

10
2 ej

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO



FACULTAD DE INGENIERIA

CERTIFICACION DE CALIDAD
DE SISTEMAS DE INFORMACION

T E S I S

QUE PRESENTAN:
PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION

ANA MARIA BRISERO GARCIA
MONICA ALEJANDRA LOPEZ FLORES

Director de Tesis
ING. SERGIO RUIZ PALACIOS



México, D. F.

1991

TESIS CON
FALLA



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está 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.

INDICE	i
INDICE DE FIGURAS	v
INDICE DE TABLAS	vii

INTRODUCCION

Situación actual de la certificación de calidad del software.	1
Fases en el desarrollo de sistemas	4
Errores en el desarrollo de sistemas	6
Problemas en la modificación y mantenimiento del software	8
Estandarización del software	8

CAPITULO I. Conceptos básicos de la certificación de calidad

1.1 Calidad	9
1.2 Control de calidad	9
1.3 Software	10
1.4 Calidad de Software	11
1.5 Certificación de calidad de Software	12
1.6 Medición para distintos tipos de Software	12

CAPITULO II. Marco de referencia para la certificación de calidad de software.

2.1 Introducción	16
2.2 Modelo jerárquico de la calidad de software	17
2.3 Procedimientos para identificar los factores de calidad	18
2.4 Consideraciones de las características básicas de la aplicación	18

CAPITULO II...cont

2.5 Factores de calidad	21
2.6 Consideraciones de las implicaciones del ciclo de vida	22
2.7 Identificación de criterios de calidad	22
2.8 Definición de criterios	24
2.9 Definición de métricas	25
2.10 Clasificación de métricas	25
2.11 Métricas de Software	
2.11.1 Métricas de tamaño	26
2.11.2 Métricas de estructuras de datos	34
2.11.3 Métricas de estructura lógica	36
2.11.4 Métricas de composición	52
2.11.5 Métricas de diseño	59

CAPITULO III. Costos de la certificación de calidad

3.1 Introducción	62
3.2 Costos de la certificación de calidad	62
3.3 Costo total de calidad	65
3.4 Análisis de actividades	66
3.5 Enfoque de costos	71
3.6 Clasificación de costos	
3.6.1 Costos que no son de calidad	73
3.6.2 Costos de calidad	73
3.7 Abuso potencial del costo	74
3.8 Productividad	76

CAPITULO IV Herramientas de certificación de calidad

4.1	Introducción	80
4.2	SMF-II	
4.2.1	Descripción de la herramienta	81
4.2.2	Reportes	88
4.3	CHECK-OUT	
4.3.1	Descripción de la herramienta	92
4.3.2	Reportes	99
4.4	MEASURE	
4.4.1	Descripción de la herramienta	102
4.4.2	Reportes	108
4.5	PATHVU	
4.5.1	Descripción de la herramienta	112
4.5.2	Reportes	117

CAPITULO V Evaluación y certificación de calidad de un sistema de aplicación.

5.1	Descripción del sistema de Inventario de Ordenes de pago bancarias.	123
5.1.1	Ciclo del vida del sistema	
5.1.1.1	Diagrama del ciclo de vida	124
5.1.1.2	Descripción del ciclo de vida	125
5.1.2	Desarrollo del sistema	126
5.1.3	Módulos del sistema	127
5.1.4	Descripción de módulos	130

CAPITULO V ...cont

5.2 Aplicación de las métricas al sistema 132

5.3 Resultados comparativos 207

CONCLUSIONES 211

BIBLIOGRAFIA 213

ANEXO 1. Estrategia de conteo para el lenguaje Pascal

ANEXO 2. Formulario de metricas de certificación de calidad

ANEXO 3. Diagramas de Flujo

ANEXO 4. Programas Fuente

INDICE DE FIGURAS

INTRODUCCION

Figura I.1	Distribución de esfuerzo total para el desarrollo y mantenimiento del software	4
Figura I.2	Fases de detección de errores	7

CAPITULO II

Figura 2.1	Programa de ejemplo 1.	28
Figura 2.2	Métrica de contador de tokens	30
Figura 2.3	Programa de ejemplo 2.	34
Figura 2.4	Lista de referencia cruzada	35
Figura 2.5	Conteo de variables	35
Figura 2.6	Diagrama de flujo 1.	37
Figura 2.7	Diagrama de flujo 2.	38
Figura 2.8	Prueba condicional que selecciona una opción	40
Figura 2.9	Salto hacia atrás incondicional utilizado para crear ciclos	40
Figura 2.10	Salto hacia atrás siguiendo una prueba condicional	41
Figura 2.11	Representación por diagrama de flujo de un condicional compuesto	43
Figura 2.12	Diagrama de flujo con 13 nodos y 16 líneas de dirección	45
Figura 2.13	Diagrama de flujo simple de una secuencia de código lineal	46
Figura 2.14	Flujo de una subrutina	48
Figura 2.15	Programa de ejemplo 3	50

CAPITULO III

Figura 3.1	Análisis de actividades por departamento	68
Figura 3.2	Forma para determinación de actividades	69
Figura 3.3	Análisis de actividades de requerimientos	70
Figura 3.4	Análisis de actividades	72
Figura 3.5	Análisis de calidad	75

CAPITULO IV

Figura 4.1	Diagrama general del SMF-II	83
Figura 4.2	Flujo de utilización del PATHVU	115
Figura 4.3	Proceso de PATHVU	116

CAPITULO V

Figura 5.1	Diagrama del ciclo de vida del sistema	124
------------	--	-----

INDICE DE TABLAS

INTRODUCCION

Tabla I.1	Evolución del software	3
-----------	------------------------	---

CAPITULO I

Tabla 1.1	Medición para los distintos tipos del software	15
-----------	--	----

CAPITULO II

Tabla 2.1	Forma de reconocimiento de requerimientos de factores de calidad de software	19
-----------	--	----

Tabla 2.2	Características del sistema y factores de calidad relacionados	20
-----------	--	----

Tabla 2.3	Impacto de no especificar o medir los factores de calidad	23
-----------	---	----

Tabla 2.4	Relación entre factores de calidad	23
-----------	------------------------------------	----

Tabla 2.5	Vector de métricas	52
-----------	--------------------	----

CAPITULO III

Tabla 3.1	Categorías del costo de calidad	73
-----------	---------------------------------	----

Tabla 3.2	Actividades del Software	78
-----------	--------------------------	----

Tabla 3.3	Atributos del software para diseño	79
-----------	------------------------------------	----

I N T R O D U C C I O N

SITUACION ACTUAL DEL CONTROL DE CALIDAD DEL SOFTWARE

El término informatica o computación se refiere no solo al equipo de comuto, sino también a los aspectos relacionados con su uso en sistemas de informacion y las estructuras administrativas requeridas para su emoleo.

En un principio, las computadoras empezaron a jugar el papel de simbolo de estatus y avance tecnologico para las instituciones que las poseian, sin que estas lograrán obtener en realidad, beneficios acordes a las inversiones realizadas.

Esta situacion cambia radicalmente en unos cuantos años en un numero extraordinario de empresas e instituciones, y muy especialmente del sector de servicios donde se logra utilizar con éxito esta tecnología para incrementar las ventas y conquistar así nuevos mercados.

Durante la década de los 60's, un número creciente de empresas toman parte en esta competencia economica y financiera, que gira alrededor de la construcción de equipos mas grandes y de la definicion de nuevos mercados.

Para fines de la decada de los sesenta, unas cuantas firmas permanecen en el mercado (IBM, Burroughs, Univac, CDC), seguidas por empresas de reciente creacion como DEC y Honeywell Bull.

El desarrollo comercial de la computacion se concentra en E.U., y esto se explica en terminos financieros. Este mismo factor determino posteriormente el crecimiento en la industria Japonesa y Alemana que actualmente ocupan 2ndo y 3er lugar respectivamente.

Los primeros esfuerzos comerciales estuvieron orientados a la producción de hardware, mientras que el software era considerado como una actividad auxiliar realizada mas por los usuarios. Sin embargo, esto se ha ido transformando en los últimos años.

En la actualidad la tendencia principal del desarrollo de software esta orientada a construir soluciones integrales para un tipo especifico de establecimientos. Esta idea se ha visto reforzada, tanto por la existencia de una infraestructura de comunicaciones telefonicas, como por los avances recientes de microprocesadores y la entrada al mercado de las microcomputadoras.

La tendencia que se define cada vez con mayor claridad es la de la sustitucion del software de subsistencia por uno de tipo comercial de mayor calidad y flexibilidad.

En los paises subdesarrollados, las actividades relacionadas con la tecnologia de comuto, aparecen con un retraso de 10 a 12 años.

Las naciones atrasadas inician esfuerzos serios de asimilacion tecnologica desde la decada de los 70's, aun con los riesgos de adquirir patrones de administracion totalmente extraños.

El desarrollo de arquitecturas y componentes (hardware) no habria tenido sentido, sin un desarrollo equilibrado de software. Asi por ejemplo, la operacion actual de un sistema de varios procesadores centrales, que controlan el proceso de informacion o consultas y actualizan bases de datos, implican la existencia y operacion simultanea de un gran numero de programas de soporte que desempeñan una amplia gama de funciones en forma transparente para los usuarios.

La realizacion del potencial de las computadoras depende de la habilidad de producir software confiable a un costo razonable. Se ha visto que el costo del software se incrementa y el hardware se decremента. Esto se debe a :

- La falta de especialistas y por lo tanto el alto costo de estos.
- El bajo porcentaje de ganancias en productividad de programacion.
- La demanda de mayor seguridad, facilidad de uso y mas ingenieria humana de software.
- El hardware se produce en serie, por lo que el costo se puede prorratear a diferencia del software que se no normalmente no se produce en serie.

El costo del software se eleva generalmente por un deslizamiento en el plan de desarrollo. Una vez que un proyecto cae fuera del plan es difícil de recuperar por la naturaleza secuencial del proceso de desarrollo. Para estar a tiempo, se aumentan más horas hombre, sin embargo esto resulta contraproducente incrementándose el costo del producto.

El desarrollo del software, además de facilitar la interacción entre hombre y máquina en forma operativa, ha permitido construir herramientas y definir metodologías que dan al analista y al programador actual una mayor capacidad, que la de los analistas y programadores anteriores.

En la tabla 1. se enumeran algunos de los avances logrados en esta dirección y sus implicaciones en las actividades de programación.

CONCEPTO (utilización comercial)	AÑO	FACTOR DE AVANCE	CAPACIDAD
Lenguaje de máquina primitiva.	1955	---	1
Ensamblador (MAP, FAP) etc.	1958	1.8	2.5
Interpretes (Bell, Lisp, etc.)	1962	1.5	2.7
Compiladores (primeros lenguajes)			
Fortran.	1964	2.5	6.75
Compiladores (Algol 50, Cobol, etc)	1967	1.2	8.10
Editores en línea (tiempo, compart.)	1968	1.5	12.15
Memoria virtual	1970	1.1	13.4
Programación segmentada/inter- comunicación de lenguajes.	1972	1.2	16.0
Metodologías de documentación y programación estructurada	1974	1.3	20.8
Administración de base de da- tos	1976	1.2	25.2
Metaprocesadores para Cobol y Fortran.	1977	1.2	30.0
Editores de página.	1978	1.1	33.0
Nuevos lenguajes de Programación (Pascal, PL1, c, etc.)	1979	1.2	39.6
Cuarta generación (SQL, DB-IV, etc)	1983		

FASES EN EL DESARROLLO DE SISTEMAS

Con el crecimiento de la complejidad, el desarrollo de los sistemas ha hecho necesaria la realización de actividades secundarias de planeación, administración, evaluación e intercomunicación, que sustenten las actividades básicas de análisis y programación.

El periodo de vida de un producto de programación es de uno a tres años en su desarrollo y de cinco a quince en su uso, incluyendo el mantenimiento. La distribución de esfuerzos entre desarrollo y mantenimiento se ha informado como a razón de 40/60.

Si se hace una normalización del esfuerzo total, se obtienen los resultados de estas distribuciones que son presentadas en la siguiente figura.

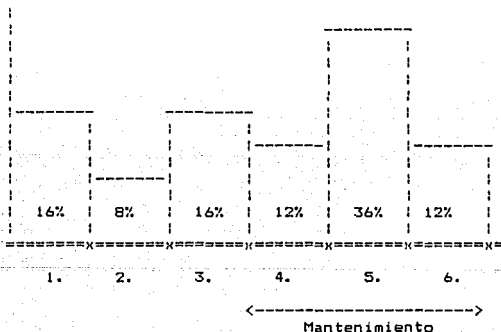


FIG. I.1

1. Análisis y diseño.
2. Implementación.
3. Pruebas e instalación.
4. Adaptación.
5. Mejoras.
6. Arreglos.

ANALISIS.

La fase de análisis concierne principalmente a la generación de las especificaciones de las funciones requeridas y características de ejecución del software, así como los recursos necesarios para construir y modificar el sistema.

DISEÑO.

El diseño puede dividirse en 2 etapas.

En la primera se diseña en forma general la configuración del sistema, el lenguaje de implementación, los módulos principales con sus interfaces, las bases de datos y el plan para su prueba. La otra etapa debe ser más detallada, dándose en esta la definición específica de cada módulo.

IMPLEMENTACION.

La producción de los módulos en el lenguaje especificado en el diseño.

PRUEBAS E INSTALACION.

Se supervisan los requisitos y especificaciones del diseño, el código y las pruebas de aceptación o pilotos. En caso de ser aceptado, el producto es liberado.

MANTENIMIENTO.

En esta fase se descubren errores adicionales, se corrigen, se hacen cambios en el código y en los manuales, se adicionan y se depuran funciones.

ERRORES EN EL DESARROLLO DE SISTEMAS

Los errores mas frecuentes en la produccion de un sistema son:

1.- Se desconoce el ambiente tanto de hardware como de software en el que la aplicacion sera utilizada.

2.- Se sacrifica calidad y facilidad de mantenimiento por eficiencia.

3.- No existe un programa formal de revision interna para verificar que lo codificado satisfaga lo que se quiere hacer.

4.- Falta de Estructura.

5.- Complejidad excesiva.

6.- Falta de documentacion.

7.- Falta de apego a estandares de programacion.

Estos errores, provocan que en ocasiones los sistemas o bien nunca lleguen a funcionar o que padezcan de entrega tardia, sobrecosto, poca confiabilidad, ineficiencia o poca aceptacion por parte del usuario.

En general, el software no es desarrollado para un mantenimiento facil y esto es notorio pues los gerentes de software asignan al personal mas experimentado para esta tarea.

Los errores siguen presentes a pesar de las metodologias modernas de desarrollo de software tales como diseo top-down, programacion estructurada, etc. que se practican actualmente. La solucion a todos estos problemas se basa esencialmente en la simple aplicacion de tecnicas de Control de Calidad durante el desarrollo de los sistemas.

Los errores mas costosos son los que son detectados en el campo. Un estudio de los costos relativos de correccion de errores, por fase de desarrollo, es dada en la figura 1.2.

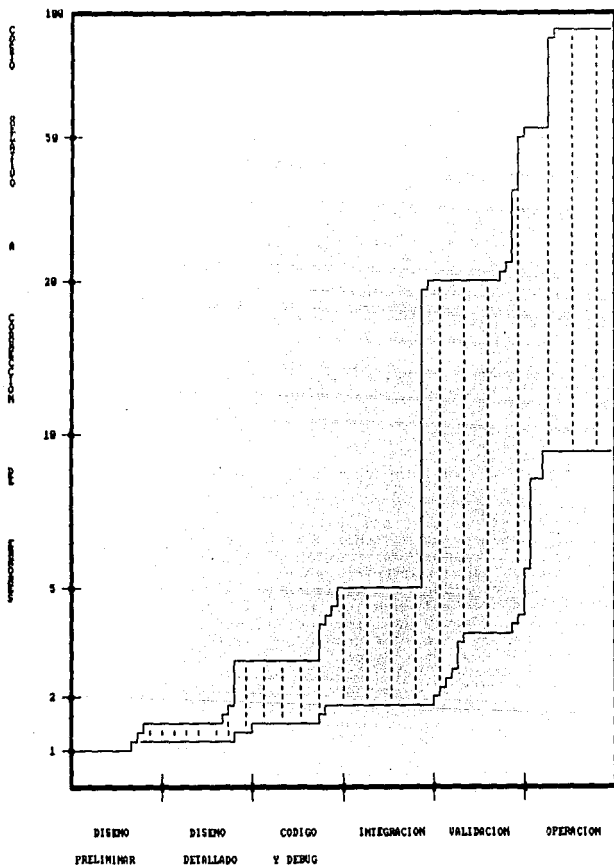


FIGURA 1.2

Muchas de las razones para incrementar los costos de corrección de errores como un proyecto progresivo son:

1. Las pruebas llegan a ser más complejas y lentas.
2. La documentación de los cambios llega a ser más extensiva y costosa.
3. La comunicación de problemas y cambios envuelve a muchas personas.
4. La repetición de pruebas anteriores es costosa.
5. Una vez que la operación está comenzada, el equipo de desarrollo es desintegrado y reasignado.

El Problema de la Modificación del Software y Mantenimiento.

Nos encontramos, que tan grande como es el costo de desarrollo, los gastos de mantenimiento del software son siempre más grandes. El término "costo de mantenimiento" se refiere al gasto necesario para encontrar errores en el software que ha sido oficialmente liberado y colocado en operación. Para los sistemas largos de software, que han sido diseñados para un cierto número de años y han sido modificados de una manera u otra, los esfuerzos asociados con estos cambios, que llamaremos rediseño, son muy grandes y pueden aproximarse o exceder el costo inicial del software. Del 30% al 80% del presupuesto del procesamiento de datos son gastados en rediseño y mantenimiento del software.

Estandarización del Software.

Uno de los fenómenos evidentes en el software es el síndrome de la "reinención de la rueda". La gente está constantemente diseñando muchos programas o grandes partes de programas, una y otra vez.

El progreso en la estandarización ha sido lenta por distintas razones. Lo más importante es el gran esfuerzo necesario para realizar el trabajo y vencer la limitación partidaria y las sutilezas que ocurren con todos los esfuerzos de estandarización.

CAPITULO I

CONCEPTOS BASICOS DE LA CERTIFICACION DE CALIDAD.

1.1 CALIDAD .

La calidad es un concepto abstracto y subjetivo. La cual se puede considerar como un conjunto de cualidades intrínsecas del objeto. Cualquier característica que hace a un objeto bueno o malo, recomendable o censurable y que marca el grado de excelencia que posee, habla de su calidad.

Para la industria, la calidad es lo mejor para el consumidor dentro de ciertas condiciones como son su uso actual y el precio de venta del producto.

La confiabilidad es una parte de la calidad total requerida, del producto, así como, los aspectos de conservación, servicios, resistencia, etc..

Esta calidad es la resultante de una combinación de características de ingeniería y de fabricación que determinan el grado de satisfacción que el producto proporcionará al consumidor durante su uso.

1.2 CONTROL DE CALIDAD .

El control de calidad es una herramienta desarrollada durante la segunda guerra mundial que se basa en métodos estadísticos incluyendo técnicas de muestreo random, pruebas de mediciones, etc..

Desde entonces el control de calidad se ha convertido en un arma poderosa dentro de las industrias para sobrevivir en los mercados modernos.

El control de calidad generalmente se define como un modelo planeado y sistemático de todas las acciones necesarias para proporcionar la confianza de que el artículo o producto se ajuste a los requisitos técnicos establecidos.

Las actividades de control de calidad pueden ser resumidas en :

- Establecimiento de estándares de calidad
- Estructuración de la concordancia de los estándares
- Acción cuando se sobrepasan los estándares
- Proyecto para el mejoramiento de los estándares

El control de calidad es un auxiliar, no un sustituto de los trabajos de diseño, ni de las actividades de inspección siempre requeridas para producir artículos de alta calidad.

Los beneficios que resultan de los programas de control de calidad son:

- Mejor calidad de diseño y producto.
- Reducción del costo de operación.
- Reducción de pérdidas.
- Mejoramiento de la moral del trabajador
- Reducción de tropiezos en la línea de producción.

Además se obtienen beneficios secundarios como :

- Mejoría en métodos de inspección.
- Establecimiento racional de estándares de tiempo en la mano de obra.
- Programas preventivos definidos para el mantenimiento.
- Oportunidad para acumular datos importantes para el uso en la propaganda y la adquisición de bases de actualidad para el cálculo de costos debido a desperdicios, reproceso e inspección.

1.3 SOFTWARE.

El Software son los programas, procedimientos, reglas y posible documentación asociada y datos pertenecientes a la operación de un sistema de computadora.

El software tiene las siguientes características:

- Estructura con lógica y propiedades funcionales
- Creados y mantenidos dentro de varias formas que se representan dentro de su ciclo de vida.
- Diseñadas y ajustadas para un proceso dentro de un completo estado de vida.

Existen dos formas básicas de software:

FORMA NO EJECUTABLE.

Los documentos como especificaciones de funciones que serán ejecutadas y sus entradas y salidas, listados de programación y diagramas lógicos de flujo de las funciones.

La información depositada dentro de dispositivos de la máquina como son discos, tarjetas, casets, cintas, etc..

FORMA EJECUTABLE.

Consiste de secuencias ejecutables de información accesable directamente, que ajusta los procesos de la máquina.

Dentro del software ejecutable que generalmente se maneja en una computadora encontramos:

SOFTWARE DE APLICACION.

Son desarrollados por el usuario y facilitan la interacción con la máquina. Satisfacen necesidades específicas del usuario.

SOFTWARE DEL SISTEMA OPERATIVO.

Hace posible la operación general del sistema, tiene previstas tareas como la localización de memoria, la planeación de trabajo de la unidad de procesos, la operación de los dispositivos de E/S y por último, dispone las prioridades de operación e interrupción.

1.4 CALIDAD DE SOFTWARE .

La calidad de software significa primordialmente un desarrollo de software capaz de producir datos de calidad para el usuario, que a su vez sea entendible, confiable, modificable y transportable.

Los once factores de calidad de software son:

Corrección
Integridad
Eficiencia
Utilización
Pruebas
Flexibilidad
Transportabilidad
Reutilización
Conectividad
Funcionalidad
Facilidad de Mantenimiento

Hay que tomar en cuenta que los requerimientos del programa son en realidad los requerimientos de calidad que el software debe presentar para su funcionalidad.

Las actividades descritas para la calidad de software y su medición son ejecutadas frecuentemente como actividades independientes que pueden darse de muchas maneras.

1.5 CERTIFICACION DE CALIDAD DE SOFTWARE

La certificación de calidad del software es la actividad que provee todas las evidencias necesarias para establecer que la función de calidad esta siendo ejecutada adecuadamente. Son actividades sistematicas que proveen evidencia de las características de uso del producto total de software.

1.6 MEDICION PARA DISTINTOS TIPOS DE SOFTWARE.

1. SISTEMAS DE MISION CRITICA

Son aquellos que se ejecutan en recursos de misión crítica, que incluyen computadoras y servicios para el conducto de misiones militares de Departamento de Defensa. En estos sistemas lo más importante es la confiabilidad.

2. SISTEMAS DE SOFTWARE

Un sistema de software es una colección de programas escritos para coordinar todos los circuitos de la computadora para que corra eficientemente. Un sistema de software es el sistema operativo el cual está compuesto de programas de control y programas de procesamiento.

El área para entocar el control de la calidad será el manejo de la configuración del software (SCM) y la documentación.

3. SISTEMAS EN TIEMPO REAL

La ejecución debe ser muy estricta, está restringida por constantes de tiempo, usualmente el software es complejo y existe como firmware, lenguaje ensamblador y microcódigo.

Lo principal a verificar es la eficiencia y para verificar la eficiencia se necesitan pruebas. El SCM es importante debido al tiempo de cada programa al ser unidos en un sistema.

A través de las pruebas, incrementando y controlando cercanamente los nuevos incrementos del software adicionado, la persona de certificación de calidad del software, puede calcular significativamente el éxito del sistema.

Deben considerarse también las revisiones, auditorías y documentación, sin lo cual, el sistema no estaría completo.

4. SISTEMAS INTERACTIVOS

Los sistemas interactivos son aquellos que manejan sistemas que requieren de una interface con un operador humano en tiempo real y no-real.

Algo importante es el entrenamiento del personal para manejar el sistema. El control del sistema es lo más importante, siendo secundario las revisiones, auditorías y documentación.

5. SISTEMAS DE NEGOCIOS

La principal aplicación es en el procesamiento electrónico de datos (EDP).

El trabajo para el responsable de la certificación de calidad del software es ejecutar una auditoría en el sistema EDP para determinar la adecuación de procedimientos, seguridad del sistema, documentación adecuada y verificar que el sistema funcione correctamente. Debido a que un alto porcentaje de sistemas de negocios están en fase de mantenimiento, la importancia de la documentación es muy alta.

ACTIVIDADES RELACIONADAS

6. ADMINISTRACION DE CONFIGURACION DE SOFTWARE (SCM)

Es una función interna dentro de una organización para identificar, controlar y evaluar el estatus del software y de la documentación.

La importancia de SCM es de la más alta prioridad cuando el sistema de software ha sido aprobado y vendido a un cliente. El software está sujeto a cambios y los cambios requeridos después de que un producto ha sido desarrollado tienen el más alto costo.

El personal de certificación de calidad debe asegurarse que cada nueva corrección resuelva un problema conocido y que el impacto del cambio en otras funciones del sistema esté registrada, probada y verificada para eliminar o minimizar cualquier efecto.

7. MANTENIMIENTO DE SOFTWARE

Los principales problemas son :

1. Adecuación de sistemas de aplicación diseñando especificaciones.
2. Calidad en programación original de sistemas de aplicación.
3. Calidad en la documentación de los sistemas de aplicación.

B. VERIFICACION INDEPENDIENTE Y VALIDACION

Tiene como actividades producir independientemente pruebas de documentos y correr pruebas de software producidos por otros. El personal de certificacion de calidad debe monitorear las pruebas y asegurarse de que cada prueba es repetida y que sigue el procedimiento de prueba.

TABLA DE MEDICION PARA LOS DISTINTOS TIPOS DE SOFTWARE.

TIPUS DE SOFTWARE	SCM	PRUEBAS	REVISIONES Y AUDITORIAS	DOCUMENTACION
1	1	4	3	2
2	3	1	2	2
3	2	1	3	4
4	1	2	3	4
5	4	3	1	2
ACTIVIDADES RELACIONADAS				
6	-	1	3	2
7	1	2	4	3
8	4	1	3	2
ESCALA		ENFASIS		
1		Mayor		
2				
3				
4		Menor		

TABLA1.1.

CAPITULO II

MARCO DE REFERENCIA PARA EL CONTROL DE CALIDAD DE SISTEMAS DE INFORMACION.

2.1 INTRODUCCION.

La clave de un grupo de certificación de calidad exitoso es definir y medir la calidad.

Se ha incrementado en los últimos años la conciencia acerca de los problemas críticos que se han encontrado en el desarrollo de sistemas de software de gran escala. Estos problemas no solo incluyen el costo y el esquema típico de los esfuerzos de desarrollo y la pobre ejecución de los sistemas una vez liberados, sino que también incluye el alto costo de mantenimiento, la falta de portabilidad y la alta sensibilidad de cambios en requerimientos.

Las métricas de software miden varios atributos de software y relacionan los diferentes aspectos de la calidad del software. El potencial de los conceptos de métricas de software pueden ser realizadas por su inclusión en programas de control de calidad. Proveen más disciplina, aproximación ingenieril de certificación de calidad y un mecanismo que permita un punto de vista de calidad de software. Los beneficios derivados de sus aplicaciones son realizadas en una reducción de costo del ciclo de vida.

Durante la pasada década, la evolución de prácticas de programación modernas, técnicas de desarrollo estructuradas y disciplinas y requerimientos para mayor estructuración y documentación efectiva, ha incrementado la factibilidad de la medición adecuada de la calidad de software.

Una estructura o modelo de calidad de software ha sido construido. Un modelo establecido esta descrito en la perspectiva de como puede ser usado para establecer los requerimientos de calidad de software, para una aplicación específica.

2.2 MODELO JERARQUICO DE CALIDAD DE SOFTWARE.

El vehículo para establecer los requerimientos de ejecución es un modelo jerárquico de calidad de software.

Este modelo tiene en su nivel más alto un conjunto de factores de calidad de software que son expresados como términos orientados a manejo de usuarios y representan las características que abarca la calidad de software.

El siguiente nivel considera para cada factor de calidad un conjunto de criterios que proveen las características representadas por cada factor. Estos criterios son términos relacionados sobre el software.

En el nivel más bajo están las métricas, las cuales son medidas cuantitativas de los atributos de software definidos por los criterios.

FACTOR	orientado a dirección
	vista de un producto
	de calidad
v	
CRITERIOS	orientado a software
	atributos los cuales
	proveen calidad
v	
METRICAS	medidas cuantitativas
	de estos atributos

Una vez que los requerimientos de calidad hayan sido determinados de acuerdo a los procedimientos descritos, deben ser transmitidos al equipo de desarrollo. Los requerimientos de calidad deben ser documentados de la misma forma que los otros requerimientos del sistema.

2.3 PROCEDIMIENTO PARA IDENTIFICAR LOS FACTORES DE CALIDAD

Las herramientas básicas serán la forma de reconocimiento de requerimientos de calidad de software que se muestra en la Tabla 2.1. Una reunión preliminar debe darse entre los responsables de las decisiones que usarán las tablas y figuras. Esto deberá hacerse sólo para medir los factores significantes para un proyecto.

Entre los responsables que tomen las decisiones deben encontrarse el gerente propietario del sistema, el usuario, el gerente de desarrollo, y el gerente de certificación de calidad.

2.4 CONSIDERACIONES DE LAS CARACTERISTICAS BASICAS DE LA APLICACION

Debe considerarse que los requerimientos de calidad para cada sistema son únicos y son influenciados por sistemas o características dependientes de la aplicación. Hay características básicas las cuales afectan los requerimientos de calidad; de esta manera cada sistema de software debe ser evaluado por sus características básicas. Tabla 2.2.

Por ejemplo, si el sistema está siendo desarrollado en un ambiente en el cual hay un alto porcentaje de adelantos técnicos en el diseño de hardware, la portabilidad tomará una gran significancia. Si el ciclo de vida esperado en un sistema es largo, el mantenimiento llega a ser de un considerable costo crítico. Si la aplicación es un sistema experimental donde las especificaciones de software tendrán un alto porcentaje de cambios, la flexibilidad en el producto del software es altamente deseable. Si las funciones de un sistema son esperadas para ser requeridas en un tiempo largo, mientras el propio sistema pueda ir cambiando considerablemente la reutilización será de primera importancia en estos módulos los cuales implementan la mayoría de las funciones del sistema. Con el advenimiento de más capacidades de comunicación y redes de computadoras, los sistemas están incrementando sus requerimientos de interface con otros sistemas y el concepto de interoperabilidad es extremadamente importante. Estas y otras características de sistemas deben de ser consideradas para la identificación de los factores de calidad importantes.

1. FAVOR DE INDICAR COMO SE CONSIDERA CADA UNO DE LOS FACTORES SIGUIENTES: SIENDO MUY IMPORTANTE (M), IMPORTANTE (I), ALGO IMPORTANTE (A) O NO IMPORTANTE (N) DEPENDIENDO DE LOS REQUISITOS DISEÑADOS PARA SISTEMA EN EL QUE ACTUALMENTE SE ENCUENTRE TRABAJANDO.

OBJETIVO

FACTOR

CORRECCION
 INTEGRIDAD
 EFICIENCIA
 UTILIZACION
 PRUEBAS
 FLEXIBILIDAD
 TRANSPORTABILIDAD
 REUTILIZACION
 COELECTIVIDAD
 FUNCIONALIDAD
 MANTENIMIENTO

2. EN QUE TIPO(S) DE APLICACION ESTA USTED ACTUALMENTE ENVUELTO?

3. ESTA USTED ACTUALMENTE EN :

- 1. FASE DE DESARROLLO
 ----- 2. FASE DE OPERACION/MANTENIMIENTO

4. POR FAVOR, INDIQUE EL PUESTO QUE MEJOR DESCRIBA SU POSICION

- 1. GERENTE DE PROGRAMA
 ----- 2. CONSULTOR TECNICO
 ----- 3. ANALISTA DE SISTEMA
 ----- 4. OTRO (FAVOR DE ESPECIFICAR) -----

FORMA DE RECONOCIMIENTO DE REQUERIMIENTOS DE
 FACTORES DE CALIDAD

TABLA 2-1

CARACTERISTICAS

- * SI LAS VIDAS HUMANAS SON AFECTADAS
- * LARGO CICLO DE VIDA
- * SOBRE UNA APLICACION EN COMPUTADORA
- * PROCESOS DE INFORMACION CLASIFICADA
- * SISTEMAS INTERRELACIONADOS

FACTORES DE CALIDAD

- INTEGRIDAD
- CORRECCION
- PRUEBAS
- MANTENIMIENTO
- FLEXIBILIDAD
- TRANSPORTABILIDAD
- EFICIENCIA
- INTEGRIDAD
- CORRECCION
- EFICIENCIA
- INTEGRIDAD
- CORRECCION
- INTEGRIDAD
- COELECTIVIDAD

CARACTERISTICAS DEL SISTEMA Y FACTORES DE CALIDAD RELACIONADOS**TABLA 2 - 2**

2.5 FACTORES DE CALIDAD.

- Corrección** : Grado en el cual un programa satisface sus especificaciones y cumple los objetivos del usuario.
- Integridad** : Grado en el cual el acceso al software o datos por personas no autorizadas puede ser controlado.
- Eficiencia** : Es la manera con la cual un programa o aplicación ejecuta sus funciones sin desperdiciar recursos de máquina como: memoria, espacio en disco, y con un adecuado comportamiento en el tiempo de respuesta.
- Utilización** : Esfuerzo requerido para aprender a operar, preparar entradas e interpretar salidas de un programa.
- Pruebas** : Esfuerzo requerido para probar un programa asegurando la ejecución de la función intentada.
- Flexibilidad** : Esfuerzo requerido para modificar un programa operacional.
- Transportabilidad** : Esfuerzo requerido para transferir un programa en una configuración de hardware y/o sistema de software a otro.
- Reutilización**: Grado en el cual un programa puede ser usado en otras aplicaciones.
- Conectividad**: Grado requerido para acoplar un sistema a otro.
- Funcionalidad**: Capacidad con la que un programa o aplicación ejecuta correctamente sus funciones planteadas por el usuario e interpretadas por sus diseñadores.
- Facilidad de Mantenimiento** : Simplicidad con la cual un programa o aplicación (en operación) pueda ser entendido, modificado, probado e implementado nuevamente en operación.

2.6 CONSIDERACION DE LAS IMPLICACIONES DEL CICLO DE VIDA

Los once factores de calidad identificados en la evolución pueden agruparse de acuerdo a tres actividades del ciclo de vida asociadas con un producto de software liberado.

Estas tres actividades son operación del producto, revisión del producto y transición del producto. La relación de los factores de calidad a estas actividades se muestra en la Tabla 2.3.

Esta tabla también muestra donde las indicaciones de calidad pueden ser realizadas a través de mediciones y donde el impacto es sentido si una pobre calidad es realizada. El tamaño del impacto positivo determina el costo que puede ser esperado si un sistema de muy alta calidad es realizado a través de la aplicación de métricas.

Este costo económico es un tanto compensado por el costo de la aplicación de las métricas y el costo de desarrollar un producto de software de muy alta calidad.

El costo de implementación contra la relación de reducción del costo de ciclo de vida, existe para cada factor de calidad. El beneficio contra el costo de proveer porcentaje para cada factor es medido como alto, mediano o bajo en la columna de la derecha de la Tabla 2.3. Esta relación y las implicaciones en el ciclo de vida de los factores de calidad deben ser considerados para la selección de los factores importantes para un sistema específico.

Como resultado de los dos primeros pasos, una lista tentativa de factores de calidad debe ser producida. El siguiente paso es considerar las interrelaciones entre los factores seleccionados. La tabla 2.4 puede ser usada como guía para determinar las interrelaciones.

2.7 IDENTIFICACION DE CRITERIOS.

El siguiente nivel de requerimientos de calidad envuelve procedimientos de los factores de calidad con los criterios. El conjunto de criterios, que son atributos del software, está relacionado a varios factores por definición.

Su identificación es automática y representan una especificación más detallada de los requerimientos de calidad.

CICLO DE VIDA FASES FACTORES	DESARROLLO			EVALUACION	POST-DESARROLLO			COSTOS ESPERA SALVADOS CONT COSTOS PARA PROVEER
	ANALISIS REQUERIM.	DISEÑO	CODIGO Y DEBUG	PRUEBAS	OPERACION	REVISION	TRANSICION	
CORRECCION	**	**	**	+	+	+		ALTA
FUNCIONALIDAD	**	**	**	+	+	+		ALTA
EFICIENCIA	**	**	**		+			BAJA
INTEGRIDAD	**	**	**		+			BAJA
UTILIZACION	**	**		+		+		MEDIA
MANTENIMIENTO		**	**			+	+	ALTA
PRUEBAS		**	**	+		+	+	ALTA
FLEXIBILIDAD		**	**			+	+	MEDIA
TRANSPORTABILIDAD		**	**				+	MEDIA
REUTILIZACION		**	**				+	MEDIA
CONECTIVIDAD	**	**		+			+	BAJA

** : DONDE LOS FACTORES DE CALIDAD DEBEN SER MEDIDOS

+ : DONDE EL IMPACTO DE UNA POBRE CALIDAD ES REALIZADO

IMPACTO DE NO ESPECIFICAR O MEDIA LOS FACTORES DE CALIDAD

TABLA 2.3

FACTORES	FACTORES	FACTORES	FACTORES	FACTORES	FACTORES	FACTORES	FACTORES	FACTORES	FACTORES
CORRECCION		CORRECCION							
FUNCIONALIDAD	**	FUNCIONALIDAD							
EFICIENCIA		EFICIENCIA							
INTEGRIDAD		INTEGRIDAD	OP						
UTILIZACION	**	UTILIZACION	OP	**					
MANTENIMIENTO	**	MANTENIMIENTO	OP	**	**				
PRUEBA	**	PRUEBA	OP	**	**	**			
FLEXIBILIDAD	**	FLEXIBILIDAD	OP	OP	**	**	**		
TRANSPORTAB.		TRANSPORTABILIDAD	OP		**	**			
REUTILIZACION		REUTILIZACION	OP	OP	OP	**	**	**	
CONECTIVIDAD		CONECTIVIDAD	OP	OP			**		

EL UN ALTO GRADO DE CALIDAD SE PRESENTA POR FACTOR, QUE GRADO DE CALIDAD SE ESPERA DEL OTRO?

** = ALTO OP = BAJO BLANCO = SIN RELACION

RELACIONES ENTRE FACTORES DE CALIDAD

TABLA 2.4

2.8 DEFINICION DE CRITERIOS PARA CALIDAD DE SOFTWARE.

- Rastreo** : Proveen un hilo de los requerimientos de la implementacion con respecto al desarrollo especifico y el ambiente operacional.
- Satisfaccion** : Atributos del Software que proporcionan una de instrumentacion completa de las funciones requerimientos de las funciones requeridas por el usuario.
- Tolerancia a errores** : Atributos del Software que proporcionan continuidad de operacion bajo condiciones anormales.
- Consumo de recursos** : Atributos del Software relativo a sus requerimientos de procesador, I/O, memoria, disco e impresion, al ser ejecutado en el equipo donde va a entrar a produccion.
- Comportamiento**: Atributos del Software relativos a sus características de respuesta para los volúmenes de procesos esperados.
- Operatividad** : Atributos del Software que determinan la facilidad o dificultad de operacion del sistema.
- Estandarizacion**: Atributos del Software que proporcionan técnicas de analisis, diseño, desarrollo y documentacion uniformes.
- Complejidad** : Atributos del Software que determinan la facilidad o dificultad de entender el sistema, programa o modulo en lo que a su funcionamiento se refiere.
- Estructuración**: Atributos del Software que detrmnan su modulacion, facilidad de expansion y ejecucion ordenada de sus funciones.
- Documentacion**: Complemento textual del Software que proporciona informacion para su operacion (Manual de Operacion), uso (Manual del Usuario) y mantenimiento del sistema (Manual Técnico).
- Generalidades**: Atributos del Software que determinan el nivel de utilizacion de funciones o Bases de Datos definidas, para uso compartido de los usuarios.
- Facilidad de prueba** : Atributos del Software que determina la facilidad o dificultad de ejecucion de sistema, con la finalidad de verificar su funcionamiento.

2.9 DEFINICION DE METRICAS.

La actual medición de calidad de software es realizada por aplicación de métricas a la documentación y código fuente producido durante el desarrollo de software.

2.10 CLASIFICACION DE LA METRICAS.

Las métricas pueden ser clasificadas de acuerdo a tres categorías:

- Métricas de detección de anomalías: identifica deficiencias en documentación o código fuente. Estas deficiencias usualmente son corregidas para mejorar la calidad del producto. El cumplimiento forzoso de estándares es una forma de estas métricas.

- Métricas de predicción: son medidas de la lógica del diseño e implementación. Son concernientes a formas, estructura, densidad y complejidad. Proveen una indicación de la calidad que pueden ser realizadas en el producto final, basados en la naturaleza de la aplicación y estrategias de diseño e implementación.

- Métricas de aprobación: son aplicadas al producto final para avalar su conformidad final con los requerimientos. Las pruebas son una forma de estas métricas.

Las métricas de detección de anomalías predictivas son aplicadas durante las fases de desarrollo para identificar los problemas de calidad tempranamente de tal manera que las acciones correctivas puedan realizarse cuando sean más efectivas y económicas.

El propósito de la aplicación de las métricas de calidad de software en un programa de certificación de calidad, es mejorar la calidad del producto.

2.11 METRICAS DE SOFTWARE.

Entre las características que pueden ser medidas en un sistema de información se tienen: el tamaño en líneas de código, el costo de desarrollo, el tiempo para el desarrollo en días de trabajo, el tamaño de memoria requerida en bytes y el número de clientes complacidos después de la entrega.

Sin embargo, diferentes observadores del mismo producto de software pueden obtener diferentes medidas, siendo que se esta midiendo la misma propiedad.

Las metricas de software pueden clasificarse de la siguiente manera:

- Métricas de tamaño
- Métricas de estructura de datos
- Métricas de estructura lógica
- Métricas de composición
- Métricas de diseño

2.11.1 METRICAS DE TAMAÑO

Hay una característica que todos los programas tienen: Tamaño.

Existen muchas posibilidades para representar el tamaño o magnitud de un programa, siendo las más importantes, por su utilidad en la caracterización del software, las líneas de código, número de tokens, número de funciones y otras medidas basadas en estas básicas.

El tamaño de un programa es una medida importante primordialmente por estas razones:

- Es sencillo de contabilizar después de que el programa este terminado.
- Es el factor más importante para muchos modelos de desarrollo de software.
- La productividad esta normalmente basado en una medida del tamaño.

Los listados de programas generados por muchos compiladores y ensambladores tienen un número serial impreso para cada línea de programa. Esta característica es provista para tener una rápida manera de encontrar líneas de programas individuales en caso de tener errores en el programa. Una ventaja adicional, es que, después de que un programa este compilado, un programador inmediatamente conoce el número total de líneas en el programa.

La cantidad de esfuerzo necesario para construir un programa depende del número de líneas que están escritas. Sin embargo, la medida de las líneas de código puede no ser satisfactoria completamente debido a que algunas líneas en un programa son más difíciles de producir que otras líneas en el mismo programa.

LINEAS DE CODIGO.

La medida de software más familiar es la cuenta de líneas de código. Se representa por el símbolo Ss, cuya unidad es LOC. Para programas largos es más apropiado contar con miles de líneas de código (KLOC), representadas por S.

Muchos desarrolladores, están de acuerdo que para medir las líneas de código no se deben incluir comentarios o líneas en blanco, ya que estas son documentación interna y no afectan las funciones del programa. Además la inclusión de estas líneas provoca una falsa ilusión de alta productividad, la cual es medida en LOC/PC (líneas de código / persona/mes).

Algunos lenguajes de programación de formato libre permiten dos o más instrucciones en una línea, o que una instrucción sea extendida a dos o más líneas. Para solucionar esta inconsistencia, se determinó la definición estándar de líneas de código:

" Una línea de código es cualquier línea de texto del programa que no es un comentario o una línea en blanco, indiferente del número de instrucciones o fragmentos de instrucciones en la línea. Esto específicamente incluye todas las líneas conteniendo encabezados de programas, declaraciones de variables e instrucciones ejecutables y no ejecutables. "

Bajo este criterio, el siguiente programa cuenta con 14 líneas de código:

Línea		Nivel
1	SUBROUTINE SORT (X,N)	
2	INTEGER X(100),N,I,J,SAVE,IM1	
3	C ESTA SUBROUTINA ORDENA EL ARREGLO X	
4	C EN ORDEN ASCENDENTE	
5	IF (N.LT.2) GO TO 200	1
6	DO 210 I=2,N	2
7	IM1=I-1	3
8	DO 220 J=1,IM1	3
9	IF (X(I).GE.X(J)) GO TO 220	4
10	SAVE=X(I)	5
11	X(I)=X(J)	5
12	X(J)=SAVE	5
13	220 CONTINUE	4
14	210 CONTINUE	3
15	200 RETURN	2
16	END	1

FIGURA 2.1

CONTADOR DE TOKEN.

La medida de S_s no es consistente debido a que algunas líneas son más difíciles de codificar que otras. Una solución es darle más peso a líneas que tienen más estudio en ellas.

Un esquema natural de peso para solucionar este problema es utilizar el número de tokens, que son unidades sintácticas básicas distinguibles por un compilador. Este esquema fue utilizado por Halstead en su familia de métricas comúnmente llamada Ciencia del Software (Software Science).

Un programa de computadora es considerado como una colección de tokens que pueden ser clasificados como operadores y operandos. Esto está basado en el hecho de que todos los programas pueden ser reducidos a una secuencia de instrucciones de lenguaje de máquina, cada una de las cuales contiene un operador y un número de direcciones de operandos. Las métricas básicas están definidas como :

n^1 = número de operadores únicos

n^2 = número de operandos únicos

N^1 = total de ocurrencias de operadores

N^2 = total de ocurrencias de operandos

Generalmente cualquier símbolo en un programa que especifica una acción es considerado un operador, mientras que un símbolo utilizado para representar un dato es considerado un operando. Los operadores consisten de símbolos aritméticos, nombres de comandos, marcas de puntuación, símbolos especiales y nombres de funciones. Los operandos son las variables, constantes y niveles.

El tamaño de un programa en términos del número total de tokens utilizados es:

$$N = N^1 + N^2$$

Para la clasificación de operadores y operandos existen variaciones. Las reglas originales establecidas por Halstead excluyen el conteo de las instrucciones de declaración de variables y/o constantes, y las instrucciones de I/O. Los niveles de instrucciones no fueron contados como operandos, más bien fueron considerados una parte de transferencias directas.

Actualmente, las investigaciones tienden hacia el conteo de tokens en instrucciones de declaración de variables y de I/O, y muchos cuentan los niveles de instrucciones como operadores donde sea que se encuentren.

Esta técnica se ha empleado para producir el conteo en la siguiente tabla, en base a la Figura 2.1 :

OPERADORES	OCURRENCIAS	OPERANDOS	OCURRENCIAS
SUBROUTINE	1	SORT	1
()	10	X	8
.	8	N	4
INTEGER	1	100	1
IF	2	I	6
.LT.	1	J	5
GO TO	2	SAVE	3
DO	2	IM1	3
=	6	2	2
-	1	200	2
.GE.	1	210	2
CONTINUE	2	1	2
RETURN	1	220	3
END OF LINE	13		
$n^1=14$	$N^1=51$	$n^2=13$	$N^2=42$

FIGURA 2.2

Las reglas usadas son dependientes del lenguaje. Además, ocurren ambigüedades en el conteo de operandos y operadores.

En el Anexo 1, se encuentran el conjunto de reglas de conteo para el lenguaje Pascal, tal como fue implementado en el analizador de software distribuido por el Purdue University Software Metrics Research Group.

Nótese que en la figura 2.2, $N^1=51$ y $N^2=42$ y la medida llamada "longitud" por Halstead es :

$$N = N^1 + N^2 = 93$$

La métrica N puede ser convertida a un estimador de $S^{\#}$ vía la relación $S^{\#}=N/c$, donde la constante c es dependiente del lenguaje. Para Fortran, c equivale a un 7, por lo que :

$$S^{\#} = 93/7 = 13$$

el cual es cercano al actual $S^{\#}$ obtenido para la Figura 2.1.

La Ciencia del Software define métricas adicionales usando los términos básicos. Y son :

Vocabulario.

Esta definido por $n = n^1 + n^2$.

Acentúa el hecho de que teniendo un vocabulario de programación consistente de sólo estos operandos y operadores, se puede construir exitosamente este programa.

Volúmen.

Definido por $V = N \times \log^2 n$.

La unidad de medida es el bit, la unidad común para mediciones del tamaño actual del programa en una computadora, si el código binario uniforme es utilizado. Para la figura 2.1, $N=93$ y $n = 27$. Si el esquema de número binario es utilizado para representar cada uno de los 27 items del vocabulario, se tomarán 5 bits por item, ya que un esquema menor no genera los códigos necesarios para representarlos. Cada uno de los 93 tokens utilizados pueden ser representados en un código de 5 bits, generándose un string de $5 \times 93 = 465$ bits, que permitirán almacenar el programa entero en memoria.

Sin embargo, ya que el código de 5 bits permite 32 tokens, en vez de los 27 necesarios, en lugar de utilizar el entero 5, se utiliza el $\log^2 = 4.75$, para obtener un volumen ligeramente más pequeño, de 442 bits.

CONTADOR DE FUNCIONES.

Para programas largos, es más fácil predecir el número de módulos que de líneas de código. Sin embargo, es necesario definir estrictamente el modo en que un programa será dividido en módulos.

Una función en un programa está definida como una colección de instrucciones ejecutables que realizan cierta tarea, junto con declaraciones de parámetros formales y variables locales, manipuladas por estas instrucciones.

Un módulo consiste de una o más funciones. Una función puede computar un valor aritmético sencillo, puede cambiar estructuras de datos o archivos. El conteo de funciones en un programa puede no ser fácil de obtener a menos que el programador haya construido cada función como un módulo separado.

El número de líneas de código para cada función usualmente no deberá ser muy grande, debido al límite de la capacidad mental humana: un programador no puede manipular eficientemente información cuando la cantidad es mayor al límite.

Algunos autores definen que cada tamaño de módulo debe limitarse de 50 - 200 LOC, argumentando que esos tamaños incrementan el entendimiento y minimizan errores.

MEDIDAS DE TAMAÑO EQUIVALENTES.

No siempre se desarrolla software nuevo. De hecho, una buena parte del trabajo de programación envuelve modificar código existente.

En base a esto, para muchos programas, el tamaño tiene 2 componentes: S^n para código escrito nuevo y S^u para código adaptado de software existente.

La medida de tamaño equivalente está en función de S^n y S^u . Esto significa que el esfuerzo requerido para desarrollar el software con S^n y S^u es "equivalente" al esfuerzo requerido para desarrollar un producto con S^e "desde el principio" (sólo código nuevo sin código reutilizado).

Se considera que el código reutilizado incluye código revisado, en cuanto a que al utilizar código existente, tal vez sea necesario realizar modificaciones para que se adapte al resto del programa.

Una función propuesta por Boehm para su modelo de estimación de costos COCOMO es :

$$S^e = S^n + \frac{a}{100} S^u$$

y

$$a = 0.4 DM + 0.3 CM + 0.3 IM$$

donde :

DM : Porcentaje de modificación requerido en el diseño

CM : Código

IM : Esfuerzo necesario para la integración del código modificado.

El máximo valor de a es 100, el cual corresponde al caso en el cual es igualmente difícil adaptar código usado que reescribirlo. No se tiene ningún registro de su exitosa aplicación excepto en los estudios originales de COCOMO.

