



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

80
2EJ

FACULTAD DE INGENIERÍA

DISEÑO E IMPLEMENTACION DEL SISTEMA DE
PAGO A TRANSPORTISTAS DE PRODUCTOS DE
PEMEX A TRAVES DE UNA HERRAMIENTA C. A. S. E.

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N :

MA. DOLORES PEÑALOZA BLANCO
LAURA P. JACOME DEL MORAL
RODOLFO I. CARREOLA RIVAS



DIRECTOR DE TESIS M. en I. ING. JUAN CARLOS ROA BEIZA

1994

U.N.A.M.

1775



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.



"-El principio de la sabiduría es el temor de Jehová.
- Los insensatos desprecian la sabiduría y la enseñanza." Proverbios .

"- Porque Jehová da la sabiduría y de su boca viene el conocimiento y la inteligencia."

Proverbios.



A mi Dios :

Gracias te doy padre mío porque desde el vientre de mi madre me has cuidado y me has traído hasta concluir esta meta y siempre me has bendecido.

A mis Padres:

Porque me enseñaron a temer y amar a mi Señor y esa ha sido la mas grande enseñanza para poder terminar mi carrera.

A mi Esposa:

Gracias Amada mía esposa mía, por que eres la mas grande bendición que Dios me ha dado y porque con tu amor, tu aliento y tu ayuda he podido terminar esta carrera.

A mi Hijo :

Hijo mío eres muy pequeño, para poder decirte lo que por tí siento ya que naciste en el mismo tiempo en que termino esta meta, pero al igual que a mí me enseñaron yo te enseñare que la sabiduría es el temor al Señor.

A mis Hermanos (Ruben, Ruth, Samuel y Rachel)

Por su amor y comprensión en los momentos difíciles.

A mis Compañeros:

Dolores : Por su madurez y gran apoyo.

Laura: Por su alegría con desenfado hacia la vida.

Rodolfo: Por la admirable fidelidad a tus ideas y convicciones.



A MI MADRE:

CON TODO MI AMOR LE DEDICO ESTE TRABAJO.
POR HABERME DADO EL SER Y ENSEÑARME
EL CAMINO A SEGUIR, PORQUE CON SUS IN-
VALUABLES DONES OBTENGO LA CONFIANZA
EN MI MISMA.

A MI PADRE:

CON AMOR, POR LA CONFIANZA Y APOYO QUE
SIEMPRE ME HA BRINDADO.

A MI HERMANO JUAN MANUEL:

POR TODO LO QUE SIGNIFICAS PARA MI,
PORQUE GRACIAS A TU GRAN APOYO Y
CONFIANZA PUEDO LOGRAR UNA META MAS
EN MI VIDA.

A MIS HERMANAS TERE Y ROSA:

POR SU CARIÑO Y COMPRENSIÓN QUE SIEMPRE
ME HAN DEMOSTRADO

A ARMANDO:

PUES EL AMOR QUE NOS UNE ME HA PERMITIDO
TENER LAS FUERZAS SUFICIENTES PARA SEGUIR
SIEMPRE ADELANTE.
GRACIAS, TE AMO.

A MI AMIGA CONSUELO CHACÓN:

POR SU SINCERA Y LEAL AMISTAD

A MIS COMPAÑEROS Y AMIGOS:

DOLORES : POR SU MADUREZ Y GRAN APOYO
ISRAEL: POR MOSTRARME UNA NUEVA LUZ
RODOLFO: POR LA ADMIRABLE FIDELIDAD A TUS
IDEAS Y CONVICCIONES.
PORQUE CON SU APOYO Y CONFIANZA LOGRAMOS
JUNTOS LA CULMINACIÓN DE ESTA TESIS

A MIS PROFESORES:

POR HABERME TRANSMITIDO SUS
CONOCIMIENTOS Y DEDICADO SU TIEMPO



AGRADEZCO MUY ESPECIALMENTE AL M. EN ING.
JUAN CARLOS ROA BEIZA POR LA ORIENTACIÓN Y
ASESORÍA QUE ME BRINDO.

AL LIC. IGNACIO SOTO BORJA,
AL LIC. ALEJANDRO OCHOA IZAGUIRRE,
AL ING. LUIS PIÑERA CORONADO.
Y A TODOS Y CADA UNO DE LOS QUE
CONTRIBUYERON EN LA REALIZACIÓN DEL
PRESENTE TRABAJO.

" Abre tus ojos al inmenso cielo,
Busca una luz que mejor ilumine,
Ayuda con fe a causas nobles
...porque nadie desea llorar
...porque nadie desea morir"

Mami:

Gracias por todo lo que me has dado de manera tan incondicional, gracias porque me enseñaste a ser fuerte y libre.

Te amo y te admiro mucho porque eres una gran mujer y un hermoso ser humano, pero sobre todo porque sacaste adelante a tus hijas.

A mi Padre:

Gracias por mostrarme que el conocimiento es parte esencial de la vida del hombre .

A mis hermanos: (Claudia, Lourdes, Sofia , Fer, Lulú, Carlos y Gaby)

Gracias por haber compartido tantas vivencias a mi lado; estudio, trabajo, juegos, risas y lágrimas. Gracias porque espero que nos sigamos dando fortaleza y ánimos para intentar ser felices.

Los quiero con todo mi corazón

A ti :

Gracias por enseñarme que el amor más importante de un ser humano, es el amor con el que toma la vida.

Te amo, creo que siempre te amaré

A mis Amigos:

Laura: Por tu alegría con desenfado hacia la vida.

Israel: Por mostrarme una nueva luz.

Rodolfo: Por la admirable fidelidad a tus ideas y convicciones.

A ti amigo que me has dado tu amistad y tu tiempo.

CONTENIDO

INTRODUCCIÓN.....	iii
CAPITULO I.	
ANTECEDENTES	
I.1.- C.A.S.E.	
I.1.1. Antecedentes.....	1
I.1.2. Bloques que integran el C.A.S.E.....	12
I.1.3. Clasificación de las herramientas C.A.S.E.....	17
I.1.4. Filosofía.....	41
I.1.5. Técnicas y Metodologías Estructuradas que soportan las Herramientas C.A.S.E.	57
I.2.- I.E.W.	
I.2.1 Planeación.....	126
I.2.2 Análisis.....	159
I.2.3 Diseño.....	172
I.3.- Concepto de Base de Datos.....	187
I.4.- Topología de Base de Datos.....	216
I.5.- Modelo de Base de Datos para un Ambiente Multiusuarios	230
CAPITULO II.	
IDENTIFICACIÓN Y EVALUACIÓN DEL PROCEDIMIENTO DE PAGO A TRANSPORTISTAS .	
II.1.- Análisis de funciones y responsabilidades del área de distribución	1
II.2.- Análisis del procedimiento de pago de fletes a transportistas	39
II.3.- Planteamiento de la problemática y disponibilidad de los recursos.....	113

II.4.- Análisis de los requerimientos de usuario.....	121
II.5.- Organización y análisis de la información.....	127
II.6.- Evaluación y selección de opciones.....	155

CAPITULO III.

DISEÑO Y DESARROLLO DEL SISTEMA.

III.1.- Elaboración del modelo conceptual	1
III.2.- Explotación de la herramienta I.E.W	
III.2.1.- Diseño	4
III.2.2.- Desarrollo.....	109
III.2.3.- Diccionario de datos.....	127
III.3.- Evaluación de Resultados.....	181

CONCLUSIONES.....	A
-------------------	---

DOCUMENTOS.....	B
-----------------	---

GLOSARIO.....	C
---------------	---

BIBLIOGRAFÍA.....	D
-------------------	---

MANUAL DEL USUARIO	ANEXO
--------------------------	-------

INTRODUCCION

Como la mayoría de los proyectos de software se retrasan según lo planificado, y como los directivos están ansiosos por empezar a trabajar con el siguiente proyecto en espera de ser desarrollado, no se dedica tiempo a planear. Las presiones para cubrir las crecientes demandas de más proyectos y la permanencia en el mismo nivel técnico, han oscurecido la importancia de evaluar lo bien que funcionan las tecnologías del software en la práctica.

La utilización de las herramientas y metodologías del software, es sólo el primer paso para mejorar el proceso del software. También se debe evaluar su utilidad en la práctica. El mayor problema que bloquea las mejoras del software, es la falta de información sobre el estudio de casos de proyectos. Es difícil convencer a una dirección sobre el valor que tiene el probar nuevas tecnologías, cuando no hay soporte de tal evidencia. También es difícil especular si más adelante serán necesarios cambios o mejoras.

Lo fundamental en el proceso de producción de software de calidad, es el compromiso continuo por parte de los profesionales del desarrollo y del mantenimiento, de investigar como mejorar los desarrollos de software y su producción. Deberían definirse métodos para evaluar los efectos de tales herramientas así como, analizar los datos para descubrir los procesos de mayor complejidad, la estimación deficiente de los calendarios, de los costos y de las frecuencias de error.

Partiendo del hecho que este principio raramente se practica y poniendo de manifiesto su importancia, debe considerarse como paso fundamental la implantación de una tecnología **CASE** en una organización. La razón es proporcionar información útil para mejorar los futuros proyectos y productos de software:

- ¿Qué funcionó?
- ¿Que no funcionó?
- ¿Cuál es el próximo paso para intentar un trabajo mejor en el desarrollo y mantenimiento de los sistemas?

Es necesario tener una nueva visión del ciclo de vida, de que admita diversos enfoques para construir los sistemas de software. El uso de metodologías con el soporte de las herramientas **CASE** resulta apropiado para poder definir completamente los requerimientos de dichos sistemas desde el inicio de su desarrollo.



CAPITULO I
ANTECEDENTES



I.1 INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADORA (COMPUTER AIDED SOFTWARE ENGINEERING)

I.1.1. Antecedentes (metodología).

La historia de la tecnología CASE (Ingeniería de software asistida por computadora), nos lleva a la automatización del desarrollo de software, y comienza a principios de los años ochenta con la introducción de la documentación asistida por computadora y de las herramientas de diagramación, que soportan a este.

Estas herramientas representan parte de las primeras herramientas de desarrollo de software basadas en computadoras personales y los primeros intentos para automatizar el análisis y el diseño de tareas. Estas herramientas se idearon simplemente para el dibujo en puestos individuales.

Se emplearon para crear diagramas estructurados (de flujo de datos, organigramas y de entidad/relación). Su propósito era producir automáticamente la documentación estructurada requerida para las distintas metodologías de desarrollo estructurado.

Las diferentes herramientas CASE soportan diferentes metodologías de desarrollo estructurado, como el desarrollo estructurado de análisis y de diseño de Yourdon o, el diseño estructurado de Jackson o la ingeniería de la información de Martin.

Evolución de la tecnología CASE

Principios de los años Ochenta	Mediados de los años Ochenta	Últimos años de los Ochenta	Principios de los años Noventa
Documentación asistida por computadora	Comprobación automática del análisis del diseño	Generación automática del código a partir de las especificaciones de diseño	Conductor de metodología Inteligente
Diagramación asistida por computadora	Depósito automático de la información del sistema	Engenere del diseño automático y la programación automática	Interfaz amigable con el usuario
Herramientas de Análisis y de diseño			La reutilización como metodología de desarrollo

Fig 1.1.1.1. El CASE es una tecnología con una buena sintonía de pocos años

El CASE propone una nueva formulación del concepto de ciclo de vida del software, basada en la automatización. La idea básica que nace en el CASE es proporcionar un conjunto de herramientas bien integradas y que ahorren trabajo, enlazando y automatizando todas las fases del ciclo de vida del software.

Las tecnologías tradicionales del software son de dos tipos: herramientas y metodologías. Estas categorías incluyen herramientas de tercera, de cuarta y (más recientes) de quinta generación. La mayoría de estas herramientas son autónomas, basadas en mainframe (ordenador principal) y dirigidas a la implantación del ciclo de vida del software.

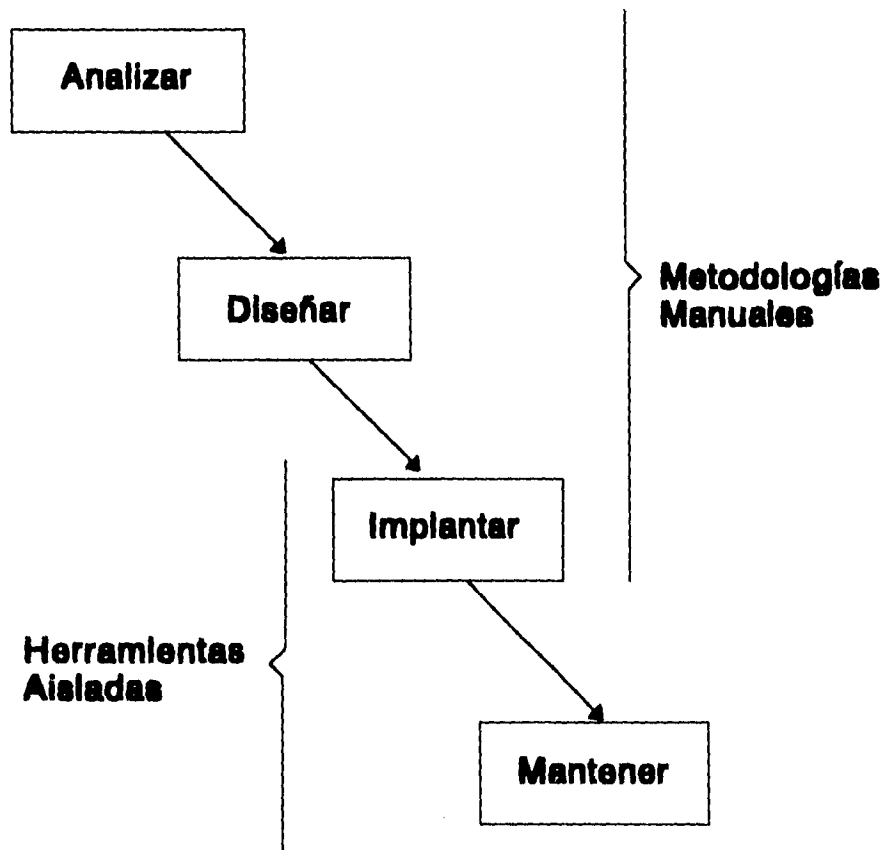


fig. I.1.1.2. Las tecnologías tradicionales de software son de dos tipos: manuales y con herramientas aisladas. La tecnología CASE es una combinación de herramientas y metodologías totalmente integrada, con especial énfasis en la automatización del ciclo de vida del software.

Capítulo I Ingeniería de Software Asistida por Computadora

Las primitivas herramientas CASE se dirigieron principalmente a la automatización de la documentación y de la comunicación como una mejora clave de la productividad del software. A mediados de los años ochenta, las herramientas CASE se mejoraron para proporcionar dos funciones muy importantes:

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

Los diagramas se revisan para asegurar que están completos y correctos antes de utilizarlos como especificaciones de diseño de la implantación del programa. La comprobación se basa en las reglas de la metodología estructurada soportada. Además se registran las referencias cruzadas de la información y se eliminan las redundancias.

Después de la validación de los diagramas, estos se almacenan en diccionarios o en bases de datos, llamados depósitos CASE. Una vez almacenados los diagramas se pueden actualizar fácilmente compartiéndolos con todo el equipo del proyecto y más tarde utilizarlos en otros proyectos de desarrollo de software.

En este punto, la aplicación importante de CASE está en el diseño automático, porque el análisis y diseño se consideran como las fases iniciales del ciclo de vida del software.

Capítulo I Ingeniería de Software Asistida por Computadora

Realizando un trabajo mejor en las fases iniciales del ciclo de vida redundará favorablemente en la calidad y costo del sistema. Con la asistencia automática de las herramientas CASE, el tiempo de diseño se reduce substancialmente, a la vez que se producen especificaciones de diseño de alta calidad y sin errores.

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 se relaciona con la asistencia por computadora de 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 de 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 automático y la programación automática se requiere un puente entre dos entornos Hardware (computadoras personales mainframes) y dos tecnologías de Software (CASE y cuarta generación). Una parte muy importante de este puente será añadir un depósito al entorno CASE.

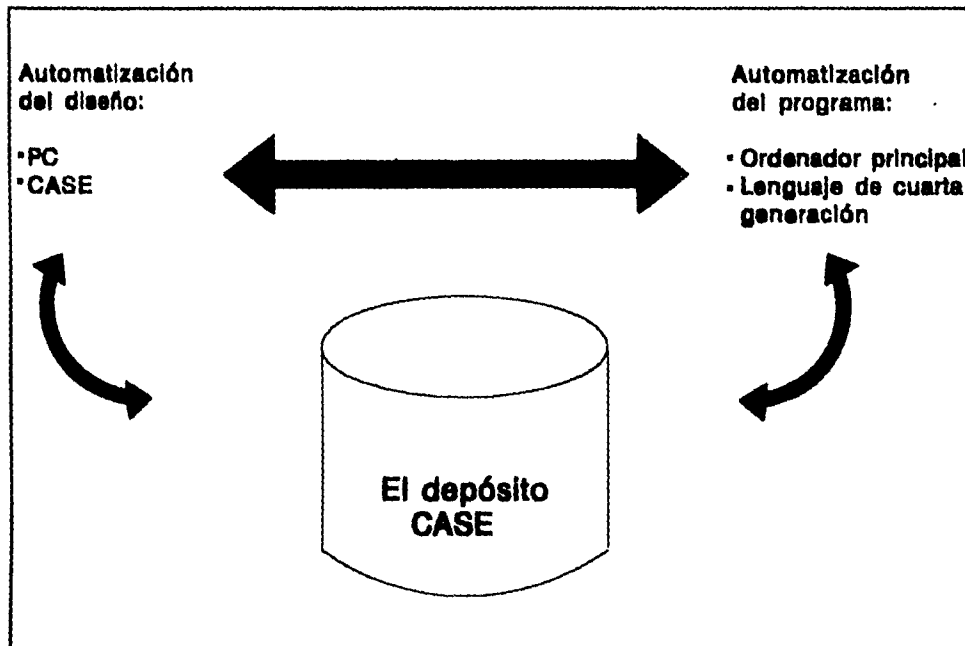


Figura I.1.1.3. La unión de la automatización del diseño con la automatización del programa significa que puede generarse automáticamente todos los sistemas del software desde las especificaciones gráficas a alto nivel del sistema. Una parte muy importante de esta unión se debe al depósito CASE.

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

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

En lugar de ir creando cada vez nuevo software, o de mejorar el dudoso, los profesionales del desarrollo de software pueden utilizar los componentes del software reutilizable almacenado en los depósitos CASE .'

No solamente podrán los profesionales reutilizar planes de proyectos, modelos de prototipos, modelos de datos y especificaciones de diseño.

OBJETIVOS DEL CASE.

El CASE ha cambiado radicalmente la forma de construir los sistemas de 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 interfaces gráficas.

Capítulo I Ingeniería de Software Asistida por Computadora

La informática y la ingeniería se unieron de esta forma. Volviendo al presente, se puede observar que encontramos ingeniería asistida por computadora (CAE), diseño asistido por computadora y fabricación integrada por computadora (CIM, sucesor del CAM) como actividades usuales en la mayoría de las empresas.

La automatización de la ingeniería no sólo ha llegado sino que se ha convertido en una parte integral del proceso, por lo cual, los ingenieros mecánicos y eléctricos no pueden volver atrás y cambiar el futuro, sin embargo de cierta forma los ingenieros del Software sí pueden; tienen la oportunidad de modelar el futuro del CASE aprendiendo de la evolución del CAE, del CAD y del CIM.

La ingeniería del Software asistida por computadora tiene el potencial de llegar a ser el avance tecnológico más importante en la historia del desarrollo de software.

Hoy en día, las herramientas CASE se añaden a la caja de herramientas del ingeniero de software. El CASE proporciona al ingeniero la capacidad de automatizar las actividades manuales y de mejorar su enfoque de trabajo.

Hoy en día el CASE está donde estaban CAE/CAD/CIM en 1975, no hay duda de que el CASE afectará la ingeniería del software de la misma forma que sus antecesores a otras ramas de la ingeniería.

Sin embargo existen algunas diferencias importantes. En sus primeros días de evolución, el CAD/CAE/CIM desarrolló prácticas de ingeniería que habían sido probadas durante los últimos 100 años.

El CASE sin embargo, proporciona un conjunto de herramientas semiautomatizadas y automatizadas que están desarrollando una cultura de ingeniería nueva para muchas empresas. Existe una diferencia importante en la aceptación y el impacto.

El CAD/CAE se centra casi exclusivamente en la resolución de problemas y el diseño. El CIM constituye su continuación hacia los procesos de fabricación.

El objetivo más importante del CASE (a largo plazo) es conseguir la generación automática de programas desde una especificación a nivel de diseño

LAS VENTAJAS DE LA TECNOLOGÍA C.A.S.E.

- 1) Permite las técnicas estructuradas.
- 2) Impone las ingeniarías de software y de la información.
- 3) Aumenta la calidad del software mediante comprobación automática.
- 4) Favorece la realización de prototipos.
- 5) Simplifica el mantenimiento del programa.
- 6) Acelera el proceso de desarrollo.
- 7) Libera al profesional de desarrollo de la principal parte creativa en el desarrollo del software.
- 8) Anima al desarrollo evolucionado y gradual.
- 9) Posibilita la reutilización de los componentes del software.

BENEFICIOS DEL C.A.S.E.

Como se puede observar de las ventajas que se listaron anteriormente se muestran las ventajas que el CASE ofrece a los profesionales del desarrollo de software. Esta exposición sugiere que el CASE no es enteramente una tecnología nueva, sino que está construida sobre técnicas y herramientas comprobadas en la práctica.

En este sentido, el CASE puede definirse como un conjunto de Conceptos Estructurados y de Metodologías con un nuevo envase. La novedad es la automatización.

I.1.2. BLOQUES QUE INTEGRAN EL C.A.S.E.

La ingeniería del software asistida por computadora puede ser tan simple como una única herramienta que permita desarrollar una actividad específica, o tan compleja como un entorno que integre distintas herramientas, una base de datos, gente, hardware, una red, sistemas operativos, estándares y muchos otros componentes.

Cada uno de los bloques que componen el CASE, constituye la base del siguiente, con las herramientas situadas en la cima de la pila. El fundamento para un CASE efectivo tiene poco que ver con las herramientas de ingeniería del software en sí mismas.

Los buenos entornos de ingeniería del software se construyen sobre una Arquitectura de Entorno que engloba los correspondientes sistemas de software y de hardware.

Además, la arquitectura de entorno debe considerar los patrones de trabajo humanos que se aplican durante el proceso de ingeniería de software.

Durante los años 60, 70 y 80, el desarrollo de software era una actividad para grandes máquinas. Los terminales estaban unidos a computadoras centrales y cada programador compartía los recursos de ésta. Las herramientas de software disponibles estaban diseñadas para operar cada entorno con terminales de tiempo compartido.

En esta época la tendencia en el desarrollo del software está lejos de las grandes computadoras y enfocada hacia las estaciones de trabajo como plataforma de ingeniería de software.

Las estaciones de trabajo individuales se interconectan mediante redes para que los ingenieros de software puedan comunicarse de forma efectiva. Las bases de datos de proyectos están disponibles a través de un servidor de archivos en red que es accesible desde todas las estaciones de trabajo.

Un sistema operativo que gestiona el hardware, la red y las herramientas mantiene todo el entorno unido.

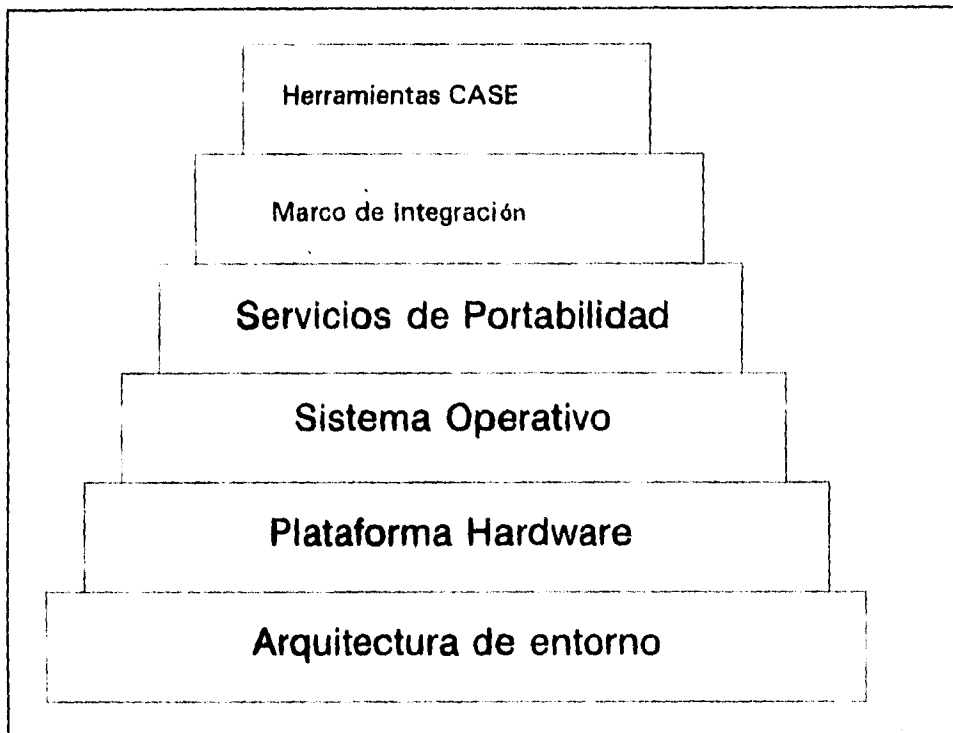


fig. 1.1.2.1 Bloques constitutivos del C.A.S.E.

La arquitectura de entorno, compuesta por la plataforma de hardware y el soporte del sistema operativo (incluida la red y la gestión de la base de datos) forman la base del CASE. Pero el entorno CASE, en sí mismo, necesita otros componentes. Un conjunto de Servicios de Portabilidad constituyen un puente entre las herramientas CASE y, su Marco de Integración y la Arquitectura de entorno.

El marco de integración es un conjunto de programas especializados que permiten a cada herramienta CASE comunicarse con las demás para crear una base de datos de proyectos y mostrar una apariencia homogénea al usuario final (el ingeniero de software). Los servicios de portabilidad permiten que las herramientas CASE y su marco de integración puedan migrar a través de diferentes plataformas de hardware y sistemas operativos sin grandes esfuerzos de adaptación.

Los bloques representan una base para la integración de las herramientas CASE, sin embargo la mayoría no han sido construidas utilizando todos los bloques componentes comentados anteriormente. Esto es, una herramienta se utiliza como ayuda en una actividad concreta de ingeniería del software (p. ej.: modelado del análisis), pero no se comunica directamente con otras herramientas; no está unida a una base de datos de proyectos y no se parte de un entorno CASE integrado (I-CASE). Aunque esta situación no es la ideal, una herramienta CASE puede ser utilizada eficientemente, aun siendo una solución puntual.

A continuación se podrán mostrar los niveles de integración del CASE. En el nivel más bajo del espectro de integración está la Herramienta individual (solución puntual).

Cuando las herramientas proporcionan facilidades para el Intercambio de datos (la mayoría lo hacen) el nivel de integración aumenta ligeramente. Estas herramientas generan una salida en un formato estándar compatible con otras herramientas que puedan leer ese formato.

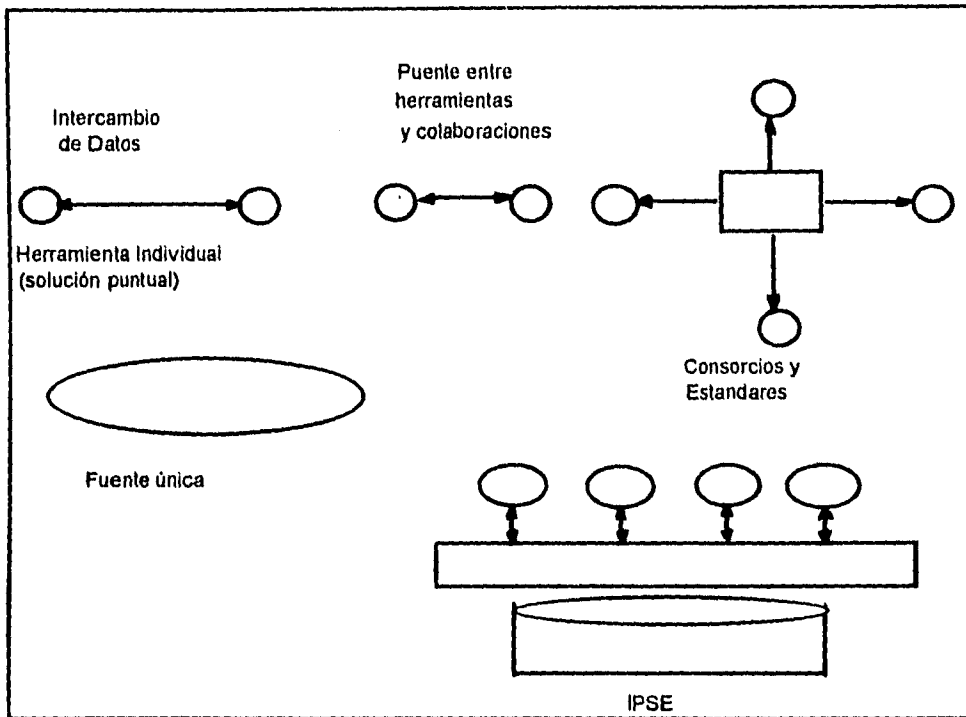


fig. I.1.2.2. Opciones de Integración

En algunos casos los que construyen herramientas CASE complementarias trabajan juntos para establecer un puente entre ellas (p. ej.: una herramienta de diseño y análisis que se une aun generador de código). Utilizando este enfoque, la compatibilidad entre herramientas puede generar productos finales que serian dificiles de desarrollar utilizando cada herramienta por separado. La Integración por Fuente Única se da cuando un constructor de herramientas CASE integra diferentes herramientas y las vende como un único paquete. Aunque este enfoque es bastante efectivo, la mayoría de los entornos provenientes de una misma fuente tienen una arquitectura cerrada que hace difícil añadir nuevas herramientas de otros vendedores.

Al final del espectro de integración está el Entorno de Soporte de Proyectos Integrado (del inglés IPSE). Se crean estándares para cada uno de los bloques componentes. Los vendedores de herramientas CASE utilizan estos estándares IPSE para construir herramientas compatibles entre sí.

I.1.3. CLASIFICACIÓN DE LAS HERRAMIENTAS C.A.S.E.

La columna vertebral de la tecnología CASE es una nueva generación de herramientas de software.

Algunas de estas herramientas están basadas en micros, tienen potentes entornos gráficos que hacen más amigable la interfaz del usuario y se incorporan a bases de trabajo donde pueden ser fácilmente llamadas, alimentadas y utilizadas por otros. Las herramientas CASE (CASE TOOLS) se enfocan hacia la productividad individual del profesional del desarrollo del software.

Con el uso de las estaciones de trabajo, de las redes del área local y de las herramientas basadas en las computadoras personales, el profesional del desarrollo de software puede trabajar por primera vez desde un entorno de desarrollo de software confiable y dedicado.

Ya que las herramientas CASE se utilizan principalmente por los profesionales del desarrollo de software y no por los usuarios finales, la tecnología CASE no debe considerarse solamente como una mejora de las herramientas de cuarta generación. Además, las herramientas CASE difieren de las herramientas de cuarta generación en otros dos aspectos.

Primero, muchas herramientas CASE están diseñadas para estaciones de trabajo, mientras que muchas herramientas de las de cuarta generación están diseñadas para computadoras principales (mainframes). Segundo, el CASE es una tecnología de software de propósito general para utilizarse en el desarrollo de todo tipo de sistemas (grandes y pequeños, comerciales y científicos, en línea y de tiempo real etc.), mientras las herramientas de cuarta generación se utilizan principalmente para desarrollar aplicaciones comerciales de tamaño pequeño o mediano. En general, las herramientas CASE se diferencian de las primitivas herramientas de software en algunas cosas muy importantes. A saber, las herramientas CASE fueron diseñadas para:

- 1) Soportar un entorno personal dedicado.
- 2) Utilizar gráficos para especificar y documentar los sistemas.
- 3) Juntar todas las fases del ciclo de vida del software.
- 4) Capturar y juntar en la computadora toda la información sobre el entorno del software en un sistema, desde los requerimientos iniciales hasta las actividades para un mantenimiento constante.
- 5) Utilizar la inteligencia artificial para realizar automáticamente muchas de las rutinarias tareas del desarrollo y mantenimiento del software.

EJEMPLOS DE HERRAMIENTAS CASE.

Aún más, las herramientas CASE realizan muchos más tipos de tareas que las herramientas de la tercera y cuarta generación. Soportan la automatización de una gran variedad de tareas de desarrollo y mantenimiento del software, incluyendo la gestión del proyecto.

Como se muestra en la siguiente figura, los ejemplos de herramientas case incluyen:

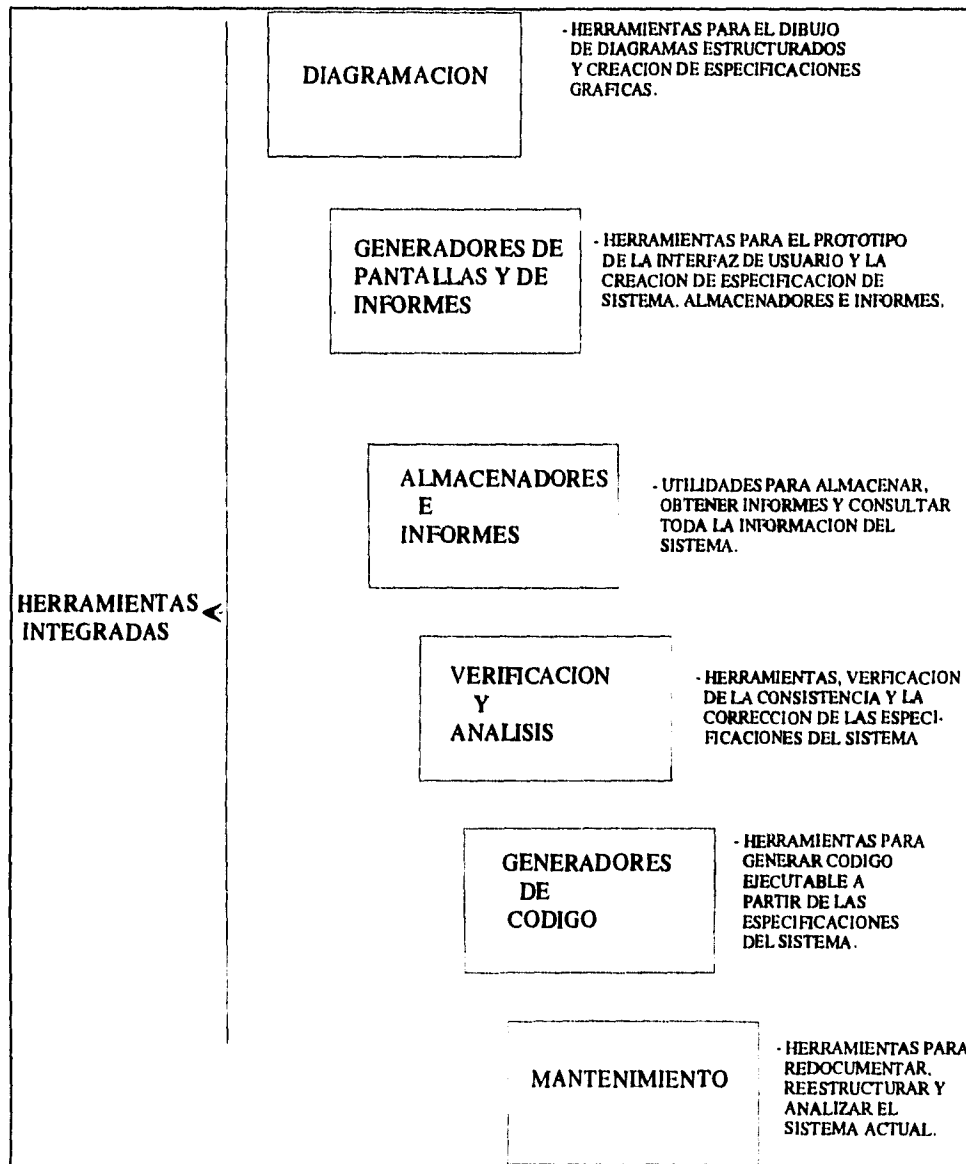


fig. I.1.3.1 Las herramientas CASE cubren muchos tipos de las tareas del desarrollo y de la automatización del software.

- 1) Herramientas de diagramación para el dibujo de diagramas estructurados y la creación de especificaciones de sistemas gráficos.
- 2) Pantallas de informes gráficos para la creación de las especificaciones de sistemas y de formularios de prototipos simples.
- 3) Diccionarios, sistemas de desarrollo de bases de datos, y utilerías para almacenamiento de información y, consulta de la información técnica y de proyectos del sistema de desarrollo.
- 4) Herramientas de validación de especificaciones para la detección automática de las que son incompletas, incorrectas sintacticamente o inconsistentes.
- 5) Generadores de código para generar automáticamente código ejecutable a partir de las especificaciones obtenidas gráficamente.
- 6) Generadores de documentación para producir la documentación técnica y del usuario requeridas por las técnicas estructuradas.

Aunque las herramientas son una parte muy importante de CASE, la tecnología CASE consta de mucho más. El CASE es una redefinición del entorno completo del software.

Una forma clave en la que CASE cambia el entorno de desarrollo del software, es que muchas herramientas CASE operan en entornos de estación de trabajo.

Esto cambia el desarrollo del software en un proceso altamente interactivo en el cual la detección de los errores comienza en los pasos iniciales de la definición de los requerimientos del sistema.

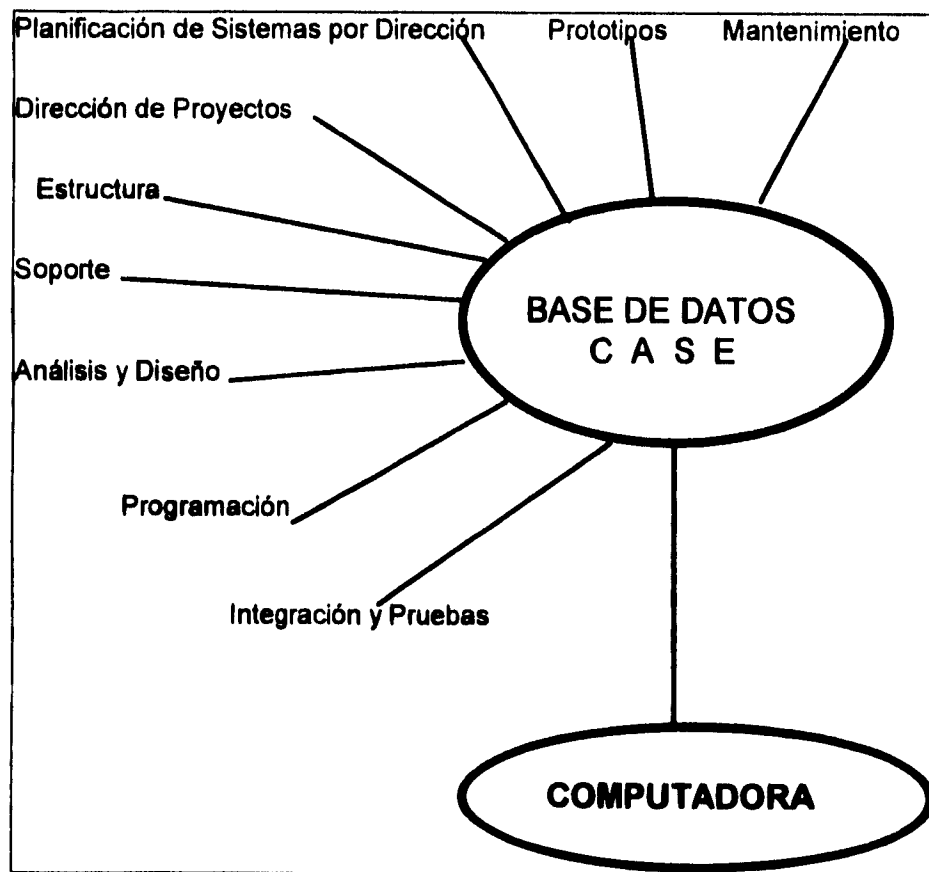


fig. I.1.3.2. Una Taxonomía de las herramientas C.A.S.E.

Las herramientas CASE se pueden clasificar por su función ó por su papel como instrumentos para el personal técnico o los directivos, por la arquitectura del entorno (hardware y software) que las soportan o incluso por el costo. La taxonomía que se presenta utiliza la funcionalidad como criterio principal.

HERRAMIENTAS DE PLANIFICACIÓN DE SISTEMAS DE DESARROLLO.

Mediante el modelado de los requisitos de la información estratégica de una organización, las Herramientas de Planificación de Sistemas de desarrollo proporcionan un "metamodelo" del cual se pueden obtener sistemas de información específicos. En lugar de centrarse en los requisitos de una aplicación específica, la información de gestión se modela según va pasando a través de las distintas entidades organizativas de una compañía.

El Objetivo principal de las herramientas de esta categoría es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la organización.

HERRAMIENTAS DE REALIZACIÓN DE PROYECTOS.

Las herramientas se centran en un elemento específico de la dirección del proyecto, en lugar de proporcionar un soporte global para la actividad de dirección..

El director del proyecto puede hacer estimaciones útiles de esfuerzo, costo y duración de un proyecto, definir una estructura de partición del trabajo (EPT), y hacer una planificación realista del mismo, y así como el seguimiento del proyecto de forma continua, además se pueden utilizar estas herramientas para recoger datos que le permitan realizar una estimación de la productividad del desarrollo y de la calidad del producto. Para aquellos directores que dirigen proyectos de desarrollo de software bajo contrato, existen herramientas CASE que van desde los requisitos de la petición de propuesta inicial hasta el trabajo de desarrollo que convierte estos requisitos en un producto final.

HERRAMIENTAS DE PLANIFICACIÓN DE PROYECTOS.

Las herramientas que caen dentro de esta categoría se centran principalmente en dos áreas: la primera, el esfuerzo y costo de un proyecto de software; la segunda la planificación del proyecto.

La estimación del costo permite al director del proyecto considerar el tamaño del mismo, utilizando una medida indirecta como lo pueden ser las líneas de código, puntos de función y descripción de las características globales del proyecto (p.e. complejidad, experiencia del personal, madurez de proceso).

Las herramientas de estimación calculan el esfuerzo, la duración del proyecto y el número de gente recomendada; el director puede cambiar la fecha de entrega del proyecto y examinar su repercusión sobre el costo final.

Las herramientas de planificación de proyectos permiten al director definir todas las tareas (estructura de partición del proyecto), crear una red de tareas (utilizando normalmente una entrada gráfica), representar las interdependencias entre tareas y modelizar la cantidad de paralelismo posible dentro del proyecto.

HERRAMIENTAS DE SEGUIMIENTO DE REQUISITOS.

El objetivo de las herramientas de seguimiento de requisitos es proporcionar un enfoque sistemático para aislar requisitos, comenzando con las especificaciones del cliente. La herramienta típica de seguimiento combina la evaluación interactiva de texto con un sistema de gestión de base de datos que almacena y valora cada requisito del sistema extraído de las especificaciones originales. El analista valora el requisito especificado en la frase y lo introduce en la base de datos. El trabajo de desarrollo posterior puede referenciarse en la base de datos, de tal forma que se asegure la satisfacción de cada requisito.

HERRAMIENTAS DE DESARROLLO Y MEDIDA.

Las herramientas de medida actuales se centran en las características del producto y del proceso. Las herramientas orientadas a la gestión parten de medidas específicas del proyecto (p.e. LDC/ hombre/mes, defectos por punto de función) que proporcionan una indicación global de la productividad y de la calidad.

Las herramientas con orientación técnica determinan medidas técnicas (p.e. complejidad ciclomática) que proporcionan una visión más amplia de la calidad del diseño y del código. La mayoría de las herramientas de medida avanzadas mantienen una base de datos de la "medida del mercado". Basándose en las características del producto y del proyecto proporcionadas por el usuario "comparan" las medidas obtenidas frente a las medidas del mercado, (y a las obtenidas en proyectos anteriores) para seguir estrategias de mejora.

Las herramientas de gestión ayudan a los directores de sistemas de información a establecer prioridades entre los diferentes proyectos que compiten por un conjunto limitado de recursos. Basándose en los requisitos y prioridades del cliente, en las restricciones de la organización y en los riesgos técnicos y financieros, estas herramientas utilizan un sistema experto para sugerir el orden en el que se debe llevar a cabo un proyecto.

HERRAMIENTAS DE SOPORTE.

La categoría de herramientas de soporte engloba las herramientas de aplicación y de sistemas que complementan el proceso de ingeniería del software. Las herramientas que caen en esta amplia categoría recogen las actividades aplicables en todo el proceso de ingeniería de software. Estas incluyen herramientas de documentación, herramientas para gestión de bases de datos y de configuración del software.

HERRAMIENTAS DE DOCUMENTACIÓN.

Las herramientas de producción de documentación y de autoedición se utilizan en casi todos los aspectos de la ingeniería de software y representan una oportunidad muy interesante. La mayoría de las organizaciones de desarrollo de software emplean mucho tiempo en el desarrollo de documentos y en muchos casos el proceso de documentación es, en sí mismo muy ineficiente.

No es raro que una empresa emplee el 20 o el 30 por 100 de su esfuerzo de desarrollo en la documentación. Por esta razón, estas herramientas constituyen una opción importante para aumentar la productividad.

Las herramientas de documentación suelen estar unidas a otras herramientas CASE por medio de una interfaz de datos suministrada por el vendedor. Por ejemplo, muchas herramientas de análisis y diseño están unidas a uno o varios sistemas de autoedición, de tal forma que los modelos y textos creados durante el análisis y el diseño pueden ser transmitidos a una herramienta de documentación y añadidos a la especificación creada utilizando la misma herramienta de documentación.

HERRAMIENTAS PARA SOFTWARE DE SISTEMAS.

El CASE es una tecnología de estaciones de trabajo. Por esto, el entorno CASE debe soportar software de redes de comunicación de alta calidad, correo electrónico, boletines electrónicos y otras posibilidades de comunicación. Aunque el sistema operativo más empleado en las estaciones de trabajo de ingeniería (y un número cada vez mayor de PCs) es el UNIX, los servicios de portabilidad proporcionados por un IPSE deben permitir una migración de las herramientas CASE a otros sistemas operativos sin grandes complicaciones.

HERRAMIENTAS DE CONTROL DE CALIDAD.

La mayoría de las herramientas CASE que se venden como orientadas al control de calidad, son en realidad herramientas de medida que comprueban el código fuente para determinar sus compatibilidad con lenguajes estándar. Otras herramientas extraen técnicas métricas como base para medir la calidad del software que se está desarrollando.

HERRAMIENTAS DE BASES DE DATOS Y DE REALIZACIÓN DE CONFIGURACIÓN DEL SOFTWARE.

El software de gestión de bases de datos sirve como base para el establecimiento de una base de datos CASE (almacén) que hemos denominado base de datos del proyecto las herramientas CASE pueden ayudar en las 5 tareas principales de la gestión de configuración del software (GCS)- identificación y control de versiones, control de cambio, auditoría y gestión de estados. La base de datos CASE proporciona un organismo para identificar cada elemento de la configuración y relacionarlo con otros elementos; el acceso fácil a los distintos elementos de la configuración facilita el proceso de auditoría ; y las herramientas CASE de comunicación pueden mejorar significativamente la gestión de estados (mandando información sobre cambios a todo aquél que necesita saberlo).

Controlando los cambios de la configuración del software, las herramientas de GCS fuerzan a que todo el mundo tenga conocimiento de cada cambio, reduciendo, de esta forma, los malentendidos y aumentando la calidad del sistema.

HERRAMIENTAS PRO/SIM (Herramientas de creación de Prototipos y de simulación).

Las herramientas de creación de prototipos y de simulación (PRO/SIM) proporcionan al ingeniero de software la capacidad de predecir el comportamiento de un sistema de tiempo real antes de que sea construido. Además, le permiten desarrollar prototipos de sistemas de tiempo real que proporcionen al cliente una visión general de la función, de la operación y de la respuesta, antes de la codificación final.

La mayoría de las herramientas PRO/SIM suministran al ingeniero medios para crear modelos funcionales y de comportamiento de un sistema. Las herramientas de esta categoría proporcionan un medio de especificación de las características de rendimiento proyectadas de cada elemento del sistema, (p.e. velocidad de ejecución de una función hardware o software), definen las características de los datos de entrada y salida (p.e. frecuencia de llegada de los datos de entrada y o características de las interrupciones) y modelan las interfaces y la interconexión entre los distintos elementos del sistema.

Muchas herramientas PRO/SIM tienen la capacidad de generar código para Ada y otros lenguajes de programación que se harán más sofisticados a medida que las herramientas evolucionen.

HERRAMIENTAS PARA EL DISEÑO Y EL DESARROLLO DE INTERFACES.

Las herramientas de diseño y desarrollo de interfaces son, en realidad, un conjunto de componentes de software, tales como menús, botones, estructuras de ventanas, iconos, mecanismos de visualización, controladores de dispositivos y otros elementos de este tipo. Sin embargo, estos conjuntos de herramientas están siendo reemplazados por herramientas para desarrollar prototipos que permiten la creación rápida en pantalla de interfaces sofisticadas ajustadas al estándar elegido para el software (p.e. X-Windows, de Motif).

Los sistemas de desarrollo de interfaces de usuario (SDIU) combinan herramientas CASE para la interacción hombre/máquina, con una biblioteca de componentes de programa que permite al diseñador construir una interfaz rápidamente. Un SDIU proporciona componentes de programa que gestionan los dispositivos de entrada, validan las entradas del usuario, manejan las condiciones de error, procesan las interrupciones, recuperan errores, proporcionan realimentación visual, cursores, ayudas, actualización de ventaneas, gestión de los datos de la aplicación, edición y visualización, aíslan la aplicación de las funciones de manejo de pantalla y permiten que el usuario personalice ciertas características del entorno.

MAQUINAS DE ANÁLISIS Y DISEÑO.

Una nueva generación de herramientas de análisis y diseño, denominadas máquinas de análisis y diseño, utilizan una arquitectura basada en reglas que permite que la herramienta sea adaptada a cualquier método de análisis y diseño.

Estas permiten que el Ingeniero de software personalice la herramienta para satisfacer los requisitos de un método específico.

HERRAMIENTAS DE CODIFICACIÓN CONVENCIONALES

Las únicas herramientas de las que disponía un Ingeniero de software eran las herramientas de codificación convencionales - compiladores, editores y depuradores. Hoy las herramientas convencionales siguen existiendo en primera línea del desarrollo del software, pero están respaldadas por todas las otras herramientas CASE.

HERRAMIENTAS DE CODIFICACIÓN DE CUARTA GENERACIÓN

Los sistemas de consulta a bases de datos, los generadores de código y los lenguajes de cuarta generación han cambiado la forma en que se desarrollan los sistemas. El objetivo final del CASE es la generación automática de código esto es, la representación de sistemas a un nivel de abstracción más alto que el de los lenguajes de programación convencionales.

Idealmente, estas herramientas de generación de código no sólo traducirán la descripción de un sistema a un programa operativo, sino que también ayudarán a verificar la corrección de la especificación del sistema, de tal forma que la salida resultante satisfaga los requisitos del usuario.

Aunque en los lenguajes de cuarta generación, los generadores de código y los generadores de aplicaciones (p. e: sistemas de consulta a bases de datos) permiten que un ingeniero de software especifique un sistema a un nivel muy alto de abstracción; cada una de estas herramientas difiere en aspectos importantes.

Capítulo I Ingeniería de Software Asistida por Computadora

Un lenguaje de cuarta generación es la entrada directa a un intérprete de L4G. Este traduce el código en L4G a código ejecutable. La entrada de generador de código es un lenguaje de especificación de procedimientos (LEP). Este es procesado por uno o varios módulos de generación de códigos que traducen el LEP al lenguaje de programación adecuada.

Un generador de aplicaciones utiliza una base de datos central o un diccionario de datos, características seleccionadas mediante menús interactivos y reglas específicas de la aplicación para crear software que pueda ser utilizado en un campo de aplicación limitada.

HERRAMIENTAS DE PROGRAMACIÓN ORIENTADAS A LOS OBJETOS.

La programación orientada a los objetos es una de las tecnologías más actuales de la ingeniería de software.

Los entornos de programación orientados a los objetos suelen estar unidos al lenguaje de programación específicos (p. ej. C++, Eiffel, Objective-C o Smalltalk). Un entorno O-O típico incorpora características de los interfases de tercera generación (ratón, ventanas, menús desplegables, operaciones sensibles al contexto, multitarea) con funciones especializadas como las del "inspector" -una función que permite al ingeniero de software examinar todos los objetos contenidos en unas bibliotecas de objetos para determinar si pueden ser reutilizados en la aplicación actual.

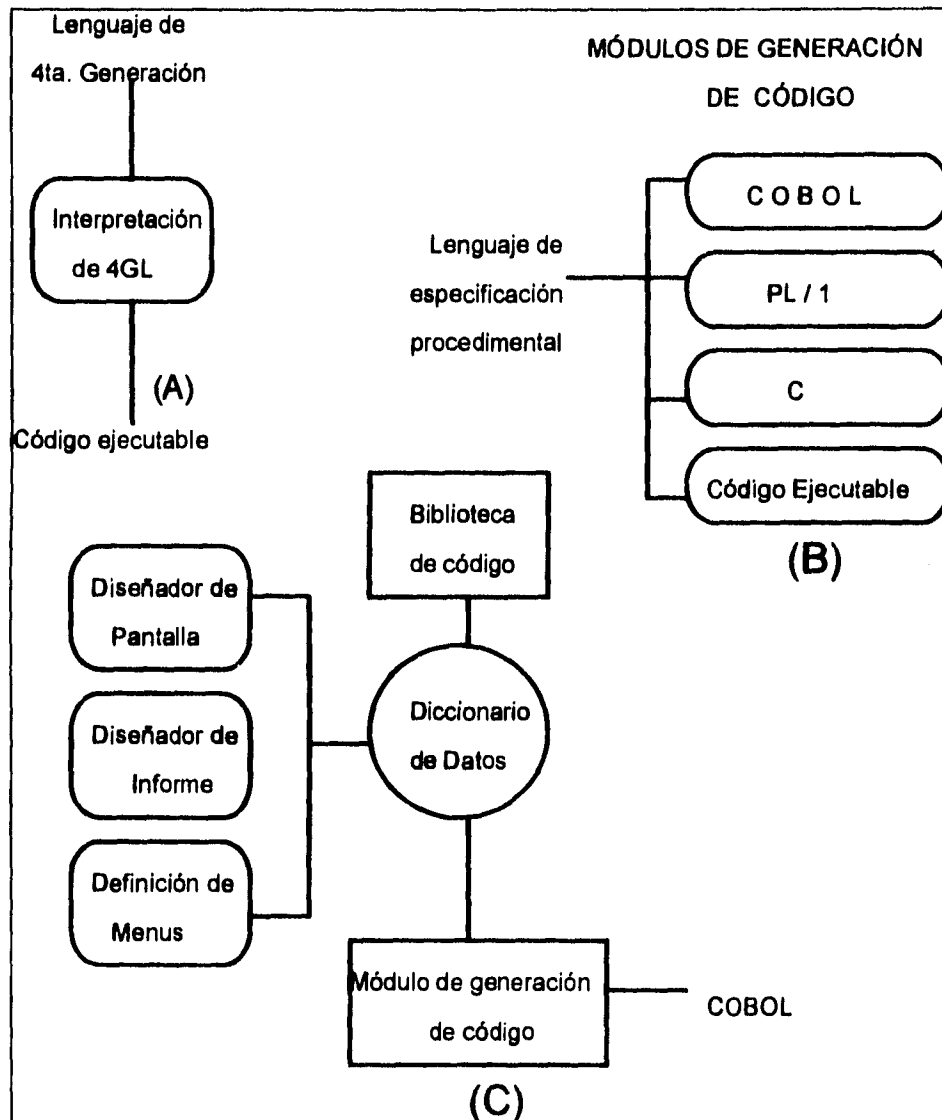


Fig. I.1.3.3. Herramientas de cuarta generación . A) 4GL b) generación de código, c) generador de aplicaciones.

HERRAMIENTAS DE INTEGRACIÓN Y PRUEBA.

Se definen las siguientes categorías:

- 1) Adquisición de datos; herramientas que adquieren datos para ser usados durante la prueba.
- 2) Medida estática; herramientas que analizan el código fuente sin ejecutar casos de prueba.
- 3) Medida dinámica; herramientas que analizan el código fuente durante la ejecución.
- 4) Simulación; herramientas que simulan la función del hardware o de otros elementos externos.
- 5) Realización de pruebas; herramientas que ayudan a la planificación, el desarrollo y control de las pruebas.
- 6) Herramientas de funcionalidad cruzada; herramientas que realizan varias de las funciones anteriores.

HERRAMIENTAS DE ANÁLISIS ESTÁTICO

Las herramientas de prueba estáticas ayudan al ingeniero de software a generar casos de prueba.

Las herramientas de prueba basadas en casos aceptan código fuente (LDP) como entrada y realizan varios análisis generando un conjunto de casos de prueba.

Los lenguajes de prueba especializados (p.ej; ATLAS) permiten al ingeniero de software escribir especificaciones de prueba detalladas que describen cada caso y la logística para su ejecución. Sin embargo, estas herramientas no ayudan al diseño de los casos.

Las herramientas de prueba basadas en requisitos reúnen los requisitos específicos del usuario y sugieren casos de prueba (o clases de prueba) que comprobarán estos. Para trabajar de forma apropiada, las herramientas de esta subcategoría deben tener acceso a una especificación formal del software.

HERRAMIENTAS DE ANÁLISIS DINÁMICO.

Las herramientas de prueba dinámicas interactúan con un programa en ejecución comprobando bifurcaciones, hipótesis sobre el valor de determinadas variables, y en general, monitoreando la ejecución de flujo del programa.

Las herramientas dinámicas pueden ser intrusivas o no intrusivas. Una herramienta intrusiva modifica el software que está probando, insertando trazas (instrucciones adicionales) que realizan las actividades mencionadas anteriormente. Las herramientas no intrusivas utilizan un procesador de hardware aparte de que se ejecutan en paralelo con el procesador que contiene el programa que está siendo verificado.

La mayoría de las herramientas de la categoría de análisis dinámico producen informes que indican el número de veces que determinados bloques de instrucciones se han ejecutado (análisis de ramificación) y su tiempo medio de ejecución (análisis de prestaciones).

HERRAMIENTAS DE REALIZACIÓN DE PRUEBAS.

Las herramientas de gestión de pruebas se utilizan para controlar y coordinar las pruebas del software en cada una de sus etapas principales. Las herramientas de esta categoría gestionan y coordinan las pruebas de regresión, realizan comparaciones para encontrar diferencias entre la salida esperada y la actual, realizan pruebas no interactivas de programas con interfaces hombre/máquina, estas sirven como controladores genéricos de pruebas.

HERRAMIENTAS DE CREACIÓN DE PROTOTIPOS.

Las herramientas de diseño de pantallas permiten a los ingenieros de software definir, de forma rápida, formatos de presentación para aplicaciones interactivas. En algunos casos, estas herramientas pueden generar el código fuente para configurar la pantalla.

Herramientas CASE más sofisticadas permiten crear un diseño de datos, incluyendo el diseño de la pantalla y el formato de los informes.

Muchas herramientas de análisis y diseño tienen extensiones que incluyen la opción de crear prototipos.

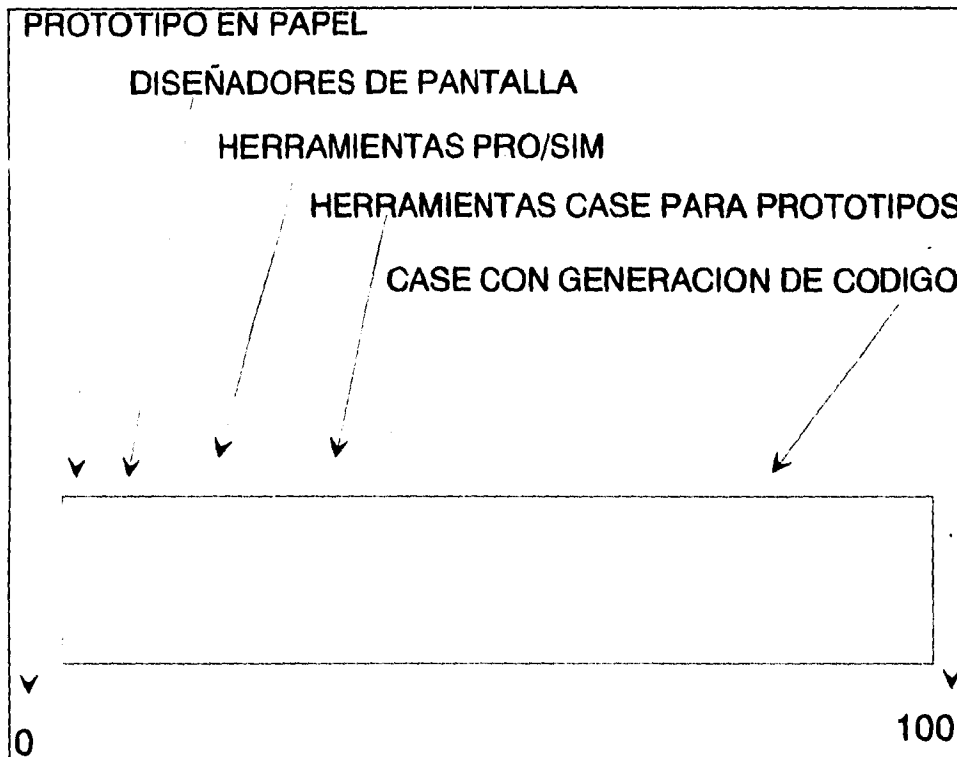


fig. I.1.3.4. Herramientas para prototipos.

HERRAMIENTAS DE MANTENIMIENTO.

Las herramientas CASE para el mantenimiento de software abarcan una actividad que actualmente ocupa, aproximadamente, el 70% del esfuerzo total dedicado al software.

La categoría de herramientas de mantenimiento puede subdividirse de la siguiente forma:

- 1) Herramientas de ingeniería inversa a especificaciones.- Toman el código fuente como entrada y generan modelos de diseño y análisis estructurado, listas de utilización y otra información relacionada con el diseño.
- 2) Herramientas de reestructuración y análisis de código.- Analizan la sintaxis del programa, generan un diagrama de flujo de control y un programa estructurado.
- 3) Herramientas interactivas de reingeniería del sistema.- Se utilizan para modificar sistemas de bases de datos (p.e. convertir archivos IDMS o DB2 al formato entidad-relación).

HERRAMIENTAS DE INGENIERÍA INVERSA.

Las herramientas de ingeniería inversa realizan el análisis de un programa una vez que este ha sido concluido.

Al igual que las herramientas de prueba, las de ingeniería inversa se pueden clasificar en estáticas o dinámicas.

Una herramienta estática de ingeniería inversa utiliza como entrada el programa fuente para extraer y analizar su arquitectura, su estructura de control, el flujo lógico y el flujo de datos.

Las herramientas dinámicas monitorean el software durante su ejecución y utilizan la información obtenida para construir un modelo de comportamiento del programa.

HERRAMIENTAS DE REINGENIERIA.

Aunque este tipo de herramientas son muy prometedoras, existen muy pocas en uso con calidad industrial, aceptan como entrada código fuente sin estructurar, realizan el análisis de ingeniería inversa y reestructuran el código ajustándolo a los conceptos modernos de programación estructurada.

Las herramientas de reingeniería de datos trabajan en el otro extremo del espectro de diseño. Analizan las definiciones de los datos o una base de datos descrita en un lenguaje de programación (normalmente cobol) o en lenguaje de descripción de base de datos.

HERRAMIENTAS DE ESTRUCTURA.

Las herramientas de estructura son herramientas que permiten la gestión de base de datos, gestión de configuraciones y la integración de otras herramientas CASE; constituyen el punto de partida en la dirección del IPSE.

Las herramientas de esta categoría tienen componentes funcionales para el tratamiento de datos, de interfaces y con capacidad de integración con otras herramientas. La mayoría trabajan con base de datos orientadas a los objetos, y con un conjunto interno de herramientas para establecer interfaces con herramientas de otros distribuidores del sistema CASE.

CASE E INTELIGENCIA ARTIFICIAL

Algunas herramientas CASE incorporan sistemas expertos, pero la gran mayoría hace uso muy reducido de las técnicas de inteligencia artificial. Muchas de las que si lo hacen utilizan esta tecnología para la validación gráfica de modelos de diseño y análisis, aplicando reglas de diseño específicas de un método a los modelos creados por el ingeniero de software.

Los investigadores están evaluando entornos de programación que hacen uso de "agentes" de diseño y evaluación, herramientas inteligentes que ayudan en el análisis, el diseño y la prueba de sistemas basados en computadora.

En lugar de evaluar simplemente el modelo que creó una persona, un agente ayudará al ingeniero en su labor de resolución de problemas.

Estos agentes deben ser específicos de un campo y acceder a una base de conocimiento que contenga las características de una clase limitada de aplicaciones y sea capaz de utilizar esta base de conocimiento para guiar al ingeniero en el análisis, el diseño y las pruebas.

Se concluye que las herramientas de ingeniería del software asistida por computadora abarcan cada etapa del proceso de ingeniería y cada actividad que se desarrolla a lo largo del mismo.

Capítulo I Ingeniería de Software Asistida por Computadora

El CASE está formado por un conjunto de bloques que comienzan en el nivel del Hardware y del Sistema operativo y acaban en cada una de las herramientas .

Las categorías de herramientas CASE incluyen las actividades técnicas y , así como la mayoría de las áreas de aplicación del software, cada categoría de herramientas se ha considerado como una "solución puntual".

Con el paso de los años el CASE ser parte del tejido de la ingeniería del software.

Al igual que los ingenieros mecánicos y eléctricos se basan en el CAD/CAE/CIM para el análisis y el diseño de productos de alta tecnología, los ingenieros de software se basarán en el CASE para el análisis, el diseño y las pruebas de los sistemas basados en computadora del siglo veintiuno.

I.1.4. FILOSOFÍA

I.1.4.1 ¿Porque usar las técnicas estructuradas y herramientas CASE?

Al iniciar los días del análisis y la programación de los sistemas, existían pocas reglas, se contaba solamente con los lenguajes de programación. Los pioneros de la programación, creaban sus propios métodos, y muchos de ellos eran por lo regular ineficientes, ocasionando infinidad de problemas.

El análisis y la programación de sistemas, han cambiado desde las metodologías iniciales, hasta los métodos "ad-hoc" a la disciplina (procesos de entrada-proceso-salida). Las técnicas estructuradas representan una búsqueda para disciplinar, el análisis y la programación; éstas técnicas no son estáticas, se encuentran en continua evolución a mejores formas.

Los objetivos primarios de las técnicas estructuradas son las siguientes:

- a. Realizar programas de alta calidad de funcionamiento predecible
- b. Realizar programas de fácil mantenimiento
- c. Simplificar programas y el proceso de desarrollo de programas
- d. Obtener un mayor control y posible predicción durante el desarrollo de los procesos
- e. Rapidez en el desarrollo de los sistemas
- f. Disminuir el costo del desarrollo de un sistema

Para poder realizar eficientemente los objetivos anteriores a continuación se enumeran algunos puntos que ayudarán a lograrlo.

Objetivos técnicos

- a. Identificar el problema y desintegrarlo en partes simples.
Las funciones complejas son desintegradas en funciones de bajo nivel, éstas descomponerlas en funciones simples y continuar. Debemos separar el diseño en módulos jerarquizados, definiendo con precisión los intereses entre ellos.
- b. Simplicidad en el diseño.
Un buen diseño, usualmente es sencillo, con interfases claras entre sus módulos; un mal diseño es complejo, enredado en sus interacciones.
- c. Complejidad controlada.
En computación se desarrollan sistemas en los cuales se incrementara la complejidad. Al controlar la complejidad de un sistema estamos realizando una minimización en el número de interacciones entre módulos separados y estandarizados de la estructura de control.
- d. Tener un pensamiento claro acerca del sistema y sus programas.
- e. Usar técnicas de diagramación que sean lo más claras posibles.
Tener buenos diagramas son de gran ayuda para comprender la complejidad del problema.
- f. Mejorar la readaptabilidad del diagramas y códigos.
Una persona debe ser capaz de comprender los diseños y códigos creados. El personal de mantenimiento también necesita tener una fácil comprensión del sistema. Estandarizar el estilo de programación.
- g. Tener comunicación con los usuarios terminales.
Muchos sistemas tienen fallas u operan pobremente porque no se conocen los requerimientos de los usuarios.

Las técnicas usadas deben ser capaces de simular los requerimientos de usuarios terminales y además checar el diseño durante su implementación.

- h. Realizar unidades de estructuras.

Un buen diseño es claro con estructuras unificadas. Similares principios de diseño y aplicación de construcciones para el sistema.

- i. Empleo consistente, de un método dócil.

El análisis, diseño y técnicas de programación necesarios para una enseñanza fácil.

- j. Emplear un conjunto estándar de estructuras de control para poder convertir en código con un mínimo esfuerzo.

Los lenguajes de 4G facilitan la representación de estructuras de control con mucho menos trabajo que los lenguajes de 3G.

- k. Tener comunicación precisa entre la gente de desarrollo.

Una de las fuentes más grande de problemas son los desacuerdos entre desarrolladores y programadores. La técnica debe minimizar las necesidades de interacción a través de representaciones formales preferentemente computarizadas.

- l. Minimizar el número de desarrolladores en un equipo.

Intentar equipos de una persona si es posible.

- m. Usar sistemas que trabajen tanto en sistemas grandes como medianos.

Evolución

Las técnicas estructuradas evolucionaron desde una metodología codificada (programación estructurada) a técnicas que incluyen análisis, diseño y metodologías de examen, también como conceptos de dirección de proyectos y herramientas de documentación.

Las técnicas estructuradas se desarrollaron para ser un paso, a través del cambio de desarrollo de software, desde una destreza manual a una disciplina ingenieril. En un sentido estricto es más una actitud que una metodología.

Asumimos una importancia fundamental con la llegada de las herramientas CASE. Fue esencial para llevar a cabo ayudas automatizadas para el diseño de sistemas, verificación computarizada del diseño y generación automatizada del código desde el diseño.

Las técnicas estructuradas fueron introducidas a la comunidad académica a finales de los 60's y empezó a conocerse en la industria a principios de los 70's, después del trabajo realizado por Terry Baker y Harlan Mills en IBM, para el New York Times. Para finales de los 70's las técnicas estructuradas han evolucionado en un conjunto de tecnologías que abarcan todo el ciclo de vida del software. Ver figura I.1.4.2.1.

Programación Estructurada

Al principio las técnicas estructuradas se enfocaban al programa. Las preguntas se dirigían a la forma del programa. ¿Qué debería un programa observar?, ¿Cuál es la relación entre su forma estática y sus estructuras dinámicas de tiempo y ejecución. ¿Cómo podemos codificar un programa para hacerlo más comprensible?, ¿Cómo controlaremos la complejidad en un programa que crece de tamaño?.

La estandarización de las formas fue la llave. La notación de la estandarización fue aplicada para la construcción de programas de control y módulos de programas.

Diseño Estructurado

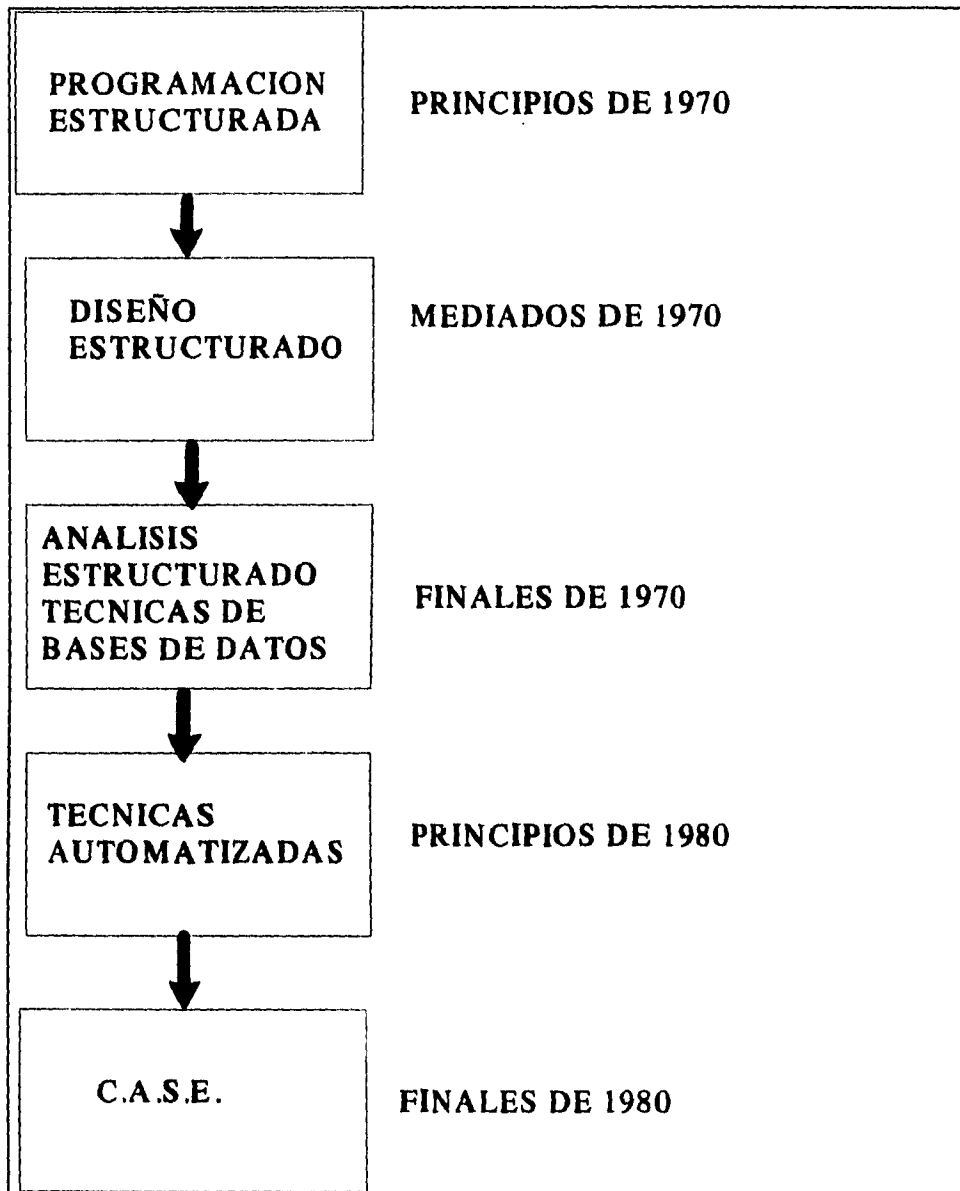
Para mediados de los 70's la filosofía estructurada se difundió a las fases del diseño. La notación de estandarización fue aplicada a los procesos de solución de problemas como un camino para introducir organización y disciplina dentro del diseño de programas.

Considerando que en sus principios, las técnicas estructuradas se concentraron en un detallado nivel de instrucciones, Ahora visto desde un programa, el diseño estructurado se enfocó en un alto nivel de vista del programa, usando el programa modular como la construcción como una construcción básica de bloques. La relación entre el problema y la solución programada fue estudiada. El concepto de modularización fue refinado por la estandarización de la estructura de un módulo y definiendo cualidades métricas del programa.

Análisis Estructurado

Cuando empezó a parecer que muchos problemas de software surgieron por una pobre definición de requerimientos, falta de una adecuada atención a la fase del análisis, las técnicas estructuradas para el análisis de sistemas y especificaciones de requerimientos empezaron a desarrollarse (a principios de los 70's). El uso de diagramas de flujo de datos se dispersó ampliamente y empezaron a desarrollarse las técnicas de base de datos.

Diferentes técnicas estructuradas han sido introducidas. Las técnicas estructuradas iniciaron como una mezcla de metodología, estrategias y herramientas que ofrecían un mejoramiento sistemático para el desarrollo del software.



La evolución de las técnicas estructuradas dentro de la tecnología CASE
Fig. I.1.4.1.1.

Ingeniería de Información

La ingeniería de información se aplicó como técnica estructurada para todo un desarrollo, no para un sistema. Es una ayuda para crear un marco de trabajo dentro del cual pueda ser diseñado por separado convenientemente un sistema. El marco de trabajo consiste de un modelo de datos estructurados y modelos de desarrollo y sus procesos.

Técnicas Automatizadas

A principios de los 80's la baja en la productividad en programación alcanzó una crisis. Las computadoras y en particular las microcomputadoras se dispersaron ampliamente, provocado una drástica reducción en los precios. Muchos usuarios terminales empezaron a adquirir una cultura computacional y empezaron a exigir más aplicaciones. El procesamiento de datos usó las metodologías de programación estructurada, diseño estructurado y análisis estructurado, conforme crecían se llegó a tener problemas con el mantenimiento. Al buscar mayor productividad permitió el desarrollo de nuevos lenguajes generados de código, todo en continuo hacia una evolución hasta llegara una herramienta completa denominada CASE, en la cual se incluyen las metodologías y técnicas antes vistas.

I.1.4.2. FILOSOFÍA DE LAS TÉCNICAS ESTRUCTURADAS

La filosofía estructurada empezó en el mundo de la programación básicamente con los cuatro puntos siguientes:

Principio de Abstracción

Para solucionar un problema , debemos separar los aspectos ligados a una realidad particular para representar en el problema en una forma general simplificada.

Principio de Formalidad

Apoyarse de una metodología probada, para solucionar un problema.

Concepto de dividir y vencer

Solucionar un problema difícil mediante la división del problema en conjuntos de pequeños e independientes problemas, los cuales sean fáciles de comprender y solucionar.

Concepto de la ordenación jerárquica

Organizar los componentes de la solución en una estructura de árbol jerárquica. Entonces la solución puede comprenderse y construirse nivel por nivel, en donde en cada nivel puede ser agregado un mayor detalle.

La filosofía originalmente se aplicó para la programación la cual representaba una importancia como la técnica para crear sistemas de cambio. Estos fueron los principios para estar en posibilidad de manejar la *complejidad*, ya sea en el desarrollo de software, diseño de chip's o en el diseño de sistemas complejos. Esto se relaciona con la introducción de la estandarización y disciplina del diseño de los procesos y la forma final del diseño. También se buscó mejorar la dirección del desarrollo de los sistemas, los procesos del desarrollo de los sistemas y el resultado de los sistemas, a través de procedimientos bien definidos, herramientas, técnicas, control el proyecto y mecanismos de comunicación.

PRINCIPIOS BÁSICOS

La filosofía de las estructuras básicas formalizó a las buenas prácticas de programación. La filosofía estructurada original se compone de las siguientes estrategias para solucionar problemas:

- a. Principio de abstracción
- b. Principio de formalidad
- c. Concepto de dividir y vencer
- d. Concepto de la ordenación jerárquica

Principio de abstracción:

Sin el principio de abstracción, la revolución con las técnicas estructuradas nunca podría haber sucedido. La *Abstracción* es la concepción o visualización de algo que existe en su realidad y que deseamos traerla a la nuestra. La manejamos como una simplificación de factores que describen que está sucediendo sin explicarnos como es que se sucede. Con la abstracción nos podemos imaginar la solución de un problema sin preocuparnos de restricciones o detalles irrelevantes alrededor del problema. Algunas de las más creativas y mejores soluciones se derivan de ésta manera. Usando el principio de abstracción, podemos ver un programa como niveles o estratos.

El nivel más alto, nos muestra el nivel más abstracto (complejo), mientras que el nivel más bajo nos muestra un detalle muy particular del problema. Podemos movernos a través de los niveles y así poder ver cada componente.

Cada nivel de abstracción es compuesto por un grupo de funciones consideradas lógicamente relacionadas. Las funciones pertenecientes a un mismo nivel y a diferentes niveles admiten comunicación de una manera restringida. Existen dos reglas de comunicación:

1. Estratos inferiores no tienen conciencia de si, y en consecuencia no tienen acceso a niveles superiores; por el contrario niveles superiores pueden invocar a niveles inferiores para que realicen tareas.
2. Cada nivel posee sus propios datos, los cuales son exclusivos y no pueden ser accedidos por otros niveles. Los datos son transferidos de un estrato a otro por petición explícita. (Cada nivel puede acceder a las áreas comunes, denominadas estructuras de base de datos).

Principio de Formalidad

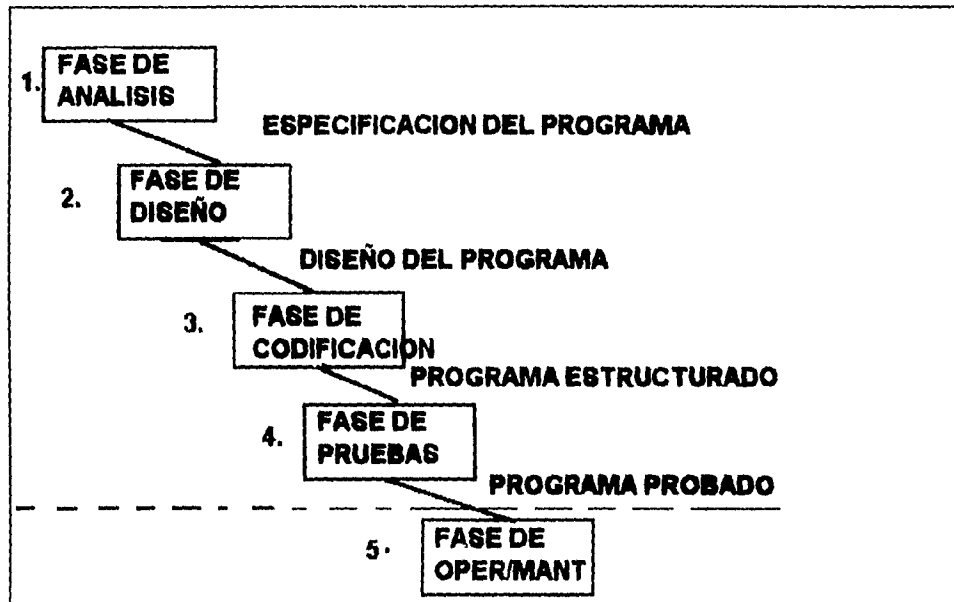
El segundo principio fundamental de la filosofía estructurada es el "Principio de la Formalidad". La palabra formalidad sugiere un método riguroso de acercamiento. Esto es precisamente lo que implica la filosofía estructurada.

El principio de la formalidad es la base para movernos de una programación por intuición a una disciplina dentro de la ingeniería. Sin la formalidad no se tendrían bases para implantar un software para controlar proyectos ni productos de control de calidad. Introduciendo la formalidad dentro de la programación significa que el programa desarrollado es guiado mediante una sistematización, procedimientos de paso a paso, además que el resultado producido por cada procedimiento puede ser definido rigurosamente, y una evaluación por pasos es incluida para determinar si el procedimiento ha sido ejecutado correctamente y sobre todo, si el resultado producido es el correcto.

Todo esto no significa que la programación es reducida a una repetición mecánica de procedimientos en la cual ninguna solución necesita de la creatividad. Al introducir la formalidad dentro de cualquier proceso, frecuentemente se presenta oposición en el ámbito, argumentando que se sofoca a la creatividad; lo cierto es que la filosofía estructurada usa la formalidad para nutrir la creatividad a través de la disciplina.

1

La formalidad es una manera positiva y poderosa de combinar el oficio y aspectos de la ingeniería de programación con el objetivo de clarificar la comunicación y para concentrar la creatividad.



El ciclo de vida del software es un proceso múltiple que consiste de cinco fases básicas

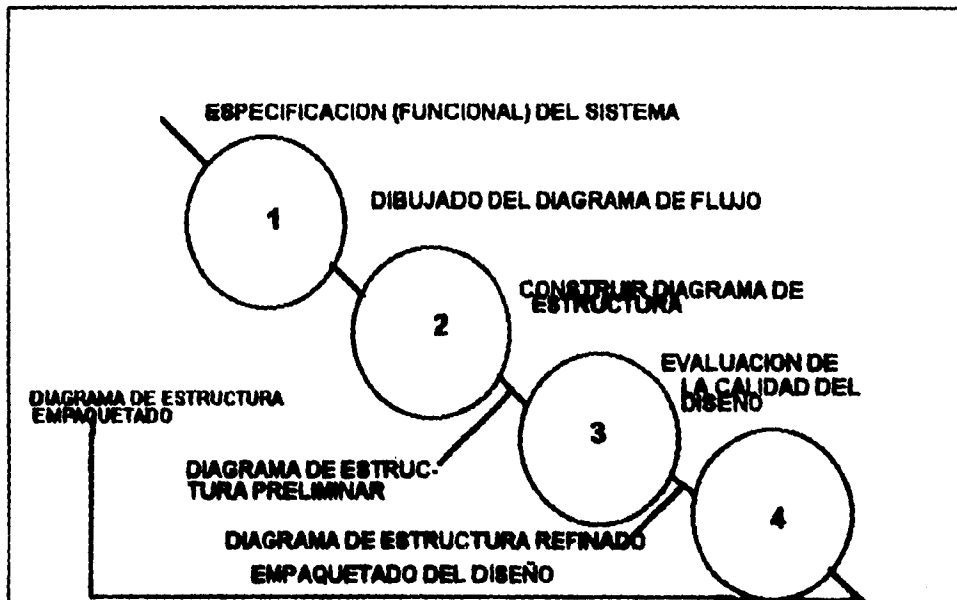
Fig. I.1.4.2.1.

2

El ciclo de vida del software es un ejemplo de la aplicación del principio de la formalidad para la programación. El ciclo define los procesos de programación como una secuencia de fases, como se muestra a continuación:

Para cada fase del ciclo de vida se detallan paso a paso los procedimientos definidos. Por ejemplo: existe una variedad de metodologías de diseño estructurado, cada una con sus propios procedimientos. A continuación se muestran los cuatro pasos básicos del diseño en la metodología de diseño estructurado de Yourdon.

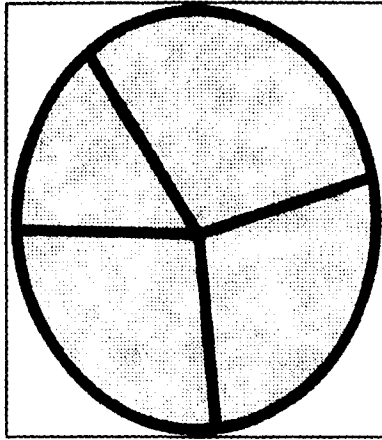
3



Los procesos para el diseño de programas según la metodología Yourdon
Fig. 1.1.4.2.2

Concepto de Dividir y Vencer

Es el concepto para solucionar problemas difíciles mediante la división de éste en pequeñas partes. Independientemente del problema, sus partes deben ser fáciles de entender y solucionar. Es una herramienta poderosa y esencial para el manejo de la complejidad. Permite a los desarrolladores de software trabajar bajo cierta confidencialidad en una parte del sistema a solucionar sin preocuparlos acerca del enorme número de detalles que conforman al sistema. El concepto de dividir y vencer ha sido utilizado desde los tiempos de Julio Cesar. Cuando es aplicado el concepto a un sistema de software, significa particionar un programa en pequeñas piezas, manejables y examinables por separado, donde cada pieza puede ser desarrollada por una persona, éste tipo de trabajo se denomina *modular* y al hecho de poder revisar cada parte por separado, *refinamiento por pasos* (*Stepwise refinement*).



Dividir un problema en partes, fáciles de entender y solucionar. Es usado para simplificar la programación (modularización) y los procesos de éste (refinamiento por pasos).

Fig. 1.1.4.2.3.

Concepto de Ordenación Jerárquica

El concepto de Ordenación Jerárquica, está muy relacionado con el concepto anterior. Siempre debemos arreglar los módulos de los programas en un árbol jerárquico, la cual es una estructura fácil de comprender.

Los programas pueden ser entendidos y construidos nivel por nivel, en donde cada nivel que se agregue significa mayor detalle. La organización jerárquica, hace posible el manejo del control, delegación y comunicación del problema, el cual puede ser tan complejo que involucre a muchas partes.

Los conceptos de dividir y vencer así como, el de ordenación jerárquica, son los fundamentos para la formulación de otras metodologías, tales como la programación "Top Down", "chief programmer team" y metodologías de diseño estructurado. Aunque teóricamente podemos manejar los conceptos de dividir y vencer y ordenación jerárquica, éstas fueron instrumentadas para programación avanzada e ir más allá de una herramienta menos poderosa como el método de programación modular.

Una de las diferencias esenciales entre la programación estructurada y la no estructurada es que la primera, está ordenada jerárquicamente mientras que la segunda está ordenada sólo secuencialmente. Una estructura jerarquizada, nos permite eliminar estratos superiores y no afectar en lo más mínimo a los estratos inferiores. También podemos remover o transferir bloques a otros sistemas, con lo cual creamos una poderosa herramienta para la creación de programas que sean fáciles de cambiar.

I.1.4.3. PRINCIPIOS BÁSICOS DE LA INGENIERÍA DE SOFTWARE

Empezaremos por distinguir entre las Técnicas Estructuradas y la Ingeniería de Software. La Ingeniería de Software es el estudio de los principios y sus aplicaciones para el desarrollo y mantenimiento de los sistemas de software.

La Ingeniería de software y las técnicas estructuradas son una colección de metodologías y herramientas, donde ambas comparten muchas ideas y principios, aunque la I. S. es una disciplina más amplia que las T.E. La I.S. es una disciplina más amplia que las T.E. porque la I.S. incluye las metodologías estructuradas y la colección de técnicas estructuradas, así como, otras herramientas y metodologías de software.

La Ingeniería de software se fundamenta en los principios que se mostraron en el cuadro anterior, en adición a este se expande esta lista al cuadro siguiente:

Principio de Ocultamiento

Esconder información esencial. Habilitar un módulo para ver solamente la información necesaria.

Principio de Localización

Ubicar cerca a los objetos relacionados lógicamente.

Principio de Integridad Conceptual

Seguir una arquitectura de diseño consistente.

Principio de Totalidad

Checar que de todo el sistema realizado, nada se haya olvidado.

Principio de Independencia Lógica

En análisis y diseño, concentrarse en las funciones lógicas ha ser acopladas, independientemente de la implementación física.

El *principio de ocultamiento* es usado para guiar de los procesos de descomposición funcional. Para controlar la complejidad un sistema debe ser modularizado y las interfases entre módulos deben ser tan simples como sea posible. Esencial el principio de ocultamiento hace a un sistema fácil de comprender debido al manejo de información estrictamente necesaria para el módulo, además también facilita el mantenimiento porque sólo se aboca al módulo afectado.

El *principio de localización* es conceptualizado como el agrupamiento siempre cercano de las relaciones lógicas de objetos. Lo aplicamos a datos y pasos de procesos. Arreglos, registros, subrutinas y procedimientos son candidatos para aplicar el principio. Evitar el uso de GOTO'S en un código bien estructurado ayuda a el principio.

El *principio de integridad conceptual* es quizás el principio más importante de la ingeniería de software. Cuando es combinado con el principio de abstracción y el de ordenación jerárquica, el principio de integridad conceptual nos da la habilidad para manejar la complejidad. El principio de integridad conceptual acentúa el valor de un plan de diseño consistente seguido durante la construcción de cada componente de un sistema. Como resultado de seguir el principio de integridad conceptual es una arquitectura de sistemas unificada. Con esto se promueve la cualidad de comprensión más que cualquier otro factor.

El *principio de totalidad* implica que cada cosa ha sido incluida y nada se ha olvidado. En un sistema complejo que involucra muchos componentes, es factible que ocurra una omisión. El principio de totalidad dirige la atención a la importancia de checar que el sistema se encuentre completo en todos sus requerimientos.

I.1.5 TÉCNICAS Y METODOLOGÍAS ESTRUCTURADAS QUE SOPORTAN LAS HERRAMIENTAS CASE

OBJETIVO:

Describir las técnicas y metodologías estructuradas cuya función es ayudar al desarrollo de sistemas mediante herramientas CASE.

La función de un CASE es soportar cada fase de los procesos de desarrollo y mantenimiento del software con un conjunto de herramientas integradas. Las herramientas CASE automatizan la producción de la documentación requerida en cada fase del desarrollo y del mantenimiento de un sistema. En las primeras fases del ciclo de vida de un sistema, las herramientas CASE, así como los lenguajes de generación de pantallas y de especificaciones ejecutables, se emplean para construir modelos, prototipos y simulaciones para establecer los requerimientos del sistema y comprobar el comportamiento del mismo. En la fase de implementación las herramientas CASE, así como los generadores de código y de base de datos, se utilizan para escribir automáticamente programas documentados a partir de la especificación de diseño de alto nivel.

Otra parte importante del soporte del ciclo de vida de un sistema es apoyarse en las metodologías que estructuradas que realizan los procesos dentro del ciclo de vida de un sistema. Un CASE soporta el empleo de metodologías estructuradas automatizando la producción de la documentación requerida por la metodología y guiando al usuario en la correcta utilización de la metodología.

Analizaremos como el CASE soporta tanto los procesos del ciclo de vida de un sistema y las metodologías estructuradas con los siguientes cuatro puntos:

- La cobertura del ciclo del vida del software.
- El soporte de prototipos.
- El soporte de metodologías estructuradas.
- La generación automática de código.

IMPORTANCIA DE LAS PRIMERAS FASES DEL CICLO DE VIDA

La importancia de la tecnología CASE, se centra en las primeras fases del ciclo de vida del software. Esto inicia con el reconocimiento de que las etapas de análisis y diseño son las más críticas del ciclo de vida. Los errores en las especificaciones pueden ser muy costosos si no se detectan y corrigen durante las primeras fases. La corrección de un error de especificación durante la fase de mantenimiento puede llegar a ser 100 veces más caro que si se hubiera corregido durante la fase de análisis. La integridad y exactitud de las especificaciones del sistema afectan el éxito del desarrollo de todos los programas. La especificación es la base de la planificación y asignaciones del proyecto, de la planificación de las pruebas, de la documentación del usuario y del diseño del programa. Una comprensión deficiente del sistema es la causa de las fallas del software.

Los errores de diseño a menudo dominan los proyectos de software dependiendo de su número y costo de corrección, especialmente cuando no se detectan pronto. En los proyectos grandes, los errores de diseño superan a menudo los errores de código, estadísticamente se habla de un 60% de errores de diseño contra un 30% de los errores de código. Cuanto más cuidado se preste al diseño menos costos y más confiable será el sistema. El diseño del sistema es la guía de su implantación si la guía no existe ó es incorrecta el sistema producido probablemente estará mal organizado, documentado y su mantenimiento será difícil.

EL SOPORTE DE LA METODOLOGÍA ESTRUCTURADA.

El case soporta metodologías estructuradas como análisis estructurado, diseño estructurado y programación estructurada al proporcionar herramientas para automatizar las metodologías. hay dos niveles de automatización:

- 1.- La preparación de la documentación.
- 2.- Los pasos del proceso de la metodología.

La preparación automatizada de la documentación significa proporcionar soporte gráfico para el dibujo de diagramas estructurados (diagramas de flujo, de transición de estado y de acción, de entidad /relación). También significa la generación automática de especificaciones textuales, miniespecificaciones y pseudocódigo. Las especificaciones textuales se emplean para proporcionar un información más detallada sobre los procedimientos del programa y de la estructura de los datos referenciados en los diagramas estructurados de un nivel más alto.

Como las diferentes metodologías estructuradas utilizan diagramas distintos para modelar y documentar un sistema, las metodologías estructuradas soportadas por un Case en particular, estarán determinadas por los tipos de diagramas estructurados y por los estándares que pueda ofrecer. La metodología de diseño estructurado de Yordan emplea un diagrama de estructura (structure chart) derivado de un diagrama de flujo para representar el diseño del programa; la metodología de diseño de Jackson utiliza el diagrama estructurado en árbol y la metodología de desarrollo de sistemas de datos estructurados de Orr, emplea los diagramas de Warnier- Orr.

El segundo nivel de soporte de metodología, la automatización de los pasos del proceso de la metodología, significa que el CASE guía al usuario en el correcto uso de una metodología estructurada. Esto requiere que al menos cierto nivel de comprensión de la metodología este incorporado en las herramientas del CASE. Esto puede ser tan simple como incorporar paneles de ayuda para describir cada paso de la metodología o de listas de comprobación (check list) que incluya las entradas y salidas requeridas en cada paso. Podría incluir un mecanismo de comprobación para asegurar que cada salida requerida por la metodología esta presente, correcta y completa antes de permitir al usuario ir al paso siguiente.

Al usuario no solamente se le guía, sino que, además, se le forza a seguir los pasos con un orden y método normalizado. El propósito, en definitiva es normalizar y sistematizar el proceso de desarrollo del software.

El segundo nivel de automatización de la metodología estructurada se encuentra incorporado dentro de la metodología CASE ya que mediante él, se introduce un control sobre el proceso del desarrollo del software y la normalización impuesta por la herramienta CASE. Además, puede realizarse un control más completo y exhaustivo sobre la calidad e integridad, pues muchas comprobaciones de calidad son específicas de la metodología. Por ejemplo las medidas de acoplamiento y cohesión son comprobaciones de diseño que forman parte solamente de la metodología de diseño estructurado de Yordan. En general, cuanto más puede definirse específicamente, mayor efectividad habrá para la automatización del software.

EL DIRECTOR DE LA METODOLOGÍA.

El segundo componente de un shell inteligente es el director de metodología. Su propósito es introducir el conocimiento sobre las metodologías de desarrollo de software en las herramientas de software.

Hay tres niveles de conocimiento que pueden ser soportados por la automatización del software:

1. Documentación
2. Tutela.
3. Proceso.

El conocimiento a nivel de documentación.

Las herramientas de software que automatizan la producción de documentación requerida por una metodología de desarrollo de software residen en el primer nivel. Los ejemplos incluyen las herramientas CASE de diagramación y documentación que soportan los símbolos gráficos y los convenios de notación requeridos por las distintas metodologías estructuradas (por ejemplo, el diagrama de flujo de datos de DeMarco utilizado para representar la especificación de un sistema en la metodología estructurada de análisis).

El conocimiento a nivel de Tutela.

A nivel de tutela, las herramientas de software guían al programador en el uso de una metodología de desarrollo de software. Técnicas tales como los paneles de ayuda, las listas de control, los menús dependientes del contexto y los mecanismos de comprobación se emplean para indicar al usuario la utilización correcta de la metodología. Estas herramientas proporcionan información dependiente del contexto sobre cómo empezar, qué hacer a continuación, qué entradas y salidas se requieren en cada paso de la metodología y cómo comprobar la calidad del sistema que se está desarrollando.

El conocimiento a nivel de proceso.

Las metodologías de automatización del software son el componente más fundamental de la automatización del software, porque es la metodología de software la que guía el proceso de desarrollo. La experiencia ha demostrado que la clave del éxito de un proyecto de software es la elección correcta de la metodología (que pueda conducir al programador a desarrollar un buen sistema de software). La elección de la metodología adecuada es más importante que utilizar las mejores y más potentes herramientas.

Las herramientas que automatizan las disponibilidades requeridas por la metodología de software son el primer paso hacia la automatización del software. Automatizan solamente la producción de las disponibilidades requeridas por la metodología, no el proceso de producción de esas disponibilidades. Las herramientas que automatizan los pasos de los procesos de la metodología son las bases de la verdadera automatización del software.

LA TUTELA AUTOMÁTICA DEL SISTEMA

- El profesional de desarrollo debe seguir los pasos de la metodología de un modo ordenado.
- La guía se implantará vía menús que limiten las opciones validas basadas en las reglas de la metodología y en las comprobaciones de la calidad (por ejemplo, el programador no puede ir al paso siguiente de la metodología hasta que la tarea en curso esté completa y se certifique que las salidas de esa tarea están completas, correctas y cumplen con los estándares en curso).
- Con la utilización de pantallas de ayuda y menús, se informará al programador del próximo paso a dar, la entrada necesaria para realizar ese paso y la salida que se producirá.
- Cuando se realice un cambio, se informará al programador de los pasos previos que se pueden ver afectados, qué salidas se verán afectadas (y se ha cambiado automáticamente) y que pasos deben revisarse antes de permitir que el programador continúe.

Las herramientas que proporcionan el primer nivel de conocimiento de la metodología, el nivel de documentación, pueden ser genéricas; es decir, que soportan los requerimientos de documentación de varias metodologías de software. Sin embargo, por su propia naturaleza, las herramientas que procesan los niveles de guía y proceso deben corresponder a metodologías específicas. Como tales herramientas animan activamente la utilización de metodologías específicas, al programador debe comprenderse con una metodología específica.

Para automatizar los niveles más altos de la metodología del conocimiento debe integrarse en las herramientas un modelo del proceso de la metodología de software. Esto presupone:

- 1) una comprensión de cómo se lleva el proceso a la práctica;
- 2) qué modelos pueden garantizar un alto grado de productividad, control de gestión y de éxito que culminen en un sistema de software de alta calidad;
- 3) una forma de codificar el modelo en las herramientas.

MENÚ PRINCIPAL	
SECCIÓN:	1. DESCRIPCIÓN DE DATOS
	2. GENERAR APLICACIÓN
	3. DESCRIPCIÓN DEL PROGRAMA
	4. REVISIÓN DE LA ESPECIFICACIÓN
	5. SIGUIENTE FUNCIÓN
	6. ENTRADA SESIÓN
	7. ESTADO
	8. SALIR DE MENÚ
	9. AYUDA
	10. SALIDA

Los menús dependientes del contexto guían a los usuarios en el uso correcto de la metodología al seleccionar el siguiente paso en el proceso.

FIG. I.1.5.1

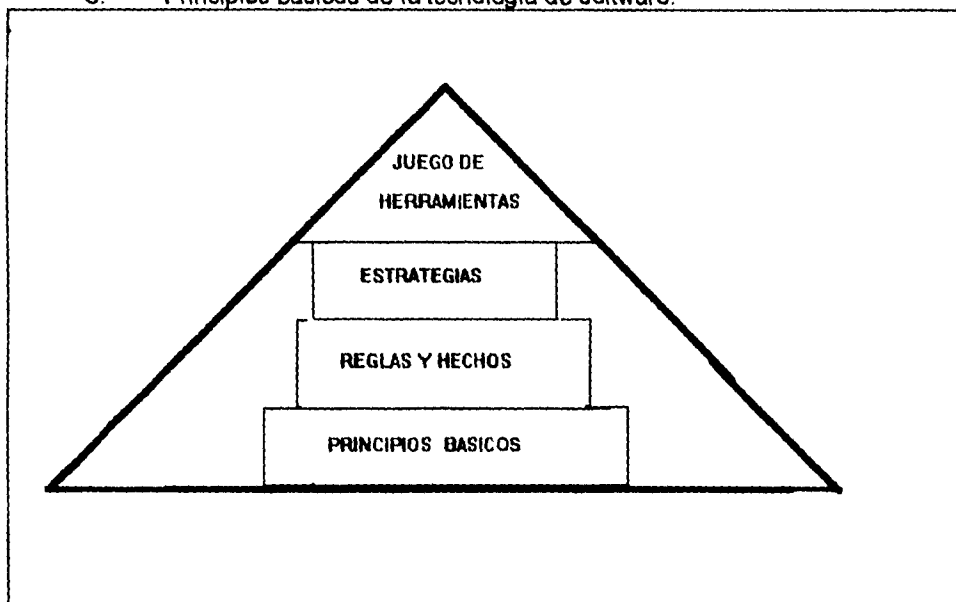
Las herramientas a nivel de guía se quedan cortas en la automatización del software, porque no incorporan un conocimiento real de la metodología que pretenden imponer. No pueden llevar adelante el proceso de la metodología por sí mismas, independientes de la guía del experto humano. Sin embargo, las del tercer nivel, el nivel de proceso, el conducto de metodología tiene esta comprensión y esta capacidad.

Es un verdadero sistema experto. Puede aconsejar como empezar, sugerir qué hacer a continuación, notificar cuándo se ha acabado, informar de lo que está equivocado y explicar por qué, sugerir un método mejor y, en muchos casos, realizar el proceso de la metodología por sí mismo. En otras partes, puede imitar el comportamiento de un experto humano en metodología del software.

LOS COMPONENTES DE UN DIRECTOR DE METODOLOGÍA.

De la figura que se muestra a continuación se presenta al director de metodología como una pirámide formada por cuatro componentes. En la cúspide hay un conjunto de herramientas de software potentes e integradas, como las herramientas de diagramación, lenguajes de especificación, herramientas de prototipos y generadores de código. La utilización de estas herramientas durante el proceso de construcción del sistema está dirigida por una base inteligente constituida por los tres niveles inferiores de la pirámide:

1. Estrategias y heurísticos del desarrollo de software.
2. Reglas y hechos de la tecnología de software.
3. Principios básicos de la tecnología de software.



El director de metodología es un sistema experto que comprende y puede llevar a cabo los pasos de una metodología de software. Consta de 4 componentes básicos. Las estrategias de software las reglas y los hechos, y los principios básicos forman la base de conocimientos que guían la utilización del conjunto de herramientas integrado y potente. fig. 1.5.1.2.

LOS PRINCIPIOS BÁSICOS.

En la base de la pirámide se encuentran los principios básicos, que son el fundamento del razonamiento de todos los sistemas expertos. Son los principios básicos de la tecnología de software que se han ido descubriendo a lo largo de la historia de la programación.

Las dos figuras anteriores exponen algunos de estos principios básicos pertenecientes a las disciplinas de las ingenierías de software y de la información, además de ser las bases de la filosofía estructurada. Empezaron en el mundo de la programación y se extendieron al diseño, el análisis y a la planificación de la información. Aunque las técnicas de creación de software cambiarán, estos principios básicos permanecerán constantes. Son principios importantes para controlar la complejidad y tratan de la introducción de la normalización y de la disciplina en los procesos y productos de software.

Divide y vencerás es uno de los principios básicos. Es el principio de la resolución de problemas difíciles, dividiendo el problema en otros más pequeños e independientes más fáciles de comprender y resolver. Es una técnica esencial y potente para tratar la complejidad. Permite al profesional del desarrollo del software trabajar con confianza en una parte de un sistema grande sin preocuparse por el enorme número de detalles que cubren el sistema. Cuando se aplica a un sistema de software significa seccionar un programa en partes independientes que son manejables y comprobables separadamente. Cuando se aplica al proceso de desarrollo del software significa seccionar el proceso en pasos manejables y comprobables separadamente. Cuando se aplica al proceso de desarrollo del software significa seccionar el proceso en pasos manejables.

El principio de la ordenación jerárquica es otro principio básico. Está relacionado con el principio de divide y vencerás. Además de descubrir que es más fácil entender un problema (y programa) si se divide en partes, la ordenación de las piezas es igualmente importante para su comprensión. Organizar siempre las partes de una estructura jerárquica en árbol mejora la comprensión. El programa puede entenderse y construirse nivel a nivel, añadiendo más detalles a cada nivel.

PRINCIPIO DE ABSTRACCIÓN:

Para resolver un problema, separar todos aquellos aspectos que estén relacionados con una realidad en particular, a fin de representar el problema de una forma general y simplificada.

PRINCIPIO DE FORMALIZACIÓN:

Seguir un procedimiento metódico y riguroso para resolver un problema.

DIVIDE Y VENCERÁS:

Para resolver un problema complejo, dividirlo en varios problemas independientes que sean más fáciles de comprender y resolver.

LOS PRINCIPIOS BÁSICOS DE LA INGENIERÍA DE SOFTWARE.

- **La ordenación jerárquica.**
Organizar los componentes de una solución es una estructura jerárquica en árbol.
- **Principio de ocultación.**
Ocultar la información innecesaria, permitiendo que un módulo conozca sólo la información que necesita.
- **Principio de colocación.**
Colocar físicamente próximos los datos relacionados lógicamente.
- **Principio de integridad conceptual.**
Seguir una filosofía y una arquitectura de desarrollo consistentes.
- **Principio de Totalidad.**
Comprobar que no falta nada.
- **Principio de independencia lógica.**
En el análisis y el diseño, concentrarse en funciones lógicas que hay que realizar, independientemente de la implantación física.

Una estructura jerárquica permite quitar los niveles superiores y tener todavía utilizables los niveles inferiores, que se pueden convertir en las piezas para construir una variante del sistema o ser la base de un nuevo sistemas. Este puede ser un mecanismo potente par la creación de programas que sean fáciles de modificar y de introducir la reutilidad como técnica de desarrollo. La característica hereditaria de los lenguajes de programación orientados al objeto es otra aplicación del principio de ordenación jerárquica.

LAS REGLAS PARA LOS MÓDULOS DE PROGRAMAS.

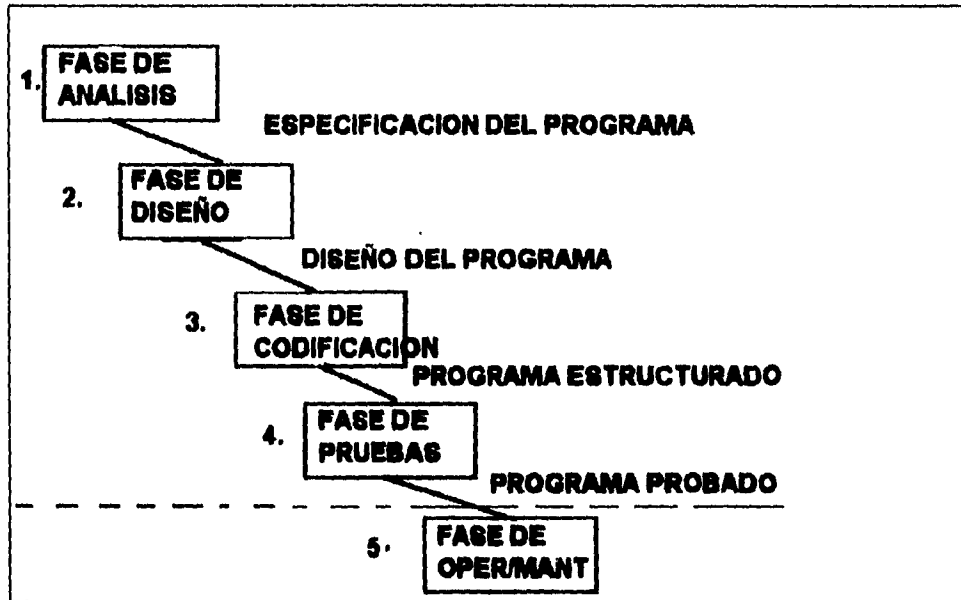
- Cada módulo representa una tarea lógica e independiente.
- Los módulos son simples.
- Los módulos son cerrados.
- Los módulos son discretos y visibles.
- Los módulos pueden probarse por separado.
- Cada módulo se implanta con una función del programa simple e independiente.
- Cada módulo tiene un solo punto de entrada y un solo punto de salida.
- Cada módulo sale a un punto de retorno estándar del módulo en el cual se ha ejecutado.
- Los módulos pueden combinarse en módulos más grandes in el conocimiento de la construcción interna de los módulos.
- Los módulos tienen interfaces bien definidos. Los módulos tienen el control de las conexiones a través de sus puntos de entrada y salida; los módulos tienen conexiones de datos a través de parámetros y de datos compartidos; los módulos tienen conexiones funcionales a través de los servicios que se realizan reciprocamente.

LAS PROPIEDADES DE UN PROGRAMA ESTRUCTURADO.

- El programa se divide en un conjunto de módulos ordenados en una jerarquía que define sus relaciones lógicas y de tiempo de ejecución.
- El flujo de ejecución de un módulo a otro se restringe a un esquema simple en el cual el control se devuelve al módulo que invoca.
- La forma del módulo es estándar y las construcciones legales de control están restringidas a la secuencia, la selección y la iteración.
- Se requiere documentación para explicar lo que hace el programa, pero no cómo lo hace.

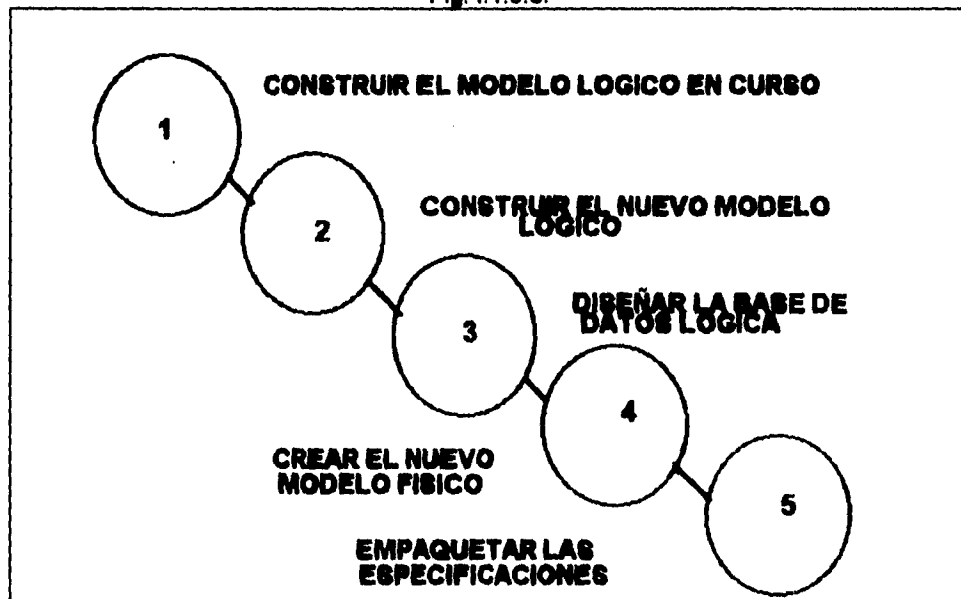
Cuando se aplican a los procesos del software, las reglas del principio de divide y vencerás toman la forma de una metodología que especifica la secuencia de los pasos seguidos para realizar una tarea. Por ejemplo, en la siguiente figura se muestra la versión tradicional en cascada del ciclo de vida del software como una secuencia de cinco fases. Hay varias metodologías estructuradas de uso común que dividen cada una de estas fases en una secuencia de pasos. En cada paso se describe la función a realizar, las entradas necesarias y las salidas producidas. El propósito es normalizar el desarrollo del software y asegurar la calidad del sistema de software obtenido.

Muchas metodologías se enfocan hacia una fase del ciclo de vida, como el análisis o el diseño. Algunas metodologías han sido diseñadas para acoplarse y alimentarse unas a otras. Por ejemplo, la metodología de análisis estructurado de DeMarco produce las entradas necesarias para la metodología de diseño estructurado de Yourdon. La metodología de la ingeniería de la información de Martin se divide en un conjunto de estados integrados que cubren todo el proceso de desarrollo de software, empezando con la planificación estratégica de los datos.



El ciclo de vida del software

Fig. I.1.5.3.



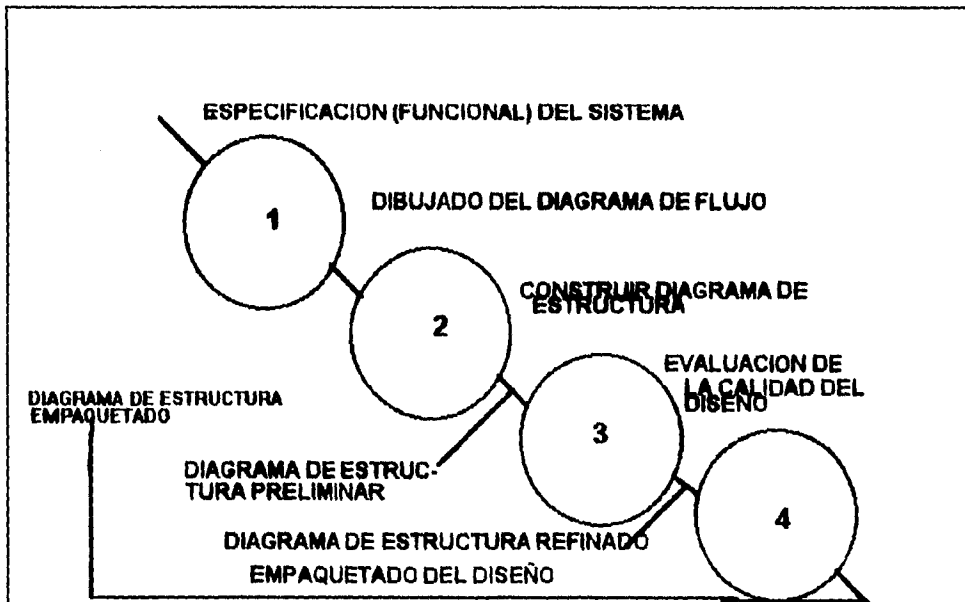
La versión del análisis estructurado de Gane-Searson consta de cinco pasos.

Fig. I.1.5.4

Estrategias y Heurísticos.

Las estrategias para desarrollar el software están un nivel por encima de las reglas y los hechos. Incluyen estrategias tales como:

- desarrollo descendente con descomposición funcional.
- Desarrollo ascendente con reutilización de los componentes del software desarrollados previamente.
- Combinación de la aproximaciones descendente y ascendente.
- Diseño orientado al objeto con descomposición de los objetos y herencia de las clases.



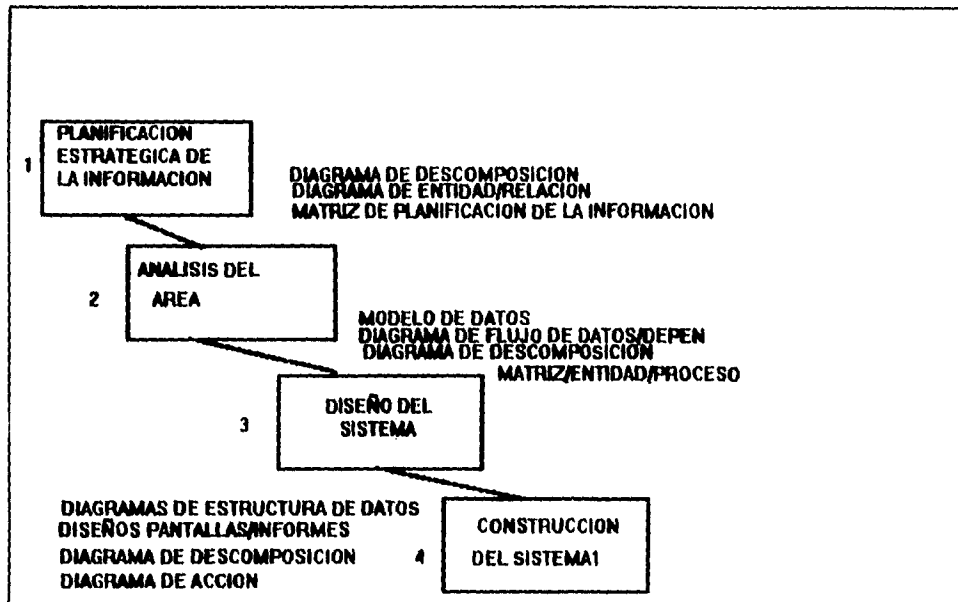
El proceso de diseño estructurado de Yourdon-Constantine
Fig. 1.1.5.5.

El director de metodología ayuda al programador a aplicar correctamente esas estrategias mediante:

- Imposición de las reglas y principios básicos de las tecnologías de software.
- Comprobando la plenitud y casos especiales.
- Informando de cómo aplicar prácticamente la estrategia.
- Seleccionando la tarea próxima a realizar.

Por ejemplo, el director de metodología puede guiar el proceso de diseño del software utilizando el método de descomposición funcional y respondiendo a cuestiones como:

- ¿Cuándo debo detener la descomposición de un componente?
- ¿Qué esquema de modularización es el mejor? O ¿Es este esquema aceptable por lo menos?
- ¿Dónde pongo las decisiones en la estructura del programa? Esto es, ¿he colocado cada decisión en el mejor sitio para minimizar su impacto en el sistema cuando necesito hacer algún cambio.
- ¿Cuál será el próximo paso?
- Existe algún componente que pude utilizar?



La ingeniería de la información es un proceso descendente y orientado hacia los datos para el desarrollo de los sistemas. Arranca con la Planificación Estratégica de la Información. Fig. I.1.5.6.

ESTRATEGIAS PARA GUIAR LA DESCOMPOSICIÓN FUNCIONAL

- descomponer en partes independientes.
- Posponer las decisiones sobre los detalles de implantación.
- Mantener interfaces simples entre los módulos.
- Ocultar información no esencial a los módulos.
- Mantener el uso de datos lo más local posible.
- Tomar primero las decisiones de diseño más fáciles.

HEURÍSTICOS.

Los heurísticos son una parte importante de las estrategias. Los heurísticos son unas reglas empíricas no escritas que comprenden mucha de la experiencia práctica que separan al constructor experto del novel. Los heurísticos incluyen conocimiento sobre atajos y cómo limitar o ignorar las reglas par conseguir que el trabajo sea mejor y más rápido. Por ejemplo, el proceso de diseño estructurado se guía por una de las dos estrategias de diseño siguientes:

La aproximación descendente emplea la descomposición funcional para crear el diseño dividiendo las funciones del sistema en subfunciones. En los pasos simples de diseño, los componentes de los procedimientos y de las estructuras de datos del sistema se definen con detalle cada vez mayor.

La aproximación ascendente sigue la estrategia opuesta. Crea el diseño del sistema combinando los componentes simple para ir formando componentes más complejos. Se prefiere la aproximación descendente cuando el programador está seguro de que es posible construir componentes de niveles más bajos (por ejemplo, los métodos de acceso a los archivos) deben posponerse. La aproximación ascendente se prefiere cuando el sistema es una generalización o extensión de un sistema existente. En este caso la reusabilidad del componente de software acorta el tiempo de desarrollo.

Sin embargo, un programador experimentado sabe que seguir estrictamente una estrategia tanto descendente como ascendente puede causar problemas y errores de diseño. Por ejemplo, cuando se sigue una aproximación descendente, las funciones comunes pueden no ser reconocidas o un diseño descendente puede requerir mucho tiempo para el desarrollo de sistemas grandes. Por otro lado, siguiendo una aproximación ascendente pura puede dar como resultado interfaces de módulos de diseño pobre, haciendo totalmente imposible la prueba minuciosa.

En la práctica, el proceso de diseño generalmente es una combinación de procesos descendentes y ascendentes.

EL NUEVO PROCESO DE DESARROLLO DEL SOFTWARE.

El empleo de un director de metodología inteligente cambia el proceso de desarrollo del software y la división del trabajo entre el programador y sus ayudantes automáticos. Desde la perspectiva de los programadores, el proceso resulta menos formal y con más especificaciones. La mayor parte del trabajo de desarrollo del sistema que ahora realiza el programador está relacionada con la especificación del problema a resolver. (Las necesidades del sistema a construir).

La forma de esta especificación cambia de una descripción narrativa a una representación gráfica y, en muchos casos, a un modelo de prototipo ejecutable. Las primeras metodologías de desarrollo estructurado requerían que se produjese el documento completo y correcto de especificaciones antes de poder desarrollar el sistema. Lo que significaba que los programadores gastaban una gran cantidad de esfuerzo para crear, revisar y corregir los documentos de especificaciones hasta que juzgaban que estaba completo y correcto. Sin embargo, esta era una tarea casi imposible, por que las necesidades cambiaban frecuentemente. Es muy difícil acertar a un blanco móvil.

El director de metodología inteligente emplea técnicas de inteligencia artificial, como lógica difusa, redundancia, razonamiento probable y revisión de las ideas, que pueden trabajar con datos incompletos e incorrectos y cambios en los datos de las necesidades. El papel del director de metodología en el proceso de desarrollo es recoger la especificación parcial e informal descrita por el programador y transformarla en un sistema completo.

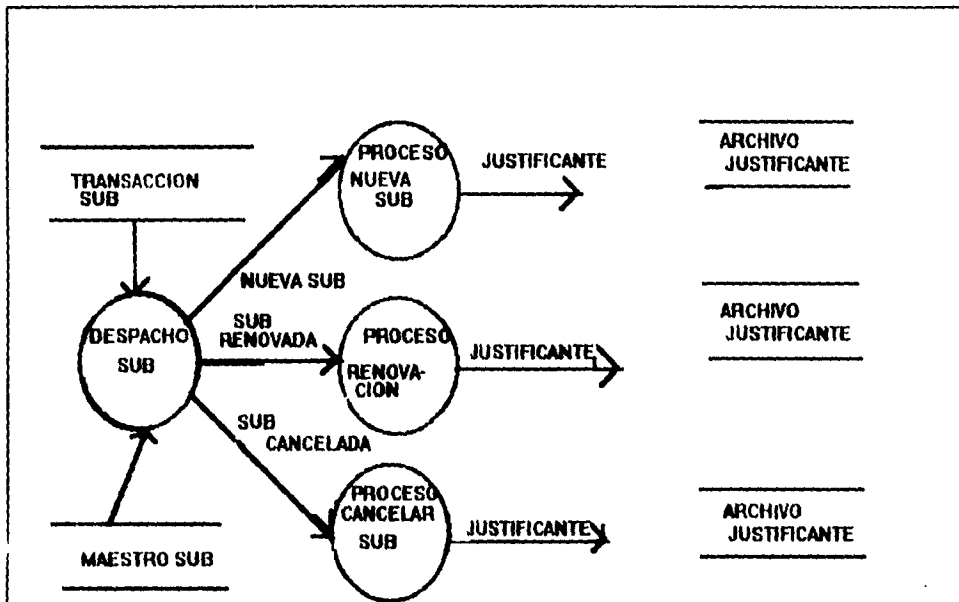
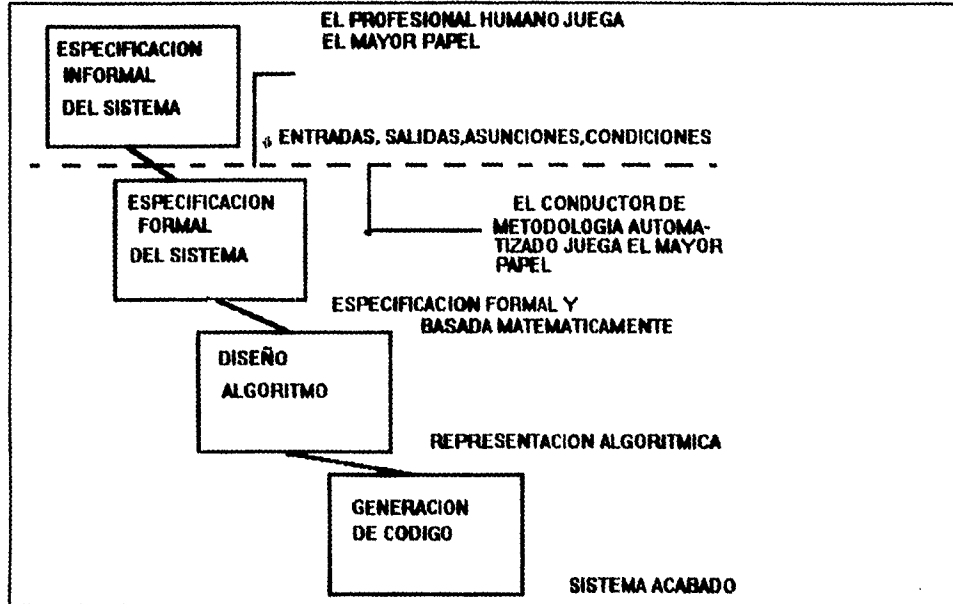


Diagrama de flujo de datos de alto nivel describiendo el sistema de suscripciones.

Fig. 1.1.5.7

Supongamos, por ejemplo, que los programadores empiezan describiendo un sistema de suscripciones que desean construir. El propósito del sistema de suscripciones es procesar tres tipos de transacciones (nuevas suscripciones, renovaciones y cancelaciones), basándose en un archivo maestro de suscripciones para emitir facturas y devoluciones. Esta descripción podría expresarse en un diagrama de flujo de datos de alto nivel como el mostrado en la figura 1.1.5.7. Describe las entradas, salidas, relaciones y suposiciones básicas para el proceso del sistema se una forma específica por la aplicación.



El empleo del conductor de metodología inteligente cambia el proceso de desarrollo de software y la distribución del trabajo entre el humano y el ayudante automatizado. El papel del profesional es definir los requerimientos del sistema. El papel del director de metodología es transformar una especificación informal en un sistema completo.

Fig. 1.1.5.8

El programador comienza el proceso de desarrollo del sistema dibujando el diagrama de flujo de datos en la pantalla de la estación de trabajo. A partir de este punto, el director de metodologías asiste al programador en la construcción del sistema de suscripciones. El primer paso hacia una descripción formal de este sistema es refinar el diagrama de flujo de datos para mostrar más detalles, como las entradas condicionales y los errores de proceso. El director de metodología asiste aquí sugiriendo que este sistema debe ser modelado según las especificaciones de un sistema existente almacenado en su base de conocimiento. Le presenta al programador una, o varias, especificación candidata, después de haberla buscado y comparado con el contenido de su base de conocimiento.

Capítulo I Ingeniería de Software Asistida por Computadora

También amplía la especificación para incluir componentes de proceso normalizados y poder manejar funciones como la validación de los datos, el acceso a las base de datos, la actualización de ficheros y el proceso de errores.

La construcción de la especificación es un proceso interactivo donde el programador y el director de metodología intercambian información y se hacen mutuamente preguntas. El director de metodología requerirá, por medio del indicador (prompt), al programador la información que le falte. Sugerirá el posible paso siguiente, pero dejará al programador la decisión, por si éste elige a otro. Mantendrá una memoria de lo que queda por hacer y recordará al programador los pasos que se ha saltado. Identificará lo que se ha ido añadiendo a la especificación y toda inconsistencia que haya que resolver.

Aunque al programador pueda parecerle que se utiliza un método informal, como la diagramación del flujo de datos, para definir la especificación del sistema lo que realmente utiliza el director de metodología es un lenguaje de especificaciones formal con base matemática. Cada objeto de la especificación que se ha definido tienen un significado formal asociado. Un conjunto de reglas formales gobiernan como puede relacionarse con otros objetos y qué operaciones puede realizarse. Por medio del director de metodología, una especificación informal se transforma en una especificación formal cuya exactitud puede comprobarse.

