITUACION DE

REPORTE

Cardinality: (many)

Get Info:

CATALOGO SOBRE LA SITUACION DE PROYECTOS

P: PENDIENTE
A: ATENDIDO
S: SUSPENDIDO

17 SITUACIONES PROYECTOS

STATUS-RST

Related Objects:

Entity: PROYECTOS
Relationship: SITUACIONES PROYECTOS#PROYECTOS
Cardinality: (many)

Get Info:

SITUACION PROYECTOS

CATALOGO DEL ESTADO DE UN PROYECTO

- ACTIVO
- SUSPENDIDO
- TERMINADO

18. SOFTWARE INSTALADO

NUM-EQUIPO
CVE-SOFT
STATUS-SIN
VERSION
CVE-CLIENTE

Related Objects:

Entity: SOFTWARE SAG
Relationship: SOFTWARE SAG#SOFTWARE INSTALADO
Cardinality: (one)
Entity: EQUIPO DE COMPUTO
Relationship: EQUIPO DE COMPUTO#SOFTWARE

INSTALADO

Cardinality: (one)

Entity: SITUACION SOFT. INSTALADO
Relationship: SITUACION SOFT. INSTALADO#SOFTWARE INSTALADO
Cardinality: (one)

Get Info:

SOFTWARE INSTALADO

**SE REFIERE AL TIPO DE SOFTWARE QUE TIENE EL CLIENTE
YA SEA PRODUCTOS DE SAG U OTROS.**

19. SOFTWARE SAG

Related Objects:

Entity: SOFTWARE INSTALADO
Relationship: SOFTWARE SAG#SOFTWARE INSTALADO
Cardinality: (many)

Get Info:

SOFTWARE SAG

ES UN CATALOGO DE LOS PRODUCTOS DE SAG COMO:

- ADABAS
- NATURAL
- NATURAL CONSTRUCT
- PREDICT
- NATURAL ARCHITECT
- NATURAL FOR WINDOWS

20. TIPOS DE REUNION

Related Objects:

Entity: VISITAS AREA COMERCIAL
Relationship: Blank+VISITAS AREA COMERCIAL
Cardinality: (one)

Get Info:
TIPOS DE REUNION

CATALOGO DEL TIPO DE REUNION:
- REUNION INFORMAL
- REUNION DE AVANCE DEL PROYECTO
- REUNION DE DIRECTORES

ETC

21. TIPOS DE CLIENTE

Related Objects:

Entity: CLIENTES
Relationship: TIPOS DE CLIENTE#CLIENTES
Cardinality: (many)

Get Info:
TIPOS DE CLIENTE

CATALOGO DE LA CLASE DE CLIENTE QUE TENEMOS

- PROSPECTO
- CLIENTE
- SUSPECTO

22 VISITAS AREA COMERCIAL

CONSULTOR
FEC-VISITA
PUNTOS-TRAT
TIPO-REUNION
CVE-PROY
•Deleted•
CVE-CLIENTE

Related Objects:

Entity: EMPLEADOS
Relationship: VISITAS AREA COMERCIAL#EMPLEADOS
Cardinality: (one)
Entity: TIPOS DE REUNION
Relationship: Blank+VISITAS AREA COMERCIAL
Cardinality: (one)

Entity: CLIENTES
Relationship: VISITAS AREA COMERCIAL#CLIENTES

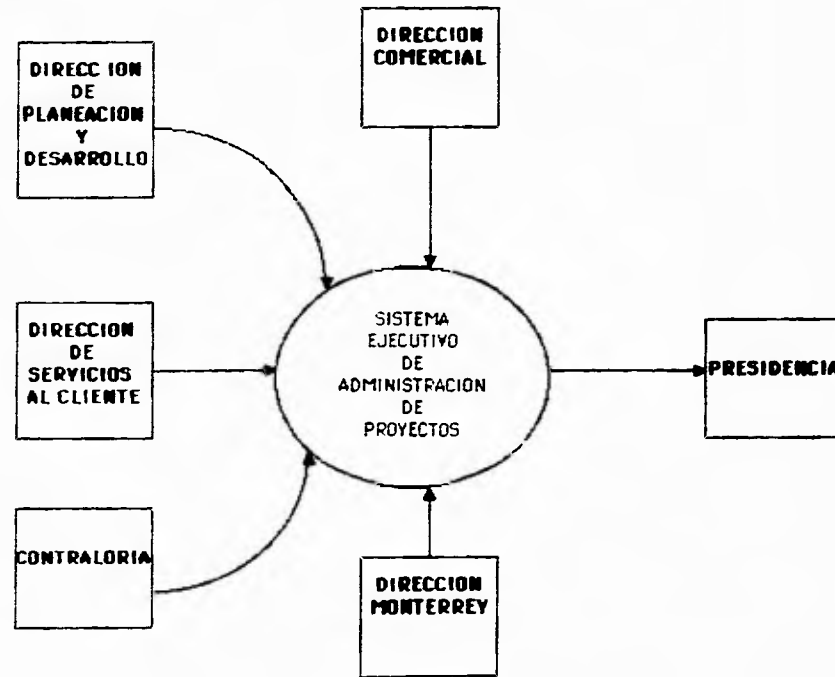
Cardinality: (one)

**Get Info:
VISITAS AREA COMERCIAL**

VISITAS QUE REALIZO EL AREA COMERCIAL AL CLIENTE

SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

DIAGRAMA DE CONTEXTO



Modelo: Context
Drawn By:
Revision Date: vier. 2 dici 1994 - 16:53

DOCUMENTACION DEL DIAGRAMA DE CONTEXTO

DFD Report: AEP-CONT

Report Generated: viernes, 2 diciembre 1994

Dictionary:

Schema Type: ADABAS

Process:

Context

Local Dataflows

[d-1] d-1

From: [d] DIRECCION COMERCIAL

To: [1] SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

DIRECCION COMERCIAL A AEP

- DATOS DE CLIENTE
- DATOS DE CONSULTORES
- FACTURAS
- DATOS DEL PROYECTO
- DATOS DEL CONTRATO

[a-1] a-1

From: [a] DIRECCION DE PLANEACION Y DESARROLLO

To: [1] SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

DIRECCION DE PLANEACION A AEP

- DATOS DE CONSULTOR
- DATOS DEL AVANCE DEL PROYECTO
- SITUACION DEL PROYECTO
- FACTURAS

[b-1] b-1

From: [b] DIRECCION DE SERVICIOS AL CLIENTE

To: [1] SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

DIRECCION DE SERV. AL CLIENTE A AEP

- REPORTE DE SOPORTE TECNICO DE UN PROYECTO
- REPORTE DE CONSULTORIA
- AMBIENTE TECNOLOGICO
- CONSULTOR QUE ATENDIO EL PROBLEMA
- CONSULTOR QUE RESOLVIO EL PROBLEMA

[c-1] c-1

From: [c] CONTRALORIA
To: [1] SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

CONTRALORIA A AEP

- GENERACION DE FACTURAS

[f-1] f-1

From: [f] DIRECCION MONTERREY
To: [1] SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

DIRECCION DE MONTERREY A AEP

- DATOS DEL CLIENTE
- DATOS DEL CONTACTO (CONSULTOR)
- DATOS DEL PROYECTO

[1-i] 1-i

From: [1] SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS
To: [i] PRESIDENCIA

AEP A PRESIDENCIA

- INFORMACION DE CARACTER IMPORTANTE PARA LA TOMA DE
DESCIONES.

- ELEMENTOS PARA ENFRENTAR EN UNA REUNION A UN CLIENTE.

[1]: SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS

SISTEMA EJECUTIVO DE ADMINSTRACION DE PROYECTOS
(AEP)

SISTEMA MEDIANTE EL CUAL EL PERSONAL PODRA CONOCER DE
FORMA
INTEGRAL EL TRABAJO QUE LAS DIFERENTES AREAS REALIZAN SOBRE
UNA CUENTA
DETERMINADA. AUTOMATIZARA EL FLUJO DE INFORMACION.

One Undefined Process

External Entities

[a]: DIRECCION DE PLANEACION Y DESARROLLO

DIRECCION DE PLANEACION Y DESARROLLO

DIRECCION ENCARGADA DEL SEGUIMIENTO DE UN PROYECTO,
DESDE LA
PLANEACION HASTA LA IMPLANTACION.

[b]: DIRECCION DE SERVICIOS AL CLIENTE

DIRECCION DE SERVICIOS AL CLIENTE

DIRECCION ENCARGADA DE BRINDAR:

- SOPORTE TECNICO
- CONSULTORIA
- TECNOLOGIA
- SEMINARIOS/CAPACITACION

[c]: CONTRALORIA

CONTRALORIA

ENCARGADA DE LA ADMINISTRACION DE LA EMPRESA, LA
FACTURACION Y
COBRANZA

[d]: DIRECCION COMERCIAL

DIRECCION COMERCIAL

ENCARGADA DE CONTACTAR AL CLIENTE, REALIZAR UNA VENTA,
DAR
SOLUCIONES VERTICALES (CANALIZARLAS A CONSULTORIA)

[f]: DIRECCION MONTERREY

DIRECCION MONTERREY

SUCURSAL DE SOFTWARE AG EN LA CIUDAD DE MONTERREY

[i]: PRESIDENCIA

PRESIDENCIA

ENCARGADA DE REVISAR EL CONTRATO FINAL, TOMA DE
SOLUCIONES
INMEDIATAS.

Total of 6 External Entities

DOCUMENTACION DEL DICCIONARIO DEDATOS

Dictionary: SIS-EJEC-ADMON-PROYECTO vier, 2 dici 1994 18:57
Schema Type: ADABAS
Dictionary Created: juev, 14 abri 1994 10:43
Dictionary Modified: vier, 2 dici 1994 18:47

Attributes: (90)

1. **ACCESORIOS** alphanum 20 Null
 Heading 1: ACCESORIOS
 ACCESORIOS QUE CONTIENE EL EQUIPO
2. **ACCION** alphanum 70 Null
3. **ACREDITADO** numeric 7.0U Null
 Heading 1: ACREDITADO
 IMPORTE ACREDITADO DE LA FACTURA
4. **CAUSA** alphanum 60 Null
 Heading 1: CAUSA
 CAUSA DEL PROBLEMA
5. **COBRADO** numeric 7.0U Null
 Heading 1: COBRADO
 IMPORTE DE LO QUE SE HA COBRADO DE LA FACTURA
6. **COMENT-TECNO** alphanum 70 Null
 Heading 1: COMENTARIOS TECNOLOGIA
 COMENTARIOS SOBRE SU PLATAFORMA TECNOLOGICA
7. **COMUNICACIONES** alphanum 70 Null
 Heading 1: COMUNICACIONES
 FACILIDADES DE COMUNICACIONES QUE TIENE CONTRATADAS EL
 CLIENTE
8. **CONCEPTO** alphanum 20 Null
 Heading 1: CONCEPTO
 CONCEPTO DE LA FACTURA
9. **CONSULTOR** numeric 3.0U Null
 Heading 1: CONSULTOR
 CLAVE DEL EMPLEADO QUE FUNGE COMO CONSULTOR
10. **CONTABLE** alphanum 9 Null
 Heading 1: CTA. CONTABLE
 Heading 2: ASIGNADA

11. **COSTOS-CPTO** alphanumeric 20 Null
Heading 1: **CONCEPTO**
CONCEPTO DE COSTOS GENERADOS POR EL PROYECTO PARA SAG
12. **COSTOS-IMP** numeric 7.0U Null
Heading 1: **IMPORTE**
Edit Mask: \$999,999
IMPORTE DE LOS COSTOS GENERADOS POR EL PROYECTO
13. **CVE-ACT** numeric 3.0U Null
14. **CVE-AREA** alphanumeric 4 Fixed
Heading 1: **CLAVE DEL AREA**
CLAVE DEL AREA
15. **CVE-CLASIF** alphanumeric 2 Null
CLAVE DE CLASIFICACION DE PROBLEMA
16. **CVE-CLIENTE** alphanumeric 10 Null
CLAVE DEL CLIENTE
17. **CVE-EMP** numeric 3.0U Null
Heading 1: **CLAVE DE EMPLEADO**
CLAVE DEL EMPLEADO
18. **CVE-PROY** alphanumeric 6 Fixed
Heading 1: **CLAVE PROYECTO**
CLAVE DEL PROYECTO
19. **CVE-SOFT** alphanumeric 3 Fixed
Heading 1: **CLAVE DE SOFTWARE**
CLAVE DE SOFTWARE
20. **DESC-PROBLEMA** alphanumeric 50 Null
Heading 1: **PROBLEMA**
DESCRIPCION DE LA FALLA QUE SE REPORTA
21. **DESCRIP** alphanumeric 30 Null
Heading 1: **DESCRIPCION**
DESCRIPCION DE LA ACTIVIDAD
22. **DESV** numeric 3.0U Null
23. **DIREC-NOMBRE** alphanumeric 25 Null
Heading 1: **NOMBRE**
NOMBRE DEL DIRECTIVO

24. **DIREC-PUESTO** alphanumeric 30 Null
Heading 1: PUESTO
PUESTO DEL DIRECTIVO
25. **DIREC-TELEFONO** alphanumeric 18 Null
Heading 1: TELEFONO
TELEFONO DEL DIRECTIVO
26. **DIRECCION** alphanumeric 60 Null
Heading 1: DIRECCION
DIRECCION DEL CLIENTE
27. **DURACION-PLAN** numeric 3.0U Null
Heading 1: DURACION (DIAS)
Heading 2: PLAN
DURACION DE LA ACTIVIDAD EN DIAS (PLANEADO)
28. **DURACION-REAL** numeric 3.0U Null
Heading 1: DURACION (DIAS)
Heading 2: REAL
DURACION DE LA ACTIVIDAD EN DIAS (REAL)
29. **EJEC-CTA** numeric 3.0U Null
EJECUCION DE CUENTA
30. **EMP-ATENDIO** numeric 3.0U Null
Heading 1: EMPLEADO ATENDIO
CLAVE DEL EMPLEADO QUE ATENDIO EL REPORTE
31. **EMP-RECIBIO** numeric 3.0U Null
Heading 1: EMPLEADO RECIBIO
CLAVE DEL EMPLEADO QUE RECIBIO EL REPORTE
32. **ES-RESPONSABLE** binary 1 Null
Heading 1: RESPONSABLE (SI/NO)
INDICA SI EL PARTICIPANTE ES RESPONSABLE DEL PROYECTO
33. **FACT-CRIT** alphanumeric 70 Null
Heading 1: FACTORES CRITICOS DE EXITO
FACTORES CRITICOS QUE INTERVIENEN PARA LLEVAR A BUEN
TERMINO DEL PROYECTO
34. **FEC-ATENCION** date Null
Heading 1: FECHA ATENCION
FECHA DE ATENCION DEL REPORTE

35. **FEC-EMISION** **date** Null
Heading 1: **FECHA EMISION**
FECHA DE EMISION DE FACTURA
36. **FEC-INI** **date** Null
Heading 1: **FECHA DE INICIO**
FECHA DE INICIO DEL PROYECTO
37. **FEC-INI-PLAN** **date** Null
Heading 1: **FECHA DE INICIO**
Heading 2: **PLAN**
FECHA DE INICIO DE LA ACTIVIDAD (PLAN)
38. **FEC-INI-REAL** **date** Null
Heading 1: **FECHA DE INICIO**
Heading 2: **REAL**
FECHA DE INICIO DE LA ACTIVIDAD (REAL)
39. **FEC-REPORTE** **date** Null
Heading 1: **FECHA REPORTE**
FECHA DEL REPORTE
40. **FEC-SITFIN** **date** Null
FECHA DE LA SITUACION FINANCIERA
41. **FEC-SOLUCION** **date** Null
Heading 1: **FECHA SOLUCION**
FECHA EN QUE SE SOLUCIONO LA FALLA
42. **FEC-TER** **date** Null
Heading 1: **FECHA DE TERMINACION**
FECHA DE TERMINACION DEL PROYECTO
43. **FEC-TER-PLAN** **date** Null
Heading 1: **FECHA DE TERMINACION**
Heading 2: **PLAN**
FECHA DE TERMINACION DE LA ACTIVIDAD (PLAN)
44. **FEC-TER-REAL** **date** Null
Heading 1: **FECHA DE TERMINACION**
Heading 2: **REAL**
FECHA DE TERMINACION DE LA ACTIVIDAD (REAL)
45. **FEC-ULT-ACT** **date** Null
Heading 1: **ULTIMA ACTUALIZACION**
FECHA DE ULTIMA ACTUALIZACION SOBRE DATOS DE AVANCE O
SITUACION GENERAL DEL PROYECTO

46. **FEC-VISITA** date Null
Heading 1: **FECHA VISITA**
FECHA DE VISITA
47. **FOLIO**alphanumeric 6 Null
NUMERO DE FOLIO
48. **FUNCION** alphanumeric 20 Null
Heading 1: **FUNCION**
FUNCION DE LOS PARTICIPANTES
49. **GRAVEDAD** numeric 1.0U Null
Heading 1: **GRAVEDAD**
GRAVEDAD DEL PROBLEMA
50. **GRP-PLAT** alphanumeric 1 Null
Heading 1: **GRUPO**
GRUPO DE PLATAFORMAS CON LAS QUE CUENTA EL CLIENTE
1 - MAINFRAME
2 - VAX
3 - UNIX
4 - OS/2
5 - WINDOWS
51. **GRUPO** alphanumeric 20 Null
Heading 1: **GRUPO**
GRUPO DEL CLIENTE
52. **IMP-ESTRAT** numeric 2.0U Null
Heading 1: **NIVEL DE IMPORTANCIA ESTRATEGI**
53. **IMP-FACTURADO** numeric 7.0U Null
Heading 1: **FACTURADO**
Edit Mask: \$9,999,999
IMPORTE TOTAL FACTURADO POR EL PROYECTO
54. **IMP-PAGADO** numeric 7.0U Null
Heading 1: **PAGADO**
Edit Mask: \$9,999,999
IMPORTE TOTAL PAGADO DEL TOTAL FACTURADO POR EL PROYECTO
55. **IMP-SECTOR** alphanumeric 4 Null
Heading 1: **IMPORTANCIA EN EL SECTOR**
IMPORTANCIA DEL CLIENTE DENTRO DEL SECTOR