Una fórmula similar fue propuesta por Bailey y Basili:

$$S^e = S^n + K S^u$$

Un valor de $K=0.2$ para todos los programas se encontró razonable para los proyectos estudiados. Sin embargo, un valor fijo para K probablemente no sea realista y deberá considerarse sólo como una primera aproximación.

Thebaut supuso que la contribución del código adaptado es no lineal, proponiendo la siguiente fórmula :

$$S^e = S^n + S^u$$

donde k es una constante no mayor a 1, cuyo valor, utilizando análisis de regresión es $k=6/7$.

2.11.2 METRICAS DE ESTRUCTURAS DE DATOS.

Un importante conjunto de métricas son aquellas que capturan la cantidad de datos de entrada, de proceso y salida de software.

CANTIDAD DE DATOS

Muchos compiladores y ensambladores tienen la opción de generar una lista de referencias cruzadas, indicando la línea donde una cierta variable es declarada y la línea o líneas donde es referenciada.

Un método para determinar la cantidad de datos es contar el número de entradas en la lista de referencias cruzadas. Se deben excluir las variables que están definidas pero nunca son utilizadas. Probablemente estén definidas para futuras referencias, pero no afectan las características operacionales o la dificultad de desarrollo. El contador de variables será referido como VARS.

```
1  Program payday (input,output),
2  type check = record
3      gross : real;
4      tax : real;
5      net : real;
6  end;
7  var pay: check;
8      hours, rate : real;
9  begin
10     while not eof(input) do begin
11         readln(hours,rate);
12         pay.gross := hours * rate;
13         pay.tax := 0.25 * pay.gross;
14         pay.net := pay.gross - pay.tax;
15         writeln(pay.gross, pay.tax, pay.net)
16     end
17 end.
```

FIGURA 2.3

check	2	7			
gross	3	12	13	14	15
hours	8	11	12		
net	5	14	15		
pay	7	12	13	13	14
	14	14	15	15	15
rate	8	11	12		
tax	4	13	14	15	

Lista de referencia cruzada

FIGURA 2.4

Para la Figura 2.3, VARS = 7, para la Figura 2.1, VARS = 6 (Teniéndose X, N, I, J, SAVE E IM1 como variables.)

El conteo de variables depende de la siguiente definición:

"Una variable es un string de caracteres alfanuméricos que esta definido por el programador y que es utilizado para representar algún valor durante, ya sea, la compilación o la ejecución"

El contador de variables también puede ser generado utilizando un analizador de software que cuente los tokens utilizados.

Existen algunos items de la lista que fueron borrados:

eof	10	
input	1	10
output	1	
payday	1	

FIGURA 2.5

Los items EOF, INPUT, OUTPUT son relativos a I/O. El nombre del programa fue denominado variable, pero debido a que ninguna de estas son variables en el sentido de que fueran creadas para producir un programa, son borradas de la Figura 2.4.

Halstead introdujo una métrica que él definió como n^2 para ser un contador de operandos en un programa incluyendo variable, constantes y niveles.

$$n^2 = \text{VARS} + \text{constantes únicas} + \text{niveles}$$

El ejemplo del SORT (Figura 2.1), tiene 6 variables (X, N, I, J, SAVE, IM1), 3 constantes (1, 2, 100) y 4 niveles (SORT, 200, 210, 220), así que $n^2 = 13$.

El nombre de la subrutina es utilizado como nivel, ya que es el nivel que será usado por cualquier otro programa o subprograma que desee acceder SORT.

Para medir la cantidad de uso de los operandos Halstead definió la métrica "Total de ocurrencias de operandos" y la llamo N^2 . El programa de la Figura 2.3 utiliza 9 operandos 32 veces, algunos son usados diferentes veces, mientras otros, solo una vez. Así $N^2 = 32$ para este programa, mientras que para el programa de la Figura 2.1, $N^2 = 42$.

2.11.3 METRICAS DE ESTRUCTURAS LOGICAS

La estructura logica de un programa le permite ejecutar diferentes operaciones dependiendo de los datos de entrada y de cálculos intermedios.

Además del poder de almacenar y acceder grandes cantidades de datos, la computadora tiene la habilidad de probar los datos y tomar una acción dependiendo del resultado de la prueba.

Las pruebas ordenadas y las acciones asociadas forman la base de los programas de computadora y son llamados **Algoritmos**. La estructura de un algoritmo es representada por una gráfica dirigida llamada **Diagrama de Flujo**.

Las Figuras 2.6 y 2.7 son los diagramas de flujo de las Figuras 2.1 y 2.3, respectivamente.

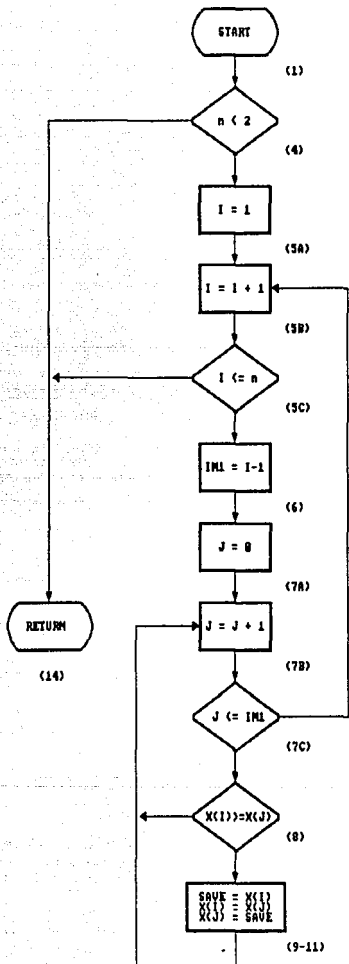


FIGURA 2.6
DIAGRAMA DE FLUJO 1

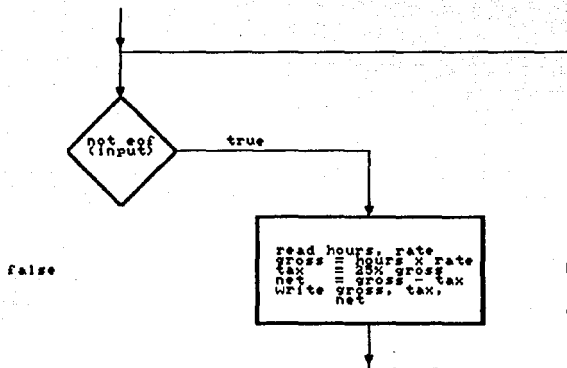


FIGURA 2.7 : DIAGRAMA DE FLUJO DE CONTROL PARA EL PROGRAMA DE LA FIGURA 2.3.

La parte del comportamiento del algoritmo con las pruebas realizadas y los saltos después de las pruebas es llamada la Estructura Lógica del Programa.

Los métodos de conteo más populares para estas métricas son:

CONTEO DE DECISIONES

El flujo de control en un programa, normalmente es secuencial. Es interrumpido en tres posibles situaciones:

Salto hacia adelante : Este sigue a una prueba condicional que lleva a una opción entre dos posibles acciones. (Figura 2.8)

Salto hacia atrás : Utilizado para crear ciclos, puede ser incondicional (Figura 2.9), puede seguir a una prueba condicional que permite seguir con la iteración o terminar el ciclo. (Figura 2.10)

Salto horizontal : Son las transferencias de control a un procedimiento o subrutina.

Una métrica sencilla es el contador de decisión (DE), que es el conteo de IF, DO, WHILE, CASE y otras instrucciones de control de ciclos y condicionales. Así en la Figura 2.1, DE = 4 y en la Figura 2.3, DE = 1.

Un programa con un DE mayor, generalmente es más complejo que otro programa con un DE más pequeño.

Muchos lenguajes de programación permiten la utilización de condiciones compuestas en IF's y otros condicionales e instrucciones de control de ciclos.

La instrucción

```
IF C1 AND C2 THEN S
```

es equivalente a :

```
IF C1 THEN IF C2 THEN S
```

o bien

```
IF C2 THEN IF C1 THEN S
```

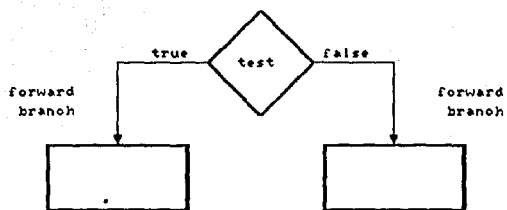



FIGURA 2.8 : PRUEBA CONDICIONAL QUE SELECCIONA UNA OPCION

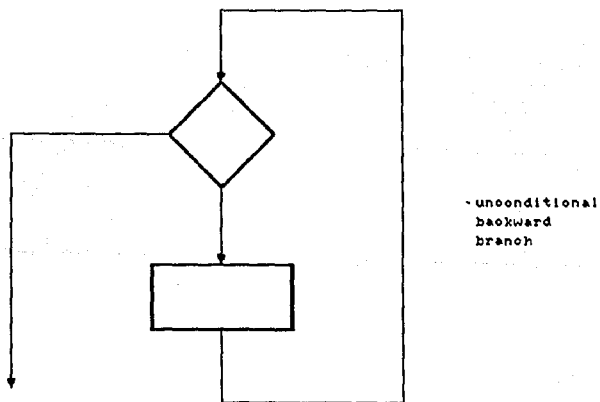
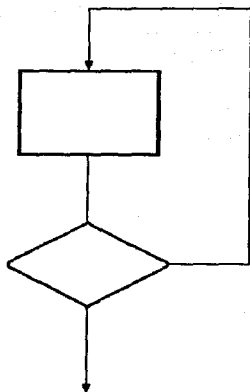


FIGURA 2.9 : SALTO HACIA ATRAS INCONDICIONAL UTILIZADO PARA CREAR CICLOS.



conditional
backward
branch

**FIGURA 2.18 : UN SALTO HACIA ATRAS SIGUIENDO
UNA PRUEBA CONDICIONAL**

La representación de esta instrucción varía en el diagrama de flujo, como puede verse en la Figura 2.11, sin embargo, para el conteo, es razonable que para una instrucción IF con dos condiciones simples, contribuya con 2 al contador de decisiones. Estas condiciones sencillas llamadas predicados, deben contarse como DE en un programa, en vez del número de ocurrencias de la palabra clave (en la mayoría de los lenguajes IF).

Similarmente una instrucción CASE es considerada una instrucción IF con múltiples predicados.

Una métrica más sofisticada y mejor conocida, basada en el número de decisiones es el número de complejidad ciclomática (V(G)) propuesta por McCabe.

El número de complejidad ciclomática está definido por :

$$v(G) = e - n + 2$$

donde : e es el número de flechas o líneas de dirección
n es el número de nodos o bloques de código

Para la Figura 2.12, si los rectángulos son instrucciones y los rombos son puntos de decisión, se tienen 13 nodos y 16 flechas en el diagrama de flujo. Por lo tanto:

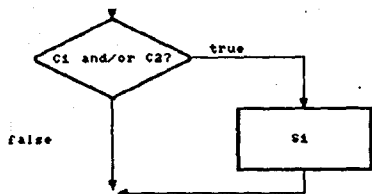
$$v(G) = 16 - 13 + 2 = 5$$

Una buena manera de visualizar el número de complejidad ciclomática es imaginar que representa el número de flechas que deben ser removidas para reducir el diagrama de flujo a su esqueleto, que es sin ciclos.

Un diagrama de flujo sencillo, que representa una secuencia lineal de código, tiene un $v(G) = 1$. Sin embargo, un conjunto lineal de nodos puede ser reducido a un simple nodo, sin que cambie su número ciclomático. Figura 2.13.

M McCabe concluyó que un $v(G) = 10$ es razonable.

(A)



(B)

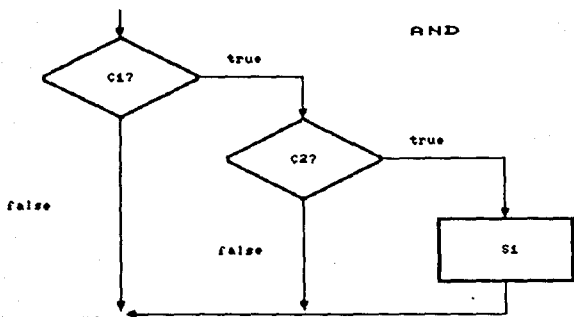


FIGURA 2.11. REPRESENTACION POR DIAGRAMA DE FLUJO DE UN CONDICIONAL COMPUESTO

< C >

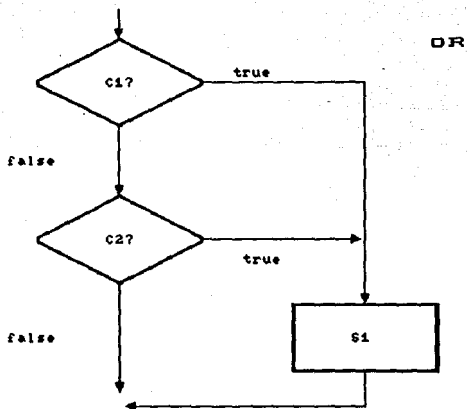


FIGURA 2.11 : REPRESENTACION POR DIAGRAMA DE FLUJO DE UN CONDICIONAL COMPUESTO

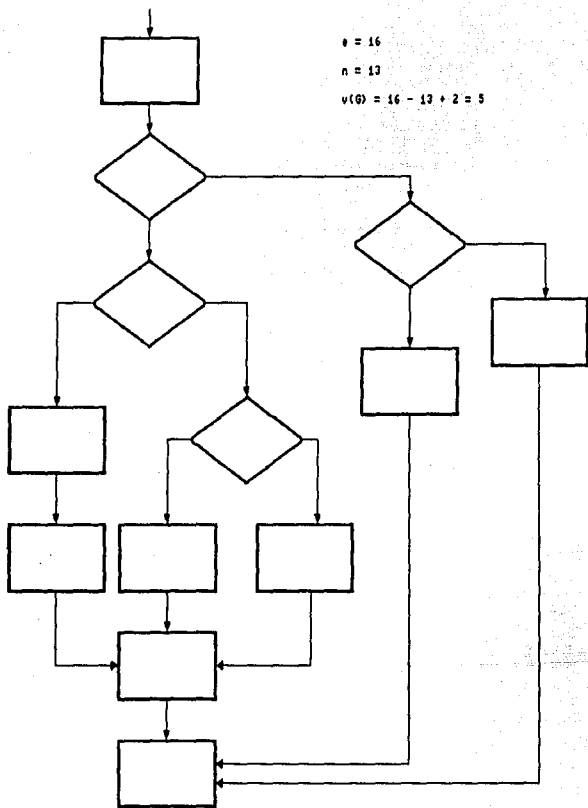
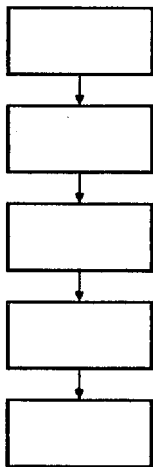


FIGURA 12 : DIAGRAMA DE FLUJO CON 13 NODOS Y 16 LINEAS DE DIRECCION

< A >

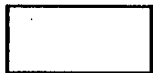


$$e = 4$$

$$n = 5$$

$$v(G) = 4 - 5 + 2 = 1$$

< B >



$$e = 0$$

$$n = 1$$

$$v(G) = 0 - 1 + 2 = 1$$

FIGURA 2.13 : DIAGRAMA DE FLUJO SIMPLE DE UNA SECUENCIA DE CODIGO LINEAL.

Basándose en el hecho, de que, la complejidad ciclomática sólo depende del número de flechas y nodos, y no de como están conectados, la relación entre flechas y nodos, $e = n - 1$, se mantendrá indiferente del número de nodos que se tengan, siempre y cuando no existan ciclos o saltos.

Debido a que cada condicional tiene dos flechas, la instrucción condicional incrementa e en 1 por cada condicional. Por ejemplo, si α es el número de predicados :

$$e = n - 1 + \alpha$$

o el equivalente

$$e - n = \alpha - 1$$

$$v(G) = e - n + 2 = \alpha + 1$$

Los predicados son equivalentes a DE, así que la fórmula puede quedar :

$$v(G) = DE + 1$$

Este concepto, puede ser fácilmente determinado por un analizador de software. Las reglas de conteo para el lenguaje Pascal, para determinar $v(G)$, se encuentran en el Anexo 1.

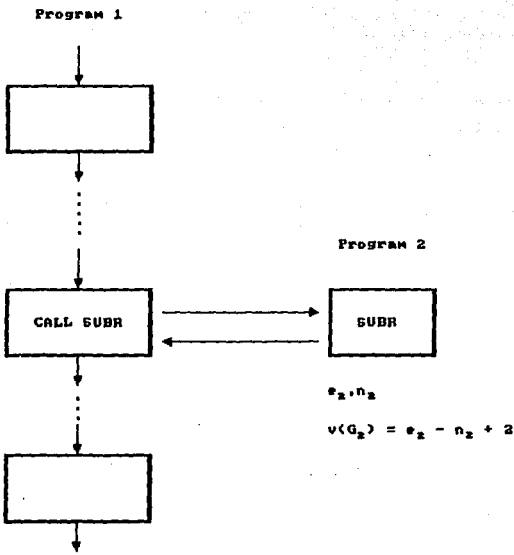
Considerando el diagrama de la Figura 2.6, que tiene $e=15$ y $n=12$, se obtiene que $v(G) = 5$, y teniendo cuatro predicados, representados por cuatro rombos, se tiene que $\alpha = DE = 4$, obteniéndose que $v(G) = 5$.

Un programa con cierto número de módulos tiene un diagrama de flujo con el mismo número de componentes conectados. Obsérvese la Figura 2.14.

El Programa 1 tiene e^1 flechas y n^1 nodos, si de uno de estos nodos es llamado el Programa 2 con e^2 flechas y n^2 nodos, se tiene :

$$v(G^1) = e^1 - n^1 + 2$$

$$v(G^2) = e^2 - n^2 + 2$$



e_1, n_1

$v(G_1) = e_1 - n_1 + 2$

$$\begin{aligned}
 v(G) &= (e_1 + e_2 + 2) - (n_1 + n_2) + 2 \\
 &= (e_1 - n_1 + 2) + (e_2 - n_2 + 2) \\
 &= v(G_1) + v(G_2)
 \end{aligned}$$

FIGURA 2.14 : PROGRAMA 1 LLAMA A PROGRAMA 2 (SUBROUTINA SUBR)

Al llamar al Programa 2 se adiciona una flecha de llamada y una de regreso. Así, para este programa, la complejidad ciclométrica, llega a ser :

$$v(G) = (e^1 + e^2 + 2) - (n^1 + n^2) + 2$$

$$v(G) = (e^1 - n^1 + 2) + (e^2 - n^2 + 2)$$

$$v(G) = v(G^1) + v(G^2)$$

Se obtiene entonces que la complejidad ciclométrica para un programa de multimódulos es la suma de los $v(G)$, para los módulos individuales.

$$v(G \text{ programa}) = \sum_{i=1}^m v(G^i) = \sum_{i=1}^m e^i - \sum_{i=1}^m n^i + 2m$$

$$v(G \text{ programa}) = \sum_{i=1}^m DE^i + m$$

MÉTRICAS DE EXTENSIÓN Y NÚMERO MÍNIMO DE TRAYECTORIAS.

Schneidewind y Hoffman definieron métricas para el número mínimo de trayectorias en un programa y la extensión de cualquier nodo.

Para determinar el número mínimo de trayectorias, N_p , se debe describir cada trayectoria, una secuencia única de arcos del nodo inicial al nodo final. Este análisis de trayectorias también sirve para determinar la extensión, R , el número de caminos únicos de secuencias de cada nodo.

Cuando existen ciclos, el número de trayectorias puede ser infinito. La determinación de N_p , excluyen trayectorias con ciclo hacia atrás atravesados más de una vez. Por ejemplo, en la Figura 2.6 existen cinco trayectorias únicas, por lo que $N_p = 5$.

1, 4, 14

1, 4, 5a, 5b, 5c, 14

1, 4, 5a, 5b, 5c, 6, 7a, 7b, 7c, 5b, 5c, 14

1, 4, 5a, 5b, 5c, 6, 7a, 7b, 7c, 8, 7b, 7c, 5b, 5c, 14

1, 4, 5a, 5b, 5c, 6, 7a, 7b, 7c, 8, 9-11, 7b, 7c, 5b, 5c, 14

Esta técnica tiende a complicarse cuando los programas son mas grandes.

La extensión promedio (R) es contabilizada del total de trayectorias dividida entre el numero de nodos.

$$R = \frac{26}{12}$$

NIVELES ANIDADOS

La anidación es un método de programación muy útil ya que permite al programador prevenir instrucciones con excesivos condicionales compuestos en IF's o WHILE's, tomando antecedentes de condiciones efectuadas antes del IF o WHILE.

Sin embargo, esto también debe ser moderado para no dificultar la comprensión de la secuencia.

Línea		Nivel
1	for i:= 1 to n-1 do begin	1
2	if x[i] > 0 then begin	2
3	while x[i] > x[i+1] do begin	3
4	if x[i] >= decr then begin	4
5	x[i] := x[i] - decr	5
6	end	
7	end	
8	end	
9	end;	

FIGURA 2.15

Así, otra métrica es la profundidad de anidación. Una instrucción simple en una parte de secuencial, como la línea 5 de la Figura 2.1, puede ser ejecutada sólo una vez, mientras que la línea 9 es ejecutada en el orden de n^2 , donde n es el tamaño del arreglo y es almacenado en la variable N , debido a que esta dentro de un ciclo.

Para la más alta profundidad o nivel de anidación, lo más difícil es evaluar las condiciones de entrada para ciertas instrucciones.

Para calcular el nivel de anidación promedio (\overline{NL}), cada instrucción ejecutable en un programa debe ser asignado a un nivel de anidación. Un procedimiento sencillo para esto es:

1. La primera instrucción ejecutable esta asignada al nivel de anidación 1.
2. Si la instrucción A es un nivel 1 y la instrucción B sigue secuencialmente la ejecución de la instrucción A, el nivel de anidación para la instrucción B también es 1.
3. Si la instrucción A es un nivel 1 y la instrucción B está en el rango de un ciclo o una transferencia condicional gobernada por la instrucción A, entonces el nivel de anidación de la instrucción B es $1+1$.

Para determinar el nivel de anidación promedio, se suman todos los niveles de anidación y se divide la suma entre el número de instrucciones.

Para el programa de la Figura 2.15 con 11 instrucciones ejecutables, la suma es 34 y el nivel de anidación promedio es 3.1.

2.11.4 METRICAS DE COMPOSICION

Una métrica simple no es capaz de representar adecuadamente la complejidad del software.

Las métricas anteriores fueron diseñadas para medir diferentes características de un programa. Un aprovechamiento correcto de estas métricas, es considerar un vector de las mismas.

Por ejemplo, podemos utilizar un par de números (a,b), de tal manera que a y b sean métricas. Un ordenamiento puede entonces ser formado: el programa P₁ con las métricas (a₁, b₁) es más complejo que el programa P₂ con las métricas (a₂, b₂), si a₁ > a₂ y b₁ > b₂ (mayor mientras uno de los pares es "mayor que" o "mayor o igual"). Usando esta técnica, se pueden ordenar los programas en la Tabla 2.6 del menos al más complejo, si a es la métrica de tamaño S_g y b es la métrica de la estructura lógica v(g).

COMPLEJIDAD	S _g	V(g)
Menor	300	7
!	320	9
!	320	11
!	350	11
v	400	15
Mayor	500	16

Vector de métricas

TABLA 2.5

Myers, considerando la complejidad ciclométrica, reconoce la inconsistencia potencial entre el conteo de instrucciones IF y el conteo de predicados. Propone que una métrica más razonable para la estructura lógica es el par (CYC-min, CYC-max) donde :

CYC-min = conteo de todos los condicionales y loops incluyendo instrucciones CASE.

CYC-mid = CYC-min + número igual a dos menos que el número de selecciones en una instrucción CASE.

CYC-max = CYC-mid + cada operador lógico (AND u OR) en una instrucción condicional.

Sin embargo, las métricas compuestas incrementan la confusión, ya que las relaciones entre programas no cubren el orden parcial.

En la Tabla 2.5 no es muy claro cuál de los dos programas es más complejo, si $a_1 > a_2$ pero $b_1 < b_2$.

La confusión aumenta si más métricas son incluidas en el vector. Debido a esto, muchos desarrolladores no utilizan estas métricas.

2.11.4.1 Métricas compuestas de la Ciencia del Software

Estas son una colección de métricas compuestas por Halstead en 1970. Ya se han discutido con anterioridad las métricas n_1 , n_2 , N_1 , N_2 y V . Las cuatro primeras son contadores básicos de tokens y V , más las restantes métricas, son funciones de estas cuatro básicas. El modelo diseñado fue muy simple: Un programa de N_1 operadores y N_2 operandos es construido seleccionando entre n_1 operadores únicos y n_2 operandos únicos.

Longitud del programa estimado

Una medida del tamaño de un programa es el total del número de tokens, referido como la longitud N , el cual es la suma total de operandos y operadores usados ($N = N_1 + N_2$).

La primera hipótesis de la Ciencia del Software es que la longitud de un programa bien estructurado es función solo del número de operadores y operandos únicos. Esta función llamada ecuación de longitud se define como:

$$N = n_1 * \log_2 n_1 + n_2 * \log_2 n_2$$

Por ejemplo, el programa de la figura 2.1, tiene 14 operadores únicos y 13 operandos únicos, como se muestra en la figura 2.2.

Suponiendo que estos números son conocidos, antes de completar el programa, posiblemente usando un lenguaje de diseño de programa y conociendo el lenguaje de programación escogido, es posible estimar la longitud N del programa en el número de tokens usando la ecuación anterior:

$$\begin{aligned}N &= 14 * \log_2 14 + 13 * \log_2 13 \\N &= 14 * 3.81 + 13 * 3.70 \\N &= 53.34 + 48.11 \\N &= 101.45\end{aligned}$$

Se predice que habrá cerca de 101 tokens usados en el programa completo, un 10% mas de lo obtenido en la figura 2.2

La ecuación de longitud puede ser vista como una relación que explica la ejecución "típica", pero falla en precisar otros ejemplos.

Volúmen del programa

El modelo de Halstead provee otra interpretación de la medida de volúmen dada anteriormente como métrica de tamaño.

Halstead sugiere :

$$V = N * \log_2 n$$

La cual puede ser interpretada como el número de comparaciones mentales necesarias para escribir un programa de longitud N. Esta interpretación asume, que durante la programación, la mente humana sigue un proceso de "búsqueda binaria" para seleccionar el siguiente token de un vocabulario de tamaño n. en el programa de la figura 2.2, el sort, nótese que $n=14+13 = 27$ y $N = 93$, así que :

$$V = 93 * \log_2 27$$

$$V = 93 * 4.75$$

$$V = 442$$

La Ciencia del Software sugiere que 442 comparaciones mentales deben hacerse en orden para escribir el programa del sort. Deben seleccionarse 93 tokens cada vez de un campo de 27 posibilidades usando una técnica de búsqueda binaria. Algunos programadores pueden sentir que no programan de esta manera y que la programación es una tarea más sofisticada que solo determinar el siguiente token a usar de un conjunto disponible. Por esta razón algunos desarrolladores encuentran la interpretación del volumen de esta manera, cuestionable.

Volúmen potencial y dificultad

Un algoritmo puede ser implementado por muy diferentes pero equivalentes programas. Entre estos programas el que tiene menor tamaño se dice que tiene el volumen potencial V^* . Halstead arguye que la implementación mínima de cualquier algoritmo fue a través de una referencia a un procedimiento previamente escrito. La implementación de este algoritmo requerirá nada más que invocar al procedimiento y suplir los operandos por sus parámetros de entrada y salida.

$$V^* = (2 + n_2^*) \log_2 (2 + n_2^*)$$

El primer término en el parentesis, 2, representa los dos operadores únicos para el procedimiento: call el nombre del procedimiento y un símbolo de agrupamiento que separa el nombre del procedimiento de sus parámetros.

n_2^* representa el número de parámetros de entrada y salida únicos, conceptualmente.

Para el sort de la figura 2.2, los parámetros son:
 x - el arreglo que mantiene los enteros a ser ordenados
 N - el tamaño de un arreglo a ser ordenado

$n_2^* = 3$, ya que x es usado como entrada y salida. Así el volúmen potencial V^* para el programa del ordenamiento es :

$$V^* = 5 \log_2 5 = 11.6$$

Mientras n_2^* puede ser determinado, probablemente para programas de aplicaciones pequeñas, es más difícil de computar para programas de una longitud muy grandes, como un compilador o un sistema operativo.

En la Ciencia del Software, cualquier programa con volumen V esta considerado para ser implementado en el Nivel de programa L, definido por:

$$L = V^* / V$$

El valor de L de un programa es máximo 1, con L = 1, representando un programa escrito en el nivel más alto posible. Así, el programa de ordenamiento puede decirse que esta implementado en el nivel

$$L = \frac{11.6}{442} = 0.026$$

Halstead llama al inverso del nivel de programa Dificultad D

$$D = 1 / L$$

Como incremento el volumen de implementación de un programa, en nivel de programa decrementa y la dificultad D incrementa. Así, las prácticas de programación, como el uso redundante de operandos o la falla al usar construcciones de control de alto nivel, tienden a incrementar la métrica de volumen, así como la métrica de dificultad. Para el programa del ordenamiento se obtiene que:

$$D = 1 / 0.026 = 38.5$$

Halstead ofrece una fórmula alternativa que estima el nivel de programa :

$$L = \frac{1}{D} = \frac{2}{n_1} * \frac{n_2}{N_2}$$

La dificultad de programación incrementa si operadores adicionales son introducidos (si $n_1/2$ incrementa) y si un operador es usado repetidamente (si N_2/n_2 incrementa). Cada parámetro puede ser obtenido contando los operadores y los operandos en un programa terminado.

El volumen potencial V^* puede ser deducido con $L = L$ para el programa del sort. $n_1 = 14$, $n_2 = 13$ y $N_2 = 42$.

$$\hat{L} = \frac{2}{14} * \frac{13}{42} = 0.044$$

el cual no es cercano al nivel 0.026 determinado antes, usando el contador de operandos únicos conceptualmente.

De este resultado, se puede sugerir que

$$V^* = V * \hat{L} = 442 * 0.044 = 19.4$$

Esfuerzo y tiempo

Halstead da la hipótesis que el esfuerzo requerido para implementar un programa de computadoras incrementa como incrementalmente el tamaño del programa. También toma mayor esfuerzo implementar un programa a un nivel bajo (más alta dificultad) que otro programa equivalente en un nivel alto (más baja dificultad). Así, el esfuerzo en Ciencia del Software está definido como:

$$E = V / \hat{L} = D * V = \frac{n_1 N_2 N \log_2 n}{2n_2}$$

Las unidades de medida de E son discriminantes mentales elementales. Para el programa del sort:

$$E = 442 / 0.044 = 10045$$

De esta manera, 10,045 discriminantes mentales elementales son requeridas para contruir el programa.

Una característica de la Ciencia del Software es la habilidad de relacionar sus métricas básicas a la implementación de tiempo. El psicólogo John Stroud sugirió que la mente humana es capaz de hacer un número limitado de discriminantes elementales por segundo.

Stroud afirma que este número B (llamado número stroud por Halstead) está en el rango de 5 y 20. Ya que el esfuerzo E utiliza discriminantes mentales elementales como su unidad de medida, el tiempo de programación T de un programa en segundos es simplemente :

$$T = E / B$$

B es normalmente 18, ya que parece dar los mejores resultados en los experimentos Halstead, el cual compara los tiempos de programación observados, incluyendo el tiempo para diseño, codificación y prueba.

El tiempo de programación estimado para el ejemplo será:

$$T = 10045 / 18 = 558 \text{ seg} = 10 \text{ minutos}$$

Este es probablemente un tiempo razonable para producir el programa, el cual es muy simple.

Nivel del lenguaje

Actualmente hay cientos de lenguajes de programación. En algunas organizaciones, diversos lenguajes pueden ser usados como una base en el desarrollo de software. Por esto es necesario una métrica que exprese el poder de un lenguaje. Halstead hipotetizó que si el lenguaje de programación se guarda fijo, como V^* incrementa, L decremente de tal manera que el producto LxV^* permanece constante. Así, este producto, llamado nivel de lenguaje, puede ser usado para caracterizar un lenguaje de programación.

$$\text{Nivel} = L * V^* = L^2V$$

Cuando Halstead analizó un número de programas escritos en diferentes lenguajes usando la ecuación para L, las métricas de nivel de lenguaje promedio fueron determinadas para ser 1.53 para PL/I, 1.21 para ALGOL, 1.14 para FORTRAN y 0.88 para lenguaje ensamblador CDC.

Los estudios han demostrado que es una función decreciente del tamaño del programa.

La ecuación anterior puede ser útil para seleccionar un lenguaje de programación para una área de aplicación.

Manipulando ecuaciones anteriores se obtiene:

$$E = \frac{V^{*3}}{2}$$

Así, para un problema dado, con V^* fija, el esfuerzo de programación E es inversamente relacionado al cuadrado del nivel de lenguaje. Esta relación puede ayudar a decidir, basado en la disponibilidad, integridad y eficiencia de varios compiladores, el lenguaje apropiado para un problema dado.

Para el problema del sort se obtiene que:

$$E = 0.044^2 * 442 = 0.86$$

Y con ese resultado se suiere que el programa fuera escrito en algo ligeramente abajo del lenguaje ensamblador CDC.

2.11.5 METRICAS DE DISEÑO

La complejidad de los programas aumenta con el tamaño: los programas más largos son más difíciles de entender y escribir, contienen más errores y son más difíciles de revisar. Para reducir esta complejidad, los diseñadores de software han incrementado la importancia de la modularidad de programas y de metodologías de diseño estructurado.

Entre las ventajas aportadas por la modularización son:

- Comprensión** : El programador y el usuario pueden entender más fácilmente la lógica del programa.
- Manejo** : Los gerentes pueden asignar personal a módulos y la responsabilidad esta más localizada.
- Eficiencia** : El esfuerzo de implementación debe ser reducido.
- Reducción de errores** : las pruebas deben ser más fáciles.
- Reducción de mantenimiento** : la identificación de los módulos debe ser más fácil ya que las diferentes funciones son ejecutadas por módulos diferentes.

Tambien existen algunas desventajas en la modularización.

Mientras más crezca el número de módulos, la necesidad para interfaces apropiadas entre los módulos y para la coordinación entre la gente asignada a codificar los módulos puede incrementar el esfuerzo para escribir el programa y el tiempo necesario para revisar el programa.

Algunos autores señalan que los módulos no deben exceder de 50 a 200 líneas, otros dicen que sin importar el tamaño de cada módulo, se debe ejecutar una función específica, otros dicen que deben ser limitados de acuerdo a un valor de una métrica de complejidad, como los módulos cuya complejidad ciclomática no exceda 10.

La metodología de diseño estructurado sugiere que la calidad de un diseño es relativa a cinco principios de diseño. Estos principios pueden ser objetivamente medidos para una o más de las métricas primitivas estudiadas.

Acoplamiento: Es el número de interconexiones entre módulos. El acoplamiento incrementa como el número de calls entre módulos incrementa o la cantidad de datos comparativos incrementa. La hipótesis es que los diseños con alto acoplamiento contendrán más errores. Las métricas primitivas relacionadas al número de interconexiones por nodo del diagrama estructurado pueden ser usadas para cuantificar el acoplamiento.

Cohesión: Es la medida de las relaciones de los elementos entre módulos. En un diseño cohesivo, las funciones relacionadas son localizadas en un módulo simple. La hipótesis es que los diseños con baja cohesión de módulo contendrán más errores. Las métricas primitivas relacionadas a la compartición de datos entre módulos pueden ser usadas para identificar la cohesión.

Complejidad: Un diseño debe ser tan simple como sea posible. La complejidad del diseño crece como el número de construcciones de control crece y como el tamaño en el número de módulos crece. La hipótesis es que los diseños con alta complejidad contendrán más errores. Las métricas primitivas relacionadas al número de nodos en el diagrama de estructuras y la profundidad de la gráfica puede ser usada para identificar la complejidad.

Modularidad: El grado de modularización afecta la calidad de un diseño, la sobremodularización es tan indeseable como la inframodularización. La hipótesis es que baja modularización indica más errores. Algunas métricas usadas para calificar la complejidad pueden ser usadas para cuantificar modularidad, pero de una manera opuesta, un bajo número de nodos pueden indicar baja modularización.

Tamaño: Un diseño que exhibe largos módulos o una gran profundidad en el diagrama de estructuras se considera indeseable. La hipótesis es que programas largos pueden contener más errores que pequeños programas. Las métricas de complejidad pueden ser utilizadas.

Se obtuvo la conclusión de que las métricas de acoplamiento mantienen las mejores correlaciones con el conteo de errores, mientras que las métricas de complejidad son las siguientes en importancia.

El coeficiente de correlación para las métricas de acoplamiento y errores fue de 0.73. Aunque este es un resultado significativo, el valor indica que hay otras variables que también afectan la calidad del software. La calidad del diseño de software puede ser mejorada minimizando el número de interconexiones entre módulos.

CAPITULO III

COSTO DE LA CERTIFICACION DE CALIDAD

3.1 INTRODUCCION

Se habla mucho de que la calidad es libre, pero siempre implica un costo. Los datos necesarios para calcular los costos de la calidad existen, por regla general, en el departamento de contabilidad, datos que debidamente arreolados, permiten organizar los costos de la calidad. Puede ser necesario como un primer paso, una relación de costos y cómo puede ponerse en uso.

3.2 COSTO DE LA CERTIFICACION DE CALIDAD

Una vez que los costos de la calidad han sido identificados y obtenidos, es necesario sujetarlos a un análisis antes de que se usen como una base para tomar acción. El análisis consiste en examinar cada elemento de costo en relación con otros elementos y con el total. Los elementos de costo de calidad del software son los siguientes :

ACTIVIDAD

En cualquier evaluación de calidad, el costo esta en relación de que se ha hecho y que no, las cosas que han sido hechas pueden ser costeadas en términos de las horas hombres desarrolladas. Las cosas que no han sido hechas, pueden ser costeadas en términos del impacto de la no ejecución de las tareas. Para determinar las tareas que deben hacerse y poder presupuestarlas, se necesita una evaluación de las horas en que sera desarrollada, el impacto en el esquema del programa (si el programa es manejado por esquema) y los beneficios derivados de las tareas en términos de la calidad del producto en liberación, costo total, recursos de poder humano, ventajas competitivas en el mercado y costos de mantenimiento después de la liberación.

Debe haber una relación positiva entre los objetivos del proyecto y las tareas a ser ejecutadas.

ENTRADAS

DATOS: En orden de ejecución de una tarea de calidad, la requisición de entrada de datos debe estar disponible. Los datos pueden ser derivados del análisis, provistos por el cliente, deducidos por la ejecución de algunas otras tareas, provistos como las salidas de otro proyecto o datos necesitados por los aspectos físicos del proyecto en si mismo.

Una vez que los datos requeridos han sido determinados, el costo de obtención de los datos puede ser determinado.

MATERIAL: Esto es todo lo necesario para realizar la tarea, como por ejemplo, formas para grabar datos o resultados de las actividades, etc. Una vez identificado el material se puede determinar el costo de compra o de creación y su disponibilidad en el mercado.

NECESIDADES DEL CLIENTE: Las necesidades del cliente es uno de los parámetros más difíciles de identificar en el conjunto de entradas de requerimientos para la evaluación de la calidad. Generalmente el cliente tiene una idea general de sus necesidades, pero no una idea específica de estas, por esto es importante determinar los requerimientos del proyecto (necesidades del cliente).

Los requerimientos que no sean bien definidos o cambien constantemente tendrán un impacto significativo en la calidad y ejecución del producto. Cada necesidad identificada debe ser asociada con una actividad y cada actividad debe tener un conjunto de datos de entrada asociados.

La variación primaria en el costo de calidad es el costo asociado con las diferentes maneras en que la producción puede ser desarrollada, cuando los requerimientos son refinados, interpretados e implementados.

ENTRENAMIENTO: Las actividades de calidad seguidas requieren entrenamiento en ciertos procesos técnicos. Los costos implicados en el entrenamiento deben de ser costeados.

Los beneficios derivados del entrenamiento en términos de la calidad del producto debe ser comparado con el costo del entrenamiento.

RECURSOS: Los recursos requeridos para proveer u obtener los datos de entrada necesarios, el material o el entrenamiento deben de ser costeados. La facilidad de los recursos para estos elementos deben también costearse en términos de espacio, utilerías y otros impactos de gastos.

PROCEDIMIENTOS: Cualquier actividad que parezca ser hecha en concordancia con cualquier conjunto de requerimientos, deberá ser en respuesta a un conjunto de procedimientos. Si los procedimientos no están disponibles, deben ser creados y el costo de tal creación tomado en cuenta.

ESTANDARES: En muchas situaciones, las actividades asociadas con el desarrollo del producto tendrán que conformarse a ciertos estándares. Estos estándares pueden ser establecidos externamente o internamente. Si una actividad es ejecutada debido al estándar y no porque pueda ser hecha de cualquier manera por su impacto de calidad positivo de todas maneras debe ser considerada un costo de calidad.

Si el estándar impone actividades, pero quita calidad debe ser cuestionado. Si el estándar adiciona calidad, pero la actividad asociada al estándar sería hecha de cualquier manera si el estándar no existiera, entonces no se adiciona costo debido al estándar.

ACCIONES DE ADICION DE VALORES: Para cada tipo de entrada, debe haber una actividad de adición de valores asociada. Estas actividades son en las cuales se adiciona un valor al producto mismo o al proceso que realiza ese producto. Esto es, el valor adicionado puede incrementar la calidad del producto, un impacto positivo en los procesos o procedimientos implementados en la producción significa una disminución en costos de producción sin impactar negativamente la calidad del producto.

Detrás de todas estas actividades, están las necesidades del cliente. Puede existir una necesidad real de un producto libre de defectos y otros en los cuales un sistema totalmente libre de defectos es deseable pero no necesario. El grado de confiabilidad del sistema debe ser optimizado en base al costo tiempo y habilidades tecnológicas disponibles.

SALIDAS

PRODUCTO: El producto resultante de las actividades desarrolladas tendrá un nivel de eficiencia determinado en la medida en que este libre de defectos y sea fácil de mantener. El producto tendrá requerimientos de eficiencia, conocidos o no dependiendo de que las entradas necesarias estén disponibles y las acciones de adición de valores hayan sido llevadas a cabo.

La confiabilidad del producto debera ser optimizada basándose en todos aquellos factores criticos como costo, tiempo y tecnologia .

SERVICIOS: Las salida de las acciones de adición de valores puede estar en la forma de un servicio ejecutado por otros departamentos. Mientras el producto no sea liberado, el servicio puede ser necesario para generar el producto con el nivel de calidad requerido. Por ejemplo: la eficiencia de diseño e inspección de código puede ser un servicio no liberable pero el impacto en la calidad del producto probablemente sea significativo. La revisión del documento es otra forma de actividades no liberables que tiende a asegurar al cliente que el producto entregado (documento) esta tan libre de defectos como el procedimiento de implementación lo permita.

INFORMACION: Frecuentemente es el resultado de las actividades de calidad, aun cuando la información por si misma no sea liberable. La información en este contexto puede tomar la forma de reportes de análisis de calidad, como el diseño y datos de inspección de códigos o reportes de auditoría.

La información puede tomar diversas formas y en cualquiera de ellas las actividades relacionadas con la calidad son parte del costo de calidad.

3.3 COSTO TOTAL DE CALIDAD

Cuando el jefe de programación, quien no entiende los elementos reales de calidad pregunta por el costo de una actividad, esta preguntando por el costo específico de la ejecución de la actividad no por el impacto resultante de hacer o no hacer la tarea.

Si la tarea cuesta una cantidad, pero el resultado de ejecutar la tarea ahorra otra cantidad, el costo real es la diferencia entre las dos. Sin embargo no es esta diferencia acerca de la que se busca información, ya que si la respuesta real es un costo negativo y si el jefe de programación no entiende realmente estas relaciones, estará tentado a pedir una reducción en el presupuesto para esa actividad.

Debido a esto, es necesario proveer de medios o recursos para evaluar el costo real de calidad en un proyecto y ofrecer un cuadro completo al jefe de programación.

Para esto cada departamento debe ir a través de un análisis de actividades de evaluación de calidad, reunir los datos apropiados y presupuestar el cuadro real al jefe de proyecto. Una vez hecho esto y educado al jefe de programación la tarea de ejecutar las actividades de calidad será fomentada.

3.4 ANALISIS DE ACTIVIDADES

El proceso de análisis de costo de calidad tiene 4 pasos principales:

1. Listar todos las actividades principales.
2. Para cada actividad:
 - .Listar entradas .Que ? De donde?
 - .Determinar su valor adicionado Porque hacerlo?
 - .Listar salidas .Que ? De donde?
3. Para cada actividad
 - .Determinar los requerimientos del cliente
 - .Determinar los requerimientos de proveedores
 - .Determinar medidas.
4. Para cada actividad
 - .Analizar los requerimientos de mediciones
 - .Determinar el nivel de esfuerzo: estimar el costo del esfuerzo
 - .Dividir el costo en categorías

Las actividades deben de ser primero identificadas , una breve descripción general debe ser provista para cada actividad principal en orden, para evaluar con propiedad esa actividad lo mejor será hacerlo a través de la forma de la figura 3.1.

El siguiente paso es determinar la información necesaria para cada tarea en términos de las entradas requeridas y los valores adicionados para la ejecución de la actividad.

La salida para la actividad también debe ser determinada. Un ejemplo de una forma para este propósito es mostrada en la figura

3.2. La ventaja de esta forma es la eliminación de trabajo supuesto y una clara indicación de que información vital se está ignorando. Cada actividad debe tener una forma propia.

La parte "impacto de lo que no se ha hecho" de la forma no siempre es fácil de especificar, especialmente en términos de elementos de costos. La respuesta a esta pregunta puede ser una evaluación basada en datos de proyectos pasados.

El siguiente paso es determinar las soluciones a las preguntas de requerimientos. La figura 3.3 muestra una forma para grabar esta información, pero para proyectos largos se puede usar otro tipo de documento.

Para identificar un conjunto de mediciones, las cuales nos proporcionarían los requerimientos, se necesitarían considerables negociaciones. En la práctica de la medición del software es solo escalar, un estado de desarrollo donde unas pocas medidas son aceptadas. Debido a esto las medidas seleccionadas por el contratista pueden no ser necesariamente aceptables para el cliente. Cada medida seleccionada debe ser identificada, justificada y comentada. Es común tener un conjunto de medidas para cada estado de desarrollo del software. Esto proporcionaría al jefe de desarrollo un recurso para continuas evaluaciones y correcciones en todas partes del proceso de desarrollo de software.

El proceso de selección de mediciones deberá estar determinado por distintos factores, como los requerimientos de datos de entrada, la disponibilidad del software de soporte para hacer las evaluaciones necesarias, la facilidad con la que los resultados pueden ser emitidos y analizados por la dirección y la cantidad de información contenida en los resultados de las mediciones. Es claro que una medida que solo indique que existe un problema es menos valiosa que una que indica que el problema existe y por que.

Una vez que las actividades, requerimientos de entrada, requerimientos de salida y requerimientos de mediciones son determinados, entonces el conjunto final de preguntas deberá realizarse. Como se ve en la figura 3.3. para cada actividad el jefe debe determinar whether la medición es posible ahora, si es posible del todo y whether la actividad realmente debe ser medida. Puede costar más ejecutar la medición que completar la tarea.

ANALISIS DE ACTIVIDADES POR DEPARTAMENTO

NOMBRE DE LA FUNCION	
NOMBRE DEL DEPARTAMENTO	NUM. DEPTO

DESCRIPCION GENERAL DEL TRABAJO DESARROLLADO EN ESTE DEPARTAMENTO		
(ACTIVIDADES MAYORES)		
FIRMA DEL GERENTE	FECHA	EXTENSION

FIGURA 3.1

ANALISIS DE ACTIVIDAD DE REQUERIMIENTOS

ACTIVIDAD :	DEPTO.	FICHA :	PREPARADO POR :
-------------	--------	---------	-----------------

CUALES SON LOS REQUERIMIENTOS DE ENTRADA QUE USTED Y SU PROVEEDOR HAN DEFINIDO ?

CUALES SON LOS REQUERIMIENTOS DE SALIDA QUE USTED Y SU CLIENTE HAN DEFINIDO ?

CUALES SON LAS MEDIDAS DE CALIDAD QUE MOSTRAN SI LA SALIDA SATISFACE LOS REQUERIMIENTOS ?

FIGURA 3.3

La actividad no debe ser costada en términos de las horas-hombre requeridas para ejecutar la tarea. La actividad puede ser puramente actividad de calidad o una actividad parcial que no es propiamente de calidad.

El costo de calidad debería estar categorizado en términos de actividad de prevención, actividad de evaluación o actividad de fallas. Ver figura 3.4.

Los costos de prevención tienen como finalidad evitar que ocurran defectos. Los costos de evaluación incluyen los gastos necesarios para conservar los niveles de calidad, por medio de una evaluación formal de la calidad de los productos, estos gastos comprenden los elementos de inspección, pruebas y auditoría de calidad. Los costos por fallas son causados por productos defectuosos.

3.5 ENFOQUE DE COSTOS

Una vez que estas acciones se han completado el director tiene la información necesaria para ir con el jefe de programación y presentar un cuadro verdadero del costo de elaborar un producto de calidad. El enfoque debe ser sobre 2 factores:

- Costo para analizar los problemas
- Costo para rehacer la tarea

El análisis de las tareas provee los medios para identificar errores dentro del sistema y priorizarlos para aplicar la acción correctiva. Concentrándose en los medios para identificar con exactitud la no conformidad, se requiere de conocer las tareas involucradas en esto.

El resultado del análisis debe documentarse y presentarse como base para tomar las decisiones.

ACTIVIDAD :

DEPTO:

FECHA :

PREPARADO POR :

ANALISIS DE MEDICIONES

- * ESTA ACTIVIDAD ESTA MEDIDA ACTUALMENTE ? NO SI
- * ESTA ACTIVIDAD PUEDE SER MEDIDA ? NO SI
- * ESTA ACTIVIDAD DEBERIA SER MEDIDA ? NO SI

NIVEL DE ACTIVIDAD

- * HORAS POR SEMANA ESTIMADAS GASTADAS EN ESTA ACTIVIDAD POR EL DEPARTAMENTO :

_____ HORAS / SEMANA

- * ES ESTA UNA ACTIVIDAD DE "COSTO DE CALIDAD" ?

NO

PARCIALMENTE

TOTALMENTE

- * HORAS POR SEMANA DE "COSTO DE CALIDAD" ESTIMADO GASTADO EN ESTA ACTIVIDAD :

_____ HORAS / SEMANA

COSTO DE LAS CATEGORIAS DE CALIDAD

- | | | | | | |
|----------------|--------------------------|--------------|-------|----------|-------|
| * PREVENCION | <input type="checkbox"/> | HORAS/SEMANA | _____ | COSTO \$ | _____ |
| * APROXIMACION | <input type="checkbox"/> | HORAS/SEMANA | _____ | COSTO \$ | _____ |
| * FALLAS | <input type="checkbox"/> | HORAS/SEMANA | _____ | COSTO \$ | _____ |
| * TOTAL | | HORAS/SEMANA | _____ | COSTO \$ | _____ |

FIGURA 3.4

3.6 CLASIFICACION DE COSTOS

-Costo que no es de calidad

Dentro de la discusión de la evaluación se determinará la parte que no es de calidad en las tareas. Para esto se debe decidir si la tarea es una de las siguientes:

-Diseño	-Desarrollo	-Creación
-Documentación	-Fabricación	
-Reunión	-Proceso	

Estas no son elementos de una tarea de calidad, son tareas que deben realizarse durante la obtención del producto, fuera de la consideración de calidad.

-Costos de calidad

Los elementos de las tareas los cuales son parte del costo de calidad son agrupados generalmente en 3 categorías y 2 clasificaciones. Las clasificaciones son conformidad y no conformidad.