Una vez que la especificación normal está completa, puede diseñarse el sistema y después el diseño puede transformarse en código ejecutable. Como se muestra en la siguiente figura, el director de metodología tiene un papel importante en los pasos restantes del desarrollo, que son una serie de transformaciones que culminan en el sistema final.

De nuevo el director de metodología utiliza su base de conocimiento, que contiene diseños de un sistema de aplicación similar para encontrar posibles algoritmos de diseño para el sistema de suscripciones. Con la ayuda de los programadores, los algoritmos se adaptan al sistema refinando los tipos de datos, el flujo de datos y el flujo de control. La última transformación del diseño en código es bastante directa y se realiza pro medio de un generador de código.

En conclusión, el director de metodología es un sistema experto que incorpora conocimientos sobre metodologías de desarrollo de software en las herramientas de software. Su propósito es guiar y dirigir al usuario en el uso correcto de las metodologías de desarrollo de software. El director de metodología comprende los pasos del proceso de una metodología en particular y es la base de la verdadera automatización del software.

EL DIRECTOR DE METODOLOGÍA.

- Incluye herramientas que imponen activamente el empleo de metodologías de software.
- Proporciona listas de control dependientes del contexto para tareas específicas.
- Automatiza el conocimiento sobre los mejores planes y prácticas.
- Ofrece sugerencias, consejos y explicaciones.
- Emplea un sistema experto que modela los procesos de desarrollo de software: imita el comportamiento de un experto en ingeniería de software.

CLASIFICACIÓN DE LAS METODOLOGÍA ESTRUCTURADAS.

Las metodologías estructuradas pueden clasificarse en pertenecientes a:

- Corriente de la ingeniería del Software
- Orientada a los procedimientos, a los datos o a la información.
- Corriente de la Ingeniería de Información.
- Desarrollo de Sistemas de Tiempo real o de información.

INGENIERÍA DEL SOFTWARE.

La ingeniería del Software es un proceso descendente (top-down) de la implantación por fases para el desarrollo de programas. Esta proporciona un proceso de desarrollo paso a paso que comienza con la visión funcional más general de lo que puede hacer el programa, descompone toda esta visión en subfunciones y repite el proceso por cada subfunción hasta que todas las subfunciones sean lo suficientemente pequeñas para ser implantadas en código de programas.

El resultado es un programa modular estructurado jerárquicamente, aunque el diseño de los componentes funcionales y el diseño de la estructura de los datos se realizan en paralelo, el diseño de la estructura de los datos se pospone hasta el diseño de los procedimientos, para proteger la independencia y la claridad de los componentes funcionales. Los requerimientos de los datos del programa se descubren a través del análisis de los requerimientos funcionales del programa. De esta forma, la ingeniería del software se enfoca hacia una visión funcional del programa.

PRINCIPIOS BÁSICOS DE LA INGENIERÍA DEL SOFTWARE.

- Un planteamiento descendente (top-down) de implantación por fases requiere un procedimiento gradual desde el nivel más alto de control y de definición de datos, descendiendo hasta los módulos funcionales y las estructuras de los datos.
- Un aspecto muy importante del proceso descendente es la existencia en la firmeza de los requerimientos.
- La solución al sistema se define en funciones; los datos se diseñan para preservar la funcionalidad del sistema del sistema y la independencia de los módulos del programa. Por tanto, los requerimientos de los datos se descubren a través del análisis de las funciones requeridas por el sistema.

LOS DIAGRAMAS DE LA INGENIERÍA DE SOFTWARE.

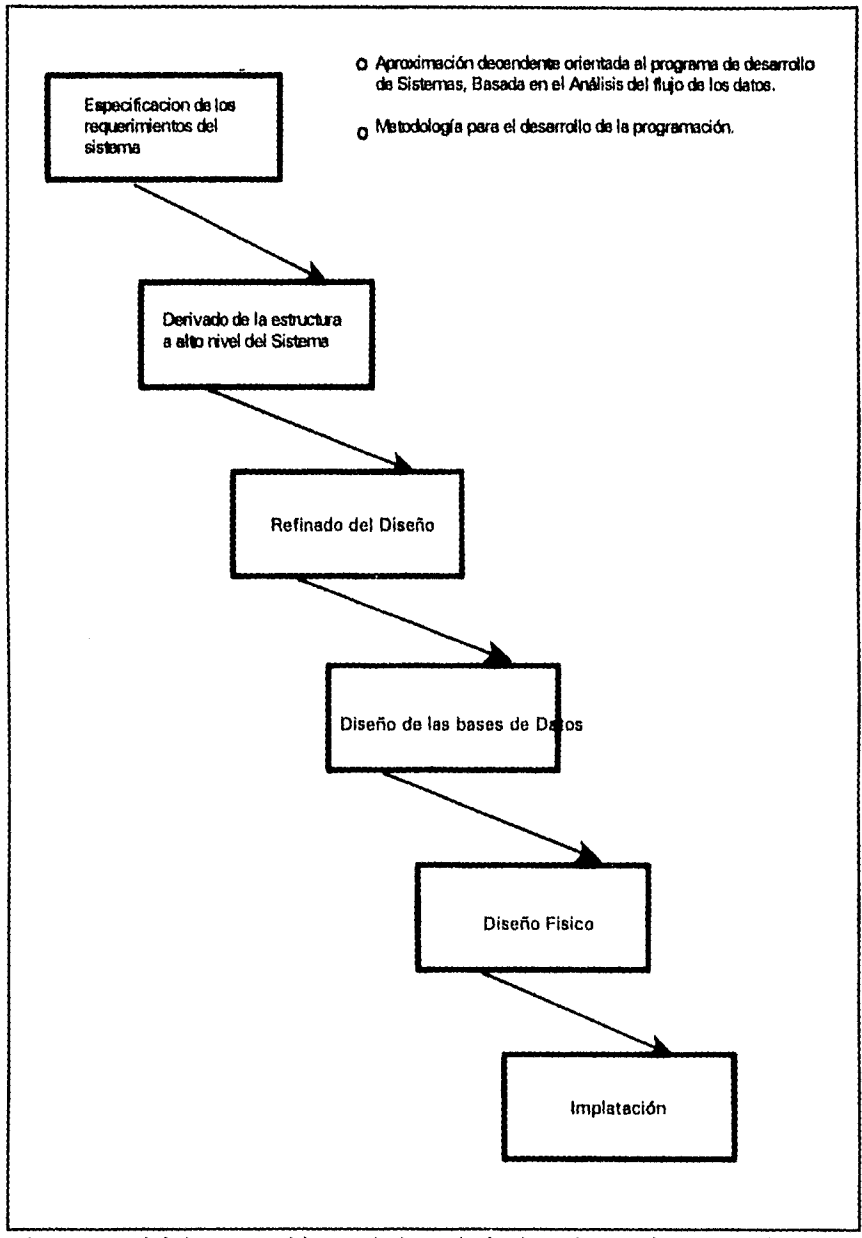
Los tipos de diagramas necesarios para soportar un planteamiento en ingeniería de software para el desarrollo del sistema son:

- Diagrama de flujo de datos: muestra los procesos funcionales desde el nivel más alto en un sistema y el flujo de datos desde la entrada de la información través del del proceso y las eventuales salidas.
- Diagramas estructurados en árbol: muestran las relaciones entre los que diseños de programa y la estructura jerárquica de los datos.
- Diagrama detallado de la lógica del procedimiento: muestra detalladamente la lógica del programa, secuencias, selección e iteración.
- Diseño de pantallas y de informes: Muestra el diseño del interfaz del usuario.

La ingeniería del Software representa uno de los primeros intentos para estructurar el proceso de desarrollo del software. Desde los años sesenta se ha avanzado en el cambio del desarrollo del software desde un arte manual a una disciplina casi mecanizada con el objetivo de introducir un enfoque normalizado al desarrollo del software. Los conceptos y técnicas involucrados en la ingeniería de software incluyen:

- El diseño descendente (top-down)
- La programación estructurada.
- La modularidad.
- El refinamiento progresivo y la descomposición funcional.
- La métrica de la calidad del software.
- La abstracción de los lenguajes de programación.

La ingeniería de software es una disciplina general de desarrollo que puede aplicarse al desarrollo de los sistemas de información y a los sistemas de tiempo real. Por ejemplo, las metodologías de diseño para sistemas de tiempo real de Ward-Mellor y de Hatley están basadas en la metodología de diseño estructurado de Yordon-Constantine empleada para desarrollar sistemas de información. La ingeniería de software está fundada sobre el modelo básico de Entrada/Proceso/Salida de un sistema. Los datos se introducen en el sistema como Entradas, se operan con ellos (se transforman) por medio de un proceso y lo abandona como salidas. Este modelo básico se utiliza en todas las metodologías de diseño estructurado. Sin embargo, el orden en que se construye el modelo depende de la metodología particular que sigue. La ingeniería de Software tradicional trata la parte del proceso como la fundamental del modelo del sistema y por tanto se describe como un enfoque del desarrollo del software orientado al procedimiento. Los datos se derivan de la función. En un enfoque de desarrollo orientado a los datos, la parte del modelo correspondiente a las entradas y salidas se considera como la más importante.



Los pasos básicos seguidos en la ingeniería de software durante el desarrollo de un programa. fig. 1.1.5.9

Por tanto, las estructuras de los datos se define primero y los componentes de los procesos se derivan de las estructuras de los datos. En la figura anterior se muestran las premisas básicas del enfoque de desarrollo dirigido por los datos. Los tipos de diagramas necesarios para soportar un enfoque de desarrollo dirigido por los datos son diagramas de árbol estructurado jerárquicamente para representar las estructuras de los datos y de los programas y el diagrama detallado de la lógica de procedimiento.

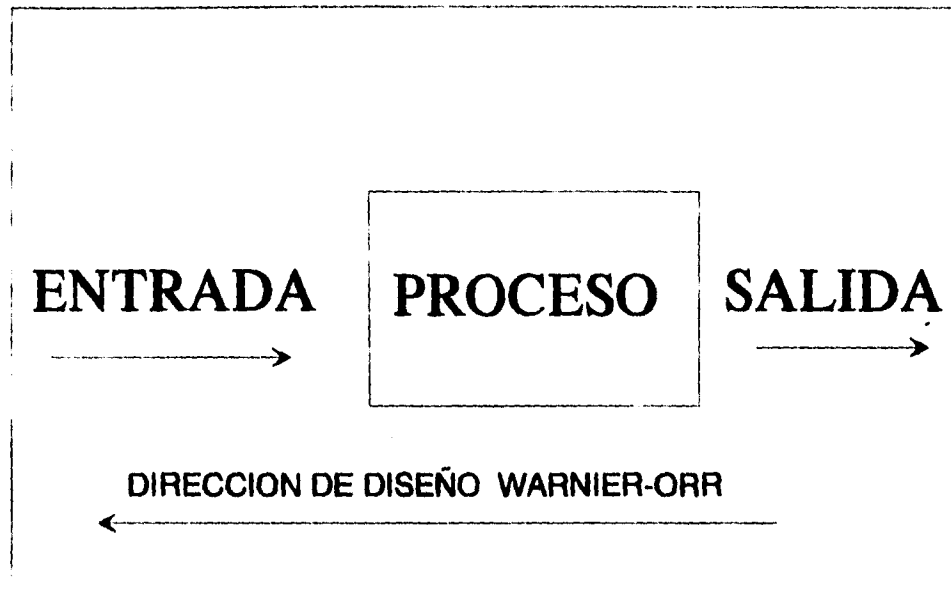
La metodología de diseño de Jackson es un ejemplo de metodología orientada a los datos. En el enfoque de Jackson las estructuras de los datos de entrada y salida se definen primero. Después, se define la estructura del programa mezclando todas las estructuras de los datos del programa.

La metodología de diseño de Warnier -Orr es otro ejemplo de enfoque orientado a los datos. Pero a diferencia del enfoque de Jackson, la formulación de Warnier-Orr se centra en las salidas del programa.

Es una metodología orientada a las salidas. Utilizando la formulación de Warnier-Orr, se comienza por la definición de las salidas del sistema y se trabaja hacia a través del modelo básico del sistema hasta definir las partes del diseño correspondientes a los procesos y a las entradas.

LAS PREMISAS BÁSICAS SON:

- La estructura del control del programa debe de ser jerárquica y debe derivarse de la estructura de los datos del programa.
- El proceso del diseño consiste en definir primero las estructuras de los datos, mezclarlas luego todas en una estructura jerárquicas de programa y después ordenar detalladamente la lógica procedimental para que se ajuste a esta estructura.
 - El diseño lógico debe estar separado del diseño físico y precederlo.
 - Los dos diagramas necesarios para soportar un enfoque dirigido por los datos son:
 - * Diagramas estructurados en árbol jerárquico.
 - * Diagrama detallado de la lógica del procedimiento.



Los datos se introducen en el sistema como entradas, se opera con ellos (se transforman) por medio de un proceso y lo abandonan como salidas.
fig. I.1.5.10

EL ENFOQUE DEL DESARROLLO ORIENTADO A LA INFORMACIÓN.

En las metodologías de desarrollo de sistemas de formulación orientada a la información, un modelo de datos lógico que represente la información utilizada a través de una organización es el punto de arranque de todo el desarrollo del sistema. El proceso empieza con un análisis de alto nivel de una compañía, sus objetivos comerciales y las necesidades que muestre información estratégica. Una visión global de las necesidades de información se presentan en un modelo que muestran todas las entidades de modelos básicos de la organización y sus relaciones entre sí. Después, basándose en este modelo se construyen individualmente los sistemas de información de la organización. Así los procedimientos se derivan de los datos.

LA INGENIERÍA DE LA INFORMACIÓN.

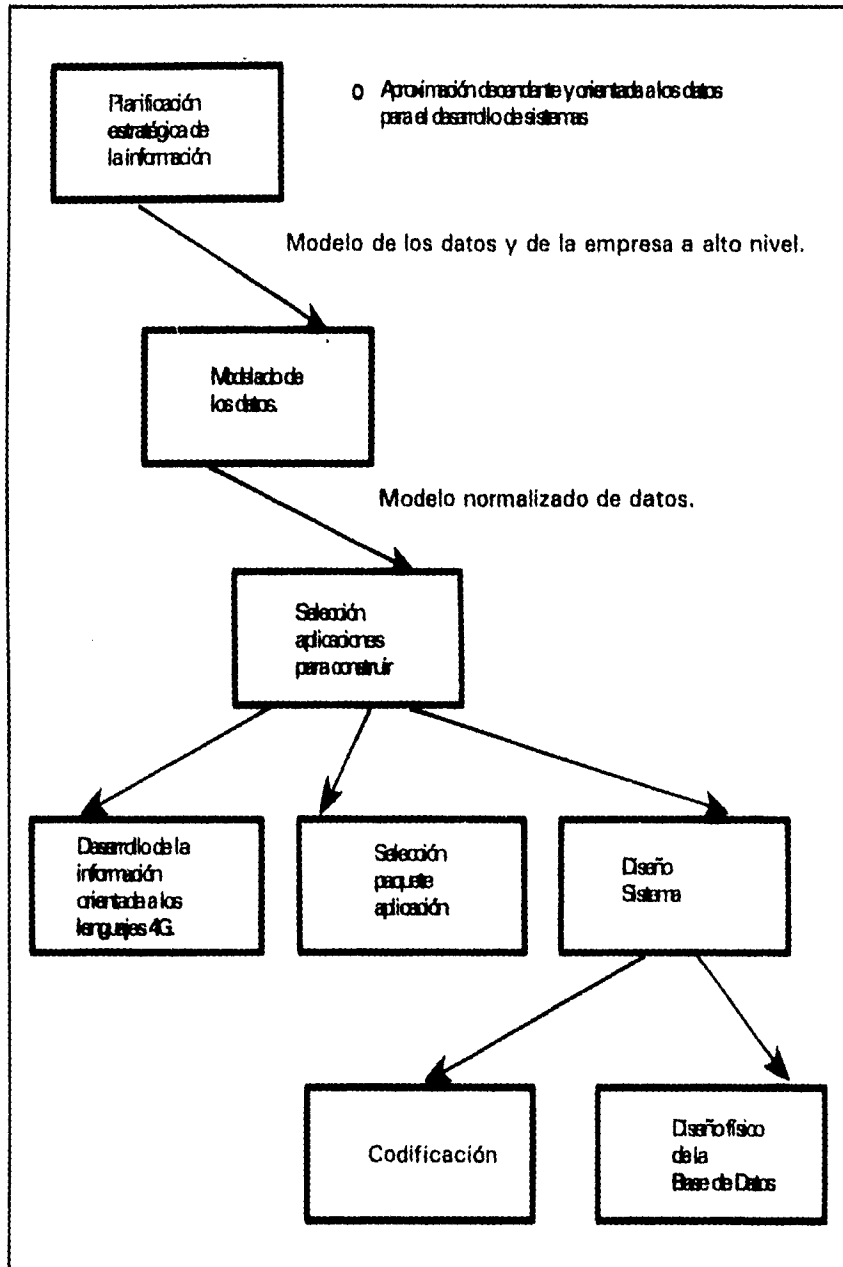
La ingeniería de la información es un ejemplo de una formulación de desarrollo orientada a la información.

LAS PREMISAS BÁSICAS DE LA INGENIERIA DE LA INFORMACIÓN.

- Se necesita una estrategia general de desarrollo del sistema centrada en una planificación de los objetivos estratégicos del negocio para construir negocios que satisfagan mejor las necesidades de la compañía.
- Sistemas de información que pueden integrarse mejor si los datos a compartir se controlan centralmente por ser parte de un mismo modelo lógico de datos.
- La representación lógica de los datos es relativamente estable, en tanto que los procedimientos que utilizan los datos cambian frecuentemente.

Por tanto, el modelo lógico de datos, que refleja lo que es organización, no como trabaja, deberá ser la base del desarrollo del sistema.

La ingeniería de la información es tanto un enfoque comercial como de ingeniería, como una formulación para la construcción de sistemas de software. Es una formulación comercial porque comienza con una planificación estratégica de la organización. Es una formulación de ingeniería porque proporciona un procedimiento paso a paso para construir sistemas de información. La siguiente figura describe los pasos lógicos de la formulación de la ingeniería de información.



Pasos básicos en la ingeniería de la información para desarrollar un sistema.
Fig. 1.5.11

Capítulo I Ingeniería de Software Asistida por Computadora

La ingeniería de la información es una disciplina más reciente que la ingeniería de Software. En cierto sentido tiene un enfoque más amplio que la ingeniería de Software. Es una construcción del Sistema, más que una disciplina de construir programas. Comienza por un nivel más alto (con la planeación estratégica) que la ingeniería de software. Sin embargo en la fase del diseño del programa, su formulación del desarrollo del programa es básicamente la misma que la ingeniería del software.

En otro sentido, sin embargo, la ingeniería de información tiene un enfoque más estrecho que la ingeniería de software. Se emplea aquella para construir sistemas de información, mientras que la ingeniería de software se emplea para desarrollar toda clase de sistemas (tanto de tiempo real , como comerciales).

La ingeniería de la información es un enfoque orientado a la información para el desarrollo de software, y por tanto difiere de los enfoques orientados a los datos, como las de Jackson y Warnier-Orr. La ingeniería de información ha sido diseñada para desarrollar sistemas de bases de datos y puede trabajar con estructuras de datos no jerarquizadas. El modelo lógico de datos y la normalización son pasos requeridos en la ingeniería de la información. La ingeniería de la información construye sistemas de información integrados, porque están contruidos sobre el mismo modelo lógico de datos.

Además el principio de la ingeniería de la información está en los datos desde el punto de vista de los requerimientos comerciales y no en los requerimientos del sistema de software. Todo el enfoque del desarrollo de la ingeniería de la información se centra en la visión lógica de como el sistema utiliza los datos. La ingeniería de información se podrá apreciar al desarrollarse la metodología de la ingeniería de la información de Martín.

EL ANÁLISIS ESTRUCTURADO

El análisis estructurado propone un enfoque sistemático por pasos para la realización del análisis del sistema y la producción de las especificaciones del sistema. Hay dos versiones similares de análisis estructurado:

- * Gane-Searson
- * DeMarco

Ambas representan una disciplina estructurada basada en los siguientes conceptos estructurados:

- * Organización jerarquizada descendente.
- * Herramientas gráficas de comunicación/documentación.

El análisis estructurado utiliza un método descendente de descomposición funcional para definir los requerimientos del sistema. La especificación del sistema producido por el análisis estructurado es un modelo descendente particionado del sistema a construir. El suministro de una descripción de los requerimientos del sistema a construir es la conexión entre el análisis y el diseño.

La unión entre el análisis y el diseño se proporciona con la descripción de los requerimientos del sistema a construir.

El objetivo primario del análisis es producir una especificación del sistema que defina la estructura del problema a resolver según la visión del usuario.

METODOLOGÍA	CLASIFICACIÓN
De Marco Análisis estructurado	<ul style="list-style-type: none"> * Ingeniería de software * Orientados al procedimiento * Sistemas de información * Informal (extensiones tiempo real)
Gane-Searson Análisis estructurado	<ul style="list-style-type: none"> * Ingeniería de software * Orientado al procedimiento * Sistemas de información * Informal (extensiones tiempo real)
Yourdon Diseño estructurado	<ul style="list-style-type: none"> * Ingeniería de software * Ingeniería de software * Orientado al procedimiento * Sistemas de información * Informal (extensiones tiempo real)
Jackson Diseño estructurado	<ul style="list-style-type: none"> * Ingeniería de software * Orientados a los datos * Sistemas de información * Informal (extensiones tiempo real)
Orr Desarrollo de sistemas	<ul style="list-style-type: none"> * Ingeniería de software * Orientado a los datos * Orientado a las salidas * Sistemas de información * Informal
Martin Ingeniería de la información	<ul style="list-style-type: none"> * Ingeniería de la información * Orientado a la información * Sistemas de información * Informal

El propósito principal del análisis es producir una especificación del sistema que defina la estructura del problema a resolver según la visión del usuario. El propósito del diseño es definir la estructura de la solución de forma que sea compatible con la estructura del problema y con los requerimientos del usuario. Los defensores del análisis estructurado sugieren que al utilizar el mismo método de construcción (descomposición funcional descendente, en la especificación y en diseño, los dos pueden unirse para representar un sistema que satisfaga las necesidades y expectativas del usuario.

LA ESPECIFICACIÓN ESTRUCTURADA

La especificación del sistema producida por el análisis estructurado es un modelo lógico, gráfico, particionado, descendente y jerárquico de los procesos del sistema y de los datos utilizados por estos procesos.

EL DIAGRAMA DE FLUJO DE DATOS

Un diagrama de flujo de datos es un diagrama estructurado que representa los procesos (funciones o procedimientos) de un sistema y los datos que les conectan. Es la herramienta central del modelado del análisis estructurado y se emplea para fraccionar jerárquicamente el sistema en procesos. En los niveles inferiores, el proceso de nivel más alto se expande para ver si hay otros procesos y los datos más detallados involucrados.

EL DICCIONARIO DE DATOS

El diccionario de datos es un conjunto de definiciones de todos los datos que aparecen en un diagrama de flujo de datos, en datos almacenados o como flujo de datos. La definición de cada dato consta de los componentes del dato que constituyen el dato y las relaciones entre ellos.

FICHERO DE CLIENTES = [Registro de clientes]

REGISTRO DE CLIENTES = Nombre-cliente + dirección-cliente + información-del-pago
+ órdenes-de-pago + tipo-de-cliente

LAS ESPECIFICACIONES DE PROCESO

Una especificación de proceso (o miniespecificación) describe lo que ocurre dentro de un cuadro de proceso en un diagrama de flujo de datos. Una miniespecificación tiene generalmente una página de extensión y explica cómo se transforman los datos de entrada en datos de salida. La miniespecificación se escribe en inglés estructurado (seudocódigo) como una tabla de decisión o como un diagrama de acción. Se crea una miniespecificación por cada proceso definido en el diagrama de flujo de bajo nivel en la especificación del sistema.

LOS PASOS DEL ANÁLISIS ESTRUCTURADO

DeMarco define el análisis estructurado en siete pasos:

PASO 1

Documenta la forma en que las cosas en curso se realizan en el entorno actual del usuario. La salida de este paso es el diagrama de flujo físico de los datos actuales. Muestra la situación, el personal, los nombres y los procedimientos manuales y automatizados, y utiliza los términos de los usuarios.

PASO 2

Utiliza el diagrama físico de flujo de datos del paso 1 para crear el modelo lógico del sistema actual. El modelo lógico, igual que el físico se representa como un conjunto de niveles jerarquizados de los diagramas de flujo de datos.

La diferencia entre los dos modelos está en que el modelo físico muestra los detalles físicos de cómo un usuario particular hace las cosas, mientras que el modelo lógico muestra lo que se hace en un nivel abstracto, distinguiendo cualquier estado particular de los procedimientos fundamentales del negocio.

PASO 3

Desarrolla el nuevo modelo lógico del sistema a construir, siendo el punto de partida el modelo lógico del paso 2. Se modifica para incorporar los cambios para el nuevo sistema. Indica los procesos manuales y los automatizables y se representa como un conjunto por niveles de diagramas de flujo de datos que describen el sistema en variados niveles de detalle. El paso 3 conlleva la mayor parte del esfuerzo de trabajo del proceso de análisis estructurado.

PASO 4

Crea un nuevo modelo físico del sistema a construir identificando el interfaz hombre/máquina. Puede crearse más de un modelo posible para representar los distintos grados de automatización.

PASO 5

Produce una estimación de los costes y una planificación por cada uno de los modelos físicos alternativos desarrollados en el paso 4.

PASO 6

Basándose en la estimación obtenida en el paso 5, se selecciona un modelo físico para utilizar en la especificación de las funciones y requerimientos del sistema a construir.

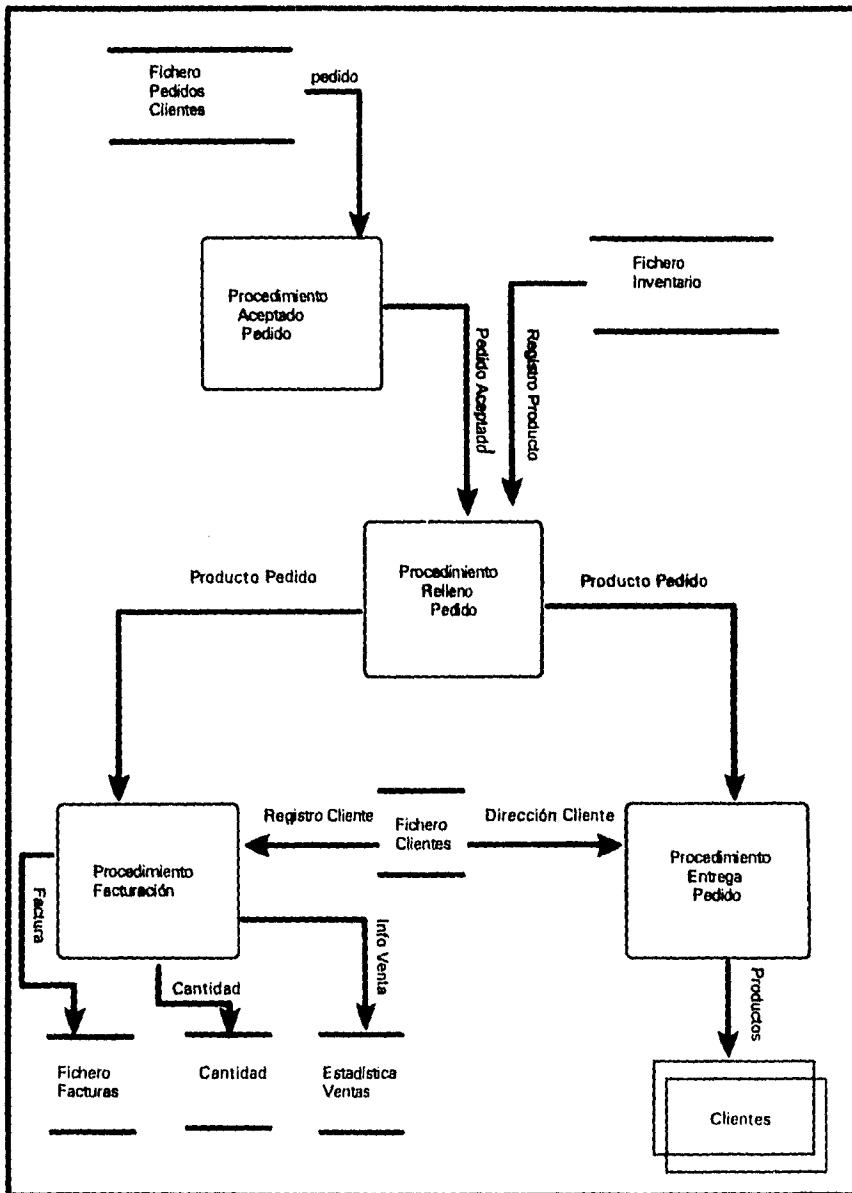


Diagrama de flujo de datos mostrando los procesos del sistema de distribución de ventas, dibujado con el INFORMATION ENGINEERING WORKBENCH de KnowledgeWare. Fig. 1.1.3.9

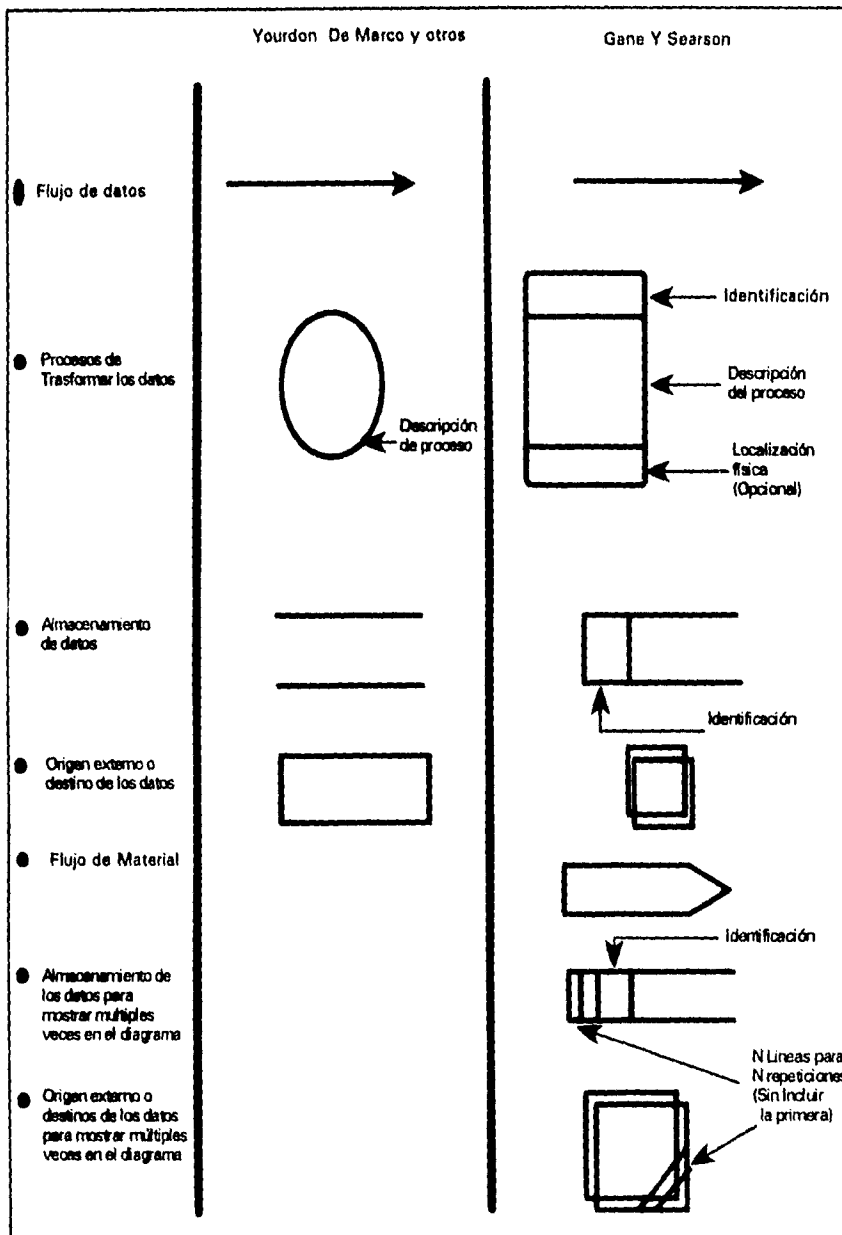
PASO 7

Se empaqueta en su forma final la especificación estructurada, que incluye un conjunto por niveles de diagramas de flujo de datos, el diccionario de datos y las miniespecificaciones.

Gane y Searson definen un proceso similar para el análisis. La principal diferencia es que Gane y Searson incluyen un paso del modelado de datos almacenados mostrados en los diagramas de flujo de datos, posiblemente en tercera forma normal. Gane y Searson incluyen el modelado de datos en el paso 2 de su versión de análisis estructurado. Después, en el paso 3, el modelo lógico de datos se emplea como entrada para diseño físico de la base de datos. También, utilizan en su versión de diagramas de flujo, un conjunto de símbolos gráficos ligeramente diferentes.

EL DISEÑO ESTRUCTURADO DE YOURDON.

La metodología de diseño estructurado de Yourdon proporciona un proceso para el diseño paso a paso de sistemas y programas detallados. Unos pasos implican el desarrollo del diseño, otros, la medición y la mejora de la calidad del diseño. Cada paso es soportado por un conjunto de estrategias de diseño, guías y técnicas de documentación. Por ejemplo, las estrategias de diseño, las de análisis de transformación y las de transacción guían al diseñador en las decisiones de diseño. Las medidas de diseño, como las de acoplamiento y cohesión proporcionan al diseñador, técnicas para medir la calidad del diseño. El principal producto obtenido por los procesos del diseño estructurado es un diagrama de estructura mostrando los componentes de procedimientos del programa, su ordenación jerárquica y los datos conectados a ellos.



Los símbolos utilizados en los diagramas de flujo de datos
Fig. I.1.5.12

EL DIAGRAMA DE ESTRUCTURA.

Es un diagrama jerárquico o de árbol que define la arquitectura global de un programa mostrando sus componentes de procedimiento y sus interrelaciones. Los bloques básicos para construir el diagrama de estructura son cajas que representan los componentes de procedimientos y las flechas que los conectan. Los datos que pasan entre los componentes de procedimientos se llaman pares y se escriben a lo largo de las flechas de conexión.

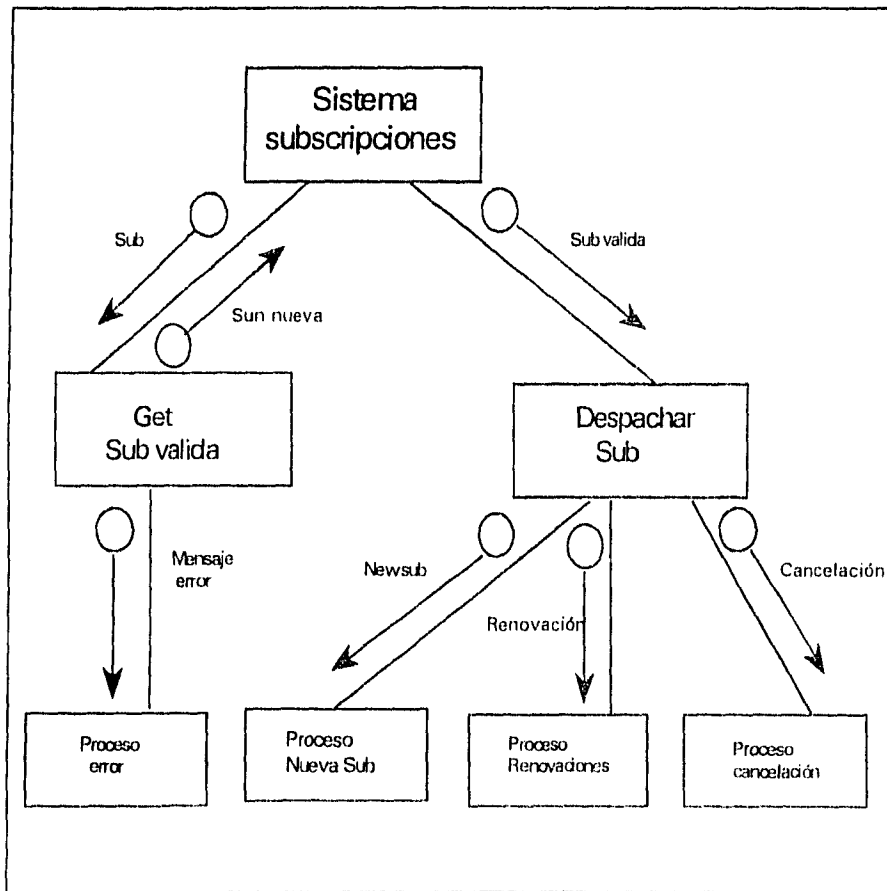
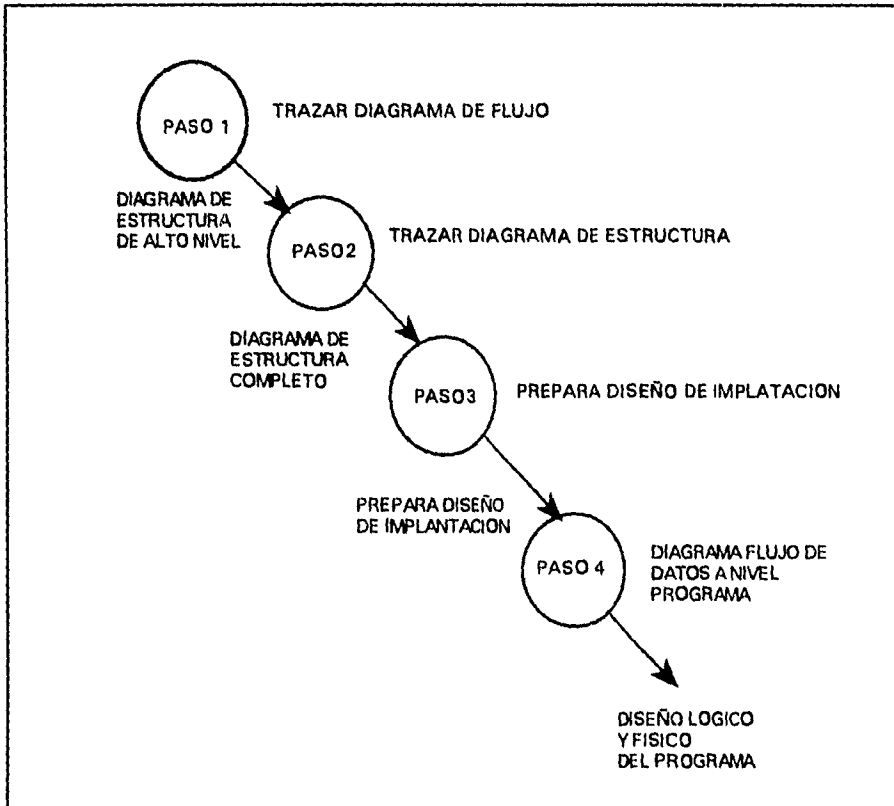


Diagrama de estructura a alto nivel derivado del diagrama de flujo de datos utilizando la estrategia de diseño de transacción Fig. I.1.5.13

LOS PASOS DE DISEÑO DE YOURDON



El proceso de diseño estructurado consta de cuatro pasos secuenciales en la metodología de diseño estructurado de Yourdon.

Fig. I.1.5.14

PASO 1: Trazar el diagrama de flujo de datos.

El primer paso en el proceso de diseño estructurado es representar el problema de diseño como el flujo de datos a través de un sistema. El sistema se compone de procesos que operan (transforman) con los datos. Estos procesos y los datos que los enlazan forman la base para definir los componentes de programa. Se utiliza un conjunto por niveles de diagramas de flujos de datos que los enlazan para representar la primera visión del diseño del programa.

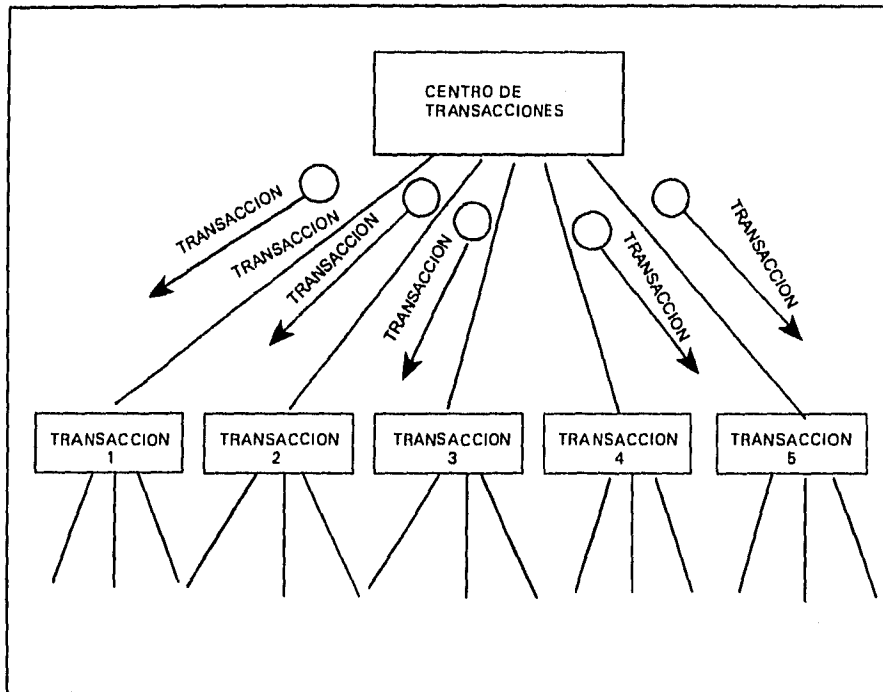
PASO 2: Trazar el diagrama de Estructuras.

El segundo paso es representar el diseño del programa como una jerarquía de componentes de procedimiento. Se emplea un diagrama de estructura para mostrar esta visión del diseño. El diagrama de estructura se deriva del diagrama de flujo de datos obtenido en el paso 1. El diseño estructurado proporciona dos estrategias de diseño para guiar la transformación de uno o varios diagramas de flujo de datos a un diagrama de estructura: los análisis de transformación y de transacción. Estas dos estrategias proporcionan dos modelos de estructuras que pueden utilizarse individual o conjuntamente para dirigir el diseño jerárquico así como un proceso paso a paso de transformación por cada estrategia.

El análisis de transformación es un modelo de flujo de información que divide el diagrama de flujo de datos en tres partes: la entrada que recibe el nombre de rama aferente, el proceso lógico llamado transformación central y la salida denominada rama eferente.

El análisis de transacción se emplea cuando se diseñan programas con proceso de transacciones. El diagrama de estructura general para un programa con proceso de transacciones se muestra en la siguiente figura:

En la parte superior de los diagramas de estructura esta el modulo de transacciones central y debajo hay varios módulos de transacciones. El diagrama de estructura se obtiene definiendo la transformación central y las transacciones como se indica en las siguientes figuras.



El modelo general de diagrama de estructura para un programa con proceso de transacciones, tiene un módulo "centro de transacciones" en lo más alto y por debajo un módulo de transacción para cada tipo distinto de transacción.

Fig. I.1.5.15

PASO 3: Evaluación del diseño .

El tercer paso en el diseño estructurado es la medición de la calidad del diseño. El acoplamiento y la cohesión son las dos técnicas de medición del diseño proporcionadas por el diseño estructurado.

El acoplamiento mide el grado de independencia entre los componentes del procedimiento (módulos) en el diagrama de estructura. Cuando existe poca interacción entre dos módulos, éstos se describen como ligeramente acoplados. Cuando la interacción entre dos módulos es grande, los módulos se describen como estrechamente acoplados.

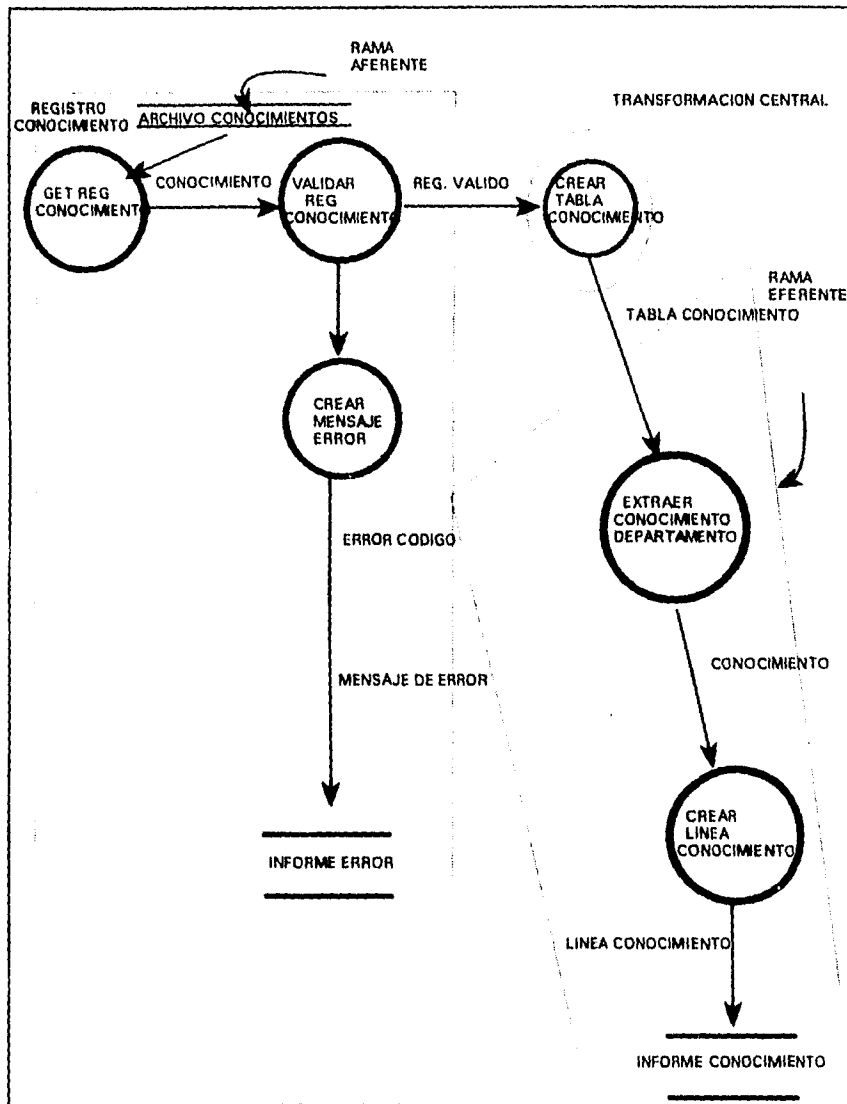


Diagrama de flujo de datos del sistema de conocimiento de empleados, indicando las ramas afrentes (entrada), las ramas eferentes (salida) y la transformación central (proceso lógico).

Fig. I.1.5.16

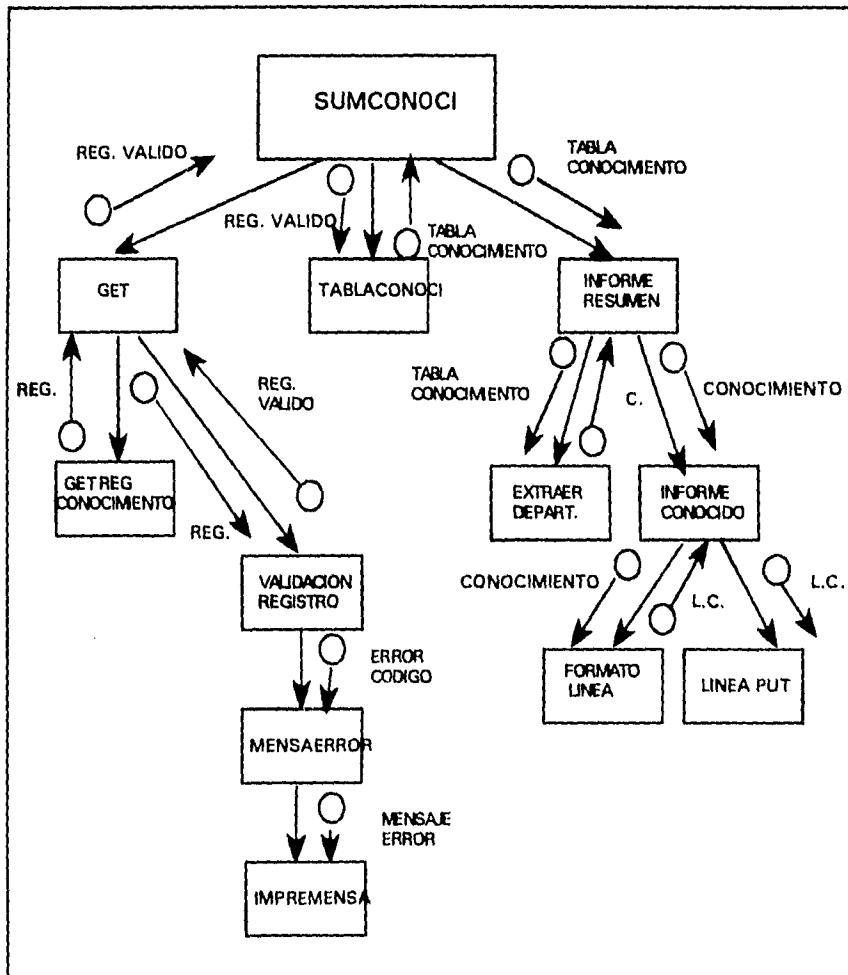


Diagrama de estructura del sistema de conocimientos de empleados derivado del diagrama de flujo de datos mostrando en la figura 1.1.5.14 Fig. 1.1.5.17

Un diseño de alta calidad significa que los módulos están lo más ligeramente acoplados posible. Hay cinco niveles de acoplamiento: datos, estampado, de control, común y por contenido.

El acoplamiento por datos es el más ligero y por tanto el mejor tipo. Mientras que el acoplamiento por contenido es el más estrecho y por tanto el peor tipo.

La cohesión mide la fuerza de las relaciones entre los elementos dentro de un módulo cuanto más fuerte, mejor. Hay siete niveles de cohesión: funcional, secuencial, de comunicación, de procedimiento, temporal, lógica y de coincidencia. La cohesión funcional es la mejor, la de coincidencia.

PASO 4: LA PREPARACIÓN DEL DISEÑO PARA LA IMPLANTACIÓN.

El último paso de un diseño estructurado es la preparación del diseño para la implantación. Esto se llama empaquetar el diseño. Empaquetar es el proceso de dividir el diseño de programa lógico en unidades físicas de implantación llamadas unidades de carga. Es un diseño físico del programa.

Es una descripción de la tradicional metodología de diseño estructurado de Yourdon. Hay variaciones y actualizaciones, alguna de las cuales han extendido las metodologías de diseño de Yourdon (que fue ideada originalmente para el diseño de sistemas de información para soportar el diseño de sistemas de tiempo real.

LA METODOLOGÍA DE DISEÑO DE JACKSON.

Como la metodología de diseño de Yourdon, la de Jackson proporciona un proceso descendente por pasos de diseño. La principal diferencia entre las dos, es que la de Jackson esta basada en el análisis de la estructura de los datos, mientras que la de Yourdon se basa en el análisis de flujo de datos. La de Jackson esta orientada a los datos y la de Yourdon al procedimiento. La formulación de Jackson aboga por una visión estática de las estructuras, mientras que la de Yourdon, utiliza una visión dinámica del flujo de datos.

La metodología de diseño de programas de Jackson es una técnica dirigida por los datos. Su objetivo es obtener la estructura del programa derivándola de las estructuras de los datos, asumiendo que se ha especificado totalmente el problema.

Jackson considera el programa como un proceso secuencial. Tiene entradas y salidas que le consideran como corrientes secuenciales de registro. El proceso de diseño consiste en definir primero la estructura de la corriente de datos y después ordenar el proceso lógico (operaciones) para ajustar la estructura de datos.

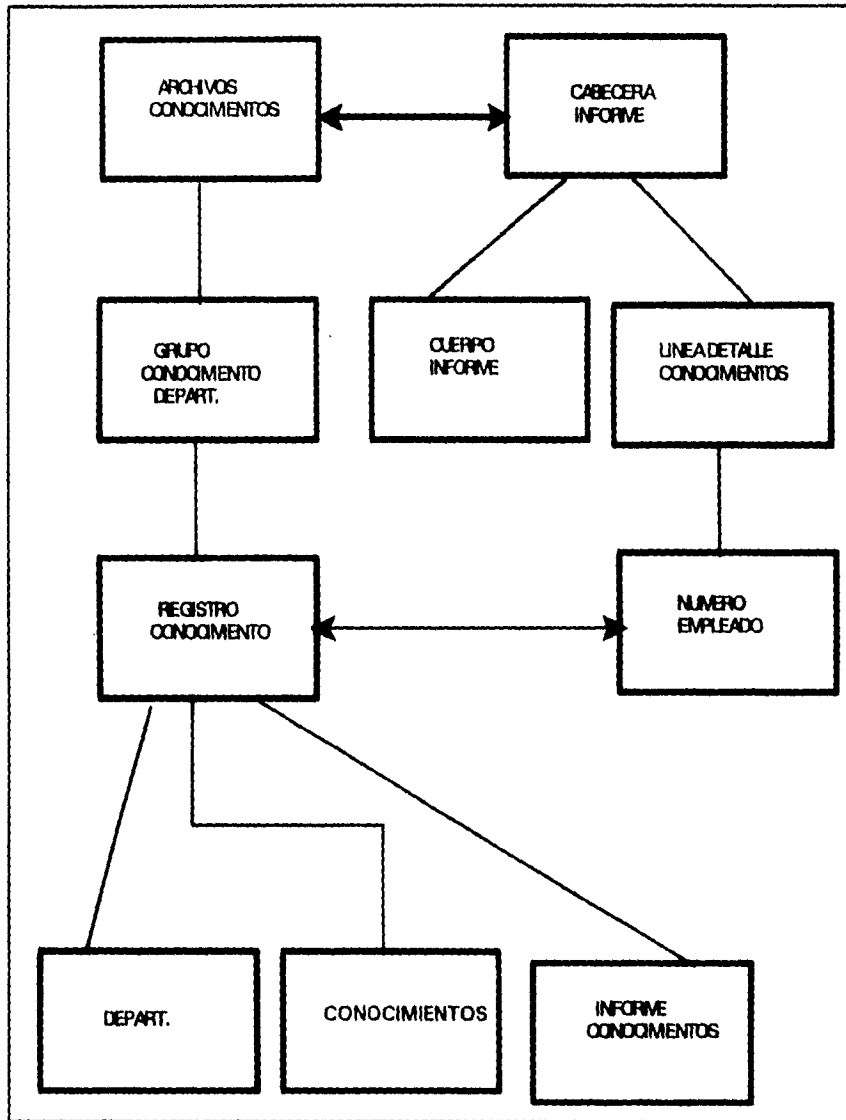
PASOS BÁSICOS EN EL PROCESO DE DISEÑO DE JACKSON:

- Paso de Datos: Dibuja un diagrama estructurado por cada entrada y salida de datos en el programa.
- Paso de Programa: Toma todas las estructuras de datos definidas en el paso de datos y forma con ellas una única estructura de programas.
- Paso de operación: Hace una lista con las operaciones necesarias para producir las salidas a partir de las entradas. Después, asigna cada operación de la lista a un componente de la estructura del programa (progresivamente desde la salida hasta la entrada).
- Paso de Texto: Transcribe la estructura del programa a texto, añadiendo la lógica condicional que rige la ejecución de los bloques y la estructura de selección.

DIAGRAMAS ESTRUCTURADOS Y PASOS DE JACKSON.

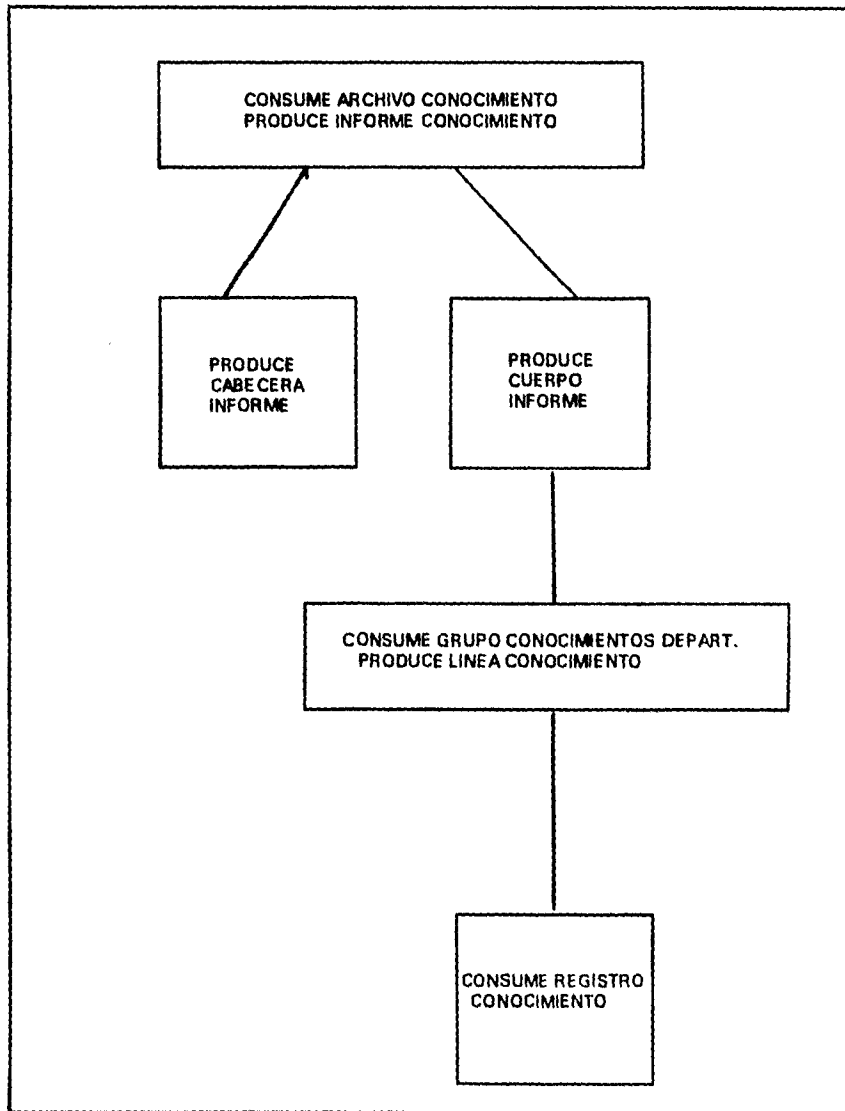
La metodología de diseño de Jackson utiliza las técnicas de diagramación estructurada en árbol para describir un diseño de programa.

- Diagrama de red: El flujo de corrientes de datos entre programas.
- Diagrama estructurado en árbol: Representación jerárquica de estructuras de datos y programas.
- Texto: en forma de pseudocódigo.



El programa de conocimientos de empleados tiene una corriente de entrada y una de salida. Se utiliza el diagrama estructurado en árbol para representar la estructura de los datos. Las flechas sombreadas muestran la relación consumo-producción entre los componentes en la estructura de datos.

Fig. 1.1.5.19



La estructura del programa del sistema conocimientos de empleados está compuesto por las estructuras de los datos de entrada y de salida mostrados en la figura I.1.5.17
Fig. I.1.5.20

PASO 2: PASO DE PROGRAMA :

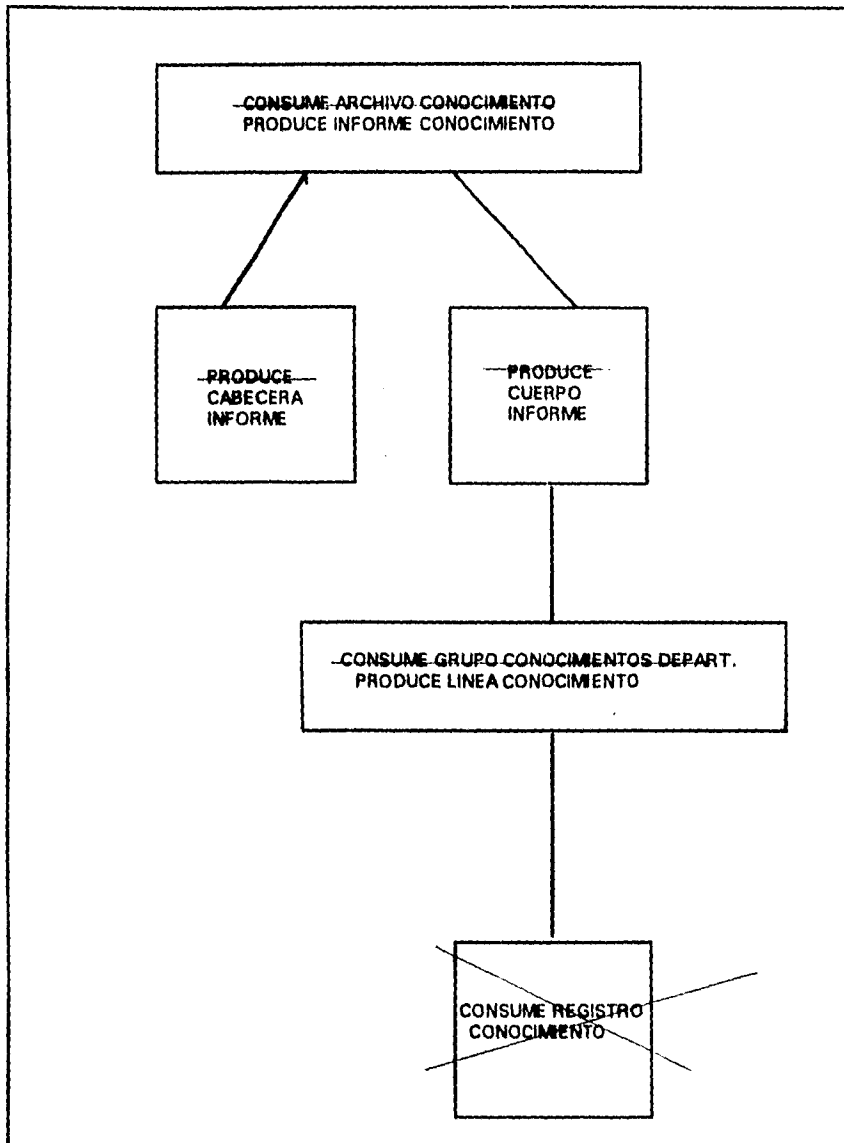
El segundo paso es combinar las estructuras de datos en una sola estructura de programa. Hay dos partes en el paso de programación. Primero, todas las correspondencias entre los componentes de la estructura de datos se identifican encontrando la relación consumo/producción entre los componentes. Se construye la estructura del programa para que se corresponda con esas relaciones.

La segunda parte consiste en verificar la corrección de la estructura de programa. Esto se consigue reduciendo la estructura del programa a sus estructuras de datos individuales. El proceso de verificación se aplica a cada corriente de datos para demostrar que el programa se puede reducir a cada una de sus estructuras de datos. Si esto puede hacerse , se asume que el diseño es correcto.

PASO 3: PASO DE OPERACIONES.

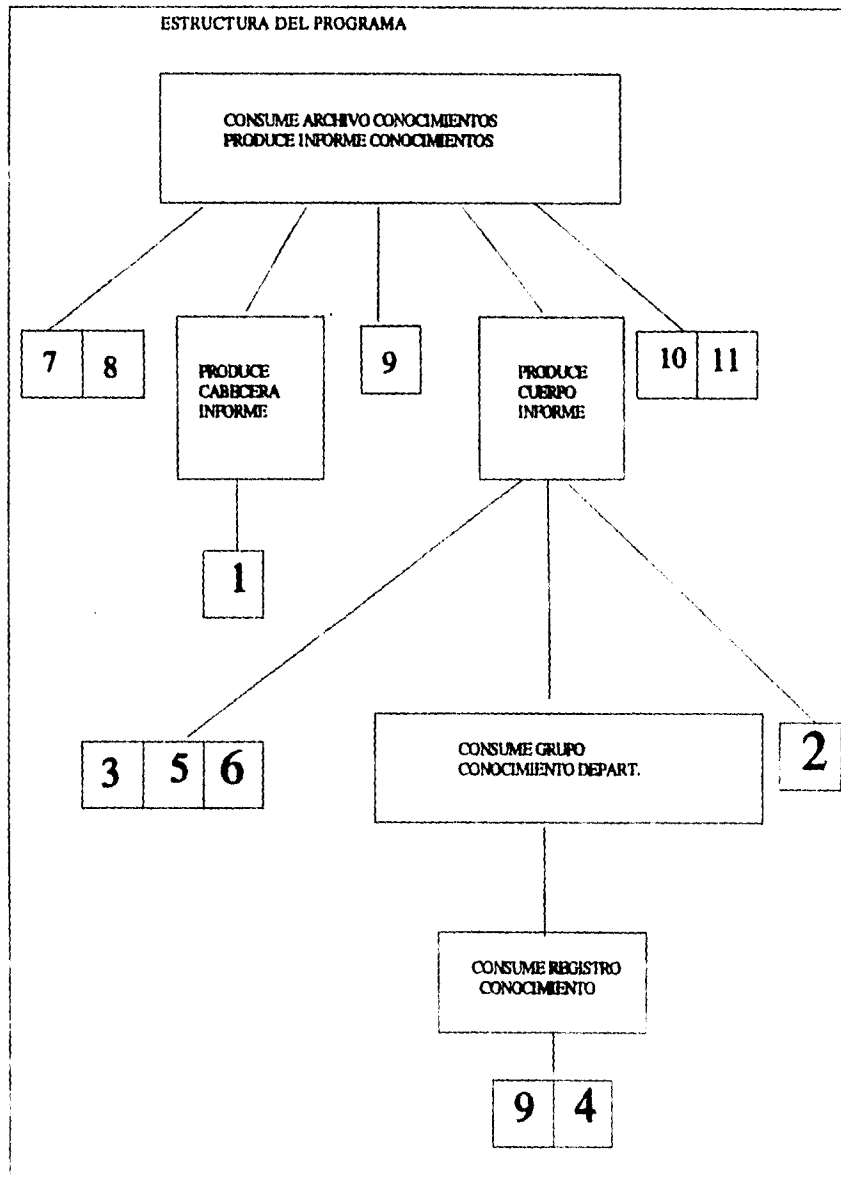
El paso de operaciones se compone de tres partes:

- 1.- Se listan las operaciones ejecutables requeridas por el programa para convertir su entrada en salida; esto se hace empezando en la salida y yendo hacia atrás hasta la entrada. Las operaciones se determinan por referencia a las estructuras de datos, las cuales, a su vez, se reflejan en la estructura del programa.
- 2.- Cada operación se asigna al lugar apropiado de la estructura del programa.
- 3.- Se verifica la corrección comprobando si se producen todas las salidas y se consumen todas las entradas. La figura anterior muestra la estructura del programa incrementa con sus pasos de operación.



La exactitud de la estructura del programa se demuestra reduciéndolo a sus estructuras de datos. En el ejemplo la estructura del programa se reduce a la corriente de datos de salida, INFORME CONOCIMIENTOS.

Fig. I.1.5.21



En el paso de operaciones se listan las operaciones ejecutables requeridas por el programa de conocimientos de los empleados.

Fig. I.1.5.22

PASO 4: PASO DEL TEXTO.

El último paso en el diseño de Jackson es el de texto. En este paso, el diagrama de estructura del programa, junto con sus correspondientes operaciones, se traduce a un texto de estructura.

El texto de estructura es un tipo formal de pseudocódigo. Tiene reglas estrictas para escribir el control de la secuencia, la selección y la iteración, pero las instrucciones a nivel de programación elemental se escriben informalmente.

Las condiciones de control que rigen la ejecución de las estructuras selección e iteración se añaden a la estructura de texto en este paso. Al final del paso de texto, el diseño de programa está listo par su implantación. El diagrama de red del sistema, la estructura en árbol de datos, la estructura en árbol del programa y la estructura de texto conforman todo el paquete de diseño.

PROCEDIMIENTO DE DISEÑO BÁSICO.

El diseño de Jackson descrito antes se denomina proceso de diseño básico. Puede utilizarse en el diseño de programas simples, pero no es directamente aplicable a programas complejos. Un programa complejo ha de dividirse antes en una secuencia de programas más simples, cada uno de los cuales se puede diseñar utilizando el procedimiento de diseño básico. La resolución de colisiones y la inversión de programas son técnicas incluidas en el proceso de diseño de Jackson para conducir el diseño de programas complejos. La siguiente figura muestra una visión más completa de los pasos del proceso de diseño de Jackson.

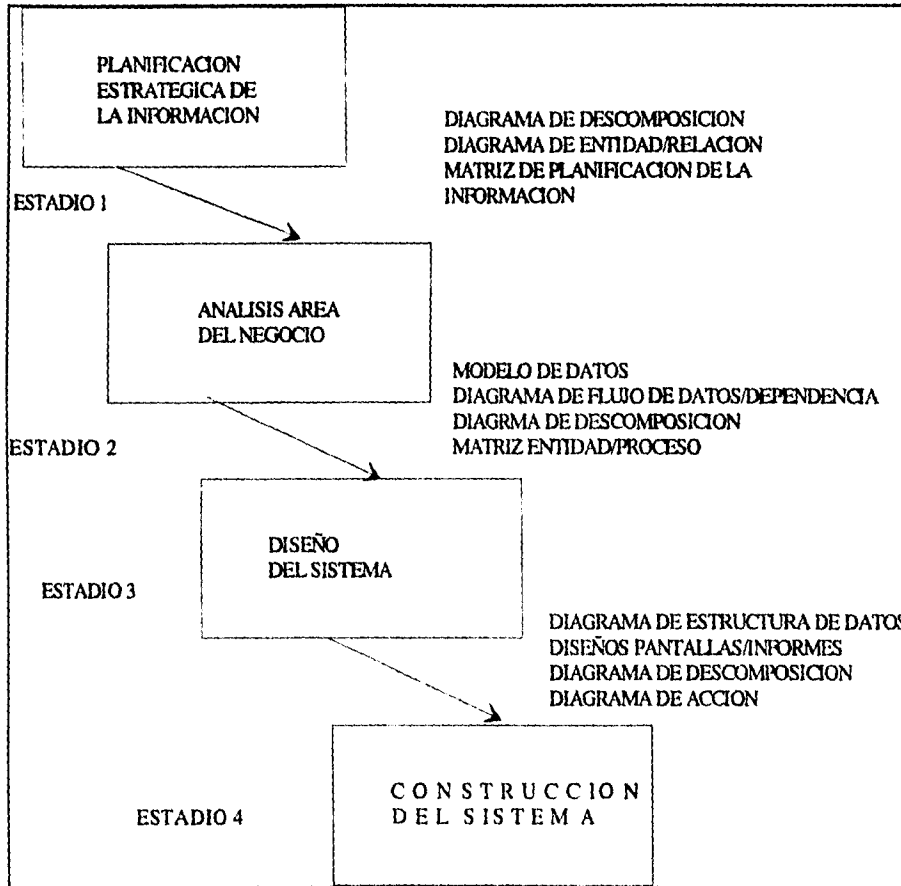
El proceso de diseño de Jackson se ha extendido al manejo de diseño de sistema, además de programas, al diseño de sistemas de tiempo real y a los sistemas de información.

METODOLOGÍA DE LA INGENIERÍA DE LA INFORMACIÓN DE MARTÍN.

La metodología de la ingeniería de la información adoptada por James Martín es una formulación descendente por pasos para la construcción de sistemas de información. Los primeros pasos de esta metodología se centran en la información y el modelo de la empresa a alto nivel. El objetivo es gestionar el desarrollo del sistema y la interacción a través del control de los datos. Los datos compartidos en el sistema de información se definen en un modelo lógico y se controlan centralmente. Así, la metodología de Martín revisa todo el sistema en el intento de identificar cómo se utiliza y se comparte la información. Las funciones de alto nivel de la empresa se definen junto con la información del negocio. En los pasos posteriores, se definen con más detalles los datos y las funciones. En el último paso, se definen e implantan los datos y lógica del programa.

Las estructuras del programa se construyen en el nivel más alto de un modelo de la empresa que establece una infraestructura común de información que aúna los sistemas de información utilizados en toda la organización. Así pues, los procedimientos se derivan de los datos. El diseño lógico está separado del diseño físico y lo precede.

La siguiente figura muestra los cuatro pasos principales de la Metodología de la ingeniería de la información de Martín.



La ingeniería de la información es un proceso descendente y orientado hacia los datos para el desarrollo de los sistemas. Arranca con la planificación estratégica de la información.

Fig. 1.1.5.23

Y en el siguiente cuadro se muestra el resumen de los tipos de diagramas estructurados utilizados en la metodología de la ingeniería de la información de Martin.

LOS DIAGRAMAS DE LA INGENIERÍA DE LA INFORMACION

- **DIAGRAMA DE ENTIDAD/RELACION:** muestra los tipos de entidades y sus relaciones.
- **DIAGRAMA DE LA ESTRUCTURA DE LOS DATOS:** Muestra los tipos de entidades, sus atributos y sus relaciones.
- **DIAGRAMA DE DESCOMPOSICION JERARQUICA ESTRUCTURADA EN ARBOL:** Muestra el modelo corporativo de alto nivel y la estructura arquitectónica jerárquica de las actividades del negocio, de los procesos, de las aplicaciones, de los procedimientos y de los programas.
- **DIAGRAMA DE FLUJO DE DATOS Y DE DEPENDENCIA:** Muestran el flujo de los datos entre procesos.
- **DISEÑOS DE PANTALLA Y DE INFORMES:** Muestran el diseño del interfaz del usuario.
- **DIAGRAMAS DE ACCIÓN:** Muestran detalladamente la lógica de los programas.

ESTADIO 1: PLANIFICACIÓN DE LA ESTRATEGIA DE LA INFORMACION:

Este estado comienza con la creación de un plan estratégico de los sistemas del negocio en el que se definen los objetivos del negocio para los próximos cinco a diez años. Junto con el plan estratégico se construyen los modelos de alto nivel y datos de la empresa. El modelo de la empresa define las funciones básicas (por ejemplo, VENTAS) y la estructura de la organización. Un diagrama estructurado en árbol jerárquico, llamado diagrama de descomposición, es el que se emplea para definir las funciones del negocio y las estructuras de la organización. Para definir los tipos de datos de la organización se emplea un diagrama de entidad/relación como el que se muestra en la siguiente figura:

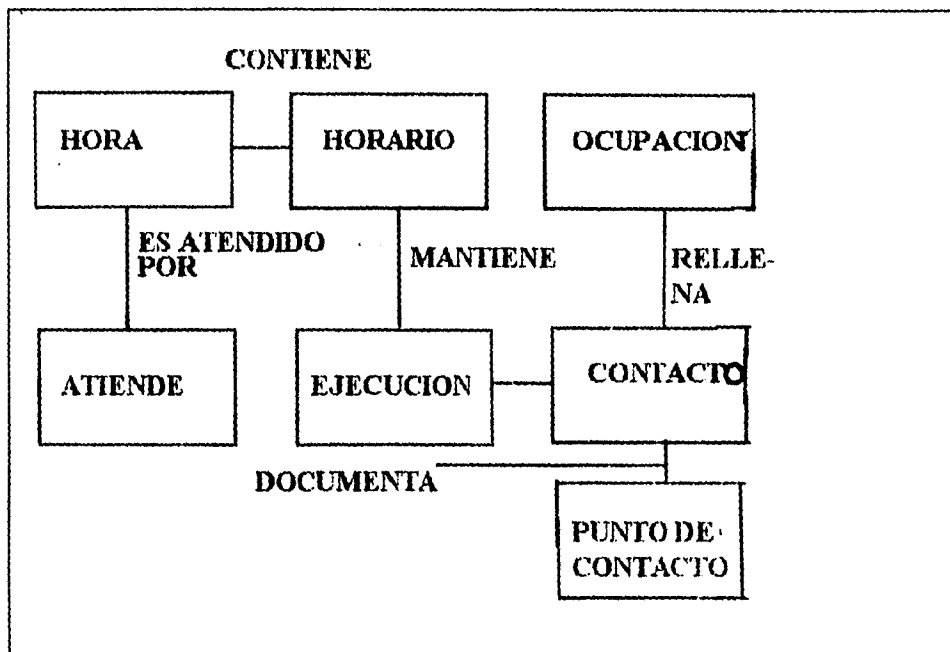


Diagrama de Entidad/Relación .
Fig. I.1.5.24.

Una entidad es cualquier información de la empresa que se pueda almacenar (como empleados, departamentos, clientes, facturas, etc.). Las entidades se descubren entrevistando los usuarios del sistema de información . Las visiones de los usuarios se analizan y luego se mezclan y clasifican para crear un modelo compuesto. Además, la ingeniería inversa de los sistemas existentes puede emplearse para revelar el modelo de datos existente, el cual puede utilizarse como punto de partida para crear el modelo necesario para soportar la dirección estratégica de la empresa.

Finalmente , se determina en este estado el orden en el cual se van a desarrollar los sistemas de información que satisfagan los objetivos de la empresa.

El usuario tiene una importante función activa en este estado y en todo el proceso de desarrollo de los sistemas de ingeniería de la información.

ESTADIO 2: EL ANÁLISIS DEL ÁREA DEL NEGOCIO

Las funciones del negocio definidas en el estado 1 se usan para dividir la empresa en áreas lógicas del negocio (por ejemplo, COMPRAS). Después se definen y construyen los sistemas de información necesarios para soportar un área del negocio en particular.

Este estado se concentra en la definición de los datos y procesos necesarios para satisfacer los objetivos de la empresa dentro de un área en particular del negocio. Una parte del diagrama entidad/relación desarrollada en el estado 1 que concierne a un área en particular del negocio se extrae y desarrolla en un modelo de datos completamente **normalizado**. Se definen completamente las entidades, sus atributos y sus relaciones recíprocas. El diagrama de descomposición del estado 1 que representa las funciones del negocio, se descompone en procesos por cada una de las áreas del negocio. El proceso puede representarse en diagramas de dependencia (un tipo de diagrama de flujo de datos), en diagramas de flujo de datos o en diagramas de descomposición como se ve en la siguiente figura. Se emplea una matriz de entidad/proceso para relacionar los datos con los procesos en los que intervienen.

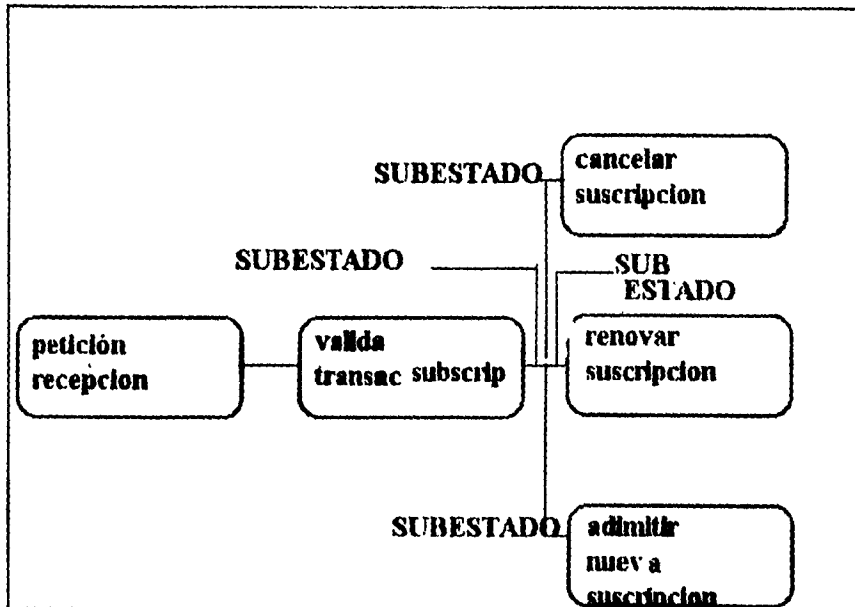


Diagrama de dependencia, un tipo especial de diagrama de flujo de datos.
Fig. 1.1.5.25.

ESTADIO 3: EL DISEÑO DEL SISTEMA

El estado 3 se ocupa de las consideraciones de diseño del sistema lógico. Los procesos del negocio definidos en el estado 2 conforman las bases para el desarrollo del sistema en este estado. Se diseñan los procedimientos necesarios para realizar los procesos y las estructuras lógicas en sus datos. Se utiliza la descomposición funcional descendente para diseñar los procesos. En este estado, la metodología de la ingeniería de la información se hace similar a la metodología tradicional de la ingeniería de software para el desarrollo de programas. El sistema diseñado se representa con diagramas de estructura de datos, con diseños y con diagramas de acción y descomposición. Los diagramas de acción muestran detalladamente la lógica del procedimiento y los accesos a las bases de datos.

El prototipo del interfaz de usuario (en pantallas, diálogo o gráficos) se emplea para descubrir los requerimientos del usuario. Es necesaria la implicación del usuario para ayudar a construir y comprobar los modelos de prototipos.

ESTADIO 4: LA CONSTRUCCIÓN

Mientras que el estado 3 trataba el diseño del sistema lógico, el estado 4 se ocupa del diseño del sistema físico y de la implantación del programa. Durante el estado 4, toda la información del sistema se entrega como una base de datos lógica, una base de datos física y un código documentado de trabajo. Las herramientas C.A.S.E., como los lenguajes de cuarta generación, herramientas de soporte de decisiones y las generaciones de códigos, se emplean en el estado 4.

¿QUE ES LA INGENIERIA DE INFORMACION ?

El término, Ingeniería de INFORMACION, se refiere al conjunto integrado de metodologías, diseñadas para crear y operar en el ambiente descrito anteriormente. Se apoya en modelos de datos completamente normalizados de la información de un proyecto. Estos modelos se mantiene en un ambiente sistematizado, utiliza herramientas computarizadas para construir las aplicaciones que se utilizan para el manejo de la información.

Estas herramientas están diseñadas para automatizar el análisis del sistema, aplicaciones de desarrollo, etc., tan rápidamente como sea posible, y hacer aplicaciones fáciles de mantener. Un mayor objetivo, es evitar el desarrollo de problemas desordenados con un mantenimiento convencional.

Mientras que las técnicas estructuradas se aplican a un sistema en lo particular, la Ingeniería de Información, aplica técnicas estructuradas a un proyecto de modo que los sistemas diseñados por equipos separados sean adecuados a una estructura diseñada. El concepto de la Ingeniería de Información contrasta con el término de la Ingeniería de Software. La idea principal de la Ingeniería de Software, es entorno a la lógica del proceso computarizado; va muy de la mano, con las técnicas estructuradas para diseño específico y escribe programas. Por otro lado, la idea base para la Ingeniería de Información es, en sí, la información, que podemos manejar de los datos que se encuentran almacenados y mantenidos por computadoras.

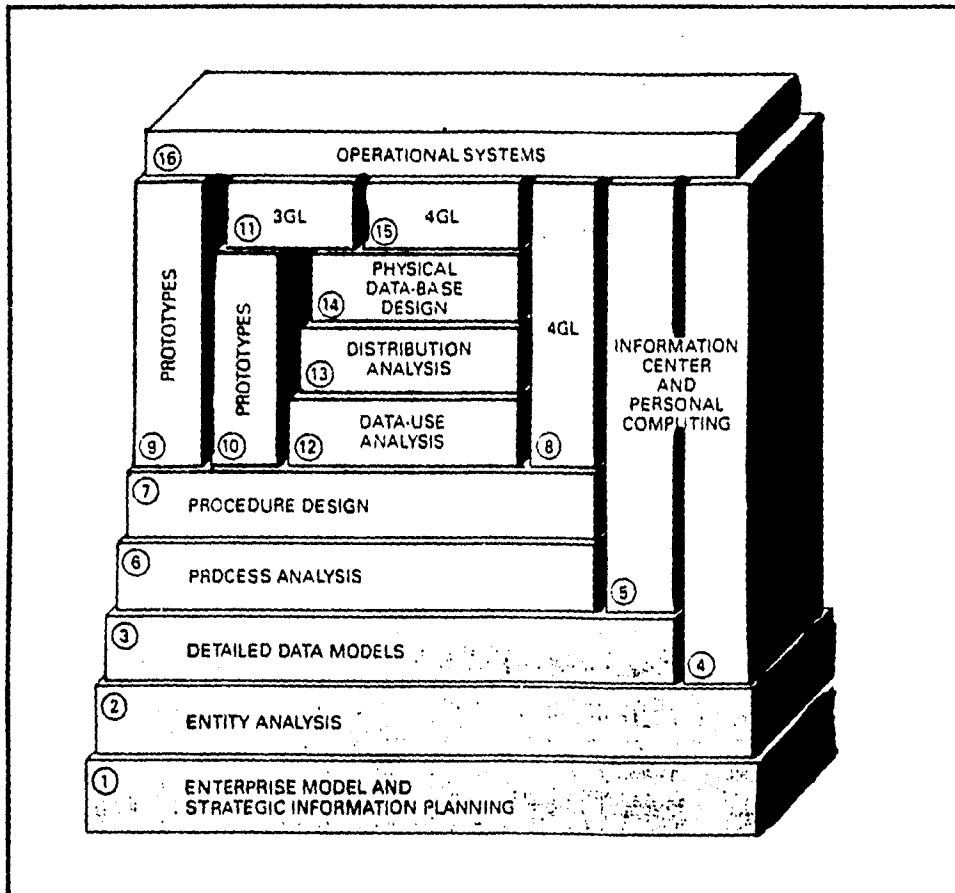
La Ingeniería de Información se apoya en el análisis y código estructurado, pero busca encontrar herramientas como los lenguajes no-procedurales, especificación de lenguajes, generadores de aplicación, diseño asistido por computadora, que minimizan el trabajo de desarrollo de programas.

CONSTRUYENDO BLOQUES DE INGENIERIA DE INFORMACION

La ingeniería de información establece un entrelazamiento entre las metodologías como se muestra en la figura siguiente. En este diagrama, cada bloque es dependiente del que se encuentra abajo de él. El diagrama está dibujado como un conjunto de bloques de edificios, debido a que los bloques se ensamblan en diferentes modos dependiendo de los diferentes analistas.

BLOQUE 1

Bloque en donde la planeación estratégica soporta a los demás. Un modelo jerárquico del proyecto es dibujado, los objetivos de cada uno de sus componentes se establece, y se intenta determinar aquella información que será necesaria para permitir al proyecto lograr sus objetivos.



Bloques que construyen a la Ingeniería de información
Fig. 1.1.5.26

BLOQUE 2

Aquí es creado un esbozo del mapa de la información necesaria para desarrollarse el proyecto. Aplicando un análisis Top-Down para los tipos de datos que deberán ser guardados, así como la forma en que se van a relacionar con los otros. El análisis de información es algunas veces hecho a través del un modelo completo o puede un modelo dividido.

BLOQUE 3

El tercer bloque es para el modelado de datos. El análisis de información estudia el tipo de datos necesarios, creado un modelo de información, el cual deberá ser un esbozo amplio pero, tomando en cuenta que no contiene todos los detalles necesarios para la instrumentación de una base de datos. El modelado de datos crea el diseño lógico detallado de la base de datos, intentando hacerla tan estable como sea posible antes de que está se instrumentada. El bloque 3 es una extensión del bloque 2, pero a un nivel de mayor detalle y aplica diversos chequeos buscando el mayor grado de confiabilidad.

Una de las realizaciones mas importante que con lleva la ingeniería de información, es la existencia de una organización la cual pueda ser descrita independientemente de como este siendo usada la información. Aunado a la necesidad de que la información sea estructurada.

La información tiene ciertas propiedades inherentes que son guían para corregir la estructuración.

Estas propiedades son, independientes a a la forma como se estén utilizando la información. Si estructuráramos la información de un modo que violara las propiedades inherentes del proyecto, es probable que tendríamos que rehacerla en el futuro. Rediseñar siempre será mas caro, porque programas que utilizan la estructura tendrán que ser reescrito. Si la información es estructurada de acuerdo con sus propiedades, tendremos un diseño estable.

BLOQUE 4 y 5

Muchos de los nuevos lenguajes y generadores, son empleados por los usuarios. Esta participación directa de los usuarios puede traer una mejora vital, necesaria en el procesamiento de los datos, siempre y cuando no puedan desarrollar estructuras propias fuera de las establecidas. En muchos casos, los usuarios están contentos de su liberación del Centro de Procesamiento de Datos. Cuando la información es compartida, los lenguajes usados por los usuarios deberán estar estrechamente vinculados con el modelo de datos, representados por bloques 4 y 5.

En algunos casos, la computación personal y la del centro de cómputo emplean datos que no han sido adecuadamente normalizados y esto por lo regular no son representados en los modelos de datos detallados. Por esta razón, el bloque 4 descansa en bloque 2. Por otra parte, cuando la información necesita estar en los modelos detallados de los administradores de información, el usuario sólo debe emplearlos. Esto salvará al usuario, de tener que diseñar su propia información, lo que conlleva un tiempo, y ayudará para garantizar el intercambio de información entre sistemas desarrollados por separado. Cuando sea posible, la computación orientada a los usuarios, deberán buscar que el bloque 5 descansa en el bloque 3.

BLOQUE 6

El Bloque 6 y los que descansen sobre él, se relacionan con el trabajo del profesional en el análisis de sistemas. El bloque 6 se refiere al análisis de una área específica del proyecto. Las funciones del área son descompuestas en procesos menores, utilizando los diagramas de descomposición. Los diagramas de dependencia o de flujo de datos, puede ser dibujados, para mostrar la interacción de los diferentes procesos.

En algunos casos, los procesos de análisis están estrechamente vinculados a la creación de modelos de datos. Mientras los procesos están siendo examinados, los tipos de entidad necesitan ser considerados. Los datos deben ser determinados y sintetizados dentro de un modelo de datos totalmente normalizado. Podemos revisar la exactitud e integridad del modelo de datos que se ha creado con alguno que ya exista dentro de la empresa.

BLOQUE 7

El bloque 7 está íntimamente relacionado con el diseño de los procedimientos.

Un diagrama de acción de los procedimientos es creado. Este puede ser desde un diagrama de navegación construido en el modelo de datos. Árboles o mesas de decisión pueden ser utilizados para que el diseño lógico sea apropiada.

Los diagramas de acción pueden ser generados directamente desde el correcto dibujo de los diagramas de descomposición, diagramas de dependencia, diagramas de navegación de información, árboles de decisión, o diagramas de estado de transición. Para acelerar el trabajo del diseñador es aconsejable usar, las herramientas de gráficos automatizadas las cuales permiten diagramar rápidamente los dibujos y estados de cambio. Las herramientas deberán imponer rigor en lo dibujado; Automáticamente se transforman las diversas representaciones en diagramas de acción; Facilitar el corte, sintaxis y edición de diagramas de acción; transforman los diagramas codificados.

BLOQUE 8 y 9

El diseño en el bloque 7 puede ser implementado directamente en un lenguaje de cuarta-generación, tal como FOCUS, RAMIS II, IDEAL, Application Factory, NATURAL, o MANTIS.

Alternativamente, estos generadores pueden ser utilizados para crear prototipos. Los prototipos pueden ser modificados sucesivamente y eventualmente codificados con el tiempo.

BLOQUE 10 y 11

En otros casos, el prototipo puede desear tener performance de desempeño máquina, insuficiente para el sistema final. El prototipo puede ser convertido a COBOL o algunos otros lenguajes de alto-nivel logrando una mayor optimización de desempeño.

BLOQUE 12

El análisis del uso de datos provee un modo formal de coleccionar y diagramar la información usada y lista para el diseño de la base de datos física. El modelado de los datos, resulta en un diseño lógico de una base de datos. Una variedad de decisiones tiene que estar hecho antes de que el diseño este instrumentado físicamente. Las decisiones dependen del estado de la información, es probable que se utilizen las trayectorias de uso a través de la base de datos, y sus volúmenes de uso y respuesta.

BLOQUE 13

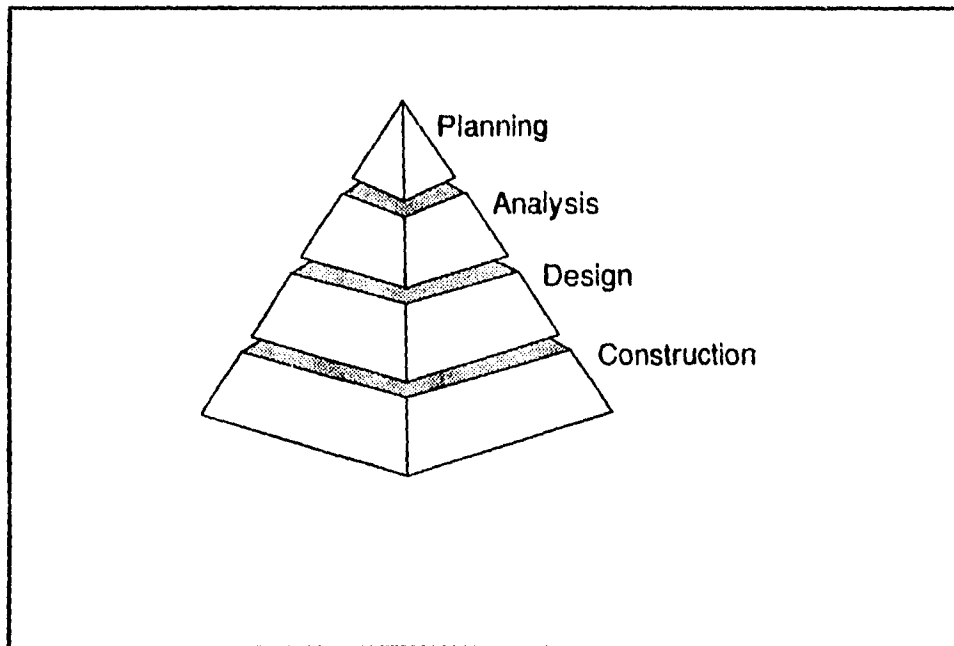
Los modelos de datos pueden o no ser divididos para su implementación en bases de datos separadas. Esto puede ser hecho por razones de "performance".

BLOQUE 14

Este bloque se relaciona con la conversión de los modelos de datos y procedimientos dentro del diseño de base de datos físicas.

I.2.1 MODULO DE PLANEACION

Comprender la importancia de planear la información, puede ayudar en un proyecto a largo tiempo para evaluar los datos que se requieran y poder hacer un mejor uso de las Bases de Datos y programas existentes. La planeación dinámica puede mantener un alto nivel de anticipación de las funciones del proyecto, su orientación para la dirección de negocios, así como los sucesos críticos, datos y toda la información necesaria. También nos ayuda a definir y otorgar prioridades a los proyectos para su adecuada implementación o futuro estudio.



Localización de la Planeación en el ciclo de vida del desarrollo de Sistemas
Fig. I.2.1.1

El módulo de planeación (IEW/PWS), es un conjunto integrado de herramientas, desarrolladas para trabajar en una computadora personal (PC), desde donde es soportada la planeación de la información.

Toda la información que se desarrollará desde la pantalla de trabajo, es almacenada en una ENCICLOPEDIA. Es llamada enciclopedia, porque - a diferencia de un diccionario de datos - desarrolla de manera lógica, la comprobación y asegura de no duplicar datos. Los modelos de objetos creados y afinados, son almacenados una sola vez.

El módulo de planeación, ayuda a capturar los factores de sucesos críticos y los objetivos de la dirección del negocio, desarrollando vistas de alto nivel sobre el proyecto, para funciones de datos y toda la información necesaria. Existen algunas tareas que complementan el uso del módulo de planeación.

1. Desarrollo de modelos de un proyecto en términos de su organización, planes estratégicos y actividades.
2. Identificación de la información necesaria para soportar el cumplimiento del proyecto.
3. Establecer el entorno del sistema de información, desarrollado a través del proyecto.
4. Conceder prioridades a los sistemas desarrollados, basándose en estratégicos planes de negocios.

El módulo de planeación ayuda a determinar que sistemas de información realmente son necesarios para poder cumplir los objetivos planteados.

El módulo es usado para afinar diagramas de entidad y desintegración de procesos. Una vez que han sido afinados, podemos usar el módulo de planeación, para iniciar la descripción de como se realizará la implementación de la información a un nivel físico.

Cualquier información almacenada en la Enciclopedia, puede ser copiada y compartida por varios miembros de un proyecto, pueden trabajar independientemente en los diferentes componentes del IEW/PWS.

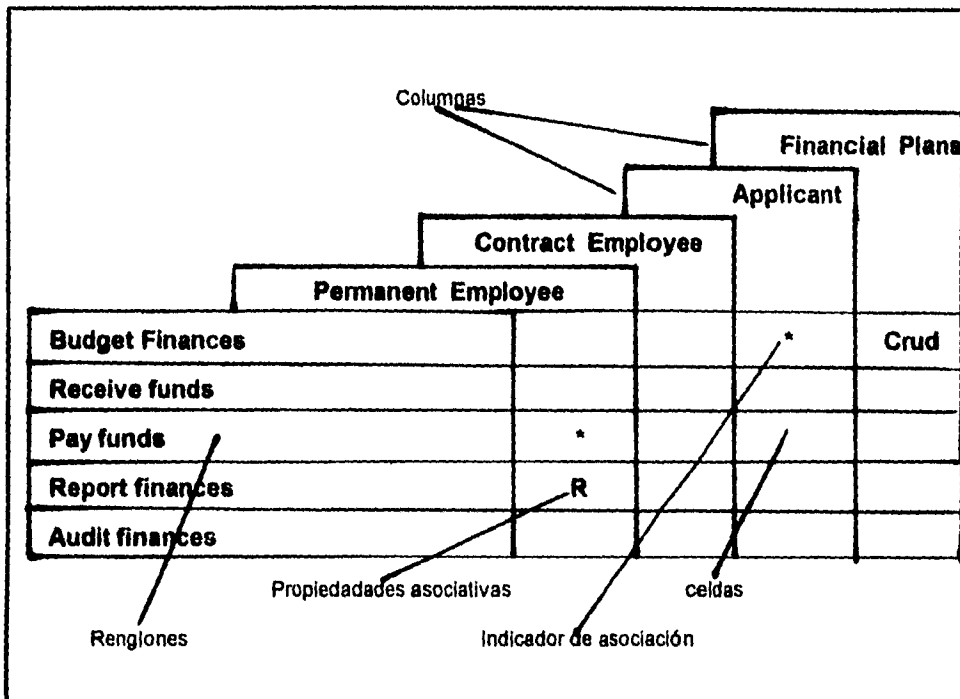
El módulo de planeación, consiste de las siguientes herramientas:

1. Diagrama de matriz
2. Diagramas de entidad
3. Diagramas de desintegración

I.2.1.1 DIAGRAMAS DE MATRIZ DE ASOCIACIÓN

El uso de los diagramas de matriz de asociación, es para la rápida creación y desplegado de las ligas asociativas entre los objetos.

Podemos usar los diagramas, para mostrar cada instancia de dos tipos de objetos asociados (por ejemplo, todos los tipos de entidades y todos los procesos), o dos instancias en particular (por ejemplo, la entidad cliente récord, involucra en el proceso a el cliente x).



Muestra de la "acción" propia para un proceso que involucra a un tipo de entidad asociativa
 Fig. I.2.1.1.1

Ciertas asociaciones son el "todo o nada" para una instancia. Una colección de datos pueden ser una composición de otra colección de datos o no, otra asociación puede tener atributo de grado o cosa. Por ejemplo, podemos especificar directamente o indirectamente el impacto de cierto problema, o definir el porcentaje del impacto para varias partes. El grado de impacto, es una propiedad de la asociación

Componentes de un Diagrama de Matriz de Asociación

Renglones:

Audit Finances
Administer taxes
Manage invest

Un renglón cruza horizontalmente a través de el diagrama. Cada renglón contiene una instancia. Todos los renglones se componen de la misma instancia para una asociación.

Columnas:

Contract Employee
Permanent Employee

Una columna cruza verticalmente el diagrama. Cada columna contiene una instancia (objeto). Todas las columnas contienen el mismo tipo de objeto para una asociación.

Celda:

*	*	
		*

Una celda es formada por la intersección de un renglón y una columna. Un caracter o caracteres en una celda, indican si existe alguna asociación entre los objetos del renglón y la columna, pudiendo tener cualquier tipo de asociación.

Asociaciones y sus propiedades

Para asociar objetos en un diagrama de matriz de asociación, debemos marcar el cruce de un renglón y columna con un caracter. El caracter puede ser un asterisco o un código alfabético, con lo cual se define el tipo de asociación.

Para desarrollar una matriz de asociación, debemos abrirla , y luego agregar o eliminar los objetos. La matriz establece una asociación entre los objetos por los renglones y columnas, además de las prioridades para éstas.

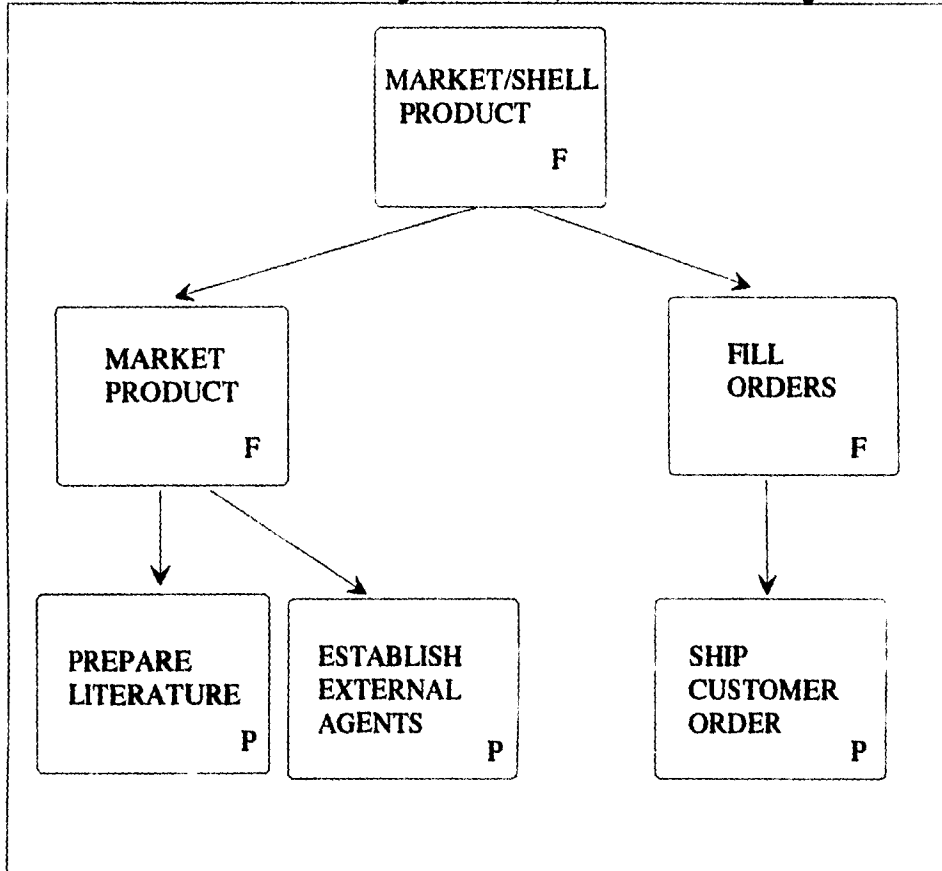
I.2.1.2 DIAGRAMAS DE DESINTEGRACIÓN

Frecuentemente moldeamos una forma de desintegración. En lo general, la desintegración describe a un objeto como, la composición de otro objeto o del mismo tipo, el cual en su turno, puede ser compuesto de aún más objetos del mismo tipo y así sucesivamente. Tenemos una sola excepción en un diagrama de desintegración para una función. Una función puede ser compuesta de subfunciones o procesos. Por ejemplo, un diseñador puede descomponer funciones de alto nivel en funciones de bajo nivel, para un mayor detalle de las funciones o procesos.

El módulo de planeación usa los siguientes objetos:

- 1) Colección de datos
- 2) Funciones
- 3) Objetivos
- 4) Información necesaria
- 5) Posición
- 6) Mecanismos
- 7) Unidades organizativas
- 8) Problemas
- 9) Procesos
- 10) Proyectos
- 11) Áreas subordinada

Podemos mostrar a través de un diagrama sencillo, los niveles de desintegración.



La función raíz "Market/Shell Product" es descompuesta en dos subfunciones
Fig. 1.2.1.2.1

Para este ejemplo, "market/sell product" es el máximo nivel de proceso u objeto raíz, apareciendo en la parte superior del diagrama. La inmediata raíz descendiente o hijos son "market product" y "fill orders". Los hijos son niveles inferiores en el diagrama de descomposición.

I.2.1.2.1 ICONOS PARA EL USO DEL DIAGRAMA DE DESCOMPOSICIÓN

Un diagrama de descomposición puede ser representado, como una estructura de árbol, en cualquier de los tipos de objetos listados previamente.

1. Desintegración de objetos :



Toda la desintegración de objetos, excepto funciones y procesos son representados por un ícono rectangular.

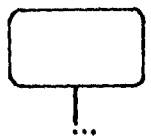


Las funciones son representadas por un ícono que contiene a la letra 'F' en la esquina inferior derecha.

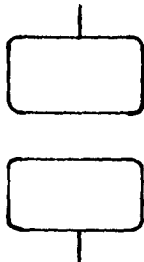


Los procesos son representados por un ícono que contiene a la letra 'P' en la esquina inferior derecha.

2. Empacado de Objetos



Cualquier descomposición de objetos puede ser contractada o empacada. Una pequeña línea y una elipsis aparecerán en la parte inferior del bloque, indicando que oculta a sus hijos.

3. Ocultamiento de parentesco o descendencia

Un objeto puede tener hijos ocultos, porque el nivel del diagrama es limitado. Una línea en la parte inferior del ícono indica uno o más hijos ocultos, y en la parte superior uno o más padres ocultos.

4. Fantasmas

Los fantasmas son objetos que han sido agregados a los diagramas, los cuales no son raíces o no han sido ligados a otros objetos. Un ícono fantasma, aparece con una 'X' desvanecida.

1.2.1.2.2. Desarrollando un diagrama de desintegración

Un diagrama de desintegración puede ser muy simple, mostrando sólo algunos objetos y su asociación, o extendiendo a muchos niveles de desintegración.

Podemos realizar:

- 1) Abrir un diagrama de descomposición
- 2) Agregar objetos a el diagrama
- 3) Establecer asociación de parentesco entre objetos
- 4) Especificar el número de niveles a desplegar
- 5) Cambiar una función a un proceso y viceversa

1.2.1.3 DIAGRAMAS DE ENTIDAD

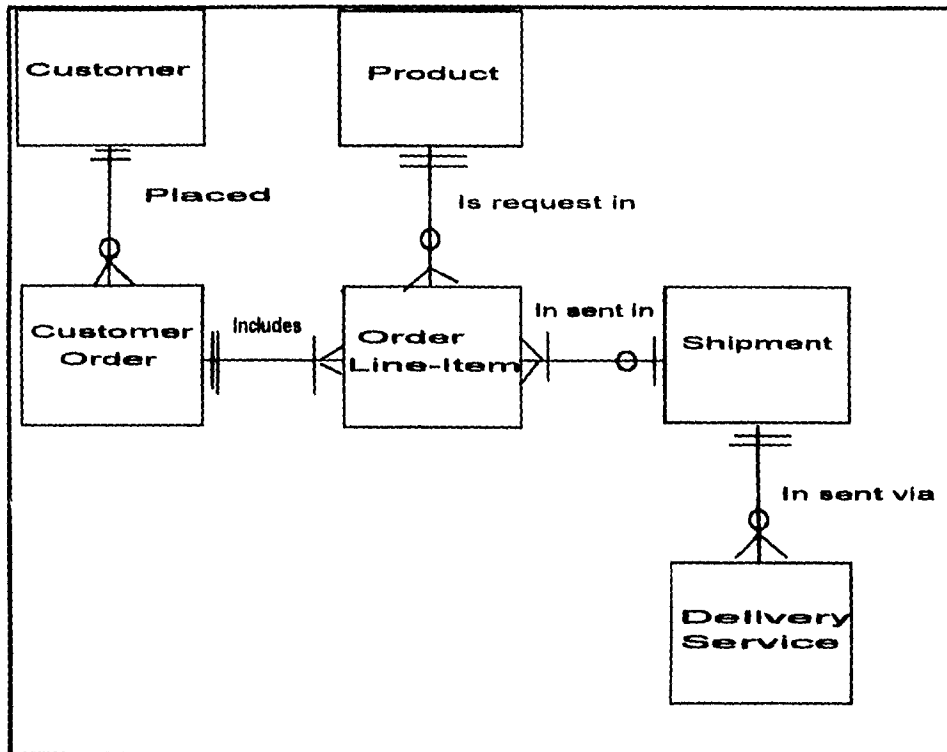
Los *diagramas de entidad* son usados en la planeación "top-down" y para el modelado detallado de datos. La planificación "top-down" identifica al tipo de entidad principal; involucrado en la ejecución, además construye y determina la mejor relación entre las entidades.

El modelado de datos puede ser usado para afinar los diagramas de entidad creados durante la planeación. Podemos usar los diagramas de entidad, ya sea para planeación y/o el análisis del módulo de planeación. Para el desarrollo de modelos, a través del análisis, podemos construir Bases de Datos relacionales.

Un diagrama de entidad (algunas veces llamado como el diagrama de entidades relacionales) nos muestra los tipos de entidades desarrolladas en el proyecto y como se están relacionando. Podemos desarrollar las propiedades de entidad y relación a través del diagrama relacional y los diálogos descritos en los capítulos previos y posteriores.

La totalidad de las entidades contenidas en la "ENCICLOPEDIA", así como, cada una de sus relaciones, comprende el "Modelo Entidad".

Podemos crear diagramas de entidad para áreas principales, procesos y colección de datos. Estos diagramas representan la vista ("view"), de cada uno de los objetos que componen el modelo entidad.



Entidades y algunos de sus tipos de relaciones
Fig. 1.2.1.3.1

1.2.1.3.1. COMPONENTES DE UN DIAGRAMA ENTIDAD

Para los diagramas de entidad existen dos componentes básicas (tipo de entidad y tipo de relación); la cardinalidad obligada nos dice cual es la limitación numérica de la relación.

Los iconos de entidad, son rectángulos que contienen el nombre del tipo entidad. El tipo de relación, aparece como una línea entre los tipos de entidad; la cardinalidad, es localizada al final de la línea de relación.

I.2.1.3.1.1 TIPOS DE ENTIDADES



Tipo de Entidad

Un tipo de entidad, es una clase de gente, un lugar, una cosa o concepto, los cuales tienen una característica de interés para el proyecto. Fundamentos y atributos usan los iconos de tipos de entidades.



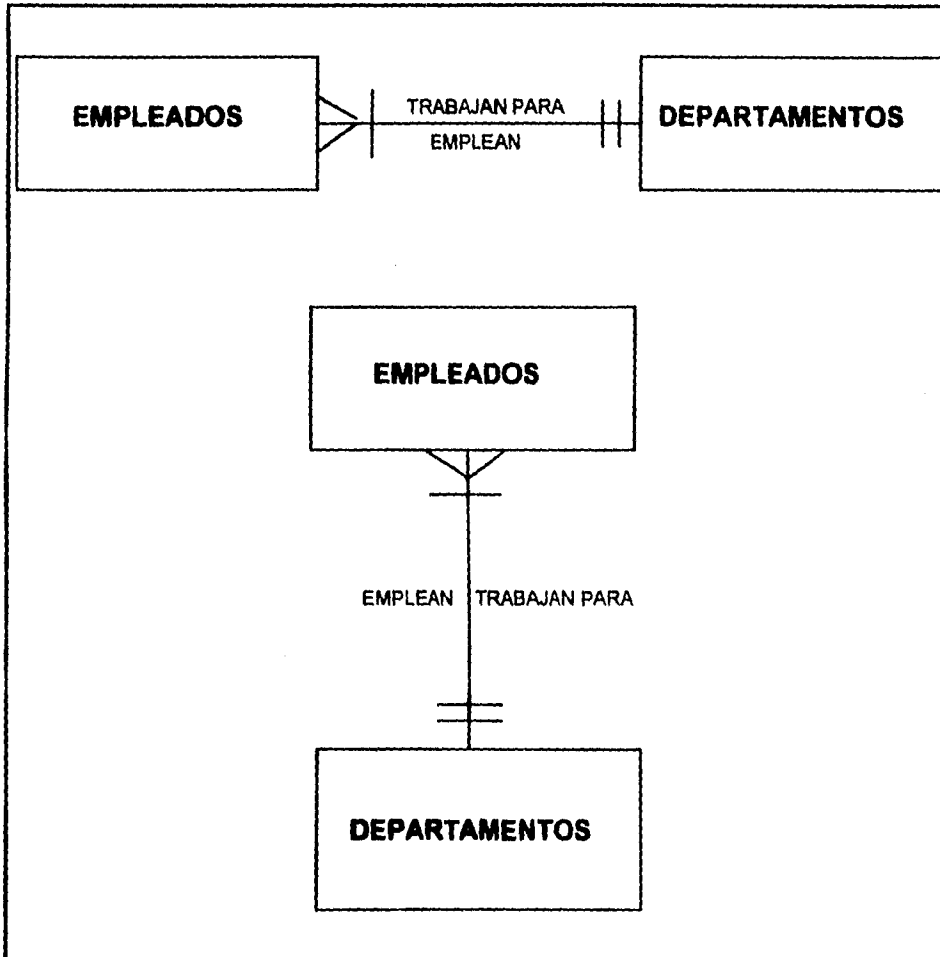
Entidad Asociativa

La asociación de un tipo de entidad existe primeramente para interrelacionar tipos de entidades. Se recomienda usar este ícono, aunque podemos usar cualquiera de los opcionales en el menú.

I.2.1.3.1.2 TIPOS DE RELACIONES

El tipo de relación aparece como una importante línea entre dos entidades. La unión de entidades establece la naturaleza de la relación. Por ejemplo; la relación, que se da entre un "Empleado" y un "Departamento", podría ser: "Trabaja para", con lo que se formaría la frase, "El empleado trabaja para el departamento". Mientras los empleados, "trabajan para" un departamento; el departamento también "emplea" a empleados y en realidad cada tipo de relación tiene dos tipos de nombre, los cuales pueden ser leídos en cada entidad, como el proyecto u objetivo de la frase:

Cuando la acción va de izquierda a derecha, aparece arriba de la línea, y la que va de derecha a izquierda, abajo de la línea. Si la relación es vertical, la acción que va de arriba hacia abajo queda descrita del lado derecho y para el caso de ir de abajo hacia arriba la acción se describe a la izquierda.



Descriptivo de las relaciones entre entidades
Fig. 1.2.1.3.1.2.1

I.2.1.3.1.3 CONSTRUYENDO LA CARDINALIDAD

La cardinalidad especifica el número de instancias para un tipo de relación que puede ocurrir entre dos entidades relacionadas. Cada cardinalidad construida, se define con un mínimo y un máximo; el valor ímimo puede ser un 0 o cualquier valor entero positivo, el máximo debe ser cualquier entero positivo o "M" (muchos), lo que significa sin límite, el máximo debe ser más grande que el mínimo.

La siguiente tabla, muestra los símbolos de cardinalidad y su significado:

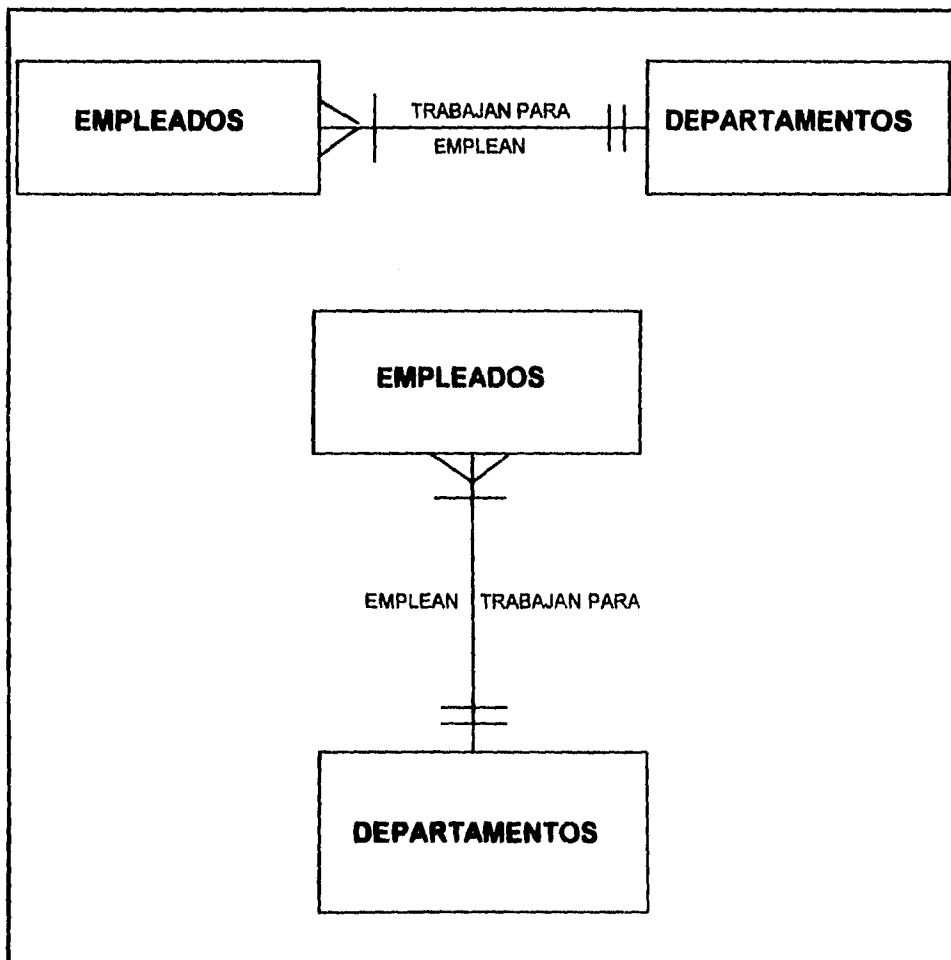
Mínimo :	0	=	Cero
		=	Uno o más
Máximo:		=	Uno
	> ó <<	=	Muchos

Si cualquiera de los símbolos de la cardinalidad es desconocido, aún no especificado, un signo de interrogación (?) aparecerá en la línea donde falte.

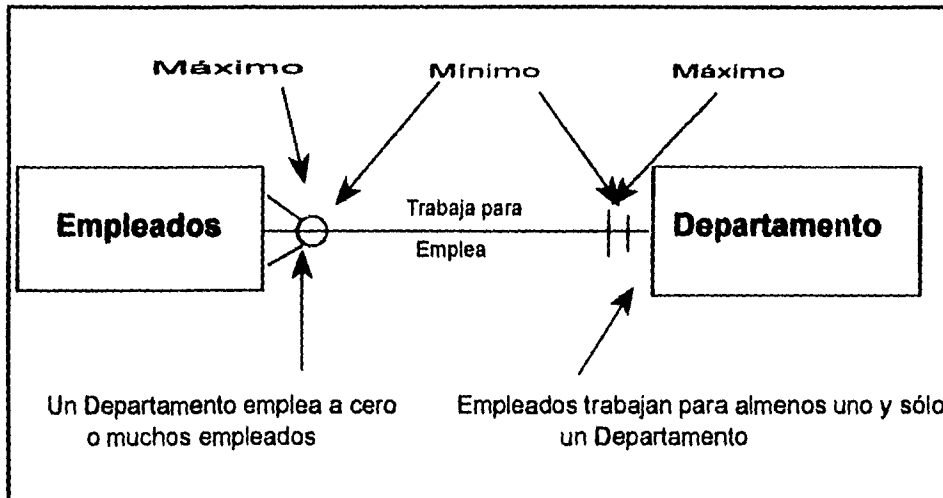
El símbolo que encierra a un ícono usado para el tipo de entidad, representa al número máximo de entidades que puede tener ésta relación. Parte el tipo de entidad al finalizar dicha asociación. Los valores máximos pueden ser uno o muchos, lo cual significa que el primer tipo de entidad puede ser asociado con al menos uno o más, de el segundo tipo de entidad referido.

El esquema siguiente muestra la relación Departamento-Epleado, de ninguno a un número ilimitado de empleados, mientras que, cada empleado puede trabajar para uno y sólo un departamento.

Cuando la acción va de izquierda a derecha, aparece arriba de la línea, y la que va de derecha a izquierda, abajo de la línea. Si la relación es vertical, la acción que va de arriba hacia abajo queda descrita del lado derecho y para el caso de ir de abajo hacia arriba la acción se describe a la izquierda.



Descriptivo de las relaciones entre entidades
Fig. 1.2.1.3.1.2.1



Muestra del uso de la cardinalidad y su interpretación
Fig. I.2.1.3.1.3.1

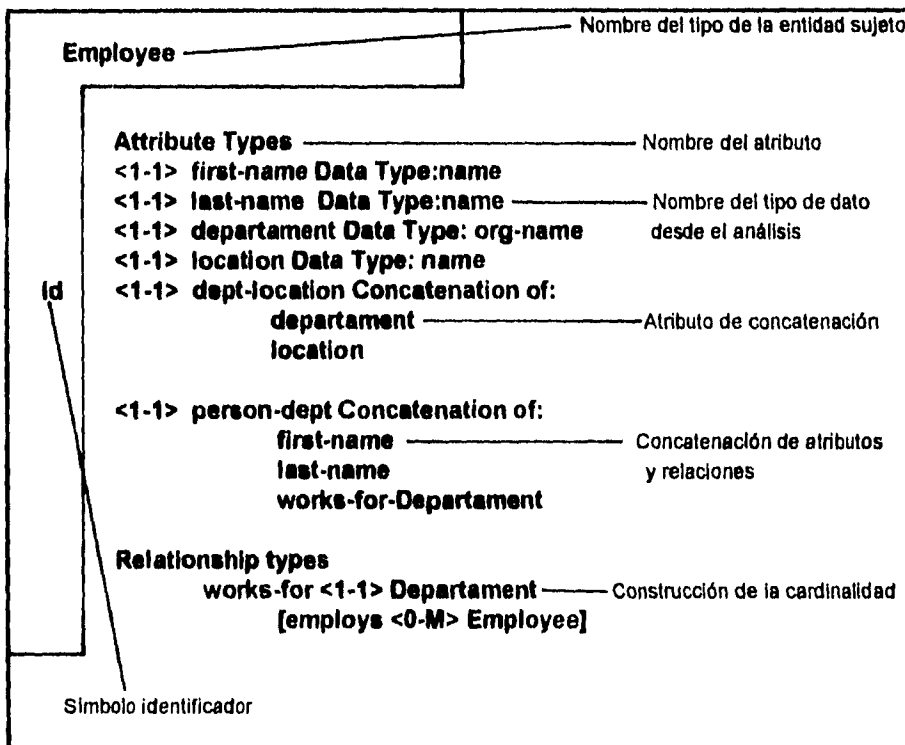
I.2.1.4 DESCRIPCIÓN DE LOS TIPOS DE ENTIDADES

En la cabeza de la ventana se muestran los atributos y tipos de relaciones dados para un tipo de entidad en el contexto de una colección de datos en particular, procesos o áreas de un proyecto (mostrados en el diagrama de desintegración), o fuera de contexto (vistos en el diagrama de modelo de entidad). Al abrir la descripción de los tipos de entidad o un diagrama de matriz de propiedad, se muestra cada atributo y tipo de relación para la entidad seleccionada.

Dentro de la ventana de tipo de entidad, podemos llevar a cabo las siguientes funciones:

- Agregar tipos de atributos
- Borrar tipos de atributos
- Concatenar tipos de atributos y relaciones
- Revisar la relación entre tipos de entidades

Los tipos de relación solamente pueden ser creados en un diagrama de entidad o en la lista de objetos.



Descripción de una ventana de tipo de entidad
Fig. 1.2.1.4.1

I.2.1.4.1. COMPONENTES DE LA DESCRIPCIÓN DE LOS TIPOS DE ENTIDADES

En la descripción de los tipos de entidades, los elementos de la descripción son caracteres tipo texto antes que íconos gráficos.

1. Tipo sujeto entidad

El nombre del tipo de entidad descrita por el atributo y tipo de relación, aparece en la equina superior izquierda del diagrama.

2. Tipo de atributo

Un tipo de atributo es una característica o propiedad, expresada por uno o más valores, que describen el tipo de entidad. Cada atributo es representado por un nombre.

3. Encogimiento de la cardinalidad

El encogimiento de la cardinalidad representa, a la liga máxima y mínima, en el número de instancias de un atributo o tipo de relación, que cada entidad puede tener.

Cada tipo de atributo tiene también un encogimiento de su cardinalidad, llamada "Max per value" (máxima por valor), la cual representa al máximo número de entidades que lo describen. Un máximo por valor de "1", para un tipo de atributo especifica que: para cada valor debe ser único entre las entidades descritas.

La cardinalidad mínima y máxima es desplegada con mayor precisión en una descripción de tipo de entidad, que en un diagrama de entidad. El mínimo puede ser cualquier entero desde 0 a 9999 o "M" (significa que no existe límite en la cardinalidad). El mínimo no debe exceder a l máximo.

- * **Encogimiento de la cardinalidad en tipos de atributos.**

El encogimiento de la cardinalidad en tipos de atributos, aparece dentro de corchetes angulares ('<' y '>'), antes del nombre del atributo. El valor anterior al guión es el mínimo para este tipo de atributo; el valor después del guión es el máximo :

<mín> - <máx>.

- * **Encogimiento de la cardinalidad en tipos de relaciones**

El encogimiento de la cardinalidad en tipos de relaciones, se despliegan igual que en los tipos de atributos, exceptuando que éstas se colocan antes del nombre del tipo de entidad relacionada. Estos encogimientos, indican el número mínimo y máximo de entidades que puede tener esta relación, para cada instancia del tipo de entidad seleccionada.

El contrario de la relación, es desplegada entre corchetes; por ejemplo, para el tópico de la entidad "Product", la relación debe leerse, "is made up of <1-M> Product Component[makes up <1-M> Product]".

4. Concatenación

Múltiples atributos o tipos de relaciones pueden ser combinadas para formar una concatenación.

5. Tipos de relaciones

Un tipo de relación, es una interacción entre tipos de entidades. Cada tipo de relación es representada por un nombre anterior y un posterior. Cuando agregamos un nombre al tipo de entidad, ésta forma un nombre de relación para el estado de la frase. Ejemplo, un empleado trabaja para <1-1> Departamento", es un tipo de relación primaria (1 a 1) y el departamento emplea de <0-M> empleados, es el tipo de relación inversa.

6. Identificadores

Un identificador es un tipo de atributo, tipo de relación o concatenación que puede ser usada para seleccionar una instancia única de un tipo de entidad sujeta. El símbolo "Id", aparece en la columna izquierda, enseguida del componente al que hace referencia. En un identificador encontramos dos características, definidas a continuación.

- * Cada entidad del tipo de entidad sujeta, tiene al menos una instancia del identificador; esto es, su mínima cardinalidad es uno o más. Con lo cual se esta garantizando que no falte el identificador de entidad.
- * Cada instancia del identificador, describe al menos una instancia del tipo de entidad sujeta; para la cual su "Max per value" es uno. Con lo que garantizamos, que cada identificador de instancia, seleccione a una entidad única.

El sistema, automáticamente inserta un "Id" a continuación de cualquier tipo de atributo, tipo de relación o concatenación que satisfaga a los dos criterios mencionados.

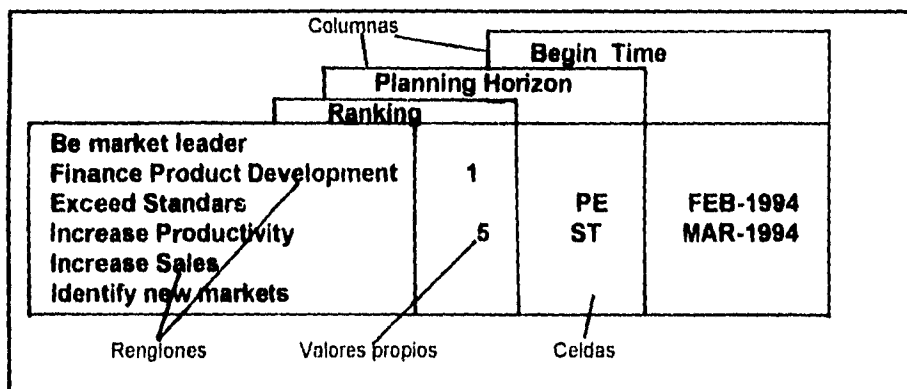
I.2.1.5 DIAGRAMAS DE MATRIZ DE PROPIEDAD

Los diagramas de matriz de propiedad, ordenan los tipos de objetos en renglones, con columnas encabezadas para cada propiedad de las instancias que componen al objeto. Si bien, el detalle del objeto lo podemos ver a través de la ventana de esta misma propiedad, encontraremos a la matriz mucho más fácil para desarrollar un borrador de una planeación de modelo, así como, una invaluable ayuda para el mantenimiento de la consistencia de la información de propiedad.

Usar un diagrama de matriz de propiedad para:

- Rápida población de la enciclopedia con objetos de un solo tipo.
- Desarrollo de propiedades para objetos iguales
- Examinar y comparar propiedades de objetos.

Cualquier información en la matriz puede ser ordenada en renglones o columnas. Esta flexibilidad, facilita la disposición de los objetos por propiedad, u objetos reordenados por instancias en orden alfabético. Podemos reordenar renglones y columnas manualmente por selección y movimiento de éstos con el "mouse" (ratón).



La matriz muestra los valores de propiedad para algunas de las Ideas
Fig. I.2.1.5.1

I.2.1.5.1. COMPONENTES DE UN DIAGRAMA DE MATRIZ DE PROPIEDAD

Una matriz de propiedad contiene renglones y columnas, formando una celda de la matriz.