56. **IMP-TOTAL** numeric 7.0U Null
Heading 1: **COSTO TOTAL**
Edit Mask: \$9,999,999
COSTO TOTAL DEL PROYECTO
57. **IMPORTANCIA** alphanum 60 Null
Heading 1: **IMPORTANCIA DEL PROYECTO**
IMPORTANCIA DEL PROYECTO
58. **IMPORTE** numeric 7.0U Null
Heading 1: **IMPORTE**
Edit Mask: \$999,999
IMPORTE DE LA FACTURA (EN DOLARES)
59. **MONEDA** alphanum 1 Null
TIPO DE MONEDA PESO/DOLAR
60. **NOMBRE** alphanum 20 Null
Heading 1: **NOMBRE DEL AREA**
NOMBRE DEL AREA
61. **NOMBRE-EMP** alphanum 30 Null
NOMBRE DEL EMPLEADO
62. **NOMBRE-PROY** alphanum 30 Null
63. **NUM-CONTRATO** alphanum 10 Null
Heading 1: **NO. CONTRATO**
NUMERO DE CONTRATO INTERNO DE SAG CORRESPONDIENTE AL PROYECTO
64. **NUM-EQUIPO** numeric 2.0U Null
Heading 1: **NO. EQUIPO DE COMPUTO**
NUMERO DE EQUIPO
65. **NUM-PRB** numeric 2.0U Null
Heading 1: **NUMERO**
NUMERO DE PROBLEMA CUENTA
66. **NUM-REPORTE** alphanum 6 Fixed
Heading 1: **NO. REPORTE**
NUMERO DE REPORTE
67. **NUMERO** alphanum 6 Fixed
Heading 1: **NO. FACTURA**
NUMERO DE FACTURA

68. PORC-AVAN-PLAN numeric 3.0U Null
Heading 1: % AVANCE PLANEADO
PORCENTAJE DE AVANCE PLANEADO
69. PORC-AVAN-REAL numeric 3.0U Null
Heading 1: % AVANCE REAL
PORCENTAJE DE AVANCE REAL
70. PORC-AVANCE numeric 3.0U Null
Heading 1: % AVANCE
PORCENTAJE GENERAL DE AVANCE DEL PROYECTO
71. PUESTO alphanum 20 Null
Heading 1: PUESTO
PUESTO QUE OCUPA EL EMPLEADO
72. PUNTOS-TRAT alphanum 60 Null
Heading 1: PUNTOS TRATADOS
PUNTOS TRATADOS EN LA VISITA
73. RAZON-SOCIAL alphanum 40 Null
74. REC-HUM-PRY numeric 2.0U Null
Heading 1: RECURSOS HUMANOS
RECURSOS HUMANOS DEL PROYECTO
75. REFERENCIA alphanum 20 Null
Heading 1: REFERENCIA
REFERENCIA DE LA ACTIVIDAD
76. RETRASO-PRY numeric 3.0U Null
Heading 1: DIAS DE RETRASO
RETRASO GENERAL DEL PROYECTO EN DIAS
77. RFC alphanum 15 Null
Heading 1: RFC
REGISTRO FEDERAL DE CONTRIBUYENTES
78. SOLUCION alphanum 60 Null
Heading 1: SOLUCION
SOLUCION DEL PROBLEMA
79. SOP-CVE-SISOP alphanum 2 Fixed
Heading 1: CLAVE SISTEMA OPERATIVO
CLAVE DEL SISTEMA OPERATIVO

- 7
80. SOP-DESCRIP alphanum 30 Null
Heading 1: DESCRIPCION SISTEMA OPERATIVO
DESCRIPCION DEL SISTEMA OPERATIVO
81. STATUS alphanum 20 Null
Heading 1: SITUACION
SITUACION DE LA FACTURA
TIPO DE CAMPO CON VALORES RESTRINGIDOS
A - ACTIVA
P - PAGADA
C - CANCELADA
82. STATUS-RST alphanum 2 Null
STATUS SITUACION DEL PROYECTO
83. STATUS-SIN alphanum 1 Null
Heading 1: SITUACION
SITUACION EN LA QUE SE INSTALO EL SOFTWARE
1 - COMPRA
2 - RENTA
3 - ARRENDAMIENTO
4 - PRESTAMO
5 - PRUEBA
84. SUP-CRIT alphanum 70 Null
Heading 1: SUPUESTOS CRITICOS
SUPUESTOS CRITICOS PARA LLEVAR A BUEN TERMINO EL PROYECTO,
ES DECIR, TODOS AQUELLOS FACTORES QUE SE ESTAN SUPONIENDO COMO
VERDADEROS PARA EL LOGRO DE LOS OBJETIVOS DEL PROYECTO.
85. TELEFONOS alphanum 15 Null
Heading 1: TELEFONOS
TELEFONOS DEL CLIENTE
86. TIPO-CLIENTE alphanum 1 Null
Heading 1: TIPO DE CLIENTE
TIPO DE CLIENTE
C - CLIENTE
P - PROSPECTO
87. TIPO-PROBLEMA alphanum 2 Null
Heading 1: TIPO DE PROBLEMA
TIPO DE PROBLEMA CUENTA
- 5

88. TIPO-REUNION alphanum 2 Fixed
Heading 1: TIPO REUNION
CLASIFICACION DEL TIPO DE REUNION
01 - VISITA PROSPECTO
02 - CONSULTORIA
03 - AVANCE DEL PROYECTO
04 - VISITA ESPECIAL
89. UBICACION alphanum 15 Null
Heading 1: UBICACION
UBICACION DEL EQUIPO DE COMPUTO
90. VERSION alphanum 10 Null
Heading 1: VERSION
VERSION DEL SOFTWARE INSTALADO

Entities: (29)

1. ACTIVIDADES POR PROYECTO

DESCRIP alphanum 30 Null
DESV numeric 3.0U Null
DURACION-PLAN numeric 3.0U Null
DURACION-REAL numeric 3.0U Null
FEC-INI-PLAN date Null
FEC-INI-REAL date Fixed
FEC-TER-PLAN date Null
FEC-TER-REAL date Null
PORC-AVAN-PLAN numeric 3.0U Null
PORC-AVAN-REAL numeric 3.0U Null
REFERENCIA alphanum 20 Null
CVE-ACT numeric 3.0U Fixed
CLAVE DE LA ACTIVIDAD

CVE-CLIENTE alphanum 10 Null
CVE-PROY alphanum 6 Fixed

2. AMBIENTE TECNOLOGICO

3. AREAS

CVE-AREA alphanum 4 Fixed
Heading 1: CLAVE DEL AREA
CLAVE DEL AREA

NOMBRE alphanum 20 Null
CONTABLE alphanum 9 Null

4. **CLASIF. DE CLIENTE**
 CVE-CLASIF alphanum 2 Null
 Heading 1: CLAVE DE PROBLEMA
 Heading 1: DESCRIPCION
 DESCRIPCION DE LA CLASIFICACION DEL PROBLEMA
- GRAVEDAD numeric 1.0U Null
 Heading 1: SEVERIDAD
 GRAVEDAD DEL PROBLEMA
5. **CLASIFICACION PROBLEMAS**
 FEC-SITFIN date Null
 MONEDA alphanum 1 Null
 EJEC-CTA numeric 3.0U Null
6. **CLIENTES**
 COMENT-TECNO alphanum 70 Null
 COMUNICACIONES alphanum 70 Null
 •Deleted• •Deleted• Fixed
 CLAVE DEL CLIENTE
- DIREC-NOMBRE alphanum 25 Null
 DIREC-PUESTO alphanum 30 Null
 DIREC-TELEFONO alphanum 18 Null
 DIRECCION alphanum 60 Null
 GRUPO alphanum 20 Null
 IMP-ESTRAT numeric 2.0U Null
 IMP-SECTOR alphanum 4 Null
 RAZON-SOCIAL alphanum 40 Null
 Heading 1: RAZON SOCIAL
 RAZON SOCIAL DEL CLIENTE
- RFC alphanum 15 Null
 TELEFONOS alphanum 15 Null
 Heading 1: TELEFONOS
 TELEFONOS DEL CLIENTE
- TIPO-CLIENTE alphanum 1 Null
7. **EMPLEADOS**
 CVE-EMP numeric 3.0U Fixed
 •Deleted• •Deleted• Null
 CVE-AREA alphanum 4 Fixed
 PUESTO alphanum 20 Null
 NOMBRE-EMP alphanum 30 Null

8. EQUIPO DE COMPUTO

•Deleted• •Deleted• Fixed
ACCESORIOS alphanumeric 20 Null

Heading 1: ACCESORIOS
ACCESORIOS QUE CONTIENE EL EQUIPO

GRP-PLAT alphanumeric 1 Null
NUM-EQUIPO numeric 2.0U Null
UBICACION alphanumeric 15 Null
SOP-CVE-SISOP alphanumeric 2 Fixed
•Deleted• •Deleted• Null
DESCRIP alphanumeric 30 Null
CVE-CLIENTE alphanumeric 10 Null

9. FACTURACION

CVE-AREA alphanumeric 4 Fixed
ACREDITADO numeric 7.0U Fixed
COBRADO numeric 7.0U Null
CONCEPTO alphanumeric 20 Null
FEC-EMISION date Null
Heading 1: FECHA EMISION
FECHA DE EMISION DE FACTURA

IMPORTE numeric 7.0U Null
NUMERO alphanumeric 6 Null
STATUS alphanumeric 20 Null
CVE-PROY alphanumeric 6 Fixed
Heading 1: CLAVE PROYECTO
CLAVE DEL PROYECTO

•Deleted• •Deleted• Fixed
CVE-CLIENTE alphanumeric 10 Null
FOLIOalphanumeric 6 Null

10. GRUPOS DE PLATAFORMA

11. GRUPOS PLATAFORMA

12. PARTICIPANTES

CVE-EMP numeric 3.0U Fixed
ES-RESPONSABLE binary 1 Null
FUNCION alphanumeric 20 Null
CVE-PROY alphanumeric 6 Fixed

13. **PROBLEMAS CLENTE**
 CVE-PROY alphanumeric 6 Fixed
 •Deleted• •Deleted• Null
 •Deleted• •Deleted• Fixed
 CAUSA alphanumeric 60 Null
 •Deleted• •Deleted• Null
 NUM-PRB numeric 2.0U Null
 SOLUCION alphanumeric 60 Null
 TIPO-PROBLEMA alphanumeric 2 Null
 •Deleted• •Deleted• Null
14. **PROBLEMAS CLIENTE**
 TIPO-PROBLEMA alphanumeric 2 Null
 CVE-PROY alphanumeric 6 Fixed
 STATUS-SIN alphanumeric 1 Null
 GRAVEDAD numeric 1.0U Null
 CVE-CLIENTE alphanumeric 10 Null
 NUM-PRB numeric 2.0U Null
 DESC-PROBLEMA alphanumeric 50 Null
 CAUSA alphanumeric 60 Null
 SOLUCION alphanumeric 60 Null
15. **PROBLEMAS CUENTA**
 CAUSA alphanumeric 60 Null
 NUM-PRB numeric 2.0U Null
 SOLUCION alphanumeric 60 Null
 TIPO-PROBLEMA alphanumeric 2 Null
16. **PROYECTOS**
 CVE-AREA alphanumeric 4 Fixed
 COSTOS-CPTO alphanumeric 20 Null
 COSTOS-IMP numeric 7.0U Null
 CVE-PROY alphanumeric 6 Fixed
 FACT-CRIT alphanumeric 70 Null
 FEC-INI date Null
 FEC-TER date Null
 FEC-ULT-ACT date Null
 IMP-FACTURADO numeric 7.0U Null
 IMP-PAGADO numeric 7.0U Null
 IMP-TOTAL numeric 7.0U Null
 IMPORTANCIA alphanumeric 60 Null
 NUM-CONTRATO alphanumeric 10 Null
 PORC-AVANCE numeric 3.0U Null
 RETRASO-PRY numeric 3.0U Null
 SUP-CRIT alphanumeric 70 Null
 REC-HUM-PRY numeric 2.0U Null
 NOMBRE-PROY alphanumeric 30 Null
 DESV numeric 3.0U Null

STATUS-SIN alphanum 1 Null
CVE-CLIENTE alphanum 10 Null
MONEDA alphanum 1 Null
EJEC-CTA numeric 3.0U Null
FEC-SITFIN date Null

17. REPORTE SOPORTE TECNICO

FEC-REPORTE date Null
FEC-ATENCION date Null
FEC-SOLUCION date Null
NUM-REPORTE alphanum 6 Null
EMP-ATENDIO numeric 3.0U Null
EMP-RECIBIO numeric 3.0U Null
DESC-PROBLEMA alphanum 50 Null
•Deleted• •Deleted• Fixed

Heading 1: SITUACION
SITUACION DEL REPORTE
1 - PENDIENTE DE ATENDER
2 - ATENDIENDOSE
3 - SUSPENDIDO
4 - RESUELTO

•Deleted• •Deleted• Fixed
STATUS-RST alphanum 2 Null
CVE-CLIENTE alphanum 10 Null
ACCION alphanum 70 Null

18. SISTEMAS OPERATIVOS

SOP-CVE-SISOP alphanum 2 Fixed
SOP-DESCRIP alphanum 30 Null

19. SITUACION DE REPORTES

20. SITUACION FACTURAS

21. SITUACION SOFT. INSTALADO

22. SITUACIONES PROYECTOS

STATUS-RST alphanum 2 Null

23. SOFTWARE INSTALADO

•Deleted• •Deleted• Fixed
Heading 1: CLAVE DE CLIENTE
CLAVE DEL CLIENTE

NUM-EQUIPO numeric 2.0U Fixed
CVE-SOFT alphanum 3 Fixed
STATUS-SIN alphanum 1 Null
VERSION alphanum 10 Null
CVE-CLIENTE alphanum 10 Null

24. SOFTWARE SAG
25. TIPOS DE CLIENTE
•Deleted• •Deleted• Fixed
Heading 1: CLAVE DE CLIENTE
CLAVE DEL CLIENTE
26. TIPOS DE CLIENTES
CVE-SOFT alphanum 3 Fixed
•Deleted• •Deleted• Null
27. TIPOS DE REUNION
28. VISITAS AREA COMERCIAL
CONSULTOR numeric 3.0U Null
Heading 1: CONSULTOR
CLAVE DEL EMPLEADO QUE FUNGE COMO CONSULTOR
- FEC-VISITA date Null
PUNTOS-TRAT alphanum 60 Null
TIPO-REUNION alphanum 2 Fixed
Heading 1: TIPO REUNION
CLASIFICACION DEL TIPO DE REUNION
01 - VISITA PROSPECTO
02 - CONSULTORIA
03 - AVANCE DEL PROYECTO
04 - VISITA ESPECIAL
- CVE-PROY alphanum 6 Fixed
•Deleted• •Deleted• Fixed
CVE-CLIENTE alphanum 10 Null
29. CVE-EMP numeric 3.0U Fixed
ES-RESPONSABLE binary 1 Null
FUNCION alphanum 20 Null
•Deleted• •Deleted• Fixed
CVE-CLIENTE alphanum 10 Null

5.7 CONSTRUCCION DE PROGRAMAS, MAPAS Y BASE DE DATOS

En la etapa de construcción se realiza físicamente lo especificado en el diseño, con las herramientas que se tiene, el medio ambiente para el sistema (Hardware y Software). Para el sistema de Administración Ejecutiva de Proyectos su ambiente es el siguiente:

- Base de Datos ADABAS
- Lenguaje de programación NATURAL
- Generador de código Natural Construct
- Diccionario de Datos PREDICT
- Equipo MAINFRAME IBM

Lo especificado en la etapa de diseño como lo fue mapas, y entidades fueron realizadas con el formato de nuestro ambiente. Con los prototipos realizados solo resta unirlos y hacer pruebas del sistema. La base de datos de producción se basa en la base de datos de prueba y tomando en cuenta el formato de la base de datos ADABAS. Realización del diagrama procesos y documentación (manual).

La construcción de mapas se realiza con la información que se obtiene del editor de Natural Architect el cual con una utilidad de predict se hace la transferencia.

La base de datos se construye con las entidades que se obtuvieron en natural architect (diagrama de entidad relacion), restando especificar los descriptores, campos múltiples, periódicos.

Para la promoción primero se utiliza la herramienta CASE generador de código Natural Construct el cual genera el código del programa ya sea un programa browse o un programa de mantenimiento.

Finalmente se realizan las pruebas del sistema por módulos como se especifica en las técnicas de RAD y JAD. Al ya haber terminado este tipo de pruebas a todos los módulos se llega a la parte final, la prueba de todo el sistema y poder liberar el sistema a producción.

5.7.1 DOCUMENTACION DE CONSTRUCCION

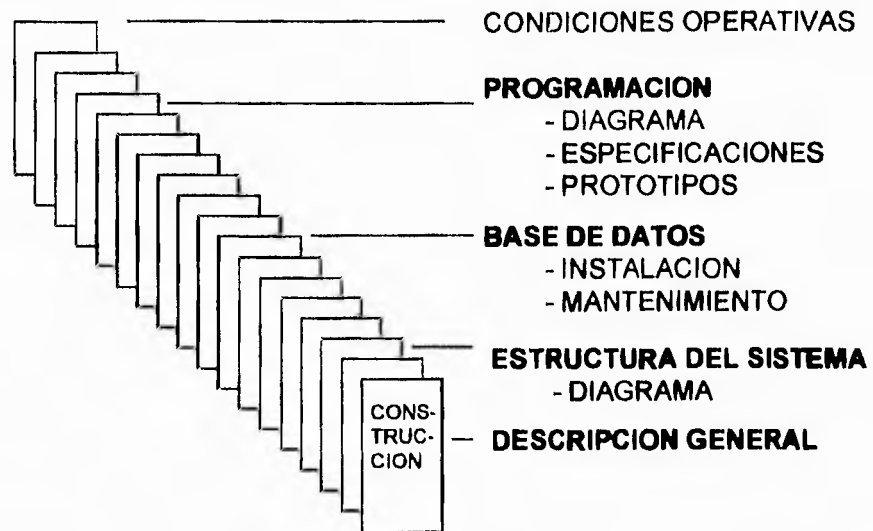


Fig. 6.2

Entre las ventajas de utilizar herramientas CASE para la etapa de construcción se cuentan la reducción del tiempo de programación, debido a que automáticamente nos genera código reutilizable, teniendo como consecuencia lógica disminución de costos en esta etapa.