Las categorías son prevención, evaluación y falla. Se ha establecido una relación entre las categorías y la clasificación. Prevención y evaluación caen dentro de la categoría de conformidad y falla dentro de no conformidad. Esta relación puede ser tabulada como se muestra:

CONFORMIDAD		NO CONFORMIDAD
PREVENCIÓN	EVALUACION	FALLA
Adiestramiento	Inspección	Rework (volver a trabajar)
Planeación	Pruebas	Servicio
Simulación	Monitoreo	Modificación
Modelado	Auditoria	Expedición
Consulta	Medición	Corrección
Certificación	Verificación	Recall (volver a llamar)
Calificación	Análisis	Retest (volver a probar)
		Análisis de errores

TABLA 3.1

Cuando la actividad de categorizar y clasificar es cumplida, cualquier otra actividad de análisis de calidad puede ser ejecutada. Esto puede hacerse usando una forma diseñada para este propósito como se muestra en la figura 3.5. Con esto se puede hacer un resumen de la información recolectada en un solo lugar haciendo fácil presentarla al director del programa.

Si el tiempo gastado y el costo incurrido por el costo de ejecutar actividades de calidad son el resultado de una producción pobre o un proceso de producción o diseño, entonces el costo de calidad puede ser reducido dentro del grado en el que el diseño, los procesos de producción y los procedimientos son mejorados.

La reducción en el costo de calidad puede ser medida y relacionada con la producción y actividades de diseño, pudiendo proporcionar el significado de una evaluación eficaz de las mejoras hechas.

3.7 ABUSO POTENCIAL DEL COSTO

Cuando el costo de análisis de calidad es publicado, se puede abusar de él. Algunos departamentos de finanzas ejecutan cualquier cosa asumiéndola como un costo de análisis de calidad.

El abuso del costo para el análisis de calidad esta generalmente relacionado al punto de enfoque del análisis. Hay una tendencia a enfatizar el costo sin tomar suficientemente en cuenta los elementos de falla del análisis. El costo es justamente un medio para atraer la atención sobre la tolerancia a los elementos de falla del análisis mismo y sus efectos dentro del programa.

Este es un requerimiento para entender el impacto de las fallas y un medio convincente de mostrar esto, es el costo gráfico. No se puede decir que las figuras sean erróneas o poco importantes, pero las fallas dan vida a los elementos de costo, siendo una cuestión de enfoque.

El enfoque podría ser sobre las fallas y como prevenirlas. Desarrollar el costo podría ser un medio de rastrear lo efectivo de reducir las fallas en el programa. Las figuras no pueden ser usadas en este caso para comparar departamentos o funciones. Nuevamente se enfatizaría sobre quitar la falla desde el inicio del desarrollo de software, marcando esta tendencia dentro de los departamentos o funciones.

DEPARTAMENTO: DESARROLLO DE SOFTWARE XYZ FECHA: 07/04/90

ACTIVIDAD	NON-COST OF QUALITY	CONFORMIDAD		NO CONFORMIDAD	TOTAL
		PREV	EVAL.	FALLAS	
DISENO DE PROGRAMAS DE SISTEMAS	160				160
REVISIONES DE DISENO		24	32		56
CODIFICACION	616				616
REVISION DE CODIFICACION		40	64		104
PRUEBAS DE UNIDADES DEL SISTEMA			96		96
MANEJO DE CONFIGURACION	80	16			96
SOLUCION DE PROBLEMAS				104	104
RETRABAJO / REPRUEBAS				328	328
TOTAL HORAS/HOMBRE	856	80	192	512	1640
% DEL GRAN TOTAL	52%	5%	12%	31%	100%

ANALISIS DE ACTIVIDAD DE CALIDAD

FIGURA 3.5

Si el enfoque y tendencia del análisis es sobre la prevención de los defectos, esto por sí mismo tendrá un significado importante dentro del costo de calidad.

Si el enfoque es sobre costos solamente, será un enfoque incompleto.

3.8 PRODUCTIVIDAD

La productividad es un aliado cercano al costo de calidad y muy relacionado con los elementos de falla.

Es fácil seguir la productividad en términos del número de líneas de código generado por unidad de tiempo. Esto es usualmente expresado en término de líneas por mes-hombre. Estas medidas no son las adecuadas y estimula el desarrollo de los departamentos fuera de la realidad ó elimina tareas de calidad y procedimientos que deben ser ejecutados.

Si la productividad es examinada en conjunto con el costo del análisis de calidad, entonces el énfasis deberá ser sobre la línea de producción correcta por mes-hombre

Para enfatizar sobre la producción como una medida de producción correcta, se requerirá de una educación al administrador de producción. Ya que usualmente el administrador recibe la productividad manejando la definición, medida, reporte del proyecto y todo sobre el costo del programa.

La producción del software puede ser considerada como una especificación funcional. Si esta especificación es producir rápido pero con muchos errores, la productividad del programador será desvirtuada. Si la producción es realizada con un mínimo de defectos, entonces se tendrá un impacto positivo sobre la producción del programador.

Los aspectos de productividad anteriores pueden ser incorporados dentro del costo del análisis de calidad. Estas actividades pueden ser elementos que mejoran el costo de calidad.

Cuando una actividad de desarrollo de software empieza a ser estimada, la relación de los componentes del costo y las salidas son una ayuda para entender las diferencias entre las actividades de software por sí mismas y los atributos de desarrollo de software.

Las actividades de software son mostradas en la Tabla 3.2, la cual da las típicas actividades de desarrollo de software, relacionando las componentes de costo y las salidas. Esto puede usarse como una base para ejecutar un análisis de costo de calidad para el departamento de desarrollo de software.

En la Tabla 3.3 se muestra la relación de desarrollo de software. Hay varios elementos de atributos de impacto como confiabilidad, exactitud y otros que son datos.

Esta figura junto con la Tabla 3.1, ayuda a formular un programa de reducción de costos de calidad, enfatizando actividades, las cuales pueden ser de un impacto significativo por lo que requieran más atención.

El costo de calidad en si mismo, debe de ser nulo, pero solo si las actividades propias, basdas en un análisis claro son ejecutadas. La calidad se provee libremente ya que un esfuerzo continuo es hecho para realizar las tareas de reducción de fallas y buscar maneras y medios para mover las actividades de eliminación, detección y prevención de fallas antes del final del proceso de desarrollo. Lo más cerca al inicio de la eliminación de fallas es realizado, lo más grande el impacto y lo más significativo la reducción de costos. Esta es la única manera que el costo de revisar calidad llega a ser menos que el costo de no realizar tareas de calidad. El costo de calidad puede ser propiamente evaluado sólo en términos del impacto si las fallas no ocurren.

ACTIVIDADES DEL SOFTWARE (TAB. 3.2)

ACTIVIDADES	COMPONENTES DE COSTO	SALIDAS
DEFINICION DEL SISTEMA	<ul style="list-style-type: none"> .DEFINICION DE REQUERIMIENTOS DE SOFTWARE .DESCRIPCION DEL SISTEMA DE SOFTWARE .PLANEACION DEL DESARROLLO DE SOFTWARE .ANALISIS DE CAMBIOS DE INGENIERIA 	<ul style="list-style-type: none"> .ESPECIFICACIONES DE REQUERIMIENTOS DEL SISTEMA DE SOFTWARE .DESCRIPCION DE LA DOCUMENTACION DEL SISTEMA DE SOFTWARE .PLAN DE DESARROLLO DE SOFTWARE .INGENIERIA DE PROPOSITOS DE CAMBIO
DISEÑO DE SOFTWARE	<ul style="list-style-type: none"> .DISEÑO FUNCIONAL .DISEÑO DE PROGRAMA .DISEÑO DE PRUEBAS .HERRAMIENTAS DE PRUEBA .EVALUACION DE DISEÑO 	<ul style="list-style-type: none"> .ESPECIFICACIONES DEL DISEÑO FUNCIONAL .ESPECIFICACIONES DEL DISEÑO DEL PROGRAMA .ESPECIFICACIONES DE DISEÑO DE PRUEBAS .UTILERIAS Y DEBUGIN
DESARROLLO DE SOFTWARE	<ul style="list-style-type: none"> .MODULO DE DESARROLLO .PRUEBAS DEL DESARROLLO .ANALISIS DE PROBLEMAS Y CORRECCION 	<ul style="list-style-type: none"> .LIBRERIAS DE DESARROLLO (MODULOS) .DESARROLLO DE PRUEBAS DE PROCEDIMIENTOS Y REPORTES .MODIFICACION DE PROGRAMAS
PRUEBAS DEL SISTEMA DE SOFTWARE	<ul style="list-style-type: none"> .PRUEBAS DE PROCEDIMIENTOS DEL SISTEMA DE SOFTWARE .PRUEBAS E INTEGRACION DE SOFTWARE 	<ul style="list-style-type: none"> .PRUEBAS DE PROCEDIMIENTOS DEL SISTEMA DE SOFTWARE .INTEGRACION DE LIBRERIAS Y PRUEBAS DE REPORTES DEL SISTEMA
SISTEMA/PRUEBAS DE ACEPTACION	<ul style="list-style-type: none"> .SOPORTE DE PRUEBAS AL SISTEMA .SOPORTE DE PRUEBAS DE ACEPTACION .SOPORTE DE OPERACION .ENTRENAMIENTO .SOPORTE DEPLOYMENT 	<ul style="list-style-type: none"> .LIBRERIAS DEL SISTEMA REPORTES DE PRUEBAS .LIBERACION DEL SISTEMA DE SOFTWARE ACEPTACION DE LIBRERIAS .NIVEL DE ASISTENCIA Y MANTENIMIENTO .NIVILE DE ESFUERZO DE MANUALES Y CURSOS .NIVEL DE ESFUERZO DE ASISTENCIA
SOPORTE GENERAL	<ul style="list-style-type: none"> .MANEJO DEL PROYECTO .CONFIGURACION DE MANEJO Y CONT. .COSTO DE INGENIERIA DE SOFTWARE .CONTROL DE CALIDAD .ADMINISTRACION DE CENTROS 	<ul style="list-style-type: none"> .NIVEL DE ESFUERZO .PROCEDIMIENTOS, ESTANDARES, LIBRERIAS DE CONTROL .AUDITORIAS Y PLANES DE CONTROL DE CALIDAD

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

ATRIBUTOS DE SOFTWARE PARA EL DISEÑO (TAB. 3.3)

	C O N F I A B L E	P R E C I S I O N	E X A M I N A B L E	M A N T E N I B L E	C A M B I A B L E
DISEÑO DE OBJETIVOS					
CARACTERÍSTICAS DE DISEÑO					
ARQUITECTURA TOP-DOWN	X		X		
TAMANO DEL MÓDULO			X	X	
BAJA COMPLEJIDAD	X		X	X	X
PROGRAMACIÓN ESTRUCTURADA	X		X	X	X
AGRUPACIÓN DE DECLARACIONES			X	X	X
PARAMETRIZACIÓN SIMBÓLICA			X	X	X
CONVENCIÓN DE UNIFICACIÓN DE NOMBRES			X	X	X
TIEMPO Y TAMAÑO DE ANÁLISIS	X		X		
PRECISIÓN MONITOREO		X	X		
NÚMERO BAJO DE PARÁMETROS	X		X	X	X
MINIMIZACIÓN DE MODELOS DINÁMICOS	X		X	X	X
ENFASIS SOBRE LA INSPECCIÓN DE DISEÑO	X		X		

CAPITULO IV

HERRAMIENTAS DE CERTIFICACION DE CALIDAD

4.1 INTRODUCCION

En la actualidad, existen algunas herramientas ya diseñadas por algunos fabricantes, para evaluar y certificar la calidad de sistemas desarrollados en sus equipos.

Estas herramientas, utilizan el análisis estático, que es una técnica para valorar las características estructurales del código fuente o cualquier representación notacional que se apeque a reglas sintácticas bien definidas. Este análisis examina la estructura del código sin afectarlo.

Las herramientas tienen como objetivo mejorar la calidad de los sistemas a través de la detección de errores de lógica y promover la facilidad de mantenimiento identificando la complejidad, estructura y estandarización de la programación en forma objetiva.

Los beneficios que el análisis estático provee son:

A gerencias de áreas de desarrollo y certificación de calidad

- Evaluación de la calidad del código en forma objetiva.
- Reducción del tiempo para la realización de revisiones estructuradas.
- Documentación de mejoras en forma automática.
- Identificación de magnitud de cambios entre distintas versiones del mismo programa.

A analistas y programadores

- Ayuda a entender código no documentado.
- Identifica errores de lógica antes de iniciar pruebas.
- Proporciona documentación actualizada en el código mismo de los programas.
- Acelerar el proceso de rastreo de errores.
- Ayuda a obtener código de calidad.

Entre las herramientas existentes, que fueron desarrolladas en base al análisis estático son :

- SNF II
- CHECK-OUT
- MEASURE
- PATHVU

4.2 SMF-II SYSTEM MANAGEMENT FACILITY

4.2.1 DESCRIPCION DE LA HERRAMIENTA

Es una herramienta de monitoreo disponible para CANDE, en equipo UNISYS serie A, que da estadísticas de uso de los recursos de computo y del comportamiento de los programas, ayudandonos a:

-Conocer el consumo del procesador, memoria, disco e I/O de programas y aplicaciones que serán liberadas a producción.

-Conocer el tiempo de respuesta de programas y/o aplicaciones que serán liberadas a producción.

Nos provee un lenguaje de consulta(query) para almacenar la información dentro de una base de datos SMFII. Proveyendo al usuario con una poderosa capacidad para examinar y analizar la ejecución del sistema.

El usuario puede dar las especificaciones del reporte en forma interactiva por medio de la terminal o puede dar estas dentro de una secuencia de datos de un archivo que facilita la ejecución en batch.

Igual que otros lenguajes de consulta el SMFII no tiene un conocimiento inherente acerca del contenido de la base de datos. El programa por medio del usuario selecciona los elementos de la base de datos, manipula estos en gráficas o en forma estática.

ARQUITECTURA DE LA BASE DE DATOS.

Los datos contenidos en la base de datos SMFII son organizados en 4 módulos funcionales. Cada módulo consiste de 1 o más archivos en disco que contienen la ejecución de los datos, y un archivo SYMBOLTABLE con la descripción de los nombres, localización y tipo de los datos de la información contenida dentro de los archivos. El contenido de los módulos es el siguiente:

MODULO	CONTENIDO
UTILIZACION	STATUS SAMPLES INFORMACION DE MEDIO AMBIENTE MEZCLA SENCILLA (MIX SANPSHOTS) INFORMACION DE TOTALES IOTRACE DATA SYMBOLTABLE

HADWARE

ERRORES E/S
ERRORES MCS
ERRORES MLIP
ERRORES B 7000 MAINFRAME
ERRORES B 6800 MAINFRAME
ERRORES B 5900 MAINFRAME
ERRORES SERIE A MAINFRAME
HALT/LOAD INFO
ERRORES DCP
TAPE ACTIVITY INFO
BNA ACTIVITY INFO
SYMBOLTABLE

WORKLOAD

DATOS HISTORICOS DE TAREAS
(TASK HISTORY DATA)
MCS ACTIVITY INFO
ARCHIVOS DE DATOS DE ACTIVIDAD
(FILE ACTIVITY DATA)
SYMBOLTABLE

AVAILABILITY

DISRUPTIVE EVENT DATA
EVENT RESPONSE DATA
SYMBOLTABLE

EL DIAGRAMA GENERAL SE PUEDE VER EN LA FIG.4.1

DIAGRAMA GENERAL DE SMF-II

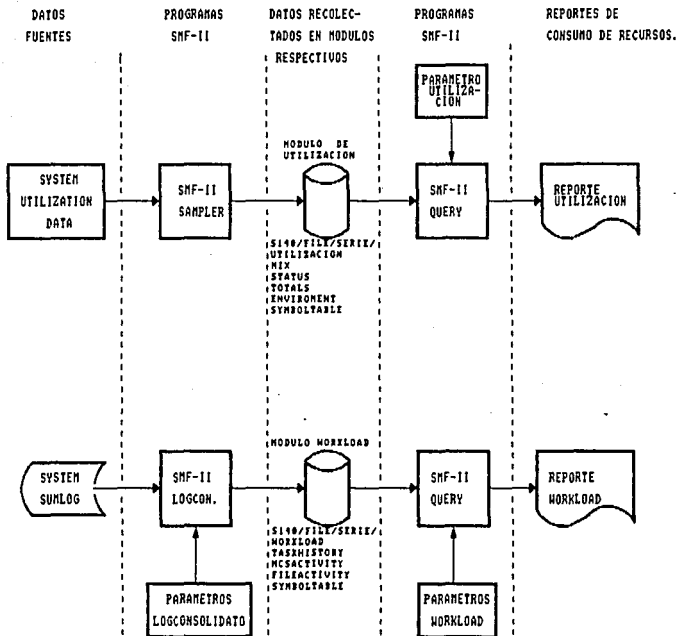


FIG. 4.1

MODULO WORKLOAD

Contiene datos relacionados con el funcionamiento y ejecución de los programas que corrieron en el sistema. Así como los recursos del sistema requeridos por cada programa individual o grupos de programas.

El programa SMF-II LOGCONSOLIDATOR extrae información desde uno o varios "sumlogs" específicos. Durante la ejecución de este programa se generan los archivos de datos de este modulo que son:

TASKHISTORY
MCSACTIVITY
FILEACTIVITY
SIMBOLETABLE

La información obtenida en el reporte de WORKLOAD es la siguiente:

Por código del usuario:

- Nombre de la tarea
- Número de mezcla
- Tiempo de procesador
- Tiempo de espera del procesador
- tiempo de entrada/salida
- Memoria de código promedio
- Memoria de datos promedio
- Tiempo transcurrido
- Número de pbits de inicio
- Tiempo de pbits de inicio
- Numero de otros pbits
- Tiempo de otros pbits
- Número de archivos de disco cerrados
- Número de archivos remotos cerrados

Por número de mezcla

- Nombre del archivo
- Tamaño del archivo
- Tamaño del bloque
- Tiempo de entrada/salida
- Número de transacciones
- Porcentaje de lecturas
- Accesos físicos
- Tipo de cerrado
- Número de unidad

- Tipo de unidad
- Tipo de archivo
- Error de cerrado
- Tipo de uso

Un ejemplo del reporte obtenido se puede ver en el punto 4.2.2 de reportes (reporte1)

MODULO DE UTILIZACION

El módulo de utilización de SMF-II contiene datos de aprovechamiento y ejecución relacionados con el uso de los recursos Hardware y Software usados por la mezcla del sistema. Los archivos del módulo de utilización contienen datos que son coleccionados en tiempo real del sistema. Esto es obtenido por medio del programa:

SMF-II/SAMPLER

La explotación de la información del módulo de utilización se puede hacer de 2 maneras:

- Para reporte general (UTILIZATION/PARAMETRO1)
- Para reporte detallado (UTILIZATION/PARAMETERS2)

Se deberá modificar los siguientes parametros en el archivo seleccionado:

En :	NOMINA/FILE/#SERIE	El nombre del archivo a explotar
	PACK	Es el disco donde estan los archivos
	10:00 12/15/89	Es la hora y fecha inicial de la información a explotar
	12:00 12/20/88	Es la hora y fecha final de la información a explotar

Se cuenta con un programa query que extraera la información requerida, generando un reporte que posteriormente se podrá mandar a imprimir.

La información obtenida en el reporte general del módulo de UTILIZACION es la siguiente:

- Nombre de la tarea
- Número de mezcla
- Memoria save máxima
- Memoria save promedio
- Memoria total máxima
- Memoria total promedio
- Tiempo de procesador máximo
- Tiempo de entrada/salida máximo
- Tiempo transcurrido

La información obtenida en el reporte detallado del módulo de UTILIZACION es la siguiente:

- Nombre de la tarea
- Número de la mezcla
- Memoria save máxima
- Memoria save promedio
- Memoria total máxima
- Memoria total promedio
- Tiempo de procesador máximo
- Tiempo de entrada/salida máximo
- Tiempo transcurrido
- Time Stamp

Un ejemplo del reporte obtenido se puede ver en el punto 4.2.2 de reportes (reporte2)

4.2.2 REPORTES DE SMF II

AS SMFII ANALYSIS FOR SYSTEM #23

FILTERING: MIYNUMBER= 7015

RANGE: 09:00:00 02/20/89 - 10:00:00 02/20/89

2 OBSERVATIONS

--- DATA EXTRACTION LIMITED TO 20000 RECORDS ---

1. NOMBRE TAREA	2. NUM MEZCLA
3. MEMORIA SAVE MAX	4. MEMORIA SAVE PROM.
5. MEMORIA TOTAL MAX	6. MEMORIA TOTAL PROM.
7. TIEMPO PROCESADOR MAX	8. TIEMPO I/O MAX
9. TIEMPO TRANSCURRIDO	10. TIMESTAMP

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
----	----	----	----	----	----	----	----	----	-----

(RECH)S019/PROGRAM/LIBRE1 ON INGENIERIAI	7015	6981	6981	7588	7588	0:00:22	0:00:58	0:01:24	09:42:08
(RECH)S019/PROGRAM/LIBRE1 ON INGENIERIAI	7015	6981	6981	7588	7588	0:01:09	0:03:04	0:04:25	09:45:03

[GRAND TOTAL]-----

13962 13962 15176 15176

>>> OUTSTANDING DEFINES/ FUNCTIONS/ AND STRING EXPANSIONS DISCARDED <<<<

AS SMFII ANALYSIS FOR SYSTEM #22

FILTERING: (LOGUSERCODE="RECH")

RANGE: 12:30:00 02/13/89 - 13:00:00 02/13/89

? OBSERVATIONS

--- DATA EXTRACTION LIMITED TO 30000 RECORDS ---

- | | |
|----------------------------|-------------------------------|
| 1. NOMBRE TAREA | 2. NUM MEDCLA |
| 3. TIEMPO PROCESADOR | 4. TIEMPO ESPERA PROC. |
| 5. TIEMPO I/O | 6. MEMORIA CODIGO PROM. |
| 7. MEMORIA DATOS PROM | 8. TIEMPO TRANSCURRIDO |
| 9. NUM.PBITS.INICIO | 10. TIEMPO PBITS INICIO |
| 11. NUM.OTROS PBITS | 12. TIEMPO OTROS PBITS |
| 13. ARCHIVO DISCO CERRADOS | 14. ARCHIVOS REMOTOS CERRADOS |

1.

	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
(RECH)CANDE/CODESSO ON INGENIERIA	3727	0:04:43	0:17:27	0:38:35	2152	43510	1:12:16	544	0:00:02	29	0:00:00	14
(RECH)OBJECT/CONVERT/DATE ON INGENIERIA	3728	0:00:00	0:00:00	0:00:00	122	908	1:12:15	12	0:00:00	1	0:00:00	1
(RECH)OBJECT/LINEA ON INGENIERIA	3412	0:00:00	0:00:00	0:00:00	103	1960	0:00:02	28	0:00:02	0	0:00:00	0
*OBJECT/CREAFORMAS	3391	0:00:02	0:00:02	0:00:01	1310	4599	0:04:40	92	0:00:00	3	0:00:00	1
(SUBTOTAL FOR NOMBRE TAREA)	3896	0:00:02	0:00:12	0:00:01	1470	4504	0:02:14	121	0:00:00	0	0:00:00	2
			2	2700		2		2		2		3
*SYSTEM/FILEDATA	3913	0:00:01	0:00:00	0:00:00	4689	3798	0:00:07	129	0:00:00	1	0:00:00	1
(GRAND TOTAL)				7	1.E4		7		7		7	24

>>> OUTSTANDING DEFINE/FUNCTIONS/ AND STRING EXPANSIONS DISCARDED <<<<

AS SMFI1 ANALYSIS FOR SYSTEM #23
 FILTERING: FAMI#NUMBER=3727
 RANGE: 12:30:00 02/13/89 - 13:00:00 02/13/89

13 OBSERVATIONS

--- DATA EXTRACTION LIMITED TO 20000 RECORDS ---

- | | |
|--------------------|------------------------|
| 1. NOMBRE ARCHIVO | 2. TAR. ARCHIVO |
| 3. TAR. BLOQUE | 4. TIEMPO I/O |
| 5. NUM. TRANSAC. | 6. PORCENTAJE LECTURAS |
| 7. ACCESOS FISICOS | 8. TIPO CERRADO |
| 9. NUM. UNIDAD | 10. TIPO UNIDAD |
| 11. TIPO ARCHIVO | 12. ERROR CERRADO |
| 13. TIPO USO | |

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
(RECH) SORT/DISK	19665	570	0:00:49	2030	50	2030	UNKNOWN	85	17	192	0	OUTPUT
(RECH) SORT/DISC	0	90	0:00:00	0	0	0	UNKNOWN	85	17	192	0	OUTPUT
(RECH) S019/FILE/DISPERION	150240	600	0:12:40	17035	100	16719	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/MAESTRO	206000	3600	0:11:10	8512	100	8030	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/GASTOS	1500	1500	0:01:19	2451	100	1379	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/SALDOS	7956	780	0:00:14	8525	100	205	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/REGIONALES	2000	1200	0:01:43	8613	100	2024	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/SUCURSALES	2040	600	0:05:01	9942	100	6916	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/ZONAS	2400	1200	0:04:15	8523	100	4473	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/DIVISIONES	2000	1200	0:00:20	8523	100	542	UNKNOWN	60	17	192	0	OUTPUT
(RECH) S019/FILE/AREAS	2000	1200	0:00:00	0	0	0	UNKNOWN	60	17	192	0	OUTPUT
*BD/0003214/0003727/0005019GASTOINDLA	8400	13	0:00:14	10571	0	783	UNKNOWN	85	17	4	0	OUTPUT
*BD/0003214/0003727/0015019G5T0GERLA	21700	13	0:00:35	49344	0	2120	UNKNOWN	85	17	4	0	OUTPUT

>>> OUTSTANDING DEFINES, FUNCTIONS, AND STRING EXPANSION DISCARDED <<<<

AS SMFIT ANALYSIS FOR SYSTEM M23

FILTERING: (USERCODE="RECH")

RANGE: 09:00:00 02/29/89 - 10:00:00 02/20/89

13 OBSERVATIONS

--- DATA EXTRACTION LIMITED TO 20000 RECORDS ---

* CONSUMO DE RECURSOS POR EL USUARIO RECH *

- | | |
|--------------------------|------------------------|
| 1. NOMBRE TAREA | 2. NUM MEZCLAO |
| 3. MEMORIA SAVE MAX. | 4. MEMORIA SAVE PROM. |
| 5. MEMORIA TOTAL MAX | 6. MEMORIA TOTAL PROM. |
| 7. TIEMPO PROCESADOR MAX | 8. TIEMPO I/O MAX |
| 9. TIEMPO TRANSCURRIDO | |

1.	2.	3.	4.	5.	6.	7.	8.	9.
(RECH)CANDE/CODESSO INGENIERIA1	7011	2062	2062	3177	3177	0:00:01	0:00:01	0:00:01
(RECH)OBJECT/CONVERT/DAT1 ON INGENIERIA1	7012	342	912	1075	1075	0:00:00	0:00:00	0:00:00
(RECH)SB19/PROGRAM/LIBRE1 ON INGENIERIA1	7015	6981	6381	7586	7586	0:01:00	0:01:00	0:01:00
JOBFILE/CONVERTER	7003	1713	1713	1823	1823	0:00:02	0:00:02	0:00:02
EXEC	7014	972	972	987	987	0:00:00	0:00:00	0:00:00
[GRAND TOTAL]								

>>>> OUTSTANDING DEFINES, FUNCTIONS, AND STRING EXPANSION DISCARDED <<<<

4.3 CHECK-OUT

4.3.1 DESCRIPCION DE LA HERRAMIENTA

Es un sistema de pruebas diseñado por Programming Aids, para uso específico de Burroughs. Muestra un listado visual de donde los programas en COBOL tienen que ser provados. Es un sistema que nos dara un reporte de una prueba o la acumalación de los resultados de varias pruebas y es tan sencillo de usar como un compilador de cobol, solo requiere del uso de algunas tarjetas de control adicionales.

El CHECK-OUT consiste de 2 programas: CHEK-OUT GENERATOR y el CHECK-OUT ANALYZER . El primero es un precompilador de COBOL que lee el programa fuente y produce un nuevo programa provado. El segundo es corrido despues de una serie de pruebas y nos produce el CHECK-OUT PROGRAM LISTING.

Ofrece 4 opciones de corrida que permiten optimizar los componenetes de los recursos requeridos. Dos de estas opciones dan una impresión como la mostrada en el reporte 3, las otras 2 muestran algo semejante en el reporte 4.

Esta herramienta nos ayuda entre otras cosas a:

- Provee los medios para un examen mas minucioso que nos da mayor confiabilida en el recién iniciado programa de Cobol.
- Identificar "Códigos muertos" (no utilizados)
- Ayuda con la optimización del programa identificando areas de alta actividad.

ESTADOS DE CONTROL

El CHECK-OUT provee estados de control que son establecidos por el programa STARTGEN el cual ejecuta el CHECK-OUT GENERATOR y el compilador de Cobol. Estos solo controlan la operación del CHECK-OUT GENERATOR.

Para la operación interactiva los estados de control son especificados dentro del prompt de una terminal remota, para la operación en batch estos estados de control son especificados por tarjetas de control que entran por medio del archivo SPECS.

Las tarjetas de control consisten de una simple palabra llave la cual puede o no ser seguida por información adicional, esta información puede ser:

- Un simple caracter alfabetico
- Un entero
- Un string

Las reglas que se aplican a estas tarjetas de control son:

- El archivo SPECS acepta tarjetas perforadas dentro de un lector de tarjetas, una baraja de datos dentro del WFL job y un archivo de datos en disco(diskpack).
- Pueden preceder a la palabra llave ninguno, uno o más espacios
- Pueden seguir a la palabra llave uno o más espacios, un signo de igual o un parentesis izquierdo
- Un entero dentro de las tarjetas de control debe ser mayor que cero y menor que 100,000
- El string puede consistir de cualquier combinación de caracteres EBCDIC
- El string puede incluir parentesis, los cuales deben aparecer en pares, uno abriendo y otro cerrando.
- El scanning de las tarjetas de control empieza en la columna uno y continua hasta la columna 80
- Si el string no cabe en las 80 columnas se podra continuar en la columna uno considerandola como la 81
- Un punto, un signo de por ciento o un punto y coma fuera del string pone fin al scanning de las tarjetas.
- Solamente puede aparecer una palabra llave dentro de las tarjetas de control
- El control de tarjetas check-out puede aparecer en cualquier orden dentro del archivo SPECS.

PALABRAS LLAVES DE TARJETAS DE CONTROL

Las palabras llaves que se utilizan en las tarjetas de control son:

1) ALL-BRANCHES -(todas las ramas) Identifica cual condición de verdadero o falso es o no es ejecutada y asocia a cada rama falsa con su correspondiente rama verdadera.
En el listado del CHECK-OUT al principio de la instrucción aparece un número causado y un indicador:

"true branch #" para los estados IF cuando el resultado es verdadero.

"false branch #" despues del estado ELSE o el punto que termina el estado IF.

Ademas del número de parejas de ramas verdaderas con sus correspondientes ramas falsas y el indicador profundida de anidamientos de los de estados IF.

Detecta las "ramas ciegas falsas" que ocurren cuando se usa el formato tested/not tested o el estado IF sin su correspondiente ELSE.

La palabra llave all-branches nos da un resumen de información adicional al final del listado con lo siguiente:

- Número de ramas dentro del programa
- Número de ramas verdaderas ejecutadas
- Porcentaje de ramas verdaderas ejecutadas
- Número de ramas falsas ejecutadas
- Porcentaje de ramas falsas ejecutadas

Formato: ALL-BRANCHES

2) CANDE -manda al sistema de CHECK-OUT el programa que normalmente sería compilado usando el comando CANDE COMPILE.

Formato: CANDE

3)CHARGE -Asigna un cargador de código al WFL job el cual ejecuta el CHECK-OUT generator y el compilador de Cobol.

Formato: CHARGE and string

3)CLASS -Asigna el WFL job el cual ejecuta el CHECK-OUT generator y el compilador de Cobol a una de las colas del sistema.Class y QUEUE son sinonimos.

Formato : CLASS y un entero

4)COMPILER -Especifica el nombre del compilador distinto que Cobol. El nombre debe ser válido para el compilador de Cobol conforme las especificaciones contenidas en la publicación "BURROUGHS PUBLICATION Número 5001464".

Formato: COMPILER y un string.

Ejemplo: COMPILER (COBOL31)

5)DESTNAME -Asigna un destino al WFL job que contendra la ejecución del CHECK-OUT generator y del compilador COBOL.

Formato: DESTNAME y un string

6)FAMILY -Asigna un estado de FAMILY al WFL job que contendra la ejecución del CHECK-OUT generator y del compilador Cobol.

FORMATO: FAMILY y un string.

EJEMPLO: FAMILY (DISK = NEWPACK OTEHRWISE OLDPACK)

7) GENERATOR -Asigna un nombre no estandar al CHECK-OUT generator.
El generador provisto por PROGRAMMING AIDS, es nombrado CHECK-OUT/GEN. Si embarco cada instalación tiene la opción de renombrar el generador. Y se tendra que indicar el nuevo nombre cada vez que se corra el programa STARTGEN.

Formato: GENERATOR y un string

8) JOB -Asigna un nombre al WFL job el cual ejecutara el CHECK-OUT generator y el compilador Cobol.

Formato: JOB y un string.

9) MANUAL-FILES -Instruye al sistema CHECK-OUT para que no genere automaticamente los titulos para los archivos operacionales del CHECK-OUT. Se tendran que proveer los titulos para este archivo operacional via archivo tarjetas de control WFL.

Formato: MANUAL-FILES

10) OPTION -Selecciona el formato para el listado generado por el programa CHECK-OUT.
.T especifica el formato tested/notested (provado/no provado).
.C especifica la ejecución con el formato de contadores.

El formato tested/notested requiere un tope menor de recursos de computadora que el formato de contadores.

Formato: OPTION y un caracter(T o C)

11) QUEUE -Asigna el WFL job el cual ejecuta el CHECK-OUT generator y el compilador a una cola del sistema. QUEUE y CLASS son sinónimos.

Formato: QUEUE y un entero

12) STARTTIME -Asigna un STARTTIME (tiempo de inicio) al WFL job que contendrá la ejecución del CHECK-OUT generator y el compilador Cobol.

Formato: STARTTIME y un string

Ejemplo: STARTTIME (22:00 ON 03/21/89)

13) SUB-PROGRAM-NO -Asigna un número al subprograma, el cual saltará al programa huésped. El número asignado a cada subprograma:

-Deberá ser único

-Deberá estar dentro del rango 1 al 99,999

-Puede estar dentro de cualquier orden

El sistema CHECK-OUT usa el SUB-PROGRAM-NO para generar un nombre de archivo interno para la operación de archivos del CHECK-OUT para cada subprograma. Estos nombres internos deberán tener la forma siguiente:

[identificador de archivo] [sub-program-no]

donde [identificador de archivo] es .QZZQSUBS o
.QZZQSUBDATA

Por ejemplo, si el Sub-program-No especificado es 16, el nombre interno del archivo dentro del subprograma será:

.QZZQSUBS16

.QZZQSUBDATA16

Formato: SUB-PROGRAM-NO y un entero

14) TARGET-PROGRAM -asigna un nombre al Target-program.

Formato: TARGET-PROGRAM y un string.

15) USER

-Asigna una combinación de usercode/password al WFL job el cual ejecuta el CHECK-OUT generator y el compilador de Cobol.

Formato: USER y un string

4.3.2 REPORTE DE CHECK_OUT.

403910	SET-UP-BLANKS.		TESTED
403920	IF IN-SUB = 81		TESTED
403930	PERFORM SET-UP-BUMP		TESTED
403940	SET IN-SUB TO 1.		TESTED
403950	IF IN-CHAR (IN-SUB)=SPACE		TESTED
403960	SET IN-SUB UP BY 1		TESTED
403970	GO TO SET-UP-WORD	***	NO ***
404100	GET-SET-UP-WORD		
404100	MOVE SPACES TO TEST-WORD		TESTED
404200	SET TEST-SUB TO 1		TESTED
404300	PERFORMS SET-UP-BLANKS		TESTED
404400	IF IN-CHAIN (IN-SUB) = "." ON "% OR ",		TESTED
404500	MOVE 1 TO SET-UP-CARD-DONE	***	NO TESTED ***
404600	GO TO G-S-U-W-EXIT	***	NO TESTED ***
*	***EXECUTION PASSED HERE (YES/NO)***	***	YES ***
404700	G-S-U-W-LOOP.		
404800	IF IN-SUB=81		TESTED
404810	PERFORM SET-UP-BUMP		TESTED
404820	SET IN-SUB TO 1		TESTED
404900	IF IN-CHAR (IN-SUB) = "." OR "X" OR ",		TESTED
405000	MOVE 1 TO SET-UP-CARD-DONE	***	NO TESTED ***
405100	GO TO G-S-U-W-EXIT.	***	NO TESTED ***
405200	IF IN-CHAR (IN-SUB)=SPACE		TESTED
405300	GO TO G-S-U-W-EXIT.		TESTED
405400	MOVE IN-CHAR (IN-SUB) TO TEST-CHAR (TEST-SUB)		TESTED
405500	SET IN-SUB TEST-SUB UP BY 1.		TESTED
405600	GO TO G-S-U-W-EXIT.		TESTED
*	*** EXECUTION PASSED HERE (YES/NO) ***	***	NO ***
405700	G-S-U-W-EXIT.		TESTED
	EXIT.	***	NO ***
*	**** EXECUTION PASSED HERE (YES/NO) **		
	SEARCH-SET-UP-WORD.	*	TESTED
406000	SET SET-UP-SUB TO 1.	*	TESTED

4.4 MEASURE

4.4.1 DESCRIPCION DE LA HERRAMIENTA

Measure colecta y despliega la ejecución estática para el sistema TANDEM NONSTOP. Para análisis de la información y el uso efectivo de esta en la ejecución del análisis, se requiere un entendimiento de como trabaja el sistema TANDEM y las aplicaciones que corren en este.

Dimensiona el consumo de recursos de aplicaciones y nos da a conocer el comportamiento de estas y el consumo de recursos con cargas reales y estimadas de trabajo.

Nos da el consumo de disco, I/O y CPU de programas y/o aplicaciones que serán liberadas en producción. Así como el tiempo de Respuesta de programas y aplicaciones.

Haciendo uso de MEASURE se puede determinar el recurso que se desea examinar, despues se pasa la información al subsistema de MEASURE dentro de la configuración de medidas. La configuración describe cada recurso en terminos de su tipo de entidad a examinar y las especificación de entidad.

-El tipo de entidad describe el recurso que se desea examinar. Los tipos son:

CLUSTER	NETLINE	CPU
DEVICE	PROCESS	PROCSSH
DESC	DISCOPEN	FILE
LINE	SYSTEM	TERMINAL
TMF	USERDEF	

-La especificación de entidades describe el recurso particular que se desea examinar. La especificación de entidad depende del tipo de entidad.

MEASURE tiene un set predefinido de contadores donde se colecta la información acerca de los recursos. Los contadores se basan en el tipo de entidad.

TOMA DE MEDICIONES

Para tomar una medición se entra con MEASCOM, definiendo la configuración de la medición. Por ejemplo se medirá una colección de información acerca de todos los CPU's y dos discos,\$DATA y \$BUYER. MEASURE automáticamente crea la medición del archivo de datos \$DATA.PERF.NOV04 en caso de que no exista; si existe MEASURE purga este.

```
>MEASCOM  
MEASURE - T9086B30 - (MAR0186)  
+ADD CPU  
+ADD DISC $DATA  
+ADD DISC $BUYER  
+START MEASUREMENT $DATA.PERF.NOV04  
+EXIT
```

Para parar la medición se usa el comando STQP de MEASCOM

EXAMEN DE LOS DATOS

Para examinar la colección de datos se entra a MEASCOME y se usa el comando ADD MEASUREMENT para identificar la medición del archivo de datos que se desea examinar y el comando LIST para desplegar la información contenida en el archivo. Por ejemplo:

```
+ADD MEASUREMENT $DATA.PERF.NOV04  
+LIST CPU 1
```

MEASURE solamente colecta y despliega la información. Será necesario analizar esta para darle uso efectivo.

DESCRIPCION DEL SUSBSISTEMA DE MEASURE

El subsistema de MEASURE consiste de un conjunto de procesos diferentes:

-Comandos de proceso de interface (MEASCOME). Acepta y procesa los comandos controlados por MEASURE.

-Cordinación de procesos (MEASMON). Empieza y termina la medición y el subsistema de MEASURE.

-Contador de control de procesos (MEASCTL). Localiza e inicializa los contadores, escribe periodicamente el valor de los contadores en el archivo de datos. Cuando se empieza con el subsistema de MEASURE, el proceso MEASMON crea un proceso MEASCTL dentro de cada cpu del sistema local.

-Proceso de manejador de archivos (MEASFH). Construye registros de contadores desde los datos dentro de la medición del archivo de datos. MEASFH crea e inicializa el archivo de datos. Cuando se requiere la medición de los datos, MEASFH lee los datos desde el archivo y traslada estos dentro de un registro con formato de contador externo.

Los siguientes archivos son parte del sistema de MEASURE:

-MEASDDL. Contiene la definición del registro DDL (Data Definition Language) para su uso por el lenguaje query ENFORM. MEASURE escribe la ejecución de datos en un archivo sin estructura, pero se puede usar MEASCOM para copiar la ejecución de datos dentro de un archivo estructurado que puede ser leído usando ENFORM. Este archivo DDL describe el formato de registro para archivos estructurados.

-MEASDECS. Contiene la declaración de la estructura y el la definición de los valores literales usados por MEASURE.

-MEASERRS. Contiene el texto de los mensajes de error.

-MEASHELP. Contiene textos de ayuda.

COMO TRABAJA MEASURE

Los siguientes pasos describen como MEASURE inicia la medición y colección de datos:

1. Se entra a MEASCOM, definiendo la configuración de medición y el inicio de esta.

2. MEASCOM crea un proceso MEASFH y la configuración de medición para este proceso.

3. MEASFH abre el archivo de datos (especificado cuando se inicio la medición), convierte la configuración de medición a un formato interno y pasa la configuración a MEASMDN.

4. MEASMDN pasa la configuración para cada proceso MEASCTL. (Cuando el subsistema MEASURE fue iniciado, MEASCOM creo un proceso MEASCTL en cada CPU).

5. El proceso MEASCTL localiza e inicializa el contador para cada entidad dentro de la configuración de medición y escribe estos valores iniciales en el archivo de datos. Si la configuración de medición especifica una entidad que ya esta bajo medición, MEASCTL escribe el valor que tenga el contador de esa entidad como el valor inicial en el archivo de datos. Esto no asigna ni inicializa un nuevo contador para la entidad.

6. Como el sistema continua corriendo, varias operaciones dentro de los procesos del sistema actualizan a los contadores. Al finalizar la medición o los intervalos de recolección de datos especificados por el usuario, MEASCTL escribe el valor de los contadores en el archivo de datos.
(ver diagrama)

Se puede leer datos desde un archivo de datos de medición o desde contadores dentro del espacio designado en el sistema. La medición del archivo de datos puede ser asociada con una medición activa o inactiva. El contador dentro del espacio del sistema debe estar asociado a una medición activa.

-La lectura de datos desde un archivo de datos de medición refleja los datos colectados desde el inicio de la medición.

Para leer el contador desde el archivo de datos, MEASCOM requiere de los datos que se encuentran en MEASFH. MEASFH lee el valor del contador y el valor del contador inicial, proporcionando la diferencia entre estos.

-La lectura de datos desde un contador activo refleja la colección de datos desde un contador que fue inicializado.

EJECUCION

MEASURE generalmente usa menos del 1% el CPU. La mayor parte de el tiempo de proceso es por la escritura en disco. MEASCTL debe escribir un registro por entidad por intervalo; adicionalmente MEASCTL debe escribir un registro por cada tiempo de inicio o fin transitorio de una entidad. Se puede tener un control de los datos que MEASURE debe escribir por una cuidadosa selección de una colección de intervalos y configuración de mediciones.

-Intervalos de colección. Los intervalos de colección pequeños hacen que haya mas datos para escribir en el archivo de datos y hace mayor el efecto en la ejecución del sistema. Para minimizar los efectos de MEASURE se especifica un intervalo de colección con una parcialidad grande.

-Número de entidades de medición. Se debe medir las entidades que se necesitan solamente.

Las mediciones concurrentes afectan la ejecución del sistema porque el proceso MEASCTL debe escribir más registros en el archivo de datos de medición.

CONSIDERACIONES PARA LA INTERPRETACION DE MEASURE

MENSAJES	Se refiere a una transacción.
NUMERO DE ACCESOS*	Se refiere al número de writes y reads que realizo el proceso.
TIEMPO PROMEDIO ESPERA DE MENSAJES	Se refiere al tiempo promedio que una transacción espera para ser atendida por el CPU (microsegundos).

MSGS RECIBIDOS*

Se refiere al número total de mensajes que recibió un proceso.

TIEMPO MAXIMO

Es el tiempo mayor que una transacción espera para ser atendida por el CPU (microsegundos)

FALLAS DE PAGINAS*

Es el número de veces que fue necesario ir a disco para traer otra página a memoria principal.

% CACHE LECTURA*

Es el porcentaje de veces que la información a leer fue encontrada en memoria cache.

%CACHE DE ESCRITURA*

Es el porcentaje de veces que la información a escribir fue depositada en memoria CACHE.

% USD CACHE*

Es el porcentaje de veces que la información fue encontrada en memoria CACHE.

* NOTA : El dato es incremental

A continuación se muestran ejemplos de reportes obtenidos (reporte 5 y 6).

4.4.2 REPORTES DE MEASURE

MEDICION DE PROCESOS
 FECHA DE REPORTE : 13/12/1988

INICIO MEDICION	FIN MEDICION	NUMERO PROCESADOR	NOMBRE PROCESO	NOMBRE ARCHIVOS	NUMERO ACCESOS	% CONSUMO CPU	TIEMPO PROMEDIO ESPERA MENSAJES	MSGS RECIBIDOS	TIEMPO PROMEDIO ESPERA MENSAJES	FALLAS PAGINAS
02:07:24PM	03:30:02PM	0	\$ADD	\$ADIR	0	00.22	0	0	0	0
	03:30:02PM			\$000101	0	00.22	0	0	0	0
02:07:25PM	03:30:02PM	0	\$ADI	\$D	4	00.00	66609	6	2	0
	03:30:02PM			\$DATA HARDCOPY	2	00.00	66609	6	2	0
	03:30:02PM			\$RECEIVE	6	00.00	66609	6	2	0
02:07:24PM	03:30:01PM	1	\$ADD	\$ADIR	0	00.00	0	0	0	0
	03:30:01PM		\$ADD	\$ADIR	0	00.00	0	0	0	0

MEDICION DE CONSUMO DE CACHE
 14/12/88

NOMBRE ARCHIVO	% USO CACHE LECTURA	% USO CACHE ESCRITURA	% USO CACHE
\$D		0	0
		0	0
\$CLXDSV AUDITL MA000125		0	0
		0	0
\$DATA HARDCOPY		0	0
\$RECEIVE	0		0
	0		0
	0		0
	0		0
	0		0

STARTING AND STOPPING MEASUREMENTS				
SIZE	SMALL	MEDIUM	LARGE	
ELAPSED TIME	15.1S	16.8S 1.1X	17.3S 1.1X	
PROCESS TIME	103.1M	119.3M 1.2X	170.1M 1.6X	(1)
PROCESS BUSY	.7%	.7% 0X	1.9% 1.4X	(2)
BYTES SENT	7104	14125 2X	21575 3X	(3)
10 MINUTE MEASUREMENT				
SIZE	SMALL	MEDIUM	LARGE	
ELAPSED TIME	10.3M	10.3M	10.3M	
PROCESS TIME	126.5M	119.3M .9X	153.5M 1.2X	(4)
PROCESS BUSY	.7%	.02%	.02%	(5)
BYTES SENT	7104	14125 2X	21575 3X	(3)

INTERVALS				
INTERVAL	10MIN	5MIN	1 MIN	10 SEC
NO. OF INTERVALS	1	2 2X	10 10X	60 60X
ELAPSED TIME	10.3M	10.9M 1.1X	10.4M 1.0X	10.3M 1.0X
PROCESS TIME	153.5M	227.5M 0X	404.1M 1.4X	1656.8M
PROCES BUSY	.02X	.04X 2X	.06X 3X	.27X 13.5X
BYTES SENT	21536	37051 1.8X	141177 6.6X	811265 37.7X
CONCURRENT MEASUREMENTS				
MEASUREMENTS	1	5 5X	30 30X	60 60X
ELAPSED TIME	10.3M	10.9M	14.9M	20.6M
PROCESS TIME	153.5M	776.6M 5.1X	5642.5M 36.8X	14115.0M 92.0X
PROCESS BUSY	.02X	.12X 6X	.63X 31.5X	1.14X 57X
BYTES SENT	21536	168857 7.8X	1493620 69.4X	4039327 187.6X

REPORTE (1 DE 2)

4.5 PATHVU

4.5.1 DESCRIPCION DE LA HERRAMIENTA

Pathvu es una herramienta disponible para los Sistemas Unisys de las Series A y V, que sirve para evaluar la complejidad y estructura de programas en COBOL68 y COBOL74, usada para el desarrollo, documentación y mantenimiento de los mismos. Sin embargo puede ser utilizado en hardware de mainframe incluyendo IBM, Honeywell DFS8/88 y Unisys 1100.

Esta herramienta hace un análisis sobre el código fuente, ó resultados obtenidos de una compilación, de programas sintácticamente correctos, el cual es procesado en batch a través de WFL. Por medio de ella, se pueden obtener métricas y gráficas que nos indican el grado de complejidad y estructura del ó de los programas procesados, así como reportes a detalle y generales cuando es necesario, por ejemplo: Reportes técnicos que proveen a detalle mapas de la lógica y estructura de un programa ó programas, maneja reportes de medición de calidad en terminos de complejidad y modularidad, reportes enfocados a un programa individual, aplicaciones, sistemas o librerías.

El PATHVU nos proporciona los siguientes beneficios:

-Reduce el tiempo requerido para entender la lógica del programa, permitiendo mayor rapidez y más accurate en los cambios.

-Reduce el costo de mantenimiento de documentación de los programas por la modificación de los comentarios en línea con cada corrida de PATHVU.

-Reduce el tiempo requerido para el debugging por diagnóstico pinpointing el cual causa algunos problemas.

-Salva el tiempo de entrenamiento dentro de la orientación de un nuevo conjunto para técnicas estandars.

-Salva el tiempo y simplifica las pruebas por isolating de los problemas lógicos despues de la ejecución del programa con datos prueba.

-Salva el tiempo de computadora por la reducción del número de compilaciones y pruebas requeridas.

-Reduce el manejo de tiempo de proyectos dentro de reviewing, análisis, aceptación e implementación de la aplicación del sistema.

Se usa a través de pantallas, en cada una de ellas aparecen una serie de campos en los que se da información referente al sistema.

Las pantallas son:

-Pantalla de inicialización : A partir de la cual se tiene acceso a las pantallas operativas del analizador. Esta pantalla se usa para establecer opciones y defaults globales para el sistema. Aquí se genera el número de sesión con el cual se puede identificar las tareas procesadas por pathvu.

-Pantalla del menú principal : En este se selecciona la tarea a realizar. Las opciones son:

- PATH Análisis estático de programa(s) en Cobol.
Procesa un programa en cobol o un directorio de programas, a los que se les realiza el análisis estático.
- PSEC Reporte del análisis por sectores.
Emite una gráfica (reporte del análisis de sectores) por cliente o global, la cual nos indica el número de programas evaluados y el sector en que caen.
- PMET Reporte del análisis de métricas y diagnósticos.
Emite el reporte del análisis de métricas y diagnósticos por cliente o global, de acuerdo a los programas analizados.
- FCOM Reporte comparativo de programas del sistema.
Emite el reporte comparativo del sistema.
- PCHG Consulta, modificación y eliminación de registros en la base de datos.
Modifica campos y consulta o borra registros de la base de datos.
- TRAK Consulta y detección de errores de las tareas procesadas por PATHVU.
Permite la consulta y detección de errores de las tareas procesadas por PATHVU, a través del estado en el que se encuentran.
- TDEL Consulta y eliminación de una tarea específica procesada por PATHVU.