1. Renglones

Be market leader
Increase productivity
Increase sales

Los renglones cruzan el diagrama de manera horizontal. Cada renglón contiene una instancia objeto. Todos los renglones contiene el mismo tipo de objeto para una propiedad dada.

2. Columnas

Ranking
Planning Horizon

Las columnas atraviesan verticalmente el diagrama. Cada columna contiene un tipo de propiedad acorde al tipo de objeto presentado en los renglones.

3. Celdas

70	TA	
		MAR-1994

Las celdas son formadas por la intersección de los renglones y las columnas. Los caracteres en la celda representa el valor de la propiedad.

I.2.1.5.2 VALORES DE LA PROPIEDAD

Los caracteres en la celda de la matriz, especifican los valores correspondientes a tipos de objetos (en el renglón), y a la propiedad (en la columna). El caracter válido que puede ser dado en la celda, depende del tipo de propiedad usada.

I.2.1.6. REPORTES

El módulo de planeación, puede analizar y reportar la información capturada en la enciclopedia; el resultado es desplegado en una ventana de reporte. Podemos posicionarnos en cualquier lugar del reporte, imprimirlo o enviarlo a un archivo de salida. Todo reporte que no sea guardado en un archivo se perderá. Si creamos y guardamos un reporte, cualquier cambio posterior que sufra la enciclopedia, no se verá reflejada en el reporte anterior. Disponibilidad de analizar y reportar

Los reportes muestran la información, acerca de los datos contenidos en la enciclopedia. No desarrolla un análisis, simplemente permite una compilación pertinente de la información para un tópico seleccionado. En la planeación podemos generar los siguientes reportes y análisis:

- Análisis de afinidad
- Reporte de Jerarquías
- Análisis de la proyección de acción
- Análisis de las vistas contenidas

I.2.1.6.1. ANÁLISIS DE AFINIDAD

El análisis de afinidad de grupos de objetos se basa en un tipo de asociación común. Determina el grado de las ligas o afinidad entre pares de objetos y usa su afinidad como una base para agrupamiento de objetos con características comunes.

Podemos usar procesos de agrupamiento para acceder tipos de entidades comunes (o funciones que involucren proyectos de áreas comunes), como una base para el desarrollo de proyectos.

La "workstation", determina la afinidad mediante el cálculo de números que pudieran tener en común, y se expresa mediante un porcentaje. Por ejemplo, un par de entidades que están involucradas en un mismo proceso, tienen una afinidad de 100. Pares de entidades no involucradas entre sí, de ninguna manera, tienen un grado de afinidad 0. El agrupamiento de objetos se basa en la afinidad calculada, pares de objetos con alta afinidad, comparados con un punto inicial especificado, forma la base de cada agrupamiento. Los restantes son asignados al agrupamiento. Los objetos restantes son asignados al agrupamiento inicial, basándose en el cálculo de peso de la afinidad.

1. Prerequisitos

Esta técnica de análisis asume :

- * La existencia de los objetos en la enciclopedia
- * Haberse descrito la asociación entre objetos

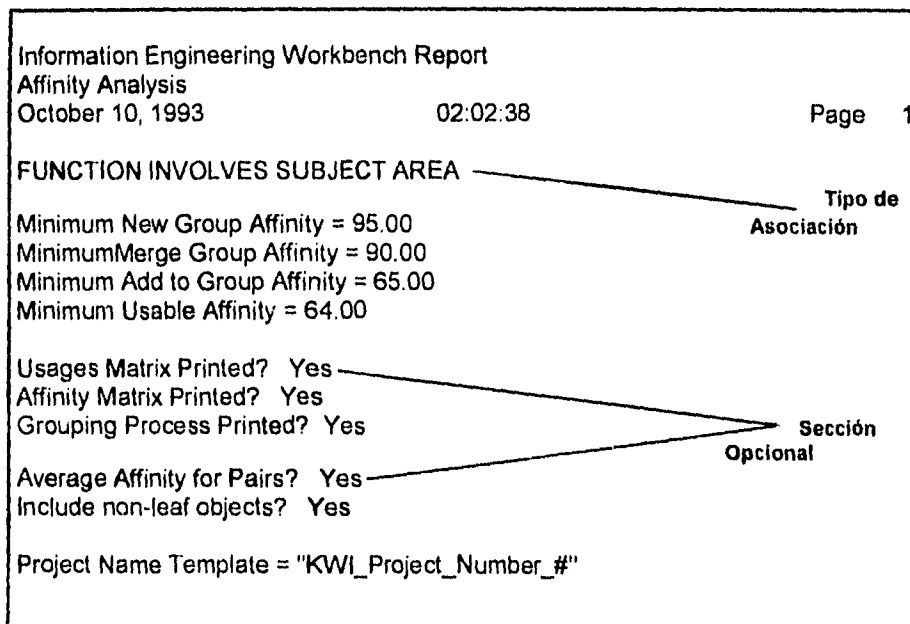
Podemos desarrollar análisis de afinidad desde el menú de la "Workstation" o desde una matriz de asociación.

2. Interpretación del reporte

El reporte de análisis de afinidad, identifica a los objetos considerados en el análisis y el agrupamiento resultante.

El reporte también puede incluir secciones intermedias, que describan al proceso y dos matrices finales donde se muestra el grupo resultante.

Cada sección del reporte se describe a continuación, usando la "Function involves Subject Area" (Función envolvente de el área sujeto) como la asociación de afinidad.



Bloques de un reporte
 Fig. 1.2.1.6.1.1

3. Sección procesamiento de opciones

En esta sección se listan los parámetros seleccionados en el panel de sección de opciones.

- * **Tipo de asociación:**
 La "workstation" , base el análisis de afinidad en su tipo de asociación, "Function involves Subject Area", en el ejemplo.
- * **Afinidad mínima para formar un nuevo grupo:**
 Para el análisis, se uso un 95% como valor de inicio.
- * **Afinidad mínima para unión de grupo:**
 Para el análisis, se uso un 90% como valor de inicio.
- * **Afinidad mínima para agregar un objeto a un grupo:**
 Para el análisis, se uso un 65% como valor de inicio.

- **Afinidad mínima para considerar objetos en todo:**
Para el análisis, se uso un 64% como valor de inicio.
- **Afinidad promedio entre pares:**
"Yes", se especificó en el análisis; la "Workstation" promedia la proporción de afinidad para pares de objetos.
- **No usar hoja objeto:**
"Yes", se especificó en el análisis; la "workstation" agrupa todos los objetos.
- **Nombre patrón del proyecto :**
Es el prefijo, para el nombre del proyecto. La "Workstation", forma nombres únicos por cada proyecto. En el ejemplo, "KWI_Project_Number_#", es el nombre patrón. Como las formas de análisis de afinidad es agrupada, el nombre del proyecto asignado, inicia con "KWI_Project_Number_1". Entonces la "Workstation", registra estos proyectos como objetos en la enciclopedia.
- **Sección de sumarización de objetos**
Aquí se lista a los objetos involucrados en el análisis de afinidad. Como ejemplo, ver figura 1.2.1.6.1.2

Instancias Objeto Izquierda:

Lista todas las instancias objeto de la columna izquierda, a través del tipo de objeto en la asociación de afinidad. Las asociaciones con estos objetos determina la afinidad entre instancias de objeto derecha (línea horizontal superior). Por ejemplo, en la asociación "Function Involves Subject Area", son funciones de instancias de objeto izquierda

Information Engineering Workbench Report		
Affinity Analysis		
October 10, 1993	02:02:38	Page 2
Left Object Instances		
1 Software Production		
2 Methodology Development		
3 Product Publishing		
4 Consulting Administration		
5 Education Administration		
6 Product Sales		
7 Product Marketing		
8 Customer Support		
9 Accounts Receivable		
10 Accounts Payable		
11 Payroll		
12 Purchasing		
Right Object Instances		
1 Product Specifications		
2 Product Publications		
3 Customer Engagements		
4 Products Course Material		
5 Orders		
6 Market		
7 Product		
8 Receivables		
9 Payables		
10 Personnel		

Bloques de un reporte
Fig. I.2.1.6.1.2

- **Instancias Objeto Derecha :**

Lista todas las instancias objeto derecha, a través del tipo de objeto en la asociación de afinidad. El análisis de afinidad agrupa a estos objetos, basándose en sus asociación común con las instancias de objeto izquierda. Por ejemplo, en la asociación "Function involves Subject Area", el área del proyecto es la "Right Object Instances".

* **Identificación de los números :**

La "Workstation usa estos números para identificar las instancias objeto que aparecen en secciones intermedias del reporte. Por ejemplo, "Right Object Instances" 1, identifica a la función nombrada "Software Production" (producción de software).

- **Sección de Tratamiento de Matrices:** la asociación entre el "Left Object Instances" y el "Right Object Instances". La matriz muestra la misma información por la asociación de afinidades. Podemos omitir esta sección del reporte.

Information Engineering Workbench Report										
Affinity Analysis										
October 10, 1993				02:02:38				Page 3		
Usages Matrix										
	1	2	3	4	5	6	7	8	9	10
1	*
2	*
3	*	*
4	.	.	*	*
5	.	.	*	*
6	*	.	*	.	.	.
7	*	*	.	.	.

Bloques de un reporte
Fig. I.2.1.6.1.3

- Afinidad

Este valor muestra como similar a dos instancias, basándose en la asociación de afinidad. Cien (100), significa que los objetos son muy parecidos, y un cero (0), no tienen algo en común con respecto a la asociación de afinidad. Ejemplo: Considerar la afinidad de las instancias de las áreas de 1 y 2 en la matriz de afinidad, mostrada en la figura anterior. Como mostramos al principio en la "Usages Matriz" (figura 1.2.1.6.2), las áreas 1 y 2 involucran la misma función una vez. Partiendo del hecho que existen tres tipos de asociaciones entre el área 1 y cualquier función, la afinidad del área 1 con la 2 es el 33% de las veces que existe una asociación entre el área 1 con las demás dentro de la misma función. Podemos ver en el renglón del área 2, dentro de la columna del área 1, 100% de las veces que existe una asociación entre el objeto del área 2 y una función cualquiera, el objeto del área 1 esta asociada con la misma función. Si la "Workstation", no promedia las afinidades entre las áreas, el "33" aparecerá en el renglón 1, columna 2, y un "100" en el renglón 2, columna 1. Sin embargo, este ejemplo muestra un promedio de afinidad "67", el cual aparece en el renglón 1, columna 2 y en el renglón 2, columna 1.

Sección de agrupamiento de procesos

En esta sección del reporte, muestra los pasos que siguió la "Workstation" para agrupar objetos similares en su afinidad.

- * **Número del ciclo :**
Identifica las etapas de los procesos. El proceso inicia en 1 y continua hasta que haya sido considerado y este por arriba del "Minimum Usable Affinity".
- * **Afinidad :**
Identifica la afinidad más alta que no haya sido considerada por el agrupamiento, y los objetos o grupos involucrados. En cada ciclo la "Workstation", considera sólo el residuo de afinidad más alto, determinado por grupo de objeto.

Information Engineering Workbench Report				
Affinity Analysis				
October 10, 1993		02:02:38		Page 6
Cycle Number	Affinity	Between Objects	Action Taken	Group Containing
1	100.00	3 and 4	Created	1
2	83.33	5 and 7	Created	2
3	66.67	1 and 2	Created	3
4	66.67	6 and 7	Added to	2

Bloques de un reporte
Fig. i.2.1.6.1.4

- * **Entre objetos :**
Lista el par de objetos que se están analizando.

* **Acción tomada :**

Describe la acción realizada por la "Workstation" con el objeto y grupo de afinidad mayor. También puede ser creación, unión, agregar o ignorancia.

* **Contenido de grupo:**

Lista los grupos contenidos en el resultado por la acción tomada.

4. Sección de grupos

En esta sección, se identifica el grupo objeto de la derecha que fue formado basándose en el valor inicial seleccionado. Si damos el nombre del proyecto, el reporte mostrará el proyecto objeto y la asociación creada entre el objeto de la derecha y él

Information Engineering Workbench Report	
Affinity Analysis	
October 10, 1993	02:02:38
	Page 7
Group 1: (Containing object n) Project KWI Project Number 1 created.	
1	3 Customer Engagements connected.
2	4 Product Course Material connected.
Group 2: (Containing object n) Project KWI Project Number 2 created.	
3	5 Orders connected.
4	6 Market connected.
5	7 Product connected
Group 3: (Containing object n) Project KWI Project Number 3 already exists.	
6	8 Orders already connected.
7	9 Market already connected.

Bloques de un reporte
Fig. 1.2.1.6.1.6

* **Grupo**

Identifica al objeto incluido en cada grupo. En el ejemplo de la figura (7:15), el proyecto área 3 y a están en el primer grupo.

* **Número de Instancia**

Muestra el número de instancias que la "Workstation" ha asignado a los objetos para el análisis..

* **Instancia objeto derecha**

Es el nombre de la instancia objeto derecha asociada con la lista de proyecto.

- **Reporte jerárquico**

Un reporte jerárquico, es usado para mostrar como las instancias objeto están relacionadas a través de sus asociaciones.

1.2.2 GENERALIDADES DE LA HERRAMIENTA IEW

El Information Engineering Workbench/Workstation (IEW/WS) ,Banco de datos de Información de INGENIERIA es un conjunto de herramientas integradas y diseñadas para trabajar conjuntamente y proporcionar ayuda automatizada para el desarrollo y el mantenimiento de software.

El IEW cuenta con un conjunto de herramientas CASE que nos ayudan a automatizar las fases de Planeación, Análisis y Diseño, en un sistema de información integrada.

Cada herramienta consiste en una serie de diagramas integradores de información, que permiten la captura y la rápida clasificación de datos, los cuales pasan a través de un chegador lógico llamado Coordinador de Conocimiento y son almacenados en una Base de Conocimiento Compartido que es denominada Enciclopedia

Hay tres conjuntos de herramientas para una Estación de Trabajo:

- 1) El Ambiente de Planeación (IEW/PWS) ayuda a capturar la información del sistema en cuestión , manejando los factores globales y críticos que intervienen dentro del planteamiento del problemas para vislumbrar las funciones, los datos y toda la información necesaria a un alto nivel de detalle.**
- 2) El ambiente de Análisis (IEW/AWS) ayuda a definir y determinar que procesos son necesarios implementar para la solución del problema definido en el módulo de planeación; así como la relación y los datos que fluyen entre ellos. Permite establecer el modelo de datos del sistema, en estudio.**
- 3) El ambiente de Diseño(IEW/DWS) ayuda a elaborar la implementación de los procesos y los datos que intervienen en el sistema a un nivel físico.**

4) El ambiente de Construcción (IEW/CWS) incluye la implementación de procedimientos y su utilización, donde ya los lenguajes de 4 generación , generadores de código y herramientas para usuario final permiten ligar la fase del diseño con la fase de construcción.

Este conjunto de herramientas permite la tarea de almacenar, mantener, y analizar el diseño de sistemas mediante diagramas. La tecnología de Inteligencia Artificial es utilizada para integrar las concepciones lógicas dentro de un almacenador denominado "Enciclopedia".

La enciclopedia almacena toda la información acerca de cada objeto sólo una vez, de esta manera las características de los objetos están actualizados en todos los lugares donde estos aparecen.

Las principales facilidades con que cuenta este conjunto de herramientas son:

- a)** La representación de conceptos a través de diagramas.
- b)** La utilización del mouse para navegar en todos los módulos.
- c)** El uso de cajas de diálogos para preguntar y para dar avisos del sistema.
- d)** Generalmente existe en las opciones un valor por default.
- e)** Las acciones del mouse pueden ser usadas por menús.
- f)** Existen combinaciones de teclas para reemplazar las selecciones del mouse.
- g)** Para conservar a integridad de los datos, envía avisos sobre el efecto de alguna operación sobre algún objeto.
- h)** Los diagramas se salvan por selección del menú para asegurar su actualización.

I.2.2.1 MODULO DE ANÁLISIS

Su propósito principal es ayudar a comprender como la información del problema en desarrollo, es intercambiada, generada y modificada a través de las funciones y los procesos requeridos por el sistema.

El Módulo de Análisis describe los datos y procesos necesarios para establecer las necesidades del sistema; así como la manera en que estos datos y procesos se interrelacionan. Estas facilidades se pueden realizar utilizando algunos de los diagramas comúnmente utilizados dentro del análisis estructurado, en los cuales se pueden crear, validar y mantener los datos lógicos y los modelos de los procesos de manera dinámica.

En forma general esta parte de la herramienta permite:

- 1)** Desarrollar a detalle, el modelo de datos de una tarea específica.
- 2)** Integra los procesos y el modelo de datos.
- 3)** Permite identificar los procesos y las funciones principales para el diseño del sistema.
- 4)** Permite identificar el conjunto de prioridades para el diseño del sistema.

El módulo de análisis da la facilidad de poder utilizar las necesidades establecidas en el módulo de Planeación, ya que toda la información es almacenada en una enciclopedia que comparte su información con todos los módulos y que a diferencia del diccionario de datos, ésta asegura la no duplicidad de los datos ya que cada dato generado y almacenado es único.

Al módulo de Análisis lo integran las siguientes herramientas:

- a) Los Diagramas de Acción.
- b) Los Diagramas de Flujo de Datos
- c) Los Diagramas de Descomposición
- d) El Diagrama Entidad/Relación

I.2.2.1.1 DIAGRAMAS DE ACCIÓN

Es una herramienta gráfica que permite describir clara y textualmente el detalle lógico de una secuencia de procesos, los cuales están compuestos a su vez , por una serie de acciones que describen un proceso secuencial y cuyos pasos o subprocesos son ejecutados a un tiempo determinado y en un orden específico.

Los Diagramas de Acción describen la actividad realizada por los procesos de nivel más bajo en el Flujo de datos jerárquico y que frecuentemente son denominados mini-especificaciones, éste consiste en realizar una serie de pasos con un propósito, con una tarea a realizar ó un conjunto de acciones que combinadas y encerradas en brackets, identifican un block. Cada block de acciones anidadas describe justamente una acción a realizar. Las acciones se ejecutan en un orden (Top-down). Las condiciones en un block determinan que acción se ejecuta y por lo tanto cuantas veces.

I.2.2.1.2 COMPONENTES DEL DIAGRAMA DE ACCIÓN

Los Diagramas de Acción utilizan el Inglés ó Pseudo-Código para describir las acciones de un proceso secuencial. Las condiciones son construidas bajo las condiciones en que serán ejecutas, y por el número de veces que serán llamadas.

Acción

Es la unidad fundamental de un Diagrama de Acción y es un texto que describe un paso elemental que será ejecutado por un proceso.

Flujo de Datos

Si un proceso abre un diagrama de acción, éste cuenta con un flujo de datos que llega y otro que sale del proceso. La referencia del flujo de datos aparece al inicio y al final del diagrama de acción.

Accesos al almacenador de Datos

Los procesos diagramados pueden crear, leer, actualizar y/o borrar entidades.

Exit

Los Exits indican que se ha terminado con la ejecución de un block de acciones, en el nivel en que es llamado.

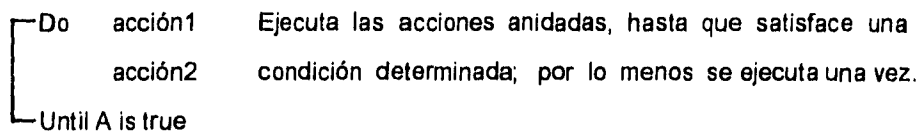
Procesos Secuenciales

Un Diagrama de acción puede hacer referencia a otros diagramas de acción que ya han sido definidos anteriormente. Cuando el ícono de un proceso secuencial aparece en un diagrama de acción, el proceso se ejecuta en ese punto. Una "S" en la parte baja de la derecha del ícono distingue al proceso secuencial.

Bloques de repetición

Permiten controlar las veces que determinadas acciones se repetirán. Existen tres clases de blocks repetitivos.

Do Until Block



Do While Block

Do while ordenes	Repite las acciones anidadas sólo mientras cierta condición se satisface; éste termina cuando la condición no logra satisfacerse.
If pagado	
registro	

For Each Block

For each orden	Cada instante en que la condición es reconocida; este block ejecuta una vez la lista de acciones.
acción 1	
acción2	

Bloques Selectivos

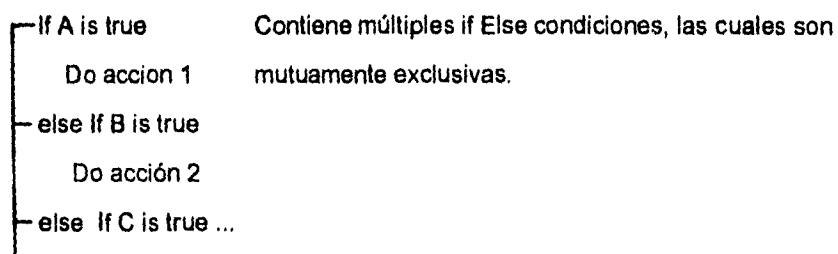
Los Bloques selectivos definen las condiciones bajo las cuales las acciones serán ejecutadas. Existen tres tipos de Blocks Selectivos:

If Block

If pago	Especifica las acciones a ejecutar si la condición se satisface.
registra pago	

If Else Block

If pago	Especifica las acciones a ejecutar si la condición dada es satisfecha, pero también muestra ejecuciones alternativas si la condición no se satisface.
registra pago	
-else	
cobra	

Multiway Block**I.2.2.1.3 DIAGRAMAS DE FLUJO DE DATOS**

Un Diagrama de Flujo de Datos despliega los procesos, los almacenadores de datos, los agentes externos, las conjunciones y el flujo de datos a un nivel de descomposición de un proceso.

El proceso descrito por un Diagrama de Flujo de Datos es un tópicos del diagrama y los procesos desplegados en el diagrama son los hijos del proceso principal.

Los procesos, los almacenadores de datos, los agentes externos y las conjunciones son denominados Nodos del Diagrama del Flujo de Datos; éstos nodos son fuente, destino y almacenadores de los datos. El flujo de datos que entra ó sale de un proceso, es conectado por medio de flechas a otros procesos que pueden encontrarse a niveles más bajos ó más altos dentro del diagrama de procesos.

I.2.2.1.4 COMPONENTES DEL DIAGRAMA DE FLUJO DE DATOS**Conjunciones de Contexto**

Representa el Flujo de datos que entra o sale de un tópicos de un proceso.

Almacenador de Datos

Son los almacenadores para los datos generados por los procesos, los cuales son reconocidos por cualquier otro proceso del diagrama.

Agentes Externos

Son interfases del sistema con la capacidad de enviar y recibir datos hacia los procesos y de los procesos del sistema.

Flujo de datos

Son Flechas que llevan datos entre los procesos, almacenadores y agentes externos.

Uniones

Ayudan a organizar y esclarecer los diagramas del flujo de datos.

Uniones Divergentes

Permiten desviar el flujo de datos de un punto hacia 2 o más salidas.

Uniones Convergentes

Permiten la combinación de dos ó más entradas dentro de un mismo flujo de datos.

Uniones Lineales

Producen una salida para una entrada.

Procesos

Los procesos manipulan datos y realizan tareas para lograr el objetivo del proceso. Los procesos reciben, transforman y transmiten datos.

I.2.2.1.5 TIPOS DE DATOS

La ventana Data type (tipo de dato) permite especificar los valores permitidos para el tipo de dato. Cada tipo describe un dato elemental dentro de la enciclopedia; existiendo de esta manera datos de tipo "global" y cuyo tipo puede ser asignado a muchos otros datos de la enciclopedia; de la misma manera, existen los datos de tipo "local" y cuyo tipo puede ser asignado a sólo un elemento de la enciclopedia.

I.2.2.1.6 DIAGRAMAS DE DESCOMPOSICION

El Diagrama de Descomposición, permite que un proceso pueda ser dividido en otros procesos más simples ó en procesos secuenciales.

De esa manera el módulo de Análisis contempla el uso de los siguientes objetos de descomposición:

- a) Unidades Organizacionales
- b) Procesos
- c) Subáreas

I.2.2.1.6.1 ICONOS DE DIAGRAMAS DE DESCOMPOSICION

Un Diagrama de Descomposición, puede representar a cualquiera de los objetos anteriormente mencionados, los cuales a su vez puede ser representados a través de iconos para construir un diagrama.

1.2.2.1.6.2 DESCOMPOSICION DE OBJETOS

Objetos

Cualquier objeto posible a descomponer se representa mediante un cuadro, excepto las funciones y los procesos que son representados por íconos rectangulares.

Funciones

Las funciones se representan con íconos rectangulares que cuentan con una "F" en la esquina inferior izquierda.

Procesos

Los procesos se representan con íconos rectangulares que cuentan con una "P" en la esquina inferior izquierda.

Procesos Secuenciales

Los procesos secuenciales se representan con íconos rectangulares que cuentan con una "S" en la esquina inferior izquierda.

1.2.2.1.6.3 CONTRACCIÓN DE OBJETOS

Cualquier descomposición de objetos, puede contractarse o empacarse. Esto puede representarse con una elipse y un fragmento de línea, los puntos bajo la línea indican que el ícono tiene procesos hijos escondidos.

Un objeto puede tener objetos padres ocultos que no pertenecen al diagrama activo. Estos son representados por un ícono y un fragmento de línea arriba lo cual indica que existe uno ó más padres del objeto que han sido escondidos.

Un objeto puede tener objetos hijos ocultos , debido a que los niveles del diagrama son limitados. Estos se representan con íconos que cuentan con una "F" en la esquina inferior izquierda. Las funciones se representan con íconos que cuentan con una "F" en la esquina inferior izquierda.

I.2.2.1.7 DIAGRAMAS ENTIDAD/RELACION

Los Diagramas entidad son utilizados en la fase de planeación "top down" y para el diseño del Modelo de Datos. En la planeación "top down" se identifican las entidades principales; que participan dentro del funcionamiento del sistema, además se construyen y determinan las relaciones más óptimas entre las mismas.

El modelo de datos puede ser utilizado en el Módulo de Análisis en donde se puede afinar el Diagrama Entidad creado en el Módulo de Planeación. Es posible utilizar el Diagrama Entidad de la fase de Planeación ó el de la fase de Análisis para desarrollar este modelo, la versión del Módulo de Análisis permite construir la base de datos relacional directamente del Modelo Entidad.

Un Diagrama Entidad, algunas veces llamado Diagrama Entidad/Relación muestra el tipo de entidades que forman un sistema y como se relacionan entre sí .Se pueden desarrollar las propiedades tanto de las Entidades y de las Relaciones a través del mismo Diagrama Relacional con la ayuda de cajas de diálogo. Las Entidades y las relaciones entre ellas, que se encuentran definidas dentro de la "enciclopedia" conforman el Modelo Entidad. Se pueden crear Diagramas Entidad para flujo de datos, almacenadores de datos, agentes externos, procesos, procesos secuenciales y subáreas. Estos Diagramas Entidad presentan las vistas de cada objeto que pertenece al Modelo Entidad.

I.2.2.1.7.1 COMPONENTES DEL DIAGRAMA ENTIDAD

Existen dos componentes básicos que son utilizados en el Diagrama Entidad, entidades y relaciones; la cardinalidad fuerza a limitar el número de relaciones entre las entidades.

Los íconos de las entidades son rectángulos que cuentan con el nombre de la entidad, la relación aparece como una línea entre las entidades y las cardinalidades se localizan al final de la línea de relación.

I.2.2.1.7.2 TIPOS DE ENTIDAD

Una entidad es una clase de gente, un lugar, un objeto ó un concepto que tiene características de interés para el problema que se esta analizando. Un tipo de entidad que existe y que es de interés para el sistema es llamada "entidad fundamental".

Una entidad que existe primordialmente para interrelacionar a las entidades entre sí; es llamada "entidad asociativa". Una entidad que sirve para describir otro tipo de entidad es llamada "entidad atributiva".

I.2.2.1.7.3 TIPOS DE RELACIONES

Una relación aparece como una línea significativa entre dos entidades y su denominación es una frase con un verbo conjugado en tiempo presente. La unión de entidades establece la naturaleza de la relación

I.2.2.1.7.4 CARDINALIDADES

La cardinalidad especifica el número de instancias que una relación puede tener ú ocurrir con respecto a otras entidades con las cuales tiene una relación.

Cada cardinalidad esta definida por un valor mínimo y un valor máximo. El valor mínimo puede ser "0" ó cualquier entero positivo; el valor máximo puede ser un entero positivo ó "M" cuyo significado es de ilimitado. El valor máximo debe ser mayor que el valor mínimo por regla.

I.2.2.1.8 ENTIDAD DESCRIPTIVA

Esta parte de la herramienta permite a través de una ventana con funciones especiales mostrar los atributos y las relaciones más relevantes de una entidad en el contexto de un flujo de datos particular, de un almacenador de datos, de un agente externo , de un proceso, de un proceso secuencial ó una subárea ó inclusive fuera del contexto.

Dentro de una entidad descriptiva se puede:

- a) Adicionar atributos
- b) Borrar atributos
- c) Concatenar atributos y relaciones
- d) Revisar relaciones entre entidades

I.2.2.1.8.1 COMPONENTES DE UNA ENTIDAD DESCRIPTIVA

En una entidad descriptiva, los elementos de la descripción son texto más que íconos gráficos.

Adjetivo de la Entidad

El nombre de la entidad descriptiva, para los atributos y las relaciones aparecen en la parte alta de la izquierda del diagrama.

Atributos

Un atributo es una característica o una propiedad expresada como uno ó más valores, que describen a la entidad. Cada atributo es representado por un nombre.

Cardinalidad

La cardinalidad representa la máxima y la mínima frontera de número de instancias, de un atributos o relaciones que cada entidad tiene.

Concatenaciones

Múltiples atributos ó relaciones pueden ser combinados para formar una concatenación.

Relaciones

Una Relación es una interacción entre entidades, cada relación es representada por un nombre en una dirección hacia adelante y otro en reversa.

Identificadores

Un identificador es un atributo, una relación o una concatenación que puede seleccionar una instancia única de la entidad.

I.2.2.1.8.2 EXPRESIONES DE FLUJO

Son utilizadas para especificar los componentes y envolventes en un flujo de datos. Un flujo de datos puede incluir entidades, atributos, relaciones u otros flujos de datos.

El Módulo de Análisis permite una gran facilidad para editar una expresión de flujo de datos a través del Diagrama de Flujo de Datos.

La expresión de flujo de datos describe la composición de un flujo de datos. Si la expresión contiene más de un componente, se pueden separar mediante operadores. Si se desea se pueden capturar la cardinalidades que describen el número de instancias de una ó más componentes en función de otra componente.

En una expresión de flujo no deben utilizarse caracteres especiales.

Los operadores que se pueden utilizar son:

- + que significa and
- | que significa or exclusivo
- . que significa no se conoce
- () determinan la precedencia de operadores.

I.2.2.1.9 REPORTES

El Módulo de Análisis permite reportar para poder analizar toda la información capturada en la enciclopedia; los resultados son desplegados en un reporte. Este reporte se puede ver mediante una ventana, la impresora o un archivo.

I.2.3 DIAGRAMAS DE ACCION POR MODULOS

Los diagramas de acción por módulos representan a programas lógicos detallados en un formato gráfico.

MÓDULO DE PARÁMETROS

La comunicación entre módulos se muestra en diagramas de acción por pares de parámetros formales y llamadas paramétricas para un modelo dado. La asignación de parámetros para un módulo especifica la ruta en la cual el módulo puede ser usado en datos, por ligado en una o más estructuras de datos. Parámetros formales definen los datos que son esperados para recibir (si es experimentado como entrada, salida o ambos) y referida a la estructura de los datos pasados a el módulo.

Todos los módulos que llaman a submódulos usan los mismos parámetros formales para la comunicación con ello. Llamadas de parámetros unen el actual dato transferido en una llamada específica a el parámetro formal esperado.

I.2.3.1 COMPONENTES DE UN DIAGRAMA DE ACCION

Los componentes de un diagrama de acción, describen la acción a tomar en el procesamiento de datos. Las acciones son empacadas por la condición bajo la cual es ejecutada, y por el número de veces a ser repetida la acción.

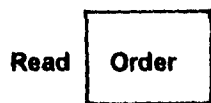
Los diagramas de acción por módulos también pueden llamar a submódulos de datos en estructuras accesadas.

Con ligeras modificaciones, diagramas de acción de "mini-spec" (mini especificaciones), creadas en el módulo de análisis de la "Workstation", puede ser incorporado dentro de un módulo de diagramas de acción.

1. ACCIONES

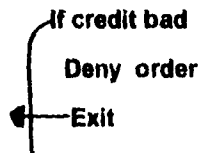
Las unidades fundamentales de un diagrama de acción, son líneas de texto que describen una operación, programa, subrutina, o línea de código.

2. ACCIONES DE DATOS



Describe el tipo de acción tomada (pudiendo ser, Create, Read, Update o Delete) y el área de datos afectados. El ícono representa el área de datos.

3. SALIDAS

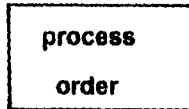


Un "Exit" termina la ejecución de un bloque de acción, para cualquier nivel. Una salida es representada por una flecha y la palabra "Exit".

4. MÓDULOS

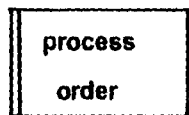
Cualquier diagrama de acción por módulos pueden mostrar a otros módulos, los cuales en su oportunidad pueden mostrar llamadas adicionales.

5. MÓDULO LLAMADOR



Las llamadas a módulos son representadas por un icono rectangular.

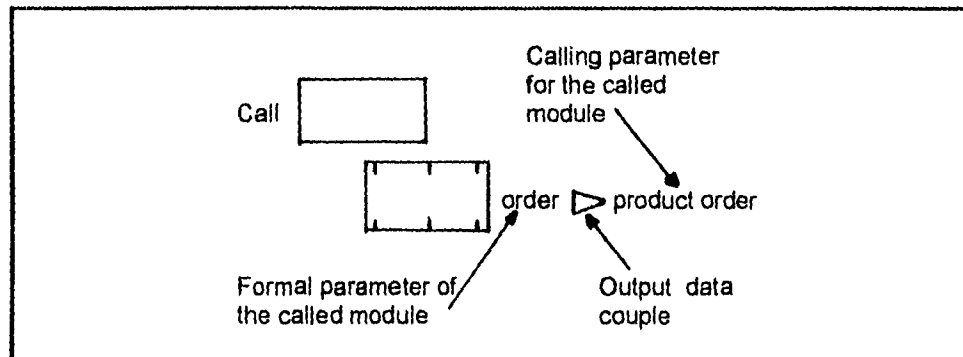
6. MÓDULO LIBRERÍA



El ícono para cualquier módulo almacenado en una librería, muestra barras verticales en sus costados.

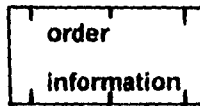
7. MÓDULO DE PARÁMETROS

Parámetros formales del módulo llamado, su estructura de datos, llamadas asociadas parámetros y símbolos de datos pareja/bandera, entrada/salida, son representados a continuación del ícono llamado.



Partes del módulo de parámetros
Fig. 1.2.3.1.1

8. ESTRUCTURA DE DATOS



Identificar la estructura de datos pasados por el módulo, puede ser usado como entrada o salida para el programa.

9. PAREJA DE DATOS

Este símbolo muestra los datos a procesar y son pasados entre los módulos llamador y llamado. La dirección de los movimientos de datos iguales es hacia donde apunta el símbolo: in, out o ambos.



Input data couple

Datos pasados dentro de este módulo desde el módulo llamado.



Output data couple

Dato saliendo de éste módulo, dentro del módulo llamado




Input/output data couple


Dato pasado entre el módulo y el módulo llamado.


10. BANDERA

Banderas son típicamente conjuntos y rangos examinados que proceden.

Frecuentemente comunican información acerca del dato. Los datos son pasados entre el módulo llamador y llamado: in, out, ambos, dentro y fuera del módulo llamado.

- 

Input flag Este módulo pasa información descriptiva al módulo llamado.
- 

Output flag El módulo llamado pasa información descriptiva a este módulo
- 

Input/Output flag Tanto el módulo llamador como el llamado, pasan información descriptiva en ejecución.

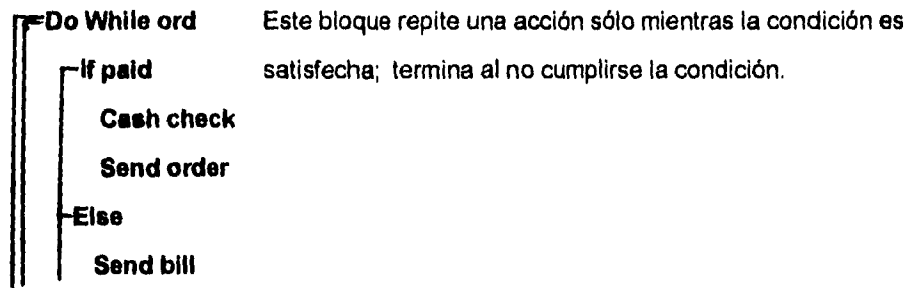
11. BLOQUES DE REPETICIÓN

Los bloques de repetición controlan la interacción de acciones. Existen tres diferentes bloques de repetición: El *Do Until*, el *Do While*, el *For Each*.

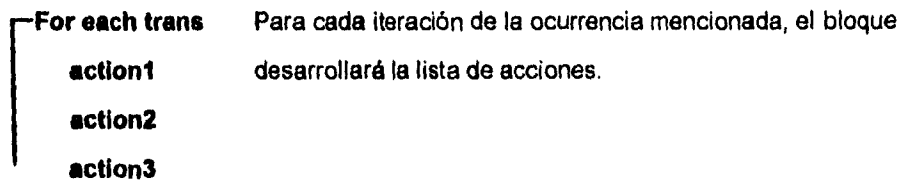
• Bloque **DO UNTIL**

<p>Do</p> <p style="padding-left: 20px;">action1</p> <p style="padding-left: 20px;">action2</p> <p style="padding-left: 20px;">action3</p> <p>Until A is true</p>	<p>El bloque <i>Do Until</i> desarrolla una acción hasta que una condición es satisfecha; al menos se ejecuta una vez.</p>
--	--

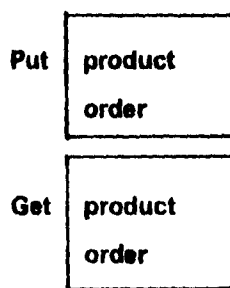
* Bloque **DO WHILE**



* Bloque **FOR EACH**



12. **VENTANAS**

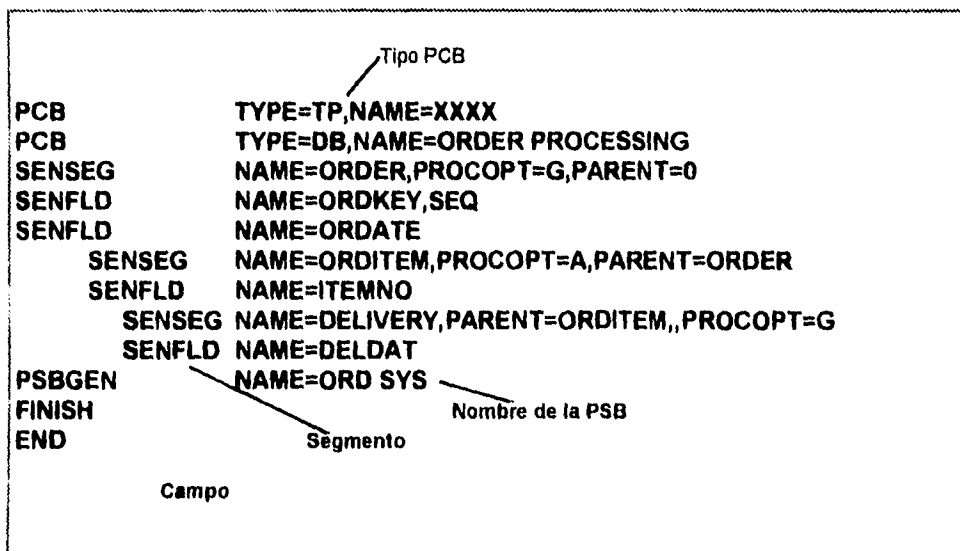


En algunas aplicaciones el programa pondrá "Put", fuera de la ventana para recibir iteraciones de usuario, la ventana de "Get", checa el cambio de datos.

1.2.3.2 BLOQUE DE COMUNICACIÓN PROGRAMADA (TEXTO PSB)

Una PSB es una serie de macro instrucciones que describen una aplicación, como una aplicación usada en segmentos y campos DBD's, y como si usáramos una terminal lógica. Una PSB, consiste de uno o más Program Communication Blocks (Bloques de Comunicación Programada, PCB). Cada DB PCB, representa una DBD que usa el programa de aplicación; Una TP PCB define a un mensaje destino.

Después de construir el conjunto de macro instrucciones PSB, pueden usarse como entrada para generar la PSB. De acuerdo a los diagramas y texto de la DBD, la PSB es usada para comunicaciones IMS.



Un texto PSB contiene dos PCBs
Fig. 1.2.3.2.1

I.2.3.2.1 COMPONENTES DE UNA PSB

Los componentes texto de una PSB son los objeto texto que pueden ser modificados y movidos.

1. PSBGEN

El predicado PSBGEN lista los nombre de la PSB.

2. DBPCB

El predicado DB PCB define a un subgrupo de segmentos de la DBD accedados por un programa en particular. La línea muestra el tipo de PCB, y nombres de la DBD desde la cual es llamada.

3. TP PCB

El predicado TP PCB define la comunicación con un monitor de teleproceso.

4. SENSEG

El predicado SENSEG lista los nombres de un segmento de búsqueda, accesos correctos a el segmento y su procedencia.

5. SENFLD

El predicado SENFLD, define un campo con una SENSEG para búsqueda, y si el programa puede actualizarla.

Relaciones Fisicas

Los niveles de identificación muestran como se relacionan los segmentos. Segmentos subordinados son identificados a la derecha de su predecesor, y el nombre de su predecesor es desplegado en el campo "PARENT =".

I.2.3.2.2. DESARROLLANDO UNA PSB

Podemos iniciar desde un PSB scratch o un PSB load.

Valores Requeridos y opcionales para una DB PCB's

Estos valores son requeridos para la generación de una PSB:

1. **DBDNAME** Valor pulsado por el usuario.
2. **PROCSEQ** Este predicado identifica a una DBD para usar como una secuencia de procesos secundario. Posicionar el cursor dentro de este campo para seleccionar PROCSEQ desde el menú de ADD.

Los siguientes valores pueden ser dados:

1. **PROCOPT** Dar cuatro letras para definir el tipo de opción para el proceso antes de desarrollarlo:

G	Obtener
I	Insertar
R	Reemplazar
D	Borrar
A	Todo, incluyendo G, I, R, D
P	usualmente requerido si D es usada
O	Sólo lectura
N	reduce la terminación anormal cuando una aplicación read-only se refiere a datos iniciados o actualizados por otra aplicación.
T	similar a N , pero causa un regreso automático de la operación

- | | |
|----|--|
| E | habilita la exclusividad de uso de una base de datos o segmentos por línea de programa |
| L | función de carga para cargado de la base de datos |
| GS | obtiene segmentos de manera ascendente |
| LS | carga segmentos de manera ascendente |
-
2. KEYLEN longitud en bytes de la llave concatenada
 3. POSITION tipo S para simple, M para múltiple, or en blanco.

Valores Requeridos y opcionales para una TP PCB's

Estos valores son requeridos para la generación de una PSB:

1. LTERM parámetro para el mensaje destino de la salida
2. NAME nombre del código de transacción para la salida destino
3. ALTRESP dar "YES", para declarar una PCB alterna
4. SAMETERM dar "YES", si se desea verificar IMS
5. MODIFY determina si la PCB alterna es modificable
6. EXPRESS especifica si los mensajes de una PCB alterna serán enviados o recibidos o si la aplicación termina anormalmente

I.2.3.3 DIAGRAMAS PARA UNA BASE DE DATOS RELACIONAL

El uso de diagramas relacionales para modelar relaciones y referencias entre ellos. La referencia muestra cual relación tiene llaves foráneas que hace referencia a llaves primarias en otra relación. La referencia también despliega la cardinalidad.

Las llaves primarias y foráneas son definidas en la estructura de datos para cada relación.

El "Analissys Workstation's Relational Translator" puede generar automáticamente un diseño físico "first cut (primer corte)" de el modelo entidad.

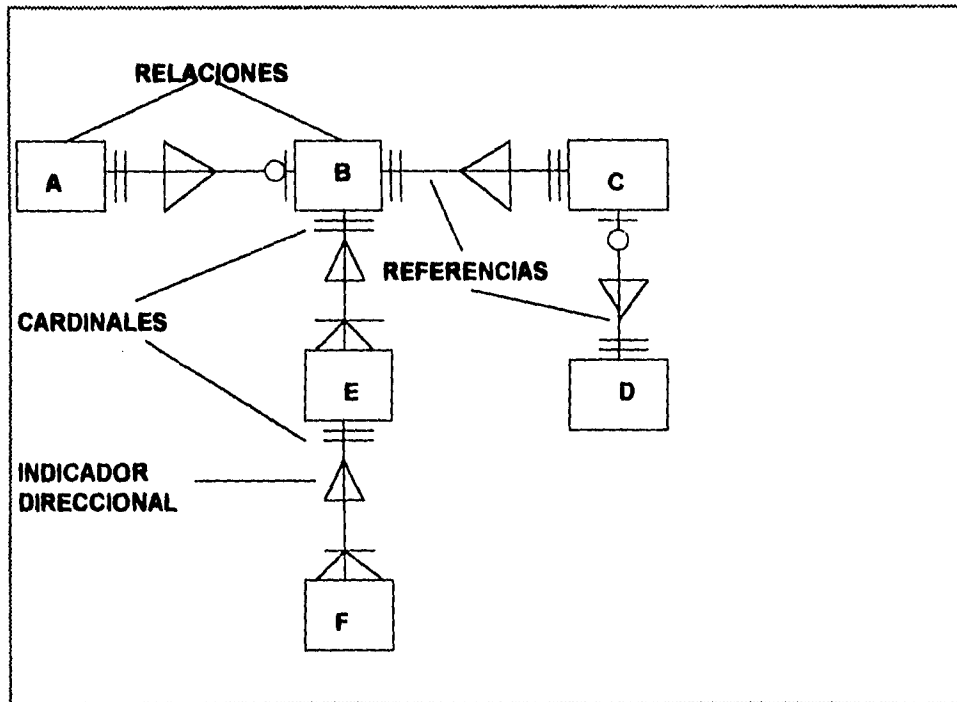


Diagrama para una base de datos relacional , donde se muestran las relaciones y las referencias entre ellos
Fig. 1.2.3.3.1

I.2.3.3.1 COMPONENTES DE UN DIAGRAMA DE BASE DE DATOS RELACIONAL

Relación

Representa a una área de datos o registro de una base de datos

Referencia

Indica una relación entre dos relaciones. La cardinalidad y el indicador direccional (llave foránea para llave primaria) también aparece en la línea.

CONSTRUCCIÓN DE LA CARDINALIDAD

Cada tipo de referencia es limitada (construida) por el número (cardinalidad) de instancias que pueden ocurrir entre las dos relaciones.

Cada cardinalidad construida es definida por un mínimo y un máximo. El mínimo puede ser cero o cualquier positivo entero; el máximo puede ser un cualquier entero positivo o "M" (muchos), lo que significa no limitado. El máximo debe ser más grande que el mínimo.

I.2.3.3.2 ESTRUCTURA DE DATOS RELACIONALES

Las relaciones son áreas de datos. Toda relación tiene una estructura de datos raíz. Una raíz de relación en una estructura de datos puede contener otra estructura de datos componentes.

Estos componentes de estructura representan campos en la relación. La estructura raíz es local por default, aunque tenemos la opción de modificarla o reemplazarla.

Los componentes de una estructura de datos (grupo de datos o elementos de datos), tiene una característica que aplica para estructuras de datos en general y algunas que son únicas para relaciones. Estas características son representadas en las columnas de los diagramas de relación para una estructura de datos. Lo siguiente es una lista de todas las características de una relación de estructura de datos, el tipo o lista de valores pueden darse y explicación del propósito.

SYNC *check mark*

Indica alineamiento de las palabras. No checa el significado de la marca los datos son alineados por byte.

PURPOSE *value , token, pointer*

Indica que la estructura representa un valor simple, un token, o un apuntador.

Primary Key *check mark*

Especifica que la estructura de sirve como el identificador principal de la relación. Si la llave primaria es compuesta de múltiples elementos Y datos , debemos de crear un grupo de datos para representar la llave primaria.

Foreign Key *check mark*

Especifica la relación existente entre un estructura actual y otra estructura. La "foreign key (llave foránea), es la "primary key (llave primaria)" en otra estructura

Identifier 1 *Check mark*

Identifica el valor de la estructura que puede ser usada para identificar una ocurrencia de la relación

Identifier 2 *check mark*

Identifica a el valor de las estructuras que pueden ser usadas para identificar una ocurrencia de la relación.

- Identifier 3** *check mark*
Indica el valor de las estructuras que pueden ser usadas para identificar una ocurrencia de la relación.
- Index** *1 a 18 caracteres*
Especifica a un índice con el nombre de su creador que corresponde a una estructura de datos.
- Mandatory** *check mark*
Indica el valor de la estructura obligatorio. Checa los campos de nulidad.
- DataBase Field** *check mark*
Indica los campo de la estructura especificada en la base de datos.
- Physical name** *1-18 characters*
Especifica el nombre a ser usado por el campo en la base de datos.
- SQL data type** *char, varchar, graphic, vargraphic, smallint, integer, decimal, real, double, precision, float, date, time, timestamp.*
Especifica el tipo de dato a ser usado por el campo en la base de datos.
- Derive SQL data type** *check mark*
Indica el tipo de dato SQL derivado automáticamente le tipo de asociación de datos. Es read-only esta celda.
- Data Type** *1 - 32 characters*
Presenta el nombre de un tipo de dato que describe el tipo, formato, y otras implementaciones características de una estructura de datos.
- Format** *una cláusula picture de COBOL*
Especifica el formato de los valores de la estructura de datos.

Nota que algunos de los valores de las características representadas en la columna puede ser dado directamente en la celda de matrix de estructura de datos, mientras otros (tales como tipo de datos y formato) son dados vía menú. El tipo de dato SQL puede ser agregado independientemente o derivado desde el tipo de datos genérico asignado a la estructura de datos. Cuando un tipo de dato SQL es elevado en prioridad para la existencia de un tipo de dato. La "Workstation" deriva el tipo de dato SQL y coloca una marca en la columna "tipo de dato SQL Derivado". Las derivaciones son hechas acorde a la siguiente tabla:

Tipo de Dato type	format	SQL data Type type
Alphanumeric	X(n)	CHAR
Date		DATE
Time		TIME
Timestamp		TIMESTAMP
Binary	S9(n), where 1<=n<=4	SMALLINT
Internal Decimal	S9(n) where 5<=n<=9	INTEGER
Internal Decimal	S9V	DECIMAL (1,0)
	SV9(n)	DECIMAL (n,n)
	S9(n)V	DECIMAL (n,0)
	S9(n)V9(m) where n,m >=1; n+m <= 15	DECIMAL (n+m,m)
Single Precision Floating Point		REAL
Double Precision Floating Poin		DOUBLE PRECISION

1.3 CONCEPTOS DE BASES DE DATOS.

En los últimos 30 años la computadora ha tomado un papel preponderante en las diversas disciplinas humanas, en especial en sistemas de información, en donde es impresionante observar como crecen en volumen e importancia los archivos de datos que utilizan las computadoras. Cuanto mayor es la cantidad de datos a que tiene acceso la computadora tanto mayor es su poder potencial. Sin embargo el uso indiscriminado de archivos y la falta de una metodología adecuada producen redundancia e inconsistencia en los datos de una organización o empresa. Los sistemas de Bases de Datos tienen entre sus objetivos evitar este tipo de fallas tan comunes.

Un sistema de manejo de bases de datos (también conocido como DBMS), consiste en un conjunto de datos relacionados entre sí y un grupo de comandos o programas para tener acceso a los datos. El conjunto de datos se conoce comúnmente como base de datos, y es en esta estructura donde se encuentra la información referente a una entidad determinada. El principal objetivo de un DBMS es crear un ambiente en el cual sea posible guardar y recuperar información en la base de datos en forma segura y confiable.

Los sistemas de bases de datos se diseñan para controlar grandes cantidades de información. El manejo de los datos incluye tanto la definición de las estructuras para el almacenamiento de la información como los mecanismos para el manejo de la misma. Entre otras cosas el sistema de bases de datos debe cuidar la seguridad de la información almacenada, lo mismo para las fallas del sistema como contra los intentos de acceso no autorizado. Si los datos van a ser compartidos por varios usuarios, el sistema debe evitar la posibilidad de tener resultados erróneos.

Por la importancia de la información en todas las empresas, la base de datos es un valioso recurso, esto permitió desarrollar un gran número de conceptos y técnicas para manejar los datos en forma eficiente.

1.3.1 Objetivos de las bases de datos.

Un sistema de Bases de Datos tiene varios fines, pero en esencia busca proveer al usuario con una mejor y más eficiente organización de la información, a través del uso de estructuras llamadas bases de datos que le permitan almacenar grandes volúmenes de información y explotarla mediante el uso de los comandos y programas del software elegido (DBMS).

Una Sistema de Bases de Datos debe permitir las cuatro operaciones básicas:

- 1) Creación de la **Base de Datos**.
- 2) Actualización de la información:
 - Alta
 - Modificación
 - Baja o eliminación
- 3) Extracción de información
 - En línea o interactivo
 - Batch o por lotes.
- 4) Reorganización de la Base de Datos

Considere que se tiene un DBMS de buena calidad, con un buen diccionario de datos y un conjunto de comandos para: recuperar información de diferentes maneras; generar diversos reportes; y dar mantenimiento a la Base de Datos. Entonces se tienen las siguientes ventajas:

- 1) Los datos pueden tener múltiples aplicaciones.
Distintos usuarios utilizan la Base de Datos para aplicaciones distintas.
- 2) Protección contra cambios.
 - Un programa puede cambiar sin afectar a los otros.
 - La estructura lógica de la Base de Datos puede crecer o disminuir sin afectar los programas existentes.
 - Los dispositivos físicos y la estructura de los datos, pueden cambiar sin afectar los programas ni la estructura lógica.
 - Distintas bases de datos pueden mezclarse sin reescribir los programas que las controlan.
- 3) Claridad
Los usuarios pueden saber y comprender fácilmente que datos tienen.
- 4) Uso transparente.
El usuario tiene acceso a la Base de Datos de manera sencilla, la complejidad del software queda oculta para él.
- 5) Uso flexible.
Se tienen diferentes rutas de acceso para buscar datos.
- 6) Posee un sistema generador de reportes
Muy poderosos que satisfaga las demandas requeridas para las diversas aplicaciones.

7) Disminuye la redundancia.

Permite optimizar el número de campos , sin que estos se repitan innecesariamente.

8) Bajo costo.

Minimización del costo del almacenamiento, del uso de los datos y de los cambios a las aplicaciones.

9) Menor proliferación de los datos.

Las necesidades de las nuevas aplicaciones serán cubiertas con los datos existentes en lugar de crear nuevos archivos.

10) Rápida respuesta a situaciones imprevistas.

Los pedidos espontáneos de información se atenderán sin necesidad de escribir un programa de aplicación, sino utilizando un lenguaje de alto nivel para averiguación o generación de reportes.

11) Facilidad para el cambio.

La Base de Datos puede crecer y cambiar sin afectar las formas establecidas de usar los datos.

12) Seguridad de acceso.

Se evitará el acceso no autorizado de datos. Los mismos datos podrán estar sujetos a diferentes restricciones de acceso para diferentes usuarios.

13) Integridad.

Los datos estarán protegidos contra: fallas, catástrofes y personas incompetentes que intenten falsificarlos.

14) Disponibilidad.

Los datos se hallarán inmediatamente disponibles para los usuarios casi todas las veces que lo necesiten.

- 7) Disminuye la redundancia.
Permite optimizar el número de campos , sin que estos se repitan innecesariamente.
- 8) Bajo costo.
Minimización del costo del almacenamiento, del uso de los datos y de los cambios a las aplicaciones.
- 9) Menor proliferación de los datos.
Las necesidades de las nuevas aplicaciones serán cubiertas con los datos existentes en lugar de crear nuevos archivos.
- 10) Rápida respuesta a situaciones imprevistas.
Los pedidos espontáneos de información se atenderán sin necesidad de escribir un programa de aplicación, sino utilizando un lenguaje de alto nivel para averiguación o generación de reportes.
- 11) Facilidad para el cambio.
La Base de Datos puede crecer y cambiar sin afectar las formas establecidas de usar los datos.
- 12) Seguridad de acceso.
Se evitará el acceso no autorizado de datos. Los mismos datos podrán estar sujetos a diferentes restricciones de acceso para diferentes usuarios.
- 13) Integridad.
Los datos estarán protegidos contra: fallas, catástrofes y personas incompetentes que intenten falsificarlos.
- 14) Disponibilidad.
Los datos se hallarán inmediatamente disponibles para los usuarios casi todas las veces que lo necesiten.

15) Independencia física de los datos.

El hardware de almacenamiento y las técnicas físicas de almacenamiento podrán ser alteradas sin obligar a la modificación de los programas de aplicación.

16) Independencia lógica de los datos.

Podrán agregarse nuevos datos, o expandirse la estructura lógica general sin que sea necesario reescribir los programas de aplicación existentes.

17) Depuración.

La Base de Datos debe ser depurable, para mejorar su desempeño sin exigir la reescritura de los programas de aplicación.

I.3.2 Abstracción de la Información.

Un sistema de manejo de bases de datos es un conjunto de archivos interrelacionados y una serie de programas que permiten a varios usuarios tener acceso a estos archivos y modificarlos. Uno de los objetivos principales de un DBMS es proporcionar a los usuarios una visión abstracta de la información. Es decir, el sistema oculta ciertos detalles relativos a la forma como los datos se almacenan y mantienen. Sin embargo, para que el sistema sea útil la información debe recuperarse en forma eficiente.

La búsqueda de la eficiencia conduce al diseño de estructuras de datos complejas para representar la información en la Base de Datos. Pero como los sistemas de bases de datos muchas veces son utilizados por personal que no cuenta con conocimientos de computación, esta complejidad debe estar escondida para los usuarios. Para ocultar la complejidad se definen varios niveles de abstracción a través de los cuales puede observarse la Base de Datos.

- Nivel físico.

Este es el nivel más bajo de abstracción, en el que se describe como se almacenan realmente los datos. Se describen en detalle las estructuras de datos complejas del nivel más bajo.

- Nivel conceptual.

Este es el siguiente nivel más alto de abstracción, en el que se describen cuáles son los datos reales que están almacenados en la Base de Datos y que relaciones existen entre estos. Este nivel contiene toda la Base de Datos en términos de unas cuantas estructuras sencillas. Aunque es posible que la implantación de las estructuras simples del nivel conceptual requiera estructuras complejas en el nivel físico no es forzoso que el usuario del Nivel conceptual se de cuenta de ello. Este nivel lo utilizan los administradores de bases de datos quienes deciden que información se guarda en ella.

- Nivel de visión.

Este es el nivel de abstracción más alto, en el cual se describe solamente una parte de la Base de Datos. Aunque en el Nivel conceptual se utilizan estructuras más simples todavía queda una forma de complejidad que resulta de gran tamaño de la Base de Datos. Muchos usuarios de la Base de Datos no tendrán que ocuparse de esta información. Solamente necesitarán una parte de ella. Para simplificar la interacción entre estos usuarios y el sistema, se define el Nivel de abstracción de visión. El sistema puede proporcionar muchas vistas diferentes de la misma Base de Datos.

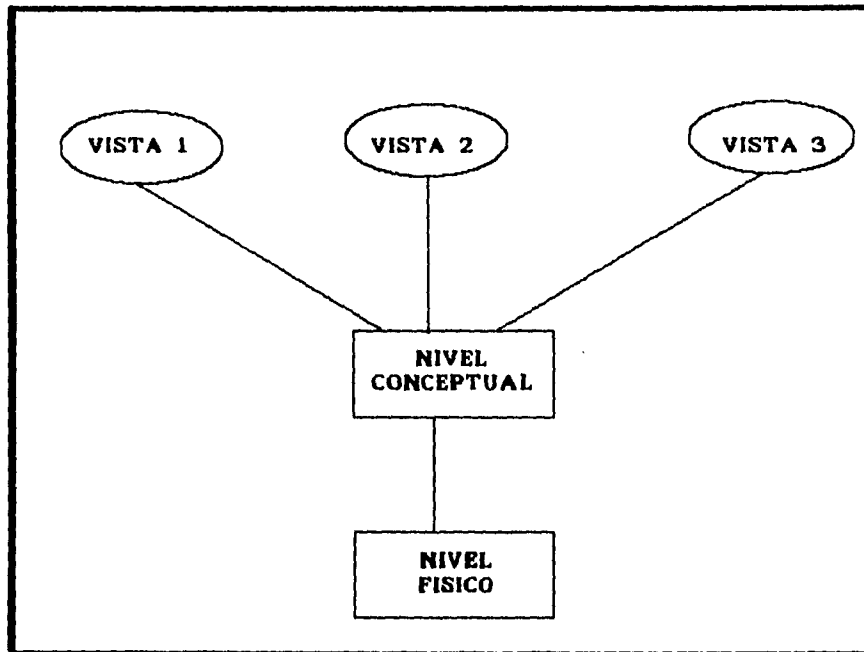


Fig. 1.3.2. Vistas de una base de datos.

1.3.3 Modelos de datos.

Para describir la estructura de una Base de Datos es necesario definir el concepto de modelo de datos. Este es un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes.

Del mundo real se tiene una percepción que corresponde a la totalidad de eventos y ocurrencias. Como un medio de ayuda para comunicarlos se utilizan los modelos del mundo real, en donde las cosas similares son generalmente agrupadas en "tipos de objetos". Un ejemplo de objeto puede ser descrito por sus características.

En nuestra percepción del mundo, las relaciones entre entidades y atributos pueden ser representadas gráficamente, donde cada nodo representa una entidad o un atributo. Pueden existir muchas relaciones, cada una con un nombre para diferenciar una relación de otra. Por medio de relaciones formamos una entidad. Existe un tipo de relación llamado asociación, por ejemplo de las entidades HOMBRE y MUJER (entidades por sí solas independientes), dos individuos casados están bajo la relación "casado con", sin embargo por el compromiso legal que contrajeron, forman una asociación. En el mismo caso se encuentra el dueño de una casa.

El objeto CASA y PERSONA (entidades por sí solas independiente) están bajo la relación "dueño de". Sin embargo, ambos objetos están asociados por los derechos de propiedad del dueño.

Los modelos de datos utilizados por los Manejadores de Base de Datos -DBMS's- pueden ser distinguidos por la forma como representan las relaciones entre los datos (atributos), pero principalmente por el manejo de las asociaciones (relaciones entre entidades). Una Base de Datos es un modelo en computadora de un sistema del mundo real. En general, si el modelo puede ser descrito en términos de estructuras naturales, el trabajo del diseñador de la base de datos se simplificará, tanto para la definición inicial como para los cambios subsecuentes.

Se han propuesto varios modelos de datos diferentes los cuales pueden dividirse en tres grupos: los modelos lógicos basados en objetos, en registros y los modelos físicos de datos.

Así por ejemplo el objeto CASA tiene características (dirección, color, estilo y precio). Es posible que un objeto sea característica de otro objeto.

Por ejemplo la dirección es un tipo de objeto cuyas características incluyen la calle y el número, el estilo de la casa puede ser rancho, duplex y bungalow, cuando una característica de manera particular describe a un objeto dentro de su conjunto se le nombra LLAVE. Así si la dirección de la casa puede ser utilizada para identificar a cada casa, entonces la dirección es la llave del objeto CASA.

Una vez identificado el mundo real, los objetos y sus características, es necesario reflejarlos en un modelo. Cuando los objetos tienen una existencia independiente y completamente definida son interpretados como entidades y sus características como atributos. Para cada entidad sus atributos tienen ciertos valores, por ejemplo el color (rojo, verde y azul) cuyo conjunto de posibles valores forman el dominio del atributo.

Es posible que varios atributos tengan el mismo dominio, como es el caso de tamaño y número de una casa cuyo dominio es el conjunto de números enteros positivos. Finalmente, es necesario elegir una representación para interpretar el mundo real bajo un modelo que esté de acuerdo a los manejadores de bases de datos. Un modelo es un patrón de acuerdo al cual los datos están lógicamente organizados. Este consiste de unidades lógicas de datos y expresan la relación entre éstos.

Uno de varios modelos puede ser utilizado para representar la información del mundo real. La principal diferencia entre los modelos de los manejadores de bases de datos (DBMS) es la manera mediante la cual representan las relaciones entre los datos.

Una RELACION es una correspondencia o mapeo entre elementos de dos conjuntos. En la tabla I.3.3 se muestra la relación entre el conjunto dirección y color.

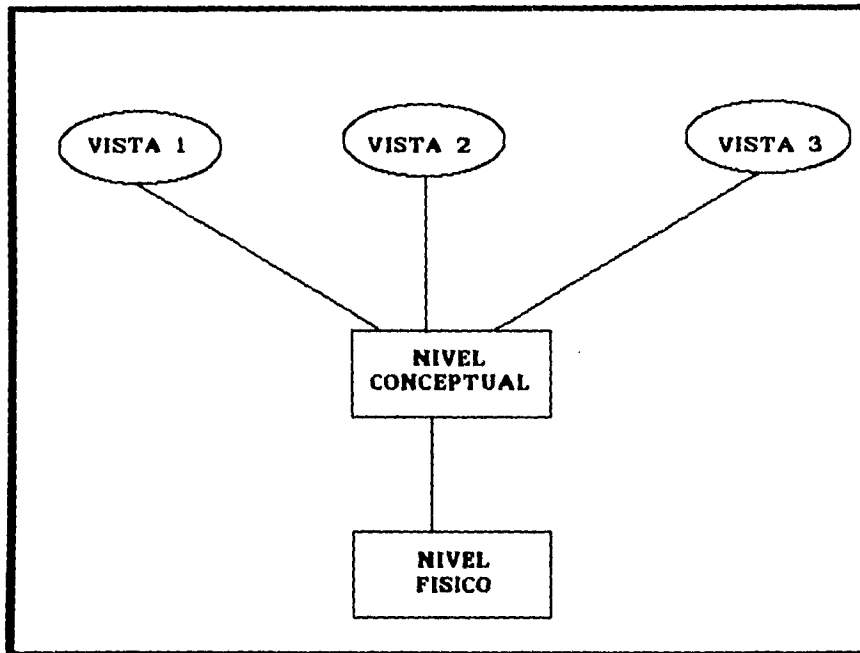


Fig. 1.3.2. Vistas de una base de datos.

1.3.3 Modelos de datos.

Para describir la estructura de una Base de Datos es necesario definir el concepto de modelo de datos. Este es un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes.

Del mundo real se tiene una percepción que corresponde a la totalidad de eventos y ocurrencias. Como un medio de ayuda para comunicarlos se utilizan los modelos del mundo real, en donde las cosas similares son generalmente agrupadas en "tipos de objetos". Un ejemplo de objeto puede ser descrito por sus características.

I.3.3.1. Modelos lógicos basados en objetos.

Los modelos basados en objetos se utilizan para describir los datos en los niveles conceptual y de visión. Se caracterizan por el hecho de que permiten una estructuración muy flexible y hacen posible especificar claramente las limitantes de los datos.

Existe una gran cantidad de modelos lógicos basados en objetos, para nuestro análisis utilizaremos el modelo Entidad-Relación (E-R), que se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y de las relaciones entre estos objetos. Una entidad es un objeto que existe y puede distinguirse de otros. La distinción se logra asociando a cada entidad un conjunto de atributos que describen al objeto.

Por ejemplo, los atributos número y saldo definen a una cuenta específica en un banco. Una relación es una asociación entre varias entidades. El conjunto de todas las entidades y relaciones del mismo tipo se denomina conjunto de entidades y conjunto de relaciones, respectivamente.

Además de entidades y relaciones, el modelo E-R representa ciertas limitantes que debe cumplir el contenido de una Base de Datos. Una de estas limitantes importantes es la cardinalidad de mapeo, que expresa el número de entidades con las que puede asociarse otra entidad por medio de un conjunto de relaciones.

La estructura lógica general de una Base de Datos puede expresarse gráficamente por medio de un diagrama E-R que consta de los siguientes componentes:

- Rectángulos.

Representan conjuntos de entidades.

- Elipses.

Representan atributos.

- Rombos.

Representan relaciones entre conjuntos de entidades.

- Líneas.

Conectan los atributos a los conjuntos de entidades y los conjuntos de entidades a las relaciones.

1.3.3.2. Modelos lógicos basados en registros.

Los modelos lógico basados en registros se utilizan para describir los datos en los niveles conceptual y de visión. A diferencia de los modelos de datos basados en objetos, estos modelos sirven para especificar tanto la estructura lógica general de la Base de Datos como una descripción en un nivel mas alto de la implantación. Sin embargo, no permiten especificar en forma clara las limitaciones de los datos. A continuación se describen los tres modelos que han tenido mas aceptación:

- Modelo relacional.

Los datos y las relaciones entre ellos se representan por medio de una serie de tablas cada una de las cuales contiene varias columnas con nombres únicos.

NOMBRE	CALLE	CIUDAD	NUMERO
Miguel	Pino	D.F.	294
Ernesto	Villa	Sonora	493
Ernesto	Villa	Sonora	320
Juan	Zarco	Puebla	563
Juan	Zarco	Puebla	493

NUMERO	SALDO
294	500
493	160
320	386
563	638

Fig. I.1.3.2.a Modelo relacional.

- Modelo de red.

Los datos en este modelo se representan por medio de conjuntos de registros (en el sentido que tiene la palabra en Pascal o PL/1) y las relaciones entre los datos se representan con ligas que pueden considerarse como apuntadores.

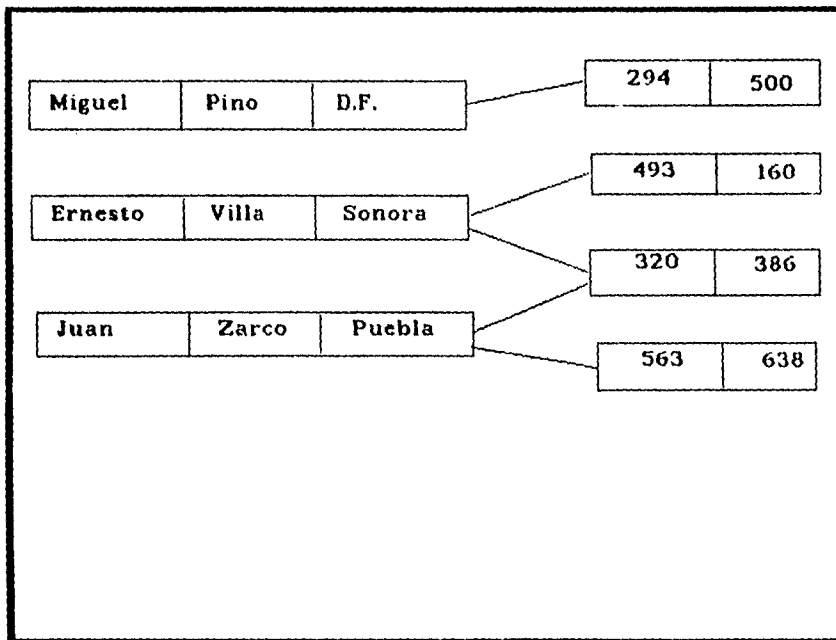


Figura 1.3.3.2.2 Modelo de red.

- Modelo jerárquico.

Es similar al modelo de red en cuanto a que los datos y las relaciones entre los datos se representan por medio de registros y ligas respectivamente. El modelo jerárquico difiere del de red en que los registros están organizados como conjuntos de árboles en vez de gráficas arbitrarias.

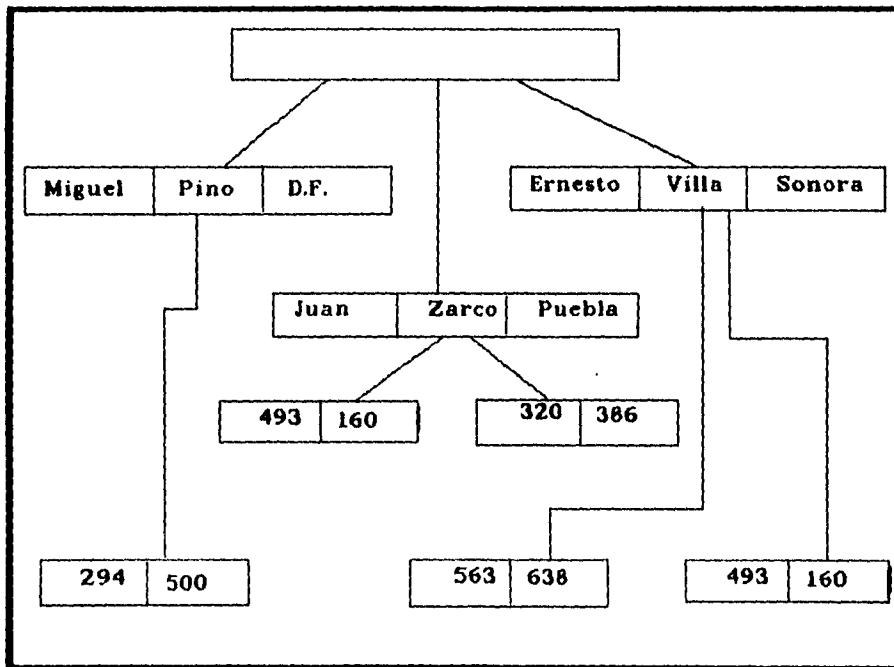


Figura I.3.3.2.3 Modelo jerárquico.

1.3.4. Instancias y esquemas.

Las bases de datos cambian con el tiempo al insertarse información en la base de datos y eliminarse de ella. El conjunto de información almacenado en la base de datos en cierto momento se denomina una *instancia* en la base de datos. El diseño general de la base de datos se llama *esquema de la base de datos*. Los esquemas se alteran muy raras veces o nunca.

El concepto de *esquema* de base de datos corresponde a la idea de definición de un lenguaje de programación. Una variable de un determinado tipo tiene un valor específico en un momento dado. Así, el concepto de valor de una variable en los lenguajes de programación corresponde al concepto de una *instancia* en una base de datos.

Existen varios esquemas en la base de datos, y éstos se dividen de acuerdo con los niveles de abstracción mencionados en la sección 1.3.2. En el nivel más bajo se tiene el *esquema físico*; en el nivel intermedio está el *esquema conceptual*, mientras que en el nivel más alto existe un *subesquema*. En general, los sistemas de bases de datos cuentan con un esquema físico, un esquema conceptual y varios subesquemas.

1.3.5. Independencia de los datos.

En la sección 1.3.2. se definieron tres niveles de abstracción en los que puede verse la base de datos. La capacidad de modificar una definición de esquema en un nivel sin afectar la definición del esquema en el nivel inmediato superior se denomina *independencia de los datos*. Existen dos niveles de independencia:

Independencia física, que es la capacidad de modificar el esquema físico sin obligar a que se vuelvan a escribir los programas de aplicación. En algunas ocasiones son necesarias modificaciones en el nivel físico para mejorar el rendimiento.

Independencia lógica, es la capacidad de modificar el esquema físico sin obligar a que se vuelvan a escribir los programas de aplicación. Las modificaciones en el nivel conceptual son necesarias siempre que se altera la estructura lógica de la base de datos.

La independencia lógica de los datos es más difícil de lograr que la independencia física, ya que los programas de aplicación dependen en alto grado de la estructura lógica de los datos a los que tienen acceso.

El concepto de independencia de los datos es similar en muchos aspectos al concepto de *tipos abstractos de datos* en los lenguajes de programación modernos. Ambos ocultan los detalles de la puesta en marcha, lo cual permite a los usuarios concentrarse en la estructura general más bien que en los detalles de la implementación en el nivel más bajo.

1.3.6. Lenguaje de definición de datos.

Un esquema de bases de datos se especifica por medio de una serie de definiciones que se expresan en un lenguaje especial llamado lenguaje de definición de datos (DDL, Data Definition Language). El resultado de la compilación de las proposiciones en el DDL es un conjunto de tablas que se almacenan en un archivo especial llamado diccionario (o directorio) de datos.

Un diccionario de datos es un archivo que contiene metadatos, es decir "datos acerca de los datos". Este archivo se consulta antes de leer o modificar los datos reales en el sistema de bases de datos.

La estructura de almacenamiento y los métodos de acceso empleados por el sistema de bases de datos se especifican por medio de un conjunto de definiciones de un tipo especial de DDL llamado lenguaje de almacenamiento y definición de los datos. El resultado de la compilación de estas definiciones es una serie de instrucciones que especifican los detalles de implantación de los sistemas de Bases de datos que normalmente no pueden ver los usuarios.

I.3.7. Lenguaje de manejo de datos.

Los niveles de abstracción no solamente se aplican a la definición o estructuración de los datos sino también al manejo de datos; esta manipulación consiste en:

- La recuperación de la información almacenada en la base de datos.
- La inserción de información nueva en la base de datos.
- La eliminación de información de la base de datos.

En el nivel físico, deben definirse algoritmos que permitan tener acceso a los datos en forma eficiente. En los niveles de abstracción mas altos lo importante es la facilidad de uso. El objetivo es lograr una interacción eficiente entre las personas y el sistema.

Un lenguaje de manejo de datos (DML, Data Manipulation Language) permite a los usuarios manejar o tener acceso a los datos que estén organizados por medio del modelo apropiado.

I.3.8. Manejador de base de datos.

El objetivo de un sistema de bases de datos es simplificar y facilitar el acceso a los datos. No debe abrumarse innecesariamente a los usuarios con los detalles físicos de la implementación del sistema. Sin embargo, uno de los factores primordiales para la satisfacción o insatisfacción del usuario con el sistema de base de datos es su funcionamiento. Si el tiempo de respuesta para una consulta es demasiado largo, el valor del sistema se reduce. El funcionamiento del sistema depende de la eficiencia de las estructuras de datos utilizados para representar los datos en la base de datos y de que tan eficientemente pueda operar el sistema con esas estructuras. Como sucede en muchos otros aspectos de los sistemas de cómputo, deben hacerse concesiones, no sólo entre el espacio y el tiempo, sino también entre la eficiencia de un tipo de operación y la de otro.

Un manejador de base de datos es un módulo de programa que constituye la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas, los programas de aplicaciones y las consultas hechas al sistema. El manejador de base de datos es responsable de las siguientes tareas:

Interacción con el manejador de archivos. Los datos sin procesar se almacenan en el disco mediante el sistema de archivos proporcionado normalmente por un sistema operativo convencional. El manejador de la base de datos traduce las diferentes proposiciones en DML a comandos del sistema de archivos de bajo nivel. Así, el manejador de base de datos se encarga realmente del almacenamiento, recuperación y actualización de los datos en la base.

Seguridad en la lectura de información. Los valores de los datos almacenados en la base deben satisfacer ciertos tipos de *limitaciones de consistencia*. Por ejemplo, si se desea almacenar en una variable valores monetarios, dicha variable no deberá tomar valores negativos. El administrador de la base de datos debe especificar estas limitantes en forma explícita. Si se especifican estas limitantes, entonces el manejador de la base de datos puede verificar si las actualizaciones realizadas a la base resultan en la violación de cualquiera de estas limitantes.

Seguridad en el acceso a la información. No es preciso que todos los usuarios de la base de datos tengan acceso a todo su contenido. Es labor del manejador de la base de datos hacer que se cumplan estos requisitos de seguridad.

Respaldo y recuperación. Un sistema de cómputo como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallas. Existen muy diversas causas de estas fallas, entre ellas la caída de las cabezas lectoras de disco, la interrupción del suministro de energía y los errores de software. En cada uno de estos casos se pierde la información de la base de datos. Es responsabilidad del manejador de la base de datos detectar estas fallas y restaurar la base al estado que existía antes de presentarse la falla. Esto se logra generalmente iniciando diversos procedimientos de respaldo y recuperación.

Control de accesos múltiples. Cuando varios usuarios actualizan la base de datos en forma concurrente, es posible que no se conserve la consistencia de los datos. Es necesario que el manejador de la base de datos controle la interacción entre los usuarios concurrentes.

Algunos sistemas de bases de datos, diseñados para utilizarse en computadoras personales pequeñas, no cuentan con varias de las funciones mencionadas. Esto da como resultado un manejador de datos de menor tamaño que requiere menos recursos físicos, sobre todo memoria principal, y su implantación es más económica. Por ejemplo, muchos sistemas pequeños incluyen la restricción de que solo un usuario puede tener acceso a la base de datos en un momento dado. Otros dejan las tareas de respaldo, recuperación e implementación de la seguridad al usuario. Aunque este enfoque de bajo costo y funciones limitadas es suficiente para bases de datos personales, no es adecuada para cumplir con los requerimientos de una empresa de tamaño mediano o grande.

I.3.9. Estructura general del sistema

Un sistema de bases de datos se divide en módulos que se encargan de cada una de las tareas del sistema general. Algunas de las funciones del sistema de base de datos pueden ser realizadas por el sistema operativo. En la mayoría de los casos, el sistema operativo proporciona únicamente los servicios más elementales y la base de datos debe partir de ese fundamento. Así, el diseño de la base de datos debe incluir una consideración de la interfaz entre el sistema de base de datos y el sistema operativo.

Un sistema de base de datos consiste en varios componentes funcionales, entre los que se encuentran:

El manejador de archivos, encargado de asignar espacio en disco y de las estructuras de datos que se van a emplear para representar la información almacenada en el disco.

El manejador de base de datos, que constituye la interfaz entre los datos de bajo nivel almacenados en la base de datos , los programas de aplicaciones y las consultas que se hacen al sistema.

El procesador de consultas, que traduce las proposiciones en lenguaje de consulta a instrucciones de bajo nivel que puede entender el manejador de la base de datos.

El precompilador de DML, que convierte las proposiciones en DML incrustadas en un programa de aplicaciones en llamadas normales a procedimientos en el lenguaje huésped. El precompilador debe interactuar con el procesador de consultas para generar el código apropiado.

El compilador de DDL, que convierte las proposiciones en DDL en un conjunto de tablas que contienen mandatos. Tales tablas se almacenan después en el diccionario de datos.

Además se requieren varias estructuras de datos como parte de la implementación del sistema físico, incluyendo:

Archivos de datos, que se guardan en la base de datos.

Diccionario de datos, que almacena la información relativa a la estructura de la base de datos. Se usa constantemente, por lo que debe tenerse mucho cuidado de desarrollar un diseño apropiado y una implementación eficiente.

Indíces, que permiten el acceso rápido a elementos de información que contienen valores determinados.

La figura I.3.9.1 muestra los componentes y las conexiones entre ellos.

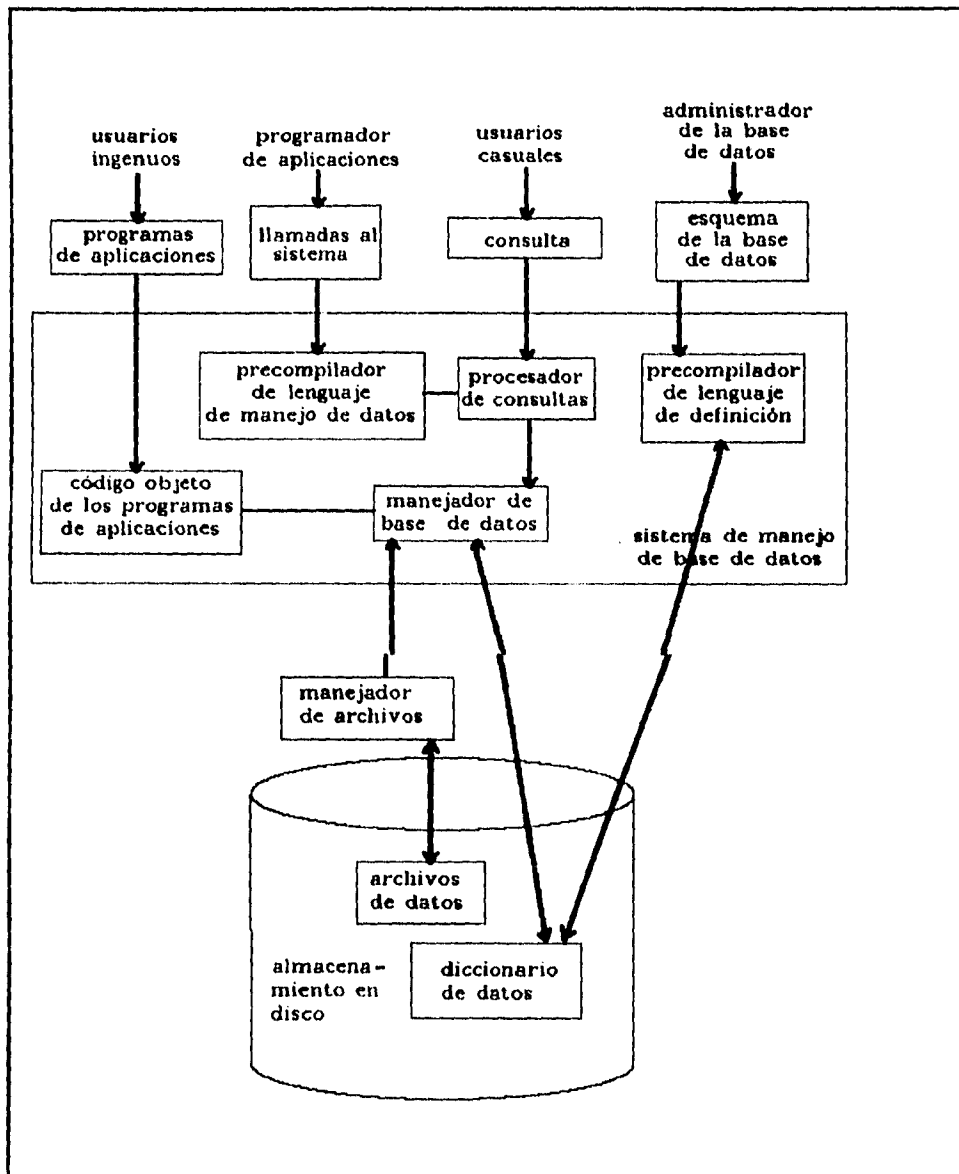


Figura I.3.9.1 Estructura General del Sistema

1.3.10. Bases de Datos Relacionales.

El objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Una de las técnicas para lograrlo consiste en diseñar esquemas que tengan una forma normal adecuada. Para determinar si un esquema de relaciones tiene una de las formas normales se requiere mayor información sobre el problema del "mundo real" que se intenta modelar mediante la base de datos. La información adicional es proporcionada por una serie de limitantes que se denominan dependencias de los datos.

Llaves primarias

Una tarea muy importante dentro de la modelación de bases de datos consiste en especificar cómo se van a distinguir las entidades y las relaciones. Conceptualmente, las entidades y las relaciones son distintas entre sí, pero desde el punto de vista de una base de datos la diferencia entre ellas debe expresarse en términos de sus atributos. Para hacer estas distinciones, se asigna una *superllave* a cada conjunto de entidades. La superllave es un conjunto de uno o más atributos que, juntos, permiten identificar en forma única a una entidad dentro de un conjunto de entidades.

El concepto de superllave no es suficiente para el modelado de la base de datos, ya que puede incluir atributos ajenos. Si K es una superllave, entonces también lo será cualquier superconjunto de K . Muchas veces lo que se busca es la superllave más pequeña posible. Es decir, se buscan superllaves para las cuales ningún subconjunto propio es una superllave. Estas superllaves mínimas se denominan *llaves candidato*.

Se utilizará el término llave primaria para denominar a la llave que elija el diseñador de la base de datos como la forma principal de identificar a las entidades dentro de un conjunto de éstas.

Dependencias funcionales

Las dependencias funcionales son las limitantes entre los atributos del sistema de base de datos que no pueden expresarse por medio de superllaves.

Para cualquier instancia de la relación, los atributos dependientes pueden tener uno y solo un valor para el atributo con el atributo del cual es dependiente. Una tabla entonces, no podrá consistir de todas las posibles combinaciones de valores del atributo para cada dominio del atributo debido a que la dependencia funcional limita los valores de los atributos dependientes. Una dependencia es llamada *dependencia funcional completa* si ningún atributo puede ser removido del conjunto de atributos sin remover la dependencia funcional.

Normalización

El proceso de determinar la correcta ubicación y función para cada atributo en el esquema relacional es llamado *normalización*.

Generalmente la normalización descompone relaciones que están incorrectamente construidas, en relaciones múltiples que estén correctamente normalizadas.

El modelo relacional requiere que todas las tablas se encuentren de inicio en la *primera forma normal (1FN)*, que quiere decir que los dominios para los atributos deben consistir exclusivamente de valores atómicos que permitan evitar la repetición de grupos. Esta es llamada *normalización estructural o sintáctica*. Los otros tipos de normalización son opcionales y semánticos por naturaleza.

Su principal tarea es evitar anomalías cuando los tuplos son actualizados, entendiéndose como anomalías, efectos inesperados en el sistema después de una operación en la base de datos.

Primera forma normal (1FN)

Para encontrar la primera forma normal, una relación debe contener dominios integrados exclusivamente por valores atómicos para cada atributo. No deben existir conjuntos de atributos, ni atributos de valores múltiples.

Considere el siguiente esquema que muestra la relación ESTUDIANTE (tabla I.3.10.1)

ESTUDIANTE (NO.ESTUD, GENERACION, PROFESOR, SALON, {CLASES})

Donde el atributo CLASES esta constituido por un campo de valores múltiples, ya que, contiene los números de clase para cada estudiante; por lo tanto no se encuentra en la primera forma normal.

NO.ESTUD	GENERACION	PROFESOR	SALON	CLASE 1	CLASE 2	CLASE 3
1022	81	Hernández	412	101-07	143-01	159-02
4123	84	Del Valle	216	201-01	211-02	214-01

Tabla I.3.10.1.

Observe que en en la tabla anterior se repite el mismo tipo de información (CLASE1, CLASE2, CLASE3) para almacenar elementos de datos que contienen muchas ocurrencias para un estudiante. Observe que si el estudiante tiene más de tres clases o menos de tres, se tendría que modificar la estructura de la base de datos (en el primer caso) o se tendrían que definir valores nulos (en el segundo caso).

Definiendo la 1FN para la tabla I.3.10.1. obtenemos la tabla siguiente:

ESTUDIANTE (NO.ESTUD, GENERACION, PROFESOR, SALON, CLASE)

NO.ESTUD	GENERACION	PROFESOR	SALON	CLASE I
1022	81	Hernández	412	101-07
1022	81	Hernández	412	143-01
1022	81	Hernández	412	159-02
4123	84	Del Valle	216	201-01
4123	84	Del Valle	216	211-02
4123	84	Del Valle	216	214-01

Tabla 1.3.10.2.

Observe en la tabla I.3.10.2. que se ha eliminado el atributo de valores múltiples CLASES, sin embargo esta simplificación implicó que se agregaran más registros a la tabla, además si el PROFESOR cambia todos sus tuplos asociado deberán ser localizados y actualizados. Estos problemas pueden ser solucionados aplicando las formas complementarias de normalización.

Segunda forma normal (2FN)

Para obtener la *segunda forma normal* la relación primero debe encontrarse en la primera forma normal y tener todos los atributos que no sean llave en dependencia funcional completa de la llave primaria. Esto significa que se deberán identificar todas las dependencias funcionales y cada atributo no llave deberá ser dependiente funcional en todos los atributos que conforman la llave primaria y no un subconjunto de la llave primaria. Todos los atributos que no cumplan con esta restricción serán removidos de esta tabla y depositados en otra relación.

Considere el ejemplo anterior donde se aplicó el procedimiento de 1FN y donde N.ESTUD es la llave primaria. Note que existen múltiples valores del número de CLASE para cada N.ESTUD. Esto significa que el número de clase no es dependiente funcional de la llave primaria y por lo tanto esta relación no se encuentra en la segunda forma normal. Es necesario descomponer el esquema ESTUDIANTE en dos tablas con el siguiente esquema (Tabla I.3.10.3. y tabla I.3.10.4.).

ESTUDIANTE (NO.ESTUD, GENERACION, PROFESOR, SALON)
 N.CLASE (N.ESTUD, CLASE)

NO.ESTUD	GENERACION	PROFESOR	SALON
1022	81	Hernández	412
4123	84	Del Valle	216

Tabla I.3.10.3.

NO.ESTUD	CLASE I
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Tabla I.3.10.4

Tercera forma normal (3FN)

Para lograr la *tercera forma normal* se requiere que la relación se encuentre en la segunda forma normal y con los atributos que no sean llaves dependientes exclusivamente de la llave primaria.

En el ejemplo anterior, note que que SALON es dependiente funcional del atributo PROFESOR. Como el atributo PROFESOR no es principal, esta es una violación de la tercera forma normal. La solución es mover el atributo fuera de la relación ESTUDIANTE observe la Tabla I.3.10.5. y Tabla I.3.10.6.

NO.ESTUD	GENERACION	PROFESOR
1022	81	Hernández
4123	84	Del Valle

Tabla I.3.10.5.

NO.ESTUD	SALON	DEPTO
1022	412	42
4123	216	42

Tabla I.3.10.6.

Cuarta forma normal (4FN)

La cuarta forma normal está relacionada con las dependencias de valores múltiples. Para llegar a la *cuarta forma normal* la tabla debe encontrarse en la tercera forma normal y el conjunto determinante de atributos en una dependencia de valores múltiples debe ser una superllave.

I.4. TOPOLOGÍA DE BASES DE DATOS

Los sistemas de bases de datos se crean con el fin de manejar un enorme volumen de información, para lo cual es necesario una definición estructural adecuada y un conjunto de instrucciones necesarias para manejar información.

Las bases de datos son los elementos de los manejadores de bases de datos. Una base de datos es un modelo en computadora de un sistema del mundo real. El mecanismo formal que se emplea para expresar la estructura lógica de datos así, como la semántica asociada es llamado modelo lógico de datos, que tienen una gran relación con las Topologías de bases de datos ha elegir. Lo anterior se puede entender fácilmente si recalcamos que las bases de datos son una colección de información del mundo real.

En esta sección se explicarán las 3 estructuras de datos:

- a) Red
- b) Relacional
- c) Jerárquico

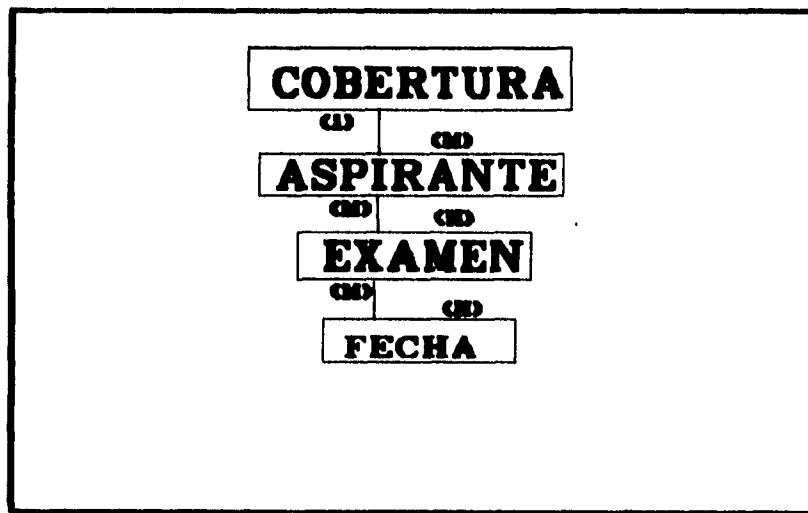
Estas estructuras de datos es posible implementarlas en una topología de las 3 existentes:

- a) Red
- b) Relacional
- c) Jerárquica

Cabe aclarar que una estructura de datos tipo red es posible implementarla en una topología relacional ó una de tipo relacional en una topología jerárquica, todo es cuestión de llevar acabo las adecuaciones pertinentes.

Los ejemplos para explicar esta sección serán relacionados con una Bolsa de Trabajo que se encarga de recopilar los datos de los interesados, así como de aplicar los exámenes pertinentes para seleccionar a las personas aptas, según lo requiera el puesto.

Posteriormente se dará una explicación más exacta de la estructura de datos que maneja, por el momento mostraremos en la siguiente figura las entidades y el tipo de relación que entre ellas guardan.



Las Entidades y sus relaciones

Figura I.4.1

I.4.1 TOPOLOGIA RELACIONAL.

Permite implementar una base de datos cuya estructura de datos sea del tipo

1:1, 1:N o N:M.

Cada cantidad queda representada por una tabla. El concepto de tabla está en correspondencia directa con el concepto matemático de relación.

Una Base de Datos Relacional consistirá de una colección de tablas, cada una de las cuales llevará el nombre de la entidad que se quiera representar, este nombre es único.

Un renglón en una tabla representa una relación entre un conjunto de valores. Dado que una tabla es una colección de tales relaciones, encierra el concepto de tabla y el modelo matemático de relación.

En la sección anterior de este capítulo se explicó ampliamente el concepto de RELACIÓN, aquí lo mencionaremos en términos menos formales: una relación será una tabla bidimensional de n Columnas, constituida de renglones ó tuplos constituidos de m objetos. Cada una de las columnas en una relación es un conjunto de datos (también llamado atributo, campo u objeto) llamado DOMINIO. Consideremos la siguiente tabla:

FECHA_EXAMEN	PERSONAS	HORA_EXAM
14 ENERO 93	10	9
14 ENERO 93	15	12
15 ENERO 93	24	9
15 ENERO 93	12	12
15 ENERO 93	10	15

TABLA I.4.1.1. PROGRAMACIÓN DE EXÁMENES (PROG_EXAM).

A esta tabla o entidad le asignamos el nombre único de prog_ex. Dicha tabla consta de 3 atributos: fecha_examen, personas y hora_exam.

Para cada atributo, existe un conjunto de valores permitidos, llamado dominio de un atributo.

Atributo	Dominio
fecha_examen	Todos los días del año excepto días festivos. A este conjunto lo llamaremos D1.
Personas	El intervalo numérico mayor que 5 pero menor que 25. A este conjunto lo llamaremos D2.
Hora_exam	Intervalo numérico mayor e igual a 8 pero menor o igual a 15. Conjunto D3.

Los dominios de los 3 atributos anteriores son D1, D2 y D3 respectivamente. Cada renglón de la tabla consiste de 3 atributos o tuplos (t1,t2,t3), donde t1 es la fecha del examen (t1 elemento del conjunto D1), t2 es el número de personas en dicho examen (t2 es elemento del conjunto D2) y t3 es la hora del examen (t3 elemento del conjunto D3). La relación es de grado 3.

NUM_EXAM	FECHA	NOM_EXAM	CALIF
930120	15/01/93	Aritmética	8.0
930120	15/01/93	Ortografía	8.0

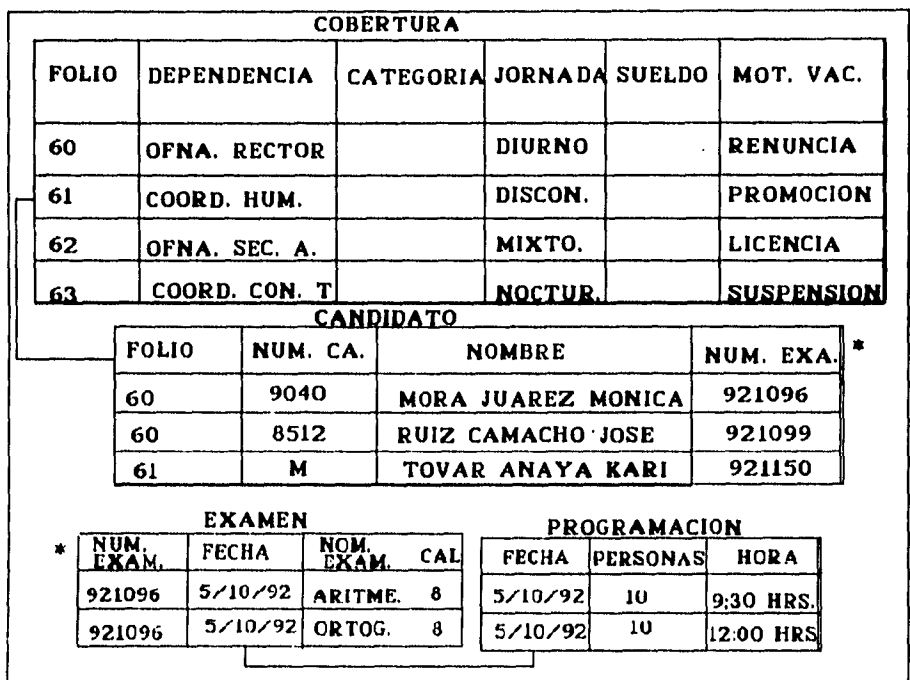
TABLA I.4.1.2. EXAMEN.

Esta tabla tiene el nombre único de EXAMEN y cuenta con 4 atributos: NUM_EXAM, FECHA, NOM_EXAM y CALIF.

Atributo	Dominio
num_exam	Las primeras dos cifras representan el año, las siguientes el mes y las últimas es un número consecutivo. Conjunto E1.
fecha	Todos los días del año excepto días festivos sábados y domingos. Conjunto E2.
nom_exam	Las materias autorizadas. Conjunto E3.
calif	Valor numérico de 0 a 10 y aproximado con decimales. Conjunto E4.

Cada renglón tiene 4 tuplos (u_1, u_2, u_3, u_4) , u_1 es elemento de E1, u_2 es elemento de E2, etc. La relación es de grado 4.

Ahora bien cada tabla o entidad, tiene un número x de renglones y cada uno se diferencian entre sí por el valor de los atributos. Cuando existe una llave común a las tablas, éstas se pueden relacionar.



Estructura de tipo Relacional Figura I.4.1.1.

La llave común es el atributo Fecha entre la entidad examen y fecha.

Puede existir más de un conjunto de dominios que pueden constituir una llave; es decir que identifican unívocamente a un tuplo y que es no redundante.

A estos conjuntos se les denomina llaves candidatas. La llave primaria es el conjunto de dominios que se selecciona para identificar a los tuplos.

I.4.1.1. VENTAJAS Y DESVENTAJAS DE LA TOPOLOGÍA RELACIONAL.**VENTAJAS****DESVENTAJAS**

No hay grupos repetitivos, cada dominio representará una sola relación

Cada columna representa el dominio del atributo. Todos los renglones son distintos.

Es posible arreglar los renglones sin alterar la información.

Cada tuplo o relación debe poseer una llave que la identifica.

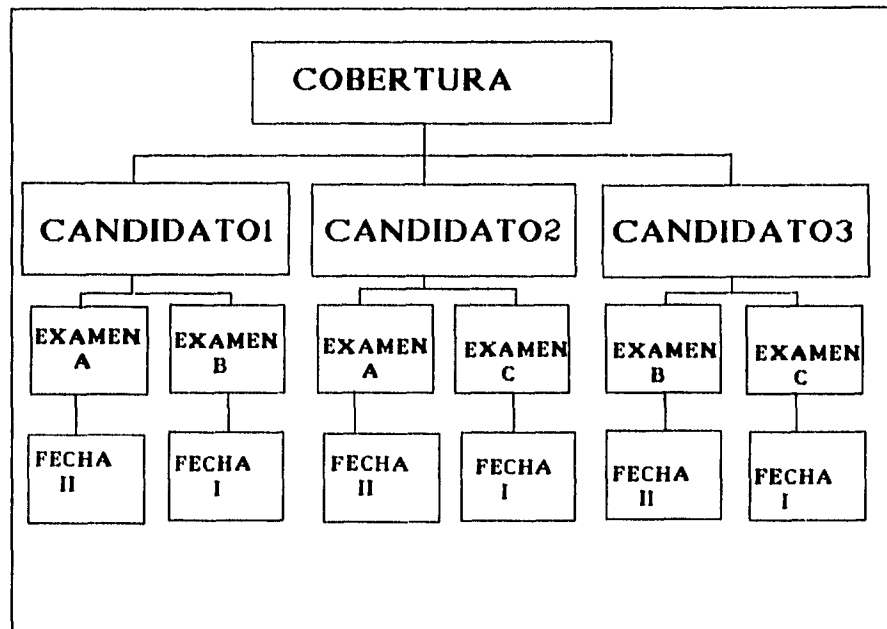
Un DBMS relacional no ha alcanzado el nivel de desempeño de un modelo de datos jerárquico o de red.

I.4.2. MODELO JERARQUICO.

Una base de datos jerárquica consiste de un conjunto de registros que se conectan entre sí por medio de ligas. Un registro es un conjunto de atributos (campos).

Una liga es una conexión entre dos registros únicamente. La topología jerárquica maneja las relaciones 1:1 y 1:N.

REGISTRO	CAMPOS
cobertura	Folio dependen categoría jornada sueldo mot_vac
candidato	folio nombre num_exam

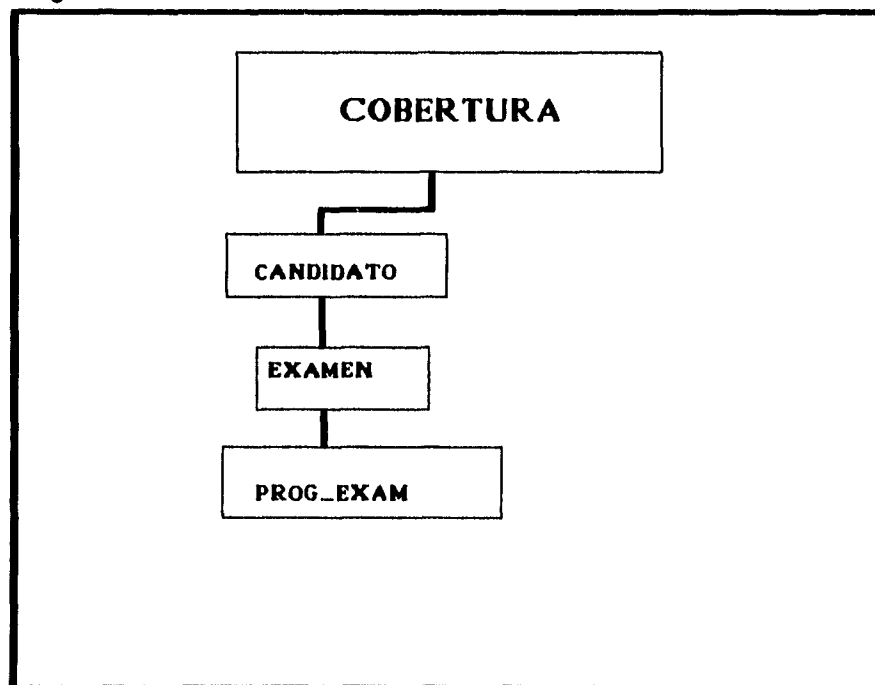


Estructura de datos tipo Jerarquica Figura I.4.3.1.

La relación entre cobertura y candidato es de 1:N.

En la figura anterior podemos observar que para cada cobertura existen uno o más candidatos. En la estructura de árbol invertido podemos identificar los siguientes elementos: nodos (cuadros, registros) y arcos (lineas, ligas).

Todos los nodos en una definición jerárquico de árbol representan tipos de registros. Todos los arcos representan una liga entre dos tipos de registros. Dado que existe al menos un arco entre cualquiera de dos nodos, en un árbol el arco no es etiquetado para ser distinguido de otros. La definición de un árbol jerárquico es básico para una base de datos jerárquica. Los tipos de registros especifican que tipos de registros son permitidos en la base de datos. Las ligas especifican la conexión permitida entre tipos de registros.

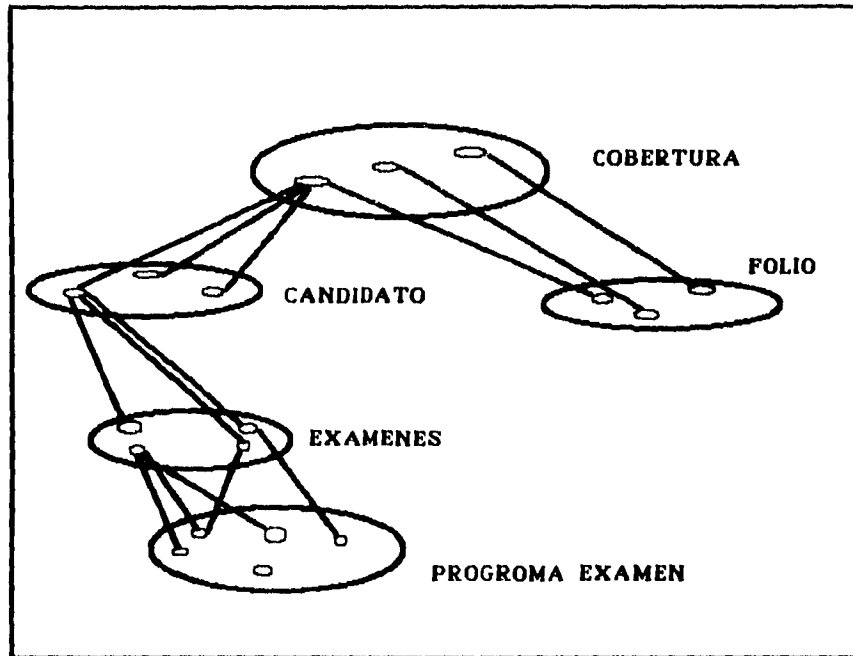


Definición Jerárquica de árbol Figura I.4.3.2.

En una definición de árbol jerárquico, existe una designación especial y corresponde al registro RAÍZ. Los otros tipos de registros son llamados DEPENDIENTES. El registro RAÍZ está en el nivel 1. Los registros dependientes son de niveles más bajos (2,3,..etc...). Así tenemos en la figura el tipo de registro COBERTURA es la Raíz (nivel 1), CAND (candidatos) y FOLIO son tipos de registros dependientes(nivel 2). El tipo de registro EXAMEN (nivel 3) y PROG (programación) se encuentra en el último nivel (nivel 4).

Una base de Datos jerárquica es una colección o selva de arboles desarticulados, es decir los nodos son registros. Cada árbol desarticulado es llamado BASE DE DATOS DE ÁRBOL y consiste de un registro RAÍZ y todos sus registros dependientes. Un registro raíz es una ocurrencia de un tipo de registro raíz, mientras que un registro dependiente es una ocurrencia de un tipo de registro dependiente. Todas las bases de Datos de Árbol son construidas de acuerdo a las conexiones permitidas en el árbol jerárquico de la Base de Datos.

En una base de Datos jerárquica los parientes y los hijos pueden ser identificados claramente de acuerdo a las conexiones entre registros. Por ejemplo:



Base de Datos Jerárquica Figura I.4.3.3.

Un registro de COBERTURA puede ser conectado a varios CANDIDATOS y a un solo FOLIO, en este caso se dice que la COBERTURA es el padre de los registros CANDIDATOS y el FOLIO. Estos últimos son por lo tanto los hijos de COBERTURA. Una ruta en una base de datos es una secuencia de registros, empezando del registro raíz. Por ejemplo la secuencia del registro COBERTURA, registro CANDIDATO, registro EXAMEN, es una ruta jerárquica, otra sería registro COBERTURA y registro FOLIO.

En una Base de datos jerárquica, puede haber un número variado de registros de cada tipo en los distintos niveles. Por ejemplo en la Figura I.4.3.3. hay varios registros de COBERTURA cada uno de los cuales puede ser conectado a varios registros de candidato, a su vez cada registro CANDIDATO puede ser conectado a varios registros EXAMEN. En conclusión todas las conexiones entre registros de acuerdo a las ligas establecidas en el árbol jerárquico son totalmente funcionales.

I.4.2.1. VENTAJAS Y DESVENTAJAS DEL MODELO JERÁRQUICO.

VENTAJAS	DESVENTAJAS
Relaciones de uno a muchos (un padre y muchos hijos)	Nodos dependientes
	Debe existir una raíz
Sencillo de implementar	Borrado en niveles altos.
Algoritmo de búsqueda mas rápidos.	

I.4.3. TOPOLOGÍA DE RED.

En general, en el modelo de red no existen restricciones en cuanto a los tipos de relaciones, así el modelo jerárquico es un caso particular de este modelo.

En general, el modelo reticular no tiene restricciones en las relaciones representadas por medio de ligas. Pueden representarse relaciones 1:1, 1:N y N:M. Sin embargo, algunos manejadores tienen sus restricciones en las ligas. Supongamos que las ligas pueden ser funcionales en una dirección, esto es, todas las ligas representan una relación 1:N. Por ejemplo, 2 registros del tipo Rj y Ri están conectados por una liga del tipo 1:N de los registros Ri a los Rj.

Además, cada registro R_i está conectado a muchos registros R_j . En este caso el registro R_i es nombrado PROPIETARIO de R_j .

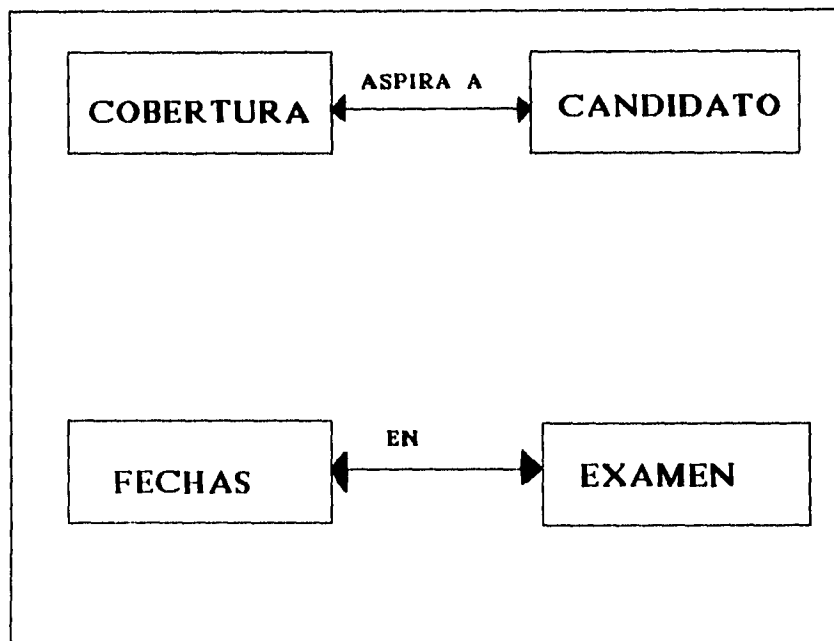
De la misma manera el conjunto de registros R_j 's conectados a un R_i , son llamados MIEMBROS del registro PROPIETARIO R_i .

El papel de los tipos de registros PROPIETARIOS y los MIEMBROS pueden ser representadas en una gráfica de red marcando las ligas directas desde el tipo de registro PROPIETARIO al tipo de registro MIEMBRO. Así una gráfica de relación es llamada DIAGRAMA DE ESTRUCTURA.

Si solo ligas del tipo 1:N son permitidas entre varios tipos de registros de una red, entonces una relación del tipo N:M representa un problema. Por ejemplo el tipo de registro EXAMEN y el tipo de registro PROG_EXAM tienen una relación N:M. En este caso un número de examen se proporciona a una persona que podrá presentar varios exámenes y serán registrados con este número, por lo tanto tendrán varias fechas de programación.

Una relación N:M puede ser representada sin utilizar ligas N:M, sino utilizando dos de tipo 1:N. Pero también es posible utilizar ligas N:M.

A diferencia del modelo jerárquico en el modelo tipo red todas las ligas deben ser identificadas por un nombre.



Estructura de datos diseñada para una topología tipo Reticular Figura I.4.4.1.

I.4.3.1. VENTAJAS Y DESVENTAJAS DEL MODELO DE RED.

VENTAJAS

Acceso muy rápido

No existe redundancia
de información

La relación 1:N
es eficiente .

DESVENTAJAS

Demasiada complejidad por el gran número de
apuntadores.

I.5. MODELO DE BASES DE DATOS PARA UN AMBIENTE MULTIUSUARIO.

Construcción de un manejador de base de datos para un ambiente multiusuario.

El diseño de un manejador de base de datos, se realiza tomando en consideración los servicios de apoyo que brinde el sistema operativo de la computadora. Este puede resultar muy simple o bastante complejo dependiendo del ambiente en que va a ser utilizado, dado que existen ambientes multiusuario y personal.

En un ambiente multiusuario, el manejador de base de datos debe de realizar las funciones siguientes:

- Interactuar con el manejador de archivos.
- Manejar el buffer.
- Manejar la integridad de la base de datos.
- Manejar la seguridad de la base de datos.
- Manejar la recuperación de caídas.
- Controlar la concurrencia.

Manejador de archivos.

El rendimiento de un sistema depende de la eficiencia de las estructuras de datos que se empleen para representar la información de la base de datos, y de que el sistema pueda operar en forma eficiente con esas estructuras.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

La estructura básica que maneja cualquier sistema operativo es el archivo, éste está organizado lógicamente como una secuencia de registros que se mapean en bloques en el disco. Una forma de mapear la base de datos a archivos es utilizar varios de éstos y almacenar solamente registros de longitud fija en un archivo dado. Una alternativa es estructurar los archivos de manera que puedan contener registros de longitudes distintas.

Entre los métodos que existen para implementar registros de longitud variable se encuentran los siguientes:

- Espacio reservado. Cuando los registros de longitud variable se fijan a una longitud máxima que nunca se va a exceder, como se muestra a continuación:

Longitud máxima de registros

Perryridge	102	Hayes	400	201	williams	900	218	Lile	700
Round Hill	305	Turne	350						
Mianus	215	Smith	700						
Centro	101	Johnson	500	100	Peterson	600			
Readwood	222	Lindsay	700						

Longitud máxima de registros Figura I.5.1.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

- Apuntadores. El registro de longitud variable se representa por una lista de registros de longitud fija, encadenados por medio de apuntadores. Este método se ejemplifica a continuación:

Apunt.	Registro			
	Round Hill	305	Turner	350
	Mianus	215	Smith	700
Ap_Reg5	Centro	101	Johnson	500
	Redwood	322	Lindsay	700
Ap_reg6		201	Williams	900
		110	Peterson	600
		218	Lyte	700

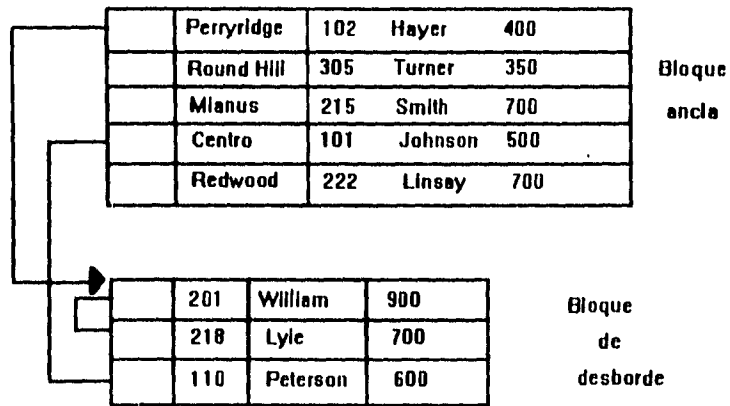
Apuntadores Figura I.5.2.

Un archivo puede considerarse como un conjunto de registros. Sin embargo, como los datos se transfieren entre el disco y la memoria principal por bloques, vale la pena asignar los registros a los bloques de tal manera que uno de éstos contenga registros relacionados entre sí y de esta manera mejorar el rendimiento del sistema evitando demasiados accesos a disco. Una estructura propuesta para lograr lo anterior es manejar dos tipos de bloques en un archivo:

- **Bloque ancla.** Contiene el primer registro de una cadena.
- **Bloque de desborde.** Contiene todos los registros que no son el primer registro de una cadena.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Para detallar esta estructura se presenta un ejemplo en la siguiente figura:

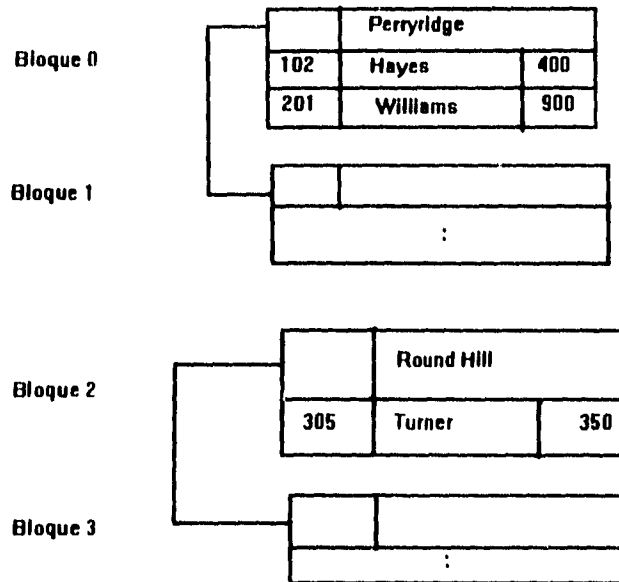


Bloque de desborde Figura I.5.3.

Así, todos los registros dentro de un bloque tienen la misma longitud. Esta estructura soluciona algunos problemas de representación y acceso a disco, sin embargo, presenta fallas en bases de datos bastante dinámicas, por lo que se propone una alternativa que ocupa mayor espacio. Esta estructura se denomina archivo cubeta, donde una cubeta consiste de varios bloques encadenados por apuntadores para representar los datos; la desventaja de esta estructura es que no puede compartirse

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

un bloque entre archivos. En la siguiente figura se muestra un ejemplo de esta estructura:



Archivo cubeta Figura I.5.4.

Uno de los objetivos primordiales del manejador de archivos es definir la estructura de los archivos que se va a emplear para la representación de los datos optimizando el número de accesos a disco, y de esta manera mejorar el rendimiento del sistema.

I.5.1. MANEJADOR DEL BUFFER.

Una forma de reducir los accesos a disco es tratar de mantener toda la base de datos en memoria principal. Dado que esto es imposible, es preciso manejar la asignación del espacio disponible en la memoria principal para almacenamiento de bloques. El buffer es la parte de la memoria principal que está disponible para almacenar copias de bloques del disco. El módulo que se encarga de manejar el espacio disponible en el buffer se denomina manejador del buffer. El manejador del buffer intercepta todas las solicitudes que hace el resto del sistema pidiendo bloques de la base de datos. Si el bloque ya está en el buffer, se pasa al solicitante la dirección del bloque en la memoria principal. Si el bloque no está en el buffer, el manejador del buffer lee el bloque del disco y lo escribe en el buffer, y pasa la dirección del bloque en memoria principal al solicitante.

El manejador del buffer, con el fin de servir adecuadamente al manejador de base de datos, realiza óptimamente las funciones antes mencionadas, utilizando técnicas depuradas para el manejo de memoria virtual, entre las que destacan las siguientes:

- **Estrategia de reemplazo.** Cuando no queda espacio disponible en el buffer, debe sacarse un bloque antes de que pueda escribirse uno nuevo. La estrategia óptima en un DBMS se basa en que el bloque al que se hizo referencia más recientemente, se graba en disco y se borra del buffer.

- **Bloques sujetos.** Para que el DBMS pueda recuperarse de caídas es necesario restringir las oportunidades para regrabar un bloque en el disco. Cuando no se permite que un bloque se vuelva a grabar a disco se dice que está sujeto.

- **Salida forzada de un bloque.** Existen situaciones en las que es necesario regrabar un bloque en el disco aunque no se necesite el espacio que ocupa en el buffer, esto se debe al hecho de que el contenido del buffer se pierde en una caída del sistema.

I.5.2. MANEJADOR DE AUTORIZACIONES Y DE INTEGRIDAD.

La información que se almacena en la base de datos debe protegerse contra: el acceso no autorizado, la destrucción o alteración con fines indebidos, y la introducción accidental de inconsistencias.

Es más fácil evitar la pérdida accidental de la consistencia de la información, que impedir el acceso mal intencionado a la base de datos.

Para lograr el objetivo anterior, es necesario que a los usuarios se les definan distintas formas de autorización respecto a partes de la base de datos. Entre éstas pueden mencionarse:

- **Autorización de lectura**, que permite leer, pero no modificar la base de datos.
- **Autorización de inserción**, que permite insertar datos nuevos, pero no modificar los ya existentes.
- **Autorización de actualizaciones**, que permite modificar la información, pero no permite la eliminación de datos.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Además de estas formas de autorización de acceso a datos, es posible autorizar al usuario para que modifique el esquema de la base de datos:

- **Autorización de índices**, que permite la creación y eliminación de índices.
- **Autorización de recursos**, que permite la creación de nuevas relaciones.
- **Autorización de alteración**, que permite agregar o eliminar atributos de una relación.
- **Autorización de eliminación**, que permite eliminar relaciones.

Estas formas de autorización, son una manera de proteger a la base de datos contra el acceso indebido o no autorizado. Pero el objetivo de las limitantes de integridad es muy diferente. Son una forma de garantizar que los cambios que hacen a la base de datos los usuarios autorizados, no van a resultar en una pérdida de la consistencia de la información, por lo tanto, las limitantes de integridad previenen el daño accidental a una base de datos.

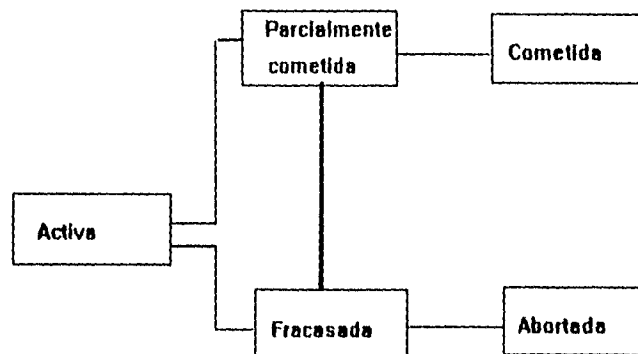
Entre los esquemas para la conservación de la integridad se tiene el de disparador, donde un disparador es una instrucción o serie de instrucciones que el sistema ejecuta en forma automática cada que se cumplen las condiciones que activan el disparador. Un esquema alternativo, para conservar la integridad de la base de datos es modificar la información.

I.5.3. MANEJADOR DE RECUPERACIONES.

Un sistema de cómputo está sujeto a fallas, donde éstas pueden ser de diversas causas: un aterrizaje de la unidad de disco, falta de energía eléctrica y errores de software. En cada una de éstas se pierde información relacionada con la base de datos, por lo que cada una debe manejarse en forma diferente. Dentro del modelo integral del manejador de bases de datos existe el módulo que es un esquema de recuperación que se encarga de detectar las fallas y restaurar la base de datos a un estado que existió antes de presentarse la falla.

En caso de que ocurra una falla, puede ser que el estado real del sistema deje de ser consistente, es decir, puede no reflejar ya el estado del mundo que se supone debe reflejar la base de datos.

Para conservar la consistencia es necesario que todas las transacciones sean atómicas, es decir que, o bien se ejecuten completamente todas las instrucciones asociadas con ella, o que no se ejecute ninguna. En la siguiente figura se muestra el diagrama de estados de una transacción:



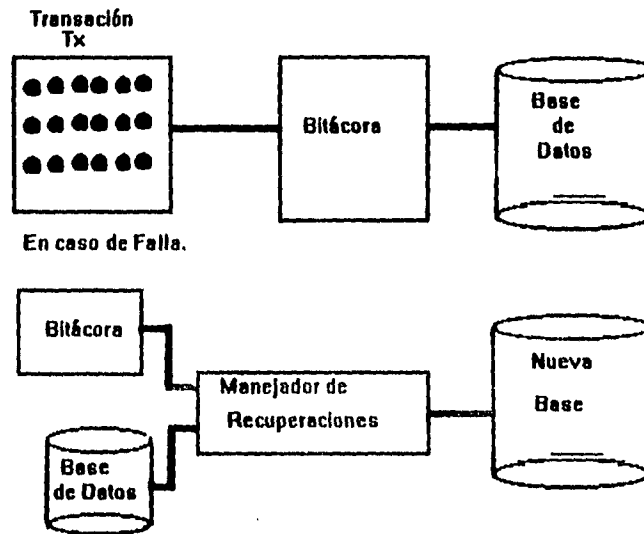
Estados de una transacción Figura I.5.3.1.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

El procedimiento de recuperación debe encargarse de mantener la propiedad de atomicidad. A continuación se describen algunos esquemas que permiten garantizar la atomicidad:

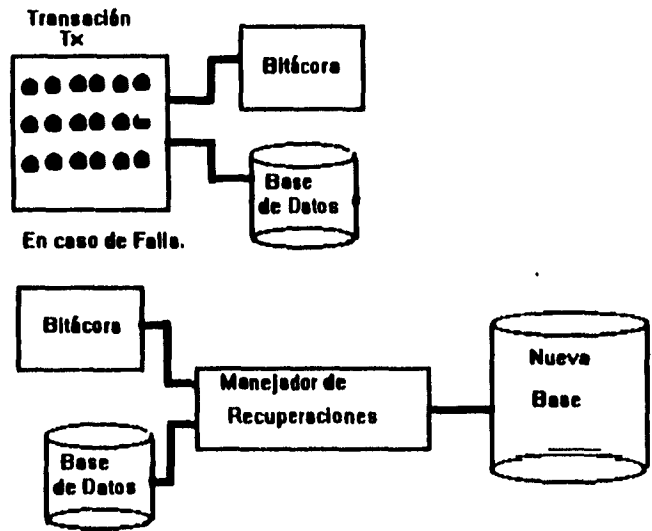
- Bitácora incremental con actualizaciones diferidas. Durante la ejecución de una transacción se postergan todas las operaciones de escritura hasta que aquella quede parcialmente completada. Todas las actualizaciones del sistema se registran en la bitácora del sistema que debe grabarse en una memoria estable. Una vez que la transacción está parcialmente completada, la información de la bitácora asociada a la transacción se utiliza para ejecutar las operaciones diferidas de escritura. Si el sistema se cae antes de que la transacción termine de ejecutarse, o bien, si la transacción aborta, se hace caso omiso de la información de la bitácora. Empleando la bitácora, el sistema puede manejar cualquier falla que no resulte en pérdida de información de memoria no volátil. Cuando no ocurre una falla, el esquema de recuperación consulta la bitácora y repite todas las transacciones completadas. Este esquema de recuperación se describe en la siguiente figura:

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios



Esquema de recuperación Figura I.5.3.2.