PROTOTIPOS DE PANTALLAS

APBML451	DD/MM/AA				
APBML452	HH:MM PM				
*** SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS***					
----- ACTIVIDADES POR PROYECTO -----					
CLAVE DE CLIENTE:	<input type="text"/>				
CLAVE DE PROYECTO:	<input type="text"/>				
ACTIVIDAD:	<input type="text"/>				
ACTIVIDAD EN CURSO:	<input type="text"/>				
FECHA DE AVANCE:	<input type="text"/>				
FECHA INICIO	FECHA FINAL	DURACION (DIAS)	% AVANCE	% AVANCE ANTERIOR	FECHA AVANCE ANTERIOR
PLAN	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
REAL	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
NOTA:	<input type="text"/>				
TECLEE <ENTER> PARA CONTINUAR <PF4> PARA CANCELAR					

File Name: APBML452
Form Name: APBML452

Page 1

Field	Title	Edit Mask	Attributes
CYE-CLIENTE	CLAVE DE CLIENTE:		Input Only
RAZON-SOCIAL			
CYE-PROY	CLAVE DE PROYECTO:		
NOMBRE-PROY			
CYE-ACT	ACTIVIDAD:		
DESCRIP			
New_Field_8	ACTIVIDAD EN CURSO:		
New_Field_9	FECHA DE AVANCE:		
FEC-INI-PLAN	PLAN		
FEC-TER-PLAN			
New_Field_10	DURACION		
PORC-AVAN-REAL	AVANCE		
PORC-AVAN-PLAN	ANTERIOR		
New_Field_12	ANTERIOR		
FEC-INI-REAL	REAL		
FEC-TER-REAL			
New_Field_11			
REFERENCIA	NOTA:		

COMPANIA	PROGRAMA	*** SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS***	DD/MM/AA									
		REGISTRO DE PROBLEMAS	HH:MM PM									
			MODO <input type="checkbox"/>									
CLIENTE	<input type="text"/>											
PROYECTO	<input type="text"/>											
NUMERO	<input type="text"/>											
DESCRIPCION	TIPO DE PROBLEMA	<input type="text"/>										
	GRAVEDAD	<input type="text"/>										
CAUSA	<input type="text"/>											
SOLUCION	<input type="text"/>											
Entr	PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9	PF10	PF11	PF12
	AYUDA	REGRE	SALIR	CANCL	SPLIT	P-SIG	P-ANT	IMPRI	P-IZQ	P-DER	PRINC	

File Name: APANL120
Form Name: APANL120

Page 1

Field	Title	Edit Mask	Attributes
#MODO	MODO		
CLAVE_DE_CLIENTE	CLIENTE		
RAZON_SOCIAL			
CLAVE_PROYECTO			
PRY-NOM	PROYECTO		
NUMERO	NUMERO		
group0			
DESCRIPCION	DESCRIPCION		
TIPO_DE_PROBLEMA	TIPO DE PROBLEMA		
#DESC-TIPO			
GRAVEDAD	GRAVEDAD		
New_Field_2			
group1			
CAUSA	CAUSA		
group2			
SOLUCION	SOLUCION		

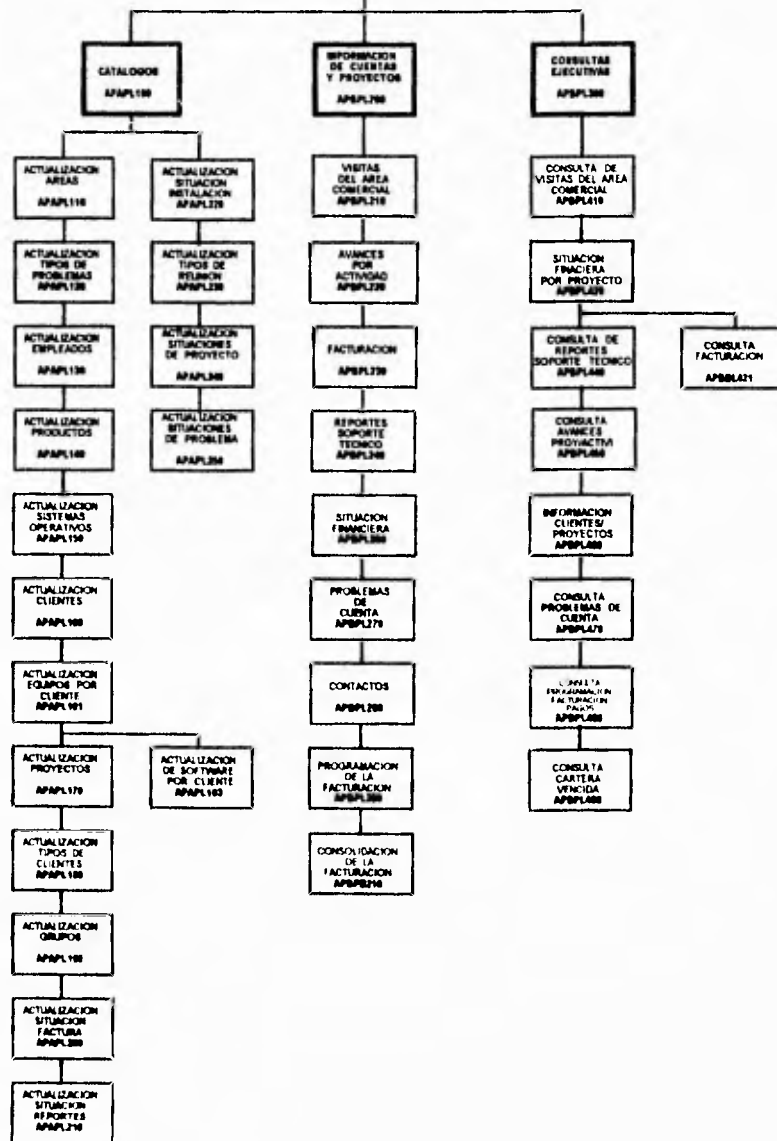
APBPL450		*** SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS ***					DD/MM/AA						
APBML450		AVANCE DE ACTIVIDADES/PROYECTO					HH:MM PM						
CUE-CLIENTE													
SEL	DESV	PROY.	DESCRIPCION	# AVR	DIAS RET.	NUM REC.	STATUS						
Entr		PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9	PF10	PF11	PF12
		AYUDA	REGRE	FIN					P-ANT	P-SIG	INPRI		MENU

<u>Field</u>	<u>Title</u>	<u>Edit Mask</u>	<u>Attributes</u>
CYE-CLIENTE	CYE-CLIENTE		
RAZON-SOCIAL			
group0			
New_Field_6	SEL		
group2			
DESY	DESY		
group4			
CYE-PROY	PROY:		
group6			
NOMBRE-PROY	DESCRIPCION		
group7			
PORC-AVANCE	AYA		
group3			
RETRASO-PRY	RET:		
group1			
REC-HUM-PRY	REC:		
group5			
STATUS-SIN	STATUS		

DIAGRAMA DE PROCESOS

ADMPROY

3/4/95



MATRIZ DE PROCESOS / ENTIDADES

MATRIZ.XLS

MATRIZ DE PROCESOS/ENTIDADES		AEP-ACTIV	AEP-AREAS*	AEP-CLAPRO*	AEP-CLIENTES	AEP-CONTAC	AEP-EMPLEA	AEP-ECCOM	AEP-FACTUR	AEP-GRUPOS*	AEP-PROB	AEP-PROY	AEP-SISOP*	AEP-SITFAC*	AEP-SITPRY*	AEP-SITREP*	AEP-SITSO*	AEP-SOFINS	AEP-SOFBAG*	AEP-SOFTEC*	AEP-TIPCLI*	AEP-TIPREU*	AEP-VISITA	AEP-SITPRO*	AEP-TIPRO*	AEP-PROFAC	
PROGRAMA	DESCRIPCION																										
APAPL110	ACT AREAS	X																									
APAPL120	ACT TIPO DE PROBLEMAS		X																								
APAPL130	ACT EMPLEADOS	X				X																					
APAPL140	ACT PROD.BAG																		X								
APAPL150	ACT S.O											X															
APAPL160	ACT CLIENTES				X																						
APAPL161	ACT TEC.CLIENTES						X	X				X															
APAPL170	ACT PROYECTOS											X										X					
APAPL180	ACT TIPOS DE CLIENTES																					X					
APAPL190	ACT GRUPO PLATAFORMA								X																		
APAPL200	ACT SIT. FACTURA												X														
APAPL210	ACT SIT REPORTES															X											
APAPL220	ACT SIT INSTALACION																X										
APAPL230	ACT.TIPO REUNION																						X				
APAPL240	ACT SIT PROYECTO													X													
APAPL250	ACT SIT PROBLEMA																							X			
APBPL210	ACT.VISITA AREA COMERCIAL						O																X	X			
APBPL220	ACT.PROYECTOS	X										X															
APBPL230	ACT FACTURACION		X					X					X														
APBPL240	ACT REPORTE SOPORTE TEC.						O									X					X						
APBPL250	ACT SIT FINANCIERA											X															
APBPL270	ACT PROB CUENTA			X							X													X	X		
APBPL280	ACT CONTACTOS DE CLIENTE		X	X																							
APBPL290	PROB DE LA FACTURACION																								X		
APBPL400	CNS SIT FINANCIERA (BRO)				X								X														
APBPL440	CNS REPROTES SOPORTE TEC.																O			O							
APBPL450	CNS FACTURACION = CLIENTE											X															
APBPL460	CNS PROG DE PAGO				X							O															
APBPL490	CNS CARTURA VENCIDA											O														X	
APBPL421	FACTURAS DE PROYECTO							O				O															
APBPL430	SITUACION FINANCIERA(INFO)		X					O			O		O	X													
APBPL461	ACTIVIDADES = PROYECTO							O			O		O														
APBPL462	CONTACTOS = PROYECTO (BRO)		O									O															
APBPL461	AVANCE DE ACTIVADES			O			O					O			O												

NOTA
X = VISTA DEFINIDA EN UN ARCHIVO
O = CONSULTA
* = TABLAS COMUNES

CODIGO DE ARCHIVOS Y PROGRAMAS

11:38

All File NO. 121

03/04/1995

```

*****
* 04/03/95                               11:37:39 *
*                               DBMS Database Services *
*                               ADABAS Report          *
*                               *                      *
*****

```

 overview of file <AEP-CLIENT> in database <SAGAEP>

```

database name      : SAGAEP
database number    : 5
file name          : AEP-CLIENT
file number        : 121
number of descriptors : 2
number of fields   : 15
length of uncompressed records : 1767
number of records  : 63
padding-factor in (%) : 0
ISN reusage        : y
space reusage      : y

```

 FDT of file <AEP-CLIENT> in database <SAGAEP>

typ	lev	sname	lname	form	leng	occ	comp	desc	uni
	1	AA	CLI-RAZON-SOCIAL	A	40		N		D
	1	AB	CLI-DIRECCION	A	60		N		
	1	AC	CLI-GRUPO	A	20		N		
	1	AD	CLI-IMP-SECTOR	A	4		N		
	1	AE	CLI-RFC	A	15		F		
	1	AF	CLI-TIPO-CLIENTE	A	1		F		
	1	AG	CLI-IMP-ESTRAT	U	2				
M	1	AH	CLI-COMENT-TECNO	A	70	3			
M	1	AI	CLI-COMUNICACIONES	A	70	3			
M	1	AJ	CLI-TELEFONOS	A	20	5	N		
P	1	AK	CLI-DIRECTIVOS			15			
	2	AL	CLI-DIREC-NOMBRE	A	25		N		
	2	AM	CLI-DIREC-PUESTO	A	30		N		
	2	AN	CLI-DIREC-TELEFONO	A	18		N		
	1	AO	CLI-CVE-CLIENTE	A	10				D