INSTRUCCIONES DE USO

PATHVU se usa a través de pantallas. En cada una de ellas aparecen una serie de campos en los que se da información referente al sistema. Algunos de estos campos presentan un valor por default que no es necesario cambiar, a menos que sean necesarios diferentes parametros.

El flujo de utilización es como el mostrado en la figura 4.2., así como los procesos de pathvu en la figura 4.3.

FLUJO DE UTILIZACION DEL PATHUU

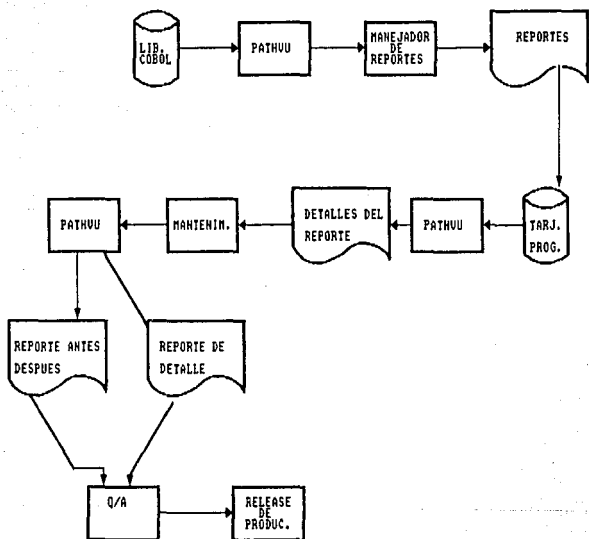


FIGURA 4.2

PROCESOS DEL PATHVU

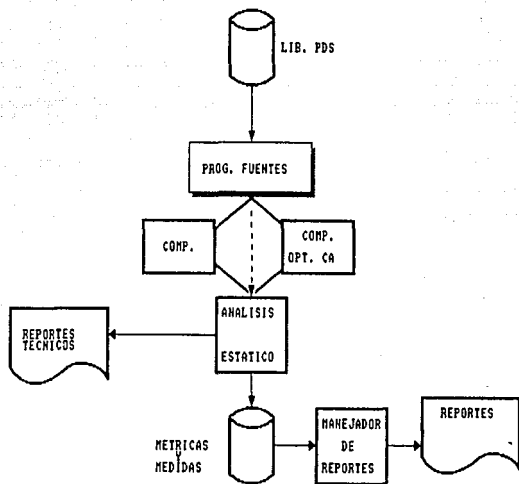


FIG. 4.3

4.5.2. REPORTES DE PATHVU

```

* .....
* . LOOPING PERFORM RANGE VIOLATION FROM 03097 0020-READ-INPUT-CARDS.
* . CONDITIONAL TRANSFER FROM 03342 0170-BYPASS-ORDER.
* . FALL THRU FROM 03097 0020-READ-INPUT-CARDS.
* . BEGIN PROCESSING LOOP RETURNING FROM 03097 0020-READ-INPUT-CARDS.
* .....

```

```

03106 013830 0030-PROCESS-INPUT-CARDS. PGM01790
03107 013840 IF OUR-ORDER-NUMBER IS NOT EQUAL TO LAST-ORDER-NR-READ PGM01800
03108 013850 PERFORM 0200-ORDER-NR-BREAK-ROUTINE PGM01810
03109 013860 THRU 0210-RETURN-FROM-SCAN-EXIT PGM01820
03110 013870 ELSE PGM01830
03111 013880 GO TO 0040-PROCESS-ITEMS. PGM01840
03112 013890 MOVE BI-SHIP-ADDRESS-TAG TO WS-SHIP-ADDRESS-TAG. PGM01950
03113 013900 MOVE BI-C-COMMENT-COUNT TO WS-C-COMMENT-COUNT. PGM01860
03114 013910 MOVE BI-M-COMMENT-COUNT TO WS-M-COMMENT-COUNT. PGM01870
03115 013920 MOVE BI-D-COMMENT-COUNT TO WS-D-COMMENT-COUNT. PGM01880
03116 013930 MOVE BI-ATTN-LINE TO WS-ATTN-LINE. PGM01890
03117 013940 MOVE BI-GOV-OR-HOSP-PURCH-ORD TO WS-GOV-OR-HOSP-PURCH-ORD. PGM01900
03118 013950 MOVE ZERO TO SAVE-DUP-ORDER. PGM01910
03119 013960 MOVE I TO SUB-D-BO. PGM01920
03120 013970 MOVE SPACES TO WS-D-SPECIAL-BO. PGM01930
03121 013980 MOVE SPACES TO WS-SAVE-C-RECS. PGM01940
03122 013990 MOVE ZERO TO SUB-C-WS. PGM01950
03123 014000 IF (NO-CHARGE-ORIGINAL OR NO-CHARGE-BACKORDER) PGM01960
03124 014010 AND NOT REENTER-ORDEROT EQUAL TO LAST-ORDER-NR-READ PGM01970
03125 014020 MOVE '14' TO SU-CODE PGM01980
03126 014030 PERFORM 0220-SAVE-SHOW-REMARK THRU 0230-SAVE-SHOW-EXIT. PGM01990
03127 014040* PGM02000
03128 014050* PGM02010

```

```

* .....
* . LOOPING PERFORM RANGE VIOLATION FROM 05877 0320-BYPASS-ITEM
* . 07463 1030--CONTINUE-PREFERENCE-SCAN, 07511 1060-MOVE-KEY,07533 1070-CONTINUE-SCAN*
* . CONDITIONAL TRANSFER FROM 03106 0030-PROCESS-INPUT-CARDS.
* . UNCONDITIONAL TRANSFER FROM 07463 1030-CONTINUE-PREFERENCE-SCAN.
* . FALL THRU FROM 03106 0030-PROCESS-INPUT-CARDS
* . BEGIN PROCESSING LOOP RETURNING FROM 03914 0050-NEXT-CHECK-AREA.
* . 05877 0320-BYPASS-ITEM, 07463 1030-CONTINUE-PREFERENCE-SCAN, 07511 1060-MOVE-KEY
* . 07533 10-70 CONTINUE-SCAN.
* .....

```

```

03129 014060 0040-PROCESS-ITEMS. PGM02020
03130 014070* PGM02030
03131 014080* ***** PGM02040
03132 014090* * PGM02050

```

REPORTE

UNENTERED PROCEDURES

SEQ PROCEDURE NAME

00751 REPORT-SPACING

UNEXECUTABLE STATEMENTS

SEQ SOURCE STATEMENT

00751 075100 STO PRUN.

RUNAWAY PATHS

PATH BRANCH TO NUMBER

99999

PERFORM RANGE VIOLATIONS

BRANCH FROM TO

00709 22-SORT-RTM	00747 24-NOT-RECOVERABLE
00731 CHECK-HDR	00322 A1-BUILD-TAPE-PR

LOOPING PERFORM RANGE VIOLATIONS

BRANCH FROM TO

00346 A1-34-A	00275 A1-4-0
00346 A1-34-A	00147 A1-34-4
00346 A1-34-A	00357 A1-0AS
00674 Z1-WRITE-DELT-HDR	00275 A1-4-0

RECURSIVE PERFORMS

PERFORM RANGE

PERFORMED BY

00667 B3-ENDT 00275 A1-BEGIN

UNRESOLVED PROCEDURE EXITS

FALL THRU TO

00623 A7-93-C

ALTERS

PROCEDURE ALTERED

TO PROCEED TO

00341 A1-34-B

00342 A1-34-2

PERFORM THRU

PROCEDURE PERFORMED

PERFORMED BY

00709 Z2-SORT-RTN

THRU Z2-SORT-RTN-EXIT.....

00307 A1-15-0

FALL THRU PROCEDURES

FALL THRU TO

FROM

00275 A1-4-0

00293 A1-4-0

00259 A1-START-JOB-STEP

00301 A1-13-0

TRANSFER CREATED LOGIC LOOPS

LOOP BEGINS AT

RETURNS FROM

00275 A1-4-0

00275 A1-4-0

00276 A1-9-0

00301 A1-27-2

PERFORM NO THRU

PROCEDURES PERFORMED

PERFORMED BY

00740 Z3-WRITE-HDR

00731 CHECK-HDR

REPORTE

BMM
SUMMARY DIAGNOSTICS

PAGE NO 1
DATE 10/27/87
TIME 13.39.05

----- DIAGNOSTICS -----

PROGRAM	SYS	UNENTERED PROCEDURES	RUNAWAY PATH	PERFM RANGE VIOL.	LOOPING PERFORM RANGE VIOL.	RECURSIVE PERFORMS	UNRESOLVED PROCEDURE EXITS	ALTERS
1	(PATHUU) S016	SOURCE/LIBRARY/BARREBCAELEC	RELO07/MTP002	0	0	0	0	0
2	(PATHUU) S016	SOURCE/LIBRARY/CTESPRODS	RELO07/MTP003	0	0	0	0	0
3	(PATHUU) S016	SOURCE/LIBRARY/DOSALPASO	RELO07/MTP001	0	0	0	0	0
4	(PATHUU) S016	SOURCE/LIBRARY/EDOCACH	RELO07/MTP001	0	0	0	0	0
5	(PATHUU) S016	SOURCE/LIBRARY/HISTORIA	RELO07/MTP001	0	0	0	0	0
6	(PATHUU) S016	SOURCE/LIBRARY/MOUTOSCHE	RELO07/MTP001	0	0	0	0	0
7	(PATHUU) S016	SOURCE/LIBRARY/PIN	RELO07/MTP001	0	0	0	0	0
8	(PATHUU) S016	SOURCE/LIBRARY/PINLOG	RELO07/MTP004	0	0	0	0	0
9	(PATHUU) S016	SOURCE/LIBRARY/SISTIPCTA	RELO07/MTP001	0	0	0	0	0
10	(PATHUU) S016	SOURCE/LIBRARY/SUBRUTFECHABIL	RELO007/MTP001	0	0	0	0	0
11	(PATHUU) S016	SOURCE/LIBRARY/S015AECTES	RELO07/MTP005	0	0	0	0	0
12	(PATHUU) S016	SOURCE/LIBRARY/S015LINEA	RELO07/MTP005	0	0	0	0	0
13	(PATHUU) S016	SOURCE/LIBRARY/UNOALPASO	RELO07/MTP001	0	0	0	0	0
14	(PATHUU) S016	SOURCE/P010	RELO07/MTP014	0	0	0	0	0
15	(PATHUU) S016	SOURCE/P015	RELO07/MTP014	0	0	0	0	0
16	(PATHUU) S016	SOURCE/P020	RELO07/MTP002	0	0	0	0	0
17	(PATHUU) S016	SOURCE/P030	RELO07/MTP002	0	0	0	0	0
18	(PATHUU) S016	SOURCE/P040	RELO07/MTP003	0	0	0	0	0
19	(PATHUU) S016	SOURCE/P050	RELO07/MTP001	0	0	0	0	0
20	(PATHUU) S016	SOURCE/P060	RELO07/MTP003	0	0	0	0	0
22	(PATHUU) S016	SOURCE/P070	RELO07/MTP003	0	0	0	0	0

COPYRIGHT 1983 BY PEAT MARWICK MAIN & CO. ALL RIGHTS RESERVED.
CONFIDENTIAL UNPUBLISHED PROPERTY OF PEAT MARWICK MAIN & CO. LIMITED DISTRIBUTION TO AUTHORIZED PERSONS ONLY.
PATHUU VERSION 03.09.00W

REPORTE

PATHU USER
 SECTOR ANALYSIS
 33 PROGRAMS 66,716 LINES OF CODE
 SAMPLE PROGRAMS

PAGE NO 2
 DATE 10/27/87
 TIME 13.39.05

C
O
M
P
L
E
X
I
T
Y

C
O
M
P
L
E
X

250

N
O
N
/
C
O
M
P
L
E
X

0 PROGRAMS
 0.0% OF TOTAL
 0 LINES OF CODE
 0.0% OF TOTALS
 0 VERBS
 0.0% OF TOTALS

SECTOR 2

9 PROGRAMS
 27.3% OF TOTAL
 36,239 LINES OF CODE
 54.4% OF TOTALS
 19,047 VERBS
 65.0% OF TOTALS

SECTOR 4

SECTOR 1

SECTOR 3

0 PROGRAMS
 0.0% OF TOTAL
 0 LINES OF CODE
 0.0% OF TOTALS
 0 VERBS
 0.0% OF TOTALS

13 PROGRAMS
 39.4% OF TOTAL
 12,668 LINES OF CODE
 19.0% OF TOTALS
 4,925 VERBS
 16.8% OF TOTALS

STRUCTURED,

100

UNSTRUCTURED

ARCHITECTURE

COPYRIGHT 1983 BY PEAT MARWICK MAIN & CO. ALL RIGHTS RESERVED.
 CONFIDENTIAL UNPUBLISHED PROPERTY OF PEAT MARWICK MAIN & CO. LIMITED DISTRIBUTION TO AUTHORIZED PERSONS ONLY.
 PATHU VERSION 03.08.00W

REPORTE

CAPITULO V

EVALUACION Y CERTIFICACION DE CALIDAD DE UN SISTEMA DE APLICACION.

5.1 INTRODUCCION

El sistema de Ordenes de Inventario de Ordenes de Pago Bancarias, es el sistema seleccionado para realizar la certificación de calidad. Se considera que se cuenta con toda la documentación necesaria, manuales de usuario y operación, describiéndose brevemente el desarrollo del mismo, aplicándose las métricas de certificación de calidad a la implementación del sistema.

5.2. DESCRIPCION DEL SISTEMA DE INVENTARIO DE ORDENES DE PAGO BANCARIAS.

En base a estudios orientados a la investigación del flujo de remesas de que envían al país los mexicanos que trabajan en Estados Unidos, así como estudios en cuanto a la concentración de población de trabajadores agrícolas de procedencia mexicana que se encuentran en Estados Unidos de manera legal o ilegal, se vió la necesidad de desarrollar un medio de transferencia de fondos adecuado y confiable.

Como solución se encontró, basándose en estudios efectuados para determinar los medios existentes para transferir fondos a México, una orden de pago bancaria.

La venta de esta orden de pago se realiza en diversos comercios establecidos, denominados agentes, que cuentan con una terminal punto de venta conectada a una red. La operación de estas ordenes de pago se realiza bajo un concepto totalmente automatizado, desde su emisión hasta su recepción para pago en una sucursal bancaria.

5.2.1. CICLO DE VIDA DEL SISTEMA.

En el manejo de las órdenes de pago, estas llevarán un ciclo de vida, el cual irá describiendo por donde va pasando y en que estado se encuentra la orden de pago. Esta secuencia de estatus irá desde su orden de impresión hasta que sea pagado, de tal manera que se lleve control, a base de registros, de la secuencia normal del ciclo o cambios de secuencia como prevenciones de pago.

5.1.1.1. DIAGRAMA DE CICLO DE VIDA DEL SISTEMA

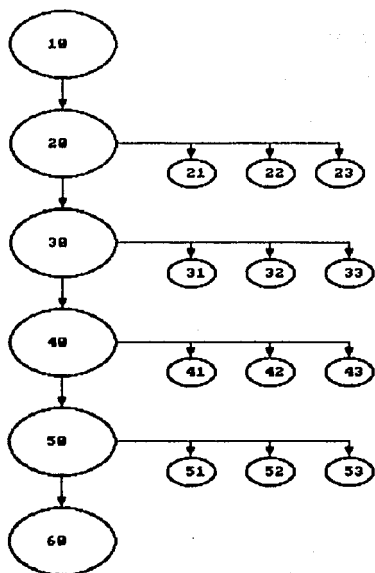


FIGURA 5.1

5.2.1.2 DESCRIPCION DEL CICLO DE VIDA.

ESTATUS	DESCRIPCION
10	Orden de impresión.
20	Ordenes de pago en bóveda.
21	Prevención de pago de anuladas en bóveda.
22	Prevención de pago de perdidas en bóveda.
23	Prevención de pago de robadas en bóveda.
30	Ordenes de pago en distribución.
31	Prevención de pago de anuladas en distribución.
32	Prevención de pago de perdidas en distribución.
33	Prevención de pago de robadas en distribución.
40	Ordenes de pago en localidad de venta.
41	Prevención de pago de anuladas en loc. de venta.
42	Prevención de pago de perdidas en loc. de venta.
43	Prevención de pago de robadas en loc. de venta.
44	Orden de pago anulada por la terminal pto. vta.
50	Orden de pago vendida.
51	Prevención de pago de robada al cliente.
52	Prevención de pago de perdida por el cliente.
53	Prevención de pago de cancelada por el cliente.
60	Orden de pago pagada.

5.2.2 DESARROLLO DEL SISTEMA.

El sistema empieza con la requisición de órdenes de pago, con lo cual se genera una orden de impresión de documentos, la cual se mandará a un imprenta, que será la responsable de la correcta impresión de los mismos. Estos documentos se encuentran en fajillas, las cuales se mandarán posteriormente a las bóvedas para su almacenamiento.

Cuando son recibidas las fajillas se dan de alta en los archivos, poniéndose disponibles para la distribución a los agentes, dependiendo de sus necesidades y requerimientos.

Cuando los agentes ya tienen en su poder las fajillas solicitadas, se manda una confirmación al sistema para señalar que las fajillas ya se encuentran en localidad de venta y están disponibles para la compra.

Después de esta confirmación, el manejo de las órdenes de pago cambia, ya que antes los documentos eran manejados por paquetes y ahora se manejarán de manera individual.

Las agencias contarán con una terminal punto de venta con un sistema de captura de información, que permitirá realizar una transacción más rápida y eficiente. En este momento pueden generarse dos estatus, que el documento sea vendido, o que sea anulado por la terminal por algún problema con la impresión.

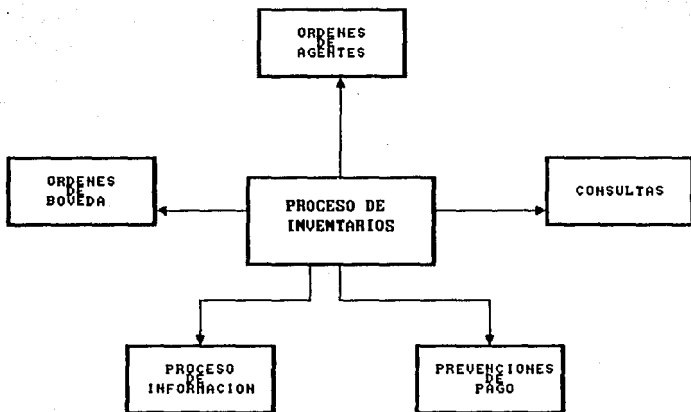
Cuando el documento ya ha sido vendido, el cliente será el responsable de mandarlo a su beneficiario en México, para su cobro. De esta manera, al recibir el documento, el beneficiario podrá acudir a cualquier sucursal autorizada para su cambio, siempre y cuando no exista la notificación de alguna prevención de pago sobre ese documento.

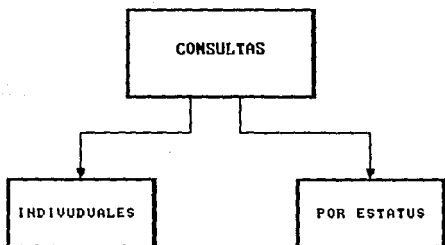
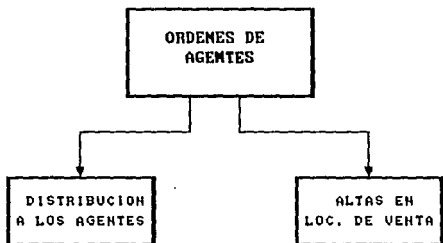
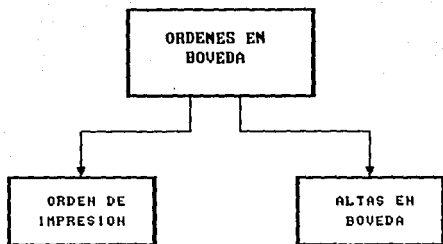
Los archivos son actualizados cada día en la noche verificándose las discrepancias e inconsistencias en la información, generándose los reportes correspondientes para su análisis.

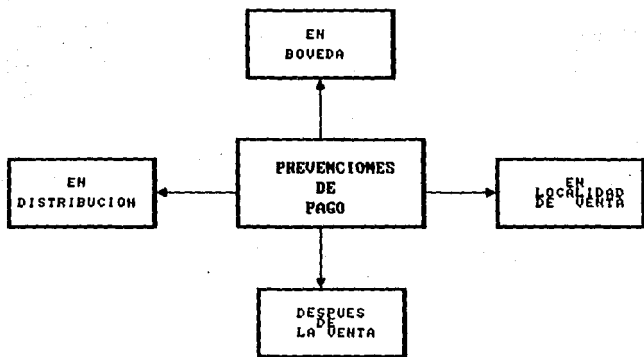
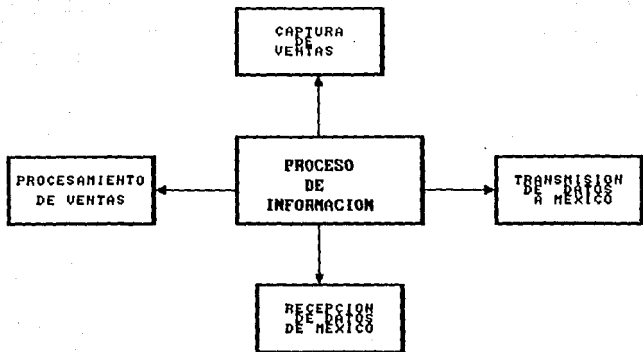
Las órdenes de pago son susceptibles de prevención de pago, ya sea por anulación, pérdida o robo en cualquier de los estatus del ciclo de vida, ya sea por paquete o individual. Estas prevenciones son registradas por el sistema, actualizando los archivos, para que junto con las ventas realizadas sean transmitidas a México, para de la misma manera, verificar los pagos realizados en las sucursales. Es mandado otro archivo de regreso a Estados Unidos con los movimientos pagados y los movimientos que tuvieron alguna inconsistencia antes de su pago.

Se cuenta, además, con un módulo de consultas, la cual podrá ser individual por número de la orden de pago de manera detallada, o consultas por estatus, de manera general.

5.1.3 MODULOS DEL SISTEMA DE INVENTARIO DE ORDENES DE PAGO







5.2.4 DESCRIPCION DE MODULOS.

ORDENES EN BOVEDA

Se cuenta con la opción de generar la orden de impresión, dando la cantidad deseada, obteniéndose un reporte donde se indican los consecutivos del paquete a imprimir, así como la fecha de la requisición. En este módulo, se captura además, la confirmación de la alta en bóveda, cuando los paquetes son entregados por la imprenta, registrándose la fecha de recepción.

ORDENES DE AGENTES

Este módulo maneja la distribución de los paquetes a los agentes, registrándose a que agente fue asignado cada paquete, los consecutivos del paquete y la fecha de envío, para posteriormente, con la confirmación del agente registrar la fecha de alta en localidad de venta.

PROCESAMIENTO DE INFORMACION.

Este módulo, cuenta con la captura manual de documentos, para aquellas órdenes de pago que no hayan sido registradas directamente por la terminal punto de venta, siendo necesario capturar el número de la orden de pago, número del agente, importe en dolares y pesos, comisión del agente, tipo de cambio y tipo de movimiento. Las ventas capturadas en este módulo son almacenadas en un archivo, para su proceso posterior, junto con las ventas transferidas automáticamente por la red, verificándose cada orden de pago, para ver si es correcta la venta, generándose un archivo de errores cuando se presente alguna inconsistencia con algún documento, como sea un tipo de cambio incorrecto o la venta de un documento por un agente diferente al asignado.

Estos archivos son transmitidos a Mexico, para actualizar las prevenciones de pago mandadas, así como las ventas correctas para poder ser pagadas posteriormente en alguna sucursal. Los documentos que sean pagados, serán retransmitidos a Estados Unidos, así como los rechazos generados en los procesos, para la actualización de los archivos.

PREVENCIONES DE PAGO.

Las prevenciones de pago, podrán hacerse por paquete o de manera individual, dependiendo del estatus en el que se encuentren. La prevención de pago desvía la orden de pago de su ciclo de vida normal, terminándose en ese momento su desarrollo. Una orden de pago puede ser puesta en prevención de pago cuando se recibe una notificación de anulación, robo o pérdida, por lo que ya no son susceptibles de pago, evitándose que un documento que ha sido marcado como robado, por ejemplo, intente ser cobrado más tarde.

CONSULTAS.

En este módulo se realiza la consulta de las órdenes de pago individuales, desplegándose su información respectiva, señalando su ciclo de vida y su estatus actual. La consulta por estatus, genera la información de todos los documentos que en ese momento se encuentren en el estatus seleccionado.

5.3 DOCUMENTACION TECNICA

En el anexo 3 y 4 se pueden ver algunos de los diagramas de flujo del sistema y programas del Sistema de Ordenes de Pago Bancarias, a los cuales se les aplicaron las métricas de certificación de calidad para después obtener la evaluación del sistema.

5.2 APLICACION DE LAS METRICAS AL SISTEMA

Conteo de operadores y operandos como métricas básicas

PROGRAMA MONEY O

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
Close all	2	store to	18
&	6	=	108
select	11	use	11
index	5	,	94
restore from	3	additive	3
do	37	<>	10
do while	30	marques	9
with	17	space	1
@	44	say	47
<	1	if	5
substr()	2	+	4
picture	9	stuff	1
str()	1	case	37
endcase	9	endif	5
endo	30	at()	8
get	8	read	8
val()_	1	docase	9
ordp	1	prnt	1
zap	1	ship	1
sloc	1	rep_dstr	1
ventas	1	init	1
tran_mex	1	lect_mex	1
bove	3	tripsami	1
dste	3	tripsdst	1
stp_pay	4	tripsuta	1
sustit	1	cons	1
constat	1	return	1
else	1	otherwise	2

PROGRAMA MONEY O

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
" "	171	.F.	4
'2'	10	'3'	6
'3'	6	'4'	6
'5'	3	'6'	2
'7'	3	'8'	2
'9'	10	'0'	7
sel_emis	2	sel_dstr	2
sel_mo	2	sel_ciclo	2
sel_ag	1	sel_trans	1
sel_dscr	1	sel_prod	2
sel_tab	2	flag	2
emision	2	dstr	2
agn_ds	1	mo	2
no_mo	1	mo_fecha	1
mo_fchpo	1	ciclo_vi	3
tabla	4	ds_tabla	1
set_tabla	1	esta_bien	21
opcion	89	conta	4
caso	3	pos	11
ren	13	inv_okt	1
"012345"	1	tempo	2
bandera	4	codiqo	8
7	2	30	33
0	9	5	2
1	9	9	2
31	5	2	12
3	1	4	1
35	6	57	6
	21	8	6
10	6	12	5
18	5	14	3
16	3	20	3
6	1	23	2
32	4	19	5
54	3	999999	1
999998	1	999997	1
999996	1	4999	1
9998	1	9996	1
9997	1	"L"	1
"C"	1	"I"	1
"R"	1	"D"	1
"S"	1	"SOLD"	5
"Sistema de orden de pago"			1

OFERANDOS

OCURRENCIAS

"Inventarios"	1
"0. Regreso Menu anterior"	1
"1.-Ordenes en Bodega"	1
"1.-Ordenes de agentes"	1
"Procesamiento de Informacion"	1
"Prevencion de pagos"	1
"Consultas"	1
"012345"	1
"Ordenes en Bodega"	1
"Menu"	1
"0. Regreso al menu anterior"	1
"1. Ordenes de impresion"	1
"2.Altas en Bodega"	1
"teclea tu opcion"	1
"012"	1
"Ordenes de Agentes"	1
"1.Distribucion a los agentes"	1
"2 Altas en localidades de Ventas"	1
"Procesamiento de informacion"	1
"1.Captura de ventas"	1
"2.Procesamiento de datos"	1
"3.Transmision de datos"	1
"4.Recepcion de datos Mexico"	1
"Prevencion de pago en Bodega"	1
"1.En Bodega"	1
"2.En distribucion"	1
"3.En localidad de venta"	1
"4.Robada al cliente"	1
"5.Perdida por el cliente"	1
"6.Cancelacion despues de la venta"	1
"7.Ordenes de pago a ips despues de venta"	1
"01234567"	1
"1.Anuladas "	1
"2.Robadas "	1
"3.Perdidas"	1
"4.En poder IFS"	1
"Orden de pago robada"	1
"Orden de pago perdida"	1
"Orden de pago cancelada"	1
"Orden de pago en poder de IPS"	1
"Orden de pago reembolsada"	1
"sold+stln+lost"	1
"Sustitucion de una orden de pago"	1
"consultas"	1
"Regreso al menu at."	1
"1.Por numero de orden de pago"	1
"2.Por status de la orden de pago"	1

PROGRAMA ORDIF

OPERADORES	OCURENCIAS	OPERADORES	OCURENCIAS
SELECT	5	&	1
USE	4	INDEX	1
STORE TO	3	DATE()	2
=	10	DO	9
MARQUES	1	WHIT	14
,	47	@	22
TO	1	DOUBLE	1
SAY	21	GET	3
PICTURE	6	READ	1
IF	5	.OR .	1
>	1	MSG_ERR	2
RETURN	3	ENDIF	5
SEEK	3	RIGHT()	1
EOF()	2	EOF()	2
SET DEVICE TO SCREEN	1	SET FILTER TO	1
GO BOTTOM	1	ELSE	1
VAL()	5	+	21
APPEND BLANK	1	REPLACE	2
LEFT()	3	SUBSTR	1
CALC_D_C	2	PRINTER	1
TIME()	1	STR()	2
*	1	!	1
AT()	1	<>	1
EJECT	1		

PROGRAMA ORDP

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
" "	37	SEL_PROD	1
PRODUCTO	3	DSC_PROD	1
'000'	1	TIPO_PROD	7
CANTIDAD	8	0	4
FECHA	3	FLAG	1
'0'	1	"INVENTARIOS"	1
"ORDEN DE IMPRESION"	1	4	4
79	1	7	3
10	4	'999'	1
'99,999'	2	13	5
999990	1	DS MOCK	1
3	1	DS_PRODI	1
EMISION	2	CNSC_FINT	15
CNSC_INIT	3	'1'	1
CNSC_FIN	3	ZEROI	3
1	18	ESTA_BIEN	2
2	6	DENOMINAC	2
FCH_EMIS	2	CANT_INI	2
'DOCUMENTO'	2	DOCUMENTO	5
MONEDA	1	DCI	3
DCF	3		2
"HORA"	1	"99:99"	1
TABLA	1	TABLA_MO	1
PRODUCTOP_CVL	1	11	1
"DIVISA:"	1	DESCRIP	4
TABLA_DN	1	34	1
PRODUCTOP_DNMC	1	17	2
'DLS'	1	"5"	1
": "	1	33	2
'99,999,999.99'	1	CNSC_INIT	1
8	2	'FECHA'	1
"NUMERO DEL TIPO DEL PRODUCTO"			1
"CANTIDAD DE PRODUCTO A IMPRIMIR"			1
"FECHA DE IMPRESION"			1
"INTERNACIONAL PAYMENT SYSTEM"			1
"SISTEMAS ORDENES DE PAGOS"			1
"ORDENES DE IMPRESION"			1
"-----"			1
"NUMERO DEL PRODUCTO"			1
"NUMERO DEL DOCUMENTO A IMPRIMIR"			1
"ORDEN DE PAGO INICIAL"			1
"ORDEN DE PAGO FINAL"			1

PROGRAMA PRINT

OPERADORES	OCURENCIAS	OPERADORES	OCURENCIAS
SELECT	4	&	1
USE	3	INDEX	1
=	12	CTOD()	24
DO	7	MARQUES	1
WITH	8	,	28
@	12	SAY	12
GET	2	PICTURE	1
READ	2	IF	4
RETURN	3	ENDIF	4
SEEK	3	RIGHT()	1
VAL()	4	ZERO_I	1
SET FILTER TO	1	+	23
LOCATE FOR	1	DTO()	1
EOF()	1	ELSE	1
-	1	CALC_D_C	2
CLEAR_	1	STR()	3
*	1	;	1
AT()	1	LEFT()	2
<>	1	RTRIM()	1
FIGHT()	1	.F.	1
SI_D_NO	1	REPLACE	1

PROGRAMA PRNT

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
" "	24	SEL_PROD	4
PRODUCTO	4	DSC_PROD	1
TIPO_PROD	8	000	1
FECHA	1	"11"	2
"INVENTARIOS"	1	"PASO A BOVEDA"	1
"9999"	1	7	4
10	2		2
3	1	EMISION	2
EXISTENCIA	1	4	1
MONEDA	1	DENOMINCA	2
999990	1	FCH_PRNT	4
MSG_ERR	1	DS_MONF	1
CNSC_INIT	4	CNS_FIN	3
CANT_INI	2	1	10
DOCUMENTO	7	DCI	3
DCF	3	TABLA	1
TABLA_MO	1	2	3
PRODUCTO_P_CLV	1	DESCRIP	3
TABLA_DN	1	PRODUCTO_P_DNMC	1
'DOCUMENTO'	1	17	1
'DLS'	1	TABLA_DESCRIP	1
"S"	1	33	1
"\$"	1	".00"	1
29	1	12	2
49	1	"999,999"	1
	2	"_"	2
ESTA_BIEN	3	FECHA_PRNT	1
"ORDEN DE PAGO FINAL"			1
"ORDEN DE PAGO INICIAL"			1
"ERROR CON DENOMINACION CON 1 LIMITE VARIABLE"			1
"NUMERO DEL PRODUCTO"			1
"NUMERO DEL TIPO DEL PRODUCTO"			1

PROGRAMA SHIP

OPERADORES	OCURENCIAS	OPERADORES	OCURENCIAS
SELECT	4	&	1
USE	5	INDEX	1
STORE TO	5	DATE ()	1
DO	15	MARQUES	1
WITH	25	,	34
@	22	SAY	16
GET	5	PICTURE	7
READ	2	DO WHILE	1
>	2	ENDO	1
SEEK	1	IF	7
EOF ()	2	DESCO	6
MSG_ERR	5	RETURN	6
ENDIF	7	=	9
SET FILTER TO	2	+	5
LOCATE FOR	1	.AND.	3
<=	1	DTCO ()	1
<>	3	.OR.	3
<	1	CLEAR	1
?	1	CHR ()	1
VAL ()	2	-	3
ZEROI	2	;	1
SI_O_NO	1	REPLACE	4
APPEND BLANK	2	LEFT ()	1
SUBSTR ()	1	RIGHT ()	1
RECND ()	1		

PROGRAMA SHIP

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
SEL_AG	1	AGENTES	2
0	6	CLAVE_AGT	5
FCH_SHIP	2	ENVIOT	12
"000000"	1	"0000"	1
CNS_FINT	1	TOCP_PROD	6
"INVENTARIOS"	1	7	6
10	7	"CLAVE DEL AGENTE"	1
39	5	"99999"	2
"9999"	2	55	1
"TIPO DEL PRODCUTO"	1	9	4
"9,999"	2	46	5
9990	1	17	1
ESTA_BIEN	10	DS_AGNF	1
STAT_AG	2	2	3
DS_AGNBLK	1	4	1
EMISION	1	PRODUCTO	2
1	7	DS_AGNT	1
DS_AGNC1	1	MONEDA	2
DENOMINAC	2	EXISTENCIA	8
99990	1	FCH_PRNT	1
"11"	1	DS_MONF	1
23	1	20	1
21	1	CNS_INIT	4
CNSC_FINT	5	CNS_FIN	4
FECHA_SHIPT	1	"99999999"	1
DS_NCNSC	1	DSTR	1
DSIR_AP	1	CLAVE_AG	1
CANT_INI	1	FCH_SHIP	1
FCH_SHIPT	1	RECO	2
TEMFO	1	"CONSECUTIVO INICIAL"	1
"CONSECUTIVO FINAL PROPUESTO"	1		1
"LOS DATOS EXPUESTOS CORRESPONDEN A UN PAQUETE PROPUESTO"	1		1
"NO ES POSIBLE OBTENER DE BOVEDA UN PAQUETE DEL TAMANO REQUERIDO"	1		1
"DISTRIBUCION DE ORDENES DE PAGO"	1		1
"DISTRIBUCION DE ORDENES DE PAGO"	1		1
"CANTIDAD DE ORDENES DE PAGO"	1		1
"FECHA DE DISTRIBUCION"	1		1
"LA CANTIDAD DEVE SER MENOR A 9990"	1		1

PROGRAMA REP_DSTR

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
SELECT	4	GO TOP	1
IF	2	.NOT.	2
EOF()	2	=	12
DO	6	MARQUES	2
WITH	6	PRINTER	1
RETURN	1	,	12
@	8	SAY	8
DO WHILE	1	.NOT.	1
+	19	ZERDI	1
GOTO	1	VAL()	1
-	1	CALC_D_C	2
PICTURE	2	EJECT	2
ENDIF	2	SKIP	1
ENDO	1	ZAP	1
SET DEVICE TO SCREEN	1	&	1
USE	1		

PROGRAMA REF_DSTR

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
" "	12	.F.	1
TEMPO	2	DC	4
CONTA	4	0	4
RENGLON	11	9	2
'1'	2	7	3
1	8	APUNTADOR	2
DSTR_AF	1	CNS_INI	5
CANT_INI	1	PRODUCTO	3
MONEDA	4	DENOMINAC	4
5	1	CLAVE_AG	1
19	1		2
44	1	68	1
"999,999"	1	55	1
27	1	'9,999'	1
ESTA_BIEN	1	SEL_PROD	1
FLAG	1		
"DISTRIBUCION DE ORDENES DE PAGO"			1
"AGENTE ORDEN PAGO INICIAL ORDEN"			1
"REPORTE"			1
"PAGO FINAL CANTIDAD"			1
"DISTRIBUCION DE OREDENS DE PAGO"			1
"AGENTE ORDEN DE PAGO INICIAL"			1
"ORDEN DE PAGO FINAL CANTIDAD"			1
"TOTAL DE DISTRIBUCIONES"			1

PROGRAMA SLOC

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
STORE	3	DO	164
MARQUES	12	WITH	14
,	19	@	16
SAY	13	GET	4
PICTURE	3	READ	2
SET FILTER TO	1	SET ORDER TO	2
=	3	+	3
VAL()	2	ZEROI	2
GETGE	1	EOF()	1
MSG_ERR	3	DESEO	4
RETURN	4	ENDIF	4
IF	4	<>	2
-	1	DATE()	1
SI_O_NO	1	.NOT.	1
CTOD()	1		

PROGRAMA SLOC

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
" "	16	0	1
CLAVE_AGT	3	"0000000"	1
CNSC_T	7	"0000"	1
TIPO_PROD	3	"INVENTARIOS"	1
7	6	10	11
39	4	"99999"	1
DSTR	1	PRODUCTO	1
MONEDA	1	DENOMINAC	1
EXISTENCIA	1	9990	1
1	1	DS_MONF	1
ESTA_BIEN	6	CLAVE_AG	1
DS_AGNCHK	1	CANT_INI	2
18	1	44	1
"9,999"	1	35	2
CNS_FIN	1	FCH_SHIP	1
"CONFIRMACION EN LOCALIDAD DE VENTA"			1
"CLAVE DEL AGENTE"			1
"CANTIDAD DE ORDENES DE PAGO"			1
"CONSECUTIVO INICIAL"			1
"CONSECUTIVO FINAL"			1
"FECHA DE DISTRIBUCION"			1

PROGRAMA VTAS

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
@	32	=	38
DO	1	.	5
&	2	<>	8
+	9	!	7
*	7	CLEAR	3
MARQUES	1	WITH	22
RETURN	3	GET	16
READ	5	SAY	16
IF	3	ENDIF	3
SELECT	3	USE	2
ZAP	1	APPEND BLANK	2
DO WHILE	4	DATE()	1
PICTURE	10	UPPER()	3
ENDO	4	.AND.	2
DTC()	1	REPLACE	5
STR()	12	STR()	12
LEFT()	2	SUBSTR()	1
RIGHT()1a	3	TRANSAC	1
ELSE	1	VAL()	1
GOTOP	1	COPY TO	1
TYPE	1		

PROGRAMA VTAS

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
" "	50	9	3
1	8	11	7
10	2	0	14
6	2	7	12
36	1	41	2
49	1	24	2
40	1	59	2
61	1	62	1
27	1	42	1
13	5	22	1
15	5	17	5
35	2	53	1
23	5	19	2
65	1	45	1
5	2	2	4
100	7	8	3
16	1	12	1
4	1	FLAG	1
ARCHS	4	TOT_VNT	4
NUM_VND	3	NUM_ANI	4
TOT_CIFS	4	TOT_CAG	4
CHANCE	9	SEL_PRODD	1
VNT_DIA	1	CORRECTO	8
CLAVE_AGT	2	COMIS-ISPT	3
COMIS_AGT	4	PROD_T	3
CNS_CT	3	DIG_T	4
FECHA_T	3	HORAT	3
MINI	3	TIPO_CMBT	3
DOLART	5	CLAVE_OPT	3
STATUS_T	7	PESOS_T	2
VNT_DIA	2	FCH	4
CLAVE_AGT	2	NO_MO	1
FCH_VNT	2	HORA_VNT	2
VALOR_VNT	3	COMIS_IPS	4
COMIS_AG	3	VALOR_MO 0	2
TIPO_CMBIO	2	NUM_TRANS	2
SDF	1	NUM_TRANS	2
"0"	2	"N"	4
"000"	1	"0000000"	1
"00"	2	"99999"	1

PROGRAMA VTAS

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
"9,099,999.99"	2	"999.99"	2
"99,999.99"	1	"!"	3
"S"	4	"V"	1
"000000000"	1	"0000000"	1
"DESEA REALIZAR OTRA VENTA"			1
"TIPO DE CAMBIO (PESOS A DOLARES)"			1
"STATUS: (<S> OLD O V<OID>)"			1
"OPERADOR"			1
"COMISION AGENTE"			1
"COMISION IPS"			1
"VALOR (EN PESOS) \$"			1
"VALOR (N DOLARES) \$"			1
"PROCESAMIENTO DE INFORMACION"			1
"CAPTURA DE VENTAS MANUALES"			1
"TECLEA EL NOMBRE DEL ARCHIVO DE VENTAS"			1
"NUMERO DE LA ORDEN DE PAGO"			1
"CLAVE DE AGENTE"			1
"FECHA DE LA VENTA"			1
"HORA DE LA VENTA"			1

PROGRAMA INIT

OPERADORES	OCURENCIAS	OPERADORES	OCURENCIAS
=	94	+	25
-	13	>	13
!	6	<>	4
<	2	<=	1
/	21	*	2
,	60	&	9
@	61	.NOT.	7
DO	15	WITH	43
USE	11	INDEX	6
ZAP	7	SAY	54
GET	1	READ	1
IF	18	ENDIF	18
APPEND FROM	2	TYPE	1
DO WHILE	4	ENDO	4
CLEAR	7	GOTOP	8
SUM TO	1	COUNT FOR	1
SET ORDER TO	1	PACK	2
PICTURE	21	ELSE	9
SET FILTER TO	1	SEEK	3
APPEND BLANCK	4	REPLACE	9
SKIP	2	WAIT	2
ON ERROR	2	RETURN	2
DATE()	1	RECCOUNT()	1
RECNO()	5	EOF()	4
VAL()	30	SUBSTR()	3
MOD()	1	DTOC()	1
STR()	1	RIGTH()	3
LEFTH()	2	ABS()	1
CTOD()	1	BUSCA	1
DESR	7	GETGE	1
DSTR_AP	1	ZEROI	1
MARQUES	1	MSG_ERR	1
DESEO	1	ICONS	1

PROGRAMA IN11

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
REG_PROC	4	REG_PROCES	4
REG_ALTA	3	REG_RCHZ	2
REG_DSCR	2	IMPOP_PROC	4
IMPOP_RCHZ	1	IMPOP_ALTA	4
IMPOP_DSCR	2	IMPOD_PROC	4
IMPOD_RCHZ	2	IMPOD_ALTA	4
IMPOD_DSCR	2	IMPOC_PROC	4
IMPOC_RCHZ	2	IMPOC_ALTA	4
IMPOC_DSCR	2	IMPOA_PROC	4
IMPOA_RCHZ	2	IMPOA_ALTA	4
IMPOA_DSCR	2	SUM_VNT	3
SUM_VND	1	SUM_ANL	1
SUM_CIPS	3	SUM_CAG	3
TOTS	3	TOTV	3
PARTICION	2	INC	2
SEL_AAG	2	SEL_PROD	3
MO_IMP	1	VNT_DIA	6
SEL_CICLO	1	VNT_DIAL	3
TEMPO	6	ESTA_BIEN	8
COMIS_AGNT	2	ARCH_VNT	5
VALOR_MO_0	6	NUM_TRANS	3
TIPO_PROD	2	CLAVE_AG	5
CONSECUTIVO	4	NO_MO	4
DS_DCI	1	FCH_VNT	4
DSTR	5	HORA_VNT	2
PRODUCTO	3	VALOR_VNT	7
MONEDA	3	COMIS_ISP	1
DENOMINAC	3	NUM_VND	3
FCH_VTA	1	NUM_AN1	3
DS_NSLOC	2	TOT_VNT	2
EXISTENCIA	7	TOT_CIPS	1
CNS_FIN	5	TOT_CAG	2
DS_AGNCHK	1	TOT_NUM	2
RECO	6	FECHA_PROC	3
TABLA_DN	1	NMR_REG	2
CNS_FINT	3	COMIS_IPS	6

PROGRAMA INIT*

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
TIP_DEN	3	COMIS_AG	3
DEN	2	TRANSAC	4
DESCRIP	2	"S"	4
DS_OUTRN	1	"V"	3
VALDR_MIN	1	SEL_EMIS	1
VALDR_MAX	1	RCHZ_IMP	1
TIPO_CAMBIO	3	DSCR_IMP	1
DS_CAMBIO	2	TEMP	2
ARCH_VNT	7		4
TPO_PROD	3	USUARIO	1
B	3	MONEDAT	2
10	6	6	7
DENOMINAT	2	40	2
28	1	FCH_SHIPT	2
11	5	SDF	2
FCH_SLOCT	2	32	4
49	1	FCH_SHIP	2
18	8	25	1
FCH_SLOC	3	36	1
4	1	CNS_TEMP	6
54	1	5	2
RESTA	5	70	1
9	1	CONT_INI	3
12	5	9990	1
DSTR_AP	1	37	1
2	3	EXIST_D	2
57	1	29	1
TRANSACCION	3	71	1
100	12	UL_VT	5
13	2	MO_TMP	1
15	7	TIFROD	1
16	11	VALOR_VTA	1
22	2	STATUS	1
27	1	"9999"	1
"/ /"	1	"REG"	1
SEL_TRANS	1	24	1
AGN_TMP	3	31	2
"RCHZ"	1	APUNTAOR	1
51	4	"9,999"	4

PROGRAMA INIT

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
DSTR_AP	1	67	4
ULT_VT	3	17	5
AGENTES	3	23	1
HR_VTA	1	3	4
CLV_OPER	1	20	5
CLAVE_OP	1	7	6
DS_TRNI	1	1	24
"NUMERO"	1	"IMPORTE"	2
"COMISION"	2	"REGISTROS"	1
"DOLARES"	1	"PESOS"	1
"IPS"	1	"AGENTE"	1
"PROC"	1	"L"	1
"D"	1	"!"	1
"-----"	2	"DSCR"	1
"TECLEA NOMBRE DEL ARCHIVO A PROC"			1
"CARGANDO ARCHIVO DE TRABAJO"			1
"ESPERA UN MOMENTO"			1
"999,999,999"			4
"999,999,999,999"			4
"9,999,999.99"			7
"OPRIME CUALQUIER TECLA PARA CONTINUAR"			1
"** EL ARCHIVO"			1
"TIENE INCONSISTENCIA CON EL ARCHIVO"			1
"NO EXISTE EN EL ARCHIVO NINGUN DOCUMENTO VALIDO"			1
"NO SE EFECTUO NINGUNA TRANSFERENCIA"			1
"ACTUALIZACION DE ARCHIVOS"			1
"ESPERA UN MOMENTO"			1
"ARCHIVO PROCESADO"			1
"NOMBRE DEL USUARIO"			1

PROGRAMA TRAN_MEX

OPERADORES	OCURENCIAS	OPERADORES	OCURENCIAS
=	170	+	52
-	1	*	3
RESTORE FROM	1	DOWHILE	3
ENDO	3	DO CASE	7
END CASE	7	RIGHT()	9
STR()	8	DO	13
DATC()	4	WITH	38
AT()	1	,	52
LEFTH()	1	@	10
REPLACE	5	<>	3
SKIP	4	&	2
GET	1	PICTURE	4
CASE	34	READ	1
SET ORDER	4	IF	7
ELSE	5	ENDIF	7
SEEK	1	SELECT	16
DELETE	1	USE	2
PACK	1	INDEX	1
SUBSTR()	1	ZAP	1
TYPE	1	APPEND BLANK	4
RETURN	1	SET FILTER TO	9
MARQUES	1	DTOC	6
CALC_DC	3	EOF()	4
COPY TO	1	GOTOP	6
OTHER WISE	3	CLEAR	5
GETGE	4	INCONS	4
LEFTH()	2	ABS()	1

PROGRAMA TRAN-MEX

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
" "	122	.T.	10
CLVE_STAT	30	ARCHT	5
EMISION	1	NUM_TRANS	7
PRODUCTO	4	VT	7
MONEDA	4	BVT	10
DENOMINAC	4	NVT	20
DSTR	2	REG	9
CANT_INI	4	SEL_PROD	1
TIPO	14	TRANS_MEX	6
CLAVE	13	MO	3
EXISTENCIA	12	FC_TRANS	1
NUM_STAT	32	D_C	8
FCH_SLOC	2	TIPO_PROD	11
NM_STAT	4	ENDIF	18
CNS_MO	10	FCH_LST	1
VALOR	4	NUM_TRANS	1
VALOR_MO_0	2	FCH_2ST	1
STATUS	4	FCH_2TR	2
STOLD	4	SDFK	1
TIPO_REG	3	REP_TRAN	1
CVE_MOV	3	10	5
OPB	4	8	1
IMPORTE	4	0	13
FECHA	2	1	31
FCH_TRANS	1	25	1
CICLO_VI	8	31	1
FCH_1TR	2	49	3
OK	3	12	1
BAJA	11	11	2
RANGO	16	2	4
4	1	7	5
100	2	10E11	2
10E2	4	00000000000	1
10E7	1	10E5	3
6	3	5	4
999999	1	999998	1
999997	1	9999	1
999996	1	9998	1
997	1	996	1
	5	"/ /"	9
"9,999"	3	"A"	3
"01"	4	"02"	2

PROGRAMA TRAN_MEX

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
"EM"	1	"ST_SOLD"	1
"ST_VDPR"	1	"BV"	1
"ST_STPR"	1	"O3"	5
"ST_LSPR"	1	"ST_VOID"	2
"ST_PRNT"	1	"S"	1
"TRAN_MEX 1"	1	"M"	7
"VB"	3	"ST_STLN"	1
LB	3	"L"	1
SB	3	"ST_LOST"	1
"DS"	1	"C"	1
"ST_VDSH"	2	"O1"	2
"ST_STSH"	2	"ST_CNCL"	1
"ST_LSSH"	2	"SV"	1
"ST_SHIP"	1	"VS"	1
"TRAN_MEX2"	1	"ST_SOLD"	3
"ST_VDSL"	2	"QS"	1
"ST_STSL"	2	COUNT FOR	1
"QL"	1	"ST_LSS1"	2
"QC"	1	"ST_SLOC"	1
"Q4"	1	"BS"	4
"M1"	1	"TRAN_MEX3"	1
"O5"	1	"LD"	1
"TRAN MEX4"	1	"SD"	2
"S1"	1	"VL"	2
"S+LNLOST CNL"	1	"SL"	2
"SLC"	1	"VD"	1
"O0"	1	"LD"	1
"LL"	1	"00000000"	1
"PROCESAMIENTO DE INFORMACION"			1
"TRANSMISION A MEXICO"			1
"!!!!!!!!!!!!!!!!!!!!!!"			1
"PROCESANDO EL REGISTRO"			1
"ESPERA UN MOMENTO"			1

PROGRAMA LECT_MEX.