- Bitácora incremental con actualizaciones inmediatas: Todas las actualizaciones se aplican directamente a la base de datos, manteniéndose en almacenamiento estable una bitácora incremental de todos los cambios que se hacen en el estado del sistema. La siguiente figura ejemplifica este método de recuperación:



Bit

acora incremental del esquema de recuperación Figura I.5.3.2.

I.5.4. CONTROLADOR DE CONCURRENCIA.

En un ambiente multiusuario, la mayoría de las veces llega a presentarse el caso de que varias transacciones se ejecutan en forma concurrente en la base de datos, lo que conlleva a la posibilidad de que se pierda la consistencia de los datos. Por lo tanto, es necesario que el sistema controle la interacción entre las transacciones concurrentes, control que se logra al utilizar algunos de los diversos mecanismos que se conocen como esquemas de control de concurrencia:

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

- Protocolos para candados.
- Ordenamiento por hora de entrada.
- Técnica de validación.
- Técnica de multiversión.

Todos estos esquemas garantizan la característica de seriabilidad que se define como: "es requisito indispensable que cualquier cuadro que se produzca al procesar un conjunto de transacciones de manera concurrente, sea computacionalmente equivalente al cuadro que se produzca al ejecutar estas transacciones en serie en algún orden".

Un protocolo para candados es un conjunto de reglas que se definen cuando una transacción puede poner o quitar candados a cada uno de los datos de la base de datos. El protocolo para candados de dos fases permite a una transacción ponerle candado a un nuevo dato, esto solo si no le ha quitado todavía el candado a algún otro.

Un esquema de ordenamiento por hora de entrada garantiza la seriabilidad escogiendo por adelantado un ordenamiento entre cada pareja de transacciones. A cada transacción del sistema se le asigna una hora de entrada única. Las horas de entrada de las transacciones determinan el orden de seriabilidad.

Un esquema de validación es el método de control de concurrencia adecuado en los casos en los que la mayor parte de las transacciones efectúan solo lecturas, por lo que la incidencia de conflictos entre transacciones es baja. Por cada transacción en el sistema se asigna una hora de entrada única. Las horas de entrada determinan el orden de seriabilidad.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

El esquema de control de concurrencia de multiversión supone que cada operación de write(Q) genera una nueva versión del dato "Q". Cuando se solicita una operación de read(Q), el esquema escoge una de las versiones del dato "Q" para leerse, de tal manera que la versión "Q" que elija garantice la propiedad de seriability.

Manejador de base de datos para un ambiente multiusuario.

El modelo que se presenta como ejemplo, es el manejador de base de datos ORACLE, bajo el sistema operativo UNIX. Este modelo está construido, tomando en consideración algunas características funcionales del sistema operativo:

- El poder compartir código y datos entre procesos.
- Seguridad de la base de datos.
- Control de acceso a usuarios.
- Facilidades de comunicación entre procesos.

La siguiente figura muestra la estructura básica de ORACLE, describiendo los módulos funcionales y la relación entre ellos:

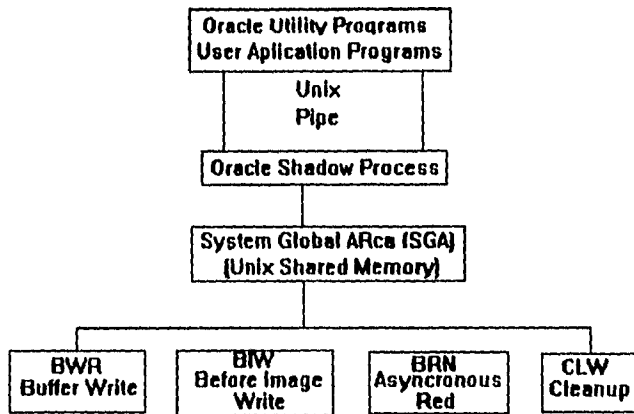


Figura 1.5.4.1

1.5.5. PROGRAMAS DE USUARIO Y UTILERÍAS DE ORACLE.

Cada que un programa de usuario, o utilidad de ORACLE es invocado para acceder la base de datos, se genera un nuevo proceso denominado "proceso shadow". Los "procesos shadow" son los encargados de acceder directamente la base de datos, ya que los programas de usuarios son bloqueados para acceder en forma directa la base de datos. La comunicación entre los programas de usuario y los "procesos shadow", se realiza por medio de los "pipes" de UNIX que permiten la interconexión de dos procesos.

Procesos shadow.

En un ambiente multiusuario, los "procesos shadow" de ORACLE colocan todas las solicitudes de servicio de los usuarios en un área de memoria compartida, denominada: área global del sistema (SGA: System Global Area). Estas solicitudes son atendidas por los procesos background de ORACLE, y los resultados generados por éstos procesos son depositados en el SGA.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Parte del área de datos de los "procesos shadow" es utilizada como área de contexto. Esta última es utilizada para almacenar información relativa a instrucciones de SQL.

Area global del sistema.

El SGA está residente en uno o más segmentos de la memoria principal. Esta área es generada durante la inicialización, y está presente hasta que se cierra la base de datos.

Para soportar el medio ambiente multiusuario, los procesos ORACLE utilizan una estructura de datos que reside en el SGA. Esta estructura incluye:

- Tabla de candados.
- Listas de espera.
- Información para la recuperación de transacciones.
- Información para control de procesos.
- Información de la configuración del sistema.
- Buffer de copia anterior.
- Buffer de la base de datos.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Procesos background de ORACLE.

Estos procesos son los que atienden las solicitudes de los usuarios a la base de datos.

Los procesos background son:

- BWR: Manejo del buffer de escritura de la base de datos.
- BIW: Manejo del archivo de copia exterior.
- ARH: Lecturas asíncronas adelantadas.
- CLR: Procesos de borrado.

Seguridad, Consistencia y Concurrencia.

La seguridad de la base de datos, está implementada por los niveles de permisos que maneja al sistema operativo UNIX.

El manejador de base de datos ORACLE provee de forma automática la protección completa de la consistencia de la base de datos. Esto se logra por medio del uso de un buffer que almacena una imagen anterior de un bloque de datos. Antes de que una unidad de trabajo sea iniciada se hace uso del buffer. Una unidad de trabajo se define como una transacción, donde ésta se compone por todas las operaciones relacionadas con cada línea de un programa.

ORACLE provee para el control de accesos concurrentes un mecanismo interno de candados; el sistema automáticamente coloca candados a datos accesados por unidades lógicas de trabajo. Una unidad de trabajo es identificada como una secuencia de instrucciones de SQL.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Los modelos de bases de datos para ambientes multiusuarios contemplan las siguientes características:

1. Acceso Múltiple.
2. Protección de datos (DEAD-LOCK).
3. Niveles de lock. Estos pueden ser a nivel:
 - Recursos.
 - Archivos.
 - Registros.
4. Interrelación de archivos (BLOQUEO).

Cuando las transacciones corren concurrentemente, existen algunos posibles intervalos de operaciones de lectura y escritura, que al acceder los mismos items, provocan efectos impredecibles.

5. Facilidad de programación:
 - Manejo de pantalla.
 - Reportes y consultas.
 - Control de procesos.
 - Generación de archivos.
 - Manejo de menús.
 - Control del sistema operativo.
 - Documentación.
 - Posibilidades para programadores expertos.

7. Facilidad para consultas no programadas (QUERIES).

Posibilidad de:

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

- Consultas en línea.
- Ordenamiento de datos.
- Selección.
- Interacción con archivos relacionados.
- Facilidad de uso.
- Repetición automática del tipo de consulta.

8. Eficiencia:

- Indexación.
- Sort.
- Ejecución de programas. - Operación con archivos.
- Generación de aplicaciones (facilidad de programación).

9. Seguridad:

- Integridad de los datos.
- Facilidad para respaldar y recuperar.
- Facilidad para reconstruir.
- Facilidad para cambiar definiciones.
- Control de acceso a programas o datos.

10. Transportabilidad.

11. Facilidad para importar y exportar datos.

12. Orientación principal.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Las bases de datos para ambientes multiusuarios cubren los estándares de la prueba ACID (atomicidad, consistencia, aislamiento y durabilidad) creada por el Consejo de Procesamiento de Transacciones (TPC).

Atomicidad.

Atomicidad significa que una transacción debe ser una unidad discreta que realiza una tarea completa, aún cuando pueda estar compuesta de un cierto número de operaciones. Una indagación o un cambio de direcciones son ejemplos de transacciones simples. Las transacciones realizan todas o ninguna de sus operaciones: no es suficiente con 99 de cada 100.

Si no puede completarse una transacción, el sistema tiene que asegurar que la transacción que se ha abandonado no tiene efecto sobre la información. Esta regla es la base para las operaciones Comprometer (Commit) y Reanudar (Rollback) que se usan en el procesamiento de las transacciones. Cuando se ejecuta una transacción, el sistema sigue la pista del trabajo en un registro, incluyendo los valores de los datos afectados. Cuando el DBMS completa con éxito la transacción, se "compromete" de forma permanente con los cambios. Si una transacción no se completa con éxito, el sistema usa el registro para restaurar o reanudar la base de datos al estado que tenía antes de la ejecución.

La atomicidad garantiza que la base de datos está siempre en un estado constante entre transacciones, aún si ocurre un error o un programa es abandonado. Una base de datos necesita un verdadero proceso de transacciones para cumplir con este requisito.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Consistencia.

La consistencia se refiere a la integridad de la información. Las bases de datos organizan la información descomponiéndola en elementos de datos que se relacionan. Una base de datos con integridad mantiene reglas que gobiernan las relaciones entre sus elementos de información:

Por ejemplo:

- Todos los empleados deben tener un número de nómina.
- Las cantidades de las facturas deben ser mayores que cero.
- Siempre que se entre una cantidad en un libro, debe hacerse una entrada similar al libro mayor.

Las bases de datos son consistentes cuando toda la información cumple con las reglas. A menos que el DBMS tenga una forma de hacer cumplir las reglas, una base de datos puede tener la consistencia después que se actualice. Algunas veces un cambio complejo hace que la base de datos no sea consistente temporalmente, pero al final, debe restablecerse el orden. La unidad de trabajo que lleva a la base de datos de un estado consistente a otro se llama una transacción.

Aislamiento.

El aislamiento, que se logra mediante técnicas de reservación, asegura que el acceso de cada usuario a un registro de la base de datos está garantizado mientras este trabajando con él. Cuando una transacción reserva un objeto, nadie más puede actualizarlo hasta que se retire la reservación.

Capítulo I Modelos de Base de Datos para un Ambiente Multiusuarios

Durabilidad.

Un DBMS pasa la prueba de durabilidad si puede preservar los efectos de todas las transacciones comprendidas, y garantizar la consistencia de los datos después de recobrase de una pérdida de alimentación o un fallo de hardware. Un sistema completamente confiable incluye protecciones para el hardware, tales como fuentes de alimentación interrumpible/no-breaks/ups y copias de discos simultáneas (disk mirroring). Pero una base de datos bien diseñada mantendrá una historia, de forma que pueda restaurarse a sí misma. Para sobrevivir a una breve pérdida de alimentación, lo que se necesita es una operación de deshacer/repertir (undo/redo) registro; pero para recuperarse de un fallo de hardware, el sistema tiene que ir hacia adelante y recrear las transacciones partiendo de un archivo.



CAPITULO II

IDENTIFICACIÓN Y EVALUACIÓN DEL PROCEDIMIENTO DE PAGO A TRANSPORTISTAS



Capítulo II Análisis de funciones y responsabilidades del área de distribución

II.1.- ANÁLISIS DE FUNCIONES RESPONSABILIDADES DEL ÁREA DE DISTRIBUCIÓN.

Introducción.

La descripción de procedimientos tiene como propósito fundamental el dar a conocer formalmente los aspectos que integran la unidad de Administración y Servicios, tales como estructura orgánica, objetivos, política y funciones así como la relación que guarda con los órganos que la integran.

Otros de los fines es el servir como, instrumento de inducción para el personal de nuevo ingreso y /o personal dentro de petróleos mexicanos que se integre a esta unidad.

Será un documento que debe conocer todo el personal que integra la unidad de administración y servicios, con el objeto de que conozca que lugar le corresponde al puesto que desempeña dentro de la estructura orgánica, así como las funciones que debe desarrollar.

La revisión y actualización de los procedimientos será cuando la estructura orgánica de la unidad así lo requiera para cumplir con sus objetivos y de acuerdo con las políticas y lineamientos marcados por la gerencia de desarrollo institucional

Capítulo II Análisis de funciones y responsabilidades del área de distribución

II.1.2. ANTECEDENTES.

Se han emitido diversos ordenamientos en materia de racionalidad, austeridad y disciplina presupuestal, a los que petróleos mexicanos se ha apegado, en cuanto a racionalizar y simplificar su estructura orgánica a efecto de utilizar los recursos estrictamente indispensables para llevar a cabo sus actividades mediante los acuerdos establecidos.

Con motivo de dicha compactación de estructura y dentro del ámbito de la subdirección comercial, se originó la coordinación ejecutiva de distribución misma que presentó la propuesta de organización. Dentro de esta estructura esta la unidad de administración de servicios, dependiendo directamente de la coordinación con el objeto de administrar y controlar los procedimientos de pago por concepto de fletes a transportistas, así como los recursos humanos, materiales y financieros de las unidades que la integran, las cuales se mencionan a continuación:

- 1) Gerencia de distribución y Transporte. (G.D.T.)
- 2) Gerencia de Transporte marítimo y administración portuaria (*)
- 3) Gerencia de arrendamiento y contrato de servicio para la distribución. (G.A.C.D.)

(*) A esta gerencia solo se mantendrá un seguimiento de control administrativo y de integración estadística para fines de diagnóstico y análisis de coordinación en materia de recursos humanos, materiales, financieros y/o presupuestales.

II.1.3. OBJETIVO GENERAL.

Lograr el uso racional de los recursos de la gerencia de distribución y transporte y la G.A.C.D., así como administrar sus sistemas de información, proporcionando de esta manera un apoyo administrativo para sugerir.

II.1.3. POLÍTICAS GENERALES.

Las políticas generales norman las actividades y planeamientos de cada una de las unidades básicas.

- 1) La unidad de administración y servicios implementará las acciones necesarias para desarrollar sus funciones dentro de un esquema de coordinación y participación, con las dependencias que integran la coordinación ejecutiva de distribución, así como las unidades institucionales responsables de las funciones administrativas y de control
- 2) Instrumentar los mecanismos adecuados para que las unidades de la coordinación, establezcan una estrecha colaboración, con todas las dependencias de la institución, el instituto mexicano del petróleo y con las dependencias gubernamentales, relacionadas con la distribución y transporte de productos.
- 3) Pugnar en forma permanente y sistemática a fin de que las acciones que se desarrollan en la unidad de administración y servicio se orienten a la racionalización de los recursos disponibles, para optimizar la consecución de los objetivos y metas generales establecidas por la coordinación.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

4) Promover que todos los presupuestos de egresos de la G.D.T. y la G.A.C.D., se elaboren con la finalidad de incrementar la productividad.

5) Mantener actualizados los programas de trabajo en las diferentes áreas con el fin de resolver los problemas prioritarios de la coordinación ejecutiva de distribución en materia de control de inventarios.

6) Promover que la selección del personal para ocupar las plazas en las unidades que integran la coordinación, este orientada a alcanzar las metas propuestas por la misma, en materia de productividad.

II.1.4. FUNCIONES GENERALES.

1) Planear , organizar, dirigir y controlar las acciones de la unidad de administración y servicios, para apoyar a las sugerencias de distribución y transporte y la de embarcaciones y contratos para la distribución.

2) Adquirir y suministrar en forma oportuna el material, equipo y refacciones que sean necesarias para la G.D.T. y la G.A.C.D..

3) Establecer con las dependencias correspondientes, un sistema integral de presupuestos por programas congruentes con un estilo de dirección dinámica, y que permite organizar el desarrollo de las actividades mediante la previsión del futuro.

4) Planear los objetivos, políticas, programas, sistemas y procedimientos que norman las actividades operativas de la G.D.T. y la G.A.C.D., en materia de presupuestos por programas en sus diversas fases.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

- 5) Establecer controles por área de responsabilidad (C.A.R.) en todas las dependencias que integran la coordinación (de sus recursos humanos y materiales).
- 6) Mantener una comunicación permanente con los diferentes representantes de la G.D.T. y la de arrendamiento de embarcaciones y contratos para la distribución, en los centros de trabajo, a fin de recibir de estos la información que se requiera, para el seguimiento físico-financiero del presupuesto, y la administración de los recursos humanos y materiales, estableciendo orden y disciplina en el ejercicio del gasto.
- 7) Participar activamente en el desarrollo y administración de políticas, programas y procedimientos para otorgar al personal trato justo y equitativo, así como satisfacción y seguridad en el trabajo.
- 8) Realizar el diagnóstico global de los sistemas administrativos de la coordinación y proponer en su caso, políticas y acciones para solución de la problemática o desviaciones detectadas.
- 9) Elaborar estados financieros consolidados de la coordinación, así como efectuar análisis e interpretación de los mismos.
- 10) Determinar los costos unitarios promedio de la coordinación por gerencia y actividad.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

11) Coadyuvar en el diseño y realización de programas de adiestramiento, capacitación y desarrollo personal, coordinando las acciones requeridas en las unidades demandantes, con la gerencia de personal y en el Instituto Mexicano del Petróleo.

12) Vigilar la adecuada implementación de los programas y proyectos de racionalización de las estructuras organizativas y los sistemas administrativos, a fin de garantizar el cumplimiento efectivo de sus cometidos.

II.1.5. ORGANIZACIÓN.

La estructura organizacional de Comercializadora Ejecutiva de Distribución (C.E.D.) fue diseñada con los siguientes criterios:

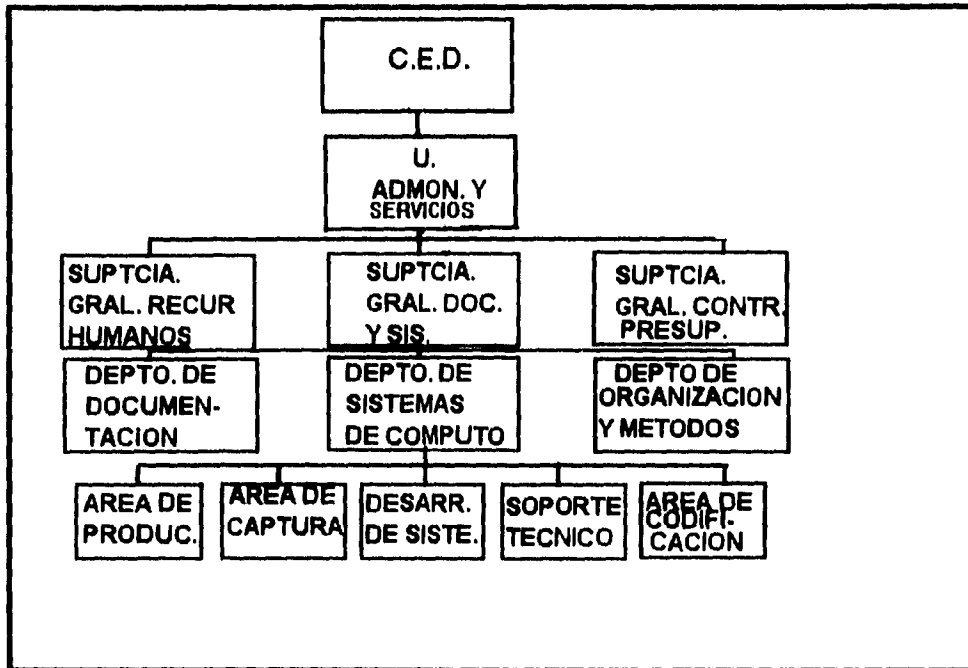


Figura II.1.5.1 Diagrama organizacional C.E.D.

II.1.5.1 ORGANIZACIÓN POR FUNCIÓN

Para el segundo extracto jerárquico el de las superintendencias generales, el tipo de organización utilizado fue funcional en virtud de la delimitación precisa de éstas:

- 1) De administración de recursos humanos y materiales.
- 2) De documentación y sistemas.
- 3) De integración y control presupuestal.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

II.1.5.2. ORGANIZACIÓN POR ESPECIALIDAD.

Para el tercer extracto jerárquico el de las superintendencias adscritas a las superintencias generales, el tipo de organización utilizado fue por especialidad en virtud a sus características funcionales:

- 1) De administración de recursos humanos.
- 2) De administración de recursos materiales.
- 3) De documentación.
- 4) De sistemas de cómputo.
- 5) De organización y métodos.
- 6) De integración presupuestal.
- 7) De control presupuestal.

II.5.1.3. ESTRUCTURA ORGÁNICA.

1. Unidad de administración y servicios.
2. Superintendencia General de administración de recursos humanos y materiales.
 - 2.1. Departamento de administración de recursos humanos.
 - 2.2. Departamento de administración de recursos materiales.
3. Superintendencia General de documentación y sistemas.
 - 3.1. Departamento de documentación.
 - 3.2. Departamento de sistemas de cómputo.
 - 3.3. Departamento de organización y métodos.
4. Superintendencia general de Integración y control presupuestal.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

4.1. Departamento de integración presupuestal.

4.2. Departamento de control presupuestal.

II.1.5.4. OBJETIVOS, POLÍTICAS, FUNCIONES Y ORGANIGRAMAS DE LAS UNIDADES ADMINISTRATIVAS CENTRALES.

1. SUPERINTENDENCIA GENERAL DE ADMINISTRACIÓN DE RECURSOS HUMANOS Y MATERIALES.

OBJETIVO.

Lograr el uso racional de los recursos humanos y materiales de la G.D.T. y la G.A.C.D., a efecto de apoyar su funcionamiento integral.

POLÍTICAS.

Actualizar los procedimientos de trabajo, que tengan como finalidad incrementar la productividad del personal de los órganos de la G.D.T. y de la G.A.C.D..

Utilizar los métodos científicos de investigación más eficaces para mantener el más alto grado de eficiencia en los sistemas y procedimientos que permitan alcanzar los objetivos que persiguen la G.D.T. y la G.A.C.D..

Lograr que los planes y programas de capacitación al personal de la coordinación sean cumplidos y asimilados y mantener el compañerismo entre los integrantes.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Promover que la selección del personal para ocupar las plazas en la G.D.T. y la G.A.C.D., este orientada a alcanzar las metas propuestas en materia de productividad.

Mantener las actividades de la superintendencia, dentro del marco de las normas establecidas por la institución y el gobierno federal.

Procurar que el personal adscrito a la gerencia de distribución y transporte y la de arrendamiento y contratos para la distribución, tenga la facilidad para la actualización de sus conocimientos técnicos y profesionales, promoviendo para ello su capacitación y desarrollo; reconociendo y estimulando sus méritos procurando alcancen un nivel ético y profesional adecuado.

FUNCIONES.

Planear , programar y coordinar las necesidades de recursos humanos, controlando permanentemente en coordinación con la superintendencia de documentación y sistemas, la integración del personal de las dependencias adscritas a la G.D.T. y la G.A.C.D..

Resolver conjuntamente con la gerencia de relaciones laborales los problemas de carácter laboral, en que incurra el personal profesional técnico, de oficina y obrero. Proponiendo también los ajustes necesarios a la plantilla de integración de los recursos humanos en coordinación con la misma gerencia.

Dirigir, coordinar y/o formular los planes y programas de capacitación y desarrollo al personal directivo, profesional, técnico de oficina y obrero de la G.D.T. y la G.A.C.D..

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Estudiar y analizar para su correcta interpretación o proposición de cambios a los reglamentos, convenios y normas correspondientes con relación al trato del personal de confianza, y al contrato colectivo de trabajo.

Controlar y en su caso realizar los trámites administrativos, en relación a contratación, promoción, prestaciones y manejo en general de los recursos humanos dependientes de la G.D.T. y la G.A.C.D., siguiendo los lineamientos y normas establecidas por la gerencia de relaciones laborales.

Integrar las necesidades de los materiales de acuerdo a los planes y programas de las dependencias operativas de la G.D.T. y la G.A.C.D., analizando las existencias en los almacenes y las solicitudes de los materiales, a fin de evitar excedentes y los obsoletos, promoviendo en su caso el rechazo o modificaciones de las mismas.

Coordinar con las gerencias de proveeduría y almacenes y ramas involucradas, todos los trámites relacionados con la gestión de requisiciones de materiales y equipo para la G.D.T.s y la de arrendamientos de embarcaciones y contratos para la distribución (muebles y vehículos); requeridos por esta dependencia, así como activar el suministro urgente de esos recursos materiales.

Establecer, mantener y/o apoyar los sistemas de control de materiales, que permita la actualización de inventarios, estadísticas de consumo, materiales defectuosos, traspasos y disponibilidad.

Integrar la información correspondiente a los recursos humanos y materiales de la coordinación ejecutiva de distribución.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

2. DEPARTAMENTO DE ADMINISTRACIÓN DE RECURSOS HUMANOS

OBJETIVO.

Coadyuvar al máximo aprovechamiento de los recursos humanos adscritos a la gerencia de la distribución y transporte y la G.A.C.D., a través del establecimiento de programas, actividades y acciones pendientes a optimizar su administración y mantener, elevar o desarrollar la capacidad de trabajo de sus integrantes, mediante adecuados planes y programas de capacitación, sin descuidar su identificación y aceptación para con la institución la coordinación y sus miembros.

POLÍTICAS.

Contribuir al cabal cumplimiento de las normas y lineamientos que institucionalmente son aplicables en la administración de recursos humanos.

Fomentar la instrumentación de mecanismos que permitan el desarrollo de los planes de trabajo en forma controlada, simplificando trámites que coadyuven a su evaluación contribuyendo de esta manera a una administración más objetiva de los recursos.

Fortalecer las relaciones a nivel interno y externo de la coordinación con otras dependencias a efecto de lograr armonía y dinámica en los asuntos a cargo de este departamento.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Pugnar para que esta área cuente con las innovaciones técnicas, así como los adelantos en materia administrativa a fin de estar permanentemente actualizados y lograr mejores niveles de calidad en los servicios a otorgar.

Procurar que los servicios de apoyo en los que se refieren a los recursos humanos, se proporcionen con calidad, suficiencia y en el momento oportuno, para contribuir al cumplimiento de los objetivos de esta unidad.

FUNCIONES.

Gestionar y controlar todo lo relativo a promociones, ascensos, capacitación e integración de personal ante la coordinación ejecutiva de recursos humanos.

Realizar y controlar los trámites administrativos inherentes al movimiento de personal de la coordinación, (contrataciones, substituciones, ascensos, vacaciones, permisos y comisiones).

Establecer y mantener permanentemente actualizado el control de las plazas de la G.D.T. y la G.A.C.D. (definitiva y extraordinaria).

Elaborar de acuerdo a los alineamientos institucionales planes y programas anuales y extraordinarios de capacitación entrenamiento y desarrollo para el personal profesional, técnico y de oficina de la G.D.T. y la G.A.C.D., aplicando los métodos y recursos educacionales idóneos.

Coordinar con el instituto mexicano del petróleo y la gerencia de relaciones laborales la planeación, desarrollo, supervisión y evaluación de los recursos de capacitación al

Capítulo II Análisis de funciones y responsabilidades del área de distribución

personal adscrito a la G.D.T. y la de arrendamiento de embarcaciones y contrato para la distribución.

Diseñar en coordinación con organización y métodos los sistemas y/o procedimientos para la mejor planeación de recursos humanos y evaluación de la actuación del personal.

Presentar a la G.D.T. y a la G.A.C.D. en las negociaciones administrativo-sindicales, necesarias para la estructuración e integración de personal de estas dependencias ante la coordinación ejecutiva de recursos humanos.

Proporcionar a organización y métodos la información necesaria para efectuar análisis y estudios de evaluación de puestos, para determinar cada nivel tabular de acuerdo a funciones y responsabilidades, proporcionando los resultados correspondientes a la coordinación ejecutiva de recursos humanos.

Estudiar y promover acciones de formación y de desarrollo internas, tales como autoformación, formación en el puesto, realización de seminarios por área de especialización, mejoramiento de canales de ascenso, comisiones de un centro de trabajo a otro, con el fin de intercambiar todas aquellas técnicas o procedimientos, tendientes a elevar el índice de productividad.

Reunir permanentemente los elementos de juicio necesarios para evaluar la aptitud del personal sugiriendo en su caso los nombres de los que deban considerarse para promociones temporales o definitivas en los casos de vacantes de nueva creación.

Elaborar programas de inducción y entrenamiento para el personal de nuevo ingreso.
Gestionar movimientos del personal de planta y transitorios.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Gestionar, controlar y verificar el pago del tiempo extra, de viáticos erogados por el personal de las gerencias de distribución y transporte y la de arrendamiento de embarcaciones y contratos para la distribución, ante la coordinación ejecutiva de recursos humanos.

Solucionar asuntos relacionados con las altas, incapacidades, jubilaciones, circulares, permisos económicos y renunciables, etc....

Recibir a la representación sindical y atender a las peticiones que en primera instancia le correspondan o sea competencia de la superintendencia de recursos humanos.

Atender a funcionarios y trabajadores de la G.D.T. y la G.A.C.D..

Gestionar, atender y solucionar los asuntos relacionados con el personal foraneo.

Integrar la información correspondiente a los recursos humanos de la coordinación ejecutiva de distribución.

1.3. DEPARTAMENTO DE ADMINISTRACIÓN DE RECURSOS MATERIALES

OBJETIVO.

Ayudar al máximo aprovechamiento de los recursos materiales de las gerencias de distribución y transporte y la de arrendamiento de embarcaciones y contratos para la distribución, a efecto de apoyar la gestión administrativa de éstas dependencias.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

POLITICAS.

Lograr que la planeación sea una herramienta aplicable en todas las fases de operación a fin de incrementar los niveles de productividad de ésta dependencia.

Procurar que los servicios de apoyo en lo que se refiere a los recursos materiales, se proporcionen con la calidad, suficiencia y en el momento oportuno, para contribuir al cumplimiento de los objetivos de esta unidad.

Fomentar la instrumentación de mecanismos que permitan el desarrollo de los planes de trabajo en forma controlada, simplificando tramites y que coadyuven a su evaluación contribuyendo de esta manera a una administración mas objetiva de los recursos materiales.

Contribuir al cabal cumplimiento de las normas y lineamientos que institucionalmente, son aplicables en la administración de los recursos materiales.

FUNCIONES.

Elaborar planes y programas para la mejor administración de los recursos materiales asignados a las gerencias de distribución y transporte y la de arrendamiento de embarcaciones y contratos para la distribución.

Aprovisionar y suministrar papelería, artículos de escritorios, mobiliario y equipo de oficina a las gerencias de distribución y transporte y la de arrendamiento de embarcaciones y contratos para la distribución.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Coordinar con la gerencia de proveeduría y almacenes todos los trámites relacionados con la gestión de requisiciones de materiales y equipo necesarios para el suministro de estos, a las gerencias de distribución y transporte y la de arrendamiento de embarcaciones y tratos para la distribución.

Establecer y mantener los sistemas de control de materiales, que permitan la actualización de inventarios, estadísticas de consumo, materiales defectuosos, traspasos y disponibilidades.

Controlar y gestionar la reparación y mantenimiento del mobiliario y equipo de oficina.

Inventario y resguardo de los bienes muebles.

Inventariar y controlar el mobiliario y equipo de oficina asignado a las gerencias de distribución y transporte y la de arrendamiento de embarcaciones y contratos para la distribución, en sus diferentes fases. detención de necesidades, elección de proveedores, elaboración de requisiciones, tramite de autorización y seguimiento de materiales, recibo y entrega en custodia del bien solicitado.

Integrar y mantener actualizado el inventario de vehículos asignados a funcionarios, así como de empleados sujetos al régimen de vehículos.

Coordinar y gestionar los tramites necesarios para la distribución de espacios para oficina.

Coordinar y gestionar los tramites necesarios para la asignación de estacionamientos.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Integrar la información correspondiente a los recursos materiales de la coordinación ejecutiva de distribución.

2. SUPERINTENDENCIA GENERAL DE DOCUMENTACION Y SISTEMAS

OBJETIVOS.

Apoyar y cimentar la toma de decisiones de la coordinación ejecutiva de distribución, mediante el desarrollo, implantación y mantenimiento de sistemas informáticos.

Atender los pagos a terceros por concepto de fletes, arrendamientos de equipo de transporte, contratos de servicio y otros, vigilando su cumplimiento dentro del tiempo marcado para su trámite.

Optimizar el funcionamiento administrativo de las dependencias adscritas a la coordinación ejecutiva de distribución, mediante la actualización de sus estructuras orgánicas y de los métodos y procedimientos de trabajo de operación.

POLITICAS.

Fomentar la utilización del procedimiento electrónico de datos, como un instrumento de apoyo en las funciones de planeación, dirección, supervisión y control.

Pugnar para que se cuente con la infraestructura informática necesaria.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Procurar la actualización técnica profesional del personal adscrito a la superintendencia general.

Promover la aplicación de técnicas avanzadas, a fin de fortalecer la función de informática, organización y sistemas administrativos.

Evitar un uso irracional de los recursos de cómputo (hardware y software).

Promover el desarrollo de sistemas informáticos bajo técnicas avanzadas y de actualidad.

Procurar que los sistemas y procedimientos de trabajo en operación, permanezcan actualizados y dentro del marco legal y administrativo que norman las actividades de la coordinación.

Pugnar porque el tramite de pagos a terceros prestadores de servicios se realice conforme a las leyes y reglamentos establecidos por las diversas autoridades internas y/o externas que regulan el pago, de productos transportados por autotank y/o carro tank.

F U N C I O N E S .

Definir y vigilar la aplicación de políticas informativas.

Desarrollar, implantar y mantener sistemas computarizados de operación, control e información que requieran la G.D.T. y la G.A.C.D., para cumplir con sus objetivos.

Planear, seleccionar, operar y mantener los equipos de procesamiento de datos requeridos para la implantación de los sistemas computarizados.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Evaluar los sistemas de información existentes para actualizarlos, mejorarlos, aplicarlos o suprimirlos.

Integrar y mantener actualizado el inventario de equipos de cómputo y software de la coordinación.

Mantener estrecha comunicación con todas las dependencias de la coordinación, a fin de proporcionarles el apoyo necesario para el cumplimiento de sus funciones.

Elaborar estudios de viabilidad económica para sustentar los requerimientos de adquisición de equipo de cómputo, su renta y/o contratación de servicios computacionales.

Mantener y actualizar los sistemas administrativos en operación.

Promover y aplicar las técnicas más avanzadas de análisis, diseño y programación de sistemas computarizados.

Capacitar y asesorar al personal de las dependencias de la coordinación que operaran los sistemas administrativos computarizados.

Analizar permanentemente la estructura organizacional y de recursos humanos de la coordinación, proponer e instrumentar las acciones preventivas y correctivas que contribuyan a optimizar su aprovechamiento, manteniendo actualizado el manual de organización de la coordinación.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Realizar los tramites de revisión y autorización de pagos de servicios prestados por el transporte de productos petrolíferos y petroquímicos de petróleos mexicanos, en sus diversos modos y medios que son transportados por autotanque y carrotanque, así como por los trabajos de mantenimiento e instalaciones de embarques y transporte y por los nuevos desarrollos de obra civil.

Verificar el aseguramiento y afianzamiento, tanto de la carga transportada como de los vehículos utilizados en el servicio.

2.2 DEPARTAMENTO DE DOCUMENTACION.

OBJETIVO.

Lograr que los tramites de pago a terceros (prestadores de servicios) que se efectúan en el ámbito de operación de transporte terrestre, se realicen conforme a las normas establecidas por las diversas autoridades internas y/o externas que regulan el pago, así como cumplir con los tiempos asignados para su trámite.

POLITICAS.

Promover la optimización del servicio de pago a terceros, en condiciones de calidad, oportunidad y costo.

Procurar que los sistemas y procedimientos de trabajos en operación, permanezcan dentro del marco legal y administrativo que norma la actividad de la unidad de administración y servicios.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Vigilar la debida sustentación del pago de servicios contratados por la G.D.T. y la G.A.C.D..

F U N C I O N E S .

Revisar la documentación de sustento y las cantidades a pagar por los servicios presentados a la coordinación de distribución por diversas empresas (transportistas, arrendadoras, constructoras, etc.).

Verificar con la unidad de administración de riesgos, el aseguramiento de los equipos propiedad de la Institución, así como la carga a transportar.

Participar conjuntamente con la superintendencia de sistemas de cómputo en los estudios que permitan integrar el sistema automatizado de pago a transportistas.

Llevar el control y seguimiento en las diferentes áreas externas del trámite de pago a transportistas hasta su cobro.

Participar con la gerencia de finanzas en el establecimiento de los mecanismos de pago de pago de fletes a transportistas con objeto de agilizar su trámite.

Llevar a cabo el trámite de pago y de los servicios prestados a pemex, por diversas empresas (transportistas, arrendadoras, constructoras, etc.), que se efectúan en el ámbito de operación terrestre.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

2.3. DEPARTAMENTO DE SISTEMAS DE COMPUTO.

OBJETIVO.

integrar el plan de los sistemas de cómputo para lograr la correcta operación de los sistemas de procesamiento de datos, en cuanto a su uso, almacenamiento y explotación de los mismos, así como mantenerlos actualizados con los recursos propios o externos en todas sus fases, de acuerdo al plan institucional de sistemas, para apoyar la operación de la G.D.T. y la G.A.C.D..

POLITICAS.

Procurar un uso racional del equipo de cómputo.

Pugnar para no crear archivos en discos, cuando no tengan un uso continuo.

Resguardar semanalmente en cinta magnética los archivos en disco.

Fomentar la productividad del personal de captura y codificación.

Pugnar para que se cuente con el equipo y dispositivos necesarios en materia de informática.

Vigilar la debida aplicación de los estandares de desarrollo en el diseño de sistemas computarizados.

Orientar los sistemas a desarrollar para procesos en línea.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Mantener los sistemas computarizados en operación debidamente actualizados en relación a los requerimientos de las diversas áreas.

F U N C I O N E S .

Coordinar diariamente los trabajos por procesar, haciendo los ajustes necesarios por desviaciones de los programas, trabajos de emergencia y problema de disponibilidad de equipo.

Cumplir con el programa de elaboración y entrega de reportes a las diversas dependencias de la gerencia de distribución y transporte y la G.A.C.D., de acuerdo con sus necesidades.

Realizar las actividades de desarrollo y mantenimiento de técnicas, rutinas, subrutinas, paquetes de utilería y/o investigación para la operación de los sistemas computarizados.

Elaborar estadísticas de producción, utilización de equipo y consumo de materiales.

Coordinar y supervisar la recepción, instalación, operación y mantenimiento de los equipos de cómputo y sus componentes, verificando que cumplan con los protocolos y especificaciones solicitados.

Determinar ; mediante el análisis de las necesidades de información y procesamiento de datos de la G.D.T. y la G.A.C.D., el programa de modernización y adecuación de sistemas mecanizados.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Programar, en coordinación con recursos humanos, los cursos y seminarios de actualización e inducción en materia computacional, para los integrantes de la G.D.T. y la de embarcaciones y contratos para la distribución, que necesiten conocer de estas técnicas para un mejor aprovechamiento del equipo de cómputo utilizado o en proceso de adquisición.

Mantener estrecho contacto con las dependencias de la institución, gubernamentales y privadas que puedan aportar conocimientos de cómputo, asesoría en adecuación e incorporación de nuevos equipos, paquetes de utilería, etc., aplicables a la G.D.T. y la de arrendamientos de embarcaciones y contratos para la distribución.

Desarrollar y aplicar criterios de evaluación de los sistemas de informes en desarrollo y operación, como base en el diagnóstico de funcionalidad de los sistemas de información de la G.D.T. y la G.A.C.D..

Asesorar e instruir a los usuarios en el uso, operación y óptimo aprovechamiento de los sistemas técnicos y administrativos liberados, durante su aplicación y revisión.

Participar en la elaboración de planes de desarrollo de sistemas técnicos y administrativos computarizados de acuerdo a las necesidades de la G.D.T. y la G.A.C.D. .

Analizar mancomunadamente con los usuarios las operaciones actuales y proyectadas de las dependencias, para determinar posibles mejoras en el diseño de sistemas técnicos y administrativos computarizados.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Diseñar e implantar los sistemas técnicos y administrativos en todas sus fases, como son: diseño de la estructura lógica, selección de los modelos matemáticos implicados, desarrollo de los algoritmos, programación en el lenguaje de computadora, diseño físico y pruebas necesarias.

Analizar las bases de datos necesarios para un sistema o conjunto de aplicaciones y definir las estructuras respectivas de acuerdo a las normas establecidas.

Realizar actividades de investigación para adecuar paquetes o sistemas técnicos y administrativos de aplicación estandar en las diferentes ramas de la computación en materia de ingeniería, planeación y toma de decisiones.

Supervisar la elaboración de manuales, instructivos de utilización de rutinas y/o subrutinas, paquetes de utilería; vigilando que estos manuales y/o instructivos cumplan con los estandares documentales vigentes.

Integrar la documentación de los sistemas en todas sus etapas y elaborar sus instructivos para su uso, operación y mantenimiento de cada sistema desarrollado.

Implementar y controlar el banco de información técnica y los archivos de sistemas documentados en proceso de desarrollo y en operación, manuales de operación y del usuario, inventario de equipo y bitácoras de servicio de mantenimiento.

Organizar y promover la creación de comites interdepartamentales para el desarrollo y/o actualización de estandares documentales.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

2.4. DEPARTAMENTO DE ORGANIZACION Y METODOS.

OBJETIVO .

Optimizar la estructura organizacional y el funcionamiento administrativo de la G.D.T. y la G.A.C.D., así como efectuar los estudios de métodos y procedimientos de trabajo a efecto de actualizar las herramientas necesarias que permitan el proceso de gestión interna.

POLITICAS .

Lograr que las modificaciones a las estructuras se realicen conforme a la políticas marcadas por la gerencia de desarrollo institucional.

Actualizar las estructuras de organización que tengan como finalidad la productividad de los órganos que integran la G.D.T. y la G.A.C.D..

Promover la constante actualización y modernización de los sistemas y procedimientos de trabajo en operación dentro del ámbito de la G.D.T. y la G.A.C.D..

Procurar que las acciones, la materia de simplificación administrativa se adapten e implanten adecuadamente en el ámbito de la G.D.T. y la de arrendamiento de embarcaciones y contratos para su distribución.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

F U N C I O N E S .

Estudiar y analizar permanentemente la estructura y el funcionamiento general de la G.D.T. y la G.A.C.D..

Llevar a cabo con las dependencias correspondientes la adecuación de la organización y de las funciones de la G.D.T. y la G.A.C.D..

Determinar el perfil de los puestos que requieran las estructuras organizativas de la G.D.T. y la G.A.C.D..

Elaborar y/o actualizar los manuales de organización de la G.D.T. y la G.A.C.D..

Asesorar a las dependencias que lo soliciten en la interpretación y aplicación de técnicas de análisis y desarrollo organizacional.

Promover y activar las labores de mejoramiento administrativo y proponer su divulgación.

Recopilar, seleccionar, clasificar y difundir en forma sistemática los documentos e información relacionada con el proceso de mejoramiento administrativo.

Efectuar las auditorías administrativas correspondientes y en su caso proponer las medidas que procedan.

Determinar las necesidades de innovaciones tecnológicas en materia de desarrollo organizacional.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Integrar las estructuras orgánicas y los procedimientos administrativos de la coordinación ejecutiva de distribución.

Evaluar permanentemente las estructuras de organización y el funcionamiento de la G.D.T. y la G.A.C.D., a través de diagnósticos organizacionales, de la racionalización administrativa y coordinación sectorial y regional.

Crear, coordinar y controlar los proyectos de racionalidad administrativa, diagnóstico organizacional, cambios planeados de organizaciones y productividad del trabajo, manteniendo relaciones constantes con la gerencia de desarrollo institucional y la gerencia de relaciones laborales.

Implementar los programas y proyectos de racionalización de las estructuras organizacionales y los sistemas administrativos a fin de garantizar el cumplimiento efectivo de sus cometidos.

Realizar el diagnóstico global de los sistemas administrativos de la coordinación y proponer en su caso políticas y acciones para la solución de las problemáticas o desviaciones detectadas.

Planear los objetivos, políticas, programas, procedimientos que norman las actividades de la G.D.T. y la G.A.C.D..

Capítulo II Análisis de funciones y responsabilidades del área de distribución

3. SUPERINTENDENCIA GENERAL DE INTEGRACION Y CONTROL PRESUPUESTAL.

OBJETIVO.

Programar, dirigir y controlar las acciones necesarias para impulsar la formulación y el sistema de control presupuestal de la G.D.T. y la G.A.C.D., para apoyar la toma de decisiones y para reportar el avance de su gestión; partiendo de la información presupuestal estadística y operativa que se produzca en la misma.

POLITICAS.

Promover la optimización de la información financiera presupuestal en condiciones adecuadas de calidad, confiabilidad y oportunidad.

Mantener en constante actualización adelantos técnicos que en materia de presupuestación sean susceptibles de ser adoptados por la G.D.T. y la G.A.C.D..

Mantener en constante análisis los adelantos tecnológicos que en materia computacional sean susceptibles de ser adoptados a la G.D.T. y la de arrendamiento de embarcaciones y contratos para la distribución.

Fomentar buenas relaciones con las áreas que producen información financiera presupuestal, susceptible de ser utilizada por la G.D.T. y la G.A.C.D..

Promover el intercambio de información e ideas entre las áreas pertenecientes a la G.D.T. y la G.A.C.D..

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Mantener las actividades presupuestales de las gerencias de distribución y transporte y la G.A.C.D., dentro del marco legal, técnico y administrativo que marcan las diversas entidades normativas dentro de la institución.

F U N C I O N E S .

Vigilar la actualización de la codificación única de centros de trabajo y departamentos de la G.D.T. y la G.A.C.D..

Coordinar y asesorar a los departamentos de integración y control presupuestal respecto de la formulación de los presupuestos de las dependencias de la G.D.T. y la G.A.C.D..

Analizar y determinar las modificaciones al presupuesto solicitado por las áreas de la G.D.T. y la G.A.C.D..

Difundir en forma oficial el presupuesto devengable autorizado a las diferentes áreas que conforman la G.D.T. y la G.A.C.D..

Supervisar la operación del sistema de control de gestión de la G.D.T. y la G.A.C.D..

Analizar la autorización o el rechazo de los pagos que gravan el presupuesto asignado de la coordinación ejecutiva de distribución.

Dirigir y coordinar la elaboración del presupuesto de la G.D.T. y la G.A.C.D..

Integrar la información correspondiente a los estados financieros presupuestales de la coordinación ejecutiva de distribución.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Vigilar que las modificaciones al presupuesto devengable se realicen de acuerdo a la normalidad, emitida por la secretaria de programación y presupuesto.

3.2. DEPARTAMENTO DE INTEGRACION PRESUPUESTAL.

OBJETIVO .

Lograr la consolidación presupuestal de la G.D.T. y la de arrendamiento de embarcaciones y contratos para distribución, a fin de optimizar la información del ejercicio presupuestal de los recursos financieros de la misma, a través de la planeación, aplicación y adecuación de los mecanismos que permitan la integración, registro y actualización del presupuesto mensual y anual, así como sus modificaciones y evaluaciones.

POLITICAS .

Cuidar que los presupuestos devengable y de flujo de efectivo aplicados, a las áreas de responsabilidad de la coordinación, sea los definidos en el marco del reglamento para la planeación, programación y presupuesto institucional.

Procurar que la estructura del ejercicio corresponda a la que utilice para el presupuesto autorizado.

Procurar que la función de registro desarrollada en lo interno de las áreas de la coordinación, sea apoyada mediante asesoría e información oportuna tanto en materia de normas y procedimientos, como de su ejercicio presupuestal.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Pugnar por que la programación del presupuesto mensual se realice en base a la realidad operativa.

F U N C I O N E S .

Integrar optimamente el anteproyecto del presupuesto anual de la G.D.T. y la G.A.C.D..

Regular y evaluar la información financiera presupuestal captada de las diferentes entidades normativas gubernamentales para difusión dentro de la coordinación ejecutiva de distribución.

Coordinar la información presupuestal de la G.D.T. y la de arrendamiento de embarcaciones y contratos para la distribución, con las diferentes áreas de la institución.

Clasificar e identificar el gasto en sus diferentes niveles de agregación, para programar el presupuesto mensual en base a la realidad operativa.

Colaborar conjuntamente con las áreas normativas de la información financiera presupuestal con la área interna del control del ejercicio presupuestal, en la fijación de los techos financieros correspondientes a las áreas que conforman la coordinación ejecutiva de distribución.

Supervisar conjuntamente con la área interna del control del ejercicio presupuestal la elaboración, tramite y autorización de las adecuaciones al presupuesto anual autorizado de la G.D.T. y la de arrendamiento de embarcaciones y contratos para su distribución.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Actualizar la codificación única institucional de centros de trabajo y departamentos de las dependencias que conforman la G.D.T. y la G.A.C.D..

Asesorar a los responsables de las dependencias de la G.D.T. y la G.A.C.D., respecto a la formalicen de sus presupuestos correspondientes

Solicitar la información para integrar y formular el anteproyecto del presupuesto de las dependencias de la G.D.T. y la de arrendamiento de embarcaciones y contratos para la distribución, identificando todos los niveles.

Recibir y clasificar el presupuesto devengable autorizado.

Difundir a las diferentes dependencias el presupuesto devengable autorizado en forma oficial institucional.

Efectuar adecuaciones al presupuesto de flujo de efectivo de acuerdo al comportamiento del ejercicio presupuestal.

Informar mediante cédulas analíticas de diferentes niveles de agregación el estado presupuestal mensual y anual de las dependencias que integran la coordinación ejecutiva de distribución.

Tramitar ante la subdirección de planeación y coordinación las adecuaciones al presupuesto autorizado conforme a la normatividad emitida por la secretaría de programación y presupuesto.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Integrar la información correspondiente al presupuesto de la coordinación ejecutiva de distribución.

3.3. DEPARTAMENTO DE CONTROL PRESUPUESTAL.

OBJETIVO.

Asegurar que el proceso de programación, evaluación y presupuestación de la coordinación ejecutiva de distribución, se lleve a cabo con la máxima eficiencia y efectividad, instrumentando y controlando los sistemas de registro.

POLITICAS.

Pugnar para que la solicitud del anteproyecto de presupuestos sea congruente a las necesidades de la G.D.T. y la G.A.C.D..

Pugnar al cumplimiento del presupuesto ejercido en relación al presupuesto solicitado por las áreas.

Procurar que las modificaciones al presupuesto asignado se ejecuten de acuerdo a los requerimientos operativos.

Coadyuvar y vigilar con el área de finanzas que la ministración de fondos presupuestales se realice en forma oportuna.

Mantener actualizado el sistema de control de gestión de la G.D.T. y la G.A.C.D..

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Determinar la procedencia o inprocedencia de los pagos que graven el presupuesto asignado.

F U N C I O N E S .

Coordinar a los responsables de las dependencias de las gerencias de distribución y tranportes y la de arrendamiento de embarcaciones y contratos para distribución, respecto a la formulación de sus presupuestos correspondientes.

Recabar la información para controlar el presupuesto de las dependencias de la G.D.T. y la G.A.C.D., identificando todos sus niveles.

Determinar modificaciones del presupuesto autorizado respecto de presupuesto solicitado por las áreas.

Programar de acuerdo a las áreas de la G.D.T. y la G.A.C.D., la ministración de fondos presupuestales conforme a sus programas de trabajo y realidades operativas.

Estudiar, diseñar, implantar y operar el sistema de control de gestión de la G.D.T. y la G.A.C.D., mediante el cual se conozcan los avances de un sistema de cómputo que muestre el estado financiero-presupuestal de los centros de trabajo y departamentos que devengan nuestro presupuesto.

Informar respecto a los pagos que graven el presupuesto asignado de acuerdo a la disponibilidad que muestre el sistema de cómputo, mismo que también generara indicadores de seguimiento físico financiero real.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Preestablecer estándares de comparación y análisis de incidencia y tendencia del abasto, que permitan optimizar el ejercicio presupuestal correspondiente.

Recortar mediante cédulas analíticas de diferentes niveles de agregación el estado presupuestal mensual y anual de la G.D.T. y la G.A.C.D..

Efectuar conjuntamente con la G.D.T. y la de arrendamiento de embarcaciones y contratos para la distribución, las adecuaciones al presupuesto autorizado, conforme a la normatividad emitida por la secretaría de programación y presupuesto.

Actualizar, difundir y controlar los catálogos inherentes al control presupuestal atendiendo a las necesidades de las áreas, de la codificación de las operaciones y de la sanción de la información presupuestal.

Generar y difundir en las áreas que integran la coordinación la información del ejercicio presupuestal, así como formular reportes del comportamiento del mismo para los niveles superiores.

Elaborar los reportes específicos para la toma de decisiones respecto a conceptos particulares y tendencias del ejercicio del presupuesto.

Incorporar al sistema computarizado el presupuesto anual autorizado, con el objeto de controlar su ejercicio.

Evaluar los resultados del ejercicio y de los sistemas que captan, integran generan la información presupuestal.

Capítulo II Análisis de funciones y responsabilidades del área de distribución

Integrar la información correspondiente al control presupuestal de la coordinación ejecutiva de distribución.

II.2 PROCESO DEL PAGO DE FLETES A TRANSPORTISTAS

OBJETIVO

Aplicar los lineamientos que establece el presente procedimiento administrativo para la revisión de liquidaciones de fletes por autotranque para autorización de pago, con el proposito de optimizar recursos y disminuir el tiempo de proceso.

AMBITO DE APLICACION

El presente procedimiento administrativo servirá de base para la unidad de administracion y servicios, la que a través de sus dependencias, cumplirá las disposiciones y lineamientos establecidos con respecto a la revisión de liquidaciones de fletes por autotranque para autorización de pago.

También se utilizará por todos aquellos Organos Institucionales, en el ámbito de su competencia, y para terceros que esten involucrados en la función operativa o administrativa de este procedimiento.

MARCO NORMATIVO EXTERNO

- 1) Ley Orgánica de Petróleos Mexicanos y su reglamento.
- 2) Ley Orgánica de la administracion publica federal
- 3) Ley del presupuesto, contabilidad y gasto publico federal y su reglamento.
- 4) Ley federal de responsabilidad de los servicios públicos.
- 5) Ley de adquisiciones, arrendamiento y prestacion de servicios relacionados con bienes e inmuebles.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- 6) Ley para el control, por parte del gobierno federal de los órganos descentralizados y empresas de participación estatal.**
- 7) Ley Orgánica de la secretaría de comunicaciones y transportes y su reglamento.**
- 8) Ley general de comunicaciones y transportes.**
- 9) Ley federal de las entidades paraestatales.**
- 10) Ley de impuesto al valor agregado.**
- 11) Reglamento de la ley de presupuesto contabilidad y gasto público federal.**
- 12) Reglamento de la S.C.T. para caminos y puentes federales de ingresos y servicios conexos.**
- 13) Código de comercio y su reglamento.**
- 14) Código fiscal de la federación.**
- 15) Acuerdo presidencial para la simplificación administrativa del sector público del 8 de agosto de 1984.**
- 16) Acuerdo del 25 de octubre de 1973 de la secretaría de comercio y transporte en el cual autorizó a Petróleos Mexicanos el uso de la forma 3-218 como carta de porte.**
- 17) Tarifa general de autotransporte federal de carga.**
- 18) Contrato colectivo de trabajo.**
- 19) Oficio no. 361-a-1923 de la Dirección Técnica de Ingenieros de la Secretaría de Hacienda y Crédito Público en el cual se comunica criterio aplicable en materia del impuesto al valor agregado en relación con el servicio de transporte público terrestre.**

MARCO NORMATIVO INTERNO

- 1) Reglamento para el manejo de fondos y valores.
- 2) Norma para la elaboración de procedimientos administrativos y su integración en manuales.

PROCEDIMIENTO

El Transportista (porteadores)

ACTIVIDADES.

1. Entregan en las ventanillas de recepción los documentos para iniciar los trámites para el pago de fletes que son los siguientes :

- 1.1 Conocimiento de embarque por autotanque o camion que contiene lo siguiente. (forma 3-218)

- 1) Código de barras
- 2) Centro embarcador
- 3) Fecha de elaboración
- 4) Folio de pago de fletes (a la implantación de la 3-218 c/código de barras los folios son únicos, con vigencia anual.)
- 5) Razón social o nombre del transportista.
- 6) Clave oficial no.
- 7) Placa número (de la unidad).
- 8) Clave de autotanque (registro de pemex)
- 9) Destino consignatario
- 10) Nombre del operador

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- 11) Flete cobrar o pago**
- 12) Valor**
- 13) Orden de volumen o peso (litros o kilos)**
- 14) Cantidad cargada.**
- 15) Producto o clave del mismo**
- 16) Litros o kilos brutos y/o temperatura en grados centígrados de carga.
kilos netos o litros a 20 grados**
- 17) No. de los sellos de garantía de la verificación de la pasarela.**
- 18) Sisas (nivel).**
- 19) Hora de salida**
- 20) Importe flete**
- 21) Observaciones o datos de facturación**
- 22) Remitente responsable (firma)**
- 23) Receptor responsable (firma)**
- 24) Porteador responsable (firma)**
- 25) Deducciones**
- 26) Neto a pagar**
 - Nota: los datos que aparecen al reverso del conocimiento de embarque o camión son únicamente en el punto de destino de Petróleos Mexicanos**
- 27) Identificación del lugar de descarga**
- 28) Fecha de llegada**
- 29) Horas de llegada**
- 30) Inicio de bombeo**
- 31) Terminación de bombeo**
- 32) Cantidad de producto recibido**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- 33) Temperatura del producto**
- 34) Litros cargados**
- 35) Litros recibidos**
- 36) Litros faltantes**
- 37) Productos sobrantes**
- 38) Productos faltantes**
- 39) Firma del receptor responsable**
- 40) Firma de entregado del chofer**
- 41) Sello del destino documentado justificando la descarga del producto el cual debe tener nombre del lugar geográfico, fecha de recepción y la razón social o planta donde se haya entregado el producto.**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

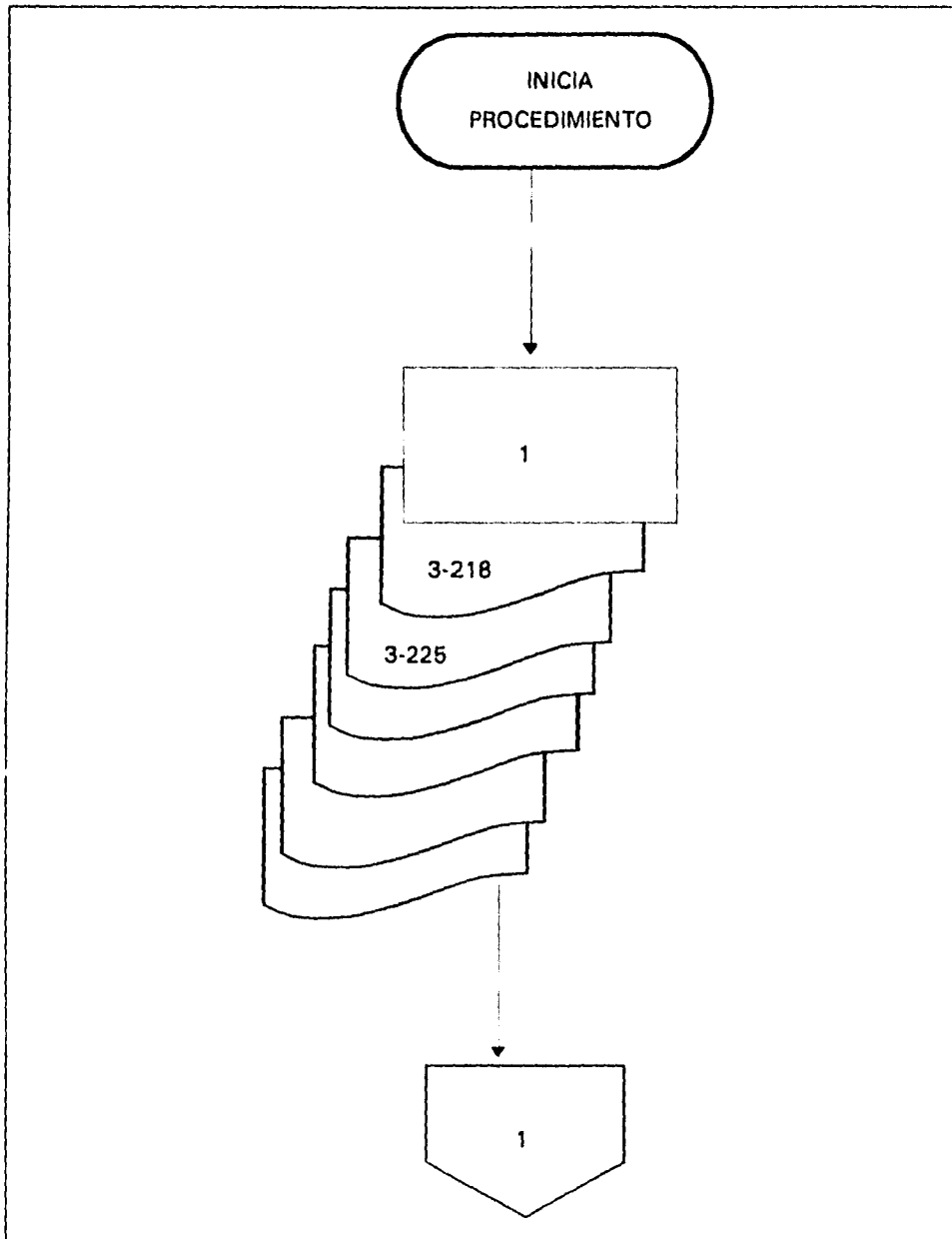


fig. II.1.2.1 Diagrama de Actividad 1.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

1.2 Liquidación de fletes de autotanques con capacidad de hasta 25 tráficos de la cual contiene los siguientes datos:(forma 3-226)

- 1) Clave porteador**
- 2) Porteador**
- 3) Domicilio**
- 4) R.F.C**
- 5) Registro I.V.A**
- 6) Registro camara de transportes**
- 7) Número de liquidación**
- 8) Fecha**
- 9) Cliente**
- 10) Origen**
- 11) Clave (origen)**
- 12) Número de folio 3-218**
- 13) Destino**
- 14) Producto**
- 15) Autotanque**
- 16) Cantidad transportada**
- 17) Cuota por litros o kilo-gramos**
- 18) Importe flete**
- 19) Cifra control (total de embarques e importes).**
- 20) Total.**

2. ¿ El transportista entrega la documentación completa y en originales ?

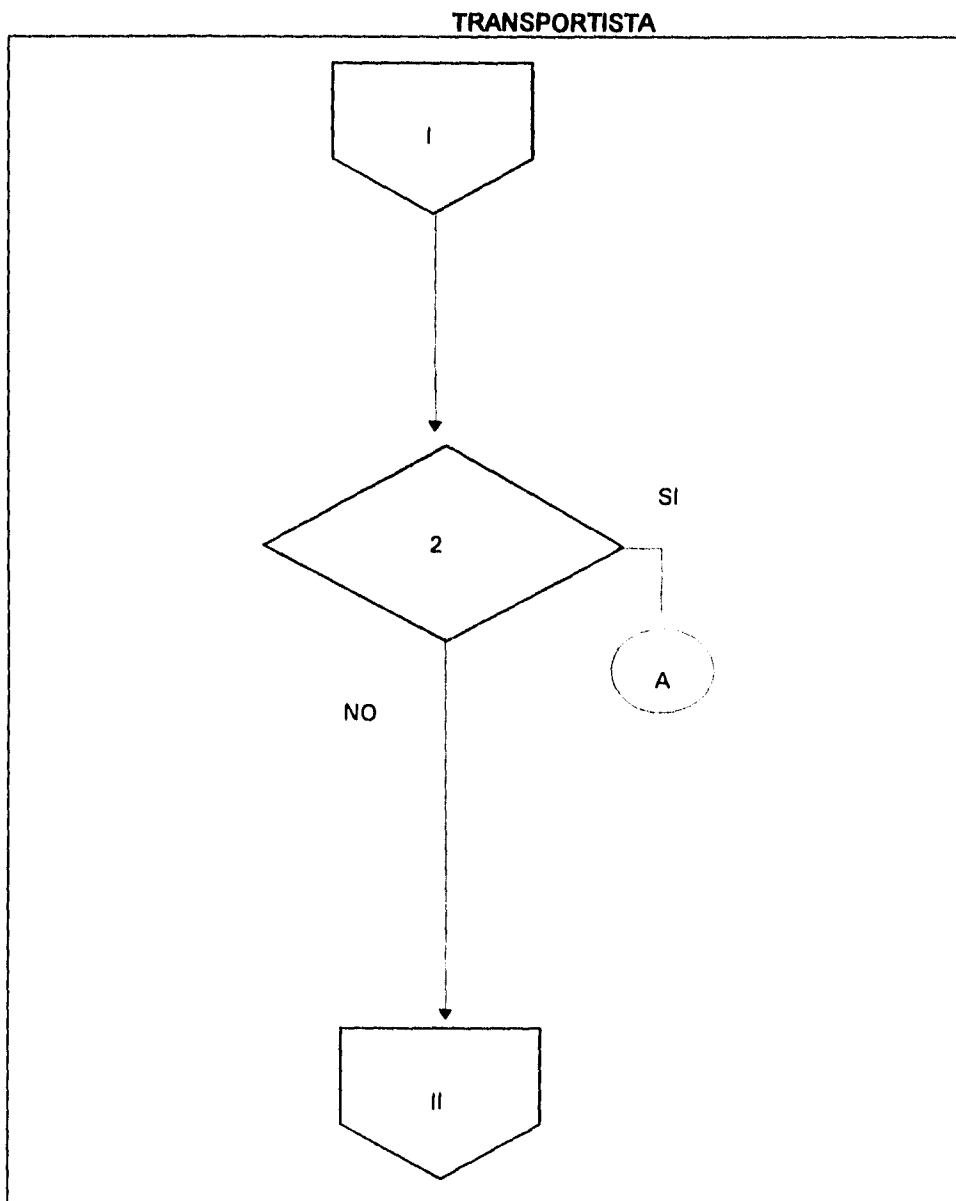


fig. II.1.2.2. Diagrama de Actividad 2

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

si pasa a la actividad no. 64

no pasa a la actividad no. 3

3. Solicita su trámite el pago de fletes con copia del original del conocimiento de embarques por autotanque o camión, para lo cual deberá tramitar y presentar en la unidad de administración y servicios, los siguientes documentos :

- a) Solicitud por escrito en la que debiera especificar las causas del extravío del documento original, haciendo mención expresa de que se hace responsable**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

del posible mal uso que se de al documento original extraviado. así mismo, deberá hacer referencia al número (folio) del conocimiento de embarque por autotanque o camión extraviado. (forma: solicitud)

b) Anexar a la solicitud la copia del conocimiento de embarque o camión la cual deberá ostentar los sellos originales, del centro receptor que avale la descarga física del producto. (forma: 3-218)

c) Oficio autorizado, tanto por el centro embarcador como del centro receptor a fin de avalar la copia del conocimiento de embarque por autotanque o camión el cual deberá especificar:(forma: oficio)

1) Nombre firma y puesto del responsable de la información que sera invariablemente la más alta autoridad del centro embarcador y del centro receptor.

2) Deberá especificar todos los datos contenidos en la copia del conocimiento de embarque o autotanque o camión del documento original extraviado.

d) Liquidación de fletes de autotanques. elaborada por el transportista, en la que indicara en número de copias del conocimiento de embarque por autotánque o camión que pretende tramitar para su cargo. (forma: 3-226)

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Nota : los sellos originales que ostentarán la copia del conocimiento de embarque o camión, así mismo el oficio que la avala, deberán ser tramitados por el transportista interesado.

Unidad de Administración y Servicios

4. Recibe los siguientes documentos :

- 1) Solicitud por escrito del transportista. (forma: solicitud)**
- 2) Copia del conocimiento de embarque por autotankue o camión el cual debera cumplir con los requisitos señalados en el inciso 3.b. (forma: 3-218)**
- 3) Liquidación de fletes por autotankues. (forma: 3-226)**

Notas: Analiza que los motivos expuestos en el escrito justifiquen el extravío del original. da instrucciones para que se elabore el formato para control de correspondencia y turna a la jefatura del departamento de documentación en documentos señalados.

Señalará prioridad para efectuar los trámites subsecuentes de acuerdo a la fecha de presentación de los documentos y a la fecha de expedición del conocimiento de embarque por autotankue o camión.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe del departamento de documentación.

5. Recibe los documentos, acusa de recibo y turna los documentos en el mismo formato de control de correspondencia, fechando foliando y turnando el escrito solicitud del porteador a la jefatura de área, registrándolos previamente en la libreta de control de correspondencia. (formas: formato).

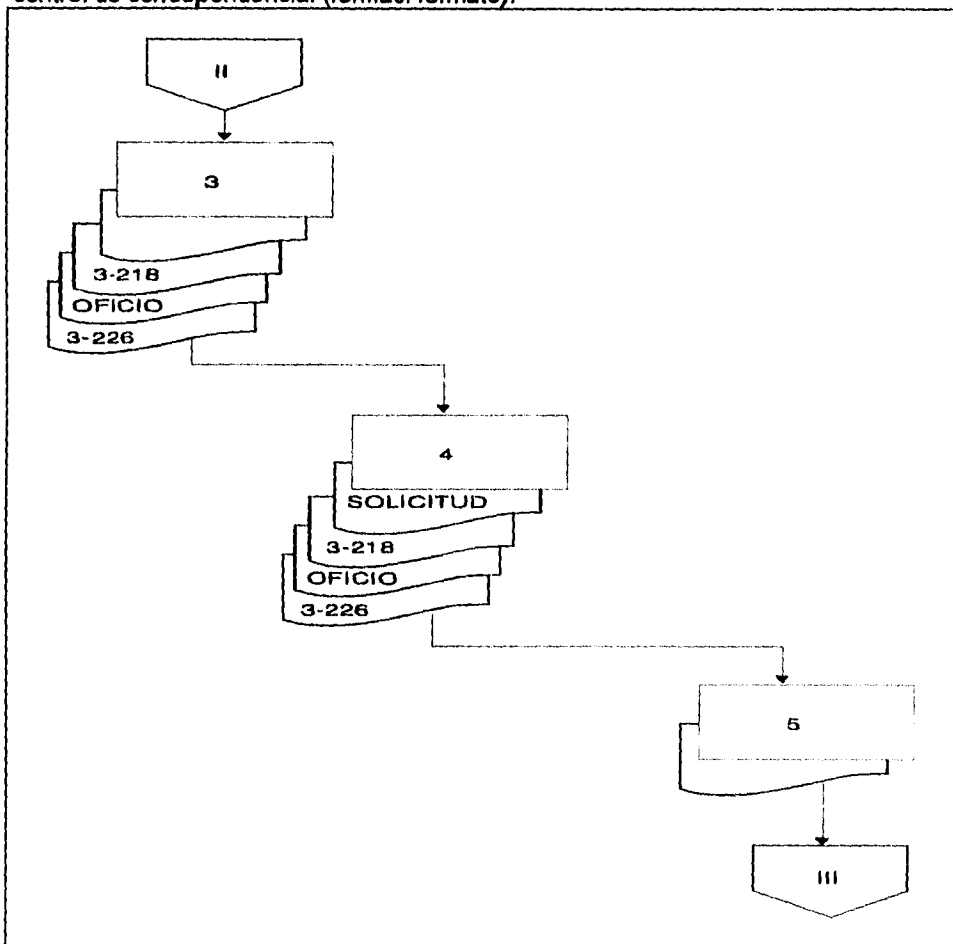


fig. II.1.2.3. Diagrama de Actividad 3, 4 y 5

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área

6. Recibe los documentos descritos en la actividad no. 5 analiza que cumplan con los requisitos señalados en la actividad no. 3 anota los documentos en su libreta de control, rubrica los documentos y los turna para su revisión a la mesa de extravíos.
(formas: lote)

Mesa de revisión de extravíos (técnico en revisión de fletes).

7. Recibe los documentos descritos, procede a abrir expediente, hace su revisión y comprueba la autenticidad de los documentos verificando los siguientes datos :
(formas: lote y expediente).

a) Que exista la solicitud por escrito del transportista para el trámite de pago con copia del conocimiento de embarque por autotank o camión.

b) Que la copia del conocimiento de embarque por autotank o camión exhiba los sellos originales del centro receptor y embarcador.

c) Que el original del oficio que avala la copia del conocimiento de embarque por autotank o camión describa todos los datos contenidos en la misma.

d) Que los documentos estén libres de tachaduras, enmendaduras, alteraciones que se preste a duda o confusión.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

e) Que cumpla con el requisito del formato de control de correspondencia expedido por la unidad de administración y servicios, (actividad no. 4)

f) Que los tráficos no se encuentren prescritos.

g) Que exista la certeza de que los documentos no se hayan presentado con anterioridad (perforados).

h) Que los documentos que hayan motivado el trámite se encuentren debidamente autorizados y contengan todos los datos necesarios para realizar las operaciones y los cálculos para realizar correctamente su pago.

8.- ¿ La documentación revisada cumple con todas las especificaciones ?

no pasa la actividad no.9

si pasa la actividad no.10

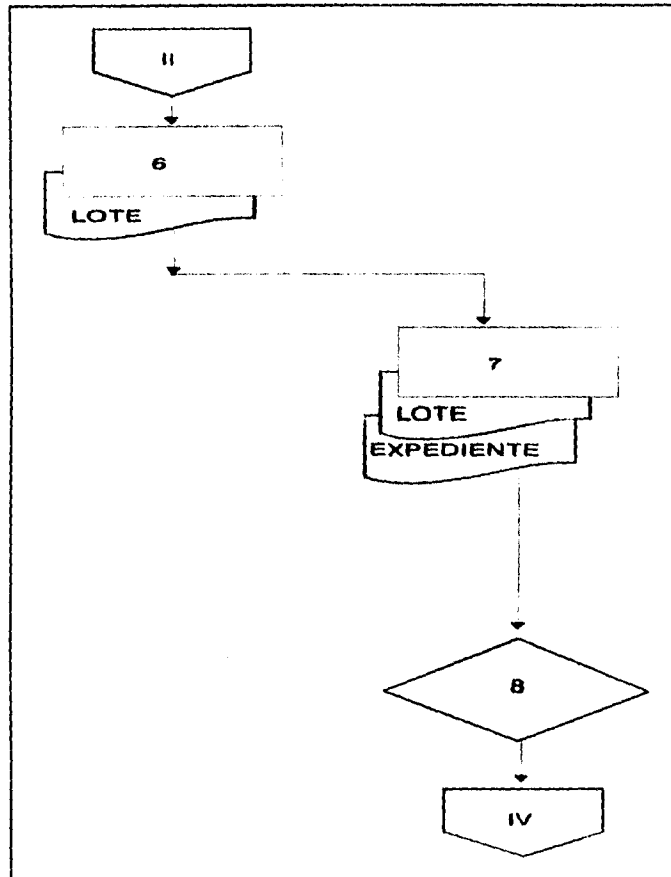


fig. II.1.2.4 Diagrama de Actividad 6 y 7

.9. Regresa la documentación, la cual registro en la libreta de control correspondencia a la jefatura de departamento para que la devuelva al transportista acompañada de un oficio firmado por el jefe de la unidad de administración y servicios. en el que se exponen las razones por las que se devuelve la documentación (se archiva copia del oficio en la unidad de administración y servicios, para cualquier aclaración).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Ventanilla de recepción.

10. Procede a foliar las liquidaciones, registra los números en la libreta de control, haciendo la observación de que los números de folio corresponden a extravíos y entrega al transportista copia de las liquidaciones foliadas en los números correspondientes.

Nota : Se establece un indicador (rango) que permita identificar el movimiento por número de reporte y el número de liquidación de fletes de autotranque,

Transportista (porteador)

11. Recoge en la ventanilla de recepción copias de las liquidaciones de fletes de autotranque con los números de folios y firma de recibido en la libreta de control, posteriormente se canjearán por los contrarrecibos.

Mesa de revisión de extravíos (técnicos en revisión de fletes).

12. Recibe las liquidaciones de fletes de autotranques con los números de folios que corresponden, y previa revisión y confirmación de que no hayan sido tramitadas con anterioridad procede a elaborar un memorandum para firma del jefe de la Unidad de administración y servicios, el que se solicita a la superintendencia gral. de documentación y sistemas que incluya en el proceso computarizado las liquidaciones elaboradas con las copias de los conocimientos de embarque o camión, turna el memorandum con todos los documentos de sustento al jefe de área.(formas: 3-226)

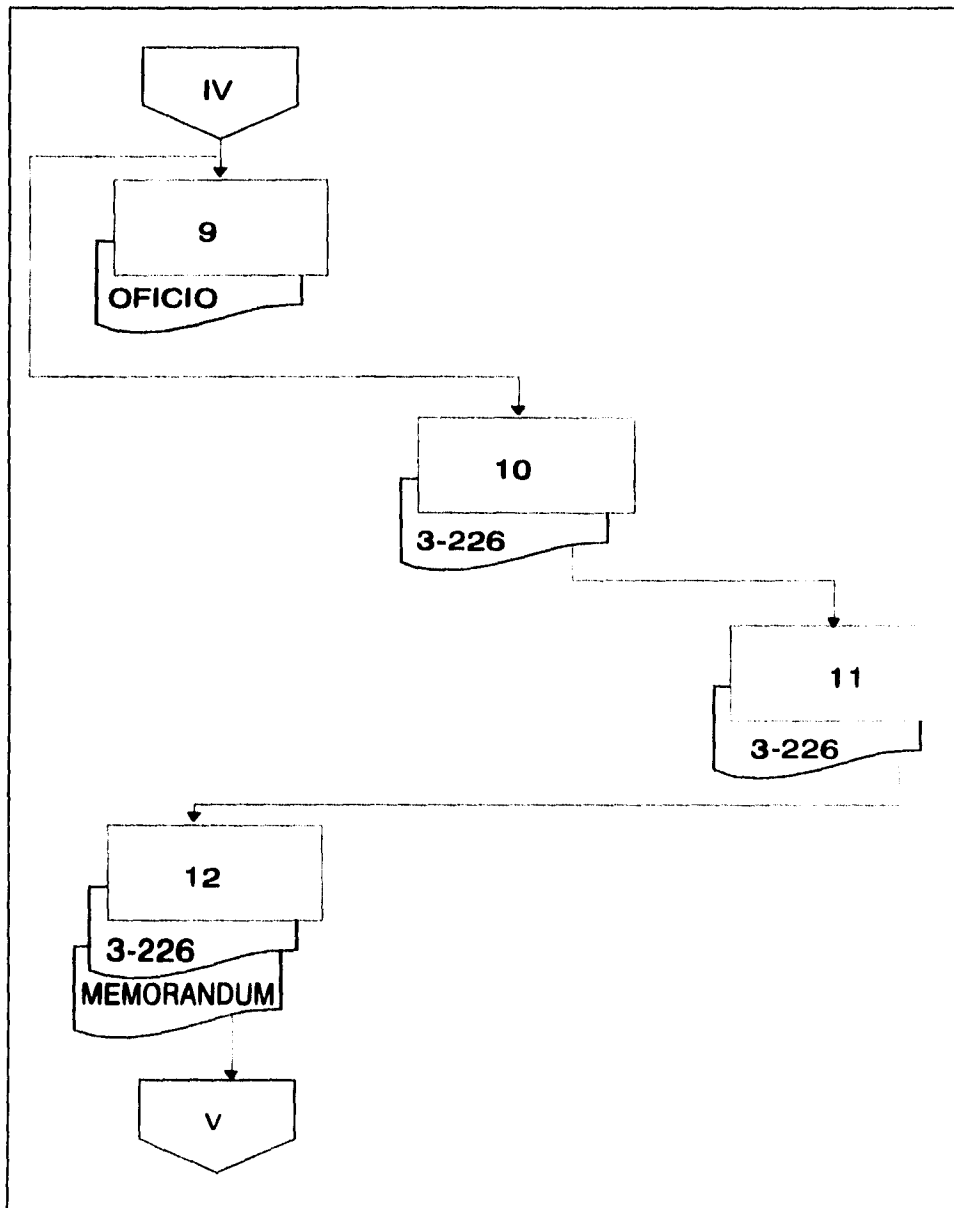


fig 11.1.2.5 Diagrama de Actividad 9, 10, 11 y 12

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área.

13. Recibe el memorandum y revisa que su contenido este debidamente sustentado con la siguiente documentación:(formas: memorandum).

- a) Solicitud por escrito del transportista. (formas: solicitud).

- b) Copia del conocimiento de embarque por autotanque o camión con sellos del centro embarcador y centro receptor. (formas: 3-218).

- c) Liquidaciones de fletes de autotanes en la que se relaciona(n) la(s) del conocimiento de embarque por autotanque o camión. (formas: 3-226)

- d) Oficio del centro embarcador y centro receptor con todos los datos de la copia del conocimiento de embarque por autotanque o camión. (formas: oficio).

- e) Formato de control de correspondencia de la unidad de administración y servicios despues de haber comprobado que la documentación esta completa, que aparece registrada en sus controles y que lleva la antefirma del técnico en revisión de fletes lo rubrica y lo turna al jefe del departamento de documentación. (formas: formato).

Jefe del depto. de documentación.

14.-Recibe el memorandum con toda la documentación que lo respalda, verifica que lleve las antefirmas correspondientes, lo rubrica y lo turna a la unidad de administración y servicios para firma. (formas: memorandum lote).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Unidad de administración y servicios.

15.- Recibe memorandum y da instrucciones para que se registre el número consecutivo del mismo en los documentos que dieron origen al trámite revisa antefirmas firma el memorandum y lo turna al jefe del depto. de documentación. (formas: memorandum).

Jefe del depto. de documentación.

16.- Recibe, verifica la firma del memorandum y lo turna a la mesa de revisión de extravíos con toda la documentación de soporte. (formas: memorandum y lote).

Mesa de revisión de extravíos (técnicos en revisión de fletes).

17.- Recibe el memorandum debidamente autorizado y procede a preparar el lote de documentos para turnarlos a la jefatura de área, anexando: (formas: memorandum).

- a) Solicitud por escrito del transportista.
- b) Copia del conocimiento de embarque por autotanque o camión con sello original.
- c) Formato de control de correspondencia.
- d) Liquidación (es) de fletes de autotanques.
- e) Memorandum solicitando se proceda a verificar la información en la suptcia. gra. de documentación y sistemas.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

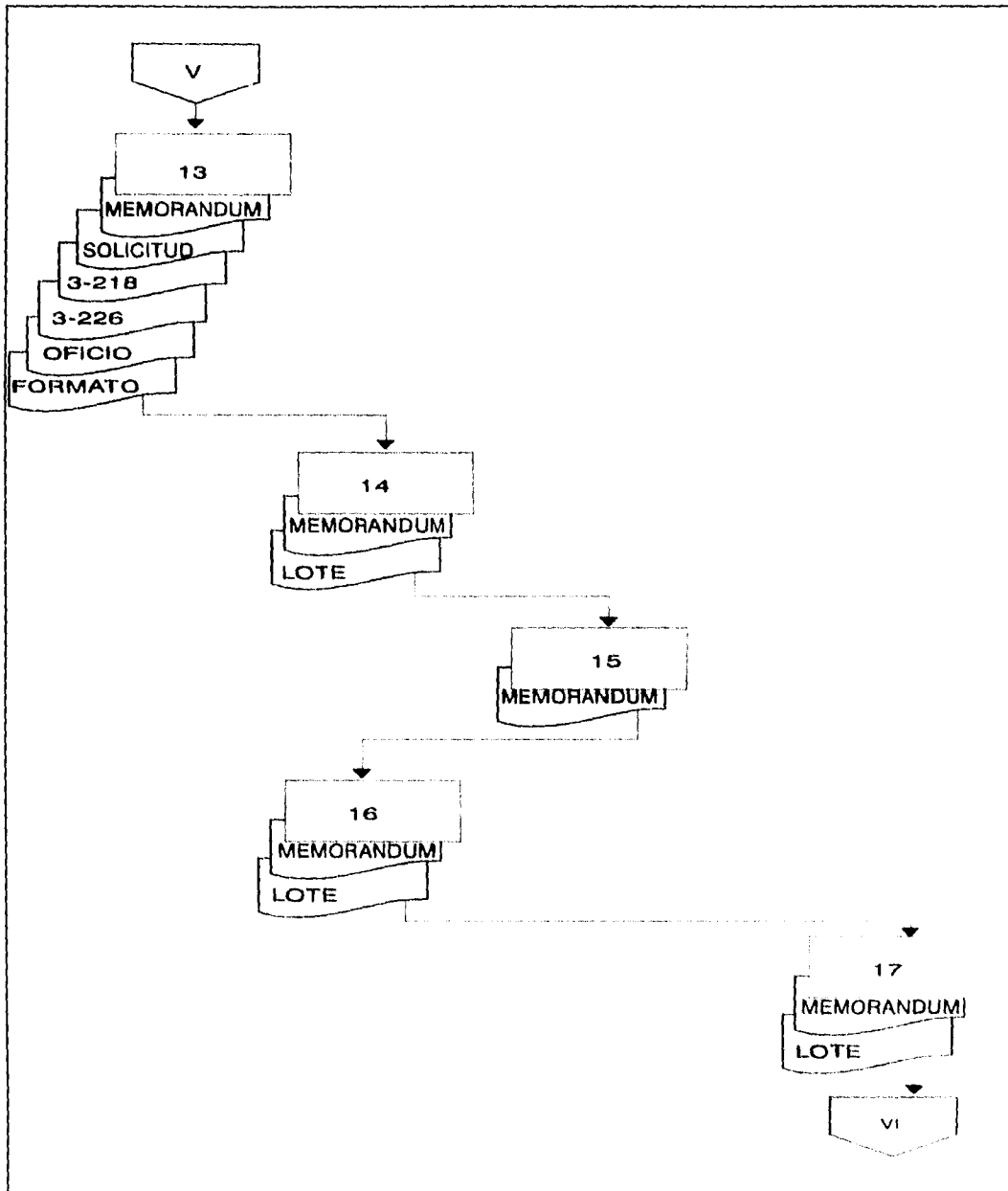


fig. II.1.2.6. Diagrama de Actividad 13, 14, 15, 16 y 17

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área.

18.- Recibe el lote de documentos procede anotar en la libreta de registro y control de lote (batch), anota el número consecutivo y el concepto de las liquidaciones; (extravíos) turna la información a la suptcia. gral. de documentación y sistemas, y envía copia de acuse de recibo a la unidad de administración y servicios. (formas: lote)

Suptcia. gral. de documentación y sistemas.

19.-Recibe y acusa de recibido en el memorandum los siguientes documentos:

- a) Copia del conocimiento de embarque. (formas: 3-218).
- b) Liquidación de fletes por autotanque. (formas: 3-226).
- c) Memorandum solicitando su integración en el sistema de procesamiento de datos. (formas: memorandum).
- e) Formato de control de lote o batch. (formas: formato).

Suptcia. gral. de documentación y sistemas.

20.- Turna al área de codificación la documentación recibida.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

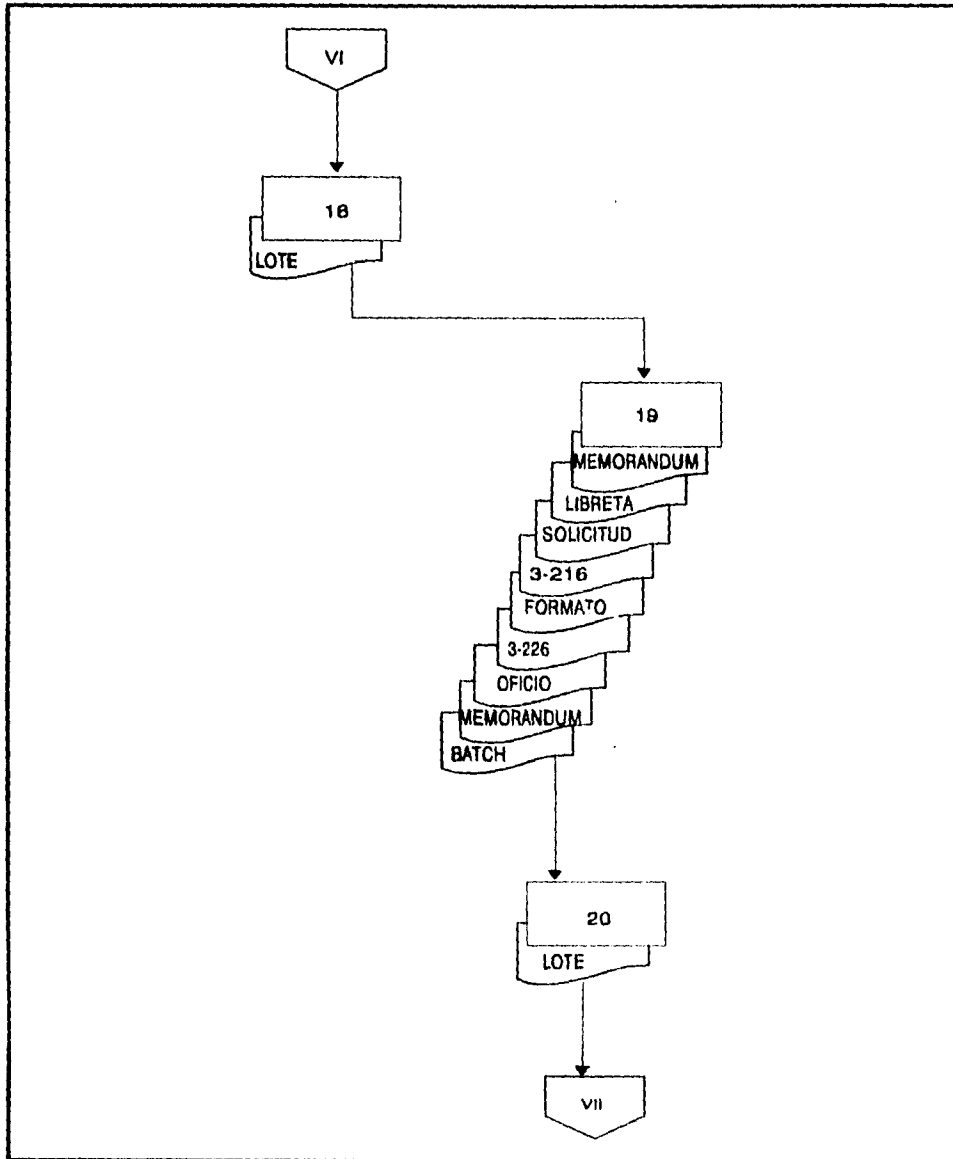


fig. II.1.2.7 Diagrama de Actividad 18, 19 y 20

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Codificador.

21.- Recoge el lote o batch de documentos y proceden a codificarlo, colocando las claves previamente especificadas en el sistema de pago a transportistas de carga liquida basandose en los siguientes catálogos: (formas: lote).

- 1) Catálogo de portadores. porteador.
- 2) Catálogo de origen destino
- 3) Catálogo de productos.

Codificador

22. Revisa que la cifra de control o número de batch este de acuerdo al número de liquidaciones y las copias de los tráficos recibidos, además verifican sellos originales de origen y destino, las reexpediciones y los faltantes.

Codificadores.

23.-Complementan el control diario de embarques (batch) llenando las columnas número de contrarrecibo, clave de transportista, cantidad de tráficos y el recuadro superior derecho anotando nombre del codificador y fecha de codificación. (formas: batch).

24.- Turna al área de captura los siguientes documentos:

- a) Control copia del registro de batch. (formas: batch).
- b) Copia del conocimiento de embarque por autotanque. (formas: 3-218).
- c) Liquidaciones de fletes por autotanque. (formas: 3-226).
- d) Memorandum solicitando su integración para su proceso (en el sistema de pago a transportistas de carga liquida).(formas: memorandum).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

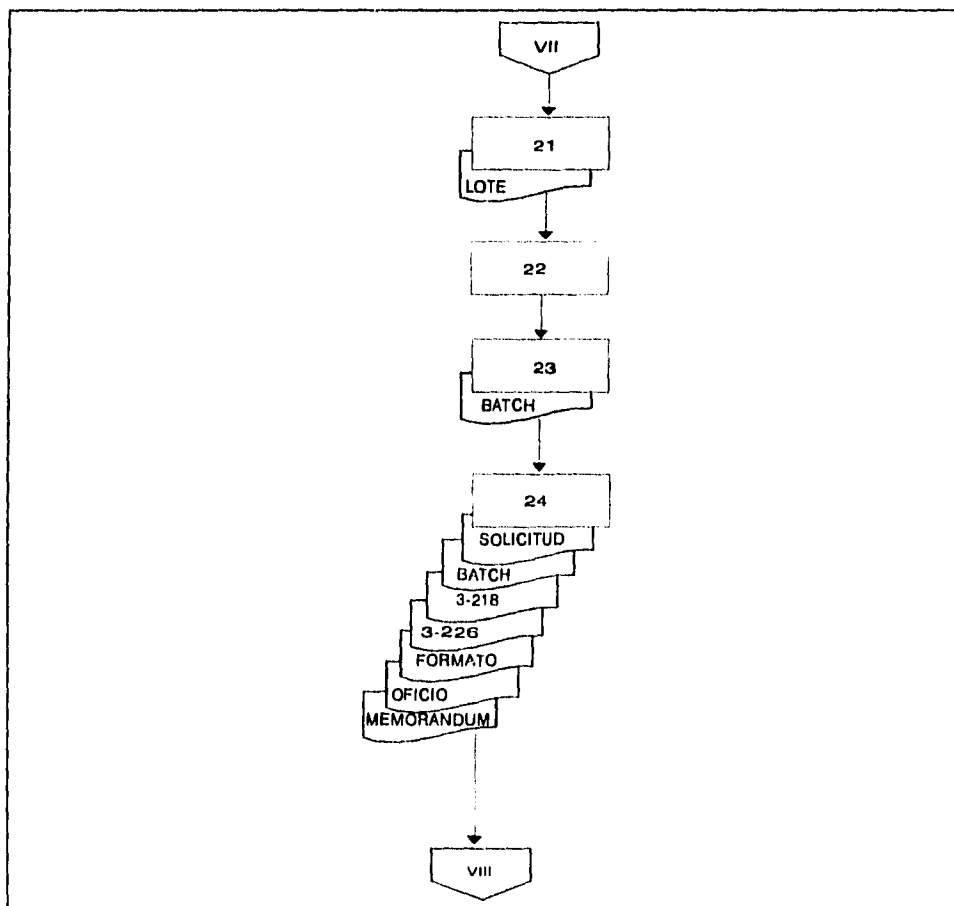


fig. II.1.2.8. Diagrama de Actividad 21, 22, 23 y 24

Area de captura

25.-Recibe el lote de documentos relacionados en la actividad anterior y procede a grabar los datos de acuerdo al formato de captura. (formas: lote).

Nota: se capturan los datos que permiten calcular el importe por flete, y generar la estadística de operación del autotanque.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Area de captura de datos.

26.- Anota en la forma para control y registro de batch el nombre o iniciales del capturista y la fecha en que fue grabado el lote. (formas: formato).

27.- Entrega al analista preparador de información los documentos referidos en la actividad no. 24, así como el número de la cinta en que fue grabado.

Analista preparador de información.

28.- Recibe documentación y efectúa lo siguiente:

Solicita a la cintoteca de la gerencia de informática institucional (g.i.i.) la cinta para transferir la información grabada en el área de captura de la superintendencia general de documentación y sistemas. con lo anterior alimenta la información al sistema de pago a transportistas de carga líquida realiza los cálculos computarizados necesarios para calcular el importe del embarque vigilando, a través del sistema informático que dichos embarques no se hayan pagado con anterioridad).

Nota: en el caso de extravíos para determinar que no se pagaron con anterioridad los tráficos durante el proceso de validación de la información se realiza una comparación con el archivo histórico de la fecha de elaboración del conocimiento de embarque por autotank y de fecha de presentación para su pago el tiempo promedio de proceso de los documentos es de cuatro días hábiles, producto del sistema es el mecanizado previo.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

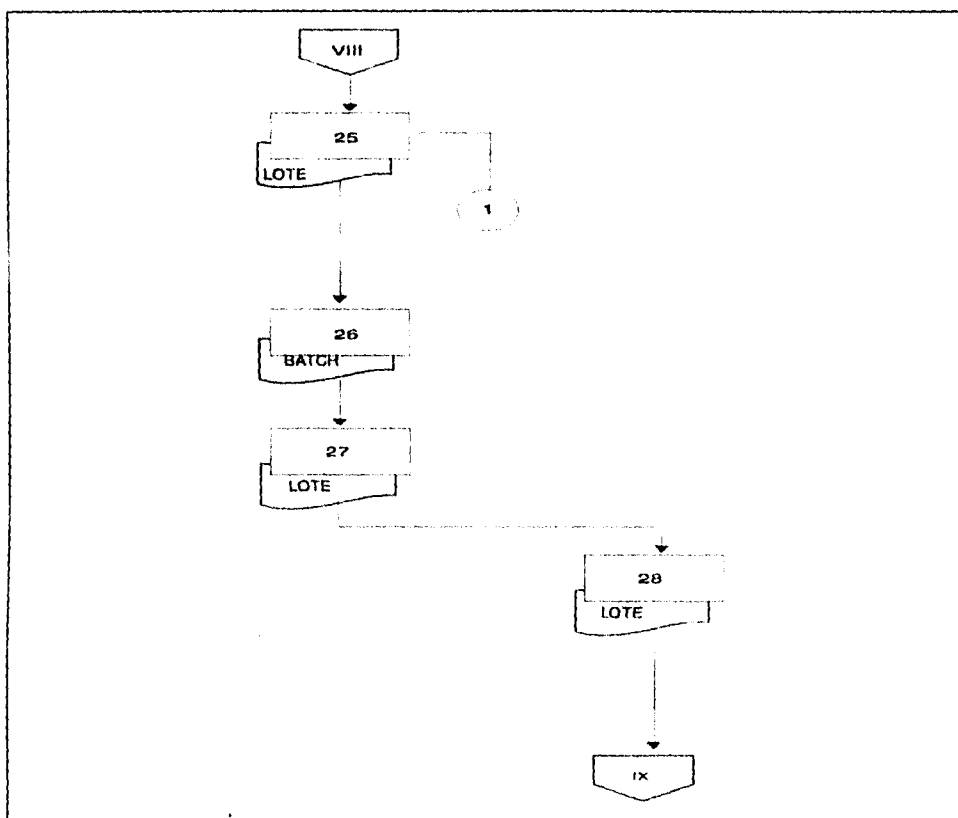


fig. II.1.2.9. Diagrama de Actividad 25, 26, 27 y 28

29.- Procesa un reporte previo donde las formas se validan y las principales características del talón de embarque como son: (formas: 3-218).

- a) Número de folio de la liquidación.
- b) Nombre de la empresa porteadora.
- c) Origen.
- d) Destino.
- e) Producto.
- f) Temperatura de carga y descarga.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

30.- Envía mecanizado previo al departamento de documentación. (formas: mecanizado).

Jefe de área.

31.- Recibe mecanizado previo y distribuye entre los técnicos en revisión de fletes encargados de los extravíos. (formas: mecanizado).

32.- recibe mecanizado y validan la información contenida en estos. (formas: mecanizado).

33.- La información contenida en el mecanizado previo.

¿es correcto?

si: pasa a la actividad no. 38.

no: pasa a la actividad no. 34.

Mesa de revisión de extravíos (técnicos en revisión de fletes).

34.- Turna al analista preparador de información para que efectuen las correcciones, marcadas en el mecanizado previo. (formas: mecanizado).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

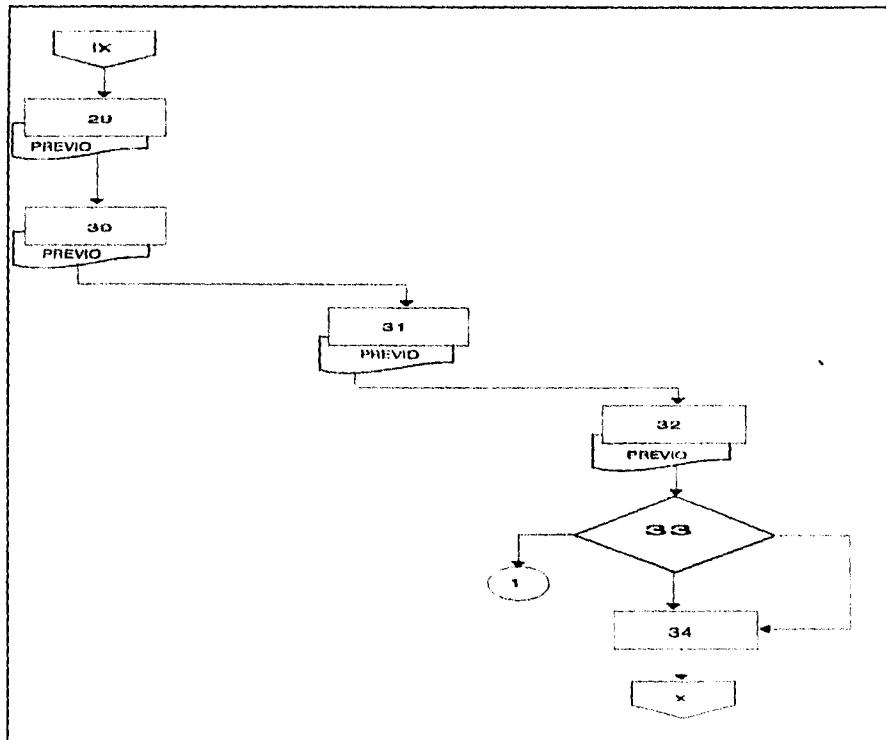


fig. II.1.2.10. Diagrama de Actividad 29, 30, 31, 32, 32, 33 y 34

Analista preparador de información

35.- Turna a los capturistas para que se efectuen las correcciones correspondientes.

Capturistas (área de captura).

36.- Realizan las correcciones y notifican al analista preparador de información al termino de esta actividad.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Analista preparador de información.

37.- Procede a emitir el reporte de validación de movimientos para pago de fletes por autotanque para turnarlo al departamento de documentación con la documentación fuente:

- a) Solicitud por escrito del transportista. (formas: solicitud).
- b) Copia del conocimiento de embarque por autotanque o camión.
(formas:3-218).
- c) Oficio de validación de la copia del conocimiento de embarque, tanto del centro embarcador como del centro receptor.
- d) Liquidación de fletes de autotanques. (formas: 3-226)
- e) Formato control de correspondencia. (formas: formato).
- f) Copia del memorandum solicitando su integración para su proceso en el sistema de pago a transportista de carga líquida. (formas: memorandum).
- g) Formato de control de lote o batch. (formas: batch).

Nota: el sistema verifica, vía un archivo magnético, que el flete no se haya pagado con anterioridad para la cual se comprueba que los datos en el talón de embarque presentado no coincidan con algun otro ya registrado, para lograr esto, se toman de la copia del conocimiento de embarque por autotanque o camión los siguientes datos:

- a) Número de folio del talón de embarque.
- b) Origen.
- c) Destino.
- d) Fecha de Embarque

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- e) Porteador.
- f) Número de autotanque.

Jefe de área

38.- Recibe el reporte de validación de movimientos para pago de fletes por autotanque con toda la documentación fuente, firma de recibido en la hoja de control de producción del departamento de sistemas y turna la información a la mesa de revisión de extravíos; anotándolo previamente en sus controles.

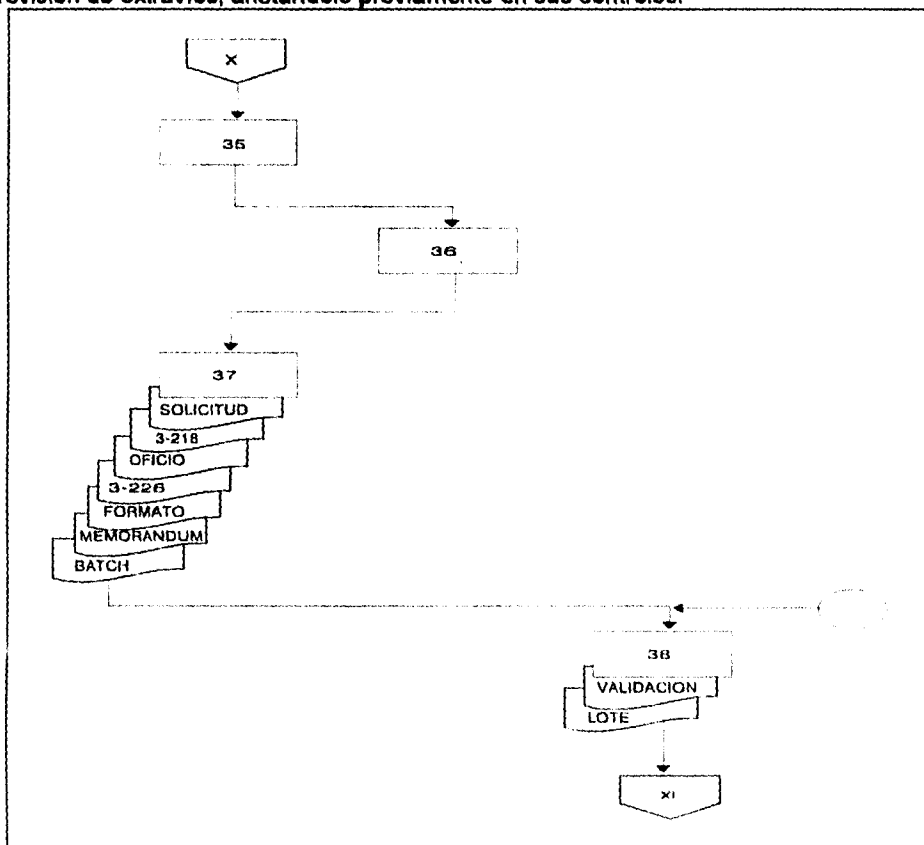


fig. II.1.2.11. Diagrama de Actividad 35, 36, 37 y 38

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Mesa de revisión de extravíos (técnicos en revisión de fletes).

39.- Recibe la documentación y procede a su revisión de acuerdo a los siguientes puntos:

- a) Por tratarse de una excepción en el sistema normal de liquidación de fletes por autotransporte, deberá comprobar que los documentos se encuentran debidamente autorizados y cumplan con los requisitos señalados en la actividad no. 7.**
- b) Comprobar que los datos que están grabados correspondan al porteador de que se trata.**
- c) Verificar que el origen y el destino sean los contenidos en la copia del conocimiento de embarque por autotanque o camión (formas: 3-218).**
- d) Verificar que el tipo de producto sea el que corresponda de acuerdo con las claves utilizadas.**
- e) Verificar que se hayan grabado los faltantes a la entrega del producto se hubieran registrado.**
- f) Verificar que las temperaturas de carga y descarga sean las que están descritas en el conocimiento de embarque por autotanque o camión.**
- g) Comprobar en el reporte las fechas de expedición. (formas: 3- 218).**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

h) No aceptar documentos en que exista duda de haber sido presentado con anterioridad y que se haya pagado (perforados).

i) Comprobar que las claves correspondientes a las tarifas que se aplican sean las autorizadas a la fecha en que se efectúe el embarque.

40 . ¿ La información contenida en el reporte de validación de movimientos este correcta?

no: pasa a la actividad no. 41

si: pasa a la actividad no. 42

41 . Realiza las observaciones para que el procesamiento de datos corrija la información, regresa a la actividad no. 25.

42 . Turna al jefe de área el reporte de validación para que se envíe al área de producción del departamento de sistemas. (formas: reporte de validación).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

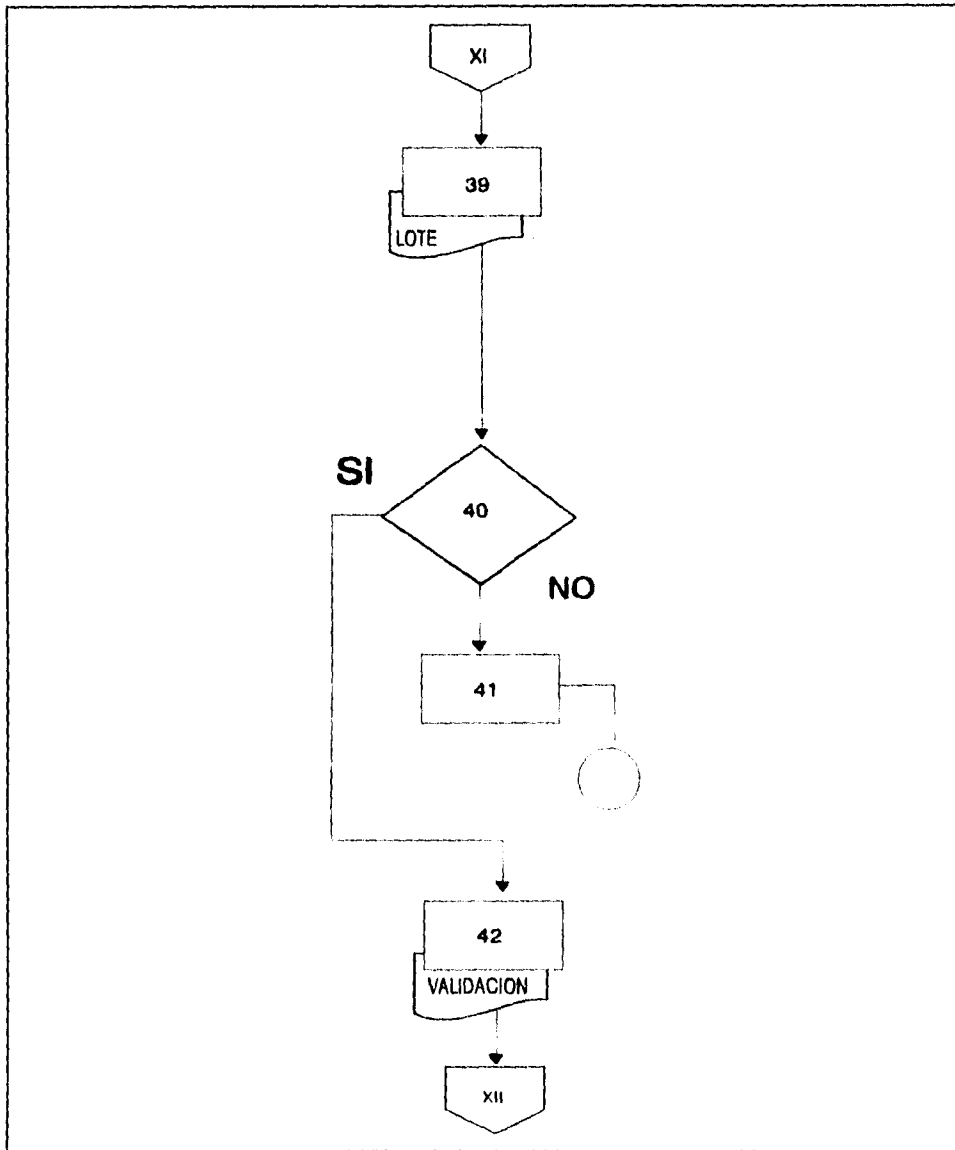


fig. II.1.2.12. Diagrama de Actividad 39, 40, 41 y 42

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área.

43 . recibe el reporte de validación y procede a llenar la solicitud de mecanizados definitivos , validación, analítico, cédula de pago y relación de contrarrecibos.

Suptcia. gral. de documentación y sistemas.

44 . Recibe el reporte de validación acusa de recibo en la solicitud del área de producción del departamento de sistemas y procede a llenar el formato de transferencia para procesar los reportes definitivos. dicho formato contiene los siguientes datos : (formas: solicitud y transferencia).

- a) Fecha definitivo.**
- b) Diagrama de proceso (número de cinta en la cual se grabo el concepto de los movimientos registrados, locales, manuales, foráneos, extravíos, reexpediciones).**
- c) Número de registros.**
- d) Número de cinta.**
- e) Número de batch por archivo.**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

45. Se emiten los reportes definitivos : (formas: reporte definitivo).

- a) Validación de documentos.
- b) Analítico de liquidaciones.
- c) Cédula de pago.
- d) Contrarrecibos.

46. Elabora formato de control de producción el sistema de pago a transportistas de carga liquidada que contiene los siguientes datos:

- a) Tipo de movimiento; local, foraneo, manual, extravíos.
- b) Fecha
- c) Recepción
- d) Observaciones.

47. Turna al jefe de área de autotanques del departamento de documentación: (formas: reporte de validación analítico de liquidaciones cédula de pago forma 3-78).

- a) Reporte de validación (original y dos copias).
- b) Cédula de pago.
- c) Contrarrecibo.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

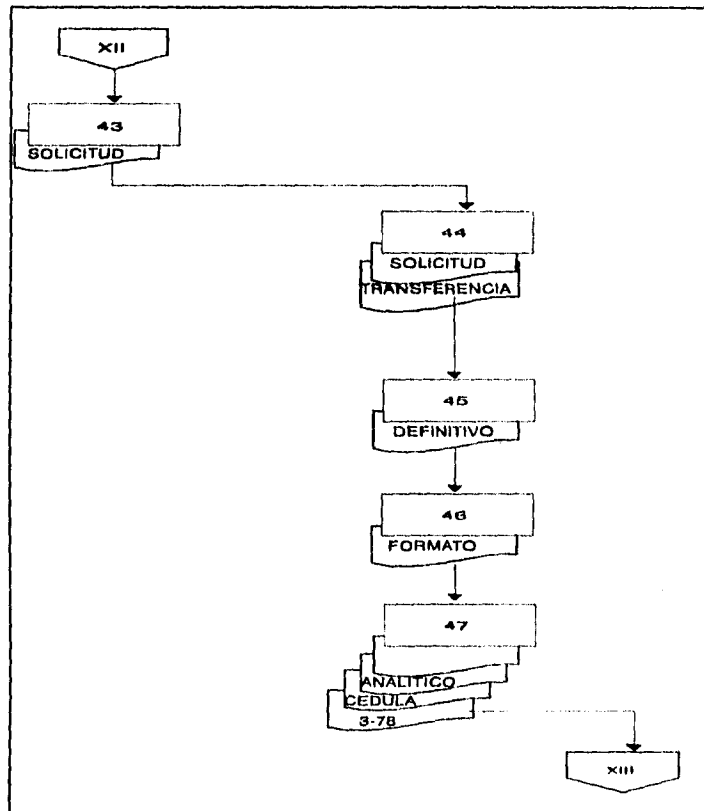


Fig. II.1.2.13. Diagrama de Actividad 43, 44, 45, 46 y 47

Jefe de área.

48. Recibe la documentación descrita en la actividad anterior, firma de recibido en la hoja de control de producción y procede a desglosar quedándose con el analítico de liquidaciones y cédula de pago, el contrarrecibo lo envía a la ventanilla de recepción y turna a la mesa de revisión de extravíos el siguiente documento: (formas: reporte de validación).

- a) Reporte de validación.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Mesa de revisión de extravíos (tecnico en revisión de fletes).

49. Recibe la documentación y procede a su verificación vigilando que cumpla con todos los requisitos para el trámite, con los puntos señalados en la actividad no. 39 y en su caso poniendo especial énfasis en las correcciones que se hubieran detectado en la revisión del reporte de validación de movimiento para pago de fletes por autotranque. (formas: lote).

50. ¿Los documentos cumplen con los requisitos señalados para la revisión?

no: pasa a la actividad no. 51

si: pasa a la actividad no. 52

51. Separa los documentos y elabora un oficio para firma de la unidad de administración y servicios exponiendo las razones por las cuales la documentación ha sido rechazada y la devuelve al transportista. (formas: oficio).

52. Conforme con la revisión procede a rubricar y sellar las liquidaciones y el reporte de validación para turnarlos al jefe del área, así mismo procede a entregar los contrarrecibos. (formas: 3-226 reporte de validación forma 3-78).

Nota: el sello debe contener lo siguiente :

a)Nombre

b)Firma

c)Fecha

d)Fecha de revisión (día, mes, año).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

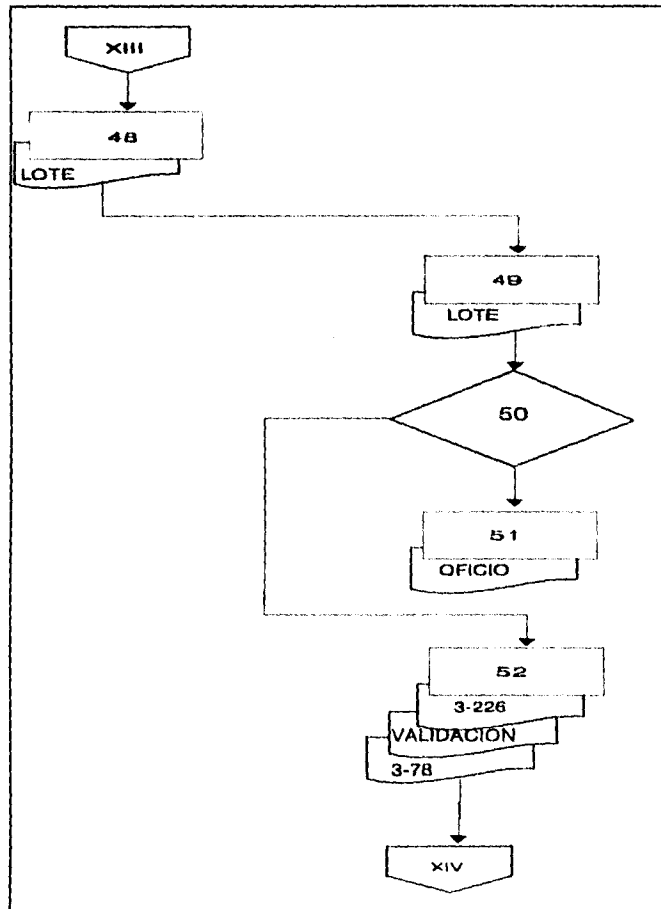


fig. II.1.2.14. Diagrama de Actividad 48,49, 50, 51 y 52

Transportista (porteador).

53. Presenta en la ventanilla de recepción la copia de la liquidación para canjearla por el contrarrecibo con el cual realizará el seguimiento de los documentos hasta recibir el cheque correspondiente en la caja general. (formas: 3-226, 3-78).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área.

54. Recibe las liquidaciones de autotanques y el reporte de validación, revisa la antefirma del técnico en revisión de fletes, el sello que avala la revisión y verifica contra la libreta de control y registro que se hayan procesado los documentos registrados, procede a firmar la cédula de pago prosigue a recabar las autorizaciones correspondientes, turnando al jefe de departamento de documentación. (formas: 3-226, reporte de validación, cédula de pago, analítico de pago).

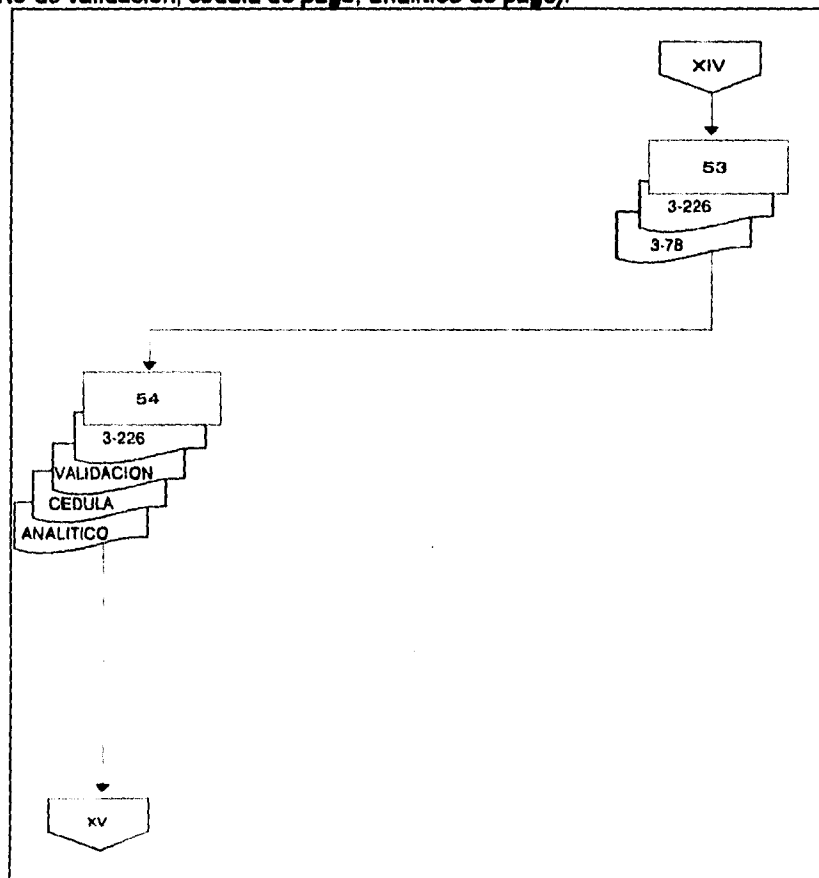


fig. II.1.2.15. Diagrama de Actividad 53 y 54

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de departamento de documentación.

55. Recibe la relación de pagos y el analítico de liquidaciones y procede a firmar dichos documentos para turnarlos al superintendente general de documentación y sistemas.

Superintendente general de documentación y sistemas.

56. recibe la relación de pagos y el analítico de liquidaciones y procede a firmar en el lugar de la correspondencia para turnarlo al jefe de área. (formas: relación de pago, analítico de pago).

Jefe de área.

57. Recibe la relación de pagos y el analítico de liquidaciones verifica la firma de suppte. gral. de documentación y sistemas y lo turna a la jefatura de la unidad de administración y servicios. (formas: relación de pago, analítico de pago).

Unidad de administración y servicios.

58. Recibe la relación de pagos y analítico de liquidaciones, verifica las firmas antes referidas y procede a firmar en el lugar que le corresponde para turnarlos a la gerencia. (formas: relación de pago, analítico de pago).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Gerencia.

59. En acuerdo con el jefe de la unidad de administración y servicios, procede a firmar las relaciones de pagos verificando las firmas precedentes y es enviado a la unidad de administración y servicios. (formas: relación de pago).

Unidad de administración y servicios.

60. Recabada la firma del gerente la turna al jefe del depto. de documentación.

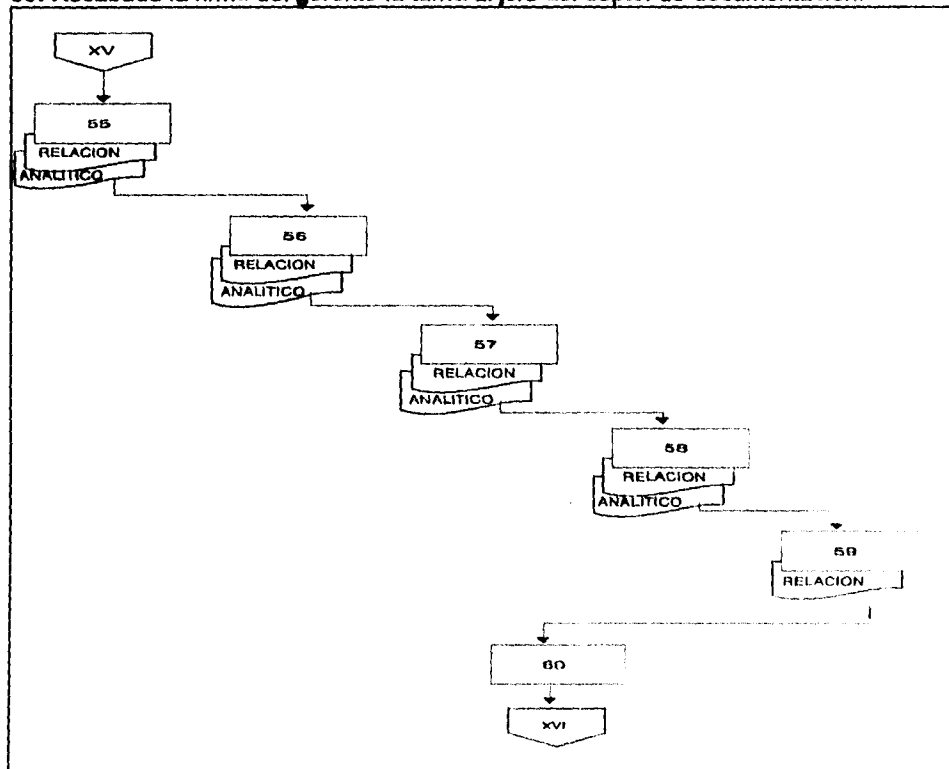


fig. II.1.2.16. Diagrama de Actividad 55, 56, 57, 58, 59 y 60

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de depto. de documentación

61. Recibe las relaciones de pago y el analítico de liquidaciones, verifica firmas y los turna al jefe de área de autotanques, junto con toda la documentación de soporte. (formas: relación de pago, analítico de pago).

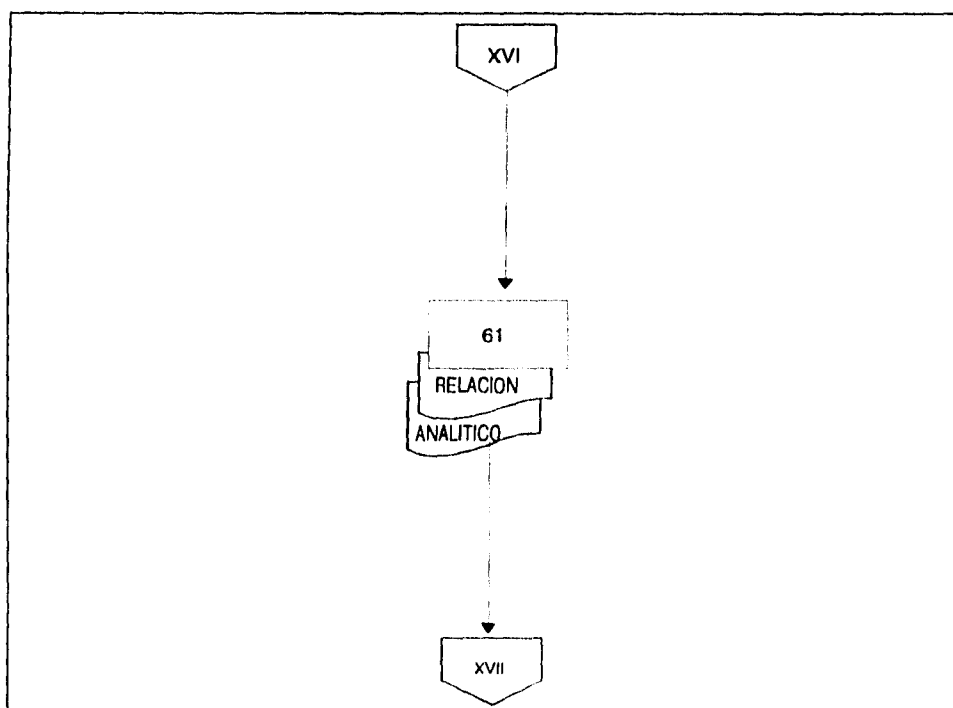


fig. II.1.2.17. Diagrama de Actividad 61

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área.

62. Recibe y envía a la superintendencia de control presupuestal cédula de resumen o relación de pago por concepto de fletes. las liquidaciones de fletes de autotanques, así como las cuentas por pagar de estas por transportista y copia del conocimiento de embarque por autotanque o camión de cada liquidación, solicitud por escrito del transportista, forma de control de correspondencia oficio que avale la copia del conocimiento de embarque por autotanque o camión tanto del centro embarcador como del centro receptor y memorandum solicitando su trámite de pago. (formas: cédula de pago, forma 3-226, cuentas por pagar, forma 3-218, solicitud, control de correspondencia, oficio, memorandum).

Nota : A partir de esta actividad la documentación descrita entra al flujo normal del procedimiento para revisión de liquidaciones por fletes por autotanque. cabe señalar que invariablemente la cédula resumen ira acompañada de la documentación que dio origen al trámite de pago y se sujetaran a la realización de las actividades que describen en la actividad no.122.

Ventanilla de recepción.

63. Recibe original de liquidaciones de fletes de autotanques, las anexa a los originales de los conocimientos de autotanque y entregan al transportista (porteador) el folio en el termino de 24:00 horas (el horario de recepción es de 8:00 a 10:30 de lunes a viernes y de 11:00 a 15:00 horas lunes y miercoles para carga regular). (formas: 3-226, 3-218).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

64. Verifica código de barras de las formas pemex 3-218, mediante lapiz lector optico.

65. registra y sella las liquidaciones en libreta control anotando:

- a) Fecha de recepción**
- b) Número de folio**
- c) Nombre del porteador**
- d) Importe**

Nota : el sello debe contener los siguientes datos del tecnico en revisión de fletes:

- a) Nombre**
- b) Firma**
- c) Ficha**
- d) Fecha de revisión**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

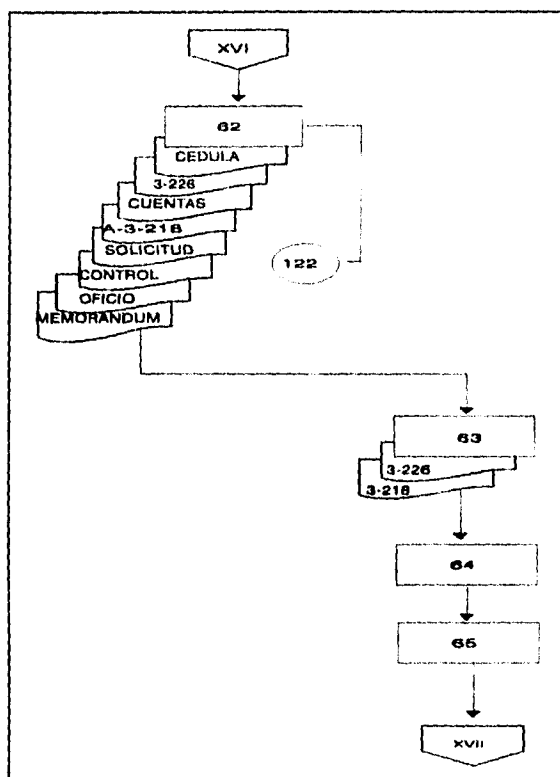


fig. II.1.2.18. Diagrama de Actividad 62, 63, 64 y 65

66. Entregan al jefe de área las liquidaciones de fletes de autotanques para que las distribuya de acuerdo a las cargas de trabajo a los técnicos en revisión de fletes.