PROGRAMAS

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
0010 **SAG GENERATOR: BROWSE-SELECT-MD                               Version: 3.2.1 UP 1
0020 **SAG TITLE: LISTA DE CLIENTES
0030 **SAG SYSTEM: ADMPROY
0040 **SAG GDA: SAGG000
0050 **SAG DESCS(1): ESTE PROGRAMA MUESTRA UNA LISTA DE SELECCION
0060 **SAG DESCS(2): DE LAS VISITAS DEL AREA COMERCIAL A NUESTROS CLIENTES
0070 **SAG HEADER1: SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS
0080 **SAG HEADER2: VISITAS DEL AREA COMERCIAL
0090 **SAG DIRECT-COMMAND-PROCESS:
0100 **SAG PRIME-FILE: AEP-VISITA
0110 **SAG PRIME-KEY: SD-CLI-FVI
0120 **SAG MAP-NAME: APBML210
0130 **SAG MAX-WINDOWS: 01
0140 **SAG MAX-PAGES: 99
0150 **SAG SINGLE-REDEFINE-PROMPT: X
0160 **SAG DYNAMIC-ATTRIBUTES: ){
0170 **SAG ACTION-FORMAT: A
0180 **SAG ACTION-LENGTH: 01
0190 **SAG MAXIMUM-ACTIONS: 10
0200 **SAG ACTION-LINE: 10
0210 **SAG ACTION-COLUMN: 03
0220 **SAG ACTIONS: X XX X
0230 *****
0240 * PROGRAM   : APBPL210
0250 * System    : ADMPROY
0260 * Title     : LISTA DE CLIENTES
0270 * Generated: Jul 05,94 at 02:05 PM
0280 * Function  : ESTE PROGRAMA MUESTRA UNA LISTA DE SELECCION
0290 *           : DE LAS VISITAS DEL AREA COMERCIAL A NUESTROS CLIENTES
0300 *
0310 *
0320 * History
0330 **SAG DEFINE EXIT CHANGE-HISTORY
0340 * Changed on May 12,94 by DSEGV for release _____
0350 * >
0360 * >
0370 * >
0380 **SAG END-EXIT
0390 *****
0400 DEFINE DATA
0410 GLOBAL USING SAGG000
0420 LOCAL USING CDACTA /* Used to specify actions that program supports.
0430 LOCAL USING CDDIALDA /* Used by dialog objects.
0440 LOCAL USING CDENVIRA /* Used to capture/restore previous environment.
0450 LOCAL USING CDFLIPA /* Used to change the KD lines.
0460 LOCAL USING CDKEYLDA /* Used to set function keys and their names.
0470 LOCAL USING CDSETGBA /* Used to set logical pagesize.
0480 LOCAL USING CDWILDA /* Wild card characters in input key
0490 LOCAL
0500 *
0510 * Constants.
0520 01 #MAX-PAGES(P3) CONST<99> /* Maximum scroll pages
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
0530 01 #MAX-PANELS(P3) CONST<1> /* Max left/right panel.
0540 01 #SEL-TBL-SIZE(P3) CONST<10>
0550 01 #TOTAL-ACTION-LINES(P3)CONST<10> /* Max sel lines
0560 01 #UNCONDITIONALLY(L) CONST<TRUE> /* Used to REJECT always.
0570 *
0580 * Miscellaneous variables.
0590 01 #ACTION(A1)
0600 01 #BKWRD-TABLE-DECREMENT(P1) /* Bkwrд scroll table decrement.
0610 01 #END-OF-DATA(L) /* End of logical read.
0620 *
0630 * Save first key and ISN found in case scroll back record was deleted.
0640 01 #FIRST
0650 02 #KY(A14) /* 1st KEY when looking for match.
0660 02 #ISN(P10) /* 1st ISN when looking for match.
0670 01 #FIRST-ACTION-LINE(P3) INIT<10>
0680 /* Line where action column starts
0690 01 #FIRST-INPUT(L) INIT<TRUE> /* Trigger End of Page.
0700 01 #FIRST-ISN-FOUND(L) /* Found the first ISN
0710 01 #FORWARD(L) INIT<TRUE> /* Not backwards left or right.
0720 01 #HARDCOPY(L) /* Request Hardcopy.
0730 01 #HARDCOPY-START-KEY(A14) /* Start Key for Hardcopy.
0740 01 #HARDCOPY-PS(P3) INIT<50>
0750 01 #HEADER1(A60)
0760 INIT!'***** SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS *****
0770 01 #HEADER2(A58)
0780 INIT' - VISITAS DEL AREA COMERCIAL -'>
0790 *
0800 * Define input fields.
0810 01 #INPUT
0820 02 VAC-CVE-CLIENTE(A10)
0830 02 VAC-FEC-VISITA(D)
0840 01 REDEFINE #INPUT
0850 02 VAC-SE-CLI-FVI(A14) /* Input key value.
0860 01 REDEFINE #INPUT
0870 02 #FIRST-CHAR-ON-SCREEN(A1) /* Allow '.' to terminate.
0880 01 #INSIDE-READ-LOOP(L) /* Executing statement inside read.
0890 *
0900 * Value of the scroll key on current record
0910 01 #KEY(A14) /* Key of record.
0920 01 REDEFINE #KEY /* Redefine #KEY into a struct to
0930 02 #KEY-STRUCTURE /* use MOVE BY NAME to build super
0940 03 VAC-CVE-CLIENTE(A10)
0950 03 VAC-FEC-VISITA(D)
0960 01 #KEY-CV(C) INIT<(AD=I)> /* Determine if key was modified.
0970 01 #NEXT-LINE(P3) /* Next screen line to be written.
0980 01 #NULL-KEY(A:4) /* Null key of record.
0990 01 #NULL-ISN(P10) /* Null ISN of record.
1000 01 #NULL-ACTION(A1)
1010 01 #LEFT-PROMPT(A9) /* Screen prompt.
1020 01 #MATCH-FOUND(L) /* Matched Unique Id from table.
1030 01 #MAX-KEY(A14) /* Current maximum key value
1040 01 REDEFINE #MAX-KEY
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
1050 02 #MAX-KEY-STRUCTURE
1060 03 VAC-CVE-CLIENTE(A10)
1070 03 VAC-FEC-VISITA(D)
1080 01 #MIN-KEY(A14) /* Current minimum key value
1090 01 REDEFINE #MIN-KEY
1100 02 #MIN-KEY-STRUCTURE
1110 03 VAC-CVE-CLIENTE(A10)
1120 03 VAC-FEC-VISITA(D)
1130 01 #PREFIX-FOUND(L) /* Recs exist with current prefix.
1140 01 #PANEL(P3) INIT<1> /* Left/Right panel number.
1150 01 #REDISPLAY-SCREEN(L) /* If true, don't scroll forward.
1160 01 #RIGHT-PROMPT(A9) /* Screen Prompt
1170 01 #SCR-CV(C) INIT<(AD=I)> /* Determine if any field modified.
1180 01 #SEL-INDX(P5) /* Select table index.
1190 01 #SEL-TBL(1:#TOTAL-ACTION-LINES) /* Selection table.
1200 02 #KY(A14)
1210 02 #ISN(P10)
1220 02 #ACTIONS(A1) /* Function code column.
1230 02 #ACTION-CV(C) INIT<(AD=NP)> /* Control var. for func. code col
1240 01 #SELECTED-KEY(A14) /* Key of selected record.
1250 01 #SELECTED-ISN(P10) /* ISN of selected record.
1260 *
1270 * Key value for starting position of read.
1280 01 #START
1290 02 #KY(A14)
1300 02 REDEFINE #KY
1310 03 VAC-CVE-CLIENTE(A10)
1320 03 VAC-FEC-VISITA(D)
1330 02 #ISN(P10)
1340 *
1350 * Tables for backward/sideways scrolling feature
1360 01 #SCROLL-INDX(P3) /* Index of Scroll Table.
1370 01 #SCROLL-TBL(1:#MAX-PAGES)
1380 02 #KY(A14) /* Store top key/page.
1390 02 #ISN(P10) /* Store top ISN on each page.
1400 01 #STD-KEY-PRESSED(L) /* Return, Quit, Flip, Main.
1410 01 #TOP-LINE(P3)
1420 01 #TOP-OF-PAGE(L) /* Top of Page was triggered.
1430 *
1440 * Fields used for interpreting wild card characters
1450 * in the input key.
1460 01 #WCTABLE(1:1) /* Wild card characters table
1470 02 #WILD-FIELDS(A10) /* Input field to be checked
1480 02 REDEFINE #WILD-FIELDS
1490 03 #WILD-FIELDS-CHAR(A1/1:10)
1500 02 #LAST-CHAR(A1) /* Last character of input field
1510 *
1520 01 #CURRENT-LAST-CHAR(A1) /* Last character on input
1530 01 #HIGH-VALUE(A14) /* Used for setting max values
1540 01 #LAST-POS(P3) /* Last position filled on input
1550 01 #WILD-FIELD(A10) /* Used for wild character check
1560 01 #WINDX(P3) /* Temporary index
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
1570 *
1580 * Views.
1590 *
1600 * Primary view being scrolled.
1610 01 AEP-VISITA VIEW OF AEP-VISITA
1620 02 VAC-CONSULTOR
1630 02 VAC-FEC-VISITA
1640 02 VAC-TIPO-REUNION
1650 02 VAC-CVE-CLIENTE
1660 02 VAC-CVE-PROY
1670 02 VAC-PUNTOS-TRAT(1:3)
1680 02 VAC-COMPROMISO(1:3)
1690 02 VAC-FEC-COMP
1700 **SAG DEFINE EXIT LOCAL-DATA
1710 LOCAL
1720 **
1730 01 VISITA VIEW OF AEP-VISITA
1740 02 VAC-CONSULTOR
1750 02 VAC-FEC-VISITA
1760 02 VAC-TIPO-REUNION
1770 02 VAC-CVE-CLIENTE
1780 02 VAC-CVE-PROY
1790 02 VAC-PUNTOS-TRAT(1:3)
1800 02 VAC-COMPROMISO(1:3)
1810 02 VAC-FEC-COMP
1820 LOCAL
1830 01 #LLAVE (A60)
1840 01 REDEFINE #LLAVE
1850 02 #LLA-TIPREU (A02)
1860 01 #DATOS (A250)
1870 01 REDEFINE #DATOS
1880 02 #DAT-TIPREU (A30)
1890 LOCAL
1900 01 #MOD-CV (C)
1910 01 #DIAS (N3)
1920 01 #CLI-CV (C)
1930 01 #FEC (D)
1940 01 #LLA-CV (C)
1950 01 #SW-CANC (L)
1960 01 #SW (L)
1970 01 #ISN-T (P08)
1980 01 #MIN-PAR (A05)
1990 01 #MOD0 (A20)
2000 01 #NOM-CONSULTOR(A15)
2010 01 #CLIPRY (A50)
2020 01 REDEFINE #CLIPRY
2030 02 #CLI (A10)
2040 02 #DPRY (A40)
2050 01 #NOM-CONSU (A15)
2060 01 #D-TIPREU (A30)
2070 01 #CONSULTOR (A15)
2080 01 #FEC-CLI (A40)
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
2090 01 #PUNTOS      (A30)
2100 01 #DES-PROY   (A40)
2110 01 #DES-TIPREU (A30)
2120 01 #FECHA-A    (A08)
2130 01 REDEFINE #FECHA-A
2140 02 #FECHA-N    (N08)
2150 **SAG END-EXIT
2160 END-DEFINE
2170 *
2180 * Define Formats
2190 FORMAT KD=ON LS=133 SG=OFF ES=OFF ZP=ON
2200 *
2210 PERFORM INITIALIZATIONS
2220 *
2230 *
2240 *
2250 **SAG DEFINE EXIT START-OF-PROGRAM
2260 *
2270 * Processing to be performed once at the start of the program.
2280 MOVE *PROGRAM TO #PROG
2290 BACKOUT TRANSACTION
2300 SET CONTROL 'N'
2310 **SAG END-EXIT
2320 PROG.
2330 REPEAT /* Repeat loop to allow escape of program from within subroutine.
2340 *
2350 ***** Start of Main Program Logic *****
2360 *
2370 RESET #FIRST-ISN-FOUND #REDISPLAY-SCREEN #MATCH-FOUND
2380 #REDISPLAY-SCREEN
2390 *
2400 * If we are scrolling backwards or left/right then obtain our starting
2410 * value from the table if one exists. If the table does not yet
2420 * contain a starting value the currently specified input value will
2430 * be used.
2440 IF #SCROLL-INDX LT 1 THEN /* Nothing in table yet.
2450 ASSIGN #FORWARD = TRUE
2460 END-IF
2470 IF #FORWARD THEN /* Use input values.
2480 IF #CURRENT-LAST-CHAR = CDWILDA.#WILD-CARD-CHARS(*) THEN
2490 RESET #START.#KY
2500 ELSE
2510 ASSIGN #START.#KY = #INPUT.VAC-SD-CLI-FVI
2520 ASSIGN #MATCH-FOUND = TRUE
2530 END-IF
2540 ELSE /* Use table values.
2550 ASSIGN #START.#KY = #SCROLL-TBL.#KY(#SCROLL-INDX) /* Match this key.
2560 ASSIGN #START.#ISN = #SCROLL-TBL.#ISN(#SCROLL-INDX)
2570 /* Match this unique val.
2580 END-IF
2590 *
2600 * Don't allow a key less than the minimum key specified
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
2610 IF #START.#KY < #MIN-KEY THEN
2620   ASSIGN #START.#KY = #MIN-KEY
2630 END-IF
2640 NEW-READ.
2650 REPEAT /* Escape this repeat if no end of data processing.
2660   /*
2670   /* Very first time trigger a new page to input starting value.
2680   IF #FIRST-INPUT THEN
2690     RESET #FIRST-INPUT
2700     WRITE ' ' /* Trigger newpage.
2710     NEWPAGE
2720   END-IF
2730 *
2740   RESET #END-OF-DATA
2750   RESET #PREFIX-FOUND
2760 READ-FILE. /* Escape this label if end of data processing is needed.
2770   READ AEP-VISITA BY VAC-SD-CLI-FVI
2780   STARTING FROM #START.#KY
2790 **SAG DEFINE EXIT READ-CONDITION
2800   IF AEP-VISITA.VAC-CVE-CLIENTE NE #INPUT.VAC-CVE-CLIENTE
2810     ESCAPE BOTTOM(READ-FILE.)
2820   END-IF
2830 **SAG END-EXIT
2840   ASSIGN #INSIDE-READ-LOOP = TRUE
2850   /*
2860   /* Build #KEY to equal super descriptor VAC-SD-CLI-FVI components.
2870   MOVE BY NAME AEP-VISITA TO #KEY-STRUCTURE
2880   /*
2890   /* If we reached the maximum allowed key do end of data processing
2900   IF #KEY > #MAX-KEY THEN
2910     ESCAPE BOTTOM(READ-FILE.) IMMEDIATE
2920   END-IF
2930   /*
2940   /* If scrolling back/sideways, try to find the same key and
2950   /* ISN as in scroll table.
2960   DECIDE FOR FIRST CONDITION
2970     WHEN #FORWARD IGNORE /* Don't check table at all.
2980     WHEN #MATCH-FOUND IGNORE /* Already matched record.
2990     WHEN *ISN(READ-FILE.) = #START.#ISN
3000     ASSIGN #MATCH-FOUND = TRUE /* This record matches table.
3010     WHEN NONE
3020     /*
3030     /* We are still trying to match the record from the previous
3040     /* top of page.
3050     IF NOT #FIRST-ISN-FOUND THEN
3060     /*
3070     /* Save the first mismatch in case we don't find a match.
3080     ASSIGN #FIRST.#ISN = *ISN(READ-FILE.)
3090     ASSIGN #FIRST.#KY = #KEY
3100     ASSIGN #FIRST-ISN-FOUND = TRUE
3110   END-IF
3120   IF #KEY > #START.#KY THEN /* Record was deleted
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
3130          /*
3140          /* We have gone passed the key value without finding a match
3150          /* reassign the search values to the first record we found.
3160          ASSIGN #SCROLL-TBL.#KY(#SCROLL-INDX) = #FIRST.#KY
3170          ASSIGN #SCROLL-TBL.#ISN(#SCROLL-INDX) = #FIRST.#ISN
3180          PERFORM EXIT-READ-LOOP
3190          ELSE
3200              REJECT #UNCONDITIONALLY          /* Keep looking for match.
3210          END-IF
3220          END-DECIDE
3230          ASSIGN #PREFIX-FOUND - TRUE          /* Records exist with current prefix
3240          **SAG DEFINE EXIT WRITE-FIELDS
3250          RESET #NOM-CONSULTOR #DES-TIPREU #DES-CLI
3260          CALLNAT 'APANL130' AEP-VISITA.VAC-CONSULTOR #NOM-CONSULTOR #ISN-T #SW
3270          CALLNAT 'APANL160' AEP-VISITA.VAC-CVE-CLIENTE #DES-CLI #ISN-T #SW
3280          MOVE AEP-VISITA.VAC-TIPO-REUNION TO #LLA-TIPREU
3290          CALLNAT 'APANL001' 'AEP-TIPREU' #LLAVE #DATOS #SW
3300          MOVE #DAT-TIPREU TO #D-TIPREU
3310          WRITE(0)
3320              6X #ACTION
3330              6X AEP-VISITA.VAC-FEC-VISITA
3340              3X AEP-VISITA.VAC-PUNTOS-TRAT(1) (EM=X(30))
3350              3X #NOM-CONSULTOR(EM=X(15))
3360          **SAG END-EXIT
3370          PERFORM UPDATE-SELECTION-TABLE
3380          /*
3390          /* Display headings and load the scrolling table with key and
3400          /* Unique record identifier for backward/sideways scrolling.
3410          AT TOP OF PAGE
3420              IF #HARDCOPY THEN
3430                  ASSIGN CDSETGBA.#GLOBAL-PAGESIZE = #HARDCOPY-PS
3440                  CALLNAT 'CDSETGB' CDSETGBA
3450              END-IF
3460              ASSIGN #TOP-OF-PAGE = TRUE
3470              ASSIGN #BKWRD-TABLE-DECREMENT = 1
3480              RESET INITIAL #SEL-TBL(*)          /* Clear select table.
3490              MOVE EDITED *TIMX(EM=HH:'II' 'AP) TO #RIGHT-PROMPT
3500              REPEAT UNTIL *LINE-COUNT GE #FIRST-ACTION-LINE
3510                  WRITE NOTITLE(ES=OFF) ' '
3520              END-REPEAT
3530          /*
3540          /* If we are scrolling forward, and we have found a record
3550          /* and we have not hit end of data then put entry in table.
3560          IF #FORWARD AND #INSIDE-READ-LOOP THEN
3570              ADD 1 TO #SCROLL-INDX
3580          /*
3590          /* If scroll table is filled, shift out oldest entry.
3600          IF #SCROLL-INDX GT #MAX-PAGES THEN
3610              ASSIGN #SCROLL-INDX = #MAX-PAGES
3620              MOVE BY NAME #SCROLL-TBL(2:#MAX-PAGES)
3630                  TO #SCROLL-TBL(1:#MAX-PAGES - 1)
3640          END-IF
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
3650         ASSIGN #SCROLL-TBL.#KY(#SCROLL-INDX) = #KEY
3660         ASSIGN #SCROLL-TBL.#ISN(#SCROLL-INDX) = *ISN(READ-FILE.)
3670     END-IF
3680     ASSIGN #TOP-LINE = #FIRST-ACTION-LINE
3690 END-TOPPAGE
3700 /*
3710 /* Input the screen and check users PF keys.
3720 AT END OF PAGE
3730 /*
3740 /* Add final record to our selection table.
3750 PERFORM UPDATE-SELECTION-TABLE
3760 /*
3770 /* Protect all but the first line of each new record.
3780 FOR #SEL-INDX = 1 TO #TOTAL-ACTION-LINES
3790     IF #SEL-TBL.#ISN(#SEL-INDX) = #NULL-ISN THEN
3800         MOVE (AD=NP) TO #SEL-TBL.#ACTION-CV(#SEL-INDX)
3810     ELSE
3820         MOVE (AD=I) TO #SEL-TBL.#ACTION-CV(#SEL-INDX)
3830     END-IF
3840 END-FOR
3850     ASSIGN #NEXT-LINE = 1 /* To avoid duplicate ISNs in #SEL-TBL.