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
Additive	1	Marques	1
Do	24	Msg_err	1
Use	1	Deseo	1
Select	11	Rep_Lect	1
With	37	Rep_Ds1	1
Type	1	Dscr_Mex	1
Clear	4	Rast2_po	1
EndDo	2	.Not.	5
Go Bottom	1	.Or.	2
Zap	2	EOF()	2
On Error	2	RECNO()	4
Do While	2	CTOD()	5
Say	8	SUBSTR()	7
Get	1	RIGHT()	5
Read	1	LEFT()	5
If	10	SUM()	1
Return	5	RECCOUNT()	1
Endif	10	DATE()	2
Append From	2	&	5
Restore from	1	=	39
Wait	3	@	15
Go Top	3	,	47
For	1	+	22
To	1	<>	4
Do Case	1	-	1
Picture	1	<	1
Case	17		
Replace	9		
Else	2		
Otherwise	1		
EndCase	1		
Delete	1		
Skip	1		
Pack1	1		
SDF	1		

PROGRAMA LECT_MEX

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
St_tabla	1	Ds_rchz10	1
Arch_mex	5	Ds_rchz11	1
No_dscr	1	Ds_rchz12	1
Sel_dscr	2	Ds_rchz13	1
Dsmx_tmp	2	Ds_rchz14	1
Sel_prod	1	Ds_rchz15	1
Lect_mex	7	Ds_rchz16	1
Esta_bien	8	Ds_rchz17	1
Ds_infd	1	Ds_rchz18	1
No_rec	3	Ds_rchz19	1
Aut_pago	1	Ds_rchz20	1
Tot_imp	2	0	3
Importe	2	1	11
Ultimo_req	2	2	16
Id_trailer	2	3	10
Clv_rchz	18	4	1
Fecha_proc	4	5	1
Fch_pago	11	7	4
Sm_importe	2	8	1
Num_rec	1	9	1
Lectura	2	10	4
Registro	2	11	1
Clv_stat	2	12	3
Es_bueno	2	13	2
Fch_rc	2	14	1
Mo	2	15	1
Fch_recep	2	16	1
Ciclo_vi	1	17	1
Dscr_mex	3	18	1
Status	4	19	1
Suc_prom	4	20	2
Suc_oper	4	22	2
Aut_pago	4	23	1
Fch_rcp_pg	2	25	1
Fch_regp	8	28	1
Flag_cnf1	1	31	1
Ds_ncv	1	32	1
Rep_dsl	1	49	1
Rep_lect	1	.f.	3
Ds_rchz1	1	.t.	3
Ds_rchz2	1	'9,999'	1
Ds_rchz3	1	'/'	3
Ds_rchz4	1	'/'	9
Ds_rchz5	1	'A'	1

LECT_MEX

OPERANDOS

OCURENCIAS

'P'	2
'fch_irc	1
'fch_tr'	1
'sold'	1
'mo->fch_trans'	1
'PROCESAMIENTO DE INFORMACION'	1
'RECEPCION DE DATOS DE MEXICO'	1
'TECLEA EL ARCHIVO A PROCESAR'	1
'EL ARCHIVO ESTA VACIO'	1
'PRESIONA CUALQUIER TECLA PARA CONTINUAR'	3
'ERROR EN LAS CIFRAS DE CONTROL'	1
'PROCESANDO EL REGISTRO : '	1
'ESPERA UN MOMENTO'	1
'FIN DEL PROCESO'	1

BOVE

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
Store to	23	EndIf	8
Do	19	R#str Mo	1
Do case	1	Select	3
Case	3	EndCase	1
Marques	1	Say	5
Get	12	Picture	10
Read	1	Calc_d_c	2
With	46	If	8
Msg_err	4	Deseo	5
Return	7	Zerol	3
Si_o_no	1	Replace	4
Append Blank	3	Set Filter to	1
Sort on	1	Use	2
Erase	1	Rename to	1
Date()	2	=	21
DtoC()	1	+	20
Space()	2	-	9
Val()	13	<>	4
@	17	,	47
()	2	:	3

BOVE

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
"0"	6	Fecha	3
"00"	2	60	2
0	11	Causa1	3
"0000000"	2	Causa2	3
" "	1	Codigo	4
ProductoP	5	999997	1
MonedaP	5	999998	1
DenominaP	5	999999	1
Cncsp	9	"_"	2
D_CP	3	Esta_Bien	11
ProductoS	7	.F.	2
MonedaS	7	Clv_Stat	3
DenominaS	7	D_CF	5
CncsS	8	Ds_DcI	2
D_CS	4	"Msg_Err"	1
Cnsc	1	Mo1	2
Cnsc_FinL	4	Mo2	2
Tiraje_L	5	Emission	5
Exist_L	3	Cnsc_Fin	9
Cnsc_FinA	5	Cant_ini	6
Tiraje_A	3	Ds_rnginv	1
Exist_A	3	"PRNT"	1
Cnsc_FinF	4	Existencia	5
Tiraje_F	5	Ds_Stinv	1
Exist_F	3	Fche	2
Fche_Emis	3	7	10
Fch_P	3	17	3
Fch_prnt	3	10	7
Ciclo_vi	1	14	1
Tipo_prod	1	16	1
Cnsc_Mo	1	45	2
Fch_1St	1	46	2
Clv_Stat	1	48	2
"EM"	1	50	2
Causa_St1	1	57	2
Causa_St2	1	58	2
Producto	2	3	1
Moneda	2	8	2
Denominac	2	1	5
Temporal	2	"9"	6
"Observaciones"	1	"99"	2
"Rango inicial"	1	"9999999"	2
"Rango Final"	1		
"Prevencciones en Boveda"	3	"Ordenes de Pago Anuladas"	1
"Ordenes de paco Perdidas"	-1	"Ordenes de Pago Anuladas"	1

DS1E

OPERADORES	OCURRENCIAS	OPERADORES	OCURRENCIAS
Store to	23	Rename To	1
If	9	Do Case	2
Case	8	With	34
Do	16	Marques	1
EndCase	2	Else	1
EndIf	9	Say	5
Get	12	Picture	10
Read	1	Calc_D_C	11
Index On To	1	Msg_Err	4
Deseo	5	Return	7
Rastr_Mo	1	Select	3
Zeroi	3	Si_o_No	1
Append Blank	3	Replace	6
Set Order to	1	Set Filter to	1
Sort on to	1	Use	2
Erase	1	Date()	2
=	23	DtoC()	1
+	19	Space()	1
<>	5	Val()	13
-	9	.And.	1
>	4	.Or.	3
<	1	.Not.	2
.	48	()	2
!	3	e	17

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
"0"	6	Fecha	3
"00"	2	60	2
0	12	Causa1	3
"0000000"	2	Causa2	3
" "	3	Bandera	1
ProductoP	5	Codigo	9
MonedaP	5	9996	2
DenominaP	5	9997	2
CnscP	9	9998	2

DSIE

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
D_CF	3	9999	2
ProductoS	7	Esta bien	11
MonedaS	7	"_"	2
DenominaS	7	Clv_Stat	5
CnscS	9	.F.	2
D_CS	4	D_CT	3
Cnsc	1	Ds_DcI	3
Cnsc FinL	4	"Msg_Err"	1
Tiraje_L	5	Mo1	2
Exist_L	3	Mo2	2
Cnsc_FinA	5	Dstr	5
Tiraje_A	3	Cnsc_Fin	8
Exist_A	3	Cant_Ini	6
Cnsc_FinF	3	Ds_RngInv	1
Tiraje_F	5	"Ship"	1
Exist_F	3	"Sloc"	1
Existencia	5	Ds_STInv	1
Clave_agt	3	Clave_ag	4
Fche	2	Fch_Ship	2
Fcho	3	Fch_Sloc	3
Ciclo_Vi	1	Tipo_prod	1
Cnsc_mo	1	Fch_1St	1
"Ds"	1	Causa_St1	1
Causa_St2	2	Moneda	2
Denominac	1	Temporal	2
Agnt_Ds	1	Producto	2
7	10	10	7
17	3	45	2
46	2	48	2
50	2	57	2
58	2	14	1
"9"	6	16	1
"99"	2	8	2
"9999999"	2	3	1
1	5	"Observaciones"	1
"Previsiones en Distribucion"	4	"Previsiones en Loc. de venta"	4
"Ordenes de pago Robadas"	2	"Rango Inicial"	1
"Ordenes de pago Anuladas"	2	"Rango Final"	1
"Ordenes de pago Perdidas"	2	"Ordenes de pago en poder de ips"	2

STP_FAY

OPERADORES

OCURRENCIAS

Parameters	1
Restore From	1
Additive	1
Do	9
Marques	1
With	17
Store to	6
Say	14
Get	7
Picture	8
Read	2
Rastr_Mo	1
If	6
Deseo	3
Return	3
Endif	6
Select	3
Msg_Err	2
Si_o_No	1
Replace	4
Set Filter to	1
Seek	2
Incons	1
Else	2
Append Blank	1
Use	2
Index	1
Date()	2
At()	1
EOF()	2
()	1
-	1
>	1
&	2
*	1
+	14
:	2
=	6
@	21
,	21

STP_PAV

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
Titulo	2	0	2
Stat_Ant	2	6	8
Nv_Stat	3	3	5
St_Tabla	1	45	1
ProductoT	5	46	1
MonedaT	5	48	1
DenominaT	5	50	1
Consecutivo	5	57	1
DCT	3	58	1
Esta_Bien	8	8	6
Clv_Stat	9	10	4
Mo	1	12	2
Ds_Stinv	1	23	2
Dstr->Clave_aa	2	42	3
Mo->Tipo_Camb	2	67	1
Mo->fch_vta	1	66	1
Mo->valor_vta	3	64	1
Pesos	2	14	1
Status	2	16	1
Ciclo_vi	1	17	1
St_Sold	1	"0"	1
Tipo_prod	2	"00"	1
Fch_2st	1	"0000000"	1
Cnsc_Mo	1	"9"	3
Fch_1st	1	"99"	2
Causa_St1	1	"9999999"	1
Causa_St2	1	.F.	1
Sel_Aa	1	" "	2
Agentes	2	"99.999.99"	1
Acum_vta	2	"999.999.99"	1
Mo->Comis_ipa	1	"999.999.999.99"	1
Ds_Aanf	1	"St "	1
"Msa_Err"	1	"+"	1
"Clave del agente"	1	"-"	1
"Tipo de cambio: \$"	1	"Observaciones"	1
"Importe en pesos: \$"	1	"Sto Pavi"	1
"Previsiones de pago"	1	"Fecha de venta:"	1
"Numero de la orden de pago: "	1		
"Importe en Dolares: \$"	1		

CONS

OPERADORES OCURENCIAS

Do while	2
Restore from	1
Additive	1
Do	26
Marques	1
With	26
Say	44
Get	2
Picture	8
Read	1
Getq	2
If	24
Msg_err	2
Exit	9
Endif	24
Clear	5
Imp_st	11
Do case	5
Case	19
Otherwise	4
Incons	9
EndCase	5
Else	4
Select	6
Set Filter to	5
Seek	4
Set Order to	1
EndDo	2
Deseo	1
Return	1
Left()	2
Dtoc()	4
Val()	4
Substr()	2
Str()	1
At()	2
EOF()	6
()	2
~	4
>	5
&	2
*	1
+	11
:	1
=	66
@	51
,	19

CONS

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
Ds_tabla	1	Fecha1	2
No_Mo_1	3	Flag_Cnfl	3
No_Mo_2	3	Fch_lrc	1
No_Mo	4	Mo->Sucur_prom	1
D_C	2	Mo->Sucur_Oper	1
Ren_St	5	Mo->Aut_Pago	1
Clv_Stat	21	Fch_pago	1
Transmis	5	Fch_Rco_Pg	1
Tipo_Prod	6	Esta_Bien	1
Producto	3	"0000"	1
Moneda	3	"0000000"	1
Denominac	3	"000000000000"	1
Consecutivo	6	"0"	1
Ds_monf	1	" "	10
Ds_mochk	1	"Consultas"	1
Fch_Emis	3	"9999"	1
Fch_Prnt	2	"999999"	1
Existencia	2	"999999999999"	1
Dstr->Fch_Ids	1	"-"	2
Fch_Ips	1	"9"	1
Ciclo_vi	2	"Ordo"	1
Tipo_prod	1	" / / "	4
Emission->cns_c_fin	1	"Prnt"	1
Fch_1st	1	">"	4
Fch_itr	1	"V DPR"	2
Causa_St1	1	"STPR"	2
Causa_St2	1	"LSPR"	2
Emission->fch_ips	2	"BIPS"	2
Dstr	2	"Cons 5"	4
Dstr->Clave_Aq	1	"EM"	1
Fch_Sloc	1	"SE"	1
Fch_Ship	1	"Cons 6"	2
Dstr->Cnec_Fin	1	"Observaciones"	3
Mo	1	"Rastr_Moi"	1
Mo->Valor_Vta	1	"Clave_aq"	1
Mo->Tipo_Cambio	1	"SHIP"	1
Mo->Valor_Mo_o	1	"SLOC"	1
Mo->Comis_ips	1	"VDSH"	1
Mo->Comis_Aq	1	"STSH"	1
Mo->Clv_Oper	1	"LSSH"	1
Status	4	"DIPS"	1
Fch_Vta	2	"VDSL"	1
Fch_trans	1	"STSL"	1
Cont0	5	"LSSI"	1
Cont1	4	"LIPS"	1

CONS

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
"DS"	1	8	3
"SD"	1	29	10
"Rastr_mo"	1	54	1
"Tipo de Cambio"	1	47	2
"Importe en Dolares"	1	15	1
"Importe en pesos"	1	5	10
"Comision los"	1	39	2
"Comision Agente"	1	45	2
"Clave operador"	1	1	22
"9.999.999.99"	1	50	2
"9.999.999.999.99"	1	51	2
"99.999.99"	1	6	1
"V "	1	7	9
"VOID"	1	17	1
"SOLD"	1	18	4
"I "	1	19	7
"VIFS"	1	10	5
"- -v -p -i "	1	57	2
"Rastr_Mo4"	1	70	1
" StIn Lost Cncl"	1	20	3
" SLC"	1	9	2
"Fch _"	2	11	2
"ST"	1	12	2
"Tr"	1	14	2
"A"	1	28	1
"R"	1	79	4
"Confirmado el "	1	999990	2
"Rechazado el"	1	999999	2
"P"	1	999998	2
"Sucursal Promotora"	1	999997	2
"Sucursal Operadora"	1	999996	4
"Autorizacion de Pago"	1	9999	2
"PAID"	1	9998	2
"<"	1	9997	2
16	1	9996	4
2	1	9990	1
4	1	30	1
43	1	55	1
21	1	56	1
"Stop Payment por confirmar"	1	22	1
"Ordenes de pago individuales"	1		
"Numero de la orden de pago"	1		
"Status Fecha Comunicacion"	1		
"-----"	1		

CONSTAT

OPERADORES OCURENCIAS

Restore from	1
Additive	1
Do While	23
Clear	1
Do	77
Marques	48
With	83
Say	249
Get	11
Picture	11
Read	11
EndDo	23
Do Case	11
Case	47
Consultas	36
EndCase	11
Return	1
=	123
<>	11
,	323
@	260
AT()	11

CONSTAT

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
Consul	1	"Alor"	2
Opcion	148	"Boveda"	2
" "	68	"3. en poder de ips"	2
Cn	1	"Prnt"	1
0	12	"Vdpr"	1
"0"	11	"Lsor"	1
"0. Salida"	11	"Stpr"	1
"1. Imoresion"	1	"Esor"	1
"2. Boveda"	1	"En poder de ips"	1
"3. Distribucion"	1	"Bips"	1
"4. Localidad de vta"	1	"En boveda"	1
"5. Vendidos"	1	"Alor"	1
"6. Anulados"	1	"Distribucion"	1
"7. Paados"	1	"Agente"	1
"Teclaa tu opcion"	11	"Fecha"	10
"01234567"	1	"Shio"	1
"9"	11	"Vdsh"	1
"1"	11	"Lssh"	1
"2"	11	"Stsh"	1
"3"	11	"Essh"	2
"4"	10	"Dios"	1
"5"	1	"En ditribucion"	1
"6"	1	"Alsh"	4
"7"	1	"Localidad de venta"	1
"Consultas"	7	"Sloc"	4
"Impresion"	1	" "	2
"0123"	1	"Vdsl"	2
"Cantidad"	26	"lssl"	2
"Status"	34	"stsl"	2
"OrdP"	7	"Essl"	4
"1. Anuladas"	4	"Lips"	1
"2. Perdidas"	5	"Alsl"	2
"3. Robadas"	5	"Orden de pago"	3
"4. Cualquer status"	9	"Fecha de venta"	1
"01234"	9	"Fecha transmis"	1
"Anuladas"	4	"Sold"	1
"VDOR"	1	"vnd"	1
"="	23	"1. Canceladas"	1
"Perdidas"	5	"Canceladas"	1
"LSOR"	1	"Lvnd"	1
"Robadas"	5	"Svnd"	1
"Stor"	1	"Lvnd"	1
"Esor"	1	"Evnd"	1
"Ordenes de Paoo"	6	"Ivnd"	1
"En impresion"	1	"Avnd"	1

CONSTAT

OPERANDOS	OCURRENCIAS	OPERANDOS	OCURRENCIAS
"Vendidas"	1	6	174
"Void"	2	8	24
"Fecha de Pago"	2	10	11
"Paid"	2	12	11
"Ordenes de pago anuladas"	1	14	11
"Ordenes de pago pagadas"	1	16	7
"Ordenes de pago vendidas"	7	18	11
"En Localidad de Venta"	1	20	1
"Ordenes de Pago en Localidad de venta "	7	24	11
"Ordenes de Pago en Distribucion"	7	30	32
"Ordenes de Pago en Boveda"	7	35	39
"Con prevencion de Pago"	10	46	23
"Ordenes de pago en impresion"	6	25	14
"Menu de consultas"	1	2	14
"Sistema Ordenes de pago"	1	68	13
"Con status normal"	5	54	5
"Orden de pago inicial"	26	21	10
"Orden de pago final"	26	55	13
64	24	32	18
73	19	56	24
4	9		
62	9		

CONSULTAS

OPERADORES

OCURENCIAS

Parameters	3
If	15
Select	3
Set Filter to	8
Else	12
Endif	15
Go Top	3
Do While	3
Do	13
Oprime1	3
Clear	2
ZeroI	2
With	10
Calc_D_C	5
Say	14
Picture	2
Stat_Mo	2
Skip	3
EndDo	3
Substr()	1
At()	1
Left()	14
Val()	8
EOF()	3
DtoC()	4
t	1
=	33
+	43
>	6
<>	10
@	19
-	6
,	45
*	1
&	11

CONSULTAS

OPERANDOS	OCURENCIAS	OPERANDOS	OCURENCIAS
Clave1	9	Fch_vta	1
Clave2	8	Fch_pago	1
Clave3	5	Fch_trans	1
Signo	5	8	7
Salte	7	0	4
"Cl_"	1	1	25
Renqlon	29	999995	1
Tecia	2	7	2
" "	4	12	1
DC	11	35	1
ST	5	57	2
"Drdp"	2	69	1
"Prnt"	1	9995	1
Sel_Emis	1	6	1
Existencia	14	28	1
Fch_Prnt	2	48	2
" / / "	4	65	1
Cnca_ini	12	72	1
Cnca_fin	10	5	2
Cant_ini	2	21	1
Producto	8	67	2
Moneda	8	4	2
Denominac	8	36	1
"_"	5		
"999,999"	1		
"Ship"	2		
"Sloc"	1		
Sel_Dstr	1		
Fch_Sloc	2		
"9,999"	1		
Clave_ag	2		
Fch_Ship	1		
Sel_Mo	1		
"E"	2		
"A"	1		
Status	9		
"Y"	2		
"P"	3		
"I"	1		
Tipo_prod	2		
Cnsc_Mo	2		
" S C L P V I"	1		
"Sold Stln Cncl Lost Paid Void Vips"			1

MARQUES

OPERADORES	OCURENCIAS	OPERANDOS	OCURENCIAS
Parameters	1	Cabezal	3
If	2	Cabeza2	3
Clear	1	Flag	3
EndIf	2	"0"	2
Say	8	0	4
Picture	1	1	6
Else	1	2	4
To	1	3	1
Double	1	4	2
Return	1	8	1
/	2	22	1
-	1	62	1
=	3	69	1
.	12	40	2
@	9	"1"	1
Time()	1	79	1
Date()	1	"Hora: "	1
Len()	1	"99:99"	1
		"Fecha: "	1
		"-----"	
		"International	
		Payment Systems"	1

RUTAS PROGRAMA ORDP

NODO	RUTA
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,2,3,4,5
6	1,2,3,4,5,6
7	1,2,3,4,5,6,7
8	1,2,3,4,5,6,8
9	1,2,3,4,5,6,8,9
10	1,2,3,4,5,6,8,9,10
11	1,2,3,4,5,6,8,9,11
12	1,2,3,4,5,6,8,9,11,12
13	1,2,3,4,5,6,8,9,11,12,13
14	1,2,3,4,5,6,8,9,11,12,14
15	1,2,3,4,5,6,8,9,11,12,13,15 1,2,3,4,5,6,8,9,11,12,14,15
16	1,2,3,4,5,6,8,9,11,12,13,15,16 1,2,3,4,5,6,8,9,11,12,14,15,16
17	1,2,3,4,5,6,8,9,11,12,13,15,16,17 1,2,3,4,5,6,8,9,11,12,14,15,16,17
18	1,2,3,4,5,6,8,9,11,12,13,15,16,17,18 1,2,3,4,5,6,8,9,11,12,14,15,16,17,18
19	1,2,3,4,5,6,8,9,11,12,13,15,16,17,18,19 1,2,3,4,5,6,8,9,11,12,14,15,16,17,18,19
20	1,2,3,4,5,6,8,9,11,12,13,15,16,17,18,19,20 1,2,3,4,5,6,8,9,11,12,14,15,16,17,18,19,20

NODO	RUOTA
21	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21
22	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22
22	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22
23	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
26	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
27	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
28	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28
29	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29
30	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30

NODO	ruta
31	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31
32	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 32
33	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 32, 33 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 33
34	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 33, 34
35	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 33, 34, 35
36	1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36

NODO

RUTA

- 37 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37
1, 2, 3, 4, 5, 6, 7, 37
1, 2, 3, 4, 5, 6, 8, 9, 10, 37
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 37
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 37
- 36 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38
1, 2, 3, 4, 5, 6, 7, 37, 38
1, 2, 3, 4, 5, 6, 8, 9, 10, 37, 38
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 15, 16, 17, 18, 37, 38
1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 37, 38

RUTAS PROGRAMA PRNT

NODO	RUTA
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,2,3,4,5
6	1,2,3,4,5,6
7	1,2,3,4,5,6,7
8	1,2,3,4,5,6,7,8
9	1,2,3,4,5,6,7,9
10	1,2,3,4,5,6,7,9,10
11	1,2,3,4,5,6,7,9,11
12	1,2,3,4,5,6,7,9,11,12
13	1,2,3,4,5,6,7,9,11,12,13
14	1,2,3,4,5,6,7,9,11,12,13,14 1,2,3,4,5,6,7,9,11,12,13,25,14
15	1,2,3,4,5,6,7,9,11,12,13,14,15 1,2,3,4,5,6,7,9,11,12,13,25,14,15
16	1,2,3,4,5,6,7,9,11,12,13,14,15,16 1,2,3,4,5,6,7,9,11,12,13,25,14,15,16
17	1,2,3,4,5,6,7,9,11,12,13,14,15,16,17 1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17
18	1,2,3,4,5,6,7,9,11,12,13,14,15,16,17,18 1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17,18
19	1,2,3,4,5,6,7,9,11,12,13,14,15,16,17,18,19 1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17,18,19

NODO RUTA

- 20 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20
- 21 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21
- 22 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22
- 23 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
- 24 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
- 25 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25
- 26 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26
- 27 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26, 27
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26, 27
- 28 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26, 27, 28
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27, 28
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26, 27, 28

- 29 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26, 27, 28, 29
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 26, 27, 28, 29
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27, 28, 29
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
24, 26, 27, 28, 29
- 30 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26, 27, 28, 29, 30
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 26, 27, 28, 29, 30
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27, 28, 29, 30
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
24, 26, 27, 28, 29, 30
- 31 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26, 27, 28, 29, 30, 31
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 26, 27, 28, 29, 30, 31
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27, 28, 29, 30, 31
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
24, 26, 27, 28, 29, 30, 31
- 32 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
24, 26, 27, 28, 29, 30, 31, 32
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 26, 27, 28, 29, 30, 31, 32
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27, 28, 29, 30, 31, 32
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
24, 26, 27, 28, 29, 30, 31, 32
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 26, 27, 28, 29, 30, 32
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24
26, 27, 28, 29, 30, 32
1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 25, 14, 15, 16, 17, 18, 19, 20, 21, 22,
24, 26, 27, 28, 29, 30, 32

NODO RUTA

- 33 1,2,3,4,5,6,7,9,11,12,13,14,15,16,17,18,19,20,21,22,23
24,26,27,28,29,30,31,32,33
1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17,18,19,20,21,22,
23,24,26,27,28,29,30,31,32,33
1,2,3,4,5,6,7,9,11,12,13,14,15,16,17,18,19,20,21,22,24
26,27,28,29,30,31,32,33
1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17,18,19,20,21,22,
24,26,27,28,29,30,31,32,33
1,2,3,4,5,6,7,9,11,12,13,14,15,16,17,18,19,20,21,22,23
24,26,27,28,29,30,32,33
1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17,18,19,20,21,22,
23,24,26,27,28,29,30,32,33
1,2,3,4,5,6,7,9,11,12,13,14,15,16,17,18,19,20,21,22,24
26,27,28,29,30,32,33
1,2,3,4,5,6,7,9,11,12,13,25,14,15,16,17,18,19,20,21,22,
24,26,27,28,29,30,32,33
1,2,3,4,5,6,7,8,33

RUTAS PROGRAMA SLOC

NODO	ruta
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,2,3,4,5
6	1,2,3,4,5,6
7	1,2,3,4,5,6,7
8	1,2,3,4,5,6,7,8
9	1,2,3,4,5,6,7,8,9
10	1,2,3,4,5,6,7,8,9,10
11	1,2,3,4,5,6,7,8,9,10,11
12	1,2,3,4,5,6,7,8,9,10,11,12
13	1,2,3,4,5,6,7,8,9,10,11,13
14	1,2,3,4,5,6,7,8,9,10,11,12,14
15	1,2,3,4,5,6,7,8,9,10,11,13,15
16	1,2,3,4,5,6,7,8,9,10,11,13,15,16
17	1,2,3,4,5,6,7,8,9,10,11,12,13,,17
18	1,2,3,4,5,6,7,8,9,10,11,13,17,18
19	1,2,3,4,5,6,7,8,9,10,11,12,13,17,18,19
20	1,2,3,4,5,6,8,9,11,12,13,17,18,19,20
21	1,2,3,4,5,6,7,9,10,11,12,13,17,18,19,20,21
22	1,2,3,4,5,6,7,9,10,11,12,13,17,18,19,20,21,22
23	1,2,3,4,5,6,7,9,11,12,13,17,18,19,20,21,22,23
24	1,2,3,4,5,6,7,9,10,11,12,13,17,18,19,20,21,22,23,24

NDDO	RUTA
25	1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 17, 18, 19, 20, 21, 22, 23, 24, 25
26	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
27	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 25, 14, 15, 16, 17, 19, 20, 27
28	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 27, 28
28	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 29 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 15, 16, 29 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 18, 19, 20, 27, 28, 29

RUTAS PROGRAMA SHIP

NODO	RU TA
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,2,3,4,5
6	1,2,3,4,5,6
7	1,2,3,4,5,6,7
8	1,2,3,4,5,6,7,8
9	1,2,3,4,5,6,7,8,9 1,2,3,4,5,6,7,8,6,9
10	1,2,3,4,5,6,7,8,9,10 1,2,3,4,5,6,7,8,6,9,10
11	1,2,3,4,5,6,7,8,9,10,11 1,2,3,4,5,6,7,8,6,9,10,11
12	1,2,3,4,5,6,7,8,9,10,11,12 1,2,3,4,5,6,7,8,6,9,10,11,12
13	1,2,3,4,5,6,7,8,9,10,11,12,13 1,2,3,4,5,6,7,8,6,9,10,11,12,13
14	1,2,3,4,5,6,7,8,9,10,14 1,2,3,4,5,6,7,8,6,9,10,14
15	1,2,3,4,5,6,7,8,9,10,14,15 1,2,3,4,5,6,7,8,6,9,10,14,15
16	1,2,3,4,5,6,7,8,9,10,14,15,16 1,2,3,4,5,6,7,8,6,9,10,14,15,16
17	1,2,3,4,5,6,7,8,9,10,14,15,16,17 1,2,3,4,5,6,7,8,9,10,14,15,16,17
18	1,2,3,4,5,6,7,8,9,10,14,18 1,2,3,4,5,6,7,8,9,10,14,18

NODO	ruta
19	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 19 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 19
20	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 19, 20 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 19, 20
21	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21
22	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22
23	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23
24	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24
25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25
26	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26
27	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 27 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 27
28	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28
29	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29
30	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30
31	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31

NODO	RUTA
32	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32
33	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33
34	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33, 34 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33, 34 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33, 34 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33, 34
35	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33, 34, 35 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33, 34, 35 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33, 34, 35 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33, 34, 35
36	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33, 34, 35, 36 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31 32, 33, 34, 35, 36 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33, 34, 35, 36 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31 32, 33, 34, 35, 36

NODO

RUTA

51

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31
 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49
 50, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31
 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49
 50, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31
 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50
 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31
 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50
 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31
 32, 33, 34, 35, 39, 40, 41, 50, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31
 32, 33, 34, 35, 39, 40, 41, 50, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31
 32, 33, 34, 35, 40, 41, 50, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 28, 29, 30, 28, 31
 32, 33, 34, 35, 40, 41, 50, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 51
 1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 11, 12, 13, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 16, 17, 51
 1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 15, 16, 17, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 19, 20, 51
 1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 14, 18, 19, 20, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 27, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 27, 51
 1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 27, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31
 32, 33, 34, 35, 36, 37, 38, 51
 1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31
 32, 33, 34, 35, 36, 37, 38, 51
 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 28, 29, 30, 28, 31
 32, 33, 34, 35, 36, 37, 38, 51
 1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 14, 18, 21, 22, 23, 24, 25, 26, 28, 29, 30, 28, 3
 32, 33, 34, 35, 36, 37, 38, 51

RUTAS PROGRAMA REP DSTR

NODO	RUTA
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,2,3,4,5
6	1,2,3,4,5,6
7	1,2,3,4,5,6,7
8	1,2,3,4,5,6,7,8
9	1,2,3,4,5,6,7,8,9
10	1,2,3,4,5,6,7,8,9,10
11	1,2,3,4,5,6,7,8,9,10,11
12	1,2,3,4,5,6,7,8,9,10,11,12 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 23,24,25,26,27,28,11,12 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 27,28,11,12
13	1,2,3,4,5,6,7,8,9,10,11,12,13 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 23,24,25,26,27,28,11,12,13 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 27,28,11,12,13
14	1,2,3,4,5,6,7,8,9,10,11,12,13,14 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 23,24,25,26,27,28,11,12,13,14 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 27,28,11,12,13,14
15	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 23,24,25,26,27,28,11,12,13,14,15 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, 27,28,11,12,13,14,15

NODD

RUTA

- 16 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
23,24,25,26,27,28,11,12,13,14,15,16
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16
- 17 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,,17
23,24,25,26,27,28,11,12,13,14,15,16,17
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17
- 18 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,,17
23,24,25,26,27,28,11,12,13,14,15,16,17,18
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17,18
- 19 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,,17
23,24,25,26,27,28,11,12,13,14,15,16,17,18,19
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17,18,19
- 20 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,,17
23,24,25,26,27,28,11,12,13,14,15,16,17,18,19,20
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17,18,19,20
- 21 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,,17
23,24,25,26,27,28,11,12,13,14,15,16,17,18,19,20,21
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17,18,19,20,21
- 22 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,,17
23,24,25,26,27,28,11,12,13,14,15,16,17,18,19,20,21,22
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17,18,19,20,21,22
- 23 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22
23
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,17
23,24,25,26,27,28,11,12,13,14,15,16,17,18,19,20,21,22,23
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
27,28,11,12,13,14,15,16,17,18,19,20,21,22,23

NDDO

RUTA

- 24 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 24
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17
23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
- 25 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 24, 25
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17
23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
- 26 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 24, 25, 26
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17
23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
- 27 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 24, 25, 26, 27
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17
23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26, 27
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
27
- 28 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 24, 25, 26, 27, 28
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17
23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
25, 26, 27, 28
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
28
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
27, 28

NODO	RUTA
29	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 23, 24, 25, 26, 27, 28, 29 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17 23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 27, 28, 29
30	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 23, 24, 25, 26, 27, 28, 29, 30 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17 23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 27, 28, 29, 30
31	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 23, 24, 25, 26, 27, 28, 29, 30, 31 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17 23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 27, 28, 29, 30, 31
32	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 17 23, 24, 25, 26, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 25, 26, 27, 28, 29, 30, 31, 32 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27, 28, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 27, 28, 29, 30, 31, 32

RUTAS PROGRAMA VIAS

NODO	RUTA
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,2,3,4,5
6	1,2,3,4,5,6
7	1,2,3,4,5,6,7
8	1,2,3,4,5,6,7,8
9	1,2,3,4,5,6,7,8,9
10	1,2,3,4,5,6,7,8,9,10
11	1,2,3,4,5,6,7,8,9,10,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22 23,24,26,27,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22 23,25,26,27,,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,14,16,17,18,19,20,21, 22,23,24,26,27,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,14,16,17,18,19,20,21, 22,23,25,26,27,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,16,18,19,20,21, 22,23,24,26,27,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,16,18,19,20,21, 22,23,25,26,27,,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,14,16,17,16,18,19,20, 21,22,23,24,26,27,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,14,16,17,16,18,19,20, 21,22,23,25,26,27,28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,27,28,29, 30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,14,16,17,16,18,19,20, 28,29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,16,18,19,20,28, 29,30,31,11 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,14,16,17,16,18,19,20, 21,22,23,24,26,27,28,29,30,31,11

NDDO

ruta

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 28,
29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 14, 16, 17, 18, 19, 20,
21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 24, 26, 27, 28, 29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
23, 25, 26, 27, 28, 29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 14, 16, 17, 18, 19, 20, 21,
22, 23, 24, 26, 27, 28, 29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 14, 16, 17, 18, 19, 20, 21,
22, 23, 25, 26, 27, 28, 29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 14, 17, 16, 18, 19, 20, 21,
22, 23, 24, 26, 27, 28, 29, 30, 31, 30, 11, 12, 13, 14, 15, 16, 17

RESULTADOS COMPARATIVOS DE METRICAS

TAMANO

Programa	Ss	n ₁	n ₂	N ₁	N ₂	N	Ssest	n	V	C _{fp}	C _{fsist}
MoneyD	314	56	141	624	614	1238	310	197	9436.77	20	4
OrdP	79	46	73	225	194	419	105	119	2878.92	2	10
Prnt	56	43	60	162	142	304	76	103	2032.69	4	10
Ship	91	49	69	249	160	409	102	118	2815	9	5
Rep_Dstr	44	31	40	106	97	203	51	71	1248	2	3
Sloc	53	33	36	134	86	220	55	69	1343.87	2	8
Vtas	115	40	104	293	273	556	141	68	3445.90	8	2
Init	311	69	196	655	603	1258	315	265	10126.70	8	3
Tran_mex	289	56	145	293	273	1199	298	201	9173.60	8	3
Lect_mex	156	61	105	329	254	583	146	166	4299.71	5	8
Bove	125	44	89	328	282	610	153	133	4300.50	7	4
Dste	145	48	96	339	307	646	162	144	4625.26	7	4
Stp_pay	76	41	76	183	152	335	84	117	2301.45	6	3
Cons	237	52	179	518	413	931	233	231	7308.35	5	6
Constat	540	22	137	1373	1273	2646	662	159	19342.26	2	1
Consultas	119	38	68	348	268	616	154	106	4139.52	4	6
Marques	18	19	21	49	41	90	23	40	478.80	-	3

ESTRUCTURA DE DATOS

ESTRUCTURA LOGICA

Programa	VARS	DE	v(G)	R	Np	NL
MoneyD	141	72	64	-	-	7
OrdP	73	6	6	80	8	3
Prnt	60	4	5	48	5	3
Ship	69	10	9	156	20	2
Rep_Dstr	40	3	4	100	5	4
Sloc	36	4	5	32	4	2
Vtas	104	7	8	366	22	4
Init	196	34	23	-	-	11
Tran_mex	145	44	41	3685960	121970	7
Lect_mex	105	30	32	-	-	-
Bove	89	14	11	-	-	-
Dste	96	20	15	-	-	-
Stp_pay	76	6	8	-	-	-
Cons	179	46	50	-	-	-
Constat	137	70	71	-	-	-
Consultas	68	18	21	-	-	-
Marques	21	2	5	-	-	-

METRICAS COMPUESTAS

Programa	N	V	V*	L	D	E	T(s)
MoneyD	1331.88	9439.77	77.28	0.01	121.95	1150253.10	63930 0.64
OrdP	705.90	2889.72	47.26	0.02	81.72	176586.28	9010 0.22
Print	587.74	2032.69	39.94	0.02	50.89	103430.00	5746 0.76
Snip	596.60	2815.00	44.54	0.02	56.81	159924.63	8885 0.81
Rep_Distr	366.45	1249.39	33.21	0.03	37.59	46923.85	2607 0.88
Sloc	352.58	1343.87	34.00	0.03	39.41	52971.00	2945 0.86
Vtas	909.72	3445.50	65.65	0.02	52.50	181039.00	10057 1.24
Init	1913.97	10126.70	95.00	0.01	106.14	1074907.30	59717 0.90
Iran_mer	1336.29	9173.00	73.95	0.00	123.97	1137289.00	63183 0.60
Lect_mer	1066.77	4299.71	58.04	0.01	74.07	318497.03	17670 0.79
Bave	816.52	4303.72	61.54	0.01	69.93	300959.44	16702 0.88
Dst	900.23	4631.77	60.21	0.01	76.92	356390.00	19779 0.78
Stp_pay	694.50	2301.57	55.92	0.02	41.15	97183.95	5347 1.35
Cons	1636.00	7309.97	121.34	0.02	60.24	440359.63	24430 2.01
Constat	1070.53	19349.88	187.69	0.01	103.09	1994832.99	110507 1.82
Consultas	613.36	4144.39	85.12	0.01	75.18	311608.20	17288 0.73
Marques	172.94	478.97	25.81	0.05	18.54	8866.27	494 1.39

RESULTADOS DE LA CERTIFICACION DE CALIDAD DEL SISTEMA EN PRODUCCION.

Al analizar un sistema que se encuentra actualmente en produccion, se evaluó la calidad de este, mediante la aplicacion de las métricas mencionadas en el capítulo 2, obteniéndose los siguientes resultados :

Al aplicar las métricas de tamaño, se obtuvo que el estimador S_s obtenido es mayor a la métrica de líneas de código S_s real para cada uno de los programas. La relación entre el estimador y la métrica de líneas de código es inversa al tamaño de cada programa, es decir, mientras mayor número de líneas tiene un programa, es menor la diferencia entre estas y el estimador obtenido. Esta diferencia puede ser debida a las ambigüedades en el conteo de operandos y operadores o a la falta de consistencia en el contador de líneas de código, ya que hay líneas más difíciles de codificar que otras.

Sin embargo, a pesar de las ambigüedades que pueden producirse en el conteo de operandos y operadores, se considera conveniente darle mayor importancia al valor obtenido en el estimador S_s , aún cuando sea más fácil obtener el número de líneas de programa.

Basándonos en que el tamaño de los módulos debe ser entre 50 y 200 líneas de código (LOC), encontramos que la mayoría de los módulos cumplen esta condición, exceptuando un mínimo de programas que se considera que requieren una mayor estructuración. Sin embargo, podemos decir que el sistema se encuentra modularizado.

Al analizar la estructura lógica de cada programa, basándose en el conteo de decisiones y la complejidad ciclomática, los módulos desarrollados más complejos, son el programa principal del sistema (Programa Moneyo), y el programa del modulo de consultas (Programa Constat). Considerando además, que según McCabe, un $v(G)$ de complejidad ciclomática igual a 10 es lo razonable, encontramos que un 50% de los modulos sobrepasa este rango de complejidad aceptada.

La representación de las instrucciones complejas varían en los diagramas de flujo, pudiéndose representar con un solo símbolo de decisión o descomponerse en varios símbolos de decisiones, aumentando la contabilización con sus correspondientes líneas de dirección. Debido a esto, no se cumple la relación de igualdad entre $v(G)$ y $DE+1$ cuando la condición compleja utiliza solo un símbolo de decisión, ya que las condiciones compuestas no generan una línea de dirección en el diagrama de flujo, pero si afectan el contador de decisiones.

El análisis realizado para la estructura lógica del sistema, fue considerando cada módulo por separado, ya que la certificación final de calidad se evaluará basándose en los resultados por módulos.

Otra forma de medir la complejidad de un programa, es mediante la métrica de extensión y número mínimo de trayectorias. Esta métrica no es práctica aplicarla en programas muy largos y/o que contengan ciclos dentro de ciclos, debido a que las trayectorias empiezan a aumentar de forma desproporcionada, llegando a ser muy difícil de cuantificar. Un ejemplo de este problema fue con el programa principal Moneyo, el cual debido a su extensión, produce demasiadas trayectorias, siendo imposible terminar con la cuantificación.

El nivel de anidación también es un factor importante en la medición de la complejidad de un programa, ya que mientras mayor sea este, el seguimiento de la secuencia es cada vez más complicado.

En base al análisis realizado, se propone como un nivel de anidación adecuado un valor de 5, ya que con un número mayor la facilidad para perder la continuidad de las condiciones efectuadas anteriormente aumenta.

La primera hipótesis de la Ciencia de Software, la ecuación de longitud, falló en precisar el número de tokens usados en los programas prediciendo un número mayor del que realmente se utilizó.

Pudimos observar que el valor obtenido del volumen potencial, es inferior al volumen real. Y a pesar de que se toma en cuenta que el volumen real puede ser reducido, se considera que el volumen potencial es un valor muy utópico, ya que ninguno de los programas, por mayor simplificación y modularización que se le aplique, alcanzaría estos valores.

El nivel de programación, fue en general deficiente, reflejándose esto, en la cantidad de espacio requerido para almacenar el código y en el alto grado de dificultad de los programas. Incrementándose en consecuencia el tiempo y esfuerzo en programación.

El nivel de lenguaje, se considera que llega a ser una métrica útil para referencias posteriores o si en lugar de analizar el código, la evaluación se realiza con el pseudocódigo.

CONCLUSIONES.

En base a los resultados obtenidos y al proceso de desarrollo de la certificación, las conclusiones generales para la certificación de un sistema son :

1. El análisis de los datos sólo puede ser útil al comparar "manzanas con manzanas", esto es, se debe tener cuidado en no combinar datos conteniendo métricas de software con diferentes definiciones. Comparando dos diferentes conjuntos de datos con items llamados "líneas de código", se debe tener cuidado de que los items realmente midan la misma cosa. Es obvio que hay diferentes definiciones de líneas de código. Si en un conjunto de líneas de código excluyen comentarios y declaraciones, mientras que en el otro las incluye, llega a ser difícil obtener conclusiones de las combinaciones del conjunto de datos sin introducir serias inclinaciones.

Por esta razón, los analistas deben tener cuidado cuando analizan datos de diferentes organizaciones o datos obtenidos bajo diferentes reglas en la misma organización.

2. Muchos modelos de desarrollo de software requieren calibración, que es, determinar coeficientes y constantes de datos históricos obtenidos en un ambiente específico. Muchos modelos de software no pueden ser transportables directamente de un ambiente a otro sin re-calibrar. El precio pagado por la violación de este estándar son predicciones erróneas.

3. El resultado de los modelos de software pueden auxiliar, pero no reemplazar el proceso de toma de decisiones de gerentes con experiencia. Se consideran las métricas y modelos discutidas como unas herramientas valiosas para los que toman decisiones, pero nunca pueden reemplazar a los gerentes, desde que en general, las métricas normalmente direccionan sólo una pequeña parte del proceso de desarrollo del producto. Además, mientras estén razonablemente validadas, las métricas enfocan sólo una pequeña parte de las variables envueltas y por su naturaleza producen medidas de las características del sistema promedio. Debido a las variaciones extremas en el desempeño humano, como medidas promedio, pueden dar un figura distorsionada cuando se aplican a un sistema particular. Similarmente, alguna métricas y modelos pueden producir pobres resultados en ambientes substancialmente diferentes. En particular, los gerentes deben ser libres de discernir las implicaciones de los modelos de software si ellos creen que el proyecto debe ir diferente en alguna manera que el modelo este dando una lectura errónea. Su experiencia e intuición debe ser su guía. Sin embargo, la información de métricas y modelos (cuando coinciden con la intuición) puede ayudar a proveer gran precisión para estimadores y decisiones.

Las métricas y modelos de software son propuestas para ser usadas en el manejo de los productos, no para evaluar al equipo técnico. Sin embargo, si estas métricas son exitosas, se pueden utilizar como medidas de productividad a manera de predecir costos y como una manera de asegurar el relativo desempeño de cada programador.

Una limitación en el uso de las métricas es la dificultad y costo de computar estas métricas. El costo de obtener métricas puede reducir los beneficios potenciales. Las métricas más útiles son aquellas que pueden ser obtenidas automáticamente y con muy poco esfuerzo extra del programador. Están disponibles diversas herramientas de software para diversos lenguajes que pueden computar tamaño, estructura de control y otras métricas complejas para programas de computadora. Algunas instalaciones automáticamente computan métricas en todo el código producido, así pueden tomarse las acciones apropiadas cuando estas métricas exceden ciertos límites preescrito.

Algo de suma importancia es el problema en los modelos predictivos de confiar en estimar ciertas variables, las cuales seguidas no son conocidas exactamente. Por ejemplo, muchos modelos utilizan el recurso del tamaño del programa como un parámetro de entrada. Todavía la habilidad de estimar el tamaño tempranamente en el proceso de desarrollo es muy pobre. Desafortunadamente el modelo de predicción no puede ser más exacto que la predicción del tamaño.

BIBLIOGRAFIA

Handbook of Software Quality Assurance
G. Gordon Schulmeyer and James I. McManus
New York, New York, 1987

Software Engineering Metrics and Models
S.D. Conte, H.E. Dunsmore, V.Y. Shen
The Benjamin/Cummings Publishing Company Inc
1986

Ingeniería de Software
Richard Fairley
McGraw Hill
México, 1987

The Art of Software Testing
Myers, G.J.
New York, New York
John Wiley & Sons

La Computación en México
Diagnóstico, prospectiva y estrategias de desarrollo
Conacyt
Fundación Arturo Rosenblueth para el Avance de la Ciencia

ANEXO 1

Estrategia de conteo para el Lenguaje Pascal

ANEXO 1

ESTRATEGIA DE CONTEO PARA PASCAL

1. Todas las partes de sentencias del programa, encabezados y partes de declaraciones deben ser consideradas. Las únicas instrucciones ignoradas deben ser los comentarios.
2. Las variables, constantes (incluyendo las constantes estandares FALSE, TRUE y MAXINT), tipos definidos por el usuario, literales, nombres de archivos y la palabra reservada NIL son contadas como operandos. Todos los operandos son contados como si fueran globales. Es decir, las variables locales con el mismo nombre en diferentes procedimientos son contadas como múltiples ocurrencias del mismo operando.
3. Las siguientes entidades son contadas siempre como un operador simple :

*	/	DIV	MOD	<	>=
<>	>=	>	:=	^	,
NOT	AND	OR	IN	PACKED	TO
INTEGER	REAL	TEXT	CHAR	LABEL	PROGRAM
EXTERN	FORWARD	PROCEDURE	DOWNTO	FUNCTION	

CDC Pascal Release 3 operadores únicos:

Fortran	Seqtext	Segmented	Alfa	Dynamic
Otherwise	Run			

4. Las siguientes entidades múltiples son contadas como un operador simple.

Begin end	Case end	While do	Repeat until
If then	If then Else	For do	With do
Set of	File of	Record end	Array of

5. Las siguientes entidades o pares de entidades son contadas como operadores simples sujetos a las condiciones correspondientes.

VAR es contado como un operador en las listas de parámetros y no es contada como un nivel de sección.

= es contada ya sea como un operador relacional en expresiones o como un operador de definición en secciones no ejecutables del programa.

+ es contado ya sea como un + unario o + binario dependiendo de su función. El + binario no está diferenciado entre el uso aritmético y el uso set.

- es contado como un - unario o un - binario dependiendo de su función. El - binario no está diferenciado entre su uso binario y su uso set.

. es contado ya sea como un símbolo selector de componente de registros o como terminación del programa dependiendo de su función.

:

! es un operador de definición en las sección VAR y listas de parámetros. Es un operador de separación siguiendo CASE o niveles de GOTO.

() es contado ya sea como un operador en la lista de argumentos o un operador de expresión dependiendo de su función.

[] es contado ya sea como un operador subscript o un operador de conjunto dependiendo de su función.

6. Las llamadas a procedimientos y funciones son contadas como operadores. Los nombres de las subrutinas siguiendo un FUNCTION o PROCEDURE no son contadas, ya que es el operando para el operador FUNCTION o PROCEDURE.
7. Las instrucciones GOTO (ex. GOTO y un nivel) son contadas como el operador GOTO y el operando "label".
8. Las declaraciones de niveles no son enumeradas - todos los tokens después del operador LABEL hasta el siguiente punto y coma (inclusive) son ignorados.

9. Los siguientes son dispositivos sintácticos y no son contados:

CONST TYPE VAR (para secciones de variables)

10. Las siguientes son reglas pertenecientes a Pascal 6000 versión 3:

VALUE es un dispositivo sintáctico y no es contado. Comas, () e = en las secciones de VALUE son contadas como en la sección TYPE.

OF en las secciones VALUE es un dispositivo sintáctico y no es contada. OTHERWISE es contado como otro nivel de CASE.

11. La métrica de McCabe, $v(G)$, es contada como sigue:

incremento en las palabras reservadas:

WHILE	FOR	REPEAT	IF	OTHERWISE
AND	OR	PROCEDURE	FUNCTION	PROGRAM

AND y OR incluyen ciclos y saltos controlados por variables booleanas. Los condicionales no podrían ser contados directamente como funciones booleanas y las variables no serían contadas contadas correctamente.

También se cuentan los niveles de CASE por:

Incremento en:

Puntos en la parte ejecutable del programa pero afuera de la lista de parámetros de WRITE(LN).

Comas en la lista de niveles del CASE.

Decremento en:

Palabra reservada LABEL.
Comas en la instrucción LABEL.

El decremento es necesario para remover los niveles GOTO, si hay.

12. VARS es contabilizado por la substracción del número de constantes numéricas y cadenas de caracteres en el programa del conteo del punto 2.