Jefe de área.

67. Recibe las liquidaciones de fletes de autotanques y elabora la relación para batch, asignando el número de folio consecutivo correspondiente.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

68. Turna a los técnicos en revisión de fletes las relaciones para batch, acompañados de las liquidaciones de fletes de autotranques.

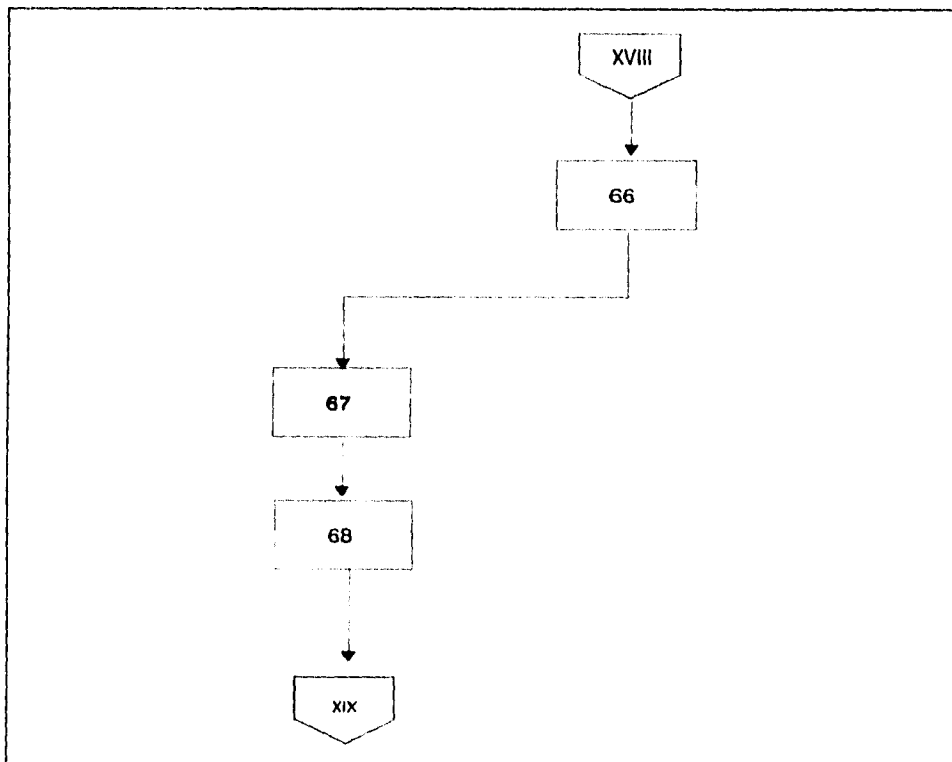


fig. II.1.2.19. Diagrama de Actividad 66, 67 y 68

Técnico en revisión de fletes.

69. Recibe la documentación referida en la actividad no. 68 la registra y procede a verificar si llena los siguientes requisitos:

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- a) Los conocimientos de embarque por autotanque o camión y las liquidaciones de fletes de autotanque deben contener los datos especificados en la actividad no. 1 (la carencia y/o alteración de cualquiera de dichos datos implica su devolución). (formas: 3-218).
- b) La cantidad de conocimientos de embarque por autotanque o camión amparados en la liquidación de fletes de autotanques debe ser precisamente la cantidad entregada de los conocimientos de embarque por autotanque o camión y que se relaciona en la liquidación de fletes por autotanques.
- c) Los conocimientos de embarque por autotanque o camión relacionados con la liquidación de fletes por autotanques deben corresponder al mismo origen, mes y porteador además coincidir con los datos adicionales que se registran en liquidación de fletes por autotanques.
- d) El sello destino, debe corresponder al destino documentado originalmente, así al consignatario que se indica, con la fecha y lugar de recepción claramente visible (salvo en los casos de expedición o cambio de destino en cuyo caso, el sello de recepción deberá permitir identificar claramente la fecha y lugar de descarga).
- e) Los documentos no deben presentar tachaduras ni enmendaduras.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

70. ¿ La documentación revisada cumple con todas las especificaciones ?

no: pasa a la actividad no. 72 y 73.

si: pasa a la actividad no. 74.

Ventanilla de entrega de documentación.

71. Devuelve al transportista los documentos (liquidaciones de fletes de autotanques con el original de los conocimientos de embarque por autotanque o camión) y registra su cancelación en la libreta de control, registrando los datos complementarios que permiten identificar las características de dicha devolución.

Transportista.

72. Recibe la documentación que no cumpla con los registros indicados en la actividad no. 69 y acusa de recibido en la libreta de control de liquidaciones.

Ventanilla de entrega de documentación.

73. Asigna número de folio a la liquidación de fletes de autotanques con el conocimiento de embarque por autotanque o camión anexo y separa una copia de la misma para ser entregada al transportista.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Nota : Anteriormente se entregaba al transportista la forma pemex 3-78 (contrarrecibo) actualmente se asigna un número consecutivo a la liquidación de fletes de autotanque (folio) el que se canjea posteriormente por el contrarrecibo (s) correspondiente (s), generado(s) por el sistema y, en la libreta de control de liquidaciones de fletes de autotanques, el porteador acusa recibido por una de las liquidaciones de fletes por autotanques con el número de folio que le son entregadas.

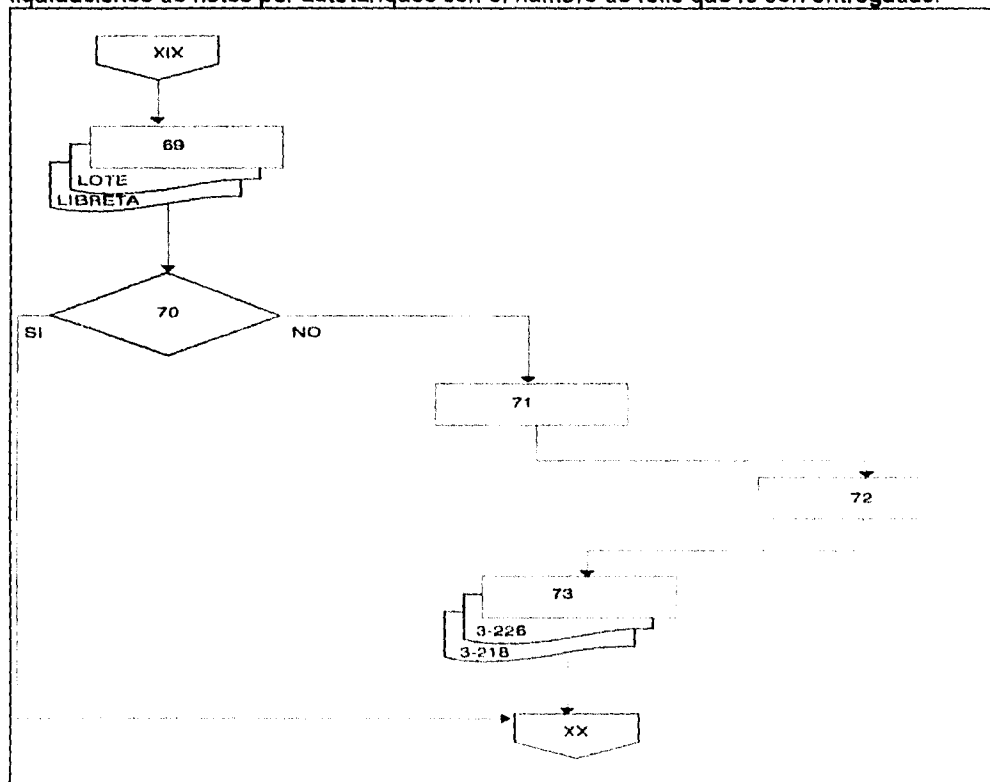


fig. II.1.2.20. Diagrama de Actividad 69, 70, 71, 72 y 73

74. Registra en la libreta de control: el número de folio, nombre del porteador, la cantidad de tráficos, clave del porteador y el importe total. (formas: libreta).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

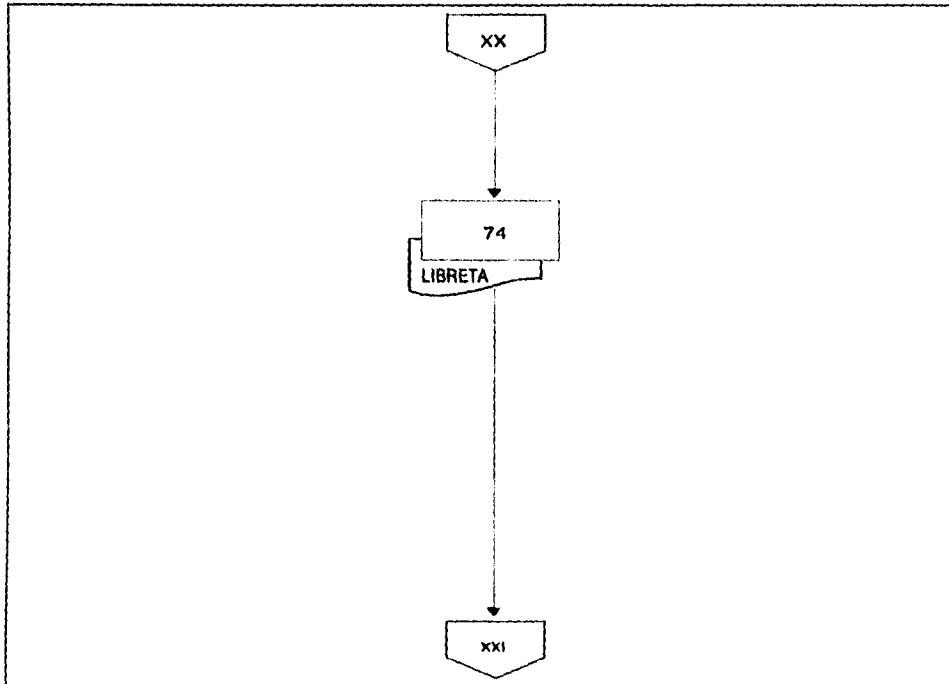


fig. II.1.2.21. Diagrama de Actividad 74

75. Elabora la forma de control y registro de batch, que tiene capacidad para 40 lotes, con los siguientes datos : (formas: batch).

- a) Número de batch
- b) Fecha
- c) Control registro no.
- d) Cantidad de conocimiento de embarques por autotanque o camión.
- e) Origen
- f) Modalidad
- g) Ventanilla
- h) Nombre del tecnico en revisión de fletes

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- i) Folio**
- j) Clave transportista.**

Ventanilla de recepción.

76. Al día hábil siguiente a la entrega de documentación para su revisión preliminar, entrega al transportista copia de la liquidación de fletes por autotanque con el número de folio. (formas: 3-226).

77. Turna al personal verificador, foleador y de registro las liquidaciones de fletes de autotanque con los números de folios para que efectúen el desglose del lote de documentos.

Personal verificador, foleador y de registro.

78. Recibe las liquidaciones de fletes por autotanque con los números de folio correspondientes y los registra en la libreta denominada control y registro de liquidaciones.

79. Regresa al jefe de área para que a su vez, este las turne a los técnicos en revisión de fletes.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

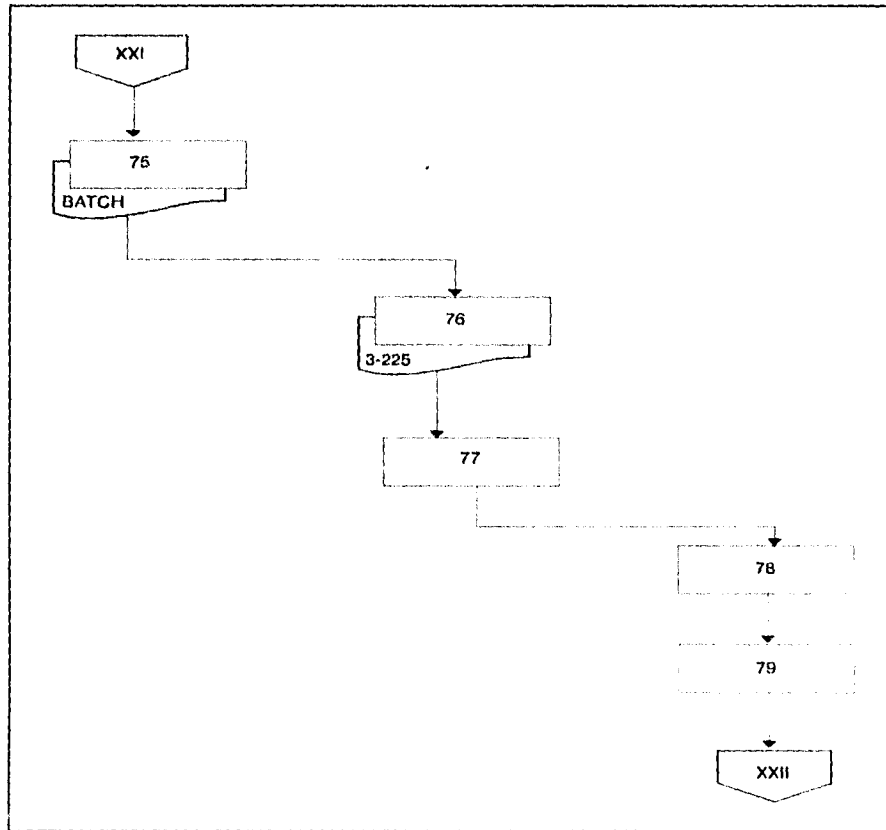


fig. II.1.2.22. Diagrama de Actividad 75, 76, 77, 78 y 79

Jefe de área.

80. Turna a los técnicos en revisión de fletes para que asignen número de lote (batch).

81. Reciben y asignan número de lote (batch) y lo anotan en la libreta de registro y control de batch agregando el número consecutivo de batch y el concepto de las liquidaciones de fletes por autotanques (cambios de destino, locales manuales y reexpediciones).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

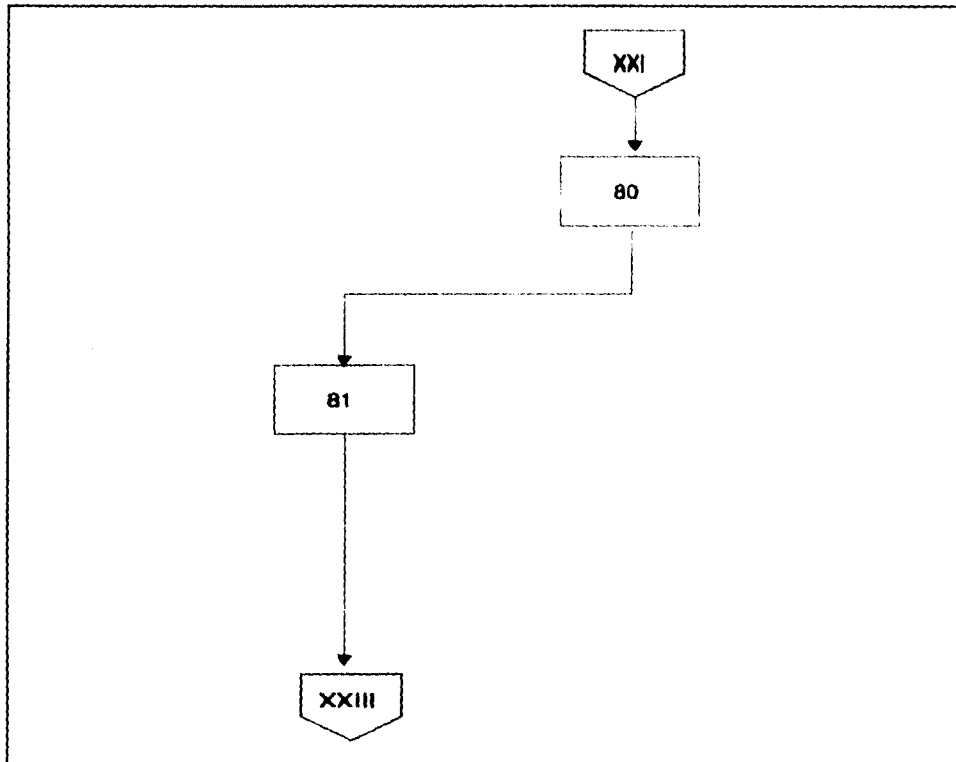


fig. II.1.2.23. Diagrama de Actividad 80 y 81

82. Regresan al jefe de área para que solicite el proceso informático.

Jefe de área.

83. Recibe y entrega la información al analista preparador de información, para su proceso computarizado.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Analista preparador de información.

84. Recibe y acusa de recibo en una copia de la forma de control y registro de batch, los siguientes documentos que integran el lote :

- a) Forma de control y registro de batch (original). (formas: batch).
- b) Liquidaciones de fletes por autotanque (original y cuatro copias. (formas:3-226).
- c) Conocimiento de embarques por autotanque (original). (formas: 3-218).

85. Coloca, en el área de codificación, dentro de un gabinete y por orden número de batch los documentos para que se proceda a su codificación.

Codificadores. (s g d s).

86. Recogen el lote (batch) de documentos y proceden a codificarlo; es decir colocan las claves correspondientes, previamente especificadas por el departamento de sistemas de la superintendencia general de documentación y sistemas, basandose en los siguientes catálogos : (formas: batch).

- 1) Catálogo de porteadores el cual contiene : (formas: catálogo de porteadores).
- 2) Catálogo de origen-destinos, clasificados en orden alfabético ascendente por origen y destino. (formas: catálogo de origen-destino).
- 3) Catálogo de productos, clasificado en orden alfabético, ascendente. 86.4 se verificará sellos de origen y destino.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

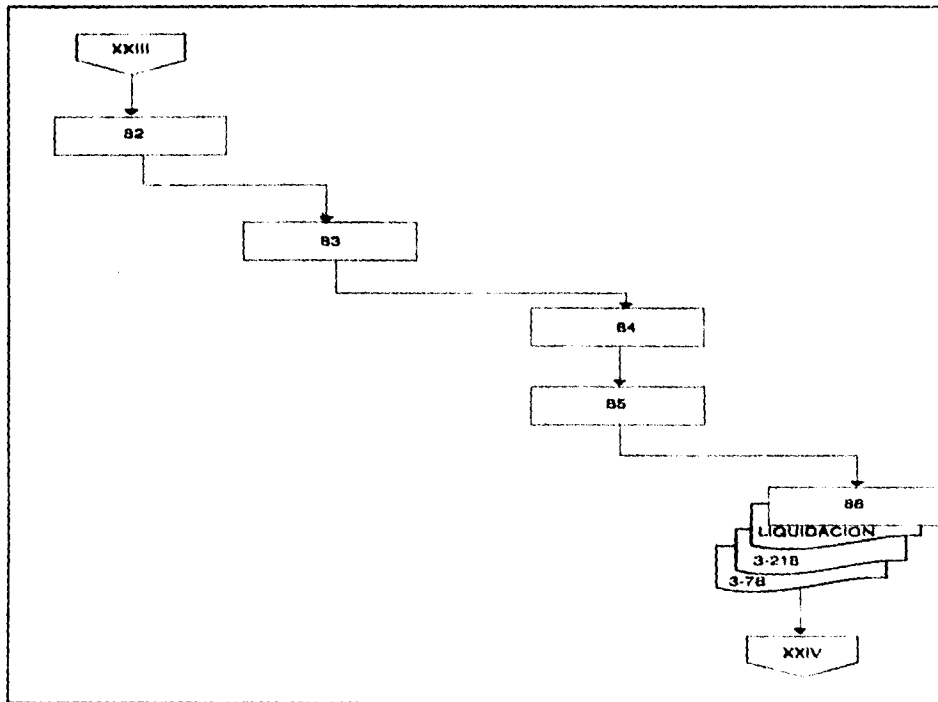


fig. II.1.2.24. Diagrama de Actividad 82, 83, 84, 85 y 86

87. Complementan el control diario de embarques (batch) llenando las columnas número de contrarrecibo, clave de transportista, cantidad de tráficos y en recuadro superior derecho anotando su nombre y fecha de codificación. (formas: batch)

88. Turna a mesa de capturistas los lotes (batch) codificados mismos que incluyen lo siguiente : (formas: batch).

- a) Control de registro de batch;
- b) Liquidaciones de fletes de autotanque. (formas: 3-226).
- c) Conocimiento de embarque por autotanque o camión (tráficos).
(formas: 3-218)

Capturistas.

89. Recibe el lote de documentos relacionados en la actividad anterior y procede a grabar los datos de acuerdo con el formato de captura (batch).

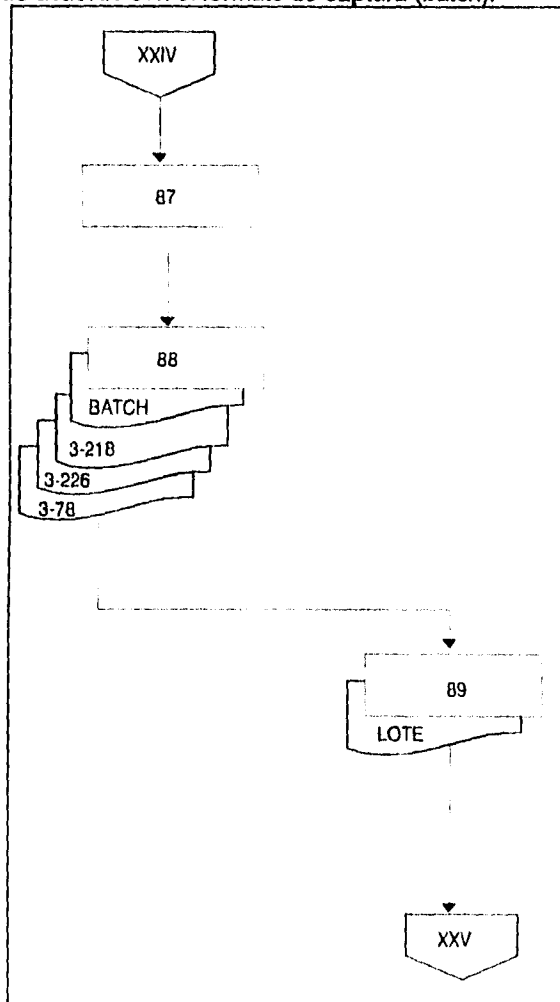


fig. II.1.2.25. Diagrama de Actividad 87, 88 y 89

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Analista preparador de información.

90. Recibe la documentación y efectúa las siguientes operaciones:

- a) Ordena en forma consecutiva los paquetes (batch) de acuerdo a la modalidad que se trate (locales o foraneos).**
- b) Selecciona una cinta disponible, con base a la rotación de las mismas.**
- c) Solicita a la cintoteca g.i.i. le sea proporcionada la cinta seleccionada para trabajar con ella.**
- d) Solicita la transferencia de los batch capturados del disco del equipo de captura a la cinta seleccionada y que previamente clasifiquen en forma consecutiva de acuerdo a su modalidad (local y/o foraneo).**
- e) Reintegra la cinta a la cintoteca de g.i.i. a fin de que pueda ser utilizada por el sistema cuando este lo solicite.**

91. Procede a elaborar un mecanizado previo para detectar posibles errores de codificación y/o captura de datos, efectuando las correcciones identificadas, tanto de codificación como de captura, a fin de depurar, en lo posible, la información a procesar, repite la actividad número 90 c).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

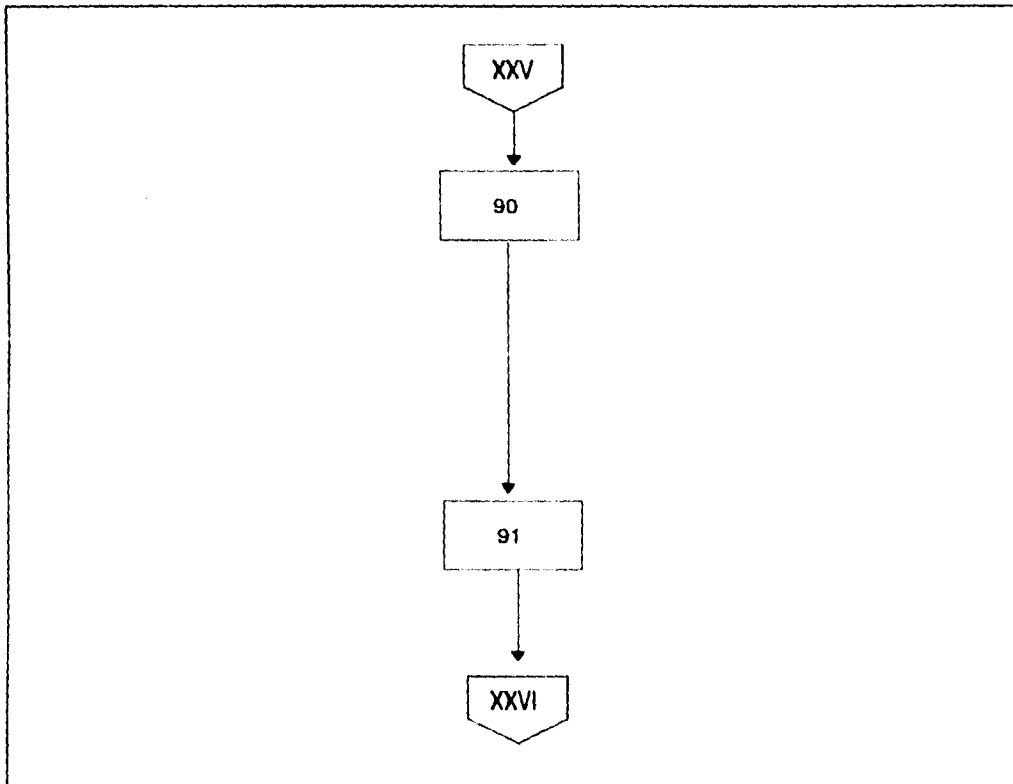


fig. II.1.2.26. Diagrama de Actividad 90 y 91

92. Envía previo al departamento de documentación (jefe de área).

Depto. de documentación (jefe de área).

93. Recibe previo y los distribuye entre sus técnicos en revisión de fletes para su validación con la documentación fuente entregada por el transportista.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Técnico en revisión de fletes.

94. Valida y revisa los mecanizados previos determinando si es correcto o no.

si es correcto pasa a la actividad no. 95.

no es correcto pasa a la actividad no. 96.

Jefe de área.

95. Solicita proceso definitivo.

96. Solicita se hagan las correcciones de codificación y/o captura marcadas en los mecanizados.

97. Solicita se corrija la información según los errores detectados por los técnicos en revisión de fletes repitiendo la actividad no. 90 c). una vez modificada la información.

98. Solicita al analista preparador de información que expida los contrarrecibos y las cuentas por pagar.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Analista preparador de información.

99. Recibe solicitud mediante la forma "solicitud de pago".

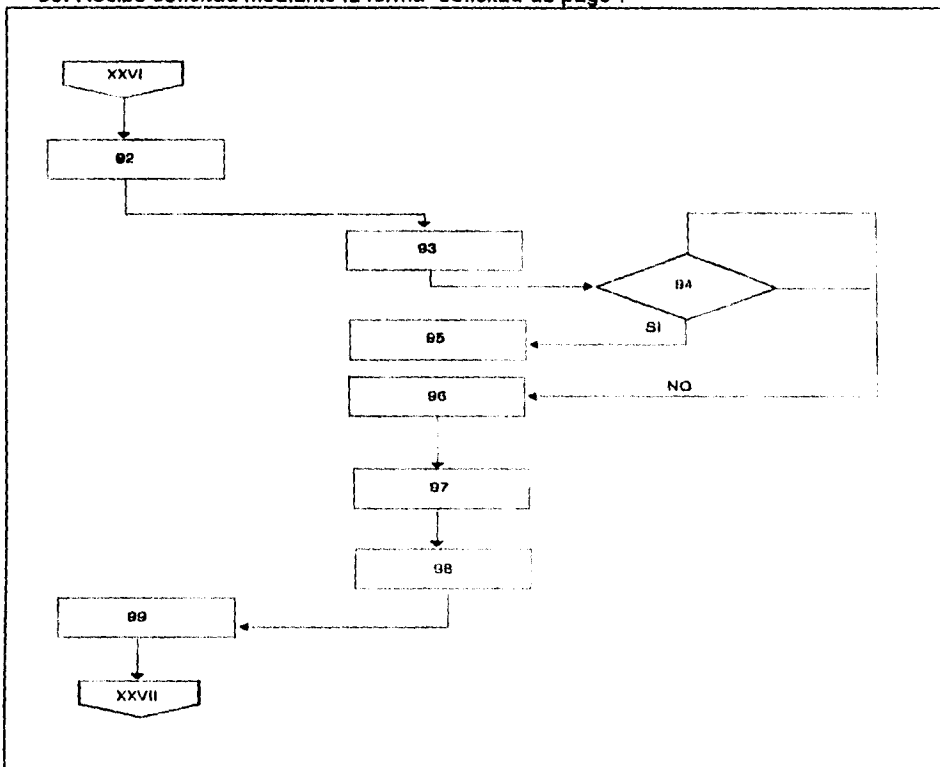


fig. II.1.2.27. Diagrama de Actividad 92, 93, 94, 95, 96, 97, 98 y 99

100. Otorga parámetro para el procesamiento dando entrada al número de cuenta por pagar, fecha, clave de autorización, cinta de movimientos, cinta acumulada, saldo anterior, obteniendo:

- a) Cuentas por pagar**
- b) Cifras control**

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

101. Verifica cifra control de cuentas por pagar y manda la información de archivo al histórico de cuentas por pagar, cotejando que lo incluido este correcto y envía la información de detalle para su inclusión en el archivo de cheques. creando un archivo pendiente de emisión de cheques.

102. Produce las cuentas por pagar y los contrarrecibos, turnandolos al jefe de área del departamento de documentación.

103. Procesa definitivo, produciendo :

- a) Mecanizado de validación**
- b) Cédula de pago**
- c) Analítico de pagos**
- d) Contra-recibos**

104. Genera, además un archivo para utilizarse en la emisión de cheques.

105. Separa de los mecanizados las hojas con cifras control y los archiva.

106. procede a hacer la entrega al jefe de área del departamento de documentación de los productos mencionados en la actividad no. 103.

Jefe de área.

107. Recibe y distribuye mecanizados definitivos entre los técnicos en revisión de fletes para que efectuen una nueva revisión y determinen si se realizaron las correcciones marcadas en el previo, así como si no existieran modificaciones o errores adicionales que justifiquen un reproceso.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Técnicos en revisión de fletes.

108. Reciben y revisan nuevamente el definitivo y comprueban si efectuaron las correcciones solicitadas en el previo y la inexistencia de modificaciones o errores adicionales.

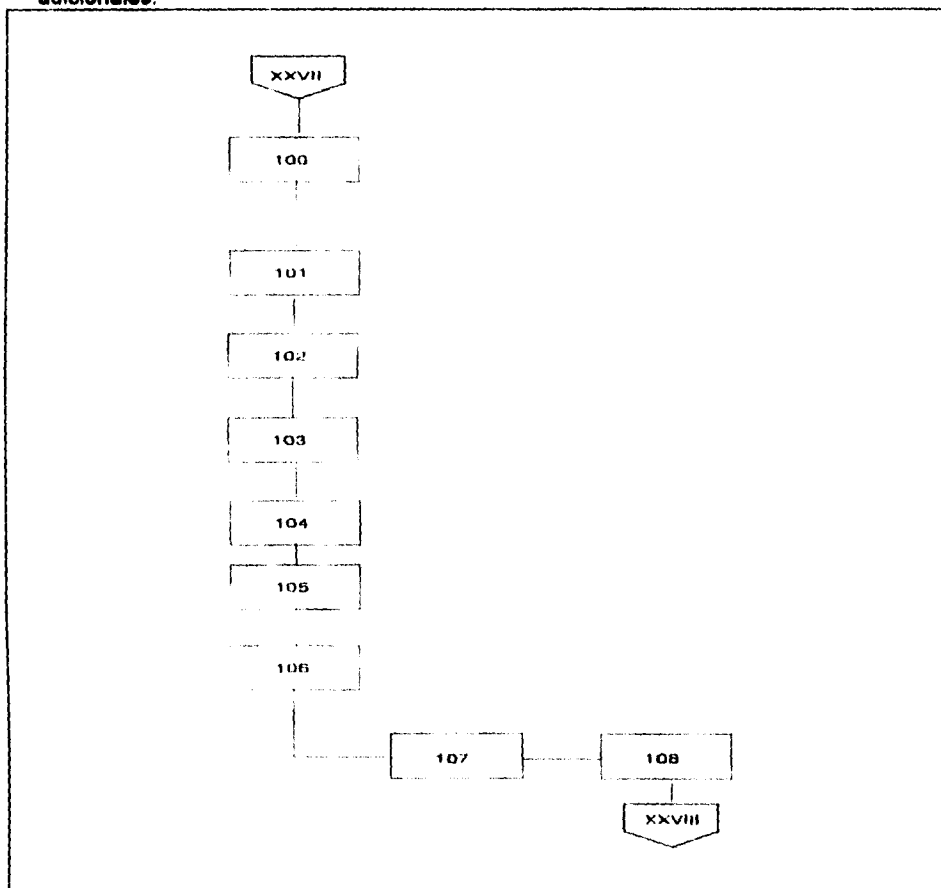


fig.II.1.2.28. Diagrama de Actividad 100, 101, 102, 103, 104, 105, 106, 107 y 108

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

109. Avisan al jefe de área de que el definitivo recibido puede seguir su trámite.

Jefe de área.

110. Recibe las cuentas por pagar y los contrarrecibos e indica la integración de las cédulas de pago a las cuentas por pagar turna a las ventanillas de recepción los contrarrecibos para que sean canjeados por los folios que les entregaron a los transportistas (porteadores).

Ventanilla de recepción.

111. Reciben los contra-recibos correspondientes para canjearlos a los porteadores mediante la presentación de los folios que les entregaron.

Transportistas (porteadores).

112. Presentan y entregan en las ventanillas de recepción los folios que le fueron proporcionados recibiendo a cambio los contra-recibos correspondientes.

Jefe de área.

113. entrega los mecanizados definitivos debidamente rubricados por los técnicos en revisión de fletes como por el mismo al jefe del departamento de documentación para que haga lo conducente además de turnar a la superintendencia de control presupuestal los documentos que se mencionan a continuación:

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- a) Cuentas por pagar
- b) Analítico de liquidaciones
- c) Liquidaciones de fletes de autotanque
- d) Conocimientos de embarque por autotanque.

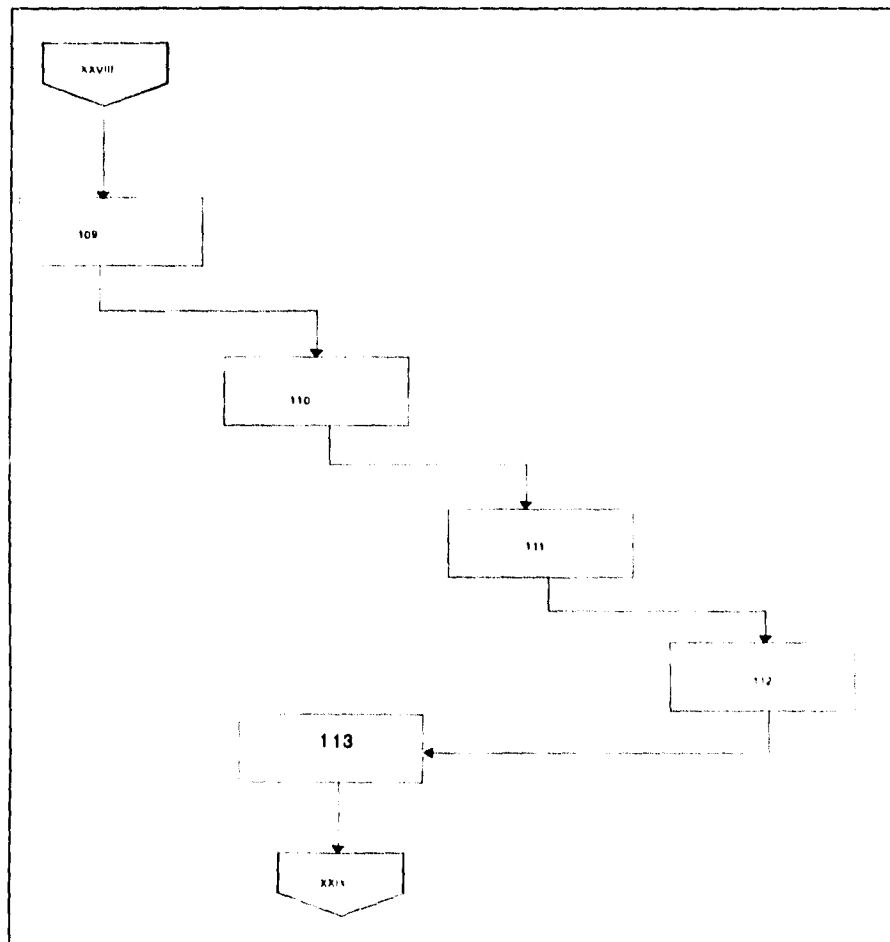


fig. II.1.2.29. Diagrama de Actividad 109, 110, 111, 112 y 113

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe del departamento de documentación.

114. Recibe y revisa que los documentos cumplan con los requisitos señalados, así como que contengan las firmas de los responsables de su revisión y la del jefe de área, seguidamente rubrica los documentos los turna al superintendente general de documentación y sistemas.

Superintendente general de documentación y sistemas.

115. Recibe y verifica las firmas en la relación de pagos y el analítico de liquidaciones y firmas en el lugar que le corresponde y lo turna al jefe del departamento de documentación.

Jefe del departamento de documentación.

116. Recibe la relación de pagos por concepto de fletes de autotanques y el analítico de liquidaciones, verifica la firma del superintendente general de documentación y sistemas y lo turna a la unidad de administración y servicios.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Unidad de administración y servicios.

117. Recibe y revisa la relación de pagos por concepto de fletes de autotanque y el analítico de liquidaciones que contengan las firmas referidas y firma en el lugar que corresponde y procede a turnar a la gerencia de distribución y transporte.

Gerencia de distribución y transportes.

118. Recibe y verifica que las relaciones de pago por concepto de fletes y los analíticos de liquidaciones estén debidamente firmados por el jefe del departamento de documentación, el superintendente general de documentación y sistemas, el jefe de la unidad de administración y servicios, y los turna a la unidad de administración y servicios.

Unidad de administración y servicios.

119. Recibe y turna al jefe del departamento de documentación las relaciones de pago y analítico de liquidaciones debidamente requisitadas.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

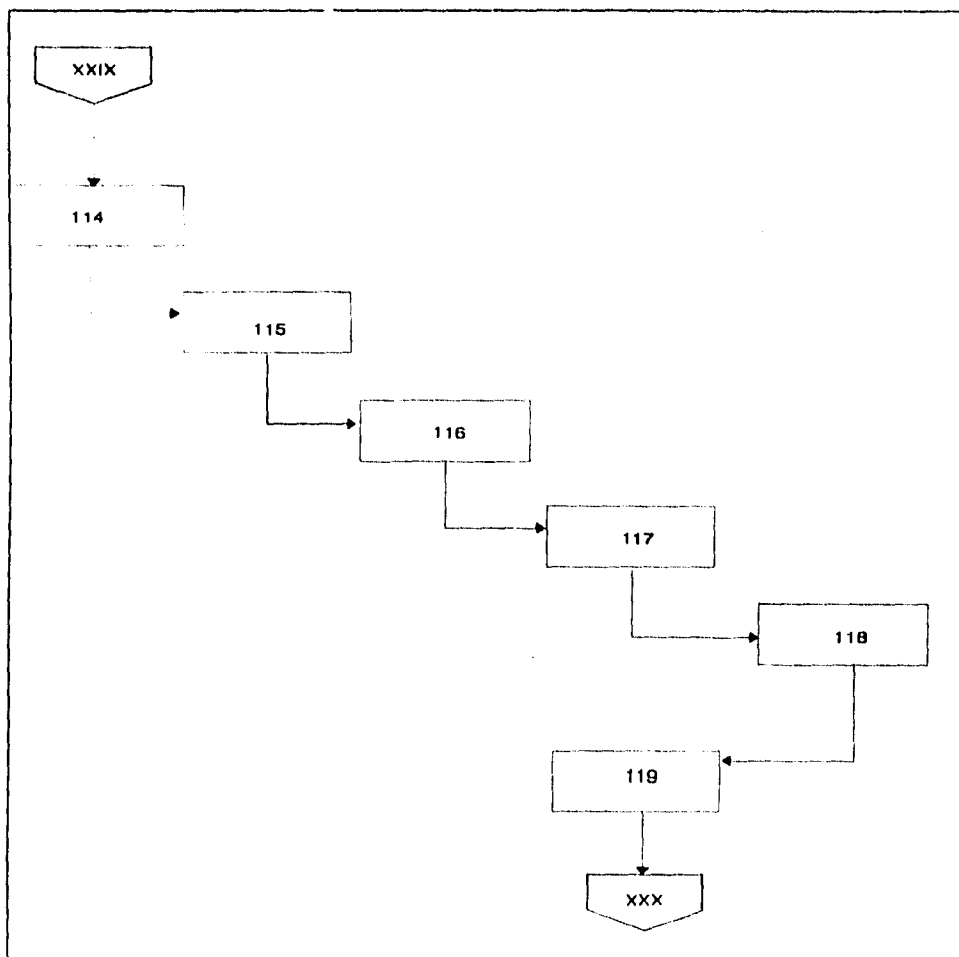


fig. II.1.2.30. Diagrama de Actividad 114, 115, 116, 117, 118 y 119

Jefe del departamento de documentación.

120. Recibe, revisa y verifica que las relaciones de pago analíticas de liquidaciones esten debidamente requisitadas y la turna al jefe de área.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

Jefe de área.

121. Recibe y turna a la superintendencia de control presupuestal la cédula resumen y las cuentas por pagar correspondientes a las liquidaciones de fletes de autotanques por autotanques, transportista y conocimientos de embarque por autotanque.

Suptica. de control presupuestal.

122. Recibe cédula resumen de las liquidaciones de autotanque, cuentas por pagar y conocimientos de embarque por autotanque o camión, firma de recibido en la libreta de control interno del departamento de documentación y registra los documentos por : (formas: cédula de pago, cuentas por pagar, forma 3-218).

- a) Número de mecanizado.
- b) Clave de autorizacion que asigno al reporte la superintendencia general de documentación y sistemas.
- c) Número consecutivo de cuentas por pagar.

123. Verifica que todos los documentos (cédula resumen, liquidaciones de fletes de autotanques y conocimientos de embarque por autotanque o camión coincidan por:

- a) Nombre clave del transportista
- b) Liquidaciones de fletes de autotanques cotejadas contra las cuentas por pagar.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

124. ¿ están elaborados correctamente los documentos y coinciden los datos ?

no: continúa en la actividad no. 125.

si: continúa en la actividad no. 126.

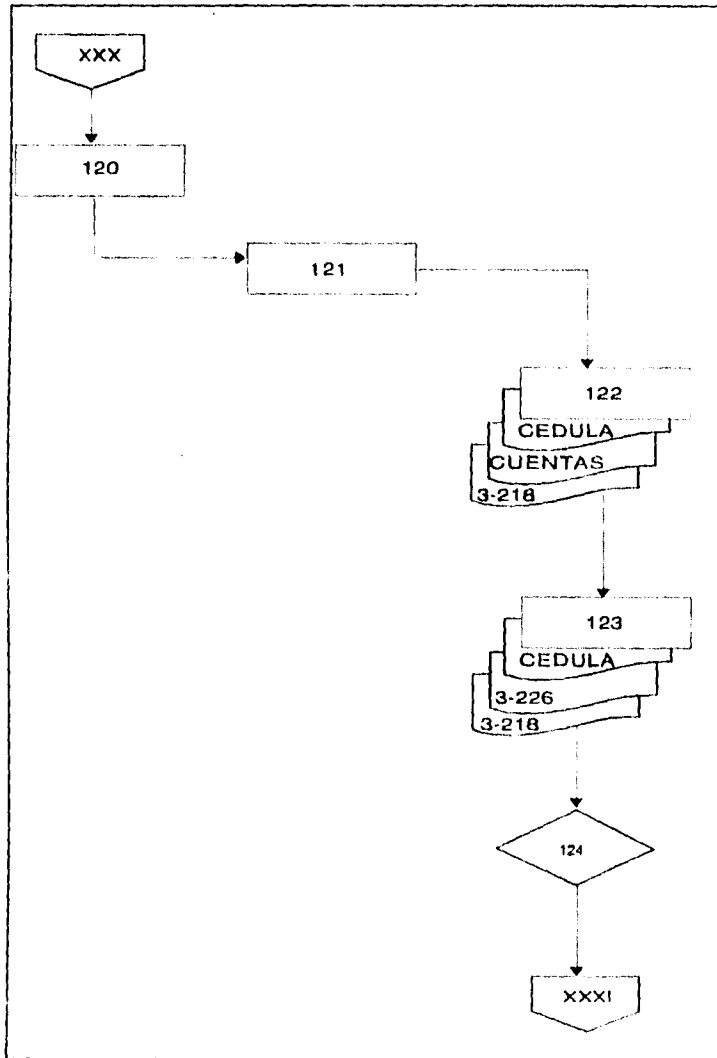


fig. II.1.2.31. Diagrama de Actividad 120, 121, 122, 123 y 124

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

125. Se cancela el folio con el que se recibieron los documentos (cédula resumen, cuentas por pagar, liquidaciones de fletes de autotanques y el conocimiento de embarque por autotanque o camión y se envía al departamento de documentación para que realice las correcciones pertinentes.

126. Perfora la cédula resumen y los conocimientos de embarque por autotanque o camión y las envía al departamento de documentación, con esta operación se evita una posible duplicidad en el trámite de dichos documentos. (formas: cédula de pago, forma 3-218).

Jefe de área.

127. Recibe las relaciones de pagos debidamente firmadas y las turna al personal foleador, verificador y de registro para sacar tres juegos de copias fotostáticas. (formas: cédula de pago, forma 3-218).

128. Recibe de la superintendencia de control presupuestal la documentación que se relaciona a continuación y la turna al personal foleador, verificador y de registro :

- a) Relación de pagos por concepto de fletes de autotanques. (formas: relación de pagos).
- b) Cuentas por pagar. (formas: cuentas por pagar).
- c) Liquidaciones de fletes de autotanque. (formas: 3-226).
conocimiento de embarque por autotanque o camión. (formas: 3- 218).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

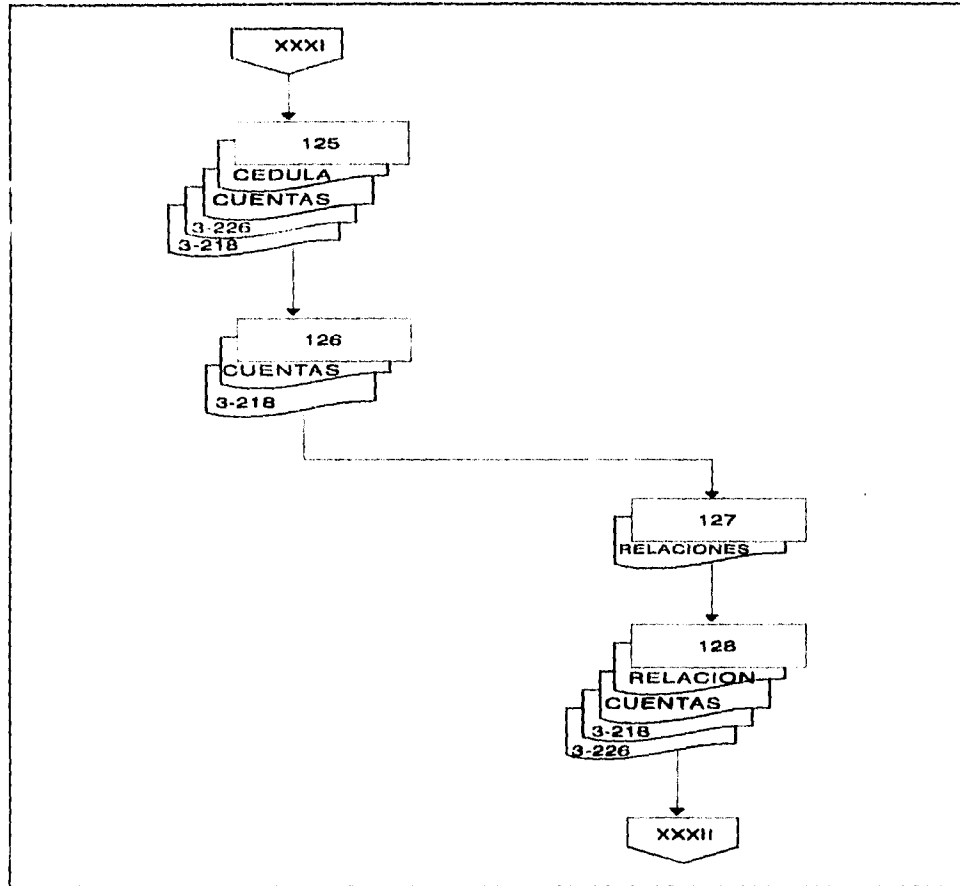


fig. II.1.2.32. Diagrama de Actividad 125, 126, 127 y 128

Personal foliador, verificador y de registro.

129. Recibe la documentación referida en la actividad anterior y procede a conjuntar los juegos con la siguiente información para turnar a la contaduría de transportes y obtener el acuse de recibo en una de las fotocopias del resumen de pago debidamente autorizado :

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

- a) Cuentas por pagar
- b) Analítico de liquidaciones
- c) Liquidaciones de fletes de autotanque

- d) Contra-recibo copia
- e) Conocimiento de embarque por autotanque o camión.

Contaduría de transportes.

130. Recibe la documentación referida en la actividad 129 y prosigue con los trámites para su pago, acusa de recibo.

Empleados del departamento de documentación.

131. Obtienen el acuse de recibo de la contaduría de transportes, así como de la gerencia de tesorería y de la caja general y turnan al jefe de área de autotanques la copia de la relación de pagos (cédula de pago) debidamente sellada. (formas: cédula de pago).

Jefe de área.

132. Recibe las cuentas por pagar, así como la copia fotostática de la misma, con los sellos de contaduría de transportes gerencia de tesorería y caja general y las archiva para aclaraciones posteriores. (formas: cuentas por pagar).

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

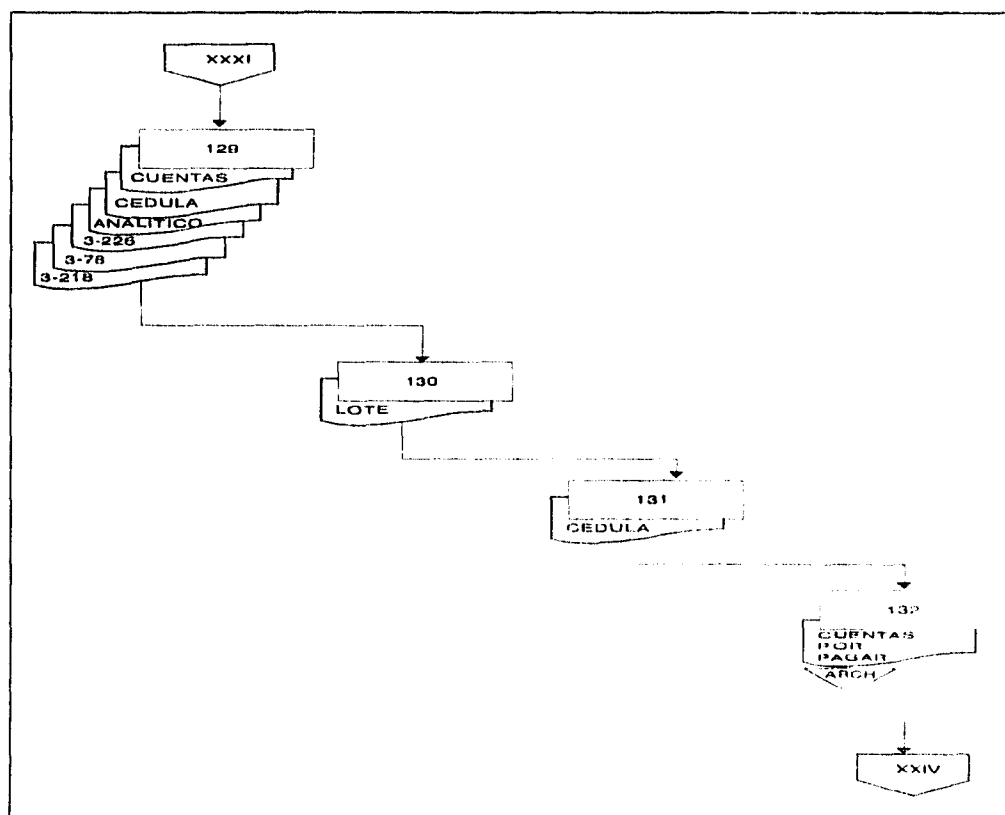


fig. II.1.2.33. Diagrama de Actividad 129, 130, 131 y 132

Contaduría de trasportes.

133. Recibe los documentos descritos en la actividad no. 129 y procede a separarlos conforme al reporte de validación y analítico de liquidaciones.

Capítulo II Análisis del Procedimiento de Pago de Fletes a Transportistas

134. Verifican que la documentación (cédula resumen, cuentas por pagar y conocimientos de embarque por autotranque o camión contengan la perforación (cancelación) hecha por la maquina perforadora de la superintendencia de control presupuestal. (formas: cédula de pago, cuentas por pagar, forma 3-218).

Contaduria de transportes.

135. Envía a la caja general las cuentas por pagar, reporte de validación, analítico de liquidaciones, contrarrecibo, copia y conocimientos de embarque por autotranque o camión. (formas: cuentas por pagar, reporte de validación, analítico de pagos, forma 3-78, forma 3-218).

termina procedimiento.

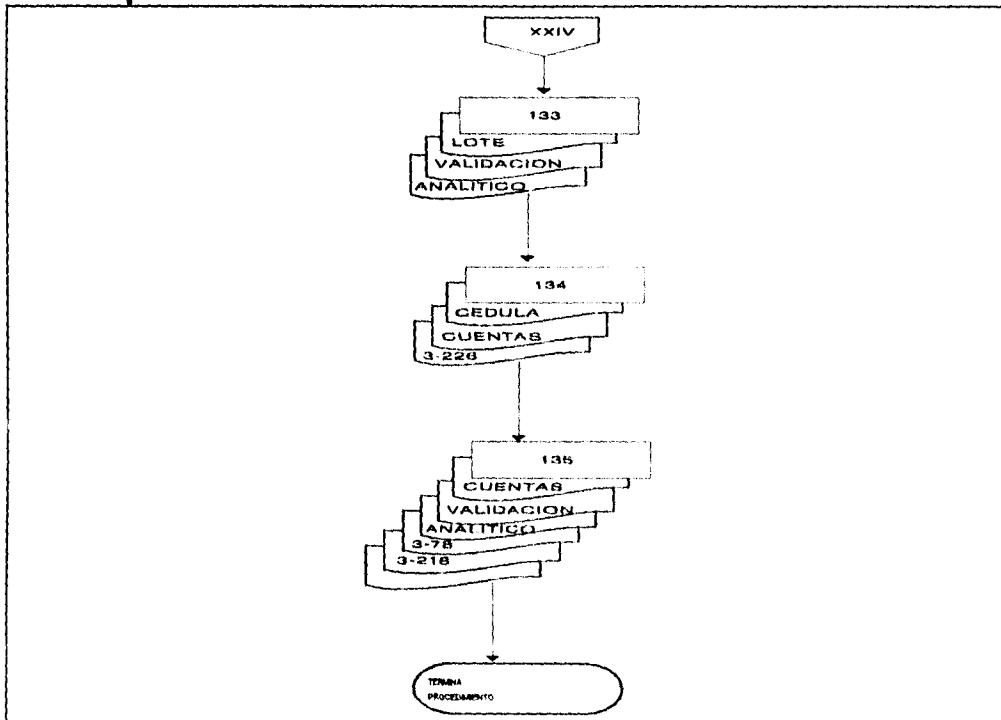


fig. II.1.2.34. Diagrama de Actividad 133, 134 y 135

II.3 PLANTEAMIENTO DE LA PROBLEMÁTICA Y DISPONIBILIDAD DE RECURSOS DEL ÁREA DE DISTRIBUCIÓN

Petróleos Mexicanos, efectúa el movimiento de sus productos desde los centros de producción, el almacenamiento, la distribución y venta de los mismos, a través de los centros embarcadores, terminales y agencias de ventas distribuidas en todo el país.

El crecimiento de la demanda de los productos petrolíferos y petroquímicos en el país, a ocasionado que los sistemas de venta y distribución sean complejos, por lo que se requiere contar con elementos para establecer un buen control, evitar operaciones erróneas y optimizar los procedimientos administrativos.

Por lo anterior y con el propósito de mejorar y agilizar la atención a clientes, así como para adecuar a los sistemas modernos de documentación y controlar las operaciones de los centros embarcadores, terminales y agencias de ventas, la Subdirección Comercial a la cual pertenece el área de Distribución, decidió implantar la automatización de dichos centros de trabajo.

Por otro lado, la estructura informática con que cuenta la C.E.D (Coordinación Ejecutiva de Distribución) a nivel de hardware, está constituida por un equipo heterogeneo y que en los centros pagadores foraneos, pertenecen a otras dependencias de Petróleos Mexicanos. Esto ha dado lugar a la implantación de diferentes versiones de los sistemas en función de las características de cada equipo y de las herramientas de desarrollo instaladas en los mismos.

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

En particular, los sistemas de Pago de Fletes vía Autotanque y Camión convencional, actualmente se encuentran implantados en tres diferentes equipos de cómputo, dos de los cuales pertenecen a la Gerencia de Informática, misma que también brinda apoyo a través de su equipo de captura, el cual por su obsolescencia, quedó fuera de servicio el año anterior.

Esta plataforma de equipo y por consecuencia, la necesidad de homogeneizar el software de desarrollo para los Sistemas de Pago, ha dado lugar al planteamiento del presente proyecto de migración y/o rediseño de los sistemas, a un ambiente que aproveche los avances tecnológicos, tanto a nivel de hardware como software, con los que se contarán a corto plazo.

II.3.1 PROBLEMATICA

El equipo de captura en el Distrito Federal será dado de baja y en el cual se capturan aproximadamente 2200 documentos diarios.

El proceso y administración del pago de fletes para carga regular se lleva manualmente, lo que trae como consecuencia, posibilidades de errores en los cálculos y principalmente falta de mecanismos dinámicos que permitan administrar la información ágil y oportunamente.

Se ha requerido continuamente capacitar al personal de sistemas en los diversos productos y versiones del software instalado en los diferentes equipos de cómputo, con los que se han construido los sistemas referidos.

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

Los cambios de políticas de pago y/o la incorporación de nuevos conceptos tiene que reflejarse en los diferentes sistemas implantados, dando lugar a comisiones de trabajo que van de días a semanas en los centros pagadores. Los cambios de equipo de cómputo y/o versiones de software han dado lugar a comisiones de trabajo de varios meses en el centro pagador.

Por otro lado, conjuntamente a la salida del Sistema Institucional de Contabilidad (S.I.C), al sistema de cheques de la caja general, y a la información operativa solicitada periódicamente por las áreas de asesoría de la C.E.D , se han ido desarrollando e implantando sobre la marcha, varios módulos que con el tiempo han requerido diferentes dependencias, tales como:

- 1) Auditoría de embarques documentados
- 2) Asignación del gasto por área de responsabilidad.
- 3) Información operativa de estas modalidades de transporte.

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

II.3.2 RECURSOS DE HARDWARE Y SOFTWARE

AÑO	RECURSOS DE HARDWARE Y SOFTWARE	
1980	HP 9845B Basic	Flow Tecnology 8020 Ensamblador
1983	SPPARSAM DATA SSD-2000	MB3
1984	IBM S/36	BASIC
1985	HP 3000 / 45	COBOL Speed Ware e IMAGE
1987	IBM S/36	COBOL
1988	IBM 3090	COBOL, ADABASE-NATURAL
1989	NCR Tower XP	UNIX, COBOL
1990	NCR Tower 32/600	UNIX, COBOL
1992	NCR 3000 y UNISYS 6000	UNIX, C, INFORMIX, RM COBOL

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

II.3.3 RECURSOS HUMANOS

Los recursos humanos que integran el área de Distribución, que soportan y/o administran directamente el procesos automatizado del pago son:

CENTRO PAGADOR	ANALISTA PRINCIPAL		PROGRAMADOR		
		PREPARADOR INF.		CODIFICADOR	CAPTURISTA
MEXICO	2	2	2	7	6
CADEREYTA	1			2	2
COATZACOALCOS	1			2	2

II.3.4 OBJETIVO DEL PROYECTO

Implantar en el presente año una infraestructura tanto en equipo de cómputo para la Coordinación Ejecutiva de Distribución que permita:

- 1) Contar con equipos de cómputo propios y software homogéneos para las diferentes zonas de pago.
- 2) Consolidar y unificar en un ambiente integral, los sistemas de Pago de Autotanques y de Carga Regular.
- 3) Evitar en lo posible que estos Sistemas dependan de equipos de otras áreas.

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

II.3.3 RECURSOS HUMANOS

Los recursos humanos que integran el área de Distribución, que soportan y/o administran directamente el procesos automatizado del pago son:

CENTRO PAGADOR	ANALISTA PRINCIPAL		PROGRAMADOR		
		PREPARADOR INF.	CODIFICADOR	CAPTURISTA	
MEXICO	2	2	2	7	6
CADEREYTA	1			2	2
COATZACOALCOS	1			2	2

II.3.4 OBJETIVO DEL PROYECTO

Implantar en el presente año una infraestructura tanto en equipo de cómputo para la Coordinación Ejecutiva de Distribución que permita:

- 1) Contar con equipos de cómputo propios y software homogéneos para las diferentes zonas de pago.
- 2) Consolidar y unificar en un ambiente integral, los sistemas de Pago de Autotanques y de Carga Regular.
- 3) Evitar en lo posible que estos Sistemas dependan de equipos de otras áreas.

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

- 4) Consolidar y administrar la información obtenida de las operaciones de Pago de Fletes del país, para la oportuna toma de decisiones a nivel operativo y ejecutivo.**
- 5) Crear interfases que fortalezcan el intercambio de información con otros sistemas locales y corporativos.**

II.3.5 ALCANCES.

Proporcionar una herramienta veraz, dinámica y eficientes a la Superintendencia General de Documentación y sistemas que le permita:

- 1) Llevar un control y seguimiento del banco de información de las compañías transportistas, así como de los autotranques autorizados para transportar productos de PEMEX.**
- 2) Registrar , Validar y Procesar los pagos por concepto de fletes realizados para PEMEX de acuerdo a la dinámica de las políticas y a la normatividad establecida por entidades internas y externas de PEMEX.**
- 3) Proporcionar un banco de información veraz y oportuno que apoye a la toma de decisiones, tanto a nivel operativo como a nivel ejecutivo.**
- 4) Establecer los mecanismos de intercambio de información eficientes entre las áreas y/o sistemas de contabilidad, caja, presupuestos, S.A.T (Sistema**

Capítulo II Planteamiento de la problemática y disponibilidad de los recursos

Automatizado de Terminales) y la propia Coordinación Ejecutiva de Distribución.

- 5) Contar con un banco de información histórica para la generación de estadísticas que proporcionen elementos de juicio en la toma de desciciones.

II.3.6 SITUACION ACTUAL

El propósito de este inciso , es dar a conocer un esquema general de los puntos que se consideran releventes en la operación, en los recursos y en la problemática existente hoy en día, en la Superintendencia General de Documentación y Sistemas a fin de entender las causas que originaron el planteamiento del proyecto que se describe a continuación.

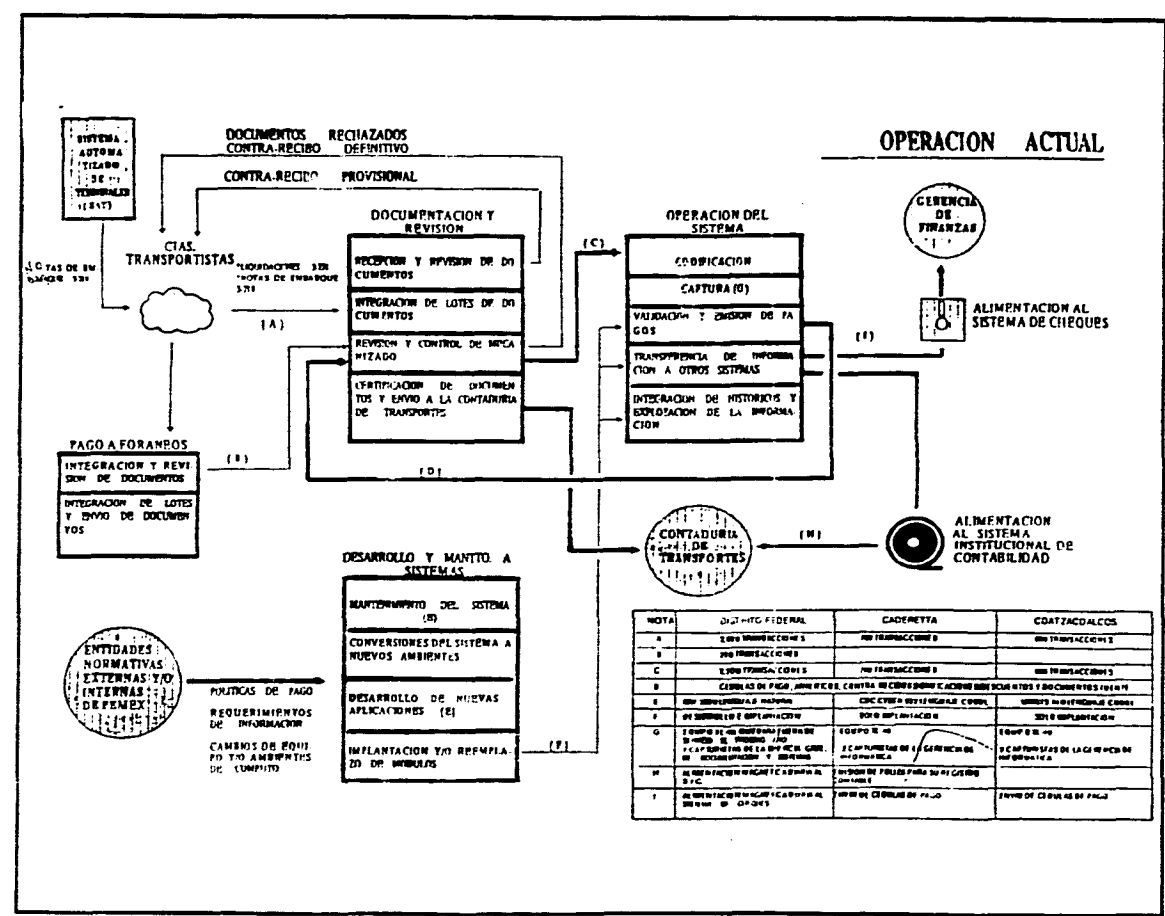


Fig 2.2.5.1 Diagrama Operación actual del SPF-AT

NOTA	DGT-ITFC FEDERAL	CADEREITA	COATZACOALCOS
A	200 TRANSACCIONES	NO TRANSACCIONES	NO TRANSACCIONES
B	200 TRANSACCIONES	NO TRANSACCIONES	NO TRANSACCIONES
C	200 TRANSACCIONES	NO TRANSACCIONES	NO TRANSACCIONES
D	CEBILAS DE PAGO, JORNALEROS, CONTRA-RECEPCION, EAGRE, BANCOS, BANCOS Y BANCOS	CEBILAS DE PAGO, JORNALEROS, CONTRA-RECEPCION, EAGRE, BANCOS, BANCOS Y BANCOS	CEBILAS DE PAGO, JORNALEROS, CONTRA-RECEPCION, EAGRE, BANCOS, BANCOS Y BANCOS
E	NO IMPLEMENTACION	NO IMPLEMENTACION	NO IMPLEMENTACION
F	NO IMPLEMENTACION	NO IMPLEMENTACION	NO IMPLEMENTACION
G	CONVERSIONES DEL SISTEMA A NUEVOS AMBIENTES	CONVERSIONES DEL SISTEMA A NUEVOS AMBIENTES	CONVERSIONES DEL SISTEMA A NUEVOS AMBIENTES
H	DESARROLLO DE NUEVAS APLICACIONES	DESARROLLO DE NUEVAS APLICACIONES	DESARROLLO DE NUEVAS APLICACIONES
I	IMPLEMENTACION Y/O REEMPLAZO DE EQUIPOS	IMPLEMENTACION Y/O REEMPLAZO DE EQUIPOS	IMPLEMENTACION Y/O REEMPLAZO DE EQUIPOS

II.4 ANÁLISIS DE LOS REQUERIMIENTOS DEL USUARIO

Debido a la problemática planteada, los requerimientos por área serán los siguientes:

II.4.1 REQUERIMIENTOS DEL AREA DE DISTRIBUCION.

El área requiere de implementar una infraestructura tanto de equipo de cómputo como de sistemas que permita:

Contar con equipos propios y software homogéneos para las diferentes zonas de pago.

Unificar en forma integral, los sistemas de pago de autotanques y carga regular.

Administrar la información producto de las operaciones de pago de fletes del país, para informes estadísticos y ejecutivos.

II.4.2 REQUERIMIENTOS DE LA ÁREA DE FINANZAS.

En esta área se requiere que se transmita, la información de los importes mensuales que genera el área de distribución por concepto de pago de fletes a transportistas.

Por ejemplo si de Coahuila se envía gasolina a Tula el costo de transportación debe ser informado al área de finanzas, para su control y aplicación.

Los datos mas importantes en esta área son :

Grupo producto
Agencia pagadora
Fecha inicial vigencia
Fecha fin vigencia
Valores iniciales mensuales
Hoja inicial de documento
Hoja inicial adefas
Número documento inicial
Número inicial adefas
Número inicial pago 2a. parte
Valores siguiente proceso:
Hoja siguiente documento
Hoja adefas
Número de adefas
Número pago 1a. parte
Número pago 2a. parte
Fecha de elaboración
Clave contaduria
Tipo de documento
Tipo de registro
Concepto de Origen
Concepto Origen I.V.A

Fig. II.4.2.1. Datos de Finanzas

Continuación
Dígito Aviso
Dígito Aviso de I.V.A.
Clave de proyecto
Centro de Trabajo Cargo
Departamento de Cargo
Cuenta Cargo
Subcuenta Cargo
Cuenta presupuestal Cargo
Centro de Trabajo Abono
Departamento de Abono
Cuenta Abono
Subcuenta Abono
Cuenta Presupuestal Abono
Centro trabajo Cargo I.V.A
Departamento Cargo I.V.A
Cuenta cargo I.V.A
Subcuenta Cargo I.V.A
Cuenta Presupuestal I.V.A
Centro de Trabajo Abono I.V.A.
Departamento de Abono I.V.A.
Cuenta Abono I.V.A.
Subcuenta Abono I.V.A.
Cuenta Presupuestal Abono I.V.A.

fig. II.4.2.2 Continuación

II.4.3 REQUERIMIENTOS DEL ÁREA DE CONTABILIDAD.

En esta área se requiere que se transmita la información de los cargos y abonos para la aplicación en cuentas contables y se aplique a cada centro de costos específico.

Por ejemplo el pago a una compañía transportista genera un cargo a la cuenta contable y se aplica al centro de costos determinado.

Los datos mas importantes en esta área son:

Grupo producto
Agencia pagadora
Fecha inicial vigencia
Fecha fin vigencia
Valores iniciales mensuales:
Hoja inicial de documento
Hoja inicial adefas
Número documento inicial
Número inicial adefas
Número inicial pago 2a. parte

fig. II.4.3.1. Datos de Contabilidad.

<p>Continuación:</p> <p>Valores siguiente proceso:</p> <ul style="list-style-type: none">Hoja siguiente documentoHoja adefasNúmero de adefasNúmero pago 1a. parteNúmero pago 2a. parteFecha de elaboración <p>Clave contaduría</p> <p>Tipo de documento</p> <p>Tipo de registro</p> <p>Concepto de Origen</p> <p>Concepto Origen I.V.A</p> <p>Dígito Aviso</p> <p>Dígito Aviso de I.V.A.</p> <p>Clave de proyecto</p> <p>Centro de Trabajo Cargo</p> <p>Departamento de Cargo</p> <p>Cuenta Cargo</p> <p>Subcuenta Cargo</p> <p>Cuenta presupuestal Cargo</p>

fig. II.4.3.2. Continuación

Continuación.....

Centro de Trabajo Abono

Departamento de Abono

Cuenta Abono

Subcuenta Abono

Cuenta Presupuestal Abono

Centro trabajo Cargo I.V.A

Departamento Cargo I.V.A

Cuenta cargo I.V.A

Subcuenta Cargo I.V.A

Cuenta Presupuestal I.V.A

Centro de Trabajo Abono I.V.A

Departamento de Abono I.V.A

Cuenta Abono I.V.A.

Subcuenta Abono I.V.A.

Cuenta Presupuestal Abono I.V.A.

fig. II.4.3.3 Continuación.

II.5 ORGANIZACION Y ANALISIS DE LA INFORMACION

De acuerdo al análisis de los requerimientos del usuario realizado anteriormente, se definieron las siguientes estructuras:

1) ACUMULADO MENS. EMB. PAG.

Año

Mes

Origen

Destino

Clave Producto

Grupo Producto

Clave Porteador

Número de Embarque

Cantidad Embarcada

Importes:

Flete

Deducción

P.E.C.E. (Pacto de Estabilidad y Crecimiento Económico)

I.V.A. (Importe al Valor Agregado)

Peaje

Kilómetros Recorridos

Kilógramo por kilómetro

Fig. II.5.1. Estructura Acumulado Mens. Emb. Pag.

2) AUTOTANQUES
Clave porteador
Clave autotanque
Tipo de servicio
Marca de Tractor
Modelo Tractor
Número de serie
Marca
Modelo
Números de ejes
Capacidad
Número de serie
Número de folio calibración
Fecha de folio calibración
Tipo de situación
Fecha de situación
Número de documento
Fecha de documento
Observaciones
Fecha ingreso autotanque
Fecha último viaje autotanque

Fig. II.5.2 Estructura Autotanques

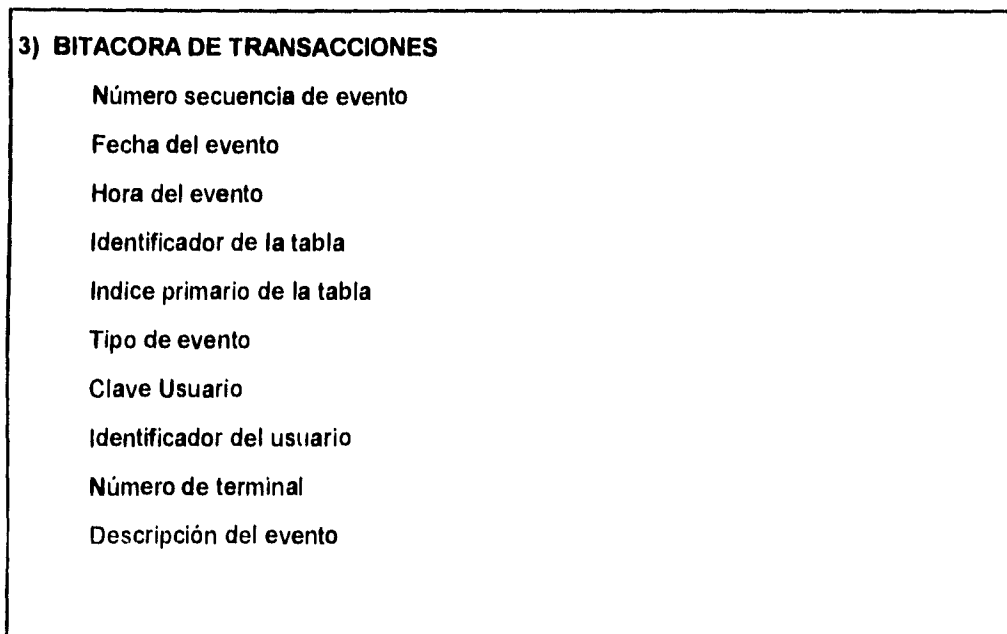


Fig. II.5.3. Estructura Bitacora de Transacciones

4) BONIFICACIONES/DESCUENTOS
Clave de porteador
Número de documento
Fecha de documento
Clave concepto documento
Importe de documento
Número de cédula
Clave tipo de cédula
Porcentaje a aplicar
Fecha de aplicación
Saldo
Estatus de registro

Fig. II.5.4. Estructura Bonificaciones / Descuentos

5) CALIBRACIONES SITUACIONES
Clave porteador
Clave autotanque
Fecha inicial de calibración
Fecha final de calibración
Tipo de movimiento
Capacidad
Fecha de documento
Número de documento

Fig. II.5.5. Estructura Calibraciones Situaciones

6) CARACTERISTICAS RUTAS
Origen
Destino
Ruta
Kilómetros
Importe Bonificación
Porcentaje Bonificación
Fecha de Inicio
Fecha Final
Kilómetros-Terracería
Porcentaje Adicional de Gas
Importe Transbordador
Clave Autorización

Fig. II.5.6. Estructura Características Rutas

7) CASETAS POR RUTA
Origen
Destino
Ruta
Clave caseta-puente

Fig. II.5.7. Estructura Caseta por Ruta

8) CEDULA DE PAGO

Números de cédulas

Clave tipo de cédula

Fecha de proceso

Fecha programado de pago

Agencia pagadora

Clave porteador

Número de contrarecibo

Importes

Flete total

Deducción por faltante

I.V.A.

P.E.C.E.

Peaje total

Por bonificaciones

Por descuentos

Datos flujo de efectivo

Fecha de recepción

Fecha de depósito bancario

Fig. II.5.8. Estructura Cédula de Pago

Datos Contables
Fecha elaboración
Números de documento
Hoja de documento
Número de adefas
Clave de autorización
Datos Financieros
Folio finanzas
Estatus Finanzas

Fig. II.5.8a. Continuación

9) CONCEPTOS DE BON/DESC.
Clave concepto
Tipo de bonificación-descuento
Descripción bonificación-descuento

fig. II.5.9. Estructura Conceptos de bon/Desc.

10) COSTOS
Clave Caseta/Puente
Fecha Inicial
Fecha Final
Costos:
2 Ejes
3 Ejes
5 Ejes
6 Ejes
9 Ejes
Adicional por Eje

Fig. II.5.10. Estructura Costos

11) CONTROL DE BATCH'S

Identificador del usuario

Número de batch

Fecha de captura

Tiempo de captura

Clave usuario verificación

Identificación usuario verificador

Clave estatus del batch

Número de liquidaciones

Número de embarques

Fig. II.5.11. Estructura Control de Batches

12) CUOTAS PEMEX

Kilometraje

Fecha inicio cuota

Fecha final cuota

Cuota grupo 1

Cuota grupo 2

Cuota grupo 3

Cuota grupo 4

Cuota grupo 5

Cuota grupo 6

Cuota grupo 7

Cuota grupo 8

Fig. II.5.12. Estructura Cuotas Pemex

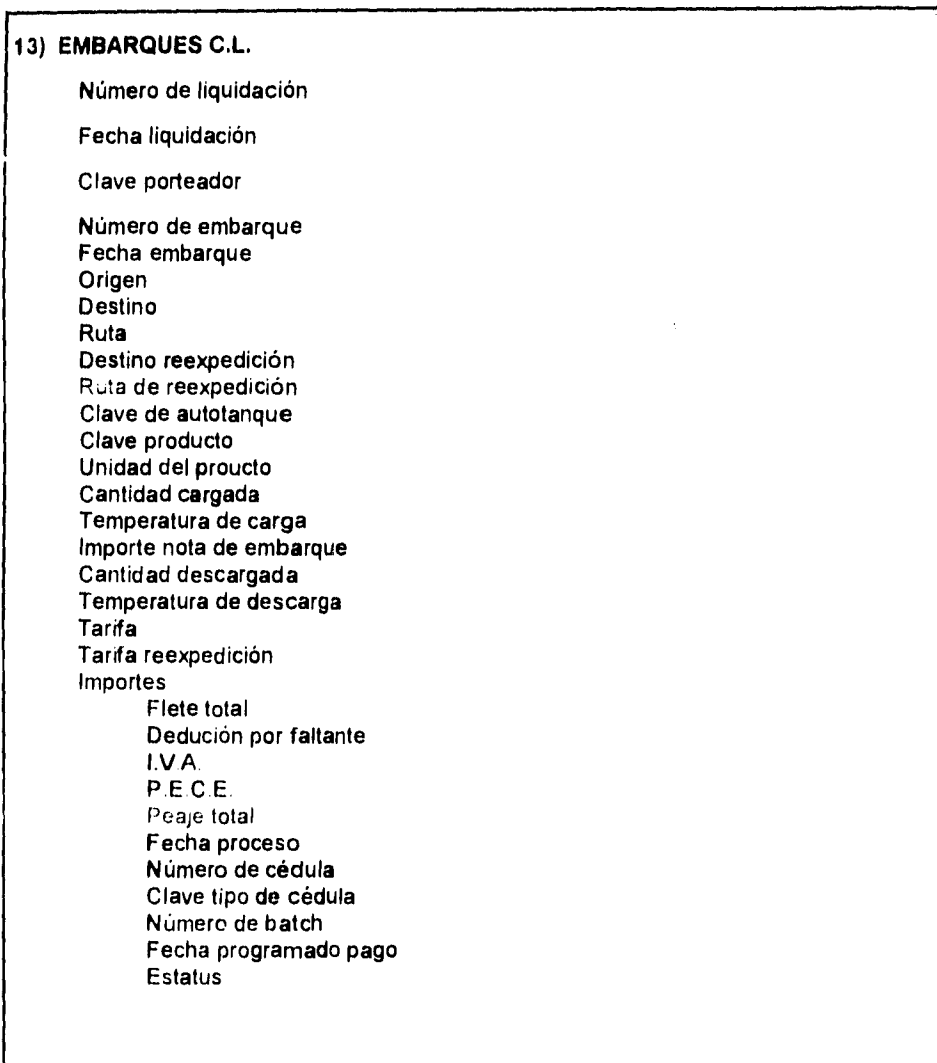


Fig. II.5.13. Estructura Embarques C.L.

14) EMBARQUES NO PAGADOS

Clave porteador

Número embarque

Fecha embarque

Origen

Destino

Ruta

Destino reexpedición

Ruta de reexpedición

Clave de autotanke

Clave producto

Unidad del producto

Cantidad cargada

Temperatura de carga

Importe nota de embarque

Cantidad descargada

Temperatura de descarga

Fig. II.5.14. Estructura Embarques No Pagados

15) HISTORICO EMB. PAG. C.L.
Número Liquidación
Fecha Liquidación
Clave Porteador
Número Embarcación Histórico
Letras Núm. Embarque
Dígitos Núm. Embarque
Fecha Embarque
Origen
Destino
Ruta
Destino Reexpedición
Clave Autotanque
Clave Producto
Unidad del Producto
Deducción por Faltante
I.V.A. (Importe al Valor Agregado)
P.E.C.E. (Pacto de Estabilidad de Crecimiento Económico)
Peaje Total
Fecha Proceso
Número de Cédula
Clave Tipo de Cédula
Datos de Pago
Fecha Programado Pago
Grupo de Producto
Peso Específico
Kilómetros
Cantidad Cargada
Temperatura de Carga
Cantidad Descargada
Temperatura de Descargada
Tarifa
Tarifa Reexpedición
Importes
Flete Total
Número Contrarrecibo
Agencia Pagadora
Status

Fig. II.5.15. Estructura Histórico Emb. Pag. C.L.

16) EMBARQUES REPORTADOS

Embarque Inicial
Embarque Final
Número Documento
Fecha Documento

Fig. II.5.16. Estructura Embarques Reportados.

17) GRUPOS DE PRODUCTOS

Grupo de producto
Descripción grupo producto

Fig. II.5.17. Estructura Grupo de Productos

18) GRUPOS EMPRESARIALES

Clave grupo empresarial
Nombre del grupo empresarial

Fig. II.5.18. Estructura Grupos Empresariales

19) HIST. EMB. C.L. SELECCIONADO

Número de liquidación
Fecha liquidación
Clave porteador
Número embarque histórico
 Letras número embarque
 Digitos números embarque
Fecha embarque
Origen
Destino
Ruta
Destino reexpedición
Clave autotanque
Clave producto
Unidad del producto
Grupo de producto
Peso específico
Kilómetros
Cantidad cargada
Temperatura de carga
Cantidad descargada
Temperatura de descarga
Tarifa
Tarifa reexpedición
Importes
 Flete total
 Deducción por faltante
 I.V.A.
 P.E.C.E.
Peaje total
Fecha proceso
Número de cédula
Clave tipo de cédula
Número programado pago
Número contrarecibo
Agencia pagadora
Estatus

Fig. II.5.19. Estructura Hist. Emb. C.L. Seleccionado

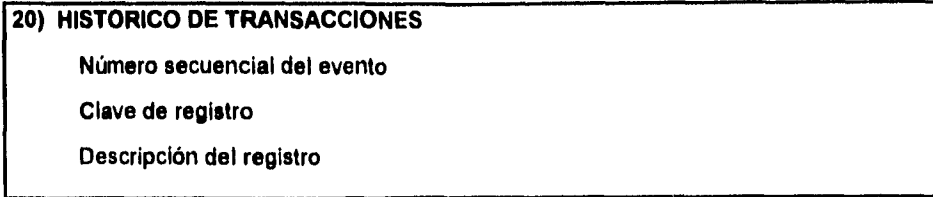


Fig. II.5.20. Estructura Histórico de Transacciones

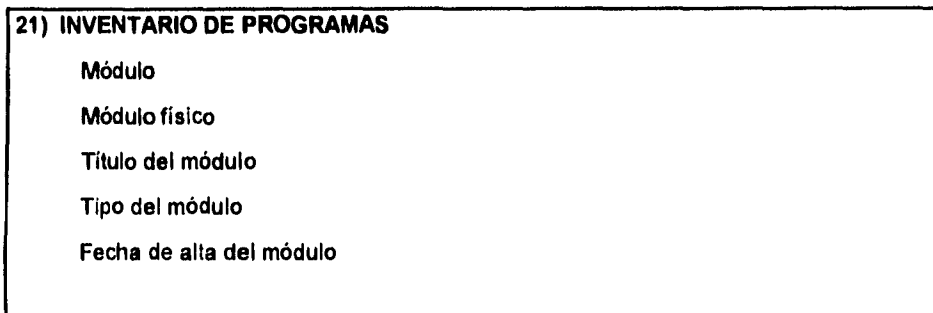


Fig. II.5.21. Estructura Inventario de Programas



Fig. II.5.22. Estructura Inventario de Tablas

23) LIQUIDACIONES
Número de cédula
Clave tipo de cédula
Clave porteador
Número liquidación
Fecha liquidación
Importes
Flete total
Deducción por faltante
I.V.A.
P.E.C.E.
Peaje total
Agencia pagadora

Fig. II.5.23. Estructura Liquidaciones

24) MARCAS DE TRACTORES Y TANQUES
Marca tractor
Tipo tanque-tractor
Nombre marca

Fig. II.5.24. Estructura Marcas de Tractores y Tanques

25) MOVIMIENTOS DIARIOS
Número de bach
Número de registro
Número de cédula
Tipo de batch
Número liquidación
Fecha liquidación
Clave porteador
Número embarque
Fecha embarque
Origen
Destino
Ruta
Destino reexpedición
Ruta reexpedición
Clave de autotanque
Clave producto
Unidad del producto
Cantidad cargada
Temperatura de carga
Importe nota de embarque
Cantidad descargada
Temperatura de descarga
Clave de estatus

Fig. II.5.25. Estructura Movimientos Diarios

26) MOV.RECHAZADOS POR CALC.
Número Liquidación
Fecha Liquidación
Clave Porteador
Número Embarque
Fecha Embarque
Origen
Destino
Ruta
Destino Reexpedición
Ruta de Reexpedición
Clave Autotanque
Clave Producto
Unidad de Producto
Cantidad Cargada
Temperatura de Carga
Importe Nota de Embarque
Cantidad Descargada
Temperatura de Descarga
Tarifa
Tarifa Reexpedición
Importes:
Flete Total
Deducción por Faltante
I.V.A.
P.E.C.E.
Peaje Total
Fecha Proceso
Número de Cédula
Clave tipo de Cédula
Número de Batch
Código de Error

Fig. II.5.26. Estructura Mov. Rechazados por Calc.

27) ORIGENES/DESTINOS
Origen / Destino
Nombre largo

Fig. II.5. 27 . Estructura Origenes / Destinos

28) PAGOS APLICADOS
Clave porteador
Número documento
Número de cédula
Clave tipo de cédula
Origen del registro
Fecha del registro
Importe aplicado
Fecha aplicación
Saldo

Fig. II.5.28. Estructura Pagos Aplicados

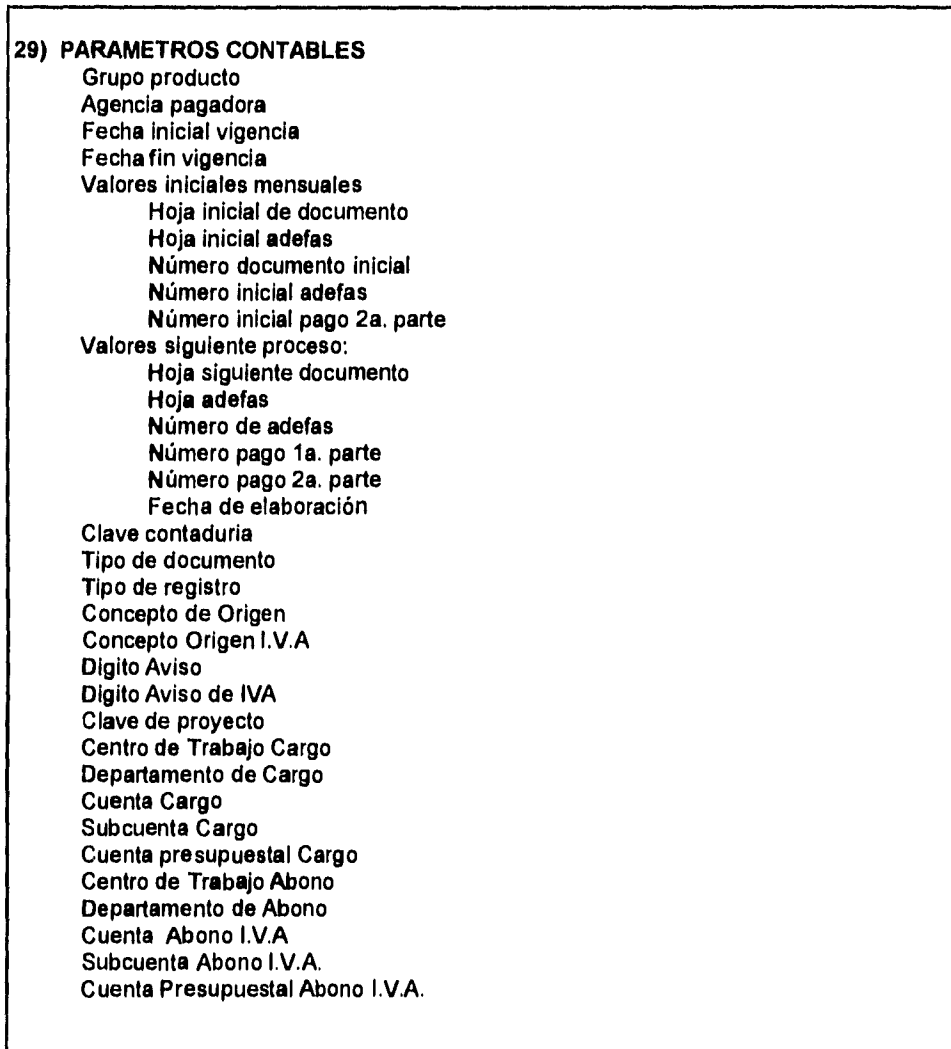


Fig. II.5.29. Estructura Parámetros Contables

Departamento de Abono
Cuenta Abono
Subcuenta Abono
Cuenta Presupuestal Abono
Centro de Trabajo Cargo I.V.A.
Departamento Cargo I.V.A.
Cuenta Cargo I.V.A.
Subcuenta Cargo I.V.A.
Cuenta Presupuestal I.V.A.
Centro de Trabajo Abono I.V.A.
Departamento de Abono I.V.A.

Fig. II.5.29a. Continuación

30) PARAMETROS DEL SISTEMA

- Clave Registro de Parámetros
- Clave Usuario que Elabora
- Identificador de quien Elabora
- Clave Usuario que Revisa
- Identificador de quien Revisa
- Clave Usuario que Autoriza
- Identificador de quien Autoriza
- Secuencial de Contra-Recibos

Fig. II.5.30. Estructura Parámetros del Sistema

31) PEAJE TOTAL

Origen

Destino

Ruta

Costos:

Total 2 Ejes

Total 3 Ejes

Total 5 Ejes

Total 6 Ejes

Total 9 Ejes

Total adicional por Eje

Fecha Inicio Vigencia

Fecha Final Vigencia

fig. II.5.31. Estructura Peaje Total.

32) PORTEADORES

Clave Porteador

Razón Social

Clave de Grupo Empresarial

Tipo de Empresa

Calle

Colonia

Ciudad

Código Postal

Apartado Postal

Municipio Estado

Teléfono

Extensión

Fax

Responsable de la Compañía

Centro Pagador

Fecha de Movimiento al Expediente

Tipo situación del Registro

Fecha Situación del Registro

Status Bonificación/Descuento

Fig. II.5.32. Estructura Porteadores

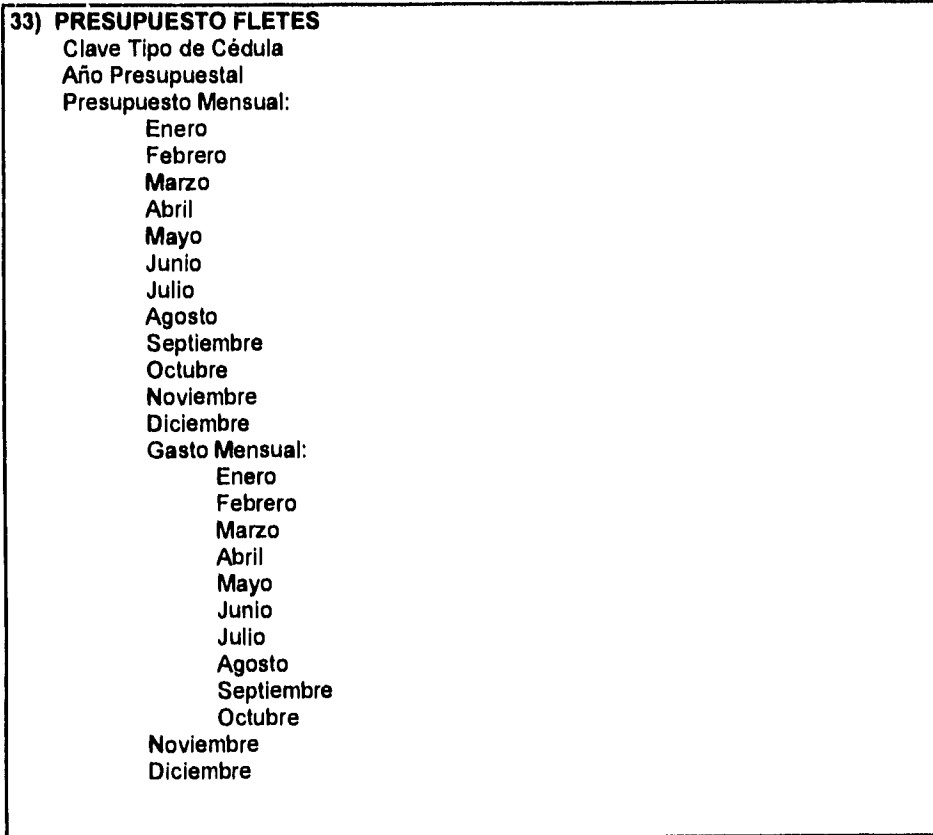


Fig. II.5.33. Estructura Presupuesto Fletes

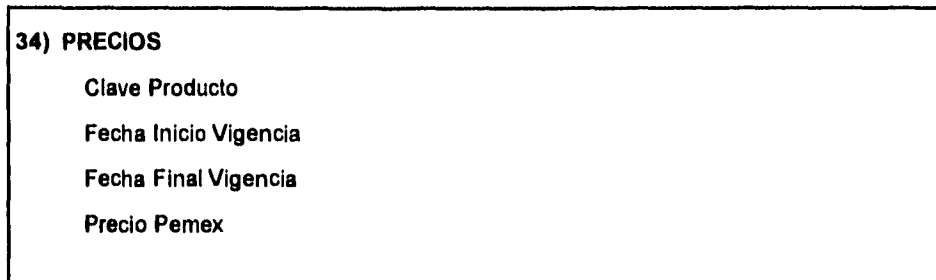


Fig. II.5.34. Estructura Precios

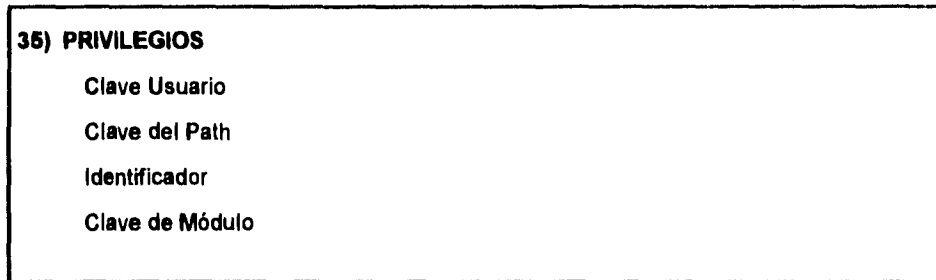


Fig. II.5.35 Estructura Privilegios

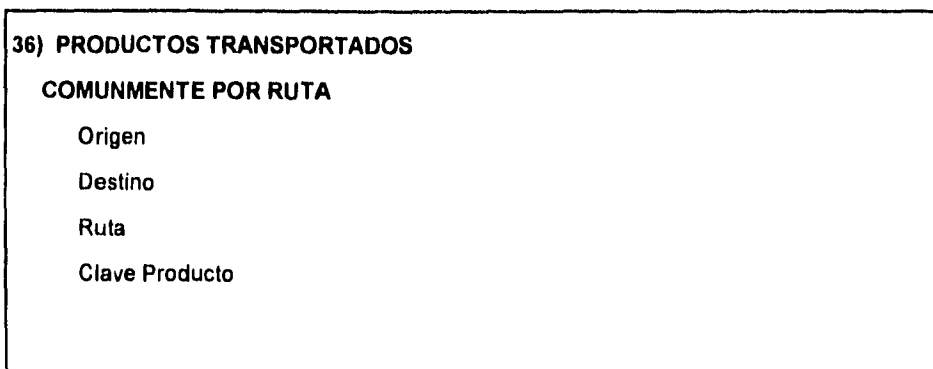


Fig. II.5.36. Estructura Productos Transportados Comúnmente por Ruta

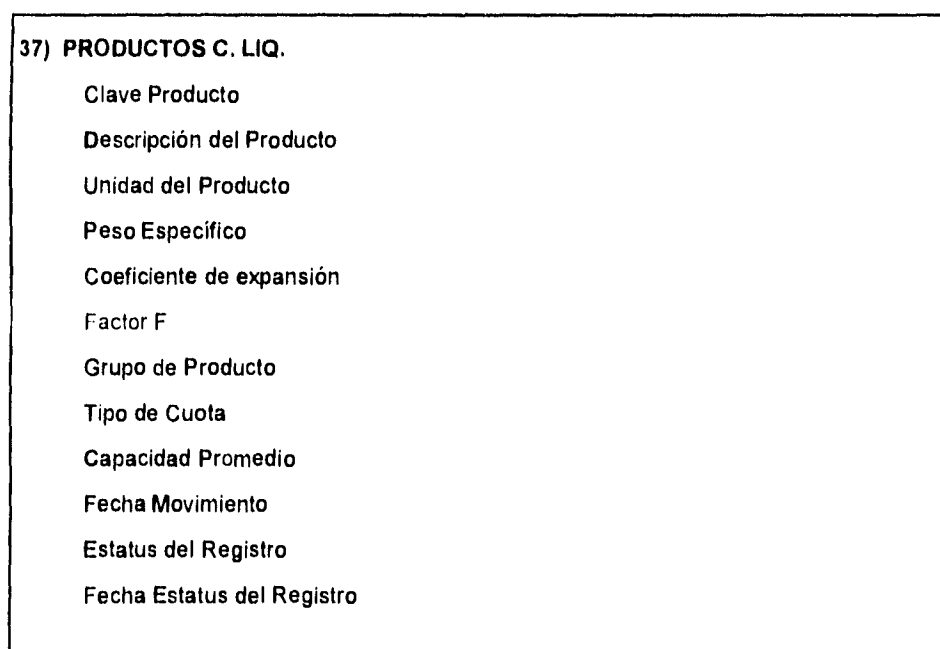


Fig. II.5.37. Estructura Productos C. Liq.

38) Puentes y Casetas

Clave Caseta/Puente

Tipo de Caseta/Puente

Nombre o Descripción

Fig. II.5.38. Estructura Puentes y Casetas

39) Tarifas Especiales

Origen

Destino

Ruta

Clave Producto

Fecha Inicio Tarea

Fecha Final Tarifa

Importe Tarifa

Fig. II.5.39. Estructura Tarifas Especiales

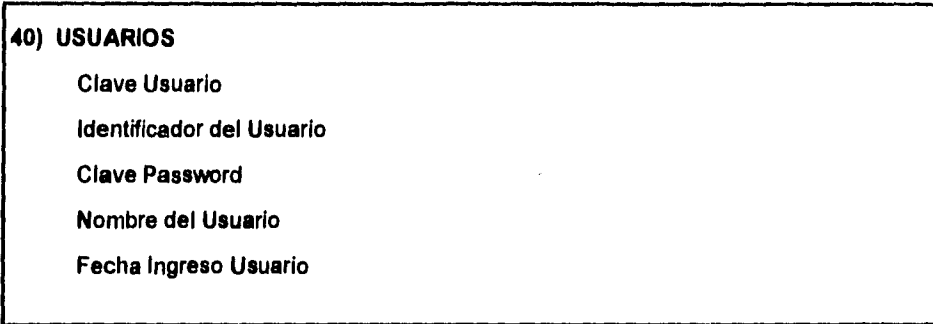


Fig. II.5.40 Estructura Tarifas Especiales

II.6 EVALUACIÓN Y SELECCIÓN DE OPCIONES DE SOLUCIÓN

Con la idea de mejorar y optimizar los procedimientos actuales para llevar a cabo con éxito este proyecto, es necesario satisfacer una serie de requerimientos técnicos y humanos que se apeguen a las políticas de desarrollo de los sistemas de información establecidos por la empresa. Dichos lineamientos son con el deseo de normar y controlar el crecimiento del esquema informático de las diversas áreas de desarrollo de sistemas que integran a la empresa.

Es por ello la necesidad de proporcionar una herramienta veraz, dinámica y eficiente con la cual la Superintendencia General de Documentación y Sistemas pueda:

- 1) Llevar un control y seguimiento del banco de información de las compañías transportistas, así como de los autotanques autorizados para transportar productos de PEMEX.
- 2) Registrar, Validar y Procesar los pagos por concepto de fletes realizados para PEMEX de acuerdo a la dinámica de las políticas y a la normatividad establecida por entidades internas y externas de PEMEX.
- 3) Proporcionar un banco de información veraz y oportuno que apoye a la toma de decisiones, tanto a nivel operativo como a nivel ejecutivo.

- 4) Establecer los mecanismos de intercambio de información eficientes entre las áreas y/o sistemas de contabilidad, caja, presupuestos, S.A.T (Sistema Automatizado de Terminales) y la propia Coordinación Ejecutiva de Distribución.

- 5) Contar con un banco de información histórica para la generación de estadísticas que proporcionen elementos de juicio en la toma de decisiones.

II.6.1 ESTANDARES ESTABLECIDOS

A NIVEL DE HARDWARE

Petróleos Mexicanos incluye en sus operaciones una gran cantidad de especialidades, que van desde la exploración y explotación del petróleo hasta la distribución de productos terminados, pasando por actividades como la exportación, refinación, petroquímica y para funciones de apoyo financieras, manejo a personal, servicios médicos, etc., tiene la necesidad, para cumplir con los esquemas marcados por el Gobierno Federal de modernización de la empresa pública, de prácticamente todo tipo de servicios de procesamiento de datos y presentación de información.

Estas amplias necesidades que se tienen, requieren de la coordinación de todos los aspectos tecnológicos informáticos para garantizar que la información se capte en el lugar de origen y se lleve hasta los centros de consumo, es por esto que mediante la estandarización de la tecnología informática se debe buscar una congruencia que nos permita garantizarle a la empresa, la conectividad y el transporte de la información, minimizando entre otras cosas el estar capacitando de manera repetitiva al personal en productos similares.

A NIVEL DE SOFTWARE

- 1) Sistema Operativo UNIX**
- 2) Tecnología de Bases de Datos Relacional**
- 3) Tecnología abierta (Open Systems) sistemas abiertos**
- 4) Lenguaje C**
- 5) Manejadores Transaccionales (OLTP's)**
- 6) SQL ansi**
- 7) Lenguajes de 4a. generación para desarrollo**
- 8) Tecnología CASE para la automatización de desarrollo**
- 9) Sistema Operativo de red Lan-Manager para Unix ó compatible**

Estos estandares tecnológicos, aplican en todos los ámbitos informáticos y se irán adecuando a las necesidades de la empresa derivadas del proceso de descentralización y desconcentración.

Los beneficios de la estandarización son :

- 1) Independencia tecnológica de proveedores.
- 2) Conectividad entre todos los equipos de Pemex.
- 3) Transparencia en la transferencia de información.
- 4) Obtención de mejores equipos a menores costos.
- 5) Mercado oferente de soluciones más amplio.
- 6) Soluciones modernas y eficientes.

BASES DEL CONCURSO PARA ADQUISICIÓN DE EQUIPO:

Petróleos Mexicanos tiene por política en lo que se refiere a adquisiciones , lanzar concursos para que cualquier empresa respetable pueda entrar a ellos. Estos concursos necesitan unas bases técnicas y funcionales que tendrán que ser cumplidas por los proveedores y las cuales van de acuerdo a las necesidades específicas del proyecto.

Las siguientes características son los requerimientos que se necesitaban para el desarrollo y la implantación del Sistema de Fletes de Carga Líquida:

- 1) Memoria de 16 MB, libres para usuario
- 2) TCP-A = 25 TPS
- 3) AIM performance rating = 25 AIM's
- 4) Máxima carga de usuarios = 210 usuarios
- 5) Máxima carga de trabajo = 250 jobs/min

- 6) Memoria cache de 8 KB
- 7) Espacio en Disco de 200 MB; con tiempo de acceso de 18 ms, y crecimiento hasta de 2GB.
- 8) Cartucho para cinta magnética de 150 MB compatible con el de 46 MB.
- 9) Unidad de diskette de 3 1/2" de 1.44 MB
- 10) 6 puertos seriales asíncronos
- 11) Puerto paralelo para impresora
- 12) Tarjeta para comunicaciones SNA/X.25 con 2 puertos High Order
- 13) Tarjeta para red Ethernet (servidor)
- 14) batería de respaldo
- 15) Monitor para la consola super VGA con pantalla 14" en diagonal
- 16) Teclado tipo qwerty de 101 teclas en español, incluyendo 12 teclas de función y teclado de cursos independiente del teclado numérico
- 17) Cable de señal
- 18) Terminales monocromáticas con pantalla de 12" en diagonal
- 19) Teclado tipo qwerty de 101 teclas en español, incluyendo 12 teclas de función y teclado de cursos independiente del teclado numérico
- 20) Capacidad de transmisión de la señal hasta de 150 mts, sin necesidad de usar Modem o cualquier otro dispositivo electrónico para amplificación/repeticón de señal

A dicho concurso entraron 5 proveedores: HP, IBM, UNISYS y NCR cuyas propuestas permitieron realizar un Dictamen Técnico y otro Económico.

La evaluación técnica y operativa se calificó, además de cumplir con las bases técnicas fué aplicado un "Benchmark", pruebas que consistieron en medir el rendimiento de la máquina corriendo pruebas de AIM's para el procesador y para el ambiente de bases de datos pruebas de TPC/A de donde finalmente salió ganador la Cía. NCR.

En lo referente a adquisición de Software el área en la que se desarrollará el proyecto es la encargada de evaluar los puntos de vista necesarios para elegir el productos indicado.

Por lo que para la elección de la Base de Datos, se realizó un análisis del mercado tomando en cuenta las bases de datos más sobresalientes como: **PROGRESS, SYSDATABASE, ORACLE e INFORMIX.**

De las cuales se eligieron **INFORMIX Y ORACLE**, debido a su mayor presencia en el mercado mexicano y por estar trabajando en Sistema Operativo UNIX. Finalmente se analizaron ambas opciones y se consideró a **ORACLE** con ciertas desventajas como:

- a) Mayor Costo
- b) Consume muchos recursos (memoria, espacio en disco etc.) con una proporción 3 veces más que **INFORMIX**.
- c) Su lenguaje 4GL para desarrollo es muy pobre y menos explotable

CARACTERISTICAS DE INFORMIX

VENTAJAS DE INFORMIX ONLINE:

OnLine brinda un amplio rango de beneficios que cumplen con los requerimientos demandados por los ambientes computacionales: alto performance, alta disponibilidad, consistencia de datos, tecnología multimedia y mucho más.

HIGH PERFORMANCE:

Informix diseño OnLine con desempeño inherente. En las series de TPC benchmarks y pruebas de comportamiento por productos evaluadores, Informix provee un desempeño superior en tiempo, después del tiempo obtenido por los productos de la competencia.

OnLine logra este alto performance mediante los eficientes métodos de almacenamiento de datos para el rápido acceso a ellos.

Los datos que se encuentran en buffering minimizan el acceso a disco, utilizando características de multiprocesador habilita simultáneos accesos por diferentes procesadores; y determinando automáticamente la estrategia de búsqueda más eficiente.

HIGH AVAILABILITY:

Cuando las compañías dependen de un ininterrumpido flujo de información de otros sistemas OLTP, cierta cantidad de tiempos muertos, pueden ser perjudiciales o incluso desastrosos.

Cualquiera de esos compromisos de habilidad de el sistema, rutinas de mantenimiento, problemas de software, o fallas en el sistema reducen la productividad y desvían la rentabilidad.

OnLine minimiza el impacto de los tiempos muertos al proveer mirroring al duplicar los datos críticos.

Online archiva para permitir interrupciones en las operaciones de las bases de datos, y capacidad de arranque para transacciones en firme. Todas estas características encuentran la habilidad de los requerimientos demandados hoy en día por los negocios.

CONSISTENCIA DE DATOS:

Cuando usted depende de sus datos para tomar una decisión exacta, no se puede cuestionarse esta integridad. OnLine asegura que aquellos datos están guardados correctamente y consistentes, no obstante un número muy grande de usuarios pueden estar accesorando. OnLine hace esta vía en distintos mecanismos estructurales- a los sistemas y niveles de transacción, sin ningún posicionamiento innecesario barre la información accesorando y uniforme performance.

LAS CAPACIDADES DEL AMBIENTE DISTRIBUIDO CLIENTE/SERVIDOR:

Como un proceso necesita extender sus clientes locales distribuidos, las comunicaciones cliente/servidor se convierten en críticos para compartir datos entre el Servidor/cliente de datos y los diferentes clientes.

OnLine hace transparente la manipulación de Base de datos múltiples en diferentes máquinas.

FÁCIL ADMINISTRACIÓN:

OnLine provee diferentes herramientas para disponer del mejor uso de las herramientas del Sistema.

ADMINISTRACIÓN DE RAW DISK:

Para acceder datos rápidamente Online consigue espacio contiguo en disco de almacenaje por medio de un raw devices.

OnLine usa raw disks par almacenaje por los archivos nativos de Unix no garantizan alocaación de espacio contiguo en disco. La localidad de espacio en un raw disk, opuestamente OnLine crea un dueño del sistema de almacenamiento de datos.

El almacenamiento de datos en el raw device permite el acceso directo de Online a la memoria.

MEMORIA COMPARTIDA:

La memoria compartida se refiere al acceso compartido para los mismos segmentos de memoria por más de un proceso OnLine.

Sus ventajas son: El tiempo de ejecución es reducido, el espacio de la memoria compartida es utilizada para OnLine a través de tablas internas y paginas de buffers.

CARACTERÍSTICAS DE MULTIPROCESADOR:

OnLine ha implementado un proceso en paralelo de ordenamiento de paquetes y pruebas de procedimientos.

ALTA VIABILIDAD:

Hace seguros la disponibilidad de los datos cuando y donde se necesiten.

OnLine implementa múltiples mecanismos de Red para restaurar datos por fallas de software o de Hardware del sistema.

MIRRORING:

Existen un par de unidades para el mirroring en el espacio de disco OnLine de igual tamaño el mirroring mantiene una copia completa de los datos primarios que transparente pueden tomar el lugar del disco primario se llegará a ocurrir una falla del Hardware.

RESTAURACION DE DATOS:

El usuario puede también restaurar fallas de hardware un archivo completo de las bases de datos y registros de todas las transacciones, ambos almacenados en cintas. Un archivo es copiado de los datos creados por Online en un tiempo específico. Otro elemento crítico para la restauración de datos es el mantenimiento de los registros de transacción de las bases de datos.

Las transacciones lógicas son almacenadas completamente en localidades especiales situadas en algún lugar del disco llamado Logical.log

El Logical.log contiene el registro de toda la actividad hasta el último instante antes de la caída del Sistema. Si alguna falla ocurre el usuario puede restaurar sus transacciones fácilmente.

CONSISTENCIA DE LOS DATOS:

Mientras asegura un alta disponibilidad integridad y disponibilidad de los datos el nivel del Sistema asegura consistencia en los datos y a nivel de transacciones.

PROCEDIMIENTOS DE ALMACENAMIENTO:

Los procedimientos de almacenamiento son comandos SQL y programas con rutinas que son almacenados en las bases de datos con un nombre de procedimientos para asegurar una implementación consistente de operaciones comúnmente utilizadas.

SEGURIDAD:

OnLine provee de dos niveles de privilegios de acceso para asegurar la base de datos. Los privilegios de la base de datos controlan el acceso a la base de datos y los privilegios para crear tablas e índices en la base de datos los privilegios de una tabla especifican las operaciones que un usuario puede realizarle a esa tabla.

ESPECIFICACIONES PARA INFORMIX**TABLA DE CAPACIDADES(Basado en tamaño de pagina de 2 Kilobytes)**

Máximo número de registros por tabla	4,278 billones
Máximo número de paginas de datos por tabla	16,775 millones
Máximo tamaño por tabla(sin blobs)	33,822 billones bytes
Máxima longitud por registro	32,767 bytes
Máximo número de columnas por tabla	32,767 columnas
Máximo número de columnas especiales (blobs)	241blobs
Máximo número de índices por tabla	96 índices
Máximo número de columnas por índice	16 columnas
Máximo tamaño del índice	255 bytes

CAPACIDAD DE CAMPOS

Máximo Tamaño de un campo tipo caracter	32,767 bytes
Máxima longitud variable de un campo caracter	255 bytes
Máximo tamaño de un string con quote	255 bytes
Máximo tamaño de un texto o campo byte	2 gigabytes

CAPACIDADES DE ACCESO

Máximo número de bases definidas por on_line	100 bases
Máximo número de tablas definidas por on_line	14,049 millones
Máximo número de usuarios activos por on_line	1,000 usuarios
Máximo número de tablas abiertas activas	32,000
Máximo número de tablas unidas	máx. núm tablas abiertas actualmente
Máximo número de candados por sistema on_line	256,000
Máximo número de buffers de datos	32,000 buffers
Máximo número de manejadores on_line por equipo	255 manejadores

Máximo número accesible de BD's distribuidas	26 índices
Máximo número de mapeo recursivo	16 columnas
Máximo número de tablas con seguro	32 tablas por usuario

RECURSOS REQUERIDOS**MEMORIA**

Memoria para los procesos del servidor On_line	1,130 kilobytes
Espacio requerido para direccionamiento vituales por usuario	80 kilobytes
Memoria compartida para cada manejador On_line	1-64 megabytes
Memoria para los procesos demon de On_line	393 kilobytes

DISCO

Espacio en Disco requerido por On_line	5 megabytes
---	--------------------



CAPITULO III

DISEÑO Y DESARROLLO DEL MODELO ELEGIDO



III DISEÑO Y DESARROLLO DEL SISTEMA

III.1 ELABORACIÓN DEL MODELO CONCEPTUAL

Se propone rediseñar, construir e implantar el sistema en un equipo NCR 3000 perteneciente a la Coordinación Ejecutiva de Distribución utilizando al administrador de base de datos Informix y su lenguaje de 4a. generación para el desarrollo.

Este desarrollo será realizado a nivel central por la superintendencia general de documentación y sistemas con la colaboración de la gerencia de planeación comercial, y el sistema será instalado en los centros pagadores foraneos, únicamente a nivel de ejecución y consulta, lo que permitirá mantener una integridad de los pagos, un mantenimiento uniforme en cada centro y la posibilidad de atender localmente los requerimientos de información.

La actualización de los catálogos seguirá efectuándose vía cinta, mientras se dispone de una red de comunicación que permita el enlace entre los diferentes equipos de computo.

A mediano plazo, se contempla la interacción formal con el S.A.T. a fin de utilizar parte de la información que genera este sistema como entrada al sistema de pago de autotanques, evitando con ello la duplicidad de captura de los datos de una nota de embarque.

Se aplicarán técnicas de diseño y programación estructurada, mediante el uso de herramientas tipo C.A.S.E. que permitirá entre otras ventajas el abatimiento de los costos de mantenimiento.

Por otra parte, se pretende establecer una infraestructura de cómputo propia y homogénea para los centros de pagos, acorde a las necesidades actuales y futuras de la Coordinación Ejecutiva de Distribución, para ello primeramente se requiere sustituir el equipo de captura XL-40, por un equipo NCR 3000 cuyas características sean similares al equipo recientemente aprobado por el comité de adquisiciones, para el área de distribución y específicamente para sus centros embarcadores. Con lo anterior, se tendrá una sola versión del sistema y se podrán alcanzar altos niveles de conectividad entre los propios equipos de la C.E.D. y C.E.C.I.

Ver Esquema Conceptual del Sistema Pago de Fletes a Transportistas Fig.III.1.1

III.1.2 CARACTERISTICAS GENERALES DEL SISTEMA

- a) Sistema orientado a usuarios finales.
- b) Sistema desarrollado con tecnología de bases de datos y lenguaje de 4a. generación (INFORMIX ON- LINE 4GL)
- c) Captura y proceso de documentos fuente en lote y actualización en línea de los catálogos.
- d) Mecanismos de seguridad en el acceso del sistema.
- e) Menús Dinámicos en base al perfil de acceso del usuario.
- f) Facilidad para Auditoria de los movimientos realizados a las tablas del sistema.
- g) Ayuda en línea de acuerdo al contexto.
- h) Generación de Informes y Consultas en forma Dinámica

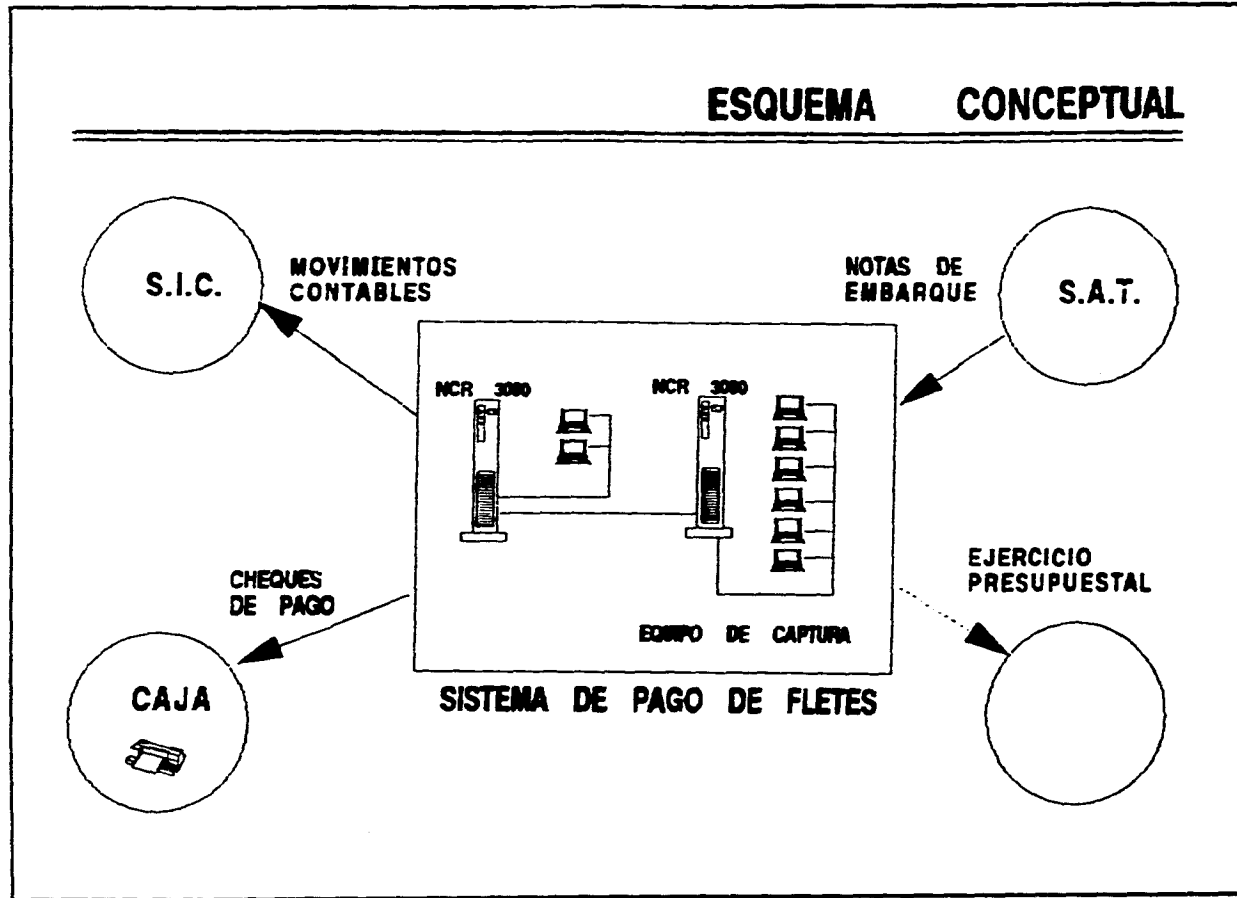


Fig. III. 1.1.1 Esquema Conceptual del Sistema

III.2 EXPLOTACION DE LA HERRAMIENTA IEW**III.2 .1 DISEÑO****METODOLOGIA DEL DISEÑO DEL SISTEMA**

Como se explicó en el capítulo I, existen diversas metodologías que nos brindan las capacidades y facilidades para poder realizar las diferentes fases del desarrollo de un sistema y aunque no todas llenan un esquema completo, es decir que cubren el desarrollo de todas las fases que lo integran, si nos ayudan en la mayoría de ellas y aunque tienen sus diferencias y puntos débiles; todas ellas nos proponen un proceso de desarrollo paso a paso , de manera estructurada y con un enfoque que promueve la normalización dentro del desarrollo de sistemas.

Los principios de la ingeniería de información, establecen un entrelazamiento entre diversas metodologías, un conjunto de técnicas formales rigurosa y altamente integradas, agrupadas en un ambiente bien definido y manipulable que permiten realizar la planeación , el análisis, el diseño y la construcción de un sistema en una empresa.

Estas metodologías, se apoyan en modelos de datos normalizados que se mantienen dentro de un ambiente sistematizado, utilizando herramientas computarizadas y así evitar el desarrollo de problemas desordenados con un mantenimiento convencional.

Esto es posible debido a que los bloques que la conforman, se pueden ensamblar en diferentes modos dependiendo del planteamiento de los diferentes analistas.

Como se puede observar en el siguiente diagrama de bloques donde se muestra la independencia y versatilidad que se tiene para llevar a cabo el desarrollo de un sistema .

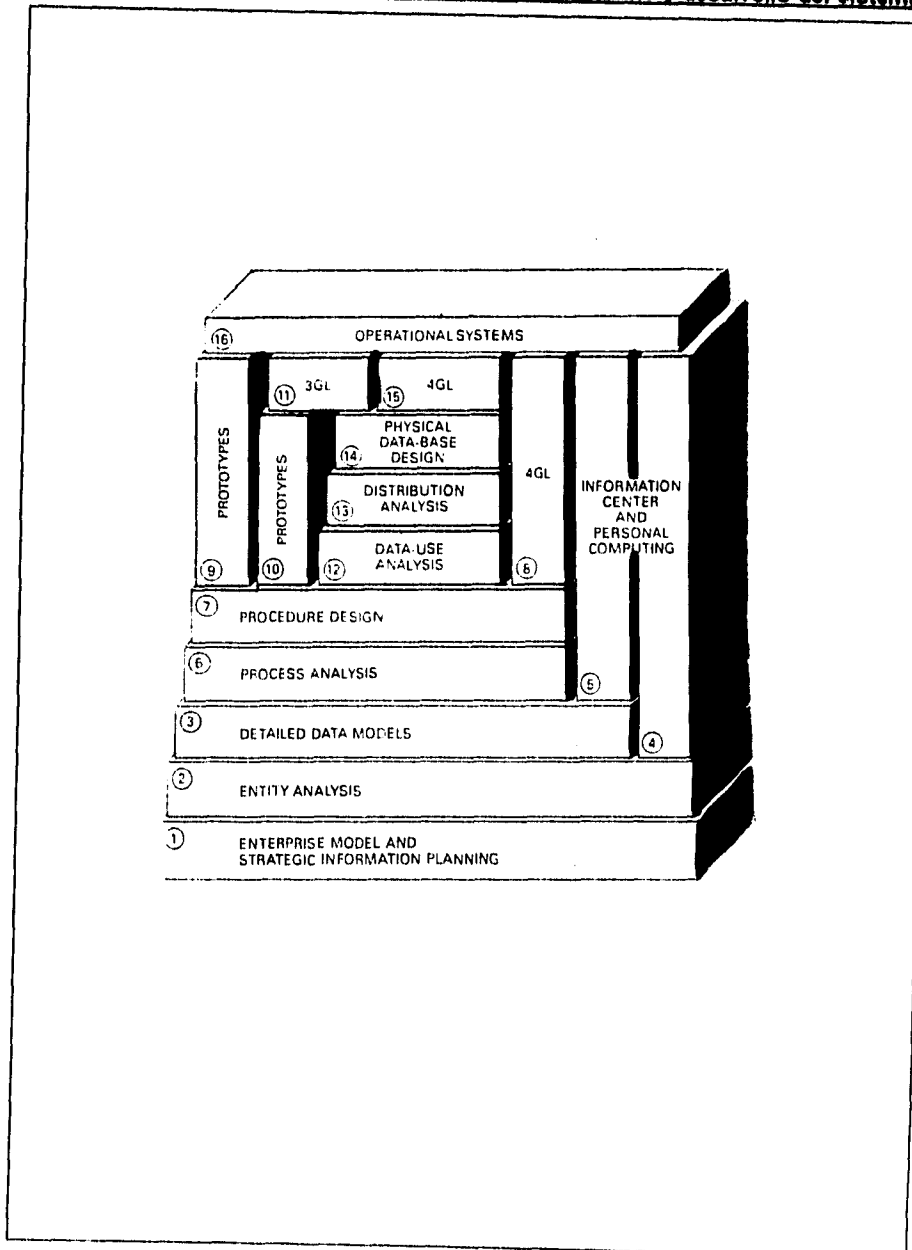


Fig.III.2.1.1 Conjunto de Bloques que soportan la Metodología de desarrollo utilizada

Algunas otras características de la metodología utilizada son:

- Ayudar a obtener lo máximos beneficios de la tecnología de la información.
- La base de su arquitectura permite la realización de sistemas integrados.
- Asegura la generación de proyectos de tamaño de fácil manipulación
- Optimiza el uso de comunicación gráfica.
- Incorpora la coordinación del desarrollo, mediante el manejador de control de funciones, que coordina todos los aspectos del desarrollo.

Por las razones anteriores se decidió llevar a cabo el diseño y desarrollo del sistema, siguiendo las fases ó etapas propuestas y descritas anteriormente por la Metodología de la ingeniería de la información, adoptada por James Martin y la cual se compone por las siguientes etapas en forma global:

- 1.- Planificación de la estrategia de la información
- 2.- Análisis del área del proyecto.
- 3.- El diseño del sistema.
- 4.- La Construcción del sistema.

Los objetivos fundamentales de esta son de comprender los objetivos principales y estratégicos del sistema, para poder desarrollar sistemas integrados que darán un soporte consistente a la empresa.

Aunque explícitamente hasta este punto se hace una descripción de la metodología utilizada para el desarrollo del sistema, cabe mencionar que ésta se implemento desde la etapa del análisis, no se partió desde la planeación ya que el objetivo y las necesidades del nuevo sistema ya se tenían bien definidos por el área usuaria.

ETAPA 1

Conforme a la metodología elegida, se realizó en primera instancia de acuerdo al bloque 1, el Diagrama jerárquico (Diagrama de Descomposición) de las áreas de la empresa que están involucradas con el sistema. Esto, con el propósito de mostrar y entender el entorno organizacional del sistema Pago de Fletes "Carga Liquida", así como la estructura orgánica en la cual se ubica el desarrollo y la operación de dicho sistema.