3860 **SAG DEFINE EXIT BEFORE-INPUT
3870 *
3880 * This exit is invoked just prior to the END-OF-PAGE INPUT
3890     MOVE (AD=I) TO #ACTION-CV(01)
3900 *     RESET #INPUT.VAC-CVE-CLIENTE
3910 *     #INPUT.VAC-FEC-VISITA
3920 **SAG END-EXIT
3930 IF #HARDCOPY THEN
3940     SET CONTROL 'H'
3950 ELSE
3960     INPUT WITH TEXT MSG-INFO.##MSG,
3970         MSG-INFO.##MSG-DATA(1), MSG-INFO.##MSG-DATA(2),
3980         MSG-INFO.##MSG-DATA(3) MARK *#INPUT.VAC-CVE-CLIENTE
3990     USING MAP 'APBML210' NO ERASE
4000     RESET MSG-INFO
4010     PERFORM VALIDATE-ACTION-CODES
4020     INCLUDE CSTDCHK /* Check standars pf-keys or '.' if first pos
4030     IF #STD-KEY-PRESSED THEN
4040         ASSIGN #FORWARD = FALSE
4050         PERFORM EXIT-READ-LOOP
4060     END-IF
4070 **SAG DEFINE EXIT AFTER-INPUT
4080 RESET #DES-CLI
4090 MOVE #INPUT.VAC-CVE-CLIENTE TO AEP-VISITA.VAC-CVE-CLIENTE
4100 CALLNAT 'APAN160' AEP-VISITA.VAC-CVE-CLIENTE #DES-CLI #ISN-T #SW
4110 **SAG END-EXIT
4120 IF *PF-KEY = CDKEYLDA.#HARDCOPY-KEY
4130     ASSIGN #HARDCOPY = TRUE
4140     IF #SCROLL-INDX > 0 THEN
4150         ASSIGN #START.#KY = #SCROLL-TBL.#KY(#SCROLL-INDX)
4160         ASSIGN #HARDCOPY-START-KEY = #SCROLL-TBL.#KY(#SCROLL-INDX)
```


11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
4170         ASSIGN #FORWARD = FALSE
4180         PERFORM EXIT-READ-LOOP
4190     END-IF
4200 END-IF
4210 /*
4220 /* Check scroll keys.
4230 DECIDE ON FIRST *PF-KEY
4240     VALUE CDKEYLDA.#BACKWARD-KEY /* Scroll backward
4250     /*
4260     /* Get previous starting value from the table.
4270     IF #SCROLL-INDX LE #BKWRD-TABLE-DECREMENT THEN
4280         REINPUT 'Top of current scroll session reached' ALARM
4290     END-IF
4300     SUBTRACT #BKWRD-TABLE-DECREMENT FROM #SCROLL-INDX
4310 ANY
4320     ASSIGN #FORWARD = FALSE
4330     PERFORM EXIT-READ-LOOP
4340 NONE
4350     IF NOT(*PF-KEY = 'ENTR' OR = CDKEYLDA.#FORWARD-KEY
4360         OR = CDKEYLDA.#HARDCOPY-KEY) THEN
4370         REINPUT 'Invalid program function key' ALARM
4380     END-IF
4390     ASSIGN #FORWARD = TRUE
4400 END-DECIDE
4410 /*
4420 /* Check whether the user changed (overtyped) the input values.
4430 DECIDE FOR FIRST CONDITION
4440     WHEN #KEY-CV MODIFIED /* Key was modified start new read.
4450         PERFORM CHECK-WILD-CHARACTER
4460         RESET #SCROLL-INDX
4470         ASSIGN #FORWARD = TRUE
4480     WHEN #SCR-CV MODIFIED OR #REDISPLAY-SCREEN
4490     /*
4500     /* If user fields were modified or FLIP key was pressed,
4510     /* show the current screen again.
4520     ASSIGN #FORWARD = FALSE
4530     WHEN ANY
4540         PERFORM EXIT-READ-LOOP
4550     WHEN NONE
4560         IF #END-OF-DATA THEN
4570             REINPUT 'Fin de datos, teclee nuevo valor de inicio' ALARM
4580         END-IF
4590     END-DECIDE
4600     END-IF /* #HARDCOPY
4610     END-ENDPAGE /* AT END OF PAGE
4620     END-READ /* read of file prime file.
4630     ASSIGN #INSIDE-READ-LOOP = FALSE
4640     ASSIGN #END-OF-DATA = TRUE
4650     /*
4660     /* Give message if no records were found with the current
4670     /* value of the prefix
4680     IF NOT #PREFIX-FOUND THEN
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
4690     ASSIGN MSG-INFO.##MSG = 'No hay registros para el valor actual'
4700     END-IF
4710     /*
4720     /* Determine whether the 'End of Data' line is the only line written
4730     /* to the screen. If so, don't decrement #SCROLL-INDX for PF7.
4740     ASSIGN #TOP-OF-PAGE = FALSE
4750     WRITE NOHDR 27T '*****' 'End of Data' (I) '*****'
4760     IF #HARDCOPY THEN
4770         ASSIGN CDSETGBA.#GLOBAL-PAGESIZE = 19
4780         CALLNAT 'CDSETGB' CDSETGBA
4790         ASSIGN #START.#KY = #HARDCOPY-START-KEY
4800         NEWPAGE
4810         RESET #HARDCOPY
4820         ESCAPE TOP
4830     END-IF
4840     IF #TOP-OF-PAGE = TRUE THEN
4850         ASSIGN #BKWRD-TABLE-DECREMENT = 0
4860     END-IF
4870     NEWPAGE
4880 *
4890 *****
4900 DEFINE SUBROUTINE EXIT-READ-LOOP
4910 *****
4920 *
4930     ESCAPE BOTTOM(NEW-READ.) IMMEDIATE
4940 END-SUBROUTINE /* EXIT-READ-LOOP
4950 *
4960     END-REPEAT /* NEW-READ.
4970     PERFORM PROCESS-SELECTION-COLUMN
4980     /*
4990     IF #STD-KEY-PRESSED THEN
5000         RESET #STD-KEY-PRESSED
5010         INCLUDE CCSTDKEY
5020     END-IF
5030 *
5040 * Subroutines (in alphabetical order).
5050 *
5060 *
5070 *****
5080 DEFINE SUBROUTINE CHECK-WILD-CHARACTER
5090 *****
5100 *
5110 * Check for wild characters in the input key
5120 * Reset minimum and maximum values for the key accordingly
5130 ASSIGN #WILD-FIELDS(1) = #INPUT.VAC-CVE-CLIENTE
5140 RESET #LAST-CHAR(*)
5150 FOR #WINDX = 1 TO 1
5160     ASSIGN #WILD-FIELD = #WILD-FIELDS(#WINDX)
5170     EXAMINE #WILD-FIELD FOR ' ' GIVING LENGTH #LAST-POS
5180     IF #LAST-POS NE 0 THEN
5190         /*
5200         /* Find last character and save it in array
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
5210 ASSIGN #LAST-CHAR(#WINDX) = #WILD-FIELDS-CHAR(#WINDX,#LAST-POS)
5220 /*
5230 /* Drop wild character
5240 IF #LAST-CHAR(#WINDX) = CDWILDA.#WILD-CARD-CHARS(*) THEN
5250     RESET #WILD-FIELDS-CHAR(#WINDX,#LAST-POS)
5260 END-IF
5270 END-IF
5280 END-FOR
5290 *
5300 RESET #CURRENT-LAST-CHAR
5310 DECIDE FOR FIRST CONDITION
5320 WHEN #LAST-CHAR(1) = CDWILDA.#ASTERISK OR =
5330     CDWILDA.#LESS-OR-EQUAL OR = CDWILDA.#GREAT-OR-EQUAL
5340     ASSIGN #CURRENT-LAST-CHAR = #LAST-CHAR(1)
5350 /*
5360 /* Assign Min/Max for subsequent components
5370 RESET #MIN-KEY.VAC-FEC-VISITA
5380 MOVE EDITED '26991231' TO #MAX-KEY.VAC-FEC-VISITA (EM=YYYYMMDD)
5390 DECIDE ON FIRST VALUE OF #LAST-CHAR(1)
5400     VALUE CDWILDA.#ASTERISK
5410     /*
5420     /* Assign Min/Max for current alpha component
5430     ASSIGN #MIN-KEY.VAC-CVE-CLIENTE = #WILD-FIELDS(1)
5440     COMPRESS #WILD-FIELDS(1) #HIGH-VALUE INTO
5450     #MAX-KEY.VAC-CVE-CLIENTE LEAVING NO
5460     VALUE CDWILDA.#LESS-OR-EQUAL
5470     /*
5480     /* Assign Min/Max for current alpha component
5490     MOVE ALL H'00' TO #MIN-KEY.VAC-CVE-CLIENTE
5500     ASSIGN #MAX-KEY.VAC-CVE-CLIENTE = #WILD-FIELDS(1)
5510     VALUE CDWILDA.#GREAT-OR-EQUAL
5520     /*
5530     /* Assign Min/Max for current alpha component
5540     ASSIGN #MIN-KEY.VAC-CVE-CLIENTE = #WILD-FIELDS(1)
5550     MOVE ALL H'FF' TO #MAX-KEY.VAC-CVE-CLIENTE
5560     NONE IGNORE
5570 END-DECIDE
5580 WHEN NONE
5590 /*
5600 /* Re-assign initial Min/Max keys
5610 RESET #MIN-KEY.#MIN-KEY-STRUCTURE
5620 MOVE ALL H'FF' TO #MAX-KEY.VAC-CVE-CLIENTE
5630 MOVE EDITED '26991231' TO #MAX-KEY.VAC-FEC-VISITA (EM=YYYYMMDD)
5640 END-DECIDE
5650 *
5660 END-SUBROUTINE /* CHECK-WILD-CHARACTER
5670 *
5680 *****
5690 DEFINE SUBROUTINE FINAL-PROCESSING
5700 *****
5710 *
5720 INCLUDE CPEND /* Restore environment
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
5730 SET CONTROL 'KO' /* Reset *PF-KEY
5740 **SAG DEFINE EXIT END-OF-PROGRAM
5750 *
5760 * Processing to be performed once at the end of the program.
5770 **FETCH 'SAGP255' #MENU-PDR #MENU-SIS #MENU-PDR
5780 ESCAPE ROUTINE IMMEDIATE
5790 **SAG END-EXIT
5800 END-SUBROUTINE /* FINAL-PROCESSING
5810 *
5820 *.....
5830 DEFINE SUBROUTINE INITIALIZATIONS
5840 *.....
5850 *
5860 ASSIGN CENVIRA.#CAPTURE-PS = TRUE
5870 INCLUDE CCBEGIN /* Capture environment.
5880 *
5890 * Initialize help parameters and current program name.
5900 ASSIGN #SYSTEM = 'ADMPROY'
5910 MOVE ALL H'FF' TO #HIGH-VALUE
5920 IF *LEVEL = 1
5930 ASSIGN ERROR-INFO.##LAST-PROGRAM = *PROGRAM
5940 END-IF
5950 MOVE EDITED *DATX(EM=LLL-DD', 'YY) TO #LEFT-PROMPT
5960 SET CONTROL 'WBM' /* Set window size and location.
5970 *
5980 * Specify actions that program supports.
5990 ASSIGN CDACTA.#ADD(1) = 'X'
6000 ASSIGN CDACTA.#PURGE(1) = 'X'
6010 ASSIGN CDACTA.#DISPLAY(1) = 'X'
6020 ASSIGN CDACTA.#MODIFY(1) = 'X'
6030 ASSIGN CDACTA.#CALLING-PROGRAM = *PROGRAM
6040 CALLNAT 'CDACT' CDACTA
6050 ASSIGN CDDIALDA.#KD-LINES(*) = CDACTA.#KD-LINES(*)
6060 ASSIGN CDFLIPA.#AVAILABLE-COMMANDS = TRUE
6070 *
6080 * Initialize minimum/maximum key values
6090 RESET #MIN-KEY.#MIN-KEY-STRUCTURE
6100 MOVE ALL H'FF' TO #MAX-KEY.VAC-CVE-CLIENTE
6110 MOVE EDITED '26991231' TO #MAX-KEY.VAC-FEC-VISITA (EM=YYYYMMDD)
6120 *
6130 * Set Global Pagesize.
6140 ASSIGN CDSETGBA.#GLOBAL-PAGESIZE = 19
6150 CALLNAT 'CDSETGB' CDSETGBA
6160 *
6170 * Specify which PF-KEYS are valid by using RESET INITIAL.
6180 INCLUDE CCPFRNT /* Print Key
6190 INCLUDE CCPFSTD /* Standard PF-Keys
6200 INCLUDE CCPFUD /* Scroll up-down keys
6210 PERFORM SET-KEYS
6220 END-SUBROUTINE /* INITIALIZATIONS
6230 *
6240 *.....
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
6250 DEFINE SUBROUTINE PROCESS-SELECTED-RECORD
6260 *****
6270 *
6280 **SAG DEFINE EXIT PROCESS-SELECTED-RECORD
6290 * This user exit is invoked once for each record that is marked with a
6300 * valid action. The following variables are set prior to invoking this
6310 * exit:
6320 * #ACTION          /* The action that was entered by the user
6330 * #SELECTED-KEY /* The prime key (ie. browse key) of the record selected
6340 * #SELECTED-ISN /* ADABAS files only.
6350 * #SELECTED-UQ /* Unique key value of record selected (non-ADABAS file)
6360 * #SELECTED-SEQ /* Sequence number for records with the same key value
6370 * Based on these variables you can perform whatever desired function you
6380 * wish.
6390   RESET #SW-CANC #DIAS
6400   MOVE #INPUT.VAC-CVE-CLIENTE TO VISITA.VAC-CVE-CLIENTE
6410   MOVE #INPUT.VAC-CVE-CLIENTE TO #CLI
6420   MOVE (AD=PI) TO #CLI-CV
6430   DECIDE ON FIRST VALUE #SEL-TBL.#ACTIONS(#SEL-INDX)
6440     VALUE CDACTA.#ADD(*)
6450     COMPRESS 'ALTA' '****' TO #MODO
6460     MOVE 'DATX' TO VISITA.VAC-FEC-VISITA
6470     MOVE (AD=DI) TO #LLA-CV #MOD-CV
6480     RESET #NOM-CONSU #D-TIPREU #DPY #DES-TIPREU
6490     PERFORM ACTUALIZA
6500     STORE VISITA
6510     IF #INPUT.VAC-CVE-CLIENTE = ' ' THEN
6520       MOVE VISITA.VAC-CVE-CLIENTE TO #INPUT.VAC-CVE-CLIENTE
6530       MOVE TRUE TO #FORWARD
6540     ELSE
6550       IF VISITA.VAC-CVE-CLIENTE LE #INPUT.VAC-CVE-CLIENTE
6560         MOVE VISITA.VAC-CVE-CLIENTE TO #INPUT.VAC-CVE-CLIENTE
6570         MOVE TRUE TO #FORWARD
6580       END-IF
6590     END-IF
6600     VALUE CDACTA.#MODIFY(*)
6610     COMPRESS 'MODIFICACION' '****' TO #MODO
6620     MOVE (AD=PI) TO #LLA-CV
6630     MOVE (AD=DI) TO #MOD-CV
6640     LEE-MOD. GET VISITA #SELECTED-ISN
6650     PERFORM LEE-INFORMACION
6660     PERFORM ACTUALIZA
6670     UPDATE (LEE-MOD.)
6680     VALUE CDACTA.#DISPLAY(*)
6690     LEE-CON. GET VISITA #SELECTED-ISN
6700     COMPRESS 'CONSULTA' '****' TO #MODO
6710     MOVE (AD=PI) TO #LLA-CV #MOD-CV
6720     PERFORM LEE-INFORMACION
6730     PERFORM ACTUALIZA
6740     VALUE CDACTA.#PURGE(*)
6750     COMPRESS 'BAJA' '****' TO #MODO
6760     MOVE (AD=PI) TO #LLA-CV #MOD-CV
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
6770     LEE-BAJ. GET VISITA #SELECTED-ISN
6780     PERFORM LEE-INFORMACION
6790     PERFORM ACTUALIZA
6800     IF NOT #SW-CANC
6810         DELETE (LEE-BAJ.)
6820     END-IF
6830     NONE IGNORE
6840 END-DECIDE
6850     IF #SW-CANC
6860         MOVE 'PROCESO CANCELADO' TO ##MSG
6870         BACKOUT TRANSACTION
6880     ELSE
6890         IF #MOD-CV MODIFIED OR
6900             #SEL-TBL.#ACTIONS(#SEL-INDX) = CDACTA.#PURGE(*)
6910             MOVE 'REGISTRO ADICIONADO O MODIFICADO' TO ##MSG
6920             END TRANSACTION
6930         END-IF
6940     END-IF
6950     RESET VISITA #SW-CANC
6960     SET CONTROL 'WB'
6970 **SAG END-EXIT
END-SUBROUTINE /* PROCESS-SELECTED-RECORD
6990 *
7000 *****
7010 DEFINE SUBROUTINE PROCESS-SELECTION-COLUMN
7020 *****
7030 *
7040     FOR #SEL-INDX = 1 TO #TOTAL-ACTION-LINES
7050         IF #SEL-TBL.#ACTIONS(#SEL-INDX) NE #NULL-ACTION THEN
7060             ASSIGN #SELECTED-KEY = #SEL-TBL.#KY(#SEL-INDX)
7070             ASSIGN #SELECTED-ISN = #SEL-TBL.#ISN(#SEL-INDX)
7080             ASSIGN #ACTION      = #SEL-TBL.#ACTIONS(#SEL-INDX)
7090             PERFORM PROCESS-SELECTED-RECORD
7100         END-IF
7110     END-FOR
7120     RESET #SEL-TBL.#ACTIONS(*)
7130 END-SUBROUTINE /* PROCESS-SELECTION-COLUMN
7140 *
7150 *****
7160 DEFINE SUBROUTINE SET-KEYS
7170 *****
7180 *
7190     INCLUDE CCSETKEY /* Set specified keys.
7200     INCLUDE CCSETHLP /* Set PF1=HELP
7210 **AVAILABLE-COMMAND-LIST                                poner 1 aqui ----->
7220 END-SUBROUTINE /* SET-KEYS
7230 *
7240 *****
7250 DEFINE SUBROUTINE UPDATE-SELECTION-TABLE
7260 *****
7270 *
7280     IF #TOP-OF-PAGE THEN
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
7290     ASSIGN #SEL-INDX = 1
7300     RESET #TOP-OF-PAGE
7310     ELSE
7320     ASSIGN #SEL-INDX = #NEXT-LINE
7330     END-IF
7340     COMPUTE #NEXT-LINE = *LINE-COUNT + 1 - #TOP-LINE
7350     IF #SEL-INDX = 1 THRU #TOTAL-ACTION-LINES AND #INSIDE-READ-LOOP THEN
7360     ASSIGN #SEL-TBL.#KY(#SEL-INDX) = #KEY /* Load key into table
7370     ASSIGN #SEL-TBL.#ISN(#SEL-INDX) = *ISN(READ-FILE.) /* Load ISN
7380     END-IF
7390     END-SUBROUTINE /* UPDATE-SELECTION-TABLE
7400     *
7410     *****
7420     DEFINE SUBROUTINE VALIDATE-ACTION-CODES
7430     *****
7440     *
7450     * Validate all function codes before processing.
7460     FOR #SEL-INDX = 1 TO #TOTAL-ACTION-LINES
7470     /*
7480     /* Make sure all non-blank selections are in our mnemonic table.
7490     IF #SEL-TBL.#ACTIONS(#SEL-INDX) NE #NULL-ACTION THEN
7500     DECIDE FOR FIRST CONDITION
7510     WHEN NOT ( #SEL-TBL.#ACTIONS(#SEL-INDX) = CDACTA.#ADD(*)
7520     OR #SEL-TBL.#ACTIONS(#SEL-INDX) = CDACTA.#PURGE(*)
7530     OR #SEL-TBL.#ACTIONS(#SEL-INDX) = CDACTA.#DISPLAY(*)
7540     OR #SEL-TBL.#ACTIONS(#SEL-INDX) = CDACTA.#MODIFY(*) )
7550     REINPUT #INVALID-ACT-MSG, #INVALID-ACT-MSG-DATA
7560     MARK *#SEL-TBL.#ACTIONS(#SEL-INDX) ALARM
7570     WHEN *PF-KEY NE 'ENTR'
7580     REINPUT 'Linea de accion puede especificarse en conjunto' -
7590     ' con <ENTER>'
7600     MARK *#SEL-TBL.#ACTIONS(#SEL-INDX) ALARM
7610     WHEN NONE
7620     ASSIGN #REDISPLAY-SCREEN = TRUE
7630     END-DECIDE
7640     END-IF
7650     END-FOR
7660     END-SUBROUTINE /* VALIDATE-ACTION-CODES
7670     **SAG DEFINE EXIT MISCELLANEOUS-SUBROUTINES
7680     *-----
7690     DEFINE SUBROUTINE ACTUALIZA
7700     FORMAT KD=OFF LS=80
7710     INPUT USING MAP 'APBML211'
7720     IF #SEL-TBL.#ACTIONS(#SEL-INDX) = CDACTA.#ADD(*) OR= CDACTA.#MODIFY(*)
7730     * OR= CDACTA.#PURGE(*)
7740     DECIDE FOR EVERY CONDITION
7750     WHEN *PF-KEY = 'PF4'
7760     MOVE TRUE TO #SW-CANC
7770     SET CONTROL 'WB'
7780     ESCAPE ROUTINE
7790     WHEN NOT (*PF-KEY = 'PF4' OR= 'ENTR')
7800     REINPUT FILL 'PF INVALIDA' MARK 1 ALARM
```