ANEXO 2

Formulario de Métricas de Certificación de Calidad

ANEXO 2

* METRICAS DE CERTIFICACION DE CALIDAD

METRICAS DE TAMANO

* Líneas de código (Ss)

* Contador de Tokens

n_1 = Número de operadores

n_2 = Número de operandos

N_1 = Total de ocurrencias de operadores

N_2 = Total de ocurrencia de operandos

* Tamaño

$$N = N_1 + N_2$$

* Estimador Ss

$$Ss = N / c$$

donde c = cte dependiendo del lenguaje

* Métricas adicionales

Vocabulario

$$n = n_1 + n_2$$

Volúmen

$$V = N \times \log_2 n$$

* Contador de Funciones

METRICAS DE ESTRUCTURAS DE DATOS

- * Cantidad de datos

$$V_2 = \text{Variables} + \text{constantes} + \text{niveles}$$

METRICAS DE ESTRUCTURAS LOGICAS

- * Conteo de decisiones (DE)

- * Complejidad ciclomática

$$v(G) = e - n + 2$$

donde :

e = número de líneas de dirección en el diagrama de flujo .

n = número de nodos en el diagrama de flujo

Además:

$$v(G) = DE + 1$$

y para programas de módulos

$$v(G) = v(G_1) + v(G_2) + \dots$$

- * Métricas de extensión y número mínimo de trayectorias

N_p = número mínimo de trayectorias

R = Extensión

\bar{R} = Extensión promedio = R / número de nodos

- * Niveles anidados (NL)

METRICAS COMPUESTAS

- * Longitud del Programa

$$\bar{N} = n_1 \times \log_2 n_1 + n_2 \log_2 n_2$$

- * Volúmen del programa

$$V = N \times \log_2 2n$$

- * Volúmen potencial

$$V^* = V \times L^{\wedge}$$

donde:

$$L^{\wedge} = (2 / n_1) \times (n_2 / N_2)$$

- * Nivel del programa

$$L = V^* / V$$

- * Dificultad

$$D = 1 / L$$

- * Esfuerzo

$$E = V / L = D \times V = (n_1 N_2 \log_2 n) / 2n_2$$

- * Tiempo de programación

$$T = E / B$$

donde:

$$B = \text{aprox. } 18 \text{ seg}$$

- * Nivel de Lenguaje

$$x = L \times V^* = L^2 V$$

Anexo 3: Programas Fuente

ANEXO 3
Diagramas de Flujo

DIAGRAMA DE FLUJO PROGRAMA MONEY_0

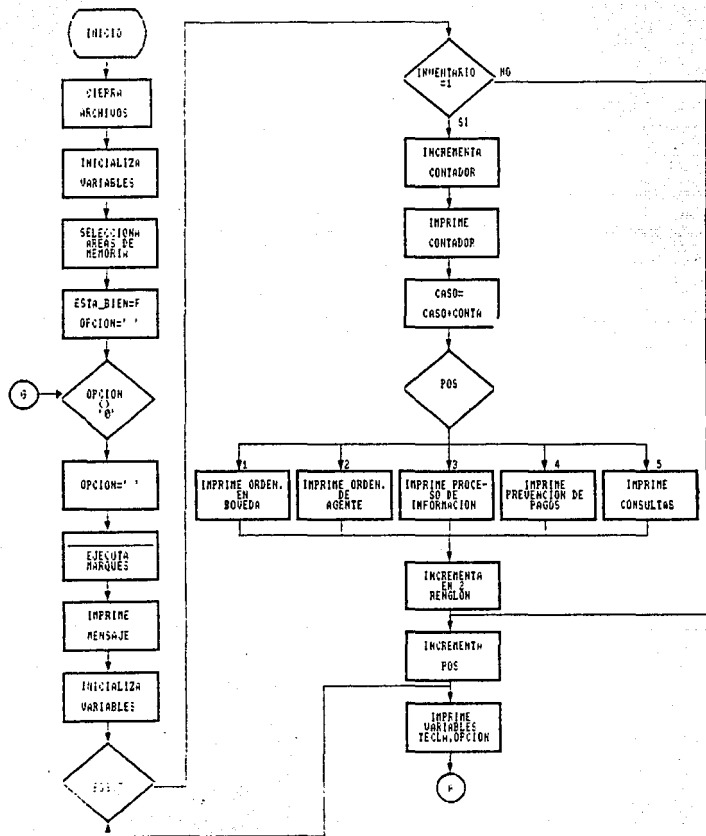


DIAGRAMA DE FLUJO PROGRAMA MONEY

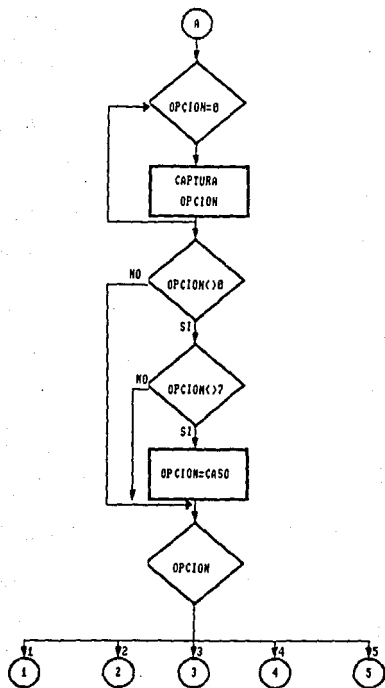


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

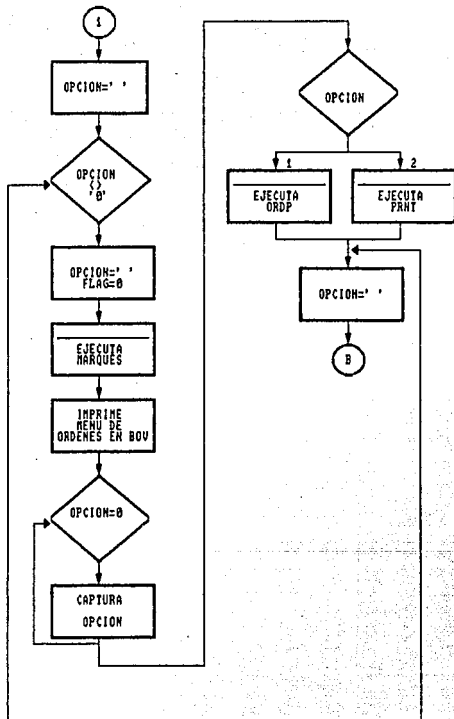


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

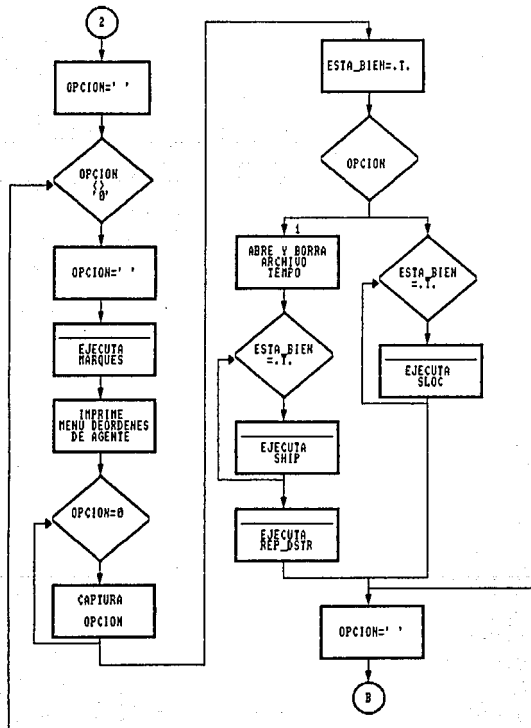


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

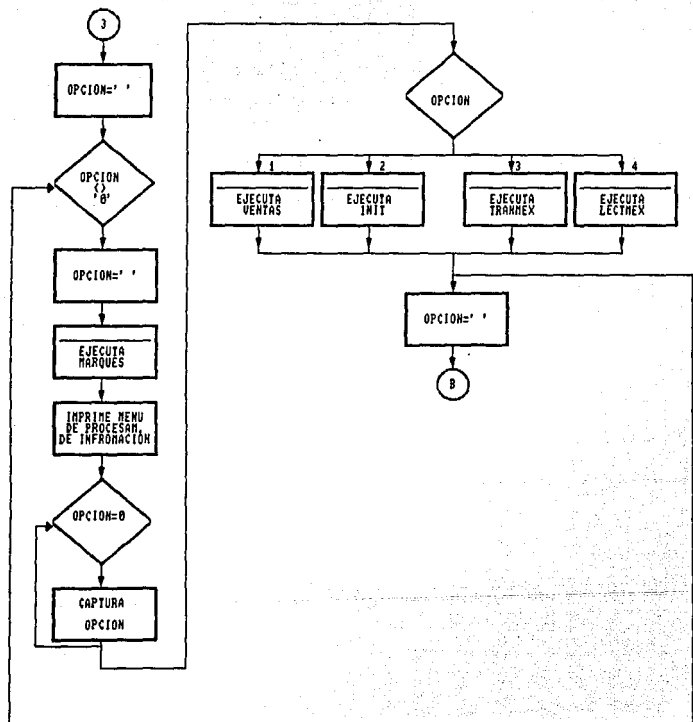


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

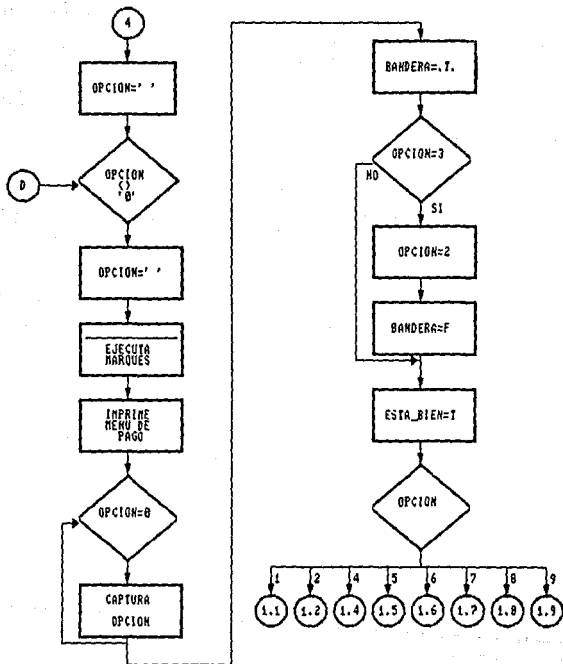


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

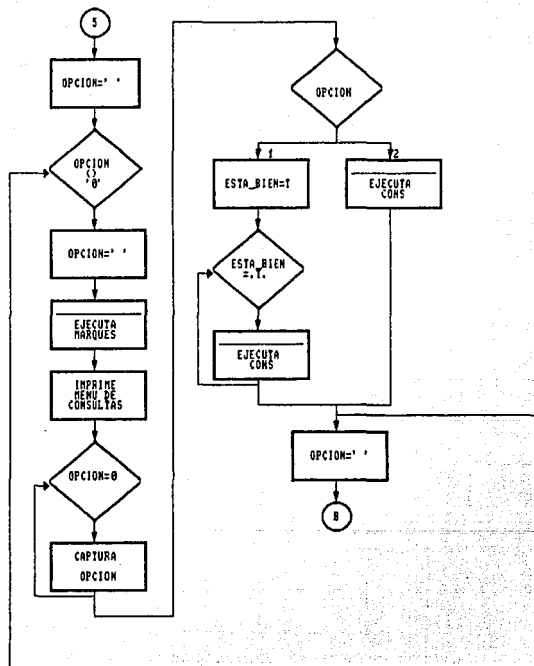


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

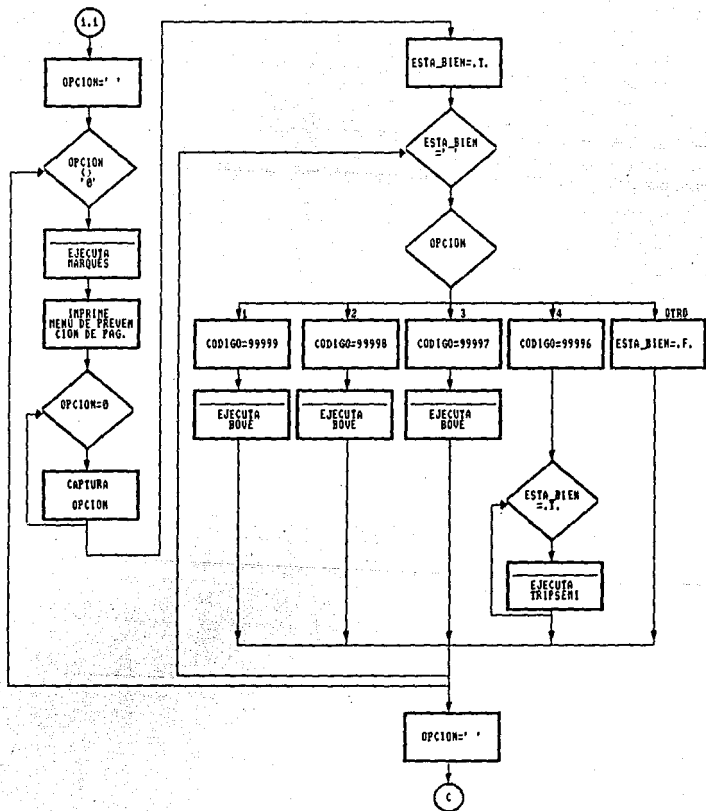


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

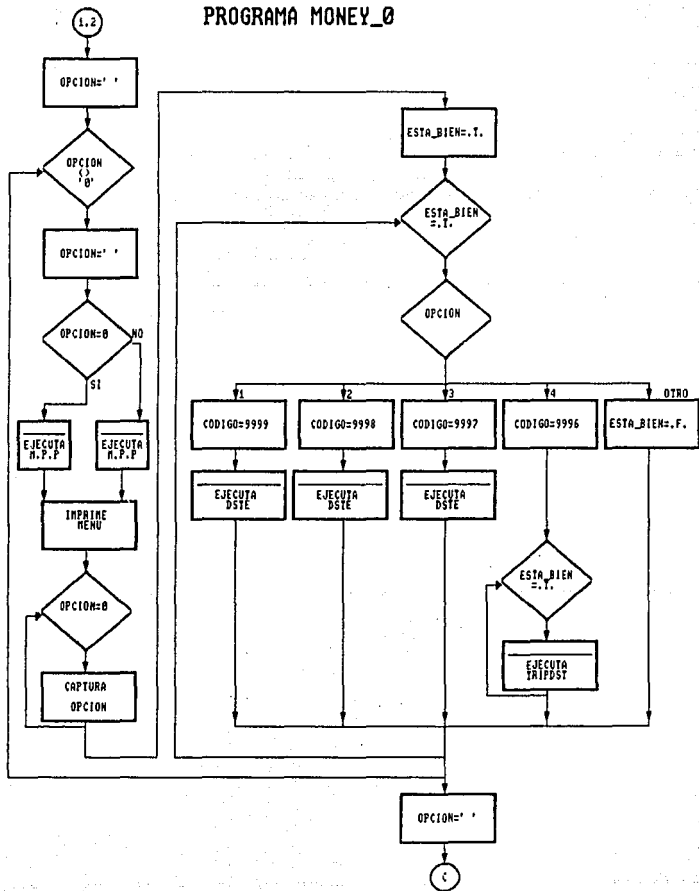


DIAGRAMA DE FLUJO PROGRAMA MONEY_0

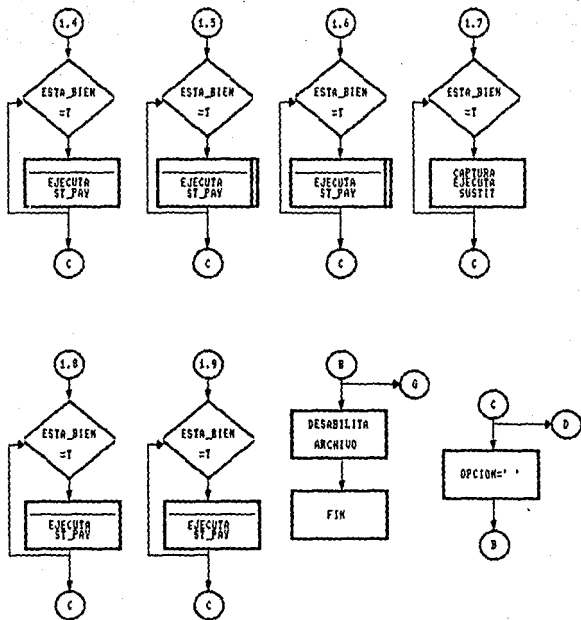


DIAGRAMA DE FLUJO PROGRAMA ORDP

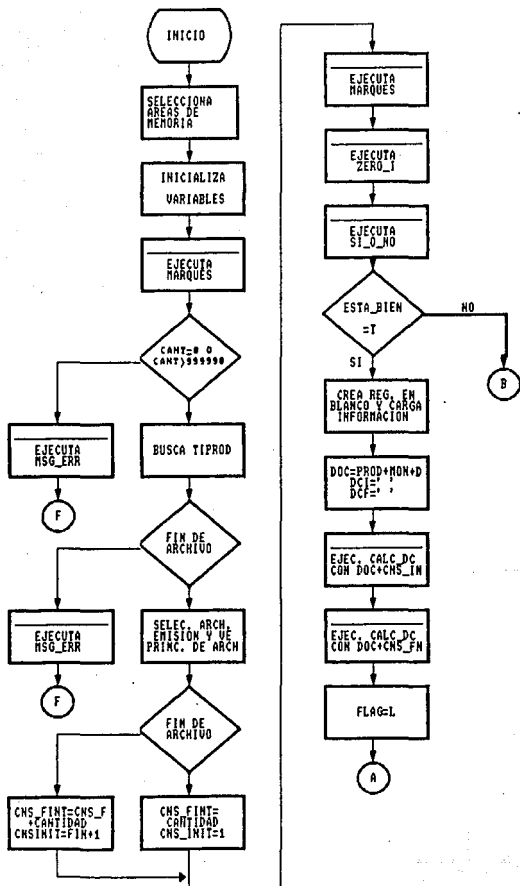


DIAGRAMA DE FLUJO PROGRAMA ORDP

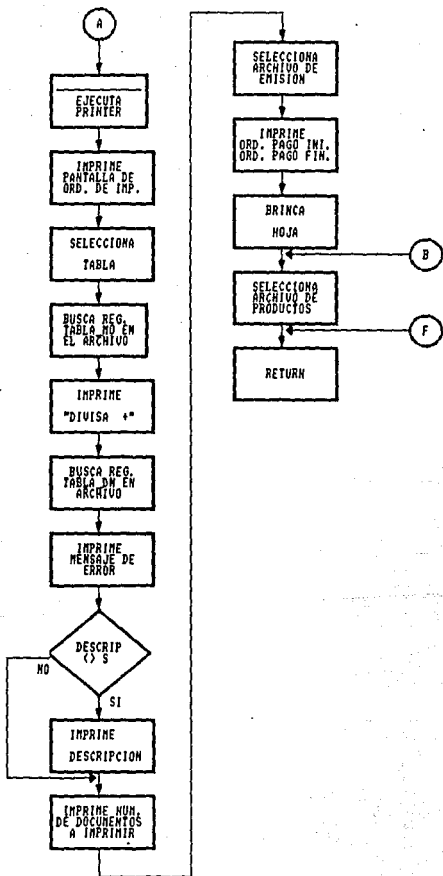


DIAGRAMA DE FLUJO PROGRAMA PRNT

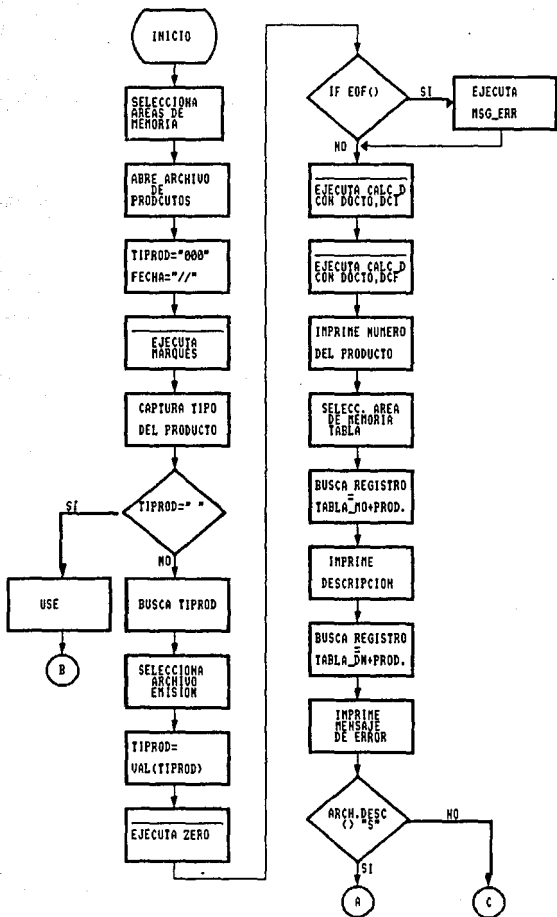


DIAGRAMA DE FLUJO PROGRAMA PRNT

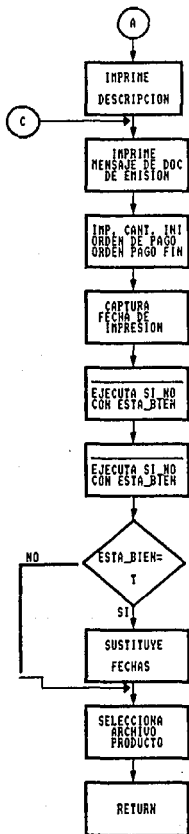


DIAGRAMA DE FLUJO PROGRAMA REP_DSTR

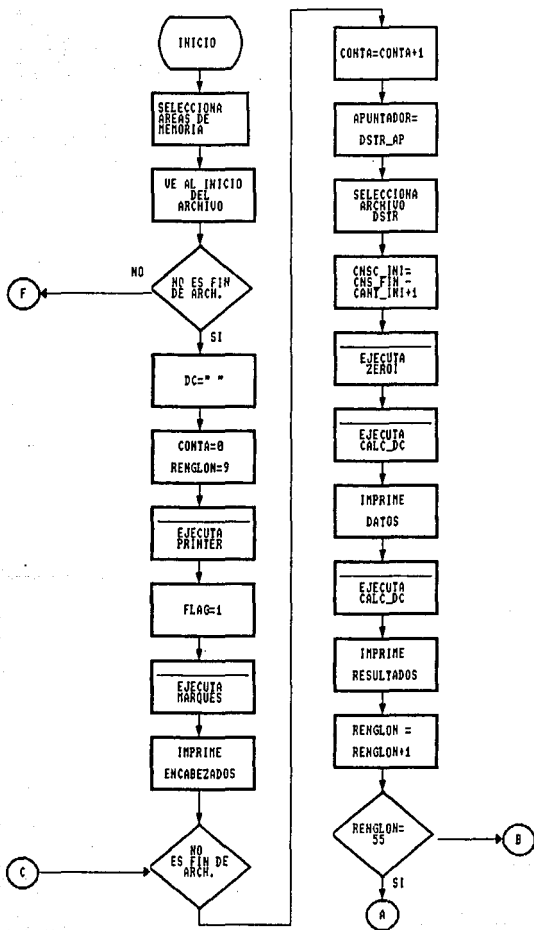


DIAGRAMA DE FLUJO PROGRAMA REP_DSTR

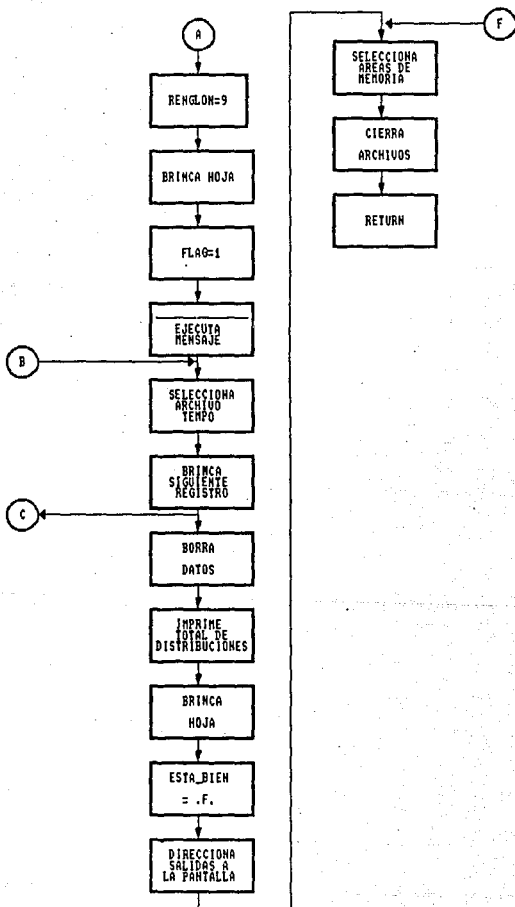


DIAGRAMA DE FLUJO PROGRAMA UTAS

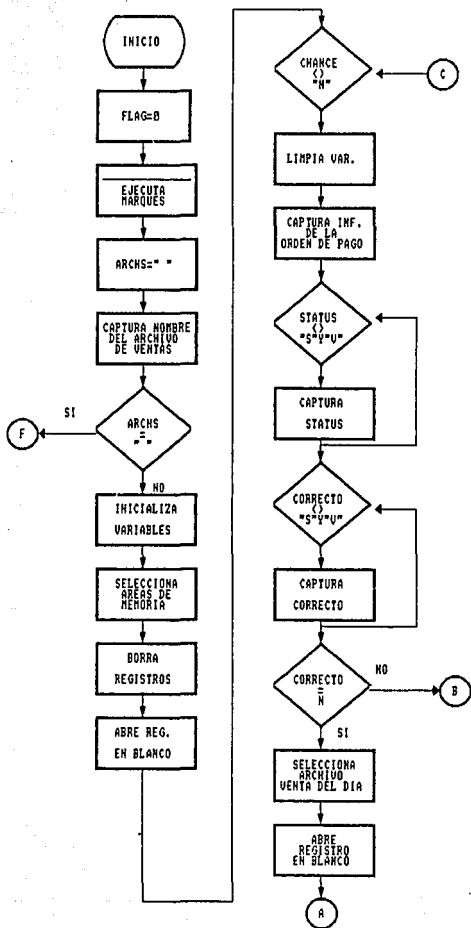


DIAGRAMA DE FLUJO PROGRAMA SLOC

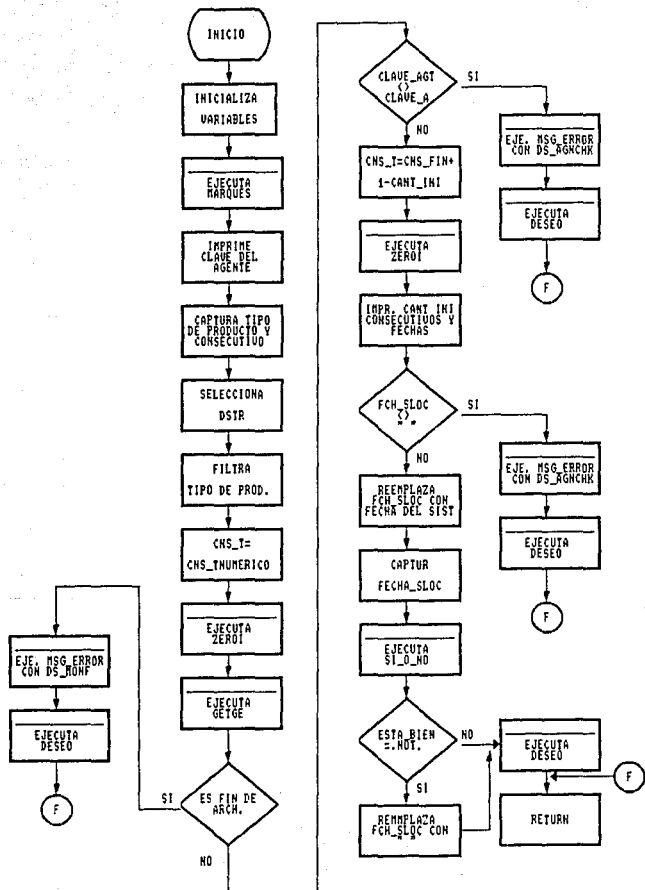


DIAGRAMA DE FLUJO PROGRAMA UTAS

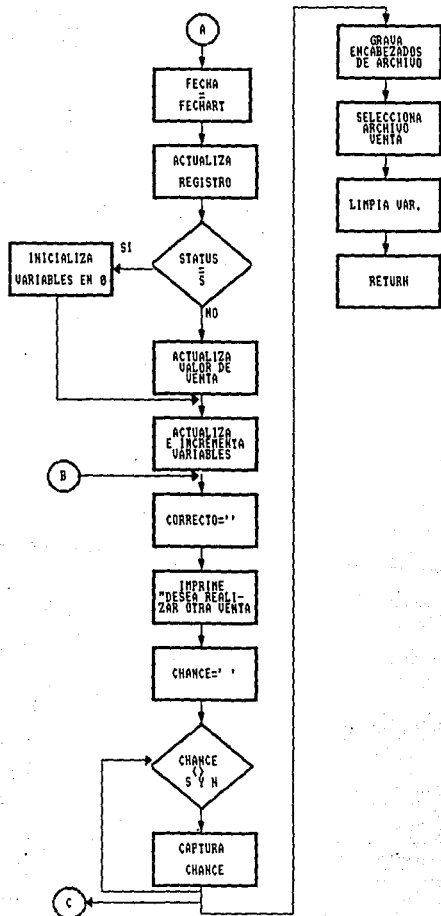


DIAGRAMA DE FLUJO PROGRAMA INIT

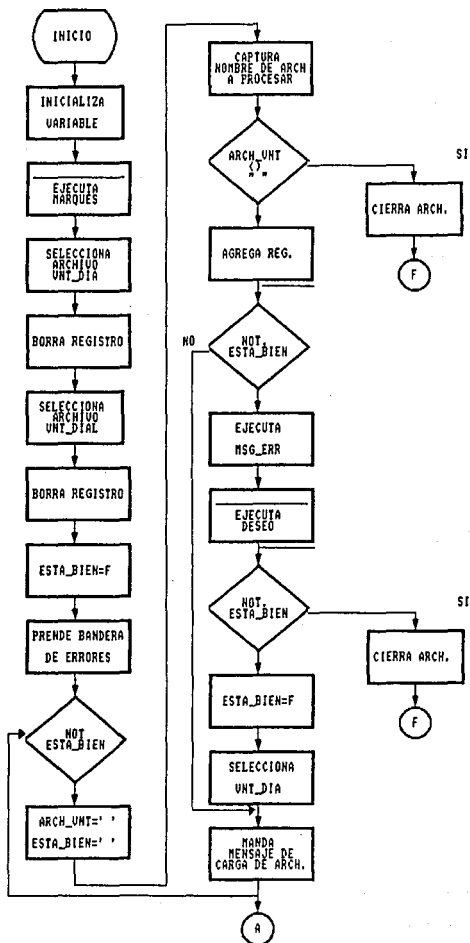


DIAGRAMA DE FLUJO PROGRAMA INIT

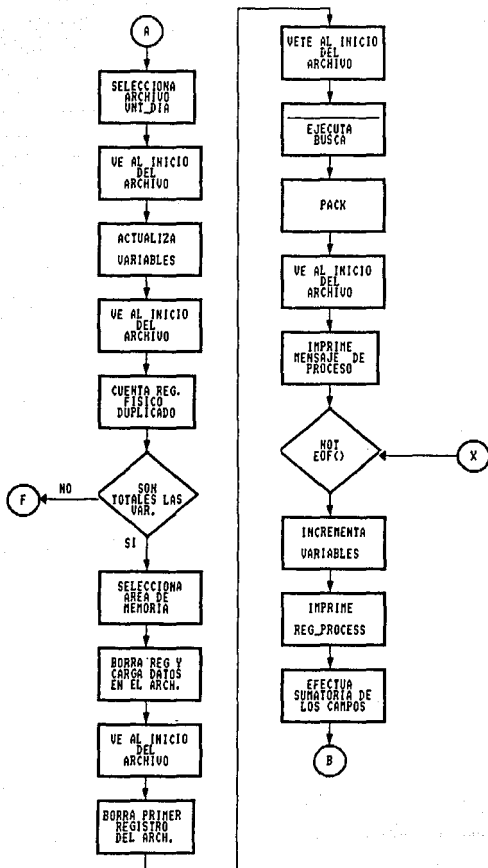


DIAGRAMA DE FLUJO PROGRAMA INIT

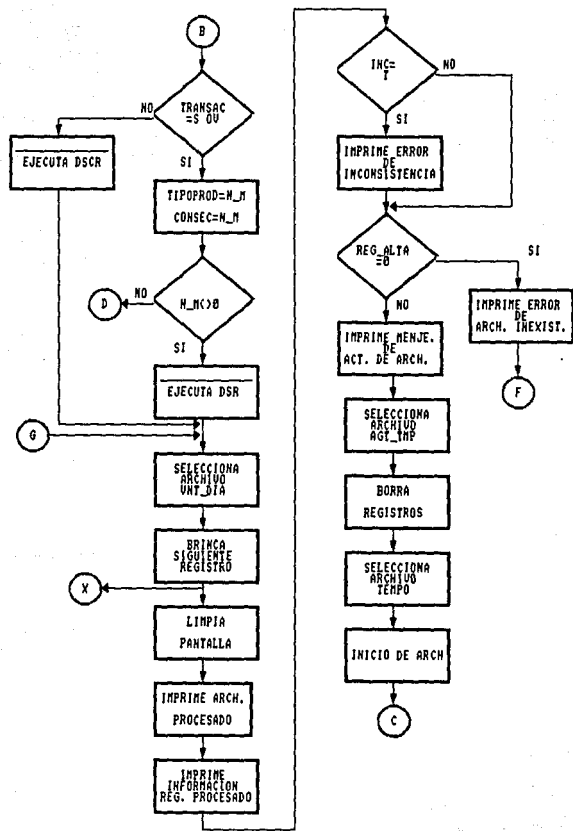


DIAGRAMA DE FLUJO PROGRAMA INIT

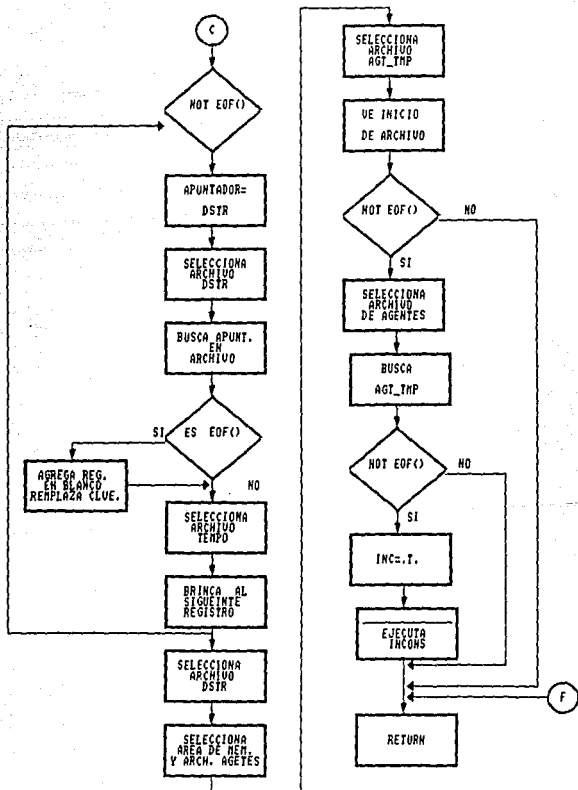


DIAGRAMA DE FLUJO PROGRAMA INIT

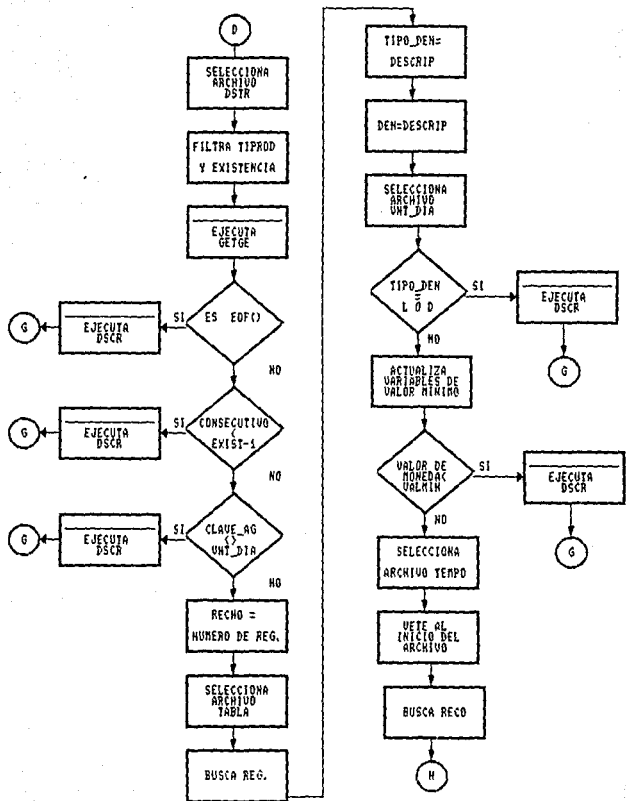


DIAGRAMA DE FLUJO PROGRAMA INIT

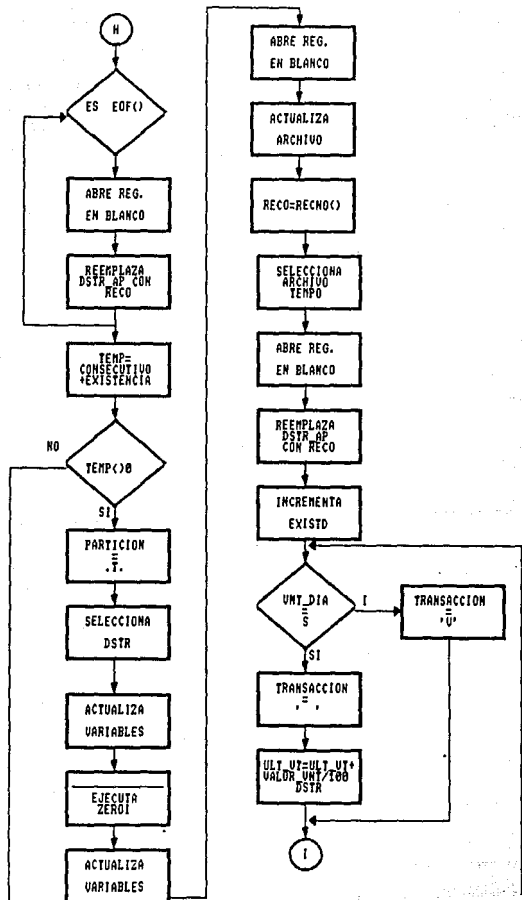


DIAGRAMA DE FLUJO PROGRAMA INIT

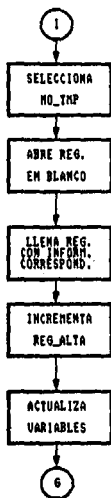


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

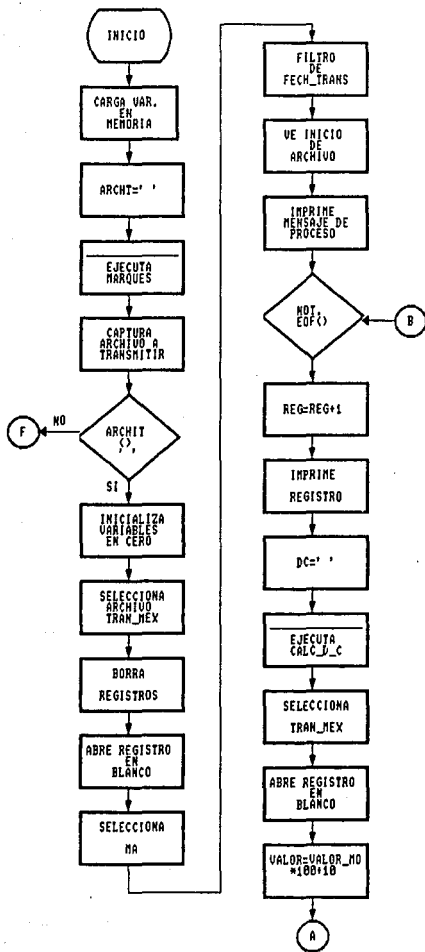


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

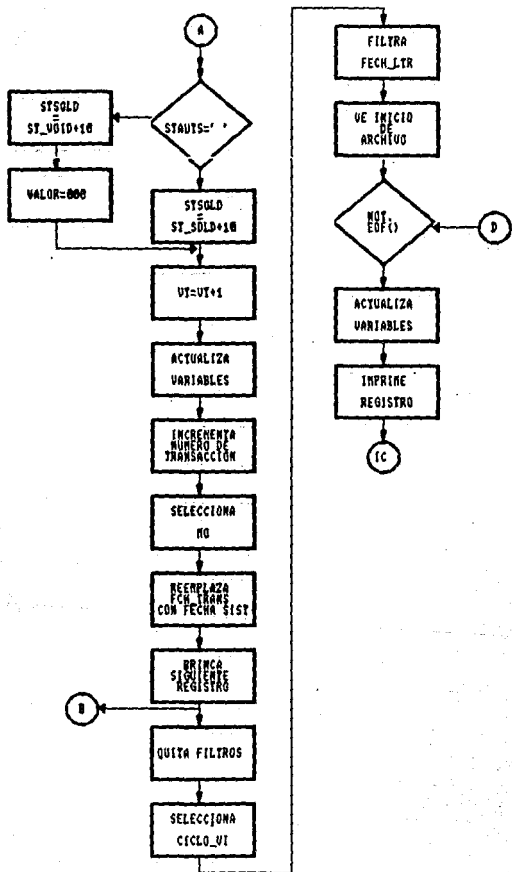


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

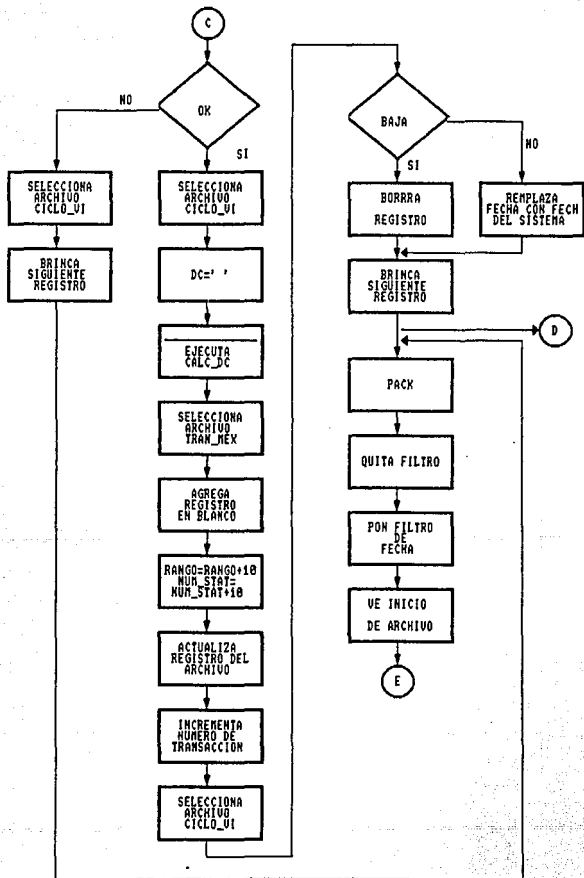


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

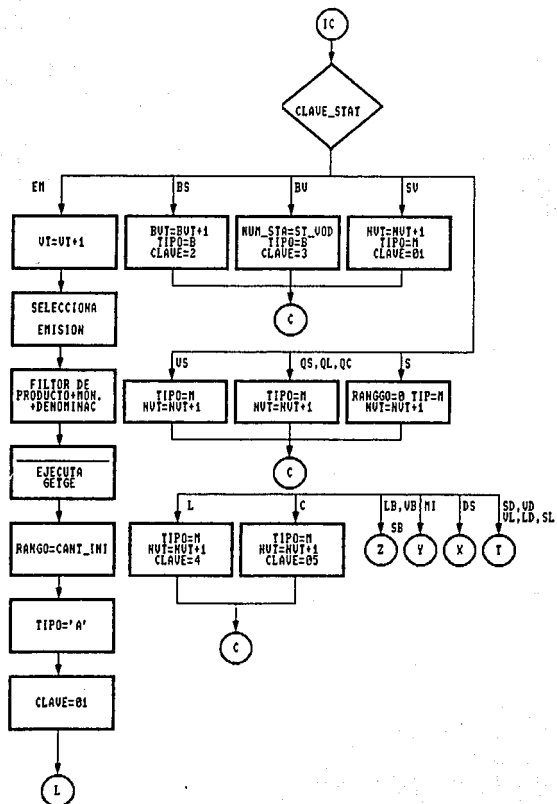


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

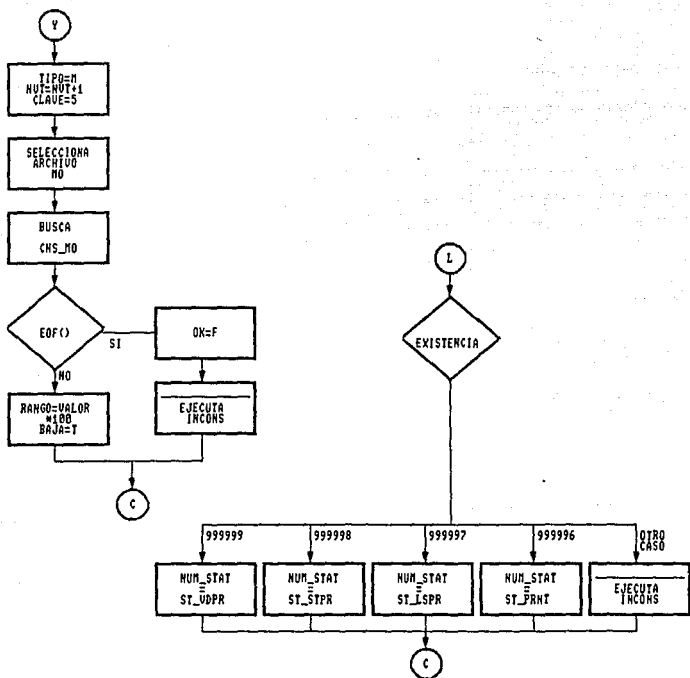


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

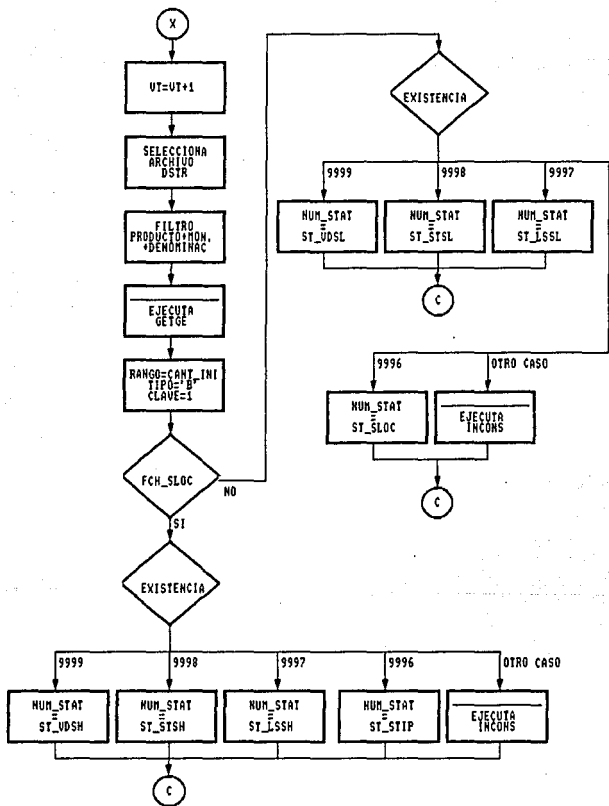


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

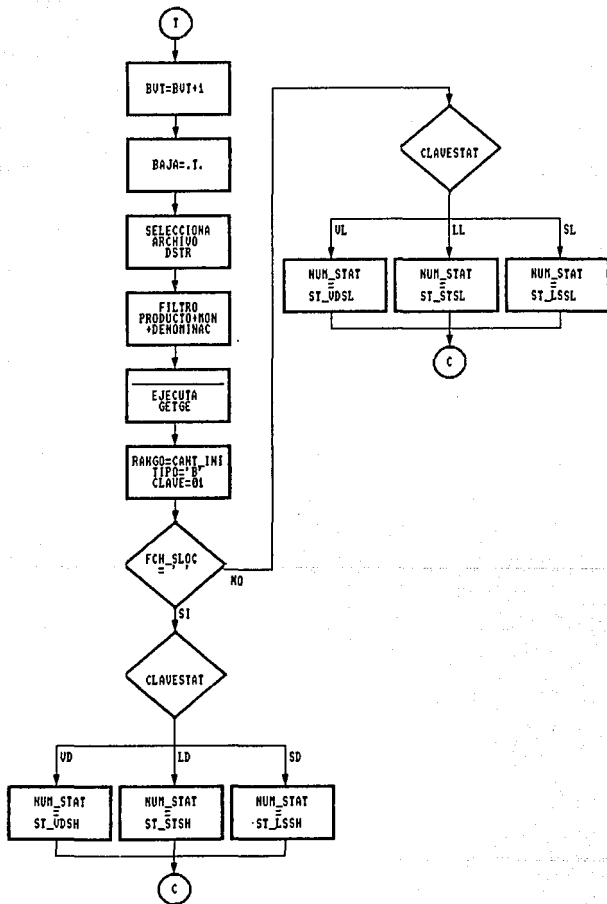


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

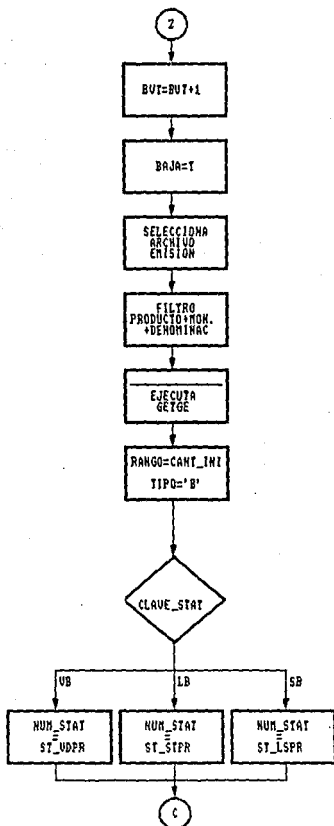


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

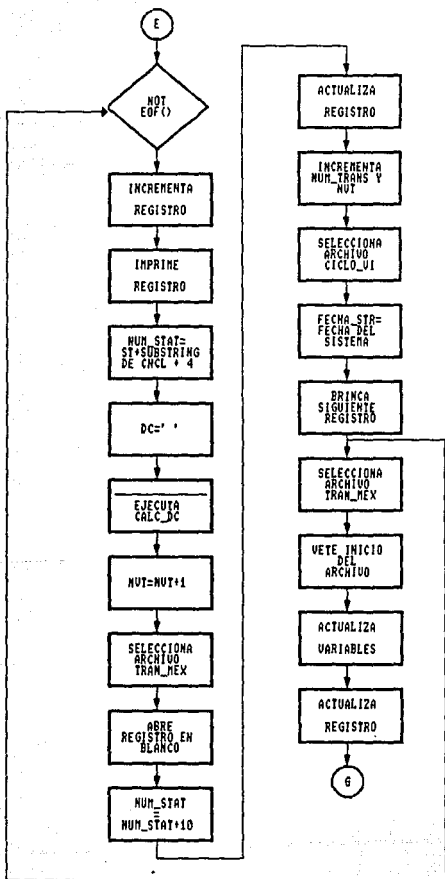


DIAGRAMA DE FLUJO PROGRAMA TRAN_MEX

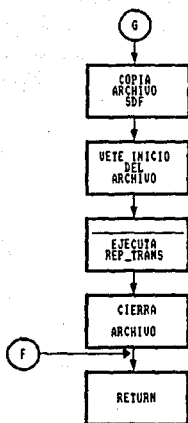


DIAGRAMA DE FLUJO PROGRAMA SHIP

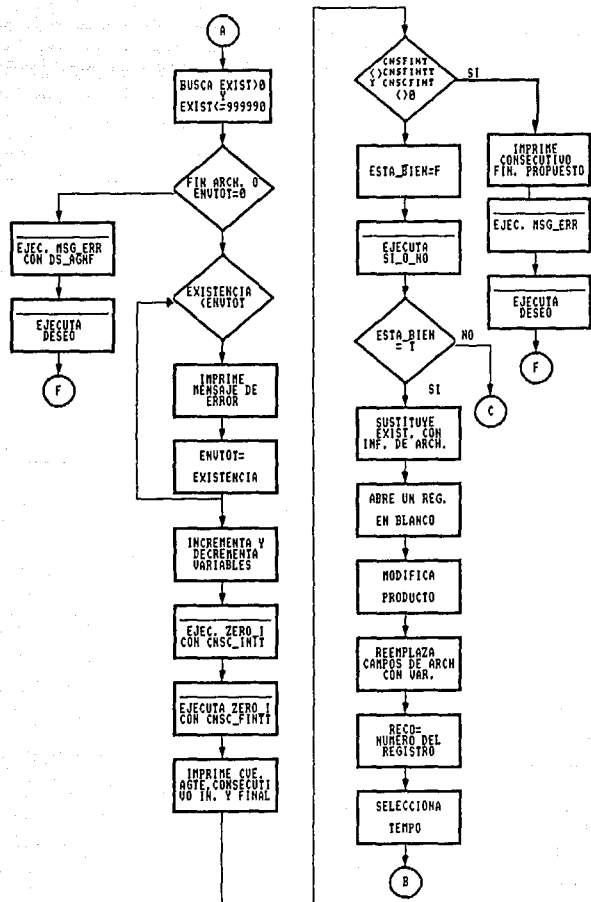
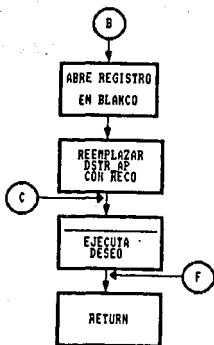