El diagrama de descomposición es una construcción jerárquica en la cual, actividades, áreas y unidades organizacionales se despliegan en niveles sucesivos de detalle y donde se pueden documentar sus funciones principales.

De ésta etapa se generó el Diagrama de Descomposición de la Estructura Organizacional Ver Fig.III.2..1.2

ETAPA 2

Después de acuerdo al bloque 2, un esbozo del mapa de información, mediante el análisis de la información se detectaron los tipos de datos y la manera en como se van a relacionar unos con otros. El análisis de información permite crear un modelado de información que da una vista amplia de las necesidades del sistema, sin llegar al detalle lógico del diseño de la base de datos y permite crear un modelo de información de la manera más estable posible, antes de que se implemente físicamente.

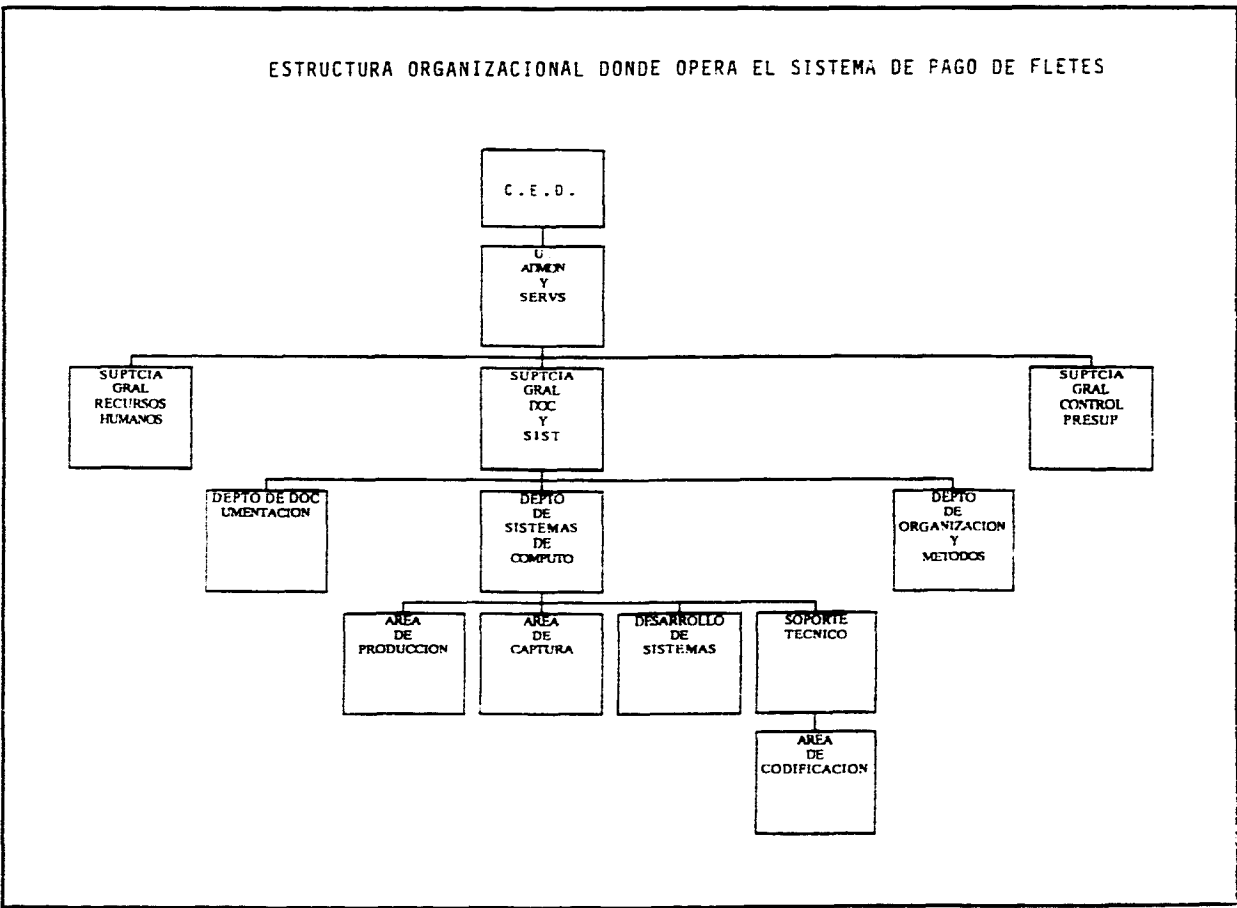


Fig. III.2.1.2 Estructura Organizacional donde opera el SPF_AT generado por IEW

Una capacidad importante de la ingeniería de información, es que los datos que existen en una organización pueden ser descritos independientemente de la manera en que esos datos serán utilizados, más aún sin importar la estructura de datos que se necesita implementar. Esto se debe a que los datos cuentan con propiedades inherentes que permiten una estructura correcta.

Esta etapa dió como resultado la generación del Diagrama Entidad/Relación el cual muestra los tipos de entidades y sus relaciones entre sí. Ver Fig.III.2.1.3

ETAPA 3

De acuerdo al tercer bloque, esta etapa es una extensión del bloque 2, pero a un nivel de mayor detalle y aplicando diversos chequeos buscando el mayor grado de confiabilidad . El Diagrama Entidad/Relación nos permite realizar el modelado de datos y una forma de identificar los tipos de entidades principales; así como las relaciones existentes entre ellas.

Este es utilizado principalmente para la planeación top-down y para la generación de Modelos de Datos detallados y a diferencia del modelo de información, contiene todos los detalles necesarios para la instrumentación de la base de datos.

De esta manera, el banco de información que soporta las políticas y procedimientos del Pago de fletes, está dado a través del siguiente Diagrama del Modelo de Datos , que muestra las entidades de información principales, así como sus relaciones mismas.

Ver Fig.III.2.1.4 Diagrama del Modelo de Datos.

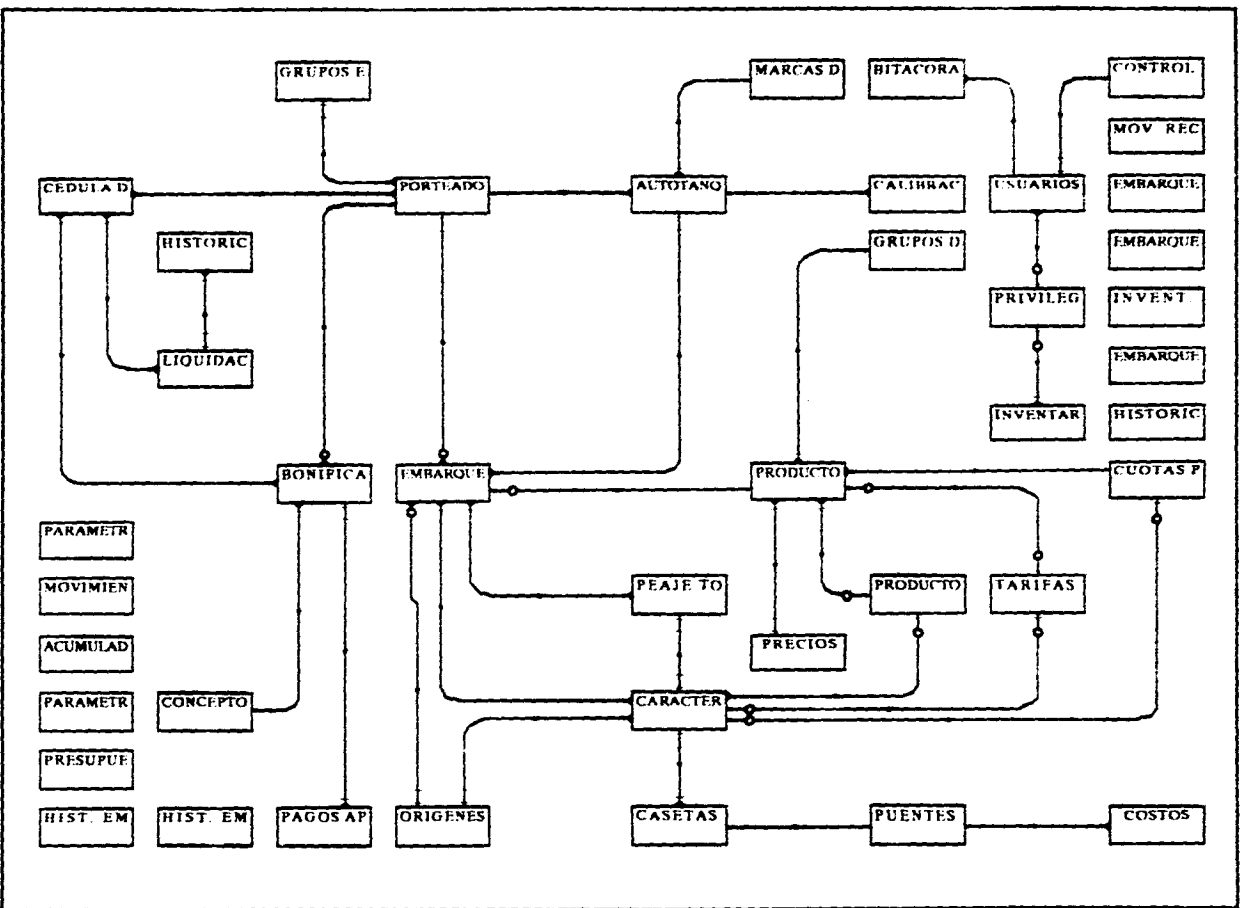


Fig III.2.1.3 Diagrama Entidad/Relación del SPF_AT generado por IEM

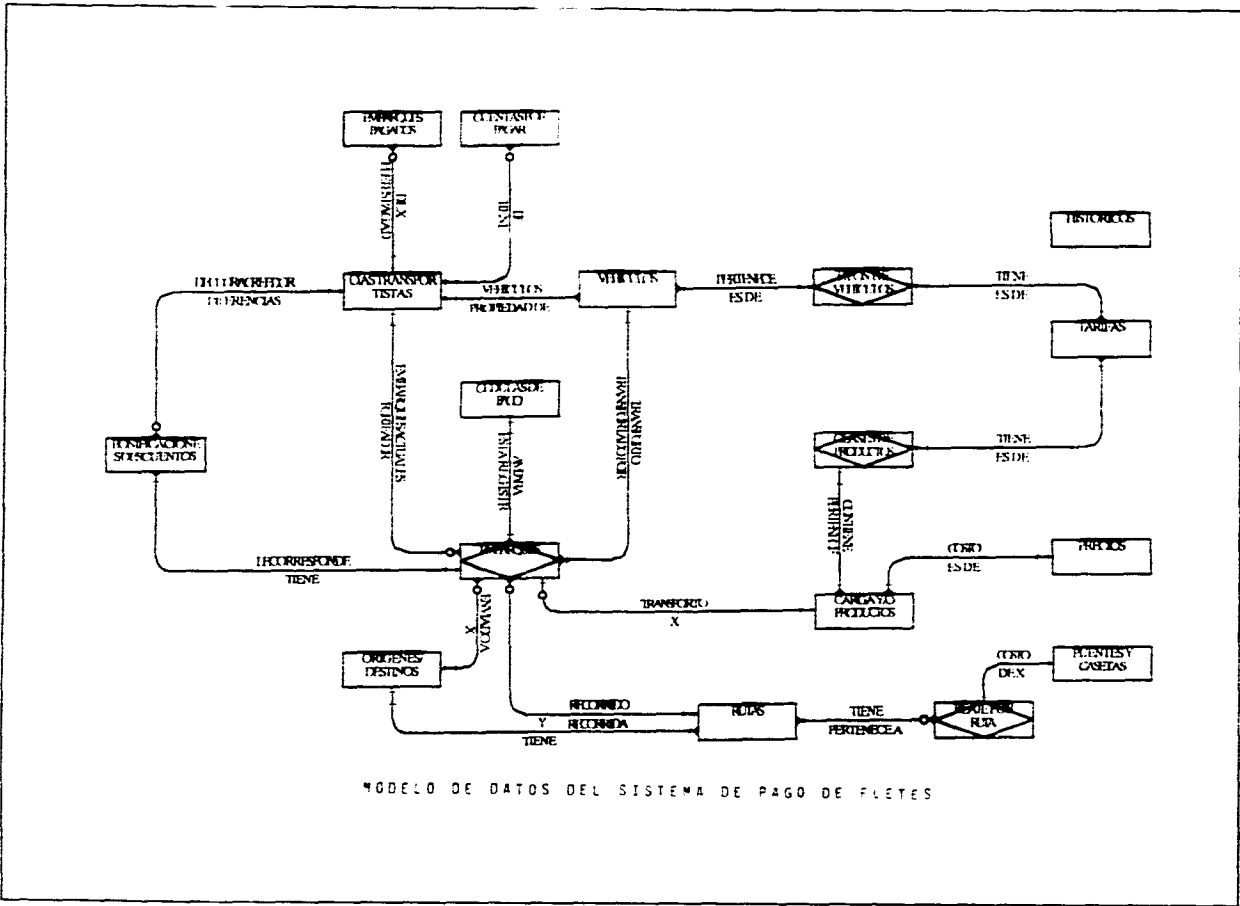


Fig III 2.1.4 Diagrama de Modelo de Datos del SPF_AT generado por IEW

ETAPA 4

Esta etapa se relaciona directamente con el análisis de sistemas. De acuerdo al bloque 6 el cual se refiere al análisis de las funciones de una área específica, permite que éstas sean descompuestas en procesos menores, utilizando diagramas de flujo de datos o de dependencia y pueden ser dibujados para mostrar la interacción de los diferentes procesos.

Una vez que los datos fueron determinados y sintetizados dentro de un modelo de datos totalmente normalizado, se realizó una revisión detallada del procedimiento del pago de fletes, para poder identificar el flujo de datos y los procesos involucrados.

Mediante la simbología de Yordon, los diagramas pretenden mostrar la secuencia lógica y los elementos de entrada y salida de los procesos involucrados en las distintas tareas de la operación del sistema, para ello, se presentan los niveles jerárquicos en los procesos, es decir, el primer diagrama muestra el diagrama de flujo de la operación general del sistema; los diagramas subsecuentes son la explotación a un nivel más detallado de cada uno de los procesos del diagrama general.

A continuación se presenta el Diagrama de Flujo de Datos General (Ver fig. III.2.1.5), así como los procesos más importantes que forman parte del Sistema de Pago de Fletes en forma de diagrama, de la misma manera una relación de estos procesos. Todo ello generado automáticamente por la herramienta IEW.

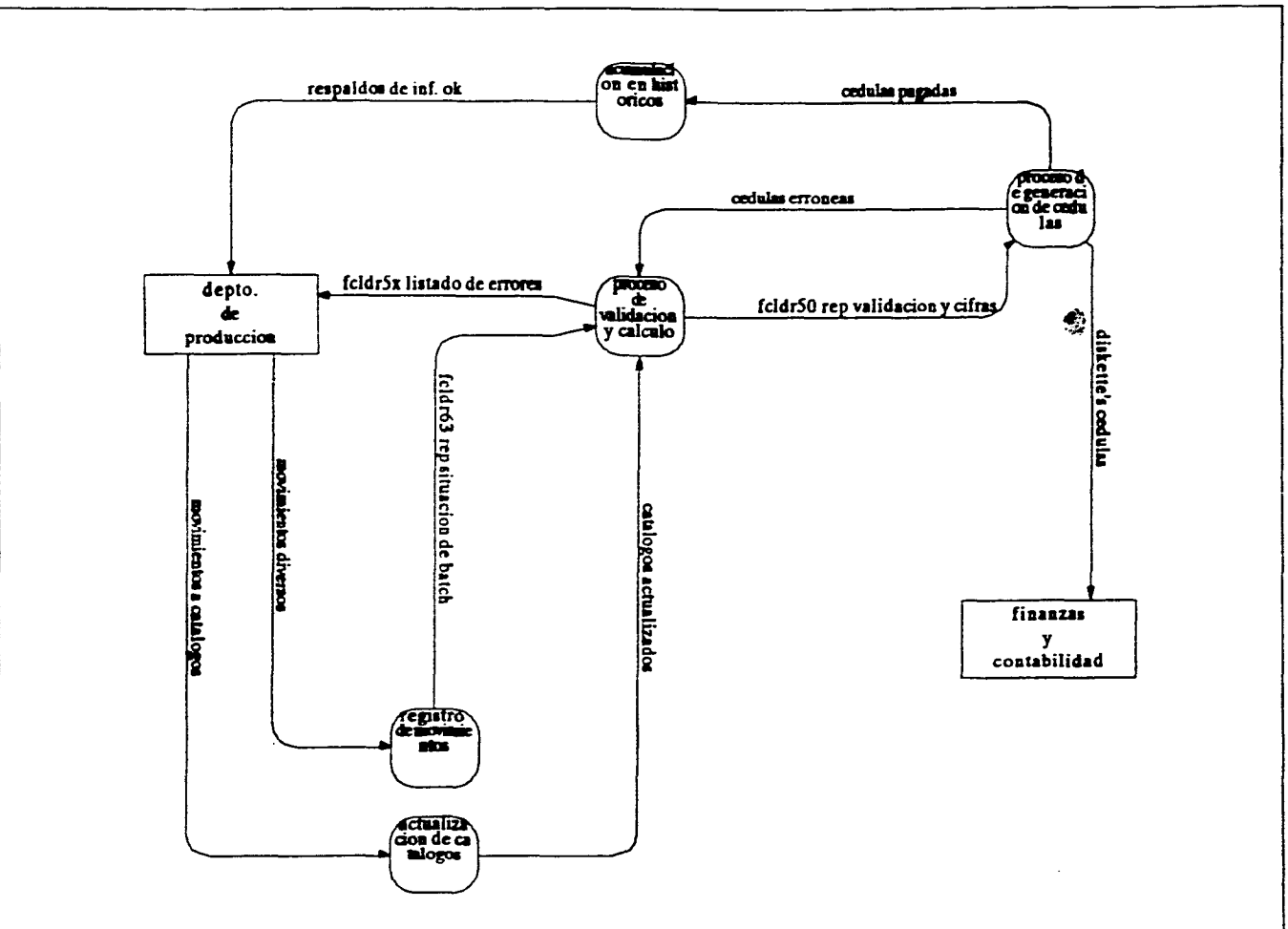


Fig. III.2.1.5 Diagrama de Flujo de Datos General del SPF_AT generado por IEW

Petroleos Mexicanos C.A.S.E

Object List

(Lista de Objetos Tipo Proceso Generada Por IEW)

process actualiza #'s embarques pagados
 process actualiza autotanques
 process actualiza cuotas
 process actualiza embarques reportados
 process actualiza origenes/destinos
 process actualiza parametros contables
 process actualiza peaje total
 process actualiza porteadores
 process actualiza precios de productos
 process actualiza productos
 process actualiza rutas
 process actualiza tarifas especiales
 process actualizacion de catalogos
 process Acumula Embarques A Historico M.
 process Acumulacion En Historicos
 process Captura
 process Control Presupuestal
 process Ejecucion De Proceso Automatizad
 process Emision De Cheques
 process Emite Analitico De Peaje
 process Emite Analitico Por Liq
 process Emite Cedula Y Bon/Des
 process Emite Contrarecibos
 process Fclbg02 Genera Ascii Contab.
 process Fclfa01 Actualiza Porteadores
 process Fclfa03 Actualiza Autotanques
 process fclfa05 actualiza origen/destino
 process fclfa08 actualiza productos
 process fclfa09 actualiza cuotas
 process fclfa10 actualiza tarifas esp.
 process fclfa12 actualiza rutas
 process fclfa13 actualiza peaje total
 process fclfa17 actualiza precios
 process fclfa18 registro de bonif/desc
 process fclfa19 actualiza emb reportados
 process fclfa31 validacion y calculo
 process fclfa35 actualiza Num embarque
 process fclfa37 actualizacion de param.
 process fclfa40 transf embarq pag a hist
 process fclfa41 acumula embar a hist men
 process fclfa50 restaura cedula

Subgerencia de Distribucion

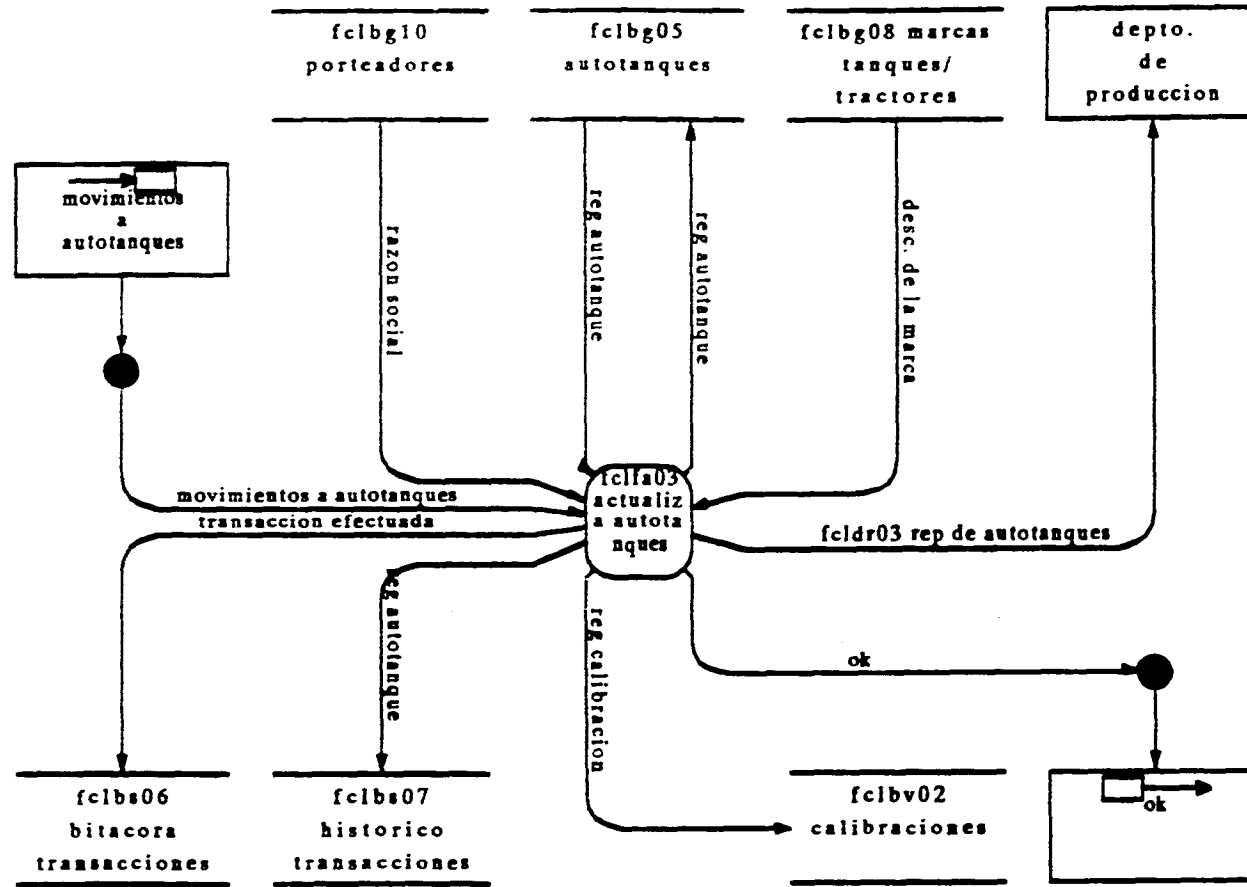
Petroleos Mexicanos C.A.S.E

Object List

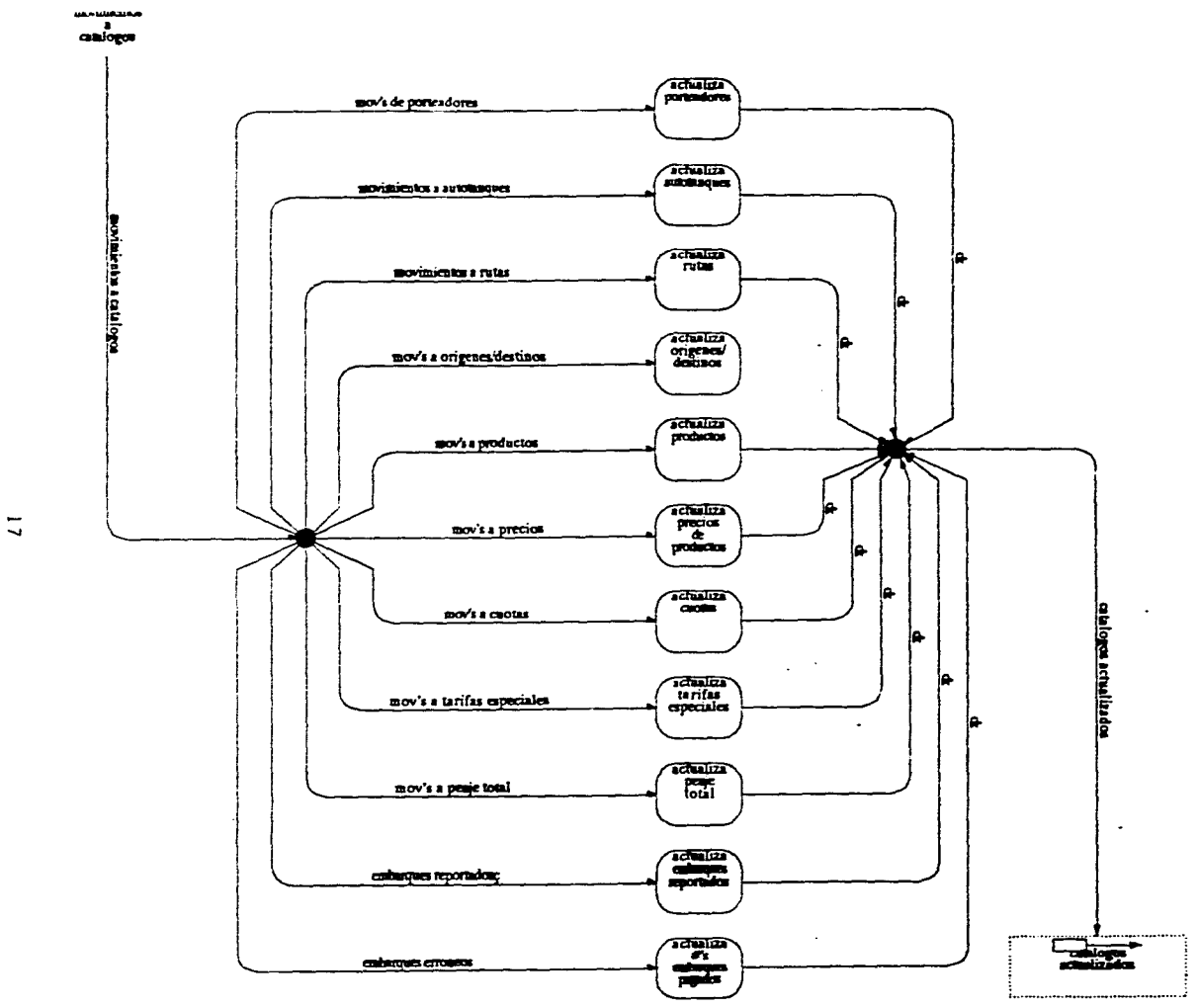
(Lista de Objetos Tipo Proceso Generada por IEW)

process fclfc63 genera rep. de batch's
process fclfg01 genera ascii para IBM
process fclfg02 genera ascii contab.
process fclfg03 genera ascii finanzas
process fclfg05 genera diskette's c. f.
process fclfg06 respaldo mov's diarios
process fclfk01 captura de embarques
process fclfk03 verificacion de batch's
process fclfk04 correcciones
process fclfm01 generacion de mecanizad
process fclfr52 emite analitico por liq.
process fclfr55 emite contrarecibos
process fclfr56 emite cedulas y bon/des
process fclfr57 emite analitico de peaje
process finanzas
process genera analitico de peaje
process genera analitico liquidaciones
process genera archivo de contabilidad
process genera archivo de finanzas
process genera cedulas
process genera cedulas y bonif/desc
process genera contrarecibos
process genera mecanizados
process genera validacion
process proceso de generacion de cedulas
process proceso de mecanizado definitivo
process proceso de validacion y calculo
process proceso general spf-at
process registro batch mecanizado
process registro de bonif/desc
process registro de movimientos
process respalda a diskette contab/finan
process respalda embarques pagados
process respalda movimientos diarios
process restaura cedulas
process transfiere embarques a historico
process validacion de datos

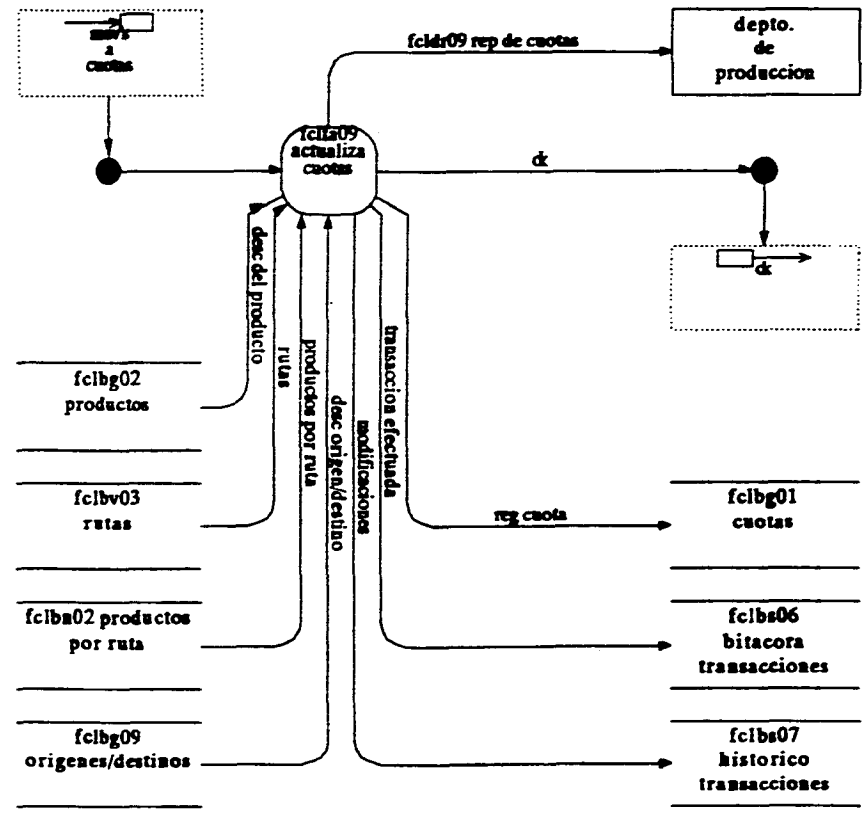
Subgerencia de Distribucion



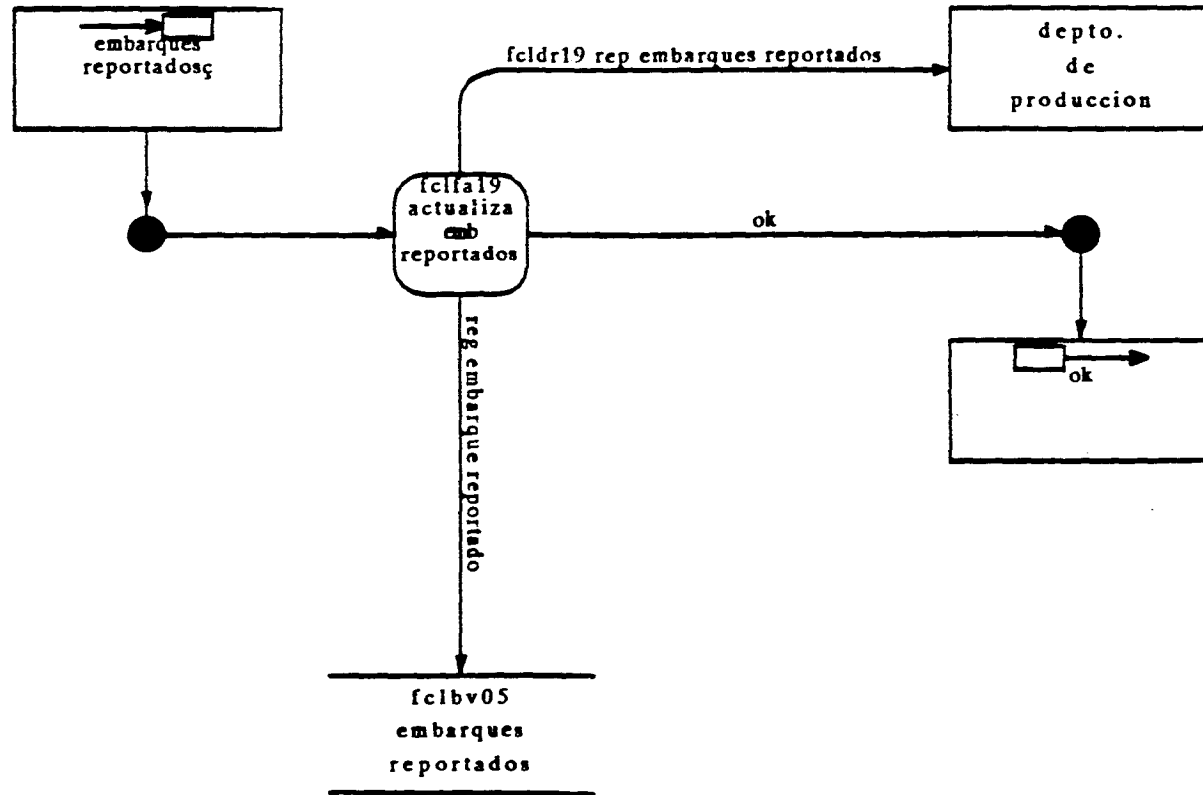
Actualiza catálogo de Autotánques



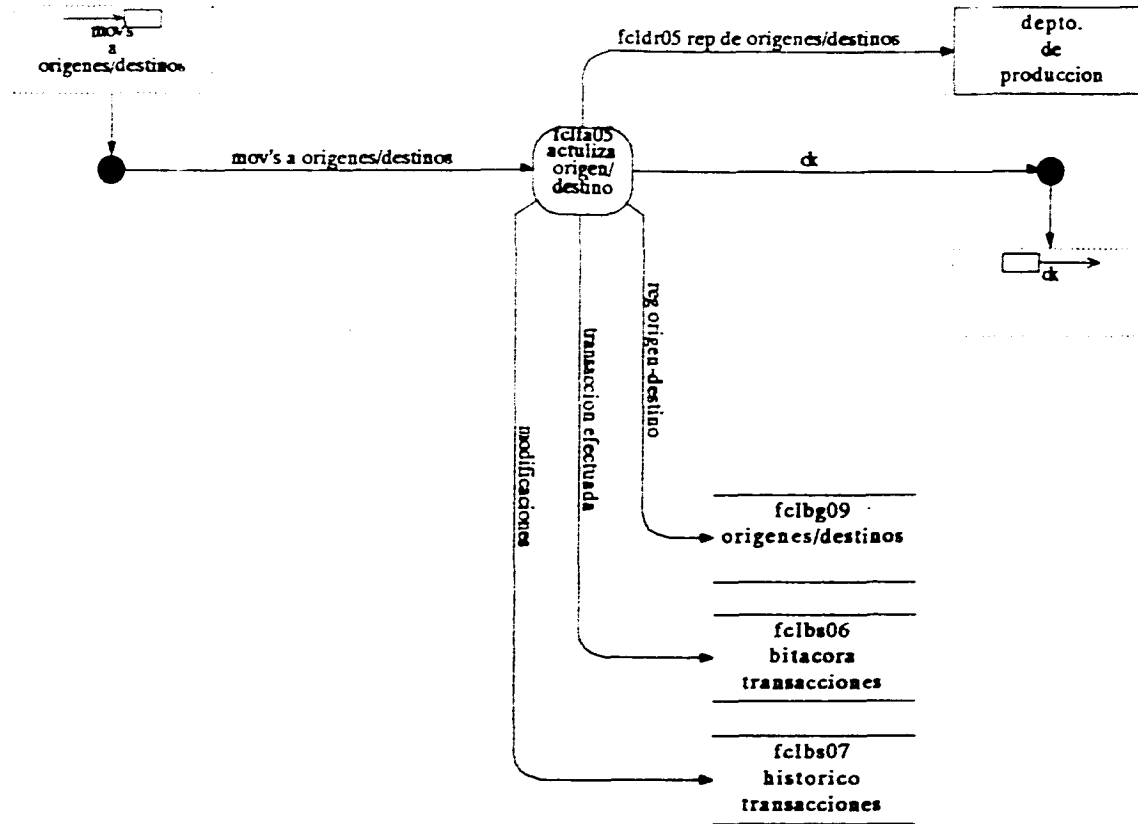
Actualización de Catálogos



Actualiza Catálogo de Cuotas

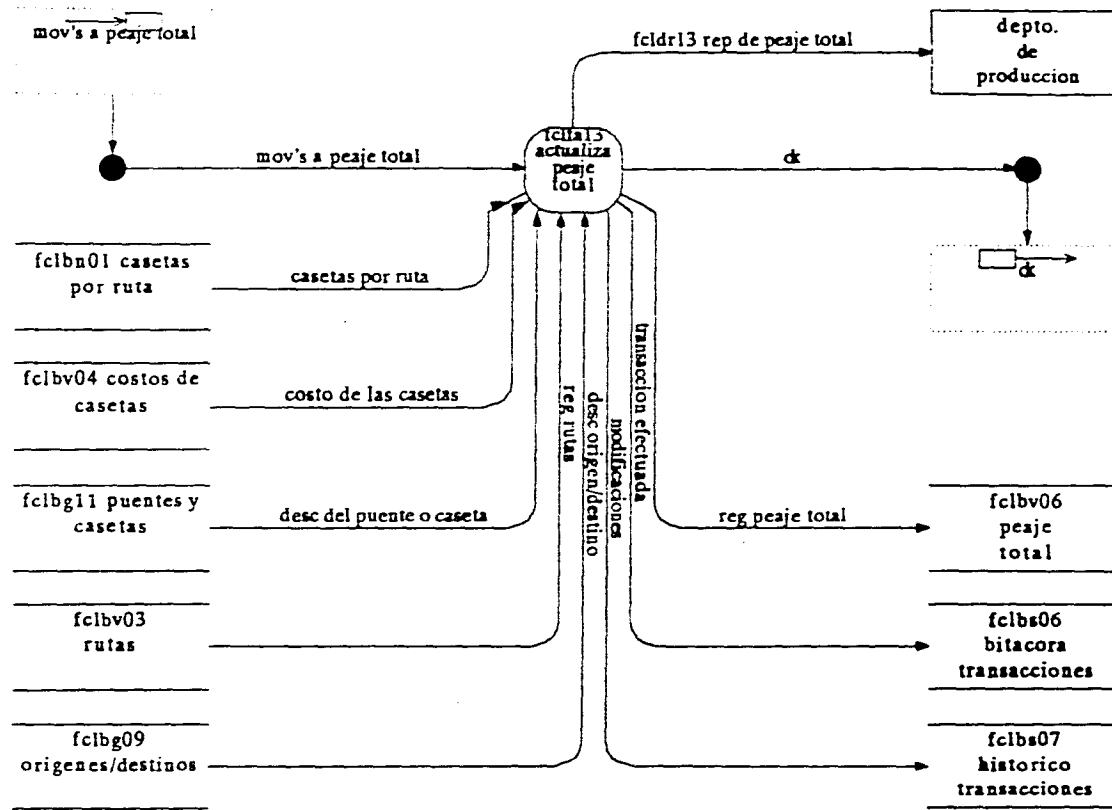


Actualiza archivo de Embarques Pagados

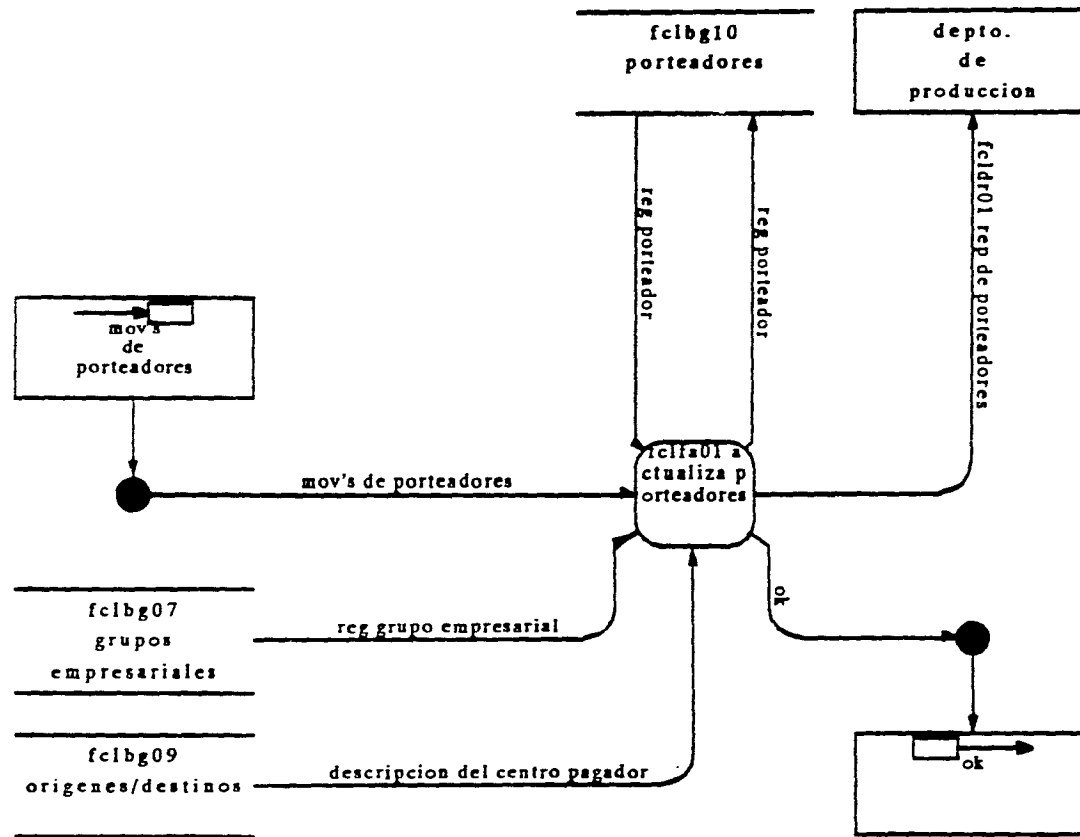


2.1

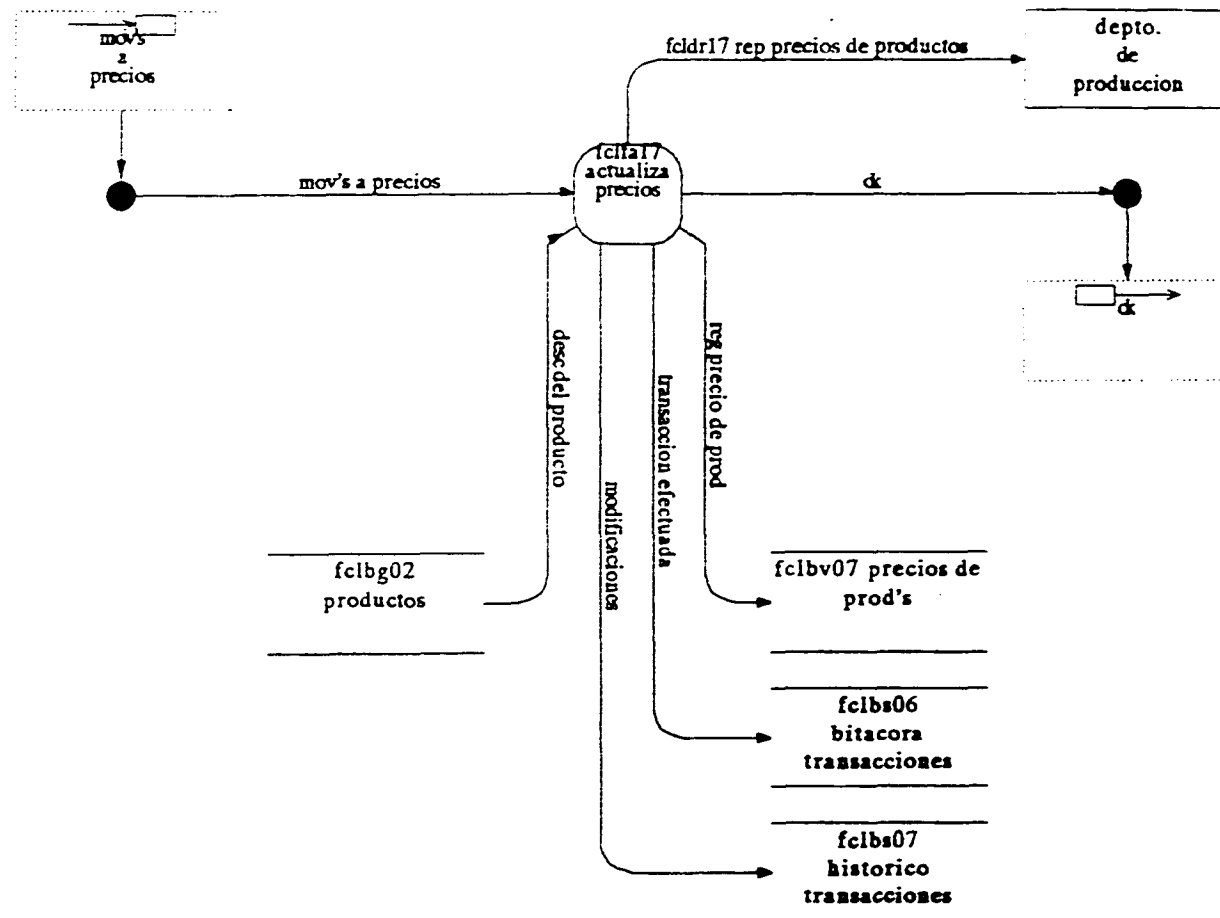
Actualización de Catálogo Origenes/Destinos



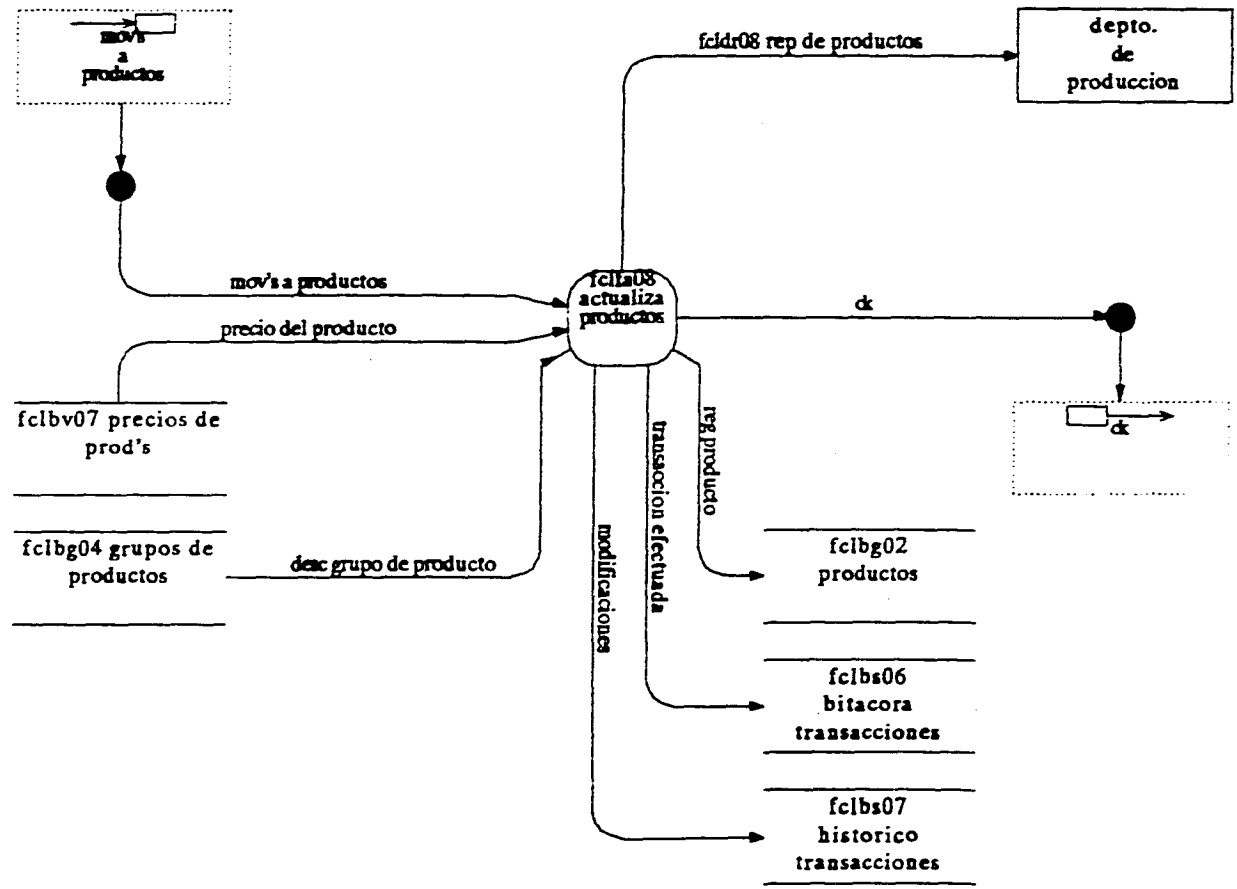
Actualización de Archivo de Peaje Total



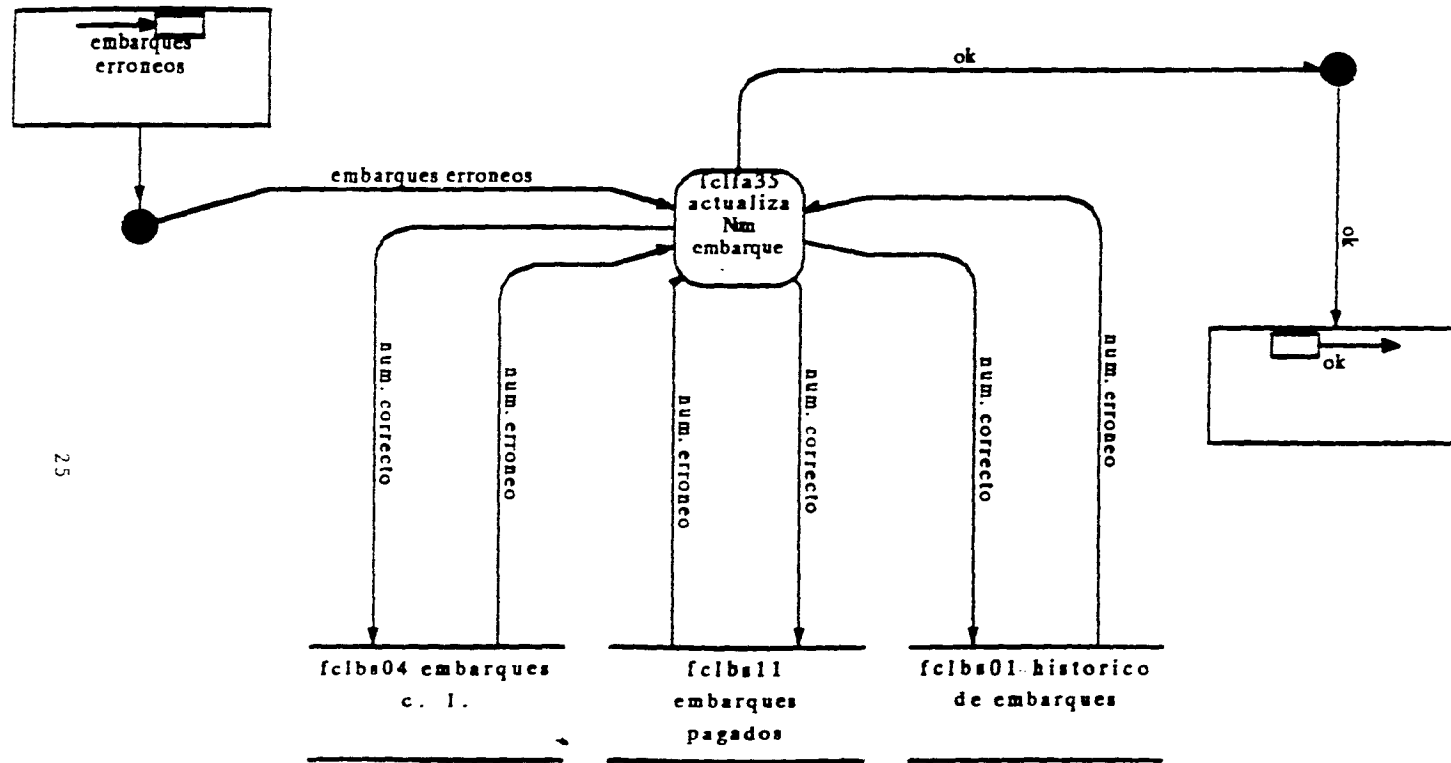
Actualización de Catálogo de Porteadores



Actualización de Catálogo de Precios

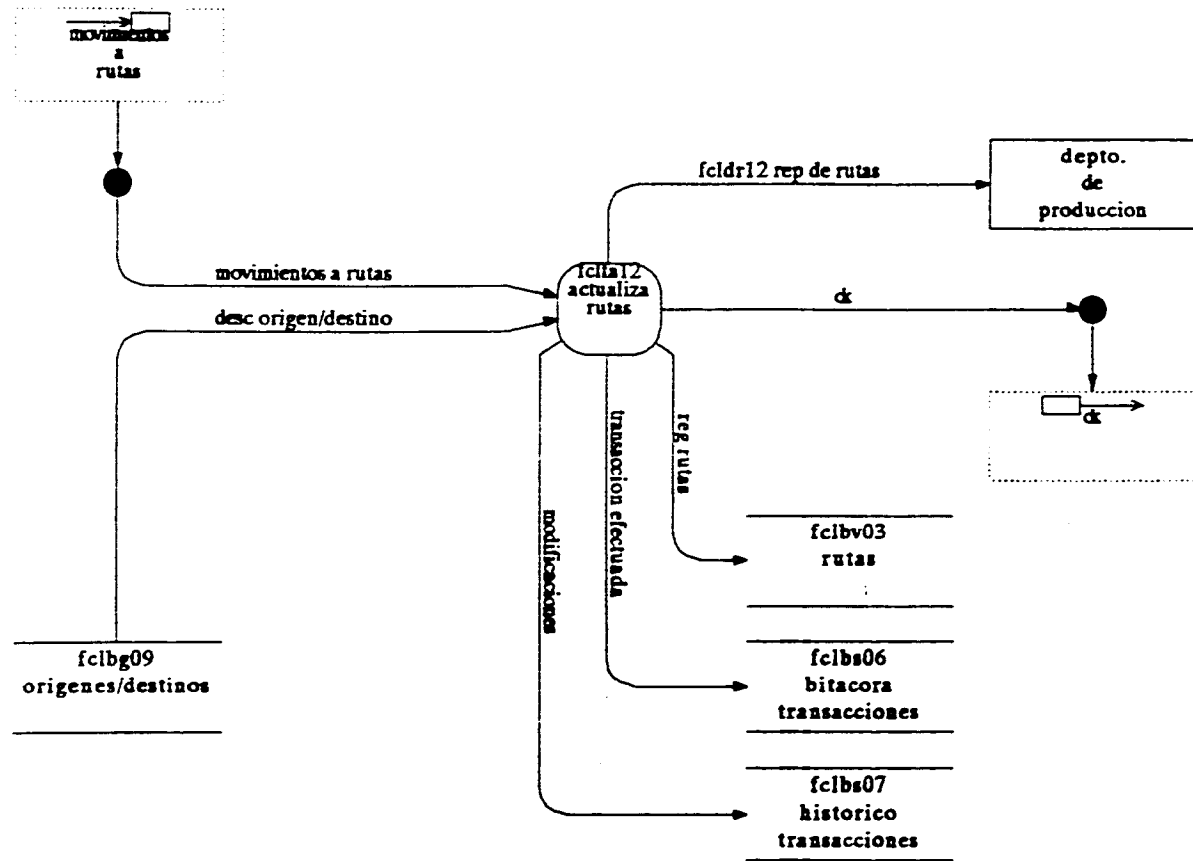


Actualización de Catálogo de Productos

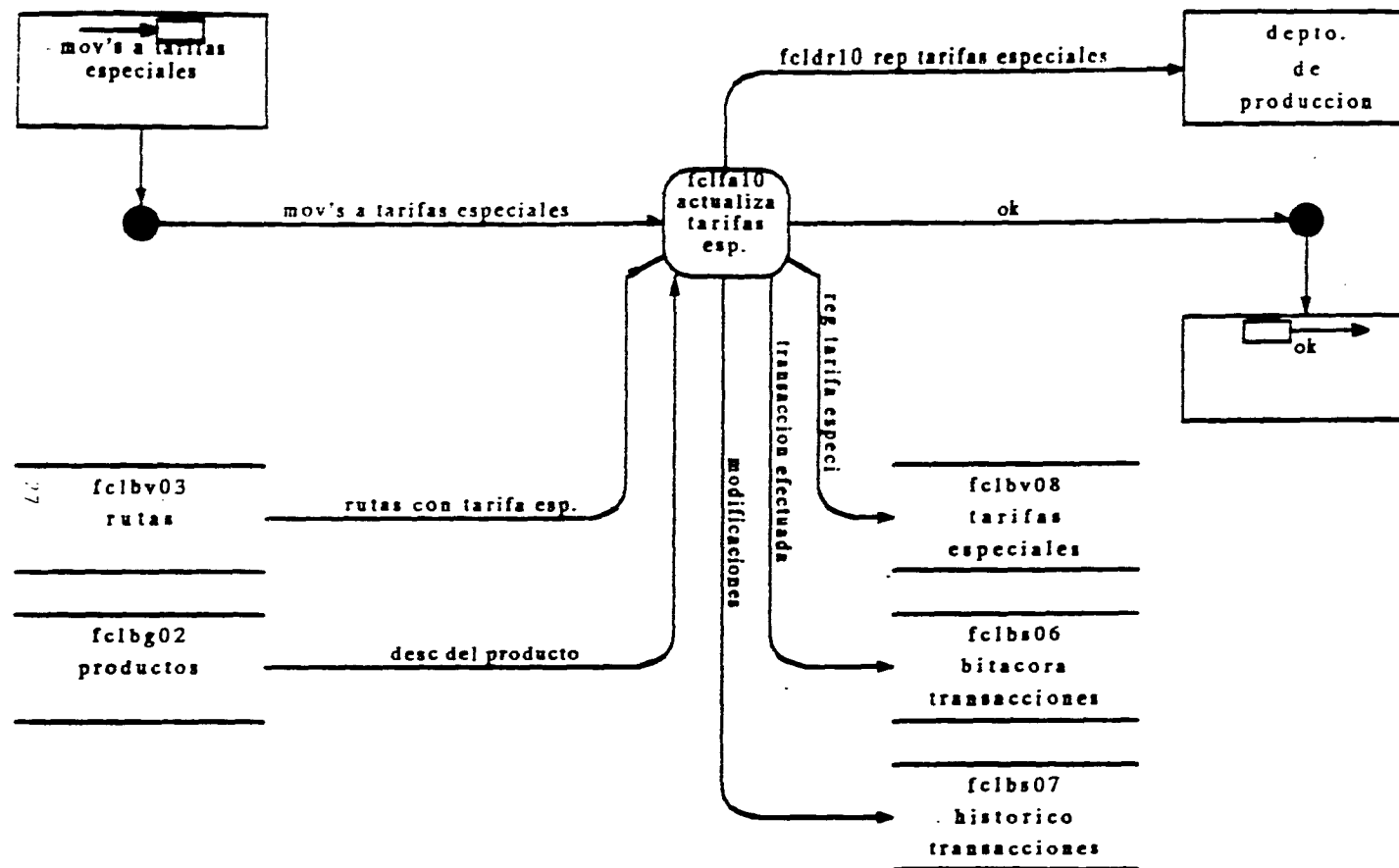


25

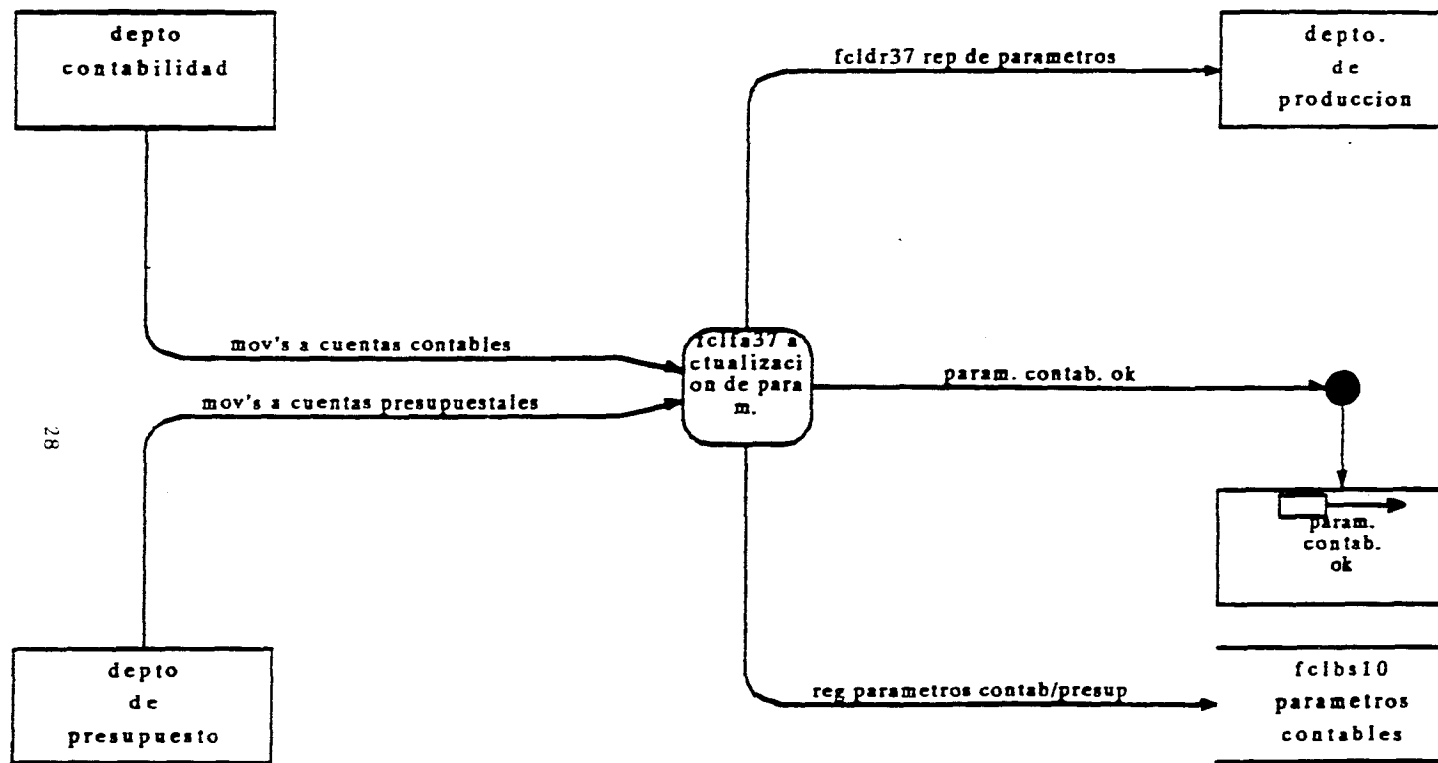
Actualización de Número de embarque



Actualización de Catálogo de Rutas

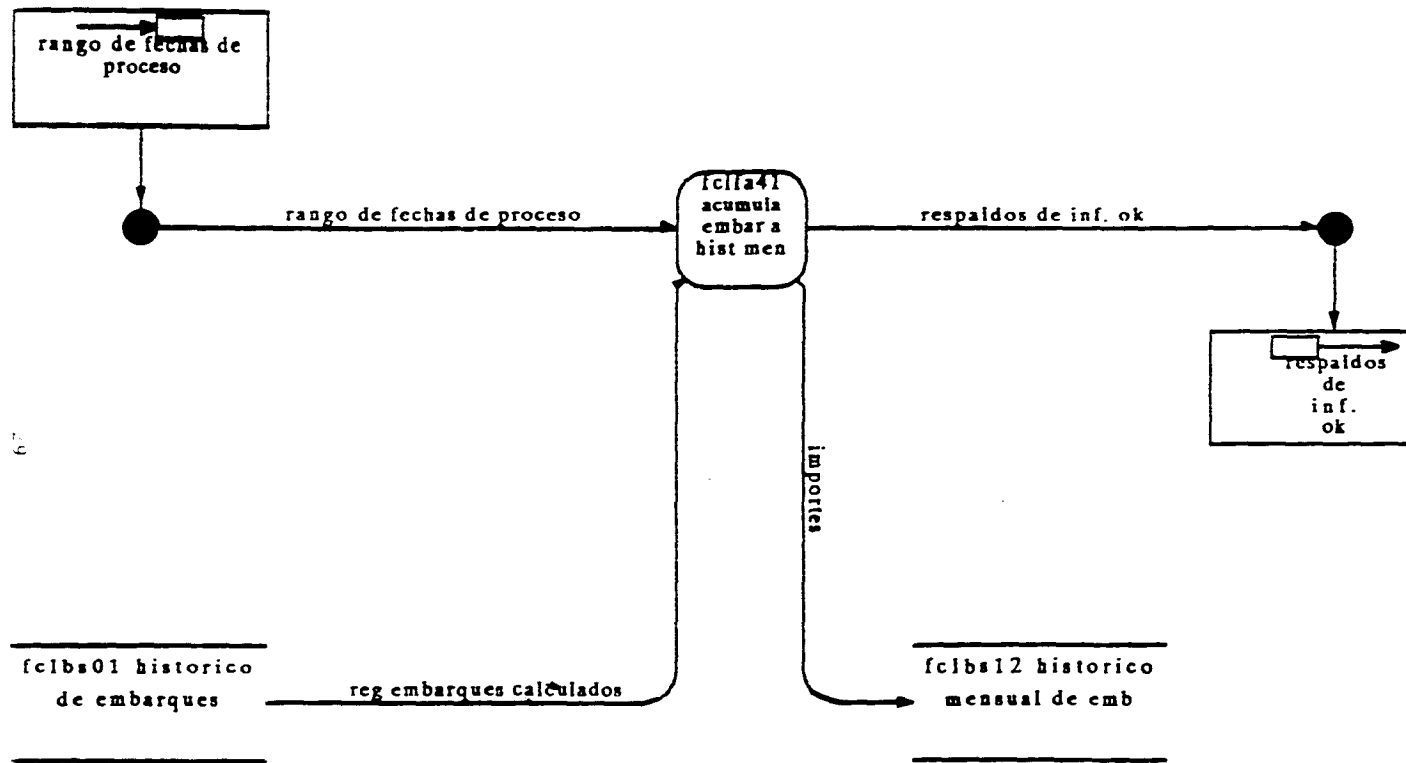


Actualización del Archivo de Tarifas Especiales

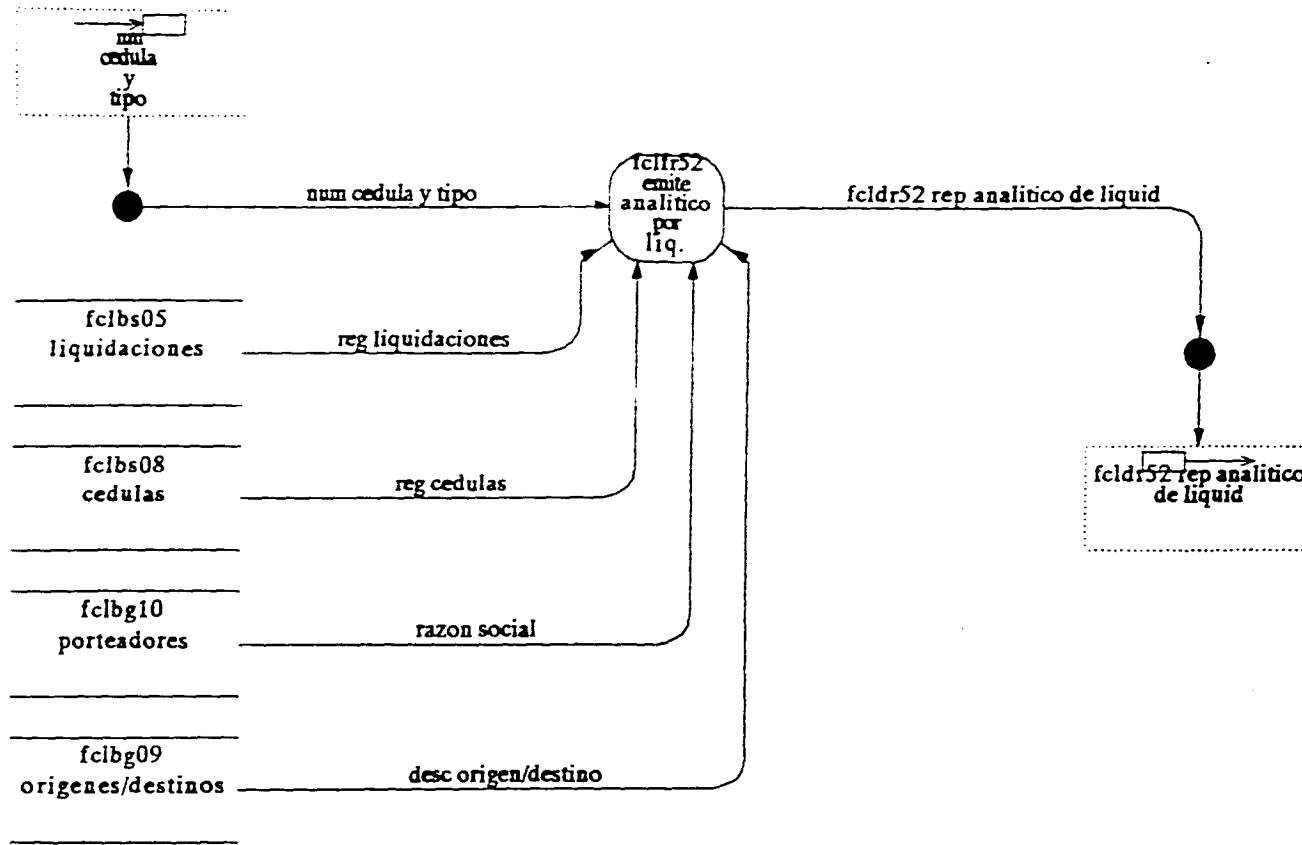


28

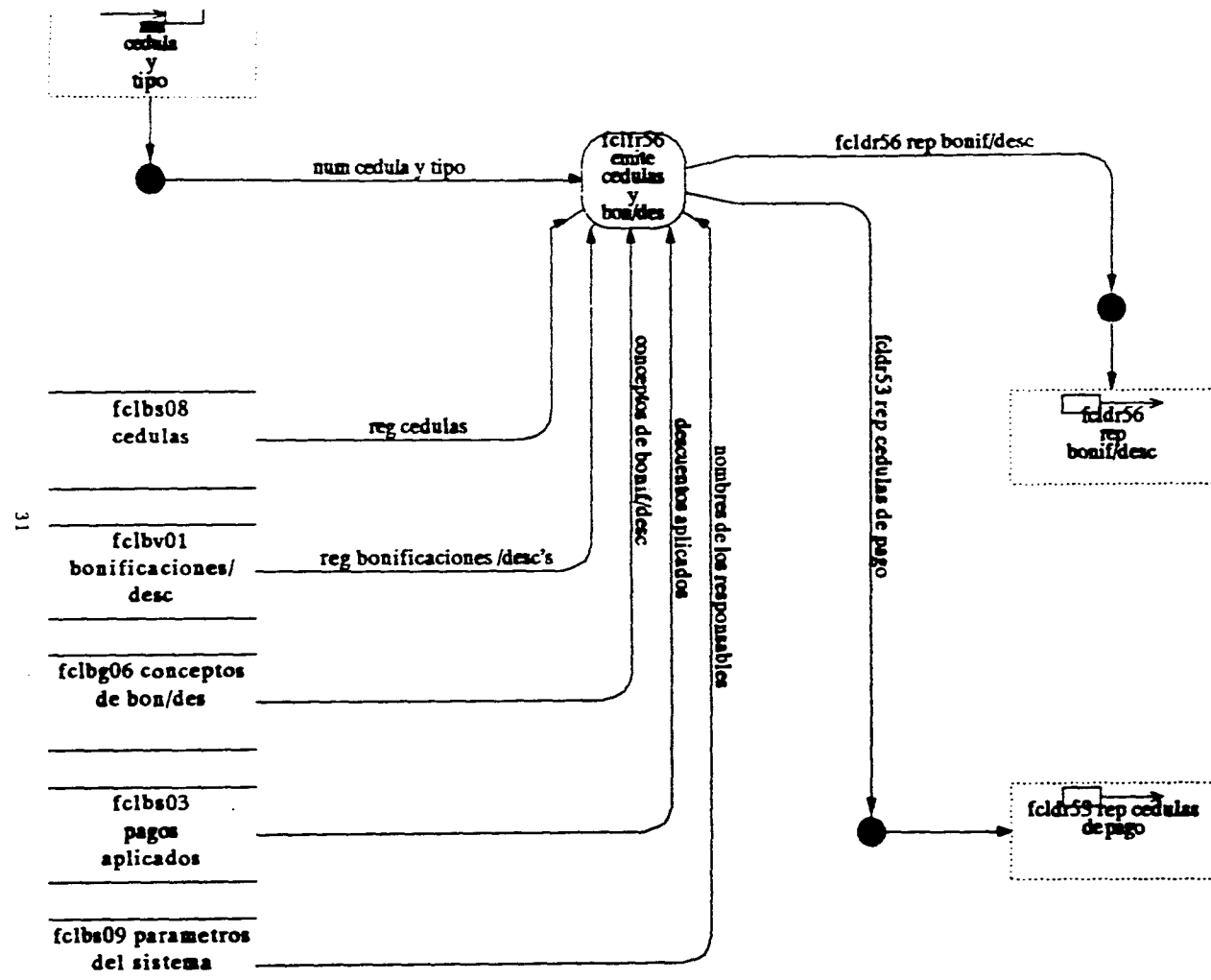
Actualización de Parámetros Contables



Acumula Embarques a Histórico Mensual

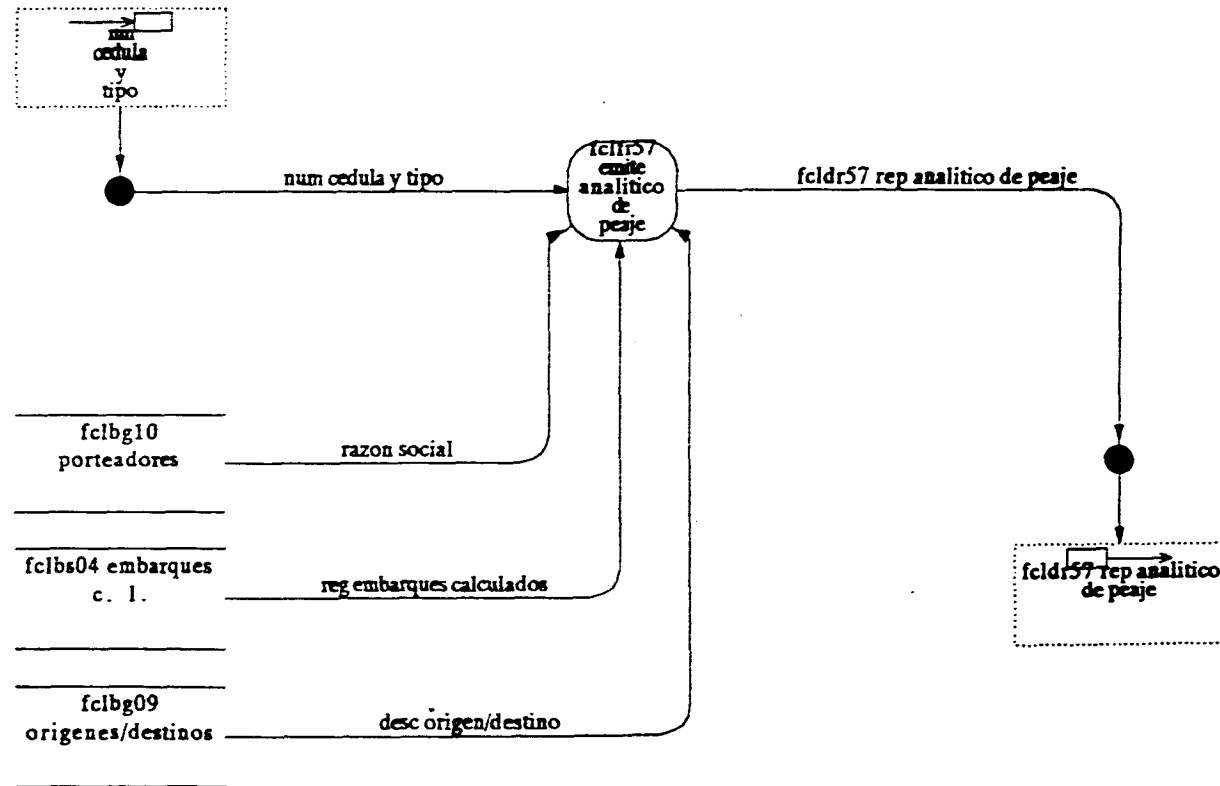


Emite Analítico por Liquidación

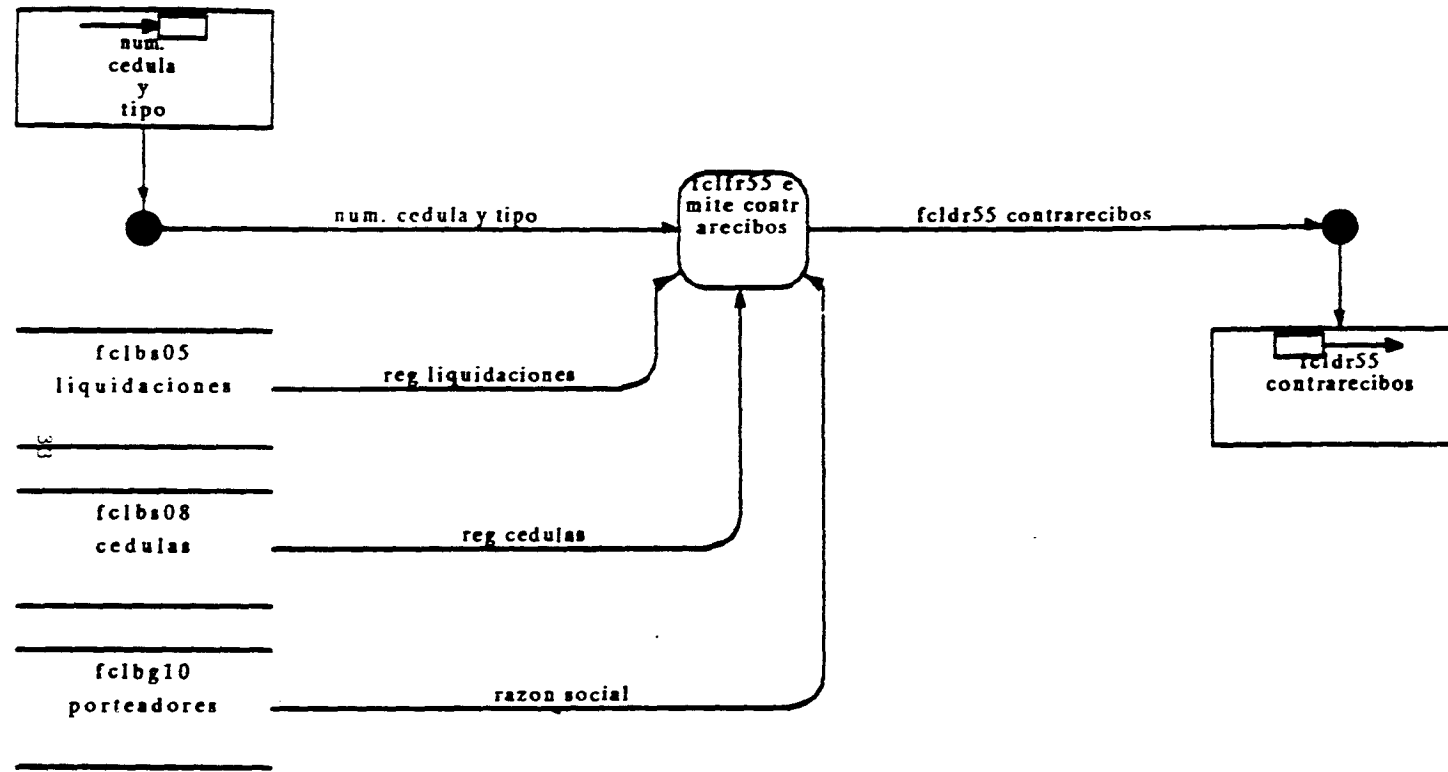


31

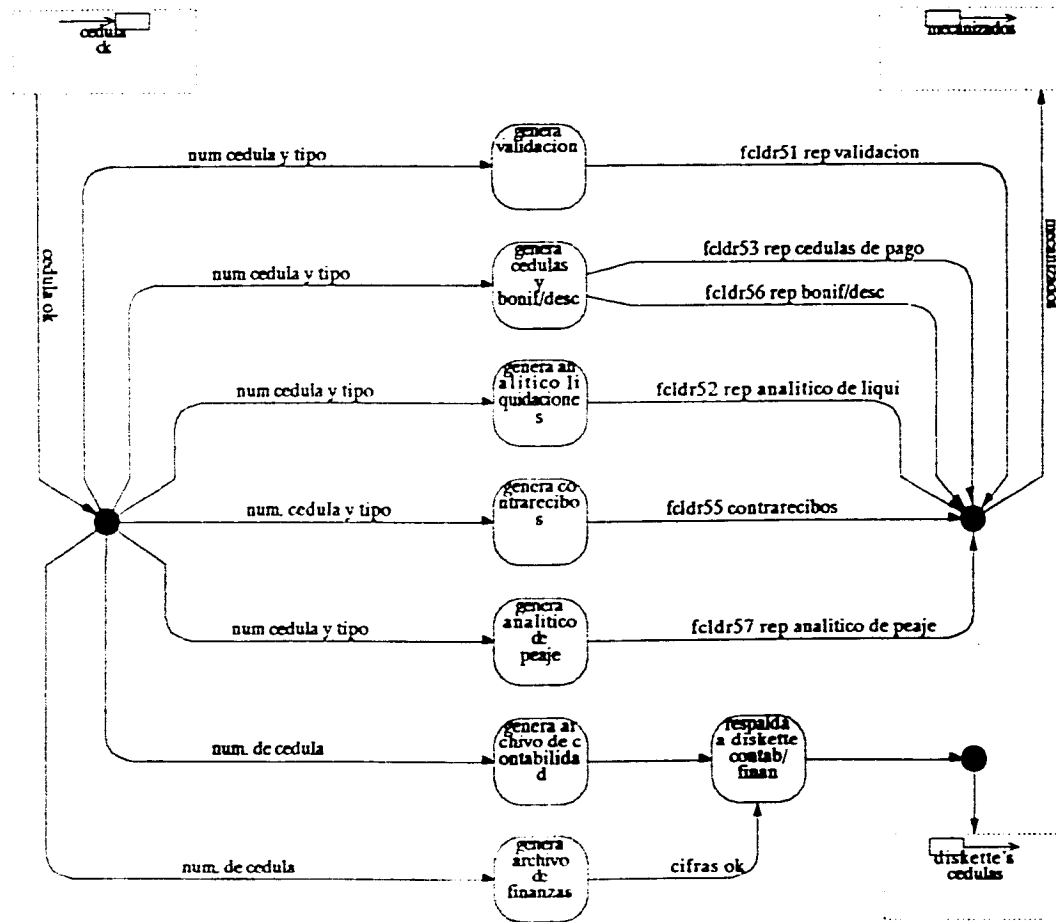
Emite Cédulas y Bonificaciones y Descuentos



Emite Analítico de Peaje

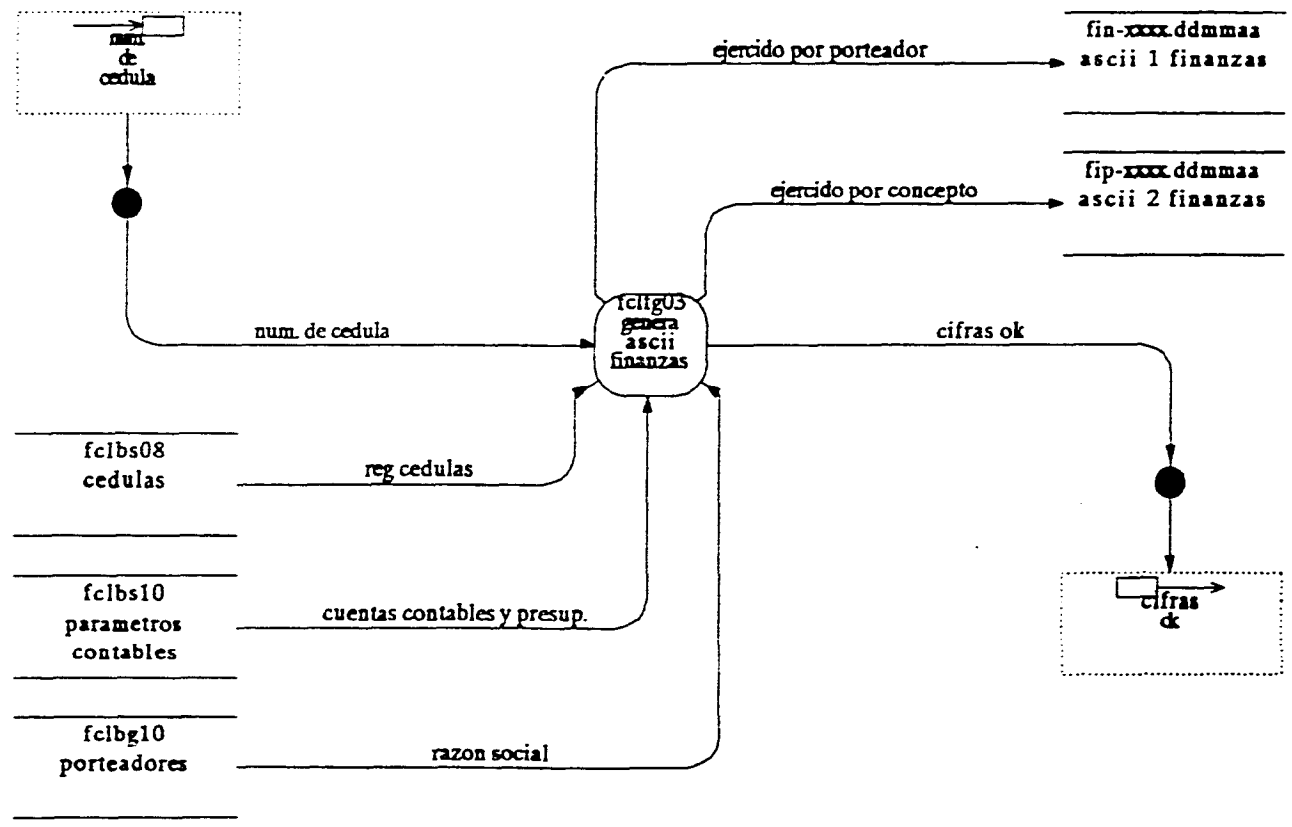


Emite Contra-Recibos

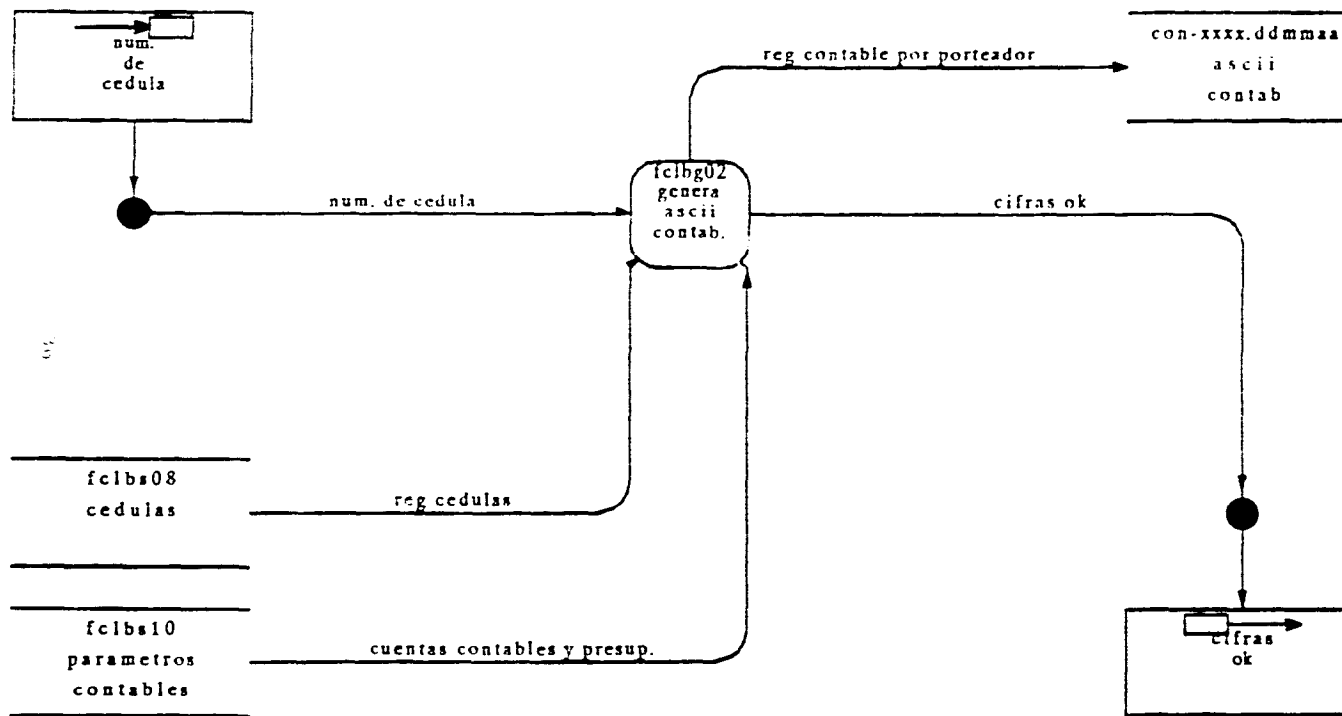


Genera Mecanizados

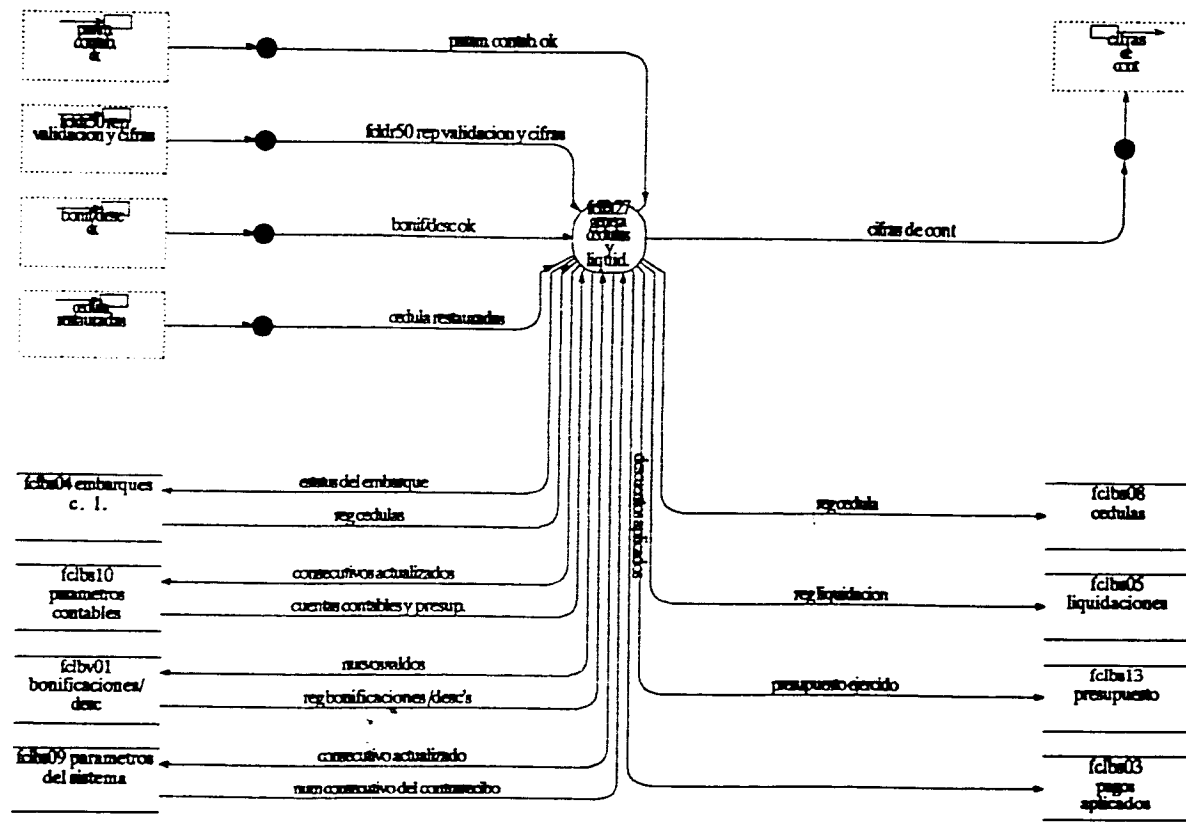
35



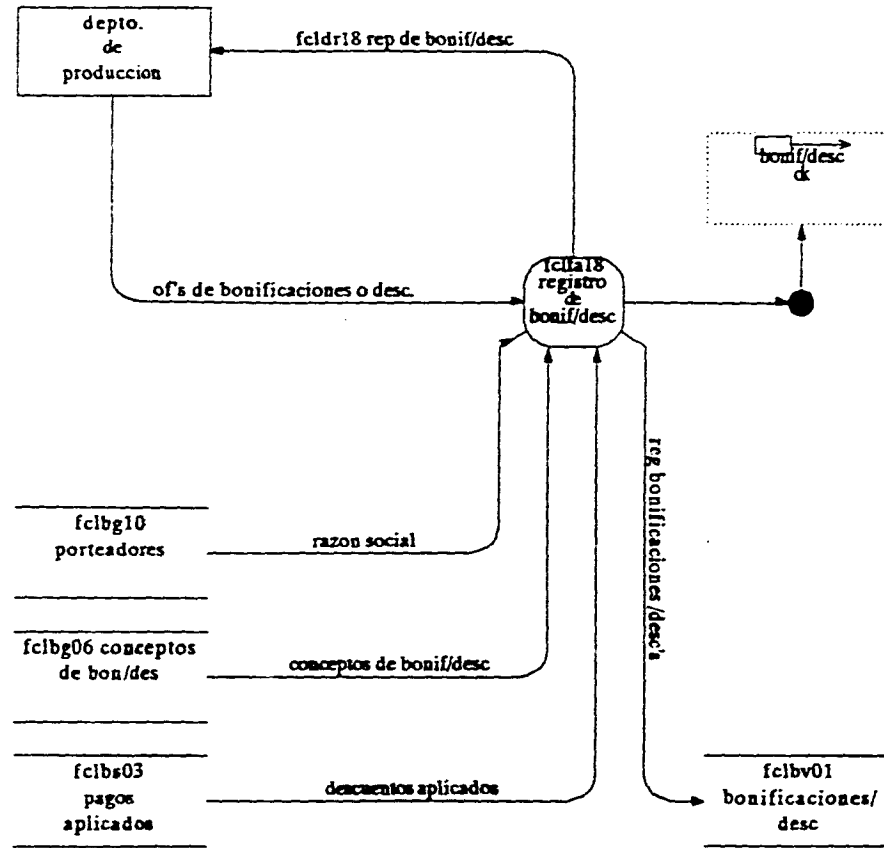
Genera Archivo Ascii para Finanzas



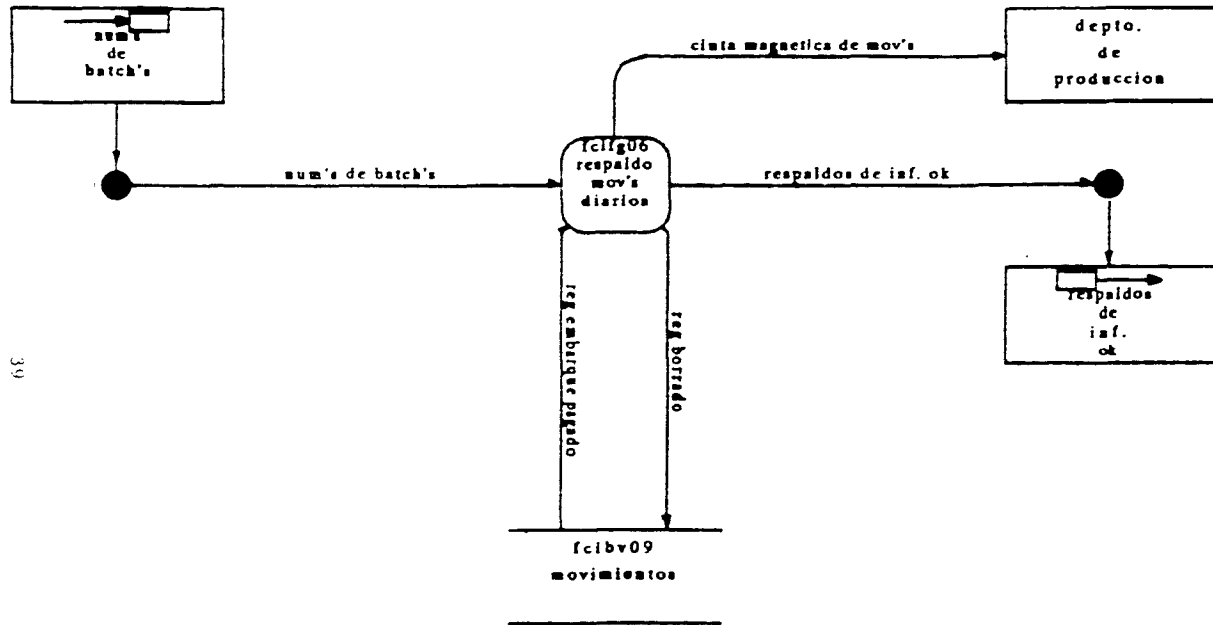
Genera Archivo Ascii para Contabilidad



Genera Cédulas y Liquidaciones

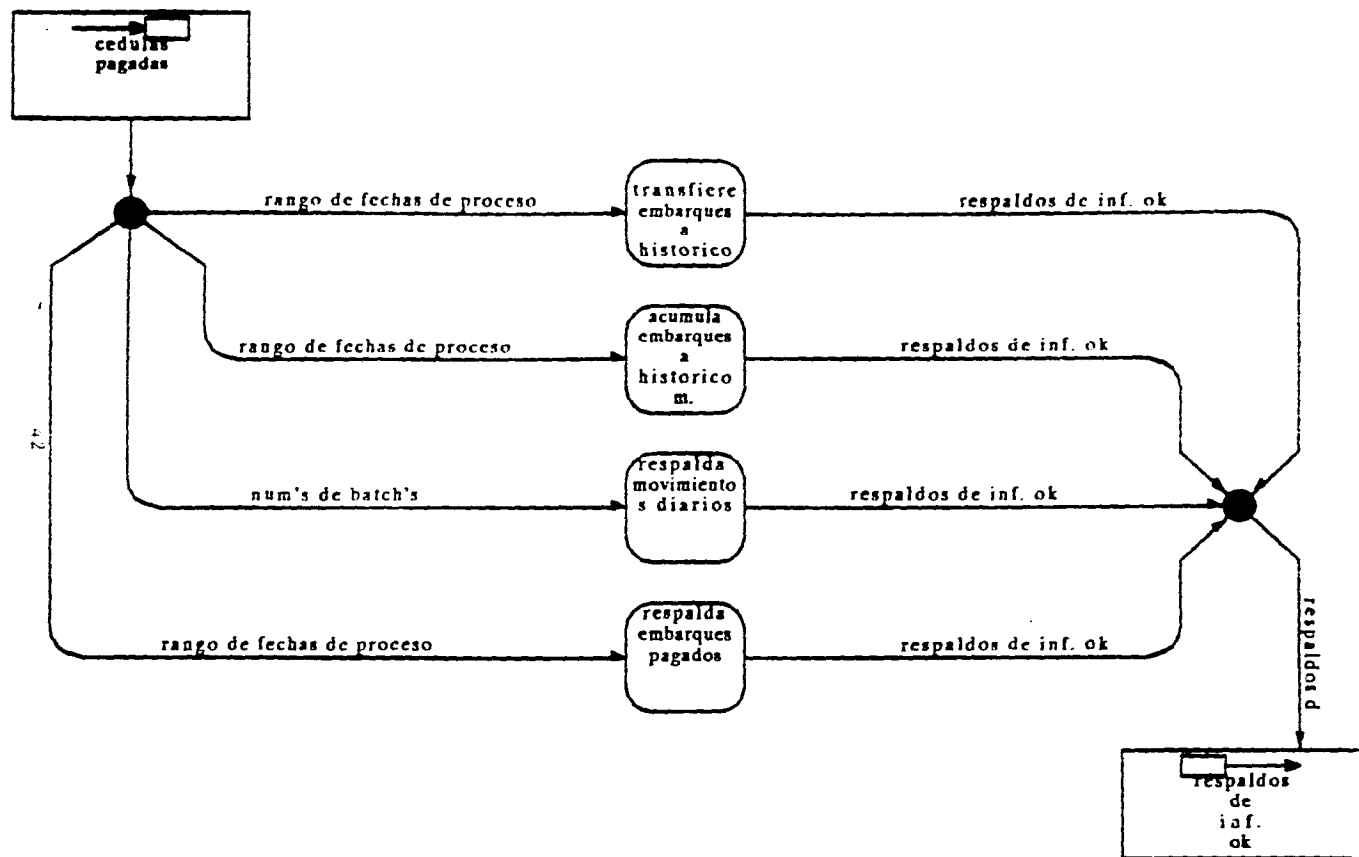


Registra Bonificaciones y Descuentos

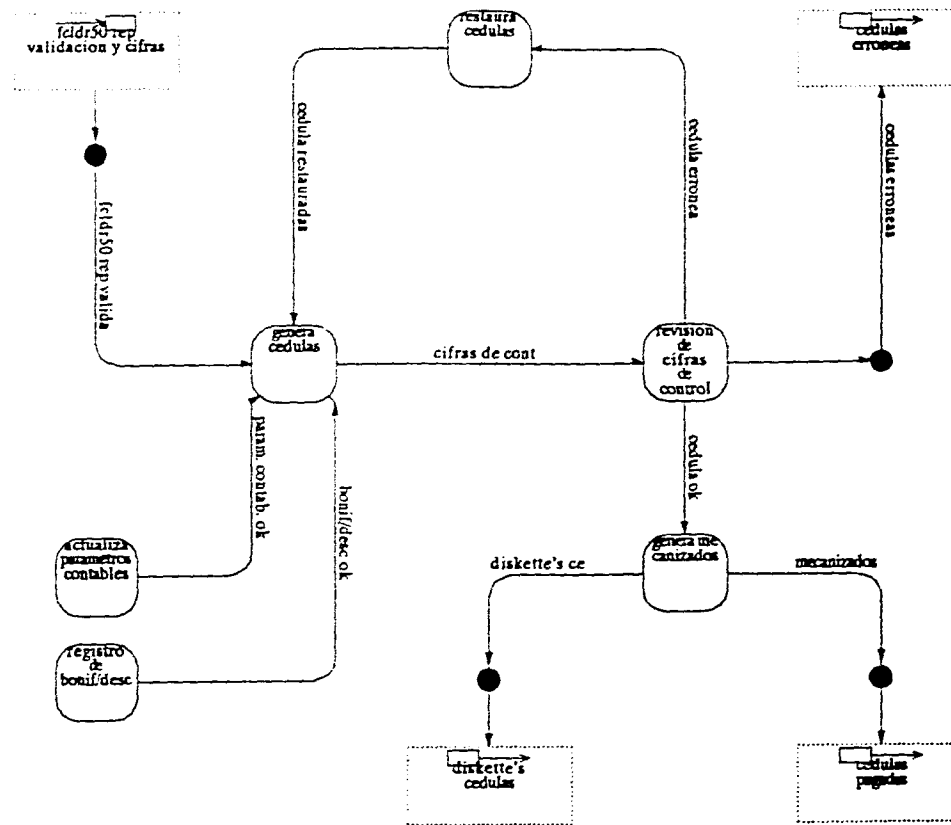


39

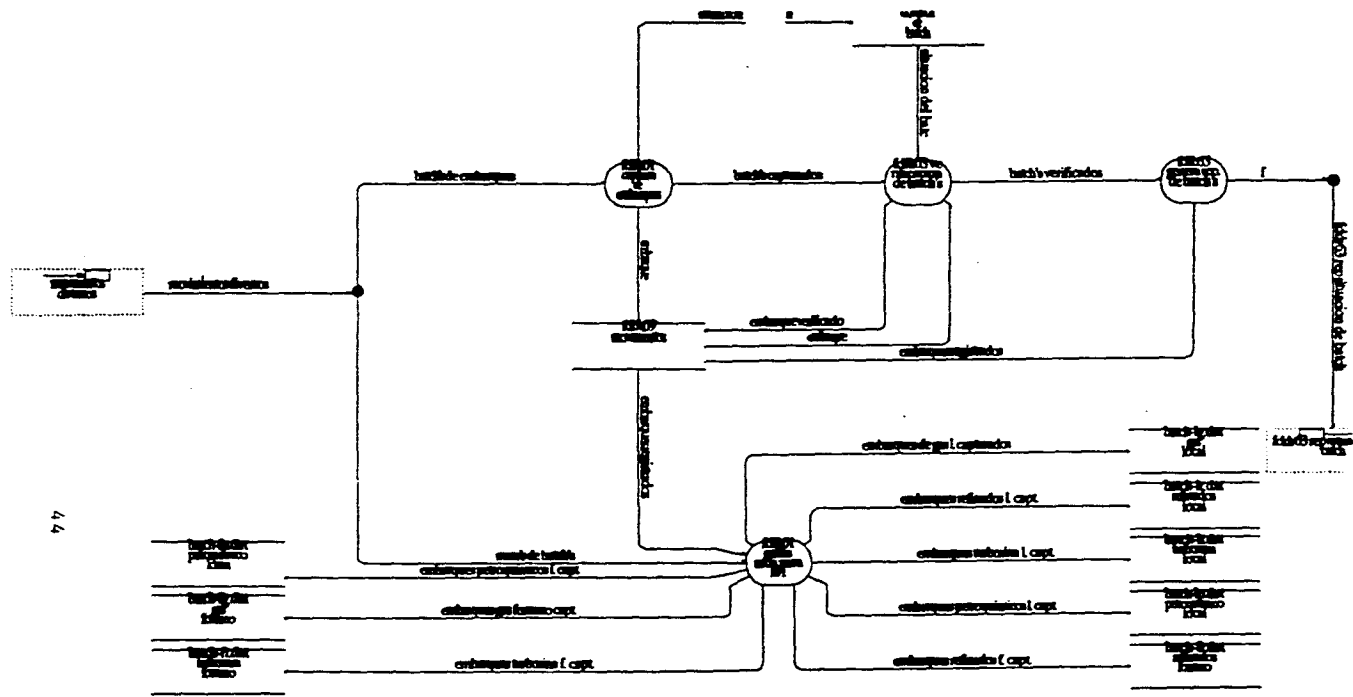
Respaldo de Movimientos Diarios



Generación de Acumulados

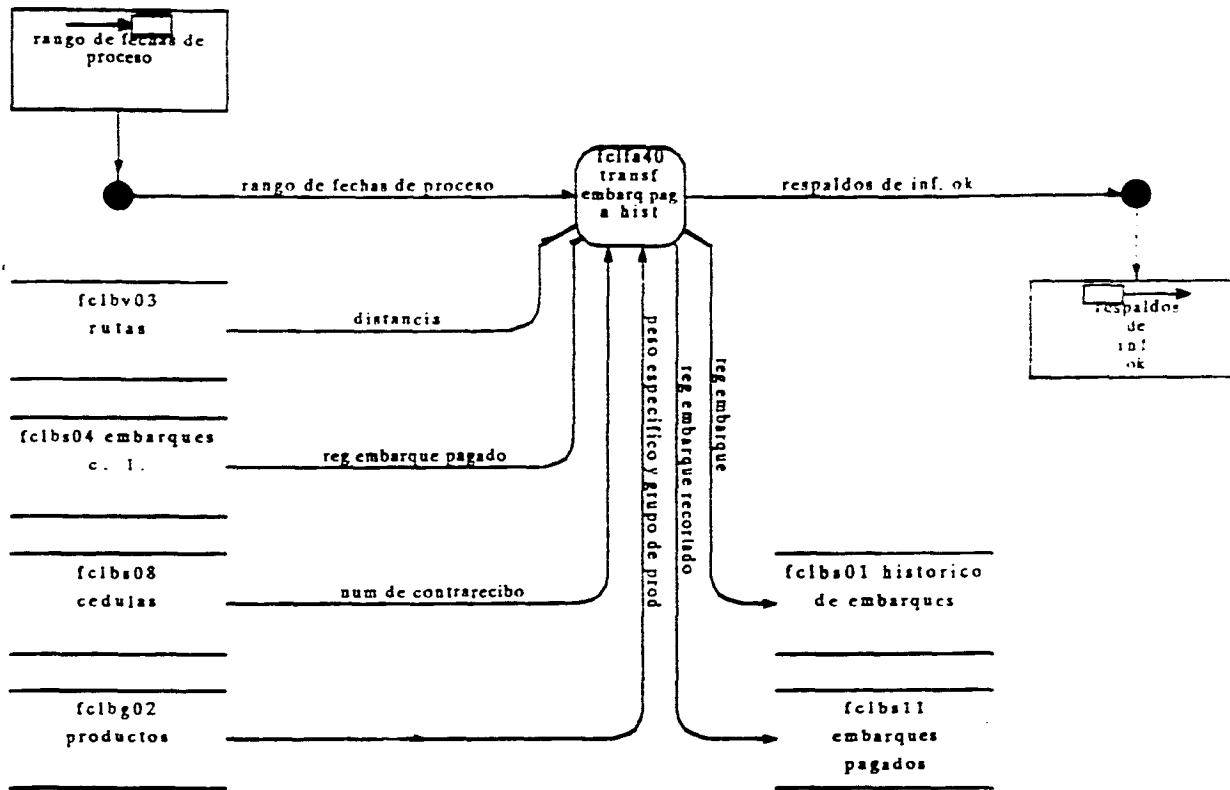


Proceso de Generación de Cédulas y Mecánizados



7/7

Genera Archivo Ascii para IBM



4.5

Generación de Histórico de Embarques Pagados

ETAPA 5

De acuerdo al bloque 7, 8 y 9 se realizó el diseño de los procedimientos y se implementó, aunque generando código en Informix-4gl un prototipo de lo que serían los módulos de procesos tipo transaccional es decir aquellos que tuvieran altas, modificaciones, bajas, generación de reportes y consultas dinámicas.

Para la realización del prototipo se probó y estableció, un estandar para el despliegue de pantallas (incluyendo mensajes de error, leyendas, fecha, tipo de movimiento, mensajes de ayuda etc.), se analizó la manera más óptima de captura, así como los atributos para una mejor visión y manipulación de datos.

Se estandarizó la implementación de consultas y reportes dinámicos, esto significa el poder realizar una consulta mediante cualquier criterio de selección, así mismo generar reportes bajo cualquier criterio.

ETAPA 6

Se realizó conforme al bloque 14 que se relaciona con la conversión de los modelos de datos y procedimientos dentro del diseño de la base de datos. Se implementó la conversión de nuestro "Modelo de datos" al diseño físico de la base de datos.

Esto nos permitió generar automáticamente el DDL para la implementación física de la Base de datos para el desarrollo; así como las estructuras de las tablas que incluyen los campos e índices y su documentación.

De esta etapa se obtuvo el diagrama del diseño físico de la Base de Datos (Ver fig.III.2.1.6), así como la estructura de cada una de las tablas que la integran y una relación de éstas.

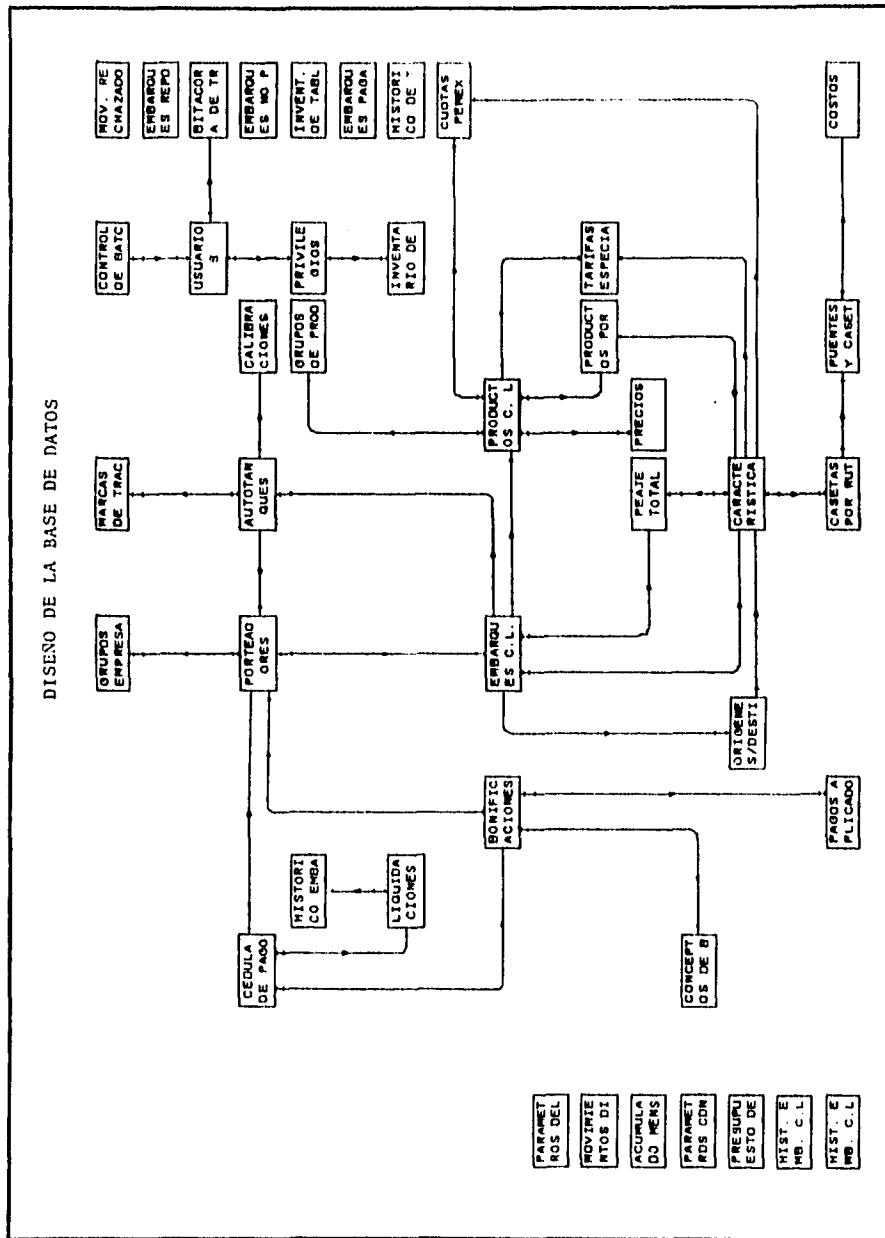


Fig. III.2.1.6 Diseño Físico de la Base de Datos del SPF_AT generado por IEW

Petroleos Mexicanos C.A.S.E

Object List

(Lista de Objetos Tipo Relacion Generada por IEW)

relation acumulado mensual emb.pag.
relation autorizaciones
relation autotanques
relation bitacora de transacciones
relation bonificaciones/descuentos
relation calibraciones situaciones
relation características rutas
relation casetas por ruta
relation cedula de pago
relation conceptos de autorizacion
relation conceptos de bon/desc
relation control de batch's
relation costos
relation cuentas por pagar
relation cuotas pemex
relation embarques c.l.
relation embarques no pagados
relation embarques pagados recortado
relation embarques reportados
relation grupos de productos
relation grupos empresariales
relation hist. emb. c.l. anterior
relation hist. emb. c.l. seleccionado
relation historico de transacciones
relation historico embarques pag. c.l.
relation invent. de tablas
relation inventario de prog's
relation liquidaciones
relation marcas de tractores y tanques
relation mov. rechazados por calculo
relation movimientos diarios
relation origenes/destinos
relation pagos aplicados
relation parametros contables
relation parametros del sistema
relation peaje total
relation porcentajes autorizados
relation porteadores
relation precios
relation presupuesto de fletes
relation privilegios

Subgerencia de Distribución

Petroleos Mexicanos C A.S.E

Object List

(Lista de Objetos Tipo Relacion Generada por IEW)

relation productos c. liq.
relation productos por ruta
relation puentes y casetas
relation tarifas especiales
relation usuarios

Subgerencia de Distribución

	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
ACUMULADO MENSUAL EMB. PAG.					
LLAVE ACUM. ESTADISTICOS:	✓	FCLIC40		FCLES12	
: ANIO				F-ANIO	SMALLINT
: MES				F-MES	SMALLINT
: ORIGEN:				C-ORIGEN	INTEGER
: DESTINO:				C-DESTINO	INTEGER
: CLAVE PRODUCTO:				C-PROD	SMALLINT
: GRUPO DE PRODUCTO:				C-GRUPO	CHAR(1)
: CLAVE FORTEADOR:				C-PORTEA	SMALLINT
: DATOS FECHA EMBARQUE					
: NUMERO EMBARQUE:				N-EMFTE	INTEGER
: CANTIDAD ENBARCADA				N-CANDETE	DECIMAL(15,0)
: IMPORTE FLETE				N-IMFLEFE	DECIMAL(15,0)
: IMPORTE DEDUCCION				N-IFALFE	DECIMAL(15,0)
: IMPORTE P. E. C. E.				N-IMECFE	DECIMAL(15,0)
: IMPORTE I. V. A.				N-IVAFE	DECIMAL(15,0)
: IMPORTE FEAJE				N-IFAJFE	DECIMAL(15,0)
: KM. RECORRIDOS				N-KMFE	INTEGER
: KG / KM				N-KGME	DECIMAL(15,0)
: DATOS FECHA PAGAR					
: NUMERO EMBARQUE:				N-EMFPP	INTEGER
: CANTIDAD ENBARCADA				N-CANDEFP	DECIMAL(15,0)
: IMPORTE FLETE				N-IMFLEFP	DECIMAL(15,0)
: IMPORTE DEDUCCIONES				N-IFALFP	DECIMAL(15,0)
: IMPORTE P. E. C. E.				N-IMECFPP	DECIMAL(15,0)
: IMPORTE IVA				N-IVAFPP	DECIMAL(15,0)
: IMPORTE FEAJE				N-IFAJEFP	DECIMAL(15,0)
: KM. RECORRIDOS				N-KMFP	INTEGER
: KG / KM				N-KGMP	DECIMAL(15,0)

Estructura Para la Tabla de Acumulado Mensual

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type	Data Type
:AUTORIZACIONES					FCRBS01		
:CLAVE PORTEADOR				✓	FCRICO1	C-PORTEA	SMALLINT
:NUMERD DE EMBARQUE				✓	FCRICO1	N-EMBARO	INTEGER
:FECHA EMBARQUE				✓	FCRICO1	F-EMBARO	DATE
:CLAVE AUTDRIZACION						C-AUTDRIZ	SMALLINT
:IMPORTE AUTORIZADO						N-IMPAUTD	DECIMAL(10.0)
							LOCAL

Estructura Para la Tabla de Autorizaciones

	SAC	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:AUTOTANQUES							FCLB305
Entity Type: VEHICULOS							
LLAVE TANQUES:					✓	FCLIC01	
CLAVE PORTEADOR:			POINTER				C-PORTEA SMALLINT
CLAVE AUTOTANQUE:			POINTER				C-AUTO CHAR(6)
SERVICIO							
Attribute Type: SERVICIO			TOKEN				C-SERV CHAR(1)
DATOS TRACTOR							
MARCA TRACTOR			POINTER				C-MARTRA SMALLINT
Attribute Type: MARCA TRAC							
MODELO TRACTOR			TOKEN				N-MODTRA SMALLINT
Attribute Type: MODELO TRA							
NUM. SERIE TRACTOR			TOKEN				N-SERIETRA CHAR(17)
DATOS TANQUE							
MARCA			POINTER				C-MARTAN SMALLINT
MODELO			TOKEN				N-MODTAN SMALLINT
EJES			TOKEN				N-EJES SMALLINT
CAPACIDAD			TOKEN				N-CAPAC INTEGER
NUMERO SERIE			TOKEN				N-SERIE CHAR(17)
CALIBRACION							
NUMERO FOLIO DE CALIBRACION			TOKEN				N-FOCAL CHAR(8)
FECHA FOLIO DE CALIBRACION			TOKEN				F-FOLIO DATE
SITUACION							
TIPO SITUACION			TOKEN				C-STATUS CHAR(1)
Attribute Type: TIPO SITUA							
FECHA SITUACION			TOKEN				F-STATUS DATE
Attribute Type: FECHA SITU							
NUMERO DE DOCUMENTO			TOKEN				N-DOCU M CHAR(8)
FECHA DE DOCUMENTO			TOKEN				F-DOCU M DATE
OBSERVACIONES			TOKEN				D-OBSER CHAR(20)
Attribute Type: OBSERVACIO							
FECHA INGRESO			TOKEN				F-INGRESO DATE
Attribute Type: FECHA INGRESO							
FECHA ULTIMO VIAJE			TOKEN				F-VIULT DATE
Attribute Type: FECHA ULTIMO							

Estructura Para la Tabla de Autotanques

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type	Data Type	
:BITACORA DE TRANSACCIONES							FCLBS06	
NUMERO SECUENCIAL DEL EVENTO:	POINTER					N-SECUEN	INTEGER	SECUENCIA
FECHA DEL EVENTO:	TOKEN	✓			FCLIC52	F-EVENTO	DATE	F-EVENT
HORA DEL EVENTO:	TOKEN	✓			FCLIC52	N-TIME	TIME	HORA
IDENTIFICADOR DE LA TABLA:	POINTER	✓			FCLIC52	C-TABLA	CHAR(7)	IDEN-TAB
INDICE PRIMARIO DE LA TABLA:	TOKEN					N-INDICE	CHAR(34)	LOCAL
TIPO DE EVENTO:	POINTER					C-EVENTO	CHAR(1)	TIPO-EVEN
CLAVE USUARIO:	POINTER					C-USER	CHAR(7)	LOCAL
IDENTIFICADOR DEL USUARIO:	POINTER					N-IDENT	SMALLINT	IDENT-USU
NUMERO DE TERMINAL:	TOKEN					N-TERMINAL	CHAR(8)	LOCAL
DESCRIPCION DEL EVENTO:	TOKEN					D-EVENTO	CHAR(25)	DES-EVEN

Estructura Para la Tabla de Bitácora de Transacciones

	Index	Physical Name	Data type	SQL Data type	Purpose
:BONIFICACIONES/DESCUENTOS					
Entity Type: BONIFICACIONES/DESC					
LLAVE BONIFICA:					
CLAVE PORTEADOR:	FOINTER	✓	FCLIC03	C-PORTEA	SMALLINT OVE PORT
:NUMERO DOCUMENTO	FOINTER			N-DOCPA	CHAR(17) LOCAL
Attribute Type: NUMERO DOC					
:FECHA DOCUMENTO	TOKEN			F-DOCPA	DATE LOCAL
Attribute Type: FECHA DOCUMEN					
:CLAVE CONCEPTO DOCUMENTO	FOINTER	✓	FCLIC01	C-CONCEP	SMALLINT OVE CONCEP
Attribute Type: CONCEPTO DOCU					
:IMPORTE DOCUMENTO	TOKEN			N-I-DOCPA	DECIMAL(10,0) LOCAL
Attribute Type: IMPORTE DOCUM					
NUMERO DE CEDULA:	FOINTER			N-CEDULA	INTEGER NUM CED
CLAVE TIPO DE CEDULA:	FOINTER			C-TIPOCEDU	CHAR(1) TIP-BATCH
:IMPORTE PORCENTAJE A APLICAR	TOKEN			N-I-APLIC	DECIMAL(12,2) LOCAL
Attribute Type: IMPORTE APLIC					
:FECHA APLICACION	TOKEN			F-APLIC	DATE LOCAL
Attribute Type: FECHA APLICAC					
:SALDO	TOKEN	✓	FCLIC04 DESC	N-I-SALDO	DECIMAL(10,0) LOCAL
:ACTIVADO DESACTIVADO	TOKEN			C-ACT-DESA	SMALLINT LOCAL
Attribute Type: ACTIVADO DESA					

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type	Data Type
CALIBRACIONES SITUACIONES					FCLBVO2		
LLAVE CALIBRACION		✓			FCLICOS		
CLAVE PORTEADOR	POINTER				C-PORTEA	SMALLINT	CVE PORT
CLAVE AUTOTANQUE	POINTER				C-AUTO	CHAR(5)	CV-AUTO
FECHA INICIAL DE CALIBRACION	POINTER		OESC		F-INICALB	DATE	LOCAL
FECHA FINAL DE CALIBRACION	POINTER				F-FINALB	DATE	LOCAL
TIPO DE MOVIMIENTO	TOKEN				N-TIPOMOV	CHAR(1)	LOCAL
CAPACIDAD	TOKEN				N-CAPAC	INTEGER	LOCAL
FECHA DE DOCUMENTO	TOKEN				F-DOCUM	DATE	LOCAL
NUMERO DE DOCUMENTO	TOKEN				N-DOCUM	CHAR(8)	LOCAL

Estructura Para la Tabla de Calibraciones y Situaciones

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SOL Data Type	Data Type
:CARACTERISTICAS RUTAS							FCLBVO3
LLAVE RUTA:		✓			FCLIC06		
ORIGEN:	POINTER					C-ORIGEN	INTEGER C-ORDES
DESTINO:	POINTER					C-DESTINO	INTEGER C-ORDES
RUTA:	POINTER					C-RUTA	SMALLINT C-RUTA
KILOMETROS:	TOKEN					N-KMS	INTEGER LOCAL
:IMPORTE BONIFICACION	TOKEN					N-BONIF	DECIMAL(8,0) LOCAL
:PORCENTAJE BONIFICACION	TOKEN					N-PORSON	DECIMAL(8,5) LOCAL
:FECHA INICIO	TOKEN	✓			FCLIC06 DESC	F-INIRUTA	DATE LOCAL
:FECHA FINAL	TOKEN	✓			FCLIC06	F-FINRUTA	DATE LOCAL
:KMS-TERRACERIA	TOKEN					N-K-TERRA	SMALLINT LOCAL
:PORCENTAJE ADICIONAL DE GAS	TOKEN					N-PORGAS	DECIMAL(7,3) LOCAL
:IMPORTE FANCORDADOR	TOKEN					N-IMPTRANS	DECIMAL(8,0) LOCAL
:CLAVE AUTORIZACION 9 EJES	TOKEN					C-AUTSEJE	CHAR(1) LOCAL

Estructura Para la Tabla de Características de Rutas

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:CASETAS POR RUTA					FCLBN01	
Entity Type:PEAJE POR RUTA						
LLAVE RUTA:		✓			FCLIC07.FCLIC08	
ORIGEN:	POINTER				C-ORIGEN	INTEGER
DESTINO:	POINTER				C-DESTINO	INTEGER
RUTA:	POINTER				C-RUTA	SMALLINT
:CLAVE CASETA/PUENTE	POINTER	✓	✓		FCLIC07.FCLIU02	C-CA-PUE
						CHAR(3)

Estructura Para la Tabla de Casetas por Ruta

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type	
CEDULA DE PAGO:							
LLAVE CEDULA:		✓			FCLIC09		
NUMERO DE CEDULA:	POINTER				N-CEDULA	INTEGER	
CLAVE TIPO DE CEDULA:	POINTER				C-TIPOCEDU	CHAR(1)	
FECHA PROCESO:	POINTER				F-PROCES	DATE	
FECHA PROGRAMADO PAGO:	TOKEN				F-PROGRA	DATE	
AGENCIA PAGADORA:	POINTER				C-ORIDES	INTEGER	
CLAVE PORTEADOR:	POINTER	✓			FCLIC09	C-FORTEA	SMALLINT
NUMERO CONTRARRECIBO:	TOKEN				N-CONTRA	INTEGER	
IMPORTES:							
FLETE TOTAL:	TOKEN				N-IMPFFLETE	DECIMAL(10,0)	
DEDUCCION POR FALTANTE:	TOKEN				N-IMPXFALT	DECIMAL(10,0)	
I.V.A.:	TOKEN				N-IMPIVA	DECIMAL(10,0)	
P.E.C.E.:	TOKEN				N-IMPECE	DECIMAL(10,0)	
REAJE TOTAL:	TOKEN				N-IMPEAJE	DECIMAL(10,0)	
IMPORTE TOT. POR BONIFICACIONES	TOKEN				N-IMPBOHIF	DECIMAL(10,0)	
IMPORTE TOT. POR DESUENTOS:	TOKEN				N-IMPDESC	DECIMAL(10,0)	
DATOS CONTABLES:							
FECHA DE ELABORACION:	TOKEN				F-ELABORA	DATE	
NUMERO DE DOCUMENTO:	TOKEN				N-DOCSIG	SMALLINT	
HOJA DE DOCUMENTO:	TOKEN				N-HOJSIDOC	SMALLINT	
NUM. DE ADEFAS:	TOKEN				N-SIGADF	INTEGER	
CLAVE DE AUTORIZACION:	TOKEN				C-AUTOFINA	CHAR(11)	
DATOS FINANCIEROS:							
FOLIO FINANZAS:	TOKEN				N-FOLFINA	CHAR(7)	
STATUS FINANZAS:	TOKEN				C-STAFINA	CHAR(1)	
DATOS FLUJO DE EFECTIVO:							
FECHA DE RECEPCION:	TOKEN				F-RECEP	DATE	
FECHA DE DEPOSITO BANCARIO:	TOKEN				F-DEPOST	DATE	

Estructura Para la Tabla de Cedula de Pago

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:CONCEPTOS DE AUTORIZACION					FCRB001	
:CLAVE DE AUTORIZACION	POINTER	✓	FCRIU03		C-AUTORIZ	SMALLINT
:DESCRIPCION DE AUTORIZACION	TOKEN				D-AUTORIZ	CHAR(25)

Estructura Para la Tabla de Conceptos de Autorización

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
CONCEPTOS DE BON/DESC						FCLB606
:CLAVE CONCEPTO	POINTER	✓			FOLIUS	C-CONCEP SMALLINT
:TIPO BONIFICACION/DESCUENTO	TOKEN					N-TIPBON CHAR(1)
:DESCRIPCION BONIF/DESCUENTO	TOKEN					D-CONCEP CHAR(25)

Estructura Para la Tabla de Conceptos de Bonificaciones/Descuento

SQL Data Type				
Physical Name				
Index				
Foreign Key				
Primary Key				
Purpose				
CONTROL DE BATCH'S				FCLBV12
LLAVE USUARIO:		✓	FCLIC32	
CLAVE USUARIO:	PDINTER		C-USER	CHAR(7)
IDENTIFICADOR DEL USUARIO:	POINTER		N-IDENT	SMALLINT
LLAVE CONTROL BATCH:		✓		
NUMERO DE BATCH:	POINTER	✓	FCLIC32, FCLIU18	N-BATCH
FECHA DE CAPTURA:	POINTER		F-BATCH	DATE
TIEMPO DE CAPTURA:	TOKEN		N-TIEMPO	TIME
CLAVE USUARIO VERIFICACION:	TOKEN		C-USERV	CHAR(7)
IDENTIF. USUARIO VERIFICADOR:	TOKEN		N-IDENTV	SMALLINT
CLAVE STATUS DEL BATCH:	TOKEN		C-STATUS	CHAR(1)
NUMERO DE LIQUIDACIONES:	TOKEN		N-NUMLIQUI	SMALLINT
NUMERO DE EMBARQUES:	TOKEN		N-NUMEMBARD	SMALLINT

Estructura Para la Tabla de Control de Batch

			SQL Data Type	Physical Name	
: COSTOS				FCLBV04	
LLAVE COSTOS:		✓	FCLIC10		
CLAVE CASETA/PUENTE:	POINTER	✓	FCLIU04	C-CA-PUE	CHAR(3)
: FECHA INICIAL	POINTER		DESC	F-INICOST	DATE
: FECHA FINAL	POINTER			F-FINICOST	DATE
: COSTO 1 EJES	TOKEN			N-CX2EJE	DECIMAL(6.0)
: COSTO 2 EJES	TOKEN			N-CX2EJE	DECIMAL(6.0)
: COSTO 5 EJES	TOKEN			N-CX5EJE	DECIMAL(6.0)
: COSTO 6 EJES	TOKEN			N-CX6EJE	DECIMAL(6.0)
: COSTO 9 EJES	TOKEN			N-CX9EJE	DECIMAL(6.0)
: COSTO ADICIONAL POR EJE	TOKEN			N-CXEJE	DECIMAL(6.0)

Estructura para la Tabla de Costos por Casetas

Estructura Para la Tabla de Cuentas por pagar.