11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
7810 WHEN VISITA.VAC-FEC-VISITA = 0 AND #MODO=SCAN 'ALTA'
7820 REINPUT FULL 'FECHA REQUERIDA' MARK *VISITA.VAC-FEC-VISITA
7830 WHEN VISITA.VAC-FEC-VISITA NE 0 AND #MODO=SCAN 'ALTA'
7840 MOVE EDITED VISITA.VAC-FEC-VISITA(EM=YYYYMMDD) TO #FECHA-A
7850 IF NOT(#FECHA-N = 18000101 THRU 23001231 OR
7860 #FECHA-N = MASK (YYYYMMDD))
7870 REINPUT FULL 'FECHA ERRONEA' MARK *VISITA.VAC-FEC-VISITA
7880 END-IF
7890 ** WHEN VISITA.VAC-FEC-COMP = 0 AND #MODO=SCAN 'ALTA'
7900 * WHEN VISITA.VAC-CVE-PROY = ' '
7910 * REINPUT FULL 'CLAVE PROYECTO REQUERIDO' MARK *VISITA.VAC-CVE-PROY
7920 WHEN VISITA.VAC-CVE-PROY NE ' '
7930 CALLNAT 'APANL170' VISITA.VAC-CVE-CLIENTE
7940 VISITA.VAC-CVE-PROY #DPRY #ISN-T #SW
7950 IF NOT #SW
7960 REINPUT FULL 'CLAVE PROYECTO ERRONEO' MARK *VISITA.VAC-CVE-PROY
7970 END-IF
7980 WHEN VISITA.VAC-TIPO-REUNION = ' '
7990 REINPUT FULL 'TIPO DE REUNION REQUERIDO'
8000 MARK *VISITA.VAC-TIPO-REUNION
8010 WHEN VISITA.VAC-TIPO-REUNION NE ' '
8020 MOVE VISITA.VAC-TIPO-REUNION TO #LLA-TIPREU
8030 CALLNAT 'APANL001' 'AEP-TIPREU' #LLAVE #DATOS #SW
8040 IF NOT #SW
8050 REINPUT FULL 'CLAVE DE TIPO DE REUNION ERRONEA'
8060 MARK *VISITA.VAC-TIPO-REUNION
8070 END-IF
8080 WHEN VISITA.VAC-CONSULTOR = 0
8090 REINPUT FULL 'CLAVE CONSULTOR REQUERIDO'
8100 MARK *VISITA.VAC-CONSULTOR
8110 WHEN VISITA.VAC-CONSULTOR NE 0
8120 CALLNAT 'APANL130' VISITA.VAC-CONSULTOR #NOM-CONSU #ISN-T #SW
8130 IF NOT #SW
8140 REINPUT FULL 'CLAVE CONSULTOR ERRONEO'
8150 MARK *VISITA.VAC-CVE-PROY
8160 END-IF
8170 WHEN VISITA.VAC-FEC-COMP NE 0 /* AND #MODO=SCAN 'ALTA'
8180 MOVE EDITED VISITA.VAC-FEC-COMP(EM=YYYYMMDD) TO #FECHA-A
8190 IF NOT(#FECHA-N = 18000101 THRU 23001231 OR
8200 #FECHA-N = MASK (YYYYMMDD))
8210 REINPUT FULL 'FECHA ERRONEA' MARK *VISITA.VAC-FEC-COMP
8220 END-IF
8230 IF VISITA.VAC-FEC-VISITA LE VISITA.VAC-FEC-COMP
8240 THEN
8250 IGNORE
8260 ELSE
8270 REINPUT FULL 'LA FECHA DEBE SER MAYOR QUE FECHA VISITA'
8280 MARK *VISITA.VAC-FEC-COMP
8290 END-IF
8300 WHEN NONE IGNOPE
8310 END-DECIDE
8320 END-IF
```


11:10

APBPL210 [ADMPROY:Program]

03/04/1995