DIAGRAMA DE FLUJO PROGRAMA SHIP



ANEXO 4

Programas Fuente

 Programa de manejo de las Ordenes de Pago
 H O N E Y O

```

Close all
Set Date ANSI
Store "1" to Sel_Bain
Store "2" to Sel_Datr
Store "3" to Sel_Mo
Store "4" to Sel_Ciclo
Store "5" to Sel_Ig
Store "6" to Sel_Trans
Store "7" to Sel_Dscr
Store "8" to Sel_Prod
Store "9" to Sel_Tabla
Flag = "0"
Select &Sel_Bain
Use Emision
Select &Sel_Datr
Use Datr Index Agnt_Ds
Select &Sel_Mo
Use Mo Index No_Mo, No_Fecha, No_Fpg, No_Hora, No_Pch, No_Pcbp
Select &Sel_Ciclo
Use Ciclo_Vi Index Ciclo_Vi
Select &Sel_Tabla
Use Tabla Index Tabla
Restore From Tabla Additive
Restore From Ds Tabla Additive
Restore From St Tabla Additive
Esta_Bien = ".P."
* En este archivo vienen los n/meros de las tablas en uso
*      Tabla_BP - N/mero de tabla de bancos pagadores
*      Tabla_Mo - N/mero de la tabla de las monedas
*      Tabla_Dn - N/mero de las diferentes denominaciones
*      Tabla_Ds - N/mero de las discrepancias (errores)
*      Tabla_St - N/mero de los posibles estatus de un P.O.

Opcion = ""
Do While Opcion <> "0"
  Opcion = "1"
  Do Marques With "Sistema Ordenes de Pago", "Inventarios"
  @ 6 10 SAY "0. Regreso al Men/ anterior"
  COMTA = 0
  CASO = SPACE(5)
  POS = 1
  REM = 8
  DO WHILE POS < 8
    IF SUBSTR(INV OKT, POS, 1) = "1"
      COMTA = COMTA + 1
      @ REM, 10 SAY COMTA PICTURE "9"
      CASO = STUFF(CASO, COMTA, 1, STR(POS, 1))
      DO CASE
        CASE POS = 1
          @ REM, 31 SAY "- Ordenes en B'veda"
        CASE POS = 2
          @ REM, 31 SAY "- Ordenes de Agentes"
        CASE POS = 3
          @ REM, 31 SAY "- Procesamiento de Informaci'n"
        CASE POS = 4
          @ REM, 31 SAY "- Prevenciones de Pago"
        CASE POS = 5
          @ REM, 31 SAY "- Consultas"
        CASE POS = 6
          @ REM, 31 SAY "- Reportes"
        CASE POS = 7
          @ REM, 31 SAY "- Utillerias"
      ENDCASE
      REM = REM + 2
    ENDDIF
    POS = POS + 1
  ENDDO
  REM = REM + 1
  @ REM, 35 SAY "Teclea tu Opci'n:"
  Do While At(Opcion, "01234567") = 0
    @ Rem, 57 Get Opcion Picture "9"
  Read
EndDo
If Opcion <> "0"
  If Opcion <> "1"
    Opcion = SubStr(Caso, Val(Opcion), 1)
  EndIf
EndIf
Do Case

```



```

* * * * *
Flag = "0"
Do Marques With "Ordenes en B'veda", "M E N U"
@ 8, 30 SAT "0. Regreso al Men/ anterior"
@ 10, 30 SAT "1. Orden de Impresi'n"
@ 12, 30 SAT "2. Altas en B'veda"
@ 18, 35 SAT "Tecla tu Opci'n:"
Do While At(Opcion, 012) = 0
@ 18, 57 Get Opcion Picture "9"
Read
EndDo
Do Case
Case Opcion = "1"
Do OrdP
Case Opcion = "2"
Do Prnt
EndCase
EndDo
Opcion = " "
Case Opcion = "3"
Opcion = " "
Do While Opcion <> "0"
Opcion =
Do Marques With "Ordenes de Agentes", "M E N U"
@ 8, 30 SAT "0. Regreso al Men/ anterior"
@ 10, 30 SAT "1. Distribuci'n a los Agentes"
@ 12, 30 SAT "2. Altas en Localidad de Venta"
@ 18, 35 SAT "Tecla tu Opci'n:"
Do While At(Opcion, 012) = 0
@ 18, 57 Get Opcion Picture "9"
Read
EndDo
Esta_Bien = .T.
Do Case
Case Opcion = "1"
Select ASel_Prod
Use Tempo index Tempo
Zap
Do While Esta_Bien
Do Sblp
EndDo
Do Rep_Dstr "2"
Case Opcion = "2"
Do While Esta_Bien
Do Sloc
EndDo
EndCase
EndDo
Opcion = " "
Case Opcion = "4"
Opcion = " "
Do While Opcion <> "0"
Opcion =
Do Marques With "Procesamiento de Informaci'n", "M E N U"
@ 8, 30 SAT "0. Regreso al Men/ anterior"
@ 10, 30 SAT "1. Captura de Ventas"
@ 12, 30 SAT "2. Procesamiento de Ventas"
@ 14, 30 SAT "3. Transmis'i'n de Dptos a Mxico"
@ 16, 30 SAT "4. Captura de Pagos"
@ 18, 30 SAT "5. Recepci'n de Dptos de Mxico"
@ 22, 35 SAT "Tecla tu Opci'n:"
Do While At(Opcion, 012345) = 0
@ 22, 57 Get Opcion Picture "9"
Read
EndDo
Do Case
Case Opcion = "1"
Do Ventas
Case Opcion = "2"
Do Init
Case Opcion = "3"
Do Tran_Mex
Case Opcion = "4"
Do Pagos
Case Opcion = "5"
Do Lect_Mer
EndCase
EndDo
Opcion = " "
Case Opcion = "4"
Opcion = " "
Do While Opcion <> "0"
Opcion =
Do Marques With "Previsiones de Pago", "M E N U"
@ 8, 30 SAT "0. Regreso al Men/ anterior"
@ 8, 30 SAT "1. En B'veda"
@ 10, 30 SAT "2. En Distribuci'n"
@ 12, 30 SAT "3. En Localidad de Venta"
@ 14, 30 SAT "4. En Localidad de Venta"

```

```

@ 16, 30 SAY "5. Perdida por el Cliente"
@ 18, 30 SAY "6. Cancelaci' n despues de la Venta"
@ 20, 30 SAY "7. Ordenes de pago a IPS despues de la venta"
@ 16, 30 SAY "8. Reembolso"
@ 20, 30 SAY "9. Sustituci' n"
@ 23, 35 SAY "Tecl ea tu Opci' n:"
Do While At{Opcion,"01234567"} = 0
@ 23, 57 Get Opcion Picture "9"
Read
EndDo
Bandera = .F.
IF OPCION = "1"
  OPCION = "2"
  Bandera = .F.
EndIf
Esta_Bien = .F.
Do Case
  Case Opcion = "1"
    Opcion = " "
    Do While Opcion <> "0"
      Opcion =
      Do Marques With "Previsiones de Pago en B'veda","M E N U"
        @ 8, 35 SAY "0. Regreso al Men# anterior"
        @ 10, 35 SAY "1. Anuladas"
        @ 12, 35 SAY "2. Robadas"
        @ 14, 35 SAY "3. Perdidas"
        @ 16, 35 SAY "4. En Poder de IPS"
        @ 19, 32 SAY "Tecl ea tu Opci' n:"
      Do While At{Opcion,"01234"} = 0
        @ 19, 54 Get Opcion Picture "9"
      Read
    EndDo
    Esta_Bien = .F.
  Do While Esta_Bien
    Do Case
      Case Opcion = "1"
        Codigo = 999999
        Do BorE
      Case Opcion = "2"
        Codigo = 999998
        Do BorE
      Case Opcion = "3"
        Codigo = 999997
        Do BorE
      Case Opcion = "4"
        Codigo = 999996
        Do While Esta_Bien
          Do TrIPSEnt
        EndDo
      Otherwise
        Esta_Bien = .F.
    EndCase
  EndDo
EndDo
OPCION = " "
Case Opcion = "2"
  OPCION =
  Do While Opcion <> "0"
    Opcion =
    IF Bandera
      Do Marques With "Previsiones de Pago en Distribuci' n","M E N U"
    Else
      Do Marques With "Previsiones de Pago en Loc. de Venta","M E N U"
    EndIf
    @ 8, 35 SAY "0. Regreso al Men# anterior"
    @ 10, 35 SAY "1. Anuladas"
    @ 12, 35 SAY "2. Robadas"
    @ 14, 35 SAY "3. Perdidas"
    @ 16, 35 SAY "4. En Poder de IPS"
    @ 19, 32 SAY "Tecl ea tu Opci' n:"
    Do While At{Opcion,"01234"} = 0
      @ 19, 54 Get Opcion Picture "9"
    Read
  EndDo
  Esta_Bien = .F.
  Do While Esta_Bien
    Do Case
      Case Opcion = "1"
        Codigo = 9999
        Do DstE
      Case Opcion = "2"
        Codigo = 9998
        Do DstE
      Case Opcion = "3"
        Codigo = 9997
        Do DstE
      Case Opcion = "4"
        Codigo = 9996
        Do DstE
    EndCase
  EndDo
EndDo

```

```

000000 - 2000
Do While Esta_Bien
  Do TrIPSDat With Bandera
  EndDo
  Otherwise
    Esta_Bien = .F.
  EndCase
EndDo
EMDDO
Opcion = " "
Case Opcion = "4"
  Do While Esta_Bien
  Do Stp_Pay With "Orden de Pago Robada", "SOLD+", "S"
  EndDo
Case Opcion = "5"
  Do While Esta_Bien
  Do Stp_Pay With "Orden de Pago Perdida", "SOLD+", "L"
  EndDo
Case Opcion = "6"
  Do While Esta_Bien
  Do Stp_Pay With "Orden de Pago Cancelada", "SOLD+", "C"
  EndDo
Case Opcion = "7"
  Do While Esta_Bien
  Do TrIPsvta With "Orden de Pago En Poder de IPS", "SOLD+", "I"
  EndDo
Case Opcion = "8"
  Do While Esta_Bien
  Do Stp_Pay With "Orden de Pago Reembolsada", "SOLD+STLN+LOST+", "R"
  EndDo
Case Opcion = "9"
  Do While Esta_Bien
  Do Sustit With "Sustituci"n de una Orden de Pago", "SOLD+STLN+LOST+", "D"
  EndDo
EndCase
EMDDO
Opcion = " "
Case Opcion = "5"
  Opcion = "4"
  Do While Opcion (<) "0"
    Opcion = " "
    Do Marques With "C O N S U L T A S", "M R N D"
    # 3, 10 SAY "0. Regreso al Menf Anterior"
    # 11, 10 SAY "1. Por Numero de la Orden de Pago"
    # 13, 10 SAY "2. Por Status de la Orden de Pago"
    # 19, 12 SAY "Teclsa tu Opci"n:"
    Do While At(Opcion,"012") = 0
      # 19, 54 Get Opcion Picture "9"
    EndDo
    Read
  EndDo
  Do Case
  Case Opcion = "1"
    Esta_Bien = .F.
    Do While Esta_Bien
    Do Cons
    EndDo
  Case Opcion = "2"
    Do Constat
  EndCase
EndDo
EMDDO
Opcion = " "
Case Opcion = "6"
  Opcion = " "
  I =
  Restore from Consul Additive
  Do While I (<) "0"
    I =
    Do Marques With "Sistema Ordenes de Pago", "R e p o r t e s"
    # 6, 20 Say "0.- Regreso al Menf Anterior"
    # 7, 20 Say "1.- Inventario de IPS"
    # 8, 20 Say "2.- Inventario por Agente"
    # 9, 20 Say "3.- Margen de Utilidad por Venta por Agente por Fecha"
    # 10, 20 Say "4.- Margen de Utilidad (Resumen por Agente)"
    # 11, 20 Say "5.- Ventas Semanales (Resumen)"
    # 12, 20 Say "6.- Ventas por Condado por Agente (Resumen Semanal)"
    # 13, 20 Say "7.- Ventas por Agente (Resumen Semanal)"
    # 14, 20 Say "8.- Ventas por Agente por Hora de Venta (Resumen)"
    # 15, 20 Say "9.- Ventas por Condado por Fecha por Agente"
    # 16, 20 Say "A.- Ventas por Condado por Fecha"
    # 17, 20 Say "B.- Pagos por Sucursal (Resumen Semanal)"
    # 18, 20 Say "C.- Pagos Semanales (Resumen)"
    # 19, 20 Say "D.- Pagos"
    # 20, 20 Say "E.- Ventas"
    # 23, 24 Say "Teclsa el Tipo de Reporte deseado: "
    Do While At(I,"0123456789ABCDE") = 0
      # 23, 60 Get I Picture " "
      I = Upper(I)
    EndDo
    Read
  EndDo

```

```

Do Case
  Case X = '1'
    Do RepoStat
  Case X = '2'
    Do Repinvpa
  Case X = '3'
    Do RepMarUt
  Case X = '4'
    Do Reprearu
  Case X = '5'
    Do Repreagr
  Case X = '6'
    Do Repreacn
  Case X = '7'
    Do Repreang
  Case X = '8'
    Do Reprenbr
  Case X = '9'
    Do Repvncnd
  Case X = 'A'
    Do Repvncn
  Case X = 'B'
    Do Repspag
  Case X = 'C'
    Do Repspag
  Case X = 'D'
    Do Rep_Pago
  Case X = 'E'
    Do Rep_Vtas
EndCase

EndDo
Opcion = ""
Case Opcion = "1"
  Opcion = "1"
  Do While Opcion <> "0"
    Opcion = "1"
    Do Marques With "Utilerias de Inventarios", "M E N U"
    @ 9, 20 SAT "0. Regreso al Menú anterior"
    @ 10, 20 Say "1. Pusi'n de Archivos OutGoing"
    @ 11, 20 SAT "2. Baja de Paquetes"
    @ 12, 20 Say "3. Liberaci'n de Previsiones de Pago antes de la Venta"
    @ 13, 20 Say "4. Liberaci'n de Previsiones de Pago despues de la Venta"
    @ 14, 20 SAT "5. Reasignaci'n de Paquetes en Distribuci'n"
    @ 15, 20 SAT "6. Modificaci'n de Fechas de Emissi'a y B'veda"
    @ 16, 20 SAT "7. Modificaci'n de Fechas de Distribuci'n y Loc. Venta"
    @ 17, 20 SAT "8. Anulaci'a de un P.O. Vendido"
    @ 18, 20 SAT "9. Rehabilitaci'a de un P.O. Anulado"
    @ 19, 20 SAT "A. Modificaci'n del Importe de un P.O. Vendido"
    @ 23, 20 SAT "Teclas tu Opci'n:"
    Do While At(Opcion,"0123456789A") = 0
    @ 23, 38 Get Opcion Picture "!"
    Read
  EndDo
  Esta_Bien = .T.
  Do Case
    Case Opcion = "1"
      Do Unir
    Case Opcion = "2"
      Opcion = "2"
      Do While Opcion <> "0"
        Opcion = "2"
        Do Marques With "Baja de Paquetes", "M E N U"
        @ 8, 20 SAT "0. Regreso al Menú anterior"
        @ 10, 20 SAT "1. Baja de Paquetes en Orden de Impresi'n"
        @ 12, 20 SAT "2. Baja de Paquetes en B'veda"
        @ 14, 20 SAT "3. Baja de Paquetes en Distribuci'n"
        @ 16, 20 SAT "4. Baja de Paquetes en Localidad de Venta"
        @ 18, 20 SAT "5. Baja de Ventas o Cancelaciones en la T.P.V."
        @ 22, 31 SAT "Teclas tu Opci'n:"
        Do While At(Opcion,"012345") = 0
        @ 22, 50 Get Opcion Picture "9"
        Read
      EndDo
      Esta_Bien = .T.
    Do Case
      Case Opcion = "1"
        Do While Esta_Bien
          Do BajaOrdP
        EndDo
      Case Opcion = "2"
        Do While Esta_Bien
          Do BajaPrnt
        EndDo
      Case Opcion = "3"
        Do While Esta_Bien
          Do BajaShip
        EndDo
    Case Opcion = "4"

```

```

Case Opcion = 7
  Do While Esta_Bien
    Do BajaSloc
  EndDo
Case Opcion = "5"
  Do While Esta_Bien
    Do BajaSold
  EndDo
EndCase
EndDo
Opcion = " "
Case Opcion = "3"
  Opcion = " "
  Do While Opcion <> "0"
    Opcion =
    Do Marques With "Liberaci"o de Prevenciones de Pago", "M E N U"
    @ 8, 20 SAY "0. Regreso al Men/ anterior"
    @ 10, 20 SAY "1. Liberaci"o de un Paquete en B"veda"
    @ 12, 20 SAY "2. Liberaci"o de un Paquete en Distribuci"o"
    @ 14, 20 SAY "3. Liberaci"o de un Paquete en Loc. de Venta"
    @ 22, 35 SAY "Teclea tu Opci"o:"
  Do While At(Opcion, "0123") = 0
    @ 22, 57 Get Opcion Picture "9"
  Read
  EndDo
  Esta_Bien = .T.
  Do Case
    Case Opcion = "1"
      Do While Esta_Bien
        Do LibPrnt
      EndDo
    Case Opcion = "2"
      Do While Esta_Bien
        Do LibShip
      EndDo
    Case Opcion = "3"
      Do While Esta_Bien
        Do LibSloc
      EndDo
  EndCase
  EndDo
  Opcion = " "
  Case Opcion = "4"
    Do While Esta_Bien
      Do LibVen
    EndDo
  Case Opcion = "5"
    Do While Esta_Bien
      Do Reasigna
    EndDo
  Case Opcion = "6"
    Do While Esta_Bien
      Do FecEmis
    EndDo
  Case Opcion = "7"
    Do While Esta_Bien
      Do FecDist
    EndDo
  Case Opcion = "8"
    Do While Esta_Bien
      Do Ven_Anu
    EndDo
  Case Opcion = "9"
    Do While Esta_Bien
      Do Anu_Ven
    EndDo
  Case Upper(Opcion) = "A"
    Do While Esta_Bien
      Do Camb_imp
    EndDo
  Opcion = " "
  EndCase
EndDo
Opcion = " "
EndCase
EndDo
* Deshabilita reas de archivos.
Select &Sel_Emis
Use
Select &Sel_Dstr
Use
Select &Sel_No
Use
Select &Sel_Ciclo
Use
Select &Sel_Ag
Use
Select &Sel_Trans
Use

```

```
---
Select &Sel_Docr
Use
Select &Sel_Prod
Use
Select &Sel_Tabla
Use
Release ST *
Release Da *
Release Tabla*
Retorna      AA Fin del programa Principal de Inventario [ Money0 ]
```

 Subrutina Orden de Impresi'n
 O R D P

```

Select &Sel_Prod
Use Producto Index Dsc_Prod
Store '0000' to Tipo_Prod
Store 0 to Cantidad
Store Dgte(1) to Fecha
Flag = 0
Do Marquee With "Inventarios", "Orden de Impresi'n"
@ 0, 0 to 4, 75 Double
@ 1, 10 Say "Numero del tipo de producto que desea imprimir" Get Tipo_Prod Picture "9999"
@ 10, 10 Say "Cantidad de productos a imprimir" Get Cantidad Picture "99,999"
@ 13, 10 Say "Fecha de impresi'n" Get Fecha
Read
If Cantidad = 0 .or. Cantidad > 999990
  Use
  Do Msg_Err With DS_NoCht
  Return
EndIf
Seek Right(Tipo_Prod, 3)
If Eof()
  Use
  Do Msg_Err With DS_ProdI
  Return
EndIf
Select Basisn
Set Filter to          !! Para visualizar el #ltimo consecutivo que se
Coto Bottom          !! ha mandado imprimir.
If Eof()
  Cnsc_PinT = Cantidad
  Cnsc_IniT = 1
Else
  Cnsc_PinT = Val(Cnsc_Pin) + Cantidad
  Cnsc_IniT = Val(Cnsc_IniT) + 1
EndIf
Do ZeroI With Cnsc_PinT, ?
Do ZeroI With Cnsc_IniT, ?
Do Si_o_No With Esta_Bien
If Esta_Bien
  Append Blank
  Replace Producto with Left(Tipo_Prod, 1)
  Replace Honeda with SubStr(Tipo_Prod, 2, 2), Denominac with Right(Tipo_Prod, 1), Cnsc_Pin with Cnsc_PinT,
  Feb_Emis with Fecha, Cant_Ini with Cantidad, Existencia with Cantidad
  Documento = Producto + Honeda + Denominac
  DCI = ""
  DCP = ""
  Do Calc_D_C With Documento + Cnsc_IniT, DCI
  Do Calc_D_C With Documento + Cnsc_Pin, DCP
  Flag = 1
  Do Printer
  @ 1, 02 Say "Hora:"
  @ 1, 06 Say "Fin( ) Picture '99:99'"
  @ 1, 22 Say "Interrogational Payment Systems, Inc."
  @ 1, 82 Say "Fecha:"
  @ 1, 69 Say "Date( )"
  @ 2, 1 Say "                Sistema de Ordenes de Pago BANANEI"
  @ 3, 1 Say "                Orden de Impresion"
  @ 4, 1 Say "-----"
  @ 8, 1 Say "Numero de Producto:"
  @ 8, 21 Say Tipo_Prod
  Select Tabla
  Seek Str(Tabla_No,2)+Str(Val(Producto->P_Clv_Mn),4)
  @ 11, 1 Say "Divisa" : " + Descrip
  Seek Str(Tabla_Dn,2)+Str(Val(Producto->P_Dnsc),4)
  @ 13, 1 Say "Documento" +SubStr("          Error con denominacion    con limite    valorable ',,1+1!';
  @ (Left(Descrip,1),1) +SubStr("          DLS",1,17)
  If Left(Descrip,1) < ">
  @ 11, 33 Say "
  @ 13, 34 Say Val(Right(Descrip,29)) Picture "99,999,999.99"
EndIf
@ 17, 1 Say "Numero de Documentos a imprimir:"
@ 17, 33 Say Cantidad Picture "99,999"
Select Basisn
@ 20, 1 Say "Orden de pago inicial" : " + Documento + " + Cnsc_IniT + " + DCP
@ 22, 1 Say "Orden de pago final" : " + Documento + " + Cnsc_PinT + " + DCP
Eject
EndIf
Set Device to Screen
Select Producto
Use
Return          !! Fin [ OKDP ]

```

Rutina de paso a boveda

P A M T

 * Ultima modificacion: 13 de Febrero de 1989

```

Select &Sel_Prod
Use Producto (order Desc_Prod
Tipo_Prod = "0000"
Fecha = CtoD( " / / " )
Do Marques With "Inventarios", "Paso a B'veda"
@ 1, 10 Say "Numero del tipo de producto" Get Tipo_Prod Picture "9999"
Read
If Tipo_Prod = " "
  Use
  Return
EndIf
Seek Right(Tipo_Prod, 3)
Select Emision
Tipo_Prod = Val(Tipo_Prod)
Do Zero1 with Tipo_Prod, 1
Set filter to Producto + Wongda + Denominac = Tipo_Prod .and. Existencia ( 999990
Locate For DtoC(Fch_Prat) = . . .
If Eof()
  Do Msg_Err With DS_HOWF
Else
  Cnsc_IniF = Val(Cnsc_Fin) - Cant_Ini + 1
  Do Zero1 with Cnsc_IniF, 1
  Docuemplo = Producto + Wongda + Denominac
  DCF = ""
  Do Calc_D_C With Documento + Cnsc_IniF, DCF
  Do Calc_D_C With Documento + Cnsc_Fin, DCF
  @6, 1 Clear
  @7, 1 Say "Numero de producto : "
  @7, 21 Say Tipo_Prod
  Select Tabla
  Seek Str(Tabla_No, 2) + Str(Val(Producto->P_Clv_No), 4)
  @8, 1 Say " Divisa: " + Descrip
  Seek Str(Tabla_No, 2) + Str(Val(Producto->P_Dnnc), 6)
  @10, 1 Say " Documento + SubStr( " Error con denominaci'o con limite valorable ", 1:17);
  At(Left((Descrip, 1), DLS ), 1:17)
  If Left(Tabla->Descrip, 1) <> "S"
  @10, 33 Say " : " + &Str(Right(Descrip, 15)) + ".00"
  EndIf
  @12, 1 Say "Numero de documentos contenidos en la Basi'n: "
  Select Emision
  @12, 49 Say Cant_Ini Picture "999,999"
  @14, 1 Say "Orden de Pago Inicial " + Documento + " " + Cnsc_IniF + " - " + DCF
  @15, 1 Say "Orden de Pago Final " + Documento + " " + Cnsc_Fin + " - " + DCF
  Fch_PratF = Date()
  @ 17, 1 Say "Fecha de Impresi'n: " Get Fch_PratF
  Read
  Esta_Bien = .F.
  Do Si_O_No With Esta_Bien
  If Esta_Bien
    Replace Fch_Prat with Fch_PratF
  EndIf
EndIf
Select Producto
Use
Return AA Fin ( PRMT )

```

 Rutina de Distribucion a los Agentes
 S R I P

 * Ultima modificaci'n: 13 de Febrero de 1989

```

Select &Sel_Ag
Use Agentes Index Agentes
Store 0 to Clave_AgT
Store date() to Fch_ShipT
Store 0 to EnvioT
Store "000000" to Cnsc_PinT
Store "0000" to Tipo_Prod
Do Margues With "Inventarios", "Distribuci'n de Ordenes de Pago"
@ 1, 10 Say Clave del Agente: "99999"
@ 1, 39 Get Clave_AgT Picture "99999"
@ 1, 55 Say "Tipo de Producto: " Get Tipo_Prod Picture "9999"
@ 3, 10 SAY "Cantidad de Ordenes de Pago:"
@ 3, 39 Get EnvioT Picture "9,999"
@ 13, 10 Say "Fecha de Distribuci'n:"
@ 13, 46 Get Fch_ShipT
Read
Do While EnvioT > 9990
@ 17, 10 Say "La cantidad debe ser menor a 9,990"
@ 3, 39 Get EnvioT Picture "9,999"
Read
EndDo
Seek Clave_AgT
If EOF()
Use
Do Msg_Err With DS_AGNP
Do Deseo With Esta_Bien
Return
Endif
If Stat_Ag = 2
Use
Do Msg_Err With DS_AgoBlk
Do Deseo With Esta_Bien
Return
Endif
If Stat_Ag = 4
Use
Do Msg_Err With DS_AgoCal
Do Deseo With Esta_bien
Return
Endif
Use
Select Emision
Set Filter to Producto + Moneda + Denominac = Tipo_Prod
Locate For Existencia > 0 .and. Existencia <= 999990 .and. DtoC(Fch_Prat) <> " . . ."
If EOF() .or. EnvioT = 0
Do Msg_Err With DS_MONP
Do Deseo With Esta_Bien
Return
Endif
If Existencia < EnvioT
@ 23, 1 Clear
? Chr(1)
@ 20, 1 Say " No es posible obtener de b'veda un paquete del tamaño requerido"
@ 21, 1 Say " Los datos expuestos corresponden a un paquete propuesto"
EnvioT = Existencia
Endif
Cnsc_Init = Val(Cnsc_Pin) - Existencia + 1
Cnsc_PinTT = Cnsc_Init + EnvioT - 1
Do Zerol With Cnsc_Init, 1
Do Zerol With Cnsc_PinTT, 1
@ 7, 39 Say Clave_AgT Picture "99999"
@ 9, 39 Say EnvioT Picture "9,999"
@ 11, 10 Say "Consecutivo inicial:"
@ 11, 46 Say Cnsc_Init
@ 13, 10 Say "Consecutivo final:"
@ 13, 46 Say Cnsc_PinTT
@ 15, 46 Say Fch_ShipT
If Cnsc_PinT < Cnsc_PinTT .and. Val(Cnsc_PinT) <> 0
@ 13, 10 SAY "Consecutivo Final Propuesto:"
@ 13, 46 Say Cnsc_PinT Picture "9999999"
Do Msg_Err With Dg_NCosc
Do Deseo With Esta_bien
Return
Endif
Esta_Bien = .P.
Do Si_o_No With Esta_Bien
?? P.-. etc..
  
```

```

** cond_picu
Replace Existencia With Existencia - EnvioT
Set Filter to
Select Dstr
Append Blank
Replace Producto with Left(Tipo_Prod, 1)
Replace Clave_Ag with Clave_AgT, Moneda With SubStr(Tipo_Prod, 2, 2), Denominac With Right(Tipo_Prod, 1);
Cnsc_Pin with Cnsc_PinT, Cant_Ini with EnvioT, Existencia with EnvioT, Pch_Ship with Pch_ShipT
Reco = RecNo()
Select Tempo
Append Blank
Replace Dstr_Ap with Reco
Endif
Do Deseo With Esta_Bien
Return && Pin [ 'SHIP' ]

```

 * Rutina de Confirmacion en Localidad de Venta
 * S L O C
 *-----

*-----
 * Ultima modificaci"n: 13 de Febrero de 1989
 *-----

```

Store 0 to Clave_AgT
Store "0000000" to Cnsc_T
Store "0000" to Tipo_Prod
Do Marques With "Inventarios", "Confirmaci"n en Localidad de Venta"
@ 7, 10 Say "Clave del Agente:"
@ 7, 39 Get Clave_AgT Picture "99999"
@ 7, 55 Say "Tipo de Producto: Get Tipo_Prod Picture "9999"
@ 10, 10 Say "Consecutivo cua|quiera contenido"
@ 10, 46 Get Cnsc_T Picture "9999999"
Read
Select Dstr
Set Filter to Tipo_Prod = Producto + Moneda + Denominac .and. Existencia < 9990
Cnsc_T = Val( Cnsc_T )
Do Zerol With Cnsc_T, 7
Set Order to
Do GetGE With Cnsc_T
Set order to I
If Rof()
  Do Mag_Brr With DS_MOMP
  Do Deseo With Esta_Bien
  Return
EndIf
If Clave_AgT <> Clave_Ag
  Do Mag_Brr With DS_AGWCHK
  Do Deseo With Esta_Bien
  Return
EndIf
Cnsc_T = Val(Cnsc_Fin) - Cant_Ini + 1
Do Zerol With Cnsc_T, 7
@ 18, 10 Say "Cantidad de Ordenes de Pago"
@ 18, 44 Say "Cant_Ini Picture "9,999"
@ 10, 10 Say "Consecutivo inicial"
@ 10, 35 Say "Cnsc_T"
@ 12, 16 Say "Consecutivo final"
@ 12, 35 Say "Cnsc_Fin"
@ 14, 10 Say "Fecha de Distribuci"n"
@ 14, 39 Say "Fch_Ship"
@ 16, 10 Say "Fecha de Localidad de venta"
If DtoD( Fch_Sloc ) <> . . .
  @ 16, 39 Say "Fch_Sloc"
  Do Mag_Brr With DS_RSHIP
  Do Deseo With Esta_bien
  Return
EndIf
Replace Fch_Sloc with Date{}
@ 16, 39 Get Fch_Sloc
Read
Do Si_o_No With Esta_Bien
If .NOT. Esta_Bien
  Replace Fch_Sloc With CtoD(" . . . ")
EndIf
Do Deseo With Esta_bien
Return ** Fin [ SLOC ]

```

Ventas Manuales
V E N T A S

* Rutina para generar un archivo el cual contenga Ordenes de Pago vendidas.

```

@ 1, 1 Clear
Flag = 0
Do Barques with "Procesamiento de Informaci"n,"Captura de Ventas Manuales"
ArchS =
@ 11, 10 Say "Teclea el Nombre del Archivo de Ventas " Get ArchS
Read
If ArchS = "
Return
EndIf
Num_Franq = 0
Tot_Vnt = 0
Num_Vnd = 0
Num_Anl = 0
Tot_CIPS = 0
Tot_CAg = 0
Chance = "
Select &Sel_Prod
Use Vnt_Dia index Vnt_Dia
Zap
Append Blank
Do While Chance <> "N"
@ 6, 1 Clear
Chance = "
Correcto = "
Clave_AgT = 0
Comis_IPST = 0
Comis_AgT = 0
Prod_T = "0000"
Cnsc_T = "0000000"
Dig_T = "0"
FechaT = Da[e]
HoraT = "00"
MinT = "00"
Tipo_CabT = 0
DolarT = 0
PesosT = 0
Clave_Opt = 0
Status_T = "
@ 7, 7 Say "N"mero de la Orden de Pago: "
@ 7, 36 Get Prod_T
@ 7, 41 Get Cnsc_T
@ 7, 49 Get Dig_T
@ 9, 7 Say "Clave del Agente"
@ 9, 24 Get Clave_AgT
@ 11, 7 Say "Fecha de la Venta," Get FechaT
@ 11, 40 Say "Hora de la Venta"
@ 11, 59 Get HoraT
@ 11, 61 Say ":"
@ 11, 62 Get MinT
@ 13, 7 Say "Valor (en d"lares) $"
@ 13, 27 Get DolarT
@ 13, 42 Say "Valor (en pesos) $"
@ 13, 59 Get PesosT
@ 15, 7 Say "Comision IPS "
@ 15, 22 Get Comis_IPST
@ 15, 35 Say "Comision Agente "
@ 15, 53 Get Comis_AgT
@ 17, 7 Say "Operador"
@ 17, 24 Get Clave_Opt
@ 17, 35 Say "Status: { (S)old o (V)oid } "
@ 19, 7 Say "Tipo de cambio (pesos por dolar) $"
@ 19, 41 Get Tipo_CabT
Read
Do While Status_T <> "S" .and. Status_T <> "V"
@ 17, 65 Get Status_T
Status_T = Upper(Status_T)
EndDo
Do While Correcto <> "S" .and. Correcto <> "N"
@ 23, 10 Say "Estan correctos los Datos? (S/N) "
@ 23, 45 Get Correcto
Correcto = Upper(Correcto)
EndDo
If Correcto <> "N"
Select Vnt_Dia
Append Blank
Fch = DtoC(FechaT)
Replace Clave Ag With Str(Clave_AgT.5), No,No With Prod_T + ;

```

AA Picture "9999"

AA Picture "99999999"

AA Picture "9"

Picture "99999"

AA Picture "99"

AA Picture "99"

Picture "9,999,999.99"

Picture "9,999,999.99"

Picture "999.99"

Picture "999.99"

Picture "99"

Picture "99,999.99"

Picture "99,999.99"

Picture "1"

Picture "1"

```

Cosc_T + Dig_T, Pch_Vnt With Left(Pch,2) + SubStr(Pch,4,2) + Right(Pch,2)
Replace Hora_Vnt With HoraF+MinF, Clave_Op With Str(Clave_Opt,2),;
Transac With Left(Status_T,1)
If Status_T = "S"
  Replace Valor_Vnt With Str(DolarF*100,9), Cois_ Ips With Str(Cois_ IpsT * 100, 8);
  Cois_Ag With Str(Cois_Agt * 100, 8), Valor_Mo_0 With Str(PesosF*100,11);;
  Tipo_Cambio With Str(Tipo_CabT*100,7)
  Num_Vnd = Num_Vnd + 1
Else
  Replace Valor_Vnt With "000000000", Cois_ Ips With "000000000";;
  Cois_Ag With "000000000", Valor_Mo_0 With "000000000000", Tipo_Cambio With "00000000"
  Num_Anl = Num_Anl + 1
  DolarF = 0
EndIf
Num_Trans = Num_Trans+1
Tot_Vnt = Tot_Vnt + DolarF * 100
Tot_CIPS = Tot_CIPS + Val(Cois_ Ips)
Tot_CAg = Tot_CAg + Val(Cois_Ag)
EndIf

Correcto = " "
@ 23, 1 Clear
@ 23, 16 Say "Deseas realizar otra Venta? (S/N) "
Chance = " "
Do While Chance (<) "S" .and. Chance (<) "N"
  @ 23, 51 Get Chance
  Chance = Upper(Chance)
  Beep
EndDo
EndDo
Goto Top
* Graba el Header del archivo.
Replace Clave_Ag With Str(Num_Trans,5), No_Mo With Str(Tot_Vnt,12), Pch_Vnt With Str(Num_Vnd,6),;
Hora_Vnt With Str(Num_Anl,4), Valor_Vnt With Right(Str(Tot_Cips),9), Cois_ Ips With Right(Str(Tot_Cag),8)
Copy to &&Arch5 Type SDF
Select Vnt_Dia
Use
Return ** FIN PROGRAMA DE PRUEBA | VENTAS ]

```

 * Transmision de datos a Mexico
 * TRAN_MEX

Restore From St_Tabla Additive

ArchT =

Do Marques With "Procesamiento de Informaci"n, "Transmis"n a Mexico"

@ 8, 10 Say "Teclca el Nombre del Archivo de Transmis"n" Get ArchT Picture "8!"

Head

If ArchT (< " "

Num_Trans = 0

Vt = 0

BVt = 0

NVt = 0

Reg = 0

Tot_Dlls = 0

Select &Sel_Prod

Use Tran_Mex Index Tran_Mex

Zap

Append Blank

Select No

Set Filter to DtoC(Fcb_Trans) = " . . . "

Go Top

@ 07, 01 Clear

@ 10, 25 Say "Procesando el Registro: "

@ 20, 31 Say "Espera un momento"

Do While .Not. Eof()

Reg = Reg + 1

@ 10, 49 Say Reg Picture "9,999"

D_C =

Do Calc D_C with Tipo_Prod + Cnsc_No, D_C

Select Tran_mex

Append Blank

Valor = Right(Str(No->Valor_No_0 * 100 + 10^11, 12), 11)

Dlls = Right(Str(No->Valor_Vta * 100 + 10^9, 10), 9)

Tipo_C = Right(Str(No->Tipo_Camb * 100 + 10^7, 8), 7)

If No->Status = "V"

StSold = Right(Str(St_Void + 10^2), 2) aa Status 44

Valor = "000000000000"

Else

StSold = Right(Str(St_Sold + 10^2), 2) aa Status 50

EndIf

Vt = Vt + 1

Replace Tipo_Reg With "A", Cve_Mov With "01", Opb With No->Tipo_Prod + No->Cnsc_No + D_C,;

Status With StSold, Importe With Valor, Fecha With No->Fcb_Vta,;

Imp_Dlls With Dlls, Tipo_Camb With Tipo_C

Tot_Dlls = Tot_Dlls + No->Valor_Vta

Num_Trans = Num_Trans + 1

Select No

Replace Fcb_Trans With Date()

Skip

EndDo

Set Filter to

Select Ciclo_Vi

Set Filter to DtoC(Fcb_1Tr) = "And. DtoC(Fcb_1St) (< " . . . "

Go Top

Do While .Not. Eof()

Ok = .T.

Baja = .F.

Rango = 0

Reg = Reg + 1

@ 10, 49 Say Reg Picture "9,999"

Do Case

Case Clve_Stat = "EM"

Vt = VE + 1

Select Emission

Set Filter to Producto + Moneda + Denominac = Ciclo_Vi->Tipo_Prod

Do GetGE with Ciclo_Vi->Cnsc_No

Rango = Capt_1ai

Tipo = "A"

Clave = "01"

Do Case

Case Existencia = 999999

Num_Stat = "St_VDP"

Case Existencia = 999998

Num_Stat = "St_STP"

Case Existencia = 999997

Num_Stat = "St_LSP"

Case Existencia = 999996

Num_Stat = "St_PMT"

Otherwise

@ 7, 1 Clear

Do Incons With "TRAN_MEX 1"

EndCase

Case Clve_Stat = "VB" .OR. Clve_Stat = "LB" .OR. Clve_Stat = "SB"

```

*.. - 01"
Baja = .T.
Select Dstr
Set Filter to Producto + Moneda + Denominac = Ciclo_Vi->Tipo_Prod
Do GetDB With Ciclo_Vi->Canc_No
Rango = Cant_Ini
Tipo = "0"
Clave = "01"
Do Case
  Case Ciclo_vi->Clve_Stat = "VB"
    Num_Stat = "St_VDPA"
  Case Ciclo_vi->Clve_Stat = "LB"
    Num_Stat = "St_SVPA"
  Case Ciclo_vi->Clve_Stat = "SB"
    Num_Stat = "St_LSPA"
EndCase
Case Clve_Stat = "DS"
  Vt = Vt + 1
  Select Dstr
  Set Filter to Producto + Moneda + Denominac = Ciclo_Vi->Tipo_Prod
  Set Order to
  Do GetDB With Ciclo_Vi->Canc_No
  Set Order to 1
  Rango = Cant_Ini
  Tipo = "A"
  Clave = "01"
  If DtoC(Pch_Sloc) = " . . ."
    Do Case
      Case Existencia = 9999
        Num_Stat = "St_VDPA"
      Case Existencia = 9998
        Num_Stat = "St_SVPA"
      Case Existencia = 9997
        Num_Stat = "St_LSPA"
      Case Existencia = 9996
        Num_Stat = "St_SGIP"
      Otherwise
        @ 1, 1 Clear
        Do Incons With "TRAN_MEX 2"
    EndCase
  Else
    Do Case
      Case Existencia = 9999
        Num_Stat = "St_VDSL"
      Case Existencia = 9998
        Num_Stat = "St_SVSL"
      Case Existencia = 9997
        Num_Stat = "St_LSSL"
      Case Existencia = 9996
        Num_Stat = "St_SLOC"
      Otherwise
        @ 1, 1 Clear
        Do Incons With "TRAN_MEX 3"
    EndCase
  EndIf
Case Clve_Stat = "VD" .OR. Clve_Stat = "LD" .OR. Clve_Stat = "SD" .OR. Clve_Stat = "VL" .OR.;
  Clve_Stat = "LL" .OR. Clve_Stat = "SL"
  Bvt = Bvt + 1
  Baja = .T.
  Select Dstr
  Set Filter to Producto + Moneda + Denominac = Ciclo_Vi->Tipo_Prod
  Set Order to
  Do GetDB With Ciclo_Vi->Canc_No
  Set Order to 1
  Rango = Cant_Ini
  Tipo = "0"
  Clave = "01"
  If DtoC(Pch_Sloc) = " . . ."
    Do Case
      Case Ciclo_vi->Clve_Stat = "VD"
        Num_Stat = "St_VDSH"
      Case Ciclo_vi->Clve_Stat = "LD"
        Num_Stat = "St_SVSH"
      Case Ciclo_vi->Clve_Stat = "SD"
        Num_Stat = "St_LSSH"
    EndCase
  Else
    Do Case
      Case Ciclo_vi->Clve_Stat = "VL"
        Num_Stat = "St_VDSL"
      Case Ciclo_vi->Clve_Stat = "LL"
        Num_Stat = "St_SVSL"
      Case Ciclo_vi->Clve_Stat = "SL"
        Num_Stat = "St_LSSL"
    EndCase
  EndIf
Case Clve_Stat = "BS"
  Bvt = Bvt + 1
*..

```

```

*...
Rango = 0
Tipo = "B"
Clave = "02"
Num Stat = "ST SOLD"
Case Clve_Stat = "BV"
  Bvt = Bvt + 1
  Baja = .T.
  Rango = 0
  Tipo = "B"
  Clave = "03"
  Num Stat = "ST VOID"
  Case Clve_Stat = "S"
    Nvt = Nvt + 1
    Rango = 0
    Tipo = "M"
    Clave = "03"
    Num Stat = "ST STLN"
    Case Clve_Stat = "L"
      Nvt = Nvt + 1
      Rango = 0
      Tipo = "M"
      Clave = "03"
      Num Stat = "ST LOST"
      Case Clve_Stat = "C"
        Nvt = Nvt + 1
        Rango = 0
        Tipo = "M"
        Clave = "03"
        Num Stat = "ST CWCL"
        Case Clve_Stat = "SV"
          Nvt = Nvt + 1
          Baja = .T.
          Rango = 0
          Tipo = "M"
          Clave = "01"
          Num Stat = "ST VOID"
          Case Clve_Stat = "VS"
            Nvt = Nvt + 1
            Baja = .T.
            Tipo = "M"
            Clave = "02"
            Num Stat = "ST SOLD"
            Select No
            Seek Ciclo_Vi->Casc_No
            If Bof()
              Ok = .F.
              Do Incons With "TRAN_MEX 4"
            Else
              Rango = No->Valor_No_0 * 100
              Baja = .T.
            Endif
          Case Clve_Stat = "QS" .Or. Clve_Stat = "QL" .Or. Clve_Stat = "QC"
            Nvt = Nvt + 1
            Baja = .T.
            Rango = 0
            Tipo = "M"
            Clave = "04"
            Num Stat = "ST SOLD"
            Case Clve_Stat = "NI"
              Nvt = Nvt + 1
              Tipo = "M"
              Clave = "05"
              Baja = .T.
              Num Stat = "ST SOLD"
              Select No
              Seek Ciclo_Vi->Casc_No
              If Bof()
                Ok = .F.
                Do Incons With "TRAN_MEX 4"
              Else
                Rango = No->Valor_No_0 * 100
                Baja = .T.
              Endif
            EndCase
          EndCase
        EndCase
      EndCase
    EndCase
  EndCase
EndCase
If Ok
  Select Ciclo_Vi
  D_C =
  Do Calc_D_C with Tipo_Prod + Casc_No, D_C
  Select tran_Mex
  Append Blank
  Rango = Right(Str(Rango + 10^11, 12), 11)
  Wa_Stat = Right(Str(ANum_Stat + 10^21, 2), 2)
  Replace Tipo_Mex With Tipo, Cve_Mor With Clave, Opb With Ciclo_Vi->Tipo_Prod + Ciclo_Vi->Casc_No + D_C.;
  Replace Importe With Rango, Status With Wa_Stat, Fecha With Ciclo_Vi->Fch_IST
  Num_Trans = Num_Trans + 1
  Select Ciclo_Vi
  If Baja
    ...

```



```

        Else
            Replace Pch_1Tr With Date()
        Endif
        Skip
    Else
        Select Ciclo_Vi
        Skip
    Endif
EndDo
Pack
Set Filter to
* Busca registros de un segundo Stop_Payment para un N.O. individual.
Set Filter to DtoC(Fch_2St) (<) . . . .And. DtoC(Fch_2Tr) = . . . .
Go Top
Do While .Not. EOF()
    Reg = Reg + 1
    @ 10, 19 Say Reg Picture "9,999"
    Num_Stat = "ST_" + SubStr("STLN LOST CNCL",-4 + 5 * At( Left( Clve_Stat, 1), "SLC"),4)
    D_C =
    Do Calc D_C with Tipo_Prod + Casc_Mo, D_C
    NVT = NVT + 1
    Select Tran_Mex
    Append Blank
    Num_Stat = Right(Str(Num_Stat + 10^2), 2)
    Replace Opb With Ciclo_Vi->Tipo_Prod + Ciclo_Vi->Casc_Mo + D_C, Status With Num_Stat, Importe With "0000000000";
        Fecha With Ciclo_Vi->Fch_2St, Tipo_Reg With "R", Cve_Mov With "03"
    Num_Trans = Num_Trans + 1
    NVT = NVT + 1
    Select Ciclo_Vi
    Replace Fch_2Tr With Date()
    Replace Fch_1Tr With Date()
    Skip
EndDo
* Crea el Header del Archivo
Select Tran_Mex
Go Top
Num_Trans = Right(Str(Num_Trans + 10^7, 8), 7) + Right(Str(NVT + 10^5, 6), 5)
Vt_NVT = Right(Str(Vt + 10^5, 6), 5) + "." + Right(Str(NVT + 10^5, 6), 5)
Dolares = "0000" + Right(Str(Tot_Dlls * 100 + 10^-12, 13), 12)
ImpoDlls = Left(Dolares, 9)
T_Cambio = Right(Dolares, 7)
Replace Tipo_Reg With "R", Cve_Mov With "00", Opb With Num_Trans, Status With "00", Importe With Vt_NVT;
    Fecha With Date(), Imp_Dlls With ImpoDlls, Tipo_Camb With T_Cambio
Copy to &&ArchT Type SDP
Go Top
Do Rep_Tran
Use
ArchT = " "
* El archivo &&ArchT contiene las Ordenes de Pago vendidas y la Ord. Pago que est a en Stop Payment
Endif
Return && | Fin de Tran_Mex |

```

 * Lectura de Datos Provenientes de Mxico
 * L E C T U R A
 * -----

```

Restore from St_Table Additive
Arch_Mex =
No_Dscr = 0
Do Marques With "Procesamiento de Informaci"n, "Recepci"n de Datos de Mxico"
Select ASel_Dscr
Use Dscr_Tap
Zap
Select ASel_Prod
Use Lect_Mex
Zap
Esta_Bien = .F.
On Error Esta_Bien = .F.
@ 7, 1
Do While .Not. Esta_Bien
  Esta_Bien = .T.
  @ 8, 10 Say "Teclea el Archivo a Procesar : " Get Arch_Mex
  Read
  If Arch_Mex = " "
    Return
  EndIf
  Append From &Arch_Mex Type SDF
  If .Not. Esta_Bien
    Do Msg_Err With Ds_PWD
    Do Deseo With Esta_Bien
    If .Not. Esta_Bien
      Return
    EndIf
    Select Lect_Mex
    Esta_Bien = .F.
  EndIf
  Arch_Mex = Arch_Mex
  Arch_Mex =
  @ 7, 1 Clear
EndDo
On Error
Go Bottom
If EOF()
  @ 12, 28 Say "El Archivo est vacio"
  @ 22, * 1
  Wait *
  Use
  Return
EndIf
* Se lee el Trailer
No_Rec = Aut_Pago    AA Numero de registros del Archivo.
Tot_Import = Importe    AA Suma de importes del archivo.
Ultimo_Reg = RecNo()    AA Ultimo registro ffsico.
Id_Trailer = Clv_Rchz    AA Identificador de registro trailer.
*
Fecha_Proc = CtoD(Left(Pch_Pago, 2) + "." + Substr(Pch_Pago, 3, 2) + "." + Right(Pch_Pago, 2))
Go Top
Sum_Importe to Sum_Importe for RecNo() (<) Ultimo_Reg
If No_Rec (<) RecCount() - 1 .or. Sum_Importe (<) Tot_Import .or. [d_Trailer (<) 9
  @ 12, 23 Say "Error en las Cifras de Control del Trailer."
  @ 22, * 1 Clear
  Wait *
  Use
  Return
EndIf
* Se lee la Informaci"n del Archivo INCOMING
Go Top
Num_Rec = 1
@ 7, 1 Clear
@ 10, 25 Say "Procesando el Registro: "
@ 20, 31 Say "Espera un momento"
Do While RecNo() (<= No_Rec
  Lectura = .T.
  Registro = RecNo()
  @ 10, 49 Say Registro    Picture "9,999"
  Clv_Stat =
  Es_Buena = .T.
  If Clv_Rchz = 0
    Pch_Rc = "Pch |Rc"
    Pch_Tr = "Mo->Pch_Trans"
  Do Mast2_Po
  Select Lect_Mex
  EndIf
  If Es_Buena
    Do Case
      Case Clv_Rchz = 0            AA Registro de P.O. Pagado.
        * Se actualizan log_archivos.