SQL Data Type	Physical Name	Dotbase Field	Index	Foreign Key	Primary Key	Purpose
						Cuentas por pagar
SMALLINT	CLAVE PORTADOR	C-PORTA	✓			POINTER
SMALLINT	NUMERO DE DOCUMENTO	N-DOCUMENTO	✓			VALUE
SMALLINT	CLAVE CONCEPTO DOCUMENTO	C-CONCEPTO	✓			TOKEN
DECIMAL(10,0)	IMPORTE DOCUMENTO	N-IMPORTE	✓			VALUE
INTEGER	NUMERO DE CEDULA	N-CEDULA	✓			TOKEN
CHAR(1)	CLAVE TIPO DE CEDULA	C-TIPOCED	✓			POINTER
DECIMAL(10,0)	IMPORTE PORCENTAJE A APLICAR	N-IMPAPLICA	✓			VALUE
DATE	FECHA APLICACION	F-APLICA	✓			TOKEN
DECIMAL(10,0)	SALDO	N-SALDO	✓			TOKEN

	Purpose	Primary Key	Foreign Key	Index	Database Field	Physical Name	SQL Data Type
CUOTAS PEMEX						FCLB601	
Entity Type TARIFAS							
LLAVE CUOTAS					FCLIC11		
KILOMETRAJE	POINTER					N-KMS	SMALLINT
FECHA INICIO CUOTA	POINTER		DESC			F-TCUOTA	DATE
FECHA FINAL CUOTA	POINTER					F-FCUOTA	DATE
CUOTA GRUPO 1	TOKEN					N-CUOC1	DECIMAL(8,2)
CUOTA GRUPO 2	TOKEN					N-CUOC2	DECIMAL(8,2)
CUOTA GRUPO 3	TOKEN					N-CUOC3	DECIMAL(8,2)
CUOTA GRUPO 4	TOKEN					N-CUOC4	DECIMAL(8,2)
CUOTA GRUPO 5	TOKEN					N-CUOC5	DECIMAL(8,2)
CUOTA GRUPO 6	TOKEN					N-CUOC6	DECIMAL(8,2)
CUOTA GRUPO 7	TOKEN					N-CUOC7	DECIMAL(8,2)
CUOTA GRUPO 8	TOKEN					N-CUOC8	DECIMAL(8,2)

Estructura para la Tabla de Cuotas Pemex

Estructura Para la Tabla de Embarques C.L.

Entity Type: EMBAQUE	Field	Type	Key	Field	Type
EMBAQUE C.L.				FECHORD	
Entity Type: EMBAQUE					
NUMERO EMBAQUE:	NUMBER	PK	PK	N-EMBAQ	INTEGER
FECHA EMBAQUE:	DATE			F-EMBAQ	DATE
DATOS GENERALES DE EMBAQUE:					
CLAVE EMBAQUE:	NUMBER	PK	PK	C-EMBAQ	SMALLINT
NUMERO EMBAQUE:	NUMBER			N-EMBAQ	INTEGER
FECHA EMBAQUE:	DATE			F-EMBAQ	DATE
ORIGEN:	NUMBER			C-ORIGEN	INTEGER
DESTINO:	NUMBER			C-DESTINO	INTEGER
RUJA:	NUMBER			C-RUJA	SMALLINT
DESTINO REEMISION:	NUMBER			C-DESTRE	INTEGER
RUJA DE REEMISION:	NUMBER			C-RUJARE	SMALLINT
CLAVE AUTOMOVE:	NUMBER			C-AUTO	CHAR(6)
CLAVE PROYECTO:	NUMBER			C-PROY	SMALLINT
UNIDAD DEL PROYECTO:	NUMBER			N-UNID	CHAR(1)
CANTIDAD CARGA:	NUMBER			N-CARGA	INTEGER
TEMPERATURA DE CARGA:	NUMBER			N-TEMPAR	SMALLINT
IMPORTE UNDA DE EMBAQUE:	NUMBER			N-IMPUR	INTEGER
CANTIDAD LESADAMA:	NUMBER			N-LESMA	INTEGER
TEMPERATURA DE LESADAMA:	NUMBER			N-TEMPLES	SMALLINT
TARIFA:	NUMBER			N-TARIFA	DECIMAL(8,4)
TARIFA REEMISION:	NUMBER			N-TARRE	DECIMAL(8,4)
IMPORTE:					
FLETE TOTAL:	NUMBER			N-IMPLETE	DECIMAL(10,0)
REDUCCION POR FALTA DE:	NUMBER			N-IMPFA	DECIMAL(10,0)
I.V.A.:	NUMBER			N-IMPIVA	DECIMAL(10,0)
P.E.C.E.:	NUMBER			N-IMPICE	DECIMAL(10,0)
IMPORTE TOTAL:	NUMBER			N-IMPTE	DECIMAL(10,0)
CLAVE ATRESO:	NUMBER	PK	PK		
FECHA ATRESO:	DATE			F-ATRES	DATE
NUMERO DE CELLA:	NUMBER			N-CELLA	INTEGER
CLAVE TIPO DE CELLA:	NUMBER			C-TIPOCELL	CHAR(1)
NUMERO DE BACH:	NUMBER			N-BACH	INTEGER
FECHA PROGRAMADO PAO:	DATE			F-PAO	DATE
STATUS:	NUMBER			C-STATUS	CHAR(1)

Physical Name	SI Data type	Index	Foreign Key	Primary Key	Purpose
:EMBARQUES NO PAGADOS					FCLEB02
DATOS GENERALES DE EMBARQUES:					
LLAVE EMBARQUES:		✓	FCLIC15		
CLAVE PORTEADOR:	FOINTER			C-PORTEA	SMALLINT
NUMERO EMBARQUE:	FOINTER			N-EMBARO	INTEGER
FECHA EMBARQUE:	FOINTER		DESC	F-EMBARO	DATE
ORIGEN:	FOINTER			C-ORIGEN	INTEGER
DESTINO:	FOINTER			C-DESTINO	INTEGER
RUTA:	FOINTER			C-RUTA	SMALLINT
DESTINO REEXPEDICION:	FOINTER			C-DESTRE	INTEGER
RUTA DE REEXPEDICION:	FOINTER			C-RUTARE	SMALLINT
CLAVE AUTOTANQUE:	FOINTER			C-AUTO	CHAR(5)
CLAVE PRODUCTO:	FOINTER			C-PROD	SMALLINT
UNIDAD DEL PRODUCTO:	TOKEN			N-UNID	CHAR(1)
CANTIDAD CARGADA:	TOKEN			N-CARGA	INTEGER
TEMPERATURA DE CARGA:	TOKEN			N-TEMPAR	SMALLINT
IMPORTE NOTA DE EMBARQUE:	TOKEN			N-IMPOR	INTEGER
CANTIDAD DESCARGADA:	TOKEN			N-DESCA	INTEGER
TEMPERATURA DE DESCARGA:	TOKEN			N-TEMDES	SMALLINT

Estructura Para la Tabla de Embarques no Pagados

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
EMBARQUES PAGADOS RECORTADO						FCLBS11
DATOS LIQUIDACION:						
NUMERO LIQUIDACION:		POINTER				N-LIOUI
FECHA LIQUIDACION:		TOKEN				F-LIOUI
CLAVE PORTEADOR:		POINTER				C-PORTEA
LLAVE EMBPAG				✓	FCLIC39	
NUMERO EMBARQUE:		POINTER		✓	FCLIU17	N-EMBARQ
FECHA EMBARQUE:		POINTER				F-EMBARQ
DATOS PROCESO:						
FECHA PROCESO:		POINTER				F-PROCES
NUMERO DE CEDULA:		POINTER				N-CEDULA
CLAVE TIPO DE CEDULA:		POINTER				C-TIPOCEDU
FECHA PROGRAMADO PAGO:		TOKEN				F-PROGRA

07

Estructura Para la Tabla de Embarques Pag. Recortado

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:EMBARQUES REPORTADOS						FCLBV05
LLAVE REPORTADO:			✓		FCLIC18	
:EMBARQUE INICIAL	POINTER					N-EMBINI INTEGER
:EMBARQUE FINAL	POINTER					N-EMFIN INTEGER
:NUMERO DOCUMENTO	TOKEN					N-DOCTO CHAR(17)
:FECHA DOCUMENTO	TOKEN					F-DOCTO DATE

Estructura Para la Tabla de Embarques Reportados

69

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:GRUPOS DE PRODUCTOS					FCLB604	
:GRUPO DE PRODUCTO	POINTER	✓			FCLIU05	C-GRUPO CHAR(1)
:DESCRIPCION GRUPO PRODUCTO	TOKEN					D-GRUPO CHAR(30)

Estructura Para la Tabla de Grupos de Productos

70

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:GRUPOS EMPRESARIALES						FCLBG07
:CLAVE GRUPO EMPRESARIAL	POINTER	✓			FCLIU06	C-EMPRE SMALLINT
:NOMBRE DEL GRUPO EMPRESARIAL	TOKEN					D-EMPRE CHAR(30)

Estructura Para la Tabla de Grupos Empresariales

Historico de C.L. SELECCIONADO				
NUMERO LIQUIDACION	RUNER	J	FOLIO 47	NOBIA
FECHA LIQUIDACION	YDEN			FOLIO
DATOS HISTORICO DE PRO:				
LINEA DE PRO:		J	FOLIO 46	
CAE RUTADOR:	RUNER	J	FOLIO 47	CORREA
NUM. ENFER. HISTORICO		J	FOLIO 48	
LETRAS NUM. ENFERIALE	RUNER			CULEB
LETRAS NUM. ENFERIALE	RUNER			NOBIA
FECHA ENFERIALE	RUNER	J	FOLIO 48 DES	FERRAZ
ORDEN	RUNER			GORDEN
DESTINO	RUNER			CRISTINO
RUA	RUNER			CRIFA
LISTRO RESECCION	RUNER			CRISTE
CAE AUTOMATE	RUNER			CAJO
CAE PRODUCTO	RUNER			CRID
NUMERO DEL PRODUCTO	YDEN			NUMO
GRUPO DEL PRODUCTO	RUNER			GRUPO
RESECCION	YDEN			NRES
KILOMETROS	YDEN			NRES
CANTIDAD CARGA	YDEN			NORCA
TEMPERATURA DE CARGA	YDEN			NENCAR
CANTIDAD DESARROLA	YDEN			NRECA
TEMPERATURA DE DESTACA	YDEN			NRELES
UNIFA	YDEN			NREFA
UNIFA RESECCION	YDEN			NREXP
IMPRES				
PLA. TOTAL	YDEN			NRETE
UNIDAD POR FALTANTE	YDEN			NREALT
I VA	YDEN			NREVA
PECE	YDEN			NRETE
RECE TOTAL	YDEN			NREAE
LINEA PROESO				
FECHA PROESO	RUNER			FPROES
NUMERO DE CILLA	RUNER			NOCILLA
CAE TIPO DE CILLA	RUNER			C-TIPOC
DATOS DE PROSO				
FECHA PROSO PROSO	YDEN			FPROPA
NUMERO CONTINENTE	YDEN			NCONTPA
ABRISA PROSO	RUNER			GORDES
SINUS	YDEN			COSINUS

Estructura Para la Tabla de Liquidaciones.

73

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:HISTORICO DE TRANSACCIONES					FCLS07	
NUMERO SECUENCIAL DEL EVENTO:	POINTER	✓			FCLIC45	N-SECUEN INTEGER
CLAVE DEL REGISTRO:	POINTER	✓			FCLIC45	C-REGISTRO CHAR(1)
DESCRIPCION DEL REGISTRO:	TOKEN					D-REGISTRO CHAR(300)

Estructura Para la Tabla del Histórico Transaccion

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:INVENTARIO DE TABLAS					FCLB015	
:CLAVE DE LA TABLA	TOKEN	✓			FCLIU08	C-TABLA CHAR(7)
:DESCRIPCION DE LA TABLA	TOKEN				D-NOMBRE	CHAR(40)

Estructura Para la Tabla del Inventario de Tablas

Estructura Para la Tabla de Liquidaciones.

INVENTARIO DE PROG.S	FCLB01Z					
MODULO	C-MODULO	✓	FCL014	POINTER	✓	
MODULO FISICO	C-MODFIS	✓		TOKEN		
TITULO DEL MODULO	D-MODTIT	✓		TOKEN		
TIPO DEL MODULO	C-TIPO	✓		TOKEN		
FECHA DE ALTA DEL MODULO	F-ALTMOD	✓		TOKEN		DATE

SQL Data Type	Physical Name	Dotbase Field	Index	Foreign Key	Primary Key	Purpose
---------------	---------------	---------------	-------	-------------	-------------	---------

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
LIQUIDACIONES					FCLBS05	
LLAVE LIQUIDACION:		✓			FCLIC19	
NUMERO DE CEDULA:	POINTER		✓		FCLIC35	N-CEDULA INTEGER
CLAVE TIPO DE CEDULA	POINTER		✓		FCLIC35	C-TIPOCEDU CHAR(1)
CLAVE PORTEADOR:	POINTER					C-PORTEA SMALLINT
NUMERO LIQUIDACION:	POINTER					N-LIQUI INTEGER
FECHA LIQUIDACION	TOKEN					F-LIQUI DATE
IMPORTES:						
FLETE TOTAL:	TOKEN					N-IMPFFLETE DECIMAL(10.0)
DEDUCCION POR FALTANTE:	TOKEN					N-IMPXFALT DECIMAL(10.0)
I.V.A.:	TOKEN					N-IMPIVA DECIMAL(10.0)
P.E.C.E.:	TOKEN					N-IMPECE DECIMAL(10.0)
PEAJE TOTAL:	TOKEN					N-IMPEAJE DECIMAL(10.0)
AGENCIA PAGADORA	POINTER					C-ORIDES INTEGER

Estructura Para la Tabla de Liquidaciones

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:MARCAS DE TRACTORES Y TANQUES					FCLB008	
:LLAVE MARCA		✓			FCLIC24	
:MARCA TRACTOR	POINTER				C-MARTRA	SMALLINT
:TIPO TANQUE/TRACTOR	POINTER				C-TANTRA	CHAR(1)
:NOMBRE MARCA	TOKEN				D-MARTRA	CHAR(20)

Estructura Para la Tabla de Marcas de Tractores/Tanques

PARTE I MOV. REC. X CAL						
NUM C:						
NUMERO LIQUIDACION:	POINTER	/		INTEGER		LOCAL
FECHA LIQUIDACION:	INDEX	/		DATE		LOCAL
DATOS GENERALES DE EMBARQUES						
LLAVE EMBARQUE:		/				
CLAVE PORTADOR:	POINTER	/		SMALLINT		CV-PORT
NUMERO EMBARQUE:	POINTER	/		INTEGER		NUM-EMBAR
FECHA EMBARQUE:	POINTER	/		DATE		LOCAL
ORIGEN:	POINTER	/		INTEGER		C-ORIGEN
DESTINO:	POINTER	/		INTEGER		C-ORIGEN
RUJA:	POINTER	/		SMALLINT		C-RUJA
DESTINO REEMISION:	POINTER	/		INTEGER		C-ORIGEN
RUJA DE REEMISION:	POINTER	/		SMALLINT		C-RUJA
CLAVE AUTOMANTE:	POINTER	/		CHAR(6)		CV-AUTO
CLAVE PRODUCTO:	POINTER	/		SMALLINT		CV-PROD
UNIDAD DEL PRODUCTO:	INDEX	/		CHAR(1)		CV-UNID
CANTIDAD CARGADA:	INDEX	/		INTEGER		CVI-CAR
TEMPERATURA DE CARGA:	INDEX	/		SMALLINT		TEM-CAR
IMPORTE NOTA DE EMBARQUE:	INDEX	/		INTEGER		IMP-EMBA
CANTIDAD DE CARGADA:	INDEX	/		INTEGER		LOCAL
TEMPERATURA DE DESCARGA:	INDEX	/		SMALLINT		LOCAL

ESTRUCTURA PARA LA TABLA DE MOVIMIENTOS POR CALCULO PARTE I

08

	Purpose	Primary Key	Foreign Key	Index	Database Field	Physical Name	SQL Data Type
PARTE 2 MOVS REC X CAL							
MVFC2:							
TARIFA:	TOKEN				✓		DECIMAL(8,4)
TARIFA REEXPEDICION:	TOKEN				✓		DECIMAL(8,4)
IMPORTES:							
FLETE TOTAL	TOKEN				✓		DECIMAL(10,0)
DEDUCCION POR FALTANTE	TOKEN				✓		DECIMAL(10,0)
I V A	TOKEN				✓		DECIMAL(10,0)
P E C E	TOKEN				✓		DECIMAL(10,0)
PEAJE TOTAL	TOKEN				✓		DECIMAL(10,0)
LLAVE XPROCESO							
FECHA PROCESO	POINTER				✓		DATE
NUMERO DE CEDULA	POINTER				✓		INTEGER
CLAVE TIPO DE CEDULA	POINTER				✓		CHAR(1)
NUMERO DE BATCH	POINTER				✓		INTEGER
:CODIGO DE ERROR	TOKEN				✓		DECIMAL(15,0)

ESTRUCTURA PARA LA TABLA DE MOVIMIENTOS RECHAZADOS POR CALCULO PARTE II

MOVIMIENTOS DIARIOS				FCLBVS	
NUMERO DE BATCH:	POINTER	✓	FCLIC44	N-BATCH	INTEGER
NUMERO DE REGISTRO:	POINTER	✓	FCLIC44	N-NUMREG	SMALLINT
NUMERO DE CEDULA:	POINTER	✓	FCLIC20	N-CEDULA	INTEGER
TIPO DE BATCH:	TOKEN	✓	FCLIC20	C-TIPOBATCH	CHAR(1)
NUMERO LIQUIDACION:	POINTER			N-LIQUI	INTEGER
FECHA LIQUIDACION:	TOKEN			F-LIQUI	DATE
DATOS GENERALES DE EMBARQUES:					
LLAVE EMBARQUES:					
CLAVE PORTEADOR:	POINTER			C-PORTEA	SMALLINT
NUMERO EMBARQUE:	POINTER			N-EMBARO	INTEGER
FECHA EMBARQUE:	POINTER			F-EMBARO	DATE
ORIGEN:	POINTER	✓	FCLIC20	C-ORIGEN	INTEGER
DESTINO:	POINTER	✓	FCLIC20	C-DESTINO	INTEGER
RUJA:	POINTER			C-RUJA	SMALLINT
DESTINO REEXPEDICION:	POINTER			C-DESTRE	INTEGER
RUJA DE REEXPEDICION:	POINTER			C-RUJARE	SMALLINT
CLAVE AUTOTANQUE:	POINTER	✓	FCLIC20	C-AUTO	CHAR(6)
CLAVE PRODUCTO:	POINTER			C-PROD	SMALLINT
UNIDAD DEL PRODUCTO:	TOKEN			N-UNID	CHAR(1)
CANTIDAD CARGADA:	TOKEN			N-CARGA	INTEGER
TEMPERATURA DE CARGA:	TOKEN			N-TEMPAR	SMALLINT
IMPORTE NOTA DE EMBARQUE:	TOKEN			N-IMPORT	INTEGER
CANTIDAD DESCARGADA:	TOKEN			N-DESCA	INTEGER
TEMPERATURA DE DESCARGA:	TOKEN			N-TEMDES	SMALLINT
CLAVE DE STATUS:	TOKEN	✓	FCLIC20, FCLIC44	C-STATUS	CHAR(1)

Estructura Para la Tabla de Movimientos Diarios

SQL Data Type

Physical Name

Index

Foreign Key

Primary Key

Purpose

ORIGENES/DESTINOS				FCLB003	
Entity Type:ORIGENES/DESTINOS					
:ORIGEN/DESTINO	POINTER	✓	FCLIU03	C-ORIDES	INTEGER
Attribute Type:ORIGEN DESTIN					
:NOMBRE LARGO	TOKEN			D-ORIDEL	CHAR(30)
Attribute Type:NOMBRE LARGO					

82

Estructura Para la Tabla de Origenes/Destinos

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type	
: PAGOS APLICADOS					FCLBS93		
LLAVE PAGO:		✓			FCLIC22		
CLAVE PORTEADOR:	POINTER				C-PORTEA	SMALLINT	
: NUMERO DOCUMENTO	POINTER				N-DOCPA	CHAR(17)	
NUMERO DE CEDULA:	POINTER	✓			FCLIC36	INTEGER	
CLAVE TIPO DE CEDULA:	POINTER	✓			FCLIC36, FCLIC22	C-TIPOCEDU	CHAR(1)
: ORIGEN DEL REGISTRO	TOKEN				C-ORIRE0	CHAR(1)	
: FECHA DEL REGISTRO	TOKEN				F-REGIS	DATE	
: IMPORTE APLICADO	TOKEN				N-I-APLIC	DECIMAL(10,0)	
: FECHA APLICACION	TOKEN				F-APLIC	DATE	
: SALDO	TOKEN				N-I-SALDO	DECIMAL(10,0)	

Estructura Para la Tabla de Pagos Aplicados

PARTE 1 PARAMETROS				
por 1:				
LLAVE PARAMETROS: fclfc38				
GRUPO DE PRODUCTO:	POINTER		✓ C-grupo	CHAR(1)
AGENCIA PAGADORA:	POINTER		✓ c-oridee	INTEGER
FECHA INIC. VIGENCIA:	POINTER		✓ f-inipar	DATE
FECHA FIN VIGENCIA:	POINTER		✓ f-finpar	DATE
VALORES INICIALES MENSUALES:				
HOJA INICIAL DE DOCUMENTO:	TOKEN		✓ n-hojindo	SMALLINT
HOJA INICIAL ADEPAS:	TOKEN		✓ n-hojinade	SMALLINT
N.M. DOC. INICIAL:	TOKEN		✓ n-docini	SMALLINT
N.M. INICIAL ADEPAS:	TOKEN		✓ n-iniade	INTEGER
N.M. INIC. PAGO 2a PARTE:	TOKEN		✓ n-inipag2	SMALLINT
VALORES SIGUIENTE PROYECTO:				
HOJA SIG. DOCUMENTO:	TOKEN		✓ n-hojsidoc	SMALLINT
HOJA ADEPAS:	TOKEN		✓ n-hojsiade	SMALLINT
N.M. DE DOCUMENTO:	TOKEN		✓ n-docsig	SMALLINT
N.M. DE ADEPAS:	TOKEN		✓ n-sigadef	INTEGER
N.M. PAGO 1a. PARTE:	TOKEN		✓ n-sigpag1	SMALLINT
N.M. PAGO 2a. PARTE:	TOKEN		✓ n-sigpag2	SMALLINT
FECHA DE ELABORACION:	TOKEN		✓ f-elabora	DATE
DATOS GENERALES:				
CONTADURIA:	TOKEN		✓ n-ideconta	CHAR(4)
TIPO DE DOCUMENTO:	TOKEN		✓ n-tipdocum	SMALLINT
TIPO DE REGISTRO:	TOKEN		✓ n-tipreg	SMALLINT
CONCEPTO ORIGEN:	TOKEN		✓ c-conorig	INTEGER
CONCEPTO ORIGEN IVA:	TOKEN		✓ c-conorigi	INTEGER
DIGITO AVISO:	TOKEN		✓ n-digaviso	CHAR(1)
DIGITO AVISO IVA:	TOKEN		✓ n-digaviva	CHAR(1)
CLAVE DE PROYECTO:	TOKEN		✓ c-proyec	CHAR(6)

Estructura Para la Tabla de Parámetros Contables

:PARTE 2 PARAMETROS					
por 2:					
CARGO FLETES:					
CENTRO DE TRABAJO CARGO:	TOKEN		√	n-cenCarf	SMALLINT
DEPARTAMENTO DE CARGO:	TOKEN		√	n-depcarf	INTEGER
CUENTA CARGO:	TOKEN		√	n-ctacarf	SMALLINT
SUBCUENTA CARGO:	TOKEN		√	n-subcarf	CHAR(9)
CUENTA PRESUPUESTAL CARGO	TOKEN		√	n-precarf	SMALLINT
ABONO FLETES:					
CENTRO DE TRABAJO ABONO:	TOKEN		√	n-cenaboi	SMALLINT
DEPARTAMENTO ABONO:	TOKEN		√	n-depabof	INTEGER
CUENTA ABONO:	TOKEN		√	n-ctaabof	SMALLINT
SUBCUENTA ABONO:	TOKEN		√	n-subabof	CHAR(9)
CUENTA PRESUPUESTAL ABONO	TOKEN		√	n-preabof	SMALLINT
CARGO IVA:					
CENTRO DE TRABAJO CARGO I	TOKEN		√	n-cecari	SMALLINT
DEPARTAMENTO CARGO IVA:	TOKEN		√	n-depcari	INTEGER
CUENTA CARGO IVA:	TOKEN		√	n-ctacari	SMALLINT
SUBCUENTA CARGO IVA:	TOKEN		√	n-subcari	CHAR(9)
CUENTA PRESUPUESTAL IVA:	TOKEN		√	n-precari	SMALLINT
ABONO IVA:					
CENTRO DE TRABAJO ABONO I	TOKEN		√	n-cenaboi	SMALLINT
DEPARTAMENTO ABONO IVA:	TOKEN		√	n-depaboi	INTEGER
CUENTA ABONO IVA:	TOKEN		√	n-ctaaboi	SMALLINT
SUBCUENTA ABONO IVA:	TOKEN		√	n-subaboi	CHAR(9)
CTA. PRESUPUESTAL IVA:	TOKEN		√	n-preaboi	SMALLINT

SQL Data Type					
Physical Name					
REGISTRO DE AGENCIAS	INDEX		REGISTRO	REGISTRO	
REGISTRO DE AGENCIAS	Foreign Key		REGISTRO	REGISTRO	
REGISTRO DE AGENCIAS	Primary Key		REGISTRO	REGISTRO	
REGISTRO DE AGENCIAS	Purpose		REGISTRO	REGISTRO	
PARAMETROS DEL SISTEMA			FCLBS09		
CLAVE REGISTRO DE PARAMETROS:	POINTER	✓	FCLIU16	C-REG	SMALLINT
CLAVE USUARIO QUE ELABORA:	TOKEN			C-USEELA	CHAR(7)
IDENTIFICADOR DE QUIEN ELABORA:	TOKEN			N-IDELAB	SMALLINT
CLAVE USUARIO QUE REVISAS:	TOKEN			C-USEREV	CHAR(7)
IDENTIFICADOR DE QUIEN REVISAS:	TOKEN			N-IDREVI	SMALLINT
CLAVE USUARIO QUE AUTORIZA:	TOKEN			C-USEAUT	CHAR(7)
IDENTIFICADOR DE QUIEN AUTORIZA:	TOKEN			N-IDAUTO	SMALLINT
SECUENCIAL DE CONTRA-RECIBOS:	TOKEN			N-SECONT	SMALLINT

86

Estructura Para la Tabla de Parámetros del Sistema

	Purpose	Primary Key	Foreign Key	Index	Physical Name	Data Type
PEAJE TOTAL						FC1BVO6
LLAVE RUTA:			✓		FCLIC23	
ORIGEN:	POINTER					C-ORIGEN INTEGER
DESTINO:	POINTER					C-DESTINO INTEGER
RUTA:	POINTER					C-RUTA SMALLINT
:COSTO TOTAL 2 EJES	TOKEN					N-T2EJE DECIMAL(8,0)
:COSTO TOTAL 3 EJES	TOKEN					N-T3EJE DECIMAL(8,0)
:COSTO TOTAL 5 EJES	TOKEN					N-T5EJE DECIMAL(8,0)
:COSTO TOTAL 6 EJES	TOKEN					N-T6EJE DECIMAL(8,0)
:COSTO TOTAL 9 EJES	TOKEN					N-T9EJE DECIMAL(8,0)
:COSTO TOTAL ADICIONAL POR EJE	TOKEN					N-TXEJE DECIMAL(8,0)
FECHA INICIO VIGENCIA:	POINTER	✓			FCLIC23 DESC	F-INICOST DATE
FECHA FINAL VIGENCIA:	POINTER	✓			FCLIC23	F-FINCOST DATE

Estructura Para La Tabla de Peaje Total

82

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:PORCENTAJES AUTORIZADOS					FCRBG02	
:CLAVE DE AUTORIZACION	POINTER	✓			FCRIU04	C-AUTOR SMALLINT
:PORCENTAJE DE APLICACION	TOKEN				N-PORAUT	DECIMAL(4,2)
:FECHA INICIO AUTORIZACION	TOKEN				F-INIAUT	DATE
:FECHA FINAL AUTORIZACION	TOKEN				F-FINAUT	DATE

Estructura Para la Tabla de Porcentajes Autorizados

PORTEADORES		SECUNDO CLAVE		
Entity Type: CIAS TRANSPORTISTAS		Physical Name		
Primary Key		Secondary Key		
Attribute	Type	Code	Code	Code
CLAVE PORTEADOR:	POINTER	POI0011	C-POITEA	DW(11)
RAZON SOCIAL	TEXT		C-RAZON	DW(55)
Attribute Type: RAZON SOCIAL				
CLAVE GRUPO EMPRESARIAL	POINTER		C-EMPRE	DW(11)
TIPO DE EMPRESA	TEXT		N-EMPRE	DW(1)
DOMICILIO DE LA EMPRESA				
Attribute Type: DOMICILIO DE				
CALLE	TEXT		C-CALLE	DW(25)
CALLEJA	TEXT		C-CAL	DW(25)
CALLEJO	TEXT		C-CALD	DW(25)
CODIGO POSTAL	TEXT		C-CP	DW(5)
AFRANTAO POSTAL	TEXT		C-AP	DW(10)
MUNICIPIO Y ESTADO	TEXT		C-MP-ED	DW(25)
TELEFONO	TEXT		N-TELS	DW(20)
EXTENSION	TEXT		N-EXT	DW(7)
FAX	TEXT		N-FAX	DW(20)
RESPONSABLE DE LA CIA	TEXT		C-RESPON	DW(30)
Attribute Type: RESPONSABLE D				
CENTRO PASADIA	POINTER		C-ENPAR	DW(5)
FECHA DE MONTO AL EXPEDIR	TEXT		F-MONEX	DATE
TIPO SITUACION DEL REGISTRO	TEXT		C-STATUS	DW(1)
FECHA SITUACION DEL REGISTRO	TEXT		F-STATUS	DATE
STATUS EMAN/DESC	TEXT		C-STATUS	DW(1)

Estructura para la Tabla de Portadores

06

	Purpose	Primary Key	Foreign Key	Index	Physical Name	SQL Data Type
:PRECIOS					FCLBV07	
Entity Type:PRECIOS						
LLAVE PRECIO:		✓			FCLIC25	
CLAVE PRODUCTO:	POINTER				C-PROD	SMALLINT
FECHA INICIO VIGENCIA:	POINTER				F-INIPREC	DATE
FECHA FINAL VIGENCIA:	POINTER				F-FINPREC	DATE
:PRECIO PEMEX	TOKEN				N-IMPREC	DECIMAL(7,2)
Attribute Type:PRECIO PEMEX						

Estructura Para la Tabla de Precios de Productos.

PRESUPUESTO DE FLETES				
CLAVE PRESUPUESTAL	/	PLIC41	PLB513	
CLAVE TIPO DE CEDULA	POINTER		C-TIROCEDU	04F(1)
AÑO PRESUPUESTAL	POINTER		F-ANTOPRESJ	SHALLINT
PRESUPUESTO MENSUAL				
PRESUPUESTO ENERO	TO/EN		N-PRESENE	DECIMAL(15,0)
PRESUPUESTO FEBRERO	TO/EN		N-PRESFEB	DECIMAL(15,0)
PRESUPUESTO MARZO	TO/EN		N-PRESMAR	DECIMAL(15,0)
PRESUPUESTO ABRIL	TO/EN		N-PRESABR	DECIMAL(15,0)
PRESUPUESTO MAYO	TO/EN		N-PRESMAY	DECIMAL(15,0)
PRESUPUESTO JUNIO	TO/EN		N-PRESJUN	DECIMAL(15,0)
PRESUPUESTO JULIO	TO/EN		N-PRESJUL	DECIMAL(15,0)
PRESUPUESTO AGOSTO	TO/EN		N-PRESAGO	DECIMAL(15,0)
PRESUPUESTO SEPTIEMBRE	TO/EN		N-PRESSEP	DECIMAL(15,0)
PRESUPUESTO OCTUBRE	TO/EN		N-PRESOCT	DECIMAL(15,0)
PRESUPUESTO NOVIEMBRE	TO/EN		N-PRESNOV	DECIMAL(15,0)
PRESUPUESTO DICIEMBRE	TO/EN		N-PRESDIC	DECIMAL(15,0)
GASTO MENSUAL				
GASTO ENERO	TO/EN		N-GASENE	DECIMAL(15,0)
GASTO FEBRERO	TO/EN		N-GASFEB	DECIMAL(15,0)
GASTO MARZO	TO/EN		N-GASMAR	DECIMAL(15,0)
GASTO ABRIL	TO/EN		N-GASABR	DECIMAL(15,0)
GASTO MAYO	TO/EN		N-GASMAY	DECIMAL(15,0)
GASTO JUNIO	TO/EN		N-GASJUN	DECIMAL(15,0)
GASTO JULIO	TO/EN		N-GASJUL	DECIMAL(15,0)
GASTO AGOSTO	TO/EN		N-GASAGO	DECIMAL(15,0)
GASTO SEPTIEMBRE	TO/EN		N-GASEP	DECIMAL(15,0)
GASTO OCTUBRE	TO/EN		N-GASOCT	DECIMAL(15,0)
GASTO NOVIEMBRE	TO/EN		N-GASNOV	DECIMAL(15,0)
GASTO DICIEMBRE	TO/EN		N-GASDIC	DECIMAL(15,0)

Estructura Para La Tabla de Presupuesto de Fletes

	Purpose	Primary Key	Foreign Key	Index	Database Field	Physical Name	SQL Data Type
: PRIVILEGIOS						FCLBG13	
CLAVE USUARIO:	POINTER		✓		FCLIU30	✓ C-USER	CHAR(7)
CLAVE DEL PATH:	TOKEN					✓ C-PATH	INTEGER
CLAVE DEL MODULO:	TOKEN					✓ C-MODULO	SMALLINT
NUMERO SECUENCIAL DEL EVENTO:	VALUE					✓ N-SECUENCIAL	INTEGER

ESTRUCTURA PARA LA TABLA DE PRIVILEGIOS

	Purpose	Primary Key	Foreign Key	Index	Physical Name	Std. Data Type
PRODUCTOS CARGA LIQUIDA						FCLB602
Entity Type: CARGA Y/O PRODUCTOS						
CLAVE PRODUCTO:	POINTER	✓			FCLIU12	C-PROD SMALLINT
:DESCRIPCION DEL PRODUCTO	TOKEN					D-PROD CHAR(30)
Attribute Type: DESCRIPCION L						
UNIDAD DEL PRODUCTO	TOKEN					N-UNID CHAR(1)
PESO ESPECIFICO:	TOKEN					N-PESP DECIMAL(5.4)
:COEFICIENTE DE EXPANSION	TOKEN					N-COEFE DECIMAL(7.6)
:FACTOR F	TOKEN					N-FACTF DECIMAL(6.5)
GRUPO DE PRODUCTO:	POINTER					C-GRUPO CHAR(1)
:TIPO DE CUOTA	POINTER					C-CUOTA SMALLINT
CAPACIDAD PROMEDIO:	TOKEN					N-CAPAP INTEGER
FECHA MOVIMIENTO:	TOKEN					F-MOVEXP DATE
CLAVE STATUS REG:	TOKEN					C-STATUS CHAR(1)
FECHA STATUS REG:	TOKEN					F-STATUS DATE

8.6

Estructura Para la Tabla de Productos C.L.

ETAPA 7

Finalmente, se realizó un Diagrama de estructura el cual muestra una vista del sistema como un sistema de módulos jerárquicos que llaman o contienen módulos subordinados donde se que pueden mostrar inclusive, los datos que son pasados de un módulo a otro. Aquí se da la facilidad de documentar las actividades que realiza cada módulo que pertenece al sistema, generando de esta forma las especificaciones de los módulos para su desarrollo e implementación.

A continuación se muestra una vista general del primer y segundo nivel de los módulos principales del Diagrama de Estructura (Ver fig. III.2.1.7), que nos permite ver los módulos que integran al sistema, así como una relación de los mismos generada automáticamente por IEW.

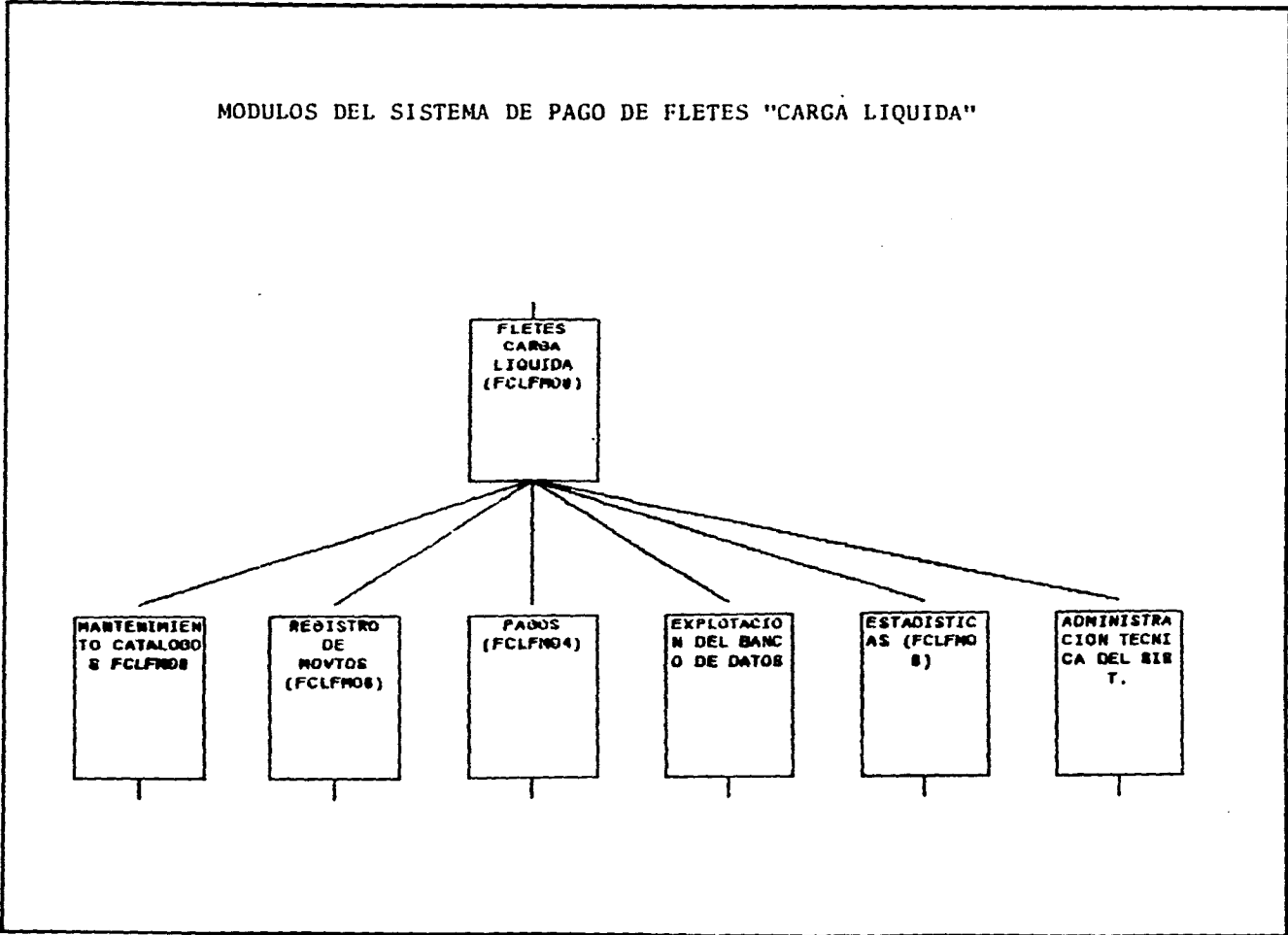


Fig.III.2.1.7 Diagrama de Estructura del SPF_AT generado por IEV

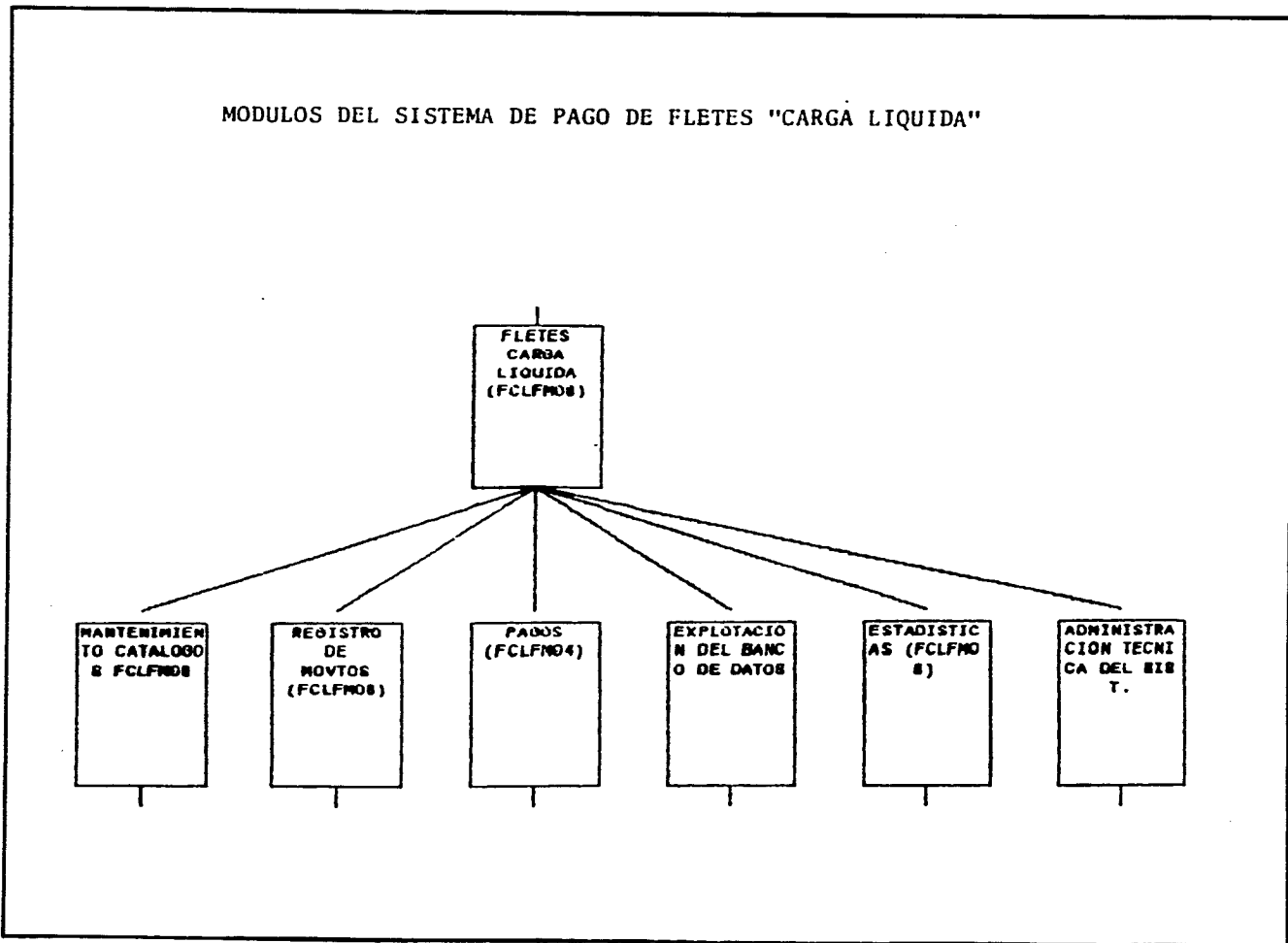


Fig. III.2.1.7 Diagrama de Estructura del SPF_AT generado por IEW

Petroleos Mexicanos C.A.S.E

Object List

(Lista de Objetos tipo Modulo Generada por IEW)

module (fclfc01) consulta cat. portea
module (fclfc03) consulta cat. autotan
module (fclfc05) consulta cat. ori/dest
module (fclfc06) consulta cat.casetas
module (fclfc07) consulta cat.bonificac
module (fclfc08) consulta cat. de prod.
module (fclfc09) consulta cat. cuotas
module (fclfc10) consulta cat.tarifas e
module (fclfc12) consulta cat.de rutas
module (fclfc14) consulta cat.peaje tot
module (fclfc15) consulta cat.calibrac.
module (fclfc16) consulta cat.costos c.
module (fclfc17) consulta cat. precios
module (fclfc26) consulta cat.marcas t.
module (fclfc33)consulta prod's x ruta
module (fclfr01) cat. portedores
module (fclfr03) cat. autotanques
module (fclfr05) cat. orig/dest
module (fclfr06) cat. puentes y casetas
module (fclfr07) cat. concep. bond/desc
module (fclfr08) cat. de productos
module (fclfr09) rep. cat. cuotas
module (fclfr10)cat. tarifas especiales
module (fclfr12) cat. caract.por ruta
module (fclfr13) cat. peaje x caseta
module (fclfr14) casetas por ruta
module (fclfr15) calibraciones auto.
module (fclfr16) cat costos casetas y p
module (fclfr17)cat. precios de produc.
module (fclfr26) cat. marcas trac/auto
module (fclfr33) rep. prod's x ruta
module act.recep/dep.de ceds. (fclfa39)
module acumula en historico (fclfa40)
module acumula mensual (fclfa41)
module acumulados historicos
module administracion tecnica del sist.
module analitico de peaje (fclfr57)
module analitico por liquidac.(fclfr52)
module ascii para contabilidad fclfg02

Subgerencia de Distribucion

Petroleos Mexicanos C.A.S.E
Object List
(Lista de objetos tipo Modulo generada por IEW)

module ascii para finanzas (fclfg03)
module auditoria transacciones(fclfr81)
module autotanques (fclfa03)
module batch's para ibm (fclfg01)
module bonif y/o descetos (fclfr56)
module bonificaciones/desc (fclfa18)
module calculo (fclfa31)
module calibraciones (fclfa15)
module captura batch's (fclfk01)
module captura de embarques
module características rutas (fclfa12)
module casetas por ruta (fclfa14)
module cedula pago (fclfr53)
module cédulas (fclfa27)
module conceptos bonif-desc (fclfa07)
module cons. batch's capturados fclfc63
module cons. hist. emb. pag. fclfc23
module consulta a liquidaciones fclfc62
module consulta a tarifas (fclfc60)
module consulta de mov. diarios(fclfc61)
module consulta presup mensual fclfc64
module consultas
module consultas estadísticas
module consultas on-line
module contra-recibos (fclfr55)
module correc. nums. embarques fclfa35
module correcc.a la captura (fclfk04)
module costos (fclfa16)
module cuotas (fclfa09)
module embarques reportados (fclfa19)
module estadis. p/gpo de produc fclfr87
module estadist prod-orig-dest fclfr85
module estadist. orig-dest-prod fclfr84
module estadist. p/gpo empresa fclfr86
module estadísticas (fclfm08)
module estadístico mens. gral. fclfr83
module estadístico por producto fclfr82
module explotación del banco de datos
module fletes carga líquida (fclfm08)

Subgerencia de Distribucion

Petroleos Mexicanos C.A.S.E

Object List

(Lista de objetos tipo Modulo Generada por IEW)

module fletes carga regular
module generacion de cedula (fclfa24)
module grupos de productos (fclfa11)
module grupos empresariales (fclfa02)
module interfases
module invent. de programas (fclfa20)
module lanzador de mecanizado (fclfm01)
module mantenimiento a catalogos
module mantenimiento usuarios (fclfa21)
module marcas tractores y at (fclfa26)
module monitoreo de procesos (fclsh04)
module origenes destinos (fclfa05)
module pagos (fclfm04)
module param. contab.y presup.(fclfa37)
module parametros del sistema (fclfa32)
module peaje total (fclfa13)
module perfiles de usuarios (fclfa22)
module porteadores (fclfa01)
module precios (fclfa17)
module presup. mensual de ftes. fclfa38
module product. x ruta (fclfa33)
module productos c. liquida (fclfa08)
module puentes y casetas (fclfa06)
module rechazados
module registro de movtos (fclfm08)
module rep gral tarif. y cuotas cl1fr09
module rep. gral autotanques (cl1fr03)
module rep. gral casetas x ruta cl1fr14
module rep. gral. calibracion (cl1fr15)
module rep. gral. origen/dest (cl1fr05)
module rep. gral. peaje total (cl1fr13)
module rep. gral. porteadores (cl1fr01)
module rep. gral. precios (cl1fr17)
module rep. gral. rutas (cl1fr12)
module reportes
module reportes de estadisticos
module reportes estadisticos
module reportes fletes c. liq.
module resp. ced's y liqui's (fclfg07)

Subgerencia de Distribucion

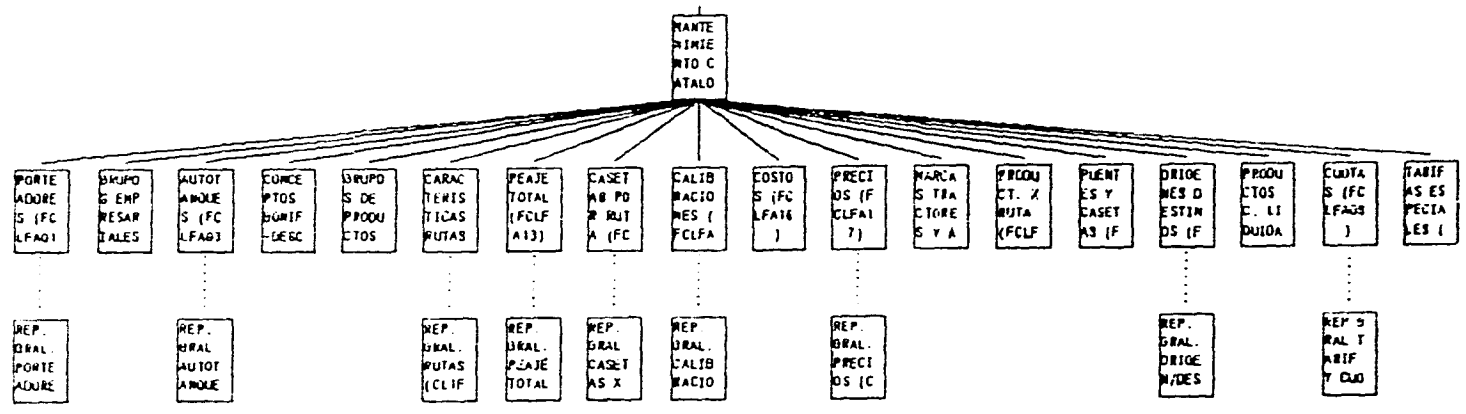
Petroleos Mexicanos C.A.S.E

Object List

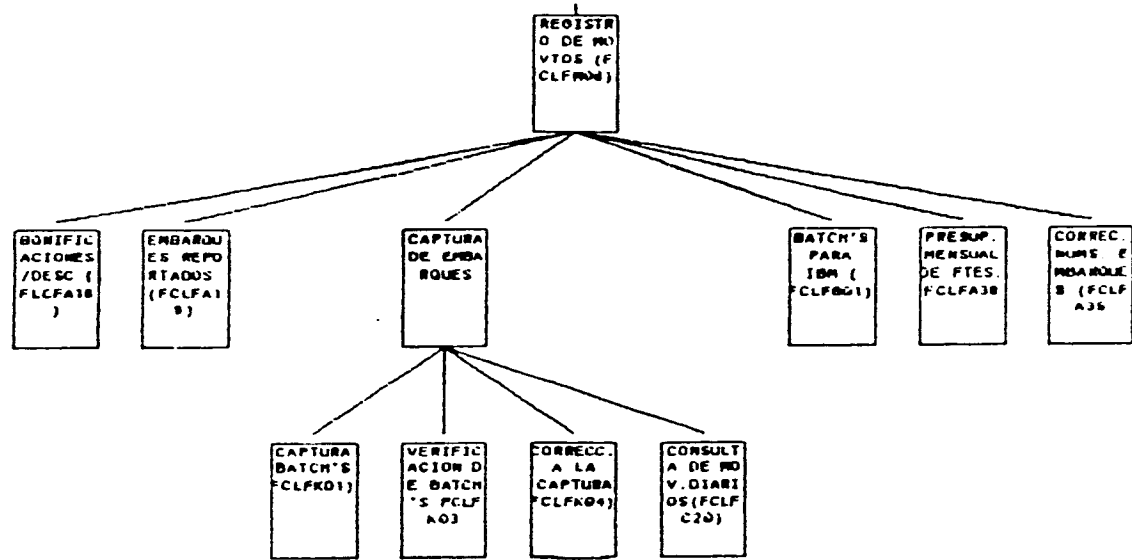
(Lista de objetos tipo Modulo Generada por IEW)

module resp. embarq. pagados (fclfg08)
module resp. en disk. cont/fin. fclfg05
module resp. movtos.diaros (fclfg06)
module resp.bonif.y pag.aplic.(fclfg09)
module respaldos de informacion
module restaura cedula (fclfa50)
module sistema de fletes (fclfm08)
module sistema de pago a transportistas
module tarifas especiales (fclfa10)
module validacion definitivo (fclfr51)
module verificacion de batch's fclfk03

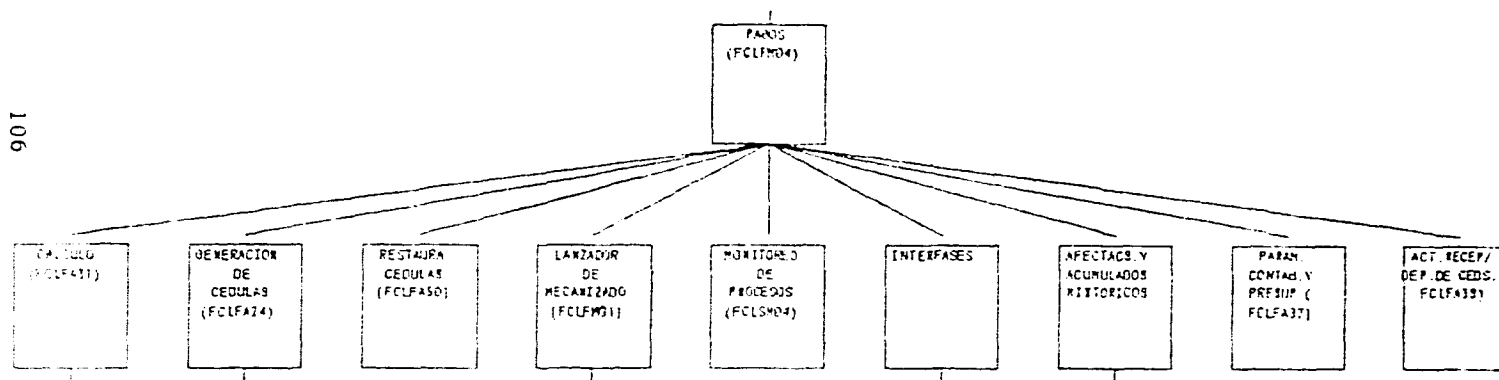
Subgerencia de Distribucion



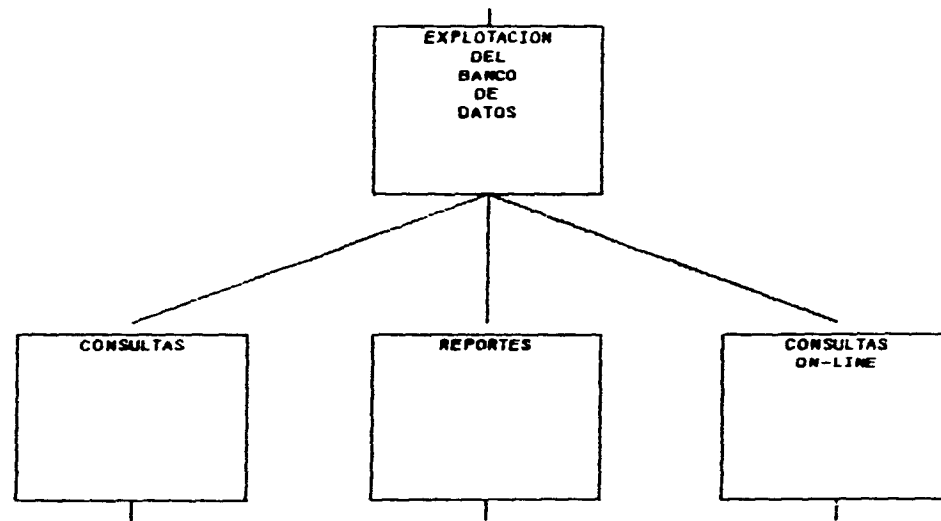
Módulos de la Vista Mantenimiento a Catálogos



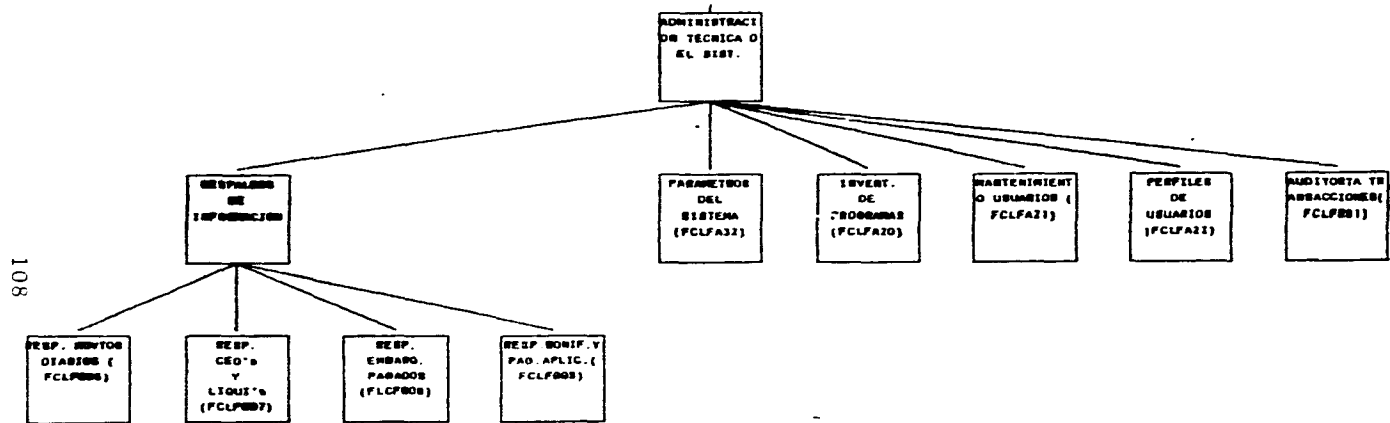
Módulos de la Vista Registro de Movimientos



Módulos de la Vista Generación de Pagos



Módulos de la Vista de Explotación del Banco de Datos



Módulos de la Vista Administración Técnica del Sistema

III.2.2 DESARROLLO

Para llevar a cabo el presente proyecto, a continuación se describen las etapas principales para su desarrollo e implantación.

1) Establecimiento de la administración del proyecto que incluye :

- Definición de grupos de trabajo.
- Definición de responsabilidades.
- Establecimiento de mecanismos de evaluación del proyecto.
- Definición de estándares.
- Políticas de operación de los equipos.

2) Definición de las bases funcionales del sistema:

- Volúmenes y transacciones actuales.
- Dimensionamiento de equipo
- Recopilación de Información.

3) Diseño detallado del sistema:

- Identificación y Diseño de los productos del Sistema.
- Diseño de la Estructura Física del Sistema.
- Diseño Funcional del Sistema.
- Diseño de las Interfases con otros Sistemas.

4) Construcción y Pruebas del Sistema.

5) Implantación y Liberación del Sistema.

6) Documentación Automática en Paralelo al Desarrollo del Sistema.

III.2.2.1 DESCRIPCION DE LA ESTRATEGIA DE DESARROLLO

DEFINICION DE GRUPOS DE TRABAJO

Se establecieron 2 grupos de trabajo, uno integrado por personal del area donde operaría el sistema y otro por el area de desarrollo. De esta manera se tuvo muy cerca el análisis y el establecimiento de las necesidades reales de los usuarios finales.

DEFINICION DE ESTANDARES

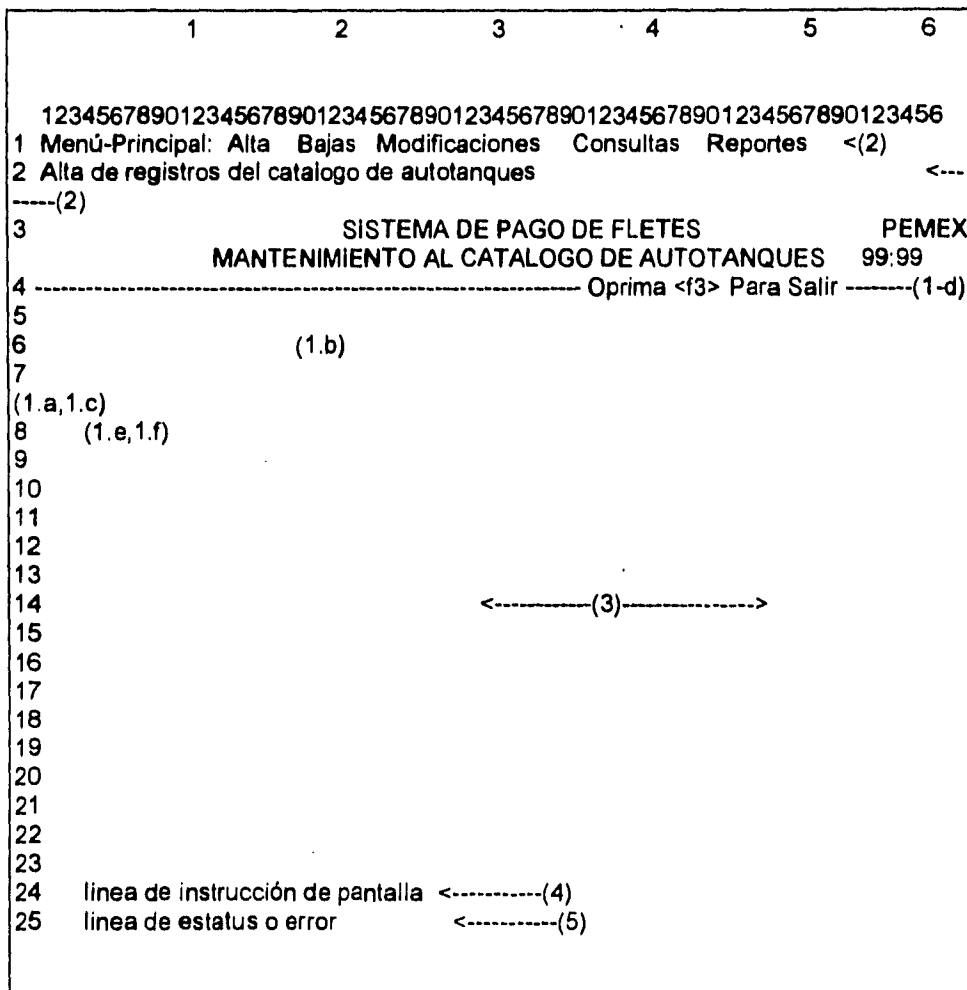
Con el propósito de normar el desarrollo del sistema y unificar la vista final; es decir los formatos de las pantallas y los diversos reportes que se emitirían hacia los usuarios finales, se pretendió definir y establecer los estandares de programación, que deberían ser observados en el diseño del sistema en ambiente UNIX, y específicamente con el manejador de base de datos Informix.

DISEÑO DE FORMATOS DE PANTALLA

Este estandar tiene como propósito presentar de manera uniforme las pantallas de las aplicaciones, identificandolas por su tipo y los datos.

Las pantallas de transacción deberán ser constituidas por las siguientes áreas en el orden que se presentan (Ver pantalla de transacción

PANTALLA DE TRANSACCION



1 Encabezados

En esta área deberán incluirse los siguientes datos:

- a fecha
- b nombre de la aplicación
- c hora
- d identificación de la pantalla
- e proceso que se está ejecutando
- f identificación de la terminal

2 Menu

En esta área se deberá desplegar el menu del tipo Pull-Down. Las opciones podrán elegirse a través de flechas o con una letra distintiva de la opción y por lo tanto no deberá repetirse.

3 Aplicación

En esta área se llevará a cabo la transacción sea cual sea su naturaleza, ocupando la parte central de la pantalla.

4 Instrucción

Esta área es destinada para indicar las teclas habilitadas y/o clarificar la función de determinada opción.

5 Estatus

Esta área se ubica al final de la pantalla en la cual se deberán desplegar los mensajes informativos o de error.

6 Salida

La salida de la aplicación deberá ser a través del menú principal.

CONVENCION DEL TECLADO

TECLAS

- F1** Solicitar ayuda (si la aplicación la maneja).

- F3** Confirmar la captura de datos.

- ESCAPE** Cancelar operación y regresar al menú anterior.

DISEÑO DE FORMATOS DE REPORTES

Tiene como propósito presentar de manera uniforme el diseño para reportes de las aplicaciones, identificando los datos y áreas comunes que deben contener. (Ver formato de reportes)

1 Area de encabezado

Esta contiene los siguientes datos:

- a Leyenda "PETROLEOS MEXICANOS".
- b Nombre del departamento.
- c Nombre oficial de la aplicación.
- d Identificador o título del reporte.
- e Fecha de emisión.
- f Identificación módulo que emite el reporte.
- g Otras fechas relativas al proceso en particular, como día de proceso, rango de fechas de los movimientos incluidos (opcional).
- h Numero de página.

2 Area de información

En esta área se reportan los datos del reporte

- a Cada columna de datos debe tener encabezado descriptivo de la información contenida. Este renglon deberá tener una línea punteada arriba y abajo del encabezado.
- b Detalle de la información.
- c Cuando se incluyan subtotales, describir claramente el criterio de subtotalización mediante la leyenda correspondiente.

3 Area de pies de página (opcional).**4 Area de terminación de reporte**

Incluir datos sumarios del reporte, cuando proceda.

En esta zona pueden incorporarse leyendas relativas a las limitaciones de la información, como exclusión de movimientos capturados posterior a la fecha de actualización de los archivos utilizados, o referencias del reporte.

ASIGNACION DE NOMBRES A ARCHIVOS UNIX (FUENTES, SHELL, ETC,)

Tomando como premisa que el diseño de sistemas establece una estructura jerárquica de vistas y /o procesos, los nombres de los archivos deberán identificar fácilmente las dependencias y ubicación de estos, en la estructura del sistema, así como la función que realizan.

REPORTE: XXXXXXXXXXXXXXXXXXXX <(1.f)
OFICINA : XXXXXXXXXXXXXXXXXXXX <(1.i)

PETROLEOS MEXICANOS
GERENCIA DE DISTRIBUCION
CATALOGO DE AUTOTANQUES

<-(1.a) (1.e) FECHA: DD/MM/AA
<-(1.b) (1.e)-> HOJA : HH:MM:SS

<----->
(1.k)

REFERENCIA	DESCRIPCIÓN
------------	-------------

115

XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

<----->
(2.b)

FIN DE REPORTE <----->
(4)

Para ello se deberán observar las siguientes reglas:

La longitud de los nombres deberá ser invariables de 8 caracteres más una extensión de 3 caracteres.

La estructura general es la siguiente:

fclttnn.ext

donde:

- fcl Es el identificador del sistema Fletes Carga Liquida
- t Es el tipo de función o actividad que realiza el archivo, y los valores permitidos son.
 - m Mantenimiento.
 - b Backup
 - c Calculo
 - k Captura
 - h Ayuda
 - l Libreria
 - r Reporte
 - a Actualizador
 - s Shell.
- nn Son digitos del 0 al 9 que identifican el número secuencial del programa.
- ext Indica el tipo de archivo.

No obstante que en el ambiente UNIX no existe la extensión de un archivo como tal, y por razones técnicas del manejador de base de datos Informix, que requiere una extensión para identificar los tipos de archivos que maneja, por lo tanto se deberá invariablemente anexarse la extensión al archivo, de acuerdo a la siguiente tabla:

- .4gl** programa fuente en 4GL
- .sql** programas fuentes en SQL
- .per** formatos de pantalla fuentes para informix
- .c** programas fuentes c
- .ace** programas fuentes del reporteador de informix
- .sh** shell's en UNIX
- .dat** archivos planos de datos en ASCII
- .txt** archivos de texto(reportes, documentos, etc..)
- .o** archivos de código objeto
- .4gi** programas ejecutables de informix rds
- .exe** programas ejecutables de informix-4gl y cualquier otro lenguaje
- .frm** formatos de pantalla ejecutable para informix
- .db** tabla de base datos.

ESTANDARES DE PROGRAMACION (ESTILO, VARIABLES, COMENTARIOS, ETC..)

El propósito del presente apartado sirve para uniformar y facilitar las funciones del programador, estableciendo normas para el desarrollo de sistemas, con el fin de simplificar la labor de mantenimiento a los mismos en un ambiente de base de datos.

GENERALES

- 1 La programación se hará bajo el concepto de programación estructurada.
- 2 Cualquier mensaje emitido al usuario del programa, así como, la documentación interna de módulos, deberá escribirse en español.
- 3 El líder de proyectos será responsable de autorizar cualquier desviación del estándar cuando el sistema lo requiera.

AUTODOCUMENTACION

Para cada programa, o módulo elaborado, incluir los siguientes elementos de documentación:

- 1 Cabecera de programas:
 - Sistema
 - Programa
 - Funciones
 - Nombre del autor
 - Fecha de elaboración
 - Descripción de parámetros
 - Variables globales afectadas
 - Modificaciones
 - * Autorizo
 - * Autor:
 - * Fecha
 - *Descripción del cambio

2 Comentarios a los bloques de código que:

- Realicen manipulación de datos importantes.
- Realicen manejo de excepciones.
- Utilice espacios y líneas en blanco, alrededor de comentarios para aumentar su legibilidad.

3 Usar nombres descriptivos (mnemónicos) para las variables, subprogramas, pantallas y reportes.

NOMBRES DE VARIABLES Y DE RUTINAS

1 La longitud máxima permitida para el nombre de variables, será hasta de 18 caracteres, aunque sólo los primeros ocho serán significativos.

2 Los nombres de variables, programas, subprogramas, pantallas y reportes serán descritos en minúsculas.

3 Las siguientes convenciones serán usadas para nombrar a las variables de memoria utilizadas en la programación.

La sintaxis para la asignación de nombres de variables será la siguiente:

a_nombre mnemónico

donde:

a Es el alcance o temporalidad de la variable; los valores posibles pueden ser:

g = Cuando se trate de una variable global.

p = Cuando se trate de un parámetro.

w = Cuando se trate de una variable local del módulos o función.

ejemplo:

g_bandera

w_fecha

MANEJO DE RUTINAS

Debe de evitarse en lo posible, el uso de rutinas, sin embargo, el paso de valores entre una rutina invocada y el programa que lo llama será mediante el uso de parámetros de la rutina.

Ejemplo:

call funcion1(parametro1, parametro2,...)

returning var1, var2

La estructura de los programas. rutinas o funciones deberán seguir el siguiente modelo:

Para el programa principal:

#Cabecera del programa

 DATABASE xxxxxxx

 MAIN

Definición de variables

 DEFINE

Cuerpo del programa

 END MAIN

DEFINICION DE LAS BASES FUNCIONALES DEL SISTEMA

VOLUMENES DE INFORMACION ACTUALES

ESTRUCTURAS CON UN RANGO DE 50 A 100 REGISTROS

BONIFICACIONES/DESCUENTOS	FCLBV01
CONCEPTOS DE BON/DESC	FCLBG06
GRUPOS DE PRODUCTOS	FCLBG04
GRUPOS EMPRESARIALES	FCLBG07
INVENTARIO DE TABLAS DEL SIST.	FCLBG15
MARCAS DE TRACTORES Y TANQUES	FCLBG08
MODULOS DEL SISTEMA	FCLBG12
PARAMETROS CONTABLES	FCLBS10
PARAMETROS DEL SISTEMA	FCLBS09
PORTEADORES	FCLBG10
PRIVILEGIOS	FCLBG14
USUARIOS	FCLBG13

ESTRUCTURAS CON UN RANGO DE 100 A 3500 REGISTROS

AUTOTANQUES	FCLBG05
ORIGENES/DESTINOS	FCLBG09
BITACORA DE TRANSACCIONES	FCLBS06
CALIBRACIONES SITUACIONES	FCLBV02
CARACTERISTICAS RUTAS	FCLBN01
CASSETAS POR RUTA	FCLBV03
CONTROL DE BATCH'S	FCLBV12
MOVIMIENTOS RECHAZ. POR CALC.	FCLBV11
PAGOS APLICADOS	FCLBS03
PRECIOS	FCLBV07
PRESUPUESTO DE FLETES	FCLBS13
PRODUCTOS C. LIQ.	FCLBG02
PRODUCTOS TRANSPORTADOS COMUNMENTE POR RUTA	FCLBN02
PUNTES Y CASSETAS	FCLBG11
TARIFAS ESPECIALES	FCLBV08

ESTRUCTURAS CON UN RANGO DE 100000 A 300000 REGISTROS

ACUMULADO MENSUAL EMB. PAG.	FCLBS12
CEDULAS DE PAGO	FCLBS08
COSTOS DE CASSETAS Y PUNTES	FCLBV04
CUOTAS PEMEX	FCLBG01
EMBARQUES C.L.	FCLBS04
EMBARQUES NO PAGADOS	FCLBS02
EMBARQUES PAGS. RECORTADO	FCLBS11
EMBARQUES REPORTADOS	FCLBV05
HISTORICO EMB. C.L. ANTERIOR	FCLBS15
HISTORICO EMB. C.L. SELECCIONADO	FCLBS14
HISTORICO EMBARQUES PAGADOS	FCLBS01
HISTORICO DE TRANSACCIONES	FCLBS07
LIQUIDACIONES	FCLBS05
MOVIMIENTOS DIARIOS	FCLBV09
MOVIMIENTOS RECHAZADOS	FCLBV10
PEAJE TOTAL	FCLBV06

DIMENSIONAMIENTO DEL EQUIPO

Para la realización y desarrollo del proyecto, fué necesario contar con el equipo de cómputo y el software que se describe a continuación:

- Computador multiusuario
- 16 MB de memoria
- Disco de 1 GB
- Unidad de disco flexible 3.5" y 1.4 MB
- Unidad de Cinta de 150 MB
- Capacidad de 24 usuarios
- Puerto de Comunicaciones de alta velocidad
- 8 Terminales
- 4 impresoras de velocidad media, 800 líneas por minuto.
- 1 impresora de alta velocidad , 1200 líneas por minuto.
- Sistema Operativo UNIX
- Compilador C
- Base de Datos Informix/Online
- Compilador RM COBOL
- Paquete de comunicaciones UNIX-UNIX, UNIX-IBM, UNIX-PC

RECOPIACION DE INFORMACION

La recopilación de la información se realizó de los datos almacenados en la IBM- 3090, de los cuales se extrajeron las estructuras implementadas para el nuevo sistema. Se realizó la creación de archivos planos , para ser transferidos y cargados en la nueva base de datos.

Las estructuras que fueron creadas con información no tomada en cuenta con anterioridad, fueron capturados por primera vez aclarando que fué una minoría; además de ser las de menos cantidad de registros; alguna otra información se extrajo de pequeños sistemas de control llevados por PC, pudiendo pasar los datos necesarios al nuevo ambiente.

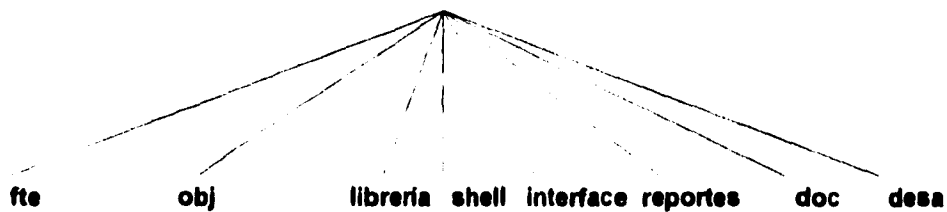
III.2.2.1 DESARROLLO DEL SISTEMA

En primera instancia se detectaron los procesos solicitados al sistema, identificandose principalmente:

- Mantenimiento a Catálogos
- Modulos de proceso
- Modulos de registro de movimientos
- Modulos de interfase con otros Sistemas
- Módulos de Explotación de Información.

III.2.2.2 DISEÑO DE LA ESTRUCTURA FÍSICA DEL SISTEMA

El sistema de pago de fletes carga líquida, se desarrolló en el ambiente del sistema operativo unix, para ello se estableció la siguiente estructura de directorios que da soporte a la aplicación.



- fte.-** contiene los programas y formas fuentes
- obj.-** contiene los programas y formas ejecutables
- librería.-** contiene programas y formas fuente, que son de uso común
- shell.-** contiene programas tipo shell que utiliza el sistema
- interface.-** aqui se generan los archivos ascii para contabilidad y finanzas
- reportes.-** aqui se depositan todos los reportes que genera el sistema y que posteriormente seran enviados a la impresora.
- doc.-** contiene los archivos de la documentación del sistema (manual de operacion).
- desa.-** contiene los directorios de desarrollo de los programadores asignados al sistema.

III.2.2.3 DISEÑO FUNCIONAL DEL SISTEMA

El establecimiento funcional del sistema, se estableció en base a un mecanismo de seguridad en el acceso al sistema. Dicho mecanismo está apoyado en un Generador de Menús dinámicos, que permite crear perfiles a los usuarios de acuerdo a las funciones operativas del mismo. El acceso al sistema, es de acuerdo a los privilegios otorgados a los usuarios.

III.2.2.4 DISEÑO DE INTERFASES CON OTROS SISTEMAS

Se estableció la generación de archivos planos, con datos para el área de finanzas y de control presupuestal, de acuerdo a las estructuras establecidas por ambas áreas y en el periodo establecido. Esta información se tomaría de los datos generados por el proceso de Pago de Fletes a transportistas.

III.2.2.5 IMPLEMENTACION Y PRUEBAS

La etapa de Implementación del sistema se inició, una vez que se tenía la fase del diseño de la Base de Datos, al contar con la generación y documentación de especificaciones para los módulos de mantenimiento, módulos de captura y los procesos de generación de pago.

Otra etapa fue la del establecimiento del método de la administración técnica, en la que intervendría también el respaldo de la información capturada; así como el control de transacciones hechas al sistema. Por último fueron implementados los módulos pertenecientes a la explotación del banco de datos y las interfaces a otros sistemas.

Las pruebas se realizaron con datos reales, siendo lo más importante el proceso de captura con 5 usuarios a la vez, además de lanzar los procesos de cálculo de pago y generación de mecanizados (cédula de pago, peaje total, generación de liquidaciones etc.,)

III.2.2.6 LIBERACION

La implementación y liberación de los productos del sistema , se realizó de manera paulatina a medida que se terminaban modulos del sistema, ya que se probaban y se integraban a la vista general del sistema, dando principalmente una estructura funcional e integral al sistema.

III.2.2.7 DOCUMENTACION

La documentación del sistema se realizó de manera simultánea a las etapas de análisis y diseño pues es una capacidad que nos brinda una herramientas CASE. A continuación se presentan una serie de documentos que permiten mostrar la información que se tiene como resultado de la fase de documentación que se genera con la utilización de una herramienta CASE.

III.2.3 DICCIONARIO DE DATOS

El diccionario de datos es parte de la documentación que nuestra herramienta CASE genera de manera automática a continuación se presenta una relación y un resumen de los datos utilizados en el sistema.

RELACION DE TABLAS E INDICES DEL SISTEMA DE CARGA LIQUIDA

Nombre Lógico	Nombre Físico	Indice	Tipo	Columnas
ACUMULADO MENS. EMB. PAG.	FCLBS12	FCLIC40	UNICO	f_anio,f_mes,c_origen,c_destino,c_prod,c_grupo,c_portea
AUTOTANQUES	FCLBG05	FCLIC01	UNICO	c_auto,c_portea
BITACORA DE TRANSACCIONES	FCLBS06	FCLIC52	DUPS	f_evento,c_tabla,n_time
BONIFICACIONES/DESCUENTOS	FCLBV01	FCLIC03 FCLIC04 FCLIU01	UNICO DUPS DUPS	c_portea,n_docpa c_portea,n_i_saldo(desc) c_concep
CALIBRACIONES SITUACIONES	FCLBV02	FCLIC05	UNICO	c_auto,c_portea,f_inicalb (desc), f_finicalb,n_tipomov
CARACTERISTICAS RUTAS	FCLBV03	FCLIC06	UNICO	c_origen,c_destino,c_ruta, f_iniruta(desc),f_finruta
CASSETAS POR RUTA	FCLBN01	FCLIC07 FCLIC08 FCLIU02	UNICO DUPS DUPS	c_origen,c_destino,c_ruta,c_ca_pu c_origen,c_destino,c_ruta. c_ca_pue
CEDULA DE PAGO	FCLBS08	FCLIC09	DUPS	n_cédula,c_tipocedu,c_portea
CONCEPTOS DE BON/DESC	FCLBG06	FCLIU03	UNICO	c_concep
COSTOS	FCLBV04	FCLIC10 FCLIU04	UNICO UNICO	c_ca_pue,f_inicost(desc),f_fincost c_ca_pue
CONTROL DE BATCH'S	FCLBV12	FCLIU18 FCLIC32	DUPS DUPS	n_batch c_user,n_ident,f_batch
CUOTAS PEMEX	FCLBG01	FCLIC11	UNICO	n_kms,f_icuota(desc),f_fcuota
EMBARQUES C L	FCLBS04	FCLIC12 FCLIC13 FCLIC14 FCLIC42 FCLIU15	UNICO DUPS DUPS UNICO DUPS	c_portea,n_embarq,f_embarq(desc) c_portea,n_liqui n_cédula,c_tipocedu,f_proces(desc) n_embarq,f_embarq n_embarq
EMBARQUES NO PAGADOS	FCLBS02	FCLIC15	UNICO	c_portea,n_embarq,f_embarq(desc)

Capítulo III**Diseño y desarrollo del sistema**

Nombre Lógico	Nombre Físico	Índice	Tipo	Columnas
HISTORICO EMB. PAG. C.L.	FCLBS01	FCLIC16	UNICO	c_portea,c_letemb,n_digemb, f_embarq(desc)
		FCLIC17	DUPS	c_portea,n_liqui
		FCLIC29	UNICO	c_letemb,n_digemb,f_embarq(desc)
EMBARQUES REPORTADOS	FCLBV05	FCLIC18	UNICO	n_embini,n_embfin
GRUPOS DE PRODUCTOS	FCLBG04	FCLIU05	UNICO	c_grupo
GRUPOS EMPRESARIALES	FCLBG07	FCLIU06	UNICO	c_empre
EMB. PAGADOS RECORTADO	FCLBS11	FCLIC39	UNICO	n_embarq,f_embarq
		FCLIU17	DUPS	n_embarq
HIST. EMB. C.L. SELECCIONADO	FCLBS14	FCLIC46	UNICO	c_portea,c_letemb,n_digemb, f_embarq(desc)
		FCLIC47	DUPS	c_portea,n_liqui
		FCLIC48	UNICO	c_letemb,n_digemb,f_embarq
HIST. EMB. C.L. ANTERIOR	FCLBS15	FCLIC49	UNICO	c_portea,c_letemb,n_digemb, f_embarq(desc)
		FCLIC50	DUPS	c_portea,n_liqui
		FCLIC51	UNICO	c_letemb,n_digemb,f_embarq
HISTORICO DE TRANSACCIONES	FCLBS07	FCLIC45	UNICO	n_secuen, c_registro
INVENTARIO DE PROGRAMAS	FCLBG12	FCLIU14	UNICO	c_modulo
INVENTARIO DE TABLAS	FCLBG15	FCLIU08	UNICO	c_tabla
LIQUIDACIONES	FCLBS05	FCLIC19	DUPS	n_cédula,c_tipocedu,c_portea,n_liq
		FCLIC35	DUPS	n_cédula,c_tipocedu
MARCAS DE TRACT. Y TANQ.	FCLBG08	FCLIC24	UNICO	c_martra,c_tantra
MOVIMIENTOS DIARIOS	FCLBV09	FCLIC20	UNICO	n_cédula,c_status,c_tipobatch, c_origen,c_destino,c_auto
		FCLIC44	UNICO	n_batch,c_status,n_numreg
MOVIMIENTOS RECHAZADOS	FCLBV10	FCLIC21	UNICO	c_portea,n_embarq,f_embarq
MOV.RECHAZADOS POR CALC.	FCLBV11	FCLIC28	UNICO	c_portea,n_embarq,f_embarq
ORIGENES/DESTINOS	FCLBG09	FCLIU09	UNICO	c_ondes
PAGOS APLICADOS	FCLBS03	FCLIC22	UNICO	c_portea,n_docpa,n_cédula, c_tipocedu
		FCLIC36	DUPS	n_cédula,c_tipocedu

Capítulo III**Diseño y desarrollo del sistema**

Nombre Lógico	Nombre Físico	Índice	Tipo	Columnas
PARAMETROS CONTABLES	FCLBS10	FCLIC38	UNICO	c_grupo,c_orides,f_inipar,f_finpar
PARAMETROS DEL SISTEMA	FCLBS09	FCLIU16	UNICO	c_reg
PEAJE TOTAL	FCLBV06	FCLIC23	UNICO	c_origen,c_destino,c_ruta, f_inicost(desc),f_fincost
PORTEADORES	FCLBG10	FCLIU11	UNICO	c_portea
PRESUPUESTO FLETES	FCLBS13	FCLIC41	UNICO	c_tipocedu,f_aniopresu
PRECIOS	FCLBV07	FCLIC25	UNICO	c_prod,f_iniprec,f_finprec
PRIVILEGIOS	FCLBG14	FCLIC27	UNICO	c_user,c_path,n_sec
PRODUCTOS TRANSPORTADOS COMUNMENTE POR RUTA	FCLBN02	FCLIC37	UNICO	c_origen,c_destino,c_ruta, c_prod
PRODUCTOS C. LIQ.	FCLBG02	FCLIU12	UNICO	c_prod
PUNTES Y CASSETAS	FCLBG11	FCLIU13	UNICO	c_ca_pue
TARIFAS ESPECIALES	FCLBV08	FCLIC26	UNICO	c_origen,c_destino,c_ruta, c_prod,f_initar(desc),f_fintar
USUARIOS	FCLBG13	FCLIC30	UNICO	c_user,c_pass
		FCLIC33	UNICO	c_user,n_ident

**Petroleos Mexicanos, C.A.S.E
object summary report
(Reporte Sumario de Objetos tipo Relación Generado por IEW)**

Relation: acumulado mensual emb.pag.

Definition

fclbs12 Acumulado mensual de embarques pagados

Comments

Es la tabla que contiene información a nivel de estadísticos mensuales de los embarques pagados, por producto, grupo de producto y porteador.

Relation: Autorizaciones

Definition

Pagos autorizados

Comments

Contiene el detalle de los importes extras autorizados mediante la forma gdt-3 para un embarque de carga regular específico.

Relation: Autotanques

Definition

fclbg05 catálogo de autotanques

Comments

Esta tabla contiene los datos y las características particulares de los autotanques pertenecientes a las compañías transportistas autorizadas por Pemex para la distribución de carga líquida.

Relation: Bitácora de transacciones

Definition

fclbs06 bitácora de transacciones

Comments

Es el archivo que contiene el registro de todos los movimientos tales como: altas, bajas y actualizaciones realizados por los usuarios, a las tablas del sistema que están auditadas

Relation: Bonificaciones/Descuentos

Definition

fclbv01 archivo de bonificaciones/descuentos

Comments

Es el archivo que contiene el registro de los movimientos de bonificaciones y/o descuentos, que se aplican en cada proceso de pago por concepto de flete a las Cias. transportistas

Relation: Calibraciones situaciones

Definition

fclbv02 Calibraciones y situaciones

Comments

Es el archivo que contiene la historia de las diferentes situaciones en que han estado los autotanques, así como la información de los certificados de calibración que determinan la capacidad de los mismos y su vigencia.

Relation: Características rutas

Definition

fclbv03 archivo de características rutas

Comments

Es el archivo que contiene la información referente a las características propias y kilómetros existentes de las rutas, dado un origen y un destino, para la realización de un flete, al igual que otros archivos, este contiene la historia de las rutas a través de su vigencia.

Relation: Casetas por ruta

Definition

fclbn01 catálogo de casetas por ruta

Comments

Contiene la relación de claves de casetas y puentes que existen en las rutas definidas y registradas por Pemex.

Relation: Cédula de pago

Definition

fclbs08 archivo, cédula de pago

Comments

Contiene los importes por concepto de flete, faltante, iva, pece, bonificaciones y descuentos, que determinan el total del pago a cada uno de los transportistas, incluidos en una cédula.

Relation: Conceptos de autorización

Definition

conceptos de autorizaciones

Comments

Contiene el catálogo de los diferentes conceptos de autorizaciones que un embarque puede tener .

Relation: Conceptos de bon/desc

Definition

fclbg06 catálogo de conceptos de bonificaciones/descuentos

Comments

Contiene la definición de los diferentes conceptos por bonificación o descuento, a que puede hacerse acreedor o deudor un transportista.

Relation: Control de batch's

Definition

fclbv12 archivo de control batch's

Comments

Contiene la información relacionada con el control de batch's que han sido capturados, así como la identificación del capturista, tiempo de captura, si fueron cerrados, verificados, enviados, etc.

Relation: Costos

Definition

fclbv04 archivo de costos

Comments

Archivo que contiene las vigencias y costos de las casetas y puentes ubicados en las diferentes rutas registradas en Pemex.

Relation: Cuentas por pagar

Definition

bdfletes.cuentas por pagar

Comments

Es el archivo que contiene la información contable, referente a la liquidaciones pendientes de pago, a las compañías transportistas.

Relation: Cuotas Pemex

Definition

fclbg01 tarifa de cuotas Pemex

Comments

Contiene las cuotas por concepto de flete para cada 10 kilometros recorridos, asignadas por la s.c.t para las diferentes clases de productos transportados por Pemex.

Relation: Embarques C:L

Definition

fclbs04 archivo de embarques carga líquida

Comments

Es el archivo que contiene la información de los embarques realizados por los transportistas, que han pasado el proceso de captura, calculos y validación y se encuentran en trámite de pago.

Relation: Embarques no pagados

Definition

fclbs02 archivo de embarques no pagados

Comments

Es el archivo que contiene la información relativa a los embarques de carga líquida realizados por los transportistas, y que aun se encuentran pendientes de pago.

Nota:

la implementación de este archivo esta pendiente, hasta que se tenga la interfase con el sistema sat.

Relation: Embarques pagados recortado

Definition

fclbs11 archivo de embarques pagados recortado

Comments

Archivo con embarques pagados a la fecha y cuatro meses atras.

Relation: Embarques Reportados

Definition

fclbv05 archivo de embarques reportados

Comments

Es el archivo que contiene la información de los embarques que por algun motivo son reportados, con la indicación de que no se realice el pago correspondiente.

Relation: Grupos de productos

Definition

fclbg04 catálogo de grupos de productos

Comments

Contiene el catálogo de las diferentes agrupaciones de los productos de carga líquida.

Relation: Grupos Empresariales

Definition

fclbg07 catálogo de grupos empresariales

Comments

Contiene el catálogo de los grupos empresariales a que pertenecen las Cías. transportistas que prestan servicio de transportacion para los diferentes productos de Pemex.

Relation: Hist. Emb. C.L. Anterior

Definition

fclbs15 archivo, historico de embarques pagados carga líquida anterior

Comments

Es un archivo de paso para bajar información historica de embarques pagados de años anteriores, y poder generar estadísticas.

Relation: Hist.Emb. C.L. Seleccionado

Definition

fclbs14 archivo, histórico de embarques pagados carga líquida

Comments

Es un archivo de paso temporal para generar reportes del archivo histórico de embarques, seleccionados previamente, es decir del año actual o anterior.

Relation: Histórico de Transacciones

Definition

fclbs07 archivo, histórico de transacciones

Comments

Contiene el reflejo de como estaban antes y/o despues los registros de las tablas del sistema que fueron modificados y registrados en la bitácora de transacciones. Este archivo esta totalmente ligado con el de bitácora de transacciones. se utiliza en la auditoria del sistema.

Relation: Histórico Embarques Pag. C.L.

Definition

fclbs01 archivo de embarques pagados carga líquida

Comments

Es el archivo que contiene toda la información relacionada con los embarques de carga líquida realizados por los transportistas, y que ya han sido liquidados.

Relation: Invent. de Tablas

Definition

fclbg15 archivo de inventario de tablas del sistema

Comments

Es la tabla que contiene la clave de identificación de todos los archivos que existen en el sistema.

Relation: Inventario de Prog's

Definition

fclbg12 archivo de inventario de programas

Comments

Es el archivo que contiene la información relacionada con los programas que conforman todo el sistema, así como las indicaciones para su operación y explotación.

Relation: Liquidaciones

Definition

fclbs05 archivo de liquidaciones

Comments

Es el archivo que contiene la sumaria de los importes pertenecientes a los embarques contenidos por liquidación, así como el número de cédula en la que fueron procesados .

Relation: Marcas de Tractores y Tanques

Definition

fclbg08 catálogo de marcas de tractores y tanques

Comments

Es el catálogo que contiene los nombres de las diferentes marcas de tanques y tractores que utiliza Pemex para el transporte de sus productos.

Relation: Mov. Rechazados por Calculo

Definition

fclbv11 archivo de movimientos rechazados por calculo

Comments

Archivo que contiene la información relativa a los embarques que fueron rechazados, por encontrar errores e incongruencias durante el proceso de validación:

Relation: Movimientos Diarios

Definition

fclbv09 archivo de movimientos diarios

Comments

Es el archivo que contiene la información de los embarques realizados por los transportistas, embarques que ya han sido capturados, y que esperan pasar por el proceso de pago.

Relation: Origenes/Destinos

Definition

fclbg09 catálogo de orígenes/destinos

Comments

Es el catálogo que contiene la clave y el nombre de los diferentes sitios olugares en donde existen instalaciones de Pemex, considerados como orígenes o destinos, de donde se envían o reciben productos y/o materiales.

Relation: Pagos Aplicados

Definition

fclbs03 archivo de pagos aplicados

Comments

Este archivo que contiene la información de los pagos parciales aplicados a transportistas en las cédulas, por concepto de descuento.

Relation: Parametros contables

Definition

fclbs10 archivo de parametros contables

Comments

Contiene los diferentes parametros contables y presupuestales a nivel de tipo de producto y agencia pagadora, que se utilizan para realizar la afectación contable en las cédulas procesadas que se envían a contabilidad.

Relation: Parametros del Sistema

Definition

fclbs09 archivo de parametros del sistema

Comments

Contiene los parametros generales del sistema.

Relation: Peaje Total

Definition

fclbv06 archivo de peaje total

Comments

Contiene los costos totales de peaje para cada una de las rutas autorizadas por Pemex, así como el histórico y su vigencia.

Relation: Porcentajes Autorizados

Definition

porcentajes y/o cuotas de autorizaciones

Comments

Contiene los porcentajes autorizados a pagar por los diferentes conceptos extras que un embarque puede tener así, como su vigencia.

Relation: Porteadores

Definition

fclbg10 catálogo de porteadores

Comments

Archivo que contiene los datos generales de los expedientes de las compañías de transporte que prestan sus servicios a Pemex.

Relation: Precios

Definition

fclbv07 catálogo de precios

Comments

Contiene el catálogo (histórico) de precios para los productos comercializados por Pemex, así como su vigencia.

Relation: Presupuesto de Fletes

Definition

fclbs13 archivo de presupuesto de fletes

Comments

Contiene el presupuesto mensual autorizado y ejercido por concepto de flete y grupo de producto.

Relation: Productos C. Liq.

Definition

fclbg02 catálogo de productos carga líquida

Comments

Catálogo que contiene la información relativa a los productos que se consideran como carga líquida, que son comercializados por Pemex.

Relation: Productos por Ruta

Definition

fclbn02 archivo de productos por ruta

Comments

Esta tabla contiene la información desagregada de la transportacion a nivel producto por ruta.

Relation: Puentes y Casetas

Definition

fclbg11 catálogo de puentes y casetas

Comments

Catálogo de las casetas y puentes registrados por Pemex.

Relation: Tarifas Especiales

Definition

fclbv08 catálogo de tarifas

Comments

Es el archivo que contiene el catálogo de tarifas aplicables a la ransportación de materiales de importación.

Relation: usuarios

Definition

fclbv08 catálogo de usuarios

Comments

es el archivo que contiene el catálogo de los usuarios que tendran acceso al sistema mediante el mecanismo de seguridad implementados.

C.A.S.E. Petróleos Mexicanos

RELACION DE MODULOS DEL SISTEMA DE CARGA LIQUIDA

MENUS DEL SISTEMA

ADMINISTRACION TECNICA DEL SISTEMA	(FCLFM08)
ESTADISTICAS	(FCLFM08)
EXPLOTACION DEL BANCO DE DATOS	(FCLFM08)
INTERFASES	(FCLFM08)
MANTENIMIENTO A CATÁLOGOS	(FCLFM08)
PAGOS	(FCLFM08)
REGISTRO DE MOVIMIENTOS	(FCLFM08)
REPORTES DE PRODUCCION	(FCLFM08)

MODULOS DE MANTENIMIENTO A ARCHIVOS

ACTUALIZA FECHA REC./DEP. CED.	(FCLFA39)
ACUMULA EN HISTORICO	(FCLFA40)
ACUMULA EN MENSUAL	(FCLFA41)
AUTOTANQUES	(FCLFA03)
BONIFICACIONES/DESCUENTOS	(FCLFA18)
CALIBRACIONES	(FCLFA15)
CAPTURA BATCH DE EMBARQUES	(FCLFK01)
CARACTERISTICAS RUTAS	(FCLFA12)
CASSETAS POR RUTA	(FCLFA14)
CONCEPTOS BONIF./DESC.	(FCLFA07)
CORRECCION A NUMEROS DE EMB.	(FCLFA35)
COSTOS CASSETAS/PUENTES	(FCLFA16)
CUOTAS	(FCLFA09)
EMBARQUES REALIZADOS	
EMBARQUES REPORTADOS	(FCLFA19)
GRUPOS DE PRODUCTOS	(FCLFA11)
GRUPOS EMPRESARIALES	(FCLFA02)
INVENTARIO DE PROGRAMAS	(FCLFA20)
MANTENIMIENTO AL ARCHIVO DE PARAMETROS DEL SISTEMA	(FCLFA32)
MANTENIMIENTO DE USUARIOS	(FCLFA21)
MARCAS DE TRACTORES Y TANQUES	(FCLFA26)
ORIGEN DESTINOS	(FCLFA05)
PARAMETROS CONTABLES Y PRESUPUESTO	(FCLFA37)
PEAJE TOTAL	(FCLFA13)
PERFILES DE USUARIOS	(FCLFA22)
PORTEADORES	(FCLFA01)
PRECIOS	(FCLFA17)
PRESUPUESTO MENSUAL AUTORIZADO	(FCLFA38)

Capítulo III**Diseño y desarrollo del sistema**

PRODUCTOS CARGA LIQUIDA	(FCLFA08)
PRODUCTOS POR RUTA	(FCLFA33)
PUENTES Y CASSETAS	(FCLFA06)
TARIFAS ESPECIALES	(FCLFA10)
VALIDACION Y CALCULO DE EMBARQUES	(FCLFA31)
VERIFICACION DE BATCH'S	(FCLFK03)

FUNCIONES Y PROCESOS ESPECIALES

CALCULO DE TARIFAS NORMALES	(FCLLA01)
CONSULTA DINAMICA DE TABLAS DEL SIST.	(FCLLA03)
CORRECCION A LA CAPTURA	(FCLFK04)
EJECUTA LA GENERACION DE MECANIZADOS	(FCLSH02)
FUNCION DE REGISTRO DE TRANSACCIONES	(FCLLA02)
GENERACION DE ASCII PARA CONTABILIDAD	(FCLFG02)
GENERACION DE ASCII PARA FINANZAS	(FCLFG03)
GENERACION DE BATCH PARA IBM	(FCLFG01)
GENERACION DE CEDULAS Y LIQUIDACIONES	(FCLFA27)
GENERACION DE PEAJE TOTAL	(FCLFA34)
LANZA CEDULA DE PAGO	(FCLFA24)
LANZA PROCESO DE VALIDACION Y CALCULO	(FCLFA30)
LANZADOR DE MECANIZADOS	(FCLFM01)
MONITOREO DE MECANIZADOS	(FCLSH03)
MONITOREO DE PROCESOS	(FCLFM02)
MONITOREO DE PROCESOS ACTIVOS	(FCLSH04)
PAGINA IMPRESION DE MANUAL	(FCLSH08)
PRODUCTOS POR RUTAS	(FCLFA33)
RESTAURA CEDULAS	(FCLFA50)
SUSTITUYE (#) POR BLANCOS EN ARCHIVO	
GENERADO PARA CONTABILIDAD	(FCLSH06)
VERIFICA SI EXISTE UN ARCHIVO	(FCLSH01)
VERIFICACION DE PROCESOS ACTIVOS	(FCLSH05)

RESPALDOS DE INFORMACION

RESPALDO CEDULAS Y LIQUIDACIONES	(FCLFG07)
RESPALDO DE BONIF. Y PAG. APLICADOS	(FCLFG09)
RESPALDO DE MOVIMIENTOS DIARIOS	(FCLFG03)
RESPALDO EMBARQUES PAGADOS	(FCLFG08)
RESPALDO EN DISKETTE DE ARCHIVOS PARA CONTABILIDAD Y FINANZAS	(FCLFG05)

MODULOS DE CONSULTAS

CONSULTA DE BATCH'S CAPTURADOS	(FCLFC63)
CONSULTA DE CEDULAS DE PAGO	(FCLFC66)
CONSULTA DE PRESUP. AUTORIZADO MENS.	(FCLFC64)
CONSULTA DE HISTORICO DE EMB	(FCLFC23)
CONSULTA DE LIQUIDACIONES	(FCLFC62)
CONSULTA DE MOVIMIENTOS DIARIOS	(FCLFC61)
CONSULTA DE TARIFAS	(FCLFC60)

REPORTES

REPORTE ANALITICO DE PEAJE	(FCLFR57)
REPORTE ANALITICO POR LIQUIDACION	(FCLFR52)
REPORTE DE BATCH'S CAPTURADOS	(FCLFR80)
REPORTE DE BONIF./DESC.	(FCLFR56)
REPORTE DE CEDULA DE PAGO	(FCLFR53)->LIGADO EN EL
(FCLFR56)	
REPORTE DE CONTRA-RECIBO	(FCLFR55)
REPORTE DE CUENTAS POR PAGAR	(FCLFR54)
REPORTE DE VALIDACION	(FCLFR51)

DOCUMENTACION DE PROCESOS

REGLAS DE VALIDACION DE TRAFICOS

- Valida Reportados

Que el número del embarque no esté en el archivo de Embarques Pagados, si no ent. error 35.

- Validación del Importe

Que el importe sea numérico y mayor que cero, si no ent. error 20.

- Valida Duplicados

Que no exista el número del embarque en el Histórico de Liquidaciones de acuerdo a la Fecha de Embarque y Número de Embarque, si no ent. error 25.

- Validación de la Fecha de Liquidación

Que el número de liquidación sea numérico y mayor que cero.

Que la fecha de liquidación tenga el formato (AAMMDD), si no ent. error 1.

Y que Fecha de Liquidación \leq Fecha Máquina.

- Validación de la Fecha de Embarque

Que la fecha de Embarque tenga el formato (AAMMDD), si no ent. error 2.

Que Fecha del Embarque $<$ Fecha-Máquina, si no ent. error 34 y que Fecha del Embarque $<$ Fecha de Liquidación.

- Validación de la clave del porteador

Que la clave del porteador exista en el Catálogo de Porteadores, si no ent. error 3.

Que el porteador esté activo al momento del embarque si no ent. error 4.

- Validación de la Clave del Autotanque

Checar que exista el autotanque en el Catálogo de Autotanques por medio de alguna de las siguientes claves: Porteador-Autotanque o Porteador-Autotanque-Anterior, si no existe el autotanque en el Catálogo ent. error 6.

Checar en el catálogo de autotanques que (Fecha del Embarque $<$ Fecha-Situación y que Tipo-Situación = 1 si no ent. error 7)

Checar en el Histórico de Situaciones que (Fecha Inicial \leq Fecha del Embarque $<$ = Fecha Final y que Tipo Situación = 1 si no ent. error 9.

- **Validación del tipo de tanque**

Cantidad de tanques:

menos de 1 => error 10
o mas de 2

Fecha <= Fecha
Calibración Embarque

calbr = capacidad
donde,
calbr = calibración

unidad = Kgs
medida

1 ó 2

Cantidad <= calbr/0.90
Cargada

Fecha =.0 => error 41
Calibración

- **Validación del Producto**

Que la clave del producto sea numérica y mayor que cero.

Que el producto exista en el CATÁLOGO DE PRODUCTOS, si no entonces error 11.

Que en el CATÁLOGO DE PRODUCTOS la unidad de medida sea 010 o 030, si no entonces error 28.

- **Validación del Destino**

Que exista la clave del destino en el CATÁLOGO ORIGEN-DESTINO, si no entonces error 13 y que la clave del destino sea mayor que cero.

- **Validación del Destino de Reexpedición**

Si la clave de destino de reexpedición es mayor que cero, entonces checar que exista en el CATÁLOGO, si no existe en el CATÁLOGO, entonces error 14.

Que no traiga error la Fecha del Embarque, si no ent. error 44.

- Validación del Origen

Que clave-origen sea numérica y que exista en el catálogo, si no entonces error 15.

- Validación de Origen-Destino-Ruta

Checar que la fecha de embarque cumpla con Fecha Inicial < = Fecha Embarque < = Fecha-Final, si no entonces error 43.

Checar que exista la relación Origen-Destino en el Catálogo Origen-Destino-Ruta, si no ent. error 16.

Checar que exista la relación Destino-Destino de reexpedición en el Catálogo, si no ent. error 17.

Si (la Fecha de Embarque no tiene error y clave-origen = 00051
y clave-producto = 0120 y (clave-destino = 02051 or 01035)
y Fecha-Embarque = 880317 THRU 999999)
entonces ruta-flete = 02

Si (la Fecha de Embarque no tiene error y clave-origen = 00051
y clave-producto = 0120 y (clave-destino = 14029 or 00029)
or clave-destino = 04029 or clave-destino = 03029
y Fecha-Embarque = 890219 THRU 999999)
entonces ruta-flete = 02

- Validación de la Cantidad Embarcada

Si (no hay error en la unidad de medida) entonces

Unidad de medida:

kgs => cant. aux. = $\frac{\text{cantidad embarcada}}{\text{PESO ESPECIFICO}} * 0.9$

lts => cant. aux. = cantidad embarcada

Si cant. aux. > calibración entonces error 18

PROCEDIMIENTO DE CALCULO**1. Normalización de volúmenes a 20°C**

Unidad-medida:

Lts => vol. carga normalizado = $(1+(20-TC)*cf)$ * cantidad embarcada
vol. descarga normalizado = $(1+(20-TC)*cf)$ * cantidad descargada

donde:

TC = temperatura a la carga o descarga
cf. = coeficiente de dilatación

Kgs => vol. carga normalizado = cantidad embarcada
vol. descarganormalizado=cantidad descargada

2. Cálculo del faltante

Si Cantidad Descargada = Cantidad Cargada => faltante = 0
independientemente de la temperatura de carga y de descarga

Unidad-medida:

Lts => faltante = vol. carga normalizado - vol. descarga normalizado

Kgs => faltante = 0

3. Cálculo de la Tarifa Total

Si hay destino de reexpedición

tarifa total = tarifa (origen - destino de reexp.)
+ tarifa (origen - destino) * percent. bonificación
+ tarifa (destino - dest. reexp.) * percent. bonificación

Si el producto es GAS

tarifa total = tarifa total
+ tarifa (origen - destino) * percent. gas
..+ tarifa (destino - dest. reexp.) * percent. gas

Fin si

En otro caso

Tarifa Total = Tarifa (origen - destino)
+ tarifa(origen-destino) * percent. bonif.

Si el producto es GAS

Tarifa Total = Tarifa Total
+ tarifa(origen-destino) * percent. GAS

Fin si

Fin si

Note: Las tarifas se calculan de acuerdo al procedimiento de CALCULO DE TARIFA y los porcentajes de bonificación se obtienen del catálogo de rutas.

4. Cálculo del Importe del Flete

Importe del flete = cantidad cargada * tarifa total

Si descuento total > importe del flete ent

****Error****

Fin si

5. Cálculo del Importe a Descontar

En caso de que haya faltante de producto a la entrega, se aplicarán dos descuentos:

a) Importe a Descontar = Faltante * Precio del producto

b) Descto proporcional de flete = faltante * TARIFA TOTAL

DesctoTotal= Importe a Descontar + Descto proporcional de flete

6. Peaje Total

Peaje Total = suma de diferencia de casetas
+ suma Imp. puentes
+ suma Imp. transbordador

El Peaje Total se obtiene del catálogo de peajes.

7. Cálculo del importe del Flete Total

Imp. flete total = importe del flete + importes de bonificación

Si $20 < \text{importe flete 3-218}$
 - (Imp. Flete Total + Importes de Transbordador) < -10 ent.

****Error**** El importe calculado **no** coincide con el capturado
 Fin si

Si no hay error en importe calculado y faltante = 0 ent.
 Imp. Flete Total ← Importe Flete 3-218

En otro caso

Imp. Flete Total = Imp. Flete Total - Importe a Descontar
 + Importes de Transbordador

Fin si

8. Cálculo de la Tarifa

El cálculo de la tarifa esta en función de la DISTANCIA y la UNIDAD de MEDIDA
 Si **hay** tarifa especial entonces

TARIFA = IMPORTE-TARIFA-ESPECIAL

El Importe Tarifa Especial se obtiene del
 Catálogo de Tarifas-Transbordadores-Impuestos (fcbv08)

Si **no** hay tarifa especial entonces

1) Se determina la distancia recorrida

distancia:

< 20 => distancia = 20
 > 999 => distancia = 1000

2) Se calcula la cuota correspondiente a la distancia recorrida

distancia:

múltiplo de 10 => CUOTA=TARIFA-SCT, se obtiene de.CATÁLOGO

no múltiplo de 10 dist. aux = distancia menos unidades de distancia
 CUOTA 1 =TARIFA-SCT corresp. a dist. aux.
 CUOTA 2 =TARIFA-SCT corresp. a (dist. aux. +10)

$$CUOTA = CUOTA 1 + \frac{(distancia - dist. aux) * (CUOTA2 - CUOTA1)}{10}$$

3) Si la capacidad promedio no es cero entonces se calcula la tarifa correspondiente a la distancia y unidad de medida.

unidad de medida:

kgs => TARIFA = .. $\frac{CUOTA * DISTANCIA}{Capacidad promedio * Peso Esp.}$

Nota: Peso Esp. = 0 => Peso Esp. = 1

Lts => TARIFA =..... $\frac{CUOTA * DISTANCIA}{Capacidad promedio}$

DICCIONARIO DE DATOS

Petroleos Mexicanos C.A.S.E

Object List

(Lista de Objetos tipo Dato generada por IEW)

ABONO FLETES
ABONO IVA
AGENCIA PAGADORA
CANTIDAD CARGADA
CANTIDAD DESCARGADA
CANTIDAD EMBARCADA
CAPACIDAD PROMEDIO
CARGO FLETES
CARGO FLETES
CARGO IVA
CEDULA DE PAGO
CENTRO DE TRABAJO ABONO
CENTRO DE TRABAJO ABONO IVA
CENTRO DE TRABAJO CARGO
CENTRO DE TRABAJO CARGO IVA
CENTRO PAGADOR
CENTRO TRABAJO CARGO
CLASES DE PRODUCTOS
CLAVE AUTOTANQUE
CLAVE CASETA/PUENTE
CLAVE DE AUTORIZACION
CLAVE DE PROYECTO
CLAVE DE STATUS
CLAVE DEL EVENTO
CLAVE DEL MODULO
CLAVE DEL PATH
CLAVE DEL REGISTRO
CLAVE PASSWORD
CLAVE PORTEADOR
CLAVE PRODUCTO
CLAVE REGISTRO DE PARAMETROS
CLAVE STATUS DEL BATCH
CLAVE STATUS REG
CLAVE TIPO BATCH
CLAVE TIPO DE CEDULA
CLAVE USUARIO
CLAVE USUARIO QUE AUTORIZA
CLAVE USUARIO QUE ELABORA
CLAVE USUARIO QUE REvisa

Petroleos Mexicanos C.A.S.E

Object List

(Lista de Objetos tipo Estructura de Datos generada por IEW)

CLAVE USUARIO VERIFICACION
CONCEPTO ORIGEN
CONCEPTO ORIGEN IVA
CTA. PRESUESTAL IVA
CUENTA ABONO
CUENTA ABONO IVA
CUENTA CARGO
CUENTA CARGO IVA
CUENTA PRESUPUESTAL ABONO
CUENTA PRESUPUESTAL CARGO
CUENTA PRESUPUESTAL IVA
DEDUCCION POR FALTANTE
DEPARTAMENTO ABONO
DEPARTAMENTO ABONO IVA
DEPARTAMENTO CARGO
DEPARTAMENTO CARGO IVA
DEPARTAMENTO DE CARGO
DESCRIPCION DEL EVENTO
DESCRIPCION DEL REGISTRO
DESTINO
DESTINO REEXPEDICION
DIGITO AVISO
DIGITO AVISO IVA
DIGITOS NUM. EMBARQUE
FECHA DE CAPTURA
FECHA DE DEPOSITO BANCARIO
FECHA DE ELABORACION
FECHA DE RECEPCION
FECHA DEL EVENTO
FECHA EMBARQUE
FECHA FIN VIGENCIA
FECHA FINAL VIGENCIA
FECHA INGRESO USUARIO
FECHA INIC. VIGENCIA
FECHA INICIO VIGENCIA
FECHA LIQUIDACION
FECHA MOVIMIENTO
FECHA PROCESO
FECHA PROGRAMADO PAGO
FECHA STATUS REG

Petroleos Mexicanos C.A.S.E**Object List****(Lista de Objetos tipo Data generados por IEW)**

FLETE TOTAL
FOLIO FINANZAS
GRUPO DE PRODUCTO
HOJA ADEFAS
HOJA DE DOCUMENTO
HOJA INICIAL ADEFAS
HOJA INICIAL DE DOCUMENTO
HOJA SIG. DOCUMENTO
HORA DEL EVENTO
I.V.A
IDENTIF. USUARIO VERIFICADOR
IDENTIFICA CEDULA
IDENTIFICADOR
IDENTIFICADOR DE LA TABLA
IDENTIFICADOR DE QUIEN AUTORIZA
IDENTIFICADOR DE QUIEN ELABORA
IDENTIFICADOR DE QUIEN REvisa
IDENTIFICADOR DEL USUARIO
IMPORTE DEDUCCIONES
IMPORTE FLETE
IMPORTE IVA
IMPORTE NOTA DE EMBARQUE
IMPORTE TOT. POR BONIFICACIONES
IMPORTE TOT. POR DESCUENTOS
KILOMETROS
LETRAS NUM. EMBARQUE
IMPORTE PECE
NOMBRE DEL USUARIO
NUM. DE ADEFAS
NUM. DE DOCUMENTO
NUM. DOC. INICIAL
NUM. EMBAR. HISTORICO
NUM. INIC. PAGO 2a PARTE
NUM. INICIAL ADEFAS
NUM. PAGO 1a. PARTE
NUM. PAGO 2a. PARTE
NUMERO CONTRARRECIBO
NUMERO DE BATCH
NUMERO DE CEDULA
NUMERO DE DOCUMENTO
NUMERO DE EMBARQUES

Petroleos Mexicanos C.A.S.E

Object List

(Lista de Objetos tipo Data generados por IEW)

NUMERO DE LIQUIDACIONES
NUMERO DE REGISTRO
NUMERO DE TERMINAL
NUMERO EMBARQUE
NUMERO LIQUIDACION
NUMERO MECANIZADO
NUMERO SECUENCIAL DEL EVENTO
ORIGEN
P.E.C.E
PEAJE TOTAL
PESO ESPECIFICO
RUTA
RUTA DE REEXPEDICION
SECUENCIAL DE CONTRA-RECIBOS
STATUS
STATUS FINANZAS
SUBCUENTA ABONO
SUBCUENTA ABONO IVA
SUBCUENTA CARGO
SUBCUENTA CARGO IVA
TARIFA
TARIFA REEXPEDICION
TEMPERATURA DE CARGA
TEMPERATURA DE DESCARGA
TIEMPO DE CAPTURA
TIPO DE BATCH
TIPO DE CEDULA
TIPO DE DOCUMENTO
TIPO DE EVENTO
TIPO DE REGISTRO
UNIDAD DEL PRODUCTO
VALORES INICIALES MENSUALES
VALORES SIGUIENTE PROCESO

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Agencia pagadora

Comments

Se refiere a la agencia en donde se efectuó el pago, es un campo numérico de 5 posiciones, acepta valores del 1 al 99999.

Definition

Cantidad cargada

Comments

Se refiere al volumen total de producto cargado, es un campo numérico de 5 posiciones.

Definition

Cantidad descargada

Comments

Se refiere al volumen del producto una vez que ha sido descargado, es un campo numérico de 5 posiciones.

Definition

Capacidad promedio

Comments

Se refiere a la capacidad promedio registrada en la tarifa elaborada por la Secretaría de Comunicaciones y Transportes y que indica la columna de acuerdo al factor f, y tipo de cuota, es un campo numérico de 8 dígitos.

Definition

Cedula de pago

Comments

Contiene toda la información relacionada con elaboración del proceso de pago ya calculado y autorizado en determinada fecha, este proceso se genera una vez a la semana.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo de Datos Generada por IEW)

Definition

Centro pagador

Comments

Este campo indica el centro pagador en el que se realizara el pago de las liquidaciones autorizadas a cobrar, es un campo numérico con valores predefinidos del 1 al 99999.

Definition

Clave autotanque

Comments

Se refiere a la clave asignada por Pemex para identificar al autotanque y que forma parte del catálogo, es un campo alfanumérico de 6 posiciones.

Definition

Clave caseta / puente

Comments

Se refiere a la clave que identifica un puente o una caseta de cobro y que se encuentran dentro de las rutas autorizadas, es un campo alfanumérico de 3 posiciones.

Definition

Clave de autorización

Comments

Se refiere a la clave que identifica el motivo de autorización.

Definition

Clave del modulo

Comments

Se refiere a la clave del módulo relacionado con la vista asigna al usuario para acceder dentro del sistema, es un campo numérico.

Definition

Clave del path

Comments

Se refiere a la clave que direcciona al usuario y al nivel que tendra permitido acceder dentro del sistema, es un campo caracter.

Petroleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Clave de status

Comments

Bandera que nos permite ver cual es el estado de un registro de captura:

1 = registro capturado

2 = registro validado

3 = registro enviado

Definition

Clave del registro

Comments

Se refiere al estatus de los registros pertenecientes a los catálogos que contienen datos con vigencia y los valores son:

1= registro activo

2= registro inactivo (o dado de baja)

Definition

Clave password

Comments

Se refiere a la clave propia de cada usuario que le permite la entrada al sistema, es un campo alfanumérico de 5 posiciones.

Definition

Clave del porteador

Comments

Se refiere a la clave asignada por Pemex, para identificar al transportista, y esta contenida en el catálogo de porteadores, es un campo numérico con valores permitidos del 1 al 32767.

Definition

Clave del producto

Comments

Se refiere a la clave asignada por Pemex y que identifica el producto, misma que debe existir en el catálogo de productos, es un campo numérico de 4 posiciones.

Petroleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Clave de registro de parametros

Comments

Se refiere a la clave única de acceso a este archivo

Definition

Clave status del batch

Comments

Se refiere a la clave que identifica en que estado de tramite se encuentra un batch de liquidaciones en captura , es un campo de un solo caracter, y admite valores del 1 al 9 , y de la "a" a la "z".

Definition

Clave del status del registro

Comments

Se refiere a la clave que puede ser activo o inactivo el registro del producto, es un campo que unicamente acepta los siguientes valores:

0 que significa inactivo

1 que significa activo

Definition

Clave tipo batch

Comments

Este campo graba el tipo de batch que se esta capturando, distingue entre local o foraneo y los valores permitidos son:

local = l

foraneo = f

Definition

Clave usuario

Comments

Se refiere a la clave asignada al usuario para tener acceso al sistema, es un campo alfanumerico de 7 posiciones

Petroleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Clave tipo de cédula

Comments

Se refiere a la clave con que se identifica a la cédula ya sea local o foránea, es un campo alfanúmerico de 1 caracter que acepta valores de la "a" a la "h" , con la siguiente separacion:

- a.- gas
- b.- petroquimicos
- c.- turbosina
- d.- refinados u otros
- e.- gas
- f.- petroquimicos
- h.- refinados u otros

Definition

Clave del usuario que autoriza

Comments

Se refiere a la clave que identifica al usuario que autoriza una cédula de pago.

Definition

Clave del usuario que revisa

Comments

Se refiere a la clave que identifica al usuario que revisa la cédula de pago una vez que fué elaborada.

Definition

Clave usuario verificacion

Comments

Se refiere a la clave del usuario que realizo la verificacion de la captura del batch, es un campo alfanúmerico de 7 posiciones.

Definition

Datos contables

Comments

Es el grupo de campos que contienen la información de interes para los registros contables del area.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Datos de pago

Comments

Se refiere a los datos referentes al pago efectuado en una cédula de pago autorizada.

Definition

Datos financieros

Comments

Es el grupo de campos que tienen la información de interés para el área de finanzas de Pemex.

Definition

Datos de flujo de efectivo

Comments

Es el grupo de campos que informan del movimiento del efectivo asignado al área para el pago de movimientos realizados durante un mes.

Definition

Datos generales del embarque

Comments

Se refiere al grupo de datos que identifican el embarque, dicho grupo esta formado por:

- 1.- clave del porteador
- 2.- numero de embarque
- 3.- fecha de embarque

Definition

Datos de la liquidación

Comments

Se refiere al grupo de datos que identifican la liquidación en la que se encuentra un embarque determinado.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Datos del proceso

Comments

Se refiere al grupo de datos que identifican el proceso, este grupo esta formado por:

- 1.- fecha de proceso
- 2.- número de cédula
- 3.- clave del tipo de cédula
- 4.- fecha programada del pago

Definition

Deducción por faltante

Comments

Se refiere a la cantidad a deducir al porteador por existir faltante del producto entregado, es un campo numérico de 10 posiciones.

Definition

Descripcion del evento

Comments

se refiere a la descripción de el evento motivo por el cual se da de alta el registro.

Definition

Descripcion del evento

Comments

Se refiere a la descripción de la clave que identifica el tipo de evento.

Definition

Destino

Comments

Se refiere al lugar en el cual debe ser entregado el producto que origino el flete, es un campo numérico de 5 dígitos, y sus valores permitidos son del 1 al 99999 utilizados para identificar la clave del destino de la ruta, de acuerdo con el catálogo de orígenes y/o destinos.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Destino de reexpedición

Comments

Se refiere al lugar considerado como nuevo destino, una vez que por alguna razón el producto no puede ser recibido en el destino original, es un campo numérico de 5 posiciones, admite valores del 1 al 99999, de acuerdo al catálogo de orígenes/destino.

Definition

Digitos del número de embarque

Comments

Se refiere únicamente a los dígitos que forman parte del número que identifica un embarque, es un campo numérico de 6 posiciones.

Definition

Fecha de captura

Comments

Se refiere a la fecha en que se capturó el batch, se presenta en formato ddmmaa en donde, dd=día, mm=mes, aa=año, y solamente se usan dos dígitos para cada dato.

Definition

Fecha de depósito bancario

Comments

Se refiere a la fecha en que fue hecho el depósito a favor del porteador.

Definition

Fecha de elaboración

Comments

Se refiere a la fecha en que se elaboró la cédula de pago, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos dígitos para cada dato.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Fecha de recepción

Comments

Se refiere a la fecha en que se recibo documentacion del porteador para la tramitación de su pago.

Definition

Fecha del evento

Comments

Se refiere a la fecha que registra en el sistema al porteador cuando se accesa esta tabla, se presenta en formato "ddmmaa" en donde , dd=día, mm=mes, aa=año.

Definition

Fecha del embarque

Comments

Se refiere a la fecha en la cual fué emitido el talon de embarque, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se usan dos digitos para cada dato.

Definition

fecha de fin de vigencia

Comments

Se refiere a la fecha en la cual termina la autorizacion de la agencia pagadora habilitada como tal.

Definition

Fecha final de la vigencia

Comments

Se refiere a la fecha en la que termina la vigencia que autoriza la agencia pagadora, se presenta con formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos digitos para cada dato.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Fecha de ingreso del usuario

Comments

Se refiere a la fecha en que se registro la autorizacion a un usuario, para tener acceso al sistema, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año.

Definition

Fecha inicial de vigencia

Comments

Se refiere a la fecha en que inicia la autorización que habilita a la agencia pagadora.

Definition

Fecha inicio vigencia

Comments

Se refiere a la fecha en que inicia la vigencia, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos digitos para cada dato.

Definition

Fecha de liquidación

Comments

Se refiere a la fecha de emision de la liquidación (factura del porteador) , se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año.

Definition

Fecha de movimiento

Comments

Se refiere a la fecha en que se realizo el movimiento, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos digitos para cada dato.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Fecha de proceso

Comments

Se refiere a la fecha en que la liquidación fue procesada, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos dígitos para cada dato.

Definition

Fecha del pago programado

Comments

Se refiere a la fecha en que se efectuara el pago al transportista, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos dígitos para cada dato.

Definition

Fecha status registro

Comments

Se refiere a la fecha en que fue registrado su último status, activo o inactivo, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos dígitos para cada dato.

Definition

Flete total

Comments

Se refiere al importe total del flete originado por el transporte del producto, es un campo numérico de 10 posiciones.

Definition

Fecha de movimiento

Comments

Se refiere a la fecha en que se realizó el movimiento, se presenta en formato "ddmmaa" en donde, dd=día, mm=mes, aa=año, y solamente se utilizan dos dígitos para cada dato.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Folio finanzas

Comments

Se refiere al número consecutivo asignado por el area de finanzas, para admitir su conocimiento de los pagos realizados por concepto de fletes.

Definition

Grupo de producto

Comments

Se refiere a la clave asignada por Pemex, y que identifica el grupo al que pertenece un producto, es un campo numérico de un digito.