```
8330 FORMAT KD=ON
8340 END-SUBROUTINE
8350 **-----
8360 DEFINE SUBROUTINE LEE-INFORMACION
8370 RESET #DPRY
8380   CALLNAT 'APANL170' VISITA.VAC-CVE-CLIENTE
8390         VISITA.VAC-CVE-PROY #DPRY #ISN-T #SW
8400   CALLNAT 'APANL130' VISITA.VAC-CONSULTOR #NOM-CONSU #ISN-T #SW
8410   MOVE VISITA.VAC-TIPO-REUNION TO #LLA-TIPREU
8420   CALLNAT 'APANL001' 'AEP-TIPREU' #LLAVE #DATOS #SW
8430   MOVE #DAT-TIPREU TO #D-TIPREU
8440   COMPUTE #DIAS = (VISITA.VAC-FEC-COMP - *DATX)
8450   IF #DIAS LT 0
8460     MOVE 0 TO #DIAS
8470   END-IF
8480 END-SUBROUTINE
8490 **-----
8500 **SAG END-EXIT
8510 END-REPEAT /* PROG.
8520 PERFORM FINAL-PROCESSING
8530 END
```

APENDICE

**MANUAL DE USUARIOS DEL
SISTEMA EJECUTIVO DE ADMINISTRACIÓN DE PROYECTOS**

CONTENIDO

- I INTRODUCCION**
- II. DESCRIPCION GENERAL**
- III. ACCESO AL SISTEMA**
- IV. CARACTERISTICAS**
- VI. ESTRUCTURA**
- VII. DESCRIPCION FUNCIONAL**
- VIII. DESCRIPCION DE PROCEDIMIENTOS**

MANUAL DE USUARIOS DEL SISTEMA EJECUTIVO DE ADMINISTRACIÓN DE PROYECTOS

I. INTRODUCCION

El Sistema Ejecutivo de Administración de Proyectos (SEAP), es un sistema mediante el cual el personal de la empresa podrá conocer de forma integral el trabajo que las diferentes áreas realizan sobre una cuenta determinada.

La consulta o actualización de la información se lleva a cabo por medio de categorías de acceso, cada categoría es evaluada y asignada por el administrador del sistema.

El sistema lo integran dos grandes módulos, y a cada uno de ellos lo integran varios procedimientos, en los que podrá consultar o actualizar la información.

En este documento se especifica ampliamente, el procedimiento de acceso al sistema y en forma muy general, la descripción de los módulos y procedimientos que lo integran, también se mencionan las teclas de funciones y las áreas de las pantallas de proceso y de menú.

De esta manera se pretende dar al usuario una visión general del SEAP.

II. ACCESO AL SISTEMA

Al acceder el sistema se presentará la siguiente pantalla:

Sistema Dinámico
Software AG de México, S.A. de C.V.

NOTA
POR FAVOR CONSULTAR PROCEDIMIENTOS DE ALTA DE USUARIOS O
CONSULTAR PROCEDIMIENTOS RELACIONADOS CON USUARIOS
PARA MÁS CONSULTAS CON EL ADMINISTRADOR
DEL SISTEMA.

CLAVE: _____ PASSWORD: _____ NUEVO PASSWORD: _____

Fig. 1

En esta pantalla deberá digitar la clave de usuario y el password requerido, ambas deberán ser asignadas por el administrador del sistema.

Puede ser que el usuario se encuentre firmado es decir, que por algún motivo no se haya ejecutado la correcta salida del sistema, las causas pueden ser: que falle la energía eléctrica o que se trabe la máquina, para solucionarlo pida ayuda al Administrador del sistema.

Una vez ejecutado lo anterior, el sistema desplegará el menú principal del Sistema Ejecutivo de Administración de Proyectos.

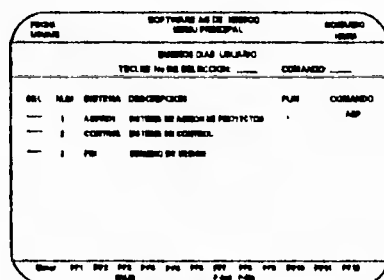


Fig. 2

III. CARACTERISTICAS

El SEAP es un sistema muy dinámico que permite:

- Tener un alto nivel de seguridad en el acceso de la información.
- Información disponible para ser consultada en cualquier momento y a cualquier nivel.
- La navegación a través de sus pantallas es muy sencilla, ya que cuenta con áreas fácilmente identificables.

IV. ESTRUCTURA

El sistema cuenta con dos módulos y cada uno de ellos con submódulos, los cuales están compuestos por varios procedimientos.

Los procedimientos a su vez, cuentan con cuatro acciones que en su conjunto permiten dar mantenimiento a la información.

Los módulos y procedimientos que integran al sistema son:

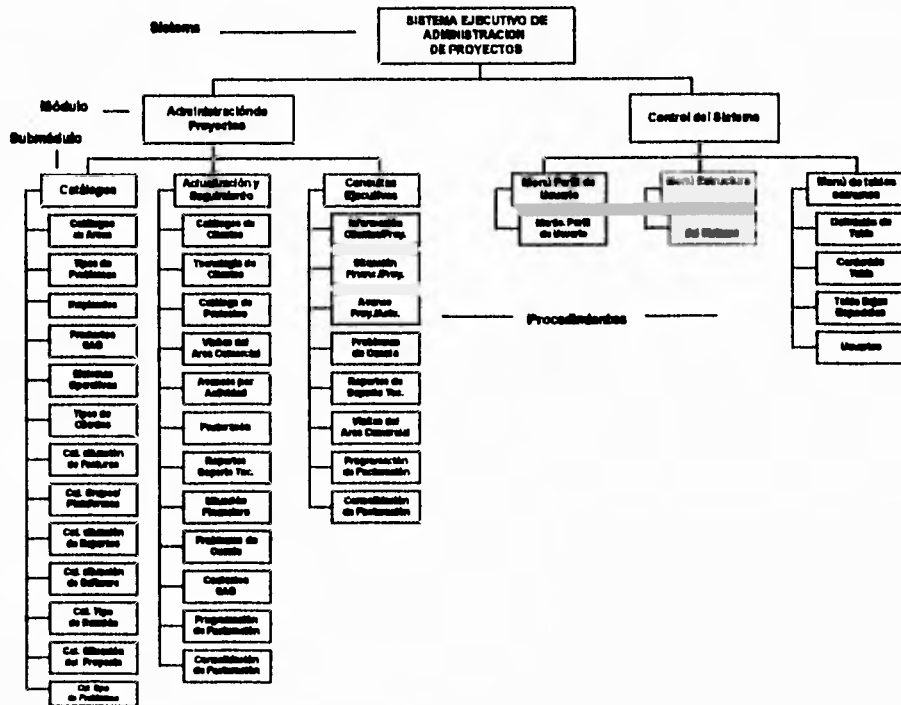


Fig. 3

1. ADMINISTRACION DE PROYECTOS

Este módulo nos da una visión integral del trabajo que realizan las diferentes áreas de la empresa en una cuenta, al permitirnos actualizar la información referente a la labor comercial, el avance de los proyectos y la situación financiera de los clientes.

A continuación se describen cada uno de ellos:

1.1. CATALOGOS

Aquí se registra y actualiza la información de los catálogos del sistema, observe en la estructura del sistema, los procedimientos que integran este submódulo. En este submódulo, es el que proporcionará la información necesaria en los desplegados de ayuda.

1.2. ACTUALIZACION Y SEGUIMIENTO

En este submódulo se le da seguimiento y actualización a los clientes y a los proyectos, se registran sus datos generales, la tecnología con que cuenta, las visitas que realiza el área comercial, los proyectos que tienen asociados, el avance de dichos proyectos, la situación financiera de la cuenta, los principales problemas tecnológicos que han tenido, el personal de la empresa que atiende la cuenta, el calendario de facturación programado en el contrato y la consolidación de la facturación.

1.3. CONSULTAS EJECUTIVAS

Aquí, el personal ejecutivo puede hacer consultas necesarias para tener una visión clara e integral del proyecto de cada cliente, considera la situación financiera, el avance del proyecto y las actividades, los problemas que tienen diferentes cuentas, las visitas que realiza el área comercial, la facturación y la cartera vencida

2. CONTROL DEL SISTEMA

En este módulo se podrá realizar el mantenimiento a: el perfil del usuario, a la estructura del sistema y a las tablas comunes, considerando para estas últimas: la definición, contenido y bajas especiales.

1. ADMINISTRACION DE PROYECTOS

Este módulo nos da una visión integral del trabajo que realizan las diferentes áreas de la empresa en una cuenta, al permitirnos actualizar la información referente a la labor comercial, el avance de los proyectos y la situación financiera de los clientes.

A continuación se describen cada uno de ellos:

1.1. CATALOGOS

Aquí se registra y actualiza la información de los catálogos del sistema, observe en la estructura del sistema, los procedimientos que integran este submódulo.

En este submódulo, es el que proporcionará la información necesaria en los desplegados de ayuda.

1.2. ACTUALIZACION Y SEGUIMIENTO

En este submódulo se le da seguimiento y actualización a los clientes y a los proyectos, se registran sus datos generales, la tecnología con que cuenta, las visitas que realiza el área comercial, los proyectos que tienen asociados, el avance de dichos proyectos, la situación financiera de la cuenta, los principales problemas tecnológicos que han tenido, el personal de la empresa que atiende la cuenta, el calendario de facturación programado en el contrato y la consolidación de la facturación.

1.3. CONSULTAS EJECUTIVAS

Aquí, el personal ejecutivo puede hacer consultas necesarias para tener una visión clara e integral del proyecto de cada cliente, considera la situación financiera, el avance del proyecto y las actividades, los problemas que tienen diferentes cuentas, las visitas que realiza el área comercial, la facturación y la cartera vencida

2. CONTROL DEL SISTEMA

En este módulo se podrá realizar el mantenimiento a: el perfil del usuario, a la estructura del sistema y a las tablas comunes, considerando para estas últimas: la definición, contenido y bajas especiales.

V. DESCRIPCION FUNCIONAL

El sistema cuenta con dos tipos de pantallas: la pantalla de menú, en donde se selecciona el procedimiento que se desea ejecutar y la pantalla de procedimiento, en la que se realizan las diferentes acciones (Altas, Bajas, Cambios y Modificaciones). En algunas pantallas no aparecerá ninguna acción especificada, esto se debe a que sólo se permite la consulta.

SISTEMA DE SELECCION DE PROCEDIMIENTOS				
MENU DE SELECCION				
NO.	DESCRIPCION	FUNCION	OPCION	ACCION
1	Consultas	+	Consultas	
2	Actualización y Borrado	+	Bajas	
3	Consultas Especiales	+	Cambios	

Botón PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12

Acción: Consultas Bajas Cambios

Pantalla de Menú

Fig. 4

SISTEMA DE SELECCION DE PROCEDIMIENTOS				
MENU DE SELECCION				
NO.	DESCRIPCION	FUNCION	OPCION	ACCION
1	Consultas	+	Consultas	
2	Actualización y Borrado	+	Bajas	
3	Consultas Especiales	+	Cambios	

Botón PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12

Acción: Consultas Bajas Cambios

Pantalla de Procedimiento

Fig. 5

Para hacer más fácil el manejo del sistema, se han implementado 9 funciones identificadas como PF, estas se ubican en la parte inferior de la pantalla, abajo de cada PF se encontrará una leyenda que especifica su uso.

Las PF's que considera el sistema son:

- PF1 Ayuda para ese campo
- PF2 Regresar a la opción anterior
- PF3 Salir del Sistema
- PF4 Cancelar la función ejecutada
- PF7 Acceso a la pantalla anterior
- PF8 Acceso de la pantalla posterior
- PF10 Acceso de la pantalla izquierda
- PF11 Acceso de la pantalla derecha
- PF12 Menú principal

En cada procedimiento se visualiza sólo las funciones que ese procedimiento necesite.

Pueden considerarse como funciones fijas: PF2, PF3 y PF12.

Una vez identificadas las pantallas de menú y de procedimientos y las funciones del sistema, describiremos el flujo de información para un procedimiento particular.

Consideremos el procedimiento "Tecnología de Clientes" para ejemplificar el funcionamiento del sistema.

TECNOLOGÍA DE CLIENTES

Este procedimiento permite registrar y darle mantenimiento a la información de la plataforma tecnológica que tiene cada cliente. Considera por cada equipo una descripción general, la plataforma, el sistema operativo y los accesorios que posee, así como la especificación de su ubicación física.

El acceso de este procedimiento se ejecutará al seleccionar el número 2 del menú de Actualización y Seguimiento.

SOFTWARE AG DE SERVICIO
Actualización y Seguimiento

USUARIO: _____

TECLAS No DE SELECCION: _____ COMANDO: _____

DEL	NUM	DESCRIPCION	FUN	COMANDO
---	1	CATALOGO DE CLIENTES	.	CLIENTES
---	2	TECNOLOGIA DE CLIENTES	.	SEMPRO
---	3	CATALOGO DE PROYECTOS	.	PROYECTOS
---	4	VISITAS DEL AREA CORPORATIVA	.	AVANCIOS
---	5	AVANCIOS POR ACTIVIDAD	.	FACTURAS
---	6	FACTURACION	.	BOFFEC
---	7	REPORTE'S REPORTE TECNICO	.	PROBLEMAS
---	8	BITUN CON PRENSION	.	CONTACTOS
---	9	PROBLEMAS DE CLIENTA	.	PROFAC
---	10	CONTACTOS DE ID	.	CONFAC
---	11	PROCESAMIENTO DE FACTURACION	.	CONFAC
---	12	CONSEJERIA DE FACTURACION	.	CONFAC

Botón: P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22
Ayuda: Ayuda: F10 F11 F12

Fig. 6

Al seleccionar el procedimiento, el sistema desplegará su respectiva pantalla:

SERVICIO AL CLIENTE DE ADMINISTRACION DE PROYECTOS

AYUDA AL CLIENTE

CLAVE: _____

CLIENTE: _____

Botón: P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22
Ayuda: Ayuda: F10 F11 F12

Fig. 7

Para obtener información, digite en el campo cliente la clave del cliente que desea consultar.

El campo cliente, es un campo con ayuda por lo tanto, tiene que estar previamente dado de alta en el catálogo de clientes del Submódulo de Actualización y Seguimiento.

Para obtener ayuda, pulse la tecla de función PF1 y obtendrá un cuadro con ayuda para que seleccione de ahí, la clave del cliente requerido. Para salir del cuadro pulse PF2 y regresará a la pantalla de procedimiento.

Las acciones ejecutables para este procedimiento son: Altas, Bajas, Consultas y Modificaciones.

Altas. Posiciónese en el campo de selección, digite <A> y después pulse <Enter>.

El sistema desplegará una segunda pantalla, en la que podrá registrar información para todos los campos.

En esta acción los campos requeridos son:

- No. equipo de cómputo
- *Plataforma
- Descripción
- *Sistema operativo

Los campos plataforma y sistema operativo son campos con ayuda, por lo tanto deben registrarse primero en el módulo de catálogos.

Al registra los datos en los campos en listados, confirme la información en el sistema pulsando <Enter> o <PF4> si desea cancelar esta acción.

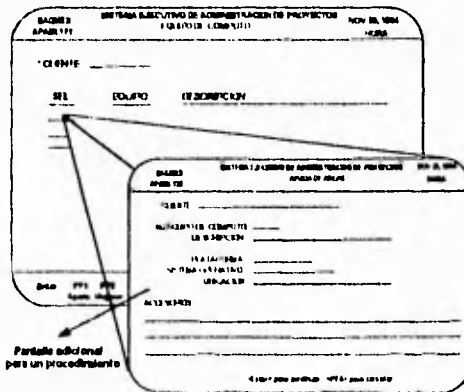


Fig. 8

Descripción de Campos:

No. equipo de cómputo: Número secuencial que se incrementa en una unidad para cada nuevo registro, permite enunciar los tipos de equipo que tiene un cliente.

Descripción: Indica el nombre, modelo y marca del equipo en cuestión.

Plataforma: Indica el tipo de ambiente al que pertenece.

Sistema operativo: Indica el sistema operativo que tiene instalado.

Ubicación: Señala la dirección del lugar donde está físicamente el equipo.

Accesorios: Indica los periféricos que tiene el equipo y su número.

Baja Para realizar la baja de un equipo específico, posicione en el campo de selección que le corresponda, digite y después pulse <Enter>.
El sistema desplegará la información y le indicará que confirme la baja del registro pulsando <Enter>.
Podrá cancelar esta acción pulsando <PF4>.

Consultas Seleccione el equipo que desea modificar, posicionándose en el campo de selección que corresponda, digite <C> y después pulse <Enter>.
El sistema desplegará toda la información para ser consultada si se desea salir de esta acción pulse <PF2>.

Modificaciones: Seleccione el equipo que desea modificar, posicionándose en el campo de selección que corresponda, digite <M> y después pulse <Enter>.
El sistema desplegará toda la información para que sea actualizada.
Para efectuar una modificación, posicione en el campo que desea actualizar y digite los cambios.
Al terminar, confirme la información en el sistema.

VI. DESCRIPCION DE PROCEDIMIENTOS

En esta sección se describe ampliamente el primer módulo, "Administración de Proyectos", considerando los tres submódulos que lo integran.

- Catálogos

- Actualización y seguimiento
- Consultas ejecutivas

Para el submódulo "Catálogos", ejemplificaremos sólo un procedimiento, ya que el funcionamiento es el mismo para todos los demás.

Para el segundo submódulo, "Actualización y Seguimiento" se describirán todos los procedimientos, ya que es aquí donde se realiza el mantenimiento de toda la información registrada en el sistema.

En la descripción de los procedimientos, se considerarán las diferentes acciones ejecutables y los campos requeridos, así como una breve explicación para aquellos campos que se consideren necesarios.

Para el submódulo "Consultas Ejecutivas" se mencionarán para cada procedimiento las diferentes formas de consulta y los campos requeridos para cada uno de estos.

El registro de información inicial, se lleva a cabo en el submódulo "Catálogos", ya que es éste el que generará los desplegados de ayuda y con ello la información necesaria para algunos campos requeridos.

1. SUBMODULO CATÁLOGOS

En este submódulo se debe registrar la información inicial, para llevar a cabo un procedimiento.

Como se mencionó en páginas anteriores, se describirá sólo un procedimiento, el de Empleados para ejemplificar todos los demás.

Empleados

Este procedimiento permite registrar a todos los empleados, considera información básica para la identificación del empleado dentro de la empresa.

Las acciones para este procedimiento son Altas, Bajas, Consultas y Modificaciones.

El acceso de este procedimiento presentará la siguiente pantalla:

```

SISTEMA SUBMÓDULO DE ADMINISTRACIÓN DE PROYECTOS
CATALOGO DE EMPLEADOS
NOV 11, 1994
HORA

ACCION  (A,B,C,L,M,E)

CLAVE DEL EMPLEADO  _____

NOMBRE DEL EMPLEADO  _____

CLAVE DEL AREA      _____

PUESTO:  _____

-----
Salir  001  002  003  004  005  006  007  008  009  010  011  012
Ayuda  Regreso  Salir  Salir  Salir  Salir  Salir  Salir  Salir  Salir  Salir  Salir

```

Fig. 9

Para la acción de altas, los campos requeridos son: clave del empleado, nombre del empleado, clave del área y puesto.

El campo clave del área es un campo con ayuda, por lo tanto deberá registrarse antes la información necesaria para dar de alta a un empleado.

Para las acciones: bajas, consultas y modificaciones el campo requerido es clave del empleado.

2. SUBMÓDULO "ACTUALIZACION Y SEGUIMIENTO"

Este submódulo permite actualizar y dar seguimiento a la información de clientes, proyectos, visitas realizadas por el área comercial, información de la situación financiera de un proyecto y la programación y consolidación de la facturación.

El acceso de este submódulo se lleva a cabo digitando el número 2 en el campo de selección del menú del módulo Administración de Proyectos.

A continuación se describen los procedimientos que integran este submódulo.

2.1 CATALOGO DE CLIENTES

Este procedimiento permite actualizar la información de los clientes, considerando la clave, razón social, dirección, R.F.C, e información del personal directivo, así como la importancia estratégica, grupo financiero al que pertenece, importancia dentro del sector y tipo de cliente.

El acceso de este procedimiento presentará las siguientes pantallas:

Pantalla de inicio

SISTEMA SUBSISTEMO DE ADMINISTRACION DE PROYECTOS
ACTUALIZACION DE CLIENTES

ACCION: (A,B,C,M,S)

CLAVE DEL CLIENTE: _____

RAZON SOCIAL: _____

DIRECCION: _____

R.F.C.: _____

INFORMACION DE DIRECTIVOS

NOBRE	APELLIDO	TELEFONOS
_____	_____	_____
_____	_____	_____
_____	_____	_____

Inicio P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22
Ayuda Inicio Salir P-1 P-2 P-3 P-4 P-5 P-6

Pantalla siguiente

SISTEMA SUBSISTEMO DE ADMINISTRACION DE PROYECTOS
ACTUALIZACION DE CLIENTES

IMPORTANCIA ESTRATEGICA: _____

GRUPO FINANCIERO: _____

IMPORTANCIA DENTRO DEL SECTOR: _____

TIPO DE CLIENTE: _____

CLAVE PARA SU TELEFONO: _____

DE REESTRUCTURACION FINANCIACIONES: _____

Inicio P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22
Ayuda Inicio Salir P-1 P-2 P-3 P-4 P-5 P-6

Para acceder a pantalla siguiente digite *1911*

Fig. 10

Los campos requeridos para dar de alta un registro son:

- Clave del cliente
- Razón social, y
- Tipo de cliente

Para ejecutar esta acción, el campo Clave y Tipo de cliente deben registrarse previamente en el submódulo de catálogos.

Para las acciones: bajas, cambios y modificaciones, el campo requerido es Clave del Cliente.

Descripción de campos:

Importancia estratégica: Indica el grado de importancia que tiene la cuenta para la empresa, puede ser: 1=Muy importante, 2=Importante, 3=Medio y 4=Poco importante.

Grupo de negocio: Deberá especificar el nombre del grupo de negocio o corporación a la que pertenece el cliente.

Importancia dentro del sector: Es la posición de la empresa en el mercado de su ramo (primero, segundo, etc).

Comentarios Tecnológicos: Este es un campo abierto, por lo tanto, será necesario capturar la información específica que describa la situación tecnológica actual de la empresa (e.i. Tecnología obsoleta, interés de ir hacia tecnología Unix, etc.)

Infraestructura de comunicaciones: Este también es un campo abierto, por ello, su especificación deberá indicar cuál es el esquema de comunicación hacia sus filiales y oficinas, si existieran.

2.2 TECNOLOGIA DE CLIENTES

Este procedimiento permite registrar y darle mantenimiento a la información de la plataforma tecnológica que tiene cada cliente. Considera por cada equipo una descripción general, la plataforma, el sistema operativo y los accesorios que posee, así como la especificación de su ubicación física.

Para obtener información, digite en el campo cliente, la clave del cliente que desea consultar y después pulse <Enter>.

El sistema presentará una lista de todo el equipo registrado para ese cliente.

Para consultar información de un equipo específico, seleccione el número del equipo que desea consultar y digite <S> en el campo de selección (sel) y después pulse <Enter>.

El sistema desplegará una segunda pantalla en la que podrá consultar la información registrada para ese equipo.

Para las acciones bajas, consultas y modificaciones el campo requerido es clave del cliente.

2.3 CATALOGO DE PROYECTOS

Este procedimiento, permite registrar los proyectos que tienen nuestros clientes. Se registrará información como: área responsable, monto del proyecto, tipo de moneda, fecha de inicio y termino, así como responsable comercial y del proyecto. Además se podrán registrar comentarios respecto a la importancia del proyecto, factores críticos de éxito y los supuestos críticos.

El acceso de este procedimiento presentará las siguientes pantallas:

Pantalla de inicio

VENTANA SUBCRIPTIVO DE ADMINISTRACION DE PROYECTOS
CATALOGO DE PROYECTOS

ACCION: _____ (A,B,C,I,M,U)

CLAVE DEL CLIENTE: _____

CLAVE DEL PROYECTO: _____

AREA RESPONSABLE: _____

TIPO MONEDA: _____ (D) CLAVES (P) LEJOS

MONTO DEL PROYECTO: _____

FECHA INICIO: _____ (AA-MM-DD)

FECHA TERMINACION: _____ (AA-MM-DD)

REC. NUMERO ASESINADOR: _____

RESP. PROYECTO: _____

RESP. COMERCIAL: _____

FECHA DE SITUACION FINANCIERA: _____

EST. AVANCE AL: _____ (AA-MM-DD)

% AVANCE PLAN: _____

% AVANCE REAL: _____

RETRABO: _____ (DMS)

Boton: P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12
Ayuda Ho.yema Bot P.Ant P.Sig P.Clar P.Ant P.Sig

Pantalla izquierda

VENTANA SUBCRIPTIVO DE ADMINISTRACION DE PROYECTOS
CATALOGO DE PROYECTOS

IMPORTANCIA DEL PROYECTO

FACTORES CRITICOS DE EXITO

SUPOSITOS CRITICOS

Boton: P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12
Ayuda Ho.yema Bot P.Ant P.Sig P.Clar P.Ant P.Sig

Fig. 11

Los campos requeridos para dar de alta un proyecto son:

- Clave del cliente
- Nombre del Proyecto
- Area responsable
- Tipo de moneda
- Fecha de inicio (AA-MM-DD)
- Fecha de terminación (AA-MM-DD)
- Status
- Responsable del Proyecto
- Responsable comercial, y
- No. de contrato

Para ejecutar esta acción, los campos Clave del cliente, área responsable, estado, responsable del proyecto y responsable comercial son campos con ayuda por lo tanto, deberán registrarse previamente en el submódulo de catálogos.

El campo status se registra en el catálogo "Situación del Proyecto".

Para las acciones: bajas, cambios y modificaciones, los campos requeridos son: Clave del Cliente y Clave del Proyecto.

Descripción de campos:

Fecha de situación financiera: Esta fecha se actualiza automáticamente, al ejecutar el proceso de consolidación.

Importancia de proyecto: Es la importancia que el proyecto tiene para la empresa.

Factores críticos de éxito: Son los aspectos internos o externos relacionados con el proyecto, que tienen gran influencia sobre su cumplimiento y son medibles (e.i. contar con personal capacitado, etc).

Supuestos críticos: Son los factores externos a la organización que afectan al proyecto, no se pueden controlar y para los cuales hay que estar preparado (e.i. Devaluaciones, Cambio de legislaciones, etc).

2.4 VISITAS DEL AREA COMERCIAL

Este procedimiento permite registrar las visitas realizadas por el área comercial a nuestros clientes, se considera información de: fecha de visita, clave del proyecto, tipo de reunión y consultor.

El acceso de este procedimiento desplegará la siguiente pantalla:

El diagrama muestra dos pantallas de una interfaz de usuario. La pantalla superior, titulada 'SISTEMA SUBSISTIVO DE ADMINISTRACION DE PROYECTOS VISITAS DEL AREA COMERCIAL', tiene un campo 'CLIENTE' y una tabla con las siguientes columnas: 'ACCION', 'FECHA', 'PUNTOS TRATADOS' y 'CONSULTOR'. La columna 'ACCION' está desplegada, mostrando una segunda pantalla con los siguientes campos: 'CLIENTE', 'FECHA VISITA', 'PROYECTO', 'TIPO DE REUNION', 'CONSULTOR', 'PUNTOS TRATADOS', 'FECHA DE COMPROMISO' y 'DIAS DISPONIBLES'. En la parte inferior de esta segunda pantalla, se indica 'El día para continuar: <PPA> para cancelar'.

Fig. 12

La consulta general de la información, se genera a través de la clave de un cliente; la actualización de esa información se registra con las acciones ejecutables para este procedimiento, (altas, bajas, consultas y modificaciones).

Para dar de alta un registro, posicione en el campo de acción y digite <A>, el sistema presentará una segunda pantalla, en la que registrará toda la información requerida.

Descripción de campos

Días disponibles: Son los días hábiles disponibles que hay de la fecha de visita a la fecha de compromiso.

Compromisos adquiridos: El sistema calculará automáticamente los días que hay, entre la fecha actual y la de compromiso.

Si la fecha de compromiso, es menor a la actual el sistema registrará cero.

2.5 AVANCES POR ACTIVIDAD

En este procedimiento se actualizará el avance de las diferentes macro actividades que conforman un proyecto.

El acceso de este procedimiento presentará la siguiente pantalla:

SISTEMA ELECTRONICO DE ADMINISTRACION DE PROYECTOS (ACTIVIDADES POR PROYECTO)

CLIENTE _____

PROYECTO _____

SER.	ACT.	DESCRIPCION	DURACION (DIAS)	PLAN.	REAL.	PLAN.	REAL.	OBSERV.

SISTEMA ELECTRONICO DE ADMINISTRACION DE PROYECTOS (ACTIVIDADES POR PROYECTO)

CLAVE DEL PLAN _____

CLAVE DEL PROYECTO _____

ACTIVIDAD _____

PLAN DE AVANCE _____

PLAN DE AVANCE _____

FECHA INICIO _____ FECHA FIN _____ DURACION (DIAS) _____ N. AVANCE _____ N. AVANCE _____ FECHA AVANCE _____

PLAN INICIO _____ PLAN FIN _____

NOTA _____

Fig. 13

La consulta general del proyecto por actividad, se realiza a través de la clave del cliente, y la especificación de la clave del proyecto.

Para ejecutar una acción, digite en el campo de selección el carácter que la identifica y después pulse <Enter>.

Para la acción Altas, los campos requeridos son: Actividad, fecha de inicio planeada, fecha final planeada y duración.

Si la acción es Cambios, Bajas o Modificaciones, bastará posicionarse en el campo de selección y digitar el carácter correspondiente a la acción a ejecutar (B=Bajas, C=Consultas y M=Modificaciones). El sistema desplegará toda la información para actualizarla.

Descripción de campos:

Actividad Es el número consecutivo de cada macro actividad que integra un plan de trabajo.

2.6 FACTURACIÓN

Este procedimiento permite actualizar la información de nuestros clientes.

SISTEMA SUBMÓDULO DE ADMINISTRACIÓN DE PROYECTOS
FACTURACIÓN DE CLIENTES

CLIENTE: _____
PROYECTO: _____

SELECCIÓN	ALTA	FACT	RECIBO	CONCEPTO	IMP. BRU	IMP. CUBO	SITUA
_____	_____	_____	_____	_____	_____	_____	_____

ALTA

CLAVE LEYENDA: _____
CLAVE PROYECTO: _____
CLAVE AREA: _____
FECHA DE EMISIÓN: (AA-MM-CC) _____
FECHA DE PAGO: (AA-MM-CC) _____
IMPORTE FACTURADO: _____
IMPORTE COBRADO: _____
SITUACIÓN: _____

© 1997 IBM CORP. TODOS LOS DERECHOS RESERVADOS

Fig. 14

La consulta general se realiza a través de la clave del cliente y del proyecto.

La ejecución de las diferentes acciones se lleva a cabo, digitando en el campo de selección el carácter que las identifica.

Si la acción es Altas, los campos requeridos son: clave área, fecha de emisión, fecha de pago, importe facturado e importe cobrado.

El campo clave área, es campo con ayuda, por lo tanto deberá ser capturado anteriormente en el submódulo Catálogos.

Para las acciones Consultas, Bajas y Modificaciones bastará digitar la acción solicitada en el campo de selección.

Descripción de campos

Situación: En caso de que el importe facturado este pendiente de facturar o facturado, Si el importe cobrado está por cobrar o cobrado.

2.7 REPORTE DE SOPORTE TÉCNICO

En este procedimiento se registrarán los reportes del área de soporte técnico.

El acceso de este procedimiento presentará la siguiente pantalla.

El diagrama muestra una interfaz de usuario con dos pantallas. La pantalla superior, titulada 'SISTEMA SUBMÓDULO DE ADMINISTRACIÓN DE PROYECTOS REPORTE DE SOPORTE TÉCNICO', contiene un campo 'CLIENTE' y una tabla con las siguientes columnas: 'ACCION', 'NUMERO REPORTE', 'FECHA RECEPCION' y 'DESCRIPCION DEL REPORTE'. La pantalla inferior, también titulada 'SISTEMA SUBMÓDULO DE ADMINISTRACIÓN DE PROYECTOS REPORTE DE SOPORTE TÉCNICO', muestra los siguientes campos: 'CLAVE PROYECTO', 'NUMERO REPORTE', 'SITUACION', 'CRITICO IN PORTANTE', 'FECHA', 'CADA DIA', 'RESPONSABLES', 'RECEPCION ATENCION' y 'LA RECEPCION DEL REPORTE'. En la parte inferior de esta pantalla se encuentran los botones 'ACCIONES TOMADAS' y 'RECEPCION ATENCION'.

Fig. 15

La consulta general de los diversos reportes se realiza al digitar la clave del cliente en el campo requerido.

El sistema presentará una lista (si existiera) de los reportes, considerando información tal como: número de reporte, fecha de recepción y descripción del reporte.

Para ejecutar cualquier acción posicione en el campo de acción correspondiente y digite el carácter que la identifica.

El sistema desplegará una segunda pantalla en la que podrá realizar la acción (A,B,C,M) correspondiente.

Para la acción Altas, los campos requeridos son: Clave proyecto, número de reporte, situación, recepción y responsables.

Los campos situación y responsables son campos con ayuda, por ello deberán registrarse primero en el submódulo de catálogos.

Descripción de campos:

Acciones tomadas: Es la respuesta de nuestros consultores, a los posibles problemas que se presenten con el cliente, los cuales deben ser enunciados en base a acciones tomadas.

2.8 SITUACION FINANCIERA

En este procedimiento sólo se podrán realizar consultas o modificaciones a la información financiera de un cliente.

El acceso de este procedimiento presentará la siguiente pantalla:

SERVICIO DE ADMINISTRACION DE PROYECTOS
SITUACION FINANCIERA

ACCION: _____ (A,B,C,L,M,R)
CLAVE DEL CLIENTE: _____
CLAVE PROYECTO: _____
PORCENTAJE DE AVANCE: _____ DÍAS DE RETARDO: _____

GASTOS

SITUACION FINANCIERA	CONCEPTO	IMPORTE
FACTURADO	_____	_____
PAGADO	_____	_____

Botón: P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12
Ayuda Punt Esc F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 Fin

Fig. 16

Los datos que aquí se presentan, dependerán de la información registrada y actualizada en el procedimiento: Catálogo Situación Financiera. De tal manera que al ser modificados los datos en el procedimiento correspondiente se actualizará inmediatamente la información.

2.9 PROBLEMAS DE CUENTA

En este procedimiento se registrarán los diferentes problemas a los que enfrenta la empresa.

El acceso de este procedimiento presentará la siguiente pantalla:

Fig. 17

La información general se accesa a través de la clave de un cliente.

Para realizar cualquier acción, posicione en el campo de acción y digite el carácter que identifica.

Para la acción Altas, los campos requeridos son: Proyecto, tipo problema, status y fecha del problema.

Los campos proyecto, tipo de problema y status son campos con ayuda.

Descripción de campos:

Causa: En este campo se deberá especificar la causa que está originando el problema que se está enfrentando.

Solución: Describe la acción tomada para resolver un problema.

2.10 CONTACTOS DE LA EMPRESA

En este procedimiento se registrarán los contactos que la empresa tiene con sus clientes.

El acceso de este procedimiento presentará la siguiente pantalla:

SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS
CONTACTOS USU

CLIENTE: _____

ACCION	CLAVE CONTACTO	NOMBRE EMPLEADO	PROYECTO	FUNCION QUE REALIZA
ALTA	_____	_____	_____	_____
BAJA	_____	_____	_____	_____
MODIFICAR	_____	_____	_____	_____

ACCION

CLAVE: _____

CLAVE EMPLEADO: _____

PROYECTO: _____

FUNCION QUE REALIZA: _____

RESPONSABLE DE LA CUENTA: _____

Fig. 17

Al igual que los anteriores procedimientos, la información general de éste, se realiza a través de la clave del cliente.

Para actualizar o registrar la información, ejecute las diferentes acciones.

Para la acción altas los campos requeridos son: clave empleado, proyecto, función que realiza y responsable de la cuenta.

Descripción de campos.

Clave contacto: Semanejan también como clave del empleado.

Función que realiza: Es la función específica que realiza el contacto en ese proyecto, que puede ser contacto comercial, responsable de proyecto, etc.

2.11 PROGRAMACIÓN DE FACTURACIÓN

En este procedimiento se registra el calendario programado de la facturación asociada a un proyecto según su contrato.

El acceso de este procedimiento presentará la siguiente pantalla:

El diagrama muestra dos pantallas de un sistema de gestión de proyectos. La pantalla superior, titulada 'SISTEMA EJECUTIVO DE ADMINISTRACIÓN DE PROYECTOS PROGRAMACIÓN DE LA FACTURACIÓN', contiene un formulario con los campos 'CLIENTE' y 'PROYECTO'. Debajo de estos campos hay una tabla con las siguientes columnas: 'SEL.', 'NO. PAGO', 'MES PROGRAMADO', 'CONCEPTO', 'MONTO' y 'FACTURA'. La pantalla inferior, también con el mismo título, muestra los detalles de un pago con los campos: 'CLIENTE', 'PROYECTO', 'NO. PAGO', 'MES PROGRAMADO', 'CONCEPTO', 'MONTO', 'FACTURA ASOCIADA', 'COMERCIO PARA EL COTIZADO PAGO' y 'DETALLE'. En la parte inferior de esta pantalla se indica la fecha '14 SEP 2013 09:58:07' y el usuario '001110011001'.

Fig. 18

La información general se accedera a través de la clave del cliente y de proyecto.

Para realizar una acción, posicione en el campo de selección y digite la acción correspondiente.

Para la acción de altas, los campos requeridos son: número de pago, mes programado, concepto e importe.

Para las acciones B, C, y M, bastará digitar, en el campo de selección el caracter que las identifica.

Descripción de campos:

Factura asociada: Se refiere a la existencia de una concentración superior de los conceptos se que están facturando. (Es el caso que una factura que ampare diferentes productos o servicios).

Condición para efecto de pago: Generalmente las condiciones de pago están asociadas a un evento señalado en el contrato. Consulte el contrato para registrar las condiciones en el sistema.

2.12 CONSOLIDACION DE FACTURACIÓN

En este procedimiento se consolida la facturación.

El acceso de este procedimiento presentara la siguiente pantalla:

* CLIENTE _____ FECHA _____
* PROYECTO _____

SE REALIZA EL PROCESO (SI): _____

Fig. 19

Consolidar implica sumar todas las facturas relacionadas con un cliente o un proyecto determinado.

Esta información se refleja en los procedimientos de Situación financiera y Monto facturado.

3. SUBMÓDULO “CONSULTAS EJECUTIVAS”

Este submódulo permite consultar la información de clientes, avances de proyectos, visitas realizadas por el área comercial, información de la situación financiera de un proyecto y la programación de la facturación.

El acceso de este submódulo se lleva a cabo digitando el número 3 en el campo de selección del menú del Administrador de Proyectos.

A continuación se describen los procedimientos que integran este submódulo.

3.1 INFORMACION CLIENTES/PROYECTO

Este procedimiento permite consultar la información general de nuestro cliente y la situación de su proyecto.

El acceso de este procedimiento presentará la siguiente pantalla:

Fig. 20

Los dos parámetros utilizados para la consulta son C y S que indican la consulta de los Datos del Cliente y los proyectos del mismo respectivamente.

El acceso de cada una se especifica a continuación.

Para consultar los datos generales del cliente, digite <C> en el campo de acción y después pulse <Enter>. De esta manera, accederá información general del cliente.

Fig. 21

Si la consulta se hace a través de <S> "Proyectos del Cliente", el sistema presentará la siguiente pantalla, en la que podrá consultar información del proyecto. Para obtener información específica de un proyecto, digite <C> en el campo de selección, y después pulse <Enter>, el sistema desplegará una segunda pantalla en la que podrá consultar el detalle de la información de los proyectos que tiene asociados.

CLIENTE	PROYECTO	DESCRIPCION	STATUS

Fig. 22

3.2 SITUACION FINANCIERA/PROYECTO

Este procedimiento permite consultar la situación financiera de un proyecto específico.

El acceso de este procedimiento presentará las siguientes pantallas.

ACCION	PROYECTO	NOMBRE DEL PROYECTO	SE IN	TOTAL	FACTURADO	COBRADO

CLIENTE	
PROYECTO	
SE IN	
COBRADO	
TOTAL DE INGRESOS	
TOTAL DE GASTOS	
TOTAL DE RESERVA	

Fig. 23

La consulta general se obtiene digitando la clave del cliente en el campo requerido, obtendrá información de los proyectos, del monto total facturado y cobrado.

Para consultar la situación financiera de un proyecto, posicione en el campo de acción y digite <C>, el sistema presentará la pantalla correspondiente. A través de esta última pantalla se puede consultar la facturación presionando <PF7>.

3.3 AVANCE DE ACTIVIDADES/PROYECTO

La consulta de avance de un proyecto se lleva a cabo en este procedimiento.

El acceso de este procedimiento presentará la siguiente pantalla:

Fig. 24

Digite la clave del cliente en el campo requerido, el sistema desplegará una lista de todos sus proyectos.

Para consultar un proyecto, posicione en el campo, seleccione y digite <C>, el sistema desplegará una segunda pantalla que le permitirá consultar la información general del proyecto.

Fig. 25

Para obtener información específica de cada macro actividad, digite <C> en la columna de selección y después pulse <Enter>.

3.4 PROBLEMAS DE CUENTA

Este procedimiento consultar los problemas que existen en cada cuenta.

El acceso de este procedimiento presentará las siguientes pantallas:

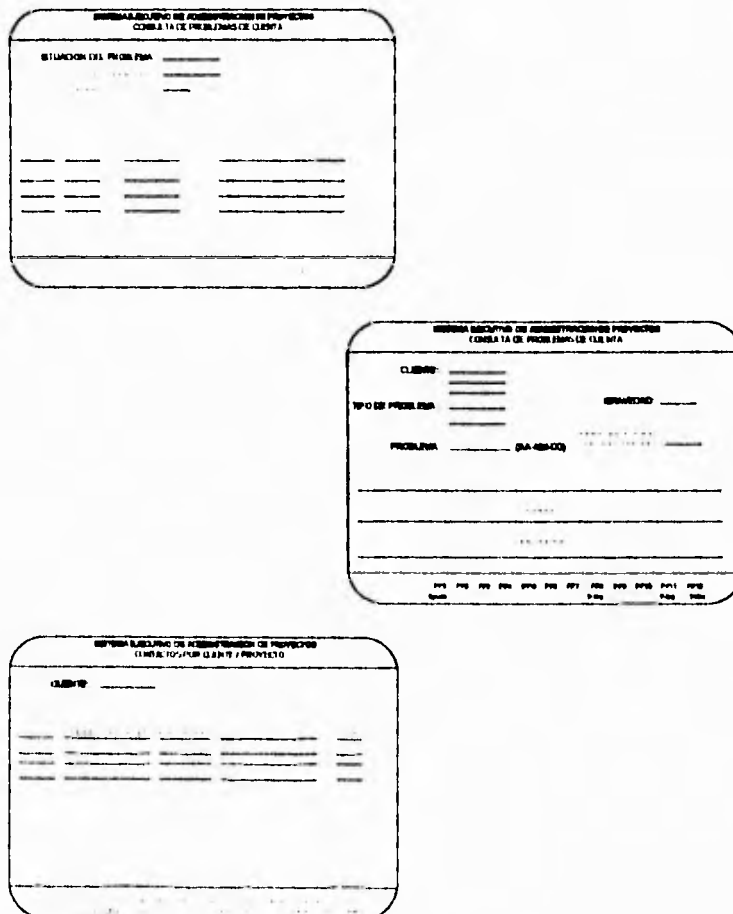


Fig. 26

3.5 REPORTES DE SOPORTE TECNICO

Este procedimiento permite consultar la información de los reportes del Area de Soporte Técnico.

El acceso de este procedimiento presentará la siguiente pantalla:

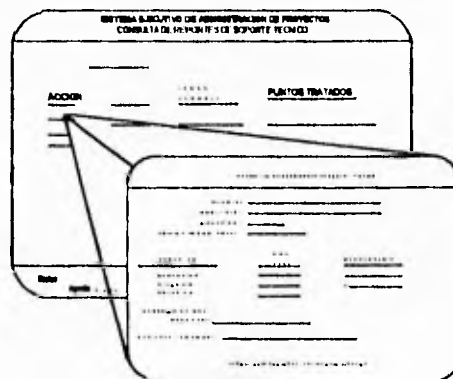


Fig. 27

La consulta general se realiza por medio de la clave del cliente, digite la clave del cliente que desea consultar y presione <Enter>, el sistema desplegará una lista de la información registrada en el sistema (Primera pantalla).

Si desea consultar información específica de un reporte, posicione en el campo de acción correspondiente y digite <C>, el sistema accederá una segunda pantalla para mostrar toda la información del reporte.

3.6 VISISTAS DEL AREA COMERCIAL

Este procedimiento permite consultar todas las visitas que ha realizado el área comercial.

El acceso de este procedimiento presentará la siguiente pantalla:

El diagrama muestra dos pantallas de un sistema de gestión de visitas. La primera pantalla, titulada 'SISTEMA INTEGRADO DE ADMINISTRACION DE PROPIEDADES COMERCIALES VISITAS DEL AREA COMERCIAL', contiene un formulario con el encabezado 'VISITAS EFECTUADAS DESPUES DEL:' y un campo 'VISITA'. La segunda pantalla, que se muestra desplegada desde el campo de acción de la primera, contiene un formulario con los siguientes campos: 'NOMBRE', 'DIRECCION', 'TELÉFONO', 'CIUDAD', 'ESTADO', 'CATEGORIA', 'FECHA DE VISITA', 'FECHA DE REGISTRO' y 'FECHA DE CANCELACION'.

Fig. 28

Para consultar la información, registre la fecha (AA-MM-DD) después de la cual desea la información.(Primera pantalla).

Para consultar la información específica de una visita, posicione en el campo de acción y digite <C>, el sistema desplegará una segunda pantalla en la que podrá consultar la información del área comercial.

3.7 PROGRAMACION DE FACTURACIÓN

Este procedimiento permite consultar la programación de la facturación.

El acceso de este procedimiento presentará la siguiente pantalla:

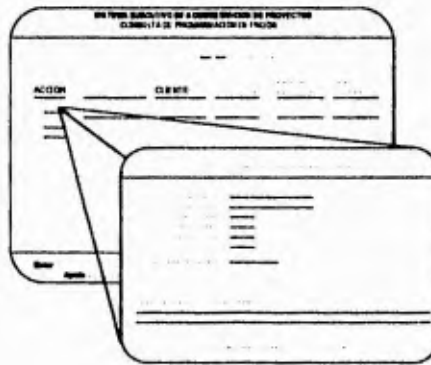


Fig. 29

La consulta se realiza respecto a un mes y año determinado, para accesar la información digite, el mes y el año requerido. El sistema desplegará un listado de todos los pagos programados a esa fecha.

Para consultar la información específica de un pago, posicóñese en el campo de selección deseado y pulse <Enter>.

El sistema desplegará una segunda pantalla, en la que podrá consultar toda la información registrada.

3.8 CARTERA VENCIDA

Este procedimiento permite consultar la situación de la cartera vencida, es decir las facturas pendientes de cobrar.

El acceso de este procedimiento presentará las siguientes pantallas:

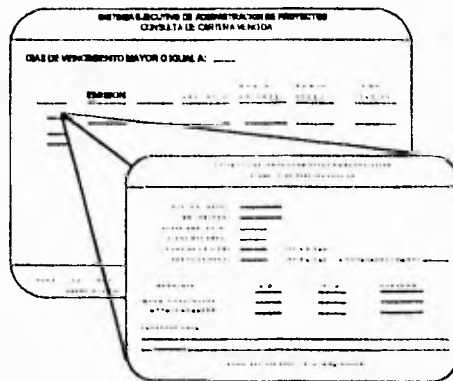


Fig. 30

Para obtener información general de la cartera vencida, registre el día de vencimiento que desea consultar. El sistema desplegará una lista de los clientes que se encuentran en esa situación.

Para consultar información específica de un cliente, digite <S> en el campo de acción y obtendrá la información deseada.

CONCLUSIONES

La realización del SISTEMA EJECUTIVO DE ADMINISTRACION DE PROYECTOS responde a las necesidades planteadas de centralizar y automatizar el flujo de información de las áreas involucradas con la atención al cliente, con un acceso actualizado, inmediato y de carácter relevante, que evitará demoras para la toma de decisiones y permitir que la empresa cumpla con sus objetivos de prestación de servicios con la calidad que se requiere para el mercado actual.

El desarrollo del sistema con herramientas CASE nos ha permitido además de aplicar los conocimientos adquiridos en nuestra formación académica profesional tanto de teoría de Base de Datos y de Ingeniería de Software, adentrarnos en la automatización de sistemas a través de las herramientas proporcionadas por CASE de Natural, teniendo como beneficios la disminución del tiempo de desarrollo y la optimización del mismo en conjunto con las técnicas RAD, elegidas por adecuarse a las características de CASE en cuanto a que nos permite la generación rápida de código y prototipos.

La importancia de la utilización de las técnicas de RAD y JAD se debe a que nos permite tener una buena administración del uso de nuestras herramientas CASE, darles un óptimo aprovechamiento y brindarle al usuario la oportunidad de participar en todas las etapas del ciclo de vida del proyecto, que se sienta comprometido en participar y el verdadero dueño del sistema.

Con JAD logramos brindarle al usuario una buena atención durante nuestras sesiones de trabajo, las cuales fueron muy frecuentes como lo plantea la técnica. Evitando que al usuario le resulte tediosa la sesión, que se traten puntos clave y no divagar en la discusión.

El sistema fué realizado en una plataforma de mainframe bajo un ambiente UNIX. Por requerimientos posteriores se migró a un ambiente PC utilizando el lenguaje Natural for Windows.

El sistema cubrió las funciones de:

- * Información sobre las juntas y reuniones con el cliente (los puntos tratados, quienes participaron).
- * El avance de los proyectos (días de retrasos, motivos).
- * Los problemas técnicos que se tienen en algunas aplicaciones y el porque de estos (la persona que recibió la queja y quien la resolvió).
- * La información referente con la facturación y cobranza de los proyectos.

* El poder informar al cliente con una semana de anticipación de su próxima fecha de pago y si él lo requiere informarle sobre su facturación. Evitando posibles conflictos en cuanto al pago.

También cubre con una correcta consistencia de información, debido a que el sistema es alimentado por varias áreas, para lo cual se coordinaron y se comprometieron de hacer un buen uso del sistema.

BIBLIOGRAFIA

**BUTLER BLOOR LTD, CHALLENGE HOUSE
SHERWOOD DRIVE, BLETCHLEY
MILTON KEYNES
UNITED KINGDOM**

**ROGER PRESSMAN
INGENIERIA DE SOFTWARE
MC GRAW HILL**

**CARMA MC CLURE, CASE
LA AUTOMATIZACION DEL SOFTWARE
ADDISON WESLEY IBEROAMERICANA, 1993**

SEMINARIO DE DISEÑO ESTRUCTURADO

**THE MYTHICAL MAN-MONTH: ESSAYS ON SOFTWARE ENGINEERING
FEDERIC P. BROOKS
ADDISON WESLEY**

**JOIN APPLICATION DESIGN: THE GROUP SESSION APPROACH TO
SYSTEM DESIGN**

**JUDY H. AUGUST.
PRENTICE HALL/ YOURDON PRESS 1991**

**STRUCTURED RAPID PROTOTYPING: AN EVOLUCIONARY APPROACH TO
SOFTWARE DEVELOPMENT**

**JOHN L. CONNELL Y LINDA BRICE
PRENTICE HALL / YOURDON PRESS 1989**

MANUALES

**ADABAS
NATURAL
PREDICT
NATURAL ARCHITECT
NATURAL CONSTRUCT**

DE SOFTWARE A.G. DE MEXICO

ARTICULOS

**SOFTWARE ENGINEERING
CONTROLLED RAPID PROTOTYPING
WARREN KEUFFEL
DATA BASED ADVISOR, 1992**

REVISTAS

**SOFTWARE DEVELOPMENT
DATABASE PROGRAMMING AND DESIGN**