```

```

If @My_Stat = "0000"
Select No
Replace Fch_Recep With Fecha_Proc
Replace Fch_Pago With CtoD[Left[LECT_MEX->Fch_Pago,2]]+"*"+Subst[LECT_MEX->Fch_Pago,3,2]+"<";
Right[LECT_MEX->Fch_Pago,2]]
Replace Status With Subst["P" + Status,1,3], Suc_Prom With Lect_MEX->Suc_Prom
Replace Suc_Oper With Lect_MEX->Suc_Oper, Aut_Pago With Lect_MEX->Aut_Pago, Fch_Rcp_Pg With Date[ ]
Replace Fch_RegP With CtoD[Left[LECT_MEX->Fch_RegP,2]]+"*"+Subst[LECT_MEX->Fch_RegP,3,2]+"<";
Right[LECT_MEX->Fch_RegP,2]]

Else
Select Ciclo_Vi      AA P.O. Pagado en Prevencion de Pago.
If .Not. Bof[ ]
Replace A_Fch_Rc With Fecha_Proc,
Flang_CofI With "A"
Select No
Replace Fch_Recep With Fecha_Proc
Replace Fch_Pago With CtoD[Left[LECT_MEX->Fch_Pago,2]]+"*"+Subst[LECT_MEX->Fch_Pago,3,2]+"<";
Right[LECT_MEX->Fch_Pago,2]]
Replace Status With Subst["P" + Status,1,3], Suc_Prom With Lect_MEX->Suc_Prom
Replace Suc_Oper With Lect_MEX->Suc_Oper, Aut_Pago With Lect_MEX->Aut_Pago, Fch_Rcp_Pg With Date[ ]
Replace Fch_RegP With CtoD[Left[LECT_MEX->Fch_RegP,2]]+"*"+Subst[LECT_MEX->Fch_RegP,3,2]+"<";
Right[LECT_MEX->Fch_RegP,2]]

Else
Do Docr_Mex With Dc_MCV
EndIf
EndIf
EndIf
Case Clv_Rchz = 1 AA Error en Fecha de Pago del Sistema On-Line.
Do Docr_Mex With Ds_Rchz1
Case Clv_Rchz = 2 AA La Orden de Pago no existe en el Archivo.
Do Docr_Mex With Ds_Rchz2
Case Clv_Rchz = 3 AA Status inv lido.
Do Docr_Mex With Ds_Rchz3
Case Clv_Rchz = 4 AA Importe inv lido.
Do Docr_Mex With Ds_Rchz4
Case Clv_Rchz = 5 AA Docto. autorizado previamente.
Do Docr_Mex With Ds_Rchz5
Case Clv_Rchz = 10 AA Duplicaci'n de Alta en MONETORD.
Do Docr_Mex With Ds_Rchz10
Case Clv_Rchz = 11 AA P.O. ya ha sido pagado.
Do Docr_Mex With Ds_Rchz11
Case Clv_Rchz = 12 AA Stop Payment de un Docto. que ya ha sido pagado.
Do Docr_Mex With Ds_Rchz12
Case Clv_Rchz = 13 AA Stop Payment de un Docto. que no est como vendido.
Do Docr_Mex With Ds_Rchz13
Case Clv_Rchz = 14 AA Docto. que no se puede poner en Stop_Payment.
Do Docr_Mex With Ds_Rchz14
Case Clv_Rchz = 15 AA Transmisi'n de un registro con Status inv lido.
Do Docr_Mex With Ds_Rchz15
Case Clv_Rchz = 16 AA Error en el D'gito de Control.
Do Docr_Mex With Ds_Rchz16
Case Clv_Rchz = 17 AA Importe omitido en la Venta.
Do Docr_Mex With Ds_Rchz17
Case Clv_Rchz = 18 AA Tipo de registro invalido.
Do Docr_Mex With Ds_Rchz18
Case Clv_Rchz = 19 AA Clave de movimiento invalido.
Do Docr_Mex With Ds_Rchz19
Case Clv_Rchz = 20 AA No se puede borrar el registro.
Do Docr_Mex With Ds_Rchz20
OtherWise
Do Docr_Mex With Ds_WrChI
EndCase
EndIf
Select Lect_Mex
If .Not. Lectura
Delete
EndIf
Skip
EndDo
Select Lect_Mex
Pack
Go Top
Do Rep_Lect
Use
Do Rep_Del
Select ASel_Docr
Use Docr_Mex
Select Docr_Mex
Append From DcMx_Tap
@ 7, 1 Clear
@ 13, 32 Say "Fin del Proceso"
@ 22, 1 Clear
Wait
Return AA [ Fin Lect_Mex ]
Preiona cualquier Tecla para continuar"

```

 Rutina de Prevenciones por Paquetes
 B O V R

```

Store "0" to ProductoP
Store "00" to MonedaP
Store "0" to DenominaP
Store "0000000" to CuscP
Store "0" to D_CP
Store "0" to ProductoS
Store "00" to MonedaS
Store "0" to DenominaS
Store "0000000" to CuscS
Store "0" to D_CS
Store " " to Cusc
Store 0 to Cusc_FinL
Store 0 to TiraJe_L
Store 0 to Exist_L
Store 0 to Cusc_FinA
Store 0 to TiraJe_A
Store 0 to Exist_A
Store 0 to Cusc_FinP
Store 0 to TiraJe_P
Store 0 to Exist_P
Store Date{} to Fecha
Fecha = DtoC( Fecha )
Store Space(80) To Causal
Store Space(60) To Causal2
Do Case
  CaseCodigo = 999997
  Do Marques With "Prevenciones en B'veda", "Ordenes de Pago Perdidas"
  CaseCodigo = 999998
  Do Marques With "Prevenciones en B'veda", "Ordenes de Pago Robadas"
  CaseCodigo = 999999
  Do Marques With "Prevenciones en B'veda", "Ordenes de Pago Anuladas"
EndCase
@ 7, 17 Say "Rango Inicial:"
@ 7, 45 Get ProductoP Picture "g"
@ 7, 46 Get MonedaP Picture "99"
@ 7, 48 Get DenominaP Picture "g"
@ 7, 50 Get CuscP Picture "9999999"
@ 7, 51 Say " "
@ 7, 58 Get D_CP Picture "g"
@ 10, 17 Say "Rango Final:"
@ 10, 45 Get ProductoS Picture "g"
@ 10, 46 Get MonedaS Picture "99"
@ 10, 48 Get DenominaS Picture "g"
@ 10, 50 Get CuscS Picture "9999999"
@ 10, 51 Say " "
@ 10, 58 Get D_CS Picture "g"
@ 14, 3 SAT "Observaciones:"
@ 16, 8 GET CAUSAL
@ 17, 8 GET CAUSAL2
Read
Esta_Bien = ".P."
Clv_Stat = " "
D_CT = " "
Do Calc_D_C With ProductoP + MonedaP + DenominaP + CuscP, D_CT
If D_CT < 5 D_CP
  Do Msg_Err With Ds_DCI
  Do Desno With Esta_Bien
  Return
EndIf
Do Calc_D_C With ProductoS + MonedaS + DenominaS + CuscS, D_CT
If D_CT < 5 D_CS
  Do Msg_Err With Ds_DCI
  Do Desno With Esta_Bien
  Return
EndIf
Do Rastr_Mo With ProductoS + MonedaS + DenominaS + CuscS + D_CS, Clv_Stat, "Msg_Err", Esta_Bien
If .Not. Esta_Bien
  Do Desno With Esta_Bien
  Return
EndIf
Mo1 = ProductoP + MonedaP + DenominaP
Mo2 = ProductoS + MonedaS + DenominaS
Select Eximion
If (Mo1 < Mo2) .or. Val(Cusc_Fin) - Cant_Ini + 1 > Val(CuscP) .or. CuscS < CuscP
  Do Msg_Err With Ds_Englnv
  Do Desno With Esta_Bien
  Return
EndIf
If Clv_Stat <> "PRNT" .Or. Val(Cusc_Fin) - Existencia + 1 > Val(CuscP)
  Do Msg_Err With Ds_Sltar
  Do Desno With Esta_Bien

```

```

Return
EndIf
Cnsc_FinL = Val(Cnsc_Fin)
Tiraje_L = Val(Cnsc_Fin) - Val(CnscS)
Exist_L = Tiraje_L
Cnsc_FinA = Val(CnscS)
Tiraje_A = Val(CnscS) - Val(CnscP) + 1
Exist_A = Codigo
Cnsc_FinF = Val(CnscP) - 1
Tiraje_F = Val(CnscP) - (Val(Cnsc_Fin) - Cant_Ini) - 1
Exist_F = Tiraje_F - Cant_Ini + Existencia
FchB = Fch_Bais
FchP = Fch_Prot
Do ZeroI with Cnsc_FinA, ?
Select Ciclo Vi
Do Si_o_No With Bata_Bien
If .not. Bata_Bien
Return
EndIf
Append Blank
Replace Tipo_Prod With ProductoP + MonedaP + DenominaP, Cnsc_Mo with Cnsc_FinA, Fch_Inst with Date(,);
Clve_Stat with "EM", Causa_St1 with Causal, Causa_St2 with Causa2
Select Baision
Replace Cnsc_Fin with Cnsc_FinA, Cant_Ini with Tiraje_A, Existencia with Exist_A
If Tiraje_F > 0
Do ZeroI with Cnsc_FinF, ?
Append Blank
Replace Producto with ProductoS, Moneda with MonedaS, DenominaC with DenominaS, Cnsc_Fin with Cnsc_FinF,;
Fch_Bais with FchB, Cant_Ini with Tiraje_F, Existencia with Exist_F, Fch_Prot with FchP
EndIf
If Tiraje_L > 0
Do ZeroI with Cnsc_FinL, ?
Append Blank
Replace Producto with ProductoS, Moneda with MonedaS, DenominaC with DenominaS, Cnsc_Fin with Cnsc_FinL,;
Fch_Bais with FchB, Cant_Ini with Tiraje_L, Existencia with Exist_L, Fch_Prot with FchP
EndIf
Set Filter to
Sort on Cnsc_Fin to Temporal
Use
Erase Baision.dbf
Rename Temporal.Dbfc to Baision.Dbfc
Use Baision
Bata_Bien = .P.
Return AA Fin { BOVB }

```

 Rutina de Previsiones por Paquetes
 D S T E

```

Store "0" to ProductoP
Store "00" to MonedaP
Store "0" to DenominaP
Store "0000000" to CnscP
Store "0" to D_CP
Store "0" to ProductoS
Store "00" to MonedaS
Store "0" to DenominaS
Store "0000000" to CnscS
Store "0" to D_CS
Store "0" to Cnsc
Store 0 to Cnsc.PigL
Store 0 to TiraJe_L
Store 0 to Exist_G
Store 0 to Cnsc.PinA
Store 0 to TiraJe_A
Store 0 to Exist_A
Store 0 to Cnsc.PigP
Store 0 to TiraJe_P
Store 0 to Exist_F
Store Date{ } to Fecha
Fecha = DtoC( Fecha )
Store Space{50} To Causa1
Store Space{50} To Causa2
If Bandera
  Do Case
    Case Codigo = 9996
      Do Marques With "Previsiones en Distribuci"n", "Ordenes de Pago en Poder de IPS"
    Case Codigo = 9997
      Do Marques With "Previsiones en Distribuci"n", "Ordenes de Pago Perdidas"
    Case Codigo = 9998
      Do Marques With "Previsiones en Distribuci"n", "Ordenes de Pago Robadas"
    Case Codigo = 9999
      Do Marques With "Previsiones en Distribuci"n", "Ordenes de Pago Anuladas"
  EndCase
Else
  Do Case
    Case Codigo = 9996
      Do Marques With "Previsiones en Loc. de Venta", "Ordenes de Pago en Poder de IPS"
    Case Codigo = 9997
      Do Marques With "Previsiones en Loc. de Venta", "Ordenes de Pago Perdidas"
    Case Codigo = 9998
      Do Marques With "Previsiones en Loc. de Venta", "Ordenes de Pago Robadas"
    Case Codigo = 9999
      Do Marques With "Previsiones en Loc. de Venta", "Ordenes de Pago Anuladas"
  EndCase
EndIf
e 7, 17 Say "Rango Inicial:"
e 7, 45 Get ProductoP Picture "9"
e 7, 46 Get MonedaP Picture "99"
e 7, 48 Get DenominaP Picture "9"
e 7, 50 Get CnscP Picture "9999999"
e 7, 57 Say " "
e 7, 58 Get D_CP Picture "9"
e 10, 17 Say "Rango Final:"
e 10, 45 Get ProductoS Picture "9"
e 10, 46 Get MonedaS Picture "99"
e 10, 48 Get DenominaS Picture "9"
e 10, 50 Get CnscS Picture "9999999"
e 10, 57 Say " "
e 10, 58 Get D_CS Picture "9"
e 14, 3 SAY "Observaciones:"
e 16, 8 GET CAUSA1
e 17, 8 GET CAUSA2
Read
Esta_Bien = ".F."
Clv_Stat =
D_CT =
Do Calc_D_C With ProductoP + MonedaP + DenominaP + CnscP, D_CT
If D_CT <> D_CP
  Do Msg_Err With Dm_DCI
  Do Deseo With Esta_Bien
  Return
EndIf
Do Calc_D_C With ProductoS + MonedaS + DenominaS + CnscS, D_CT
If D_CT <> D_CS
  Do Msg_Err With Dm_DCI
  Do Deseo With Esta_Bien
  Return
EndIf
Do Mantr_No With ProductoS + MonedaS + DenominaS + CnscS + D_CS, Clv_Stat, "Msg_Err", Esta_Bien

```

```

If .Not. Esta_Bien
  Do Deseo With Esta_Bien
  Return
EndIf
Mo1 = ProductoP + MonedaP + DenominaP
Mo2 = ProductoS + MonedaS + DenominaS
Select Dstr
If (Mo1 <> Mo2) .or. Val(Casc_Pin) - Cant_Ini + 1 > Val(CascP) .or. CascS < CascP
  Do Msg_Err With Ds_Enlva
  Do Deseo With Esta_Bien
  Return
EndIf
If Clv_Stat <> "SRIP" .and. Clv_Stat <> "SLOC" .Or. Val(Casc_Pin) - Existencia + 1 > Val(CascP)
  Do Msg_Err With Ds_Stlv
  Do Deseo With Esta_Bien
  Return
EndIf
Casc_PinL = Val(Casc_Pin)
Tiraje_L = Val(Casc_Pin) - Val(CascS)
Exist_L = Tiraje_L
Casc_PinA = Val(CascS)
Tiraje_A = Val(CascS) - Val(CascP) + 1
Exist_A = Codigo
Casc_PinF = Val(CascP) - 1
Tiraje_F = Val(CascP) - (Val(Casc_Pin) - Cant_Ini) - 1
Exist_F = Tiraje_F - Cant_Ini + Existencia
Clave_Agt = Clave_Ag
FchS = Fch_Ship
FchP = Fch_Sloc
Do Zerol with Casc_PinA, 1
Select Ciclo_Vi
Do Si_o_No With Esta_Bien
If .not. Esta_Bien
  Return
EndIf
Append Blank
Replace Tipo_Prod With ProductoP + MonedaP + DenominaP, Casc_No with Casc_PinA, Fch_let with Date(),
  Clve_Stat with "DS", Causa_StI with Causal, Causa_StL with Causal
Select Dstr
Replace Casc_Pin with Casc_PinA, Cant_Ini with Tiraje_A, Existencia with Exist_A
If Tiraje_F > 0
  Do Zerol with Casc_PinF, 1
  Append Blank
  Replace Producto with ProductoS
  Replace Moneda with MonedaS, DenominaC with DenominaS, Casc_Pin with Casc_PinF, Fch_Ship with FchS,
    Cant_Ini with Tiraje_F, Existencia with Exist_F, Fch_Sloc with FchP, Clave_Ag with Clave_Agt
EndIf
If Tiraje_L > 0
  Do Zerol with Casc_PinL, 1
  Append Blank
  Replace Producto with ProductoS
  Replace Moneda with MonedaS, DenominaC with DenominaS, Casc_Pin with Casc_PinL, Fch_Ship with FchS,
    Cant_Ini with Tiraje_L, Existencia with Exist_L, Fch_Sloc with FchP, Clave_Ag with Clave_Agt
EndIf
Set Order to 0
Set Filter to
Sort on Casc_Pin to Temporal
Use
Erase Dstr.dbf
Rename Temporal.Dbfc to Dstr.Dbfc
Use Dstr
Index On Clave_Ag to Agot_Ds
Esta_Bien = .F.
Return AA Pin | DSTB |

```

Subrutina de Manejo de Stop Payments

Stp_Pay

```

-----
Parameters Titulo, Stat_Ant, Nv_Stat
Restore From St_Pabla Additive
Do Margena With "Prescripciones de Pago", Titulo
Store "0" to ProductoF
Store "00" to MonedaF
Store "0" to Denominat
Store "0000000" to Consecutivo
Store "0" to D_CT
Store 0 To Pesos
@ 6, 3 Say "Numero de la Orden de Pago:"
@ 6, 45 Get ProductoF Picture "9"
@ 6, 46 Get MonedaF Picture "99"
@ 6, 48 Get Denominat Picture "9"
@ 6, 50 Get Consecutivo Picture "9999999"
@ 6, 57 Say "-"
@ 6, 58 Get D_CT Picture "9"
Read
Clv_Stat = " "
Esta_Bien = .F.
Do Bnstr_No With ProductoF + MonedaF + Denominat + Consecutivo + D_CT;
, Clv_Stat, "Msg_Err", Esta_Bien
If .Not. Esta_Bien
Do Deseo With Esta_Bien
Return
EndIf
Select No
If At(Clv_Stat + "+", Stat_Ant) = 0
Do Msg_Err With Dr_Stlrv
Do Deseo With Esta_Bien
Return
EndIf
@ 6, 3 Say "Numero de la Orden de Pago:"
@ 6, 3 Say "Clave del Agente:"
@ 6, 23 Say "Dstr->Clave_Ag"
@ 6, 42 Say "Tipo de Cambio: $"
@ 6, 67 Say "Mo->Tipo_Camb" Picture "99,999.99"
@ 10, 3 Say "Fecha de Venta:"
@ 10, 23 Say "Mo->Fch_Vta"
@ 10, 42 Say "Importe en d'lares: $"
@ 10, 66 Say "Mo->Valor_Vta" Picture "999,999.99"
Pesos = Mo->Tipo_Camb * Mo->Valor_Vta
@ 12, 42 Say "Importe en pesos: $"
@ 12, 64 Say "Pesos" Picture "999,999,999.99"
Do Sl_o_No With Esta_Bien
If Esta_Bien
Replace Status With Nv_Stat + Status
Select Ciclo_Vl
Clv_Stat = "St_" + Clv_Stat
If At(Clv_Stat, "St_Sold")
Set Filter to Tipo_Prod = ProductoF + MonedaF + Denominat
Seek Consecutivo
If Hof()
Do Incons With "STP_PAY1"
EndIf
Replace Pch_2st With Date(), Clve_Stat with Nv_Stat + Clve_Stat
Else
Append Blank
Replace Tipo_Prod With ProductoF + MonedaF + Denominat, Conc_No with Consecutivo, Pch_1st with Date(),
Clve_Stat with Nv_Stat +
EndIf
@ 14, 3 Say "Observaciones:"
@ 16, 8 Get Causa_St1
@ 17, 8 Get Causa_St2
Read
Select ASel_Ag && Ajuste al Acumulado de Ventas del Agente.
Use Agentes index Agentes
Seek Dstr->Clave_Ag
If .Not. Hof()
Replace Acus_Vt with Acus_Vt - (Mo->Valor_Vta + Mo->Coais_IPS)
Else
Do Msg_Err With Dn_AgNF
EndIf
Use
EndIf
Do Deseo With Esta_Bien
Return && Pin [ "STP_PAY" ]

```


Rutina de Consultas por orden de pago
C O W S

```

Do While .T. AA Etiqueta de salida
Restore from Ds Tabla Additive
No_no_1 = "0000"
No_no_2 = "0000000"
No_No = "000000000000"
D_C = "0"
Ren_St = 8
Clv_Stat = " "
Transmis = " "
Do Marques with "Consultas", "Ordenes de Pago Individuales"
@ 5, 1 SAY "Numero de la Orden de Pago:"
@ 5, 39 Get NO_NO_1 PICTURE "9999"
@ 5, 45 Get NO_NO_2 PICTURE "9999999"
Read
NO_NO = NO_NO_1 + NO_NO_2
Tipo_ProdT = Substr(No_No,1,4)
Consecutivo = Substr(No_No,5,7)
Do Calc_D_c with No_No, D_C
Select Emision
Set Filter to Tipo_ProdT = Producto + Moneda + Denominac
Do GetGE with Consecutivo
If Erf()
Do Msg_Err with Ds_NoNP
Exit
EndIf
If Producto <> Left(No_No,1)
Do Msg_Err with Ds_NoChk
Exit
EndIf
@ 5,0 Clear
@ 5, 39 Say NO_NO PICTURE "999999999999"
@ 5, 50 Say " "
@ 5, 51 Say D_C PICTURE "9"
@ 6, 1 Say "Status Fecha Comunicaci"n"
@ 7, 1 Say "-----"
Do Imp_St With "ORDP", Fch_Emigr, " ", 0
If DtoC(Fch_Prt) <> " "
Do Imp_ST With "PRNT", Fch_Prt, " ", 0
EndIf
If Existencia > 999990
Transmis = " "
If DtoC(Fch_Prt) = " "
Do Case
Case Existencia = 999999
Clv_Stat = "VDPR"
Case Existencia = 999998
Clv_Stat = "STPR"
Case Existencia = 999997
Clv_Stat = "LSPR"
Case Existencia = 999996
Clv_Stat = "BIPS"
Fch_lps = Fch_Prt
Otherwise
@ 7, 1 Clear
Do Incons With "Cons5"
EndCase
Else
Do Case
Case Existencia = 999999
Clv_Stat = "VDPR"
Case Existencia = 999998
Clv_Stat = "STPR"
Case Existencia = 999997
Clv_Stat = "LSPR"
Case Existencia = 999996
Clv_Stat = "BIPS"
Fch_lps = Fch_Prt
Otherwise
@ 7, 1 Clear
Do Incons With "Cons5"
EndCase
EndIf
IF EXISTENCIA <> 999996
Select Ciclo_Vi
Set Filter to Tipo_Prod = Tipo_ProdT
Seek Emision->Cnsc_Fin
If Erf().or.(Clv_Stat <> "EM" .and. Clv_Stat <> "SE")
Do Incons With "Cons6"
EndIf
Do Imp_St With Clv_Stat, Fch_lst, Transmis, Fch_ltr
@ 17, 1 SAY "Observaciones : "

```

```

@ 18, 10 SAY CICLO_VI->CAUSA_ST1
@ 19, 10 SAY CICLO_VI->CAUSA_ST2
Exit
Else
@ ren_st, 1 Say Clv_Stat
@ Ren_st, 7 Say emision->Fch_IPS
Exit
ENDIF
ENDIF
IF EXISTENCIA (<) 999996
If Val(Consecutivo) >= Val( Cnsc_Fin ) - Existencia + 1
Exit
ENDIF
ENDIF
ENDIF
Select Distr
Set Filter to Tipo_ProdT = Producto + Moneda + Denominac
Set order to
Do GetGE with Consecutivo
Set Order to 1
If Eof()
Do Incons With "RASTR_M01"
ENDIF
@ 5, 57 SAY "Clave Agente"
@ 5, 70 SAY "DSTR->CLAVE_AG"
Do Imp_St With "SHIP" & Fch_Ship, " ", 0
If DtoC(Fch_Sloc) <> " "
Do Imp_St With "SLOC", Fch_Sloc, " ", 0
ENDIF
If Existencia > 9990
Transmis = " "
If DtoC(Fch_Sloc) = " . . ."
Do Case
Case Existencia = 9999
Clv_Stat = "VDS"
Case Existencia = 9998
Clv_Stat = "STR"
Case Existencia = 9997
Clv_Stat = "LSS"
Case Existencia = 9996
Clv_Stat = "DIPS"
Fch_ips = Fch_Ship
Otherwise
@ 1, 1 Clear
Do Incons With "Cons5"
EndCase
Else
Do Case
Case Existencia = 9999
Clv_Stat = "VDS"
Case Existencia = 9998
Clv_Stat = "STR"
Case Existencia = 9997
Clv_Stat = "LSS"
Case Existencia = 9996
Clv_Stat = "DIPS"
Fch_ips = Fch_Sloc
Otherwise
@ 1, 1 Clear
Do Incons With "Cons5"
EndCase
ENDIF
IF EXISTENCIA (<) 9996
Select Ciclo_VI
Set Filter to Tipo_Prod = Tipo_ProdT
Seek Distr->Cnsc_Fin
If Eof() .or. (Clv_Stat <> "DS" .and. Clv_Stat <> "SD")
Do Incons With "Cons6"
ENDIF
Do Imp_St With Clv_Stat, Fch_Ist, Transmis, Fch_ITR
@ 18, 1 SAY "Observaciones:"
@ 19, 10 SAY CICLO_VI->CAUSA_ST1
@ 20, 10 SAY CICLO_VI->CAUSA_ST2
Else
@ Ren_st, 1 SAY Clv_Stat
@ Ren_st, 7 Say Distr->Fch_IPS
ENDIF
Exit
ENDIF
IF EXISTENCIA (<) 9996
If Val(Consecutivo) >= Val( Cnsc_Fin ) - Existencia + 1
Exit
ENDIF
ENDIF
ENDIF
Select No
Set Filter to Tipo_Prod = Tipo_ProdT
Seek Consecutivo
If Eof()
Do Incons With "RASTR_M02"

```

```

EndIf
@ 10, 28 TO 15, 79
@ 1, 29 Say "Tipo de cambio"
@ 2, 29 Say "Importe en D'lares:"
@ 3, 54 Say "Mo->Valor Vta Picture "9,999,999.99"
@ 4, 57 Say "Mo->Tipo Camb Picture "99,999.99"
@ 5, 29 Say "Importe en Pesos:"
@ 6, 50 Say "Mo->Valor Mo 0 Picture "9,999,999,999.99"
@ 7, 29 Say "Comisi'n IPS:"
@ 8, 47 Say "Mo->Comis IPS Picture "99,999.99"
@ 9, 29 Say "Comisi'n Agente:"
@ 10, 47 Say "Mo->Comis Agnt. Picture "99,999.99"
@ 11, 29 Say "Clave Operador:"
@ 12, 45 Say "Mo->Clv_Oper
If Status = "V"
Do Imp_St With "VOID", Pch_Vta, " ", 0
Exit
EndIf
Do Imp_St With "SOLD", Pch_Vta, " ", Pch_Trans
If Status = "I"
Do Imp_St With "VIPS", Pch_IPS, " ", 0
EndIf
If At("."+Status, "- -V -P -I ") = 0
Select Ciclo_VI
Set filter to Tipo_Prodt = Tipo_Prod
Seek Consecutivo
If Eof()
Do Incons With "RASTR_MO4"
EndIf
Cont0 = 2
Cont1 = 1
Do while Cont0 >= 1
Clv_Stat = SubStr(" SYLN LOST CNCL";
At(SubStr(Clve_Stat,Cont0,1), "SLC") * 5 - 4, 4)
Cont0 = Cont0 - 1
If Clv_Stat < " "
Pecha = "PCH " + Str(Cont1,1) + "ST"
PechaT = "PCH " + Str(Cont1,1) + "TR"
Cont1 = Cont1 + 1
Do Imp_St With Clv_Stat, APecha, " ", APechaT
EndIf
EndDo
Do Case
Case Plag_Cnfl = " "
@ 19, 29 Say "Stop Payment por Confiruar"
Case Plag_Cnfl = "A"
@ 19, 29 Say "Confirando el"
@ 19, 43 Say Pch_IRC
Case Plag_Cnfl = "R"
@ 19, 29 Say "Rechazado el"
@ 19, 43 Say Pch_IRC
EndCase
@ 20, 1 SAY "Observaciones : "
@ 21, 10 SAY CAUSA_ST1
@ 22, 10 SAY CAUSA_ST2
Select No
EndIf
If "P" = Left(Status,1)
@ 15, 28 TO 20, 79
@ 16, 29 Say "Sucursal Promotora:"
@ 17, 30 Say "Mo->Suc Prom"
@ 18, 45 Say "Sucursal Operadora:"
@ 19, 46 Say "Mo->Suc Oper"
@ 20, 29 Say "Autorizaci'n de Pago:"
@ 20, 51 Say "Mo->Aut Pago"
Do Imp_St with "PAID", Pch_Pago, "C", Pch_Ecp_Pg
EndIf
Exit
EndDo
Do Deseo With Esta Bien
Return @ Pin | COMS |

```

 Rutina de consultas
 CONSTAT

Restore from Consul Additive

Opcion =

Cn = 0

Do While Opcion (<) "0"

Clear

Opcion = " "

Do Marques With "Sistema Ordenes de Pago", "Menu de Consultas"

@ 6, 30 Say "0. Salida"
 @ 8, 30 Say "1. Impresi"n"
 @ 10, 30 Say "2. B'veda"
 @ 12, 30 Say "3. Distribuci"n"
 @ 14, 30 Say "4. Localidad de Venta"
 @ 16, 30 Say "5. Vendidos"
 @ 18, 30 Say "6. Anulados"
 @ 20, 30 Say "7. Pagados"

@ 24, 35 SAY "Tecla tu Opci"n:"

Do While At(Opcion, "01234567") = 0

@ 24, 57 Get Opcion Picture "9"

Read

EndDo

Do Case

Case Opcion = "1"

Opcion = " "

Do While Opcion (<) "0"

Opcion =

Do Marques With "Consultas", "Impresi"n"

@ 8, 30 SAY "0. Salida"
 @ 10, 30 SAY "1. Con Status Normal"
 @ 12, 30 SAY "2. En Prevenci"n de Pago"
 @ 14, 30 SAY "3. Cualquier Status"
 @ 16, 35 SAY "Tecla tu Opcion:"

Do While At(Opcion, "0123") = 0

@ 18, 57 GET OPCION PICTURE "9"

Read

EndDo

Do Case

Case Opcion = "1"

Opcion = " "

Do Marques With "Ordenes de Pago en Impresi"n", "Con Status Normal"

@ 6, 8 Say "Orden de Pago Inicial"
 @ 6, 32 Say "Orden de Pago Final"
 @ 6, 55 Say "Cantidad"
 @ 6, 68 Say "Status"

Do Consultas with "ORDP", "ORDP", "<"

Case Opcion = "2"

Opcion = " "

Do While Opcion (<) "0"

Opcion =

Do Marques With "Ordenes de Pago en Impresi"n", "Con Prevenci"n de Pago"

@ 8, 35 Say "0. Salida"
 @ 10, 35 Say "1. Anulados"
 @ 12, 35 Say "2. Perdidas"
 @ 14, 35 Say "3. Robadas"
 @ 16, 35 Say "4. Cualquier Status"

@ 21, 32 Say "Tecla tu Opci"n:"

Do While At(Opcion, "01234") = 0

@ 21, 54 Get Opcion Picture "9"

Read

EndDo

Do Case

Case Opcion = "1"

Opcion = " "

Do Marques With "Ordenes de Pago en Impresi"n", "Anulados"

@ 6, 8 Say "Orden de Pago Inicial"
 @ 6, 32 Say "Orden de Pago Final"
 @ 6, 55 Say "Cantidad"
 @ 6, 68 Say "Status"

Do Consultas with "ORDP", "VDOR", "="

Case Opcion = "2"

Opcion = " "

Do Marques With "Ordenes de Pago en Impresi"n", "Perdidas"

@ 6, 8 Say "Orden de Pago Inicial"
 @ 6, 32 Say "Orden de Pago Final"
 @ 6, 55 Say "Cantidad"
 @ 6, 68 Say "Status"

Do Consultas with "ORDP", "LSOR", "="

Case Opcion = "3"

Opcion = " "

Do Marques With "Ordenes de Pago en Impresi"n", "Robadas"

@ 6, 8 Say "Orden de Pago Inicial"
 @ 6, 32 Say "Orden de Pago Final"

```

@ 6, 35 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "ORDP", "STOR", "="
Case Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en Impresi"n,"En Preveni"n de Pago"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "ORDP", "ESOR", "="
EndCase
ENDDO
Opcion = "2"
Case Opcion = "3"
Opcion = "3"
Do Marques With "Ordenes de Pago"n,"En Impresi"n"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "ORDP", "ALOR", "="
EndCase
ENDDO
Opcion = "2"
Case Opcion = "2"
Opcion = "2"
Do While Opcion (<) "0"
Opcion = "2"
Do Marques With "Consultas","B"veda"
@ 6, 30 SAY "0. Salida"
@ 8, 30 SAY "1. Con Status Normal"
@ 10, 30 SAY "2. Con Prevensi"n de Pago"
@ 12, 30 SAY "3. En Poder de IPS"
@ 14, 30 SAY "4. Cualquier Status"
@ 18, 35 SAY "Teclas tu Opci"n:"
Do While At(Opcion,"01234") = 0
@ 18, 57 GET OPCION PICTURE "s"
Read
EndDo
Do Case
Case Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en B"veda","Con Status Normal"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "PRNT", "PRNT", "<"
Case Opcion = "2"
Opcion = "2"
Do While Opcion (<) "0"
Opcion = "2"
Do Marques With "Ordenes de Pago en B"veda","Con Prevensi"n de Pago"
@ 8, 35 Say "0. Salida"
@ 10, 35 Say "1. Anuladas"
@ 12, 35 Say "2. Perdidas"
@ 14, 35 Say "3. Robadas"
@ 16, 35 Say "4. Cualquier Status"
@ 21, 32 Say "Teclas tu Opci"n:"
Do While At(Opcion,"01234") = 0
@ 21, 54 Get Opcion Picture "s"
Read
EndDo
Do Case
Case Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en B"veda","Anuladas"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "PRNT", "VDPN", "="
Case Opcion = "2"
Opcion = "2"
Do Marques With "Ordenes de Pago en B"veda","Perdidas"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "PRNT", "LSPN", "="
Case Opcion = "3"
Opcion = "3"
Do Marques With "Ordenes de Pago en B"veda","Robadas"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"

```

```

Do Consultas with "PRNT", "STPR", "="
Case Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en B'veda", "Con Prevenci'n de Pago"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "PRNT", "ESPR", ">"
EndCase
ENDDO
Opcion = "3"
Case Opcion = "3"
Opcion = "3"
Do Marques With "Ordenes de Pago en B'veda", "En Poder de IPS"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "PRNT", "BIPS", "="
Case Opcion = "4"
Opcion = "4"
Do Marques With "Ordenes de Pago", "En B'veda"
@ 6, 8 Say "Orden de Pago Inicial"
@ 6, 32 Say "Orden de Pago Final"
@ 6, 55 Say "Cantidad"
@ 6, 68 Say "Status"
Do Consultas with "PRNT", "ALPR", ">"
EndCase
ENDDO
Opcion = "3"
Case Opcion = "3"
Opcion = "3"
Do While Opcion <> "0"
Opcion = "3"
Do Marques With "Consultas", "Distribuci'n"
@ 8, 30 SAY "0. Salida"
@ 10, 30 SAY "1. Con Status Normal"
@ 12, 30 SAY "2. Con Prevencion de Pago"
@ 14, 30 SAY "3. En Poder de IPS"
@ 16, 30 SAY "4. Cualquier Status"
@ 18, 35 SAY "Tecla tu Opcion:"
Do While At(Opcion, "01234") = 0
@ 18, 57 GET OPCION PICTURE "9"
Read
EndDo
Do Case
Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en Distribuci'n", "Con Status Normal"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SHIP", "SHIP", "<"
Case Opcion = "2"
Opcion = "2"
Do While Opcion <> "0"
Opcion = "2"
Do Marques With "Ordenes de Pago en Distribuci'n", "Con Prevenci'n de Pago"
@ 8, 35 Say "0. Salida"
@ 10, 35 Say "1. Anuladas"
@ 12, 35 Say "2. Perdidas"
@ 14, 35 Say "3. Robadas"
@ 16, 35 Say "4. Cualquier Status"
@ 21, 32 Say "Tecla tu Opcion:"
Do While At(Opcion, "01234") = 0
@ 21, 54 Get Opcion Picture "9"
Read
EndDo
Do Case
Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en Distribuci'n", "Anuladas"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SHIP", "VDSR", "="
Case Opcion = "2"
Opcion = "2"
Do Marques With "Ordenes de Pago en Distribuci'n", "Perdidas"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"

```

```

    @ 6, 46 Say "Cantidad"
    @ 6, 56 Say "Agente"
    @ 6, 64 Say "Status"
    @ 6, 73 Say "Fecha"
    Do Consultas with "SHIP", "LSSB", "="
Case Opcion = "3"
Opcion = "3"
Do Marques With "Ordenes de Pago en Distribuci"n,"Robadas"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SHIP", "STSB", "="
Case Opcion = "4"
Opcion = "4"
Do Marques With "Ordenes de Pago en Distribuci"n,"Con Prevenci"n de Pago"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SHIP", "BSSB", "="
EndCase
EWDDO
Opcion = "3"
Case Opcion = "3"
Opcion = "3"
Do Marques With "Ordenes de Pago en Distribuci"n,"En Poder de IPS"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SHIP", "DIPS", "="
Case Opcion = "4"
Opcion = "4"
Do Marques With "Ordenes de Pago"n,"En Distribuci"n"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SHIP", "ALSH", "="
EndCase
EWDDO
Case Opcion = "4"
Opcion = "4"
Do While Opcion <> "0"
Opcion = "4"
Do Marques With "Consultas","Localidad de Venta"
@ 6, 30 SAY "0. Salida"
@ 8, 30 SAY "1. Con Status Normal"
@ 10, 30 SAY "2. Con Prevencion de Pago"
@ 12, 30 SAY "3. En Poder de IPS"
@ 14, 30 SAY "4. Cualquier Status"
@ 18, 35 SAY "Teclea tu Opci"n:"
Do While At(Opcion,"01234") = 0
@ 18, 57 GET OPCION PICTURS "9"
Read
EndDo
Do Case
Case Opcion = "1"
Opcion = "1"
Do Marques With "Ordenes de Pago en Localidad de Venta","Con Status Normal"
@ 6, 2 Say "Orden de Pago Inicial"
@ 6, 25 Say "Orden de Pago Final"
@ 6, 46 Say "Cantidad"
@ 6, 56 Say "Agente"
@ 6, 64 Say "Status"
@ 6, 73 Say "Fecha"
Do Consultas with "SLOC", "SLOC", "<"
Case Opcion = "2"
Opcion = "2"
Do While Opcion <> "0"
Opcion = "2"
Do Marques With "Ordenes de Pago en Localidad de Venta","Con Prevenci"n de Pago"
@ 8, 35 Say "0. Salida"
@ 10, 35 Say "1. Anuladas"
@ 12, 35 Say "2. Perdidas"
@ 14, 35 Say "3. Robadas"
@ 16, 35 Say "4. Cualquier Status"
@ 21, 32 Say "Teclea tu Opci"n:"

```

```

Do While At(Opcion,"01234") = 0
  # 21, 54 Get Opcion Picture '9'
  Read
EndDo
Do Case
  Case Opcion = "1"
    Opcion = "1"
    Do Marques With "Ordenes de Pago en Localidad de Venta", "Anuladas"
      # 6, 2 Say "Orden de Pago Inicial"
      # 6, 25 Say "Orden de Pago Final"
      # 6, 46 Say "Cantidad"
      # 6, 56 Say "Agente"
      # 6, 64 Say "Status"
      # 6, 73 Say "Fecha"
    Do Consultas with "SLOC", "VDSL", "-="
  Case Opcion = "2"
    Opcion = "2"
    Do Marques With "Ordenes de Pago en Localidad de Venta", "Perdidas"
      # 6, 2 Say "Orden de Pago Inicial"
      # 6, 25 Say "Orden de Pago Final"
      # 6, 46 Say "Cantidad"
      # 6, 56 Say "Agente"
      # 6, 64 Say "Status"
      # 6, 73 Say "Fecha"
    Do Consultas with "SLOC", "LSSL", "-="
  Case Opcion = "3"
    Opcion = "3"
    Do Marques With "Ordenes de Pago en Localidad de Venta", "Robadas"
      # 6, 2 Say "Orden de Pago Inicial"
      # 6, 25 Say "Orden de Pago Final"
      # 6, 46 Say "Cantidad"
      # 6, 56 Say "Agente"
      # 6, 64 Say "Status"
      # 6, 73 Say "Fecha"
    Do Consultas with "SLOC", "STSL", "-="
  Case Opcion = "4"
    Opcion = "4"
    Do Marques With "Ordenes de Pago en Localidad de Venta", "Con Preveñi"n de Pago"
      # 6, 2 Say "Orden de Pago Inicial"
      # 6, 25 Say "Orden de Pago Final"
      # 6, 46 Say "Cantidad"
      # 6, 56 Say "Agente"
      # 6, 64 Say "Status"
      # 6, 73 Say "Fecha"
    Do Consultas with "SLOC", "BSSL", ">"
  EndCase
  EWDDO
  Opcion = "3"
  Case Opcion = "3"
    Opcion = "3"
    Do Marques With "Ordenes de Pago en Localidad de Venta", "En Poder de IPS"
      # 6, 2 Say "Orden de Pago Inicial"
      # 6, 25 Say "Orden de Pago Final"
      # 6, 46 Say "Cantidad"
      # 6, 56 Say "Agente"
      # 6, 64 Say "Status"
      # 6, 73 Say "Fecha"
    Do Consultas with "SLOC", "LIPS", "-="
  Case Opcion = "4"
    Opcion = "4"
    Do Marques With "Ordenes de Pago", "En Localidad de Venta"
      # 6, 2 Say "Orden de Pago Inicial"
      # 6, 25 Say "Orden de Pago Final"
      # 6, 46 Say "Cantidad"
      # 6, 56 Say "Agente"
      # 6, 64 Say "Status"
      # 6, 73 Say "Fecha"
    Do Consultas with "SLOC", "ALSL", ">"
  EndCase
  EWDDO
  Opcion = "4"
  Case Opcion = "4"
    Opcion = "4"
    Do While Opcion (>) "0"
      Opcion = "4"
      Do Marques With "Consultas", "Ordenes de Pago Vendidas"
        # 8, 30 SAY "0. Salida"
        # 10, 30 SAY "1. Con Status Normal"
        # 12, 30 SAY "2. Con Preveñi"n de Pago"
        # 14, 30 SAY "3. En Poder de IPS"
        # 16, 30 SAY "4. Cualquiera Status"
        # 18, 35 SAY "Teclas tu Opci"n:"
      Do While At(Opcion,"01234") = 0
        # 18, 57 GET OPCION PICTURE '9'
      Read
    EndDo
  Do Case
    Case Opcion = "1"

```



```

Opcion = " "
Do Marques With "Ordenes de Pago Vendidas", "Con Status Normal"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "VMD", " "
Case Opcion = "2"
Opcion = " "
Do While Opcion <> "0"
Opcion = " "
Do Marques With "Ordenes de Pago Vendidas", "Con Prevenci"n de Pago"
@ 8, 15 Say "0. Salida"
@ 10, 35 Say "1. Canceladas"
@ 12, 35 Say "2. Robadas"
@ 14, 35 Say "3. Perdidas"
@ 16, 35 Say "4. Cualquier Status"
@ 21, 32 Say "Tecla tu Opci"n:"
Do While At(Opcion, "0123") = 0
@ 21, 54 Get Opcion Picture "9"
Read
EndDo
Do Case
Case Opcion = "1"
Opcion = " "
Do Marques With "Ordenes de Pago Vendidas", "Canceladas"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "CVMD", " "
Case Opcion = "2"
Opcion = " "
Do Marques With "Ordenes de Pago Vendidas", "Robadas"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "SVMD", " "
Case Opcion = "3"
Opcion = " "
Do Marques With "Ordenes de Pago Vendidas", "Perdidas"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "LVMD", " "
Case Opcion = "4"
Opcion = " "
Do Marques With "Ordenes de Pago Vendidas", "Con Prevenci"n de Pago"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "BVMD", " "
EndCase
ENDDO
Opcion = " "
Case Opcion = "3"
Opcion = " "
Do Marques With "Ordenes de Pago", "En Poder de IPS"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "IVMD", " "
Case Opcion = "4"
Opcion = " "
Do Marques With "Ordenes de Pago", "Vendidas"
@ 6, 4 Say "Orden de Pago"
@ 6, 24 Say "Agente"
@ 6, 35 Say "Status"
@ 6, 46 Say "Fecha Venta"
@ 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "SOLD", "AVMD", " "
EndCase
ENDDO
Opcion = " "
Case Opcion = "6"
Opcion = " "
Do Marques With "Consultas", "Ordenes de Pago Anuladas"
@ 6, 4 Say "Orden de Pago"

```

```

# 6, 24 Say "Agente"
# 6, 35 Say "Status"
# 6, 46 Say "Fecha Venta"
# 6, 62 Say "Fecha Transmisi"n"
Do Consultas with "VOID", "VOID", "="
Case
  Opcion = "*"
  Do Marques With "Consultas", "Ordenes de Pago Pagadas"
  # 6, 4 Say "Orden de Pago"
  # 6, 24 Say "Agente"
  # 6, 35 Say "Status"
  # 6, 46 Say "Fecha Venta"
  # 6, 65 Say "Fecha Pago"
  Do Consultas with "PAID", "PAID", "="
EndCase
EndDo
Return      LL Fin del programa Consultas

```

 * Procedure Consultas

```

Parameters Clave1, Clave2, Signo
Salte = .P.
Clave3 = "CL_" + Clave2
Renglon = 8
Tecla = " "
Dc = " "
St = " "
If Clave1 = "ORDP" .Or. Clave1 = "PNMT"
  Select &Sel_Bais
  If Clave1 = "ORDP"
    Set Filter to Existencia &Signo &Clave3 .And. DtoC(Pch_Prt) = " . . " .And. Existencia > 0
  Else
    Set Filter to Existencia &Signo &Clave3 .And. DtoC(Pch_Prt) (<) " . . " .And. Existencia > 0
  EndIf
  Go Top
  Do While .Not. Eof()
    If Renglon = 22
      Renglon = 8
      Do Oprimel
        @ 8, 1 Clear
      EndIf
    If .Not. Salte
      If Existencia > 999995
        Cnsc_Ini = Val(Cnsc_Pin) - Cant_Ini + 1
      Else
        Cnsc_Ini = Val(Cnsc_Pin) - Existencia + 1
      EndIf
      Do Zerof With Cnsc_Ini, 7
      Do Calc_D_C with Producto + Moneda + Denominac + Cnsc_Ini, Dc
        @ Renglon, 12 Say Producto + Moneda + Denominac + Cnsc_Ini + "-" + Dc
      Do Calc_D_C with Producto + Moneda + Denominac + Cnsc_Pin, Dc
        @ Renglon, 15 Say Producto + Moneda + Denominac + Cnsc_Pin + "-" + Dc
      Do Stat.No With Clave1, Left(Clave2,1), Existencia, St
        @ Renglon, 69 Say St
      Renglon = Renglon + 1
    Else
      Salte = .P.
    EndIf
    Skip
    EndDo
Else
  If Clave1 = "SHIP" .Or. Clave1 = "SLOC"
    Select &Sel_Dstr
    If Clave1 = "SHIP"
      Set Filter to Existencia &Signo &Clave3 .And. DtoC(Pch_Sloc) = " . . " .And. Existencia > 0
    Else
      Set Filter to Existencia &Signo &Clave3 .And. DtoC(Pch_Sloc) (<) " . . " .And. Existencia > 0
    EndIf
    Go Top
    Do While .Not. Eof()
      If Renglon = 22
        Renglon = 8
        Do Oprimel
          @ 8, 1 Clear
        EndIf
      If .Not. Salte
        If Existencia > 9995
          Cnsc_Ini = Val(Cnsc_Pin) - Cant_Ini + 1
        Else
          Cnsc_Ini = Val(Cnsc_Pin) - Existencia + 1
        EndIf
        Do Zerof With Cnsc_Ini, 7
        Do Calc_D_C with Producto + Moneda + Denominac + Cnsc_Ini, Dc
          @ Renglon, 6 Say Producto + Moneda + Denominac + Cnsc_Ini + "-" + Dc
        Do Calc_D_C with Producto + Moneda + Denominac + Cnsc_Pin, Dc
          @ Renglon, 28 Say Producto + Moneda + Denominac + Cnsc_Pin + "-" + Dc
        Do Stat.No With Clave1, Left(Clave2,1), Existencia, St
          @ Renglon, 48 Say Val(Cnsc_Pin) - Val(Cnsc_Ini) + 1 Picture "9,999"
          @ Renglon, 57 Say Clave_Ag
          @ Renglon, 65 Say St
          @ Renglon, 72 Say Pch_Ship
          Renglon = Renglon + 1
        Else
          Salte = .P.
        EndIf
        Skip
        EndDo
    Else
      Select &Sel_Mo
      If Left(Clave2,1) (<) "P" .And. Left(Clave2,1) (<) "A"
        Set Filter to Left(Status,1) = Left(Clave2,1)
      EndIf
    EndIf
  EndIf
EndIf

```

```

Else
  If Left(Clave2,1) = "E"
    Set Filter to Left(Status,1) <> " " .And. Left(Status,1) <> "V" .And. Left(Status,1) <> "P";
    .And. Left(Status,1) <> "I"
  Else
    Set Filter to Left(Status,1) <> "V" .And. Left(Status,1) <> "P"
  EndIf
EndIf
Go Top
Do While .Not. Eof()
  If Renglon = 22
    Renglon = 8
    Do Oprime!
    @ 8, 1 Clear
  EndIf
  If .Not. Salte
    Do Calc_D_C with Tipo_Prod + Cnsc_No, DC
    @ Renglon, 5 Say Tipo_Prod + Cnsc_No + "-" + DC
    @ Renglon, 21 Say Clave_Ag
    @ Renglon, 36 Say Substr("SOLD STLW CWCL LOST PAID VOID VIPS",At(Left(Status,1),"SCLPVI")+5-4,4)
    @ Renglon, 48 Say Fcb_Vta
    If Left(Status,1) = "P"
      @ Renglon, 67 Say Fcb_Pago
    Else
      @ Renglon, 67 Say Fcb_Trans
    EndIf
    Renglon = Renglon + 1
  Else
    Salte = .F.
  EndIf
  Skip
EndDo
EndIf
Set Filter to
Do Oprime with Tecla
Return @@ [ Pin Consultas(Subrutina) ]

```

```
-----
Subrutina que pone Encabezados
Marques (ina)
-----
Parameters Cabeza1, Cabeza2
If Flag = "0"
  # 0, 0 Clear
EndIf
# 1, 02 Say "Hora:"
# 1, 08 Say "Time() Picture "99:99"
# 1, 22 Say "International Payment Systems, Inc."
# 1, 62 Say "Fecha:"
# 1, 65 Say "Date:"
# 2, 40-len(Cabeza1)/2 Say Cabeza1
# 3, 40-len(Cabeza2)/2 Say Cabeza2
If Flag = "1"
  # 1, 1 Say "-----"
  Flag = "0"
Else
  # 0, 0 To 4, 79 Double
EndIf
Return
#{BOP}
```