Definition

Hoja de documento

Comments

Hoja de documento asignada a la cédula de pago.

Definition

Hora del evento

Comments

Se refiere a la hora que registra el sistema al acceder a esta tabla, es un campo alfanumérico de 8 caracteres.

Definition

Impuesto al valor agregado

Comments

Este refiere al monto del impuesto al valor agregado generado por la operacion, es un campo numérico de 10 posiciones.

Definition

Identificador usuario verificador

Comments

Se refiere al identificador del usuario que realizo la verificacion de la captura del batch de liquidaciones, es un campo numérico de 7 posiciones.

Petróleos Mexicanos, C.A.S.E
Object Summary Report
(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition
Identificador

Comments
Se refiere al identificador consecutivo que asigna el sistema a un usuario, es un campo numérico.

Definition
Identificador de la tabla

Comments
Se refiere a la clave que identifica la tabla.

Definition
Identificador de quien autoriza

Comments
Se refiere al identificador (consecutivo) asignado al usuario que autorizo la cédula.

Definition
Identificador de quien elabora

Comments
Se refiere al identificador (consecutivo) asignado a la persona que elabora la cédula.

Definition
Identificador de quien revisa

Comments
Se refiere al identificador (consecutivo) asignado al usuario que revisa la cédula una vez que fue elaborada.

Definition
Identificador del usuario

Comments
Se refiere al identificador secuencial que el sistema asigna al usuario, en el momento de acceder el sistema, es un campo numérico de 7 posiciones

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Importe de la nota de embarque

Comments

Se refiere al total del importe de la nota de embarque a pagar.

Definition

Importe total por bonificaciones

Comments

Se refiere al monto total de la cantidad bonificada, es un campo numérico de 10 posiciones.

Definition

Importe total por descuentos

Comments

Se refiere al monto total de la cantidad descontada, es un campo numérico de 10 posiciones.

Definition

Importes

Comments

Es el grupo de campos que contienen la información relacionada con el importe del embarque.

Definition

Indice primario de la tabla

Comments

Se refiere a la llave primaria que permite acceder a la tablas relacionadas con pago de embarques..

Definition

Kilometros

Comments

Se refiere a la distancia en kilometros a recorrer para cubrir la ruta establecida, es un campo numérico de 4 digitos.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Letras del número de embarque

Comments

Se refiere unicamente a las letras que forman parte del número que identifica al embarque, es un campo de alfanumérico de 2 posiciones.

Definition

Llave acumulado estadísticos

Comments

Es el grupo que forman los campos que permiten el acceso al archivo, son: año, mes, origen, destino, clave de producto, grupo de producto, y clave de porteador.

Definition

Llave bonifica

Comments

Es el grupo de campos que forman la llave de acceso al archivo de bonificaciones son :

- 1.- clave del porteador
- 2.- número de documento

Definition

Llave calibracion

Comments

Es el grupo de campos que permiten el acceso al archivo de calibraciones, son:

- 1.- clave porteador
- 2.- clave autotanque
- 3.- fecha inicial de calibración
- 4.- fecha final de calibración

Definition

Llave control de batch (fclic31)

Comments

Es el grupo de campos que forman la llave primaria de acceso al archivo, son:

- 1.- número de batch
- 2.- fecha de captura

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Llave costos (fclic10)

Comments

Se refiere a la llave de acceso a la tabla de Costos, que esta formada por dos campos que son :

- 1.- fecha inicial
- 2.- fecha final

Definition

Llave cuotas

Comments

Se refiere a la llave de acceso a este archivo que esta formada por :

- 1.- kilometraje
- 2.- fecha inicio cuota
- 3.- fecha final cuota

Definition

Llave de usuarios

Comments

Es el grupo que forman los campos que permiten el acceso al archivo de usuarios son:

- 1.- clave del usuario
- 2.- identificador del usuario
- 3.- clave password

Definition

Llave embarques pagados carga líquida

Comments

Se refiere a los campos que forman la llave de acceso al archivo de datos histórico de embarques pagados:

- 1.- clave del porteador
- 2.- número de embarque histórico
- 3.- fecha de embarque

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Llave de embarques

Comments

Se refiere a la llave de acceso al archivo de embarques, y que esta formada por tres campos que son:

- 1.- clave del porteador
- 2.- número de embarque
- 3.- fecha de embarque

Definition

Llave histórico de transacciones

Comments

Es la llave de acceso al archivo de transacciones, que esta formada por :

- 1.- número secuencial del evento
- 2.- clave del registro

Definition

Llave liquidación

Comments

Es el grupo de campos que forman la llave de acceso al archivo de cédula de pago, y son :

- 1.- número de cédula
- 2.- clave tipo de cédula
- 3.- clave del porteador
- 4.- número de liquidación

Definition

Llave pago

Comments

Es el grupo formado por tres campos que permiten el acceso al archivo de cédula de pago, en donde los valores permitidos son:

- 1.- clave del porteador
- 2.- número de documento
- 3.- número de cédula

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Llave parámetros

Comments

Es el grupo formado por cuatro campos que permiten el acceso al archivo de Parámetros del sistema:

- 1.- grupo de producto
- 2.- agencia pagadora
- 3.- fecha inicial de vigencia
- 4.- fecha final de vigencia

Definition

Llave precios

Comments

Es el grupo formado por tres campos que permiten el acceso al archivo de precios de productos son:

- 1.- clave producto
- 2.- fecha inicial vigencia
- 3.- fecha final vigencia

Definition

Llave privilegios

Comments

Es el grupo de campos que forma la llave de acceso al archivo de privilegios de usuarios y son :

- 1.- clave usuario
- 2.- clave del path
- 3.- identificador

Definition

Llave productos ruta (fcllc37)

Comments

Es el grupo de campos que permiten la entrada al archivo de productos transportados comunmente en una ruta, son:

- 1.- clave origen
- 2.- clave destino
- 3.- clave ruta
- 4.- clave producto

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Llave reportados.

Comments

Se refiere a la llave de acceso al archivo de embarques reportados que forman los campos:

- 1.- embarque inicial
- 2.- embarque final

Definition

Llave ruta fclic37

Comments

Se refiere al grupo de campos que permiten el acceso al archivo, son :

- 1.- clave del origen
- 2.- clave del destino
- 3.- clave de la ruta

Definition

Llave de acceso al archivo de autotanques

Comments

La llave de acceso al archivo de autotanques, esta formada por dos datos :

- 1.- clave del porteador
- 2.- clave del autotanque

Definition

Llave tarifa (fclic26)

Comments

Es el grupo de campos que permiten el acceso al archivo de tarifas considerando como la llave principal, ya que incluye una llave secundaria de acceso (llave de ruta).

Definition

Llave usuario (fclic32)

Comments

Es el grupo de campos que permite el acceso al archivo de usuarios del sistema, son :

- 1.- clave del usuario
- 2.- identificador del usuario

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Llave xproceso

Comments

Se refiere a la llave de acceso formada por:

- 1.- fecha proceso
- 2.- número de cédula
- 3.- clave tipo de cédula

Definition

Nombre del usuario

Comments

Se refiere al nombre del usuario con acceso autorizado al sistema, es un campo alfanumérico que admite una cadena de 30 caracteres.

Definition

Número de adefas (adeudos fiscales anteriores).

Comments

Se refiere al número que identifica un adefas (adeudo fiscal anterior).

Definition

Número de embarque histórico

Comments

Se refiere al número que identifica un embarque histórico, este número esta formado por letras y dígitos por lo que este campo anida dos campos mas para especificar las letras y los dígitos que lo conforman.

Definition

Número de contrarrecibo

Comments

Se refiere al número que identifica el contrarrecibo otorgado al porteador y que ampara una o varias liquidaciones, es un campo numérico de 6 posiciones.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

llave embarques pagados carga liquida

comments

se refiere a los campos que forman la llave de acceso al archivo de datos histórico :

- 1.- clave del porteador
- 2.- número de embarque histórico
- 3.- fecha de embarque

Definition

Llave embarques pagados carga liquida

Comments

Se refiere a los campos que forman la llave de acceso al archivo de embarques pagadosde histórico :

- 1.- clave del porteador
- 2.- número de embarque histórico
- 3.- fecha de embarque

Definition

Número de batch (fclíc32)

Comments

Se refiere al número que identifica un batch, es un campo numérico de 4 posiciones.

Definition

Número de cédula

Comments

Se refiere al número unico que identifica a la cédula de pago, es un campo numérico de 4 posiciones.

Definition

Número de documento

Comments

Se refiere al número asignado al documento que ampara una cédula de pago.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Número de embarques

Comments

Se refiere al total de embarques que amparan todas las liquidaciones incluidas en un batch, es un campo numérico de 3 posiciones .

Definition

Número de liquidaciones

Comments

Se refiere a la cantidad de liquidaciones incluidas en un batch, es un campo numérico de 3 posiciones.

Definition

Número de registro

Comments

Se refiere al número secuencial que le asigna el sistema a un embarque dentro del batch.

Definition

Número de terminal

Comments

Se refiere al número asignado a la terminal en uso.

Definition

Número del embarque.

Comments

Se refiere al número que identifica al embarque, es un campo numérico de 8 posiciones.

Definition

Número de liquidación

Comments

Se refiere al número de la liquidación (factura del porteador) emitida por el porteador y que contiene uno o varios embarques, es un campo numérico de 5 posiciones.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Número secuencial del evento

Comments

Se refiere al número que el sistema asigna en forma secuencial al evento que se registra, es un campo de tipo secuencial.

Definition

Origen

Comments

Se refiere al lugar en donde se origina el flete, es un campo numérico de 5 dígitos y los valores permitidos son del 1 al 99999 que identifican a la clave del origen de la ruta de acuerdo al catálogo de orígenes y/o destinos.

Definition

Pacto de estabilidad y crecimiento económico (PECE)

Comments

Se refiere a la cantidad aplicada al pago por apoyo al pacto de estabilidad y crecimiento económico, es un campo numérico de 10 posiciones.

Definition

Peaje total

Comments

Se refiere a la cantidad total del peaje que genera el transporte de un producto, es un campo numérico de 10 posiciones.

Definition

Peso específico

Comments

Se refiere al peso específico del producto que interviene en el cálculo del pago de un flete, es un campo de numérico de 8 dígitos.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Ruta

Comments

Es el trayecto establecido que debe seguir el embarque, para ser entregado a su destino, es un campo numérico de un dígito, con valores permitidos del 1 al 3 que se refieren a la clave de la ruta, de acuerdo con el catálogo de rutas.

Definition

Ruta de reexpedición

Comments

Se refiere a la ruta utilizada en el transporte del producto al nuevo destino, es un campo de un dígito que admite valores del 1 al 3, que se refieren a la clave de la ruta, de acuerdo con el catálogo de rutas.

Definition

Secuencial de contra-recibos

Comments

Se refiere al número secuencial asignado al contra-recibo entregado al porteador.

Definition

Status

Comments

Se refiere a la clave del status que tiene el embarque, es un campo alfanumérico, que permite valores del 1 al 9 o bien de la "a" a la "z".

Definition

Status finanzas

Comments

Se refiere al conocimiento de los datos alusivos al área de finanzas.

Definition

Tarifa

Comments

Se refiere al monto de la tarifa aplicada, es un campo numérico de 8 posiciones en donde 4 son enteros y 4 son decimales.

Petróleos Mexicanos, C.A.S.E

Object Summary Report

(Reporte de Descripción de Objetos tipo Dato Generada por IEW)

Definition

Tarifa reexpedicion

Comments

Se refiere al monto de la tarifa aplicada por la reexpedicion del producto, es un campo numérico de 8 posiciones, en donde 4 son enteros y 4 son decimales.

Definition

Temperatura de carga

Comments

Se refiere a la temperatura en grados centigrados a la cual se encuentra el producto en el momento de cargar en el autotanque, es un campo numérico de 2 posiciones.

Definition

Temperatura de descaga

Comments

Se refiere a la temperatura en grados centigrados del producto en el momento de descargar, es un campo numérico de dos posiciones.

Definition

Tiempo de captura

Comments

Se refiere al tiempo que tarda el capturista en realizar la captura del batch, es un campo alfanumérico de 8 posiciones.

Definition

Tipo de evento

Comments

Se refiere a la clave que identifica el tipo de evento o de movimiento efectuado

Definition

Unidad del producto

Comments

Se refiere a la unidad de medida con que se cuantifica el volumen del producto a transportar, es un campo de un caracter, y los valores que acepta son :

- 1 para kilos
- 3 para litros.

III.2.4 Operación

En esta fase se pone a disposición de los usuarios el sistema desarrollado, probado y liberado, además de documentar los requerimientos finales del usuario tanto de información como de equipo.

III.3 Evaluación de Resultados**Sistema**

Las características más sobresalientes del sistema una vez desarrollado e implantado son las siguientes:

- Captura y proceso de documentos fuentes en lote .
- Actualización en línea de los catálogos.
- Mecanismo de seguridad en el acceso al sistema.
- Menús dinámicos en base al perfil de acceso al usuario.
- Facilidad de auditoria de los movimientos realizados a las tablas del sistema.
- Ayuda en línea sensitiva al contexto.
- Generación de Informes y consultas en forma dinámica.

Consideramos que el sistema desarrollado, satisfizo a un cien porciento las expectativas y necesidades de los usuarios finales y del área de distribución; aún más se implemento una interface para el personal ejecutivo de consultas y reportes gerenciales aun buen nivel de detalle y con una manipulación bastante sencilla, pero con capacidades muy potentes y novedosas.

Otro factor que se cuidó en buena parte fué la seguridad de la información así como el acceso al sistema, la que se implementó con la generación de un perfil específico para cada usuario, de acuerdo a las funciones del mismo. Para este aspecto se diseñó una bitácora de transacciones para tener controladas las modificaciones, las bajas y la adición de información clave de la base de datos, teniendo como datos informativos de cada transacción de este tipo el usuario quien la realizó, la hora, que tabla afectó, en que terminal y el tipo de la transacción.

Finalmente pensando siempre en brindar un sistema para los usuarios finales con mejores capacidades y facilidades las ayudas en línea, ventanas con consultas de catálogos, descripciones; el uso de mensajes en datos relevantes etc, desempeñaron un papel benéfico para una muy buena aceptación del sistema en forma integral.

Herramientas

A consecuencia de la rápida evolución del software y sobre todo el mayor número de usuarios con mayor cultura computacional y la falta de planeación estratégica en los desarrollos de sistemas; han puesto en crisis a las áreas de sistemas.

Hoy en día es muy difícil para las empresas invertir cantidades grandes de recursos para las áreas de sistemas con personal que no tiene bien definidas sus funciones y que carecen de objetivos claramente marcados, sistemas en producción que no cuentan con la documentación adecuada y otros aún en desarrollo que no saben bien hacia donde van. Todo ello por falta de una metodología organizacional y una metodología implementada para el control del desarrollo de sistemas.

Dada esta problemática, Pemex introdujo como parte de una solución global la implementación del uso de herramientas CASE, con las cuales se pueden estandarizar

los desarrollos de una gran empresa para tratar de normar el desarrollo de sistemas, así como la manipulación y el mantenimiento de los mismos.

Con el desarrollo del Sistema Pago de Fletes a Transportista se logro introducir una metodología para el desarrollo, estandarización para varios aspectos de la programación y una nueva mentalidad para la solución de necesidades

Es importante recordar que el Ciclo de Vida del Software debe iniciar con una fase de planeación estratégica de la información y un modelado de datos con gran visión. La tecnología CASE esta compuesta por un conjunto de herramientas altamente integradas y no son sólo eso, son elementos unidos por una interfaz de usuario común, con un lenguaje común para alimentarse uno a otro de manera automática con el fin de cumplir la tarea encomendada.

Puede decirse que las empresas se estan enfrentando a una crisis, como consecuencia del crecimiento descontrolado sobre las peticiones de necesidades de los usuarios. Es por ello que se ha buscado controlar estas necesidades mediante una adecuada planificación y reutilización del software.

Buscando integrar en un sólo producto las variadas necesidades del usuario y apoyandonos en una herramienta CASE, es como se desarrolló el sistema



CONCLUSIONES



Conclusiones

CONCLUSIONES

Las herramientas de Ingeniería de Software, asistida por computadoras (C.A.S.E), abarcan cada etapa del proceso de la ingeniería y cada actividad que se desarrolla a lo largo del mismo, su apoyo va desde la etapa de planeación estratégica hasta la posibilidad de generación automática de código.

Con el transcurso del tiempo, las herramientas CASE se integran al tejido de la ingeniería de Software. Actualmente las gentes que conforman las áreas de sistemas deben apoyarse en ellas para optimizar y realizar una adecuada fase de análisis, diseño e implementación de los sistemas.

El entorno CASE combina mecanismos de integración de datos, herramientas y sobre todo, una adecuada interacción hombre-máquina. La integración de datos, se alcanza por medio del intercambio común de archivos, a base de compartir o interoperar con los datos totalmente integrados.

La integración hombre-máquina se consigue a través de la interfase de estándares que adoptan un papel cada vez más importante en la Ingeniería del Software.

CASE se encuentra basado y orientado a los objetos que constituyen "el centro de acumulación y almacenamiento" de la información. Este sistema se implementa utilizando un sistema de desarrollo de bases de datos que proporciona todas las características de estandarización.

Conclusiones

El propósito de utilizar una herramienta C.A.S.E como es el IEW fué para mejorar, la planeación, el diseño y el desarrollo, así como el de utilizar una metodología eficiente capaz de automatizar cualquiera de las etapas que lo integran.

Nuestra herramienta C.A.S.E nos permitió planificar eficazmente el proyecto, así como el corregir fácilmente los errores detectados dentro del diseño y poder optimizar el código resultante de nuestros programas. Dando como resultado:

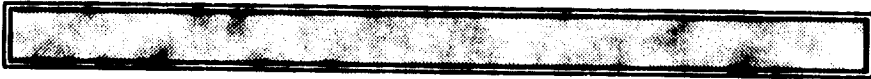
- Mejor rendimiento del sistema.
- Detección de fallas del diseño en la implementación del modelo de datos.
- Corregir y perfeccionar nuestra base de datos.
- Formar un área de sistemas con una disciplina y una forma de trabajo común más adecuada a los avances tecnológicos.

Además, el trabajo de tesis desarrollado nos permitió hacer uso de los conocimientos y habilidades adquiridos durante el periodo de formación universitaria, también nos permitió el poder seleccionar los medios y las herramientas adecuadas para dar una solución al problema planeado.

Los conocimientos adquiridos en el área de ingeniería de sistemas, bases de datos, comunicaciones nos brindaron las bases para enfrentar este problema, comprobando una vez más que la formación académica de un profesionalista, es la herramienta fundamental con la que se cuenta para proporcionar soluciones eficientes dentro del área de sistemas y de cualquier área del desarrollo humano.



DOCUMENTOS



```

globals
  "clqfa08.4gl"          (INCLUYE LA DEFINICION DE VARIABLES GLOBALES)
)
{*****
FUNCION:      CONSULTA
OBJETIVO:     CONSULTAR DINAMICAMENTE LA INFORMACION DE LA TABLA DE
              PRODUCTOS
AUTOR:        Rodolfo Isaac Carreola R.
PARAMETROS:   NINGUNO
QUIEN LA LLAMA: FCLFA08.4GL, FCLFC08.4GL
*****}
function consultas()

  open form forma01 from "fclfa08" (ABRE EL ARCHIVO QUE CONTIENE EL FORMATO)
  display form forma01             (DESPLIEGA EL FORMATO DE CONSULTA)
  display w_idpro at 3,1
  let w_fecha = today
  display w_fecha using "dd/mm/yy" at 4,1
  display w_idtty at 4,72

  call limpia("CONSULTAS DINAMICAS",1)

  while true
    {=== LEEMOS DE PANTALLA EL CRITERIO DE BUSQUEDA QUE EL USUARIO TECLEO ===}
    construct by name w_where on fclbg02.*

    {==== CONSTRUIMOS EL SELECT QUE LEERA LOS REGISTROS CORRESPONDIENTES ====}
    let w_select="select * from fclbg02 where ",w_where clipped,
                " order by c_prod"

    {===== PREPARA EL SELECT ANTERIOR PARA QUE PUEDA SER EJECUTADO =====}
    prepare w_read from w_select

    {==== SE DECLARA UN APUNTADOR PARA PODER LEER UNO A UNO LOS REGISTRO ====}
    {==== ENCONTRADOS POR EL SELECT CONSTRUIDO ANTERIORMENTE          =====}
    declare w_apun scroll cursor for w_read

    open w_apun                    (SE ABRE EL APUNTADOR, Y SE POSICIONA AL INICIO)
                                  (DE LOS REGISTROS LEIDOS POR EL SELECT )

    fetch first w_apun into w_produ.*          (LEE EL PRIMER REGISTRO)
    if status=notfound then                    (VALIDA SI LO LOCALIZO)
      error "                                REGISTRO NO LOCALIZADO "
      prompt " Desea otra consulta (S/N) ? " for char x
      if x matches "[sS]" then
        continue while
      end if
      exit while
    end if
    fetch next w_apun into w_produ.*
    if status = notfound then
      fetch first w_apun into w_produ.*
      call display_prod()                    (DESPLIEGA EL REG EN PANT)
      prompt " Desea otra consulta (S/N) ? " for char x
      if x matches "[sS]" then
        continue while
      end if
      exit while
    end if
  end while
}

```

```

else
  fetch first w_apun into w_produ.*      (HAY VARIOS REGISTROS A DESPLEGAR)
  call display_prod()                    (DESPLIEGA EL REG EN PANT)
  (===== GENERA EL MENU DE CONSULTA DINAMICA =====)
menu "CONSULTA"
  command "Primer reg" "Consulta el primer registro"
    help 1441
    fetch first w_apun into w_produ.*    (LEE EL PRIMER REGISTRO)
    call display_prod()                  (DESPLIEGA EL REG EN PANT)
    next option "Siguiente"
  command "Siguiente" "Consulta el siguiente registro"
    help 1442
    fetch next w_apun into w_produ.*     (LEE EL SIGUIENTE REGISTRO)
    if (status = notfound) then
      error "          NO HAY MAS REGISTROS          "
    next option "Anterior"
    end if
    call display_prod()                  (DESPLIEGA EL REG EN PANT)
  command "Anterior" "Consulta el registro previo"
    help 1443
    fetch prior w_apun into w_produ.*    (LEE EL REGISTRO ANTERIOR)
    if (status = notfound) then
      error "          NO HAY MAS REGISTROS          "
    next option "Siguiente"
    end if
    call display_prod()                  (DESPLIEGA EL REG EN PANT)
  command "Ultimo" "Consulta el ultimo registro"
    help 1444
    fetch last w_apun into w_produ.*     (LEE EL ULTIMO REGISTRO)
    call display_prod()                  (DESPLIEGA EL REG EN PANT)
  command key("Y") "aYuda"
    "Muestra ayuda para construir criterio de seleccion"
    call showhelp(9999)
  command key(f3)
    clear form
  exit menu
end menu
prompt " Desea otra consulta (S/N) ? " for char x
if x matches "[sS]" then
  continue while
end if
exit while
end if
end while
close form forma01
end function

```

```

(*****
FUNCION:   DISPLAY_PROD
OBJETIVO:  DESPLIEGA EN PANTALLA EL REGISTRO DE LA TABLA DE PRODUCTOS,
           ASI COMO LA DESCRIPCION DEL GRUPO DEL PRODUCTO
PARAMETROS: NINGUNO
*****)
function display_prod()
  display by name w_produ.c_prod thru w_produ.n_unid
  if (w_produ.n_unid ="1") then
    display "litros" to formonly.d_unid
  else
    display "kilos" to formonly.d_unid
  end if
end if

```

```

display by name w_produ.n_pesp
display by name w_produ.n_coefe
display by name w_produ.n_factf
display by name w_produ.c_grupo
if valida_gru() then
    display w_grupo.d_grupo to formonly.d_grupo
else
    display "*** NO EXISTE EL GRUPO ***" to formonly.d_grupo
end if
display by name w_produ.c_cuota
display by name w_produ.n_capap
if not valida_pre("123199") then
    let w_preci.n_imprec = 0
end if
display by name w_preci.n_imprec
display by name w_preci.f_iniprec
display by name w_preci.f_finprec
display by name w_produ.f_movexp
display by name w_produ.c_status
if (w_produ.c_status = "1") then
    display "Activo" to formonly.d_status
else
    display "Baja" to formonly.d_status
end if
display by name w_produ.f_status
end function

{*****
FUNCION:      VALIDA_GRU
OBJETIVO:    VALIDAR QUE EXISTA O NO UN REGISTRO DE LA TABLA DE GRUPOS DE
              PRODUCTO
PARAMETROS:  NINGUNO
*****}
function valida_gru()
    select * into w_grupo.* from tclbg04                (LEE UN SOLO REGISTRO)
        where c_grupo = w_produ.c_grupo
        if status = notfound then
            return 0                                    (NO ENCONTRO EL
REGISTRO)
        end if
        return 1                                      (SI ENCONTRO EL REGISTRO)
end function

{*****
FUNCION:      LIMPIA
OBJETIVO:    LIMPIAR MENSAJES DE LA PANTALLA
PARAMETROS:  EL TITULO DE LA PANTALLA Y LA BANDERA QUE INDICA SI LA LIMPIEZA DE
              MENSAJES ES AL INICIO O AL FINAL
*****}
function limpia(w_texto,w_tipo)
    define w_texto      char(50),
        w_tipo          smallint,
        w_columna      smallint
    let w_columna = (80 - length(w_texto)) / 2
    if w_tipo then
        (===== BORRA EL MENU DE MANTENIMIENTO =====)
        display "" at 1,1
        display "" at 2,1
        (=====)
        display w_texto clipped AT 5,w_columna attribute (reverse)
    end if
end function

```

```
    let sw1=0          (OJO .. SE INICIALIZA EL SWITCH QUE CONTROLA AL LOOP)
  else
    clear form
    display "-----" at 5,30 (BORRA EL TITULO DE LA PANTALLA)
    display "" at 23,1        (BORRA EL AVISO DE MOVIMIENTO REALIZADO)
  end if
end function
```

□

```

globals
  "clgfa17.4gl"          {INCLUYE LA DEFINICION DE VARIABLES GLOBALES}

{*****}
FUNCION:      CONSULTA
OBJETIVO:     CONSULTAR DINAMICAMENTE LA INFORMACION DE LA TABLA DE
              PRECIO DE PRODUCTOS
PARAMETROS:   NINGUNO
NOMBRE:       ISRAEL ROMERO M.
QUIEN LA LLAMA: FCLFA17.4GL, FCLFC17.4GL
{*****}
}
function consultas()
  define
    ind smallint
    open form forma01 from "fclfa17" {ABRE EL ARCHIVO QUE CONTIENE EL FORMATO}
    display form forma01           {DESPLIEGA EL FORMATO DE CONSULTA}
    let w_fecha = today
    display w_idpro at 3,1
    display w_fecha using "dd/mm/yy" at 4,1
    display w_idtty at 4,72

    call limpia("CONSULTAS DINAMICAS",1)

    {=== LEEMOS DE PANTALLA EL CRITERIO DE BUSQUEDA QUE EL USUARIO TECLEO ===}
    construct by name w where on fclbv07.*
    {===== CONSTRUIMOS EL SELECT QUE LEERA LOS REGISTROS CORRESPONDIENTES =====}
    let w_select="select * from fclbv07 where ",w_where clipped,
                " order by c_prod, f_iniprec, f_finprec"

    {===== PREPARA EL SELECT ANTERIOR PARA QUE PUEDA SER EJECUTADO =====}
    prepare w_read from w_select

    {===== SE DECLARA UN APUNTADOR PARA PODER LEER UNO A UNO LOS REGISTRO =====}
    {=== ENCONTRADOS POR EL SELECT CONSTRUIDO ANTERIORMENTE          =====}
    declare w_apun cursor for w_read
    let ind = 0
    foreach w_apun into w_preci.*
      let ind = ind + 1
      let p_precio[ind].c_prod = w_preci.c_prod
      let p_precio[ind].f_iniprec = w_preci.f_iniprec
      let p_precio[ind].f_finprec = w_preci.f_finprec
      let p_precio[ind].n_imprec = w_preci.n_imprec
      if not valida_pro() then
        let w_produ.d_prod = "*****SIN DESCRIPCION*****"
      end if
      if w_produ.c_status = 1 then
        let p_precio[ind].d_status="Activo"
      else
        let p_precio[ind].d_status="baja"
      end if
      let p_precio[ind].d_nompro=w_produ.d_prod
      if ind >= 400 then
        exit foreach
      end if
    end foreach
    open window win1 at 12,3
    with form "cllfa17" attribute(message line first,border)
    message "Oprima <F3>=Pag. Sig. <F4>=Pag. Ant. <ESC>=Para Terminar"
  end define
}

```

```

    call set_count(ind)
    display array p_precio to s_arreg.*
    close window win1
end function

{*****}
FUNCION: LIMPIA
OBJETIVO: LIMPIAR MENSAJES DE LA PANTALLA
PARAMETROS: EL TITULO DE LA PANTALLA Y LA BANDERA QUE INDICA SI LA LIMPIEZA DE
MENSAJES ES AL INICIO O AL FINAL
{*****}
function limpia(w_texto,w_tipo)
    define w_texto char(50),
           w_tipo smallint,
           w_columna smallint
    let w_columna = (80 - length(w_texto)) / 2
    if w_tipo then
        {===== BORRA EL MENU DE MANTENIMIENTO =====}
        display "" at 1,1
        display "" at 2,1
        {=====}
        display w_texto clipped AT 5,w_columna attribute (reverse)
        let sw1=0 {OJO .. SE INICIALIZA EL SWITCH QUE CONTROLA AL LOOP}
    else
        clear form
        display "-----" at 5,30 {BORRA EL TITULO DE LA PANTALLA}
        display "" at 22,1 {BORRA EL AVISO DE MOVIMIENTO REALIZADO}
        display "" at 23,1 {BORRA EL AVISO DE MOVIMIENTO REALIZADO}
    end if
end function

```

```

globals
  "c11fa33.4gl"                {INCLUYE LA DEFINICION DE VARIABLES GLOBALES}

{*****}
FUNCION:          CONSULTA
OBJETIVO:         CONSULTAR DINAMICAMENTE LA INFORMACION DE LA TABLA DE
                  PRODUCTOS POR RUTA
PARAMETROS:      NINGUNO
NOMBRE:          LAURA JACOME DEL M.
QUIEN LA LLAMA:  FCLFA33.4GL, FCLFC33.4GL
{*****}
function consultas()
  define
    w_amod like fclbg02.c_prod,
    i,k,k_old,j,
    sw,liñ      smallint

  open form forma01 from "fclfa33" {ABRE EL ARCHIVO QUE CONTIENE EL FORMATO}
  display form forma01             {DESPLIEGA EL FORMATO DE CONSULTA}
  let w_fecha = today
  display w_idpro at 3,1
  display w_fecha using "dd/mm/yy" at 4,1
  display w_idtty at 4,72
  call limpia("CONSULTAS DINAMICAS",1)

  while true
    {=== LEEMOS DE PANTALLA EL CRITERIO DE BUSQUEDA QUE EL USUARIO TECLEO ===}

    options
      input no wrap {ESTA OPCION ES PARA QUE EL CURSOR NO REGRESE A HOME }
                  {DESPUES DEL ULTIMO CAMPO DE LA PANTALLA Y SE TERMINE }
                  {EL INPUT CON EL ULTIMO <Enter> }

    display "" at 23,1

    {=== LEEMOS DE PANTALLA LA RUTA A BUSCAR DENTRO DE LA TABLA ===}
    input by name w_prodrut.c_origen thru w_prodrut.c_ruta
      on key(f1)
        case
          when infield(c_origen)
            call showhelp(1076)
          when infield(c_destino)
            call showhelp(1077)
          when infield(c_ruta)
            call showhelp(1078)
        end case

    {===== VALIDA LA CLAVE DE ORIGEN EN CATALOGO =====}
    after field c_origen      {DESPUES DE TECLEAR LA CLAVE DEL }
                              {ORIGEN }

    display "" at 23,1
    let w_val = w_prodrut.c_origen
    if not valida_ori(w_val) then {VALIDA QUE EXISTA LA CLAVE EN CAT }
      error
      "      La clave del Origen no existe en el catalogo      "
    next field c_origen
    end if
    display w_orides.d_oridel to formonly.d_origen

```



```

(===== VALIDA LA CLAVE DEL DESTINO EN CATALOGO =====)
after field c_destino      (DESPUES DE TECLEAR EL DESTINO )
  let w_val = w_prodrut.c_destino
  if not valida_ori(w_val) then (VALIDA QUE EXISTA LA CLAVE EN CAT)
    error
"                               La clave que tecleo no existe
"
    sleep 1
    next field c_destino
  end if

(===== DESPLIEGA LA DESCRIPCION DEL DESTINO =====)
  display w_orides.d_oridel to formonly.d_destino

after field c_ruta      (DESPUES DE TECLEAR LA RUTA )
  if not valida_rut() then (VALIDA QUE EXISTA EN TABLA DE
    (CARACTERISTICAS RUTAS)
    error " La clave Origen-Destino-Ruta, no existe en Catalogo "
    next field c_origen
  end if

  on key(f3)
    let int_flag = true
    exit input
  end input

if int_flag then
  let int_flag = false
  exit while
end if

declare apu01 scroll cursor for
  select * from fclbn02 whe:~
    c_origen = w_prodrut.c_origen and
    c_destino= w_prodrut.c_destino and
    c_ruta   = w_prodrut.c_ruta

open apu01      (SE ABRE EL APUNTADOR, Y SE POSICIONA AL INICIO)
                (DE LOS REGISTROS LEIDOS POR EL SELECT )

fetch first apu01 into w_prodxrut.*      (LEE EL PRIMER REGISTRO)
if status = notfound then                (VALIDA SI LO LOCALIZO)
  error "                                REGISTRO NO LOCALIZADO

"
  return
end if

(==== DESPLIEGA LA FORMA DEL ARREGLO PARA PRODS =====)

open window w3 at 8,5 with form "c11fa33" attribute (border)
display " F2=Pag Ant F4=Pag Sig " at 1,1 attribute (reverse)
  let j = 1
  while true
    let sw = 0
    let k = 0
    for i = j to j+9
      fetch absolute i apu01 into w_prodxrut.*
      if status = notfound then

```

```

        let sw = 1
        exit for
    end if

    select * into w_prod.* from fclbg02
        where c_prod = w_prodxrut.c_prod
    if status = notfound then
        let w_prod.d_prod = " No existe en Catalogo"
    end if

    let k = k + 1
    let p_prod[k].c_prod = w_prodxrut.c_prod
    let p_prod[k].d_nombre = w_prod.d_prod
end for
if k = 0 then
    if j < 0 then
        let j = j + 10
    else
        let j = j - 10
    end if
    let k = k_old
end if
call set_count(k)
let k_old = k
display array p_prod to s_prod.*
    on key (f4) (SIGUIENTE PAGINA)
        let sw = 2
        exit display
    on key (f2) (PAGINA ANTERIOR)
        let sw = 3
        exit display
    on key (f3) (SALIR)
        let int_flag = true
        exit display
end display
if int_flag = true then
    let int_flag = false
    exit while
end if
if sw = 2 then
    let j = j + 10
    continue while
end if
if sw = 3 then
    let j = j - 10
    continue while
end if
let lin = arr_curr()
end while
close window w3
close apu01
end while
end function

```

```

*****
PROGRAMA:   FCLFK01.4GL
OBJETIVO:  CONSULTA DE LOS MOVIMIENTOS DIARIOS DE EMBARQUES DEL SISTEMA
           DE PAGO DE FLETES "CARGA LIQUIDA"
FUNCION:   CONSULTAS
AUTOR:    MA. DOLORES PENALOZA
           03/02/93
*****

```

```

database fletes          (ABRE LA BASE DE DATOS DE FLETES)

```

```

(===== DECLARACION DE VARIABLES GLOBALES =====)

```

```

globals
  define w_movi record like fclbv09.*,      (BUFFER DEL ARCHIVO DE MOVIMIEN)
        w_combat record like fclbv12.*,    (BUFFER DE LA TABLA DE CONT BATC)
        w_fecha date,                     (FECHA DEL SISTEMA )
        w_vig,                             (VIGENCIA ACEPTABLE DE FECHAS)
        w_batch,                           (NUMERO DE BATCH A CAPTURAR)
        w_liqui integer,                   (NUMERO DE LIQUIDACION TEMPORAL )
        w_timei datetime hour to second,  (HORA DE INICIO DE CAPTURA )
        w_timef datetime hour to second,  (HORA DE FIN DE CAPTURA DE BATCH)
        w_sw,                              (BANDERA DE CONTROL DEL INPUT )
        w_sw1,                             (BANDERA DE CONTROL DE BATCH'S )
        w_prod,                            (NUMERO DE PRODUCTO TEMPORAL )
        w_h1,                              (LA HORA DE INICIO DE CAPTURA )
        w_h2,                              (LA HORA DE FIN DE CAPTURA )
        w_mi,                              (LOS MINUTOS DE INICIO DE CAPTU )
        w_m2,                              (LOS MINUTOS DE FIN DE CAPTURA )
        w_s1,                              (LOS SEGUNDOS DE INICIO DE CAPTU)
        w_s2,                              (LOS SEGUNDOS DE FIN DE CAPTURA )
        p_anio smallint,                   (ANIO DE LA EMBARCACION )
        x,
        w_tipobatch char(1),               (TIPO DE BATCH Local, Foraneo )
        w_user char(7),                    (CLAVE DEL USUARIO )
        w_ident smallint                   (IDENTIFICADOR DEL USUARIO )
end globals

```

```

(===== PROGRAMA PRINCIPAL =====)

```

```

main
  defer interrupt

  (===== ABRE Y DESPLIEGA EL FORMATO DE CAPTURA =====)
  open form forma01 from "fclfk01"
  clear screen
  display form forma01
  let w_fecha = today
  display w_fecha at 4,1
  (===== PROCESO INTERACTIVO DE CAPTURA =====)
  call consultas()
  clear screen
end main
(===== FIN DEL PROGRAMA PRINCIPAL =====)

```

```

*****
FUNCION:   CONSULTA
OBJETIVO:  CONSULTAR DINAMICAMENTE LA INFORMACION DE LA TABLA DE
           USUARIOS
PARAMETROS: NINGUNO
*****

```

```

)
function consultas()

```

```

call limpia("CONSULTAS DINAMICAS",1)

{=== LEEMOS DE PANTALLA EL CRITERIO DE BUSQUEDA QUE EL USUARIO TECLEO ===}
construct by name w_where on fclbv09.*

{===== CONSTRUIMOS EL SELECT QUE LEERA LOS REGISTROS CORRESPONDIENTES =====}
let w_select="select * from fclbg13 where ",w_where clipped

{===== PREPARA EL SELECT ANTERIOR PARA QUE PUEDA SER EJECUTADO =====}
prepare w_read from w_select

{=== SE DECLARA UN APUNTADOR PARA PODER LEER UNO A UNO LOS REGISTRO ===}
{=== ENCONTRADOS POR EL SELECT CONSTRUIDO ANTERIORMENTE =====}
declare w_apun scroll cursor for w_read

open w_apun          {SE ABRE EL APUNTADOR, Y SE POSICIONA AL INICIO}
                    {DE LOS REGISTROS LEIDOS POR EL SELECT      }

fetch first w_apun into w_usuar.*          {LEE EL PRIMER REGISTRO}
if status=notfound then                    {VALIDA SI LO LOCALIZO}
  error
  "          REGISTRO NO LOCALIZADO          "
  call limpia(" ",0)          {BORRA AVISOS DE LA PANTALLA}
  return
end if

call display_usua()          {DESPLIEGA EL REG EN PANT}

{===== GENERA EL MENU DE CONSULTA DINAMICA =====}
menu "CONSULTA"
  command "Primer reg" "Consulta el primer registro"
    fetch first w_apun into w_usuar.*          {LEE EL PRIMER REGISTRO}
    call display_usua()          {DESPLIEGA EL REG EN PANT}
    next option "siguiente"
  command "Siguiente" "Consulta el siguiente registro"
    fetch next w_apun into w_usuar.*          {LEE EL SIGUIENTE REGISTRO}

FUNCION:  ALTAS
OBJETIVO:  DAR DE ALTA UN NUEVO REGISTRO EN LA TABLA DE MOVIMIENTOS
PARAMETROS:  NINGUNO
*****}
function altas()
  {options
  input wrap)          {ESTA OPCION ES PARA QUE EL CURSOR REGRESE A HOME      }
                    {DESPUES DEL ULTIMO CAMPO DE LA PANTALLA          }

{===== PROCESO INTERACTIVO PARA DAR DE ALTA REGISTROS =====}
while true

  {===== LEE LOS DATOS DE PANTALLA =====}
  construct by
  input by name w_movi.n_liqui thru w_movi.n_embarq, p_anio,
    w_movi.f_embarq thru w_movi.c_destino,
    w_movi.c_destre,w_movi.c_auto thru
    w_movi.n_temdes without defaults
  {===== VALIDA NUMERO DE LIQUIDACION =====}
  after field n_liqui
  if val_liq() then
    error "**** El Numero de liquidacion no puede ser NEGATIVO ****"

```

```

        next field n_liqui
    end if
    if w_liqui = w_movi.n_liqui then (VALIDA SI ES LA MISMA LIQUI)
        next field n_embarq
    end if
    let w_liqui = w_movi.n_liqui
    let w_prod = 0

(===== VALIDA FECHA DE LIQUIDACION =====)
after field f_liqui
    if val_fec_liq() then
        error "*** La Fecha de liquidacion esta fuera de rango ***"
    next field f_liqui
end if

(===== VALIDA CLAVE DE PORTEADOR =====)
after field c_portea
    if val_por() then
        error "*** El Numero de Porteador no puede ser NEGATIVO ***"
    next field c_portea
end if

(===== VALIDA NUMERO DE EMBARQUE =====)
after field n_embarq
    if val_num_emb() then
        error
        "*** El Numero de Embarque no puede ser NEGATIVO o CERO ***"
    next field n_embarq
end if

(===== VALIDA EL ANIO DEL NUMERO DE EMBARQUE =====)
after field p_anio
    if val_anio() then
        error
        "*** El Anio del numero de Embarque no puede ser NEGATIVO o CERO ***"
    next field p_anio
end if

(===== VALIDA FECHA DE EMBARQUE =====)
after field f_embarq
    if val_fec_emb() then
        error "*** La Fecha del embarque esta fuera de rango ***"
    next field f_embarq
end if

(===== VALIDA CLAVE DE ORIGEN =====)
after field c_origen
    if val_ori() then
        error "*** La clave de Origen no puede ser NEGATIVO ***"
    next field c_origen
end if

(===== VALIDA CLAVE DE DESTINO =====)
after field c_destino
    if val_des() then
        error "*** La clave de Destino no puede ser NEGATIVO ***"
    next field c_destino
end if
let w_movi.c_ruta = 1

```

```

)
(===== VALIDA CLAVE DE DESTINO DE REEXPEDICION =====)
after field c_destre
  if w_movi.c_destre = 0 then
    let w_movi.c_rutare = 0
    next field c_auto
  end if
  if val_des_ree() then
    error "*" La clave de Destino de reexp. no puede ser NEGATIVO *"
  next field c_destre
  end if
  let w_movi.c_rutare = 1

(===== VALIDA CLAVE DE AUTOTANQUE =====)
(
  after field c_auto
  if val_aut() then
    error "*** El Numero de Autotanque no puede ser NEGATIVO ***"
  next field c_auto
  end if
)

(===== VALIDA CLAVE DE PRODUCTO =====)

(===== OJO AQUI SE OMITI LA CAPTURA DEL PRODUCTO SI ESTE ES =====)
(===== GAS O TURBOSINA =====)
before field c_prod
  if w_prod = 280 or w_prod = 120 then
    next field n_carga
  end if

after field c_prod
  if val_pro() then
    error "*** El Numero de Producto no puede ser NEGATIVO ***"
  next field c_prod
  end if

(===== VALIDA UNIDAD DE PRODUCTO =====)
after field n_unid
  if val_uni() then
    error "*** La unidad de Producto es incorrecta <1,2,3> ***"
  next field n_unid
  end if

(===== VALIDA CANTIDAD CARGADA DE PRODUCTO =====)
after field n_carga
  if val_can() then
    error
    "*** La Cantidad Cargada no puede ser NEGATIVA o CERO ***"
  next field n_carga
  end if
  if w_movi.n_unid = "3" then
    next field n_import
  end if

(===== VALIDA LA TEMPERATURA DE CARGA DE PRODUCTO =====)
(
  after field n_temcar
  if val_tem_car() then
    error "*** La Temperatura de Carga no puede ser NEGATIVO ***"
  next field n_temcar
  end if
)

```

```

(===== VALIDA CANTIDAD DE DESCARGA DE PRODUCTO =====)
after field n_desca
  if val_can_des() then
    error
    "*** La Cantidad Descargada no puede ser NEGATIVA o CERO ***"
  next field n_desca
end if

(===== VALIDA TEMPERATURA DE DESCARGA DE PRODUCTO =====)
after field c_temdes
  if val_tem_des() then
    error "*** La Temperatura de descarga es incorrecta ***"
  next field c_temdes
end if

(===== VALIDA IMPORTE DEL EMBARQUE =====)
after field n_import
  if val_imp() then
    error
    "*** El Importe del embarque no puede ser NEGATIVO o CERO ***"
  next field n_import
end if
  if w_movi.n_unid = "3" then
    let w_movi.n_temcar = 0
    let w_movi.n_desca = w_movi.n_carga
    let w_movi.n_temdes = 0
    exit input
  end if

(===== VALIDA SI OPRIMIO F2 PARA LIMPIAR PANTALLA =====)
on key (f2)
  let w_sw = 1
  exit input
on key (f3)
  let int_flag = true
  exit input
on key (f4)
  let w_timef = time (OBTIENE LA HORA EN QUE TERMINO DE CAPTURAR)
  let w_sw = 2
  exit input
on key (f1)
  let w_sw = 3
  exit input

end input

(===== VALIDA SI DESEA ABORTAR EL PROCESO ITEACTIVO DE CAPTURA =====)
if int_flag then
  let int_flag = false
  exit while
end if

(===== VALIDA SI DESEA LIMPIAR LA PANTALLA Y ABORTAR EL REGISTRO =====)
if w_sw = 1 then
  let w_sw = 0
  call Inicializa_var()
  continue while
end if

(===== TERMINA DE CAPTURAR UN LOTE DE REGISTROS =====)

```



```

let w_vig          = 365.25*5          (FECHAS HASTA CON 5 ANIOS DE
ATRASO)
let w_liqui        = 0
let w_prod         = 0
let w_sw           = 0
initialize w_movi.* to null
let w_movi.c_ruta  = 1
let w_movi.c_destre = 0
let w_movi.c_rutare = 1
let w_movi.n_import = 0
end function

{*****}
FUNCION:    VAL_LIQ
OBJETIVO:   VALIDAR EL NUMERO DE LIQUIDACION
PARAMETROS: NINGUNO
{*****}
}
function val_liq()
  if w_movi.n_liqui <= 0 then
    return 1
  end if
return 0
end function

{*****}
FUNCION:    VAL_FEC_LIQ
OBJETIVO:   VALIDAR LA FECHA DE LIQUIDACION
PARAMETROS: NINGUNO
{*****}
}
function val_fec_liq()
  if ((w_fecha - w_movi.f_liqui) > w_vig) or (w_movi.f_liqui > w_fecha) then
    return 1
  end if
return 0
end function

{*****}
FUNCION:    VAL_POR
OBJETIVO:   VALIDAR EL NUMERO DE PORTEADOR
PARAMETROS: NINGUNO
{*****}
}
function val_por()
  if w_movi.c_porlea <= 0 then
    return 1
  end if
return 0
end function

{*****}
FUNCION:    VAL_NUM_EMB
OBJETIVO:   VALIDAR EL NUMERO DE EMBARQUE
PARAMETROS: NINGUNO
{*****}
}
function val_num_emb()
  if w_movi.n_embarq <= 0 then
    return 1

```

```

    end if
    return 0
end function

{*****}
FUNCION:    VAL_ANIO
OBJETIVO:   VALIDAR EL ANIO DEL NUMERO DE EMBARQUE
PARAMETROS: NINGUNO
{*****}
}
function val_anio()
    if p_anio <= 0 then
        return 1
    end if
    return 0
end function

{*****}
FUNCION:    VAL_FEC_EMB
OBJETIVO:   VALIDAR LA FECHA DE EMBARQUE
PARAMETROS: NINGUNO
{*****}
}
function val_fec_emb()
    if ((w_fecha - w_movi.f_embarq) > w_vig) or (w_movi.f_embarq > w_fecha)
then
        return 1
    end if
    return 0
end function

{*****}
FUNCION:    VAL_ORI
OBJETIVO:   VALIDAR LA CLAVE DE ORIGEN
PARAMETROS: NINGUNO
{*****}
}
function val_ori()
    if w_movi.c_origen <= 0 then
        return 1
    end if
    return 0
end function

{*****}
FUNCION:    VAL_DES
OBJETIVO:   VALIDAR LA CLAVE DE DESTINO
PARAMETROS: NINGUNO
{*****}
}
function val_des()
    if w_movi.c_destino <= 0 then
        return 1
    end if
    return 0
end function

{*****}
FUNCION:    VAL_DES_REE
OBJETIVO:   VALIDAR LA CLAVE DE DESTINO DE REEXPEDICION

```

```

PARAMETROS: NINGUNO
.....
)
function val_des_ree()
  if w_movi.c_destre <= 0 then
    return 1
  end if
return 0
end function

{.....}
FUNCION: VAL_AUT
OBJETIVO: VALIDAR LA CLAVE DE AUTOTANQUE
PARAMETROS: NINGUNO
.....
)
function val_aut()
  if w_movi.c_auto <= 0 then
    return 1
  end if
return 0
end function

{.....}
FUNCION: VAL_PRO
OBJETIVO: VALIDAR LA CLAVE DEL PRODUCTO
PARAMETROS: NINGUNO
.....
)
function val_pro()
  if w_movi.c_prod <= 0 then
    return 1
  end if
return 0
end function

{.....}
FUNCION: VAL_UNI
OBJETIVO: VALIDAR LA CLAVE DE UNIDAD DEL PRODUCTO
PARAMETROS: NINGUNO
.....
)
function val_uni()
  if w_movi.n_unid not matches "[1|3]" then
    return 1
  end if
return 0
end function

{.....}
FUNCION: VAL_CAN
OBJETIVO: VALIDAR LA CANTIDAD CARGADA DEL PRODUCTO
PARAMETROS: NINGUNO
.....
)
function val_can()
  if w_movi.n_carga <= 0 then
    return 1
  end if
return 0

```

```

end function

{.....}
FUNCION: VAL_CAN_DES
OBJETIVO: VALIDAR LA CANTIDAD DESCARGADA DEL PRODUCTO
PARAMETROS: NINGUNO
{.....}
}
function val_can_des()
  if w_movi.n_desca <= 0 then
    return 1
  end if
return 0
end function

{.....}
FUNCION: VAL_IMP
OBJETIVO: VALIDAR EL IMPORTE DEL EMBARQUE
PARAMETROS: NINGUNO
{.....}
}
function val_imp()
  if w_movi.n_import <= 0 then
    return 1
  end if
return 0
end function

{.....}
FUNCION: SOLICITA_BATCH
OBJETIVO: OBTENER EL NUMERO DE BATCH QUE SE DESEA CAPTURAR Y VALIDAR QUE NO
ESTE CERRADO
PARAMETROS: NINGUNO
{.....}
function solicita_batch()

  define w_tit char(7)

  display "Oprima <F3> para abortar" at 5,54
  (==== ABRE UNA VENTANA PARA SOLICITAR EL NUMERO ====)
  open window v3 at 10,25 with 6 rows, 40 columns
  attribute (border,prompt line last - 1)

  while true

    (===== SOLICITA EN NUMERO DE BATCH =====)
    prompt " Numero de Batch a Capturar: " for w_batch
    on key (f3)
      let int_flag = true
    end prompt
    if int_flag then
      let int_flag = false
      close window v3
      return 0
    end if

    (==== VALIDA SI YA ESTA CERRADO EL BATCH ====)
    if esta_cerrado() then
      error
    end if
  end while
end function

```

```

"          EL NUMERO DE BATCH QUE TECLEO YA FUE CAPTURADO
"
    continue while
end if

{==== VERIFICA SI YA SE HABIA CAPTURADO ALMENOS UN REGISTRO DEL =====}
{==== BATCH, PARA YA NO PEDIR EL TIPO DE BATCH =====}
if w_tipobatch not matches "[L|F]" then
  {===== SOLICITA EL TIPO DE BATCH, LOCAL O FORANEO =====}
  while true
    prompt " Tipo de Batch <L>ocal <F>oraneo: " for char w_tipobatch
    on key (f3)
      let int_flag = true
    end prompt
    if int_flag then
      let int_flag = false
      close window v3
      return 0
    end if
    if w_tipobatch not matches "[L|l|F|f]" then
      error
" *** Error *** Teclee <F> si el batch es FORANEO; <L> si es LOCAL "
      continue while
    end if
    exit while
  end while
end if

if w_tipobatch matches "[l|L]" then
  let w_tipobatch = "L"
  let w_tit = "LOCAL"
else
  let w_tipobatch = "F"
  let w_tit = "FORANEO"
end if
display " El Numero que tecleo fue : ",w_batch at 2,1
display " y el tipo de bach es :      ",w_tit at 3,1
prompt " Es Correcto S/N ? " for char x
  on key (f3)
    let int_flag = true
  end prompt
  if int_flag then
    let int_flag = false
    close window v3
    return 0
  end if
  if x matches "[s|S]" then
    let w_sw1 = 1      (INDICA QUE ES EL PRIMER REGISTRO A CAPTURAR DE ESE
                      BATCH. ESTO EL PARA PODER GRABAR EN LA TABLA DE
                      CONTROL DE BATCH, SOLO LA PRIMERA VEZ      )
  end if
  exit while
end if
display "" at 2,1
display "" at 3,1
end while
close window v3
display "-----" at 5,54
return 1
end function

```

```

*****
FUNCION:      CIERRA_BATCH
OBJETIVO:    MARCAR COMO TERMINADA LA CAPTURA LOS REGISTRO DE TODO UN BATCH
PARAMETROS:  NINGUNO
*****
function cierra_batch()
  define w_liqui,
         t_liqui      integer,
         w_cont_liq,
         w_cont_emb   smallint,
         w_t           datetime hour to second

  declare apu cursor for
    select n_liqui from fclbv09 where n_batch = w_batch
    order by n_liqui

  let w_cont_liq = 0
  let w_cont_emb = 0
  let t_liqui = 99999999

  (==== CONTABILIZA EL NUMERO DE REGISTROS CAPTURADOS ====)
  foreach apu into w_liqui
    if w_liqui <> t_liqui then          (CONTROLA EL CORTE POR LIQUIDACION)
      let t_liqui = w_liqui
      let w_cont_liq = w_cont_liq + 1 (CONTADOR DE NUMERO DE LIQUIDACIONES)
    end if
    let w_cont_emb = w_cont_emb + 1    (CONTADOR DE NUMERO DE EMBARQUES)
  end foreach

  open window v4 at 12,20 with 7 rows, 40 columns
  attribute(border, prompt line last - 1)
  display " Numero de Batch: " ,w_batch at 2,1
  display " Liquidaciones Capturadas: " ,w_cont_liq at 3,1
  display " Embarques Capturados: " ,w_cont_emb at 4,1

  if w_cont_liq = 0 then
    prompt " Oprima cualquier tecla para continuar " for char x
    close window v4
    return 0
  end if

  prompt " Es Correcto S/N ? " for char x

  if x matches "{SS}" then

    (== ACTUALIZA EL STATUS DE LOS MOVS, COMO BATCH TERMINADO DE CAPTURAR ==)
    update fclbv09 set c_status = "2"
    where n_batch = w_batch

    (== ACTUALIZA EL REG DE CONTROL DEL BATCH ==)
    (let w_t = w_timef - w_timei TIEMPO TRANSCURRIDO DE CAPTURA)
    update fclbv12 set c_status = "2",
                     n_tiempo = w_t,
                     n_numliqui = w_cont_liq,
                     n_numembarq = w_cont_emb
    where n_batch = w_batch and
          c_user = w_user and
          n_ident = w_ident and
          f_batch = today

    display "" at 2,1
    display "" at 3,1
  end if
end function

```

```

        display "" at 4,1
        display " Batch ",w_batch," CERRADO " at 3,1
        prompt " Oprima cualquier tecla para continuar " for char x
        close window v4
        return 1
    end if
    close window v4
    return 0
end function

(*****
FUNCION:    esta_cerrado()
OBJETIVO:  VERIFICAR SI EL NUMERO DE BATCH QUE TECLEO, ESTA CERRADO, ES DECIR
           QUE YA FUE CAPTURADO Y TERMINADO
PARAMETROS: NINGUNO
*****)
function esta_cerrado()
    define w_status,
           w_tipo      char(1)
    let w_tipobatch = " "
    declare apu2 scroll cursor for
        select c_tipobatch, c_status from fclbv09 where n_batch = w_batch
    open apu2
    fetch first apu2 into w_tipo,w_status
    if status = notfound then
        close apu2
        return 0    (NO HAY NINGUNO, ESTO QUIERE DECIR QUE NO ESTA CERRADO)
    end if
    if w_status = "2" or w_status = "3" or w_status = "4" then
        close apu2
        return 1    (EL BATCH YA ESTA CERRADO)
    end if

    (===== AQUI SE OBTIENE EL TIPO DE BATCH, LOCAL O FORANEO, DE LOS =====)
    (===== REGISTROS QUE PREVIAMENTE FUERON CAPTURADOS PARA EL BATCH =====)
    let w_tipobatch = w_tipo

    return 0
end function

(*****
FUNCION:    control_batch()
OBJETIVO:  GRABAR UN REGISTRO DE CONTROL DE CADA UNO DE LOS BATCH'S QUE SE
           CAPTURAN
           NOTA: ESTO SE REALIZA AL CAPTURAR EL PRIMER REGISTRO DE CADA BATCH
           Y CUANDO SE TERMINA
PARAMETROS: NINGUNO
*****)
function control_batch()
    select * into w_conbat.* from fclbv12
        where n_batch = w_movi.n_batch and
              c_user   = w_user       and
              n_ident  = w_ident      and
              f_batch  = today
    if status = notfound then
        insert into fclbv12 values (w_user,w_ident,w_movi.n_batch,today,
                                   "00:00:00","1",0,0)
    end if
    return
end function

```

```

*****
FUNCION:   display_ultimo()
OBJETIVO:  DESPLEGAR EL ULTIMO REGISTRO LEIDO DE UN BATCH QUE AUN NO SE HA
           TERMINADO DE CAPTURAR
PARAMETROS: NINGUNO
*****}
function display_ultimo()
  define      w_anio      smallint
  clear form
  declare xx scroll cursor for
            select * from fclbv09 where n_batch = w_batch
  open xx
  fetch last xx into w_movi.*
  if status = notfound then
    error
  lear
  □

```



```

globals
  "clgfa08.4gl"                (INCLUYE LA DEFINICION DE VARIABLES GLOBALES)

(*****
FUNCION:      REPORTES
OBJETIVO:     CONSTRUIR UN MENU DE OPCIONES PARA GENERAR LOS REPORTES DEL
              CATALOGO DE PRODUCTOS
PARAMETROS:   NINGUNO
AUTOR:       RODOLFO ISAAC CARREOLA RIVAS
QUIEN LA LLAMA: FCLFA08.4GL, FCLFR08.4GL
*****)
function reportes()
(===== CONSTRUYE MENU PARA REPORTES =====)
  menu "REPORTES"
    command key ("C") "por Clave"
      "Genera reporte clasificado por clave de Producto"
      help 1451
      let w rep = 1
      call menu impresion()          (SOLICITA DISPOSITIVO DE SALIDA)
    command key ("N") "por Nombre"
      "Genera reporte clasificado por nombre del Producto"
      help 1452
      let w rep = 2
      call menu impresion()          (SOLICITA DISPOSITIVO DE SALIDA)
    command "Selectivo"
      "Genera reporte de acuerdo al criterio de seleccion deseado"
      help 1453
      let w rep = 3
      call menu impresion()          (SOLICITA DISPOSITIVO DE SALIDA)
    command "Imprimir"
      "Envia a la impresora un archivo de reportes generado previamente"
      help 1454
      run "lp /home/liquida/reportes/fcldr08"
    command "Ver"
      "Despliega en pantalla reporte previamente generado"
      help 1455
      run " echo '\033[73h' "
      run "pg -42 /home/liquida/reportes/fcldr08"
      run " echo '\033[73l' "
    command key("Y") "aYuda"
      "Muestra ayuda para construir criterio de seleccion"
      call showhelp(9999)
    command key(f3)
      close form forma01
    exit menu
  end menu
(=====)
end function

(*****
FUNCION:      MENU IMPRESION
OBJETIVO:     INICIALIZAR EL REPORTE YA SEA EN PANTALLA, IMPRESORA O UN ARCHIVO
              EN DISCO
PARAMETROS:   NINGUNO
*****)
function menu_impresion()
  define b smallint
  let x="*"          (SE DA UN VALOR INICIAL A "X" PARA QUE ENTRE AL WHILE)
  prompt "Desea el reporte por <P>antalla, <I>mpresora o <A>rchivo " for char
  x

```

```

    on key (f3)
      let int_flag = true
    end prompt
    if int_flag then
      let int_flag = false
      return
    end if

    let b = 0
    case
      case
        (VALIDA LA OPCION QUE
        TECLEARON)
        when x matches "[pP]"

          start report repl to "/home/liquida/reportes/fcldr08" (INICIALIZA EL
          REPORTE POR PANTALLA)
          let b = 1
          call lee_registros(b) (LEE REGS Y GENERA REPORTE)

        when x matches "[iI]"

          start report repl to printer (INICIALIZA EL REPORTE POR IMPRESORA)
          call lee_registros(b) (LEE REGS Y GENERA REPORTE)

        when x matches "[aA]"

          start report repl to "/home/liquida/reportes/fcldr08" (INICIALIZA EL
          REPORTE POR ARCHIVO)
          call lee_registros(b) (LEE REGS Y GENERA REPORTE)

        otherwise
          error " " , " OPCION INVALIDA "
          sleep 1
        end case
    end case
  end function

```

```

(*****
FUNCION: LEE REGISTROS
OBJETIVO: CONSTRUIR DINAMICAMENTE LA INSTRUCCION "SELECT" QUE REALICE LA
LECTURA DE REGISTROS DE LA TABLA DEPENDIENDO DEL CRITERIO ELEGIDO
PARAMETROS: NINGUNO
*****)
function lee_registros(b)

```

```

  define w_preact,
    w_preact like fclbv07.n_impres,
    w_fecact,
    w_fecant like fclbv07.f_finprec,
    w_total smallint,
    b smallint

```

```

message "Espere un momento... Reporte Generandose"
case
  when w_rep = 1 (POR CLAVE DE PRODUCTO)
    let w_total=0
    let w_select="select * from fclbg02 ",
      "where c_status=1 order by c_prod"
  when w_rep = 2 (POR NOMBRE DEL PRODUCTO)
    let w_total=0
    let w_select="select * from fclbg02 ",
      "where c_status=1 order by d_prod"

```

```

when w_rep = 3                                     (POR CRITERIO DE SELECCION)
  open form forma01 from "fclfa08"
  display form forma01
  let w_fecha = today
  display w_idpro at 3,1
  display w_fecha using "dd/mm/yy" at 4,1
  display w_idtty at 4,72
  construct by name w where on fclbg02.*
  let w_select="select * from fclbg02 where ",
              w_where clipped
  let w_total=0
end case

(==== PREPARA LA INSTRUCCION select ARMADA EN w_select PARA QUE SE ===)
(==== PUEDA EJECUTAR ===)
prepare w_lee from w_select

(===== SE DECLARA UN APUNTADOR PARA LEER LOS REGISTRO DE LA TABLA =====)
declare apun2 cursor for w_lee

(===== LEE REGISTRO POR REGISTRO Y LOS ENVIA AL FORMATO DEL REPORTE =====)
foreach apun2 into w_produ.*
  let w_preact = 0
  let w_preant = 0
  let w_fecact = "010101"
  let w_fecant = "010101"
  if valida_pre("123199") then
    let w_preact = w_preci.n_impres
    let w_fecact = w_preci.f_iniprec
    if valida_pre(w_preci.f_iniprec - 1) then
      let w_preant = w_preci.n_impres
      let w_fecant = w_preci.f_iniprec
    end if
  end if
  output to report repl(w_produ.*,w_preact,w_preant,w_fecact,w_fecant)
end foreach

(===== TERMINA EL REPORTE =====)
finish report repl
if b = 1 then
  run 'echo "\033[?3h"'
  run "pg /home/liquida/reportes/fcldr08"
  run 'echo "\033[?3l"'
end if
end function
(*****
FUNCION:      REPl
OBJETIVO:    FORMATEO DE LA SALIDA DEL REPORTE
PARAMETROS:  REGISTRO LEIDO DE PRODUCTOS (w_preci,w_preact,w_preant,w_fecact,
          w_fecant)
*****}
report repl(r,w_preact,w_preant,w_fecact,w_fecant)
define r record like fclbg02.*,          (ESTRUCTURA QUE CONTIENE EL REGIS)
          (TRO ENVIADO POR PARAMETRO    )

  w_preact,
  w_preant          like fclbv07.n_impres,
  w_fecact,
  w_fecant          date,
  w_total,
  w_cont            smallint

```

```

output
  left margin 0
  right margin 132
format
  page header
    print "ID: FCLFR08",
      column 47, "COORDINACION EJECUTIVA DE DISTRIBUCION",
      column 119, "Pag:",pageno using "###"
    print "P E M E X ",
      column 47, "GERENCIA DE DISTRIBUCION Y TRANSPORTES",
      column 119, "SUBDIR. COM."
  skip 1 lines
  if w_rep = 2 then
    print column 42, "CATALOGO DE PRODUCTOS CLASIFICADO ALFABETICAMENTE",
      column 119, "FEC:",today using "dd/mm/yy"
    print "-----",
      column 50, "-----",
      column 100, "-----"
  else
    print column 45, "CATALOGO DE PRODUCTOS CLASIFICADO POR CLAVE",
      column 119, "FEC:",today using "dd/mm/yy"
    print "-----",
      column 50, "-----",
      column 100, "-----"
  end if
  print column 1, " CLAVE ", column 8, " GPO ",
    column 13, " P R O D U C T O ",
    column 36, " PESO ",
    column 44, " COEFICIENTE ",
    column 56, " -- P R E C I O Y V I G E N C I A -- ",
    column 93, " UNIDAD ",
    column 97, " FACTOR F.",
    column 103, " CAPACIDAD ",
    column 108, " TIPO"
  print column 1, " PROD.",column 34, " ESPECIF.",
    column 45, " DILATAION ",
    column 56, " -- A C T U A L-- -- A N T E R I O R--",
    column 128, "CUOTA"
  print "-----",
    column 50, "-----",
    column 100, "-----"
  skip 1 lines
  let w_cont = 0
  on every row
  let w_total= w_total + 1
  if (w_preact != 0) then
    if (w_preant != 0) then
      print r.c_prod, 3 spaces, r.c_grupo,3 spaces,r.d_prod clipped,
        column 35, r.n_pesp, column 45,r.n_coefe,
        column 55,w_preact,2 spaces,w_fecact using "dd/mm/yy" ,
          w_preant,
        2 spaces,w_fecant using "dd/mm/yy",
        7 spaces, r.n_unid ;
      if (r.n_unid ="3") then
        print " Kgs";
      else
        print " Lts";
      end if
      print 3 spaces,r.n_factf using "#.#####",
        r.n_capap,r.c_cuota
    end if
  end if

```

```

else
  print r.c_prod,3 spaces, r.c_grupo,3 spaces,r.d_prod clipped,
  column 35, r.n_pesp, column 45,r.n_coefe,
  column 55,w_preact,2 spaces,w_fecact using "dd/mm/yy",
  w_preant,
  17 spaces, r.n_unid;
  if (r.n_unid="3") then
    print " Kgs";
  else
    print " Lts";
  end if
  print 3 spaces,r.n_factf using "#.#####",
  r.n_capap,
  r.c_cuota
end if
else
  print r.c_prod,3 spaces, r.c_grupo,3 spaces,r.d_prod clipped,
  column 35, r.n_pesp, column 45,r.n_coefe,
  column 55,w_preact,column 74,w_preant,12 spaces,
  5 spaces, r.n_unid;
  if (r.n_unid="3") then
    print " Kgs";
  else
    print " Lts";
  end if
  print 3 spaces,r.n_factf using "#.#####",
  r.n_capap, r.c_cuota
end if
let w_cont = w_cont + 1      (OJO... ESTO ES PARA CONTROLAR)
if (w_cont mod 5) = false then (PCR PANTALLA, UNA PAUSA ENTRE)
  pause (CADA CINCO LINEA DE IMPRESION)
end if
on last row
skip 5 lines
print column 3,"TOTAL DE REGISTROS IMPRESOS: ",w_total using "###"
let w_total=0
end report
□

```

```

*****
PROGRAMA: FCLFR56.4gl
OBJETIVO: GENERA EL REPORTE DE CEDULA DE PAGO POR CONCEPTO DE FLETES
          REALIZADOS POR LAS CIAS TRANSPORTISTAS
PARAMETROS: No. DE CEDULA, TIPO DE CEDULA (EXTERNOS)
AUTOR:     Ma. DOLORES PE&ALOZA B
FECHA:     13-marzo-93
*****
database fletes                                     (ABRE LA BASE DE DATOS DE FLETES )

globals

define w_reg record like fclbs08.*,
      w_linea smallint,
      w_bonif record like fclbv01.*,
      w_pagos record like fclbs03.*,
      w_concep record like fclbg06.*,
      w_select char(200),
      w_cedula integer,
      w_tipo char(1),
      w_mes, w_feprogra date,
      w_param record like fclbs09.*,
      w_elabo like fclbg13.d_nombre,
      w_revis like fclbg13.d_nombre,
      w_autor like fclbg13.d_nombre,
      w_razon like fclbg10.d_razon,
      w_locfc char(10),
      w_descrip char(70),
      w_desmes char(10),
      w_bondes record
          c_portea like fclbv01.c_portea,
          d_razon like fclbg10.d_razon,
          c_concep like fclbv01.c_concep,
          d_concep like fclbg06.d_concep,
          n_dccpa like fclbv01.n_dccpa,
          f_aplic like fclbv01.f_aplic,
          n_cedula like fclbv01.n_cedula,
          c_tipocedu like fclbv01.c_tipocedu,
          n_impbon like fclbv01.n_i_aplic,
          n_impdes like fclbv01.n_i_aplic,
          c_orides like fclbs08.c_orides
      end record
end globals

main
define w_reporte char(40)
let w_cedula = arg_val(1)
let w_tipo = arg_val(2)
if w_tipo = "X" then
    let w_select =" select * from fclbs08 where n_cedula =",
                w_cedula using "#####",
                " order by c_tipocedu,c_orides,c_portea"
else
    let w_select =" select * from fclbs08 where n_cedula =",
                w_cedula using "#####",
                " and c_tipocedu = '",w_tipo,'" ,
                " order by c_orides,c_portea"
end if
select * into w_param.* from fclbs09
      where c_reg = 1

```

```

select d_nombre into w_elabo from fclbg13
  where c_user = w_param.c_useela and
        n_ident = w_param.n_idelab
select d_nombre into w_revis from fclbg13
  where c_user = w_param.c_userev and
        n_ident = w_param.n_idrevi
select d_nombre into w_autor from fclbg13
  where c_user = w_param.c_useaut and
        n_ident = w_param.n_idauto

prepare w_lee from w_select

declare apun cursor for w_lee
let w_reporte = "/home/liquida/reportes/ced",w_cedula using "###",w_tipo,
  ".",today using "ddmmyy"
start report rep1 to w_reporte
let w_reporte = "/home/liquida/reportes/bon",w_cedula using "###",w_tipo,
  ".",today using "ddmmyy"
start report rep2 to w_reporte

foreach apun into w_reg.*
  select d_razon into w_razon from fclbg10
    where c_portea = w_reg.c_portea
  if status = notfound then
    let w_razon = " SIN DESCRIPCION "
  end if
  output to report rep1(w_reg.*,w_razon,w_elabo,w_revis,w_autor)

  if (w_reg.n_imponif > 0) or (w_reg.n_impdesc > 0) then
    declare apun2 cursor for
      select * into w_bonif.* from fclbv01
        where n_cedula = w_reg.n_cedula and
              c_tipocedu = w_reg.c_tipocedu and
              c_portea = w_reg.c_portea
    foreach apun2 into w_bonif.*
      select * into w_concep.* from fclbg06
        where c_concep = w_bonif.c_concep

      let w_bondes.c_portea = w_reg.c_portea
      let w_bondes.d_razon = w_razon
      let w_bondes.c_concep = w_bonif.c_concep
      let w_bondes.d_concep = w_concep.d_concep
      let w_bondes.n_docpa = w_bonif.n_docpa
      let w_bondes.f_aplic = w_bonif.f_aplic
      let w_bondes.n_cedula = w_bonif.n_cedula
      let w_bondes.c_orides = w_reg.c_orides

      if w_concep.n_tipbon = "B" then
        let w_bondes.c_tipocedu = w_bonif.c_tipocedu
        let w_bondes.n_impon = w_bonif.n_i_aplic
        let w_bondes.n_impdes = 0
      else
        select * into w_pagos.* from fclbs03
          where c_portea = w_bonif.c_portea and
                n_docpa = w_bonif.n_docpa and
                n_cedula = w_bonif.n_cedula and
                c_tipocedu = w_bonif.c_tipocedu
        let w_bondes.c_tipocedu = w_pagos.c_tipocedu
        let w_bondes.n_impon = 0
        let w_bondes.n_impdes = w_pagos.n_i_aplic
    end foreach
  end if
end foreach

```

```

        end if
        output to report rep2(w_bondes.*)
    end foreach
end if
end foreach
finish report rep1
finish report rep2
end main
{*****}
FUNCION:      REP1
OBJETIVO:    FORMATEO DE LA SALIDA
PARAMETROS:  REGISTRO LEIDO DEL ARCHIVO DE CEDULAS DE PAGO
{*****}
report rep1 (r,w_razon,w_elabo,w_revis,w_autor)
define r record like fclbs08.*,
w_razon like fclbg10.d_razon,
w_cenpag record like fclbg09.*,
w_param record like fclbs09.*,
w_elabo like fclbg13.d_nombre,
w_revis like fclbg13.d_nombre,
w_autor like fclbg13.d_nombre,
w_locfo char(10),
w_descrip char(70),
w_desmes char(10),
w_mes, w_feprogra date,
w_pag,
w_lineas smallint,
w_flete,
w_tot_flete ,
w_tot_pece,
w_tot_iva,
w_tot_neto,
w_tot_peaje,
w_neto decimal (10,0)

output
left margin 0
right margin 132
format
first page header
let w_lineas = 0
select * into w_cenpag.* from fclbg09
where c_orides = r.c_orides
if status = notfound then
let w_cenpag.d_oridel = " SIN DESCRIPCION "
end if
let w_pag = 1
case
when r.c_tipocedu = "A"
let w_locfo = "LOCAL"
let w_descrip = "
UNICAMENTE GAS"
when r.c_tipocedu = "B"
let w_locfo = "LOCAL"
let w_descrip = "
UNICAMENTE PETROQUIMICOS"
when r.c_tipocedu = "C"
let w_locfo = "LOCAL"
let w_descrip = "
UNICAMENTE TURBOSINA"
when r.c_tipocedu = "D"
let w_locfo = "LOCAL"
let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR
GAS,TURBOSINA Y PETROQUIMICOS )"

```



```

when r.c_tipocedu = "E"
  let w_locfo = "FORANEO"
  let w_descrip = "
                                UNICAMENTE GAS"
when r.c_tipocedu = "F"
  let w_locfo = "FORANEO"
  let w_descrip = "
                                UNICAMENTE PETROQUIMICOS"
when r.c_tipocedu = "G"
  let w_locfo = "FORANEO"
  let w_descrip = "
                                UNICAMENTE TURBOSINA"
when r.c_tipocedu = "H"
  let w_locfo = "FORANEO"
  let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR
(GAS, TURBOSINA Y PETROQUIMICOS )"
when r.c_tipocedu = "I"
  let w_locfo = "GUADALAJARA"
  let w_descrip = "GUADALAJARA "
end case
let w_mes = r.f_progra
case
  when MONTH(w_mes) = 01
    let w_desmes = "ENERO"
  when MONTH(w_mes) = 02
    let w_desmes = "FEBRERO"
  when MONTH(w_mes) = 03
    let w_desmes = "MARZO"
  when MONTH(w_mes) = 04
    let w_desmes = "ABRIL"
  when MONTH(w_mes) = 05
    let w_desmes = "MAYO"
  when MONTH(w_mes) = 06
    let w_desmes = "JUNIO"
  when MONTH(w_mes) = 07
    let w_desmes = "JULIO"
  when MONTH(w_mes) = 08
    let w_desmes = "AGOSTO"
  when MONTH(w_mes) = 09
    let w_desmes = "SEPTIEMBRE"
  when MONTH(w_mes) = 10
    let w_desmes = "OCTUBRE"
  when MONTH(w_mes) = 11
    let w_desmes = "NOVIEMBRE"
  when MONTH(w_mes) = 12
    let w_desmes = "DICIEMBRE"
end case
print "ID: FCLFR53",
      column 47, "COORDINACION EJECUTIVA DE DISTRIBUCION",
      column 119, "Pag:", w_pag using "###"
print "P E M E X",
      column 47, "GERENCIA DE DISTRIBUCION Y TRANSPORTES",
      column 109, "CEDULA No:", r.n_cedula using "&&&&" ,
      1 space, w_locfo clipped
skip 2 lines
print column 18, "RELACION DE PAGOS POR CONCEPTO DE FLETES ",
      column 59, "POR AUTOTANQUE DE EMBARQUES PROCESADOS AL:",
      today using "dd/mm/yy"
print column 35, w_descrip clipped
skip 1 lines
print column 7, "FECHA PROGRAMADA PARA PAGO: ",
      day(w_mes) using "##", " DE ",
      w_desmes clipped, " DE ", year(w_mes) using "####"

```

```

        skip 1 lines
print column 7, "AGENCIA PAGADORA: ", r.c_orides, 1 spaces,
        w_cenpag.d_oridel clipped,
        column 80, "CVE. AUTORIZACION (CIF)", 1 spaces,
        r.c_autofina
        skip 1 lines
print column 5, "---- DATOS DEL PORTEADOR ----",
        column 66, "IMPTE FLETES",
        column 84, "IMPTE I.V.A.",
        column 103, "IMPTE NETO",
        column 116, "No.CTR/RECIBO "
print column 2, "CVE.      R A Z O N      S O C I A L ",
        column 68, "P.E.C.E",
        column 84, "DIF. CASETAS"
        skip 2 lines

page header
let w_lineas = 0
select * into w_cenpag.* from fclbg09
        where c_orides = r.c_orides
if status = notfound then
        let w_cenpag.d_oridel = " SIN DESCRIPCION "
end if
let w_pag = w_pag + 1
case
        when r.c_tipocedu = "A"
                let w_locfo = "LOCAL"
                let w_descrip = "                                UNICAMENTE GAS"
        when r.c_tipocedu = "B"
                let w_locfo = "LOCAL"
                let w_descrip = "                                UNICAMENTE PETROQUIMICOS"
        when r.c_tipocedu = "C"
                let w_locfo = "LOCAL"
                let w_descrip = "                                UNICAMENTE TUREOSINA"
        when r.c_tipocedu = "D"
                let w_locfo = "LOCAL"
                let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR
(GAS, TURBOSINA Y PETROQUIMICOS )"
        when r.c_tipocedu = "E"
                let w_locfo = "FORANEO"
                let w_descrip = "                                UNICAMENTE GAS"
        when r.c_tipocedu = "F"
                let w_locfo = "FORANEO"
                let w_descrip = "                                UNICAMENTE PETROQUIMICOS"
        when r.c_tipocedu = "G"
                let w_locfo = "FORANEO"
                let w_descrip = "                                UNICAMENTE TURBOSINA"
        when r.c_tipocedu = "H"
                let w_locfo = "FORANEO"
                let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR
(GAS, TURBOSINA Y PETROQUIMICOS )"
        when r.c_tipocedu = "I"
                let w_locfo = "GUADALAJARA"
                let w_descrip = "GUADALAJARA"
end case
let w_mes = r.f_progra
case
        when MONTH(w_mes) = 01
                let w_desmes = "ENERO"
        when MONTH(w_mes) = 02

```

```

        let w_desmes = "FEBRERO"
    when MONTH(w_mes) = 03
        let w_desmes = "MARZO"
    when MONTH(w_mes) = 04
        let w_desmes = "ABRIL"
    when MONTH(w_mes) = 05
        let w_desmes = "MAYO"
    when MONTH(w_mes) = 06
        let w_desmes = "JUNIO"
    when MONTH(w_mes) = 07
        let w_desmes = "JULIO"
    when MONTH(w_mes) = 08
        let w_desmes = "AGOSTO"
    when MONTH(w_mes) = 09
        let w_desmes = "SEPTIEMBRE"
    when MONTH(w_mes) = 10
        let w_desmes = "OCTUBRE"
    when MONTH(w_mes) = 11
        let w_desmes = "NOVIEMBRE"
    when MONTH(w_mes) = 12
        let w_desmes = "DICIEMBRE"
    end case
print "ID: FCLFR53",
    column 47, "COORDINACION EJECUTIVA DE DISTRIBUCION",
    column 119, "Pag:",w_pag using "###"
print "P E M E X",
    column 47, "GERENCIA DE DISTRIBUCION Y TRANSPORTES",
    column 109, "CEDULA No:",r.n_cedula using "####" ,
    1 space ,w_locfo clipped
skip 2 lines
print column 18, "RELACION DE PAGOS POR CONCEPTO DE FLETES ",
    column 59, "POR AUTOTANQUE DE EMBARQUES PROCESADOS AL ",
    today using "dd/mm/yy"
print column 35, w_descrip clipped
skip 1 lines
print column 7, "FECHA PROGRAMADA PARA PAGO: ",
    day(w_mes) using "##", " DE ",
    w_desmes clipped, " DE ", year(w_mes) using "####"
skip 1 lines
print column 7, "AGENCIA PAGADORA: ",r.c_orides, 1 spaces,
    w_cenpag.d_oridel clipped,
    column 80, "CVE. AUTORIZACION (CIF)", 1 spaces,
    r.c_autofina
skip 1 lines
print column 5, "---- DATOS DEL PORTEADOR ----",
    column 66, "IMPTE FLETES",
    column 84, "IMPTE I.V.A.",
    column 103, "IMPTE NETO",
    column 116, "No.CTR/RECIBO "
print column 2, "CVE.      R A Z O N      S O C I A L ",
    column 68, "P.E.C.E",
    column 84, "DIF. CASETAS"
skip 2 lines

page trailer
print column 20, "ELABORO",
    column 60, "REVISO",
    column 92, "AUTORIZO"
skip 3 lines
print column 10,"-----", 9 spaces,

```

```

"-----", 4 spaces,
"-----"
print column 13, w_elabo,
      column 50, w_revis,
      column 84, w_autor
skip 1 lines
print "*** CON DESCUENTO Y/O BONIFICACION APLICADO.",
      column 44, " FAVOR VER ANEXO"

before group of r.c_tipocedu
case
  when r.c_tipocedu = "A"
    let w_locfo = "LOCAL"
    let w_descrip = "
                                     UNICAMENTE GAS"
  when r.c_tipocedu = "B"
    let w_locfo = "LOCAL"
    let w_descrip = "
                                     UNICAMENTE PETROQUIMICOS"
  when r.c_tipocedu = "C"
    let w_locfo = "LOCAL"
    let w_descrip = "
                                     UNICAMENTE TURBOSINA"
  when r.c_tipocedu = "D"
    let w_locfo = "LOCAL"
    let w_descrip = "
(GAS, TURBOSINA Y PETROQUIMICOS )"
    let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR"
  when r.c_tipocedu = "E"
    let w_locfo = "FORANEO"
    let w_descrip = "
                                     UNICAMENTE GAS"
  when r.c_tipocedu = "F"
    let w_locfo = "FORANEO"
    let w_descrip = "
                                     UNICAMENTE PETROQUIMICOS"
  when r.c_tipocedu = "G"
    let w_locfo = "FORANEO"
    let w_descrip = "
                                     UNICAMENTE TURBOSINA"
  when r.c_tipocedu = "H"
    let w_locfo = "FORANEO"
    let w_descrip = "
(GAS, TURBOSINA Y PETROQUIMICOS )"
    let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR"
  when r.c_tipocedu = "I"
    let w_locfo = "GUADALAJARA"
    let w_descrip = "GUADALAJARA"
end case
skip to top of page
before group of r.c_orides
  let w_tot_flete = 0
  let w_tot_pece = 0
  let w_tot_iva = 0
  let w_tot_neto = 0
  let w_tot_peaje = 0
skip to top of page
on every row
  let w_lineas = w_lineas + 1
  let w_flete = (r.n_impflete - r.n_impxfalt)
  let w_neto = (r.n_impflete + r.n_impiva + r.n_impeaje
               + r.n_imponif) - (r.n_impxfalt + r.n_impece
               + r.n_impdesc)
  let w_tot_flete = (w_tot_flete + w_flete)
  let w_tot_pece = (w_tot_pece + r.n_impece)
  let w_tot_iva = (w_tot_iva + r.n_impiva)
  let w_tot_neto = (w_tot_neto + w_neto)
  let w_tot_peaje = (w_tot_peaje + r.n_impeaje)

```

```

print column 66, w_flete using "###,###,###",
      column 84, r.n_impiva using "###,###,###"
if r.n_impbonif > 0 or r.n_impdesc > 0 then
  print column 1, "+*",
      column 2, r.c_portea using "####",1 spaces,w_razon,
      column 48, "-->",
      column 66, r.n_impece using "###,###,###",
      column 84, r.n_impeaje using "###,###,###",
      column 101, w_neto using "###,###,###",
      column 116, "MX-", r.n_contra using "#####","/",
      (year(w_mes) - 1900) using "##"
else
  print
  print
  column 3, r.c_portea using "####",1 spaces,w_razon,
  column 48, "-->",
  column 66, r.n_impece using "###,###,###",
  column 84, r.n_impeaje using "###,###,###",
  column 101, w_neto using "###,###,###",
  column 116, "MX-", r.n_contra using "#####","/",
  (year(w_mes) - 1900) using "##"
end if
skip 1 lines
if w_lineas > 10 then
  skip to top of page
end if
after group of r.c_orides
  skip 2 lines
  print column 31, "TOTAL DEL CENTRO -->",
      column 64, w_tot_flete using "#,###,###,###",
      column 84, w_tot_iva using "###,###,###",
      column 99, w_tot_neto using "#,###,###,###"
  print column 64, w_tot_pece using "#,###,###,###",
      column 84, w_tot_peaje using "###,###,###"
end report

(*****
FUNCION: REP2
OBJETIVO: FORMATEO DE LA SALIDA
PARAMETROS: REGISTRO LEIDO DEL ARCHIVO DE PAGOS APLICADOS
*****)
report rep2 (r1)
  define r1 record
    c_portea like fclbv01.c_portea,
    d_razon like fclbg10.d_razon,
    c_concep like fclbv01.c_concep,
    d_concep like fclbg06.d_concep,
    n_docpa like fclbv01.n_docpa,
    f_aplic like fclbv01.f_aplic,
    n_cedula like fclbv01.n_cedula,
    c_tipocedu like fclbv01.c_tipocedu,
    n_impbon like fclbv01.n_i_aplic,
    n_impdes like fclbv01.n_i_aplic,
    c_orides like fclbs08.c_orides
  end record,
  w_cenpag record like fclbg09.*,
  w_locfo char(10),
  w_descrip char(70),
  w_tot_bonif ,

```

```

w_tot_desc decimal (10,0)

output
left margin 0
right margin 132
format
page header
select * into w_cenpag.* from fclbg09
where c_orides = w_bondes.c_orides
if status = notfound then
let w_cenpag.d_oridel = " SIN DESCRIPCION "
end if

before group of rl.c_tipocedu
case
when rl.c_tipocedu = "A"
let w_locfo = "LOCAL"
let w_descrip = "
UNICAMENTE GAS"
when rl.c_tipocedu = "B"
let w_locfo = "LOCAL"
let w_descrip = "
UNICAMENTE PETROQUIMICOS"
when rl.c_tipocedu = "C"
let w_locfo = "LOCAL"
let w_descrip = "
UNICAMENTE TURBOSINA"
when rl.c_tipocedu = "D"
let w_locfo = "LOCAL"
let w_descrip = "TODOS LOS PROD. SIN CCNSIDERAR
(GAS,TURBOSINA Y PETROQUIMICOS )"
when rl.c_tipocedu = "E"
let w_locfo = "FORANEO"
let w_descrip = "
UNICAMENTE GAS"
when rl.c_tipocedu = "F"
let w_locfo = "FORANEO"
let w_descrip = "
UNICAMENTE PETROQUIMICOS"
when rl.c_tipocedu = "G"
let w_locfo = "FORANEO"
let w_descrip = "
UNICAMENTE TURBOSINA"
when rl.c_tipocedu = "H"
let w_locfo = "FORANEO"
let w_descrip = "TODOS LOS PROD. SIN CONSIDERAR
(GAS,TURBOSINA Y PETROQUIMICOS )"
when rl.c_tipocedu = "I"
let w_locfo = "GUADALAJARA"
let w_descrip = "GUADALAJARA"
end case
skip to top of page

print "ID: FCLFR56",
column 47, "COORDINACION EJECUTIVA DE DISTRIBUCION",
column 119, "Pag:", pageno using "###"
print "P E M E X",
column 47, "GERENCIA DE DISTRIBUCION Y TRANSPORTES",
column 109, "CEDULA No:",rl.n_cedula using "###" ,
1 space, w_locfo clipped
skip 2 lines
print column 20,"RELACION DE PORTEADORES CON DESCUENTOS Y/O ",
column 44,"BONIFICACIONES APLICADOS AL: ",rl.f_aplic
using "dd/mm/yy"
print column 35, w_descrip clipped
skip 1 lines

```

```

print "AGENCIA PAGADORA: ", r1.c_orides using "####", 1 spaces,
      w_cenpag.d_oridēl clipped
      skip 1 lines
print column 5, "<---- DATOS DEL PORTEADOR ---->",
      column 64, "CVE. CONCEPTO",
      column 91, "OFICIO",
      column 108, "BONIFICACION", 3 spaces, "DESCUENTOS"
print column 2, "CVE.      R A Z O N      S O C I A L",
      column 65, "DESCRIPCION",
      column 91, "NUMERO"
before group of r1.c_portea
  let w_tot_bonif= 0
  let w_tot_desc= 0
  skip 1 lines

on every row
if r1.n_impbon > 0 then
  let w_tot_bonif = w_tot_bonif + r1.n_impbon

  print r1.c_portea using "####", 1 spaces,
        r1.d_razon clipped,
        column 64, r1.c_concep using "&&",
        1 spaces, r1.d_concep clipped,
        column 91, r1.n_docpa,
        column 108, r1.n_impbon using "###,###,###",
        column 121, r1.n_impdes using "###,###,##0"
else
  let w_tot_desc = w_tot_desc + r1.n_impdes
  print r1.c_portea using "####", 1 spaces,
        r1.d_razon clipped,
        column 57, r1.c_concep using "&&",
        1 spaces, r1.d_concep clipped,
        column 85, r1.n_docpa,
        column 108, r1.n_impbon using "###,###,##0",
        column 121, r1.n_impdes using "###,###,###"
end if

after group of r1.c_portea
  skip 1 lines
  print column 40, "TOTAL PORTEADOR ---->",
        column 108, w_tot_bonif using "###,###,###",
        column 121, w_tot_desc using "###,###,###"

end report

```

```

globals
  "clgfa33.4gl"

{*****
FUNCION:   REPORTES
OBJETIVO:  CONSTRUIR UN MENU DE OPCIONES PARA GENERAR LOS REPORTES DEL
           CATALOGO DE PRODUCTOS COMUNMENTE TRANSPORTADOS POR RUTA
PARAMETROS: NINGUNO
AUTOR:    LAURA JACOME DEL MORAL
QUIEN LA LLAMA: FCLFA33.4GL, FCLFR33.4GL
*****
}

function reportes()
  (===== CONSTRUYE MENU PARA REPORTES =====)
  menu "REPORTES"
    command "Alfabetico"
      "Genera reporte clasificado en orden ALFABETICO de la ruta"
      help 10751
      let w_rep = 1
      call menu_impresion()          (SOLICITA DISPOSITIVO DE SALIDA)
    command "Selectivo"
      "Genera reporte de acuerdo al criterio de seleccion deseado"
      help 10752
      open form forma01 from "fclfa33"
      display w_idpro at 3,1
      display form forma01
      let w_fecha = today
      display w_fecha using "dd/mm/yy" at 4,1
      display w_idtty at 4,72
      let w_rep = 2
      construct by name w where on c_origen,c_destino,c_ruta
      call menu_impresion()          (SOLICITA DISPOSITIVO DE SALIDA)
      close form forma01
    command "Imprimir"
      "Envia a la impresora un archivo de reportes generado previamente"
      help 10753
      run "lp /home/liquida/reportes/fcldr33"
    command key(f3)
      exit menu
  end menu
  (=====)
end function

{*****
FUNCION:   MENU IMPRESION
OBJETIVO:  INICIALIZAR EL REPORTE YA SEA EN PANTALLA, IMPRESORA O UN ARCHIVO
           EN DISCO
PARAMETROS: NINGUNO
*****
}

function menu_impresion()
  define b _smallint

  let x="*"          (SE DA UN VALOR INICIAL A "X" PARA QUE ENTRE AL WHILE)
  prompt "Desea el reporte por <P>antalla, <I>mpresora o <A>rchivo " for char x
  on key (f3)
    let int_flag = true
  end prompt
  if int_flag then

```



```

        let int_flag = false
        return
    end if

    let b = 0

    case
        when x matches "[pP]"
            start report repl to "/home/liquida/reportes/fcldr33"
                (VALIDA LA OPCION QUETECLEARON)
                (INICIALIZA EL REPORTE POR PANTALLA)
            let b = 1
            call lee_registros(b)
                (LEE REGS Y GENERA REPORTE)

        when x matches "[iI]"
            start report repl to printer
                (INICIALIZA EL REPTE POR IMPRESORA)
            call lee_registros(b)
                (LEE REGS Y GENERA REPORTE)

        when x matches "[aA]"
            start report repl to "/home/liquida/reportes/fcldr33"
                (LEE REGS Y GENERA REPORTE)
            call lee_registros(b)
                (LEE REGS Y GENERA REPORTE)

        otherwise
            error "
                sleep 1
                ",x , " OPCION INVALIDA "
    end case
end function

{*****
FUNCION: LEE_REGISTROS
OBJETIVO: CONSTRUIR DINAMICAMENTE LA INSTRUCCION "SELECT" QUE REALICE LA
LECTURA DE REGISTROS DE LA TABLA DEPENDIENDO DEL CRITERIO ELEGIDO
PARAMETROS: NINGUNO
*****}
function lee_registros(b)

    define b smallint,
           w_origen,
           w_destino char(30)

    message "Espere un momento... Reporte Generandose"

    case

        when w_rep = 1
            let w_select="select * from fclbn02"
                (POR CLAVE DE ORIGEN/DEST/RUTA ALFA)

        when w_rep = 2
            let w_select="select * from fclbn02 where ",
                w where clipped,
                "order by c_origen,c_destino,c_ruta"
                (POR CRITERIO DE SELECCION)

    end case

    (==== PREPARA LA INSTRUCCION select ARMADA EN w_select PARA QUE SE ===)
    (==== PUEDA EJECUTAR ===)
    prepare w_lee from w_select

    (===== SE DECLARA UN APUNTADOR PARA LEER LOS REGISTRO DE LA TABLA =====)
    declare apun2 cursor for w_lee

    (===== LEE REGISTRO POR REGISTRO Y LOS ENVIA AL FORMATO DEL REPORTE =====)

```

```

foreach apun2 into w_prodrut.*
  select * into w_ruta.* from fclbv03
    where c_origen = w_prodrut.c_origen and
          c_destino= w_prodrut.c_destino and
          c_ruta   = w_prodrut.c_ruta and
          f_finruta= "12/31/99"

  select * into w_orides.* from fclbg09
    where c_orides= w_prodrut.c_origen
          let w_origen = w_orides.d_oridel

  select * into w_orides.* from fclbg09
    where c_orides= w_prodrut.c_destino
          let w_destino = w_orides.d_oridel

  output to report repl(w_prodrut.*,w_origen,w_destino)
end foreach

(===== TERMINA EL REPORTE =====)
finish report repl
if b = 1 then
  run 'echo "\033[?3h"'
  run "pg /home/liquida/reportes/fcldr33"
  run 'echo "\033[?3l"'
end if
end function

{*****
FUNCION:      REPl
OBJETIVO:    FORMATEO DE LA SALIDA DEL REPORTE
PARAMETROS:  REGISTRO LEIDO DE PRODUCTOS TRANSP.COMUNMENTE POR RUTAS (W_PRODRUT)
*****}

report repl(r,w_origen,w_destino)
  define r record like fclbn02.*,      {ESTRUCTURA QUE CONTIENE EL REGIS}
    w_origen,
    w_destino char(30),
    w_desc like fclbg02.d_prod,
    w_cont,
    w_cont_rut,
    w_cont_tot smallint
  output
    left margin 0
    right margin 132

  order by w_origen,w_destino

  format
    page header

    print "ID:FCLFR33",
      column 45, "COORDINACION EJECUTIVA DE DISTRIBUCION",
      column 110, "Pag. ", pageno using "###"
  print "P E M E X",
    column 45, "GERENCIA DE DISTRIBUCION DE TRANSPORTE",
    column 110, "SUBDIR. COM."
  skip 1 lines
  print column 45, "      CATALOGO DE RUTAS POR CLAVE",
    column 110, "FEC:", TODAY using "dd/mm/yy"
  skip 1 lines

```

```

        print "ORIGEN:",r.c_origen using "#####", 1 space, w_origen
        print "-----"
        let w_cont = 0
        skip 2 lines
        print "D E S T I N O",
            column 36, "RUTA",
            column 45, "C V E   P R O D U C T O"
        print "-----"

        let w_cont = 0
        before group of w_origen
        let w_cont_rut = 0
        skip to top of page

        before group of ENTRE)
        pause (CADA CINCO LINEA DE IMPRESION)
        end if
end report
on every row
    select d_prod into w_desc from fclbg02
        where c_prod = r.c_prod

    print r.c_destino using "#####",2 spaces, w_orides.d_oridel,
        column 37, r.c_ruta using "##",
        6 spaces, r.c_prod using "#####", 2 spaces, w_desc
    let w_cont = w_cont + 1 (OJO... ESTO ES PARA CONTROLAR)
    if (w_cont mod 10) = false then (POR PANTALLA, UNA PAUSA

```

```

*****
PROGRAMA:   FCLFA17.4GL
OBJETIVO:  DAR MANTENIMIENTO AL CATALOGO DE PRECIO DE PRODUCTOS, ASI COMO
           LA GENERACION DE SUS REPORTES
FUNCION:   ALTAS, BAJAS, CAMBIOS, CONSULTAS Y GENERACION DE REPORTES DEL
           CATALOGO DE PRECIO DE PRODUCTOS
AUTOR:    ISRAEL ROMERO MARTINEZ  FECHA:11-11-92
*****

```

```

database fletes          (ABRE LA BASE DE DATOS DE FLETES)

```

```

globals
  "clgfa17.4gl"          (INCLUYE LA DEFINICION DE VARIABLES GLOBALES)

```

```

(===== PROGRAMA PRINCIPAL =====)

```

```

main
  options
    help key f1,
    help file "/home/liquida/doc/ayuda"

  (OBTIENE LOS PARAMETROS, DE CONTROL Y AUDITORIA. CLAVE DE USUARIO, IDENTIFI)
  (CADOR DE USUARIO, IDENTIFICADOR DE LA TERMINAL E IDENTIFICADOR DEL PROG )
  (ENVIADOS POR EL PROGRAMA GENERADOR DE MENUS FCLFM08 )
  let w_user = arg_val(1)
  let w_ident = arg_val(2)
  let w_idtty = arg_val(3)
  let w_idpro = arg_val(4)

```

```

defer interrupt
  (===== ABRE Y DESPLIEGA EL FORMATO DE MANTENIMIENTO =====)
  open form forma01 from "fclfa17"
  display form forma01
  let w_fecha = today
  display w_idpro at 3,1
  display w_fecha using "dd/mm/yy" at 4,1
  display w_idtty at 4,72

```

```

(===== SE CONSTRUYE MENU PARA MANTENIMIENTO =====)

```

```

menu "PRECIOS"
  command "Altas" "Registro de nuevos Precios de prod."
    help 151
    call altas()
  command "Bajas" "Baja de Precios"
    help 152
    call baja_fisica()
  command "Correcciones" "Correcciones al registro de Precios"
    help 153
    call modifica()
  command "Nuevos precios" "Registro de nuevos Precios de productos"
    help 154
    call cambio()
  command key ("O") "cConsultas" "Consultas al Catalogo de Precios"
    help 155
  close form forma01
  call consultas() (LLAMA A LA FUNCION EXTERNA DE CONSULTAS)
  open form forma01 from "fclfa17"
  display form forma01
  display w_idpro at 3,1
  display w_fecha using "dd/mm/yy" at 4,1

```

```

    display w_idtty at 4,72
command "Reportes" "Generacion de reportes del Catalogo de Precios"
    help 156
    close form forma01
    call reportes() (LLAMA A LA FUNCION EXTERNA DE REPORTES)
    open form forma01 from "fclfal7"
    display form forma01
    display w_idpro at 3,1
    display w_fecha using "dd/mm/yy" at 4,1
    display w_idtty at 4,72
command key ("y") "aYuda"
    "Muestra ayuda para construir criterio de seleccion"
    call showhelp(9999)
command key ("!") (OJO... OPCION ESCONDIDA)
    call comando()
command key (f3)
    clear screen
    exit menu
end menu
end main
(===== FIN DEL PROGRAMA PRINCIPAL =====)
{*****}
FUNCION: ALTAS
OBJETIVO: DAR DE ALTA UN NUEVO REGISTRO EN LA TABLA DE PRECIOS DE PROD.
PARAMETROS: NINGUNO
{*****}
function altas()

options
    input wrap (ESTA OPCION ES PARA QUE EL CURSOR REGRESE A HOME ;
              (DESPUES DEL ULTIMO CAMPO DE LA PANTALLA )

call limpia("ALTAS",1) (BORRA EL MENU DE MANTENIMIENTO)

(===== PROCESO INTERACTIVO PARA DAR DE ALTA REGISTROS =====)
let descev="ALTA PRECIO NUEVO PROD."
while true
    input by name w_preci.* (LEE DE PANTALLA TODOS LOS CAMPOS)
    on key (f1)
        case
            when infield(c_prod)
                call showhelp(157)
            when infield(f_iniprec)
                call showhelp(158)
            when infield(f_finprec)
                call showhelp(159)
            when infield(n_impres)
                call showhelp(1510)
        end case

    after field c_prod (DESPUES DE TECLEAR LA CLAVE DEL )
        (PRODUCTO )

        display "" at 23,1
        if not valida_pro() then
            let w_produ.d_prod = "***** SIN DESCRIPCION *****"
            error " el producto no existe en catalogo"
            next field c_prod
        end if
        display w_produ.d_prod to formonly.d_nompro

```

```

CAT)      if valida_prec("123199") then {VALIDA SI EXISTE EL PREC.PROD EN
          display by name w_preci.*
          error "           El registro ya existe           "
          sleep 1
          clear form
          initialize w_preci.* to null
          next field c_prod
        end if

        after field f_iniprec
          display "31/12/99" to f_finprec
          let w_preci.f_finprec = "12/31/99"
          next field n_impres

        after field n_impres
          if valida_imp() then
            error "           Precio no valido           "
            next field n_impres
          end if
        on key (f3)
          let int_flag = true
          exit input
        end input

        (===== VALIDA SI OPRIMIO F3 PARA ABORTAR EL PROCESO DE ALTAS =====)
        if int_flag then
          let int_flag = false
          exit while
        end if

        (===== INSERTA UN NUEVO REGISTRO EN LA TABLA =====)

        insert into fclbv07 values (w_preci.*)

        call arma_regbv07(w_preci.*,1)
        let r2=" "
        call transac("I","fclbv07",w_ktot,w_user,w_ident,w_idtty,descev,r1,r2)

        clear form
        display "Registro dado de alta" at 23,23 attribute (reverse)
        sleep 1
      end while
      (===== FIN DE ALTAS =====)

      call limpia(" ",0)           (BORRA AVISOS DE LA PANTALLA)

end function

(*****
FUNCION:      BAJA FISICA
OBJETIVO:    DAR DE BAJA FISICAMENTE UN REGISTRO DE LA TABLA DE PRODUCTO
PARAMETROS:  NINGUNO
*****}
function baja_fisica()
  options
    input no wrap   {ESTA OPCION ES PARA QUE EL CURSOR NO REGRESE A HOME }
                   {DESPUES DEL ULTIMO CAMPO DE LA PANTALLA Y SE TERMINE }
                   {EL INPUT CON EL ULTIMO <Enter> }

```

```

call limpia("BAJAS FISICAS",1)                (BORRA EL MENU DE MANTENIMIENTO)

(===== PROCESO INTERACTIVO PARA DAR DE BAJAS REGISTROS =====)
let descev="BAJA FISICA PRECIO PROD."
let descevl="MODIFI. POR BAJA PRECIO"
while true
  input by name w_preci.c_prod thru w_preci.f_finprec
  on key (f1)
    case
      when infield(c_prod)
        call showhelp(157)
      when infield(f_iniprec)
        call showhelp(158)
      when infield(f_finprec)
        call showhelp(159)
    end case
  after field c_prod
  display "" at 23,1
  if not valida_pro() then
    error
    "          La Clave del producto no existe en catalogo
  "
    next field c_prod
  end if
  display w_produ.d_prod to formonly.d_nompro

  on key (f3)
    let int_flag = true
    exit input
end input
(===== VALIDA SI OPRIMIO F3 PARA ABORTAR EL PROCESO DE BAJA =====)
if int_flag then
  let int_flag = false
  exit while
end if

(=====VALIDA QUE EXISTA Y QUE NO ESTE DADO DE BAJA EL REG=====)
select * into w_preci.* from fclbv07
  where c_prod      = w_preci.c_prod and
        f_iniprec   = w_preci.f_iniprec and
        f_finprec   = w_preci.f_finprec
if status = notfound then
  error
  "          El registro que desea dar de baja no existe
"
  continue while
end if

(===== PRESENTA EL REGISTRO A BORRAR EN LA PANTALLA =====)
call display_prec()
prompt " Confirma dar de baja este registro (S/N) ? " for char x
if x matches "[S|s]" then          (VALIDA LA CONFIRMACION DE LA
BAJA)
  if w_preci.f_finprec = "12/31/99" then
    (let w_fech_ant   = w_preci.f_finprec)
    let w_fech_ant   = w_preci.f_iniprec
    delete from fclbv07          (AQUI SE HACE LA BAJA
FISICA)
    where c_prod = w_preci.c_prod and
          f_finprec = "12/31/99"

```

```

call arma_regbv07(w_preci.*,1)
let r2=""
call transac("B","fclbv07",w_ktot,w_user,w_ident,w_idtty,descev,
r1,r2)
select * into w_preor.* from fclbv07
where c_prod = w_preci.c_prod and
f_finprec = w_fech_ant - 1
if status<>notfound
then
call arma_regbv07(w_preor.*,1)
let w_preor.f_finprec="12/31/99"
call arma_regbv07(w_preor.*,2)
call transac("A","fclbv07",w_ktot,w_user,w_ident,w_idtty,
descevl,r1,r2)

update fclbv07 set f_finprec = "12/31/99"
where c_prod = w_preci.c_prod and
f_finprec = w_fech_ant - 1
end if

else
let w_fech_act = w_preci.f_finprec
let w_fech_ant = w_preci.f_iniprec

delete from fclbv07
where c_prod = w_preci.c_prod and
f_iniprec = w_preci.f_iniprec and
f_finprec = w_preci.f_finprec

call arma_regbv07(w_preci.*,1)
let r2=""
call transac("B","fclbv07",w_ktot,w_user,w_ident,w_idtty,descev,
r1,r2)

select * into w_preor.* from fclbv07
where c_prod = w_preci.c_prod and
f_finprec = w_fech_ant - 1
if status<>notfound
then
call arma_regbv07(w_preor.*,1)
let w_preor.f_finprec= w_fech_act
call arma_regbv07(w_preor.*,2)
call transac("A","fclbv07",w_ktot,w_user,w_ident,w_idtty,
descevl,r1,r2)

update fclbv07 set f_finprec = w_fech_act
where c_prod = w_preci.c_prod and
f_finprec = w_fech_ant - 1
end if

end if
display "Registro dado de baja" at 23,23 attribute (reverse)
sleep 1
clear form
end if
end while

(===== FIN DE BAJAS =====)

```



```

clear form
call limpia(" ",0)                                (BORRA AVISOS DE LA PANTALLA)

end function

{*****
FUNCION: CAMBIO DE PRECIO DE PRODUCTO
OBJETIVO: DAR DE ALTA UN NUEVO REGISTRO EN LA TABLA DE PRECIOS DE PROD.
PARAMETROS: NINGUNO
*****}
)
function cambio()
options
input wrap          (ESTA OPCION ES PARA QUE EL CURSOR REGRESE A HOME          )
                   (DESPUES DEL ULTIMO CAMPO DE LA PANTALLA                    )

call limpia("NUEVOS PRECIOS",1)                    (BORRA EL MENU DE MANTENIMIENTO)

(===== PROCESO INTERACTIVO PARA DAR NUEVOS PRECIO =====)
let descov="NUEVO PRECIO PRODUCTO"
while true
  initialize w_preci.* to null
  input by name w_preci.c_prod                      (LEE DE PANTALLA TODOS LOS
CAMPOS)
  help 157
  after field c_prod
  if not valida_prec("12/31/99") then (VALIDA SI EXISTE EL PROD EN
CAT)
    error "          El registro no existe          "
  next field c_prod
  end if

  if not valida_pro() then
    let w_produ.d_prod = "***** SIN DESCRIPCION *****"
  end if
  display w_produ.d_prod to formonly.d_nompro
  display by name w_preci.f_iniprec thru w_preci.n_imprec
  exit input
  on key (f3)
    let int_flag = true
    exit input
  end input

(===== VALIDA SI OPRIMIO F3 PARA ABORTAR EL PROCESO DE CAMBIO =====)
if int_flag then
  let int_flag = false
  exit while
end if

let w_fecha      = today
let w_fech_ant = w_preci.f_iniprec
input by name w_preci.f_iniprec,
              w_preci.n_imprec without defaults
on key(f1)
case
  when infield(f_iniprec)
    call showhelp(158)
  when infield(n_imprec)
    call showhelp(1510)
end case

```

```

after field f_iniprec
  if w_preci.f_iniprec > w_fecha then
    error "*** LA FECHA NO PUEDE MAYOR A HOY ***"
  next field f_iniprec
end if
if w_preci.f_iniprec <= w_fech_ant then
  error
  "*** LA FECHA NO PUEDE SE MENOR O IGUAL A LA VIGENTE ***"
  next field f_iniprec
end if
after field n_impres
  if valida_imp() then
    error "          Precio no valido"
  next field n_impres
end if
on key (f3)
  let int_flag = true
  exit input
end input

(===== VALIDA SI OPRIMIO F3 PARA ABORTAR EL PROCESO DE CAMBIO =====)
if int_flag then
  let int_flag = false
  exit while
end if

select * into w_preor.* from fclbv07
  where c_prod = w_preci.c_prod and
        f_finprec = "12/31/99"
call arma_regbv07(w_preor.*,1)
let w_preor.f_finprec=( w_preci.f_iniprec -1 )
call arma_regbv07(w_preor.*,2)
call transac("A","fclbv07",w_ktot,w_user,w_ident,w_idtty,descev,r1,r2)

update fclbv07 set f_finprec =( w_preci.f_iniprec - 1)
  where c_prod = w_preci.c_prod and
        f_finprec = "12/31/99"

(===== INSERTA UN NUEVO REGISTRO EN LA TABLA =====)
insert into fclbv07 values (w_preci.*)

call arma_regbv07(w_preci.*,1)
let r2=""
call transac("I","fclbv07",w_ktot,w_user,w_ident,w_idtty,descev,r1,r2)

clear form
display "Cambio de precio realizado" at 23,23 attribute (reverse)
sleep 1
end while
(===== FIN DE CAMBIO =====)
clear form
call limpia(" ",0)          (BORRA AVISOS DE LA PANTALLA)
end function

```

```

{*****
FUNCION:  MODIFICACIONES
OBJETIVO: MODIFICAR LA INFORMACION DE UN REGISTRO DE LA TABLA DE PRODUCTO
PARAMETROS: NINGUNO
*****
}
function modifica()

    call limpia("MODIFICACIONES",1)      (BORRA EL MENU DE MANTENIMIENTO)

    (===== PROCESO INTERACTIVO PARA MODIFICAR REGISTROS =====)
    let descev="MODIFICACION PRECIO PROD."
    while true

        display "" at 22,23 attribute (reverse)
        options
            input no wrap (ESTA OPCION ES PARA QUE EL CURSOR NO REGRESE A HOME )
                        (DESPUES DEL ULTIMO CAMPO DE LA PANTALLA Y SE TERMINE )
                        (EL INPUT CON EL ULTIMO <Enter> )

        input by name w_preci.c_prod      (LEE SOLO LA CLAVE DEL PRODUCTO)
            help 157
            on key (f3)
                let int_flag = true
            exit input
        end input

        (===== VALIDA SI OPRIMIO F3 PARA ABORTAR EL PROCESO DE MODIF. =====)
        if int_flag then
            let int_flag = false
            exit while
        end if

        (===== VALIDA QUE EXISTA Y QUE NO ESTE DADO DE BAJA EL REG =====)
        if valida_prec("12/31/99") = false then
            error " El registro que desea modificar no existe"
            continue while
        end if

        (===== PRESENTA EL REGISTRO A MODIFICAR EN LA PANTALLA =====)
        call display_prec()
        call arma_regbv07(w_preci.*,1)

        (===== LEE DE PANTALLA LAS MODIFICACIONES =====)
        display "Teclee sus modificaciones " at 22,23 attribute (reverse)
        sleep 1
        options
            input wrap      (ESTA OPCION ES PARA QUE EL CURSOR REGRESE A HOME )
                        (DESPUES DEL ULTIMO CAMPO DE LA PANTALLA )

        let w_preor.n_imprec = w_preci.n_imprec
        let w_fech_ant = w_preci.f_iniprec
        input by name w_preci.f_iniprec, w_preci.n_imprec without defaults
            on key (f1)
                case
                    when infield(f_iniprec)
                        call showhelp(158)
                    when infield(n_imprec)
                        call showhelp(1510)
                end case
        after field f_iniprec
            if w_preci.f_iniprec < w_fech_ant then
                error
            end if
    end while
end function

```

```

"*** ERROR *** Ya existe un reg. historico que incluye esta fecha de inicio
"
        next field f_iniprec
    end if

    on key (f3)
        let int_flag = true
        exit input
    end input

    (===== VALIDA SI OPRIMIO F3 PARA ABORTAR EL PROCESO DE MODIF. =====)
    if int_flag then
        let int_flag = false
        exit while
    end if
    (===== VERIFICA SI HUBO ALMENOS UN CAMBIO EN LOS CAMPOS =====)
    if w_preci.f_iniprec <> w_fech_ant then
        update fclbv07 set f_iniprec = w_preci.f_iniprec
            where c_prod = w_preci.c_prod and
                f_iniprec = w_fech_ant
        update fclbv07 set f_finprec = w_preci.f_iniprec - 1
            where c_prod = w_preci.c_prod and
                f_finprec = w_fech_ant - 1
    end if
    if w_preci.n_impres <> w_preor.n_impres then
        update fclbv07 set n_impres = w_preci.n_impres
            where c_prod = w_preci.c_prod and
                f_finprec = w_preci.f_finprec
    end if

    let w_preci.f_finprec=w_preci.f_iniprec-1
    call arma_regbv07(w_preci.*,2)
    call transac("A","fclbv07",w_ktot,w_user,w_ident,w_idtty,descev,r1,r2)

    clear form
    display "Registro Modificado      " at 23,23 attribute (reverse)
    sleep 1
end while
(=====FIN DE MODIFICACIONES =====)

clear form
call limpia(" ",0)                                (BORRA AVISOS DE LA FANTALLA)

end function

{*****
FUNCION:      VALIDA IMP
OBJETIVO:    VALIDAR QUE EXISTA O NO UN REGISTRO DE LA TABLA DE PRODUCTO
PARAMETROS:  NINGUNO
*****}
function valida_imp()
    if w_preci.n_impres <= 0 then
        return 1
    end if
    return 0      (NO ENCONTRO EL REGISTRO)
end function

```

```

(*****
FUNCION: COMANDO
OBJETIVO: EJECUTAR UN COMANDO DEL SISTEMA OPERATIVO, SIN SALIR DE LA
          APLICACION
PARAMETROS: NINGUNO
*****
)
function comando()
  define w_comando char(80) (AQUI SE DEPOSITA EL COMANDO A EJECUTAR)
  clear screen
  let x = "!"
  while (x = "!")
    prompt "#" for w_comando
    run w_comando
    prompt "Teclee RETURN para continuar" for char x
  end while
  clear screen
  display form forma01
  display w_idpro at 3,1
  display w_fecha using "dd/mm/yy" at 4,1
  display w_idtty at 4,72
end function

(*****
FUNCION : arma_regbv07
OBJETIVO : CONSTRUIR UN REGISTRO (Tipo char) CON LOS DATOS DEL REGISTRO
          DE PRECIOS DE PRODUCTOS, QUE SERA ALMACENADO EN LA BITACORA
          DE TRANSACCIONES
PARAMETROS : EL REGISTRO DE LA TABLA Y EL SWITCH (1 o 2) QUE INDICA EN
          QUE VARIABLE (r1 o r2) ARMAR EL REGISTRO
*****)
function arma_regbv07(r,nr)
define r record like fclbv07.*,
      nr smallint,
      rc char(80),
      cprod char(5),
      finiprec char(8),
      ffiniprec char(8),
      nimprec char(10)

let rc=" "
let cprod=r.c_prod using "#####"
let finiprec=r.f_iniprec using "dd/mm/yy"
let ffiniprec=r.f_finiprec using "dd/mm/yy"
let nimprec=r.n_impres using "#####.##"
let rc =cprod,"(",finiprec,"(",ffiniprec,"(",nimprec
if nr=1 then
  let r1=rc
else
  let r2=rc
end if
let w_ktot =cprod,"(",finiprec,"(",ffiniprec
end function
□

```

G L O S A R I O

Action (Acción)

Una instrucción dada directamente por un usuario a la "Workstation" para hacer algo.

Active Window (Ventana activa)

Ventana en la cual un usuario puede desplegar información y hacer cambios. Una ventana activa siempre despliega los bloques del menú, cajas cerradas, etc., alrededor de su borde.

Activity (Actividad)

Algo que se desarrolla mediante un proyecto, llevando a cabo un propósito establecido. Una actividad puede ser una función o un proceso.

Affinity Analysis (Análisis de afinidad)

Para objetos asociados, reporta el análisis en un rango de las asociaciones que tienen en común contra el número de asociaciones no tienen nada en común.

Analysis (Análisis)

Una operación que genera un desplegado de información o un reporte derivado desde el contenido de la enciclopedia.

Glosario

Ancestor (Predecesor)

Cualquier objeto en una descomposición que tiene parentesco, parentesco del parentesco, y así continuar de un objeto específico.

Association Instance (Asociación de instancias)

Conexión que existe entre las instancias objeto de dos tipos especificados en la enciclopedia.

Association Matrix (Matriz de asociación)

Un diagrama describiendo un tipo de asociación entre instancias de objeto (muestra a los renglones y columnas de la matriz).

Association Type (Tipo de asociación)

Tipo de conexión o asociación que puede existir entre dos objetos de tipo específico en la enciclopedia.

Associative Entity Type (Asociación de tipo de entidades)

Un tipo de entidad que existe primordialmente para interrelacionar otros tipos de entidades.

Attribute Instance (Instancia de atributo)

Valor que describe una característica o propiedad de una instancia de entidad.

Attribute Type (Tipo de atributo)

Tipo de característica o propiedad descrita en términos de un valor de entidades de un tipo dado.

Glosario

Attribute Entity Type (Atributo de tipo de entidad)

Tipo de entidad de una instancia que describe ampliamente entidades de otros tipos en acuerdos con atributos y dependientes de la existencia de instancias de entidad de otro tipo.

Cardinality Constraint (Construcción de la Cardinalidad)

El área máxima y mínima en el número de instancias en un tipo con respecto a una instancia de otro tipo. En el contexto de un tipo de atributo, un tipo es un tipo de entidad y el otro tipo es un tipo de valor. En el contexto de un tipo de relación ambos tipos son de tipos de entidades.

CASE (Computer Aided Software Engineering)

Ingeniería de Software Asistida por Computadora.

Check Box (Marcas de caja)

Un conjunto de alternativas presentadas en una caja de dialogo que no es mutuamente exclusiva con respecto a cualquier otra.

Concatenation (Concatenación)

En un modelo entidad, una combinación nombrada de dos o más atributos o tipos de relaciones (llamados, sus componentes), todo lo que describe a un tipo de entidad. Una instancia de tipos de componentes, arreglados en un orden específico.

Glosario

Critical Assumption (Suposición Crítica)

Una suposición es una ayuda para el proyecto, ambiente de negocios, competencia o industria que soporta o valida a un factor de sucesos críticos.

Critical Success Factor (Factor de sucesos críticos)

Ambición fundamental del proyecto, el cual se lleva a cabo si es necesario para encontrar otras ideas y objetivos.

Data Collection (Colección de datos)

Una colección de datos es implementada como una unidad, la cual contiene factores de interés para el proyecto.

Declared Root (Declarar raíz)

La punta de una estructura de árbol que ha sido declarada como raíz a pesar del número de parientes que tenga.

Decomposition (Desintegración o Descomposición)

Una descripción de las estructuras o composición de un objeto como consistencia de varios objetos del mismo tipo el cual en turno consiste de más objetos del mismo tipo y así hasta un número arbitrario de niveles. En general, cualquier jerarquía u orden de red formada por asociación recursiva (o encadenamiento de asociaciones) de un tipo particular entre objetos de un tipo particular. Los procesos de analizar un objeto así como para producir una descripción de ésta ordenación.

Glosario

Decomposition Diagram (Diagramas de descomposición)

Un diagrama en el cual son creadas y vistas la desintegración de objetos.

Descendat (Descendiente)

Cualquier objeto --en un diagrama de desintegración -- que es hijo, hijo del hijo, o hijo del hijo del hijo, etc.

Detail Window (Ventana de detalle)

Una ventana contiene detalles acerca de instancias de objetos descritas en la enciclopedia.

Diagram (Diagrama)

Representación gráfica usada de los objetos en contenidos en la enciclopedia.

Dialog Box (Caja de dialogo)

Área de la ventana en la cual obtenemos información o una decisión del usuario.

Encyclopedia (Enciclopedia)

Base de datos en donde están definidas toda la información relevante de los componentes manipulados por "Workstation"; depósito central que contiene por entero el proyecto a desarrollar o desarrollado.

Encyclopedia Model (Modelo enciclopedia)

Modelo entidad que define los tipos de información que puede almacenarse en la enciclopedia.

Glosario

Enterprise (Proyecto)

Parte del negocio u organización para la cual la ingeniería de información es usada para definir e integrar un sistema de información.

Enterprise Model (Modelo de proyecto)

Modelo de el proyecto

Entity (Entidad)

Instancia de un tipo de entidad: Una persona, un lugar, algo o concepto que tiene características de interés para un proyecto; un almacenamiento de datos; las cosas que guardan información acerca de un sistema de información.

Entity Instance (Instancia de entidad)

Entidad representada como pertenencia para un tipo de entidad en un sistema de información en particular, con atributos y tipos de relaciones en común.

Entity Model (Modelo entidad)

Modelo de un tipos de entidad, sus tipos de atributo y los tipos de relaciones entre tipos de entidad que representan la clase de información necesaria para satisfacer un requerimiento de información.

Entity Modeling (Modelado de entidad)

Actividad de la ingeniería de información que enfocada producción de modelos de entidad.

Glosario

Entity Type (Tipo de entidad)

Clase de gente, lugares, cosas o conceptos que tienen características de interés para un proyecto.

Entity Type Description (Descripción de tipo de entidad)

Todos los atributos y relaciones que describen una entidad en particular en el sistema de información.

Function (Función)

Grupo de procesos que juntos soportan un aspecto del proyecto.

Goal (Objetivo)

Origen o fin del todo o una parte del proyecto.

Herramienta CASE

Una herramienta de software que automatiza (al menos en parte) una tarea en particular del ciclo de vida del software.

Icon (Icono)

Símbolo gráfico usado para representar un tipo particular de objeto, propiedad o asociación en la enciclopedia.

Identifier (Identificador)

Modelo entidad, tipo de atributo, tipo de relación o concatenación que tiene uno o varios valores únicos para cada instancia de su tipo de entidad.

Glosario

Implied Root (Raíz incluido)

La punta de una estructura de árbol que es ligada a más de un contexto.

Inactive Window (Ventana inactiva)

Ventana sin frontera activa. Sus contenidos no pueden ser accesados hasta que este sea reactivado.

Information Engineering (Ingeniería de información)

Una metodología para desarrollar un sistema de información integrado basado en el principio de compartir la información que es común.

Information Engineering Workbench/Workstation (Workbench/Workstation de la Ingeniería de Información).

El producto de software de propietario de knowledgeware que incluye varios componentes de software integrado y una Enciclopedia central que es sostenida por la ingeniería de información. Un conjunto integrado de herramientas CASE diseñadas para trabajar conjuntamente y automatizar (o proporcionar ayuda automatizada) totalmente el ciclo de vida del software, que incluye el análisis, el diseño, la codificación y la prueba.

Information Need (La necesidad de la información)

Una declaración de necesidad de información que es crítica para el suceso de los objetivos de la empresa.

Information System (Sistema de Información)

Un sistema de información y proceso que puede ser utilizado para registrar y mantener la información.

Glosario

Juego de Herramientas CASE

Un conjunto de herramientas integradas diseñadas para trabajar conjuntamente y automatizar total o parcialmente una fase del ciclo de vida del software o una clase particular de función del software

Leaf (hoja)

Un objeto en una descomposición que no tiene ningún hijo.

Leaf Entity Type (Hoja de Tipo Entidad)

Una designación arbitraria por uno de los dos tipos de entidad que participan en un tipo de relación.

Local Data Type (Tipo de Dato Local)

Un tipo de dato utilizado para describir uno y solamente un tipo de información de atributo elemento.

Menú (Menú)

Una lista de acciones disponible para el usuario que puede seleccionar señalando y activando con el "click" del botón de ratón.

Metodología CASE

Una metodología estructurada automatizable que define una formulación técnica y disciplinada para todos o algunos de los aspectos del desarrollo y del mantenimiento del software.

Glosario

Model (Modelo)

Representación de algún aspecto la empresa almacenada en la Enciclopedia como objeto, con el fin de ejemplifica y obtener descripciones.

Object (Objeto)

Persona, lugar, cosa, o concepto que tiene características de ingeniería de información. Algo que es guardado en la Enciclopedia para sostener la información; La información que se mantiene acerca de algo.

Object Description (Descripción del Objeto)

Todas las propiedades y las asociaciones que describen un objeto particular en la Enciclopedia.

Object Instance (Instancias Objeto)

Clase de gente, lugares, cosas, o los conceptos teniendo características de interés para la Enciclopedia

Plataforma Hardware CASE

Sistema de arquitectura hardware que proporciona una plataforma de operación a las herramientas CASE.

Process (Proceso)

Una actividad que es ejecutada repetidamente dentro de un proyecto, cada ejecución resultado en una clase específica que afecta entidades, o información acerca de las entidades de tipos específicos.



BIBLIOGRAFÍA



Bibliografía**Libros, Revistas y Manuales**

AUTOR	TÍTULO	EDITORIAL
Carma McClure,	CASE la automatización del Software	Addison-Wesley Iberoamericana, 1993.
Edward Yourdon y Larry Constantine	Structured Desing	Yourdon Press, 1978.
F.L. Bauer	Software Engineering	North Holland Publishing. 1972.
M. Page Jones	the Practical Guide to Structured Systems Design	Yourdon Inc. 1980.
C.Gane y T. Sarson	Structured Systems Analysis: Tools and Techniques	New York, 1977.
J. Martin	Fourt-Generation Languages	Prentince-Hall, 1985.
Roger S. Pressman	Ingeniería del Software (un enfoque practico)	Mc. Graw Hill
J. Martin y Carma McClure	Structured Techniques: The Basis for CASE	Prentice Hall, 1988
IEW	Planning Workstation-Guia de instalación	KnowledgeWare, Inc. 1992
IEW	Analysis Workstation-Guia de instalación	KnowledgeWare, Inc. 1992
IEW	Desing Workstation-Guia de instalación	KnowledgeWare, Inc. 1992
IEW	Planning Workstation-Guia del usuario,	KnowledgeWare, Inc. 1992
IEW	Analysis Workstation-Guia del usuario	KnowledgeWare, Inc. 1992
IEW	Design Workstation-Guia del usuario	KnowledgeWare, Inc. 1992

Bibliografía**Libros, Revistas y Manuales**

AUTO	TITULO	EDITORIAL
IEW	Strategies & Analysis, Volumen I y II	KnowledgeWare, Inc. 1992
IEW	Seminar Workbook	KnowledgeWare, Inc. 1992
Pc. Computer	DBMS-Client/Server Computing	Byte